A process monitoring module based on fuzzy logic and pattern recognition

A. Devillez a,*, M. Sayed-Mouchaweh b, P. Billaudel b

a LPMM—UMR CNRS 7554, Institut Supérieur de Génie Mécanique et Productique, Université de Metz, Ile du Saulcy, 57045 Metz Cedex 01, France
b Laboratoire d'Automatique et de Microélectronique, Institut de Formation Technique Supérieure, 7, boulevard Jean Delautre, 08000 Charleville-Mezieres, France

Received 1 February 2003; accepted 1 October 2003
Available online 30 December 2003

Abstract

This article presents a plastic injection moulding monitoring module based on knowledge built on-line using feedback from production data. A fuzzy classifier was especially developed for this application. It is based on unsupervised and supervised classification methods. The role of the first one is to identify the functioning modes of the process whereas the role of the second one is to associate the state of the process to one of the identified functioning modes at the moment where a workpiece is injected. Furthermore this diagnosis module integrates an on-line learning method which allows to enrich and upgrade the initial knowledge during production. The results obtained show that the monitoring system is a solution for quality and productivity control having serious economical advantages. For example maintenance tasks can be anticipated and the size of the training set can be considerably reduced. The computing times show that the monitoring system can be used for the purpose of industrial applications without any decrease of production rate.

© 2003 Elsevier Inc. All rights reserved.

Keywords: Process monitoring; Knowledge processing; Adaptive system; On-line learning; Diagnosis; Decision making; Fuzzy clustering; Fuzzy discrimination; Approximate reasoning

*Corresponding author. Tel.: +33-387-54-70-09; fax: +33-387-31-53-66.
E-mail address: devillez@lpmm.univ-metz.fr (A. Devillez).
1. Introduction

Diagnosis in real time is a problem currently met in industrial modern societies. A lot of techniques are used to monitor the functioning of the process. For example we can cite the statistical process control (SPC), which is achieved from control maps. These methods are based on the recording of data acquired from the maintaining, production, quality or methods departments. In other words these techniques use an unorganised knowledge of the process.

Wear, failures or production circumstances cause a lot of scraps or progressive derivatives of the process, which must be detected in order to warrant quality and productivity. The works we present in this paper have for main aims the development and the adaptation of fuzzy classification methods for industrial process monitoring. Furthermore our purpose is to increase the performances of such monitoring systems by integrating a permanent training enriching the knowledge database.

Some structures exist to design monitoring systems in the field of diagnosis and industrial control. Monitoring is defined as the control of the running of a task by a human or machining [31]. A lot of works are integrated in monitoring: diagnosis, prognostic, decision making. As reason of the limits of techniques based on modelling, statistical methods or expert systems, monitoring becomes an application field of pattern recognition and classification methods [15,16,23]. The application we propose in this paper is to monitor a plastic injection moulding process from control parameters of an injection press.

2. Description of a diagnosis module

2.1. Basis structure

The basically structure proposed in [16] is described in Fig. 1. It integrates the system and its sensors, which generate the information vector required for representation and classification of data. After the pre-processing of data achieved by Principal Components Analysis method, database is constituted. It regroups the knowledge about the process. Generally the first learning is realised by experts or by a fuzzy classification method. It consists in clustering the elements belonging to the same classes and in identifying these classes. This step includes also the computing the mathematical representation required by the discrimination step. This last one includes the decision step and it must be able to follow the evolution of initial classes that is to say the evolution of functioning modes during the production. This process is called continuous or incremental learning. Another updating of initial classes consists in changing the number of classes if the number of functioning modes of the process varies during the production. It is evident that our monitoring module must integrate these two proceedings in
order to be completely adaptive. Classes $C_i$ correspond to the functioning modes of the process whereas classes denoted $C_{\text{amb}}$ and $C_{\text{mem}}$ are virtual classes, which includes respectively points rejected for ambiguity and points rejected for low membership. These reject proceedings will be detailed in a next part of this paper.

2.2. Adaptation of the structure

The structure presented in Fig. 1 remains a general structure, which shows the methodology used for developing a monitoring system. It must be updated according to the classification and discrimination methods used. The principle of our module is presented in Fig. 2. It includes three main steps: pre-processing of the data, learning and discrimination which has for consequence the final decision of classification. The system presented includes:

- a continuous learning step integrating the upgrading of the database at each time that a point is affected to one of the classes identified by the initial learning,
- the upgrading of the general learning when classes of ambiguity and belonging rejects have an effective higher than a defined threshold.

These steps are not entirely automated. We choose to adopt a partially supervised solution where the driver can master the module when the learning or the control parameters must be updated and changed. However when the process is in a known functioning state identified during the initial learning, the intervention of the operator is not required.

2.3. Pre-processing of the data

Our module integrates the pre-processing of the data, which concerns the learning data and the data to classify. In some books this pre-processing is not included in the diagnosis module because it uses essentially some statistical
Fig. 2. Structure of monitoring system.
methods of data analysis [9]. Our idea is to integrate this step in the diagnosis module for two reasons:

- The main aim of the pre-processing of data is to extract the most significant attributes to define the classification space.
- When two classes draw them nearer during the functioning of the module up that they are merged because of their deformation, the upgrading of the training set is not sufficient. It becomes essential to select new attributes in order to redefine the classification space.

These two purposes show that the pre-processing of the data is very difficult to dissociate of the decision system. In our case it is performed by statistical methods such as Principal Components Analysis [26].

2.4. Learning

The second step considered in the module and presented in Fig. 2, is the learning step. Its main mission is to create the training classes in the set of points after the building of the classification space. Furthermore it must compute a reliable database including all the information required by the discrimination algorithm.

The Unsupervised Fuzzy Graph Clustering (UFGC) [11] method was chosen for our application because the number of classes is not known a priori. This method is briefly described in the following section. The aim of the UFGC method is to compute:

- a fuzzy partition into \( c \) classes from a set of \( n \) points in a space of any size,
- an over-evaluated number of subclasses for each class required by the discrimination method used.

The determination of the number of subclasses can be a step of the learning. The integration of an ambiguity reject in the learning can make the database reliable. Indeed this reject proceeding allows decreasing the affectation faults. The remaining points are affected to the classes by the maximum membership rule [25]. The result obtained is a set of \( c \) clusters of points for which the belonging is certain. Fig. 14 presents the set of points after the complete learning step for our case. The ambiguous points have been suppressed in order to limit the discrimination faults. Only the most certain points are conserved after the training step.

2.5. Recognition

The third step we have considered is the recognition step. This step is achieved by the Fuzzy Pattern Matching Mutlidensity method [12] developed
for classes of any shape. It is a possibilistic method that presents good discrimination times with a high recognition rate. Furthermore the integration of ambiguity and membership reject proceedings in the algorithm is very easy. Consequently this method has a good ability to be applied to diagnosis problems such as the one we want to raise in this paper.

3. Presentation of the studied industrial process

The press used for this study is a Sandretto machine presented in Fig. 3. Its power closing is 60 tons and the maximum injectable volume is 192 cm$^3$.

The screw diameter is 30 mm and the mould closing is realised by a toggle joint. We have used a mould for assay spoons. It has two impressions. The schema of the impression block is shown in Fig. 4. The material used for the
study is colorless shock polystyrene. Its reference is Lacqrene 1810 from Elf Atochem.

4. Pre-processing of the data

4.1. Principle of the Principal Components Analysis (PCA)

The Principal Components Analysis [22,26,28] is applied to find the correlations between parameters and to determine the most significant parameters on the quality of the workpieces. The PCA was first introduced by Pearson [26] to fit systems of points in space and first used by Hotelling [21] to analyse correlated data sets. PCA has since been used extensively in different fields of sciences and engineering. It consists to find a projection of the space defined by the set of parameters on a space of a smaller size. This new space must minimise the distortion of the distances between the points, axis defining this space are the principal axis. First, all the data are centred and reduced in order to obtain a dimensionless form for each parameter. Then, the principal axis are determined according to the eigenvectors $u_i$ associated to the eigenvalues $\lambda_i$ of the matrix $X^T \cdot X$. The matrix $X$ is the data matrix.

The PCA defines an inertia criterion or a percentage of inertia by:

$$\tau_i = \frac{\lambda_i}{\sum_{j=1}^{a} \lambda_j}.$$

The first principal axis corresponds to the highest inertia value and the following ones are determined according to the decreasing values of inertia. Then, with the criterion of inertia, it is possible to evaluate if an axis is significant or not. In the following, we suppose that two principal axes are more significant than the other ones and the projection space is of dimension 2.

For each parameter a linear correlation coefficient is calculated with respect to the two principal components associated to the two principal axes [26]. The two obtained values define a vector for each parameter in the factorial plane. As the data are centred and reduced, the vectors are included in a circle of radius 1 centred on the origin, called correlation circle. An example of such a circle is presented in Fig. 5. Points correspond to the parameters. When they are near to the circle, they are significant. The variable vectors are vectors linking the origin to the variable. The scalar product of two vectors is the correlation degree between the corresponding parameters. In our case variables $D$ and $C$ are linearly correlated, variables $E$ and $C$ are inversely correlated and variables $B$ is not significant. To finish variables $E$ and $A$, the vectors of which define an angle of $90^\circ$ are not correlated. This analysis allows one to select the data in order to find the most significant parameters of the process.
4.2. Application to plastics injection moulding process

4.2.1. Acquisition of parameters

The parameters acquisition is done automatically by using a computer. This one is connected to the press visual display unit by a RS 232 plug. The acquisition is done for each cycle in automatic mode. All the controlling parameters described in Table 1 are collected.

4.2.2. Description of the different scraps

In order to create different classes, we searched to produce different types of scraps:

Table 1
Control parameters of the plastic injection moulding process

<table>
<thead>
<tr>
<th>Parameter code</th>
<th>Parameter description</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>Plastication time</td>
</tr>
<tr>
<td>b</td>
<td>Injection time</td>
</tr>
<tr>
<td>c</td>
<td>Holding time</td>
</tr>
<tr>
<td>d</td>
<td>Cooling time</td>
</tr>
<tr>
<td>e</td>
<td>Commutation pressure</td>
</tr>
<tr>
<td>f</td>
<td>Injection pressure</td>
</tr>
<tr>
<td>g</td>
<td>Holding pressure</td>
</tr>
<tr>
<td>h</td>
<td>Commutation position</td>
</tr>
<tr>
<td>i</td>
<td>Final position of cushion</td>
</tr>
<tr>
<td>j</td>
<td>Plastication position</td>
</tr>
<tr>
<td>k</td>
<td>Nozzle temperature</td>
</tr>
<tr>
<td>l</td>
<td>Barrel temperature 1</td>
</tr>
<tr>
<td>m</td>
<td>Barrel temperature 2</td>
</tr>
<tr>
<td>n</td>
<td>Barrel temperature 3</td>
</tr>
</tbody>
</table>

Fig. 5. Example of correlation circle obtained by the PCA method.
Incomplete: piece that presents a lack of plastic as it is shown in Fig. 6a. This type of scraps is obtained by reducing injection pressure, holding pressure and plastication position.

Flash: piece that presents an excess of plastic as it is showed in Fig. 6b. This scrap is obtained by increasing the plastication position and the injection and holding pressures.

Shrink holes: piece that presents aspect defects like shown in Fig. 6c. These faults are characterised by a bad evenness. We obtain these pieces by decreasing injection and holding pressures.

During this study we have realised 640 cycles the repartition of which is presented in Table 2.

4.2.3. Reducing of the number of attributes

It is not necessary to use all the process control parameters in order to profile the general effect of the experiments. In fact, some of them are correlated and others have no effect on the quality of produced pieces. The correlation circle obtained for the plastic injection moulding process is presented in Fig. 7. On this circle it is observed that temperatures are correlated between themselves. The holding pressure is correlated with the injection pressure, which is conforming to the experience on this type of injection press. Indeed it is important to maintain a constant difference between these two parameters during the fabrication. The final position of cushion presents an invert correlation with the commutation pressure. The unsignificant parameters is the plastication time, which is not used to define the classification space. We have chosen to retain the most significant parameters to define this space. Among the four temperatures, barrel temperature in zone 3 was retained. The final
position of cushion constitutes the second axis because it is significant and uncorrelated with the temperatures. To finish we decided to retain the holding time because it is not correlated with the other parameters. The obtained set of points in this space of size 3 is presented in Fig. 8. Five classes are obtained whereas only four types of bad pieces have been created. One of the categories of pieces is represented by two classes. The aim of the learning step is to identify the classes obtained for the process.
5. Identification of the classes by Unsupervised Fuzzy Graph Clustering Method

5.1. Shape of classes

We observe that classes have non-elliptic shapes. Two ones of them can be considered as tilted whereas the three other ones are not convex. Classical fuzzy clustering methods are not able to compute a partition for this type of set of points. That is why we chose the Unsupervised Fuzzy Graph Clustering Method [10,11] is retained for the learning step.

5.2. Division of the set of points into subclasses

The first stage of this algorithm consists in using the Fuzzy C-Means algorithm [3] to divide the set into \( c' \) subclasses. This principle has been used by Frigui and Krishnapuram in the URCP algorithm [17]. The result of this algorithm is a fuzzy membership matrix \( U' \). Element \( u'_{kj} \) represents the membership degree of sample \( x_j \) to subclass \( SC_k \). Each subclass must belong to one real class only.

Therefore the number of subclasses must be much greater than the number of real classes. In our example presented in Fig. 9, \( c' \) must be higher than 5. It is possible to use some heuristic criteria to choose \( c' \). Classically we use the fuzzy hypervolume, compactness and separability criteria to that effect [6]. In our case a partition with 14 subclasses is obtained, as illustrated in Fig. 9. It is verified that each subclass belongs to one real class only.

![Barrel temperature 3 (°C)](image)

Fig. 9. Division of the set into 14 subclasses. All points marked by a similar letter belong to the same subclass.
5.3. Similarity degrees between subclasses

The notion of hard neighbourhood between subclasses is studied in [7]. The Unsupervised Graph Clustering algorithm presents a major restriction: the initial set of points must not present ambiguous points otherwise some fusions that should not exist between subclasses will occur. The clustering algorithm based on a hard graph endures a lack of hardiness that can greatly decrease its performances. That is why we have introduced the concept of fuzzy neighbourhood. The neighbourhood or proximity between two subclasses can be quantified by a coefficient. The one we are using is the similarity degree defined by Frigui and Krishnapuram [17]:

\[ n_{kl} = 1 - \frac{\sum_{x_i \in SC_k \text{ or } x_j \in SC_l} |u_{ik}' - u_{jl}'|}{\sum_{x_i \in SC_k} u_{ik}' + \sum_{x_i \in SC_l} u_{jl}'}. \]

A neighbourhood matrix \( N \), the size of which is \( c' \times c' \) is also computed. Element \( n_{kl} \) corresponds to the proximity value between subclass \( SC_k \) and subclass \( SC_l \). Nearer the value of \( n_{kl} \) is to 1, the closer the subclasses are to one another. Terms on the diagonal are all equal to 1 because each subclass is a neighbour of itself. Furthermore the matrix \( N \) is symmetrical.

The obtained matrix \( N \), which is calculated by using the fuzzy similarity degree for our example, is the following:

\[
N = 10^{-2}
\begin{pmatrix}
100 & 1.2 & 1.0 & 9.02 & 0.08 & 0.89 & 3.00 & 0.19 & 0.53 & 0.2 & 0.13 & 0.07 & 0.26 & 0.22 \\
1.2 & 100 & 0.15 & 0.20 & 0.29 & 0.24 & 0.15 & 0.07 & 0.13 & 0.56 & 0.25 & 6.71 & 0.07 & 3.78 \\
1.0 & 0.15 & 100 & 0.17 & 1.99 & 0.35 & 0.12 & 0.08 & 0.16 & 1.35 & 9.22 & 0.09 & 0.08 & 0.18 \\
9.02 & 0.20 & 0.17 & 100 & 0.21 & 0.96 & 10.32 & 0.24 & 0.52 & 0.37 & 0.24 & 0.13 & 0.30 & 0.30 \\
0.08 & 0.29 & 1.99 & 0.21 & 100 & 0.48 & 0.12 & 0.08 & 0.18 & 16.47 & 11.02 & 0.14 & 0.09 & 0.34 \\
0.89 & 0.24 & 0.35 & 0.96 & 0.48 & 100 & 0.78 & 3.61 & 9.62 & 0.43 & 0.44 & 0.18 & 5.06 & 0.20 \\
3.00 & 0.15 & 0.12 & 10.32 & 0.12 & 0.78 & 100 & 0.17 & 0.45 & 0.27 & 0.17 & 0.09 & 0.22 & 0.25 \\
0.19 & 0.07 & 0.08 & 0.24 & 0.08 & 3.61 & 0.17 & 100 & 3.33 & 0.10 & 0.09 & 0.05 & 2.73 & 0.09 \\
0.53 & 0.13 & 0.16 & 0.52 & 0.18 & 9.62 & 0.45 & 3.33 & 100 & 0.18 & 0.18 & 0.10 & 18.59 & 0.13 \\
0.20 & 0.56 & 1.35 & 0.37 & 16.47 & 0.43 & 0.27 & 0.10 & 0.18 & 100 & 4.39 & 0.27 & 0.10 & 0.45 \\
0.13 & 0.25 & 9.92 & 0.24 & 11.02 & 0.44 & 0.17 & 0.09 & 0.18 & 4.39 & 100 & 0.14 & 0.10 & 0.27 \\
0.07 & 6.71 & 0.09 & 0.13 & 0.14 & 0.18 & 0.09 & 0.05 & 0.10 & 0.27 & 0.14 & 100 & 0.06 & 13.98 \\
0.26 & 0.07 & 0.08 & 0.30 & 0.09 & 5.06 & 0.22 & 2.73 & 18.59 & 0.10 & 0.10 & 0.06 & 100 & 0.09 \\
0.22 & 3.78 & 0.18 & 0.30 & 0.34 & 0.20 & 0.25 & 0.09 & 0.13 & 0.45 & 0.27 & 13.98 & 0.09 & 100
\end{pmatrix}
\]

5.4. Construction of the fuzzy proximity graph and of the dendogram

5.4.1. Fuzzy proximity graph and reduced graph

The neighbourhood matrix \( N \) defines a fuzzy proximity graph in which the vertices represent the subclasses and the arcs represent the links. The arcs are graduated by the proximity degrees. The graph corresponding to the neighbourhood matrix of our example includes 14 vertices and 91 arcs, knowing that the arcs linking a subclass to itself are not represented and that matrix \( N \) is
symmetrical. The representation of a graph of this size is not comfortable. That is why it is possible to associate a reduced graph, which is presented in Fig. 10. It reflects structures existing in the set of points. Calculations are based on matrix $N$, consequently they take account of the complete graph. Furthermore it is possible to define the remoteness matrix $F$. It is the complementary matrix-to-matrix $N$ where $f_{kl} = 1 - n_{kl}$ whatever $k$ and $l$. The higher the value of $f_{kl}$ is, the more distant subclasses $SC_k$ and $SC_l$ are from each other. The graph and the graduated hierarchy associated to matrix $F$ are shown in Figs. 10 and 11, respectively.

![Reduced fuzzy proximity graph for our example.](image1)

![Dendogram for our example and for level of cutting $\beta = 0.94$.](image2)
5.4.2. **Dendogram**

We call dendogram or graduated hierarchy a tree the leaves of which are samples subject to classification. The hierarchy is graduated if any part $h$ is associated to a numerical value $v(h) \geq 0$ compatible with the following relation:

$$\text{If } h \subset h' \text{ then } v(h) < v(h').$$

A hierarchy is associated to the graph shown in Fig. 10. It is graduated according to the remoteness values of matrix $F$. Two subclasses or two sets of subclasses are associated at each level. This operation is repeated until all subclasses are merged into one class only. For our example we have obtained the dendogram shown in Fig. 11.

5.5. **Defuzzification of the proximity graph**

5.5.1. **Level of the cut of the graduated hierarchy**

To recover real classes it is absolutely necessary to cut the dendogram or the graduated hierarchy at a level $\beta$. For example the level $\beta = 0.94$ gives the set of subclasses shown in Fig. 12. The final partition depends on the level of the cut.

5.5.2. **Existence of a fuzzy order relation**

The neighbourhood matrix $N$ defines a fuzzy relation [8] that expresses the idea of neighbourhood between subclasses. It can be decomposed into a series of relations of level $\alpha$. If $N$ is a fuzzy order relation where $n_{kl} \in [0,1]$, the relation of level $\alpha$ which is associated to $N$ is defined by $N^\alpha$ such as:

$$n^\alpha_{kl} = 1 \text{ if } n_{kl} \geq \alpha \text{ and } n^\alpha_{kl} < \alpha.$$
This decomposition entails the defuzzification of the proximity graph into a hard graph, which determines the structure of the set of points. The remaining arcs are no longer graduated. Two subclasses are then considered as neighbouring and consequently merged if an arc exists between the two vertices that correspond to them. For the level of cut $\alpha = 0.06$, we have obtained the hard neighbourhood graph shown in Fig. 12. We can note that the connected components of this graph correspond to the set of subclasses obtained by the cut of the dendogram. Furthermore value $\alpha$ is complementary to value $\beta$. Consequently the relation of level $\alpha$ gives the same result than the cut of level $\beta$ into the dendogram. In our exercises we have used the fuzzy relation defined by matrix $N$ to recover the structure of the set of points.

5.6. Fusion of subclasses

The last stage of the algorithm consists in searching for the connected components [19] of the hard proximity graph or sets of subclasses in the dendogram. For each point we have at our disposal a membership degree for each subclass. It is necessary to achieve a fusion in order to obtain a membership degree for each real class. This fusion is performed by a limited sum [23]. When a number $s_i$ of subclasses $SC_k$ exists within a class $C_i$, the membership function of a point $x$ to this class is defined by:

$$u_i(x) = \min \left[ 1, \sum_{k=1}^{s_i} u_{SC_k}(x) \right].$$

The new membership matrix is calculated from the matrix $U''$ resulting from the Fuzzy C-Means algorithm. For each point the sum of membership degrees to each subclass is equal to 1. That is why a simple sum is achieved between lines corresponding to the subclasses which must be merged. We obtain a membership matrix $U$ the size of which is $c \times n$ with $c$ being the number of connected components and $n$ the number of points. Element $u_{ij}$ represents the membership degree of point $x_j$ to the new class $C_i$. The obtained membership matrix depends on the chosen level of the cut.

5.7. Research on the optimal cut

Few criteria can help us to determine the cut to be made into the graduated hierarchies. Those proposed in [1,2,4,18,20,24,27,32,33] give good results with classes of spherical or elliptical shapes. With classes of complex shapes there exists no adequate criteria to find the optimal partition. The proposed criterion uses the principle of the compactness criterion defined in [33]. The global compactness of a partition is defined by:
\[
\text{co}_{\text{gl}} = \frac{1}{n} \sum_{i=1}^{c} \sum_{j=1}^{n} (u_{ij} \cdot d_{ji})^2.
\]

d_{ji} represents the distance from point \(x_j\) to the nearest centroid of class \(C_i\) because of the multiprototype approach used in the UFGC algorithm. \(\text{co}_{\text{gl}}\) allows one to measure the global compactness of the set of points.

According to the same model, we define a compactness measure for each class \(C_i\):

\[
\text{co}_{C_i} = \frac{\sum_{j=1}^{n} (u_{ij} \cdot d_{ji})^2}{|C_i|}
\]

where \(|C_i|\) is the fuzzy cardinal defined by: \(|C_i| = \sum_{j=1}^{n} u_{ij}\).

The average compactness of the created classes into the set of points is defined by the following expression:

\[
\text{co}_{\text{av}} = \frac{1}{c} \sum_{i=1}^{c} \text{co}_{C_i} = \frac{1}{c} \sum_{i=1}^{c} \frac{\sum_{j=1}^{n} (u_{ij} \cdot d_{ji})^2}{\sum_{j=1}^{n} u_{ij}}.
\]

The global compactness defined by Frigui and Krishnapuram represents the average distribution of the points around the centroids of all the classes according to the membership levels. The considered cardinal using for calculating the average value is the total number of points \(n\). The compactness we have defined for one class is based on the same model. It represents the average distribution of the points around the centroids of one class. The considered cardinal is now the fuzzy cardinal \(|C_i|\) of class \(C_i\). Then the average compactness is computed from the compactness of all the classes in order to obtain a value, which translates the distribution of the points around the centroids of the classes. These definitions are equivalent on condition that the partitioning is optimal that is to say when the computed clusters correspond to the real classes. The definition of our criteria is based on this hypothesis of equivalence between average and global compactness when the optimal partition into \(c^*\) classes is attained. In this case its ratio tends towards 1 and we propose the minimisation of the following criterion for determining the number of classes existing in the set of points:

\[
K_c = \left| 1 - \frac{\text{co}_{\text{av}}}{\text{co}_{\text{gl}}} \right|.
\]

It is shown that this criterion functions for the case where all the classes include the same number of points [11]. The extension to more general problems is the object of some experimental works.
Table 3 illustrates the results we obtained for our example. We can note that the minimum is obtained for five classes. The corresponding partition is presented in Fig. 13. The computed classes are similar to the real classes.

5.8. Optimisation of the training set

To obtain a reliable training set, an ambiguity reject proceeding is applied to the partition computed by the clustering method. When the difference between the two maximum membership possibilities of a point is lower than the ambiguity threshold $T_{\text{amb}}$, the point is suppressed of the training set. The ambiguity reject rule is:

$$\text{if } \max_{i=1,c} \left( \mu_i - \max_{i=1,c-1} \left( \mu_i - \max_{i=1,c} \right) \right) \leq T_{\text{amb}}$$

then $x$ is suppressed of the training set.

The obtained training set, for an ambiguity threshold $T_{\text{amb}} = 0.4$, is presented in Fig. 14.

Table 3
Values of the criterion $K_c$ as a function of the number of classes

<table>
<thead>
<tr>
<th>$x \cdot 10^{-2}$</th>
<th>1</th>
<th>4</th>
<th>7</th>
<th>9.5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_c \cdot 10^{-2}$</td>
<td>12.9</td>
<td><strong>11.7</strong></td>
<td>21.3</td>
<td>38.7</td>
</tr>
<tr>
<td>Number of clusters</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

Fig. 13. Graphic results of classification. Points marked by a similar number belong to the same class.
6. Recognition module

6.1. Fuzzy pattern matching multiprototype

A comparative study [5] between the main fuzzy classification methods shown that the Fuzzy Pattern Matching (FPM) proposed in [14] gives the best results for real times applications when the classes have not complex shapes. To correct this drawback, the Fuzzy Pattern Matching Multiprototype (FPMM) was developed.

6.1.1. Division of each class and calculation of the possibility densities

Each class \(C_i\), including \(n_i\) learning points, is divided into \(s_i\) subclasses \(C_{ij}\) by the Fuzzy C-Mean algorithm. As previously, it is very important that each subclass belongs to only one real class. The learning consists in calculating the possibility densities [34] for each subclass and each attribute. First, we determine the probability histogram. The probability of a bar is the number of points situated in this bar divided by the total number of points of the subclass.

To simplify the algorithm, the number of bars \(h\) is the same for all the histograms. The probability values \(\{p(y_i)|i = 1,\ldots, h\}\) of the centres of intervals \(y_i\) are sorted in the decreasing order \(p(y_1) \geq p(y_2) \geq p(y_3) \geq p(y_4) \geq \cdots \geq p(y_h)\). The possibility distribution \(\{\pi(y_i)|i = 1,\ldots, h\}\) is deduced from the probability distribution by the bijective transformation introduced by Dubois and Prade [13]:

![Fig. 14. Reliable training set for an ambiguity threshold \(T_{amb} = 0.4\).](image-url)
\[ \pi(y_i) = \sum_{j=1}^{h} \min[p(y_i), p(y_j)] = i \cdot p(y_i) + \sum_{j=i+1}^{h} p(y_j). \]

Finally, the possibility densities are determined by linear linking. For our application we have decomposed the set of points into 14 subclasses. Fig. 15 shows the possibility densities computed for the three attributes.

6.1.2. Recognition phase

For a new point \( x \) appearing into the classification space, a possibility value \( \pi_{ij}^k \) is calculated by linear interpolation for subclass \( j \) of class \( i \) and attribute \( k \). A first fusion of these values, by a minimum operator, gives a membership possibility \( \eta_i^j \) of the point to subclass \( C_{ij} \):

\[ \eta_i^j = \min_{k=1}^{a} (\pi_{ij}^k). \]

Fig. 15. Possibility densities for each subclass and each attribute.
It is necessary to achieve a second fusion to obtain a membership possibility \( \mu_i \) of the point for each class. We have choose to use a limited sum which allows to have a minimum possibility value without crossing the superior limit of 1:

\[
\mu_i = \min \left( 1, \sum_{j=1}^{s_i} \eta_{ij} \right).
\]

Moreover, contrary to the maximum, this operator smoothes the membership surface in suppressing the peaks that may occur in the reconstructed classes, as shown in Fig. 16 (for two fuzzy sets.)

6.1.3. Classification and reject rules

Generally, the new point is classified in the class for which the membership possibility is maximized. In some cases this point is rejected. It does not mean that it is suppressed, but it means that it is affected to one of the two virtual

![Fig. 16. Union two fuzzy sets (a) by a maximum (b) and a limited sum (c).]
classes called ambiguity reject and membership reject classes and noted respectively \( C_{\text{amb}} \) and \( C_{\text{mem}} \).

When the difference between the two maximum membership possibilities of a new point is lower than the ambiguity threshold \( T_{\text{amb}} \), the point is situated approximately at equal distance of the two classes. It is impossible to affect it to one of these classes. The ambiguity reject rule is:

\[
\text{if } \left\{ \max_{i=1,c} (\mu_i) - \max_{i=1,c-1} \left[ \mu_i - \max_{i=1,c} (\mu_i) \right] \right\} \leq T_{\text{amb}} \text{ then } x \in C_{\text{amb}}.
\]

When the membership possibility of a new point \( x \) is lower than the membership threshold \( T_{\text{mem}} \), the point is far from all the classes. It is affected to the membership reject class. The membership reject rule is:

\[
\text{if } \max_{i=1,c} (\mu_i) \leq T_{\text{mem}} \text{ then } x \in C_{\text{mem}}.
\]

### 6.2. Incremental learning

Each new classified point adds information about the knowledge of the studied process: evolution of the shape of the classes, occurrence of new classes corresponding to new operating states, fusion of existing classes. This information improves the performances of the classification algorithm. For this reason, it is necessary to add the new points in the learning set and to recalculate new histograms and new densities of possibility in using the transformation of Dubois and Prade. This solution is costly according to the calculation time because this time rises with the increasing of the number of points. The recursive method is faster to update the densities of possibility [29,30]. Its calculation time is constant with the number of points.

We suppose that the new point is affected to a class containing \( N_i \) points, so the new number of points is \( N_i + 1 \). For an attribute, \( P \) and \( P' \) are respectively the matrixes of the latest and of the new probabilities:

\[
P = \begin{pmatrix} p_1 & p_2 & \cdots & p_h \end{pmatrix},
\]
\[
P' = \begin{pmatrix} p'_1 & p'_2 & \cdots & p'_h \end{pmatrix}.
\]

The recursive equation giving the new probabilities is:

\[
P' = \text{NPR} \cdot P + \text{EN}
\]

with

\[
\text{NPR} = \frac{N_i}{N_i + 1} \quad \text{and} \quad \text{EN} = \begin{pmatrix} 0 & \cdots & \frac{1}{N_i + 1} & 0 & \cdots \end{pmatrix}.
\]
The recursive matrix $EN$ is built by putting $1/(N_i + 1)$ in the column corresponding to the bar of the histogram containing the new point, all the other columns are equal to 0.

We organize the new probabilities $P'$ in incremental order:

$$p'_1 < p'_2 < \cdots < p'_h.$$ 

In using the transformation of Dubois and Prade, we can find the recursive matrix giving the new possibilities in incremental order:

$$\left( \begin{array}{c} \pi'_1 \\ \pi'_2 \\ \pi'_3 \\ \vdots \\ \pi'_h \end{array} \right) = \left( \begin{array}{ccccc} h & 0 & 0 & \cdots & 0 \\ 1 & h-1 & 0 & \cdots & 0 \\ 1 & 1 & h-2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 1 \end{array} \right) \left( \begin{array}{c} p''_1 \\ p''_2 \\ p''_3 \\ \vdots \\ p''_h \end{array} \right).$$

Finally, we organize the new possibilities in their original order:

$$\Pi = \left( \begin{array}{cccc} \pi'_1 & \pi'_2 & \cdots & \pi'_h \end{array} \right).$$

6.3. Performances

The number of bars $h$ is equal to 30 for the determination of all the possibility densities. The membership reject threshold is fixed to 0.2. The first learning is realised on a training set containing 125 points in using the incremental learning and the classical one. Each class is divided in three subclasses. The results of the classification of the 640 points are presented in Tables 4 and 5. The points that are not rejected are used to calculate the classification rate

<table>
<thead>
<tr>
<th>Number of subclasses for each class</th>
<th>Number of learning</th>
<th>Classification rate (%)</th>
<th>Membership rejects</th>
<th>Ambiguity rejects</th>
<th>Mean updating time for a point in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>100</td>
<td>42</td>
<td>0</td>
<td>1.2</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>100</td>
<td>30</td>
<td>0</td>
<td>2.6</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>100</td>
<td>25</td>
<td>0</td>
<td>2.8</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>100</td>
<td>19</td>
<td>0</td>
<td>3.0</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>100</td>
<td>19</td>
<td>0</td>
<td>3.2</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>100</td>
<td>19</td>
<td>0</td>
<td>3.5</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>100</td>
<td>19</td>
<td>0</td>
<td>3.5</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>100</td>
<td>17</td>
<td>0</td>
<td>3.7</td>
</tr>
</tbody>
</table>
Table 5
Number of rejected points for different number of learning, with incremental learning

<table>
<thead>
<tr>
<th>Number of subclasses for each class</th>
<th>Number of learning</th>
<th>Classification rate (%)</th>
<th>Membership rejects</th>
<th>Ambiguity rejects</th>
<th>Mean updating time for a point in seconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>1</td>
<td>100</td>
<td>28</td>
<td>0</td>
<td>0.09</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>100</td>
<td>18</td>
<td>0</td>
<td>0.09</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>100</td>
<td>16</td>
<td>0</td>
<td>0.09</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>100</td>
<td>16</td>
<td>0</td>
<td>0.09</td>
</tr>
<tr>
<td>7</td>
<td>5</td>
<td>100</td>
<td>15</td>
<td>0</td>
<td>0.09</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>100</td>
<td>15</td>
<td>0</td>
<td>0.09</td>
</tr>
<tr>
<td>9</td>
<td>7</td>
<td>100</td>
<td>13</td>
<td>0</td>
<td>0.09</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>100</td>
<td>11</td>
<td>0</td>
<td>0.09</td>
</tr>
</tbody>
</table>

Fig. 17. Classification results after the first training, $D$ signifies a membership reject.

Fig. 18. Classification results after the last training, $D$ signify a membership reject.
and are included in the learning set. After each classification phase, a new learning is realised to take into account these new points, in increasing the number of subclasses for each class.

Tables 4 and 5 show that classification rate is always 100%, in this configuration the algorithms do not make any mistake. There is no ambiguity reject indeed our goal was to obtain well-separated classes. The decrease of the number of membership rejects is more and more slow as the number of learning increases. With classical learning, the mean updating time increases with the number of learning. This time is smallest and constant in using the incremental learning. Figs. 17 and 18 show the results after the first and the last learning.

7. Integration of the diagnosis modulus

When a point is affected to a class corresponding to a bad-operating state, the process is put back into a good one in correcting its control parameters. The reject of a point can also lead to an adjustment of control parameters. This option is enabled only when the modulus is used as a supervision system. The complete diagnosis modulus is presented in Fig. 19. After the classification of each new point, the recursive method is used to update the possibility densities. This update is not necessary if the point is rejected. The evolution of the possibility densities allows to take account the extension of the subclasses and consequently of the classes. The update of the number of subclasses for each class uses heuristic criteria and Fuzzy C-Means algorithm. Therefore the calculation times are rather long which forces to do the update only after the classification of a great number of points. We have decided to perform it when all the learning is call into doubt, that is to mean when there is a great number of rejected points. The learning modulus, using UFGC algorithm, allows updating the number of classes which causes the updates of subclasses and possibility densities. It is carried out only when the number of points rejected in membership or in ambiguity exceeds respectively the thresholds $n_{\text{mem}}$ and $n_{\text{amb}}$. The goal of UFGC algorithm used in continuous learning is to create new classes or to merge existing classes. During the functioning of the process, a new operating state may appear. It is possible that the attributes used previously would not be sufficient to discriminate the corresponding class. Therefore, it is necessary to find new efficient attributes in using the pre-processing of the data, to perform a new fuzzy partition in classes in using UFGC algorithm, to compute a new decomposition of the classes into subclasses in using heuristic criteria and Fuzzy C-Means algorithm, and to calculate the new possibility densities. This entire procedure is referred to as general updating.
Fig. 19. Integration of the diagnosis modulus with UFGC and FPMM methods.
8. Conclusion

This study shows that classification is a very interesting tool for the implementation of real time monitoring systems. The results obtained show that the reliability of the developed system is high and that this system is totally adaptive, due to the integration of the incremental or continuous learning proceedings. The control of the number of rejected membership points shows that the incremental learning is particularly efficient to detect the evolutions of the shape of the classes. The first interest is the ability for the system to reveal the derives of the process during the production, which allows to anticipate the maintaining tasks coming from the wear of some parts of the process. The second interest of the incremental training is to reduce the number of training points required by the initial learning. Indeed with the integration of such a method the identification of the functioning modes is possible with a low number of training points. This consequence is particularly interesting in the actual economic context where the production of bad workpiece decreases the productivity and quality rates. This conclusion helps to develop such monitoring systems.

The observed computing times are totally compatible with the cycle injection times. Real time monitoring becomes possible with our system. The incremental learning does not increase the computing times. However the apparition of new class requires some complementary computations. That is why this process is run after the production of several workpieces. Furthermore this process is supervised by the human operator. The computing times remain compatible with the injection cycle times.

Such a monitoring system can be considered as an approximate reasoning system. Every decision is made as a function of experiences learned from situations recorded in the learning knowledge database. The system is adaptive because each new situation is recorded and integrated in the database. This training constitutes a permanent learning. This type of system could be applied in many fields such as medical sciences, biology, industrial design and technical data knowledge management.

References


[26] K. Pearson, On lines and planes of closest fit to systems of points in space, Philosophical Magazine 6 (2) (1901) 559–572.