Proceedings of the 10th Mediterranean Conference
on Control and Automation - MED2002
Lisbon, Portugal, July 9-12, 2002.

ON-LINE QUALITY CONTROL OF ROTATING MACHINERY USING PATTERN RECOGNITION

A. Pouliezos*, C. Cristalli [†]

*Dept. of Production and Management Engineering, Technical University of Crete, 73100 Hania, Greece.
fax: +308210 69410
e-mail: tasos@dpem.tuc.gr
[†]AEA srl, Via Fiume 16, 60030 Angeli di Rosora (AN), Italy.
fax: +3907318161
e-mail: ccristalli@loccioni.com

MED2002 Conference
e-mail: med2002@isr.ist.utl.pt
http://www.isr.ist.utl.pt/med2002

Keywords: quality control, pattern recognition

Abstract

In this paper it is presented an on-line quality control system for electric motors. It is comprised of accelerometers for vibration measurement and an intelligent monitoring and classification module centred around a PC. The whole system performs very well, can be easily incorporated into existing production lines and has a wide application range.

1 Introduction

Quality control is an essential part of the manufacturing process. As industrial competition increases, the need for reliable and economical quality control becomes even more pressing. Thus the incorporation of automated systems for on-line inspection of finished products is already gaining popularity. These systems, by their own nature, are highly sophisticated elements in the production line, utilizing so called “intelligent techniques” and advanced sensor technology.

Appliances incorporating rotating machinery, namely electric motors, have in particular received a lot of attention due in part to their wide field of application (washing machines, refrigerators, compressors, motors themselves, etc.). The production quality monitoring of such machines is facilitated by means of sensors that measure vibration (accelerometers) and perhaps acoustical noise produced. The problem is thus the automatic classification of the monitored production line into healthy and faulty products.

As early as 1991, Barschdorf [1] compares various approaches for detecting faults in the production of electric motors. Fogliardi [2], uses a fuzzy c-means algorithm for detecting inductive motors characterised by undesired noise due to rotor shaft vibrations. Lastly Paone *et al*. [5] and Goumas *et al*. [3] are considering the problem of quality control in washing machines. Pattern recognition methods utilizing Fourier and wavelet features are compared in these two studies.

In this paper we consider the problem of on-line quality control of electric motors. Our system comprises of a measurement subsystem which incorporates accelerometer readings of the motor’s vibration in various points, complemented by a decision subsystem which processes the measurements and produces a decision on the health state of the product. The whole system can be easily added to existing production lines, and is capable of adaptation to new environments. Its application is not limited to the specific product but can be used in different products.

2 Statement of the problem

A manufacturer is making electric motors in a production line. The assembled parts are undergoing various quality control tests, and end up in an automated test bed where final inspection takes place. The problem is to design a system for the automatic detection of manufacturing defects of the finished product. In the present study, two such defects are considered, namely,

H_1: increased brush noise

H_2: faulty bearing

These are complemented by,

H_0: no defect

The system should be able to complete the inspection process in such a time period so as not to congest the production line. In the present situation this is of the order of 20secs.

The system is comprised of a hardware and software part. The hardware part consists of the data acquisition subsystem and a PC while the software part consists of appropriate code for data processing and decision analysis. The acquisition subsystem utilizes accelerometers connected to a PC via a DSP board, while software includes mainly MATLAB routines for on-line data acquisition, data processing and pattern recognition/decision making modules.

3 Solution of the problem

To solve the problem described in the previous section, it is cast in a pattern recognition framework, and particularly in its supervised version. Its formal mathematical representation is:

There are available N associations,

$$S=\{\boldsymbol{x}_i,\ y_i\};\ i=1,\ \ldots,\ N$$

Then given a new $\boldsymbol{x}$ what should the predicted y be?

In this context $\boldsymbol{x}_i=[x_{i1},\ \ldots,\ x_{in}]$ is usually termed the *feature vector* (of dimension n), while $y_i \in \{\omega_1, \ldots, \omega_L\}$ defines the *feature class*. The set S is then the *training set*.

The training set typically reflects a functional relationship mapping inputs to outputs, though this may not be the case as for example when the outputs are corrupted by noise. When an underlying function from inputs to outputs exists it is referred to as the *target function*. The estimate of the target function which is learnt by the learning algorithm, is known as the *solution* of the learning problem. In the case of classification this function is known as the *decision function*. The solution is chosen from a set of candidate functions which map the input space to the output domain. Usually we will choose a particular set of candidate functions known as hypotheses before we begin trying to learn the correct function. Hence we can view the choice of a set of hypotheses (or *hypothesis space*) as one of the key ingredients of the learning strategy. The algorithm which takes the training data as input and selects a hypothesis from the hypothesis space is the second important ingredient, an it is called the *learning algorithm*.

The goal of a learning algorithm should be to be able to *generalize*. By this is meant that the algorithm must have the ability to classify data not in the training set. Having said that the problem becomes one of deciding on an appropriate criterion to achieve this goal. Early approaches were designed to correctly classify the training data, thus performing poorly on unseen samples. However, the shortcomings of this approach, exhibited especially in hard, real life problems, led to a change of emphasis towards the generalization phase. There are two main directions in this viewpoint: the *Bayesian* model and the *Probably Approximately Correct* (*pac*) model. In the former we are attempting to find the most likely solution while in the latter we are trying to bound the generalization error. Both approaches are compared in this study.

3.1 Statistical pattern recognition

In the statistical pattern recognition framework, it is assumed that there exists some unknown probability distribution $P(\boldsymbol{x},y)$ from which the data are drawn, i.e. they are supposed to be independently drawn and identically distributed (i.i.d.). This is a quite general assumption allowing a distribution of y for a given $\boldsymbol{x}$. However, in the sequel fixed y for given $\boldsymbol{x}$ will be assumed.

In this framework, the problem is rephrased as follows:

Given $S=\{\boldsymbol{x}_i,\ y_i\};\ i=1,\ \ldots,\ N$ find a decision rule d such that a new observation vector $\boldsymbol{x}$ is classified unambiguously into one of the L classes in an "optimum" way (in other words the sample space is divided into L exhaustive and exclusive regions R_1, …, R_L). In this way classification is framed as a multiple hypothesis testing problem where,

> H_i: hypothesis that observation $\boldsymbol{x}$ belongs to the ith class

and rule d is the adopted statistic.

The Bayesian model

In the Bayesian approach the criterion of optimality that is minimized is the "*probability of error*" or *Bayes error* defined as,

$$\varepsilon = 1 - \sum_{i=1}^{L} \left\{ P(\omega_i) \int_{R_i} p(\boldsymbol{x}|\omega_i)\mathrm{d}\boldsymbol{x} \right\} \tag{1}$$

where,

$p(\boldsymbol{x}|\omega_i)$: conditional probability density function for the ith class

$P(\omega_i)$: a priori probability of class i (probability of observation $\boldsymbol{x}$ belonging to class ω_i)

$P(\omega_i \mid \boldsymbol{x})$: a posteriori probability of ω_i given $\boldsymbol{x}$

R_i: regions where observation $\boldsymbol{x}$ is classified in class ω_i if rule d. is adopted.

$p(\boldsymbol{x})$: unconditional density function of $\boldsymbol{x}(s)$ (or *mixture density function*)

the mixture density function is defined by,

$$p(\boldsymbol{x}) = \sum_{i=1}^{L} p(\boldsymbol{x}|\omega_i)P(\omega_i) \tag{2}$$

and the a posteriori probability by,

$$P(\omega_i \mid \boldsymbol{x}) = \frac{P(\omega_i)\,p(\boldsymbol{x}|\omega_i)}{p(\boldsymbol{x})} \tag{3}$$

(Equation (1) is actually 1-(total probability of correct classification). The total probability of correct classification is a sum of terms of which the ith one represents the probability of correct classification of the ith class times the probability of occurrence of the ith class.)

It can be proved that the decision rule obtained if this criterion is adopted is the following:

d : assign $\boldsymbol{x}$ to the class with the maximum a posteriori probability $P(\omega_i \mid \boldsymbol{x})$

Considering (3) and the fact that (2) is always positive, maximisation of (3) amounts to maximising $P(\omega_i)\,p(\boldsymbol{x}|\omega_i)$, i.e. the classifying rule for the new pattern is,

d : assign $\boldsymbol{x}$ to class ω_i if

$$P(\omega_i)\,p(\boldsymbol{x}|\omega_i) > P(\omega_j)\,p(\boldsymbol{x}|\omega_j) \qquad \forall j \neq i \tag{4}$$

So all looks simple as long as $p(\boldsymbol{x}|\omega_i)$ and $P(\omega_i)$ are known. Since this is never the case in real life, every algorithm tries to estimate $p(\boldsymbol{x}|\omega_i)$ in one way or another (knowledge of the $P(\omega_i)$'s is acquired easier). When this is done, decision is straightforward. However its qualities depend on how well $p(\boldsymbol{x}|\omega_i)$ is approximated.

This is the first source of error.

There exist two categories of density estimators: parametric and non parametric. In the parametric approach, a form of the underlying distribution is assumed (e.g. normal) and its parameters (e.g. $\boldsymbol{m}_i$, $\boldsymbol{\Sigma}_i$) approximated from the available training samples.

If it can be assumed that the $p_i(\boldsymbol{x})$ are $N(\boldsymbol{m}_i, \boldsymbol{\Sigma}_i)$ (an assumption which could be tested, see later on), we can proceed a little bit further. Using (4), our decision rule boils down to the well known *quadratic rule*,

$$d = \arg\left\{\max_i\left\{\frac{1}{\left|\boldsymbol{\Sigma}_i\right|^{\frac{1}{2}}}\exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_i)^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x}-\boldsymbol{m}_i)\right\}P(\omega_i)\right\}\right\} \tag{5}$$

However (5) can only be approximated in practice since $\boldsymbol{m}_i$, $\boldsymbol{\Sigma}_i$ have to be estimated. This is done from the available measurements using the well known unbiased formulae,

$$\begin{aligned}\hat{\boldsymbol{m}}_j &= \frac{1}{N_j}\sum_{i=1}^{N_j}\boldsymbol{x}_i \\ \hat{\boldsymbol{\Sigma}}_j &= \frac{1}{N_j-1}\sum_{i=1}^{N_j}(\boldsymbol{x}_i-\hat{\boldsymbol{m}}_j)(\boldsymbol{x}_i-\hat{\boldsymbol{m}}_j)^{\mathrm{T}};\ j=1,\ldots,L\end{aligned} \tag{6}$$

where N_j is the sample size for class j. The variance of both the above estimates is proportional to $1/N_j$. The estimated values is the second source of error.

If no assumption on the underlying pdf is desirable or available, non parametric density estimators can be used. These techniques are basically variations of the *histogram* approximation of an unknown pdf, where the probability of a sample x being located in a bin is estimated for each of the bins. If N is the total number of samples and k_i of these are located in bin i, the corresponding probability is approximated by the frequency ratio,

$$P \approx k_i/N$$

The corresponding pdf is assumed constant throughout the bin and is approximated by,

$$\hat{p}(x) \equiv \hat{p}(\hat{x}) \approx \frac{1}{h(N)}\frac{k_i}{N}, |x-\hat{x}| \le \frac{h(N)}{2} \tag{7}$$

where $\hat{x}$ is the midpoint of the bin. It can be shown that $\hat{p}(x)$ converges to the true pdf as $N \to \infty$ provided certain conditions are satisfied. In words N must be “large enough”, $h(N)$ “small enough” and the number of points in each bin “large enough” too.

In the multidimensional case, instead of bins of size h, the n-dimensional space is divided into hypercubes of volume h^n. By following the same reasoning, the multidimensional equivalent of (7), can be written,

$$\hat{p}(x) = \frac{1}{h^l}\left(\frac{1}{N}\sum_{i=1}^{N}\varphi\left(\frac{x_i-x}{h}\right)\right) \tag{8}$$

where $\varphi()$ is any smooth function satisfying,

$$\begin{gathered}\varphi(\boldsymbol{x}) \ge 0 \\ \int_{\boldsymbol{x}}\varphi(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = 1\end{gathered}$$

These functions are called *Parzen windows*. If the logic in the previous section is reversed, i.e. the number of points k_i is fixed (=k) and the volume adjusted so that it contains these k points, the estimator can be written,

$$\hat{p}(\boldsymbol{x}) = \frac{(k-1)}{NV(\boldsymbol{x})} \tag{9}$$

where the dependence of the volume $V(\boldsymbol{x})$ on $\boldsymbol{x}$ is explicitly shown. This defines the *k Nearest Neighbour* (*kNN*) density estimate. (k-1) is used instead of k to make the estimate unbiased.

The classification rule resulting from this estimator is derived as follows: having received the unclassified pattern vector $\boldsymbol{x}$, its distances d from all the points in the training sample is calculated. Let $V_i(\boldsymbol{x})$ be the volume of the hypersolid centred at $\boldsymbol{x}$ and containing the nearest k_i points from ω_i. Then (4) becomes,

d : assign $\boldsymbol{x}$ to ω_i if

$$\frac{(k_i-1)P(\omega_i)}{N_iV_i(\boldsymbol{x})} > \frac{(k_j-1)P(\omega_j)}{N_jV_j(\boldsymbol{x})};\forall i \ne j$$

Distances and volumes are calculated using any appropriate metric. In the case of Mahalanobis distance, we have,

$$\begin{gathered}V_i(\boldsymbol{x}) = \pi^{\frac{n}{2}}\Gamma^{-1}(\tfrac{n}{2}+1)\left|\boldsymbol{\Sigma}_i\right|^{\frac{1}{2}}d_i^n(\boldsymbol{x}_{k_iNN}^i, \boldsymbol{x}) \\ d_i^2(\boldsymbol{y},\boldsymbol{x}) = (\boldsymbol{y}-\boldsymbol{x})^{\mathrm{T}}\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{y}-\boldsymbol{x})\end{gathered}$$

In this case the solids are hyperellipsoids. A difficulty with this approach lies in the selection of the k_i’s. In practice, simulation may give the answer. For gaussian distributions, exact results can be obtained [4].

Note: This classifier is addressed by the adjective “volumetric”. This is to distinguish it from its more frequently used relative, which we shall call “voting". Its rule will be described next though it does not really fit in the Bayesian framework.

Instead of selecting the kth nearest neighbour from each class and comparing the distances, the kNNs of an unclassified pattern are selected and the number of neighbours from each class among the k selected samples is counted. The unclassified pattern is then assigned to the class represented by a majority of the kNNs. That is, the rule becomes,

$$\mathrm{d} : \text{assign } \boldsymbol{x} \text{ to } \omega_i \text{ if } k_i = \max\{k_1, \ \dots, \ k_L\}; \\ k = k_1 + \dots + k_L$$

where k_i is the number of neighbours from ω_i (i=1, …, L) among the kNNs.

4 Implementation and results

Having outlined the various approaches towards solving general pattern recognition problems, we now proceed to defining the specific parameters for the problem at hand, i.e. the on-line quality control of electric motors.

Here the feature class is,

$$\Omega = \begin{bmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} = \begin{bmatrix} \text{brush noise} \\ \text{bearing} \\ \text{no defect} \end{bmatrix} \tag{10}$$

while the feature set is described in Table 1.

Feature no.	Phase	Accelerometer	Feature description
x_1	Transient	Longitudinal	Cepstrum Maximum
x_2	"	"	Envelope Maximum
x_3	"	"	Wavelet
x_4	"	Radial	Envelope Maximum
x_5	"	"	Short Time Frequency Analysis - Window Energy value
x_6	"	"	Cepstrum maximum
x_7	Steady State	Radial	RMS
x_8	"	Longitudinal	RMS
x_9	Steady State	Longitudinal	Cepstrum maximum

Table 1. Feature set

For testing the validity of the algorithm we used a training set consisting of N=215 samples, 39 of which belong to class 1, 70 to class 2 and 106 to class 3.

For verification purposes we used cross-validation: in k-fold cross-validation, the data is divided into k subsets of (approximately) equal size. The classifier is trained k times, each time leaving out one of the subsets from training, but using only the omitted

Table 2. Summary of results

		3-class			2-class		
		full feature set		almost Gaussian set [1 6 9]	full feature set		almost Gaussian set [1 6 9]
		Unedited	Excluding outliers		Unedited	Excluding outliers	
kNN		0.9 (6)	**0.91**(6)	0.84 (7)	0.935 (3/5/6)	0.95(6)	0.95 (5/6)
Quadratic		0.86	0.89	0.84	0.89	**0.96**	0.94
SVM	C=1	0.88	0.89	0.83	0.92	0.93	0.94
	C=10/20		0.896			0.936	
voting			0.90			0.95	

subset to compute the error criterion. If k equals the sample size, this is called the "*leave-one-out*" cross-validation.

The results of the experiments for the “leave-one-out” cross-validation are summarized in Table 2.

The two major categories, namely 3-class and 2-class, denote classification into 3 classes as defined by (10) and classification into {faulty, healthy} respectively.

Each of these categories is further subdivided into a full feature set and an “almost gaussian set”. The full feature set is defined in Table 1, while the “almost gaussian set” is composed of those features that are the closest to a normal distribution. The selection is based on a χ^2 goodness of fit test (see Fig. 1 features 1, 2, 3 and 4). As seen features 1, 6 and 9 form this “almost gaussian set”.

The full feature set is finally subdivided into an “unedited” and an “excluding outliers” column. The former denotes the training data includes all the measurements, while in the latter the “outliers” have been excluded. Outliers have been detected as follows:

Suppose $\boldsymbol{x}_i \in \Re^n$, i=1, …, N is a sample with mean vector $\boldsymbol{m}$ and covariance matrix $\boldsymbol{\Sigma}$, estimated by $\hat{\boldsymbol{m}}, \hat{\boldsymbol{\Sigma}}$. Let $\boldsymbol{x}_0$ be one of the observation vectors and define its distance from $\hat{\boldsymbol{m}}$ as,

$$D^2 = (\boldsymbol{x}_0 - \hat{\boldsymbol{m}})^{\mathrm{T}} \hat{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{x}_0 - \hat{\boldsymbol{m}})$$

The distribution of D^2 looks like a T^2 but since $\boldsymbol{x}_0$ is contained in the sample that is used to estimate $\hat{\boldsymbol{m}}, \hat{\boldsymbol{\Sigma}}$, it can be shown that is distributed as,

$$D^2 = \frac{(N-1)^2 nF}{N(N-n-1+nF)}$$

i.e. as a central F random variable with (n, N-n-1) degrees of freedom. Conversely the quantity,

$$F = \frac{(N-n-1)ND^2}{n(N-1)^2 - NnD^2} \tag{11}$$

has the F distribution with (n, N-n-1) degrees of freedom. We would therefore reject the null hypothesis of a common population mean vector for $\boldsymbol{x}_0$ and the remaining $\boldsymbol{x}_i$ at significance level α, if,

$$F > F_{\alpha;n,\, N\text{-}n\text{-}1}$$

Now suppose $\boldsymbol{x}_0$ is the observation with the maximum D^2 statistic. The distribution of this maximum statistic is rather complicated but a conservative approximation to the 100α percent upper critical value can be obtained by the Bonferroni inequality, yielding the decision rule that $\boldsymbol{x}_0$ can be considered an outlier if its statistic (11) exceeds the critical value,

$$F > F_{\alpha / N;n,\, N\text{-}n\text{-}1}$$

Four different classification algorithms have been compared: volumetric kNN, quadratic, support vector machine (SVM) [6] and voting kNN. In the kNN cases, various values for the number of nearest neighbours have been tried, the optimum ones being shown in parantheses. For the SVM three values for the regularisation parameter C are compared, namely 1, 10 and 20.

As can be seen from Table 2, for the 3-class problem the overall best results are obtained with the volumetric kNN classifier (k=6) operating on the full feature set excluding outliers: 0.91 This is followed closely by the voting kNN, the quadratic and the SVM. No significant performance differences are observed however.

For the 2-class problem the performance results are better with the quadratic classifier being slightly ahead (at 0.96). Again differences among the various classifiers are insignificant. The better performance of this case could be attributed to the fact that this presents an easier task for the classification algorithm than the 3-class problem.

The results show that it would be difficult to improve on the performance by concentrating on improving a classifier. It looks that it is the features that might not contain enough information, for the classification to be of the order of 99%, and that if one wants a better performance, he should better look into obtaining a different set of features. Moreover it is not the number of features that plays an important role, but their quality: the “almost gaussian set” even though it contains only 3 features, it does not perform much worse (especially in the 2-class case). Hence, it is concluded that it is

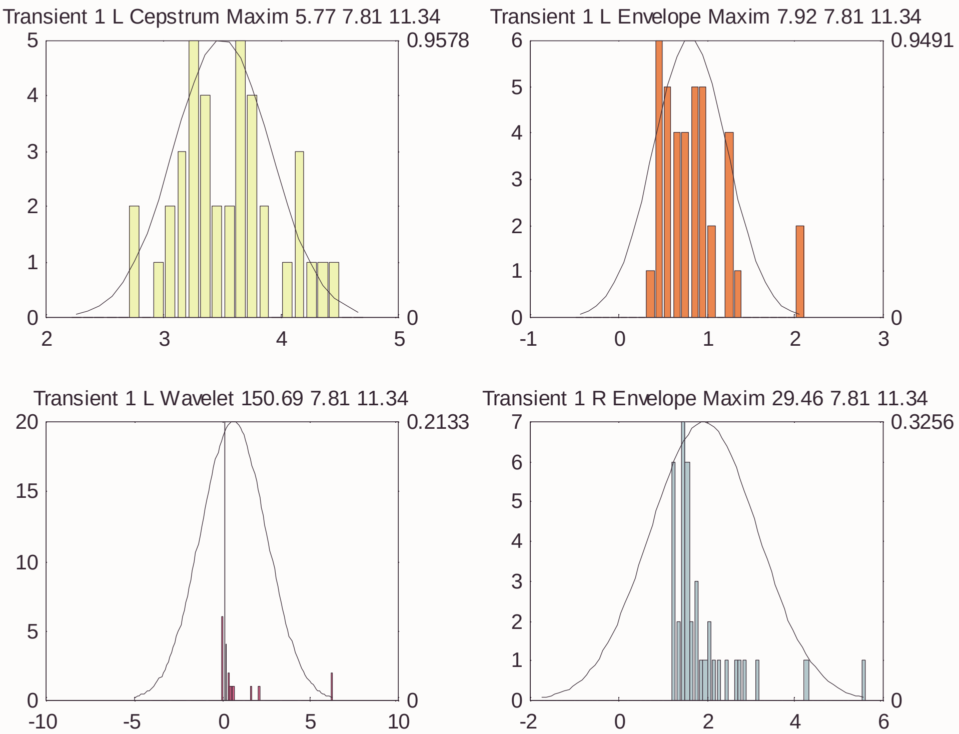

Figure 1. Feature histogram plots

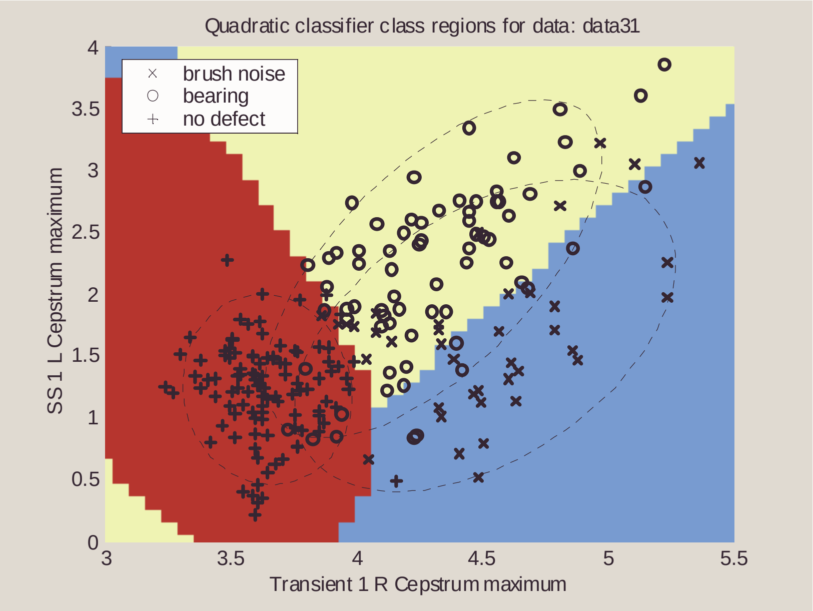

Figure 2. Quadratic regions for features 6 and 9.

the quality and not the quantity of features that matters (it has been argued that the “classifier” gets a headache with many features presented to it).

Fig. 2 shows a typical region separation for features 6 and 9 (for visualization purposes) using the quadratic classifier. One can easily seen the difficult task of the classifier, due to the heavy overlap of the regions.

A last point concerns the on-line implementation of this procedure. In order to be able to adapt to changes in the production line, a sliding window of data can be employed. The size of the window should be chosen large enough for the accurate estimation of sample statistics, and small enough for adequate sensitivity to changes.

5 Conclusions

In this paper we have presented a system for on-line quality control of electric motors. The system utilizes advanced sensors and “intelligence”. Initial results are promising, and believed to be capable of improving by a more careful selection of the feature set.

References

[1] D. Barschdorf. “Comparison of neural and classical decision algorithms”, *Proceedings, IFAC Fault Detection, Supervision and Safety for Technical Processes*, Baden-Baden, Germany, pp. 409-415, 1991.

[2] R. Fogliardi. “Fuzzy identification of noisy electric motors on the production line”, *Proceedings, EUFIT ’97*, Aachen, Germany, pp. 1755-1759 September 8-11, 1997.

[3] S. Goumas, M. Zervakis, G. Stavrakakis and A. Pouliezos. “Classification of Washing Machines Vibration Signals using Discrete Wavelet Analysis for Feature Extraction”. To appear in *Engineering Applications of Artificial Intelligence*.

[4] K. Fukunaga. “Statistical pattern recognition”. Academic Press, 1990.

[5] N. Paone, L. Scalise, G. Stavrakakis and A. Pouliezos. Fault detection for quality control of household appliances by non-invasive laser Doppler technique and likelihood classifier. *Measurement*, **25**, 237-247, 1999.

[6] Vapnik V.N. (1995). “The Nature of Statistical Learning Theory”, Springer-Verlag, New York.D.

Acknowledgments

This work has been funded by AEA srl, Angeli di Rosora, Ancona, Italy.