José Carlos PALOMARES-SALAS[1], Juan José GONZÁLEZ DE LA ROSA[1], Agustín AGÜERA-PÉREZ[1], Antonio MORENO-MUÑOZ[2]

Research Group PAIDI-TIC-168. Area of Electronics.

Intelligent Methods for Characterization of Electrical Power Quality Signals using Higher Order Statistical Features

Streszczenie. *W artykule analizowano kilka technik klasyfikacji do identyfikacji zakłóceń jakości energii. Wykorzystano narzędzia statystyki i wyznaczono geometryczne wzory obrazujące zakłócenia amplitudy i częstotliwości w sieci 50 Hz. Opracowano metodę automatycznej klasyfikacji.* ***(Inteligentne metody charakteryzacji jakości energii elektrycznej bazujące na metodach statystycznych)***

Abstract. *This paper considers a few important techniques classification for to identify several power quality disturbances. For this purpose, a process based in HOS has been realized to extract features that help in classification. In this stage the geometrical pattern established via higher-order statistical measurements is obtained, and this pattern is function of the amplitudes and frequencies of the power quality disturbances associated to the 50-Hz power-line. Once the features are managed will be segmented to form training and test sets and them will be applied in the statistical methods used to perform automatic classification of PQ disturbances. The best technique of those compared is selected according to correlation and mistake rates.*

Słowa kluczowe: Jakość energii, stany przejściowe, metody statystyczne.
Keywords: Higher-Order Statistics (HOS); Artificial Neural Networks (ANN); Power-Quality (PQ); Self Organizing Maps (SOM); Transients.

Introduction

The increasing pollution of power signals and its impact on the power quality supplied by power plants to customers are pushing forward the development of signal processing tools to monitor and control of Power Quality (*PQ*) anomalies. For this reason, the interest of the research community in *PQ* has dramatically increased over the past decade. Among the most important reasons behind such interest are [1]: (1) the release of electric power markets which generates a competitiveness based on the electric quality of service, (2) the increased complexity of power networks, and, (3) the major sensitivity of modern loads to power quality disturbances. Consequently, monitoring of *PQ* disturbances is essential to offer solutions to industrial and electrical areas.

With the increasing amount of measurement data from *PQ* monitors, it is desirable that analysis, characterization, classification, and compression can be performed automatically [2]. Further, it is desirable to find out the cause of each disturbance, for example, whether a voltage dip is caused by a fault or by some other system event such as motor starting or transformer energizing. For this purpose, companies are putting a lot of efforts in innovation and new technologies because today's equipment, and automated manufacturing devices, are highly sensitive to the power line signal's imperfections (*PQ* events), making the production cost excessive. Malfunctioning not only has to be detected, but also predicted, undoubtedly diagnosed and localized, to identify the cause and prevent the system from a similar shock. This would be reflected *a posteriori* in an enhancement of the industrial production [3].

While the needs for better management of *PQ* are imminent, progress has been relatively slow. The power quality analysis and diagnosis problem is complex for many reasons, including but not limited to, the extensive amount of system data that is currently available through *PQ* monitors and the lack of *PQ* experts who posses the required expertise. Moreover, such required expert knowledge is high level and further interdisciplinary.

Actually a good number of the real recorded disturbances are analyzed manually by specialists. Nevertheless, is possible to use several techniques to identify and classify disturbances automatically. Thus, the specialist could center the attention directed toward resolving more complicated power quality problems. Hence, it is evident that a computerized system analysis is essential for the realization of effective and efficient power quality diagnosis systems.

With the development of the wavelet transform technique, how to detect the *PQ* signals tends to be easy. However, recent results have indicated that wavelet transform-based techniques are very sensitive to the presence high power background noise [4], as such as how to classify the power quality signals is still difficult, and it is becoming more and more important [5, 6]. Another very interesting techniques are the ones that make use of second-order statistics of the error signal to detect the occurrence of disturbances [7, 8, 9]. However, it is worth mentioning that the second-order statistics are very sensitive to the presence of Gaussian noise that usually model the background noise in voltage signals. As a result, the use of such statistics could not be appropriate for those cases where the power of background noise is high. On the other hand, the use of higher-order statistics (*HOS*) such as cumulants is very interesting, because they are insensitive to the presence of Gaussian noise [10]. They are based in the following premise. Without perturbation, the 50-Hz of the voltage waveform exhibits a constant statistical behavior (stationary), generally Gaussian. Deviations can be detected and characterized via *HOS*, non-Gaussian process need at least 3^{rd} and 4^{th}-order statistical characterization in order to be completely characterized, because 2^{nd}-order moments and cumulants are not capable of differentiate non-Gaussian events. Recent works are bringing a higher-order statistics (*HOS*) based strategy, dealing with *PQ* analysis [11, 12], and other fields of Science and Technology [13, 14].

As standard $IEEE1159$ define [15], there are many types of power quality signals, such as voltage swell, voltage sag, outage and so on. It is hard to classify all these signals with the traditional tools or mathematics theory but not with the artificial intelligence theory.

Artificial intelligence techniques have had a great prosperity in the solution of the pattern classification problem and exist a significant amount of literature on automatic classification of power quality disturbances, among others [16, 17, 18, 19, 20, 21, 22]. In the state of the art there are several classification techniques, the most used are: Artificial Neural Networks (ANN), Fuzzy Logic, Genetic Algorithms, Case-based Reasoning (CBR), Bayesian Technique, Quadratic Discriminant Analysis (QDA), Linear Discriminant Analysis (LDA), Decision Trees and Support Machine Vector (SVM), among others.

Nevertheless, many open question remain, particularly concerning the most effective methods for classifying *PQ* problems.

This paper represents another step in this direction. To start with, many signals are generated by simulating different types of events causing poor power quality. These signals are used as inputs to feature extraction module which is used for selecting a number of representative coefficients of the generated signals. Such representative coefficients are based on higher-order characterization of *PQ* disturbances according to the maxima and minima of the 2^{nd}, 3^{rd}, 4^{th}, 5^{th} and 6^{th}-order cumulants at zero lags (directly related to the variance, skewness, kurtosis, standard moment 5^{th} and 6^{th}-order). Once the features vectors have been obtained, we will apply double crossvalidation method to obtain training and validation sets which will be used in variety of classify techniques. The techniques used are: linear discriminant analysis, *k*-nearest neighbors, competitive layers, self-organizing maps, *LVQ* and *SVM*. Finally, the correlation and mistake rates of different classification algorithms have been obtained for comparison of them.

The paper is structured as follows. The following Section 2 explains the higher-order statistics. A summary of the different classification techniques listed below in Section 3. Finally, the results are presented in Section 4 and conclusions are drawn in Section 5.

Higher-Order statistics

HOS measures are extensions of second-order measures (such as the autocorrelation function and power spectrum) to higher orders. The second-order measures work fine if signals has a Gaussian (Normal) probability density function, but as mentioned above, many real-life signals are non-Gaussian. Some of the key advantages to such techniques over traditional second-order techniques are:

- Second-order measures contain no phase information. As a consequence of this, non-minimum phase signals and certain types of phase coupling (associated with nonlinearities) cannot be correctly identified by 2^{nd}-order techniques.
- Any Gaussian signal is completely characterized by its mean and variance. Consequently the *HOS* of Gaussian signals are zero (e.g. the $3^{r}d$-order moment of a Gaussian signal is zero), or contain redundant information. Many signals encountered in practice have non-zero *HOS*, and many measurement noises are Gaussian, and so in principle the *HOS* are less affected by Gaussian background noise than the $2^{n}d$-order measures.

Moreover some contributions have demonstrated that *HOS*-based techniques are more appropriate to deal with non-Gaussian processes and nonlinear systems than $2^{n}d$-order-based ones. Remarkable results regarding detection, classification and system identification with cumulant-based technique have been reported.

Higher-order cumulants are used to infer new properties about the data non-Gaussian process [23, 24]. In multiple-signal processing it is very common to define the combinational relationship among the cumulants of *r* stochastic signals, $\{x_i\}_{i\in[1,r]}$, and their moments of order *p*, $p \leq r$, given by using the *Leonov-Shiryaev* formula [10]

$$Cum(x_1,...,x_r) = \sum(-1)^{p-1}\cdot(p-1)!\cdot E\{\prod_{i\in s_1} x_i\} \cdot E\{\prod_{i\in s_2} x_j\}\cdots E\{\prod_{i\in s_p} x_k\}, \tag{1}$$

where the addition operator is extended over all the partitions, like one of the form $(s_1, s_2, \ldots, s_p)$, $p = 1, 2, \cdots, r$; and $(1 \leq i \leq p \leq r)$; being s_i a set belonging to a partition of order p, of the set of integers *1,. . . ,r*.

Let $\{$*x(t)*$\}$ be an *r*th-order stationary random real-valued process. The *r*th-order cumulant is defined as the joint *r*th-order cumulant of the random variables *x(t)*, *x(t+*τ_1*)*,. . . , *x(t+*τ_{r-1}*)*,

$$\begin{aligned} C_{r,x}(\tau_1, \tau_2, \ldots, \tau_{r-1}) \\ = Cum[x(t), x(t+\tau_1), \ldots, x(t+\tau_{r-1})] \end{aligned} \tag{2}$$

The second-, third-, fourth-, fifth- and sixth-order cumulants of zero-mean *x(t)* can be expressed via [25]:

$$C_{2,x}(\tau) = E\{x(t)\cdot x(t+\tau)\} \tag{3a}$$

$$C_{3,x}(\tau_1,\tau_2) = E\{x(t)\cdot x(t+\tau_1)\cdot x(t+\tau_2)\} \tag{3b}$$

$$\begin{aligned} &C_{4,x}(\tau_1,\tau_2,\tau_3) \\ &= E\{x(t)\cdot x(t+\tau_1)\cdot x(t+\tau_2)\cdot x(t+\tau_3)\} \\ &\quad - C_{2,x}(\tau_1)C_{2,x}(\tau_2-\tau_3) \\ &\quad - C_{2,x}(\tau_2)C_{2,x}(\tau_3-\tau_1) \\ &\quad - C_{2,x}(\tau_3)C_{2,x}(\tau_1-\tau_2) \end{aligned} \tag{3c}$$

$$\begin{aligned} &C_{5,x}(\tau_1,\tau_2,\tau_3,\tau_4) \\ &= E\{x(t)\cdot x(t+\tau_1)\cdot x(t+\tau_2)\cdot x(t+\tau_3) \\ &\quad \cdot x(t+\tau_4)\} \\ &\quad - C_{2,x}(\tau_1)C_{2,x}(\tau_2-\tau_3-\tau_4) \\ &\quad - C_{2,x}(\tau_2)C_{2,x}(\tau_3-\tau_4-\tau_1) \\ &\quad - C_{2,x}(\tau_3)C_{2,x}(\tau_4-\tau_1-\tau_2) \\ &\quad - C_{2,x}(\tau_4)C_{2,x}(\tau_1-\tau_2-\tau_3) \end{aligned} \tag{3d}$$

$$\begin{aligned} &C_{6,x}(\tau_1,\tau_2,\tau_3,\tau_4,\tau_5) \\ &= E\{x(t)\cdot x(t+\tau_1)\cdot x(t+\tau_2)\cdot x(t+\tau_3) \\ &\quad \cdot x(t+\tau_4)\cdot x(t+\tau_5)\} \\ &\quad - C_{2,x}(\tau_1)C_{2,x}(\tau_2-\tau_3-\tau_4-\tau_5) \\ &\quad - C_{2,x}(\tau_2)C_{2,x}(\tau_3-\tau_4-\tau_5-\tau_1) \\ &\quad - C_{2,x}(\tau_3)C_{2,x}(\tau_4-\tau_5-\tau_1-\tau_2) \\ &\quad - C_{2,x}(\tau_4)C_{2,x}(\tau_5-\tau_1-\tau_2-\tau_3) \\ &\quad - C_{2,x}(\tau_5)C_{2,x}(\tau_1-\tau_2-\tau_3-\tau_4) \end{aligned} \tag{3e}$$

By putting $\tau_1 = \tau_2 = \tau_3 = \tau_4 = \tau_5 = 0$ in Eq. (3), we obtain

$$\gamma_{2,x} = E\{x^2(t)\} = C_{2,x}(0) \tag{4a}$$

$$\gamma_{3,x} = E\{x^3(t)\} = C_{3,x}(0,0) \tag{4b}$$

$$\gamma_{4,x} = E\{x^4(t)\} - 3(\gamma_{2,x})^2 = C_{4,x}(0,0,0) \tag{4c}$$

$$\begin{aligned} \gamma_{5,x} &= E\{x^5(t)\} - 10\gamma_{3,x}\gamma_{2,x} \\ &= C_{5,x}(0,0,0,0) \end{aligned} \qquad \text{(4d)}$$

$$\begin{aligned} \gamma_{6,x} = E\{x^6(t)\} - 15\gamma_{4,x}\gamma_{2,x} - 10(\gamma_{3,x})^2 \\ - 15(\gamma_{2,x})^3 = C_{6,x}(0,0,0,0) \end{aligned} \qquad \text{(4e)}$$

The expressions in Eq. (4) are measurements of the variance, skewness, kurtosis, fifth-order standard moment, and sixth-order standard moment of the distribution in terms of cumulants at zero lags (the central cumulants).

Normalized sixth-order standard moment, fifth-order standard moment, kurtosis and skewness are defined as $\gamma_{6,x}/(\gamma_{2,x})^3$, $\gamma_{5,x}/(\gamma_{2,x})^{5/2}$, $\gamma_{4,x}/(\gamma_{2,x})^2$ and $\gamma_{3,x}/(\gamma_{2,x})^{3/2}$, respectively. We will use and refer to normalized quantities because they are shift and scale invariant. If *x(t)* is symmetrically distributed, its skewness is necessarily zero (but not *vice versa*, almost impossible situations); if *x(t)* is Gaussian distributed, its kurtosis is necessarily zero (but not *vice versa*). In the section 0.2.4, results are obtained by using sliding cumulants, i.d. a moving window in the time domain over which to compute the each cumulant ($3rd$, $4th$, $5th$ and $6th$-order cumulants for zero time-lag).

Classification techniques

In this section we present a brief theory explication of the classification techniques that have been used. These methods are divided into two principal categories: Traditional Statistical Methods and Intelligent Statistical Methods. In the firsts of them we used two techniques: Linear Discriminant Analysis (*LDA*) and Nearest Neighbor Methods (*kNN*), and in the second ones, four techniques have been used: Competitive Neural Networks (*CNN*), Self-Organizing Feature Maps (*SOFM*), Learning Vector Quantization Networks (*LVQ*) and Support Vector Machine (*SVM*).

0.1 Traditional Statistical Methods

0.1.1 Discriminant Analysis

Linear Discriminant Analysis (*LDA*) is a classic method of classification that has stood the test of time. This is a method used in statistics, pattern recognition and machine learning to find a linear combination of features which characterize or separate two or more classes of objects. The resulting combination may be used as a linear classifier.

The main purpose of a discriminant function analysis is to predict group membership based on a linear combination of the interval variables. The procedure begins with a set of observations where both group membership (dependent categorical variable) and the values of the interval variables are known (independent variables with normal distribution). The end result of the procedure is a model that allows prediction of group membership when only the interval variables are known. A second purpose of discriminant function analysis is an understanding of the data set, as a careful examination of the prediction model that results from the procedure can give insight into the relationship between group membership and the variables used to predict group membership.

LDA works when the measurements made on independent variables for each observation are continuous quantities. Discriminant analysis often produces models whose accuracy approaches (and occasionally exceeds) more complex modern methods.

0.1.2 Nearest Neighbor Method

The *k*-nearest neighbors algorithm (*kNN*) is an alternative nonparametric method and one the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its *k* nearest neighbors (*k* is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of its nearest neighbor. It is similar to kernel method with a random and variable bandwidth. The idea is to base estimation on a fixed number of observations *k* which are closest to the desired point.

Suppose $X \in \Re^d$ and we have a training sample $\{X_1, \ldots, X_n\}$ in a multidimensional feature space, each with a class label. The training phase of the algorithm consists only of storing the feature vectors and class labels of the training samples. For any fixed point we can calculate how close each observation X_i is to *x* using the Euclidean distance $\|x\| = (x\acute{}x)^{\frac{1}{2}}$. This distance is

$$D_i = \|x - X_i\| = [(x - X_i)\acute{}(x - X_i)]^{\frac{1}{2}} \qquad (5)$$

In the classification phase, *k* is a user-defined constant, and an unlabeled vector (a query or test point) is classified by assigning the label which is most frequent among the *k* training samples nearest to that query point.

When *X* is multivariate the nearest neighbor ordering is not invariant to data scaling. Before applying nearest neighbor methods, is therefore essential that the elements of *X* be scaled so that they are similar and comparable across elements.

A drawback to the basic ”majority voting” classification is that the classes with the more frequent examples tend to dominate the prediction of the new vector, as they tend to come up in the k nearest neighbors when the neighbors are computed due to their large number. One way to overcome this problem is to weight the classification taking into account the distance from the test point to each of its k nearest neighbors.

0.2 Intelligent Statistical Methods

0.2.1 Competitive Neural Network

Competitive Networks have two layer fully connected, the input layer and the Kohonen or output layer and it is much like the Perceptron. The input layer is connected to a $1 - D$ or $2 - D$ output grid. During the training process, input data are introduced in the network through the processing elements (nodes or neurons) in the input layer. Associated with the output nodes in the Kohone layer there is a weight vector values. The weights on the connections are normally set to random positive values in the range 0 to 1. (There may be other conditions on the initial weights too.) The neurons of competitive networks learn to recognize groups of similar input vectors. The weights of the winning neuron are adjusted with the Kohonen learning rule [26]. The weight vector (w) is changed for a given neuron if the output is not equal to zero (the winning neuron), according to: $dw = \alpha(x - w)$, where x is the neuron's input, and α the learning rate. Thus, the neuron whose weight vector was closest to the input vector is updated to be even closer. The result is that the winning neuron is more likely to win the competition the next time a similar vector is presented, and less likely to win when a very different input vector is presented. The training stage stops when any of the following conditions are met: the maximum number of epochs (an epoch is a presentation of all input patterns) is reached, the performance has been minimized to the goal, or a maximum amount of time has been exceeded. One of the limitations of competitive networks is that some neurons may not always get allocated, that is, some neuron weight vectors may start out far from any input vectors and never win the competition, no matter how long the training is continued. To stop this, use biases to give neurons that only win the competition rarely (if ever) an advantage over neurons that win often. This has two good effects. First, if a neuron never wins a competition because its weights are far from any of the input vectors, its bias eventually becomes large enough so that it can win. The second advantage of biases is that they force each neuron to classify roughly the same percentage of input vectors.

0.2.2 The self-organizing map algorithm

The basic SOM consists of M neurons located on a regular low dimensional grid, usually 1 or 2 dimensional. The lattice of the grid is hexagonal, rectangular or random.

The neurons represent the inputs with reference vectors m_i, whose components correspond to synaptic weights. One reference vector is associated with each neuron called unit. The unit, indexed with *c*, whose reference vector is nearest to the input *x* wins the competition, according to Eq.(6):

$$c\left(x\right)=argmin\left\{\left\|x-m_i\right\|^2\right\}; \tag{6}$$

where the most usually used is the Euclidean metric, although others are possible as well.

The winning unit and its neighbors adapt to represent the input even better by modifying their reference vectors towards the current input. The amount the units learn will be governed by a neighborhood kernel *h*, which is a decreasing function of the distance of the units from the winning unit on the map lattice. If the locations of units *i* and *j* on the map grid are denoted by the two-dimensional vectors r_i and r_j, respectively, the instantaneous neighborhood kernel is given by Eq. (7):

$$h_{ij}\left(t\right)=h\left(\left\|r_i-r_j,t\right\|\right), \tag{7}$$

with t denoting time.

During the learning process at time *t*, the reference vectors change iteratively according to the adaptation rule given by Eq. (8), where *x(t)* is the input at instant *t*, and *c = c(x,t)* is the index of the winning unit:

$$m_i\left(t+1\right)=m_i\left(t\right)+h_{ci}\left(t\right)\left[x\left(t\right)-m_i\left(t\right)\right], \tag{8}$$

In practice, the neighborhood kernel is chosen to be wide in the beginning of the learning process, to guarantee the global ordering of the map, and both its width and height decrease slowly during the learning process, which consists of the winner's selection by Eq. (6), and the adaptation of the synaptic weights via Eq. (8). Consequently, learning can be modeled with an ANN structure in which the neurons are coupled by inhibitory connections. The main properties of such self-organizing maps can be stated as [27]:

- The distance relationships between the input data are preserves by their images in the map as faithfully as possible. While some distortion is unavoidable, the mapping preserves the most important neighborhood relationships between the data items, i.e., the topology of their distribution.
- The map allocates different numbers of nodes to inputs based on their occurrence frequencies. If different input vector appear with different frequencies, the more frequent one will be mapped to larger domains at the expense of the less frequent ones.

0.2.3 Learning Vector Quantization Network

Learning vector quantization (LVQ) [28] is a special kind of supervised neural network method and has been used in a variety of problems due to its flexibility and conceptual clarity. This method is a nearest-neighbor pattern classifier based on competitive learning.

A *LVQ* network contains an input layer, a Kohonen layer which learns and performs the classification, and an output layer. The input layer contains one node for each input feature, the output layer contains one node for each class. During the training process of the LVQ network, Euclidean distances between the vectors of the training set compounds x_i and a smaller set of weight vectors w_j in the Kohonen layer are calculated according to the formula (9).

$$d_{ij}=\left[\sum_{k=1}^{P}\left(x_{ik}-w_{jk}\right)^2\right]^{1/2} \tag{9}$$

A subset of the weight vectors is first assigned to each class. The coordinates of the weight vector closest to the original vector (labeled as w_c) are adapted according to the LVQ updating rule so that the coordinates of the model vectors will eventually be characteristic of the vectors in each class. The nearest node is declared to be the winner, and its weight vector is adjusted according to whether the winning node is in the class of the training vector:

- If the winner is the correct class, then $w_{i+1}=w_i+\alpha\left(x-w_i\right)$
- If the winner is not the correct class, then $w_{i+1}=w_i-\gamma\left(x-w_i\right)$

where w_{i+1} is the weight vector after adjustment, w_i the vector before adjustment, α and γ are learning parameters.

0.2.4 Support Vector Machine

Support vector machines are a set of related supervised learning methods used for classification and regression. Two main issues of interest in *SVM* classifiers are the generalization performance and the complexity of classifier which they make this method more attractive than many other classification methods. First is due the performance in the test set and not in training set. For a *SVM* classifier, there is a guarantee of the upper error bound on the test set based on statistical learning theory. Complexity of classifiers is a practical implementation issue. For a *SVM*, the complexity of the classifier is associated with the so-called *VC* dimension.

Given a set of training examples, each marked as belonging to one of two categories, a *SVM* training algorithm builds a model that predicts whether a new example falls into one category or the other. More formally, support vector machine introduced by Vapnik [29] uses the concept of support vector (*SV*) methods to construct a optimal separating hyperplane or set of hyperplanes in a high or infinite dimensional space. A good separation is achieved by the hyperplane that has the largest distance to the nearest training data-points of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. An *SVM* classifier minimizes the generalization error on the test set under the structural risk minimization (SRM) principle.

Considering a binary classification task with a set of linearly separable training samples $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where *x* is the input vector such that $x \in \Re^d$ (*d*-dimensional input space) and *y* is the class label such as $y_i \in \{-1, 1\}$, the goal of training is to create a suitable discriminating function

$$f(x) = \langle w \cdot x \rangle + b \tag{10}$$

where *w* is the weight vector which determines the orientation of the hyperplane $f(x) = 0$ and *b* is the bias or offset. Eq. 11 carries out the separating the data across the choosing *w* and *b* for to maximize the margin, or distance between the parallel hyperplanes that are as far apart as possible.

$$\begin{array}{ll} \langle w \cdot x_i \rangle + b \geq 1, y_i = \;\; 1, & \\ & i = 1, 2, \ldots, n. \\ \langle w \cdot x_i \rangle + b \leq 1, y_i = -1, & \end{array} \tag{11}$$

For a linear SVM, the construction of discriminating function shown in Eq. 10 is difficult to solve because its depends on the norm of *w*. Then, altering the equation reduces the problem to resolve a quadratic programming optimization problem that is represented in Eq. 12.

$$\min_{w,b} \frac{1}{2} \|w\|^2 \tag{12}$$

been $y_i (\langle w \cdot x_i \rangle + b) \geq 1, \quad i = 1, \ldots, n.$

The minimization of linear inequalities is typically solved by the application of Lagrange duality theory. By introducing Lagrange multipliers α_i, Eq. 12 is reduced to

$$\min_{w,b} \max_{\alpha} \left\{ \frac{1}{2} \|w\|^2 - \sum_{i=1}^{n} \alpha_i \left[y_i (w \cdot x_i - b) - 1 \right] \right\} \tag{13}$$

Writing the classification rule in its unconstrained dual form reveals that the maximum margin hyperplane and therefore the classification task is only a function of the support vectors, the training data that lie on the margin. Then, the dual of the SVM reduces to the following optimization problem:

$$\max_{\alpha} \left\{ \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \right\} \tag{14}$$

with $\alpha_i \geq 0, i = 1, \ldots, n.$, and to the constraint from the minimization in *b*: $\sum_{i=1}^{n} \alpha_i y_i = 0$. The end term in the member second is the Kernel function that it is defined as

$$k(x_i, x_j) = x_i \cdot x_j \tag{15}$$

According to Karush-Kuhn-Tucker (*KKT*) conditions, solutions α^*, w^* and b^* must satisfy:

$$\alpha_i^* \left[y_i (\langle w^* \cdot x_i \rangle + b) - 1 \right] = 0, \quad i = 1, \ldots, n. \tag{16}$$

The corresponding vectors x_i, for $\alpha_i \neq 0$ are the so-called support vectors. These points lie on the hyperplane and affect the decision of the machine. Finally the relevant classifier function becomes

$$f(x) = sgn \left[\sum_{i=1}^{n} \alpha_i^* y_i k(x, x_i) + b^* \right] \tag{17}$$

where N_s is the number of support vectors x_i, $x_i \in SV$, and $k(x_i, x_j)$ is the selected kernel function.

The original optimal hyperplane algorithm proposed by Vladimir Vapnik in 1963 was a linear classifier. However, in 1992, Bernhard Boser, Isabelle Guyon and Vapnik suggested a way to create non-linear classifiers by applying the kernel trick to maximum-margin hyperplanes. The resulting algorithm is formally similar, except that every dot product is replaced by a non-linear kernel function. This allows the algorithm to fit the maximum-margin hyperplane in a transformed feature space. The transformation may be non-linear and the transformed space high dimensional; thus though the classifier is a hyperplane in the high-dimensional feature space, it may be non-linear in the original input space.

Some common kernels include:

- Polynomial (homogeneous):
  $k(x_i, x_j) = (x_i \cdot x_j)^d$
- Polynomial (inhomogeneous):
  $k(x_i, x_j) = (x_i \cdot x_j + 1)^d$
- Radial Basis Function :
  $k(x_i, x_j) = exp\left(-\gamma \|x_i - x_j\|^2\right)$, for $\gamma > 0$
- Gaussian Radial Basis Function :
  $k(x_i, x_j) = exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$

Data arrangement and experimental results

In order to obtain representative signals with the most common *PQ* disturbances, we have used *Matlab* $R2008b$ software to generate them. The generated disturbances are sags, swells and oscillatory transients (*OT*). A battery of these signals is generated with the following characteristics: the employed decaying laws are exponential, linear and parabolic, and each disturbance vanishes within less than half a cycle ($2ms$), corresponding to the power line. A sample frequency of $0.1ms$ is chosen and each time-series under test for computation performance contains eight cycles of the power-line. For the *OT* a frequency and amplitude swept have been done. The frequency range is 100 to 4500 to step of 100 Hz and the amplitude range is 0.05 to 0.9 to step of 0.01 V. On other hand, an amplitude swept of -0.9 to 0.9 to step of 0.01 V for sags and swells have been performed. Once the synthetic database has been obtained, we apply the proposed technique. Such technique consists two phases: feature extraction phase and classification phase.

0.3 Feature extraction phase

As mentioned above, the use of higher-order statistical (*HOS*) based on cumulants seems to be a very promising approach for disturbance detection in voltage signals because they are more appropriate for dealing with Gaussian signals.

The signal analysis is performed using an initial condition which helps to detect disturbances in the sinusoidal power line. This condition states that the sliding window used to extract *HOS* features must enclose an exact number of cycles of the 50-*Hz* power line. In this work the analysis will be based on windows of 0.02 seconds which cover 1 cycle. Thus, displacing the sliding window along a healthy signal, the set of values analyzed by the *HOS* processing is identical, returning an specific constant value. Any disturbance or anomaly in the healthy signal will produce variations from this constant value, thereby revealing its existence.

The first step to detect disturbances in the power line is the definition of a healthy signal. In the Fig. 1 can be observed that a pure sinusoidal signal with a normalized amplitude (healthy signal) is associated to the following constant values for the *HOS* estimators: 0.5 for the variance, 0 for the skewness, -1.5 for the kurtosis, 0 for the fifth-order standard moment and 2.4813 for the sixth-order standard moment. The simple characterization of healthy signals, according to the proposed methodology is this phase, helps to detect any coupled disturbance as a deviation in the *HOS* values.

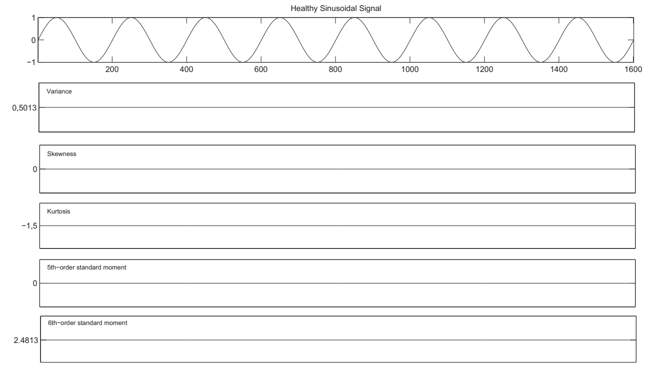

Fig. 1. Characterization of *HOS* values for healthy sinusoidal signal.

Then, for each register the second-, third-, fourth-, fifth- and sixth-order cumulants are computed according to the following procedure. For a given statistical order the feature extraction algorithm computes a cumulant over the sliding window, and then it jumps to the following starting point (overlapping 98%). Besides, each nth-order cumulant, $Cum_{n,x}\left[i\right]$, associated to the ith computation window has been normalized by $Cum_{2,x}\left[i\right]^2$, so that to give a real statistical characterization. When the computation sweep along a signal is finished, the maximum and the minimum are calculated for all the sliding windows results. Consequently, each signal is characterized by its second, third, fourth, fifth and sixth-order extremes.

This procedure can be showing in the Fig. 2 where it display the results of the proposed *HOS* analysis performed on a synthetic, which comprises the three studied disturbance kinds. The perturbations introduced in the healthy signal modify the *HOS* values of them, such it can be observed.

0.4 Classification phase

Once the signal characteristics have been obtained, which will help us in the classification process, a database were synthetically generated. This database consists of $30,600$ samples including the three studied disturbance kinds and it is composed as follows: $10,460$ *OT* signals, $10,080$ sag signals and $10,060$ swell signals. Finally we have a matrix with $30,600$ samples with three classes (corresponding to perturbation types) and 10 dimensions (corresponding to *HOS* values).

Then we apply different classification techniques as mentioned above in the section . As commented before, the techniques were divided in two categories: traditional statistical methods and intelligent statistical methods. As each studied technique has special features, we proceed to explain each of them. The algorithm used to select the training and test sets is common to all these methodologies. This algorithm is the double crossvalidation which selects two sets of available data, one for training and another for test.

The used techniques in the first category are: linear discriminant analysis (*LDA*) and nearest neighbor methods (*kNN*).

- The *LDA* performs the classification of test data in a training group. Since this technique only allows the classification in two groups, we divide the matrix of 10 dimensions in 5 vectors of 2 dimensions each. The principal division of these vectors has been the following: (minimum variance, maximum variance), (minimum skewness, maximum skewness), (minimum kurtosis, maximum kurtosis), (minimum 5 th-order standard moment, maximum 5 th-order standard moment) and (minimum 6 th-order standard moment, maximum 6 th-order standard moment). Further, the permutation of these vectors has made to carry out the classification. With each permutation we have 5 structures and after the simulations have been done, we obtain the correlation rate and mistake rate for each of them. The mean value of the five obtained values will be calculated for each permutation.
- The method *kNN* also classifies the test data into groups based on the grouping of training data and thereby we have followed the same assumptions given in the first method.

After that all possible permutations have been realized in the techniques previously seen, the method configuration with the best obtained average in each of them is selected.

Now we going to continuous with the used techniques belonging to intelligent statistical methods. The used order has been the follows: Competitive Neural Networks (*CNN*), Self-Organizing Feature Maps (*SOFM*), Learning Vector Quantization Networks (*LVQ*) and Support Vector Machine (*SVM*).

- The used parameters in *CNN* were the follows: number of neurons equal to 3, kohonen learning rate of 0.01, conscience learning rate of 0.001 and number of epochs equal to 5. Since this tool only works on a two-dimensional space, we have followed the same process of separation made in the above methods. So we have a composed classification system of 5 *CNN* networks. For each network the correlation rate and mistake rate were calculated. Finally we calculate the average of these five values.
- The parameters of *SOFM* are: size of *i*th layer dimension [3 x 1], topology function '*randtop*', distance func-

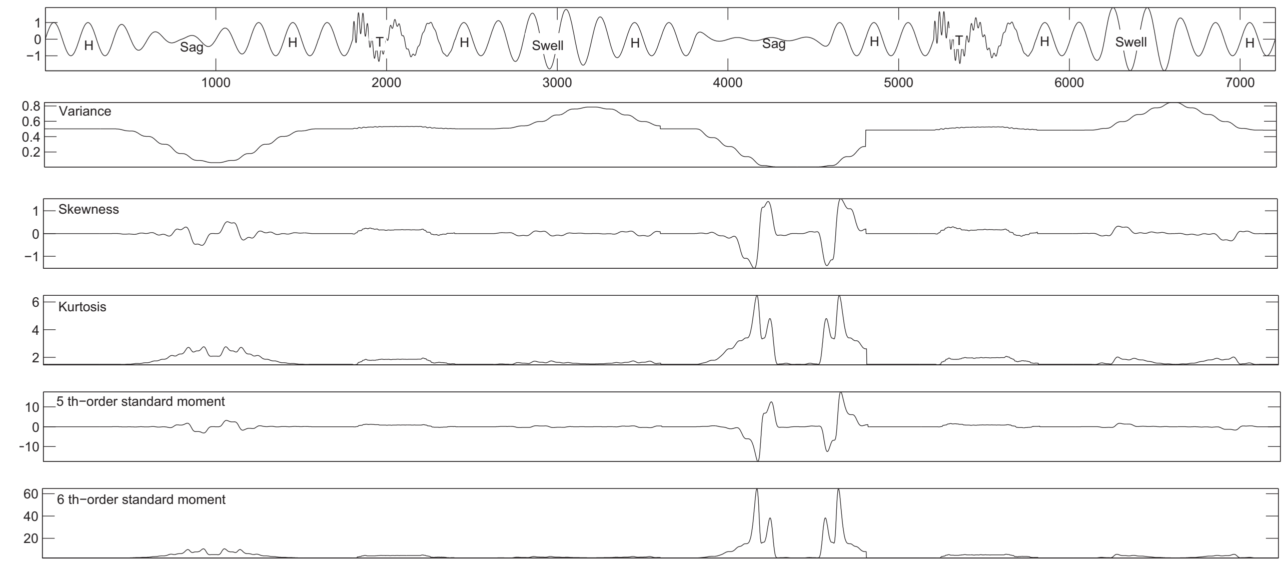

Fig. 2. On site measurement simulation which shows the evolution of the constant values associated to six statistics (variance, skewness, kurtosis, fifth-order standard moment and sixth-order standard moment).

tion '*linkdist*', 100 steps for neighborhood to shrink to 1, initial neighborhood size of 3 and number of epochs equal to 500. In this method the same procedure given above has been followed, so that we again obtain a composed classification system of 5 *SOFM* networks. In each of these networks the correlation rate and mistake rate were calculated. Several topology and distance functions to perform various simulations have been selected and the average value of these five obtained values in each simulation has been calculated.

- The introduced parameters in *LVQ* have been: number of hidden neurons equal to 15, vector of typical class percentages and learning rate of 0.1. The used training and test sets are selected from the matrix of 10 dimensions. This technique uses a composed system of 3 *LVQ* networks, one for each studied disturbance, and the correlation rate and mistake rate were calculated in each of them. The average value of these three obtained values for each experiment were calculated.
- The adjustable parameters of *SVM* are: kernel function that maps the training data into kernel space, the method sequential minimal optimization to find the separating hyperplane and box constraint of $1,000$ for the soft margin. The available kernel functions are: '*linear*', '*quadratic*', '*gaussian radial basis*', '*polynomial*' and '*multilayer perceptron*'. In the Gaussian radial basis function kernel a scale factor (*sigma*) of 0.7 is specified. For the polynomial kernel a 5*th*-degree polynomial is chosen and the scale and bias parameters in the multilayer perceptron kernel are selected to $[0.7, -1]$. As in the previous used method a composed system of three *SVM* networks is used and the correlation rate and mistake rate in each of them were calculated. Again the used training and test sets are selected from the matrix of 10 dimensions. We selected several kernel functions to perform various simulations and the average value of these three obtained values in each experiment have been calculated.

Table 1. The correlation and mistake rates of all used methods.

Model	Mean Correct Rate	Mean Error Rate
LDA	0.7783	0.2217
kNN	0.9291	0.0709
CNN	0.6408	0.3592
SOFM	0.6321	0.3679
LVQ	0.7633	0.2367
SVM	0.9994	$6.3667e^{-4}$

As seen of the first group methods, the methods configurations with better average in each of them have been selected. Table 1 shows the correlation and mistake rates of selected methods in both groups.

Conclusion

In this work an automatic detection system and classification of three kinds of *PQ* disturbance has been developed. The primary question considered to carry out the design of a classifier is the choice of appropriate features. Proper selection of features will make a more precise learning process in the automatic systems.

The features extraction phase proposed in this work is built on *HOS*. The obtained statistics in this phase are as follows: variance, skewness, kurtosis, 5 th-order standard moment and 6 th-order standard moment. The signal processing methodology is based on a tuned 50-*Hz* sliding window over which compute each statistical estimator; thus processing the 50-*Hz* events produces constant values to detect the occurrence and the start and end point of events in voltage signals. This tuned window allows signal analysis without preprocessing, i.e. preserving all the information and without introducing nonlinear behaviors.

Classifiers belonging to two categories have been discussed in detail: traditional statistical methods and intelligent statistical methods. It has been shown that results very satisfactory have been produced by the belonging methods to first category, and the nearest neighbor method has given the better average correlation rate of 92.91%. On the other hand, the classifiers belonging to the second category have produced results not as good as in the first category. Only one of them, *SVM*, has far exceeded obtaining an average correlation rate of 99.94%. This is because this one is a robust statistical classification method. In contrast, the cluster formation in *CNN* and *SOFM* methods is strongly dependent on the distridution of data, so that we can not control where they put the neurons that act as centroids of the selected clusters according to the anomalies types that will want to classify.

The advantage of statistical classifiers is obvious as they optimize the performance, which is especially important for linearly nonseparable classes, as is often the case in the real world.

Future work is designed with twofold purpose: First the objective is to deal with different types of *PQ* anomalies, like flicker and harmonics. Secondly, we will focus on a more precise learning process using hybrid learning systems, fuzzy logic and genetic algorithms or other type of *ANN* structure based in *backpropagation* networks.

Acknowledgement

The authors would like to thank the *Spanish Ministry of Science and Innovation* for funding the research projects $TEC2009\text{-}08988$ and $TEC2010\text{-}19242\text{-}C03\text{-}03$ (*SIDER-HOSAPQ*). Our unforgettable thanks to the trust we have from the *Andalusian Government* for funding the Research Group PAIDI-TIC-168 in *Computational Instrumentation and Industrial Electronics-ICEI*.

BIBLIOGRAPHY

[1] M. V. Ribeiro, C. A. G. Marques, C. A. Duque, A. S. Cerqueira, J. L. R. Pereira, Detection of disturbances in voltage signals for power quality analysis using *HOS*, Journal on Advances in Signal Processing (2007) 1–13.

[2] M. McGranaghan, Trends in power quality monitoring, IEEE Power Engineering Review 21 (10) (2001) 3–9.

[3] A. Moreno, *et al*, Mitigation Technologies in a Distributed Environment, 1st Edition, Power Systems, Springer-Verlag, 2007.

[4] H. T. Yang, C. C. Liao, A de-noising scheme for enhancing wavelet-based power quality monitoring system, IEEE Transactions on Power Delivery 16 (3) (2001) 353–360.

[5] S. Santoso, W. M. Grady, E. J. Powers, J. Lamoree, S. C. Bhatt, Characterization of distribution power quality events with fourier and wavelet transform, IEEE Transactions on Power Delivery 15 (1) (2000) 247–254.

[6] A. M. Gaouda, M. A. Salama, M. R. Sutan, A. Y. Chikhani, Power quality detection and classification using wavelet-multiresolution signal decomposition, IEEE Transactions on Power Delivery 14 (4) (1999) 1469–1476.

[7] C. A. Duque, M. V. Ribeiro, F. R. Ramos, J. Szczupak, Power quality event detection based on the divide and conquer principle and innovation concept, IEEE Transactions on Power Delivery 20 (4) (2005) 2361–2369.

[8] M. Karimi, H. Mokhtari, M. R. Iravani, Wavelet based on-line disturbance detection for power quality applications, IEEE Transactions on Power Delivery 15 (4) (2000) 1212–1220.

[9] H. Mokhtari, M. Karimi, M. R. Iravani, Experimental performance evaluation of a wavelet-based on-line voltage detection method for power quality applications, IEEE Transactions on Power Delivery 17 (1) (2002) 161–172.

[10] C. L. Nikias, J. M. Mendel, Signal processing with higher-order spectra, IEEE Signal Processing Magazine 10 (3) (1993) 10–37.

[11] J. J. G. De la Rosa, A. Moreno, C. G. Puntonet, A practical review on higher-order statistics interpretation. application to electrical transients characterization, Dynamics of continous discrete and Impulsive Systems-Series B: Applications and Algorithms 14 (4) (2007) 1577–1582.

[12] Ömer Nezih Gerek, D. G. Ece, Power-quality event analysis using higher order cumulants and quadratic classifiers, IEEE Transactions on Power Delivery 21 (2) (2006) 883–889.

[13] J. J. G. De la Rosa, I. Lloret, C. G. Puntonet, R. Piotrkowski, A. Moreno, Higher-order spectra measurement techniques of termite emissions. a characterization framework, Measurement (Ed. Elsevier) 41 (1) (2008) 105–118.

[14] J. J. G. De la Rosa, R. Piotrkowski, J. Ruzzante, Third-order spectral characterization of acoustic emission signals in ring-type samples from steel pipes for the oil industry, Mechanical Systems and Signal Processing (Ed. Elsevier) 21 (Issue 4) (2007) 1917–1926.

[15] IEEE Recommended practice for monitoring electric power quality, Tech. Rep. IEEE Std. 1159-1995, The Institute of Electrical and Electronics Engineers, Inc. (1995).

[16] J. Chung, Power disturbance classifier using a rule-based method and wavelet packet-based hidden markov model, IEEE Transactions on Power Delivery 17 (1) (2002) 233–241.

[17] P. K. Dash, S. Mishra, M. M. A. Salama, A. C. Liew, Classification of power system disturbances using a fuzzy expert system and a fourier linear combiner, IEEE Transactions on Power Delivery 15 (2) (2000) 472–477.

[18] Z. L. Gaing, Wavelet-based neural network for power disturbance recognition and classification, IEEE Transactions on Power Delivery 19 (4) (2004) 1560–1568.

[19] J. Huang, M. Negnevitsky, D. T. Nguyen, A neural fuzzy classifier for recognition of power quality disturbances, IEEE Transactions on Power Delivery 17 (2) (2002) 609–616.

[20] S. Santoso, E. J. Powers, W. M. Grady, A. C. Parsons, Power quality disturbance waveform recognition using wavelet-based neural classifier part 2: application, IEEE Transactions on Power Delivery 15 (1) (2000) 229–235.

[21] E. Styvaktakis, M. H. J. Bollen, I. Y. H. Gu, Expert system for classification and analysis of power system events, IEEE Transactions on Power Delivery 17 (2) (2002) 423–428.

[22] A. M. Youssef, T. K. Abdel-Galil, E. F. El-Saadany, M. M. A. Salama, Disturbance classification utilizing dynamic time warping classifier, IEEE Transactions on Power Delivery 19 (1) (2004) 272–278.

[23] J. J. G. De la Rosa, I. Lloret, C. G. Puntonet, J. M. Górriz, Higher-order statistics to detect and characterise termite emissions, Electronics Letters 40 (20) (2004) 1316–1317, Ultrasonics.

[24] C. L. Nikias, A. P. Petropulu, Higher-Order Spectra Analysis. A Non-Linear Signal Processing Framework, Englewood Cliffs, NJ, Prentice-Hall, 1993.

[25] A. K. Nandi, Blind Estimation using Higher-Order Statistics, Vol. 1, Kluwer Academic Publichers, Boston, 1999.

[26] T. Kohonen, Self organized formation of topologically correct feature maps, Biological Cybernetics 43 (1) (1982) 59–69.

[27] A. Varfis, C. Versino, Clustering of socio-economic data with kohonen maps, Neural Network World 2 (1992) 813–834.

[28] T. Kohonen, Self-organizing maps, Springer Series in Information Sciencies 30.

[29] V. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag, New York, 1995.

Authors: Prof. José Carlos Palomares-Salas[1],
Prof. Dr. Juan José González de la Rosa[1],
Prof. Agustín-Agüera Pérez[1],
Prof. Dr. Antonio Moreno-Muñoz[2]

[1] Univ. of Cádiz. Department of Engineering in Automation, Electronics, Architecture and Computers' Networks. Engineering School of Algeciras - EPSA. Av. Ramón Puyol S/N. E-11202-Algeciras-Cádiz-Spain.

[2] Univ. of Córdoba. Campus de Rabanales. Ed. Leonardo da Vinci. E-14071-Córdoba-Spain

email: josecarlos.palomares@uca.es; juanjose.delarosa@uca.es; agustin.aguera@uca.es; amoreno@uco.es.