Residual Generation and Visualization for Understanding Novel Process Conditions

Ignacio Díaz[1] and Jaakko Hollmén[2]

[1]Univesity of Oviedo. Area de Ingeniería de Sistemas y Automática
Campus de Viesques, Edif. 2, 33204, Gijón, Spain
e-mail: idiaz@isa.uniovi.es
[2]Helsinki University of Technology. Laboratory of Computer and Information Science
P.O. Box 5400, FIN-02015 HUT, Finland
e-mail: Jaakko.Hollmen@hut.fi

***Abstract*— We study the generation and visualization of residuals for detecting and identifying unseen faults using autoassociative models learned from process data. Least squares and kernel regression models are compared on the basis of their ability to describe the support of the data. Theoretical results show that kernel regression models are more appropriate in this sense. Moreover, experiments on vibration and current data from an asynchronous motor confirm the theory and yield more meaningful results.**

***Index Terms*—Residual generation, visualization, novelty detection, fault identification, kernel regression**

I. Introduction

VISUALIZATION and dimension reduction techniques can be very powerful tools to analyze large sets of multidimensional data and have received considerable attention in recent years [15], [13]. Particularly, in supervision and condition monitoring of complex processes, visualization methods based on radial basis function (RBF) networks [19], the generative topographic mapping (GTM) [3] or the self organizing map (SOM) [9], [16], [1], [5] have been proposed. All these techniques exploit the statistical information present in the data, capturing nonlinear relationships among the variables to build a model of the process data geometry using a low dimension manifold which allows to summarize its behavior in a few dimensions (typically 2D or 3D), being in this sense generalizations of the well known linear PCA methods. They have proven to be extremely useful in providing insightful views of the process, allowing to merge in a very efficient way information conveyed by the data with prior knowledge and experience by means of visualizations which take advantage of preattentive abilities of the human brain [7], [8].

Unfortunately, models built using these techniques only describe process behavior present in the data from which they were learned and hence they cannot explain conditions outside those present in the learning data. However, the fact that those methods explain part, maybe most, of the behavior of the process can be thought of as "work already done". *Residuals*, or innovations, can be defined as that part of the process data which is not explained by the model. A huge amount of literature has been devoted to the analysis and design of residuals, specially in the field of control of dynamical systems —see e.g. [6] for a good review. Nevertheless, most of the work in this field has been focused on residuals generated by physical or mathematical dynamical models, but much less work has been devoted to residuals or innovations generated by models learned from data. This work focuses on the latter problem.

This paper is organized as follows. In Section II we outline general concepts of autoassociative models, which are used subsequently for residual generation. In Sections III and IV, we present two autoassociative approaches for residual evaluation –kernel regression and a least squares version of it– which are similar in architecture but different in their nature. We show that the former methods take into account the support of the process data, while the latter aim to minimize the squared error in a global fashion, and we show how this can result in the implicit assumption of an unfair joint pdf in the process data, and hence in meaningless residuals. These ideas are illustrated with artificial data. In Section V we describe a simple and intuitive method to visualize the residuals and finally, in Section VI, we discuss both approaches applying the visualization method to a real case study of vibration and current data from an asynchronous motor.

II. Autoassociative Models

A. General Concepts

One way to describe the behavior of a process is the use of *autoassociative models*. A neural network can be trained to generate a map from the input space on itself, in such a way that the outputs are as close as possible to the inputs. Obviously, the raw use of this approach often leads to trivial solutions. An example of this arises if a linear mapping $\mathbf{T} : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is chosen. In this case, the minimization of the cost functional $J(\mathbf{T}) = \|\mathbf{x} - \mathbf{T}\mathbf{x}\|^2$, leads to the trivial solution $\mathbf{T} = \mathbf{I}$.

The usefulness of the autoassociative models, however, relies in bounding the complexity of the mapping. This can be done by imposing some restrictions on the class of functions used to build the mapping as, e.g., a "bottleneck" layer in autoassociative multilayer perceptrons, or in the cost function to be minimized. These restrictions, in general, aim to bound the complexity of the approximation, according to the principle of *Occam's Razor*. This is further motivated by the fact that the *general principles* according to which a physical process

evolves are essentially simple and often far from arbitrary. Indeed, this is a way in which *prior knowledge* is implicitly taken into account in autoassociative models to capture the essential physical substrate behind the data. However, despite good results [11], [2] attention is seldom paid to this in the design of autoassociative models. In next section we describe a statistical interpretation of autoassociative mappings which can help to deal with this problem in a more principled way.

B. Statistical Interpretation of Autoassociative Mappings

The problem in autoassociative computation of residuals can be seen as the estimation of $E[\mathbf{x}'|\mathbf{x}]$[1]. This quantity reduces to the trivial solution

$$E[\mathbf{x}'|\mathbf{x}] = \mathbf{x}, \quad \forall \mathbf{x} \in S \subset \mathbb{R}^n \tag{1}$$

where $S = \{\mathbf{x} \in \mathbb{R}^n | \, p(\mathbf{x}) \neq 0\}$ is the *support* of the random variable $\mathbf{x}$. Outside S, i.e. where $p(\mathbf{x}) = 0$, this quantity is not defined, as long as no outcomes of the distribution can lie outside S. This expectation is given by

$$E[\mathbf{x}'|\mathbf{x}] = \int \mathbf{x}' p(\mathbf{x}'|\mathbf{x}) d\mathbf{x}' = \int_S \mathbf{x}' \frac{p(\mathbf{x}, \mathbf{x}')}{p(\mathbf{x})} d\mathbf{x}' \tag{2}$$

where $p(\mathbf{x}, \mathbf{x}')$ is the joint pdf of the data in the augmented space $\mathbb{R}^n \times \mathbb{R}^n$ and is given by

$$p(\mathbf{x}, \mathbf{x}') = p(\mathbf{x}) \delta(\mathbf{x} - \mathbf{x}') \tag{3}$$

where $\delta(.)$ is the Dirac delta. The support of (3) is only a reduced subset of the augmented space given by $S \odot S \equiv \{(\mathbf{x}, \mathbf{x}') \in \mathbb{R}^n \times \mathbb{R}^n | \, \mathbf{x} = \mathbf{x}', \, \forall \mathbf{x}, \mathbf{x}' \in S\}$. Note also that the integrand at the rightmost part of Equation (2) is not defined for values of $\mathbf{x}$ outside the support S. This example, however, is an extreme case. When uncertainties as e.g. noise, arise in the available process data, zero densities outside a given support become unfair. A more general expression, which accounts for observation noise and is defined along the whole augmented space $\mathbb{R}^n \times \mathbb{R}^n$ is,

$$p(\bar{\mathbf{z}}) = \int p(\bar{\mathbf{x}}) p(\bar{\mathbf{z}}|\bar{\mathbf{x}}) d\bar{\mathbf{x}} = \int p(\bar{\mathbf{x}}) \frac{1}{(2\pi)^n \sigma^{2n}} e^{-\frac{\|\bar{\mathbf{z}} - \bar{\mathbf{x}}\|^2}{2\sigma^2}} d\bar{\mathbf{x}} \tag{4}$$

The previous expression, where $\bar{\mathbf{z}} \equiv (\mathbf{z}, \mathbf{z}')$ is the noisy observation of the real outcome of the process $\bar{\mathbf{x}} \equiv (\mathbf{x}, \mathbf{x}')$, is a more realistic one, accounting for Gaussian observation noise of variance σ^2, and its support being the whole augmented space. Note also that it reduces to (3) as a special case when $\sigma \to 0$. The conditional expectation $E[\mathbf{z}'|\mathbf{z}]$ obtained using (4) is now defined for all values of $\mathbf{z}$, allowing to describe the best expectation of points which lie outside the support of the original distribution with the only assumption of observation noise.

[1] As in autoassociative mappings the input and output space are the same we use the apostrophe $\mathbf{x}'$ to highlight the role of $\mathbf{x}$ as *output* vector. In general mappings, output vector will be denoted by $\mathbf{y}$.

III. Kernel Regression and Least Squares Approaches

A. Kernel Regression Mapping

Suppose that $\mathbf{x} \in \mathbb{R}^p$ and $\mathbf{y} \in \mathbb{R}^q$ are related by, $\mathbf{y} = f(\mathbf{x}) + \eta$, where η is some spherical noise in $\mathbb{R}^q$. The kernel regression estimate, rediscovered by Specht [14], allows to estimate

$$\hat{f}(\mathbf{x}) = E[\mathbf{y}|\mathbf{x}] = \int_{\mathbb{R}^q} \mathbf{y} p(\mathbf{y}|\mathbf{x}) d\mathbf{y} \tag{5}$$

The computation of the previous conditional expectation requires the knowledge of $p(\mathbf{y}|\mathbf{x})$. Specht obtains it on the basis of the Parzen kernel estimation of $p(\mathbf{x}, \mathbf{y})$ in the augmented space $\mathbb{R}^p \times \mathbb{R}^q$ which relies on similar noise hypothesis for the available samples as those used in previous section for equation (4), to come to the following expression

$$\hat{\mathbf{y}}(\mathbf{x}) = \frac{\sum_{i=1}^n \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2\sigma^2}\right) \mathbf{y}_i}{\sum_{i=1}^n \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{2\sigma^2}\right)} = \frac{\sum_i \Phi(\mathbf{x} - \mathbf{x}_i) \mathbf{y}_i}{\sum_i \Phi(\mathbf{x} - \mathbf{x}_i)}. \tag{6}$$

This class of approximation is a special case of the well known Nadaraya-Watson regression estimate [10][17], and is also often called generalized regression neural network (GRNN) [14].

B. Least Squares Mapping

A closely related type of mapping, which serves us for comparison purposes, is the following least squares interpolation version of the GRNN,

$$\hat{\mathbf{y}}(\mathbf{x}) = \sum_i \psi_i(\mathbf{x}) \mathbf{w}_i \quad \text{where,} \quad \psi_i(\mathbf{x}) \equiv \frac{\Phi_i(\mathbf{x} - \mathbf{x}_i)}{\sum_k \Phi(\mathbf{x} - \mathbf{x}_k)} \tag{7}$$

where the weights $\mathbf{w}_i$ are obtained from the input and output data matrices $\mathbf{X}$ and $\mathbf{Y}$ using a pseudoinverse approach

$$\mathbf{W} = \Psi^+ \mathbf{Y} = (\Psi^T \Psi + \lambda \mathbf{I})^{-1} \Psi^T \mathbf{Y}, \tag{8}$$

where $(\Psi)_{ij} = \psi_i(\mathbf{x}_j)$ and λ is a regularizing factor which minimizes a squared error cost function penalized with a weight decay term [12], [20].

C. Computation of Residuals

The previous approaches can be used to build an autoassociative model of the process by using the available training data both as input and output. A measure of how well the process behavior fits to the model can be given through the evaluation of the *residual vector*, whose components are the differences or *residuals* between the actual feature vector $\mathbf{x}$ and its expected value $\hat{\mathbf{x}}$ according to the model

$$\mathbf{r} = \mathbf{x} - \hat{\mathbf{x}}. \tag{9}$$

Vector $\mathbf{r}$ has n components $r_1, r_2, \ldots, r_n$ which can be regarded as individual residuals for each of the process variables or features $x_1, x_2, \ldots, x_n$ in the process feature vector $\mathbf{x}$. Residuals r_i should be close to zero as long as the process

behaves according to the model, and different from zero, in other cases. In order for the residuals to yield insightful information on the process state other than a binary "fault/no-fault" information, we also require:

1) The set of residuals which deviate significantly from zero are in some way related to the nature of the abnormality.
2) The sign and magnitude of the deviation of each residual are related to the severity of the abnormality.

In following sections, we will compare approaches described in sections III-A and III-B, and will attempt to demonstrate why the kernel-based estimation, involving support information, is important in creating meaningful residuals.

IV. Comparison of Models

A. Underlying Distributions

If we arrange models (6) and (7) together

$$\text{GRNN:} \qquad \hat{\mathbf{y}}(\mathbf{x}) = \sum_i \psi_i(\mathbf{x})\mathbf{y}_i \tag{10}$$

$$\text{LS-GRNN:} \qquad \hat{\mathbf{y}}(\mathbf{x}) = \sum_i \psi_i(\mathbf{x})\mathbf{w}_i \tag{11}$$

we see that the models are identical except for the fact that the GRNN uses the output sample vectors $\mathbf{y}_i$ while the least squares version uses vectors $\mathbf{w}_i$ computed by (8) in weighting the kernel activations. However, while the former has been derived from within a density estimation framework, the latter comes from the minimization of a cost function. This allows us to view the least squares approach in terms of the underlying joint distribution that it implicitly assumes. From this point of view, it can be seen as a GRNN estimation of the conditional expectation which assumes a different underlying joint pdf. In Fig. 1 we show for both approaches the underlying pdf's and the autoassociative mapping in the augmented space $\mathbb{R}^1 \times \mathbb{R}^1$ for the following bimodal uniform distribution

$$p(x) = \begin{cases} 0.25, & -1 \leq x \leq 1 \\ 0.25, & 2 \leq x \leq 4 \\ 0, & \text{in other case} \end{cases} \tag{12}$$

It can be noticed that, while the kernel regression (by definition) takes an estimation of the original joint pdf in the augmented space, the LS method places *ad hoc* output points on convenience to achieve the minimum squared error goal. This results in generalizations of the underlying joint pdf which alter the geometrical information on the support of the data and can lead to meaningless residuals as will be shown later.

B. 2-D Artificial Data

A simple example was designed to better illustrate the differences between the approaches in terms of the interpretability of the residuals. For the example we chose a set of 137 points $\mathbf{x}_i \in \mathbb{R}^2$ grouped in four clusters with different sizes and shapes. Autoassociative mappings were developed using both approaches. The same bandwidth parameter $\sigma = 1$ was chosen for both methods, and a small value for the regularizing parameter $\lambda = 10^{-7}$ was chosen in the LS approach to avoid degenerate solutions, while preserving the least squares nature of the approximation.

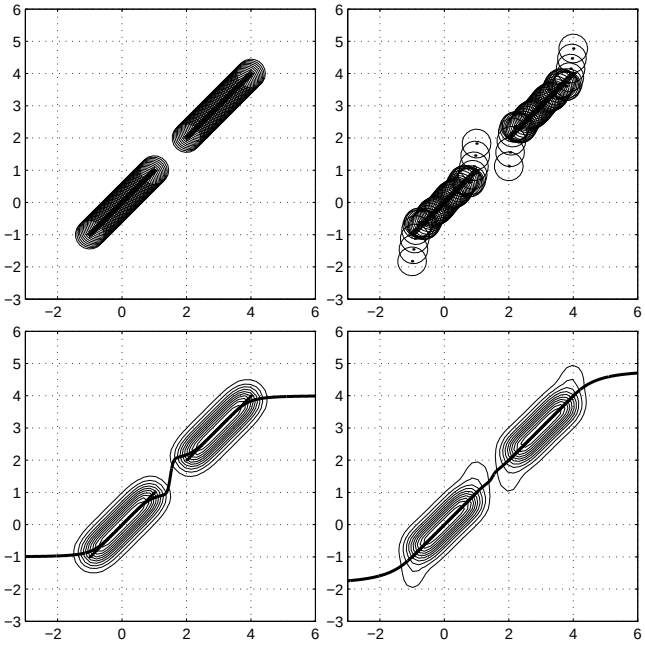


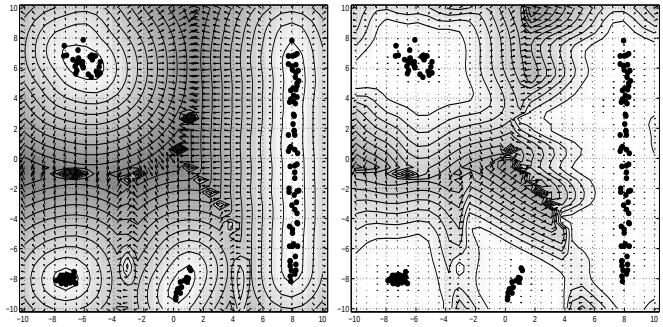
Fig. 1. Comparison in the augmented space $\mathbb{R}^1 \times \mathbb{R}^1$ of kernel (left) and LS (right) approaches for an autoassociative mapping on data of a bimodal uniform in $\mathbb{R}^1$. Note how the LS approach gets a functional mapping closer to the trivial solution $f(x) = x$ at the expense of an artificial underlying pdf.

Residual vectors for points $\mathbf{u}_i$ in a regular grid of 50×50 points covering the region $[-10, 10] \times [-10, 10]$ were obtained using both methods to compute their best expectations $\hat{\mathbf{u}}_i$. Residuals $\mathbf{r}_i$ were evaluated according to (9)

$$\mathbf{r}_i = \mathbf{u}_i - \hat{\mathbf{u}}_i \tag{13}$$

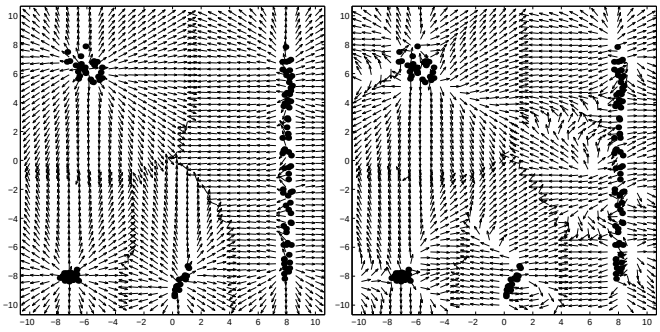
A qualitative judgment of the nature of residuals can be made by looking at the results shown in Figs. 2 and 3. In Figs. 2(a) and 2(b), the residual vectors for each point in the grid are shown. For the sake of clarity, the norm of the residuals was codified using gray levels and contour lines. It can be noticed that the norms of the residuals evaluated using the kernel approach fit quite well to the joint pdf of the data. The LS approach, in turn, yields residuals closer to zero, due to its better ability to approximate the identity mapping, but the values give no fair indication for the likelihood of the test points $\mathbf{u}_i$.

In Figs. 3(a) and 3(b), residuals have been normalized to unit length to allow comparison of their directions. In the kernel approach, arrows reflect quite well the sense of the deviation of each point with respect to its best expectation according to the pdf of the training sample. Note also that clusters of arrows emerge, revealing the areas of influence of each cluster of data. On the other hand, the LS model gives rise to areas in which the residuals take apparently arbitrary directions. Thus, for instance, in the area around $(x, y) = (2, -10)$, residual vectors produced by the LS method are almost horizontal and pointing



(a) Kernel approach. (b) LS approach.

Fig. 2. Residual vectors $\mathbf{r}_i$ and contour lines of the norms $\|\mathbf{r}_i\|$ for the kernel and LS approaches.



(a) Kernel approach. (b) LS approach.

Fig. 3. Normalized residuals $\frac{\mathbf{r}_i}{\|\mathbf{r}_i\|}$ for the kernel and LS approaches.

towards the nearby cluster. This means that residuals tell that the $\mathbf{u}$ points lying in this area, are to the left of their expected value, while it can be seen that they lie to the right of the closest cluster. While we could try to find explanations to these artifacts, it is clear that they give us a nonintuitive idea of the direction of the deviation.

C. Statistical interpretation

A strong motivation for the use of kernel approaches can be given through some theoretical results. In [18], Webb shows that the problem of minimizing the following cost function

$$V = \int \|\mathbf{f}(\mathbf{x}) - \mathbf{g}(\mathbf{z})\|^2 p(\mathbf{z},\mathbf{x}) d\mathbf{x} d\mathbf{z} = \quad (14)$$

$$= \int \|\mathbf{f}(\mathbf{x}) - \mathbf{g}(\mathbf{z})\|^2 p(\mathbf{z}|\mathbf{x}) p(\mathbf{x}) d\mathbf{x} d\mathbf{z} \quad (15)$$

where $p(\mathbf{x})$ denotes the pdf of the input data points $\mathbf{x}$, $\mathbf{g}(\mathbf{x})$ denotes the approximation function to $\mathbf{f}(\mathbf{x})$ qand $\mathbf{z}$ is a noise-corrupted version of $\mathbf{x}$ with probability $p(\mathbf{z}|\mathbf{x})$ is equivalent to obtaining

$$\mathbf{g}(\mathbf{z}) = \int \mathbf{f}(\mathbf{x}) p(\mathbf{x}|\mathbf{z}) d\mathbf{x} = E[\mathbf{f}(\mathbf{x})|\mathbf{z}] \quad (16)$$

i.e. the expected value of $\mathbf{f}(\mathbf{x})$ given the noisy input observation $\mathbf{z}$. The conditional probability $p(\mathbf{z}|\mathbf{x})$ represents indeed a noise model in the observation of inputs as long as, given a fixed outcome of the process $\mathbf{x}$, the observed value of $\mathbf{z}$ is a random variable defined by $p(\mathbf{z}|\mathbf{x})$. It can be easily seen that the kernel regression method (GRNN) is a special case of it where the noise model is a spherical Gaussian distribution:

$$p(\mathbf{z}|\mathbf{x}) = \Phi(\mathbf{z} - \mathbf{x}) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}\|\mathbf{z} - \mathbf{x}\|^2\right) \quad (17)$$

So, the kernel based methods, minimize

$$\int \|\mathbf{f}(\mathbf{x}) - \mathbf{g}(\mathbf{z})\|^2 \Phi(\mathbf{z} - \mathbf{x}) p(\mathbf{x}) d\mathbf{x} d\mathbf{z} \quad (18)$$

The previous cost function represents the error in the approximation weighted by a smoothed version of the probability distribution $p(\mathbf{x})$ achieved by convolution with spherical noise

$$\hat{p}(\mathbf{x}) = \int \Phi(\mathbf{z} - \mathbf{x}) p(\mathbf{x}) d\mathbf{z} \quad (19)$$

so, the above cost function can be shortened to

$$\int \|\mathbf{f}(\mathbf{x}) - \mathbf{g}(\mathbf{z})\|^2 \hat{p}(\mathbf{x}) d\mathbf{x} \quad (20)$$

If we envisage $\hat{p}(\mathbf{x})$ as a measure of belongingness to the input data support S, expression (20) shows up clearly the support-based nature of kernel based regression estimators for residual computation. However, the pure least squares methods bypass the support kernel $\hat{p}(\mathbf{x})$ in (20).

V. Visualization of Residuals

In previous sections we suggest that kernel based autoassociative mappings yield residuals which are more plausible from a statistical point of view, as they take into account the support of the input data. We also showed through a simple experiment how this also results in more sensible and interpretable residuals, both with respect to their norm and direction. If residuals are considered componentwise,

$$\mathbf{r} = (r_1, \ldots, r_n)^T \quad (21)$$

interpretability in norm and direction means also interpretability in the magnitude and the sign of deviation for each component r_i. Under normal (training) conditions, the residual vector will yield values close to zero as long as data input to the model come from the same distribution of training data. When an abnormal state occurs, process data lie outside the support of the training set, and some residuals deviate significantly from zero.

With a suitable visualization, kernel based residual vectors can result in a very useful tool, not only for fault detection, but also for fault identification. Preattentive capabilities of the human visual system can notice very quickly, in fractions of second changes in colors and shapes [7], [8]. This suggests the following visualization procedure:

Step 1: For each time t, compute the residual vector $\mathbf{r}(t) = \mathbf{x}(t) - \hat{\mathbf{x}}(t)$

Step 2: Build a matrix with the k last residual vectors $\mathbf{U}(t) = [\mathbf{r}(t-k+1), \ldots, \mathbf{r}(t)]$

Step 3: At each time t, visualize the elements of matrix $\mathbf{U}(t)$ in an image, using a color scale for the values of the elements of $\mathbf{U}$

In this visualization, color scales are preferable to gray levels from a perceptual point of view. Good choices for color scales are rainbow scales (e.g. a smooth hue transition through blue, cyan, green, yellow and red). This simple visualization method has the following advantages:

- While the process works in normal condition, all the window has the same color, e.g. green. This allows to remove from the representation possible states of the process which are considered normal.
- When one or more residuals deviate from zero, user can assess in fractions of second the sign of each deviation, e.g. blue for negative, and red for positive.
- Also magnitude can be assessed instantaneously, e.g. yellow, orange, red from low to high positive changes.
- Visual information of all the residuals and their history within the selected time window is deployed in a single image. The technician can quickly assess the situation using his prior knowledge about the process.
- Recent history of the process is displayed within the window. This can provide useful time-related information, such as periodicities, trends, etc.

VI. Experimental Results

A. Experimental Setup

To demonstrate the ideas exposed in this work, an experiment was carried out on a 4 kW, 2 pole-pair asynchronous motor. Two kind of faults were induced in the motor:

- *Asymmetry in the power supply*. This fault condition is provoked by the inclusion of a variable resistance R on a phase line. This produces an unbalance in the power supply modifying gradually the vibration and current patterns.
- *Mechanical asymmetry*. This fault condition is provoked by the presence of an asymmetric mass m on the axis.

The combination of both types of faults yields several operating conditions. Five sensors were installed in the motor: three vibration accelerometers $a_{bear}(t), a_x(t), a_y(t)$ and two current sensors $i_r(t), i_s(t)$. Data acquisition of the five channels was carried out at 5000 Hz using an acquisition board, after a signal conditioning stage.

B. Feature Extraction

Data were grouped into overlapped windows of 4096 elements to allow for FFT computation of the harmonics. It is known that mass asymmetries in rotating machinery are related to the frequency content at $1\times$ the rotating frequency (25 Hz) in the acceleration signals. Also, under power supply unbalance, harmonics at twice the power supply frequency are modified. Thus, for the vibration feature extraction, spectral energies at 25 Hz and 100 Hz in $a_{bear}(t), a_x(t), a_y(t)$ were computed. For the analysis of currents we used the Park vector approach [4]. It allows to summarize in a single complex sequence the behavior of the three phase supply currents

$$i(t) = i_r(t) + \mathbf{a} \cdot i_s(t) + \mathbf{a}^2 \cdot (-i_r(t) - i_s(t)) \qquad (22)$$

where $\mathbf{a} = \exp(j\frac{2\pi}{3})$ and $j = \sqrt{-1}$. A total of eight features were taken in order to visualize the motor's condition:

$$\begin{aligned}\mathbf{x} &= (x_1, \ldots, x_8)^T = \\ &= (a_{bear}^{(25\text{Hz})}, a_{bear}^{(100\text{Hz})}, a_x^{(25\text{Hz})}, a_x^{(100\text{Hz})}, a_y^{(25\text{Hz})}, a_y^{(100\text{Hz})}, \\ &\quad i^{(-50\text{Hz})},\ i^{(50\text{Hz})})^T\end{aligned}$$

C. Residual Evaluation

To assess the performance of the methods a *reference set* was built using both data from normal condition and mechanical asymmetry. This can correspond, for example, to a motor meant to work with different asymmetric loads. A *test set* was built using normal condition, mechanical asymmetry, and several unseen states such as a combination of electrical and mechanical asymmetry, and different degrees of power supply unbalance —see Fig. 4. Whereas the reference set was used to build a process model, the test set was used to calculate residuals in order to characterize the novel process states.

Feature vectors $\mathbf{x}(k)$ at each data window k, were normalized to zero mean and unit standard deviation. Autoassociative mappings using the kernel based approach (6) and the least squares approach (7) were used. For both methods, a value of $\sigma = 1$ was chosen to cover all the input pdf and a regularizing value of $\lambda = 10^{-9}$ was used for the LS method. Residuals were computed according to (9) as the difference between the actual feature vector and its approximation.

Known facts. On the presence of an electrical asymmetry, a general increase in the 100 Hz vibration energy appears as a result of unbalanced magnetic forces. Also, under ideal balanced conditions, the complex Park vector defined in (22) should have a single $+50$ Hz harmonic. On the onset of an unbalance, a -50 Hz component appears, while the $+50$ Hz component decreases. In turn, mechanical asymmetry mainly increases the vibration level at the rotating speed, 25 Hz. However, as we have considered mechanical asymmetries within the *reference condition* this increase should not be shown up as a fault.

Results[2] are shown in Fig. 4. As seen, KR residuals yield high (red tones) values for the three 100 Hz vibration features when an electrical unbalance appears. This is not observed in the LS residuals which yield near zero values for the same conditions and a strong negative (dark blue) residual in the $a_x^{(100\text{Hz})}$ for simultaneous mechanical and electrical asymmetries. Similarly, a decrement in $i^{(+50\text{Hz})}$ and an increment $i^{(-50\text{Hz})}$ should yield positive (red tones) and negative (blue tones) residuals respectively. This is clearly observed in the

[2] *Note:* A color map was used in the original electronic hardcopy. Black and white printed version may hide information on the sign of deviations. However we have tried to remove ambiguities through the text of the paper.

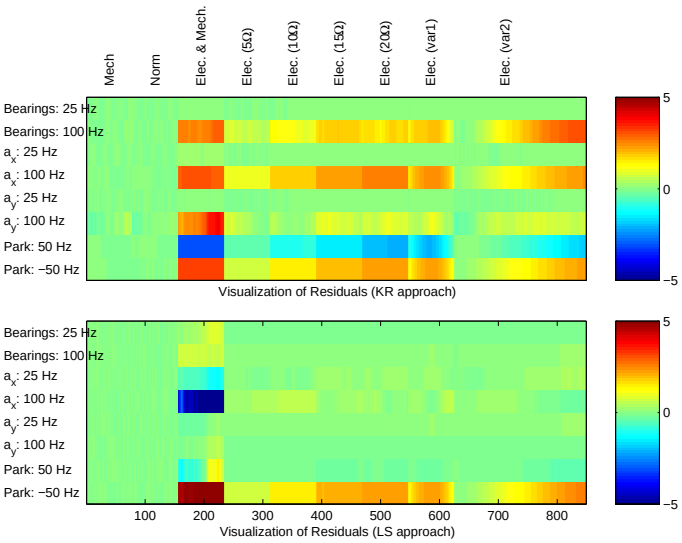


Fig. 4. Visualization of residuals for both methods. The 8 features are represented in the vertical axis. Horizontal axis represents time and color scale represents the magnitude of each residual.

KR residuals but not in the LS. Moreover, for gradual transitions (5Ω to 20Ω), the KR method shows gradual increments in the inverse sequence harmonic and in the 100 Hz components, while the LS method gives near zero residuals values. This can also be noticed in the last two records, where continuous modifications of R (*down-up-down* and *down-up*) were done.

VII. Conclusions

In this paper it transpires that a suitable residual generation can yield meaningful residuals which can be correlated with prior knowledge in a natural way through efficient visualization methods, also allowing for fault identification. We show that the kernel based approaches are more justifiable than least squares approaches for residual computation. While the former ones take into account the support of the data, the latter, in looking for a global cost minimization, develop an "ad-hoc" data joint distribution which allows for optimum least squares fitting but often lead to meaningless residuals.

Our analysis is also qualitatively plausible, and similar differences could be established in a more general context between methods with *support nature*, such as k-means, SOM or Gaussian mixture models (GMM), and others with plain least squares cost functions such as multilayer perceptrons, widely used for autoassociative mappings in industrial applications.

Questions such as the computational effort for training sets of considerable size, or the effect of outliers in the training set can arise using nonparametric methods, which have been used here only for comparison purposes. Using alternative density modeling methods, such as GMM, one can sacrifice accuracy in pdf estimation to reduce significantly the computational burden while being less sensitive to outliers. We are currently investigating these issues.

Acknowledgments

Authors are specially thankful to professors Olli Simula and Alberto Diez for making this work possible, as a result of the collaboration between the Laboratory of Computer and Information Science, at the Helsinki University of Technology and the Department of Electrical Engineering of the University of Oviedo.

References

[1] Esa Alhoniemi, Jaakko Hollmén, Olli Simula, and Juha Vesanto. Process monitoring and modeling using the self-organizing map. *Integrated Computer Aided Engineering*, 6(1):3–14, 1999.

[2] T. J. Böhme, N. Valentin, C. S. Cox, and T. Denoeux. Comparison of autoassociative neural networks and kohonen maps for signal failure detection and reconstruction. In C. H. Dagli et al., editor, *Intelligent Engineering Systems through Artificial Neural Networks 9 (Proc. of ANNIE'99)*, pages 637–644, Saint-Louis, 1999. ASME Press.

[3] Christopher M. Bishop, Markus Svensén, and Christopher K. I. Williams. *Neural Computation*, 10(1):215–235, 1996.

[4] A.J.Marques Cardoso and E.S. Saraiva. Computer-aided detection of airgap eccentricity in operating three-phase induction motors by park's vector approach. *IEEE Transactions on Industry Applications*, 29(5):897–901, Sep-Oct 1993.

[5] Ignacio Díaz, Alberto B. Diez, and Abel A. Cuadrado. Complex process visualization through continuous self organizing maps using radial basis functions. In *International Conference on Artificial Neural Networks (ICANN'01)*, pages 443–450, Viena, Austria, 2001.

[6] J.J. Gertler. Survey of model-based failure detection and isolation in complex plants. *IEEE Control Systems Magazine*, 8(6):3–11, December 1988.

[7] Christopher G. Healey, Kellogg S. Booth, and James T. Enns. Visualizing real-time multivariate data using preattentive processing. *ACM Transactions on Modeling and Computer Simulation*, 5(3), 1995.

[8] Christopher G. Healey, Kellogg S. Booth, and James T. Enns. High-speed visual estimation using preattentive processing. *ACM Transactions on Computer-Human Interaction*, 3(2), 1996.

[9] Teuvo Kohonen, Erkki Oja, Olli Simula, Ari Visa, and Jari Kangas. Engineering applications of the self-organizing map. *Proceedings of the IEEE*, 84(10):1358–1384, Oct 1996.

[10] E. A. Nadaraya. On estimating regression. *Theory of Probability and its Applications*, 9:141–142, 1964.

[11] T. Petsche, A. Marcantonio, C. Darken, S.J. Hanson, G.M. Kuhn, and I.Santoso. A neural network autoassociator for induction motor failure prediction. In D.S. Touretzky, M.C. Mozer, and M.E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, pages 924–930. Cambridge: MIT Press., 1996.

[12] Tomaso Poggio and Federico Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, September 1990.

[13] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, Dec., 22 2000.

[14] Donald F. Specht. A general regression neural network. *IEEE Transactions on Neural Networks*, 2(6):568–576, November 1991.

[15] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, Dec, 22 2000.

[16] Juha Vesanto. Som-based data visualization methods. *Intelligent Data Analysis*, 3(2):111–126, 1999.

[17] G.S. Watson. Smooth regression analysis. *Sankhya Series A*, 26:359–372, 1964.

[18] Andrew R. Webb. Functional approximation by feed-forward networks: A least-squares approach to generalization. *IEEE Transactions on Neural Networks*, 5(3):363–371, 1994.

[19] David J. H. Wilson and George W. Irwin. RBF principal manifolds for process monitoring. *IEEE Transactions on Neural Networks*, 10(6):1424–1434, November 1999.

[20] Paul Yee and Simon Haykin. A dynamic regularized radial basis function network for nonlinear nonstationary time series prediction. *IEEE Transactions on Signal Processing*, 47(9):2503–2521, September 1999.