J Electron Test
DOI 10.1007/s10836-011-5220-0

Fault Detection, Diagnosis and Prediction in Electrical Valves Using Self-Organizing Maps

Luiz Fernando Gonçalves · Jefferson Luiz Bosa · Tiago Roberto Balen · Marcelo Soares Lubaszewski · Eduardo Luis Schneider · Renato Ventura Henriques

Received: 6 September 2010 / Accepted: 16 March 2011

Abstract This paper presents a proactive maintenance scheme for fault detection, diagnosis and prediction in electrical valves. The proposed scheme is validated with a case study, considering a specific valve used for controlling the oil flow in a distribution network. The scheme is based in self-organizing maps, which perform fault detection and diagnosis, and temporal self-organizing maps for fault prediction. The adopted fault model considers deviations either in torque, in the valve's gate position or in the opening or closing time. The map which performs the fault detection, diagnosis and prediction, is trained with the energy spectral density information, obtained from the torque and position signals by applying the wavelet packet transform. These signals are provided by a mathematical model devised for the electrical valve. The training is performed by fault injection based on parameter deviations over this same mathematical model. The proposed system is embedded into an FPGA-based platform. Experimental results demonstrate the effectiveness of the proposed approaches.

Keywords Proactive maintenance · Fault prediction · Test of electromechanical systems · Self-organizing maps

A presentation based on this article was made at the Eleventh IEEE Latin-American Test Workshop, Punta del Este, Uruguay, March 28–31, 2010.

Responsible Editor: F. Vargas

L. F. Gonçalves (✉) · T. R. Balen · M. S. Lubaszewski · E. L. Schneider · R. V. Henriques
Departamento de Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Av. Osvaldo Aranha, 103, Bom Fim, CEP: 90.035-190, Porto Alegre, RS, Brasil
e-mail: luizfg@ece.ufrgs.br

T. R. Balen
e-mail: tiago.balen@ufrgs.br

M. S. Lubaszewski
e-mail: luba@ece.ufrgs.br

E. L. Schneider
e-mail: edu@ece.ufrgs.br

R. V. Henriques
e-mail: rventura@ece.ufrgs.br

J. L. Bosa
Instituto de Informática, Universidade Federal do Rio Grande do Sul, Av. Bento Gonçalves, 9500, Agronomia, CEP: 91.509-900, Porto Alegre, RS, Brasil
e-mail: jlbosa@inf.ufrgs.br

1 Introduction

In industrial plants, aircrafts, hospitals or in other environments, different types of faults may occur in the equipments in use. Bearings may jam, valves may leak or sensors may provide wrong readings, for example. A hostile environment, poor operation conditions, lack or insufficient maintenance, allied to the natural process of aging and degradation, often lead to an increasing occurrence of failures [25]. If these failures are not avoided, they may cause the growth of operational costs, they may halt the production process or they may, eventually, provoke serious accidents. For these reasons, system maintenance methods have been an important research topic for several decades.

The latest advances in computing and electronics have made it interesting to automate and integrate proactive maintenance (also known as intelligent maintenance) tasks into systems. Traditional maintenance

Published online: 12 April 2011

strategies (corrective, preventive or predictive) are based either on post-failure correction or on off-line periodic system checking. The proactive maintenance, differently from the traditional strategies, is built up from techniques for fault prediction, fault detection and fault diagnosis [2, 8, 10]. By using these three elements, it becomes possible to keep track of the equipment or the whole industrial plant, to quantify the performance degradation of the parts and thus to determine the remaining system lifetime. As a consequence, it turns feasible to schedule the replacement of degraded parts for idle or lower production periods of time, or automatically reconfigure the system for continuous (although degraded) operation until the faulty parts can be repaired.

In a proactive maintenance context, fault detection is usually performed by on-line monitoring schemes that check the validity of the operations undertaken by the system. The behavior of the monitored system must be compared to a reference to check whether or not it is working as expected [32]. This monitoring is often achieved by analyzing how a limited number of operation parameters degrade over time. In these systems, performance degradation is considered as a warning message, a sort of flag indicating an upcoming fault [2]. Therefore, it is crucial not only to determine appropriate parameters to indicate the degradation of system signals, but also to define the most suitable mathematical tools to analyze the way these signals degrade. In summary, these systems require wide instrumentation for on-line monitoring, and distributed intelligence to accurately predict the parts that start deteriorating.

A possible on-line fault detection technique consists in use a mathematical model to reproduce the dynamic behavior of the fault-free system. The mathematical model outputs are compared to the real measurements generating an error vector that gives the fault information [18, 26]. However, modeling errors may hide the real faults and may generate false alarms. Moreover, the error signals may be masked by other signals of the system, generating noise, for example [20]. For this reason, in many cases, it is necessary to employ more robust techniques, such as artificial neural networks. The major advantage of using neural networks is that they can simultaneously copy the fault-free system dynamic behavior and the error vector.

Neural networks may also be applied to perform fault diagnosis tasks. Fault diagnosis is accomplished by identifying the faulty part of the system and, whenever possible, the type of fault affecting the part may also be identified. Several fault diagnosis techniques, many based on neural networks and genetic algorithms, have been proposed in the last years for different kind of applications [21, 31, 33]. Neural networks have become one of the preferred tools of maintenance systems probably because, by grouping together similar input data, they facilitate the fault diagnosis task [17].

Besides fault diagnosis, it is possible that an on-line test system be capable of predict the occurrence of a given set of faults. This task is related to the time series prediction. Prediction can be applied to several areas, as, for example, to predict the probability of an earthquake, of rain precipitation or changes in the stock market, for example, thus saving lives or even reducing failures and increasing equipment reliability, in the case of engineering applications [5, 8, 28]. Time series prediction focuses on building models of the system using the knowledge and information that is available. Hence, the constructed model can be used first, to emulate the system behavior, and second, to simulate the future events of the system.

Many methods for system behavior prediction have been developed based on very different approaches. In the past, many were based on statistics models, such as AR (Autoregressive) and ARMA (Autoregressive Moving Average) models. Usually, the statistics models (AR and ARMA) accomplish well the prediction on a rather short term, depending on the complexity level of the problem. However, their efficiency on a longer term is questionable [7, 28]. This fact is due to the learning strategy, used to fit the data into the model, whose goal is usually to optimize the performance at a given term, most often just the next time step.

More recently, neural networks based methods for fault prediction have been proposed, usually employing networks such as MLP (Multi-Layer Perceptrons), RBF (Radial Basis Networks) or SOM (Self-Organizing Maps) [1, 8, 21, 23, 28, 29]. The default neural network method accomplishes time series prediction through feed forward functions using a particular neural network architecture, such as a standard MLP or SOM architecture [7, 8, 11]. In this context, the self-organizing maps have been raising much interest recently, since, besides giving better results than the prediction approaches based on MLP or RBF, they also provide good solutions for fault detection and diagnosis.

In this work, a proactive maintenance scheme is proposed for fault detection, diagnosis and prediction in electromechanical actuators, taking as case study an electrical valve. A self-organizing map is used for fault detection and diagnosis, and a temporal Kohonen map (a special kind of self-organizing map) is used for fault prediction. In our case study, the valves are used for controlling the flow in an oil distribution network.

Some related works address the use of neural networks for fault detection and diagnosis in electromechanical systems [21, 24]. However, to the author's knowledge this is the first time that a proactive maintenance scheme, based on fault prediction, is proposed to be applied to the kind of actuator.

The paper is organized as follows: in Section 2, the pro-active maintenance system is proposed, for this purpose a mathematical model is devised for a particular valve and a self-organizing map and a temporal Kohonen map (TKM) are proposed for fault detection, diagnosis and prediction. Section 3 presents experimental results, including fault injection experiments and the design of an embedded system for the proposed scheme. Finally, concluding remarks are given in Section 4.

2 Proactive Maintenance Scheme

Although the concepts exploited in this paper are quite general, the proactive maintenance system we are proposing here was applied in this work to a particular class of electrical valves. These valves have embedded sensors for torque and opening position measurement.

The considered fault model for the valve includes parametric deviations in torque, in the valve's gate position or in the opening or closing time. We consider as detectable faults those that lead to measurable deviations on the energy spectral density of the torque and opening position signals. The energy spectral density of this signal is computed using the wavelet packet transform (WPT) [15, 19]. It is important to point out that the torque and position signals where considered since this signals are already available from the valve, by means of an embedded instrumentation circuitry.

As a reference, the normal, degraded and faulty behaviors of the valve are learned by self-organizing maps and are on-line checked for fault detection, diagnosis and prediction (by a temporal Kohonen map) during the valve operation.

These maps are built during a training phase using a mathematical model for the valve and are consulted on-line by computing the best matching between an acquired measure and the neurons of the trained reference map.

Figure 1 shows the proposed system, which is composed of three main blocks that are further discussed in next sections: a mathematical model that represents the actuator, the valve and the pipe behavior; the signal processing and feature extraction tool (based on the wavelet packet transform); and the artificial intelligence (AI) tool based on self-organizing maps, that

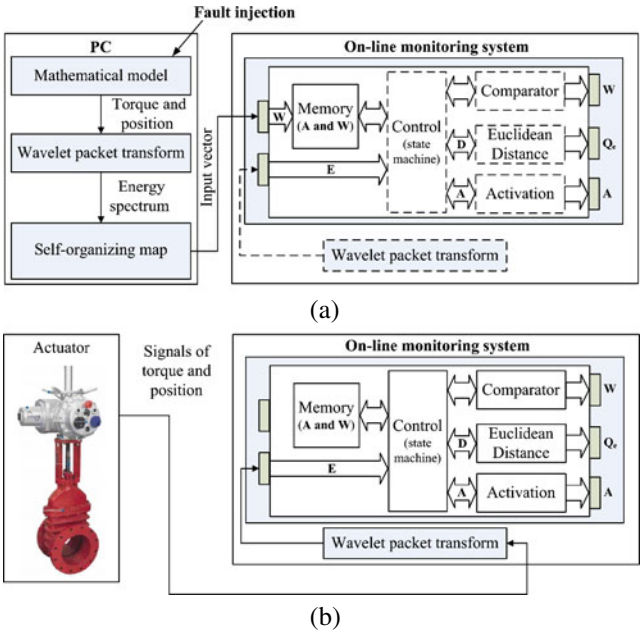


Fig. 1 The proactive maintenance scheme: **a** training phase; **b** on-line testing phase

are computed in a PC station and stored in the on-line monitoring system, afterwards. The training and testing phase are evidenced in Fig. 1a and b.

The comparator block shown in Fig. 1 is responsible for determining the winner neuron (**W**). Other two specific blocks (controlled by means of a finite state machine) are used to calculate the Euclidean distance, in order to obtain the quantization error (Q_e) and the temporal activation of the neurons (A). According to [8], in the AI field, the quantization error is the average distance between each data vector and its BMU (Best Matching Unit) or, the winner neuron.

The trained values of A and **W** (obtained by training the SOM in a PC station, with the valve model) are stored in a memory block, which is part of the on-line monitoring system.

2.1 Mathematical Model

The valves used in the industry may be of different types (e.g., drawer, sphere or globe valves) and each type suits better a different application. In this case study we adopted an electrically actuated drawer valve that is used to control the oil flow in a distribution network. In Fig. 2 the electrical actuator, the valve itself and the fluid pipe are shown.

The valve is modeled in order to evaluate its behavior in different operation conditions, including normal, degraded and faulty operation. This model takes into account a set of forces involved in the gate valve open-

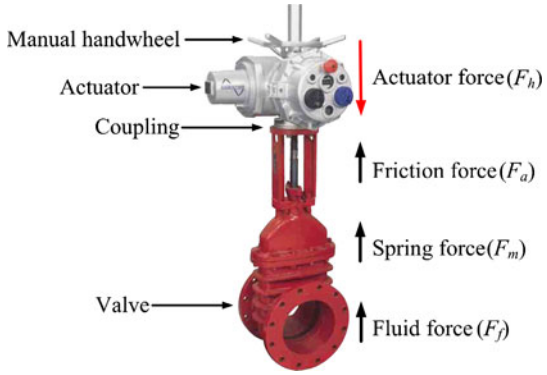


Fig. 2 Electrical actuator valve and pipe

ing and closing movements, when the actuating force is transmitted from the electrical engine (asynchronous machine) through the valve gears.

Differential and algebraic equations are devised and included in a system of non-linear equations to model the actuator behavior and the mechanical forces involved. Several physical constraints are considered in the modeling process in order to simplify the representation and, consequently, the computational effort required [4, 14].

A third-order model is chosen for the induction engine, because it can represent the permanent and transient conditions. This mathematical model is given by the differential equations (Eqs. 1a–1c), where $\dot{V}'_d$ and $\dot{V}'_q$ are internal voltages (direct and quadrature, respectively), and s the slip of the asynchronous engine. The differential equations (Eqs. 1d and 1e) complete the overall fifth-order mathematical model and relate the gear system and the pipe through the gate position (a), the speed (v_a) and the valve gate acceleration (a_a). Further information regarding part of the model described below can be found in [14].

$$\dot{s} = \frac{1}{2H_g}(T_e - T_m) \tag{1a}$$

$$\dot{V}'_d = \frac{-1}{T_0}\left[V'_d - I_{qs}(X_s - X'_s)\right] + s\,\omega_s\,V'_q \tag{1b}$$

$$\dot{V}'_q = \frac{-1}{T_0}\left[V'_q + I_{ds}(X_s - X'_s)\right] - s\,\omega_s\,V'_d \tag{1c}$$

$$\dot{a} = v_a \tag{1d}$$

$$\ddot{a} = a_a = \frac{1}{M_h}\left(F_h - F_f - F_a - F_m\right) \tag{1e}$$

where H_g is the inertia constant, T_e is the electromechanical torque, T_m is the rotor torque, T_0 is the time constant associated to the rotor inductance of the asynchronous machine, I_{qs} and I_{ds} are the machine stator currents, X_s is the stator reactance, X'_s is the transient stator reactance, ω_s is the system frequency, M_h is the weight of the gate, F_h is the force sent to gate, F_f is the force exerted by the fluid, F_a is the friction force and F_m is the spring force.

Finally, the algebraic equations that describe the electrical torque of the asynchronous engine (T_e), the voltages of the asynchronous machine (V_{ds} and V_{qs}), the flow (F_f), friction (F_a) and spring force (F_m), as well as the gate shutter force (F_h) and gate shutter torque (T_h), are given by Eqs. 2a–2h below, for $\theta = 0^o$:

$$V_{ds} = V'_d - R_s\,I_{ds} + X'_s\,I_{qs} \tag{2a}$$

$$V_{qs} = V'_q - R_s\,I_{qs} - X'_s\,I_{ds} \tag{2b}$$

$$T_e = V'_d\,I_{ds} + V'_q\,I_{qs} \tag{2c}$$

$$T_h = -T_m\,K_h\,T_{mb} \tag{2d}$$

$$F_h = \frac{T_h}{R_h\,\cos(\theta)} \tag{2e}$$

$$F_m = K_m\,a \tag{2f}$$

$$F_a = C_a\,v_a \tag{2g}$$

$$F_f = \frac{V_f^2\,A_v}{\rho\,N_R^2\,(100 - a)^2\,C_v^2} \tag{2h}$$

where R_s is the stator resistance, K_h is the inherent reduction constant, T_{mb} is the base mechanical torque, K_m is the hook coeficient of spring, C_a is the friction constant, V_f is the flow of the fluid, A_v is the area of the valve, ρ is the density of the fluid, N_R is the Reynolds number and C_v is the flow coefficient.

The mathematical model, as presented above, can be used to evaluate the normal behavior of the valve. By deviating the nominal values of the parameters in the equations, fault injection can be performed in a simple manner and misbehaviors (degraded or faulty valve) may be observed in the torque and opening position curves. The fault injection procedure will be discussed in the next section.

At the time of defining the method to obtain the system dynamic solution, we realized that it was necessary to have an open tool. The routines of this tool should provide some flexibility in the structure and modeling of the simulated system, thus allowing the simulation of any system configuration and enabling the analysis of various models of actuators and valves, as well as different situations and faults that might occur. Then, a simulator for dynamic nonlinear systems and the mathematical model previously described were developed using the Matlab software. These tools run in a PC station.

After defining the model and the simulation tool, Eqs. 1 and 2 were included into a differential and

algebraic nonlinear equation system, also known as singular or descriptor, as follows:

$$\dot{x} = f\left(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{p}\right) \tag{3a}$$

$$0 = g\left(\mathbf{x}, \mathbf{z}, \mathbf{u}, \mathbf{p}\right) \tag{3b}$$

where $f(\,)$ and $g(\,)$ are nonlinear vector functions formed by Eqs. 1a–1e and 2a–2h, respectively, $\mathbf{x}$ is a state variable vector, $\mathbf{z}$ is an algebraic variable vector, $\mathbf{u}$ is an input vector and $\mathbf{p}$ is a constant vector defined as:

$$\mathbf{u} = \left[\omega_s \;\; \dot{V}_{ds} \;\; T_m\right] \tag{4a}$$

$$\mathbf{x} = \left[\dot{s} \;\; \dot{V}'_d \;\; \dot{V}'_q \;\; \dot{a} \;\; \ddot{a}\right] \tag{4b}$$

$$\mathbf{z} = \left[I_{ds} \;\; I_{qs} \;\; T_e \;\; T_h \;\; F_h \;\; F_m \;\; F_a \;\; F_f \;\; a_a\right] \tag{4c}$$

$$\mathbf{p} = \left[H_g \;\; R_s \;\; X_s \;\; X'_s \;\; \ldots \;\; K_h \;\; K_m \;\; C_a \;\; A_v \;\; M_h\right] \tag{4d}$$

The algebraic-differential representation of the system and the structural description of functions $f(\,)$ and $g(\,)$, Eqs. 3a and 3b, respectively, are the basic input information to provide for the simulator.

The procedure implemented in the simulator calculates both the values of $\mathbf{x}$ and $\mathbf{z}$ vectors for each time step. This method is also called instant solution. This procedure includes the use of two numerical methods: one for the solution of differential equations (and initial sample) and another to solve the system of equations as a whole. The integration method used to solve the equations is the trapezoidal rule or modified Euler rule. The method used to obtain the nonlinear equation system solution at each instant of time is the method of Newton–Raphson [14].

2.2 Signal Processing

As mentioned before, the gate shutter torque (T_h) and the gate position (a), given by Eqs. 2d and 1d of the mathematical model described in Section 2.1, are the signals chosen to be monitored on-line in our case study. These signals are therefore generated by the mathematical model for normal, degraded and faulty operations. The wavelet packet transform is the signal processing tool used in this work for processing these two signals.

The wavelet packet transform (WPT) provides an alternative to the short-time Fourier transform (STFT) and to the wavelet transform (WT) in non stationary signal processing. In contrast to the fixed size of the analysis window of the STFT and to the poor frequency resolution of the WT, the WPT uses longer windows for lower frequencies and shorter windows for higher frequencies to obtain finer frequency resolution. Additionally, the capability of decomposing the signal in frequency bands makes the WPT more attractive than other signal processing tools.

Since the wavelet packet transform preserves timing and spectral information, it is a suitable tool for the analysis of non-stationary signals such as gears, bearings and actuators impulsive signals [34]. Moreover, the WPT allows tailoring the frequency bands to cover the range of injected faults [3, 15, 22]. Indeed, the choice of the appropriate signal processing tool depends on several characteristics of the signal, such as the number of samples and timing characteristics. Before deciding on the use of WPT we also tested the fourier transform (with and without windowing). However, better results were achieved by using WPT.

The numerical solution of wavelet packet coefficients ($\mathscr{W}$) is the sequence of inner products of the signal $y(t)$ with the wavelet packet functions $\psi_{\kappa\beta\eta}(t)$:

$$\mathscr{W}\left\{y, \psi\right\} = \nu_{\kappa\eta}(\beta) = \frac{1}{\sqrt{2^{\kappa}}} \int_{-\infty}^{\infty} y(t) \,.\, \psi_{\kappa\beta\eta}\, dt \tag{5}$$

where $\psi_{\kappa\beta\eta}$ are the wavelet packet functions parameterized through the integers κ, β and η.

Wavelet packets consist of a set of linearly combined usual wavelet functions. A wavelet packet function, $\psi_{\kappa\beta\eta}$, is defined by:

$$\psi_{\kappa\beta\eta} = \frac{1}{\sqrt{2^{\kappa}}} \psi_{\eta}\left(\frac{t - \beta\, 2^{\kappa}}{2^{\kappa}}\right) \tag{6}$$

The wavelets ψ_{η} are obtained from the following recursive relationships:

$$\psi_{2\eta}(t) = \sqrt{2} \sum_{\beta=-\infty}^{\infty} h(\beta)\, \psi_{\eta}(2t - \beta) \tag{7a}$$

$$\psi_{2\eta+1}(t) = \sqrt{2} \sum_{\beta=-\infty}^{\infty} g(\beta)\, \psi_{\eta}(2t - \beta) \tag{7b}$$

where $h(\,)$ and $g(\,)$ are discrete filters.

The discrete filters $h(\beta)$ and $g(\beta)$ (high-pass and low-pass filters, respectively) are quadrature mirror filters associated with the scaling function $\phi(t)$ and the mother wavelet function $\psi(t)$.

The WPT is a generalization of the wavelet analysis to enhance the decomposition procedures. A typical orthogonal wavelet packet decomposition procedure is shown in Fig. 3. In this approach, the signal is decomposed into two parts: a vector of approximation (vector A) and a vector of detail coefficients (vector D). Both approximation and detail coefficient vectors are successively decomposed into two parts.

The signal analyzed by the WPT and obtained from the equations of the system model is composed by the

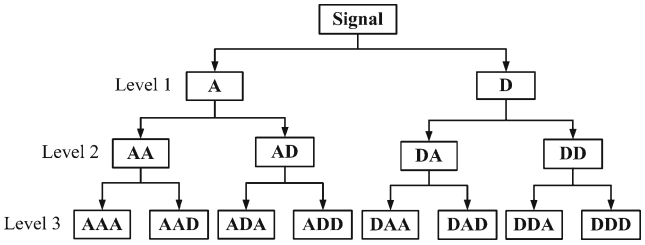


Fig. 3 Signal decomposition procedure using WPT

gate shutter torque (T_h) and the position signal (a), such as:

$$y(t) = [\ T_h(t)\ \ a(t)\ ] \tag{8}$$

The signal $y(t)$ can be decomposed into a number of sub-bands. The number of frequency sub-bands for the WPT has to be i powers of two, where i corresponds to the decomposition level of the WPT. Accordingly, the signal can be decomposed into 2^i frequency sub-bands, with the bandwidth in Hertz for each sub-band i defined by:

$$f_i = \frac{f_h}{2^i} \tag{9}$$

where f_h is the highest frequency component of the signal to be analyzed.

Then, the features can be extracted from the harmonic wavelet packet coefficients in each sub-band to provide information on the condition of the actuator being monitored. The deviation of the energy content of each sub-band of the signal $y(t)$, after decomposition, is directly related to the degree of severity of the fault. Then, it can be used as an effective indicator and a key feature of the actuator condition. More specifically, the coefficients A and D quantify the energy associated with each specific sub-band. Given the wavelet packet coefficients A and D of the actuator signal $y(t)$, the spectral energy feature vector $\mathbf{E} = [\ E_1,\ E_2,\ \ldots,\ E_N\ ]$ in each sub-band i is defined as:

$$E_i = \sum_{n=1}^{N} |y[n]|^2 \cong \sum_{n=1}^{N} |A_{k\beta}[n]|^2 + \sum_{n=1}^{N} |D_{k\beta}[n]|^2 \tag{10}$$

where $A_{k\beta}$ and $D_{k\beta}$ are the vectors of the coefficients of approximation and detail, respectively.

In summary, the spectral energy information ($\mathbf{E}$) is extracted from the gate shutter torque (T_h) and opening position (a) by the WPT. The spectral density is divided into N frequency bands, and the resulting information is used by the artificial intelligence tool to construct the self-organizing and the temporal Kohonen maps described in the following paragraphs.

Similarly to the mathematical model, the WPT tool runs in a PC station during the system training phase. During on-line testing, the WPT shall be part of the internal resources of the on-line monitoring system.

2.3 Self-Organizing Maps

The self-organizing maps, or Kohonen maps, belong to a class of neural networks that apply the unsupervised learning paradigm based on competition, cooperation and adaptation techniques [1, 12]. Figure 4 shows the basic self-organizing map structure.

Formally speaking, the ultimate goal of a self-organizing map is, after being trained, to map any input data from an $\mathbb{R}^n$ space representation into a two-dimensional, lattice-like matrix. This lattice builds a competition layer, with J neurons on the net, where $J = C_m\ L_m$, as seen in Fig. 4.

The internal processing of self-organizing map algori-thms can be simplified and divided in three different steps or stages: start up (when the synaptic weights are initialized), training (when knowledge is acquired by the map) and testing (when data inputs are classified on the map).

For the SOM start up step, the synaptic weights, $\mathbf{W}$, of all network neurons, defined as $\mathbf{W} = [W_1,\ W_2,\ \ldots,\ W_N]^T$, are initially assigned to random values. Next, in the training step the known input vectors representing the system behavior are mapped into the matrix. The input vectors, in our case, are energy spectral density values, present in the energy feature vector, $\mathbf{E}$, obtained from the WPT.

Three steps build the training phase of self-organizing maps: the competition, the cooperation and the adaptation.

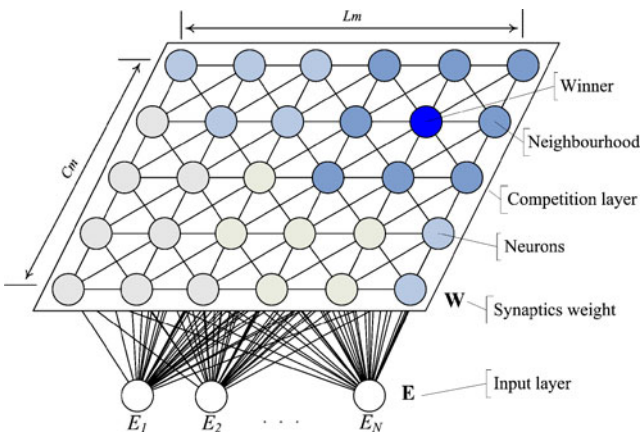


Fig. 4 Basic SOM structure

The competition step searches in the current map the neuron with the synaptic weights vector, **W**, that best matches the input vector, **E**, so minimizing the Euclidean distance:

$$D^{kj} = \|\mathbf{E}^k - \mathbf{W}^j\| = \sqrt{\sum_{i=1}^{N} \left(E_i^k - W_i^j\right)^2} \tag{11}$$

where D^{kj} is the Euclidean distance, K is the number of input vectors with $k = 1, 2, \ldots, K$; J is the number of neurons on the net for $j = 1, 2, \ldots, J$; and N is the number of **W** and **E** vector elements.

The neuron which presents the smallest value for D^{kj} is defined as the winner neuron, $\mathbf{W}_{BMU}$ [8, 27]. This process is expressed as:

$$W_{BMU} = \arg \ \min_{\vee k} \ \|\mathbf{E}^k - \mathbf{W}^j\| \tag{12}$$

where W_{BMU} is the index of the self-organizing map winner and the "arg" denotes "index".

The cooperation step is in charge of identifying the direct neighbors of the winner neuron in the map. In the adaptation step the synaptic weights **W** of the direct neighbors are updated as a function of the input vector **E**.

During the training process, the weight vector of the winner is moved towards the presented input data by a certain fraction of the Euclidean distance, as indicated by a time-decreasing learning rate α. Also, the weight vectors of the neighboring units are modified according to a spatial-temporal neighborhood function $h()$. The neighborhood function used in this work is the "Gaussian" function [16]. The learning rule may be written as:

$$\mathbf{W}^j(t+1) = \mathbf{W}^j(t) + \alpha(t)\, h(t) \left[\mathbf{E}^k(t) - \mathbf{W}^j(t)\right] \tag{13}$$

where t denotes the current learning iteration and $\mathbf{E}^k$ represents the current input energy vector. Further details on these two steps can be found in [12, 13, 28]. This iterative learning procedure leads to a topologically ordered mapping of the input data. Similar patterns are mapped onto neighboring units, whereas dissimilar patterns are mapped onto units farther apart.

For SOM in-use mapping (on-line testing), only the competition step is performed considering the measured input vector, **E**, and the synaptic weights, **W**, of the neurons in the trained map.

2.4 Temporal Self-Organizing Maps

The temporal Kohonen map is an unsupervised approach for time series prediction derived from the self-organizing map algorithm [12]. The temporal Kohonen map model uses leaky integrators to maintain the activation history of each neuron. These neurons gradually lose their activity and are added to the outputs of the other normal competitive units. These integrators, and consequently the activation decay, are modeled through the following difference equation:

$$A^k = \lambda\, A^{k-1} - \frac{1}{2}\left(D^{kj}\right) \tag{14}$$

where λ, $0 < \lambda < 1$, is a time constant, A^k is the temporal activation of the unit j at step k.

The neuron which presents the greatest value for A^k (maximum activity) is defined as the winner activation, A_{BMU}, analogously to the traditional SOM. Equation 14 preserves the trace of the past activations as a weighted sum. In fact it integrates a linear low pass filter to the outputs of the normal competitive units [27].

Except for the determination of the winner neurons, all other steps of the TKM are the same as in the SOM algorithm. On one hand, in the SOM algorithm, the winner neurons are determined by computing the Euclidean distance: the neuron with the shortest distance is the winner. On the other hand, in the TKM, the winner neurons are obtained by computing the activation: the neuron with the highest activation is the winner. The activation values are computed through the recursive summing (also based on the Euclidean distance) of the current input vector and those previously stored in the map. The winner neurons will determine the system behavior trajectory in the TKM.

3 Experimental Results

As already stated in the previous section, two maps are needed to be trained to implement the proposed scheme—one for fault detection and another for fault diagnosis and prediction. For fault detection purposes, the map is trained considering only normal system situations (fault-free valve operation). In normal operation the maximum torque is 250 Nm and the valve is able to fully open and close, reaching both ends of the gate position. For fault diagnosis and prediction purposes, the map is trained considering typical normal, degraded and faulty situations, whose misbehaviors are simulated by injecting parameter deviations into the valve mathematical model.

For fault detection, in the testing phase, the quantization error, Q_e, is computed and compared to a detection threshold. If the error exceeds this threshold,

a fault is detected. We consider a threshold of 0.01 for the quantization error, which is defined as:

$$Q_e = \left\| \mathbf{E}^k - \mathbf{W}_{BMU} \right\| \tag{15}$$

For fault diagnosis and prediction, in the training phase, the SOM is colored such that the distance between neighboring neurons can be clearly visualized. The distance is given by the difference between the synaptic weights of neighboring neurons. Closer neurons will appear clustered in the self-organizing map and will be assigned to the same color. Different colors will denote neurons under different operation conditions: normal, degraded or faulty. In the testing phase, once the winner neuron is computed for a particular input vector **E**, the current system status can be identified in the colored self-organizing map. For the temporal Kohonen map, only the testing phase is performed considering the measured input vector, **E**, the synaptic weights, **W**, of the neurons in the trained SOM and the activations A. Then, for fault prediction, the Euclidean distance and the activation are computed and, in case of deviated behavior, the degradation trajectory can be visualized in the SOM/TKM.

3.1 Fault Simulation: Detection, Diagnosis and Prediction

To train the fault detection map, a set of simulations were performed to obtain typical values for torque and opening position under normal valve operation only. For the torque case, for example, the maximum fault-free value considered equals 250 Nm, as depicted in Fig. 5a.

To train the map used for fault diagnosis and prediction, besides the simulations to obtain typical values for normal valve operation, a set of additional simulations were performed considering degraded and faulty conditions. To exemplify the fault simulations performed, K_m, K_h, and C_a parameters (see Eqs. 2d–2g) are gradually incremented in the ranges shown in Table 1, such that degraded and faulty valve behaviors can be observed. These parameters and their variation range where considered since they represent the most common sources of failures in the studied valve, according to information provided by the manufacturer engineering department.

K_m deviations simulate the elasticity loss of the valve spring along time; K_h deviations simulate the degradation of the internal valve worm gear, till it partially breaks; and C_a deviations simulate an increase of friction between the valve stem and seal. As a consequence, for these faults, for example, the over torque may reach 275 Nm and the valve may be prevented

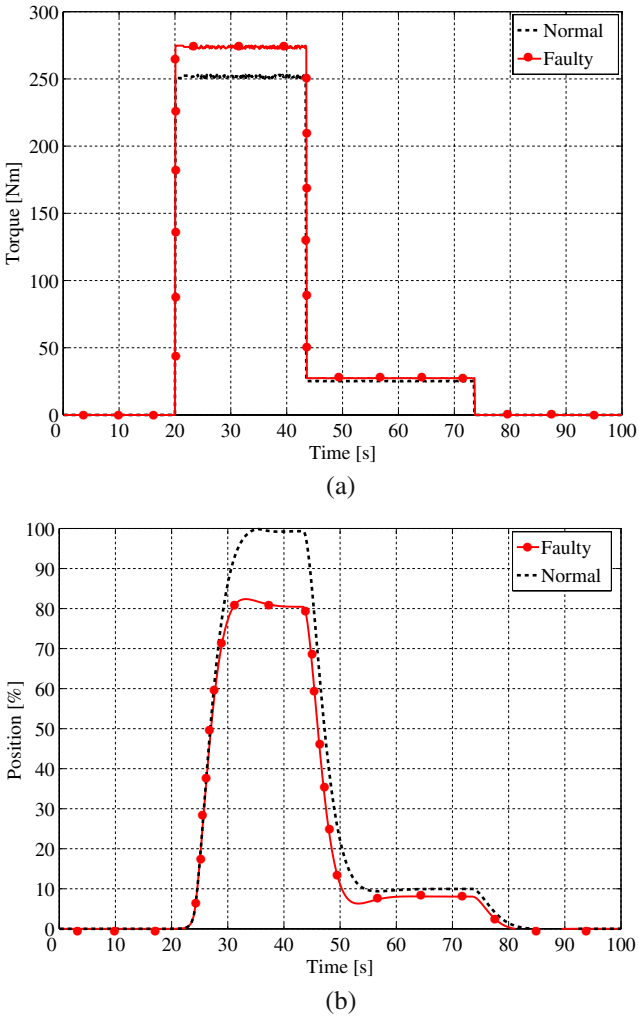


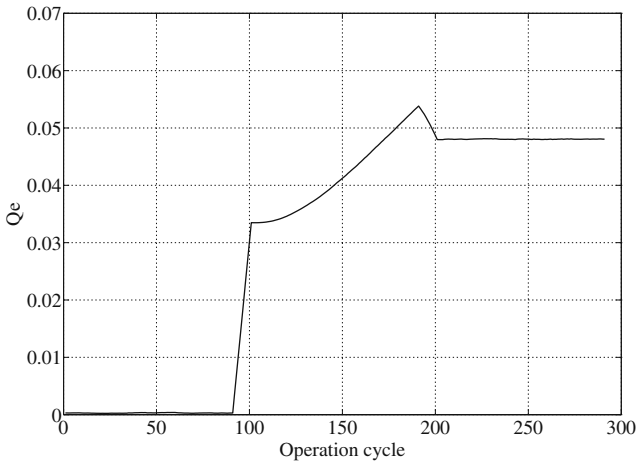
Fig. 5 Fault simulation: **a** torque and **b** position

to open the gate beyond 80% of the full range, as seen in Fig. 5. For the sake of simplicity we considered single parameter degradation and faults. This approach is suitable for the proof of concept, however, multiple faults are being considered for future works.

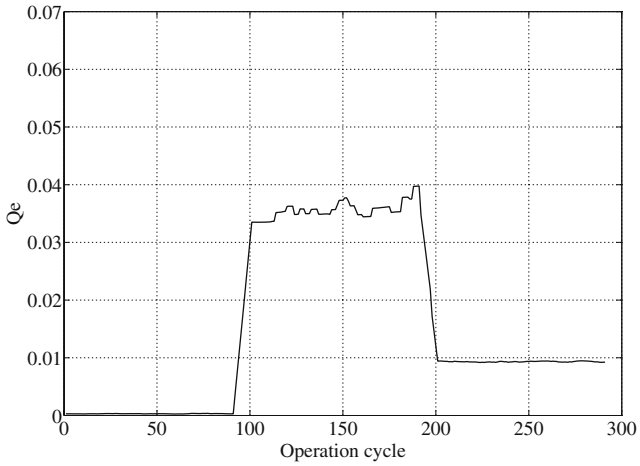
Figure 6 shows the quantization error for an increasing degradation of K_m, K_h and C_a simulated between the operation cycles 100–200 (until the cycle 100 no fault is injected). In cycle 200, the detection threshold is reached and a fault alarm can be triggered.

Table 1 Range and variation rate for K_m, K_h, and C_a

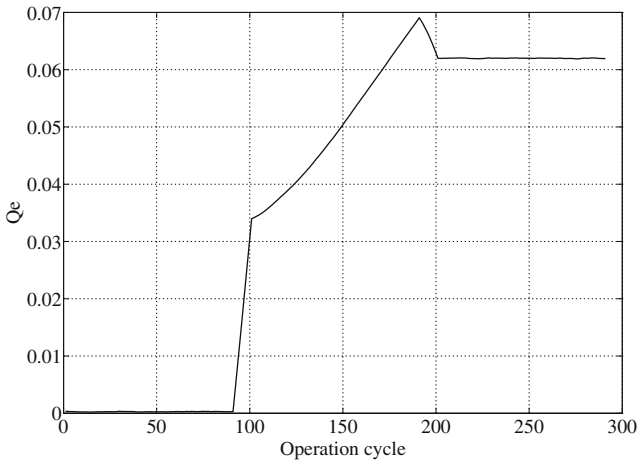
Parameter	Range	Step
K_m	4.215–5.215	0.01
K_h	11.00–12.00	0.01
C_a	16.00–21.00	0.05



(a)



(b)



(c)

Fig. 6 Quantization errors for: **a** K_m, **b** K_h and **c** C_a

In Fig. 6, from a to c, the detection thresholds obtained for K_m, K_h and C_a degradations, respectively, are shown. In this figure the degradation appears to start before the cycle 100. The reason for that is the filtering process used to obtain these data series, which reduces the number of samples.

All the experiments were performed by using the mathematical model. However, an experimental setup with a real valve is being constructed. Therefore real experimental data are expected to be included in future works.

Training for the fault diagnosis (to identify between K_m, K_h and C_a, faults) results in the map shown in Fig. 7. In this figure the neurons appear clustered around normal, degraded K_m, faulty K_m, degraded K_h, faulty K_h, degraded C_a and faulty C_a operation conditions. Each cluster in Fig. 7, corresponding to different operation conditions, is assigned to a different color.

As mentioned in Section 2, once the fault classification map is trained, D^{kj} and A^k are computed during the on-line testing phase. Then, the winner neuron computed for a given measured input vector can be easily located in this map and, consequently, the current status of the system is straightforwardly determined. Furthermore, by using the same map that was built during the training phase it is possible to carry out the failure prediction (Fig. 8).

In the temporal Kohonen map the system state can be visualized as a trajectory on the map and it is possible to follow the dynamics of the degradation process. This trajectory is described based on the winning neurons (A_{BMU}) for each temporal series. In the normal operation condition, the winners ought to follow a path inside the normal behavior region. When a fault occurs, the winner will deviate from the normal region. The deviation in the amplitude will depends on the type and severity of the fault. The fault trajectory gives important information about the failure mode. Thus, not only the degradation start can be detected, but also

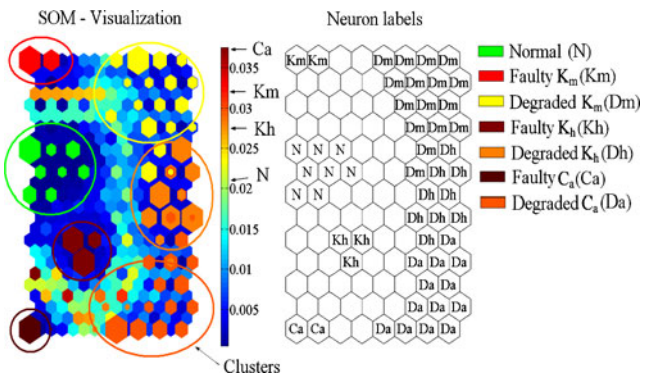


Fig. 7 Fault classification map considering faults in: K_m, K_h, and C_a

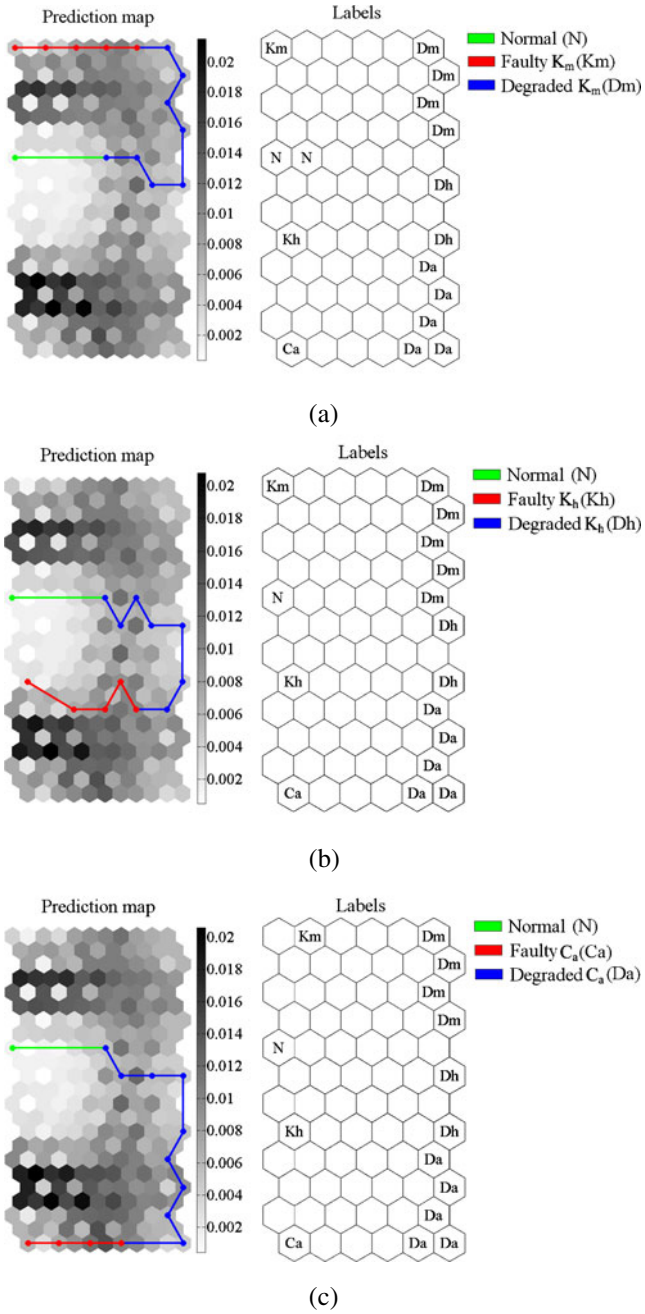


Fig. 8 Fault prediction map for faults in: **a** K_m, **b** K_h and **c** C_a

a prediction of possible failure modes can be achieved based on the trajectory trend.

Figure 8a–c show the trajectory prediction for K_m, K_h and C_a, considering the normal, faulty and degraded behavior of the valve. Three different paths, one for each simulated fault, can be seen in the figures. The trajectories start from neurons classified as normal, pass through neurons classified as degradation, and arrive to a neuron that represents the failure.

Although the results shown so far are good enough for the proof of concept, it is clear that to have a practical solution other type of faults must be considered and the impact of these faults in the SOM classification and TKM prediction capabilities must be evaluated. Additionally, for these faults, it must be checked to what extent the Euclidean distance will still be a good metric for fault detection purposes, or if other metrics such as the Hamming and the Manhattan distances should be investigated.

3.2 Design of an Embedded System for the Proactive Maintenance of Electrical Valves

According to the maintenance scheme in Fig. 1, the fault simulation and map training steps can run off-line in a PC station. However, for the on-line monitoring phase, in addition to the acquisition, conditioning and A-to-D conversion of torque and position signals, the computation of the input vector WPT, of the winner neuron W_{BMU}, and of the quantization error Q_e, may be embedded into the valve. Our proposal is to implement these functions using an FPGA platform. In this work we used the XUP Virtex2 PRO Xilinx FPGA Development Board [30] to prototype the system. This board embeds various hardware resources, including A-to-D and D-to-A converters, a Virtex2 device, a 512 MB SDRAM, an OPB bus, several I/O ports and user interfaces.

Based on this platform, we have chosen to reuse an available software implementation for the WPT computation [6], and decided to implement the $\mathbf{W}_{BMU}$ and Q_e computation blocks in hardware, as an IP-core.

Therefore, the WPT computation runs in a Microblaze processor [30] synthesized for the FPGA device. The processor runs at a 100 MHz clock. The Petalinux operational system and the WPT code are stored in the SDRAM and occupy around 2.2 MB of memory.

Figure 9 shows (a) the IP-core architecture proposed for the computation of W_{BMU} and Q_e, and (b) details of the hardware needed to obtain D^{kj}, (the Euclidean distance). Two important points are to highlight in this architecture. First is that, for the SOM implementation, the most important and most area consuming block in the FPGA is the memory (BRAM). Second is that, for the accurate computation, the operators of the Euclidean distance circuitry shall be based on a floating point representation and the Std. IEEE 754 [9] is used for this purpose.

The circuits in Fig. 9a and b were described in VHDL and have been synthesized and implemented in the Virtex2 PRO FPGA.

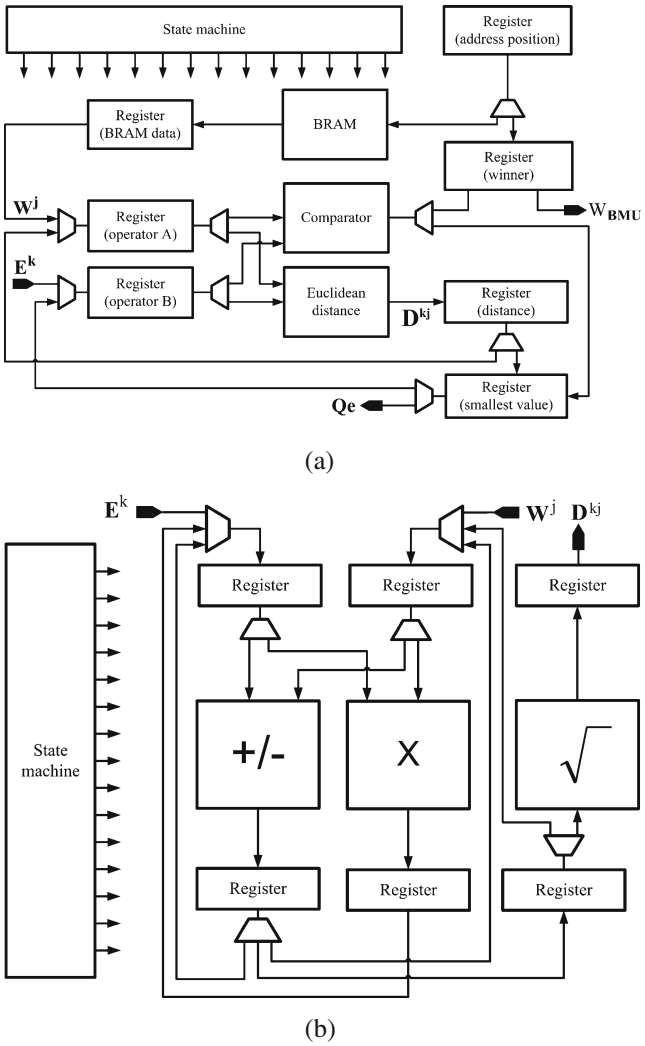


Fig. 9 Proposed IP-core: **a** SOM datapath and **b** the Euclidean distance computation

Experiments performed with this system have shown that map sizes of 90 neurons, with synaptic weight vectors of 20 elements of 32 bits each, are quite enough to reach the accuracy required for the valve application. Other applications may require greater maps to distinguish between faults with close behavior, for example. In this case, the memory requirements may become critical for the implementation of the embedded system. The FPGA device in-use can implement up to 512 neurons, with weight vectors of 32 elements of 32 bits each.

In this work, we have implemented the embedded maintenance system using a development board. However, it is likely that in real life the already existing board for the actuator control is redesigned to accommodate an FPGA device implementing the maintenance functionality. In this case, the question that arises is whether the area still available in the FPGA will be enough to implement those functions that are currently implemented by other board devices.

Table 2 FPGA area evaluation

Parameter	FPGA device	Embedded system		SOM-IP	
	Available	Used	Utilization (%)	Used	Utilization (%)
Slices	13,696	5,724	41.80	2,158	15.75
Flip-flops	27,392	5,762	21.04	2,597	9.48
LUTs	27,392	6,366	23.25	2,664	9.72
BRAMs	136	42	30.90	29	21.32
IOBs	556	134	24.00	–	–

Table 2 gives an answer to this question. As it can be noticed, keeping only the IP-core into the FPGA, there is definitely room enough to integrate the wavelet packet transform in hardware.

Another important requirement is the embedded system performance, measured here by the computation time. The gate valve takes 100 s to travel the full range, from the closed to the 100% open position and vice-versa. The whole travel is performed in 20 incremental steps, lasting 5 s each.

Therefore, in this application, $t = 5$ s is the time limit for all computations. Table 3 shows that the embedded system, as is, already meets the time requirements for the application. A much better performance is expected in the case the wavelet packet transform is implemented in hardware in the same FPGA device where the SOM-IP is programmed (this issue is under developing and are expected to be included in future works).

Finally, it should be mentioned that the power consumption of the embedded system, either implemented as a board or as a single FPGA device, is not a critical issue in the valve application. This is because the valve engine consumes much more power to move the valve gate than the embedded electronics altogether does.

Although the results obtained so far for the design of the embedded system point out to a very promising maintenance solution, it is clear that adding to the maps more valve faults, the system memory requirements may increase, if the same accuracy is required, and the maps implementation may require more hardware

Table 3 W_{BMU} and Q_e computation time for a single measurement

Algorithm	Time (s)
WPT in software	0.3812
SOM in hardware	0.0245
Total	0.4057

resources. A possible solution to reduce the overhead would be to investigate to what extent Adaptive or Kalman Filters, with less costly implementations in terms of silicon area, could be used for fault detection, diagnosis and prediction.

4 Conclusion

In this work, a proactive maintenance scheme is proposed for the detection, diagnosis and prediction of faults in electrical valves. These valves are used for flow control in an oil distribution network.

To the best of our knowledge, this is the first attempt to apply a proactive maintenance methodology to this sort of actuators that have only known corrective and preventive practices so far.

Another novelty brought in by this work is that an implementation of self-organizing map is proposed to solve the valve maintenance problem. An on-line monitoring system implements these maps for the detection, diagnosis and prediction of faults that lead to deviations either on torque, or on the valve opening position.

For fault detection and diagnosis, the self-organizing maps (and the TKM for fault prediction) are trained using a mathematical model and a fault injection procedure devised for the actuator, valve and pipe.

During the on-line monitoring phase, the system computes the best matching between an acquired measure and the neurons of the trained maps. This matching guides the fault detection and diagnosis steps that show up very effective. For fault prediction, the system computes the best matching between an acquired measure and the neurons of the trained map and respective temporal activations. This matching guides the fault prediction step.

An embedded system for the proactive maintenance of electrical valves based on self-organizing maps was prototyped using an XUP Virtex2 PRO Xilinx FPGA Development Board. The results obtained for the design of this system point out to a very promising maintenance solution. Current works are being developed addressing the investigation of the use of Adaptive or Kalman Filters for fault detection, diagnosis and prediction as an alternative to reduce the required hardware area.

Another point addressed by current works of our group is related to the fault prediction procedure. Although the scheme proposed in this work can indicate if a given component of the system is likely to fail (degradation trajectories), it is not capable of predict the remaining life-time of this component. In this sense, the computation of the first and second derivatives of the weights of the winner neurons, associated to extrapolation functions may predict the probable remaining time to failure. Although this work has considered a specific valve as case study to the proof of concept, this method can be extended to other types of actuators, if some minimal instrumentation resources are available.

Acknowledgments This research was supported by the CNPq Brazilian Research Agency under contract number 142027/2008-1, CAPES Brazilian Research Agency and Petrobras S.A.

References

1. Araújo AFR, Barreto GA (2002) Context in temporal sequence processing: a self-organizing approach and its application to robotics. IEEE Trans Neural Netw 13(1):45–57
2. Djurdjanovic D, Lee J, Ni J (2003) Watchdog agent—an infotronics based prognostic approach for product performance degradation assessment and prediction. Adv Eng Inf 17(5):109–125
3. Eren L, Devaney MJ (2004) Bearing damage detection via wavelet packet decomposition of the stator current. IEEE Trans Instrum Meas 53(2):431–436
4. Fitzgerald AE (1990) Electric machinery. McGraw-Hill, New York, 703 pp
5. Frank RJ, Davey N, Hunt S (2001) Time series prediction and neural networks. J Intell Robot Syst 31(1):91–103
6. Guleryuz OG (2009) Wavelet packets source codes. Polytechnic Institute of New York University. Available at: http://eeweb.poly.edu/onur/source.html
7. Hammer B, Micheli A, Neubauer N, Sperduti A, Strickert M. (2005) Self-organizing maps for time series. In: Proceedings of workshop on self-organizing maps, vol 1, pp 115–122
8. Huang R, Xi L, Li X, Liu CR, Qiu H, Lee J (2005) Residual life predictions for ball bearings based on self-organizing map and back propagation neural network methods. J Mech Syst Signal Process 21(1):193–207
9. IEEE (1994) IEEE standard glossary of computer hardware terminology. Institute of Electrical and Electronics Engineers, Nova York, 109 pp
10. Jinhua D, Erland O (2002) Availability analysis through relations between failure rate and preventive maintenance under condition monitoring. Department of Information Design and Product Development, Mälardalen University, Sweden, 21
11. Khoshgoftaar TM, Seliya N (2003) Fault prediction modeling for software quality estimation: comparing commonly used techniques. J Empir Softw Eng 8(3):255–283
12. Kohonen T (1997) Self-organizing maps. In: Series in information sciences, 2nd edn. Springer, Heidelberg (1997)
13. Koskela T, Varsta M, Heikkonen J, Kaski K (1998) Temporal sequence processing using recurrent SOM. In: 2nd international conference on knowledge-based intelligent electronic systems, Adelaide, Australia, vol 1(21), pp 290–297
14. Kundur P (1994) Power system stability and control. McGraw-Hill, New York, 1176 pp

15. Lathi BP (1998) Modern digital and analog communication systems. Oxford University, New York, 800 pp
16. Liu Y, Weisberg RH, Shay LK (2007) Current patterns on the west Florida shelf from joint self-organizing map analyses of HF radar and ADCP data. J Atmos Ocean Technol (Am Meteorol Soc) 24(4):702–712
17. Matos MC, Osório PLM, Johann PRS (2007) Unsupervised seismic pattern recognition using wavelet transform and self-organizing maps. J Geophys 72(1):9–21
18. Patan K, Korbicz J (2000) Application of dynamic neural networks in an industrial plant. In: Proceedings of 4th IFAC symposium on fault detection supervision and safety for technical processes, Budapeste, Hungria, vol 1(1), pp 186–191
19. Qiu H, Lee J, Lin J, Yu G (2006) Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics. J Sound Vib 289(4):1066–1090
20. Randall RB (2004) Detection and diagnosis of incipient bearing failure in helicopter gearboxes. J Eng Failure Anal 11(2):177–190
21. Saxena A, Saad A (2007) Evolving an artificial neural network classifier for condition monitoring of rotating mechanical systems. J Appl Soft Comput (Elsevier) 7(1):441–454
22. Senjyu T, Tamaki Y, Uezato K (2000) Current signature analysis of induction motor mechanical faults by wavelet packet decomposition. In: Proceedings of international conference on power system technology, vol 2(1), pp 1113–1118
23. Simon G, Lendasse A, Cottrell M, Fort JC, Verleysen M (2004) Double quantization of the regressor space for long-term time series prediction: method and proof of stability. J Neural Netw 17(1):1169–1181
24. Singh GK, Kazza SAS (2003) Induction machine drive condition monitoring and diagnostic research—a survey. J Electr Power Syst Res 64(2):145–158
25. Tan WW, Huo H (2005) A generic neurofuzzy model-based approach for detecting faults in induction motors. IEEE Trans Ind Electron 52(5):1420–1427
26. Tinós R, Terra M H (2001) Fault detection and isolation in robotic manipulators using a multilayer perceptron and a RBF network trained by the Kohonen's self-organizing map. Rev Soc Bras Autom Contr Autom 12(1):11–18
27. Varsta M, Heikkonen J, Lampinen J, Millán JR (2001) Temporal kohonen map and the recurrent self-organizing map: analytical and experimental comparison. Proc Neural Process Lett 13(1):237–251
28. Vesanto J (1997) Using the SOM and local models in time-series prediction. In: Proceedings of the workshop on self-organizing maps, Finland, pp 209–214
29. Voegtlin T (2002) Recursive self-organizing maps. Proc Neural Netw 15(8):979–991
30. Xilinx (2005) Embedded system tools reference manual. Embedded Development Kit EDK 8.1i, Xilinx Inc
31. Xu R, Wunsch D (2005) Survey of clustering algorithms. IEEE Trans Neural Netw 16(3):645–678
32. Yam RCM, Tse PW, Ling L, Tu P (2001) Intelligent predictive decision support system for condition-based maintenance. J Adv Manuf Technol 17(5):383–391
33. Yang BS, Han T, An JL (2004) Art–Kohonen neural network for fault diagnosis of rotating machinery. J Mech Syst Signal Process 18(3):645–657
34. Yan J, Lee J (2005) Degradation assessment and fault modes classification using logistic regression. J Manuf Sci Eng 127:912–914

Luiz Fernando Gonçalves received his B.Sc. in electrical engineering from Universidade Federal do Rio Grande do Sul, Brazil (2001) and M.Sc. in electric engineering from Universidade Federal of Rio Grande do Sul (2004). His experience in electric engineering focuses on the following subjects: intelligent maintenance, control systems and electromechanical systems test.

Jefferson Luiz Bosa received his B.Sc. in computer science from Universidade Estadual do Oeste do Paraná, Brazil (2006), and M.Sc. in computer science from Universidade Federal do Rio Grande do Sul (2009). His experience in computer science works mainly on the following topics: intelligent maintenance, microelectronics, system testing of hardware, computational mathematics, programming computer systems and free software.

Tiago Roberto Balen received his Electrical Engineering degree, in 2004, from the Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, Brazil, where he started his academic career working at the Prototiping and Test Laboratory. In 2006 and 2010, respectively, he received the M.Sc. degree and the Ph.D. degree also from Universidade Federal do Rio Grande do Sul. His research interests include analog and mixed-signal test, built-in-self test, programmable analog devices, fault tolerant circuits and radiation effects on electronic systems. In the past few years he has published papers on these topics in important conferences and journals, receiving a "Best Paper Award" for one of his contributions. Currently, he is professor and coordinator of the Telecommunications Engineering and Computer Engineering undergraduate courses at Centro Universitário La Salle (UNILASALLE), Canoas, Brazil. He is also with Centro de Excelência em Tecnologia Eletrônica Avançada - CEITEC civil association, where he works as Test Development Engineer.

Marcelo Soares Lubaszewski received the Electrical Engineering and M.Sc. degrees from the Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, Brazil, in 1986 and 1990, respectively. In 1994, he received the Ph.D. degree from the Institut National Polytechnique de Grenoble (INPG), France. In 2001, he joined the Laboratoire d'Informatique, Robotique et Microélectronique de Montpellier in France as an Invited Researcher for 3 months and, in 2004, the Instituto de Microelectrónica de Sevilla (IMSE) in Spain for 1 year. He is currently with UFRGS, where he has been a Professor since 1990. His primary research interests include design and test of mixed-signal, micro-electromechanical, corebased and NoC-based systems, self-checking and fault-tolerant architectures, and computer-aided testing. He has published over 200 papers in international journals and conferences on these topics. Dr. Lubaszewski has served as a Guest Editor of the Journal of Electronic Testing: Theory and Applications, of the Microelectronics Journal, as an Associate Editor of the Design and Test of Computers Magazine and as a lecturer for Latin America in the frame of the IEEE Computer Society Distinguished Visitors Program. He is presently a member of the Editorial Board of the VLSI Journal and the Editor-in-Chief of JICS, the Journal of Integrated Circuits and Systems of the Brazilian Computer and Microelectronics Societies. In the past, he has also served the IEEE CS Test Technology Technical Council as the Chair of the Latin-America Regional.

Eduardo Luis Schneider has Doctorate in Material and Metallurgical Engineering at Universidade Federal do Rio Grande do Sul, Brazil (2009). Works at Universidade Federal do Rio Grande do Sul, Brazil. He is currently professor of the Masters Program in Technology of Materials and Process Industries in the Institute for Science and Technology of Universidade Feevale. It also works as a researcher at the National Center for Supercomputing — CESUP. Has experience in Engineering Mechanics, Materials, Metallurgy and Product Design, working on the following topics: materials selection, manufacturing processes, ecodesign, product design, and metrology scales.

Renato Ventura Henriques received his B.Sc. in electrical engineering from Pontificia Universidade Catolica do Rio Grande do Sul, Brazil (1992) and M.Sc. in electrical engineering from the University of São Paulo, Brazil (1996) and Ph.D. in mechanical engineering from Universidade Federal de Minas Gerais, Brazil (2006). He is currently a professor at Federal University of Rio Grande do Sul teaching electrical engineering with emphasis in electrical process automation and electronics industry, focusing on the following topics: position control, cooperative robots, robotic welding, intelligent maintenance, Kalman filters, manufacturing systems and distributed control.