PERGAMON

Available online at www.sciencedirect.com

SCIENCE DIRECT®

Engineering Applications of Artificial Intelligence 16 (2003) 441–451

Engineering Applications of ARTIFICIAL INTELLIGENCE

www.elsevier.com/locate/engappai

Unsupervised and auto-adaptive neural architecture for on-line monitoring. Application to a hydraulic process

Christophe Lurette[a], Stéphane Lecoeuche[a,b,*]

[a] Laboratoire I3D, USTL, Bâtiment P2, 59655 Villeneuve d'Ascq, France

[b] Informatique et Productique, Ecole des Mines de Douai, 941, rue Charles Bourseul, BP838, 59508 Douai Cedex, France

Received 17 July 2002; accepted 15 May 2003

Abstract

Generally, the functioning of a monitored system is characterized by some parameters. Each evolution of these parameters may be assimilated with a drift or a fault. Therefore it may be considered to estimate the functioning state of the system by analyzing the drifts of these critical parameters evolving with time. So, this paper presents a new architecture, based on a new neural network classifier, for on-line monitoring of complex industrial systems. The first stage consists in the choice and the extraction of the significant parameters. The second stage extracts some parameter characteristics to be used in order to discriminate the different kinds of evolution. Then, in the third stage, a specialized neural network is used to classify, on-line, the evolutions for each significant parameter. The neural network is initialized with a set of synthetic and experimental data and a copy of this classifier is used to achieve the trend detection for each monitored parameter. Due to unsupervised rules and auto-adaptive abilities, each copy is adapted according to the specific evolutions of its monitored parameter in order to take account of new kinds of evolutions in situ and on-line. In the fourth stage, a decision tree is used to establish the diagnostic by analyzing the results of the trend detection stage for all the monitored parameters. The application to the monitoring of an industrial hydraulic process shows the ability of the classifier to learn and detect the different drifts.

Keywords: On-line monitoring; Fault detection; Drift; Classification; Unsupervised learning; Auto-adaptation; Hydraulic process

1. Introduction

Hydraulic systems are currently used in industrial applications, like in metal-forming forge (Frazier et al., 1997), mobile hydraulic crane (Münzer, 2000) or hot steel strip rolling mill (Bailey and Watton, 2000), because these systems provide the advantage of producing large forces at high speeds. A consequence of the strong mechanical load is a decrease of the life expectancy of some components like valves, pumps, accumulators or hydraulic jacks. So, to ensure reliable performance of hydraulic systems, curative and preventive maintenance activities are often used. Hydraulic components manufacturers recommend regular preventive controls. For example, Hydro Renée Leduc (2000) advises at least one control of the inflation pressure of its accumulators per month. In the same way, there are some preventive controls like fluid analysis and counting of particles to evaluate the oil pollution as described by Stecki (1998). Nowadays, a lot of companies propose efficient solutions to monitor the oil quality.

However, all these precautionary measures do not ensure a complete monitoring during the functioning of hydraulic systems, which is necessary for an efficient predictive maintenance. So, the early detection of faults in hydraulic components or circuits has been the subject of research works in the last decade. Most of the on-line monitoring techniques of hydraulic systems are restrictive using only vibrations and/or oil analysis. Sandt et al. (1997) have presented an on-line monitoring of the oil pollution for general hydraulic systems, and Kaye (1999) has shown the vibration analysis techniques for the monitoring of hydraulic machines. Recently, some general monitoring techniques have also been proposed by using a model of the hydraulic system and by

*Corresponding author. Tel.: +33-327-712-445; fax: +33-321-712-980.

E-mail addresses: christophe.lurette@ed.univ.lillel.fr (C. Lurette), lecoeuche@ensm.douai.fr (S. Lecoeuche).

0952-1976/03/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
doi:10.1016/S0952-1976(03)00064-2

analyzing its parameter evolutions to achieve a more complete and efficient diagnosis. For example, Zavarehi (1997) has proposed an on-line monitoring system based on the parameter estimation of the model of a valve orifice in a hydraulic process and Mourre and Burton (2001) have proposed a statistical condition monitoring system for a proportional solenoid valve based on physical measurements. Like many on-line monitoring techniques, these latter techniques often require the addition of other components to the initial equipment (here valves) and then, very often the on-line use is more expensive and more complicated. Nevertheless, this parameter extraction approach is efficient if the general monitored system could be decomposed into different component subsets and if, for each subset, one independent model could be extracted.

The estimation of parameters, characterizing the functioning of the system, for the on-line monitoring, is needed during service. The first approach is to use significant process data to define signatures of the state of the components by direct measurements. For hydraulic and fluid power systems, vibration amplitudes, noise intensity, pressures, leakage flow rates, temperatures and fluid analysis are often used as the indicators of the system health. But, in the case of complex systems, the relationships between different process data are too difficult to analyze. The second approach is to use the coefficients of an identification model of the system as parameters. To find such a parametric model of the system, i.e. a black-box model, many standard methods based on on-line identification techniques can be applied (Sjöberg, 1996; Ljung, 1999). For a hydraulic process, the estimation of these parameters during functioning is often difficult, because some non-linearities appear due to the compressibility of the hydraulic fluid or complex flow properties, and then the order estimation and delay choice are not always straightforward. So, modeling for fault detection and diagnosis must be very precise because errors in the model may be interpreted as faults.

To sum up, the monitoring and the diagnosis of the functioning of a complex system could be achieved by surveying some parameters on-line (parametric identification model of the real system, or particular physical parameters measured directly by sensors). In cases of drifts or faults, these parameters would deviate, slowly or sharply, would shift, upwards or downwards, according to the evolution of the system with time. For example, in a hydraulic process, a drift may appear at different moments due to oil pollution, or some fluctuations of the oil temperature.

Based on the analysis of the evolutions of characteristic parameters, the proposed on-line monitoring concept is defined in four steps (see Fig. 1).

In the first step, some parameters characterizing the system health are estimated (stemmed from process data or obtained by identification). In the second step, from raw data, a vector of discriminators is extracted for each parameter in order to characterize its evolutions according to an observation window. In the third step, the proposed neural network classifier is used to determine for each parameter (more precisely for each discriminators vector) its kind of evolution. Then a final decision tree permits a diagnosis based on the result of the trend classification for all the monitored parameters.

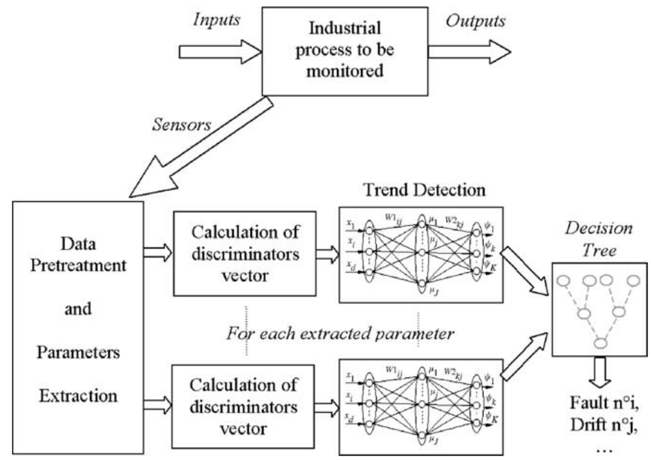


Fig. 1. Architecture of the monitoring system.

The core of the system being the trend detection by classification techniques, the present study focuses on the development of a new neural network classifier, adapted from Lurette and Lecoeuche (2001) with abilities to learn and detect different kinds of drifts and faults for an application to the on-line monitoring of an industrial system. Neural network techniques can be applied to fault diagnosis using different approaches like pattern recognition or residual generation decision (He et al., 2000; Patton et al., 2000). Crowther et al. (1998) have shown an application of a neural network to the fault diagnosis of hydraulic actuators. Artificial neural networks provide an interesting approach to represent the fault diagnosis model by a classification model and then to learn and to classify the mapping between data drifts and the diagnosis of the system. Another benefit is that the experimental faults can be diagnosed using neural networks trained both on simulation data and on trial-and-test data.

Neural networks are widely used for classification of natural data (Bishop, 1995; Looney, 1997; Gupta et al., 2002). These techniques have shown their abilities to achieve this work easily. A lot of improvements have been developed in order to label the data in the space of representation in very precise partitions by describing connex shapes (Gupta et al., 2000), radial shapes (Gomm and Yu, 2000), and elliptic shapes (Mak and Kung, 2000). This capacity is important in order to clearly learn the class model of each kind of evolution. Moreover, some future cases might not be predicted and then new partitions (new classes of trend) of the space representation would have to be created, or some classes would have to be adapted. In this way, unsupervised learning and auto-adaptation abilities have been introduced in the classification techniques (Mao and Jain, 1996; Zheng et al., 1999).

A number of neural networks have been developed with unsupervised learning or with auto-adaptation abilities but only few of them are able to adapt their architecture according to the apparition of new classes, by creating new neurons, and/or to the evolution of existing classes, by adapting their model. Among these architectures, it is possible to mention the Cluster detection and labeling network (CDL) (Eltoft and Rui Defigueiredo, 1998). Lurette and Lecoeuche (2001) have proposed a modification of this architecture to authorize its on-line use by improving the definition of the prototypes obtained per class in terms of number, size, and shape.

The developed architecture is based on unsupervised learning rules and auto-adaptive structure in order to learn new faults and drifts occurring during the life of the monitored system. Each class is defined by one or several prototypes. For the definition of each prototype, an elliptical shape is introduced to ensure a better representation of the model of classes, and to limit the number of neurons for an on-line use.

A detailed presentation of the developed neural network is given in Section 2, with the description of the architecture, the learning rules, and the auto-adaptation abilities. In Section 3, a description of its use for monitoring is given. In Section 4, the on-line monitoring of an industrial hydraulic process is realized to experiment the abilities of the classifier to detect different evolutions of the behavior of some components in the hydraulic system.

2. Description of the developed neural network classifier

The architecture of the developed neural network classifier is inspired by the CDL network (Eltoft and Rui Defigueiredo, 1998) which is a classical three-layered, feed-forward network. The main advantage of this CDL network is its auto-adaptive architecture owing to its learning rules for the creation of new neurons according to the creation of prototypes and/or classes. But, the limits of this architecture are found in the restrictive definition of the prototypes (a circular shape), a binary membership degree (which does not define the neighborhood of a prototype), and the large number of prototypes that are obtained.

2.1. Structure of the neural network

The input layer consists of as many neurons as components of the input vector called $X_i=[x_j^i \ \dots \ x_d^i]^{\mathrm{T}}$. The hidden layer is totally connected to the input layer (W^1), and each neuron represents a prototype P_j of a class (see Fig. 2). A prototype is defined by a set of close input vectors.

The output of each neuron of the hidden layer is defined as the membership degree, Eq. (1), of an example X_i to the corresponding prototype P_j.

$$\mu(P_j, X_i) = \exp\left(-\frac{(\mathrm{d}(P_j, X_i))^2}{2\alpha_{P_j}}\right). \tag{1}$$

A normalization factor α_{P_j} has been introduced in Eq. (1). The role of this factor will be presented in Section 2.2. The distance is chosen as the Mahalanobis distance, Eq. (2), in order to give the prototypes a hyper-elliptical shape, for a better description of their neighborhood.

$$\mathrm{d}(P_j, X_i)^2 = (X_i - M_{P_j})^{\mathrm{T}} \ \Sigma_{P_j}^{-1}(X_i - M_{P_j}). \tag{2}$$

So, each prototype is defined by its mean vector M_{P_j} (position in the representation space), its full covariance matrix Σ_{P_j} (shape of its neighborhood), and its normalization factor α_{P_j}.

The output layer consists of as many neurons as detected classes. The connection weights (W^2) between the hidden layer and the output layer characterize the relation between prototypes and classes. The output of each neuron of the output layer defines the membership degree, as defined by Eq. (3), of example X_i to its corresponding class C_k.

$$\Psi(C_k, X_i) = \min\left(1, \sum_{P_j \in C_k} \mu(P_j, X_i)\right). \tag{3}$$

The choice of defining one class by one or several prototypes permits the definition of complex shapes for classes. For example, a class of typical trend may be divided into several partitions of the space representation.

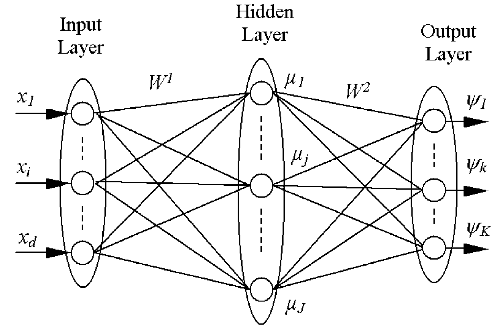


Fig. 2. Architecture of the neural network classifier.

The general architecture of the network (Fig. 2) resembles a classical RBF neural network, but the main advantage of the proposed architecture is the use of a new learning principle with unsupervised learning rules. This gives the new neural network a real and complete auto-adaptive structure in terms of hidden and output nodes.

2.2. Learning principle of the proposed neural network

A particular learning principle has been developed in order to authorize the adaptation of this architecture, not only by adapting the connection weights but also by modifying the size of the hidden layer and the size of the output layer. The iterative learning of the network is then made in three stages, as described in Fig. 3.

2.2.1. Classification without merging

The first stage is called "classification without merging". The particularity of this stage is to make the creation of new prototypes (creation of nodes on the hidden layer) and new classes (creation of nodes on the output layer) possible, when input vectors are too different from known prototypes. It is also possible to adapt existing prototypes when input vectors are not so far from known prototypes. For each new input vector, the membership degree of X_i is calculated (Eq. (1)) for each known prototype and compared with two thresholds $\mu_{\min}$ and $\mu_{\max}$. Different cases could occur; they are summarized in Table 1.

To illustrate the different cases described in Table 1, an example is presented in Fig. 4, where four prototypes P_1, P_2, P_3, P_4 are already known, and define 3 classes C_1, C_2, C_3. The notation $P_j^{C_k}$ defines the prototype P_j belonging to the class C_k. Four new input vectors, X_1, X_2, X_3, X_4, are presented successively to the input layer of the neural network.

The first case in Table 1 is illustrated by the input vector X_1. This example is too far from all existing prototypes. Hence, a new prototype P_3 and a new class C_4 are created.

The input vector X_2, representing the second case in Table 1, has a membership degree greater than the threshold $\mu_{\min}$ for prototypes P_3 and P_4 belonging to the

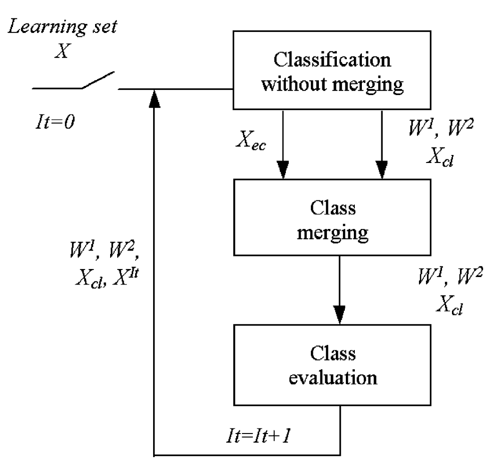


Fig. 3. Learning stages of the neural network.

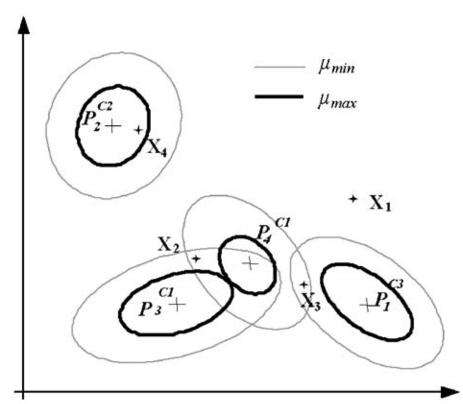


Fig. 4. Illustration of the membership degree of input vectors to prototypes (d = 2).

Table 1
Different cases for the first stage of the learning principle

	If	Then
First case	The membership degree between an example and any known prototype is smaller than the first threshold ($\mu(P_j,X_i)<\mu_{\min}$ for all P_j).	The example is not close to any prototype. It is necessary to create a new prototype (a new neuron on the hidden layer) and a new class (a new neuron on the output layer).
Second case	The membership degree is larger than the first threshold but smaller than the second threshold for prototypes P_j that belong to the same class. ($\mu_{\min}<\mu(P_j,X_i)<\mu_{\max}$).	The example is close to these prototypes, but not close enough to be associated with one of them. It is necessary to create a new prototype. This new prototype and prototypes P_j belong to the same class.
Third case	The membership degree is larger than the first threshold for multiple prototypes that belong to different classes.	The ambiguity will be analyzed during the next stage of the learning procedure. (X_{ec})
Fourth case	The membership degree is larger than the two thresholds for some prototypes P_j that belong to the same class ($\mu_{\min}<\mu_{\max}<\mu(P_j,X_i)$).	The example is associated to the closest prototype P_j and to its class. The prototype P_j is then adapted.

same class C_1. A new prototype P_6 is created and associated to C_1.

The input vector X_3 presents a membership degree greater than the threshold $\mu_{\min}$ for prototypes P_1 and P_4 belonging, respectively, to the classes C_3 and C_1 (Third case in Table 1). So, X, is considered as an ambiguous example and is memorized in X_{ec} for the next stage.

Finally, the input vector X_4 illustrates the last case. X_4 presents a membership degree greater than the threshold $\mu_{\max}$ for the prototype P_2 belonging to the class C_2. Then, X_4 leads to the adaptation of the prototype P_2. As an example in the monitoring concept, this case will correspond to the adaptation of the model of the class of normal state in order to take into account some new small variations of one characteristic of the monitored parameter.

Then, in this situation the concerned prototype P_2 is adapted in the same way as Zheng et al. (1999), but with a full covariance matrix Σ_{P_2} as is done by Mak and Kung (2000). This adaptation is performed in an iterative manner using the following relations.

$$M_{P_j}^{k+1} = \frac{k}{k+1}\, M_{P_j}^{k} + \frac{1}{k+1}\, X_i. \tag{4}$$

$$\Sigma_{P_j}^{k+1} = \frac{k-1}{k}\, \Sigma_{P_j}^{k} + \frac{1}{(k+1)}\,(X_i - M_{P_j}^{k})(X_i - M_{P_j}^{k})^{\mathrm{T}}. \tag{5}$$

In addition, the normalization factor α_{P_j}, defined in Eq. (1), is adapted owing to Eq. (6). Mak and Kung (2000) have introduced an equivalent coefficient as a smoothing parameter. Here, α_{P_j} ensures a membership degree greater than $\mu_{\max}$ for all the examples that have already been associated to the adapted prototype.

$$\alpha_{P_j}^{k+1} = \max_{X_i \in P_j} \left(\frac{(X_i - M_{P_j}^{k+1})^{\mathrm{T}}\, \Sigma_{P_j}^{-1}\, (X_i - M_{P_j}^{k+1})}{-2\,\ln(\mu_{\max})} \right). \tag{6}$$

The effect of the adaptation of a prototype with a new input vector is illustrated by Fig. 5.

At the end of this first stage, when all the input vectors have been presented, a set of classified examples is defined by X_{cl}. All the ambiguous examples are put together in X_{ec}.

2.2.2. Class merging

The "class merging" stage is dedicated to prototypes and classes that are close in the space representation. If an ambiguity is detected during the previous stage (examples memorized in X_{ec} during the first stage), as shown in the third case in Table 1, the "class merging" stage resolves it by merging the different ambiguous prototypes into the same class. So, the output layer is modified by the elimination of the neurons that defined the ambiguous classes in order to let a unique neuron which is the result of the merged classes. In the example in Fig. 4, C_3 merges with C_1.

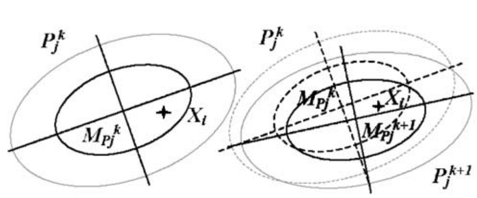


Fig. 5. Illustration of the adaptation of the center and of the covariance matrix of a prototype (d = 2).

2.2.3. Class evaluation

The "class evaluation" stage makes possible the characterization of the existing prototypes and classes. Classes and prototypes containing less assigned examples than the two respective thresholds $N_{\min}^{P}$ and $N_{\min}^{C}$ are eliminated. Then, the corresponding examples are marked as "unclassified" and will be re-presented at the next iteration until a minimum percentage of classified data (defined by $P_{\min}$) is reached. The first threshold ($N_{\min}^{P}$) is required to wait a sufficient number of examples before using Eq. (5) which makes the iterative adaptation of the full covariance matrix possible. Before this limit, the prototype is initialized with Σ_{ini}. The second one ($N_{\min}^{C}$) is used to avoid small classes, which are not significant due to very few associated examples.

3. How to use the classifier in a monitoring system

The neural network classifier presented in Section 2 has been developed to be used in a monitoring system. Its goal is to detect any deviation of different parameters, those that characterize the functioning of the system, and also to recognize the type of deviation. In this section, the overall principle of the monitoring system is described and an application to synthetic time series is given.

3.1. Principle of the monitoring system

Obviously, the first step consists in data acquisition on the system to be monitored, and the definition of a set of significant parameters. The parameters extraction is based on the use of a sliding window on which, for each datum (stemmed from process data or obtained by identification) evolving in time according to the evolution of the system health, a vector of normalized discriminators is computed. A judicious choice of the components of this vector has to be made to allow an easy discrimination of the different kinds of possible trends. According to this principle, it has been chosen to compute for each windowed data-signal the following three values:

- the maximum amplitude in the window spectrum,
- the slope of the linear regression,

- the maximum difference between two consecutive coefficients of the local ARX model (Ljung, 1999).

To be independent of the values of the data-signal and the size of the sliding window, a normalization of the data-signal is performed for each observation window. In the same way, the definition of the representative vector is followed by a normalization stage of its component discriminators in order to reduce the classification space. In this latter case, the normalization coefficients have to be fixed during the training of the neural network.

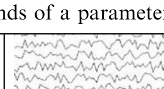
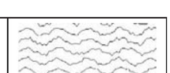
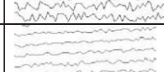
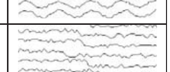
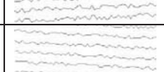
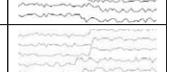
The developed neural network classifier is used in order to associate each vector to a class of trend. To do that, the neural network classifier is first of all initialized during a learning stage with the time series presented in Table 2, and/or with a set of experiences corresponding to some faults and drifts of the monitored system that are already known or voluntarily performed. When the learning stage is achieved, a copy of this initial trained network will be used for each monitored parameter to detect its evolutions (see Fig. 1). During this "trend detection" stage and owing to the network auto-adaptive abilities, an on-line learning could also be achieved in order to improve each classifier to the specific evolutions of its monitored parameter. To conclude, a decision tree permits the association of the different results of trend detection in order to establish a diagnosis. For example, the association of an upward shift of a parameter with a downward shift of another parameter permits the diagnosis of the fault no. *i*.

3.2. Illustration of the abilities of the classifier

The abilities of the neural network to learn and detect different kinds of trends have been evaluated by Lecoeuche (2002), using the "Synthetic Control Chart Time Series" database (Pham and Chan, 1998), described in Table 2. The choice of this database is interesting because each time series can be associated to a typical kind of evolution of a parameter of a real system, and therefore to a drift or a fault of this system.

Fig. 6 shows the result of the learning of the different classes of trends with the database presented in Table 2,

Table 2
Examples of trends of a parameter

Normal (zoom)		Cyclic	
Increasing trend		Downward shift	
Decreasing trend		Upward shift	

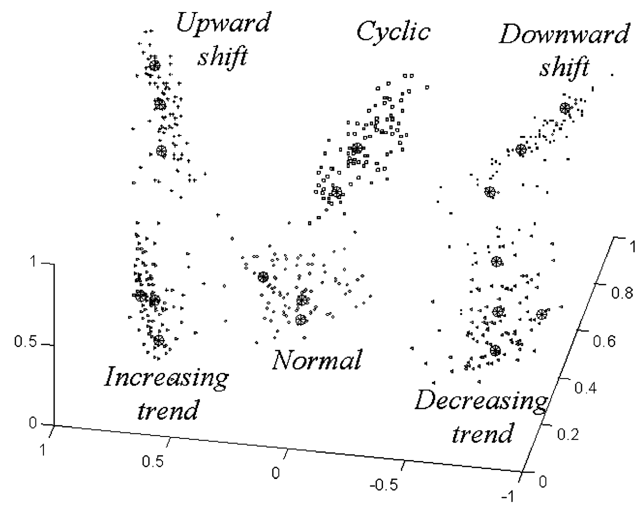


Fig. 6. Illustration of the classification of the six classes of trend proposed in Table 2, with the 18 prototypes obtained.

using the following parameters for the neural network classifier: $\mu_{\min} = 0.3$; $\mu_{\max} = 0.6$; $\Sigma_{\text{ini}} = 0.12$; $N^P_{\min} = 5$; $N^C_{\min} = 10$; $P_{\min} = 80$.

4. Application to the monitoring of an industrial hydraulic process

In this section, the proposed development of the neural network classifier is applied to the monitoring of the critical parameters of an industrial hydraulic system. The monitoring system is applied to the detection of trends of the hydraulic process to guarantee, on-line, its safe use.

4.1. Description of the hydraulic process

The used system is a real industrial hydraulic process that is illustrated in Fig. 7. Its structure is similar to the hydraulic technology defined by Frazier et al. (1997) used as a forging process.

4.1.1. Description and instrumentation

A brief description of the hydraulic components and sensors is given by Fig. 7, where the following notations are used:

P_a is the pressure in chamber A (Pa), P_b the pressure in chamber B (Pa), P_p the pressure in hydraulic circuit P (Pa), P_t the pressure in hydraulic tank (Pa), Q_c the accumulator flow rate (m^3/s), X the piston displacement (m), U the input tension of the valve (V), and C the control of the position profile (m).

The process data are acquired at a sample time of 5 ms and the considered cycle of functioning is 2 s (see Fig. 8).

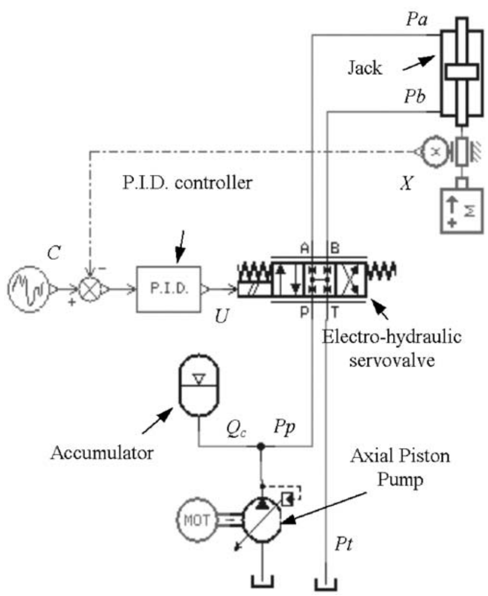


Fig. 7. Description of the hydraulic process to be monitored. (AMESIM®, produced by Image S.A.).

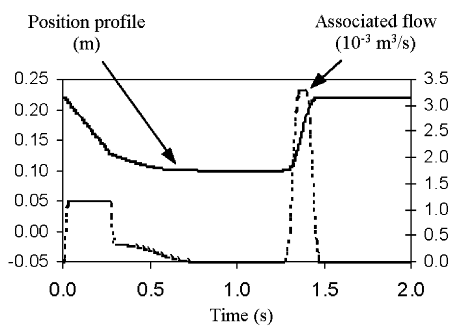


Fig. 8. Presentation of the functioning cycle used for the experiment.

4.1.2. Control of the piston displacement

The aim of the hydraulic system is to adjust the piston position according to a profile displacement C, owing to the tension U on the servo-valve. In this way, the axial piston pump is mechanically controlled to ensure a pre-defined service pressure in the hydraulic circuit. An accumulator, used as storage of hydraulic energy, is fixed on the circuit not only to compensate the response time of this pump, but also to reduce the pressure oscillations (Ijas and Virvalo, 1999). The hydraulic energy is then sent to one of the chambers of the hydraulic jack by the servo-valve whose input tension is generated by the controller.

An example of the profile position of the piston, used in the experiment, is presented in Fig. 8, with the needed flow.

4.2. Some possible faults and drifts

In the hydraulic system defined by Fig. 7, lots of faults or drifts may occur, and they are not always perceptible to the operator. The most common are the abrasion of the mechanical components of the servo-valve, the oil pollution, a degradation of the quality of functioning of the accumulator, or also the breaking of one piston of the pump. Our attention has been focused on some drifts and faults of the hydraulic accumulator and of the pump because the cost of these faults and especially of their consequences if they are not detected early are more expensive for the considered system. Before explaining the influence of the considered faults and drifts, a description of the pump and the accumulator is given for a better understanding of the choice of the parameters to be monitored.

4.2.1. Description of the pump and of the accumulator

Fig. 9 gives a simplified scheme of the pump which is more exactly an axial piston pump. The control piston permits to adjust a pressure in the hydraulic circuit by modifying the inclination of the swashplate, i.e. flow of the pump according to the volume defined by the piston assemblies.

A description of the hydraulic accumulator is given in Fig. 10, where P_c is the Azote pressure and V_c is the Azote volume at pressure P_c. So, the accumulator flow rate Q_c depends on the differential pressure between the hydraulic fluid pressure P_c in the accumulator and the hydraulic fluid pressure P_p in the circuit.

The accumulator is considered as a storage of hydraulic energy in addition to the pump.

4.2.2. Description of the considered fault and drifts

The fault, defined as fault #1, is a sudden breakdown of the membrane of the accumulator leading to the

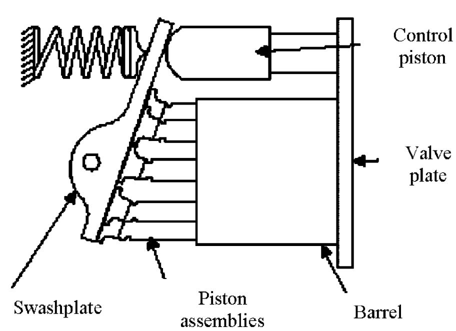


Fig. 9. Key components of the variable displacement. Axial piston pump (Dobchuk et al., 2000).

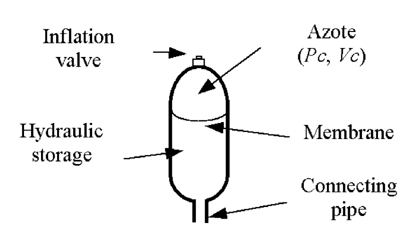


Fig. 10. Description of the hydraulic accumulator.

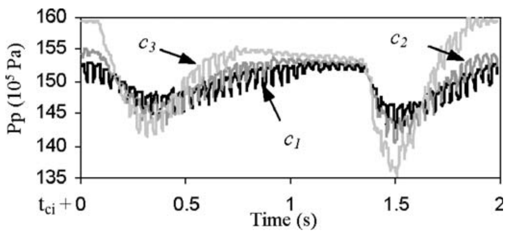


Fig. 11. Comparison of the evolution of P_p for different cycles of functioning ($t_{c_1} < t_{c_2} < t_{c_3}$) with a leakage in the inflation valve.

apparition of important oscillations of the pressure P_p in the hydraulic circuit due to the complete deterioration of the accumulator.

The first drift early considered, later referred to as drift #1, is a progressive degradation of the piston assemblies of the pump leading to a diminution of its cylinder volume and therefore of its maximal flow.

The second drift to be considered, called drift #2, is due to a diminution of the pre-charge of the accumulator which appears due to a leakage in the inflation valve.

As shown in Fig. 11, when this kind of degradation appears on the hydraulic system, more oscillations of the hydraulic pressure occur in the circuit. The comparison is made for three cycles of functioning c_1, c_2, c_3 given for three different times.

The oscillations at the highest frequencies are normal, due to the technology of the pump with its piston assemblies.

The early detection of this second drift is important because if a deterioration of the accumulator occurs and is not detected early enough, then the pump will be abnormally used due to a more constraining load and a degradation of the pump could also appear.

4.3. Monitoring of the hydraulic process

As described in Fig. 7, several sensors are used on the hydraulic process to define its state of functioning with some physical parameters. For illustrations of considered drifts and faults, here, only pressure P_b and flow Q_c will be monitored.

4.3.1. Definition of the input vector

The first step consists in the choice of the relevant parameters for the detection of the trends corresponding to the faults or drifts to be detected. In this way and according to the considered fault and drifts of the pump and of the accumulator, described in Section 4.2, two parameters are defined on each cycle:

- $P1$: the variance of the hydraulic pressure P_p in the circuit,
- $P2$: the maximum instantaneous flow rate Q_c of the accumulator.

So, the parameter $P1$ gives an information of the oscillations on the pressure signal, and $P2$ gives an indication of the level of use of the accumulator during one cycle. $P1$ and $P2$ are defined with a 2 s sampling time.

By considering a sliding window on these two parameters evolving in time, two input vectors of the neural network classifier are defined with the three discriminators presented in Section 3.1.

4.3.2. Learning procedure of the classifier

As presented in Section 3.1, the learning of the classifier is achieved by two learning sets: one with synthetic data and the other with some characteristic experimental data. This allows to avoid the use of a lot of expensive experiments to establish good knowledge while validating the first learning by introducing some experimental knowledge defined on the system. So, the learning procedure is enriched with a second stage by using a second database which corresponds to some experimental data of the evolution of concerned parameters.

Some main defaults have been voluntarily carried out on the system during a set of about 20 experiments realized on the real system in order to obtain some experimental evolutions of the parameters concerned by the monitoring process. Then, this second database has been used to improve the definition of the classes obtained by the developed neural network owing to its auto-adaptative learning abilities. The experimental database has also permitted to acquire experience of the system and to initialize some rules for the decision tree used for the diagnosis of the system.

For illustration, Figs. 12a and b represent the two evolutions of parameters $P1$ and $P2$ obtained by progressively opening the inflation valve of the accumulator. Using a sliding window of 1 h on each parameter, different input vectors have been extracted during the normal state of the system and during the drift. Then, by considering the windows no. 4 and no. 10 for each parameter, four new learning vectors are defined. Fig. 12c represents, after the learning phase, the location of these vectors. It can be seen that each vector is correctly located in the representation space obtained during the synthetic learning (see Fig. 6): $X_4^{P1}$and X_4^{P2} belong to the class of normal state, X_{10}^{P1} and X_{10}^{P2} belong, respectively, to the class of increasing trend and of decreasing trend.

4.3.3. Results of the monitoring

To conclude on the abilities of the classifier to be used in an on-line monitoring system, the results are shown in terms of detection (Table 3) and of learning by auto-adaptation during use (Fig. 13). The results correspond

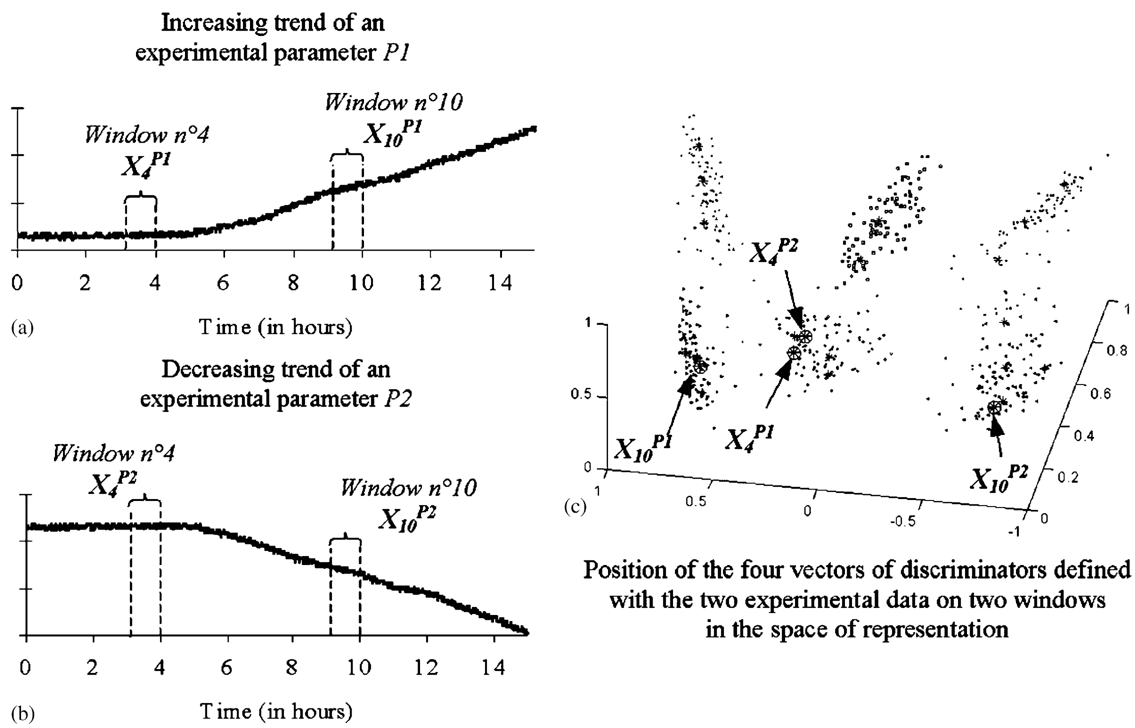


Fig. 12. Validation of the learning stage (c) with two experimental parameters (a) and (b).

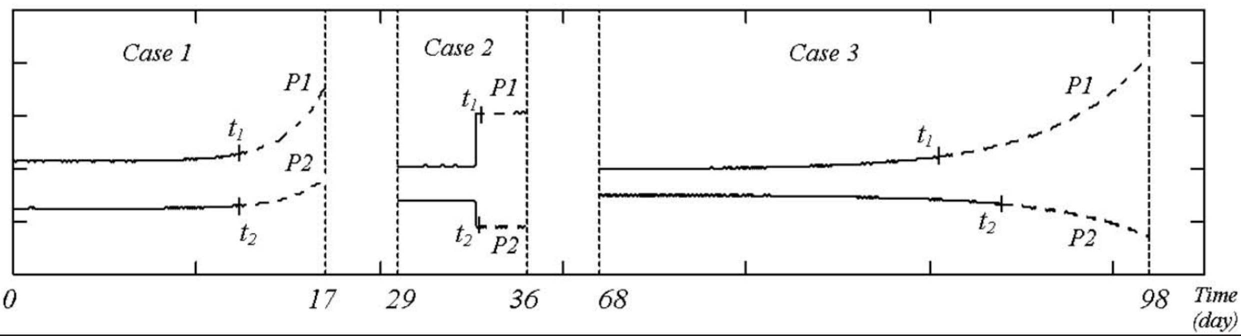
to the occurrence of the fault and drifts presented in Section 4.2, obtained during approximately 3 months of functioning. The first row of Table 3 describes the evolutions of the two parameters $P1$ and $P2$. In the second row, the results of the detection of trend on each parameter are presented and a diagnosis is made owing to a decision tree. The three cases illustrated correspond to the fault and drifts defined in Section 4.2. The first case is the drift #1, the second case is the fault #1 and the third case is the drift #2.

The evolution detection of the two parameters $P1$ and $P2$ is then realized using a sliding window. The sliding window is defined with two significant parameters: its size (N) and its offset (O). The size is fixed according to the quality of the trend detection of the parameter to be monitored. For example, if the parameter evolves as a slow trend, N will be high for a good detection. The offset is defined according to the accuracy of the shift detection. In case of a brutal breakdown, the detection has to be fast: achieved if O is small. The results presented in Table 3 have been obtained using the neural network classifier with the same parameters as those presented in Section 3.2, and by defining a sliding window of size N corresponding to 1 day of functioning, and an offset O of 15 min. In order to maintain the same signal length, the data signal is resampled at 48 s. According to the choice of these two parameters, the drifts illustrated by cases 1 and 3 are detected early with, in case 1 the same moment of detection (nearly the same, less than 2 h) for the two parameters although in case 2 there is a delay of approximately 3 days between the detection for each parameter. In the same idea, the fault illustrated by the second case will be detected (in the worst case) in 15 min. In this case, from a window with only less than 1% of data concerned by the fault, the rupture is detected by the parameter based on the ARX model.

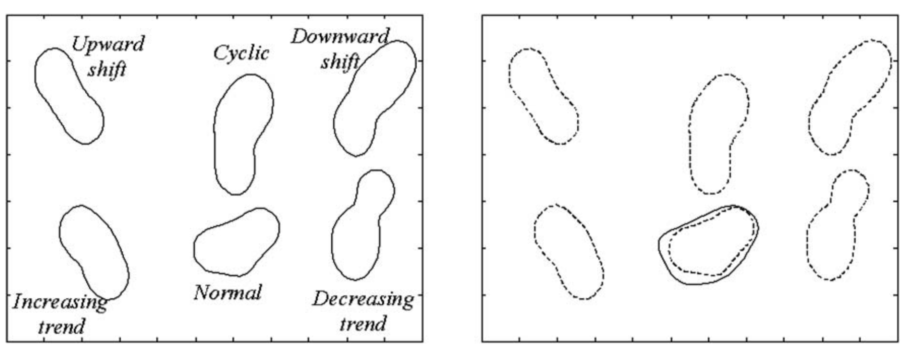
The benefits of the auto-adaptation lead to take account of new kinds of evolutions during on-line functioning. Each neural network, stemming from a unique learned network and dedicated to one monitored parameter, could be adapted by using on-line data. For example, the known classes may be adapted if the parameter data are more or less noisy. Fig. 13 illustrates the ability of the classifier to adapt the prototypes and the class of normal state during service owing to its auto-adaptive abilities and unsupervised learning rules. The first column of Fig. 13 describes the classes learned just after the initialization procedure with the time series database. The boundary of each class is defined by a membership degree $\Psi = 0.5$ (see Eq. (3)). The second column illustrates the result of the auto-adaptation of these boundaries with on-line data. The data used for this on-line learning adaptation are extracted in the same manner as those used for the detection but by using a sliding window of size N_L, with N_L three times greater than N, to assure a good adaptation

Table 3
Results of the experiment on the hydraulic in terms of detection

Evolution of the two selected parameters P1 and P2



An increasing trend of the parameter $P1$ and an increasing trend of the parameter $P2$ are detected after time $t=t_1=t_2$.	An upward shift of the parameter $P1$ and downward shift of the parameter $P2$ are detected after time $t=t_1=t_2$.	An increasing trend of the parameter $P1$ is detected after time $t=t_1$ and a decreasing trend of the parameter $P2$ is detected after time $t=t_2$.
According to the decision tree, this combination is associated to a drift of the functioning of the pump.	*According to the decision tree, this combination is associated to a sudden breakdown of the membrane of the accumulator.*	*According to the decision tree, after time $t=t_2$, this combination is associated to a leakage of the inflation valve of the accumulator.*



Definition of the six classes after the initialization. Definition of the six classes after the on-line use.

Fig. 13. Results of the experimentation on the hydraulic process in terms of learning by auto-adaptation.

of the classes and also to avoid a merging of several classes.

5. Conclusion

In this article, a new neural network classifier is developed to be used in the on-line monitoring of industrial systems in order to detect the evolution of a system early. In this way, the classifier allows the detection and the recognition of different kinds of evolutions of some parameters characterizing the state of functioning of the system.

The main advantages of the developed network are the unsupervised learning and the auto-adaptation abilities of its architecture that permit to improve the model of the classes of trend with particular input vectors. In this way, the learning rules are presented in order to not only create new prototypes or new classes if necessary, but also to adapt the existing prototypes and therefore the definition of classes. Moreover, owing to the definition of the prototypes with a full covariance matrix, the models of classes are well-defined.

The application of the neural network classifier to the on-line monitoring of an industrial hydraulic process shows good abilities to early detect some drifts and faults of the hydraulic process. The initial learning of the classifier is realized with synthetic time series and improved with a set of experiences on the system to be monitored. Based on physical parameters of the system, the proposed monitoring architecture achieves the trend detection stage by a dedicated neural network classifier to each monitored parameter. A decision tree is finally used to establish the final diagnosis according to the trend detection results.

References

Bailey, S., Watton, J., 2000. Progress of the on-line monitoring of the hydraulic systems of a hot steel strip rolling mill. Proceedings of the FPNI-PhD Symposium, Hamburg, pp. 203–213.

Bishop, C., 1995. Neural Networks for Pattern Recognition. Oxford University Press, Oxford.

Crowther, W.J., Edge, K.A., Burrows, C.R., Atkinson, R.M., Woollons, D.J., 1998. Fault diagnosis of a hydraulic actuator circuit using neural networks an output vector space classification approach. Journal of Systems & Control Engineering 212 (1), 57–68.

Dobchuck, J.W., Nikiforuk, P.N., Ukrainetz, P.R., Burton, R.T., 2000. Effect of internal pump dynamics on control piston pressure ripple. Department of Mechanical Engineering, University of Saskatchewan, Canada.

Eltoft, T., Rui Defigueiredo, J.P., 1998. A new neural network for cluster-detection-and-labeling. IEEE Transactions on Neural Networks 9 (5), 1021–1035.

Frazier, W.G., Medina, E.A., Mullins, W.M., Irwin, R.D., 1997. Modeling and simulation metal-forming equipment. Journal of Materials Engineering and Performance 49 (4).

Gomm, J.B., Yu, D.L., 2000. Selecting radial basis function network centers with recursive orthogonal least squares training. IEEE Transactions on Neural Networks 11 (2), 306–314.

Gupta, L., McAvoy, M., Phegley, J., 2000. Classification of temporal sequences via prediction using the simple recurrent neural network. Pattern Recognition 33 (10), 1759–1770.

Gupta, M.M., Homma, N., Jin, L., 2002. Static and Dynamic Neural Networks: From Fundamentals to Advanced Theory. Wiley, New York.

He, J.Z., Zhou, Z.H., Yin, X.R., Chen, S.F., 2000. Neural fault diagnosis. Conference EANN2000, Kingston, England.

Hydro Renée Leduc, 2000. Accumulateurs hydropneumatiques. Précautions maintenance. Ref AQ55 G.

Ijas, M., Virvalo, T., 1999. Experimental study of hydraulic pulsation dampers for low frequencies. In: Proceedings of Sixth Triennal International Symposium on Fluid Control, Measurement and Visualization, Sherbrooke, Canada.

Imagine S.A. Analysis and design of fluid power mechatronic systems. AMESIM®, Fluid power simulation software, Training course, Roanne, France.

Kaye, M., 1999. Cavitation monitoring in hydraulic machines by vibration analysis. Ph.D Thesis, no. 2112, Lausanne University, Switzerland.

Lecoeuche, S., 2002. Unsupervised classifier for monitoring and diagnostic of time series. Conference ESANN 2002, Bruges, Belgium.

Ljung, L., 1999. System Identification. Theory for the User, 2nd Edition. Prentice-Hall, Englewood Cliffs, NJ.

Looney, C.G., 1997. Pattern Recognition using Neural Networks: Theory and Algorithms for Engineers and Scientists. Oxford University Press, New York.

Lurette, C., Lecoeuche, S., 2001. Improvement of cluster detection and labeling neural network by introducing elliptical basis functions. Conference ICANN 2001, Vienna, Austria.

Mak, M.W., Kung, S.Y., 2000. Estimation of elliptical basis function parameters by the EM algorithm with application to speaker verification. IEEE Transactions on Neural Networks 11 (4), 961–969.

Mao, J., Jain, A.K., 1996. A self-organizing network for hyperellipsoïdal clustering (HEC). IEEE Transactions on Neural Networks 11 (4), 16–29.

Mourre, D., Burton, R., 2001. Investigation of a neural network/statistical condition monitoring technique for a proportional solenoid valve. Department of Mechanical Engineering, University of Saskatchewan, Canada.

Münzer, M.E., 2000. Control of mobile hydraulic cranes. Proceedings of the FPNI-PhD Symposium, Hamburg, pp. 475–483.

Patton, R.J., Uppal, F.J., Lopez-Toribio, C.J., 2000. Soft computing approaches to fault diagnosis for dynamic systems: a survey. Fourth IFAC Symposium on Fault Detection Supervision and Safety for Technical Processes, Budapest, Hungary.

Pham, D.T., Chan, A.B., 1998. Control chart pattern recognition using a new type of self organizing neural networks. Proceedings of the Institution of Mechanical Engineers 212 (1), 115–127.

Sandt, J., Rinkinen, J., Laukka, J., 1997. Particle and water on-line monitoring for hydraulic system diagnosis. The Fifth Scandinavian International Conference on Fluid Power 1997, Linköping, Sweden, Vol. 3. pp. 257–268.

Sjöberg, J., 1996. Non-linear system identification with neural networks. Ph.D. Thesis, Department of Electrical Engineering, Linköping University, Sweden.

Stecki, J.S., 1998. Total Contamination Control. Fluid Power Net publication Ltd., Australia, ISBN 0-646-35306-3.

Zavarehi, M.K., 1997. On-line condition monitoring and fault diagnosis in hydraulic system components using parameter estimation and pattern classification. Ph.D. Thesis, Department of Mechanical Engineering, University of British Columbia.

Zheng, N., Zhang, Z., Shi, G., Qiao, Y., 1999. Self-creating and adaptive learning of RBF networks: merging soft-competition clustering algorithm with network growth technique. International Joint Conference on Neural Networks. Washington, DC, USA.