machine diagnosis comprises the automated detection and classification of faults, whereas machine prognosis is the automated estimation of how soon and likely a failure will occur. Prognostics promises to significantly reduce expensive downtime, spares inventory, maintenance labour costs and hazardous conditions. However, prognostic is a relatively new research area and has yet to receive its prominence compared to the other areas of CBM.

Related reviews on prognostics have been reported in the literature [20]. *Pusey et al.* provided a broad overview of the development in diagnostics and prognostics technologies applicable to high-performance turbo-machines up until year 1999 [36]. *Jardine et al.* provided an overview and a catalogue of publications on data acquisition, data processing, diagnostics and prognostics of various machines up to year 2005 [23]. *Vachtsevanos et al.* defined and described intelligent fault diagnostics and prognostics approaches for engineering systems through examples [46]. *Ma et al.* discussed the need for a new paradigm shift in condition monitoring (CM) research for engineering asset management [29].

Prognostic is nowadays considered as a key feature in maintenance strategies as the estimation remaining useful life (RUL) of a system allows avoiding inopportune spending. A central problem can be pointed out: the accuracy of a prognostic system is related to its ability to approximate and predict the degradation of the equipment: starting from a "current situation", a prognostic tool must be able to forecast the "future possible situations". From the research point of view, many developments exist to support these prognostic or forecasting activities ([7], [11], [23], [46]). However, choosing an efficient technique depends on classical constraints that limit the applicability of the tools: available data-knowledge-experiences, dynamic and complexity of the system, implementation requirements (precision, computation time, etc.), available monitoring devices... Also, it can be difficult to provide effective models of dynamic systems including the inherent uncertainty of prognostic. That said, developments of this paper are founded on the following two complementary assumptions. 1) On one hand, real systems increase in complexity and their behaviour is often non-linear, which makes harder a modelling step, even impossible. 2) On the other hand, in many cases, it is not too costly to equip dynamic systems with sensors, which allows gathering real data online. According to all this, artificial neural networks (ANN) appear to be very promising prognostic tools: they learn from examples and attempt to capture the subtle relationship among data. They are computationally effective techniques and are thereby well suited for practical problems, where it is easier to gather data than to formalize the behaviour of the system being studied. Actual developments confirm the interest of using ANNs in forecasting applications ([51], [52], [1], [35], [61], [19]).

In this context, the purpose of the work is to propose an ANN as a predictive tool for prognostic purpose and to improve its prediction accuracy. More precisely, the approach combines a Recurrent Radial Basis Function network (RRBF) [54] and a proportional integral derivative controller (PID) [55]. The PID controller attempts to correct the error between the real process and the neural network predictions.

The paper is divided into four main parts. First, the concept of "prognostic" is clarified and replaced within maintenance strategies, and the relationship between prognostic and prediction is also explained; the efficiency of a prognostic system is highly dependent on its ability to perform "good" predictions. In this way, the prognostic metrics are presented and developed. Then, the use of artificial neural networks for prognostic is justified and the ways of building such models are discussed. Consequently, the combination of the RRBF and the PID controller is proposed for prognostic. In the fifth and the sixth section, data benchmark, simulation conditions, results and discussions are presented.

2. Prognostic framework and prediction

2.1. *From maintenance to prognostic*

Maintenance activity combines different methods, tools and techniques to reduce costs while increasing reliability, availability and security of equipments. Thus, one usually speaks about fault detection, failures diagnosis and response development (choice and scheduling of preventive/corrective actions). Briefly, these steps correspond to the need, firstly, of "perceiving" phenomena, secondly, of "understanding" them, and finally, of "acting" consequently. However, rather than understanding a phenomenon which has just appeared like a failure (a posteriori comprehension),

it is convenient to "anticipate" its manifestation in order to take adequate actions as soon as possible. This is what can be defined as the "prognostic process", the object of this paper.

Industrials show a growing interest in prognostic which becomes a major research framework; see recent papers dedicated to condition-based maintenance [9], [23]. The relative positioning of detection, diagnosis, prognostic and decision / scheduling in the maintenance framework is schematized in Fig. 1.a. From the phenomenological point of view, the complementarities of detection, diagnosis and prognostic can be explained as follows (see Fig. 1.b): 1) detection aims at identifying the functioning mode of the system, i.e., its current state, 2) assuming that a failure occurred, diagnosis enables to isolate and identify the component that has ceased to operate (past propagation: from effects to causes), 3) prognostic deals with the prediction of the future(s) state(s) of the system (future propagation: from causes to effects).

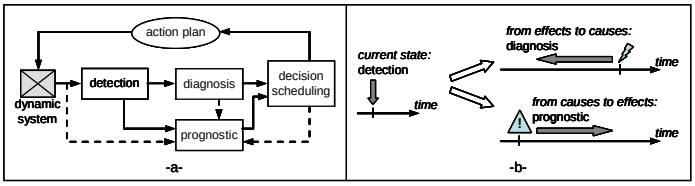

Fig. 1. *Prognostic within maintenance strategies*

2.2. From prognostic to prediction

Although there are some divergences in literature, prognostic can be defined as proposed by the International Organization for Standardization: "prognostic is the estimation of time to failure and risk for one or more existing and future failure modes" (ISO 13381-1, 2004) [21]. Prognostic is also a process whose objective is to predict the remaining useful life (RUL) before a failure occurs given the current machine condition and past operation profile [23]. Two salient aspects of prognostic appear [13]: (1) prognostic is mostly assimilated to a prediction process (a future situation must be caught), (2) prognostic is based on the failure notion, which implies that it is associated with a limit of acceptability (the predicted situation must be assessed with regard to a referential).

A central problem can be pointed out: the accuracy of a prognostic system is related to its ability to approximate and predict the degradation of an equipment; the prediction phase is a critical one. A look at prognostic metrics enables to point it out.

2.3. Prognostic metrics

There is no general agreement as to an appropriate and acceptable set of metrics that can be employed in prognostic applications, and researchers and maintenance practitioners are still working on this [46], [39], [40], [26]. Various measures emerge however from literature and are presented hereafter. As for any industrial task, prognostic can be evaluated at least in two ways:

– the main objective of prognostic is to provide the efficient information that enables the underlying decision process, i.e., the choice of maintenance actions. Thus, a first set of metrics are those that quantify the risks incurred by the monitored system. This kind of metrics can be called the prognostic measures,

– assuming that prognostic is in essence an uncertain process, it is useful to be able to judge from its "quality" in order to imagine more suitable actions. In this way, prognostic system performance measures can be constructed.

2.3.1. Prognostic measures

As mentioned earlier, the main prognostic measure pursued is the predicted time to failure (TTF), also called the remaining useful life (RUL). In addition, a confidence measure can be built to indicate the degree of certitude of the future predicted failure time. By extension, and considering that practitioners can be interested in assessing the system with regard to any performance limit, RUL and confidence can be generalized: in Fig. 2.a, TTxx refers to the remaining time to overpass the performance limit Perf/xx and Conf/xxT is the confidence with which can be taken the asset TTxx > T.

2.3.2. Prognostic system performance measures

The Timeliness of the predicted time to failure (TTF) is the relative position of the probability density function (pdf) of the prediction model along the time axis with respect to the occurrence of the failure event. This measure evolves as more data are available and reveals the expected time to perform preventive actions [46] (see Fig. 2.b). According to [17], one needs to define two different boundaries

for the maximum acceptable late and early predictions.

The Precision reveals how close predictions are grouped or clustered together and is a measure of the narrowness of the interval in which the remaining life falls. Precision follows from the variance of the predicted results for many experiments. Complementarity of accuracy and precision is illustrated in Fig. 2.c.

The Repeatability measures the ability to have the same performances if the experiments are repeated several times. This gives the robustness of the model with regard to the setup step (training process for the neural model).

The Accuracy measures the closeness of a future predicted value to a real one. It has an exponential form and is as higher as the error between the predicted value of TTF and the real one is smaller. The accuracy depends on the quality of the timeliness, precision and repeatability performances of the prediction model. This accuracy is higher: if the probability density function is closer to the real data (good timeliness performances), if the prediction values are grouped together (good precision performances) and if the model is not sensitive to the parameters setup (good repeatability).

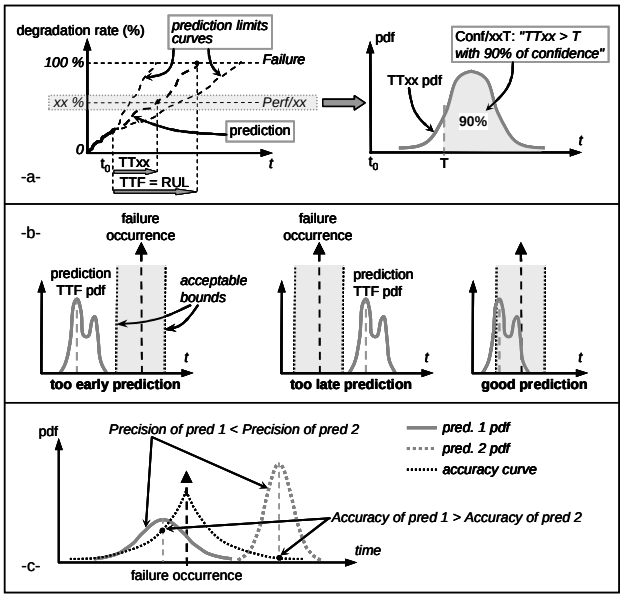

Fig. 2. *Some prognostic metrics*

2.3.3. Prognostic neural network system performance measures

Learning is an important capability of neural networks. Learning rules are algorithms for finding suitable weights W and other network parameters. Learning of a neural network can be viewed as a nonlinear optimization problem in which the goal is to find a set of network parameters minimizing the cost function for given examples. Putting it in another way, learning is an optimization process that produces an output that is as close as possible to the desired output by adjusting network parameters. For the same learning set, the neural network structure or parameters can be different at each training run. This results from the random initialization of certain parameters of the training process.

The quality of the prediction can be completely different at each running of the training algorithm. To test the prediction performances of a neural prediction model according to the prognostic metrics cited on the previous section, several running of the "training/test" process have to be done. Suppose that M represents the number of all the "training/test" running. For every running i of the training algorithm, a new value of the mean prediction error E(i) and the standard deviation std(i) are obtained for the n data of the test set as follows :

$$E(i) = \frac{1}{n}\sum_{j=1}^{n}\left(\xi^i(j) - \zeta(j)\right) \quad (1)$$

$$std(i) = \sqrt{\frac{1}{n}\sum_{j=1}^{n}\left(E(i) - \zeta(j)\right)^2} \quad (2)$$

where

$\xi^i(j)$ is the jth output obtained by the ith neural model

$\zeta(j)$ is the jth system output

The measures of the prognostic neural system performance are then processed on the variations of E(i) and std(i). The different training and test running steps are presented as follows:

for i=1 to M

- *train the NN on the training data set*
- *test the NN on the test data set*
- *calculate the mean prediction error E(i) produced by the ith neural model on the test set*
- *calculate the standard deviation std(i) produced by the ith neural model on the test set*

Next i

The prognostic metrics for a neural model are then processed as follows:

- *The timeliness is given by the global mean of all the M values of E(i):*

$$\text{Timeliness} = \overline{E} = \frac{1}{M}\sum_{i=1}^{M} E(i) \quad (3)$$

where M is the number of the training/test running, E(i) is the mean error of every running test i of the ith neural model obtained at the training step i.

For a small value of the timeliness, the probability to have a prediction close to the real value can be significant. On the contrary, if the timeliness value is important, the probability to have a wrong prediction can be significant.

- *The Precision is given by the global mean of all the M values of std(i):*

$$\text{Precision} = \overline{std} = \frac{1}{M}\sum_{i=1}^{M} std(i) \quad (4)$$

where std(i) is the standard deviation of each running test i of the ith neural model obtained by the training step i.

For a small value of the precision, the probability to have predictions grouped together can be significant. On the contrary, if the precision value is important, the probability to have a big dispersion of the predictions can be significant.

- *The Repeatability is given by the standard deviation of both E(i) and std(i). A simple way to calculate the repeatability parameter is :*

$$\text{Repeatability} = \frac{\sigma(std) + \sigma(E)}{2} \quad (5)$$

where $\sigma(std)$ and $\sigma(E)$ represent the standard deviation of the M values of respectively the std(i) and E(i) values :

$$\sigma(std) = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(\overline{std} - std(i)\right)^2} \quad (6)$$

$$\sigma(E) = \sqrt{\frac{1}{M}\sum_{i=1}^{M}\left(\overline{E} - E(i)\right)^2} \quad (7)$$

Repeatability reveals how close the different values of the E(i) and the std(i) are grouped or clustered together. This parameter reveals the dispersion of E(i) and std(i) values. For small values of $\sigma(std)$ and $\sigma(E)$, it means that at each training/running time i, the neural model gives the same performances on the test set. The repeatability parameter reveals the random initialization influence of some learning parameters. The training process is completely repeatable for small values of the repeatability parameter. The structure of the neural model is always the same at each running of the training process.

- *The accuracy is calculated from the third previous parameters. A simple way to calculate the accuracy is :*

$$\text{Accuracy} = \frac{1}{\text{Repeatability} + \text{Timeliness} + \text{Precision}} \quad (8)$$

If a neural model has a good Timeliness, Precision and is completely Repeatable, the prediction given by this neural model is very close to the real data. The prediction confidence is then very high. A big value of the accuracy parameter gives a great confidence of the prediction.

Fig. 3 illustrates the three prediction metrics: the timeliness, the precision and the repeatability described by a radar graph. This Fig. reveals the quality of the prediction made by the neural network according to the value of the three metrics. Fig. 4 gives the relation between the accuracy and the three metrics. One can see that the accuracy is very high for small values of the: timeliness, precision and repeatability. On the contrary, the accuracy is very low if, at least, one of the three metrics has a great value.

2.4. Perform good predictions: a critical issue

All prognostic metrics derive from the notion of "prediction": prognostic measures are themselves specific prediction measures and prognostic system performance measures can be seen as a way to asses the performances of the prediction in terms that can be interpreted by practitioners. As a synthesis, one should pay a particular attention to this prediction issue when choosing and adapting a prognostic tool. This aspect is developed in the next sections.

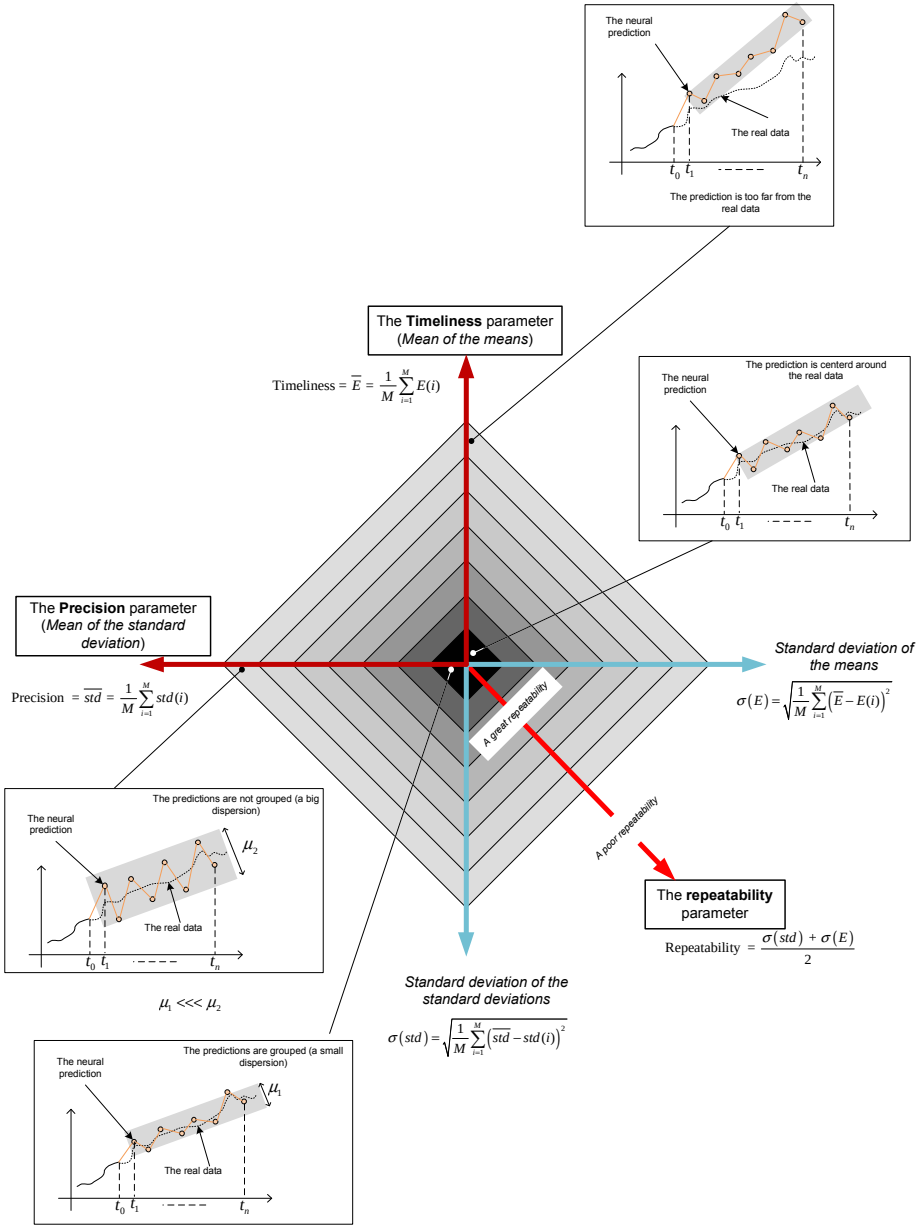

Fig. 3. *Neural Network system performance measures*

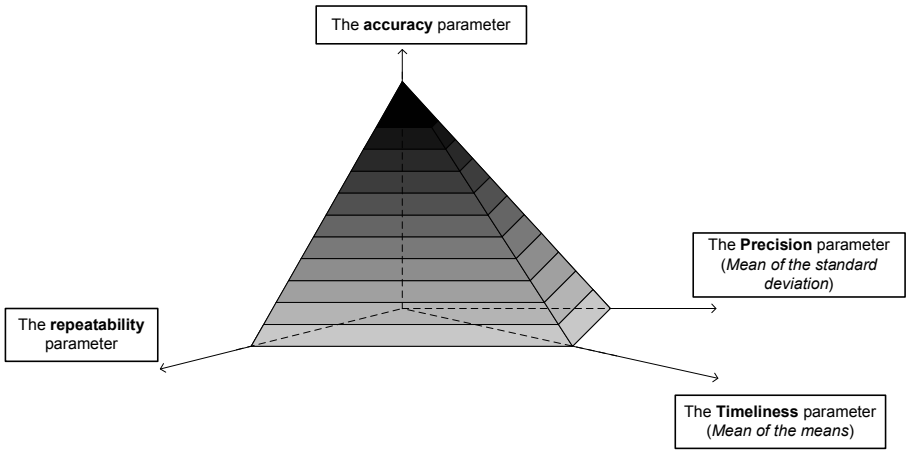

Fig. 4. *The relation between accuracy and the three other metrics.*

3. The RRBF as a prediction tool for prognostic purpose

3.1. Prediction / forecasting approaches overview

According to some authors, the methods presented in this section are sometimes labelled as "prognostic techniques". However, most of them refer to what, in this paper, is called "prediction / forecasting". That said, the aim of this part is not to dress an exhaustive overview of prediction techniques but to explain the orientations of works that are being performed.

Various prognostic approaches have been developed ranging in fidelity from simple historical failure rate models to high-fidelity physics-based models, [7]. These methods can be associated with one of the following two approaches, namely model-based (or Physics-based) and data-driven [20].

Model-based methods assume that an accurate mathematical model for the analyzed system / phenomenon can be constructed. The main advantage of these approaches is their ability to incorporate physical understanding of monitored system. Moreover, if the understanding of the system / phenomenon improves, the model can be adapted to increase its accuracy and to address subtle performance problems. But this closed relation with a mathematical model may also be a strong weakness: it can be difficult, even impossible, to catch the system's behavior.

For most industry applications, physics-based models might not be the most practical solution since the fault type in question is often unique from component to component and is hard to be identified without interrupting operation. However, physics-based models may be the most suitable approach for cost-justified applications in which accuracy outweighs most other factors and physics models remain consistent across systems, such as in air vehicles [37]. They also generally require less data than data-driven models.

Data-driven approaches attempt to derive models directly from routinely collected condition monitoring (CM) data instead of building models based on comprehensive system physics and human expertise. They are built based on historical records and produce prediction outputs directly in terms of CM data. These approaches use real data (like on-line gathered with sensors or operator measures) to approximate and track features revealing the degradation of components and to forecast the global behavior of a system. Indeed, in many applications, measured input/output data is the major source for a deeper understanding of the system degradation. The strength of data-driven techniques is their ability to capture subtle relationships among the data even if the underlying relationships are unknown or hard to describe (by a learning process).

Data-driven approaches can be divided into two categories: statistical techniques (also called conventional data-driven methods) and artificial intelligence techniques (neural networks, fuzzy systems, decision trees, etc.). The conventional data-driven methods include simple projection models, such as exponential smoothing [3], linear autoregressive [25], quadratic discriminators and partial least squares models. One major advantage of these techniques is the simplicity of their calculations, which can be carried out on a programmable calculator. However, most of these trend forecasting techniques assume that there is some underlying stability in the monitored system. They also rely on past patterns of degradation to project future degradation. This reliance could lead to inaccurate forecasts in times of change. Most of these models follow the changing pattern with a time lag of at least one observation. *Cempel et al.* introduced the Tribo-vibroacoustical (TVA) model, which can estimate the time to failure of a machine as well as forecasting the vibration amplitude or condition [8]. The model was compared with a constant trend parabolic model, an exponential trend model and an adaptive trending model in predicting a rolling bearing's peak vibration acceleration. It was reported that none of the forecasting techniques was able to predict the sudden change in the life curve.

3.2. Neural Networks – a fitted forecasting technique

Real systems are complex and their behavior is often non linear, non stationary. These considerations make harder a modeling step, even impossible. Yet, a prediction computational tool must deal with it. Moreover, monitoring systems have evolved and it is now quite easy to online gathered data. According to all this, data-driven approaches have been increasingly applied to prediction problems in general and to machine prognostic in particular. More precisely, research

works emphasize on the interest of using artificial neural networks for prediction ([2], [10], [30], [51]).

Artificial neural networks (ANNs) are currently the most commonly found data-driven techniques in the prognostics literature [20]. ANNs are a special case of adaptive networks that have been extensively explored in literature because of the following aspects. An ANN consists of a layer of input nodes, one or more layers of hidden nodes, one layer of output nodes and connecting weights. ANNs can perform nonlinear modelling without a priori knowledge: they are able to learn complex relationships among "inputs and outputs". The network learns the unknown function by adjusting its weights with repetitive observations of inputs and outputs. Numerous studies across various disciplines have demonstrated the merits of ANNs, including the abilities to (a) perform faster than system identification techniques in multivariate prognosis [27]; (b) perform at least as good as the best traditional statistical methods, without requiring untenable distributional assumptions [45], [24]; and (c) capture complex phenomenon without a priori knowledge [5]. A widely known limitation of ANNs is the lack of transparency, or rather the lack of documentation on how decisions are reached in a trained network. However, it was argued in [5] that increase in model complexity reduces the transparency of both traditional statistical models and ANN models. It is just that ANNs are more capable in modelling complex phenomenon and consequently need a more complex structure to represent the phenomenon. Rules can actually be extracted from trained ANNs to explain how decisions are reached, see [41] and [44].

The temporal NNs have two typical connection architectures depending on the type of time representation [14]: in feedforward networks (like the multi layers perceptron MLP or the radial basis function network RBF) time is represented as an external mechanism, whereas recurrent networks (like the Elman architecture or the recurrent radial basis function network RRBF) are able to treat time dimension without any external mechanism. Recurrent neural network can be described by a NARX model (Nonlinear AutoRegressive with eXogenous inputs). Both have been employed in system behavior forecasting. In order to explicitly take into account time in forecasting tools, backward networks architectures have been also developed. These NARX recurrent neural networks are fundamentally different from feedforward architectures in the sense that they not only operate on an input space but also on an internal state space. Recurrent ANNs have been compared with some of the well known methods for the prediction of non-linear time series. Results indicated that RNNs have a better forecasting performance than the classical methods and are even better than the feedforward type ANNs.

One of the first successful application of ANNs in forecasting is reported by Lapedes and Farber who designed a feedforward ANN that can accurately mimic a chaotic series [62], [52]. In general, feedforward ANNs (MLP, RBF) trained with the backpropagation algorithm have been found to perform better than classical autoregressive models for the trend prediction of non linear time series [51].

The most simple ANN-based machinery prognostics approach was time series prediction models [56], [57], [58], [60], [61]. *Tse et al.* and *Yam et al.* used recurrent neural networks (RNNs) to trend CM indices and forecast successive index value at the next time step [45], [51]. *Wang et al.* developed a recurrent wavelet neural network (RWNN) to predict rolling element bearing crack propagation [47]. The network performed satisfactorily in trending an artificially seeded and manually enlarged crack, provided that sufficient data points were used and network retraining was carried out after each time step. In [49], *Wang et al.* used a Neuro-Fuzzy (NF) network to predict spur gear condition value one step ahead. The fuzzy interference structure is determined by experts, whereas the fuzzy membership functions are trained by the neural network [59]. The NF system performed much better than RNN when there was sufficient training data. However, it could not predict well when the train-set was small or when there were fast dynamic fluctuations, such as during the chipping of gear tooth surface material or just prior to gear failure. An adaptive training technique was later proposed by Wang et al. to improve the NF model [48]. The addition of feedback links to the NF model was able to increase the forecasting accuracy. However, further work is needed to extend the prediction horizon from single step to multiple steps ahead. Feed forward neural network (FFNN) has also been used to perform single-step-ahead prediction of rolling element bearing condition by [42]. Multiple-step-ahead predictions were also performed simply by feeding the predicted value back into the network input until the desired prediction horizon was reached. The authors also proposed some rules to vary the data sampling period according to the change ratio of consecutive condition index values.

3.3. RBF and Recurrent RBF networks

3.3.1. The Radial Basis Function network

In practice, Multi-layer perceptrons (MLP) have been found to perform poorly in a number of ways, slow convergence of weights and difficulty in modelling differential responses. Radial basis function (RBF) neural networks are able to surpass the MLPs as they are simpler in structure and have the ability to model any nonlinear function in a straightforward way. On one hand MLP networks are global approximators with nonlinear input-output mappings and the representation of knowledge is distributed throughout the network. On the other hand, RBF networks are local approximators with nonlinear input-output mapping. The knowledge representation in this case is localized. Thus, RBF network are able to learn faster and suffer less from interference, as compared to MLPs.

A key feature of RBF networks is that the output layer is merely a linear combination of the hidden layer signals, there being only one hidden layer. Therefore, RBF networks allow for a much simpler weight updating procedure and subsequently open up greater possibilities for stability proofs and network robustness in that the network can be described readily by a set of nonlinear equations.

The RBF network is commonly used for the purpose of modelling uncertain and nonlinear functions. Utilizing RBF networks for modelling purposes could be seen as an approximation problem in a high-dimensional space. Consider the RBF network, which can be seen as a two-layer processing structure, as shown in Fig. 5.a. The hidden layer consists of an array of computing units (i.e., $\phi_1, \phi_2, ...\phi_M$). These hidden units provide a set of basis functions of the input vectors (i.e., $x_1, x_2, ...x_d$) as they are expanded into the higher dimension hidden-unit space. The mapping from the input vectors to the outputs of the hidden units is nonlinear, whereas the mapping from the hidden units to the final output of the RBF network is linear.

The general mapping function of the RBF network can be represented by

$$y_k = \mathbf{g}[\boldsymbol{x}, \boldsymbol{w}] = \sum_{j=1}^{M} w_{kj}\phi_j(\boldsymbol{x}) \quad (9)$$

where $\mathbf{x} = [x_1, x_2, ...x_d]^T$. For the second layer of the RBF network, it functions like a summer with a set of weights (i.e., $w_{k1}, w_{k2}, ...w_{kM}$). The commonly used RBF $\phi_j(.)$ is the Gaussian represented by

$$\phi_j(\boldsymbol{x}) = \exp\left(-\frac{\left\|\mathbf{x} - \mathbf{c}_j\right\|^2}{2\boldsymbol{\sigma}_j^2}\right) \quad (10)$$

Each RBF contains a parameter vector called a centre ($\mathbf{c}_j$), and calculates a squared distance between the centre and the input vector ($\boldsymbol{x}$). The result is then divided by the width ($\boldsymbol{\sigma}_j$) and then passed through an exponential function.

Training a RBF with linear outputs is very fast and is accomplished through two stages:

– The first stage is unsupervised and accomplished by obtaining cluster centres of the training set input vectors. A popular method for that purpose is the k-means clustering,

– the second stage consists in solving a set of linear equations, the solution of which can be obtained by a matrix inversion technique such as singular value decomposition or least squares method.

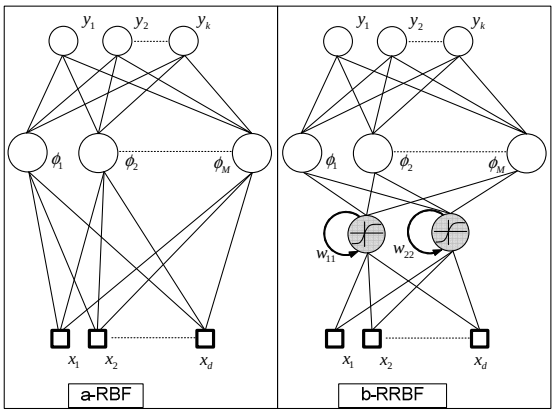

Fig. 5. *Radial Basis Function and Recurrent Radial Basis Function networks*

3.3.2. The Recurrent Radial Basis Function network

The Recurrent RBF neural network considers time as an internal representation (Fig. 5.b). The dynamic aspect is obtained by the use of an additional self-connection to the input neurons with a sigmoid activation function. The RRBF network can thus take into account a certain past of the input signal.

Every neuron of the input layer gives a summation at the instant t between its input x_i and its previous output weighted by a self-connection w_{ii}. The output of its activation function is:

$$a_i(t) = w_{ii}\xi_i(t-1) + x_i(t),\ \xi_i(t) = f\left(a_i(t)\right) \quad (11)$$

where $a_i(t)$ and $\xi_i(t)$ represent respectively the neuron activation and its output at the instant t, and f is the sigmoid activation function defined as:

$$f(x) = \left(1-\exp(-kx)\right)/\left(1+\exp(-kx)\right) \quad (12)$$

The RRBF network was described on several publications [53], [54], [38], [18], [12], [33]. A complete study of the looped neuron can be found on [15], [4]. The reader can find in a detailed mathematical demonstration of the dynamic behavior of the looped neuron. To have the longest memory, the self connection weight w_{ii} and the parameter k of the sigmoid function must respect this relation: $kw_{ii} = 2$.

4. Combining the RRBF and the PID controller for prognostic purpose

4.1. Principle

RRBF appears to be a good candidate for prediction in prognostic applications. Nevertheless, one can improve its prediction accuracy by combining the RRBF output with a Proportional-Integral-Derivative controller (PID) [55]. The use of the PID as a controller to perform the neural prediction has the same objectives as all other conventional feedback control applications. Fig. 6 shows the prediction structure where the PID controller is used to minimize the neural prediction error $\varepsilon(t)$. Let call this prediction structure the RRBF_{PID} structure.

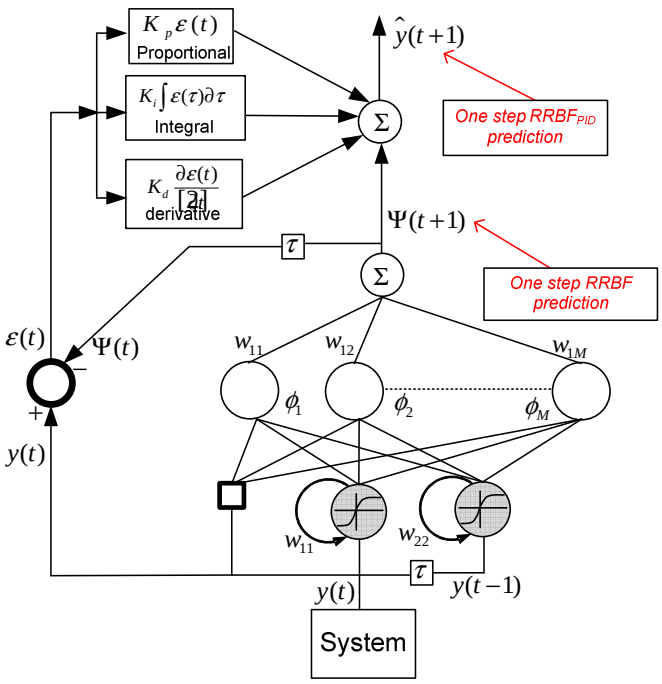

Fig. 6. *The RRBF_{PID} structure for time series prediction*

At any time *t*, the prediction error of the RRBF neural network can be expressed as:

$$\varepsilon(t) = y(t) - \Psi(t) \quad (13)$$

where $y(t)$ represents the real system output, and $\Psi(t)$ the RRBF network predicted output. A one step neural prediction is:

$$\Psi(t+1) = \mathbf{g}[\boldsymbol{y}, \boldsymbol{w}] = \sum_{j=1}^{M} w_{kj}\phi_j(\boldsymbol{y}) \quad (14)$$

The aim of the PID is to apply a corrective action on this error. The output of the final one-step ahead prediction $\hat{y}(t+1)$ obtained by the RRBF_{PID} structure is then defined as:

$$\hat{y}(t+1) = \Psi(t+1) + K_p\varepsilon(t) + K_i\int_0^t \varepsilon(\tau)\partial\tau + K_d\frac{\partial\varepsilon(t)}{\partial t} \quad (15)$$

$$\begin{aligned}\hat{y}(t+1) = \Psi(t+1) + K_p y(t) + K_i\int_0^t y(\tau)\partial\tau + K_d\frac{\partial y(t)}{\partial t} \\ - K_p\Psi(t) - K_i\int_0^t \Psi(\tau)\partial\tau - K_d\frac{\partial\Psi(t)}{\partial t}\end{aligned} \quad (16)$$

where, K_p, K_i, K_d are the proportional, integral and derivate gains of the PID. In a general way, the

prediction output $\hat{y}(t+1)$ obtained by the $RRBF_{PID}$ can be defined as follows:

$$\hat{y}(t+1) = \mathbf{F}[\boldsymbol{y}, \boldsymbol{\Psi}] \quad (17)$$

and

$$\Psi(t+1) = \mathbf{g}[\boldsymbol{y}, \boldsymbol{w}] \quad (18)$$

F[.] is the whole nonlinear function for the prediction purpose. Note that if the integrator parameter is equal to zero ($K_i = 0$), the function F is similar to the well-known NAR models (Nonlinear Auto Regressive model). In a general way, a great equivalence exists between the NARX models and the recurrent neural network (RNN) [43]. The next section presents briefly the different structures of the NARMAX models, followed by a comparative test on a NARX time series prediction.

4.2. NARMAX and NARX model for optimal predictors

The statistical approach for forecasting involves the construction of stochastic models to predict the value of an observation y_t using previous observations. A very general class of such models used for forecasting purpose is the Nonlinear AutoRegressive Moving Average with eXogenous inputs (NARMAX models) [16], [43], [28], [35] given by:

$$y(t) = \mathbf{F}[y(t-1)...y(t-n_y), e(t-1)...e(t-n_e), x(t-1)...x(t-n_x)] + e(t) \quad (19)$$

where y, e and x are output, noise and external input of the system model respectively. n_y, n_e and n_x are the maximum lags in the output, noise and input, respectively, and F is an unknown smooth function. It is assumed that *e(t)* has zero mean, is independent and identically distributed, is independent of past y and x, and has a finite variance σ^2.

The NARMAX models are nonlinear generalization of the well-known ARX models, which constitute a standard tool in linear black-box model identification. Several special cases of the general NARMAX (n_y, n_e, n_x) model which are frequently used are summarized here :

- NAR(n_y) model (Nonlinear AutoRegressive) :

$$y(t) = \mathbf{F}[y(t-1),,,,y(t-n_y)] + e(t) \quad (20)$$

- NARMA(n_y , n_e) model (Nonlinear AutoRegressive Moving Average):

$$y(t) = \mathbf{F}[y(t-1)...y(t-n_y), e(t-1)...e(t-n_e)] + e(t) \quad (21)$$

- NARX(n_y , n_x) model (Nonlinear AutoRegressive with eXogenous inputs):

$$y(t) = \mathbf{F}[y(t-1)...y(t-n_y), x(t-1)...x(t-n_x)] + e(t) \quad (22)$$

The NARX models can represent a wide variety of nonlinear dynamic behaviors and have been extensively used in various applications [35], [1], [34], [31], [50].

4.3. Example on a NARX time series model

To demonstrate the validity of our proposed predictor, the $RRBF_{PID}$ is trained and tested on a nonlinear autoregressive process with exogenous input (NARX model) generated as follows [16]:

$$y(t) = \frac{y(t-1)y(t-2)[y(t-1)+2.5]}{1+y^2(t-1)+y^2(t-2)} + x(t-1) \quad (23)$$

where the input has the form $x(t) = \sin(2\pi t/25)$. The topology of the RRBF network is estimated using the first 20 observations of the process. The predictions $\hat{y}(t+1)$ of $y(t)$ are then tested on the following observations. Fig. 7 presents the results obtained on the 100 last observations (from t=200 to t=300). To test the influence of the PID corrective action, three prediction models are compared. The first one is the RRBF model. On the second model, only the PID action is tested by considering the RRBF output equal to zero ($\Psi(t) = 0$). On the third one, the whole model is tested (RRBF + PID).

- RRBF network prediction without PID action:

$$\hat{y}(t+1) = \Psi(t+1) \quad (24)$$

- PID output without the RRBF network:

$$\hat{y}(t+1) = K_p y(t) + K_i \int_0^t y(\tau)\partial\tau + K_d \frac{\partial y(t)}{\partial t} \quad (25)$$

- RRBF with PID correction:

$$\hat{y}(t+1) = \Psi(t+1) + K_p y(t) + K_i \int_0^t y(\tau)\partial\tau + K_d \frac{\partial y(t)}{\partial t} - K_p \Psi(t) - K_i \int_0^t \Psi(\tau)\partial\tau - K_d \frac{\partial \Psi(t)}{\partial t} \quad (26)$$

The RRBF is trained by the k-means algorithm to find the Gaussian parameters and the linear regression to find the output weights. 9 Gaussians nodes were used for the RRBF model and the best PID parameters were found by testing all the possibilities of (K_p, K_i, K_d) from 0 to 1 by increments of 0,1. The best PID parameters are then: $K_p = 0.9, K_i = 0, K_d = 0.2$. With this parameter values, the structure of the PID is then similar to an Auto-Regressive model (AR). The $RRBF_{PID}$ seems to be better than the classic RRBF structure and also better than the AR model (Fig. 7). The prediction error obtained by the RRBF network is judiciously corrected by the PID structure (AR model). The next sections give more results and discussions of the $RRBF_{PID}$ obtained on three data Benchmarks.

5. Data benchmarks, simulation conditions and crossed validation

Three experimental data sets have been used to test the prediction performances of the $RRBF_{PID}$ structure with regards to the classical RRBF network. In the three cases, the aim of the predictions is to approximate a phenomenon by learning data gathered from the system.

For all the benchmarks, three sets were used: a training set, a validation set and a test set (Fig. 8). The first set was used for the training parameters of the neural network (the k centres of the Gaussian nodes and the output weights). 50 samples have been used for training set and 50 samples for validation set. All data have been normalized by range [-1,+1]. Predictions steps were made from "t+1" to "t+10" by increments of 1 (in order to measure the stability of results in time). Predictions have been performed with two past inputs at a step time "t" and "t-1" (Fig. 6).

In order to find the best RRBF structure with regard to the k centers, several models of neural networks have been created by varying the number of basis functions from 2 to 50 nodes. The k-means training algorithm has been used to find the best Gaussian centers for each k-structure. The basis width parameter of the Gaussian nodes was fixed to 1. The neural network trained with the first set is validated in the second one. Validation uses data different from the training set, thus the validation set is independent from the estimated model. This helps to select the best one among the different model parameters. To avoid overfitting or underfitting, the optimal model parameters should be selected so as to have the best performance measure associated with the validation set. Since this dataset is independent from the estimated model, the generalization error obtained is a fair estimated. The best model which gives the best mean prediction error in the validation set is then selected and tested in the third set (the test set). The model with the best generalization performance is then selected with this crossvalidation technique.

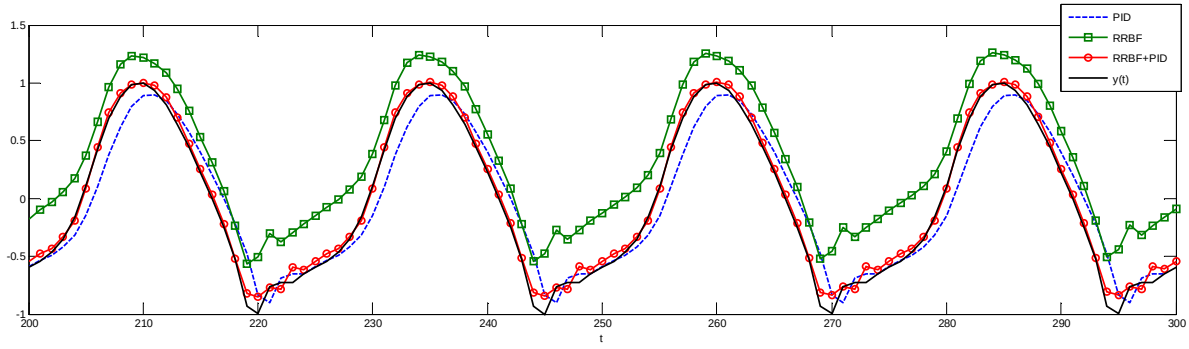

Fig. 7. *Prediction results obtained on a NARMAX example*

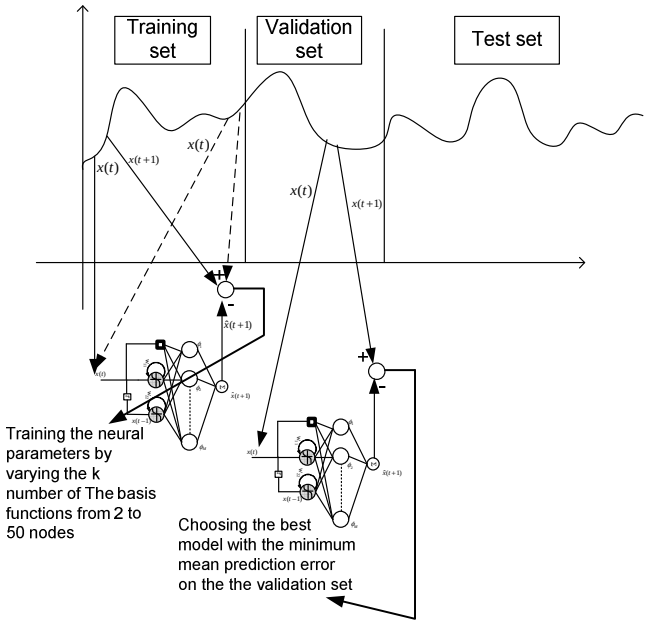

Fig. 8. *Training process and crossed validation*

Once the best RRBF model found, the second step is to find the PID parameters (Kp, Ki, Kd). At this stage, theses parameters varied from 0 to a maximum of 1 by increments of 0,1. While it is appreciated that values used for this scaling variable are extremes, these values have been chosen to encapsulate all possibilities. Two sets of PID parameters have been selected: the parameters that give the best performance (mean prediction error) on the training set ($RRBF_{PID1}$) and the parameters that give the best performance on the validation set ($RRBF_{PID2}$).

5.1. Mackey Glass

The first data set is the chaotic Mackey-Glass time series data [22]. This time series is a benchmark problem extensively used: it's a non periodic and non convergent time series. Considering our final applicative objective (the prognostic of failures) able to carry out predictions on such a signal is of good omen: what makes difficult a modeling phase are real complex systems which generally have a nonstationary and non-linear behavior. The time series is governed by the following formula:

$$\frac{dy}{dt} = b\frac{y(t-d)}{1+y(t-d)^c} - ay(t) \quad (27)$$

where a, b, c and d are real constants. The values of most commonly parameters used in literature are $a = 0.1,\ b = 0.2,\ c = 10,$ and $d = 17$ where at the initial time $t_0 = 0,\ x(t_0) = 1.2$. The differential equation is approximated by the 4^{th} order Runge-Kutta algorithm with time step equal to 1. Tests on this time series aim at predicting future values $\hat{y}(t+n)$ by using past values as follows:

$$\hat{y}(t+n) = \mathbf{F}[y(t), y(t-1)] \quad (28)$$

5.2. Lorenz

The Lorenz time series is a long synthetic chaotic time series obtained from http://www.physics.emory.edu/~weeks/research/tseries1.html. The time series is governed by the following differential equations:

$$\frac{dy}{dt} = \sigma(x_t - y_t),\ \frac{dx}{dt} = ry_t - x_t - y_t z_t,\ \frac{dz}{dt} = y_t x_t - bz_t \quad (29)$$

we take parameter setting $\sigma = 10, r = 28, b = 8/3$ and use the 4000 y-ordinate points derived from a Rung-Kutta integrator with time step 0.01. Tests on this time series aim at predicting future values $\hat{y}(t+n)$ by using past values as follows:

$$\hat{y}(t+n) = \mathbf{F}[y(t), y(t-1)] \quad (30)$$

5.3. Box–Jenkins

The third benchmark is the Box–Jenkins furnace data [6]. This data set was recorded from a combustion process of a methane-air mixture. During the process, the portion of methane was randomly changed, keeping constant gas flow rate. The data set consists of 296 pair of input-output measurements ({y(t), u(t)}, from t=1 to t=296). The input u(t) is the methane gas flow into the furnace and the output y(t) is the CO2 concentration in the outlet gas. Tests on this time series aim at predicting future values y(t+n) by using {y(t), u(t)} values as inputs. The system is described by the following NARX model:

$$\hat{y}(t+n) = \mathbf{F}[y(t), y(t-1), u(t), u(t-1)] \quad (31)$$

6. Results and discussion

To measure the prognostic system performances presented in section 2.3.3, the simulation test was running 1000 times for each benchmark (i=1 to M=1000). At each simulation running i, the mean prediction error E(i) and the standard deviation std(i) are calculated.

6.1. Mackey-Glass

Fig.s 10-11-12-13 illustrate the results obtained by the three neural models (RRBF, $RRBF_{PID1}$ and $RRBF_{PID2}$) for the Mackey-Glass data test for different horizons of prediction n (n=1 to 10). The results show that the prognostic performance measures obtained by the $RRBF_{PID2}$ are better than those obtained by the two other models.

Fig. 10 reveals the great disparity of the timeliness parameter among the three models. This Fig. shows that for the prediction horizon n=1 to 9, the $RRBF_{PID2}$ has the best timeliness value. The disparity between the models decreases with the increasing of the prediction horizons.

Fig. 11 shows the precision parameter for different models for different horizons of prediction. The precision of the $RRBF_{PID1}$ and $RRBF_{PID2}$ are similar for the different values of the horizon n. The prediction obtained with the RRBF model is more dispersed than the two other models.

Fig. 12 reveals how the prediction can be repeatable. For the prediction horizon n=0 to 6, the $RRBF_{PID2}$ is the model with the best repeatability performance.

The accuracy parameter obtained from the three metrics is shown in Fig. 13 and is calculated according to the following formula:

$$\text{accuracy} = \frac{1}{\text{Repeatability} + \text{timeliness} + \text{Precision}} \quad (32)$$

The $RRBF_{PID2}$ is the model with the best accuracy performance for n=1 to 6. This accuracy becomes the same for the three models for n>6.

6.2. Lorenz data set

The results obtained with the three prediction models for the Lorenz data set are presented by Fig.s 14-15-16-17. Prognostic performances measures obtained by the $RRBF_{PID2}$ are better than those obtained by the two other models. The $RRBF_{PID2}$ model obtains the best timeliness and precision parameters for all the prediction horizons n. For the repeatability parameter, we have good performances for the $RRBF_{PID2}$ only for n=1 to n=5. The prediction accuracy is illustrated in Fig. 17. The $RRBF_{PID2}$ model obtain the best accuracy with a great disparity for n=1, decreases until n=5 and becomes the same for the three models for n>6.

6.3 Box & Jenkins data set

Fig.s 18-19-20-21 show the results obtained by the three models for the Box&Jenkens data set for different horizons of prediction n (n=1 to 10). The performances of the $RRBF_{PID2}$ are not as better as for the two previous data set. Nevertheless, the prediction accuracy of the $RRBF_{PID2}$ is better than the two other models for n=1 to n=5. For n=6 & n=7, the prediction performance of the $RRBF_{PID2}$ is very bad for the three metrics. The accuracy prediction obtained with the $RRBF_{PID2}$ model is the best for n=1.

6.4. Discussions and future works

For all the tests, the best results of $RRBF_{PID}$ are obtained with the integrator parameter equal to zero $K_i = 0$. With this value, the PID action is reduced to a PD corrector. This correction added to the neural network output is similar to an Auto-Regressive model (AR). Fig. 9 illustrates the structure of the whole prediction model obtained after the tests.

The results obtained by the $RRBF_{PID2}$ are significantly better than the two other models for the Mackey-Glass and the Lorenz data benchmark. The performances obtained by the three models on Box&Jenkins benchmark are similar (there is no significant performance difference among the three structures). The Mackey-Glass and the Lorenz data benchmarks are pseudo periodic and are represented by a NAR model. These two benchmarks are described by differential equations. The Box&Jenkins data benchmark is a real system. There is no formal relation between the input and the output and there is no pseudo periodicity on the data. The relation between the input/ouput is described by a NARX model ($\hat{y}(t+n) = \mathbf{F}[y(t), y(t-1), u(t), u(t-1)]$). The linear PID corrector cannot perform significantly the prediction obtained by the recurrent neural network.

The experimental study presented in this paper reveals the good prediction ability of the $RRBF_{PID}$ and opens some interesting perspectives for our

future works. The feedback error performs significantly the prediction performances of the neural network with regard to the prediction metrics: Timeliness, Precision and Repeatability. A good accuracy of the prediction is then obtained by the combination of the Neural Network and the AR model (Fig. 9). Further study is necessary to investigate the robustness of the proposed method by using various training data groups. Additional tests should be made to evaluate the performances of a NARX model to process the feedback prediction error made by the ANN (use a NARX model instead of the AR model in Fig. 9).

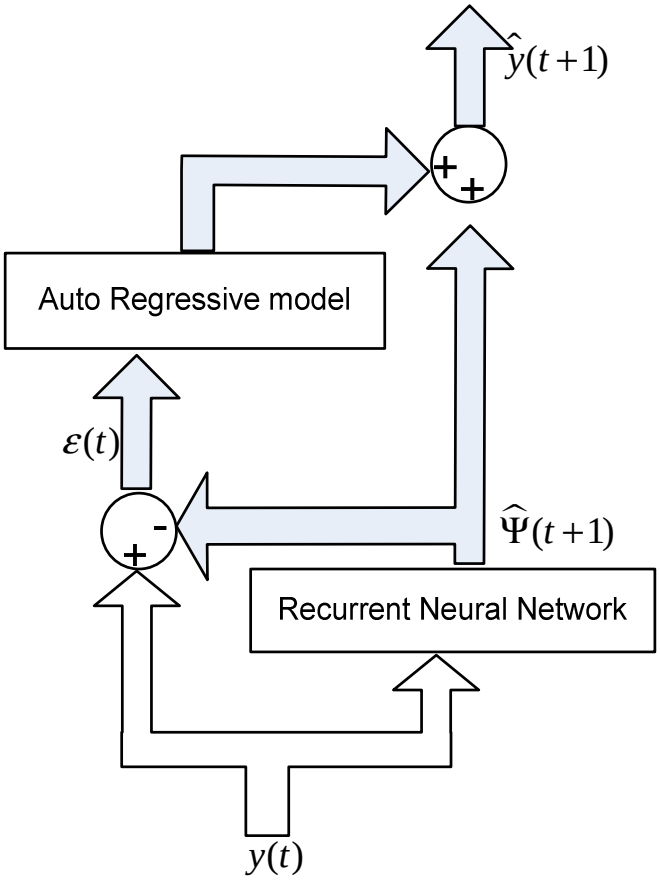

Fig. 9. *The Final prediction structure obtained after the test results.*

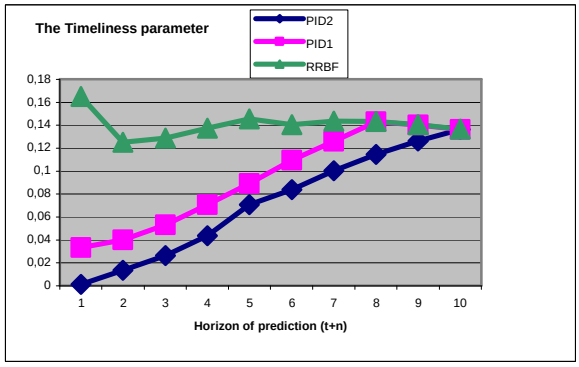

Fig. 10. *Timeliness results obtained on the Makey-Glass*

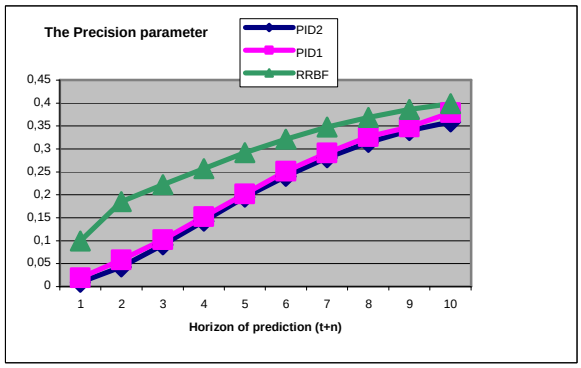

Fig. 11. *Precision results obtained on Makey-Glass*

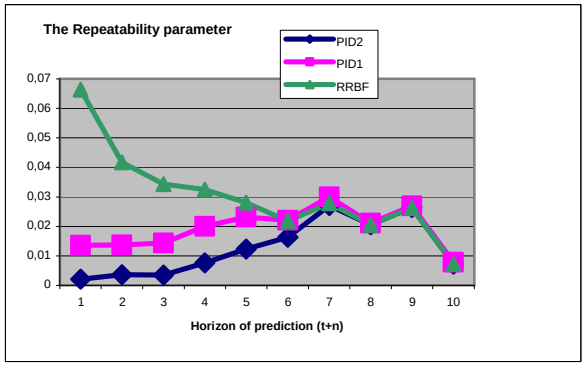

Fig. 12. *Repeatability results obtained on Makey-Glass*

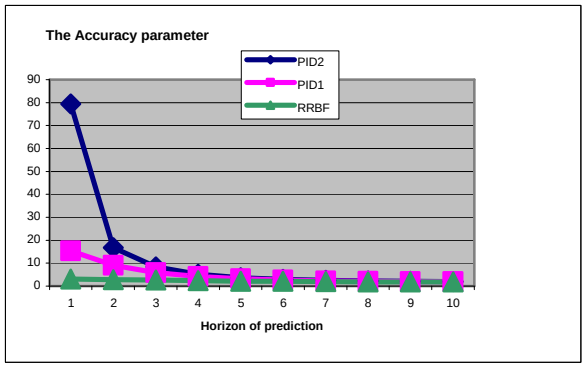

Fig. 13. *Accuracy results obtained on Makey-Glass*

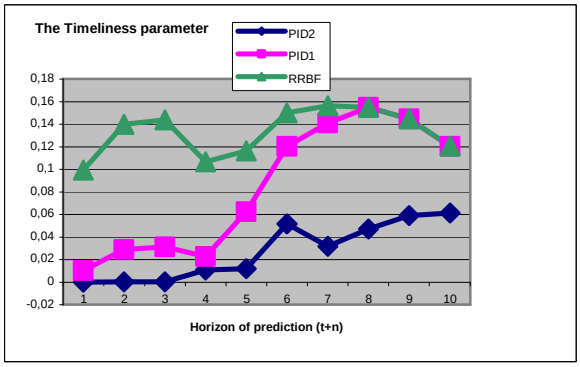

Fig. 14. *Timeliness results obtained on Lorenz data set*

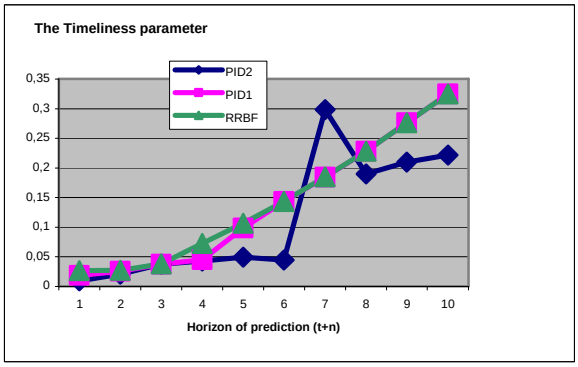

Fig. 18. *Timeliness results obtained on Box–Jenkins*

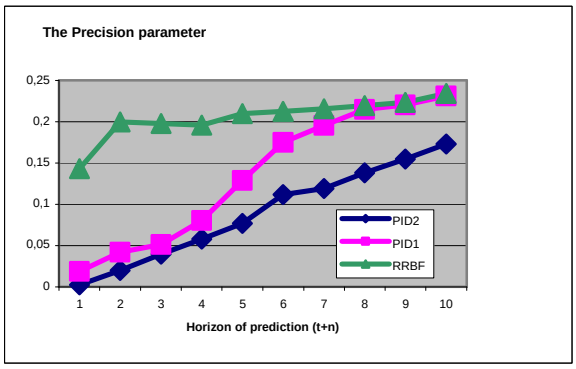

Fig. 15. *Precision results obtained on Lorenz data set*

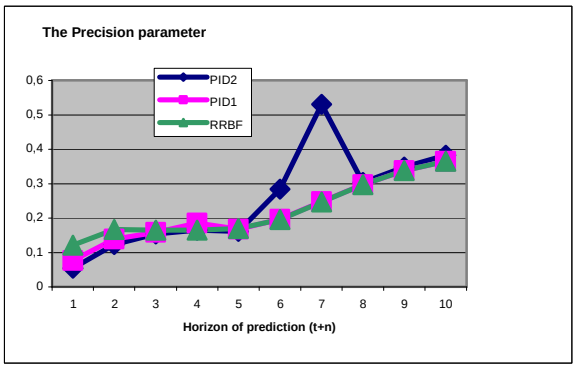

Fig. 19. *Precision results obtained on Box–Jenkins*

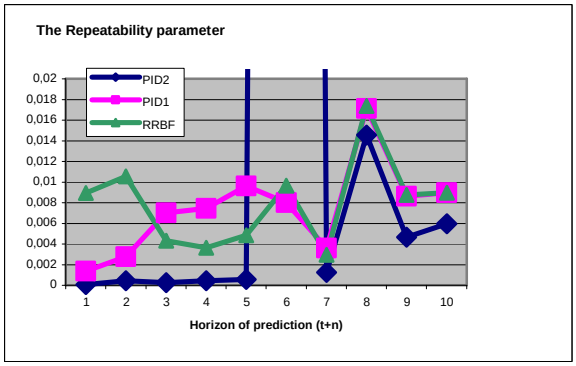

Fig. 16. *Repeatability results obtained on Lorenz data set*

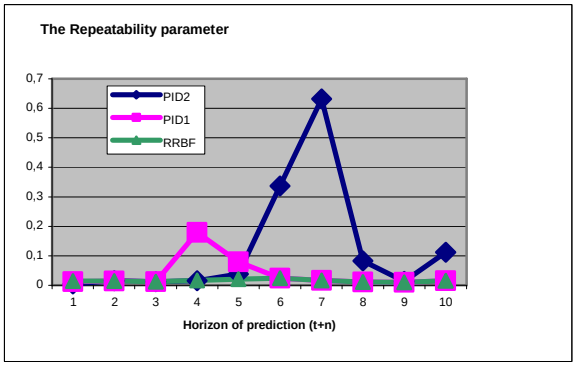

Fig. 20. *Repeatability results obtained on Box–Jenkins*

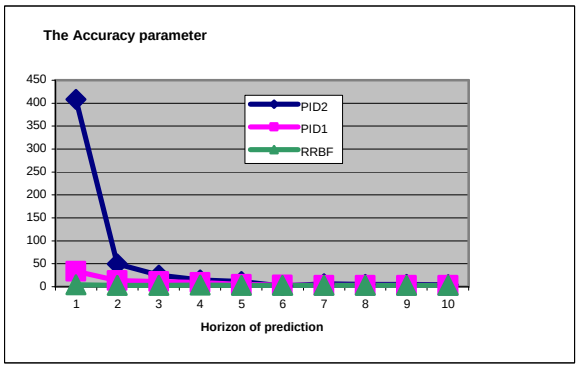

Fig. 17. *Accuracy results obtained on Lorenz data set*

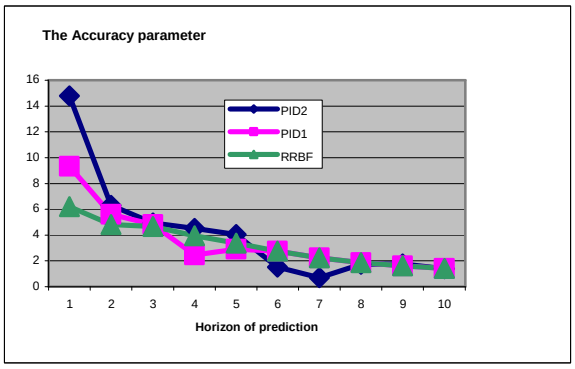

Fig. 21. *Accuracy results obtained on Box–Jenkins*

7. Conclusions

In maintenance field, prognostic is recognized as a key feature in estimating the remaining useful life of an equipment allows avoiding inopportune maintenance spending. However, it can be difficult to define and implement an adequate and efficient prognostic tool that includes the inherent uncertainty of the prognostic process. Indeed, an important task of prognostic is that of prediction. In this context, the purpose of the work reported in this paper is to point out an accurate prediction technique and to propose a way to improve its prediction performances.

The concept of "prognostic" has been positioned within the maintenance strategies in order to point out the importance of the prediction phase in prognostic. According to the global requirements that can be expected from a forecasting tool, the neural network RRBF has been presented as a candidate to support this activity. An improvement of this neural network has also been proposed by combining it with a proportional integral derivative controller (PID). The PID controller attempts to correct the error between the real process variable and the neural network predictions. Various simulations have been led with three benchmark problems. Results show that the proposed prediction structure enables the forecasting to be a more robust task without increasing complexity of treatments. The whole is of good omen for prognostic purpose.

8. References

[1] Andalib Arash and Farid Atry, « Multi-step ahead forecasts for electricity prices using NARX: A new approach, a critical analysis of one-step ahead forecasts », Energy Conversion and Management (Elsevier) Volume 50, Issue 3, March 2009, Pages 739-747.

[2] Barbounis T.G., Theocharis J.B., "A locally recurrent fuzzy neural network with application to wind speed prediction using spatial correlation", Neurocomputing, vol. 70, Issues 7-9, 2007, p. 1525-1542.

[3] Batko W., Prediction method in technical diagnostics, Doctoral Thesis, Cracov Mining Academy, 1984.

[4] Bernauer E., Les réseaux de neurones et l'aide au diagnostic : un modèle de neurones bouclés pour l'apprentissage de séquences temporelles, thèse de doctorat, LAAS/Toulouse 1996.

[5] Bostwichk D.G. and H.B. Burke, Prediction of individual patient outcome in cancer, Cancer Supplement 91 (2001), pp. 1643–1646.

[6] Box George and Jenkins Gwilym "Time series analysis: Forecasting and control", San Francisco: Holden-Day (1970).

[7] Byington C., Roemer M., Kacprzynski G., Galie T., "Prognostic enhancements to diagnostic systems for improved condition-based maintenance", IEEE Aerospace Conference, Big Sky, USA, 2002.

[8] Cempel C., Simple condition forecasting techniques in vibroacoustical diagnostics, Mechanical Systems and Signal Processing 1 (1987), pp. 75–82.

[9] Ciarapica F., Giacchetta G., "Managing the condition-based maintenance of a combined-cycle power plant: an approach using soft computing techniques", Journal of Loss Prevention in the Process Industries, vol. 19, 2006, p. 316-325.

[10] Dai Q., Chen S., "Chained DLS-ICBP neural networks with multiple steps time series prediction", Neural Processing Letters, vol. 21, 2005, p. 95-107.

[11] De Gooijer J.G., Hyndman R.J., "25 years of time series forecasting", International Journal of Forecasting, vol. 22, 2006, p. 443-473.

[12] Dodonov Evgueni, and Rodrigo Fernandes de Mello « A novel approach for distributed application scheduling based on prediction of communication events » Future Generation Computer Systems (Elsevier 2009)

[13] Dragomir O., Gouriveau R., Zerhouni N., Dragomir F., "Framework for a distributed and hybrid prognostic system", 4th IFAC Conference on Management and Control of Production and Logistics – MCPL'07, Sibiu, Roumania, 2007.

[14] Elman J.L., "Finding Structure in Time", Cognitive Science, vol. 14, 1990, p. 179-211.

[15] Frasconi P., Gori M. et Soda G., « Local Feedback Multilayered Networks », Neural Computation, Vol. 4, pp. 120-130, 1992.

[16] Gao Y., Meng Joo ER, "NARMAX time series model prediction : feedforward and recurrent fuzzy neural network approaches", Fuzzy sets and systems, 150 (2005), p. 331-350.

[17] Goebel K., Bonissone P., "Prognostic information fusion for constant load systems", Proceedings of 7th annual Conference on Fusion, vol. 2, 2005, p. 1247-1255.

[18] Gowrishankar; Ramesh, B.H.S.; Satyanarayana, P.S, "Neural network based BER prediction for 802.16e channel" 15th International Conference on Software, Telecommunications and Computer Networks, (SoftCOM 2007). 27-29 Sept. 2007 Page(s):1 - 5

[19] Graves Daniel, WitoldPedrycz « Fuzzy prediction architecture using recurrent neural networks » Neurocomputing (Elsevier) Volume 72, Issues 7-9, March 2009, Pages 1668-1678 Advances in Machine Learning and Computational Intelligence - 16th European Symposium on Artificial Neural Networks 2008, 16th European Symposium on Artificial Neural Networks 2008

[20] Heng Aiwina, Sheng Zhanga, Andy C.C. Tana and Joseph Mathewa, « Rotating machinery prognostics: State of the art, challenges and opportunities » Mechanical Systems and Signal Processing (Elsevier) Volume 23, Issue 3, April 2009, Pages 724-739

[21] ISO 13381-1, Condition monitoring and diagnostics of machines - prognostics - Part1: General guidelines, International Standard, ISO, 2004.

[22] Jang J.S.R., "ANFIS: adaptive network-based fuzzy inference system", IEEE Transaction on Systems, Man and Cybernetics 23(3), 1993, p. 24-38.

[23] Jardine A.K.S., Lin D., Banjevic D., "A review on machinery diagnostics and prognostics implementing condition-based maintenance", Mechanical Systems and Signal Processing, vol. 20, 2006, p. 1483-1510.

[24] Joshi R. and C. Reeves, Beyond the Cox model: artificial neural networks for survival analysis part II, Proceedings of the Eighteenth International Conference on Systems Engineering (2006)
[25] Kazmierczak K., Application of autoregressive prognostic techniques in diagnostics, Proceedings of the Vehicle Diagnostics Conference, Tuczno, Poland, 1983.
[26] Leao, B.P.; Yoneyama, T.; Rocha, G.C.; Fitzgibbon, K.T. « Prognostics performance metrics and their relation to requirements, design, verification and cost-benefit » in international conference on Prognostics and Health Management (PHM 2008), 2008. Denver CO,
[27] Lee J., A systematic approach for developing and deploying advanced prognostics technologies and tools: methodology and applications, Proceedings of the Second World Congress on Engineering Asset Management, Harrogate, UK, 2007, pp. 1195–1206.
[28] LUH Guan-Chun, WU Chun-Yin, CHENG Wei-Chong « Artificial immune regulation (AIR) for model-based fault diagnosis » Artificial immune systems : (Catania, 13-16 September 2004)ICARIS 2004 : international conference on artificial immune systems No3, Catania , ITALIE (13/09/2004) 2004 , vol. 3239, pp. 28-41.
[29] Ma L., Condition monitoring in engineering asset management, Proceedings of Asia-Pacific Vibration Conference, Sapporo, Hokkaido, Japan, 2007.
[30] Mandic D.P., Chambers J.A., Recurrent Neural Networks for Prediction: Learning Algorithms and Architectures and Stability, Chichester, John Wiley & Sons, 2001.
[31] Menezes Jr. José Maria P., and Guilherme A. Barreto, « Long-term time series prediction with the NARX network: An empirical evaluation » Neurocomputing Volume 71, Issues 16-18, October 2008, Pages 3335-3343 Advances in Neural Information Processing (ICONIP 2006) / Brazilian Symposium on Neural Networks (SBRN 2006).
[32] Muller A., Suhner M.C., Iung B., "Formalisation of a new prognosis model for supporting proactive maintenance implementation on industrial system", Reliability Engineering and System Safety, vol. 93, 2008, p. 234-253.
[33] Palluat N., D. Racoceanu1, and N. Zerhouni « A neuro-fuzzy monitoring system: Application to flexible production systems » Computers in Industry (Elsevier) Volume 57, Issue 6, August 2006, Pages 528-538 E-maintenance Special Issue
[34] Pham, H.T., Tran, V.T., Yang, B-S., "A hybrid of nonlinear autoregressive model with exogenous input and autoregressive moving average model for long-term machine state forecasting", Expert Systems with Applications (Elsevier) (2009)
[35] Pisoni E., Farina M., Carnevale C., Piroddi L., "Forcasting peak air pollution levels using NARX models" Engineering Applications of Artificial Intelligence (Elsevier) Vol. 22 (2009) p. 593-602.
[36] Pusey H.C. and M.J. Roemer, An assessment of turbomachinery condition monitoring and failure prognosis technology, The Shock and Vibration Digest 31 (1999), pp. 365–371.
[37] Roemer M.J., G.J. Kacprzynski, E.O. Nwadiogbu, G. Bloor, Development of diagnostic and prognostic technologies for aerospace health management applications, Proceedings of the IEEE Aerospace Conference, Big Sky, Montana, USA, 2001, pp. 3139–3147.
[38] Samantaa B., and S. Bandopadhyay « Construction of a radial basis function network using an evolutionary algorithm for grade estimation in a placer gold deposit » Computers & Geosciences (Elsevier) Volume 35, Issue 8, August 2009, Pages 1592-1602
[39] Saxena, A.; Celaya, J.; Balaban, E.; Goebel, K.; Saha, B.; Saha, S.; Schwabacher, M., « Metrics for evaluating performance of prognostic techniques », in international conference on Prognostics and Health Management (PHM 2008), 2008. Denver CO,
[40] Saxena Abhinav, Bhaskar Saha, Jose Celaya, Kai Goebel, Sankalita Saha, « On Applying the Prognostics Performance Metrics », in annual Conference of the Prognostics and Health Management Society (PHM) 2009 San Diego, CA September 27 – October 1, 2009
[41] Setiono R., W.K. Leow, and J.Y.L. Thong, Opening the neural network black box: An algorithm for extracting rules from function approximating artificial neural networks, Proceedings of International Conference on Information Systems, Brisbane, Australia, 2000, pp. 176–186.
[42] Shao Y., K. Nezu, Prognosis of remaining bearing life using neural networks, Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering, 2000, pp. 217–230.
[43] Sum J.P.F., Kan W.-K, Young G.H, "a note on the equivalence of NARMAX anf RNN", Neural Computing & Applications (Springer-Verlag), (1999), vol8, p. 33-39
[44] Tickle A.B., R. Andrews, M. Golea and J. Diederich, The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural networks, IEE Transactions on Neural Networks 9 (1998), pp. 1057–1068.
[45] Tse P. and D. Atherton, Prediction of machine deterioration using vibration based fault trends and recurrent neural networks, Transactions of the ASME: Journal of Vibration and Acoustics 121 (1999), pp. 355–362.
[46] Vachtsevanos G., Lewis F.L., Roemer M., Hess A., Wu B., Intelligent Fault Diagnosis and Prognosis for Engineering Systems, Hoboken, New Jersey, Wiley & Sons, 2006.
[47] Wang P. and G. Vachtsevanos, Fault prognostics using dynamic wavelet neural networks, Artificial Intelligence for Engineering Design, Analysis and Manufacturing 15 (2001), pp. 349–365.
[48] Wang W., An adaptive predictor for dynamic system forecasting, Mechanical Systems and Signal Processing 21 (2007), pp. 809–823.
[49] Wang W.Q., M.F. Golnaraghi and F. Ismail, Prognosis of machine health condition using neuro-fuzzy systems, Mechanical Systems and Signal Processing 18 (2004), pp. 813–831.
[50] Wei H.L., D.Q. Zhu, S.A. Billings and M.A. Balikhin, « Forecasting the geomagnetic activity of the Dst index using multiscale radial basis function networks » Advances in Space Research Volume 40, Issue 12, 2007, Pages 1863-1870
[51] Yam R.C.M., Tse P.W., Li L., Tu P., "Intelligent predictive decision support system for condition-based maintenance", International Journal of Advanced Manufacturing Technology, vol. 17, 2001, p. 383-391.
[52] Zhang G., Patuwo B.E., Hu M.Y., "Forecasting with artificial neural networks: the state of the art", International Journal of Forecasting, vol. 14, 1998, p. 35-62.

[53] Zemouri R., Racoceanu D., Zerhouni N. – b – « Réseaux de neurones Récurrents à Fonction de base Radiales :RRFR/ Application au pronostic », Revue d'Intelligence Artificielle, RSTI série RIA, Vol. 16, N°03, 2002.
[54] Zemouri R., Racoceanu D., Zerhouni N., "Recurrent Radial Basis Function network for Time-Series Prediction", Engin. Appl. of Artificial Intelligence, vol. 16, 2003, pp.453-463.
[55] Zemouri R., Paul Ciprian Patic, « Prediction Error Feedback for Time Series Prediction: a way to improve the accuracy of predictions », The 4th EUROPEAN COMPUTING CONFERENCE (ECC'10) WSEAS Conferences in the Universitatea Politehnica, Bucharest, Romania, April 20-22, 2010.
[56] EKONOMOU L., S.SP. PAPPAS, "Implementation of Artificial Intelligence in the Time Series Prediction Problem", Proceedings of the 6th WSEAS International Conference on Simulation, Modelling and Optimization, Lisbon, Portugal, September 22-24, 2006.
[57] Suyeon Kim, Byungwhan Kim, "Modeling of Ion Energy Distribution Using Time-Series Neural Network", 12th WSEAS International Conference on SYSTEMS, Heraklion, Greece, July 22-24, 2008
[58] MINJI KWON, BYUNG CHAN PARK, BYUNGWHAN KIM, "Prediction of Chamber Leak Pattern Using Time-Series Neural Network", 12th WSEAS International Conference on SYSTEMS, Heraklion, Greece, July 22-24, 2008
[59] JOANNA COLLANTES-DUARTE, FRANCKLIN RIVAS-ECHEVERRÍA, "Time Series Forecasting using ARIMA, Neural Networks and Neo Fuzzy Neurons", 3rd WSEAS International Conference on Neural Networks and Applications (NNA '02), Fuzzy Sets and Fuzzy Systems (FSFS '02), Evolutionary Computation (EC '02), Interlaken, Switzerland February 11-14, 2002
[60] Chibanga R., Jean Berlamont, Joos Vandewalle, "Use of Neural Networks to Forecast Time Series: River Flow Modeling", 3rd WSEAS International Conference on Neural Networks and Applications (NNA '02), Fuzzy Sets and Fuzzy Systems (FSFS '02), Evolutionary Computation (EC '02) Interlaken, Switzerland February 11-14, 2002
[61] Emili Balaguer Ballester, Emilio Soria Olivas, Jose Luis Carrasco Rodriguez, Secundino del Valle-Tascon, "Forecasting of surface ozone concentrations 24 hours in advance using neural networks", 2001 WSES International Conferences Neural Network and Applications (NNA 2001), Fuzzy Sets and Fuzzy Systems (FSFS 2001), Evolutionary Computation (EC 2001), Puerto De La Cruz, Tenerife, Canary Islands, Spain, February 11-15, 2001
[62] Lapedes A., Farber R, "Nonlinear signal processing using neural networks: prediction and system modelling. The. Rep. LA-UR87-2662, Los Alamos National Laboratory, 1987.