CiSJ

computing@tanet.edu.te.ua
www.tanet.edu.te.ua/computing

ISSN 1727-6209
International Scientific
Journal of Computing

INDUSTRIAL APPLICATIONS OF ARTIFICIAL NEURAL NETWORKS

Kurosh Madani

Intelligence in Instrumentation and Systems Laboratory (I2S/JE2353 Lab.)
PARIS XII University, Senart-Fontainebleau Institute of Technology,
Bât.A, Av. Pierre Point, F-77127 Lieusaint, France,
{madani ; malti ; chebira}@univ-paris12.fr

Abstract: *In a large number of real world dilemmas and related applications the modeling of complex behavior is the central point. Over the past decades, new approaches based on Artificial Neural Networks (ANN) have been proposed to solve problems related to optimization, modeling, decision making, classification, data mining or nonlinear functions (behavior) approximation. Inspired from biological nervous systems and brain structure, Artificial Neural Networks could be seen as information processing systems, which allow elaboration of many original techniques covering a large field of applications. Among their most appealing properties, one can quote their learning and generalization capabilities. The main goal of this paper is to present, through some of main ANN models and based techniques, their real application capability in real world industrial dilemmas. Several examples through industrial and real world applications have been presented and discussed.*

Keywords: – *Artificial Neural Networks, Industrial Applications, Real-Time, Software Implementation, Hardware Implementation.*

1. INTRODUCTION

In a large number of real world dilemmas and related applications the modeling of complex behavior is the central point. Difficulty could be related to several issues:

- large number of parameters to be taken into account (influencing the behavior) making conventional mathematical tools inefficient,
- strong nonlinearity of the system (or behavior), leading to unsolvable equations,
- partial or total inaccessibility of system's relevant features, making the model insignificant,
- subjective nature of relevant features, parameters or data, making the processing of such data or parameters difficult in the frame of conventional quantification,
- necessity of expert's knowledge, or heuristic information consideration,
- imprecise information or data leakage.

Examples illustrating the above-mentioned difficulties are numerous and may concern various areas. As first example, one can emphasize difficulties related to economical and financial modeling and prediction, where the large number of parameters, on the one hand, and human related factors, on the other hand, make related real world problems among the most difficult to solve. Another example could be given in the frame of the industrial processes and manufacturing where strong nonlinearities related to complex nature of manufactured products affect controllability and stability of production plants and processes. Finally, one can note the difficult dilemma of complex pattern and signal recognition and analysis, especially when processed patterns or signals are strongly noisy or deal with incomplete data.

Over the past decades, new approaches based on Artificial Neural Networks have been proposed to solve problems related to optimization, modeling, decision making, classification, data mining or nonlinear functions (behavior) approximation. Inspired from biological nervous systems and brain structure, Artificial Neural Networks could be seen as information processing systems, which allow the elaboration of many original techniques covering a large field of applications([1] to [13]). Among their most appealing properties, one can quote their learning and generalization capabilities (extrapolation of learned tasks to unknown or unlearned situation).

The main goal of this paper is to present Artificial Neural Network potential, through main ANN models and based techniques, to solve real world industrial problems. Several examples through real world industrial applications have been shown and discussed. The paper has been organized as follows: the next section will present the general principle of Artificial Neural Networks relating it to biological considerations. In the same section two classes of neural models will be introduced and discussed: Multi-layer Perceptron and Kernel Functions based Neural Networks. The section 3 and related sub-sections will illustrate real world examples of application of such techniques. Finally, the last section will conclude the paper.

2. A BRIEF OVERVIEW OF SOME OF USUAL ANN MODELS

The next sub-sections will give a very brief overview of the "Back-Propagation" (BP) based learning rule neural network, known also as "Multi-Layer Perceptron and "Kernel Functions" based neural networks trough one of their particular cases which are "Restricted Coulomb Energy/Radial Basis Functions" (RCE/RBF-like neural networks).

2.1. BACK-PROPAGATION BASED MULTI-LAYER PERCEPTRON

Back-Propagation ([13], [18], [19]) based neural models, called also Back-Propagation based "Multi-Layer Perceptron" (MLP) are sufficiently popular and known. That's why only a very brief prompt will be given here. MLP ANN model is a multi-layer neural network. A neuron in this kind of neural network operates conformably to the general ANN's operation frame described in [13]. The specificity of this class of neural network appears in the learning procedure, called "Back-Propagation of error gradient". The principle of the BP learning rule is based on adjusting synaptic weights proportionally to the neural network's output error. Examples (patterns from learning database) are presented to the neural network, then, for each of learning patterns, the neural network's output is compared to the desired one and an "error vector" is evaluated. Then all synaptic weights are corrected (adjusted) proportionally to the evaluated output error. Synaptic weights correction is performed layer by layer from the output layer to the input layer. So, output error is back-propagated in order to correct synaptic weights. Generally, a quadratic error criterion, given by equation (1), is used. Synaptic weights are modified according to relation (2). This coefficient is decreased progressively during the learning process. The learning process stops when the output error reaches some acceptable value.

$$\varepsilon_i = \frac{1}{2}\left(S_i - S_i^d\right)^2, \tag{1}$$

$$dW_{i,j}^h = -\mathbf{3} \bullet \mathbf{grad}_W\left(\varepsilon\right), \tag{2}$$

where S_i – i-th output vector's component, S_i^d – desired value of this component, $dW_{i,j}^h$ – synaptic variation of the synaptic weight connecting the j-th neurone and i-th neuron between two adjacent layers, and η – real coefficient called also "learning rate". This coefficient is decreased progressively during the learning process. The learning process stops when the output error reaches some acceptable value.

2.2. KERNEL FUNCTION BASED NEURAL MODELS

This kind of neural models belong to the class of "evolutionary" learning strategy based ANN ([12], [17], [21]). That means that the neural network's structure is completed during the learning process. Generally, such kind of ANNs includes three layers: an input layer, a hidden layer and an output layer. Figure 3 represents the bloc-diagram of such neural net. The number of neurons in input layer corresponds to the processed patterns dimensionality e.g. to the problem's feature space dimension. The output layer represents a set of categories associated to the input data. Connections between hidden and output layers are established dynamically during the learning phase. It is the hidden layer which is modified during the learning phase.

A neuron from hidden layer is characterized by its "centre" representing a point in an N dimensional space (if the input vector is an N-D vector) and some decision function, called also neuron's "Region Of Influence" (ROI). ROI is a kernel function, defining some "action shape" for neurons in treated problem's feature space. In this way, a new learning pattern is characterized by a point and an influence field (shape) in the problem's N-D feature space. In the other words, the solution is mapped thank to learning examples in problem's N-D feature space.

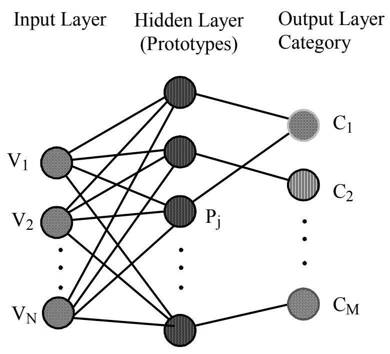

Fig.1 - Radial Basis Functions based ANN's bloc-diagram.

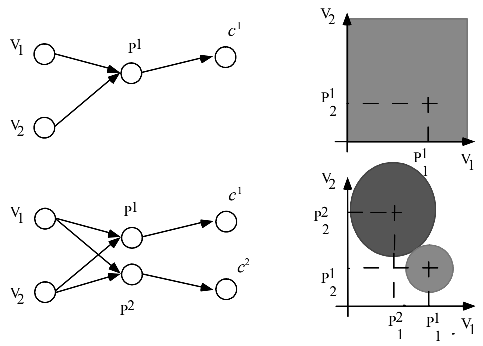

Fig.2 - Example of learning process in 2-D feature space.

The goal of the learning phase is to partition the input space associating prototypes with a categories and an influence field, a part of the input space around the prototype where generalization is possible. When a prototype is memorized, ROI of neighbouring neurons are adjusted to avoid conflict between neurons and related categories. The neural network's response is obtained from relation (3) where C_j represents a "category", $V = [V_1 \quad V_2 \quad \ldots \quad V_N]^T$ is the input vector, $P^j = [p_1^j \quad p_2^j \quad \ldots \quad p_N^j]^T$ represents the j-th "prototype" memorized (learned) thanks to creation of the neuron j in the hidden layer, and λ_j the ROI associated to this neuron (neuron j).

$$\begin{array}{lll} C_j = F\left(dist\left(V, P^j\right)\right) & If & dist\left(V, P^j\right) \leq \lambda_j \\ C_j = 0 & If & dist\left(V, P^j\right) > \lambda_j \end{array}, \qquad (3)$$

where, F(.) – neuron's activation (decision) function. Usually, this function is a radial basis function (a Gaussian function for example).

The choice of the distance calculation (choice of the used norm) is one of the main parameters in the case of the RCE-KNN like neural models (and derived approaches). The most usual function used to evaluate the distance between two patterns is the Minkowski function expressed by relation (3), where V_i is the i-th component of the input vector and p_i^j the i-th component of the j-th memorized pattern (learned pattern). Manhattan distance ($n = 1$, called also L1 norm) and Euclidean distance ($n = 2$) are particular cases of the Minkowski function and the most applied distance evaluation criterions. One can write relation (4) and (5).

$$dist = \sqrt[n]{\sum_i \left|V_i - p_i^j\right|^n}, \qquad (4)$$

$$\sum_i \left|V_i - p_i^j\right| \leq \left(\sum_i \left(V_i - p_i^j\right)^2\right)^{\frac{1}{2}} \leq \max_i \left|V_i - p_i^j\right|, \qquad (5)$$

3. NEURAL NETWORKS BASED REAL-WORLD INDUSTRIAL SOLUTIONS

If the problem's complexity, appearing through theoretical tools (modeling or conceptual complexity) needing to solve it, is the central challenge of the applicability of issued concepts, another key points characterizing application design, especially in industrial environment, is related to implementation requirements. En fact, constraints related to production conditions, market (economical conditions), quality, etc. set the above-mentioned point as a chief purpose to earn solution's viability. That is why in the next subsections, dealing with application of above-presented ANN models, the implementation issues will be of central considerations. Progress accomplished during the lasts decades concerning electrical engineering, especially in the microprocessors area, offers new perspectives in regard to the real time execution capabilities and enlarges the field in solution implementation ability.

3.1 MULTI-LAYER PERCEPTRON BASED INTELLIGENT ADAPTIVE CONTROL

Two meaningful difficulties characterize the controller dilemma, making controllers design one of the most challenging tasks: the first one is the plant parameters identification, and the second one is related to the consideration of interactions between real world (environment) and control system, especially in the case of real-world applications where controlled phenomena and related parameters deal with strong nonlinearities. Beside these two difficulties, another chief condition for conventional or unconventional control is related to the controller's implementation which deals with real-time execution capability. Neural models offer original perspectives to overcome the two firsts difficulties. On the other hand, availability of powerful microprocessors, offers new perspectives for software or hardware implementation, overcoming real time execution constraints.

Before analyzing relationship between the neural network learning and the control dilemma, let us reconsider the case of the conventional control dilemma.

3.1.1 CONTROL DILEMMA: GENERAL FRAME AND FORMALIZATION

Figure 3 gives the general bloc-diagram of two control strategies: open-loop controller and feed-back loop controller (known also as feed-back loop regulation). E_k is the "input vector" (called also "order" vector), $Y_k = \left(y_k \quad y_{k-1} \quad \cdots \quad y_{k-m}\right)^T$ is the "output vector" (plant's or system's state or response) and $U_k = (u_k \quad u_{k-1} \quad \cdots \quad u_{k-n})^T$ is the "command vector", where k is the discrete time variable. The output vector is defined as a vector which components are the m last system's outputs. In the same way, the command vector is defined as a vector which components are the n last commands. Such vectors define output and command feature spaces of the system. Taking into account the general control bloc diagram (figure 3), the goal of the command is to make converge the system's output with respect to some "desired output" noted Y_d. Associating two above mentioned feature spaces, it is usual to consider a hybrid representation called "command-output"vector:

$$\chi_k = (y_k \quad y_{k-1} \quad \cdots \quad y_{k-m}, u_k \quad u_{k-1} \quad \cdots \quad u_{k-n})^T .$$

So, considering the above-mentioned formalization, if the command vector is a subject of some modifications, then the output vector will be modified. The output modification will be performed with respect to the system's (plant, process or system under control) characteristics according to equation (6), where **J** represents the Jacobean matrix of the system.

$$dy_k = \mathbf{J}\, d\chi_k \tag{6}$$

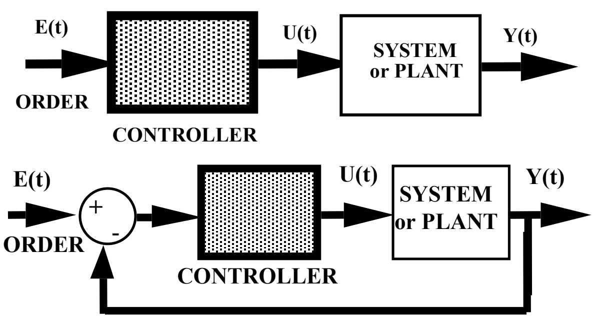

Fig.3 - General bloc-diagrams of control strategies showing open-loop controller (up) and feed-back loop controller (bottom) principles.

So, considering that the actual instant is k, it appears that to have an appropriated output ($Y_{k+1} = Y_d$), the output should be corrected according to the output error defined by: $dY_k = Y_k - Y_d$. In the frame of such formulation, supposing that one could compute the system's reverse Jacobean the command correction making system's output to converge to the desired state (or response) will be conform to relation (7). System's Jacobean is related to plant's features (parameters) involving difficulties mentioned before. Moreover, system's reverse Jacobean computation is not a trivial task. In the real world applications, only in very few cases (as linear transfer functions) the system's reverse Jacobean is available. So, typically a rough approximation of this matrix is obtained.

$$d\chi_k = \mathbf{J}^{-1}\, dy_k \tag{7}$$

3.1.2 ARTIFICIAL NEURAL NET BASED INTELLIGENT CONTROLLER

Let us consider a neural network approximating (learning) a given system (process or plant). Let Y be the system's output, U be the system's command (U becomes also the neural network's output), W_{ij} be synaptic weights of the neural network and ε be the output error representing some perturbation occurring on output. The part of output perturbation (output error) due to the variation of a given synaptic weight (W_{ij}) of the neural network noted as $\partial\varepsilon / \partial W_{ij}$ could be written conformably to relation (8). One can remark that $\partial y / \partial u$ is the system's Jacobean element and $\partial u / \partial W_{ij}$ could be interpreted as the "neural network's Jacobean" element. As the output error is related to the system's controller characteristics (represented by system's Jacobean), so the modification of synaptic weights with respect to the measured error (e.g. the neural network appropriated training) will lead to the correction of the command (dU) minimizing the output error.

$$\frac{\partial \varepsilon}{\partial W_{ij}} = \frac{\partial \varepsilon}{\partial y}\,\frac{\partial y}{\partial u}\,\frac{\partial u}{\partial W_{ij}} \tag{8}$$

Several Neural Network based adaptive control architectures have still been proposed. However, the most effective scheme is the hybrid neuro-controller ([7] to [11]). This solution operates according to the Neural Network based correction of a conventional controller. Figure 4 shows the bloc diagram of such approach. As one can see in our ANN based control strategy, the command U(t) is corrected thanks to the additional correction dU, generated by neural device and added to the conventional command component. The Neural Network's learning could be performed on-line or off-line. Several advantages characterize the proposed strategy. The first one is related to the control system stability. En fact, in the worst case the controlled plant will operate according to the conventional control loop performances and so, will ensure the control system's stability. The second advantage of such strategy is related to the fact that the proposed architecture acts as a hybrid control system where usual tasks are performed by a conventional operator and unusual operations (such as highly non linear operations or those which are difficult to be modelled by conventional approaches) are realized by neural network based component. This second advantage leads to another main welfare which is the implementation facility and so, the real-time execution capability. Finally, the presented solution takes into account industrial environment reality where most of control problems are related to existent plant dealing with an available (still implemented) conventional controller. This last advantage of the proposed solution makes it a viable option in industrial environment.

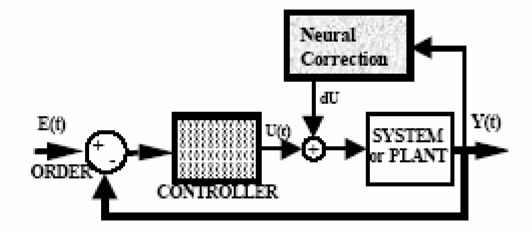

Fig. 4 - Bloc-diagram of hybrid neuro-controller.

We have applied the above-exposed neural based adaptive controller to enhance the conventional vector-control driving a synchronous 3-phased alternative motor. The goal of a vector control or field-oriented control is to drive a 3-phased alternative motor like an independent excitation D.C motor. This consists to control the field excitation current and the torque generating current separately [23]. The input currents of the motor should provide an electromagnetic torque corresponding to the command specified by the velocity regulator. For synchronous motor, the secondary magnetic flux (rotor) rotates at the same speed and in the same direction as the primary flux (stator). To achieve the above-mentioned goal, the three phases must be transformed into two equivalent perpendicular phases by using the Park transformation which needs the rotor position, determined by a transducer or a tachometer. In synchronous machine, the main parameters are Ld (inductance of d-phase), Lq (inductance of q-phase), and Rs (statoric resistor), which vary in relation with currents (Id and Iq), voltages (Vd and Vq), mechanical torque and speed (of such machine). The relations between voltages or currents depend on these three parameters defining the motor's model. However, these parameters are not easily available because of their strongly nonlinear dependence to the environment conditions and high number of influent conditions.

The neural network is able to identify these parameters and to correct the machine's reference model, feeding back their real values through the control loop. Parameters are related to voltages, currents, speed and position. The command error (measured as voltage error) could be linked to the plant's parameters values error. In the first step, the command is computed using nominal theoretical plant parameters. The neural network learns the plant's behaviour comparing outputs voltages (Vd ,Vq), extracted from an impedance reference model, with measured voltages (Vdm,Vqm).

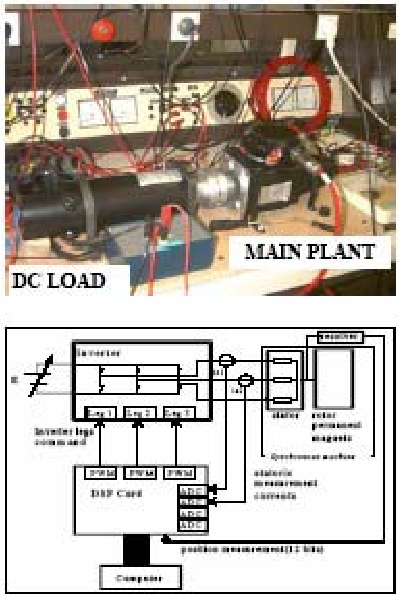

Fig. 5 - A zoom view of the main plant and the coupled load to that plant (up). Implementation block diagram (down).

In the second step when the system is learned, the neural network gives the estimated plant's parameters to the controller [23].

- The complete system, including the intelligent neuro-controller, a power interface and a permanent synchronous magnet motor (plant), has been implemented according to the bloc diagram of figure 5. Our intelligent neuro-controller has been implemented on a DSP based board. In this board, the main processor is the TMS C330 DSP from Texas Instruments. The learning data base includes 675 different values of measurement extracted motor's parameters (Ld and Lq).

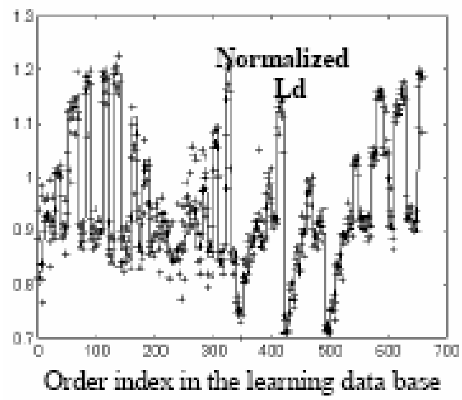

Fig. 6 - Plant parameters identification by neural net.

Different values of measurable parameters (currents, voltages, speed and position), leading to motor's parameters extraction, have been obtained for different operation modes of the experimental plant, used to validate our concepts.

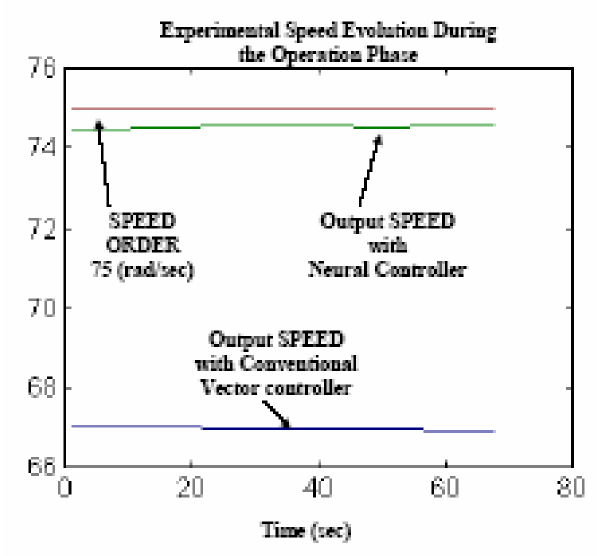

Fig. 7 - Experimental plant's speed measured when the plant is unloaded.

The ANN learning is shifted for 4 seconds after power supply application to avoid unstable data in the starting phase of the motor. Figures 6 and 7 give experimental results relative to the motor's internal parameter evolution and the plant's measured speed, respectively. One can remark from those figures that:

- Internal plant model's parameters are identified by the neural network,

Such neural based controller compensates the inefficiency of the classical control loop (achieving a 74 rad/sec angular speed).

3.2 KERNEL ANN BASED IMAGE PROCESSING FOR INDUSTRIAL APPLICATIONS

As a result of their adaptability, artificial neural networks present also good solutions for image processing and related problems which became during the last decades central points of an ever-increasing range of industrial applications. Moreover, these solutions may take advantage from the power given by the high degree of parallelism provided, on the one hand by image's parallel nature, and on the other hand, by parallelism issued from hardware implementation of ANN. This section and related subsections will focus the hardware implementation and use of such neural image processing technique to improve two different classes of applications in two different industrial domains: media-movie industry and VLSI production industry. Before presenting those two industrial dilemmas, let focus the next section on ZISC-036 neuro-processor from IBM.

3.2.1 IBM ZISC-036 NEURO-PROCESSOR

The IBM ZISC-036 ([21], [22]) is a parallel neural processor based on the RCE and KNN algorithms. Each chip is capable of performing up to 250 000 recognitions per second. Thanks to the integration of an incremental learning algorithm, this circuit is very easy to program in order to develop applications; a very few number of functions (about ten functions) are necessary to control it. Each ZISC-036 like neuron implements two kinds of distance metrics called L1 and LSUP respectively. Relations (9) and (10) define the above-mentioned distance metrics were P_i represents the memorized prototype and V_i is the input pattern. The first one (L1) corresponds to a polyhedral volume influence field and the second (LSUP) to a hyper-cubical influence field.

$$\text{L1: } dist = \sum_{i=0}^{n} \left| V_i - P_i \right| \tag{9}$$

$$\text{LSUP: } dist = \max_{i=0 \ldots n} \left| V_i - P_i \right| \tag{10}$$

ZISC-036 is composed of 36 neurons. This chip is fully cascadable which allows the use of as many neurons as the user needs (a PCI board is available with a 684 neurons). A neuron is an element, which is able to:

- memorize a prototype (64 components coded on 8 bits), the associated category (14 bits), an influence field (14 bits) and a context (7 bits),
- compute the distance, based on the selected norm (norm L1 given by relation or LSUP) between its memorized prototype and the input vector (the distance is coded on fourteen bits),
- compare the computed distance with the influence fields,
- communicate with other neurons (in order to find the minimum distance, category, etc.),
- adjust its influence field (during learning phase).

Figures 8 and 9 give the ZISC-036 chip's bloc diagram and an example of input feature space mapping in a 2-D space, respectively. A 16 bit data bus handles input vectors as well as other data transfers (such as category and distance), and chip controls. Within the chip, controlled access to various data in the network is performed through a 6-bit address bus.

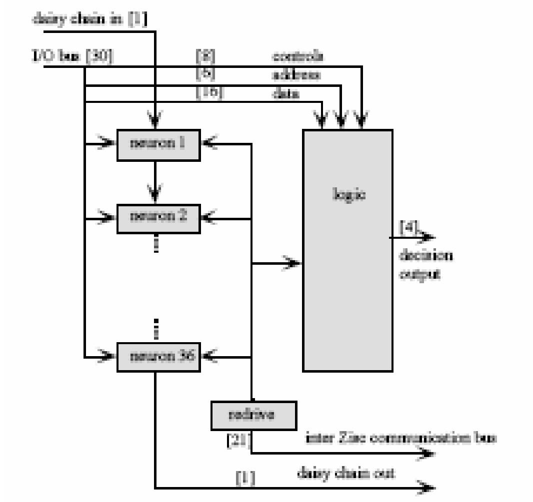

Fig. 8 - IBM ZISC-036 chip's bloc diagram.

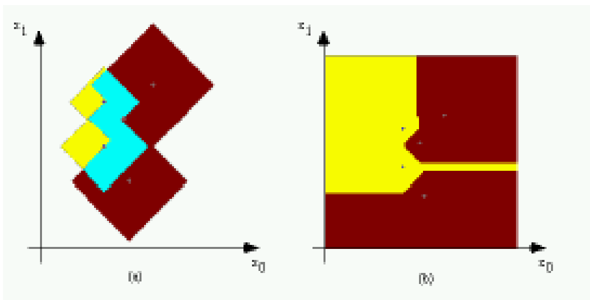

Fig. 9 - Example of input feature space mapping in a 2-D space using ROI and 1-NN modes, using norm.

3.2.2 IMAGE CORRECTION AND COLORATION IN MEDIA AND MOVIE PRODUCTION INDUSTRY

The first class of application concerns image enhancement in order to: restore old movies (noise reduction, focus correction, etc.), improve digital television, or handle images which require adaptive processing (medical images, spatial images, special effects, etc.).

The used principle is based on an image's physics phenomenon which states that when looking at an image through a small window, there exist several kinds of shapes that no one can ever see due to their proximity and high gradient (because, the number of existing shapes that can be seen with the human eye is limited). ZISC-036 is used to learn as many shapes as possible that could exist in an image, and then to replace inconsistent points by the value of the closest memorized example. The learning phase consists of memorizing small blocks of an image (as an example 5x5) and associating to each the middle pixel's value as a category. These blocks must be chosen in such a way that they represent the maximum number of possible configurations in an image. To determine them, the proposed solution consists of computing the distances between all the blocks and keeping only the most different.

The learning algorithm used here incorporates a threshold and learning criteria (Learn_Crit (V)). The learning criteria is the criteria given by relation (11) where V_l^k represents the l-th component of the input vector V^k, P_l^j represents the l-th component of the j-th memorized prototype, C^k represents the category value associated to the input vector V^k, C^j is the category value associated to the memorized prototype P^j and, α and β are real coefficients adjusted empirically.

$$Learn\_Crit\left(V^k\right) = \alpha \sum_l \left|V_l^k - P_l^j\right| + \beta\left|C^k - C^j\right| \qquad (11)$$

An example (pattern) from the learning base is chosen and the learning criterion for that example is calculated. If the value of the learning criteria is greater than the threshold, then a neuron is engaged (added). If the learning criteria's value is less than the threshold, no neuron is engaged. For each iteration, the aforementioned threshold is decreased. Once learning database is learned the training phase is stopped. Figure 10 shows two learning examples on the basis of pattern-to-category association and region-to-region association (bloc-diagram of a typical learning phase).

The image enhancement or noise reduction principles are the same as described above. The main difference lies in the pixel value associated to each memorized example. In noise reduction, the learned input of the neural network is a noisy form of the original image associated with the correct value (or form). For example, in the figure 10, for each memorized example (a block of 5x5) from the input image (degraded one), the middle pixel of the corresponding block from the output image (correct one) is used as the "corrected pixel value" and is memorized as the associated category. After having learned about one thousand five hundred examples, the ZISC-036 based system is able to enhance an unlearned image. Figure 11 and figure 12 give results corresponding to noise filtering and movie sequences coloration, respectively.

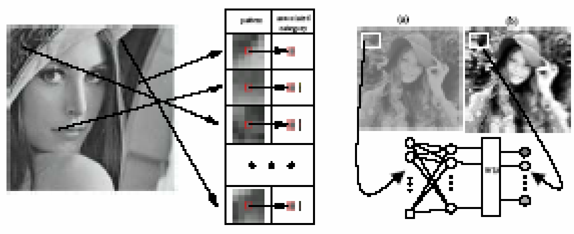

Fig. 10 - Learning process examples: associating a pixel to a category (left) and association of regions from the degraded and correct images (right).

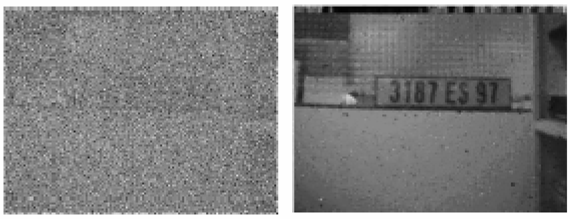

Fig. 11 - Result concerning restoration of a noisy unlearned image: input image (left) and restored image (right).

Fig.12 - Result concerning movie coloration. Image used for the learning phase (left). Coloration result obtained for an unlearned image (right).

In the case of image restoration and coloration it has been shown ([24], [25]) that the same neural concept could perform different tasks as noise reduction, image enhancement and image coloration which are necessary to restore a degraded movie. Quantitative comparative studies established and analysed in above-mentioned references show pertinence of such techniques. Figure 13 gives a quantitative comparison between colours in reconstructed images and those in the original image (which has been used as learning reference).

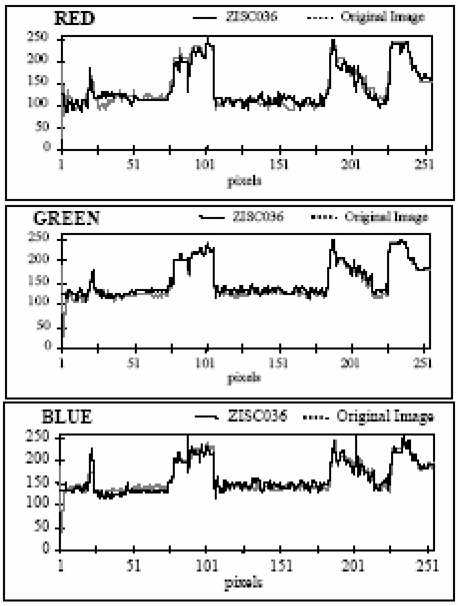

Fig.13 – Comparison of the colored (reconstructed) image with the original image in generalization phase.

3.2.3 VISUAL PROBE MARK DETECTION AND CATEGORIZATION IN VLSI PRODUCTION

One of the main steps in VLSI circuit production is the testing step. This step verifies if the final product (VLSI circuit) operates correctly or not. The verification is performed thank to a set of characteristic input signals (stimulus) and associated responses obtained from the circuit under test. A set of such stimulus signals and associated circuit's responses are called test vectors. Test vectors are delivered to the circuit and the circuit's responses to those inputs are catch through standard or test dedicated Input-Output pads (I/O pads) called also vias. As in the testing step, the circuit is not yet packaged, the test task is performed by units, which are called probers including a set of probes performing the communication with the circuit.

Figure 14 shows a picture of probes relative to such probers. The problem is related to the fact that the probes of the prober may damage the circuit under test. So, an additional step consists of inspecting the circuit's area to verify vias (I/O pads) status after circuit's testing: this operation is called Probe Mark Inspection (PMI). Figure 15 shows two examples of probe impacts produced during the circuit's test phase. The first one (left) corresponds to a correct impact (circuit hasn't been damaged) and the second one (right) to a damaged circuit.

Many prober constructors had already developed Probe Mark Inspection (PMI) software based on conventional pattern recognition algorithms with little success [20]. The difficulty lies in the response time (real time execution with production speed constraints) and method reliability compromise. Even sophisticated hardware using DSPs and ASICs specialized in image processing has not performed sufficiently well to convince industrials to switch from human visual defects recognition to electronically automatic PMI.

Fig.14 - Photograph giving an example of probes in industrial prober.

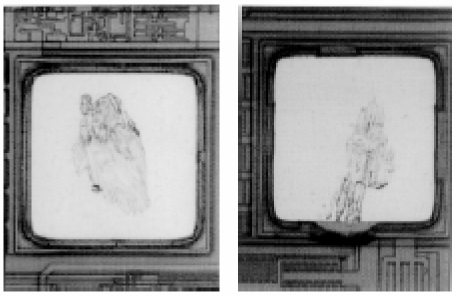

Fig.15 – Example of probes impacts corresponding to a correct (left) and damaged (right) vias, respectively.

So as reader could guess, the second kind of applications concerned the visual probe mark inspection dilemma in VLSI production. A neural network based solution has been developed and implemented on ZISC-036 neuro-processor, for the IBM Essonnes plant. The main advantages of developed solutions are real-time control and high reliability in detection and classification tasks. Our PMI application, presented in [22] and [24], consists of software and a PC equipped with this neural board, a video acquisition board connected to a camera and a GPIB control board connected to a wafer prober system. Its goal is image analysis and prober control. Figure 16 represents the bloc diagram of the application.

The process of analyzing a probe mark can be described with the following steps:

- the PC commands the prober to move the chuck so that the via to inspect is precisely located under the camera.
- an image of the via is taken through the video acquisition board.
- the application, using the ZISC-036, then:
- finds the via on the image.
- check the integrity of the border (for damage) of via.
- locates the impact in the via and estimates its surface for statistics.

The application then moves on to the next via. At the end of the process, the system shows a wafer map which presents the results and statistics on the probe quality and its alignment with the wafer. All the defects are memorized in a log file. In summary, the detection and classification tasks of our PMI application are done in two steps: localization the via on the acquired image, then, mark size estimation and probe impact classification (good, bad or none).

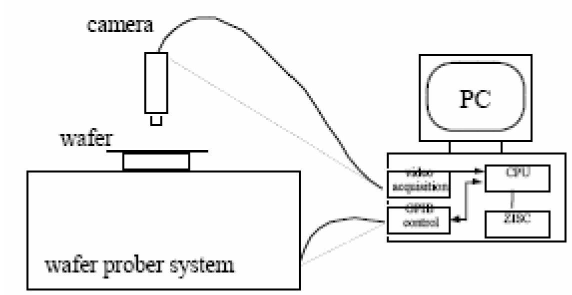

Fig.16 - Bloc-diagram of developed kernel neural networks based solution.

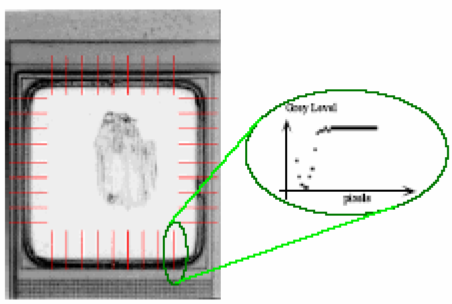

Fig.17 - Example of profiles extraction after via centring process.

Profile	Category	Profile	Category
	Faulty		OK
	Faulty		OK
	Faulty		OK

Fig.18 - Example of profiles to category association during the learning.

The method, which was retained, is based on profiles analysis using kennel functions based ANN. Each extracted profile of the image (using a square shape, figure 17) is compared to a reference learned database in which each profile is associated with its appropriated category. Different categories, related to different needed features (as: size, functional signature, etc). Figure 18 chows profile-to-fault association example. Finally, figure 19 shows impact's size's related profile (left picture) and a faulty via detected by the implemented intelligent visual probe mark inspector (right picture).

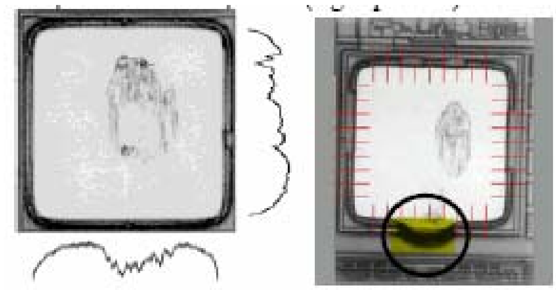

Fig.19 - Profiles extraction for size and localization of the probe mark (left). Experimental result showing a fault detection and its localization in the via (right).

Experiments on different kinds of chips and on various probe defects have proven the efficiency of the neural approach to this kind of perception problem. The developed intelligent PMI system outperformed the best solutions offered by competitors by 30%: the best response time per via obtained using other wafer probers was about 600 ms and our neural based system analyzes one via every 400 ms, 300 of which were taken for the mechanical movements. Measures showed that the defect recognition neural module's execution time was negligible compared to the time spent for mechanical movements, as well as for the image acquisition (a ratio of 12 to 1 on any via). This application is presently inserted on a high throughput production line.

3.2.4 PRODUCTION YIELD PREDICTION IN VLSI IDUSTRY

Process behaviour prediction and modelling are known as difficult classes of problems, especially when the dilemma trades with real world and real complexity constraints (industrial production systems, highly non linear systems, systems with a large number of parameters to be taken into account, etc.). Among these classes of problems, the industrial manufacturing production yield prediction dilemma is of major interest. It is important to discern the prediction from modelling (referring to the manufacturing) by the fact that the prediction concerns short term behaviour evolution estimation whereas the modelling expresses the long term behaviour evolution knowledge [26]. Of course, both of them are essential in the case of the industrial manufacturing production yield estimation dilemma. The prediction concerns the production lines logistics, whereas the modelling is used to understand the correlation between physical parameters to enhance the product quality.

We have employed on RCE-RBF neural network to solve the proposed problem. Among interests of this approach is the ZISC-O36 implementation of this kind of neural model, leading to a real time execution possibility.

Fig.20 - General bloc diagram of proposed solution.

In a more general point of view, the problem on which we are interested, deals with sampled information processing (the information here is collected sampling the production process output information). One of the key points in above mentioned class of problems is the "data pre-processing". The main goal of the data pre-processing step is to validate pertinent data (significant data) and to eliminate insignificant data. The figure 20 shows the block diagram of the proposed approach including two process stages. The first stage performs a data "pre-processing" leading to a validation of considered data. The second stage, a neural based processing stage, is a data classifier. The goal here is to use this classifier to construct some representation of the production yield evolution function (measuring the production yield during the production process). It is important to emphasize that such function is supposed to vary softly.

To validate our approaches, we use a data base of measurements consisting of 322 sets of 25 wafers (approximately). These measurements correspond to different characteristics during the manufacturing process including production yield after circuits on-site testing, electrical characterisation of produced circuits, etc.. So, in this way, for each set of wafers, data corresponding to approximately 30 parameters, including the final production yield for this set, is available. The goal here is to model the production yield short term evolution by learning the mentioned data base. For the learning process we used a part of the complete database. Then, the complete data base has been used for generalization and performance evaluation of such concepts.

3.2.4.1 PREPROCESSING AND DATA VALIDATION

The technique we propose [24] is based on the analysis of the production yield, expressed as a "category variation" (ΔCat), as function of the distance variation (Δdist). Such representation is obtained by computing the distance between each pairs (couples) of measurement (of the data base) and the variation of associated production yield. Figure 21 shows the corresponding diagram in the case of the considered data base. Our data validation technique is based on the physical phenomena continuity hypothesis: two near states of the system (in the representative future space of the system) lead to the same short term behaviour. This hypothesis is applied here to validate the used data. In fact, some of the low values of the production yield could be due to external parameters (events): for example broken wafers due to operators, etc.. In our case, data validation is performed according to the previous hypothesis, and so, points belonging to a close neighbourhood with a large production yield variation (large variation of Δcat) should be rejected: such cases correspond to circles in figure 21.

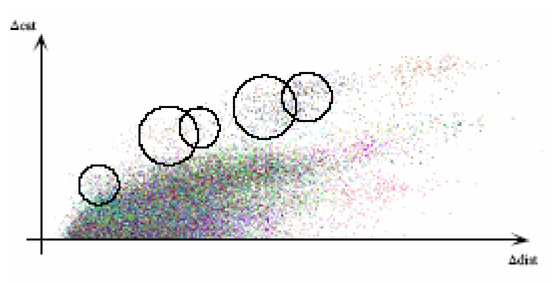

Fig.21 - Production yield variation (Δcat) versus distances variation (Δdist). The circles indicate the rejected data from data base.

3.2.4.2 YIELD ESTIMATION STAGE

As it has been mentioned previously, our technique is based on the analysis of the production yield variation, expressed as a "category variation" (ΔCat), as function of the distance variation (Δdist). So, the second processing stage performs a category classification (where the category is production yield related information) on the basis of a distance evaluation. Several parameters should be considered for an efficient learning phase:

- choice of the prototypes to be memorised,
- number of prototypes used for the learning phase (as small as possible),
- choice of the learning strategy.

In the learning strategy we used, the hidden layer's neural connections are performed according to Grow And Learn (GAL) rule. This rule can be typified by adding a Winner Take All (WTA) decision stage between the hidden layer and the output one. The main advantage related to such learning strategy is to cover all problem's feature space, insuring response stability: which is one of key conditions for industrial applications. After a preliminary data validation (based on the previous hypothesis), the most efficient strategy is to learn the furthest points in feature space (i.e. the space characterized by Δcat as a function of Δdist). For that, the learning process starts with a threshold TH with a high value which decreases during the learning phase. All prototypes for which the condition Δdist > TH is satisfied are memorised. The figure 22 shows the global error evolution (during the learning phase) with respect to the number of learned neurones (neurone in the hidden layer). In this first technique, the system learns (memorises) as many prototypes as are necessary to reach an acceptable global error.

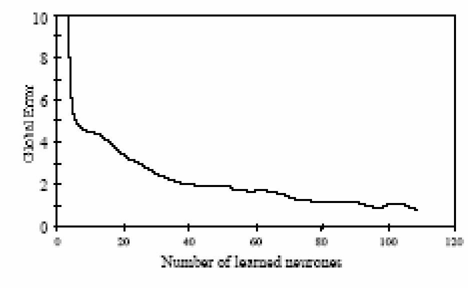

Fig.22 - Global error evolution (learning phase) versus number of connected neurones (in hidden layer).

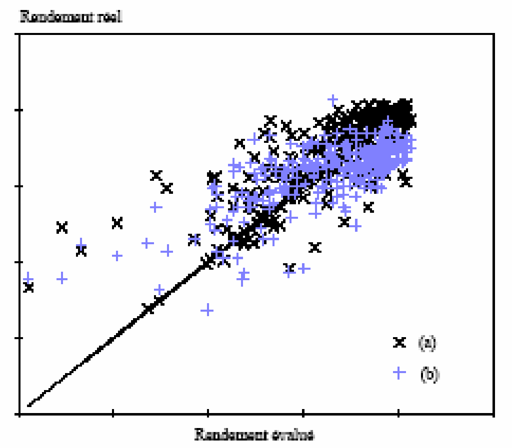

Fig.23 - Experimental results comparing the estimated production yield obtained from our technique (a), with the estimation of this yield by an expert operator (b).

The figure 23 compares the estimated production yield obtained using the neural based technique with the estimation of this yield obtained from a human expert operator. This figure shows the production yield predicted by neural system ("X" marks) and the corresponded estimation performed by a human expert ("+" marks). The continuous line "—" indicates the true prediction. Results represented by the figure 23 have been obtained sing a data base relative to 322 sets of 25 wafers. Because of industrial confidentiality related to the process data, the graduations of the figure 23 are "symbolic".

The remark which could be formulated concerning this results is related to the fact that spatial distributions (around the continuous line) of points representing expert based yield prediction and neural based one are comparable. That shows that the neural and human systems, in this case, are comparable. However, these results are not sufficient to determine if the neural based solution leads to better estimation.

4. CONCLUSIONS

Advances accomplished during last decades in Artificial Neural Networks area and issued techniques made possible to approach solution of a large number of difficult problems related to optimization, modeling, decision making, classification, data mining or nonlinear functions (behavior) approximation. Inspired from biological nervous systems and brain structure, these models take advantage from their learning and generalization capabilities, overcoming difficulties and limitations related to conventional techniques. Today, conjunction of these new techniques with recent computational technologies offers attractive potential for designing and implementation of real-time intelligent industrial solutions. The main goal of the present paper was focused on ANN based techniques and their application to solve real-world and industrial problems. Of course, the presented models and applications don't give an exhaustive state of art concerning huge potential offered by such approaches, but they could give, through above-presented ANN models and related applications, a good idea of promising capabilities of ANN based solutions to solve difficult future industrial changes.

Acknowledgements – Reported works were sponsored and supported by several research projects and industrial partners among which, French Ministry of Education, French Ministry of Research and IBM-France Company. Author whish thank especially Dr. P. Tanhoff, and Dr. G. Detremiolles from IBM-France for their partnership and joint collaboration. I would also acknowledge Dr. V. Amarger, Dr. A. Chebira and Dr. R. Malti from my research team (I^2S Lab. of PARIS XII University). I would thank too Dr. G. Mercier from PARIS XII University, who worked several years with me on intelligent adaptive control dilemma. Finally, I would express my gratitude to Mr. M. Rybnik, my Ph.D. student for useful discussions concerning the present paper.

5. REFERENCES

[1] O. Nelles, On the identification with neural networks as series-parallel and parallel models, *International Conference on Artificial Neural Networks (ICANN'95), Paris, France,* (1995).

[2] W. Faller, S. Schreck, Real-Time Prediction Of Unsteady Aerodynamics : Application for Aircraft Control and Maneuverability Enhancement, *IEEE Transactions on Neural Networks, Vol. 6, N¡ 6,* (1995)

[3] Y. Maidon, B. W. Jervis, N. Dutton, S. Lesage, Multifault Diagnosis of Analogue Circuits Using Multilayer Perceptrons, *IEEE European Test Workshop 96, Montpellier, June 12-14,* (1996).

[4] K. Madani, A. Bengharbi, V. Amarger, Neural Fault Diagnosis Techniques for Nonlinear Analog Circuit, *SPIE Vol. 3077, pp 491-502, , Orlando, Florida, U.S.A., April 21 to 24,* (1997).

[5] A. Sachenko, V. Kochan, V. Turchenko, V. Golovko, J. Savitsky, A. Dunets, T. Laopoulos, Sensor errors prediction using neural networks, *Proceedings IJCNN'2000, Jul 24-Jul 27* (2000), *Como, Italy, pp. 441-446*.

[6] Y. Zhang, Y. MA, CGHA for Principal Component extraction in the complex domain, *IEEE Trans. On Neural Networks, Vol. 8, No. 5,* (1997), *pp. 1031-1036..*

[7] M.Hormel , A Self-organizing associative memory system for control applications, *Advances Neural Information processing Systems 2,Touretzky Ed. Los Altos, CA: Morgan Kaufmann,1992 pp 332-339*.

[8] W.T Miller, Sensor based control of robotic manipulators using a general learning algorithm, *IEEE Journal of Robotics and Automation ,1987 pp 157-165*.

[9] K. Madani, G. Mercier, A. Chebire, S. Duchesne, Implementation of Neural Network Based Real Time Process Control on IBM Zero Instruction Set Computer, *SPIE International AeroSense'96, Applications and Science of Artificial Neural Networks, 9-12 April 1996, Orlando, USA, SPIE Proc. Vol. 2760, pp. 250-261*.

[10] J.S. Albus, A new Approach to Manipulator Control, *Transaction of ASME, September* (1975), *pp 220-227*.

[11] R.F Comoglio, CMAC Neural network Architecture of Autonomous Undersea Vehicle. *SPIE vol 1709. Applications of artificial Neural Networks III, pp 517-527*, (1992).

[12] L.M. Reyneri, Weighted Radial Basis Functions for Improved Pattern Recognition and Signal Processing, *Neural Processing Letters, Vol. 2, No. 3, pp 2-6, May 1995*.

[13] P. Bigot, M. Cosnard, Probabilistic Decision Trees and Multilayered Perceptrons, *Proceedings of the European Symposium on Artificial Neural Networks, ESANN'93, pp 91-96, 1993*.

[14] C.W. Anderson, S.V. Devulapalli, S.V., E.A. Stolz, Determining Mental State from EEG Signals Using Parallel Implementations of Neural Networks, *Scientific Programming, Special Issue on Applications Analysis, 4, 3, Fall, pp 171-183, 1995.*

[15] M.A. Arbib (ed.), "Handbook of Brain Theory and Neural Networks" *2ed. M.I.T. Press. 2003*.

[16] S. Hebb, The Organization of Behaviour, *Wiley and Sons, New-York, U.S.A., 1949.*

[17] T. Kohonen, Self-Organization and Associative Memory, *Springer-Verlag, Germany, 1984*.
[18] D. Rumelhart, G. Hinton, R. Williams, Learning Internal Representations by Error Propagation", *Rumelhart D., McClelland J., "Parallel Distributed Processing: Explorations in the Microstructure of Cognition", I & II, MIT Press, Cambridge MA, 1986*.
[19] M. Bogdan, H. Speakman, W. Rosenstiel, Kobold : A neural Coprocessor for Back-propagation with on-line learning, *Proc. NeuroMicro 94, Torino, Italy, pp. 110-117*.
[20] A. Davis, Using Machine Vision in Vision Semiconductor Wafer Inspection, Advanced Imaging, June 1996.
[21] G. Trémiolles (de), K. Madani, P. Tannhof, A New Approach to Radial Basis Function's like Artificial Neural Networks , *NeuroFuzzy'96, IEEE European Workshop, Vol. 6 N° 2, pp 735-745, April 16 to 18, Prague, Czech Republic, 1996.*
[22] G. Trémiolles (de), P. Tannhof, B. Plougonven, C. Demarigny, K. Madani, Visual Probe Mark Inspection, using Hardware Implementation of Artificial Neural Networks, in VLSI Production, *Lecture Notes in Computer Science - Biological and Artificial Computation : From Neuroscience to Technology, Edited by : Jose Mira, Roberto M. Diaz and Joan Cabestany - Springer Verlag Berlin Heidelberg, pp 1374-1383, 1997*.
[23] K. Madani, G. Mercier, M. Dinarvand, JC Depecker, A Neuro-Vector based electrical machines driver combining a neural plant identifier and a conventional vector controller, *SPIE International Symposium AeroSense'99, Orlando, Florida, USA, April 1999*.
[24] G. De Tremiolles, Contribution à l'étude théorique des modèles neuromimétiques et à leur validation expérimentale: mise en œuvre d'applications industrielles ("Contribution to the theoretical study of neuromimetic models and to their experimental validation: use in industrial applications"), *Ph.D. Report, University of PARIS XII – Val de Marne, March 1998*.
[25] K. Madani, G. De Tremiolles, P. Tanhoff, Image processing using RBF like neural networks : A ZISC-036 based fully parallel implementation solving real world and real complexity industrial problems, *Journal of Applied Intelligence N°18, 2003, Kluwer Academic Publishers, pp. 195-231*.
[26] A. Aussem, " Théorie et application des reseaux de neurones réccurents et dynamiques ", PhD Thesis report, René Descartes - PARIS V University, 1995.
[27] K. Madani, G. De Tremiolles, E. Williams, P. Tanhoff, Neural Based production Yield Prediction: an RBF based approach, SPIE International AeroSense'99 Symposium, Orlando, Florida, USA, 5 - 9 April 1999, *Applications and Science of Computational Intelligence II*, Vol. 3722.

*Prof. **Kurosh Madani** received his Ph.D. degree in Electrical Engineering and Computer Sciences from University PARIS XI (PARIS-SUD), Orsay, France, in 1990. From 1989 to 1990, he worked as assistant professor at Institut d'Electronique Fondamentale (Institute of Fundamental Electronics) of PARIS XI University, Orsay, France. In 1990, he joined Creteil-Senart Institute of Technology of University PARIS XII – Val de Marne, Lieusaint, France, where he worked from 1990 to 1998 as assistant professor. In 1995, he received the DHDR Doctorate degree (senior research doctorate degree) from University PARIS XII – Val de Marne. Since 1998 he works as Chair Professor in Electrical Engineering of Senart Institute of Technology of University PARIS XII – Val de Marne. Since 1992 he is head of Intelligence in Instrumentation and Systems Laboratory of PARIS XII – Val de Marne University located at Senart Institute of Technology. He has worked on both digital and analog implementation of processors arrays for image processing by stochastic relaxation, electro-optical random number generation, and both analog and digital Artificial Neural Networks (ANN) implementation. His current research interests include large ANN structures behavior modeling and implementation, hybrid neural based information processing systems and their software and hardware implementations, design and implementation of real-time neuro-control and neural based fault detection and diagnosis systems. Since 1996 he is a permanent member (elected Academician) of International Informatization Academy. In 1997, he was also elected as Academician of International Academy of Technological Cybernetics.*