PREDICTIVE MAINTENANCE ORIENTED NEURAL NETWORK SYSTEM – PREMON

Javier Ropero Peláez[1], Mariana A . Aguiar[2], Ricardo C. Destro[2],
Zsolt L. Kovács[2], Marcelo G. Simões[3]

[2]Univ. of São Paulo, Dept. of Mechatronics and Mechanical Systems Engineering
[1]Univ. of São Paulo, Dept of Electronics Engineering
[3]Colorado School of Mines, Engineering Division

E-mails: fjavier@usp.br, mariana@lac.usp.br, rdestro@lac.usp.br, kovacs@lac.usp.br, msimoes@mines.edu

Abstract – **The cost of equipment maintenance represents an important budgetary item in industrial and commercial applications. Smart machines are able to evaluate on line a number of its own vitalities helping operators to diagnose faults. Most of time the origins of the problems are buried into intractable and not usually relevant data. Some neural architectures are presented in this paper for recognizing those operational trajectories that are the early symptoms of faults in these smart machines. In order to cope with such classification problem, a neural architecture defined as PREMON (Predictive Maintenance Oriented Network) has been designed. The main advantage of the system is its brain-inspired philosophy that allow it to be applied to a great deal of systems that are degraded or damaged because of their interaction with its environment.**

I. INTRODUCTION

This paper will cover the distinctions regarding fault diagnosis and predictive maintenance. Predictive maintenance aims to prevent the occurrence of faults while fault diagnosis is concerned with the precise identification of the nature of a fault after it occurred. Various papers has been devoted to the generic topic of fault diagnosis in equipments using neural networks of the Multi-layer Perceptron type and radial basis networks [2][3][4]. However, the wide possibilities of the biologically inspired, Self Organizing Map (SOM)[5] was not fully exploited for fault detection or predictive maintenance issues. In this work a SOM approach is combined to some recent works regarding sequences detection in cerebral cortex [6][7]. The SOM is used to classify the early symptoms of the problem and afterwards the resulting classes are connected in accordance to such biological inspiration for linking events for further classification. PREMON capabilities anticipate faults having only early, or premonitory symptoms.

II. UNIVERSAL NEURAL ARCHITECTURE FOR FAULT DETECTION

A block of the main stages of an universal neural architecture for fault detection diagram is presented in Figure 1. Predictive maintenance is accomplished by introduction of time in this static architecture.

The first stage is required for reduction of dimensionality. The operating conditions of an equipment are usually assessed by a certain number of measured variables, although only a subset of these are relevant for revealing the precise nature of a failure. This subset can be isolated by means of a dimensionality reduction process such as principal component analysis; selected relevant variables are used in next stages.

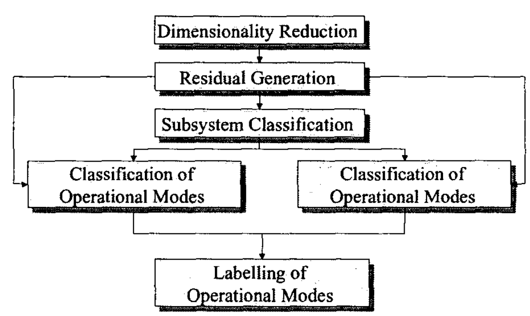

Fig. 1.Universal failure detection architecture

A second stage is essential for obtaining residuals [8] from measured variables. A residual is the difference between the measured variable and its theoretical value in optimal conditions of machine performance. In some cases the theoretical value is obtained from the dynamical equations that model the proper functioning of the machine. In other cases an auto-associative neural network [9] is trained with the same patterns in the input and in the output. These patterns correspond to optimal machine performance. After successful training a degraded input pattern reverted to a non-degraded version of at the output of the network and residuals are obtained by subtracting the input from the output. The current study is focused over a type of "smart" machines that generate fault information equivalent to residuals where the goal is to make those machines skip the above stage of residuals generation. In addition, training subsequent processing stages for normal and faulty modes are needed for correct fault discrimination.

Sub-system classification are related to the fact that usually commercial machines are composed by detached parts that must be analyzed separately because different sensors demand different kinds of information processing. Here such problem is unraveled by using a SOM approach to connect and link events. The classification of operational modes consists in clustering the different input patterns according to a distance criteria. For this purpose several statistical techniques are applicable like K-means, Forgy or nearest neighbors algorithms etc. However we will use for clustering a competitive neural network, the widely known Teuvo Kohonen's Self Organizing Feature Map[5].

The last stage, labeling of operational modes is equivalent to the presentation of targets to a standard neural network. In general, labeling of operational modes means that some of the previously obtained clusters can be arbitrarily labeled. For example, various clusters can be labeled as corresponding to a healthy situation of machine functioning and others to different kinds of faulty modes. Therefore, in this stage, previously obtained clusters are associated to different kinds of fault diagnosis. In the case of several subsystems this linkage can be accomplished through a pattern associator type neural network. Here all subsystems connect to this pattern associator network as an ultimate integrator of machine operational modes. Back-propagation or radial basis algorithms can be used in this pattern associator. In the case of an only one subsystem, and due to the orthogonal exit from the SOFM network, Widrow-Hoff algorithm can be used.

III. A PREDICTIVE MAINTANACE BASED NEURAL NETWORK

In this section we will explain how to deal with time in the previous architecture for fault detection so that it is transformed into an architecture for predictive maintenance. We will sub-divide each sub-system "classification of operational modes" blocks in figure 1 into the four stages shown in figure 2.

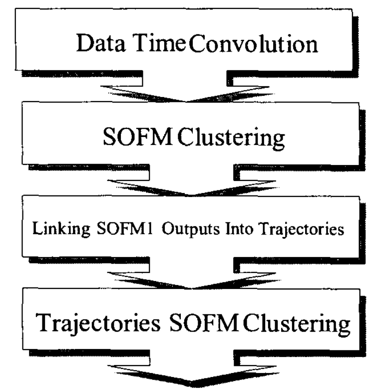

Fig. 2. Main stages in classification of operational modes

As mentioned before our study is focused on some "smart" machine that continuously outputs high level information, equivalent to residuals. The "smart" machine itself gathers this information, that is logged in a file, very much like a black-box in an airplane. In this way our attempt to access information from the machine is mediated by this file. Each line of this file has several fields, each field corresponding to a possible event or alarm. Usually only one event is logged in each line, so that each event information is a vector with only one component different from zero. Summarizing, data is logged in a file and each line of this file is interpreted as a n-dimensional vector: $X^n = [x_1 \ x_2 \ \dots \ x_n]$, being each component a certain information from specific alarm events. Next, the four stages shown in Figure 2 are explained.

A. Data Time Convolution

A log file of a smart machine shows the evolution of static situations in which only one alarm is triggered. There is no continuity between events in successive lines as to say that one event is taking part in the occurrence of the future ones or vice-versa. Therefore it is necessary to transform each line of the log file into a line in which past and future events have a certain contribution. After this transformation, the value of each component of the line will reflect how far is each alarm from the instant the line was logged. If the alarm is triggered in the moment the log line is edited, a value of 1 will appear in the correspondent alarm position inside the log line. If another alarm triggers one second before or one second after the edition of the line, a smaller value (for example 0.8) will appear in the corresponding position of this line. To model this fact, we have introduced a probabilistic smoothing over time for each recorded event as seen in Figure 3. This smoothing is accomplished by convolving the input vector Xn with a Gaussian distribution function resulting in a vector $Y^{n,m}$ which will be the input to a first SOFM network (SOFM1). By doing this we obtain the necessary temporal continuity between successive patterns in the log file. Instead of having crisp patterns from line to line, now log lines contiguity implies similarity, and similar vectors are close together in their n-dimensional space.

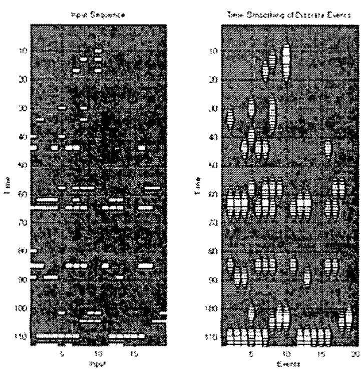

Fig. 3. Input sequence and time discret events

B. Initial clustering using SOFM

SOFM can be regarded as a network which is intended to reduce drastically the dimensionality of input patterns. One of the most important characteristics of SOFM network is to, topologically map the input into the output space. This topological organization is one of the main characteristics in the human brain. SOFM topological mapping [10] means that, input vectors that are close to each other in the input space will produce close together vectors in the two dimensional SOFM output space. In other words SOFM reduces pattern dimensionality while preserving topology. The consequence of applying the SOFM network after the previous process of data convolution is to summarize the spatio-temporal dynamics of a complex input space into a fixed number of neurons' dynamics. Complex sequences of input data are transformed into simple and tractable trajectories of winning neurons in the SOFM two dimensional output layer.

C. Linking SOFM winning neurons to form trajectories

It is necessary to link the information of isolated SOFM winning neuron to the trajectory they belong, as Figure 4 shows. A trajectory is defined as a spatially contiguous set of neurons. Trajectories are obtained by calculating the Manhattan distance between successive winners which form the current trajectory. Whenever this distance exceeds a certain threshold it is assumed that a break in contiguity occurred. At this moment, the information of all winning neurons in the sequence is replaced by the whole trajectory they belongs to, then a new trajectory is initiated. This process of linking active neurons in sequences is, very much, the same process cortical pyramidal neurons are able to perform in accordance to previous references [6][7].

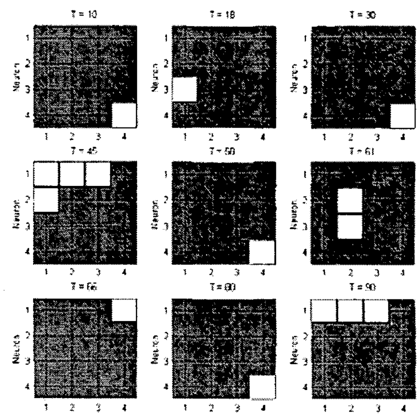

Fig. 4. Trajectories generated from SOFM1 output.

D. Trajectories Clustering through a second SOFM

SOFM has been used in the past for classifying written characters. Here a similar procedure is undertaken. But instead of a printed letter or character, SOFM2 will classify the trajectories that are "drawn" in the grid of SOFM1 output layer shown in Figure 4. This "drawing" was obtained by linking SOFM1 winning neurons, as explained in the previous section. The "drawing" of this trajectories are fed into SOFM2 which will classify them in different groups. Those groups can be understood as the different operational modes of the machine regardless if they represent a fault or not. Finally, a pattern associator network, which can be either a Widrow-Hoff or LVQ network labels the different operational modes into the different diagnoses provided by machine technicians.

IV. EVALUATION RESULTS

Preliminary tests were done using an architecture composed of two 5x5 SOFM networks and a 9x1Widrow net. The system learn 20 different alarm sequences , each one composed of 7 alarms, grouped in 10 categories after 1000 epochs for each SOFM, and 556 epochs for Widrow net. Using the same training patterns for testing, 100% of success was obtained from the very beginning of each sequence, demonstrating the predictive characteristics of the system. When an altered version of training patterns was changed by 10% random noise 99% of success was obtained on identifying each sequence for the very beginning of it where no sequences were overlapped at the beginning. Current tests are being taken to provide further performance indications.

V. CONCLUSION

Preliminarily, an universal architecture that summarizes current efforts in fault detection was introduced. Temporal dynamics was introduced in this topology, by considering time convolution and linkage of SOFM output trajectories. Thereby this topology becomes a predictive-maintenance oriented neural network system, PREMON. The analogy with the brain is twofold: First, in the initial stage, residual extraction is similar to the initial stage of nervous information processing at the level of the thalamus, which is compared elsewhere[11] to a "folded auto-associative neural network". Second, the linking and classification of the trajectories formed by winning SOFM neurons is similar to the processes taking place at a the different layers of the cortex. The reason for the widely use of SOFM architecture in this work is that the SOFM categorizes the faulty modes strictly according to a distance criteria, without any kind of previous or implicit supposition about the given data.

Finally the Performance evaluation indicated very promising results for detecting faults in a commercial machine system.

VI. REFERENCES

[1] Kovács, Z. L. Redes Neurais Artificiais: Fundamentos e Aplicações. 2ª Ed. São Paulo, Collegium Cognitio, 1996.

[2] Duin R.P.W. Machin Diagnostics by Neural Networks.
http:// ww.stw.nl/projecten/d/dtn3584.uk.html

[3] Leonard, J. A.; Kramer, M. A. Diagnosing Dynamic Faults Using Modular Neural Nets. *IEEE Expert*, **8**(2):44-53, 1993.

[4] Hines, J. W.; Miller, D. W.; Hajek, B. K. A Hybrid Approach For Detecting And Isolating Faults In Nuclear Power Plant Interacting Systems:
. http://www.engr.utk.edu/pub/hines/nuct_11.doc

[5] Kohonen T. The self-organizing Map. Proceedings of the Institute of Electrical and Electronics Engineers. **78**: 1464-1480

[6] Ropero Peláez, J. Several Hypothesis on Temporal Processing in Pyramidal Neuron's Dendrites. In Proceedings of the I Online Workshop on Soft Computing. Pages 119-123, 1996.

[7] Wehr M. &Laurent G.Odour Encoding by Temporal Sequences of Firing in Oscillating Neural Assemblies. *Nature* **384**:162-165

[8] Automation Lab. Robust Residual Generation For Model-Based Fault Diagnosis
http://www.ecs.umass.edu/mie/
labs/danai/prjresid.html

[9] Uhrig, R. E.; Tsoukalas, L. H. Applications Of Neural And Fuzzy Technologies In Soft Computing. In International Workshop On Soft Computing In Industry'96, Muroran, Japan, 1996.

[10] Self-organized formation of topologically correct feature maps. *Biological Cybernetics* **43**: 59-69

[11] Ropero-Pelaez, J. Towards a neural-network therapy for hallucinatory disorders. *Neural Networks.* (2000 Special Issue) **13**(2000): 1047-1061