Hybrid Neuro-Fuzzy System for Control of Complex Plants

B. Bona, S. Carabelli, M. Chiaberge, E. Miranda and L.M. Reyneri
`LIM` - Interdepartmental Laboratory of Mechatronics
Politecnico di Torino
C.so Duca degli Abruzzi, 24 - 10129 TORINO - ITALY
e.mail chiaberge@polito.it, fax: ++39 11 564 4099

Abstract

Artificial Neural Networks (ANNs) and Fuzzy Systems (FS) are high parallel structures that consist of a large number of elementary non-linear units (called neurons) fully interconnected.

In this article we present a hardware implementation of ANNs combined with a DSP resulting in a powerful system used in control applications. The NN processes most of the control tasks, while the DSP performs signal pre-processing and learning algorithms. In some cases the DSP also tracks some discrete states of the plant by implementing a finite state automata and/or verifying plant safety boundaries operations. The tight link with a DSP allows the NN hardware to be very simple since several operations related with the NNs (learning, weights refresh, etc.) can be performed by the DSP.

Moreover the system can implement intelligent control paradigms mixing Neuro-Fuzzy algorithms with finite state automata and/or digital control algorithms.

1 HYBRID CONTROL SYSTEM

Hybrid Neuro-Fuzzy systems have several advantages over both DSP and Neuro-Fuzzy systems working alone; interaction between them allows to implement easily several complex control algorithms [1, 2, 3, 7, 8]. Figure 1 shows a general diagram block of our hybrid system, that allows high flexibility of operations. In this system a Neuro-Fuzzy module is used to perform the plant control based on Neural Network and Fuzzy logic techniques while the DSP handles several other tasks:

- complementary Neuro-Fuzzy operations
  - finite state automata
  - fuzzy state automata [5]
  - on-line learning algorithms for intelligent and adaptive control systems.
- signal processing
  - filtering
  - signal conditioning
- management of overall system
  - user programs computation
  - interrupts handle.
  - host interface

An additional advantage of this hybrid system is its open architecture design that easily allows to add new Neuro-Fuzzy cores or other kinds of subsystems needed to signal processing of plant control; this architecture allows to share inputs between DSP and the Neuro-Fuzzy core, feed Neuro-Fuzzy core with DSP pre-processed inputs or process signals trough some modules under DSP control.

1.1 Prototype board

A general purpose system based on a TMS320C31 Texas DSP is available. The system can support up to 256 channels for A/D and D/A conversion, Neuro-Fuzzy core units, PWM encoders, digital I/O, memory expansions an so on.

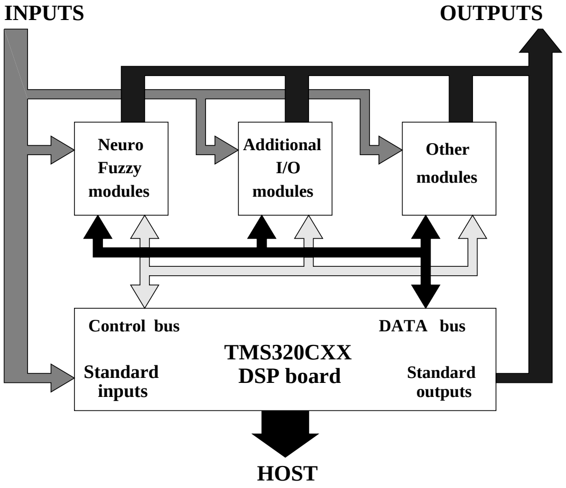

Figure 1: DSP system block diagram: each module is on a different board

The core of the system is a custom TMS320C31 board with up to 8Mbyte of RAM and 128Kbyte of Non-Volatile RAM. This board is interfaced to the additional modules of the system by using a proprietary I/O bus. In this way each module is mapped into the DSP memory space. Special modules provides a communication with a host computer (e.g. a PC, via a parallel port); these modules connect a PC serial or parallel port with the DSP board via the I/O bus.

This system is suitable both for industrial high performance control applications and educational tools depending on the amount and kind of modules plugged into it. Each module can be added with an extremely easy procedure and can be controlled by the software environment.

So far, we have manufactured several modules: the main DSP board, a reconfigurable Neuro-Fuzzy computing core, the host computer interface modules, an 8 channels 12 bit A/D and D/A converters, but the complete system can accept many others as: special-purpose processors, drivers and actuator interfaces, special sensors modules and others.

The Neuro-Fuzzy module is described in section 2, while converters modules are classical Analog to Digital and Digital to Analog converters, and also special converter modules that convert analog and digital signals into Pulse Width Modulated signals and vice-versa, these modules can be configured by the DSP by setting some parameters as: bits number for digital converters and full-scale ranges for the analog ones.

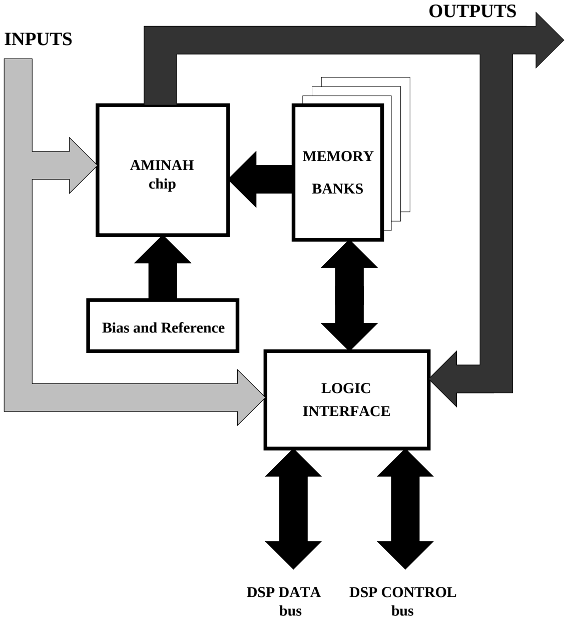

Figure 2: Neuro-Fuzzy core block diagram.

2 NEURO-FUZZY COMPUTING CORE

The proposed Neuro-Fuzzy core implements on hardware several Neuro-Fuzzy paradigms; its operation can be viewed either as a slave operation of the DSP or as an independent one. For its operation, Neuro-Fuzzy core takes as inputs either external signals (through dedicate A/D converters) or signals from the DSP subsystem via the field-bus; and also Neuro-Fuzzy core outputs can be sent either to external outputs or to the DSP.

The Neuro-Fuzzy computing core presented in this section is based on a custom Neuro-Fuzzy processor called AMINAH [9] and implements different paradigms as Multilayer Perceptron (MLP), Linear Networks (LN), Radial basis Functions (RBF) and also Weigthed Radial Basis Functions (WRBF) that are used to implement Fuzzy inference systems [6]. As shown in figure 2, Neuro-Fuzzy processor core has also a local non-volatile memory used to stores several banks of synaptic weights and some logic interface to connect AMINAH and the non-volatile memory to the main system bus; it also has special analog circuits that provide timing, biases and reference signals.

Every synaptic weights bank define a particular Neuro-Fuzzy system that includes synaptic weights and configuration information; only one of this banks can be active at the same time. A copy of active bank is also stored into the internal volatile AMINAH memory and every time that the active bank is changed the new weights and configuration information is automatically loaded into it.

Neuro-Fuzzy core is considered as a slave device the DSP can perform several tasks on it such as:

- DSP can write/read a bank of synaptic weights into/from the Neuro-Fuzzy core non-volatile memory.
- DSP can select a single bank to be the active one. The Neuro-Fuzzy subsystem will then automatically write active bank content into AMINAH.
- DSP can read inputs and outputs of Neuro-Fuzzy subsystem to execute learning algorithms or other supervision tasks.

2.1 Architecture of AMINAH

The AMINAH chip [16] (see figure 3) has two cascaded neural network layers called hidden and output layer respectively. Architecture of one layer of AMINAH, shown in figure 4, is based on an N×M synaptic array and a vector of M neurons. Each neuron j is connected to a row of N synapses plus a threshold ($S_{j,1} \dots S_{j,N}$, $S_{j,\Theta}$), while each column ($S_{1,i} \dots S_{N,i}$, $S_{\Theta,i}$) is connected to the same input X_i. The hidden layer has N=7 inputs and M=16 outputs while the output layer has N=15 inputs and M=8 outputs.

AMINAH output signals are coded using Coherent Pulse Width Modulation (CPWM), a particular Pulse Stream technique [4]. AMINAH inputs signal (X_i) can either be analog or CPWM; in the former case, they are converted using on-chip analog to CPWM converters.

Each AMINAH synapse uses two synaptic weights (called respectively excitatory and inhibitory weights) to compute the synaptic contribution. A copy of active weights is stored internally into AMINAH using two capacitive memories per synapse. The synaptic contribution depends also on the chosen neural paradigm; this can be either Multi-Layer Perceptron (MLP), Radial Basis Function (RBF), or Weighted Radial Basis Function (WRBF) of order 1, and therefore the system can also implement Fuzzy Systems [6].

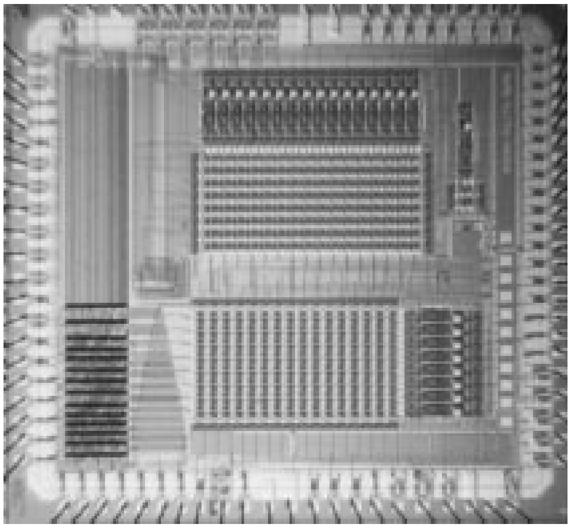

Figure 3: AMINAH microphotograph.

Synaptic contributions are charge quantities. For MLP operation each synapse generate charge quantities proportional to the product of the stored weight and width of the input CPWM signal; this product is quite linear and allows a small power dissipation; while for RBF/WRBF operation each synapse generate charge quantities proportional to the distance between stored weight and width of the input CPWM signal.

For the MLP paradigm, the synaptic contributes are presented as a differential current (I^{+}_{MLP}, I^{-}_{MLP}), while for the RBF/WRBF paradigms, they are presented as a single-ended current I^{+}_{MLP}, while I^{-}_{MLP} is not used. Each row j of synapses (Fig. 4) sums up all the currents on the two common summation lines $I^{+}_{\mathrm{MLP_j}}$ and $I^{-}_{\mathrm{MLP_j}}$ (excitatory and inhibitory respectively).

The neuron (one for each row) generate a current that is the algebraic sum of them using programmable-gain current mirrors multiplying it by a chosen factor $R_{\mathrm{I}} < 1$; the neuron integrates the resultant current on capaci-

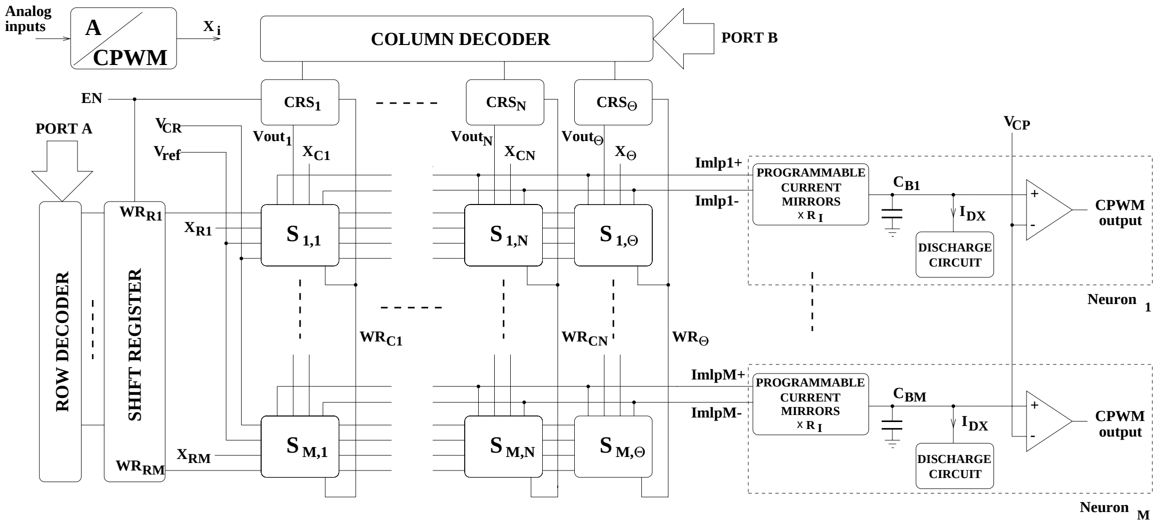

Figure 4: One-layer Neuro-Fuzzy Core diagram.

tor C_{Bj} to obtain a voltage proportional to the neuron activity. This voltage is then converted to a CPWM output signal using a non-linear conversion (by using the on-chip nonlinear converters).

Since AMINAH internal weights are stored in analog form, each column i of synapses is connected to a Common Refresh System (CRS) used for the synaptic weights long term storage to compensate leakage currents and therefore weights decay.

The most important characteristics of AMINAH [9] are:

- the chip is designed to accept any configuration of one or two layers Neural Network. If a multi-layer Neural Network (with more than two layers) is needed, a multi-chip structure can be used (every chip implements two neurons layers) or one chip multiplexing in time different weights matrices to implement more than two layers (slow operation).
- implements neural networks with at most 7 inputs, 15 hidden neurons and 8 output neurons.
- allows to work with MLP/(W)RBF algorithms and can be programmed to emulate Fuzzy systems.
- accepts two different external signals that are used to define non-linear functions for neuron computations. Typically these signals are sigmoids and exponential functions. Each neuron can be programmed to select one of these signals.
- allows to change the number of active synapses connected to a neuron by selecting one of seven current mirrors; each mirror has a fixed gain (2, 1, $\frac{1}{2}$, $\cdots$, $\frac{1}{32}$) and can be turned on/off writing a configuration register in each neuron.

3 `ONDE`: Object-oriented Neuro-Fuzzy Development Environment

`ONDE` is an Object Oriented development environment for DSPs (and also other types of microcontrollers), by which it can be possible to design, simulate and develop control systems based on DSPs.

`ONDE` will be seen as an extension to the C++ programming language, by means of predefined classes oriented to the specification of Neuro-Fuzzy systems. Each class corresponds to a specific Neuro-Fuzzy entity (such as neurons, synapses, Fuzzy rules, implicators, etc.). Compilation through a commercial C++ compiler will provide an executable code running on the desired platform (e.g. a DSP).

The `ONDE` package is tightly integrated in the $MATLAB^{\copyright}$ environment, making also available to the designer all the capabilities of $MATLAB^{©}$.

From the $MATLAB^{©}$ user's point of view, `ONDE` is like a new toolbox that allows to:

- acquire data (with or without a control algorithm running) directly into the $MATLAB^{©}$ workspace; display acquired data in real time in the time or frequency domains (a virtual oscilloscope or spectrum analyzer);
- use acquired data for real-time model identification (in open as well as in closed loop configuration);
- run a number of fixed structure controllers (PID, IIR, RTS, state feedback etc.);
- watch and modify at run time any signal or parameter of the running controller without stopping its execution.

For instance, `ONDE` may be used first to acquire data and process both in the frequency as well as in the time domain for plant or parameter identification, then to rapidly test a suitable designed filter or controller. The capacity to keep on working within the same environment greatly helps in the unavoidable cycling through the aforementioned tasks.

Moreover `ONDE` is a complete development system that extend the toolbox contents or even create new toolboxes of DSP related commands for specific subjects. For instance, `ONDE` may be used to build a high level interface for teaching as well as research laboratory experiments. Once used to prototype specific solutions, `ONDE` kernel may be used to allow a dedicated software interface to interact with the DSP hardware without modifying the DSP code. In this sense, the $MATLAB^{©}$ computing environment may be seen as the interface for the first step in the development of a virtual instrument and as its debugging and maintenance tool.

4 APPLICATIONS EXAMPLES

This section describes two applications of the proposed methodology.

4.1 Exapode walking machine

Aim of the research is to design, build and test a small hexapode walking machine controlled by purposely developed neural chips, which can be used as a remotely controlled observation device and can evolve as a true mobile robot (see figure 5). The machine can also be used as an experimental test-bed for different control strategies to investigate the possibility of small walking vehicles on different types of terrain.

The main characteristics of the machine are: simple and lightweight mechanical architecture to contain overall costs, modularity to use the machine as a research tool, flexibility leading to the possibility of adapting the gait to a variety of terrains, possibility of working as an autonomous system without the need of an umbilical cord for either energy supply or control.

For these reasons, we chose to use low-cost permanent magnets electric motors in connection with re-chargeable batteries and to adopt a gait in which the vehicle is always in conditions of static equilibrium. A "reptilian" stance is adopted as its energetic disadvantages are of small importance. The design chosen allows to assume an "insect" stance and even to switch to a quadruped "mammalian" configuration [11]. The mass of the mechanical components, including the electric motors, is 24.3 Kg. The machine is already assembled and the first tests are going on.

Each leg is made of a shinbone and a thighbone both 50 cm long; it has three degrees of freedom and is equipped with three current-controlled motors. To obtain high efficiency, switching amplifiers are employed which are perfectly matched with the CPWM encoding used in the Neuro-Fuzzy core (see section 2). All the actuators are directly controlled by power drivers. One of the actuators must carry a significant fraction of the hexapode weight (depending on the number of legs that support the body) and so its driver has a maximum output current of 20A (namely, 480W), while for the other two drivers the output current is only 3A (namely, 72W).

Since the control task is complex, the idea is to build a hierarchy of Neuro-Fuzzy controllers. Neuro-Fuzzy controllers have been chosen because of their good behavior in the presence of

Figure 5: The skeleton of the Hexapode Walking Machine.

non-linear systems and for their intrinsic generalization capability. The control system is organized in three different levels:

- Motion coordination control. It acts like a "central" controller, which gives the legs the right "high-level" control signals (such as: robot speed, robot height from ground, radius of trajectory) to let the hexapode execute properly each gait. Obstacle avoidance is achieved by giving proper trajectory information to the individual leg controls. Only large obstacles are avoided, as small ones (namely roughness of the ground, small stones, holes, etc.) are handled by each leg control.
- Leg controls. They are like "local" controls that defines the trajectory and the sequence of movements of each leg. They also recover and modify leg trajectories when small local obstacles are encountered.
- Joint position controls. They receive the angular positions and translate them into control signals for positioning motors.

Locomotion control is distributed evenly among the six legs. Independently from the different gait, each leg cycles over six main different states:

- Power: the leg leans on the ground where it supports and propels the body, moving backward to the posterior extreme position.
- Lift: the leg rises from the posterior extreme position and loses its support function. Lift phase ends when the leg is high enough to swing forward.
- Return: leg swings forward to the anterior extreme position. As soon as it reaches this point it is ready for the next phase.
- Contact: leg lowers down to the ground. During this phase it starts again to support the body weight.
- Upstep: the leg rises when it hits an obstacle (obstacle avoidance). When the height is high enough, the leg returns in the "Return" state.
- Backstep: the leg recover the right position after an obstacle avoidance or if it is in time out.

The use of Neuro-Fuzzy approach (Fuzzy-State Automata [17], in particular) allows to "smooth" the leg movements during the walk (changes of speed and of direction without slipping of the feet on the ground [11]).

The local leg controllers run simultaneously; however they are not independent of each other. To let the robot move and deal with obstacles, an inter-leg coordination is needed. For instance, when a leg is trying to step over an obstacle, it can ask the supporting legs to momentarily stop, to raise the body, or to move the robot backwards, when the obstacle it too large to step over. All these coordination functions are easily handled by the interacting Fuzzy-State Automata associated with each leg.

4.2 Rotors on active magnetic bearings

Active magnetic bearings are a well known technology in the field of high speed rotating machinery. Magnetic bearings exibit several appealing feature: very low friction, no contact between rotor and stator parts, and no need for lubricants. Turbomolecular pumps used to make the high vacuum required in the semiconductor manifatturing, maintenace free canned pumps and compressors are nowadays almost exclusively built on magnetic bearings.

In addition, active magnetic bearings are an effective means to implement a controlled suspension of a rotor. Well known rotordynamic phenomena such as critical speed crossing during rotor acceleration, bending mode damping, and unbalance compensation may take substantial advantage from the dynamic control exerted by the magnetic suspension. The Neuro-Fuzzy implementation of the bearing control allows several additional opportunities to implement advanced techniques such as adaptive and/or self-tuning control strategies.

The availability of a Neuro-Fuzzy controller is very promising in the automatic implementation of complementary and nowadays off-line tasks such as magnetic parameters identification, unbalance degree evaluatio for on-line compensation, and failure detection.

The proposed hybrid neuro-fuzzy solution is to be tested on a number of test rigs built at the Mechatronics Lab [12], [13] possible to be transferred to industrial [14], [15] prototypes.

5 CONCLUSION

This paper has presented a real-time implementation of a Neuro-Fuzzy approach to the control of complex system. Several paradigms have been integrated, such as Neural Networks, Fuzzy systems, linear controllers, DSPs and Finite and Fuzzy State Automata.

The first real applications have demonstrated that the proposed approach is very promising and can be adapted to a wide variety of different applications such as, among others, robotics, servo-controls and machines with moving parts with backlash and low stiffness.

ACKNOLEDGMENTS

This work has been supported by the Italian National Research Council project *"Architetture parallele e algoritmi per reti neuronali e loro applicazioni* and the ASI Italian Space Agency contract *Neural Network Architectures for Control of Flexible Arms.*

References

[1] D.A. White and D.A. Sofge, "Handbook of Intelligent Control", *Van Nostrand Reinhold*, New York, 1992.

[2] L. Wang, "Adaptive Fuzzy Systems and Control", *Prentice Hall*, Englewood Cliffs, New Jersey, 1994.

[3] S. Haykin, "Neural Networks: A Comprehensive Foundation", *Mc Millan College Publishing Company*, New York, 1994.

[4] L.M. Reyneri, M. Chiaberge, "CINTIA: A Neuro-Fuzzy Real Time Controller for Low Power Embedded Systems", in *IEEE MICRO*, June 1995, pp. 40-47.

[5] L.M. Reyneri, "Unification of Neural and Fuzzy Computing Paradigms", in *Proc. of AT'96, 1-st Int'l Symposium on Neuro-Fuzzy Systems*, Lausanne, August 1996, sect. 5.

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

[7] C.C. Lee, "Fuzzy Logic in Control Systems: Fuzzy Logic Controller", *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 20, no. 2, March/April 1990, pp. 404-435.

[8] G.V.S. Raju, J. Zhou, R.A. Kisner, "Hierarchical Fuzzy Control", in Int. J. Control, Vol. 54, no. 5, 1991, pp. 1201-1216.

[9] M. Chiaberge, E. Miranda Sologuren, L.M. Reyneri, "A Pulse Stream System for Low Power Neuro-Fuzzy Computation", *IEEE Transactions on Circuits and Systems*, vol. 42 no. 11, pp. 946-954.

[10] W.T. Miller, R.S. Sutton, "Neural Networks for Control", *P.J. Werbos ed.*, 1990.

[11] D. Bassani, M. Chiaberge, D. Del Corso, G. Genta, F. Zanetti, "Simulation of an Hexapode Walking Machine Controlled by Neural Network", *Proc. of ISMCR 93*, Sept. 1993, Turin (I), pp. As.II 27-32.

[12] C. Delprete, S. Carabelli, and G. Genta, "Design, construction, and testing of a five axis magnetic bearing system," in *Proc. of Int. Symp. on Magnetic Suspension Technology*, (Seattle, USA), pp. 147–161, NASA, August 1993.

[13] S. Carabelli, C. Delprete, G. Genta, and I. Moretto, "A spindle running on a five axes magnetic suspension based on conical bearings," in *Proc. of 4th Industrial Conf. on Application of Magnetic Bearings*, (Alexandria, VA, USA), August 1995.

[14] L. Mazzocchetti, E. Rava, C. Delprete, G. Genta, and S. Carabelli, "Design, construction and testing of a turbomolecular pump with five axes magnetic suspension," in *Proc. of the 4th Int. Symp. on Magnetic Bearings*, (Zurich, CH), pp. 505–510, August 1994.

[15] A. Argondizza, S. Carabelli, C. Delprete, and G. Genta, "Dry chlorine centrifugal compressor on active magnetic bearings." Accepted to 7th Int. Symp. on Transport Phenomena and Dynamics of Rotating Machinery, February 1998.

[16] M. Chiaberge, E. Miranda Sologuren, L.M. Reyneri, "A Pulse Stream System for Low Power Neuro-Fuzzy Computation", on *IEEE Trans. on Circuits and Systems - I*, Vol. 42, no. 11, November 1995, pp. 946-954.

[17] L.M. Reyneri, "An Introduction to Fuzzy State Automata", in *Proc. of IWANN 97*, Lanzarote (E), June 1997, pp. 273-283.