Brain Inspired Cognitive Systems - BICS2004:
August 29 – September 1, 2004. University of Stirling, Scotland, UK

Reference ID

A NEURO-FUZZY ARCHITECTURE FOR IMPROVING THE PERFORMANCE OF A CLASSICAL CONTROLLER

G. N. Marichal/ Department of Applied Physics, University of La Laguna La Laguna 38271. Tenerife. Spain
Tel:+34 922 318329/Fax: +34 922 318288 /Email: nico@cyc.ull.es

J. Toledo/ Department of Applied Physics, University of La Laguna La Laguna 38271. Tenerife. Spain
Tel:+34 922 318287/Fax: +34 922 318288 /Email: jonay@cyc.ull.es

L. Acosta/ Department of Applied Physics, University of La Laguna La Laguna 38271. Tenerife. Spain
Tel:+34 922 318329/Fax: +34 922 318288 /Email: lacosta@ull.es

L. Moreno/ Department of Applied Physics, University of La Laguna La Laguna 38271. Tenerife. Spain
Tel:+34 922 318329/Fax: +34 922 318288 /Email: lmoreno@ull.es

ABSTRACT

In this paper a control strategy based on a neuro-fuzzy systems have been devised. Two control system have been considered in order to test the method. Satisfactory results have been obtained in both cases.

INTRODUCTION

New Intelligent strategies have been applied in a wide range of problems [1][2]. The Fuzzy Logic techniques have been used successfully in any of them. However, in some cases it is difficult to design the most adequate Fuzzy Rules in order to solve the underlying problem.

On the other hand, Neural Networks have also been applied with a good performance [2]. Neural Networks have shown good learning properties which make them very appropriate in some tasks where a automatic adjustment is necessary. On the contrary, it is difficult to know what they are doing, that is, they are seen as black boxes which make well its work in most of cases. In spite of some studies have been done in order to find out more about these systems, most cases it is difficult to predict the number of neurons which will reach the best solution for a particular problem. On the contrary, the fuzzy logic systems are easier to interpret. The Neuro-Fuzzy systems are a form of unifying the two strategies in order to take benefit of their advantages and to overcome their disadvantages [3]. This new strategy have been applied in some control problems [4][5]. In this paper the presented Neuro-Fuzzy system will be applied to some control systems. In section 2 a description of the Neuro-Fuzzy System along with the learning algorithm is shown. Section 3 will be focused on the application of the Neuro-Fuzzy system in order to improve the behavior of a classical controller in two particular cases.

2. NEURO-FUZZY APPROACH.

In the subsequent sections, the structure along with the learning algorithm of the used Neuro-Fuzzy system is presented.

2.1 Structure of the Neuro-Fuzzy System.

The Neuro-Fuzzy System could be seen as a three-layer Network. The nodes of the network are cascaded together in layers. A diagram of the Neuro-Fuzzy System is shown in Figure 1.

The first layer or input layer comprises several nodes, each one consisting of a Radial Basis neuron. The inputs to the radial basis neuron are the inputs to the Neuro-Fuzzy System, and the outputs of the nodes are as follows:

$$p_{ij} = \exp\left(-\frac{\left(U_i - m_{ij}\right)^2}{\sigma_{ij}^{2}}\right) \qquad \begin{array}{l} j = 1,2,\ldots,N2 \\ i = 1,2,\ldots,N1 \end{array} \quad (1)$$

Where:

m_{ij} = Centre of the membership function corresponding to ith input and the jth neuron of the hidden layer.

U_i = ith Input to the Neuro-Fuzzy System.

σ_{ij} = Width of the membership function corresponding to the ith input and the jth neuron of the hidden layer.

p_{ij} = Output of the Radial Basis neuron (or degree of membership for ith input corresponding to jth neuron).

N2 = number of nodes at the hidden layer.

N1 = number of Neuro-Fuzzy System inputs.

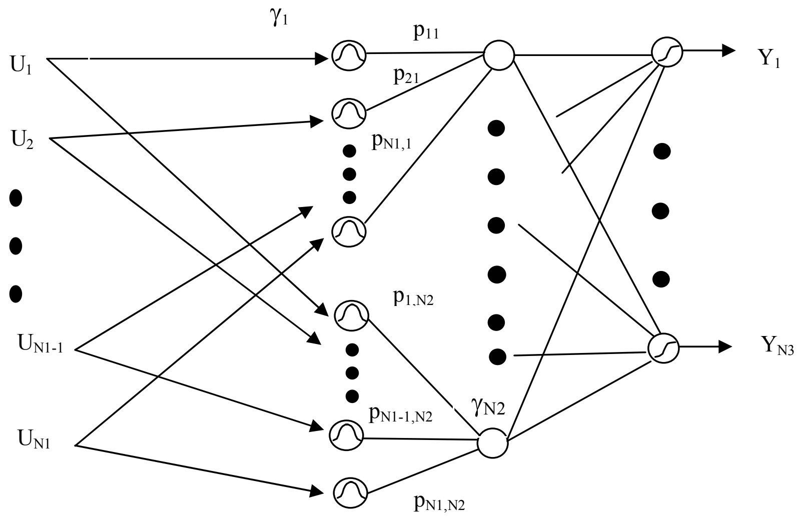

Figure 1- Diagram of the Neuro-Fuzzy System

On the other hand, the node outputs corresponding to the hidden layer are calculated as:

$$\gamma_j = min\lfloor p_{1j}, p_{2j}, \ldots, p_{ij}, \ldots, p_{N1j} \rfloor \quad j = 1,2,\ldots,N2 \quad (2)$$

Where:

γ_j = Output of the jth node at the hidden layer.

Finally, the output layer could be considered as a sigmoid neuron layer, where the weight connections between the hidden layer and the output layer are the estimated values of the outputs. The outputs of these nodes are calculated by this expression:

$$Y_k = Sigmoid\left(\beta \; \frac{\sum_j sv_{jk}\,\gamma_j}{\sum_j \gamma_j}\right) \qquad \begin{matrix} \mathrm{j} = 1,2,\ldots,\mathrm{N2} \\ \mathrm{k} = 1,2,\ldots,\mathrm{N3} \end{matrix} \quad (3)$$

Where:

Y_k = kth output of the Neuro-Fuzzy System.

sv_{jk} = Estimated value of the kth output provided by jth node at the hidden layer.

N3= number of outputs of the Neuro-Fuzzy System.

β = Gain of the sigmoid function.

Hence, it could be said that the output layer carries out the non-linear defuzzification process, providing the outputs of the Neuro-Fuzzy System.

To sum up, the structure of the Neuro-Fuzzy system could be seen as a typical Radial Basis Network, where an additional layer has been inserted between the Radial Basis layer (the input layer) and the sigmoid neuron layer (the output layer). The neurons of this additional layer calculate the degrees of membership corresponding to the different rules, that is, they apply the fuzzy operator min, being N2 the total number of Fuzzy rules. Once, these calculations have been carried out the output layer applies a defuzzification process in order to obtain numeric values for the outputs of the system. Note as this defuzzification process depends on the gain of the sigmoid function. It is possible to change the non-linear defuzzification process behavior depending on the choice of β, given the gain of the sigmoid function determines the steepness of the transition region.

In the Neuro-Fuzzy System there are some parameters which determine the relation between the inputs and outputs. In this case, the behaviour of the Neuro-Fuzzy system depends on the value of the following parameters: the membership function centres, the widths of the membership functions and the estimated output values. In order to determine these parameters a learning algorithm has to be designed. In the future, the gain of the sigmoid function will be considered as a constant.

2.2) Learning Algorithm.

The learning algorithm can be divided into three phases. The first phase and second phase are focused on obtaining initial values for some of the parameters and optimising the number of nodes at the hidden layer, that is, the number of Fuzzy rules. Whereas the last phase is focused on adjusting the different parameters of the Neuro-Fuzzy System taking as initial values the given ones in the previous phases.

First phase consists of obtaining convenient values for the membership function centres and estimated output values. These values are adapted in the third phase. In the first phase only initial values are calculated. In order to obtain these initial values a Kohonen's Self-organising feature map algorithm has been applied. The initial weight vector of the self-organising map has been chosen over a range of random numbers, being the dimension of the weight vector equal to the number of inputs (N1) plus the number of outputs (N3). A 1-D self-organising map has been used. After, an unsupervised learning algorithm has been applied, initial values for the centres of membership functions and for the estimated outputs are obtained. In the following step an optimisation process is applied in order to obtain a minimum number of rules. This second phase is based on evaluating the Euclidean distances between node weights corresponding to the 1-D self-organising map. The nodes which Euclidean distances are less than a constant value will be mixed in one node. Hence, the hidden layer of the Neuro-Fuzzy systems will be reduced given each node of the 1-D self-organising map corresponds to one node of the hidden layer. Furthermore, each node of the hidden layer corresponds to a Fuzzy rules. Hence, the Neuro-Fuzzy system will have less rules as a result of this second phase.

On the contrary, Third phase is focused on adjusting the parameters of the Neuro-Fuzzy system and improving the given ones in the previous phases. Taking into account the similarities between the typical Radial Basis Network and the Neuro-Fuzzy system, the least mean squared learning algorithm has been used as usual in a typical Radial Basis Network.

The purpose of this algorithm is to minimise the following sum-squared criterion function:

$$E = \frac{1}{2}\sum_{k=1}^{N3}\left(Y_k - \hat{y}_k\right)^2 \quad k = 1, \ldots, N3 \quad (4)$$

Where:

$\hat{y}_k$ = kth desired output.

Y_k = kth Neuro-Fuzzy system output

4. APPLICATION.

Such as it was shown in the previous section, a training process is necessary to choose conveniently the Neuro-Fuzzy system parameters. This process is carried out in an off-line mode and it is based on the experience of an expert.

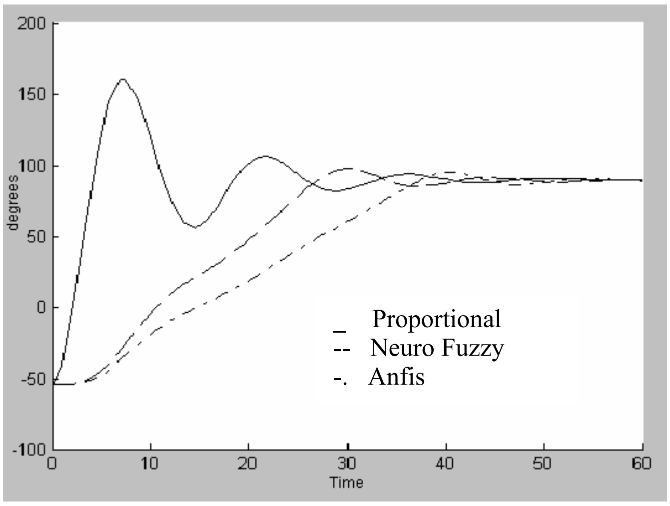

Figure 2.- Comparation beetwen Neuro-Fuzzy, ANFIS and proportional control

In this paper the Neuro-Fuzzy system will be applied to improve a motor control system. Only control systems where a classical controller is used will be considered. Moreover, a proportional controller will be considered for the sake of simplicity. The purpose of the Neuro-Fuzzy system will be to improve the behavior of the controller without modifying its structure. Because of that, each movement between an initial angular position and a determined final angular position will be divided into so steps as the electronic devices allow. Therefore, as setpoints as steps will be presented to the classical controller. Consequently, the Neuro-Fuzzy system has to be able to give the appropriate period of times where these intermediate setpoints are applied.

Using these pairs (time, position) the behaviour of the system is set. Two parameters are used to do this. First one is the time period the motor get the first position after initial position of the particular movement and the second one is an incremental interval of time where the motor gets the next positions. So the motor moves slower when it is reaching the final position and in this way the performance of the proportional action could be improved.

The expert has to provide the right timing for the different movements. These data are provided in a direct way, that is, a software application guides him in the whole process. Because of that, it is not necessary any previous knowledge about the applied strategies. This makes easier the definition of the most adequate fuzzy rules.

The Neuro-Fuzzy system is built up of two inputs and two outputs. The inputs are the current angular position and the desired angular position, respectively. The first output refers to the initial period of time while the first setpoint is being applied. The second output refers to the time frame each of the other setpoints up to the final setpoint are applied. Note as, the period of time while first setpoint is being applied has been considered different than the other setpoints. It has considered in this way because it is necessary to apply the setpoint in a greater frame time in the initialization of the movement. In this way the motor is able to acquire an adequate speed, overall taking into account that most times the motors start from the rest. In order to apply the Neuro-Fuzzy strategy it is necessary to generate a set of inputs and desired outputs for a particular Control System. Each training pattern consists of four values. First one is the current angular position expressed by degree units, however, the second one refers to the desired final angular position for that movement. Third value refers to the initial period while first setpoint is being applied. At last, fourth value refers to the period of time the other setpoints are applied up to the desired final setpoint. These parameters are provided by the expert. Because of that, a simple software has been developed in order the expert to see the effects produced by a particular choice of the parameters.

In summary, different movements are provided to the subsequent training process. In this way, the resultant Neuro-Fuzzy system will be able to show a similar behaviour to an expert would like.

In order to test the proposed strategy two particular control systems have been considered. First one corresponds to a

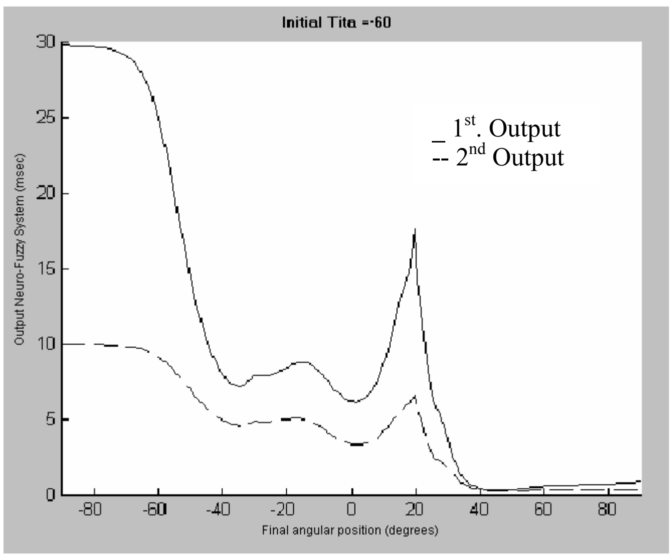

Figure 3.- Neuro-Fuzzy outputs vs. final angular position for an initial angular position of 60º.

binocular head where the motors are in charge of moving the mechanical structure. In this kind of systems sometimes it is a requirement of the problem to use small motors with simple controllers in order to reduce cost and weight of the whole system.

On the other hand, a two-link robot have been used in order to improve the performance of a classical controller (Proportional o Proportional Integral Controller). Because of the nonlinear nature of the system the proposed control strategy can improved its typical behavior.

Several trials have been carried out. In order to compare with other Neuro-Fuzzy Systems an implementation of a standard ANFIS System has been used. Concretely, the ANFIS System implemented by Matlab has been used [6]. In the following, it will be referred as ANFIS. The difference between the typical proportional action, the Neuro-Fuzzy actions and ANFIS action corresponding to one of the trials is shown in Figure 2. In this case, the training process has been started with 120 fuzzy rules. The Neuro-Fuzzy strategy (dash line) and ANFIS strategy (dot line) produces a better movement than the proportional action (solid line), arriving at the final position more or less at the same time with a better behaviour. At the end of the training process the number of fuzzy rules have been reduced to 60. It Should be considered that the ANFIS strategy fixs the number of rules, and the training is based on these rules, so an appropriate election of the number of rules it is critical for the performance of the system. Note as, the Neuro-Fuzzy system shown in this paper includes a nonlinear defuzzyfication, that is able to improve the capacity of the network generalization. It is important to remark that the results shown in Figure 2 correspond to the Neuro-Fuzzy and ANFIS system after the training process has concluded. In Figure 2 only the results corresponding to one degree of freedom have been shown, a similar process have been followed for the other degrees of freedom.

In the case of the binocular head the outputs of the Neuro-Fuzzy system versus different final angular positions starting at an initial angular position of 60 degrees are presented in Figure 3. It can be seen that the motor moves slower for positions near to zero degrees, because the motor of the binocular head finds more difficult to set this position. It is because the used electronic devices are more damage when the binocular head is moving around zero degrees. This fact shows that the proposed strategy adapts automatically to the underlying particular conditions. In the movements near to the initial position (60 degrees) it is necessary to do slow movements, and when the system is far away of the final position fast movements are necessary instead. Note that the Neuro-Fuzzy system infers this behavior from the data provided by the expert.

Other trials have been done with a two-link robot based on the same control strategy, getting a satisfactory for the control of the different links.

Taking into account the obtained results it could be said that the best performance of the Neuro-Fuzzy Strategy will be associated with links, where inertial effects are considerable. However, in the cases where great inertial effects are not present, the Neuro-Fuzzy strategy still keeps a good performance.

4. CONCLUSIONS.

In this paper, a Neuro-Fuzzy system has been used to improve the behavior of a classical controllers. The devised control strategy has been applied to two control systems obtaining satisfactory results. In addition to the proposed Neuro-Fuzzy System , the standard ANFIS Neuro-Fuzzy system have also been used. As a result of the application of the strategy, it can be seen that the proposed control strategy is able to adapt itself to the particular conditions of the underlying system.

ACKNOWLEDGMENTS

This research has been supported by the Spanish Government under the Project DPI2001-3681.

REFERENCES

[1] C.H. Chen, “Fuzzy Logic and Neural Networks Handbook”, Mc. Graw-Hill. 1996

[2] Kosko B. “Neural Networks and Fuzzy Systems: A Dynamical Systems Approach to Machine Intelligence”. Prentice-Hall.1992

[3] D. Nauck, F. Klawonn, R. Kruse (1993): Combining Neural Networks and Fuzzy Controllers. In: E.P. Klement and W. Slany (Eds.): Fuzzy Logic in Artificial Intelligence (Proc. FLAI’93), Springer, Berlin, 35-46

[4] Miller W., III, R. Sutton and P. Werbos, eds. (1990). Neural Networks for Control. The Mit Press, Cambridge

[5]L. Acosta, G. N. Marichal, L. Moreno, J.J. Rodrigo, A. Hamilton, J.A. Méndez: A Robotic system base don neural networks controllers, Artificial Intelligence in Engineering, (1999), Vol. 13, 393-398.

[6]. Fuzzy Logic Toolbox User’s Guide. (2002). The MathWorks, Inc.

[7] W. Pedrycz (1993): Fuzzy Control and Fuzzy Systems, 2. ed., John Wiley & Sons, New York.

[8] C. T. Lin and C.S. Lee (1994): Supervised and unsupervised learning with fuzzy similarity for neural network-based fuzzy logic control systems. In: R.R. Yager and L.A. Zadeh (Eds.): Fuzzy Sets, Neural Networks and Soft Computing. Van Nostrand Reinold, New York, 126-165.