A DISO NEURO-FUZZY CONTROL FOR DC SERVODRIVES.

Dan MIHAI

University of Craiova, Department of Engineering in Electromechanics, Environment and Industrial Informatics, email: dmihai@em.ucv.ro

Abstract **– The paper follows a previous work that made the synthesis of a neuro-fuzzy controller with double input and single output for positioning systems with DC motors. Now, the new system is put into operation using models and simulations for proving its qualities. Briefly, several test will certify that this single neuro-fuzzy controller (for both the position and the speed) is able to ensure good or very good performance in steady-state and dynamic regimes for a wide variety of conditions concerning: the motor parameters, the load characteristics (torque, disturbances, inertia), the reference cycle (structure, values, duration) and the sampling time. The global results for both the control structures (standard – that generated the training data and fuzzy) are compared. The next stage is to prepare the tools for an on-line implementation. Several formulas will be suggested for the real-time control using 2D and 3D look-up-tables by direct search and by interpolation. The author sustains an on-line timing based on imbricated hardware / software interrupts, that will be implemented using a low-cost microcontroller. The experimental platform is presented as well as some real-time results for the main variables of the system and by recording the on-line timing with a logic analyzer. Most of the ideas, formulas and implementation procedures presented here are useful for other systems and applications.**

Keywords**:** *DC drive, neuro-fuzzy control, LUT, real-time.*

1. INTRODUCTION

More and more applications put into operations fuzzy logic controllers (FLC) because their fast design and a robust behavior of the system. The comparisons with the conventional control algorithms like PID techniques – [1] or the sliding mode control – [4] sustain the fuzzy approach at least by analyzing the results (an analytical proof is not a strong point for the fuzzy control). Comparative study of conventional, fuzzy and neuro-fuzzy controller is made also by [10]. Sometimes, the results are quite close in term of performance but the implementation effort is much lower for the fuzzy solution – [18]. The success associated to the systems equipped with FLC is proved by many applications – [8], [18]. Although some papers still sustain the natural idea that a sampled-data fuzzy controller recovers the performance of the continuous-time fuzzy controller as the sampling period approaches zero – [5], several authors have noticed that the fuzzy control is flexible and reliable for a low rate control sampling and the author proved that an adaptive sampling frequency is feasible for FLC - [12], [13]. Lately, more and more a hybrid approach is preferred, combining fuzzy logic with other techniques from the area of soft-computing control. A very comprehensive study with a theoretical background and a strong support by the inserted results is made in [15], where the performance of the adaptive fuzzy controllers has been compared with that of several other techniques. [9] develops a design for a self tuning fuzzy logic controller, based on some desired output behavior and hence, not requiring a precise model of the machine. Most of the hybrid solutions concern the neuro-fuzzy approach, as the most natural aggregation of the best gains brought by artificial neural networks and by the fuzzy logic. Many applications concern the electrical drives field – [3]. [7] presents a neuro-fuzzy adaptive system used for generate a fuzzy model for a DC motor. In order to avoid the difficulties that overcome in the range of the low speeds, in [6] a neuro-fuzzy speed controller is proposed which has as a feedback not only the encoder information, but the measured armature current and voltage. The standard control tools become unuseful when the system is subjected to unexpected disturbances, the saturation is taken into account or the armature reaction is considered. All these drawbacks are avoided by means of a self tuning fuzzy logic controller – [9]. Some specific conditions for the embedded systems, like limited supply voltage, PWM control signals and a low sampling rate are the aim of the study from [11]. [16] suggests a new fuzzy control solution with 2-DOF PI-fuzzy controllers dedicated to a class of servo systems. The controlled plants in these systems, widely used in mechatronics applications, can be characterized by second-order dynamics with integral type. Less papers present details about the real-time implementations. [19] has such approach but the attention was focused only on the velocity control. The author is submitting now a neuro-fuzzy control for both the position and the speed of a DC servodrive, using a fuzzy inference system (FIS) previously designed – [14]. Several support tools will be put into operation for implementing such solution in real-time. For a low-cost target microcontroller unit (MCU), the variant of using a look-up-table (LUT) was considered. The data operating in real-time are extracted from the neuro-fuzzy controller previously designed. Formulas will be suggested for this purpose.

2. THE NEURO-FUZZY MODEL AND THE SIMULATION RESULTS

A first step is to certify by simulation that the obtained neuro-fuzzy controller (NFC) could bring the expected benefits. The model from the fig. 1 underlines the structure simplification related with the initial conventional structure used for the training process of the NFC – [14]. The block I/O N-F control was kept in order to compare the I/O variable evolution with the initial standard model. A 2[nd] order model in considered for the servomotor and the Load torque block is able to compose a complex load torque (including the friction component). All the parameters are taken from the target system that will be operated in real-time. The low inertia motor has a big impulsion current (and torque), so the evolution of the current is not controlled in this model. For the experimental stage, are available protection procedures in hardware and software. The results from the fig. 2 are for the initial training conditions and they are very closed with those for the conventional structure – that means the NFC was well trained. The same remark concerns the I/O variables: the position error - $\varepsilon_{\alpha k}$:, the position error change during the sampling time T - $\Delta\varepsilon_{\alpha k}$ and the control c_k. These results have, however, a theoretical meaning because the lack of limitations concerning the control range. It is important to point out that the initial conventional structure fails in doing the right positioning when the system becomes non-linear because of a saturation imposed to the power supply. The NFC is able to ensure a good control even in these conditions. The fig. 3 proves much more, because the simulation conditions are very different to those from the initial model considered for the training process:

- a strong limitation of the control voltage to the rated value of the motor;
- different motor parameters (resistance increased by 25 %);
- different load conditions (doubled the inertia, other load torque);
- consideration of a random disturbance affecting the external load;
- different positioning cycle as reference values (250 % bigger) and time length of the segments.

The main effects of the saturation for the control are the suppressing of the position overshoots (highly desirable) and the extension of the dynamic regimes (not so favorable). The most important merit factor is that the position error is made, finally, null.

3. THE DATA PROCESSING FOR THE REAL-TIME CONTROL

A normalized LUT is filled with the values generated by exploring the Rule viewer of the FIS generated with ANFIS. Only the useful data are considered. The fig. 4 shows the systemic structure and the variables involved for the LUT variant control. The control c_k activates the digital power supply (DPS) and ΔN_k means the encoder pulses counted during a sampling period T. The speed is computed from this variable. For a 2D LUT, the data arrangement (column concatenation) is suggested by the fig. 5. Be n_v the memory cells number for a value. For the experimental point (x*,y*), the on-line program must find the cell of the control c*=f(x*,y*). When n_v =1:

$$j = 1 + \frac{x^* - x_m}{r_x} \; ; \; i = 1 + \frac{y^* - y_m}{r_y} \tag{1}$$

By an appropriate correlation (N_C, N_L), i and j must be integers. IF $A_{B\,LUT}$ is the beginning address for the LUT, the position of the cell for c_k is L(i,j):

$$L(i,j) = (i-1) + (j-1) \cdot N_L + A_{BLUT} \tag{2}$$

$$L(x^*, y^*) = \frac{y^* \text{-} y_m}{r_y} + \frac{x^* \text{-} x_m}{r_x} \cdot N_L \; + A_{BLUT} \tag{3}$$

$$L(x^*, y^*) = A_{BLUT} + Displ\,(x^*, y^*) \tag{4}$$

Displ.= displacement related to the basic address. For a general case:

$$L(x^*, y^*) = A_{BLUT} + n_v \cdot Displ\,(x^*, y^*) \tag{5}$$

For multibyte controls, the signs are stored in a distinct area and the location for a specific sign is found with:

$$L_{sign}(x^*, y^*) = L_{control}(x^*, y^*) + N_C \cdot N_L \tag{6}$$

For a framing:

$$x_j \le x^* \le x_{j+1} \; ; \; y_i \le y^* \le y_{i+1} \tag{7}$$

an interpolation is necessary – fig. 6:

$$X(x^*, y^*) = A + \frac{B-A}{x_{j+1} - x_j} \cdot (x^* - x_j) + \frac{D-A}{y_{i+1} - y_i} \cdot (y^* - y_i) + (A + C - B - D) \cdot \frac{(x^* - x_j) \cdot (y^* - y_i)}{(x_{j+1} - x_j) \cdot (y_{i+1} - y_i)} \tag{8}$$

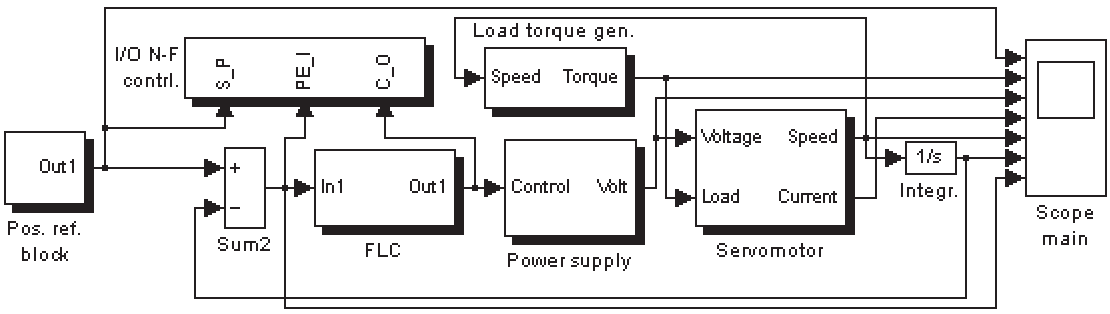


Figure 1: The model of the NFC based servosystem.

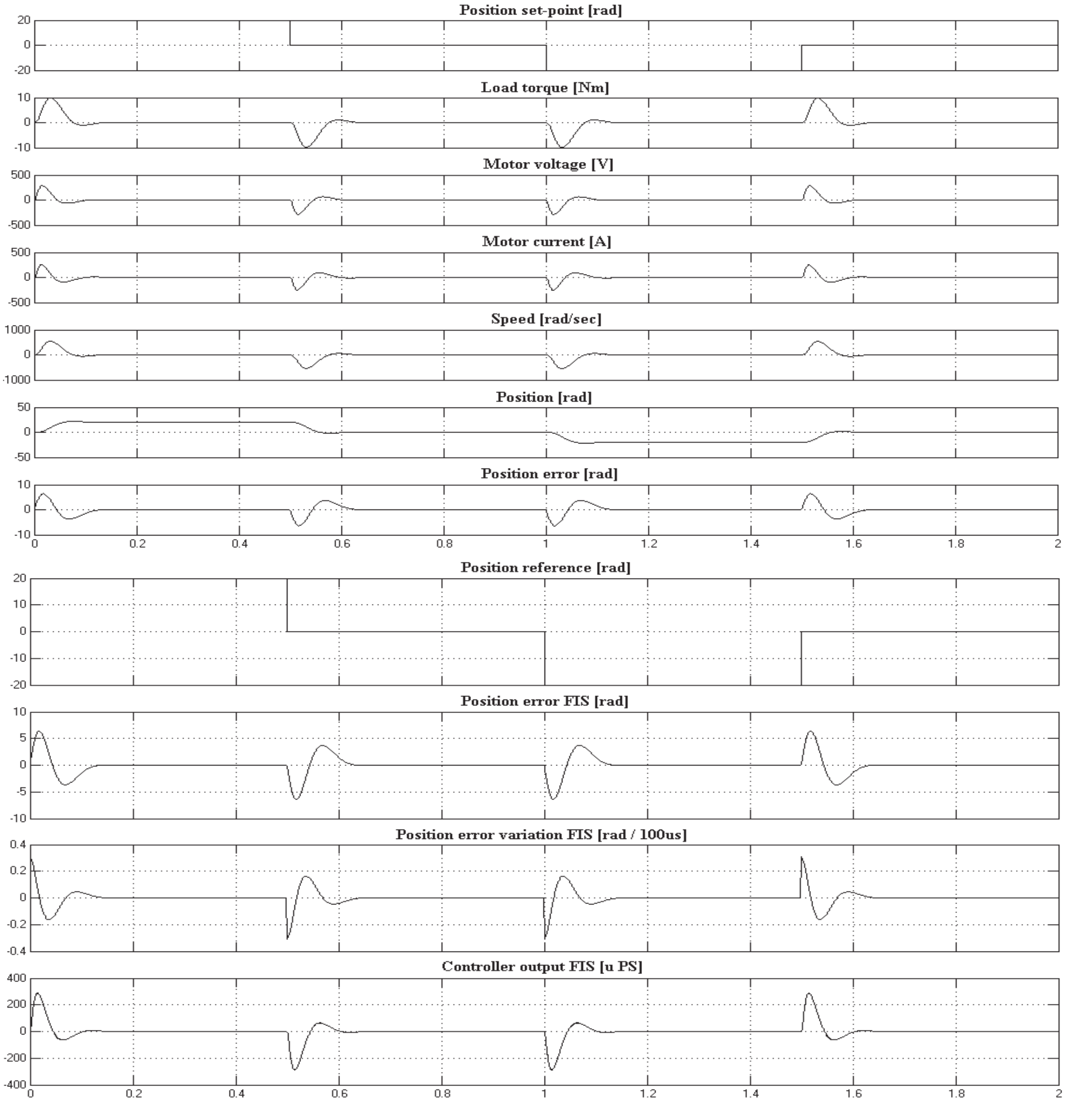


Figure 2: The main variables for the servosistem equipped with a NFC.

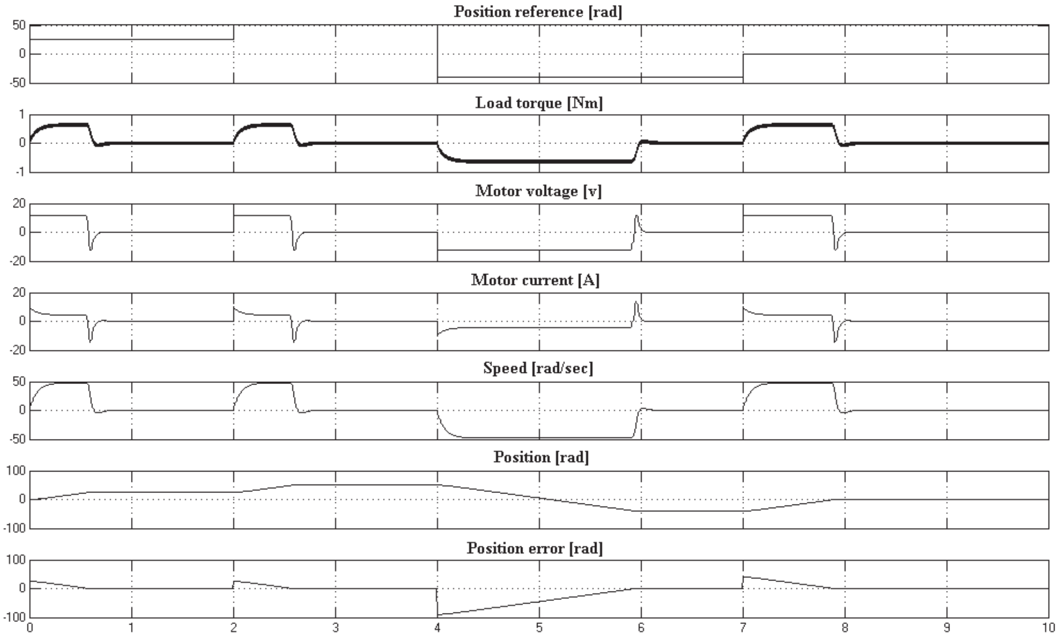


Figure 3: The behavior of the servosystem for strongly deviated condition from the training stage.

For a an advanced N-F controller, with a third input (the current of the motor), a 3D LUT is obtained by n data arrangment like in fig. 7. Each z plane has a 2D LUT and the memory amount will be N_P. N_L. N_P. n_v. The third index (P) is for the plane. Similar relations give the location of the control:

$$k = 1 + \frac{z^* - z_m}{r_z} \tag{9}$$

$$L(i,j,k) = A_{BLUT} + n_v \cdot [(k^* - 1) \cdot N_L \cdot N_C + N_C \cdot (j^* - 1) + i^*] \tag{10}$$

A 3D interpolation is possible too but it involves a more complex computation – fig. 8. The framing between the stored nodes gives the values A, ..., H. With r – the axis incements, the results could be computed as:

$$\begin{aligned} X^* = {} & A + \frac{D - A}{r_z} \times (z^* - z_k) + \frac{B - A}{r_x} \times (x^* - x_j) + \\ & + \frac{E - A}{r_y} \times (y^* - y_{i+1}) + \frac{A + H - E - D}{r_y \times r_z} \times (y^* - y_{i+1}) \times \\ & \times (z^* - z_k) + \frac{F + A - B - E}{r_x \times r_y} \times (x^* - x_j) \times (y^* - y_{i+1}) + \\ & + \frac{A + C - B - D}{r_z \times r_x} \times (z^* - z_k) \times (x^* - x_j) + \\ & + \frac{G + E + D + B - A - H - F - C}{r_x \times r_y \times r_z} \times (x^* - x_j) \times \\ & (y^* - y_{i+1}) \times (z^* - z_k) \end{aligned} \tag{11}$$

4. THE REAL-TIME CONTROL

Fig. 9 presents a simplified experimental structure. The first on-line computation is for the variables normalization. Next relations consider the ranges for the real variables and for the LUT data (M: maximum, m: minimum):

$$\begin{aligned} x_{iLUT} &= x_i \cdot \frac{x_{MLUT} - x_{mLUT}}{x_M - x_m} \\ y_{iLUT} &= y_i \cdot \frac{y_{MLUT} - y_{mLUT}}{y_M - y_m}; i = \overline{1,n} \\ c_{iLUT} &= c_i \cdot \frac{c_{MLUT} - c_{mLUT}}{c_M - c_m} \end{aligned} \tag{12}$$

Fig. 10 presents the timing for the real-time tasks. Each falling edge of the encoder pulses launches a hardware processor interrupt. During the service routine, $\varepsilon_{\alpha k}$ is updated. A software interrupt makes the control sampling. In the beginning of each T, the control is found in LUT based on the last framing coordinates of the inputs for the previous sampling period. In fig. 11 the real-time results reveals a right behavior for the system: null potion error and a slow entry to the final point. The

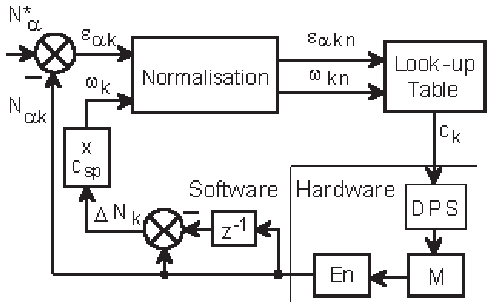


Figure 4. Systemic structure for the application.

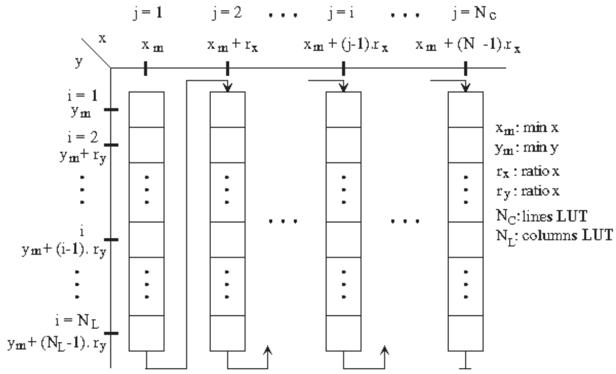


Figure 5: The stored data for the LUT.

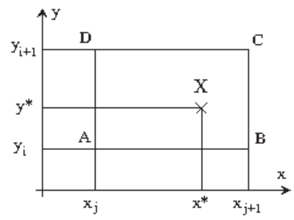


Figure 6: For the 2D interpolation

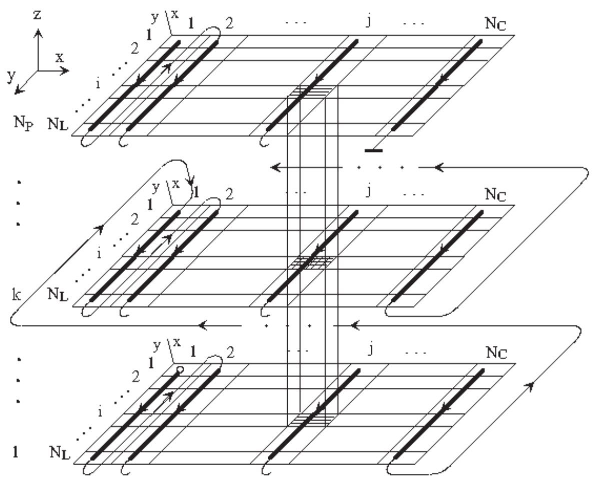


Figure 7: The data for a 3D LUT.

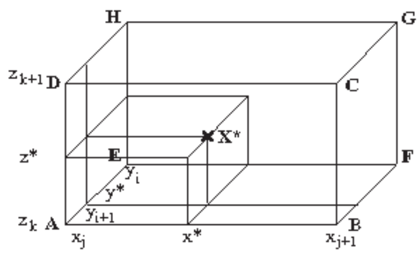


Figure: 8 For a 3D interpolation.

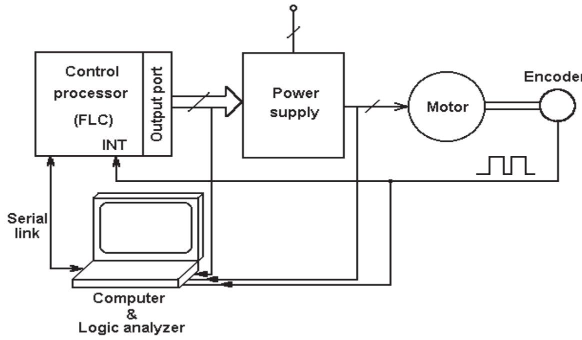


Figure 9. The simplified hardware for the servodrive.

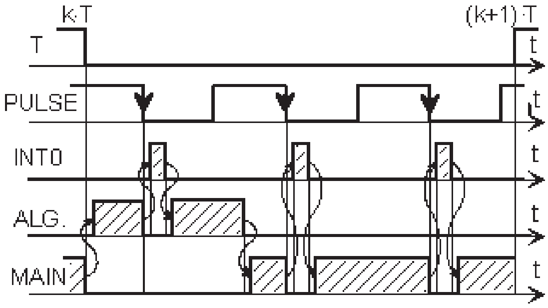


Figure 10: The timing for the on-line experiment.

ringing for the speed diagram comes from the computation manner of this variable from the position error – [14]; the real speed has a continuous variation. The timing was checked for a LUT routine written in assembly language – fig. 12 a and b. For the first case are considered the processing with integers in the range [-10,…,+10], 21x21 LUT matrix, control on 8 bits, independent LUT for the control sign. The second diagram concerns more general conditions with nodes coordinates on 16 bits and signed. More arithmetic routines are involved (like 16+16 bits, 16 x 8 bits, 16 / 8 bits, all signed. The notations meaning: ALL - All Algorithm; CADRP - Control Address Processing; CFETC - Control Fetch; SADRP - Sign Address Processing.

The next diagrams (12c and d) are for the real-time experiment with the servodrive. Now, the notations are:

SPER	Sampling period (2.456 ms)
PULSE	Encoder pulses (2500 pulses / rev)
INT0	External interrupt routine 0 activated by each encoder pulse
PROC	All control processing tasks
ARITH	Arithmetic routines
SCON	Speed loop tasks
NORM	Normalization task
LUT	Searching in look-up-table
SAVES	On-line storage in NVRAM

The fig. 12c certify a right operation of the designed timing and reveals a quite large time reserve. The fig. 12d brings a quantitative image of the final position error: after the last control cycle (all tasks are stopped when $\varepsilon_{\alpha k}$ is zero), only a pulse encoder happens at a very slow speed (for a Np/r = 2500, less that 4 rev. / min).

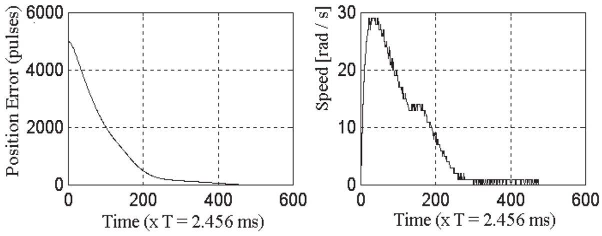
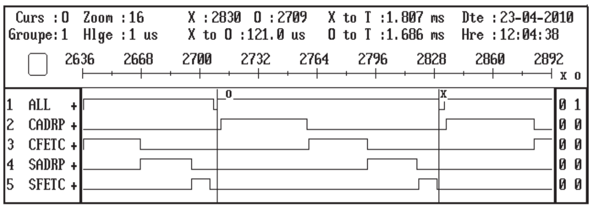
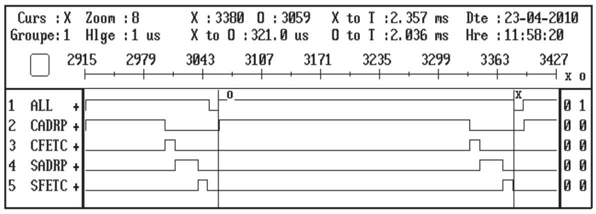


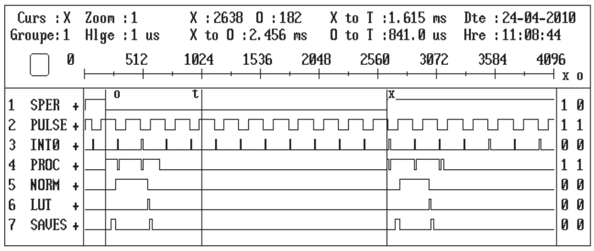
Figure 11: The on-line position error & the speed.



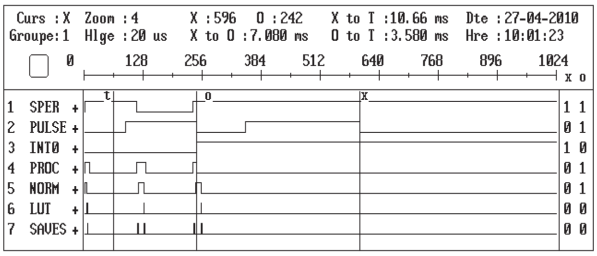
a.



b.



c.



d.

Figure 12: A recording of the real-time tasks.

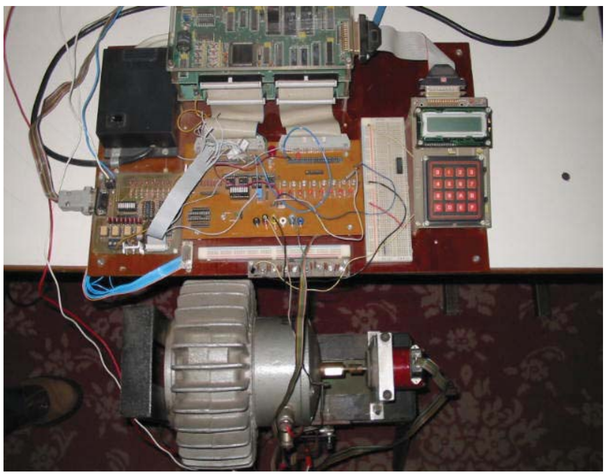


Figure 13: The experimental platform.

The use of the logic analyzer for these qualitative and quantitative results request additional software modules inside the control program. An image of the experimental platform is presented in the fig. 13.

4. CONCLUSIONS

A DISO neuro-fuzzy control brings better results than a SISO variant, mainly because the physical meaning of the second entry: the change of the position error, that meaning an image of the speed. A single controller is able to manage both the position (the main variable of a servodrive) and the speed. A large variety of test conditions was considered (motor parameters, different operation cycle and reference values as well as load disturbances) and the simulation of the neuro-fuzzy model proved very good qualities (more than expectations). The approach and several considerations could be extended to other servodrives type. The results of the study are used for an implementation in real-time and the easiest approach is to consider the look-up-table associated to the neuro-fuzzy controller. A complete tools set for the preparation and the real-time implementation is presented and the results certify the functionality of all the ideas and procedures.

References

[1] Abd Altayef J., Qun-xiong Z., Real –Time DC Motor Position Control by (FPID) Controllers and design (FLC) Using Labview Software Simulation, Int. Journ. of Eng. Studies, Vol. 1, Number 4 (2009), pp. 247–256.

[2] Allaoua B., Laoufi A., Gasbaoui B., Abderrahmani A., Neuro-Fuzzy DC Motor Speed Control Using Particle Swarm Optimization, Leonardo Electronic Journ. of Practices and Techn., 15, 2009, Cluj-Napoca, Romania pp. 1-18.

[3] Altrock C. von., Fuzzy Logic and NeuroFuzzy Applications Explained, Prentice Hall PTR, 1995.

[4] Coleman C. P., Godbole D., A Comparison of Robustness: Fuzzy Logic, PID and Sliding Mode Control, Proc. of the American Control Conference, 1994, pp. 1654-1659.

[5] Do Wan, K., Jin Bae, P., Young Hoon, J., Effective digital implementation of fuzzy control systems based on approximate discrete-time models, Automatica (IFAC Jour., Vol. 43 , Iss. 10, 2007, pp. 1671-1683.

[6] Farkas F., Halász S., Kádár I., Neuro-fuzzy Speed Controller for DC Drives Using Low Precision Shaft Encoder, Symp. on Appl. Machine Intelligence, Jan. 20-21, 2006 Herlany, Slovakia.

[7] Galende M., Sainz G. I., Fuente M. J., Herreros A., Interpretability-accuracy improvement in a neuro-fuzzy ART based model of a DC motor, Proceedings of the 17th World Congress, The International Federation of Automatic Control Seoul, Korea, July 6-11, 2008, pp. 7034-7038.

[8] Gundogdu O., Erenturk K., Fuzzy control of a dc motor driven four-bar mechanism, Mechatronics, Volume 15, Issue 4, May 2005, Pages 423-438.

[9] Ibbini, M.S., Jafar, A.S., Self-Tuning Fuzzy Logic Controller for a Series DC Motor, Proc. (369) Power and Energy Systems, Acta Press, 2002.

[10] Khuntia S.R., Mohanty K.B., Panda S., Ardil C., A comparative study of P-I, I-P, fuzzy and neuro-fuzzy controllers for speed control of DC motor drive, International Journal of Electrical and Computer Engineering 5:5 2010, pp. 287-291.

[11] Kozek, M., Lorenz, A., Kampas, Ph., Modeling and control of an electric servo drive with strong restrictions in the control variable, Int. Journal of Applied Electromagnetics and Mechanics, Vol. 25, Number 1-4, 2007,pp. 521 – 527.

[12] Mihai D., An Optimized Fuzzy Control Algorithm for Servodrives. Some Real-Time Experiments. Proceedings, IEEE Int. Conf. on Intelligent Systems IS ’06, London, UK, paper 1-4244-0195-X/06, pp. 192-197.

[13] Mihai D. Discrete Fuzzy Control Loops Based on a Motor Neuro-Fuzzy Model. Pushing too Far a Continuous Logic?, Proc., Int. Joint Conf. on Neural Networks & IEEE Int. Conf. on Fuzzy Systems, Budapest, Hungary, 2004, pp 587-592.

[14] Mihai D., An Aggregate Multi-Loop Control for a DC Servodrive. The design of the Neuro-fuzzy Controller, Sielmen 2011, Craiova (Romania) - Chisinau (R. Moldova) – in press.

[15] Ordonez R., Zumberge J., Spooner J. T., Passino K. M., Adaptive Fuzzy Control: Experiments and Comparative Analyses, IEEE Transactions on Fuzzy Systems, Vol. 5, No. 2, May 1997, pp 167-188.

[16] Precup R.E., Preitl S., Rudas I.J., Tomescu M.L., Tar J.K., Design and Experiments for a Class of Fuzzy Controlled Servo Systems, IEEE/ASME Transactions on Mechatronics Issue Feb. 2008,Vol. 13 Issue:1, pp. 22 – 35.

[17] Sharifian M.B.B., Rahnavard R., Delavari H., Velocity Control of DC Motor Based Intelligent methods and Optimal Integral State Feedback Controller, International Journal of Computer Theory and Engineering, Vol. 1, No. 1, April 2009, pp. 81 – 84.

[18] Silveura, P. E., Souza JR., R. de, Biazotto, V. M., Speed Control of an Autonomous Mobile Robot: Comparison between a PID Control and a Control Using Fuzzy Logic. J. Braz. Soc. Mech. Sci., May 2002, vol.24, no.2, pp.127-129.

[19] Tipsuwan Y., Chow M.Y., Fuzzy logic microcontroller implementation for DC motor speed control, IEEE IECON'99, San Jose, CA, Nov. 29 - Dec. 3, 1999, pp. 1271-1276.

[20] Vieiraa J., Diasb F. M. and Mota A., Artificial neural networks and neuro-fuzzy systems for modelling and controlling real systems: a comparative study, *Eng. Appl. of Artif. Intellig.* 17, 2004, pp 265–273.