Application of Learning Methodologies in Control of Power Electronics Drives

J.L. da SILVA NETO[1], L. G. ROLIM[1], W. I. SUEMITSU[2], Member, IEEE, L. O. A. P. HENRIQUES[2], P.J. COSTA BRANCO[3], Member, IEEE, and M. G. SIMÕES[4], Senior Member, IEEE

1DEE, UFRJ, Cidade Universitária, Rio de Janeiro, Brazil, e-mail : (luizneto@dee.ufrj.br, rolim@dee.ufrj.br)
2 COPPE, UFRJ, Cidade Universitária, Rio de Janeiro, Brazil, e-mail: (walter@dee.ufrj.br, porto@coe.ufrj.br)
3 Mechatronics Lab., IST, Lisboa, Portugal, e-mail: pbranco@ist.utl.pt
4 Colorado School of Mines, Golden, USA, e-mail:msimoes@mines.edu

Abstract—- This paper initially discusses the learning methodologies as an underlay approach that advances several applications in power electronics and drive systems. Two applications of learning techniques based on fuzzy logic and neural networks are presented as examples of intelligent electrical drives. Initially, an adaptive learning speed control of a PMSM drive is designed, simulated and implemented. A second example deals with neuro-fuzzy control of a SR drive to reduce pulsating torque. Those techniques are corroborated by both simulation and experimental results, achieving good performance results and fast dynamic response.

***Index Terms*—Learning techniques, Artificial intelligence applications**

I. INTRODUCTION

Artificial intelligence applications are driven by the fact that knowledge needed to solve the problem is incomplete because the source of the information may be unknown at the current time the solution is devised, for example because the environment might be changing and cannot be anticipated through analytical design. AI systems are designed under an open systems approach allowing continuous refinement and acquisition of new knowledge. The following four fields have been mostly recognized as learning methodologies for engineering applications :

Expert Systems - Traditional approach to AI, knowledge is represented at a symbolic level through the development of powerful data structures and algorithms. Declarative languages, such as Prolog, states information about structures, instead of describing manipulation of such data. The introduction of MYCIN in 1972, an expert system capable of diagnosing illnesses, drove a lot of attention of scientific community to applications in industrial fields.

Genetic Algorithms - Computational models based on the Theory of Evolution, where genes embed the problem features. Through mathematical operations describing fitness, reproduction and mutation the computer finds better solutions to evolve species which adapt to their environments.

Fuzzy Logic Systems - Initially proposed as a technique of allowing truthfulness in various degrees as an extension of ordinary set theory. It has been successfully applied to encode linguistic descriptions of variables, bringing expertise of human operators into a programming framework. In the last few years, fuzzy logic expanded as a numerical structure as well, by incorporation with neural network based systems.

Neural Networks - Initially derived from computational models of the brain, where artificial neurons are interconnected. Such a network receives input and strengths of the interconnections are trained to map an input stimulus to an output action. During the last few years hundreds of other networks have been developed based on several topologies and training methods.

Neural networks and fuzzy logic techniques are quite different, and yet with unique capabilities useful for information processing in the field of power electronics and drives with the following features:

- Specifying mathematical relationships among numerous variables in a complex system
- Performing mappings with degree of imprecision
- Control nonlinear systems to a degree not possible with conventional linear systems
- Industrial drives have found a profound influence of fuzzy and neural systems in the last ten years.
- Replacement of classical speed, position and torque controllers with AI based controllers
- New and combined control structures in various vector and direct torque control based high performance drive systems
- New and improved firing signal generation PWM, switching vector schemes and selected harmonic elimination schemes
- Compensation of non-linear effects in discontinuous operation of converters
- On-line nonlinear modeling advancing feedforward schemes
- Parameter estimation, self-commissioning systems
- Flux and torque estimators, virtual sensoring schemes
- Monitoring, diagnosis and fault tolerant drives
- Efficiency optimizers
- Peak power tracking controllers

The interest in applying learning techniques based on fuzzy logic and neural networks to design "intelligent" electrical drives has been growing in the last years, as can be seen by the promising results recently obtained in this research area [1],[2]. Also, the evolution of digital signal processors, and circuit integration, make possible the implementation of complex control techniques with low cost and high reliability. Since power electronics and drives is by nature an interdisciplinary area, the applications of learning methodologies found a quick acceptance since the introduction of such technologies. The commercial application of these methodologies in electrical drives is being considered more seriously in recent years. However, several papers have already proposed the utilization of fuzzy [3],[4],[5],[6] or neural network [7],[8],[9],[12] technologies for electric motors control. Only recently neural-fuzzy technologies have been proposed and there are only a few works in this area, and there is no full comparison yet concerning the performance of fuzzy and neural network applications. AC machines, especially induction motors, and switched reluctance (SR) drives present a very nonlinear behavior, and their control depends on the operating point [10]. Therefore, this article will present some aspects of fuzzy, neural and neural-fuzzy control of electrical motors, presenting initially fuzzy control of synchronous motors and then a neural-fuzzy system for SR drives.

II. Fuzzy Control of Synchronous Motors

This section describes an improved fuzzy adaptation method to construct or modify the knowledge base in the fuzzy logic controller (**FLC**). The objective of the fuzzy logic adaptation mechanism (**FLAM**) is to change the rules' definitions in the **FLC** rule base table, according to the comparison between a reference model output signal and the system output. The **FLAM** is composed by a fuzzy inverse model and a knowledge base modifier. The learning algorithm has a local effect but differently from previous fuzzy strategies it uses a weighting factor for each active rule and considers how much the actual output of each rule influences the control, to avoid unnecessary control signal switching. We will show the efficiency of this method in a **TMS320C30 DSP**-based speed fuzzy control of a Permanent Magnet Synchronous Motor (**PMSM**). The fuzzy logic adaptive strategy can be easily implemented. It has fast learning features and very good tracking characteristics even under severe variations of the system parameters, due to the improved algorithm.

A. Adaptation Algorithm

Although fuzzy controllers allow a more versatile tuning, an adaptation mechanism can compensate changes in the system parameters and operating conditions and even nonlinearities in the load. When used in association with an adaptation strategy the fuzzy controllers have proven indeed to be very powerful. Fuzzy controllers are more flexible. Since it has a great number of adjustable parameters, there is a multiplicity of possible adaptive approaches. Some fuzzy model reference-based learning algorithms have been proposed in recent years [13],[14],[15],[16],[17]. They differ from the fuzzy self-organizing controllers (**SOC**s) in the way that a performance index table is not needed. Instead, a reference model defines the desired dynamics.

Fig.1 shows the structure of the PMSM controller. Same reference input (ω_r) for the **FLC** is sent to the reference model. The system actual output is compared with the reference model output. The resulting error (e_m) and its change (Δe_m) are applied to the **FLAM**. The **FLAM** output signal is used to modify the knowledge base of the **FLC** to produce the required change (Δu) in the control signal. We define the tracking error and its change respectively by:

$$e_m(k) = \omega_m(k) - \omega(k) \quad (1)$$

$$\Delta e_m(k) = e_m(k) - e_m(k\text{-}1) \quad (2)$$

where k^{th} is the sampling instant.

B. The PMSM Model

The PM synchronous motor is fed by a current-controlled pulse width-modulated (**PWM**) inverter. The operation of the drive is similar to that of a current-controlled dc motor

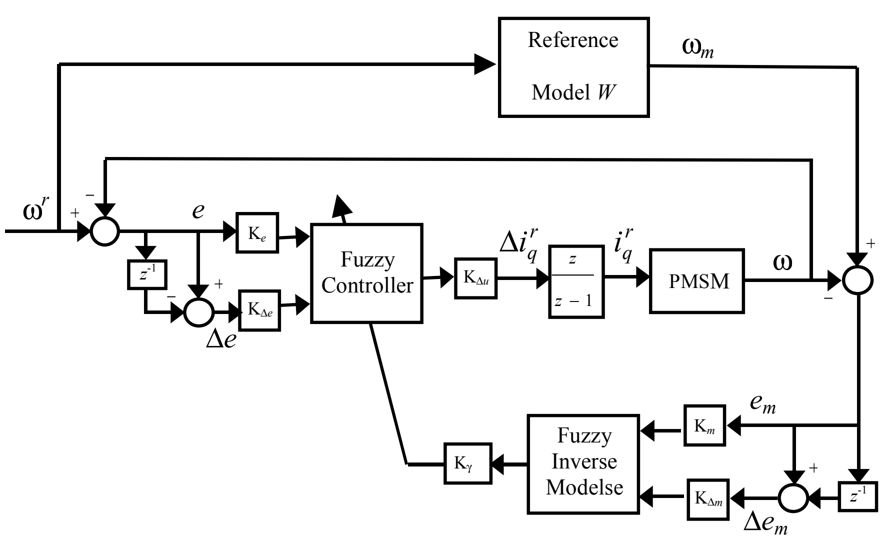

Fig. 1 Block diagram of the fuzzy learning algorithm.

and the drive behavior can be described digitally by the equations:

$$T_{em}(k) = K_T K_i\, i^r_q(k) \quad (3)$$

$$\omega(k) = \left(\frac{K}{f}\right)\frac{1-e^{-fh/J}}{z-e^{-fh/J}}\, i_q^r(k) \quad (4)$$

where $\mathbf{T_{em}}$ is the developed torque, $\mathbf{K_T}$ is the torque constant, $\mathbf{K_i}$ is the gain of the inverter, **J** is the total inertia, ***f*** is the total viscous friction coefficient, and $\mathbf{K=K_TK_i}$ (in most applications the inverter can be modeled simply by a gain relating the reference signal for the quadrature current i^r_q in volts and the real current i_q in amperes.

Such a simple model would be insufficient for most of the conventional adaptive strategies. The implementation of the adaptive control on the real system could fail since it does not account for some linear and nonlinear aspects as: driver time constant, hysteresis band, quantizing effects, etc. On a fuzzy framework however, the model (3)(4) contains enough information to tune the controller and draw conclusions concerning the learning strategy. In the simulations the unmodeled factors listed above were considered.

C. The Fuzzy Logic Controller

The **FLC** used here is a fuzzy **PI**-type controller. Its inputs are the speed error e(k) and change in speed error Δe(k) defined by:

$$e(k) = \omega_r(k) - \omega(k) \quad (5)$$

$$\Delta e(k) = e(k) - e(k\text{-}1) \quad (6)$$

The output of the **FLC** is the change in the quadrature reference current $\Delta i_q^r(k) = K_{\Delta u}\ \Delta u(k)$, with

$$\Delta u(k) = \frac{\sum_{i=1}^{N} \mu_i c_i(k)}{\sum_{i=1}^{N} \mu_i(k)} \quad (7)$$

where μ_i and c_i are respectively the inferred membership degree and the center of the output variable for the i^{th} active rule at the k^{th} instant. Equation (7) represents a typical formula of a Mamdani's type fuzzy system with a Center Average Defuzzifier [16]. The control signal is calculated by integrating the output of the controller, and is given by:

$$i_q^r(k) = i_q^r(k-1) + \Delta i_q^r(k) \quad (8)$$

Combining this expression with (4) the difference equation describing the system output is given as follows:

$$\omega(k+1) = A\omega(k) + Bi_q^r(k-1) + B\Delta i_q^r(k) \quad (9)$$

where A = $\mathbf{exp}(-fh/J)$, B = $K(1-A)/f$.

D. Fuzzy Logic Adaptive Mechanism

The learning method used in the proposed mechanism to tune the **FLC** consists in adjusting the fuzzy sets center or the singleton value c_j of the output variable for each active rule. The number of output fuzzy sets is supposed to be the same as the number of rules.

One can classify the existing approaches into two groups: *i*) using gradient method [14],[15],[16]; *ii*) using fuzzy inverse model [13],[17]. The strategies based on fuzzy inverse models are potentially more robust than those of gradient method, since they do not use explicitly the system parameters in the algorithm, and allow a nonlinear mapping from the inputs to the output. The learning mechanism imposes a change of the form:

$$\Delta c_i(k) = K_\gamma \phi_m(k) \quad (10)$$

where the right hand side represents the output of the fuzzy inverse model.

This strategy induces some *chattering* effects, due to the tendency of the learning parameters to go unbound or to saturate at predetermined levels. Although the changes imposed by the learning mechanism are restricted to the active rules, all the singletons c_j are subjected to the same displacement in the same direction and, so, are pushed beyond reasonable local levels. That leads the **FLC** to an undesirable discontinuous switching mode where a smooth operation was expected, as apparent in the results presented in [17]. The proposed algorithm can address this problem successfully by combining these two approaches. In the gradient method the goal is to minimize a cost function $\boldsymbol{J}(k+1)$. Each c_j is then changed in the direction of the negative gradient of $\boldsymbol{J}$ for some future step n that makes the derivative of the output with respect to the input positive [14]:

$$\Delta c_l(k) \propto -\frac{\partial J(k+n)}{\partial c_l(k)} = -\frac{\partial J(k+n)}{\partial \Delta u(k)}\frac{\partial \Delta u(k)}{\partial c_l(k)} \quad (11)$$

The main objective is to minimize the tracking error reducing the overshoot. Then we must consider some amount of the tracking error and its change into the cost function [13]. Since we want to prevent the learned parameters to go unbound and avoid the chattering problem, the output $\Delta u(k)$ of the **FLC** must be penalized as well. Therefore it is necessary to define $\boldsymbol{J}(k)$ as

$$J(k+1) = \gamma_e \frac{e_m^2(k+1)}{2} + \gamma_{\Delta e}\frac{\Delta e_m^2(k+1)}{2} + \gamma_{\Delta u}\frac{(\Delta u(k))^2}{2} \quad (12)$$

where γ_e, $\gamma_{\Delta e}$ and $\gamma_{\Delta u} > 0$.

From (7) and (9) we can see that each output center $c_j(k)$ affects the machine speed, and consequently the output error, one sample step ahead in time. This means that the control signal is subjected to a learning delay of one sampling period in order to minimize the cost function. Considering Equation (11) the increments for each output's center are calculated by:

$$\Delta c_l(k) \propto \left([\gamma_e e_m(k+1) + \gamma_{\Delta e}\Delta e_m(k+1)]BK_{\Delta u} - \gamma_{\Delta u}\Delta u(k)\right)\frac{\mu_l}{\sum_{i=1}^{N}\mu_i} \quad (13)$$

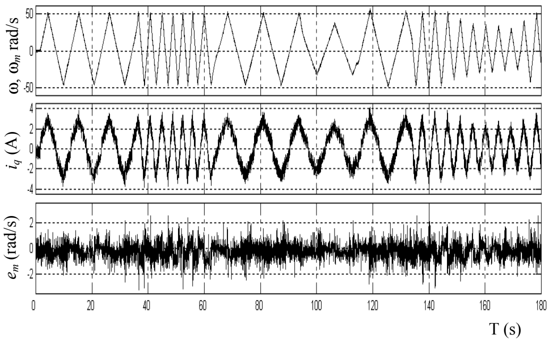

Fig. 2 Experimental results for a tracking problem.

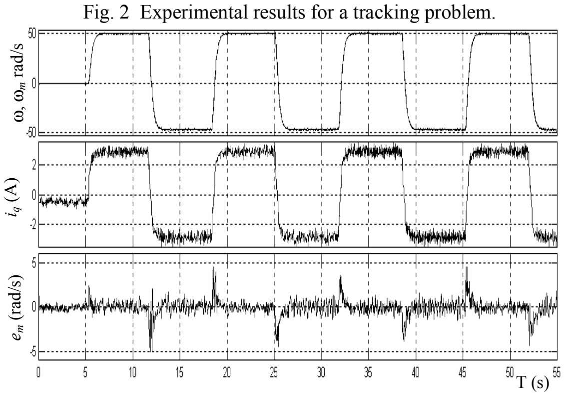

Fig.3 Experimental results for a regulation problem.

The term $[\gamma_e e_m(k+1) + \gamma_{\Delta e}\Delta e_m(k+1)]\mathrm{BK}_{\Delta u}$ is influenced by the system parameters. The proposed mechanism will preserve essentially the same form shown above with this term replaced by the output of a fuzzy inverse model. Additionally we will assume that the delay does not influence too much the learning process. Consequently the adjusting law for each $c_j(k)$ is now given by:

$$\Delta c_l(k) = (\mathrm{K}_\gamma \phi_m[\mathrm{K}_m e_m(k), \mathrm{K}_{\Delta m}\Delta e_m(k)] - \gamma_{\Delta u}\Delta u(k))\frac{\mu_l}{\sum_{i=1}^{N}\mu_i} \qquad (14)$$

In this expression the term $\gamma_{\Delta u}\Delta u(k)\frac{\mu_l}{\sum_{i=1}^{N}\mu_i}$ can be seen as the equivalent output of the l^{th} active rule, or as a projection of the output of the controller over the dimension of the consequent c_j. Fig. 1 shows the block diagram of the fuzzy learning control algorithm. It is important to note that this approach has a local effect. Through the adaptation mechanism the controller will learn to locally respond similarly to a reference model. It is essentially different from the conventional model reference strategies where the goal is to render the closed loop system globally equivalent to the reference model by adjusting the controller parameters, which have a global impact. Obviously in the latter the dynamical specifications are constrained to satisfy the model matching condition.

E. Experimental Results

The digital control was implemented in a TMS320C30 DSP-based system. A second order system was used as a reference model for the adaptive regulation problem. For the adaptive tracking problem a unity gain can be used as a reference model. Both **FLC** and **FLAM** employ as defuzzification method the Center Average Defuzzifier. As initial state of the **FLC** rule base table we assign the value 0.0 for all the consequents. Figure 2 and 3 show respectively the experimental results for a tracking and regulation problems.

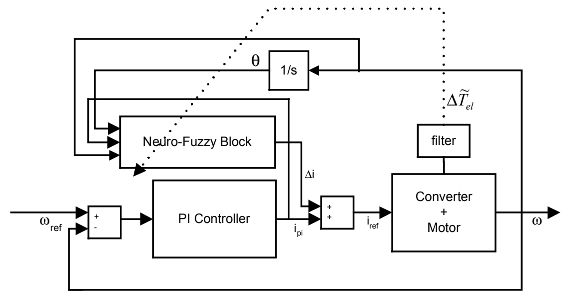

Fig 4 Block diagram of neuro-fuzzy SR control.

III. Neuro-Fuzzy Control of SR Drives

As **SR** machines present strong nonlinear characteristics, fuzzy logic and neural networks methods are well suited for its control, and thus many authors have proposed the dynamic control of **SR** drives using these artificial intelligence based methods [20],[21],[11].

The use of fuzzy logic control has been implemented with success by the authors in [22], and has shown to be effective for the **SR** speed control in applications where some degree of torque ripple is tolerated, as is the case in many industrial applications. Nevertheless, in servo control applications or when smooth control is required at low speeds, the elimination of the torque ripple becomes the main issue for an acceptable control strategy. In this case, the fuzzy logic controller is not good enough, because torque ripple changes with the **SR** motor speed and load. In this context, it is advantageous to include some learning mechanism to the **SR** control to adapt itself to new dynamic conditions. Thus, this paper presents a new methodology to control a **SR** drive that consists on the use of a **PI** speed controller with the supervision of a neuro-fuzzy block responsible for torque ripple reduction. Fig. 4 shows the block diagram of the control system.

A. Neuro-fuzzy based learning mechanism

In this work, the control has more operation flexibility, due to the presence of learning in the compensator, which makes the controller more independent of motor characteristics. If the load, the feeding voltage or the speed changes, the compensator will adapt to operate in this new operation point, adequately reducing the torque oscillation. The proposed strategy for producing this compensation signal consists in incorporating learning mechanisms, like neuro-fuzzy logic systems to the traditional **PI** control system.

The main idea is incorporate a NF compensator in the system. The compensator output is added to the PI output, like in Fig. 4.

As compensator input, we can use the motor speed, rotor position, reference current, and speed error. This operation flexibility is interesting when some data information training was not available.

B. The neuro-fuzzy control

The neuro-fuzzy compensation used is a rule learning by examples method. We utilize a mathematical model representing a neural network which neurons represent membership functions of a fuzzy logic system. The compensation block implemented with **Simulink/Matlab®** has the following input signals: rotor position, current, speed, the training error signal and a flag signal which switches on and off the training of the compensation signal. The output is the compensation current.

The compensation system has flexibility about the number and type of membership functions for each input, that is, if it is necessary to get a different type of membership function, or to change the number of membership functions for a specific input, this can be implemented.

In the first case studied, 3 membership functions for each one of the 3 inputs were used. In the second case, 7 functions were used. For both cases, the membership functions are of gaussian type, to permit that all the function rules could be activated for any input value.

The first task of the compensator is to fuzzify the inputs. After fuzzifying the inputs, a matrix with the antecedents degrees for each rule are calculated. This matrix is obtained scanning all the membership functions of the 3 inputs and multiplying all the functions' outputs. The next step is the system defuzzifying using the center of gravity method for obtaining the outputs.

C. Results

Some relevant simulation results are shown. Fig. 5, 6 and Fig 7. present respectively the phase current, the total torque and the speed error for a simulation during 0.6s, with the starting of learning in 0.2s and a motor speed of 200 rpm.

Fig 8 and Fig 9 show the experimental results of mean

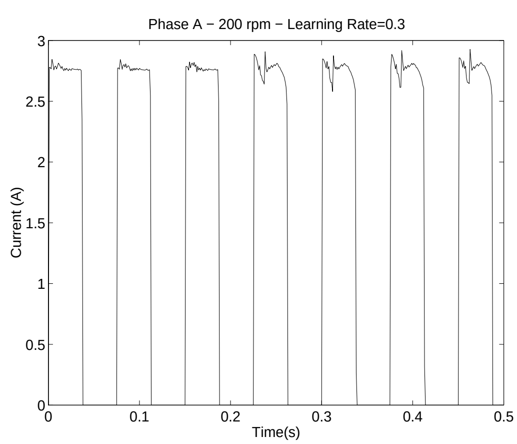

Fig. 5 Phase current profile

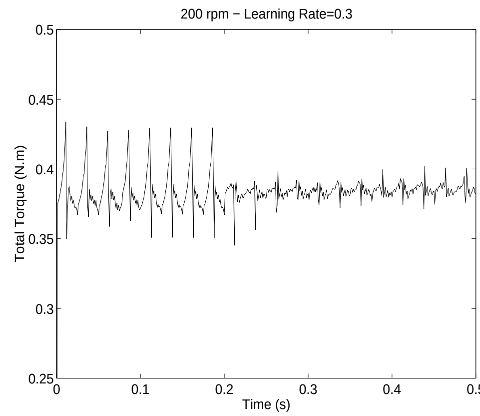

Fig. 6 Total torque

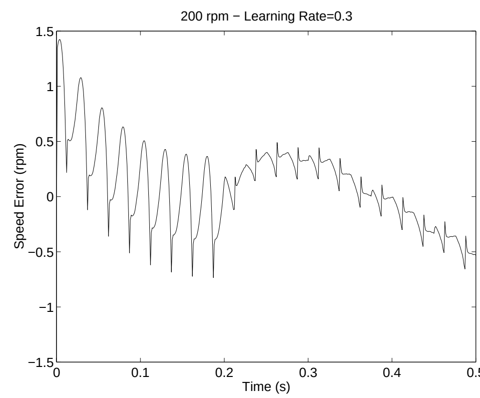

Fig. 7 Speed error

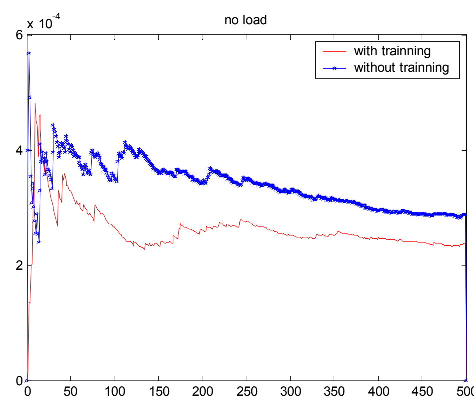

Fig. 8. Mean Square Error.

square error and torque respectively, for an online training operation of 8 minutes at 500 rpm.

IV. Conclusions

The adaptive fuzzy strategy presented applied for **PMSM** drives has proved to be very effective when applied for motion control applications. Although it has been implemented on a speed control of a **PM** motor, it can be extended for other kinds closed loop motor control.

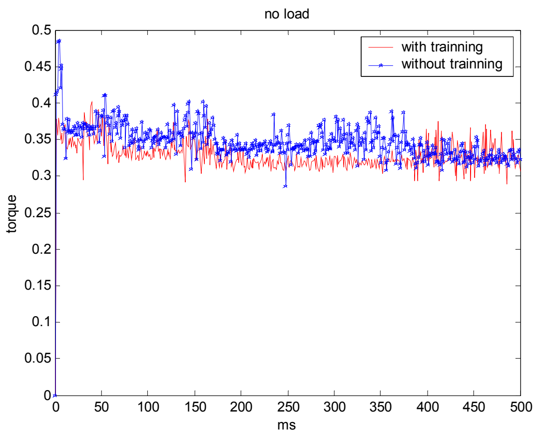

Fig 9 Torque signal

The improved algorithm demands little, although reasonable, modifications in the original mechanism based on the fuzzy inverse model approach. At each sample instant the **FLAM** will affect only the active rules, taking into account its weight in the control signal. One highlighted characteristic of this algorithm is that it can compensate non-linear load variations without the need of a completely modeled load. It should be noted that this strategy has learning properties resembling those of a neuro-fuzzy control system.

For the **SR** drive the neuro-fuzzy strategy has shown to be effective to reduce torque oscillations, which are the main problem for this type of drive. The adaptive algorithm automatically learns a current profile without the need of observers and state estimators.

V. References

[1] P. Z. Grabowski, M. P. Kazmierkowski, Bimal K. Bose and F. Blaabjerg, "A Simple Direct-Torque Neuro-Fuzzy Control of PWM-Inverter-Fed Induction Motor Drive", *IEEE Trans. on Industrial Electronics,* vol. 47, Nº 4, pp. 863-870, August 2000.

[2] A. F. Stronach, P. Vas, "Design of Fuzzy-Neural Controllers for Variable Speed Drives", *ICEM Proceedings,* vol. III, pp. 266-271, September 1996.

[3] M. Godoy Simões, B. K. Bose and R. Spiegel, "Fuzzy Logic Based Intelligent Control of a Variable Speed Cage machine Wind Generation System", *IEEE Trans. on Power Electronics,* vol. 12, Nº 1, pp. 87-95, January 1997.

[4] S. Mir, M.E. Elbuluk and I. Husain, "Torque-Ripple Minimization in Switched Reluctance Motors Using Adaptive Fuzzy Control", *IEEE Trans. on Industry Applications,* vol. 35, Nº 2, March/April 1999.

[5] M.G.Rodrigues, A.C.Siqueira, W.I.Suemitsu, H.H.Bothe, "Fuzzy Logic Control of a Switched Reluctance Motor Positioning", *Proceedings of the Second International ICSC Symposium on Intelligent Indus. Automation* pp. 231-235, Nîmes, France, September 1997.

[6] J.L. Silva Neto, H. Le-Huy, "An Improved Fuzzy Learning Algorithm for Motion Control Applications", *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society, 1998 - IECON '98,* vol. I, pp. 1-5, Aachen, Germany, September 1998.

[7] D. Fodor, G. Griva, F. Profumo, "Neural Network Flux Estimator for Universal Field Oriented (UFO) Controllers", *ICEM Proceedings,* vol. III, pp. 196-201, September 1996.

[8] K. Shimane, S. Tanaka, S. Tadakuma, "Vector-Controlled Induction Motors Using Neural Network", *Electrical Engineering in Japan,* vol. 115, Nº 1, pp. 75-82, 1995.

[9] J.O. P. Pinto, B. K. Bose, L. E. B. da Silva, M. P. Kazmierkowski, "A Neural-Network Based Space-Vector PWM Controller for Voltage-Fed Inverter Induction Motor Drive", *IEEE Trans. on Industry Applications,* vol. 36, Nº 6, pp. 1628-1636, November/December 2000.

[10] L. O. P. Henriques, P.J.C. Branco, L.G.B. Rolim, W.I. Suemitsu, and J.A. Dente Torque ripple minimization in a switched reluctance drive by neuro-fuzzy compensation *IEEE Transactions on Magnetics*. Vol. 36, No. 5/Part 1, pp. 3592-3594, September.2000.

[11] L. O. P. Henriques, P.J.C. Branco, L.G.B. Rolim and W.I. Suemitsu, Review of the ripple reduction strategies in SRM, *Proceedings of CBA2002*, Congresso Brasileiro de Automática, UFRN, Brazil September, 2002

[12] P.Almeida, R.Stephan, P.Branco, W.Suemitsu, "Rotor Flux Angle Estimation Using Neural Networks", *Electrimacs'99 Proceedings*, vol. I, pp.I-157-I-162, Lisbon, Portugal, setembro de 1999.

[13] J.R. Laynel & K.M.Passino, "Fuzzy Model Reference Learning Control for Cargo Ship Steering", *IEEE Control Systems Magazine* , December 1993, pp. 23-34.

[14] B.-M. Chung, "Tunning Effect of Fuzzy Membership Functions", *Proc.VI IFSA World Congress* vol. I, July 1995.

[15] B.-M. Chung & J.-H. Oh, "Control of dynamic systems using fuzzy learning algorithm", *Fuzzy sets and systems*, vol. 59, no. 1, pp. 1-14, 1993.

[16] B.-M. Chung & J.-H. Oh, "Autotuning Method of Membership Fuction in a Fuzzy Learning Controller ", *Journal of Intelligent and Fuzzy Systems*, vol. 1, pp. 335-349, 1994.

[17] L. Zhen & L. Xu, "A Comparison Study of Three Fuzzy Schemes for Indirect Vector Control of Induction Machine Drives ", *Proc. IEEE IAS Annual Meeting*, pp. 1725-1731, 1996.

[18] L.-X. Wang, *A Course in Fuzzy Systems and Control*, PTR Prentice Hall : Upper Saddle River, NJ, 1997.

[19] T. J. Ross, "Fuzzy Logic with Engeneering Applications", McGraw-Hill, Inc.: New York, NY, 1995.

[20] D. S. Reay, T. C. E. Green, B. W. Williams. " Applications of Associative Memory Neural Networks to the Control of a Switched Reluctance Motor" *IEEE Magazine of Control Systems*, Jun 1995

[21] S. Mir, M. E. Elbuluk, I Husain, "Torque-Ripple Minimization in Switched Reluctance Motors Using Adaptive Fuzzy Control". *IEEE Transaction on Industry Applications* v. 35, n. 2, pp. 461-468 Mar/Abr 1999

[22] M.G.Rodrigues, W.I.Suemitsu, P.Branco, J.A.Dente, L.G.B.Rolim, "Fuzzy Logic Control of a Switched Reluctance Motor", IEEE International Symposium on Industrial Electronics, pp. 527-531, Guimarães, P, julho de 1997.