Adaptive Control of Electric Drives Using Sliding-Mode Learning Neural Networks

G. L. Cascella*, F. Cupertino*, A. V. Topalov**†, O. Kaynak** and V. Giordano*

* Politecnico di Bari/Dipartimento di Elettrotecnica ed Elettronica, Via Re David, 200-70125 Bari, Italy

** Bogazici University/Electrical and Electronic Engineering Department, Bebek, 34342 Istanbul, Turkey

† On leave from Technical University Sofia – Plovdiv branch/Control Systems Department, 4000 Plovdiv, Bulgaria

Abstract—**New sliding mode control theory-based method for on-line learning in multilayer neural controllers as applied to the speed control of electric drives is presented. The proposed algorithm establishes an inner sliding motion in terms of the controller parameters, leading the command error towards zero. The outer sliding motion concerns the controlled electric drive, the state tracking error vector of which is simultaneously forced towards the origin of the phase space. The equivalence between the two sliding motions is demonstrated. In order to evaluate the performance of the proposed control scheme and its practical feasibility in industrial settings, experimental tests have been carried out with electric motor drives.**

I. Introduction

In those applications where the knowledge of the system to be controlled is fragmentary or obtainable only in a costly way through complex off-line experiments, artificial neural networks (NNs) can be an effective instrument to learn from input-output data and efficiently catch information about the most appropriate control action to apply. However the application of NNs in feedback control systems requires the study of their properties such as stability and robustness to environmental disturbances and structural uncertainties before drawing conclusions about the performances of the overall system [1]. Moreover, in neuro-adaptive systems, in order to compensate for the existing variable and unpredictable disturbances and changes in the plant parameters, robust and fast on-line learning of the neural controller is a key issue.

Recently, Variable Structure Systems (VSS)-based algorithms have been proposed for on-line tuning of NNs, showing very interesting properties and proving to be faster and more robust than the traditional learning techniques. One of the first studies on adaptive learning in single layer network architectures is due to Ramirez et al. [2]. In another paper [3], the existence of a relation between sliding surface for the plant to be controlled and the zero learning error level of the parameters of the single layer neuro-controller is discussed and the control applications of the method considered in [2] are studied. Differently from [2, 3], the sliding mode algorithms proposed in [4, 5] are for training of multilayer NNs which do not have the limited approximation capabilities of the single layer networks.

Although the results presented in [5] are quite encouraging, they have been obtained through simulation analysis only. The main goal of this work is to prove experimentally the effectiveness of the above algorithm for training of MFNN-based controllers in non-linear feedback control systems. It is also shown that the results obtained in [3] can be also extended to the MFNN controllers. The control applications studied are the speed control of a permanent magnet synchronous motor (PMSM) and of an induction motor (IM) with a nonlinear centrifugal load provided by a fan. In industrial applications, PMSM and IM drives are widely used, due to their inherent features such as versatility, ruggedness and precision. However in some applications, when uncertainties and disturbances are appreciable, traditional control techniques are not able to guarantee optimal performances or can require a considerably time-consuming and plant-dependent design stage. This has recently motivated a considerable amount of research in the field of NNs-based control of electric drives, in order to exploit the property of NNs to learn complex nonlinear mappings [6-10]. In industrial settings, the most widely used controller is still the Proportional-Integral-Derivative (PID) one and the spread of neural controllers for electric drives is contingent on the satisfaction of some critical requirements. Apart from guaranteeing good performance in a wide range of operating conditions, the computational burden presented by the neural controller should be low enough to allow its implementation on low-cost microcontrollers. Furthermore, even in the presence of a fragmentary knowledge of the plant parameters, the start-up procedure (choice of the learning rate, number of the neurons and the network layers, inputs and outputs, as well as the desired NN output) should be fast, straightforward and as general as possible, i.e. applicable for different motors and drives and thus reducing the necessary installation time, with remarkable and captivating cost savings.

The main body of the paper contains five sections. Section II gives the definitions and the formulation of the problem. Section III introduces the equivalency constraints on the sliding control performance for the plant and sliding mode learning performance for the controller. Section IV presents the experimental application of the proposed control scheme to the electrical drives. Finally, section V summarizes the results of this investigation and discusses further improvements.

II. Basic Assumptions and Problem Formulation

Consider a three layer MFNN that is to be used as a neural controller. The following definitions will be used:

0-7803-8738-4/05/$20.00 ©2005 IEEE

$X(t)=\left[x_1(t),\ \dots,x_p(t)\right]^T$ - vector of the time-varying input signals augmented by the bias term.

$U_H(t)=\left[u_{H_1}(t),\ \dots,u_{H_n}(t)\right]^T$ - vector of the output signals of the neurons in the hidden layer.

$u(t)$ - scalar signal representing the time-varying output of the network.

$W1(t)_{(n\times p)}$ - matrix of the time-varying connections' weights between the neurons in the input and the hidden layer, where each matrix's element $w1_{i,j}(t)$ means the weight of the connection of the neuron i from its input j.

$W2(t)_{(1\times n)}=\left[w2_1(t),\dots,w2_n(t)\right]$ - vector of the time-varying connections' weights between the neurons in the hidden layer and the output node. Both $W1(t)_{(n\times p)}$ and $W2(t)_{(1\times n)}$ are considered augmented by including the bias weight components for the neurons in the hidden layer and the output neuron respectively.

$N1(t)$ - vector representing the time-varying output signals of the neurons in hidden layer before applying the activation functions (the neurons net input signals).

$f(\cdot)$ - nonlinear, differentiable, monotonously increasing activation function of the neurons in the hidden layer of the network (e. g. log-sigmoid or tan-sigmoid function). The neuron in the output layer of the neural network is considered with a linear activation function.

The MFNN-based controller is assumed to operate within an adaptive control scheme, the general structure of which is presented in Fig.1, [3].

The sliding surface $s_p(e,\dot e)$ and the zero adaptive learning error level $s_c(u,u_d)$ are defined as follows

$$s_p(e,\dot e)=\dot e+\lambda e \tag{1}$$

$$s_c(u,u_d)=u-u_d \tag{2}$$

where, λ determines the slope of the sliding surface and u_d is the desired control input which is generally unknown.

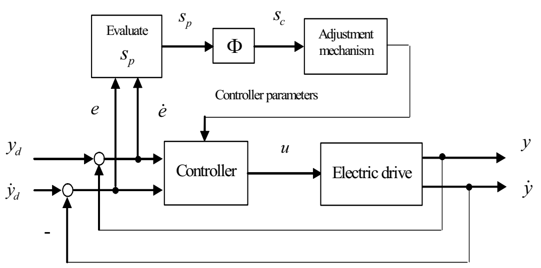

Figure 1. Block diagram of the sliding mode neuro-adaptive control system

An assumption is made that the input vector of the controller $X(t)$ and its time derivative $\dot X(t)$ are bounded, i.e.

$$\left\|X(t)\right\|=\sqrt{x_1^2(t)+\dots+x_p^2(t)}\le B_X\quad \forall t$$
$$\left\|\dot X(t)\right\|=\sqrt{\dot x_1^2(t)+\dots+\dot x_p^2(t)}\le B_{\dot X}\quad \forall t \tag{3}$$

where B_X and $B_{\dot X}$ are known positive constants.

It will be assumed also that, due to the physical constraints, the magnitude of all row vectors $W1_i(t)$ constituting the matrix $W1(t)$ and the elements of the vector $W2(t)$ are also bounded at each instant of time t by means of

$$\left\|W1_i(t)\right\|=\sqrt{w1_{i,1}^2(t)+w1_{i,2}^2(t)+\dots+w1_{i,p}^2(t)}\le B_{W1}\quad \forall t$$
$$\left|w2_i(t)\right|\le B_{W2}\quad \forall t \tag{4}$$

for some known constants B_{W1} and B_{W2}, where $i=1,2,\dots,n$.

It will be assumed that the desired control input $u_d(t)$ and its time derivative $\dot u_d(t)$ are also bounded signals, i.e.

$$\left|u_d(t)\right|\le B_{u_d},\ \left|\dot u_d(t)\right|\le B_{\dot u_d}\quad \forall t \tag{5}$$

where B_{u_d} and $B_{\dot u_d}$ are positive constants.

The output signal u_{H_i} of the i-th neuron from the hidden layer and the output signal of the controller $u(t)$ can be defined as:

$$u_{H_i}=f\left(\sum_{j=1}^{p}w1_{i,j}x_j\right) \tag{6}$$

$$u(t)=\sum_{i=1}^{n}w2_i u_{H_i} \tag{7}$$

Theorem 2.1. If the learning algorithm for the controller parameters and is chosen respectively as

$$\dot w1_{i,j}=-\left(\frac{w2_i x_j}{X^T X}\right)\alpha\ sign(s_c) \tag{8}$$

$$\dot w2_i=-\left(\frac{u_{Hi}}{U_H^T U_H}\right)\alpha\ sign(s_c) \tag{9}$$

with α being sufficiently large positive constant satisfying

$$\alpha>nB_A B_{W1}B_{\dot X}B_{W2}+B_{\dot u_d} \tag{10}$$

then, for any arbitrary initial condition $s_c(0)$, the adaptive learning error $u-u_d$ will converge to zero during a finite time t_h which may be estimated as

$$t_h\le\frac{\left|s_c(0)\right|}{\alpha-nB_A B_{W2}B_{W1}B_{\dot X}-B_{\dot u_d}} \tag{11}$$

and a sliding motion will be maintained on $s_c=0$ for all $t>t_h$.

Proof 2.1: The Lyapunov function candidate may be chosen as follows:

$$V_c=\frac{1}{2}s_c^2 \tag{12}$$

Then differentiating V_c yields

$$\dot V_c=s_c\dot s_c=s_c(\dot u-\dot u_d)=s_c\left\{\left[\sum_{i=1}^{n}w2_i f\left(\sum_{j=1}^{p}w1_{i,j}x_j\right)\right]'-\dot u_d\right\}$$

$$=s_c\left[\sum_{i=1}^{n}\dot w2_i f\left(\sum_{j=1}^{p}w1_{i,j}x_j\right)+\sum_{i=1}^{n}w2_i f'\left(\sum_{j=1}^{p}w1_{i,j}x_j\right)\sum_{j=1}^{p}\left(\dot w1_{i,j}x_j+w1_{i,j}\dot x_j\right)-\dot u_d\right]$$

$$= s_c\left[\sum_{i=1}^{n}\dot{w}2_i u_{Hi} + \sum_{i=1}^{n} w2_i A_i \sum_{j=1}^{p}\left(\dot{w}1_{i,j}x_j + w1_{i,j}\dot{x}_j\right) - \dot{u}_d\right]$$

$$= s_c\left[-\sum_{i=1}^{n}\frac{u_{Hi}}{U_H^T U_H}\alpha\, sign(s_c)u_{Hi} + \sum_{i=1}^{n}A_i\sum_{j=1}^{p}\left(-\left(\frac{w2_i x_j}{X^T X}\right)\alpha\, sign(s_c)x_j w2_i + w1_{i,j}\dot{x}_j w2_i\right) - \dot{u}_d\right]$$

$$= s_c\left(-\alpha sign(s_c) - \sum_{i=1}^{n}A_i\alpha w2_i^2\, sign(s_c) + \sum_{i=1}^{n}A_i w2_i\sum_{j=1}^{p}w1_{i,j}\dot{x}_j - \dot{u}_d\right)$$

$$= -\alpha\left|s_c\right| - \alpha\left|s_c\right|\sum_{i=1}^{n}A_i w2_i^2 + s_c\sum_{i=1}^{n}A_i w2_i\sum_{j=1}^{p}w1_{i,j}\dot{x}_j - s_c\dot{u}_d$$

$$= -\left(\alpha + \alpha\sum_{i=1}^{n}A_i w2_i^2\right)\left|s_c\right| + \left(\sum_{i=1}^{n}A_i w2_i\sum_{j=1}^{p}w1_{i,j}\dot{x}_j - \dot{u}_d\right)s_c$$

$$\le -\alpha\left|s_c\right| + s_c\left(\sum_{i=1}^{n}A_i w2_i\sum_{j=1}^{p}w1_{i,j}\dot{x}_j - \dot{u}_d\right)$$

$$\le -\alpha\left|s_c\right| + \left(nB_A B_{W2}B_{W1}B_{\dot{X}} + B_{\dot{u}_d}\right)\left|s_c\right|$$

$$= \left|s_c\right|\left(-\alpha + nB_A B_{W2}B_{W1}B_{\dot{X}} + B_{\dot{u}_d}\right) < 0 \quad \forall s_c \ne 0 \qquad (13)$$

where $A_i(t)$, $0 < A_i(t) = f'\left(\sum_{j=1}^{p}w1_{i,j}x_j\right) \le B_A \quad \forall i,j$ is the derivative of the neurons' activation function $f(.)$ and B_A corresponds to its maximum value.

The inequality (13) means that the controlled trajectories of the learning error level $s_c(t)$ converge to zero in a stable manner. It will now be shown that such a convergence takes place in finite time. Let us consider the differential equation that is satisfied by the controlled learning error trajectories $s_c(t)$ is as follows

$$\dot{s}_c(t) = -\alpha sign(s_c) - \left(\sum_{i=1}^{n}A_i w2_i^2\right)\alpha sign(s_c) + \sum_{i=1}^{n}A_i w2_i\sum_{j=1}^{p}w1_{i,j}\dot{x}_j - \dot{u}_d$$

$$= -\left(1 + \sum_{i=1}^{n}A_i w2_i^2\right)\alpha\, sign(s_c) + \sum_{i=1}^{n}A_i w2_i\sum_{j=1}^{p}w1_{i,j}\dot{x}_j - \dot{u}_d \qquad (14)$$

For any $t \le t_h$, the solution, $s_c(t)$, of this equation, with initial condition $s_c(0)$ at $t = 0$, satisfies

$$s_c(t) - s_c(0) = \int_0^t \dot{s}_c(\sigma)d\sigma \qquad (15)$$

$$= \int_0^t\left[-\alpha sign(s_c(\sigma))\left(1 + \sum_{i=1}^{n}A_i(\sigma)w2_i^2(\sigma)\right) + \sum_{i=1}^{n}A_i(\sigma)w2_i(\sigma)\sum_{j=1}^{p}w1_{i,j}(\sigma)\dot{x}_j(\sigma) - \dot{u}_d(\sigma)\right]d\sigma$$

At time $t = t_h$ the solution takes zero value and, therefore,

$$-s_c(0) = \int_0^{t_h}\left[-\alpha\, sign(s_c(0))\left(1 + \sum_{i=1}^{n}A_i(t)w2_i^2(t)\right) + \sum_{i=1}^{n}A_i(t)w2_i(t)\sum_{j=1}^{p}w1_{i,j}(t)\dot{x}_j(t) - \dot{u}_d(t)\right]dt$$

$$= -\alpha\, sign(s_c(0))\left[t_h + \int_0^{t_h}\left(\sum_{i=1}^{n}A_i(t)w2_i^2(t)\right)dt\right] + \int_0^{t_h}\left(\sum_{i=1}^{n}A_i(t)w2_i(t)\sum_{j=1}^{p}w1_{i,j}(t)\dot{x}_j(t) - \dot{u}_d(t)\right)dt \qquad (16)$$

By multiplying both sides of the equation with $-sign(s_c(0))$ the estimate of t_h in (11) can be found using the inequality (17).

$$\left|s_c(0)\right| = \alpha t_h + \alpha\int_0^{t_h}\left(\sum_{i=1}^{n}A_i(t)w2_i^2(t)\right)dt - sign(s_c(0))\int_0^{t_h}\left(\sum_{i=1}^{n}A_i(t)w2_i(t)\sum_{j=1}^{p}w1_{i,j}(t)\dot{x}_j(t) - \dot{u}_d(t)\right)dt$$

$$\ge \alpha\left(t_h + \int_0^{t_h}\left(\sum_{i=1}^{n}A_i(t)w2_i^2(t)\right)dt\right) - \left(nB_A B_{W2}B_{W1}B_{\dot{X}} + B_{\dot{u}_d}\right)t_h$$

$$\ge \left[\alpha - \left(nB_A B_{W2}B_{W1}B_{\dot{X}} + B_{\dot{u}_d}\right)\right]t_h \qquad (17)$$

Obviously, for all times $t < t_h$, taking into account the chosen sliding mode controller gain α in (10), it follows from (13) that

$$s_c(t)\dot{s}_c(t) = -\alpha\left|s_c(t)\right|\left(1 + \sum_{i=1}^{n}A_i(t)w2_i^2(t)\right) + \left(\sum_{i=1}^{n}A_i(t)w2_i(t)\sum_{j=1}^{p}w1_{i,j}(t)\dot{x}_j(t) - \dot{u}_d(t)\right)s_c(t)$$

$$\le \left(-\alpha + nB_A B_{W2}B_{W1}B_{\dot{X}} + B_{\dot{Y}_d}\right)\left|s_c(t)\right| < 0 \qquad (18)$$

and a sliding motion exists on $s_c(t) = 0$ for $t > t_h$.

As it has been already mentioned, the desired control input signal (u_d) is generally unavailable and this is the main problem in applying the design presented. If the command error is not available, s_c cannot be constructed. To overcome this difficulty an approach based on the existing relation between the s_p and s_c, will be followed in this investigation.

III. Relation Between the Sliding Mode Control of the Plant and Sliding Mode Learning of the Neural Controller

The differential relation between the sliding line s_p and the zero adaptive learning error level s_c may be specified by the following general equation:

$$s_c^{(n)} = \Phi\left(s_p^{(m)}\right) \qquad (19)$$

If assumed that $m = n = 0$ then, qualitatively, this means that if the value of s_p tends to zero, s_c goes to zero too. On the physical level, the controlled system will achieve a perfect tracking because the controller produces the desired control inputs or vice versa. It will be also true that if the learning error vector is getting away from the origin, that is s_p begins to increase in magnitude, the corresponding divergent behavior in s_c will take place or vice versa.

Let us analyze the following three conditions that the function Φ must satisfy [3]:

A. Region Condition.

The desired control input must drive the state tracking error of the controlled plant to the sliding manifold.

$$\lim_{u \to u_d} s_p = 0 \Leftrightarrow \lim_{s_p \to 0} u = u_d \qquad (20)$$

By using (1) and {2) the two equivalent limits and their consequences can be rewritten as follows

$$\lim_{s_c \to 0} s_p = 0 \Rightarrow \left\{\dot{e} \to -\lambda e \Rightarrow \begin{cases} e \to 0 \\ \dot{e} \to 0 \end{cases}\right. \qquad (21)$$

$$\lim_{s_p \to 0} s_c = 0 \Rightarrow \left\{u \to u_d\right. \qquad (22)$$

From the above statements it follows that Φ must satisfy the following condition.

$$\Phi(0) = 0 \qquad (23)$$

After analyzing the signs of s_p and s_c on the different sides of $s_p = 0$ line it follows also the relation Φ must use the first and the third quadrants of the coordinate system.

$$\Phi(\varepsilon) = \begin{cases} positive & \varepsilon > 0 \\ zero & \varepsilon = 0 \\ negative & \varepsilon < 0 \end{cases} \qquad (24)$$

B. Compatibility Condition.

The tracking performance of the feedback control system can be analyzed by introducing the following Lyapunov function candidate.

$$V_p = \frac{1}{2} s_p^2 \tag{25}$$

A similar Lyapunov function (V_c) has been introduced for the controller performance evaluation in (12). It can be concluded that only the choice of a Φ relation leading to a simultaneous minimization of both Lyapunov functions can be considered suitable since selecting other Φ candidates will violate of at least one of the design objectives.

C. Invertibility Condition.

Let us consider a family of lines drawn in accordance with the equation $s_p = \xi \left(\xi > 0\right)$ for different ξ. Obviously the tracking error vector will fall onto one of these subsets of the phase space at each instant of time. However, based on the relation between s_p and s_c each line from this family corresponds to a different situation entailing different s_c values. It may be also concluded that simultaneously with the increases of the amplitude of ξ, the magnitude of Φ must also increase, because of the increasing distance to the sliding line. Consequently the relation Φ must be invertible, or $\exists s_p \in \Re$ for $\forall s_c \in \Re$.

The above conditions clearly specify that, in order to achieve simultaneous minimization of the two quadratic functions V_p and V_c, the Φ relation must be such to perform a mapping between their horizontal axes.

Theorem 3.1. All continuous, monotonically increasing functions may serve as an Φ relation, satisfying the conditions *A* – *C*, for the establishment of an equivalency between the sliding mode control of the plant and the sliding mode adaptive learning inside the MFNN-based controller.

Proof 3.1. Stability in the Lyapunov sense requires the negative definiteness of the time derivative of the Lyapunov function.

$$\dot{V}_p = \dot{s}_p s_p = \dot{\Phi}^{-1}\left(s_c\right)\Phi^{-1}\left(s_c\right) = \frac{\partial \Phi^{-1}\left(s_c\right)}{\partial s_c}\dot{s}_c\,\Phi^{-1}\left(s_c\right) \tag{28}$$

$$= \frac{\partial \Phi^{-1}(s_c)}{\partial s_c}\left[-\left(1+\sum_{i=1}^{n} A_i w2_i^2\right)\alpha\, sign(s_c) + \sum_{i=1}^{n} A_i w2_i \sum_{j=1}^{p} w1_{i,j}\dot{x}_j - \dot{u}_d\right]\Phi^{-1}\left(s_c\right)$$

$$= \frac{\partial \Phi^{-1}(s_c)}{\partial s_c}\left[\Phi^{-1}(s_c)\left(\sum_{i=1}^{n} A_i w2_i \sum_{j=1}^{p} w1_{i,j}\dot{x}_j - \dot{u}_d\right) - \left(1+\sum_{i=1}^{n} A_i w2_i^2\right)\alpha\left(\Phi^{-1}(s_c)\, sign(s_c)\right)\right]$$

$$\leq \frac{\partial \Phi^{-1}(s_c)}{\partial s_c}\left[\left|\Phi^{-1}(s_c)\right|\left(nB_A B_{W2} B_{W1} B_{\dot{X}} + B_{\dot{u}_d}\right) - \left(1+\sum_{i=1}^{n} A_i w2_i^2\right)\alpha\left(\Phi^{-1}(s_c)\, sign(s_c)\right)\right]$$

$$= \frac{\partial \Phi^{-1}(s_c)}{\partial s_c}\left[\left|\Phi^{-1}(s_c)\right|\left(nB_A B_{W2} B_{W1} B_{\dot{X}} + B_{\dot{u}_d}\right) - \left(1+\sum_{i=1}^{n} A_i w2_i^2\right)\alpha\left|\Phi^{-1}(s_c)\right|\right]$$

$$\leq \frac{\partial \Phi^{-1}\left(s_c\right)}{\partial s_c}\left[\left|\Phi^{-1}\left(s_c\right)\right|\left(nB_A B_{W2} B_{W1} B_{\dot{X}} + B_{\dot{u}_d}\right) - \alpha\left|\Phi^{-1}\left(s_c\right)\right|\right]$$

$$= \frac{\partial \Phi^{-1}\left(s_c\right)}{\partial s_c}\left|\Phi^{-1}\left(s_c\right)\right|\left(-\alpha + nB_A B_{W2} B_{W1} B_{\dot{X}} + B_{\dot{u}_d}\right) < 0, \quad \forall s_p \neq 0$$

In above, the invertibility condition $s_p = \Phi^{-1}\left(s_c\right)$ is used to rewrite the argument of the Lyapunov function. The partial derivative $\partial\Phi^{-1}\left(s_c\right)/\partial s_c$ is positive due to the monotonically increasing behavior of the relation Φ, and $\Phi^{-1}\left(s_c\right) sign\left(s_c\right) \geq 0$ since Φ is defined on the first and the third quadrants of s_c vs. $\Phi\left(s_c\right)$ coordinate system.

Evidently from (28), choosing the bound parameter in accordance (10) enforces the value of s_c to zero level, or equivalently, s_p to zero. It is straightforward to prove that a hitting occurs in finite time (see Proof 2.1).

IV. Experimental Results

In this section experimental results are presented, showing the performance of the MFNN-based control structure with sliding mode learning as a speed controller for electric motor drives in the presence of non-linear disturbances. The different drives considered are vector-controlled drives with a permanent magnet synchronous machine (PMSM) and an induction motor (IM).

The experimental set-up is made up of a 370 W three-phase IM, a 335 W three-phase PMSM, a three-phase inverter, and a dSpace DS1103 controller board which has been designed for rapid prototyping of real-time control systems and is fully programmable in Matlab/Simulink environment through Real-Time Workshop (RTW). Incremental encoders and fans have been keyed to both motor shafts to feedback the rotor position and provide centrifugal loads. The speed controller has been implemented with a Simulink block diagram using a sampling time of 0.2 ms. The dSpace code generator compiles the Simulink program and the real-time executable code is then downloaded to the DSP memory. During motor operation, the DSP receives the feedback from the encoder and commands the appropriate control action to the inverter. The code of the implemented MFNN-based speed controller takes up a very small part of the DSP memory and can therefore be easily embedded in an industrial drive without any extra-hardware. The design of user friendly control panels and virtual instruments for on-line monitoring and parameter tuning has been realized with Control Desk.

In this study, a three-layer MFNN with a hidden layer of hyperbolic tangent neurons and a linear scalar output layer is employed. The neurocontroller has been designed with four inputs which have been chosen to be analogous to the corresponding inputs of a PID controller expressed through the following difference equation [11]:

$$u(k) = u(k-1) + K_1(k)e(k) + K_2(k)e(k-1) + K_3 e(k-2) \tag{29}$$

where $K_i\left(k\right)$, with $i = 1,2,3$, are the controller parameters at time instant k and $e\left(k\right), e\left(k-1\right)$ and $e\left(k-2\right)$ are the feedback errors at time instant $k, k-1$ and $k-2$ respectively.

In this perspective no biases are used so that the output depends on the chosen inputs only.

In order to show that the existence of a priori knowledge about the neural network configuration is not essential, every experiment has been carried out with all the network weights initialized to random values. Several tests have been conducted to choose the parameters of the neural controller, i.e. number of neurons in the hidden layer and the design parameter α, aiming at reaching the simplest configuration to alleviate the computational costs and to shorten the design stage. Thanks to the remarkable speed of learning of the proposed algorithm, only one hidden neuron is enough to guarantee good results. The parameter α has been chosen with a trial and error procedure but there is no need for fine tuning. It is sufficient to gradually increment it to increase the dynamic response of the network until rotor speed oscillations will start to appear. As a general rule of thumb, if the feedback error is big, the network should adapt more quickly so a higher value of α is needed, whereas if oscillations around the set-point arise, the network is adapting too fast and the value of α should be reduced.

Based on the tracking error, first the value of $s_p(e,\dot{e})$ is evaluated and this quantity is passed through the Φ function to get the value of s_c, which is used in the dynamic adjustment mechanism. In evaluating the value of the quantity s_p, the slope parameter of the switching line (λ) is set to unity. The following selection is made parallel to the conditions discussed in the third section

$$\Phi(x) = x \tag{30}$$

In order to reduce the chattering effect, the function in (31) has been used instead of the sign function.

$$sign(x) \approx \frac{x}{|x| + 0.05} \tag{31}$$

A. Speed control of a PMSM

In Fig. 2, the block diagram of a vector controlled PMSM is shown. The control scheme structure is based on the dynamic equations of the motor in the rotor flux reference frame (d,q) [12]. The flux and the torque of the motor are separately controlled by the references i_{sd}^* and i_{sq}^*. For the PMSM, the zero-direct-axis-current strategy ($i_{sd}^* = 0$) is used to operate up to the rated speed [13]. In this condition the PMSM behaves like a DC motor in which the main flux is provided by the permanent magnets and the armature current corresponds to i_{sq} [12]. The electromagnetic torque produced is given by $T_e = K_t \cdot i_{sq}$, where K_t is the torque constant. The MFNN-based speed controller, using the speed feedback ω_r, has the duty to produce the reference value i_{sq}^* that forces actual speed to the reference value. The d-axis component of the stator current reference i_{sd}^* (set to zero) and i_{sq}^* are directly impressed by the action of the built-in control of the inverter that uses as feedbacks the rotor position θ_r, and the stator currents i_{sa} and i_{sb}.

In Fig. 3, the speed response in p.u. is shown, where the rated speed is 4000 rpm. At first, a ramp-shaped speed reference is applied to reach half of the rated speed. After a short time interval equal to 0.25 s, in which the NN parameters are initially adapted and the static friction is overcome, the speed starts changing. Then the full rated speed, reversal speed and zero-speed references are applied. The overshoots are less than 4% of the rated speed and in the steady-state the ripple is less than 0.5% of the rated speed. It is apparent that the controller ensures high rate of convergence of the rotor speed to the reference and possesses an impressive capability to track the command speed despite changing operating conditions.

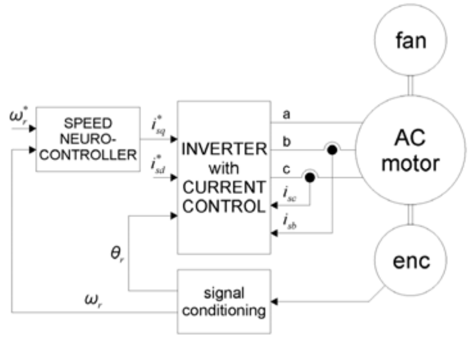

Figure 2. Control architecture for vector control strategy

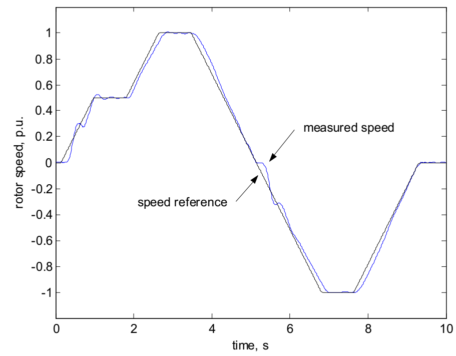

Figure 3. Speed response for the vector-controlled PMSM

It has to be noted that the static friction causes a short delay during the starting and the speed reversal but damps overshoot and removes ripple during the final zero-speed operation. The torque produced by the drive can be seen from the q-axis current response in p.u. shown in Fig. 4, where the rated q-axis base current is 2 A. During the positive rated speed operation, the motor absorbs the rated current. Due to the asymmetrical fan, the 20% over the rated current is required to track the negative rated speed. It is evident that the load torque is a non-linear function of the rotor speed.

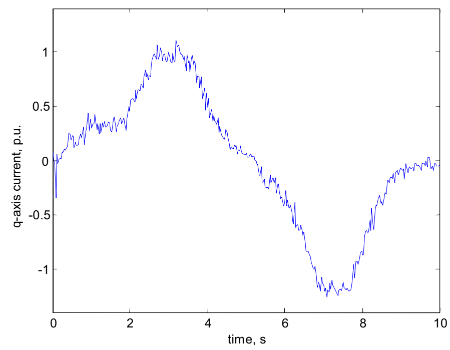

Figure 4. Current responses for the vector-controlled PMSM

B. Speed control of an IM

The scheme depicted in Fig. 2 can be used also for the speed control of and IM. The current reference i_{sd}^* is set to a constant value to guarantee the rated flux of the machine at steady-state [13]. As in the previous case, the IM can then be considered as a DC motor fed by an armature current i_{sq} equal to the output of the NN-based speed controller i_{sq}^*.

In Fig. 5, the speed response in p.u. is shown, where the rated speed is 2900 rpm. After an initial lag of 0.35 s due to the initial parameter adaptation of the NN, the rotor speed tracks the reference. The overshoots are less than 3% of rated speed and in the steady-state the ripple is less than 1% of rated speed. In this case, the static friction does not influence the speed reversal. To evaluate the torque response, the q-axis current in p.u. is reported in Figure 6.

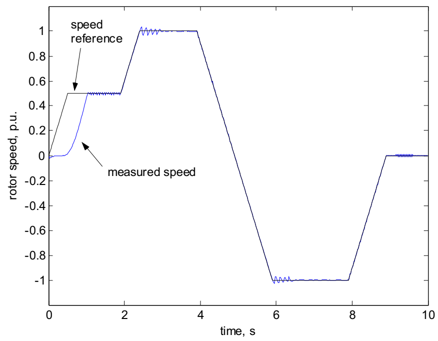

Figure 5. Speed response for the vector-controlled IM

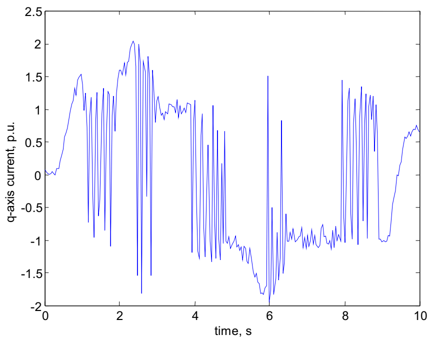

Figure 6. Current responses for the vector-controlled IM

V. CONCLUSIONS

The experimental results obtained indicate that the proposed NN-based controller possesses a number of interesting features, namely:

- good performances in several operating conditions for pump or fan applications without requiring any information about the mathematical model of the overall system;
- high speed of convergence of the algorithm that does not need an initial set-up stage before being applied to the actual system;
- no need for a-priori knowledge of the desired output of the NN for the adaptation process;
- possibility of implementation on low-cost microcontrollers;
- no skilled operator required for the tuning and maintenance stage;
- portability, i.e. distinguished adaptability to different drives with minor further adjustments

ACKNOWLEDGMENT

The authors would like to acknowledge Bogazici University Research Fund Project No: 03A202.

REFERENCES

[1] M. O. Efe and O. Kaynak, “Stabilizing and Robustifying the Learning Mechanisms of Artificial Neural Networks in Control Engineering Applications,” *Int. J. Intell. Syst.*, Vol. 15, No. 5, 2000, pp. 365-388.

[2] H. Sira-Ramirez and E. Colina-Morles, “A Sliding Mode Strategy for Adaptive Learning in Adalines,” *IEEE Trans. on Circuits and Syst.-I: Fundamental Theory and Applic.*, Vol. 42, No. 12, Dec. 1995, pp. 1001-1012.

[3] M. O. Efe, O. Kaynak, and X. Yu, "Sliding Mode Control of a Three Degrees of Freedom Anthropoid Robot by Driving the Controller Parameters to an Equivalent Regime," *ASME J. of Dynamic Syst., Measur. and Cont.*, Vol. 122, Dec. 2000, pp.632-640.

[4] G. G. Parma, B.R. Menezes, and A.P. Braga, “Sliding Mode Algorithm for Training Multilayer Artificial Neural Networks,” *Electronic Letters*, Vol. 34, No. 1, 1998, pp. 97-98.

[5] N. G. Shakev, A.V. Topalov, and O. Kaynak, “Sliding Mode Algorithm for On-line Learning in Analog Multilayer Feedforward Neural Networks,” in the book: Kaynak et al. (Eds): *Artificial Neural Networks and Neural Information Processing – ICANN/ICONIP 2003, Lecture Notes in Computer Science*, Springer-Verlag, 2003, pp. 1064-1072.

[6] A. Rubaai, E. Ricketts, and D.M. Kankam, “Development and Implementation of an Adaptive Fuzzy-Neural-Network Controller for Brushless Drives,” *IEEE Trans. on Indust. Applic.*, Vol.38, No.2, March/April 2002, pp. 441-447.

[7] M. Mohamadian, E. Nowicki, F. Ashrafzadeh, A. Chu, R. Sachdeva, and E. Evanik, “A Novel Neural Network Controller and its Efficient DSP Implementation for Vector-Controlled Induction Motor Drives,” *IEEE Trans. on Indust. Applic.*, Vol. 39, No. 6, Nov./Dec. 2003, pp. 1622-1629.

[8] T.-Ch. Chen and T.-T. Sheu, “Model Reference Neural Network Controller for Induction Motor Speed Control,” *IEEE Trans. on Energy Convers.*, Vol. 17, No. 2, June 2002, pp. 157-163.

[9] A. Rubaai, R. Kotaru, and D.M. Kankam, “A Continually Online-Trained Neural Network Controller for Brushless DC Motor Drives,” *IEEE Trans. on Industry Applic.*, Vol. 36, No. 2, Mar./Apr. 2000, pp. 475-483.

[10] Y. Yi, D.M. Vilathgamuwa, and M.A. Rahman, “Implementation of an Artificial-Neural-Network-Based Real-Time Adaptive Controller for an Interior Permanent-Magnet Motor Drive,” *IEEE Trans. on Industry Applic.*, Vol.39, No.1, Jan./Feb. 2003, pp. 96-104.

[11] F. G. Franklin, D.J. Powell, and M. Workman, *Digital Control of Dynamic Systems*, Addison Wesley, 1998.

[12] W. Leonhard, *Control of Electrical Drives*, Springer Verlag, 1985.

[13] R. Krishnan, *Electric Motor Drives: Modeling Analysis and Control*, Prentice Hall, 2001.