An experimental implementation of SPSA algorithms for induction motor adaptive control

Francesco Cupertino, Ernesto Mininno, David Naso, and Biagio Turchiano
Dipartimento di Elettrotecnica ed Elettronica,
Politecnico di Bari, Bari, 70125 Italy
E-mail: {naso,turchiano}@poliba.it, {cupertino,mininno}@deemail.poliba.it,
Phone: +39 080 5963 649, Fax: +39 080 5963 410

Abstract— **This paper describes the implementation of a self-optimizing embedded control scheme for an induction motor drive. The online design problem is formulated as a search problem and solved with a stochastic optimization algorithm. The objective function aggregates several performance indices on tracking error and control signals, and is measured directly on the hardware bench. The online optimization is performed with Simultaneous Perturbation Stochastic Approximation (SPSA) algorithms, which offer a very effective tradeoff between simplicity of implementation, speed of convergence and quality of the final solutions. The cascaded control system obtained by SPSA in about three minutes of search outperforms alternative schemes obtained with model-based linear design techniques generally used in industrial practice.**

I. Introduction

Embedded control systems are becoming increasingly widespread in industrial automation. In these systems, the actuators are equipped with relatively-low-cost microcontrollers that can also perform self-tuning and adaptation functions. The authors of this paper are engaged with the development of computationally efficient and reliable self-tuning strategies for such embedded controllers. In the case of position control of induction motors (IMs), the problem is made particularly challenging by the typical cascaded structure of the control loops (see Fig.1), which are often tuned consecutively (first the speed controller and then the position controller). Clearly, this approach neglects the possible interactions between the cascaded loops, and may lead, in principle, to suboptimal solutions. The idea investigated in this paper is to optimize the parameters of both controllers of the IM simultaneously and online, using an efficient and "computationally-light" optimization algorithm known as Simultaneous Perturbation Stochastic Approximation (SPSA) method [8]. SPSA is based on a highly efficient approximation of the gradient based on loss function measurements. In particular, for each iteration, the SPSA only needs two loss measurements to estimate the gradient, regardless of the dimensionality of the problem. Moreover, the SPSA is grounded on a solid mathematical framework that permits to assess its stochastic properties also for optimization problems affected by noise or uncertainties. Due to these striking advantages, SPSA has been recently used as optimization engine for many adaptive control problems (see e.g., [1,4, 11, 12], or the extensive survey in [8]).

With respect to the available literature, the contribution of our research is the direct implementation of SPSA on the same microcontroller running the feedback control laws. We believe that this study is interesting because (1) the preponderance of the mentioned application of SPSA has been validated in simulated case studies where the optimization algorithm has a virtually unlimited computational time for each iteration, and (2) other online optimization schemes [2] proposed for a similar hardware bench (based on DC motors instead of IMs) need a host computer to run the optimization algorithm and the various routines necessary to synchronize and reset the hardware controllers, and process the measured signals. Since our research aims at finding an adequate compromise between noise rejection, speed of convergence and quality of the final solution, we implement two variants of the SPSA algorithms and discuss their advantages and limitations.

II. Overview of SPSA algorithms

This section provides a short overview of the basic concepts related to SPSA methods. As technical details are thoroughly described in related literature [8], here we only overview the essential concepts. Consider the problem of finding the minimum of a differentiable loss function $L_n(\vartheta): R^p \to R$ (The subscript *n* here is used to indicate that the loss measurements are affected by noise, whose distribution must satisfy some important conditions [8]). There is a large variety of stochastic algorithms that could be used to find the value of ϑ (say ϑ^*) that minimizes $L_n(\vartheta)$ [7]. The SPSA method computes the estimated value $\hat{\vartheta}$ at the (*k+1*)th iteration as

$$\hat{\vartheta}_{k+1} = \hat{\vartheta}_k - a_k \hat{g}_k(\hat{\vartheta}_k) \tag{1}$$

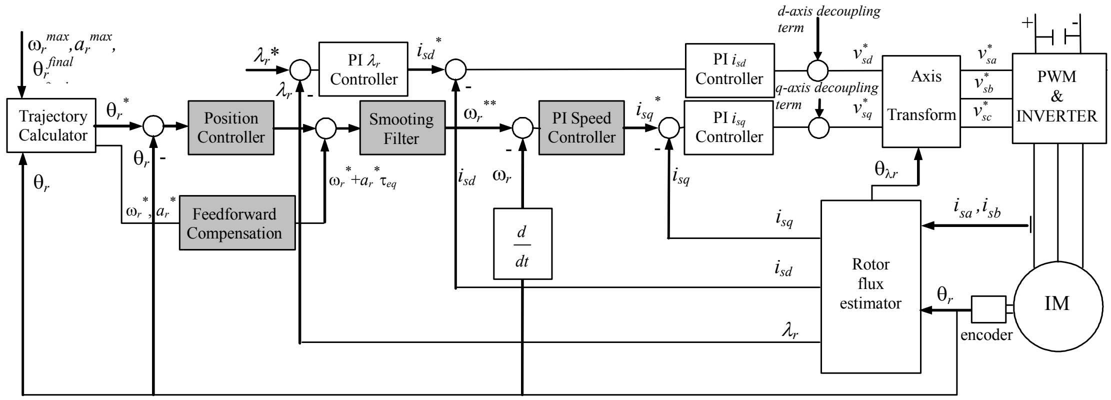

Fig.1. Induction motor drive block diagram.

where $\hat{g}_k(\bullet)$ is the estimated gradient at the kth iteration, and a_k is a gain scheduled to decrease over iterations with the law $a_k = a/(k+A)^{\alpha}$, where a, A, and α are positive configuration coefficients. SPSA estimates $\hat{g}_k(\bullet)$ using the following "simultaneous perturbation" method. Let $\Delta_k = [\Delta_{k1} \quad \Delta_{k2} \quad ... \quad \Delta_{k1}] \in R^p$ be a vector of p mutually independent zero-mean random variables (satisfying the conditions described in [8]), and let the sequence of vectors $\{\Delta_k\}$ be a mutually independent sequence with Δ_k independent of $\hat{\vartheta}_j$, $j = 0,1,...,k$. The basic SPSA (bSPSA) method computes two new points in the solution space, and evaluates the corresponding loss as follows

$$y_k^+ = L_n(\hat{\vartheta}_k + c_k \Delta_k) \tag{2}$$

$$y_k^- = L_n(\hat{\vartheta}_k - c_k \Delta_k) \tag{3}$$

where c_k is a gain sequence $c_k = c/(k+1)^{\gamma}$, and c and γ are nonnegative configuration coefficients. Then, the estimation of the gradient at the kth iteration is computed as

$$\hat{g}_k(\hat{\vartheta}_k) = \frac{y_k^+ - y_k^-}{2c_k} \left[\Delta_{k1}^{-1} \quad \Delta_{k2}^{-1} \quad ... \quad \Delta_{kp}^{-1} \right]^T. \tag{4}$$

It can be noted that all the elements of the vector $\hat{\vartheta}$ are perturbed simultaneously, and that only two measures of the loss are needed to estimate the gradient independently of the size of $\hat{\vartheta}$. Moreover, as the sequence $\{\Delta_k\}$ is usually obtained with a Bernoulli ±1 distribution with equal probability for each value, the perturbations have the same amplitude for all the components of $\hat{\vartheta}$. It has been proven [8] that under certain conditions the bias in $\hat{g}_k(\bullet)$ as an estimate of $g(\bullet)$ tends to zero as $k \to \infty$, and $\hat{\vartheta}_k$ converges "almost surely" to ϑ^*. Literature also provides effective and theoretically valid values for most configuration coefficients, as mentioned in the next sections.

In addition to the bSPSA algorithm described above, a number of effective variants have been recently developed with different aims (e.g., [9,10]). In this paper, we have also considered the *one-measurement form of SPSA (1SPSA),* proposed in [10]. Its peculiarity resides in the formula to estimate the gradient, which is the following variant of (4):

$$\hat{g}_k(\hat{\vartheta}_k) = \frac{y_k^+}{c_k} \left[\Delta_{k1}^{-1} \quad \Delta_{k2}^{-1} \quad ... \quad \Delta_{kp}^{-1} \right]^T. \tag{5}$$

In spite of the fact that this variant does not explicitly calculate the gradient using the difference between loss values, it has been shown that it shares the same nearly unbiased properties of the bSPSA. As noted in [8], since this variant uses only one loss measure for each gradient estimation, it may be advantageous in real-time algorithms as those generally needed for feedback control.

III. THE LOSS FUNCTION

The overall scheme of the control system is illustrated in Fig.1. The whole control scheme is implemented in discrete-time on the dSPACE 1104 real-time control board, equipped with a 250 MHz Motorola PPC working as microcontroller. The microcontroller runs the whole control scheme and a stability supervision algorithm that interrupts the experimentation of badly performing solutions. The various modules have different sampling times. In particular, controller and stability supervisor run with a 200 μs sampling time, while the SPSA algorithms have a 1.125 s sampling time. The execution time of a complete iteration of the control scheme is about 100 μs, and one iteration of the SPSA requires about 180 μs.

The main advantage of the proposed embedded optimization scheme is the reliability of final results. While all the model-based techniques expressly rely on the accuracy of the model (generally used in simulations), in the online tuning case the effects of the actual and unknown high-order phenomena and nonlinearities are fully accounted in the loss measurement, and the final controller (the one generating the smallest loss) is ready-for-use with known performance. This permits to obtain automatic design tools that do not require skilled expertise for system modelling, or trial-and-error controller optimization.

The design of the experiment for loss measurement plays a fundamental role in the success of the online design. In this paper, the position reference signal corresponds to the minimum time trajectory for a rotation of π radians (see Fig.2). After 0.7 s from every change of the reference signal, a step change of load torque (from 0 to 70% of motor rated torque) is applied, in order to evaluate also the overall disturbance rejection. In the final part of the loss evaluation experiment (after 1.125 s) the load torque is suddenly removed (see Fig.2).

The profiles of motor position speed and control action of the PI speed controller (the current $i_{sq}^{*}(t)$) are acquired during the experiment, and used to compute the loss, which essentially takes into account three criteria, i.e. the tracking performance, the disturbance rejection, and the smoothness of the control action. Namely, we define the following loss function (to be minimized):

$$L(\vartheta)=\int_{T_{EXP}}\left(\alpha_1 f_1(\theta)+\alpha_1 f_2(\omega(t))+\alpha_1 f_3\left(i_{sq}^{*}(t)\right)\right)dt \quad (6)$$

where T_{EXP} is the duration of the loss evaluation experiment, α_j represent positive weights, and f_j are three performance indices defined as follows:

$$f_1(\theta(t))=\begin{cases}\left|\theta^{*}-\theta\right| & if \quad \frac{\left|i_{sq}^{*}(t)\right|}{\left|isq_{\max}\right|}<0.99\\ 0 & otherwise\end{cases} \quad (7)$$

$$f_2(\omega(t))=\begin{cases}\left|\omega^{*}-\omega\right| & if \quad \frac{\left|i_{sq}^{*}(t)\right|}{\left|isq_{\max}\right|}<0.99\\ 0 & otherwise\end{cases} \quad (8)$$

$$f_3(i_{sq}^{*}(t))=\begin{cases}\left|i_{sq}^{*F}(t)-i_{sq}^{*}(t)\right| & if \quad \frac{\left|i_{sq}^{*}(t)\right|}{\left|isq_{\max}\right|}<0.99\\ 0 \quad otherwise\end{cases} \quad (9)$$

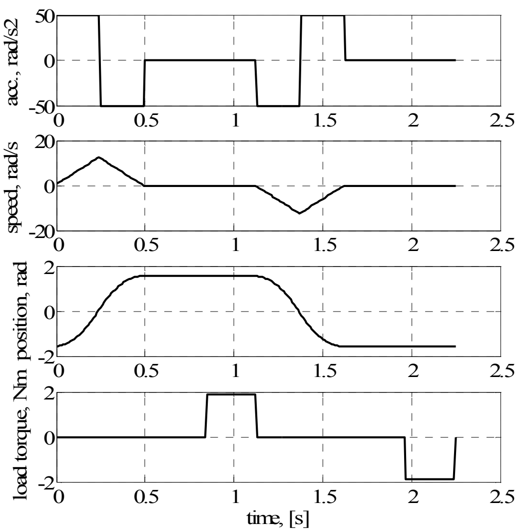

Fig.2. Acceleration, speed and position desired trajectories and load torque applied during the test.

The first two functions f_1 and f_2 take into account the tracking performance and disturbance rejection. More specifically, the position θ and speed ω tracking errors are considered only in the time intervals in which the current feed $i_{sq}^{*}(t)$ is lower than a predefined threshold isq_{max} (the tracking error due to controller saturation cannot be further reduced). The third function f_3 compares the control action $i_{sq}^{*F}(t)$ filtered by a first-order linear filter with time constant τ=0.02 s, with the unfiltered actual action $i_{sq}^{*}(t)$ itself. As smoother control actions give lower values of the integral of f_3, this index is intended to penalize controllers with an excessively oscillatory control action, which may cause stresses for the IM producing vibrations, acoustic noise, and extra losses. The hardware scheme computes each index contributing to the total loss online, i.e. updating its value at each time sample of the experiment. The online value of each loss term is constantly monitored, and whenever it exceeds a predefined threshold the current experiment is immediately stopped. This allows the system to detect unstable (or highly unsatisfactory) solutions well before the involved signals reach potentially dangerous values. In case a monitored index exceeds the prescribed threshold, the value of the loss is multiplied by a penalty factor and assigned to the individual, and the algorithm proceeds with another experiment. In this way, the stochastic search is never interrupted until the terminating condition occurs. On average, less than 1% of the experiments of the first 10-20 iterations are prematurely interrupted due to bad performance, while the remaining iterations always generate stable solutions.

The weights α_j used in loss aggregation permit to emphasize or reduce the contribution of each single performance index in the final value of the loss. In this paper, the α_j are set heuristically, i.e. performing preliminary experiments with changed weights until the desired trade-off between indices is achieved.

Finally, the vector of parameters optimized by SPSA is defined as:

$$\vartheta=\left[k_{pw} \quad k_{iw} \quad k_{pos} \quad \tau_{sm} \quad \tau_{eq}\right]^{T} \quad (10)$$

in which k_{pw} and k_{iw} are the proportional and integral gain of the anti-windup discrete-time PI speed controller (see Fig.1), k_{pos} is the gain of the proportional position controller, τ_{sm} is the time constant of the first order smoothing filter and τ_{eq} is the equivalent time constant of the position control loop (generally estimated offline with system identification procedures [3]). The stopping criterion is the maximum number of loss function calls chosen equal to 200 (i.e. the whole experiment lasts 225 seconds, although the entire algorithm converges in about one third of this interval).

IV. SUMMARY OF EXPERIMENTAL RESULTS

As mentioned, we use SPSA algorithms to optimize online a feedback control system for a vector-controlled induction motor drive. The IM is loaded using a torque controlled brushless generator, mounted on the same shaft. The IM nameplate parameters are as follows: voltage 220 V, current 3.1 A, power 750 W, speed 2860 rpm, torque 2.5 Nm, inertia 0.0012 kgm/s^2, pole pairs 1, torque constant K_c=0.7795 Nm/A.

The performances of SPSA-based controllers are compared with those obtained using controllers obtained by linear design techniques basing on the best available model of the IM [3,5]. In particular, for the model-based controllers, the gains of the current controllers are set so as to achieve a first order closed loop response with time constant equal to $\tau_{is} = 1.2ms$. The same current controllers have been used in all the experiments. For the speed controller, the plant between the output of the speed controller and the measured rotor speed is approximated to a first order system having time constant $\tau_{\Sigma\omega} = \tau_{is} + \tau_{fw} + \tau_{sh}$, i.e. it is equal to the sum of all the lags found in the speed control loop (current control τ_{is} , speed low pass filter τ_{fw} , and delays due to the digital implementation of the control scheme τ_{sh}). In this way The open loop transfer function reduces to the following:

$$G_\omega = k_{pw} \frac{1 + s\tau_{i\omega}}{s\tau_{i\omega}} \frac{1}{1 + s\tau_{\Sigma\omega}} K_c \frac{n_p}{Js} \qquad (11)$$

where $\tau_{i\omega} = k_{pw} / k_{iw}$ is the time constant of the PI controller, J is the inertia of the motor and its load and n_p the number of pole pairs. In order to obtain good disturbance rejection, we use the symmetric optimum theory [5,6], which leads to the following setting

$$\tau_{i\omega} = 4\tau_{\Sigma\omega}, \quad k_{pw} = \frac{J}{2n_p K_c \tau_{\Sigma\omega}} \qquad (12)$$

The gain of the position controller is selected equal to 1/4 of the value that gives a marginally stable system so to eliminate position oscillations near the steady state. Between the position controller and the speed control loop a first order smoothing filter is placed to avoid poorly damped speed responses. The filter time constant is chosen equal to $\tau_{sm} = 1.2 \cdot 4 \cdot \tau_{\Sigma\omega}$ [5]. Then the plant between the output of the position controller and the measured rotor position can be modelled with a first order system with time constant equal to $\tau_{eq} = 1.5 \cdot T_\theta + \tau_{sm} + 4 \cdot \tau_{\Sigma\omega}$, where T_θ is the sampling period of the position controller. Hereinafter, we will refer to the controller designed using the mathematical model as “model-based controller” (MBC). The MBC has gains $k_{p\omega} = 0.067$, $k_{i\omega} = 1.46$, $k_{pos} = 27$, $\tau_{sm} = 0.017$ and $\tau_{eq} = 0.032$.

Also for the SPSA method there are several configuration parameters that must be selected appropriately, which include c and γ, used to define the perturbed points for gradient estimation (2-4), and a, A, and α used to find the new solutions in (1). For this task, we have performed an extensive preliminary configuration study using a simulation model of the IM, also considering the general suggestions provided by literature [4],[7], [8] to properly setup SPSA methods. The main conclusions of this study are that, for our specific problem, the SPSA algorithms can find a reasonable solution in about 200 iterations, with $A=20$ (10% of the expected iterations to find optimum), and γ and α both equal to 0.3 (i.e., slightly different from the values suggested by theory). These values make c_k and a_k reach small and almost constant values after about 150 iterations, letting the SPSA spend the final 50 iterations in local refinements the of controller (see Fig. 3). For the values of a and c, a grid of possible value was investigated. In particular, each couple of parameters was tested 100 times considering different starting points in the search space and calculating the average final loss function value and the percentage of satisfactory runs (i.e., with the final loss function below a predetermined threshold of satisfaction). This study led to the following set: Parameters a=0.03 and c=0.1 for bSPSA and equal to $a=0.0025$ and $c=0.05$ for 1SPSA. The performances of both algorithms during simulations were comparable. The bSPSA reached an average loss function equal to 1.31 and satisfactory performances in the 83% of the runs. The bSPSA reached an average loss function equal to 1.20 and satisfactory performances in the 86% of the runs. Even if the cost function improvement is not much appreciable in terms of performances of the IM, the simulation results suggest that 1SPSA can be profitably used in our online problem.

Once configured as described, both bSPSA and 1SPSA were tested experimentally, obtaining comparable performances that substantially confirm the extensive simulation investigation. As an example of the behaviour of the algorithms, Fig. 4(a) shows the loss function, and the estimates of K_P and τ_{eq} during a typical run. The respective average values, measured during 10 experimental run are also reported in figure 4(b).

The best set of parameters for the selected loss function is $k_{p\omega} = 0.50$, $k_{i\omega} = 11.2$, $k_{pos} = 27$, $\tau_{sm} = 0.04$ and $\tau_{eq} = 0.034$ [hereinafter referred to as SPSA controllers (SPSACs)]. The position response obtained using this set of parameters is reported in Fig. 5 and confirms the

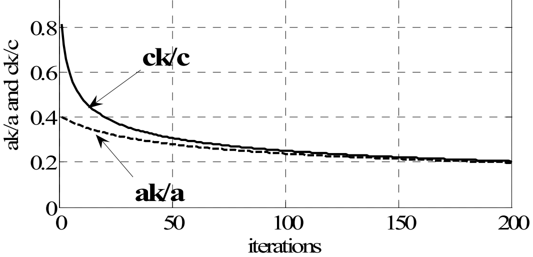

Fig.3. This figure shows the selected decrease rates for parameters c_k/c and a_k/a.

remarkable performances of the SPSA approach.

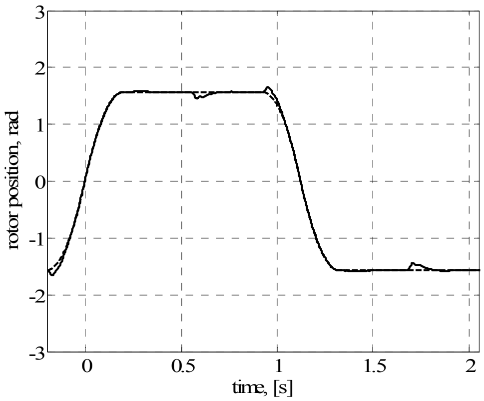

Fig.5 – Position tracking using the best set of controllers selected using SPSA method.

Figure 6-8 compare the performances obtained by using the SPSACs and the MBCs, in terms of position and speed errors and current references. The disturbance rejection of the model-based control is quite unsatisfactory. Even though trial-and error tuning may reduce the apparent overshoots, it is difficult to obtain by hand the performance achieved automatically and in less than 3 minutes by the SPSA approach. The comparison of the speed errors shown in Fig.6 evidences that the SPSACs reduce the speed oscillations using higher gains in the speed PI controllers together with a smoothing filter with a larger time constant. In this way also the disturbance rejection of the electric drive is improved. It must be underlined that this result is not easily obtainable using linear design techniques, even with very accurate models. The model-based current is clearly smoother, although the oscillations produced by the SPSA method remain admissible, and do not cause particular stress to the IM.

V. CONCLUSIONS

The experimental investigation described in this paper confirmed that the SPSA methods offer a striking tradeoff between ease of implementation, computational costs, and search efficiency, which make it possible to directly implement the optimization algorithm on the same microcontroller that runs the discrete time control law. Both considered SPSA variants permit to obtain a very effective closed loop scheme in less than three minutes, in a fairly repeatable and noise-tolerant way, and therefore can be considered fully compliant with the requirement of most embedded control devices. This research has many open aspects currently under investigations, including the evaluation of more complex parametrized controller schemes (e.g., NNs), and more advanced versions of SPSA (e.g., adaptive SPSA [9]).

REFERENCES

[1] Alessandri A., Parisini T. (1997), Nonlinear Modeling of Complex

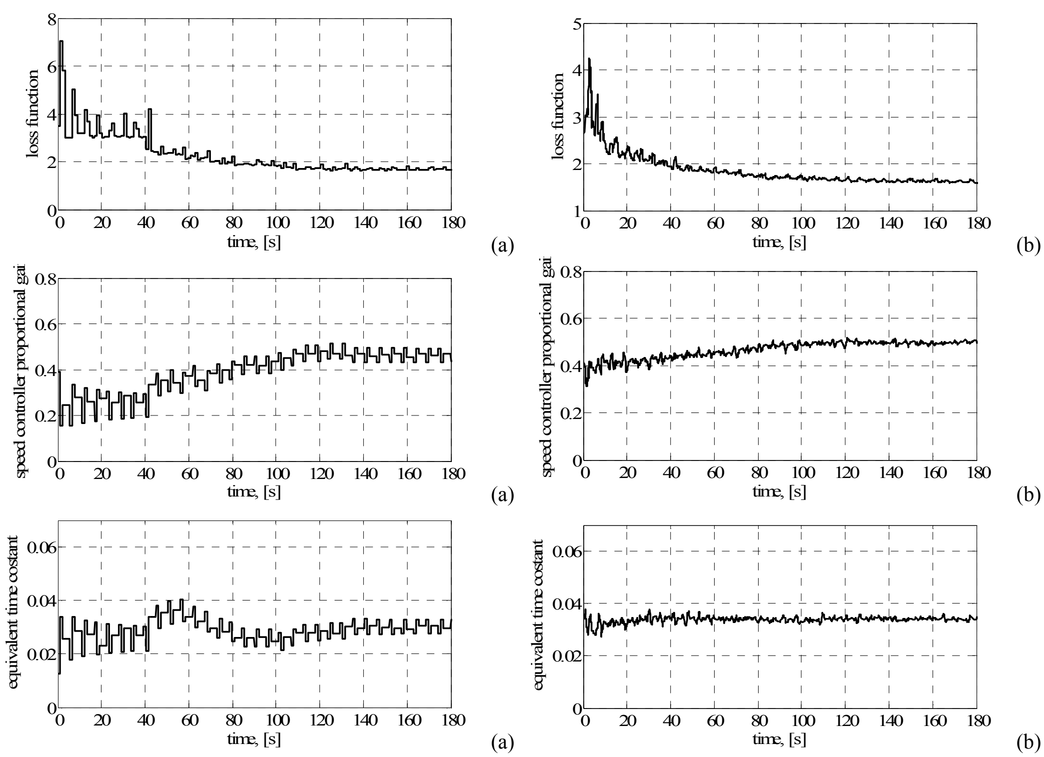

Fig.4. Evolution of loss function, K_P and τ_{eq} over time (the other components of ϑ follow a similar trend, and thus are omitted): (a) a typical run, (b) the average value over 10 runs.

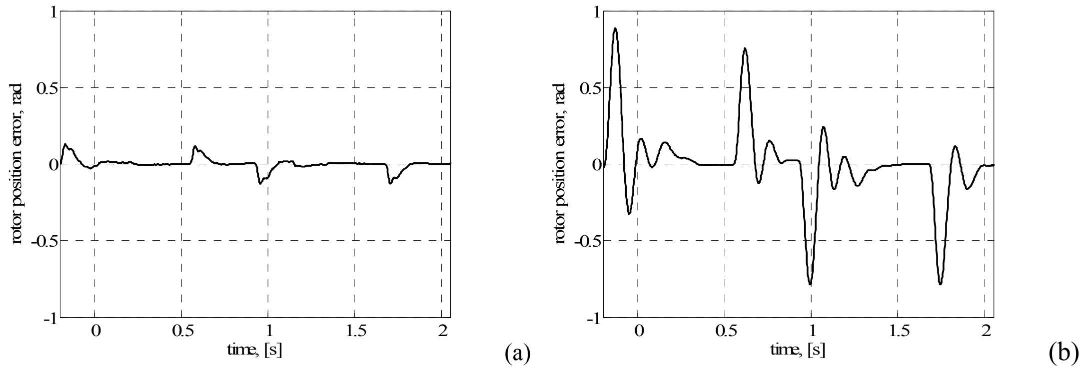

Fig.6. Comparison of position tracking errors under external load torque disturbance, (a) SPSAC, (b) MBC.

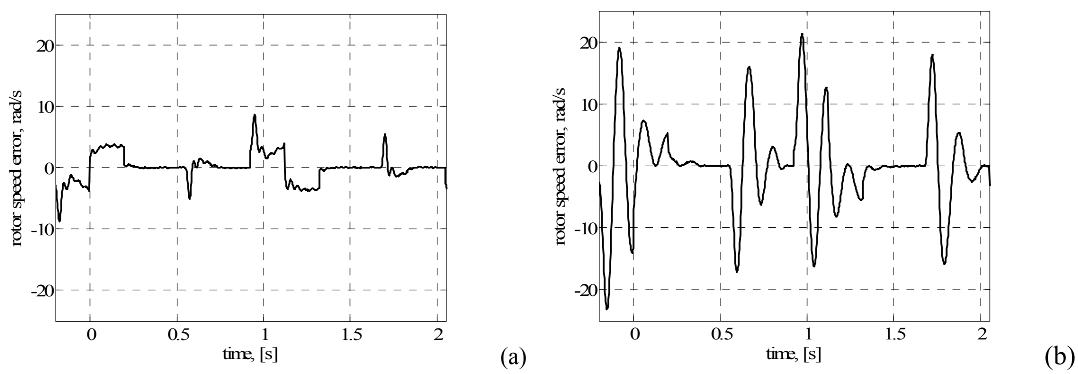

Fig.7. Comparison of speed errors (a) SPSAC, (b) MBC.

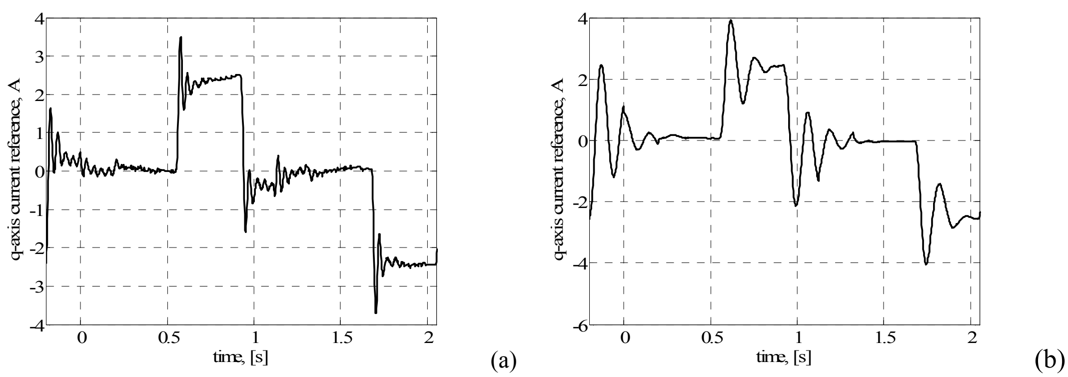

Fig.8. Comparison of current references (a) SPSAC, (b) MBC.

Large-Scale Plants Using Neural Networks and Stochastic Approximation, IEEE Transactions on Systems, Man and Cybernetics, vol.27, pp.750-757.

[2] F. Cupertino, E. Mininno, D. Naso, B. Turchiano, L. Salvatore, “On-line genetic design of anti-windup unstructured controllers for electric drives with variable load”, IEEE Transactions on Evolutionary Computation, Vol.8,. n.4, pp. 347-364, 2004.

[3] F. Cupertino, G. L. Cascella, L. Salvatore, N. Salvatore, “A Simple Stator Flux Oriented Induction Motor Control”, EPE Journal, vol. 15, n. 3, 2005.

[4] Ji X.D., Familoni B.O. (1999), A Diagonal Recurrent Neural Network-Based Hybrid Direct Adaptive SPSA Control System, IEEE Transactions on Automatic Control, vol.44, pp.1469-1473.

[5] H. Grob, J. Hamann, G. Wiegartner, Electrical feed drives in automation, Siemens, 2001.

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

[7] K. Passino, “Biomimicry for Optimization, Control, and Automation”, Springer, US, 2005.

[8] J. C. Spall, Introduction to stochastic search and optimization: estimation, simulation, and control, John Wiley and Sons, Hoboken, NJ, 2003.

[9] J. C. Spall (2000), Adaptive Stochastic Approximation by the Simultaneous Perturbation Method, IEEE Transactions on Automatic Control, vol.45, pp.1839-1853.

[10] J. C. Spall (1997), A One-Measurement Form of Simultaneous Perturbation Stochastic Approximation, Automatica vol.33, pp.109-112.

[11] J.C. Spall, J.A.Cristion (1997), A Neural Network Controller for Systems with Unmodeled Dynamics with Applications to Wastewater Treatment, IEEE Transactions on Systems, Man. And Cybernetics- Part B: Cybernetics 27 no.3, pp. 369-375.

[12] Vande-Wouwer A,. Renotte C., Remy M., On the use of Simultaneous Perturbation Stochastic Approximation for Neural Network Training, Proceedings of the 1999 American Control Conference, San Diego, California, pp.388-392.