Proc. of the 5th WSEAS/IASME Int. Conf. on Electric Power Systems, High Voltages, Electric Machines, Tenerife, Spain, December 16-18, 2005 (pp86-91)

Implementation of Two Error Compensating Methods for an On-Line Learning Speed Controller of a Switched Reluctance Machine

SILVIANO RAFAEL[1], A.J. PIRES[1] , P.J. COSTA BRANCO[2]

[1]LabSEI, Escola Superior de Tecnologia de Setúbal / Instituto Politécnico de Setúbal
Rua Vale de Chaves, Estefanilha, 2914 Setúbal
PORTUGAL

[2]Instituto Superior Técnico
Av. Rovisco Pais, 1096 Lisboa
PORTUGAL

Abstract: - A speed control system for an 8/6 switched reluctance motor was developed using a Neuro-fuzzy controller with on-line learning capability. The results, for the most common functioning regimen, are acceptable, as it is demonstrated in this work. However, in some particular cases, in the reference function tracking, it was observed a significant increase of the error value. These particular cases appear when fast time variations of the reference signal are imposed, implying a change of the machine speed.
In this paper two concrete problematic situations are presented and these ones are analyzed considering two distinct compensating methods solutions, one for each case. Also the experimental results of the proposed solutions are presented.

Key-Words: - Switched Reluctance Machine, Neuro-fuzzy Controller, Neuro-fuzzy Application

1 Introduction

The switched reluctance machine (SRM) is, more and more, used in electromechanical systems replacing successfully other electrical machines. With its characteristics; like the efficiency curve more flat than the asynchronous machine one, or being able to run at very high speed (> 3000 rpm), it becomes the choice for designers of performance productive systems. Applications of the SRM in industry, such as cleanness industry, textile industry and automobile industry among others, appear due to robustness and low cost of maintenance [1]. The command needs, however, the information of the rotor position, which can be given through an incremental encoder.
The SRM design produces considerable speed ripple, mainly at low speeds, and high acoustic noise [2]. Its nonlinear behaviour is mainly related with the inductance of the magnetic circuit that is a function of phase current and rotor position [3]. The SRM thus presents a nonlinear multivariable control structure that calls for complex nonlinear design to achieve a high dynamic performance.
The above-mentioned drawbacks are difficult to solve with conventional control techniques due to the complexity of modelling the SRM dynamics. Therefore, fuzzy controllers are today an attractive control solution to be used with these machines.

2 System Controller

One reason for the choice of such a controller is its learning capability and the possibility of generating a control law based on rule adaptation [4] and [5], minimizing the error goal function. This paper presents and discuss the results obtained with the development and implementation of a microcomputer-based neuro-fuzzy learning speed controller for an 8/6 SRM, driven by a power converter.
Some characteristics were imposed to the motor and connected mechanic machine, such as the direction of rotation and passive braking through the load. This one is very similar to the real systems composites for an electric motor and an analogical dosage pump.

2.1 Neuro-Fuzzy Design

The neuro-fuzzy system can be interpreted as a neural network composed with five layers and it is presented in figure 1. In the presented model the connections between nodes do not correspond to an attributed

weight to the connection but to the propagation of the previous node result.
The input variables x_i of the neuro-fuzzy controller are the speed error and its variation (e_k,Δe_k) defined by (1) and (2).

$$e_k = w_{ref} - w \qquad (1)$$

$$\Delta e_k = e_{k-1} - e_k \qquad (2)$$

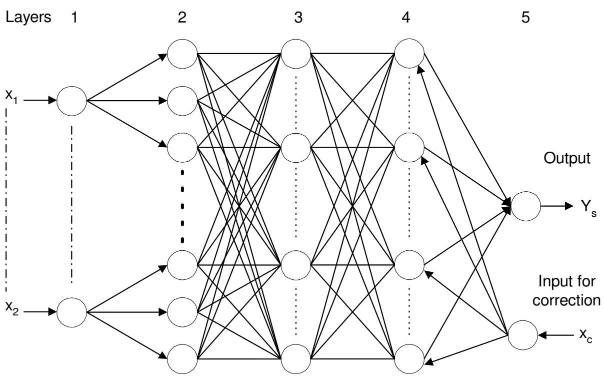


Fig.1 Diagram of the proposed fuzzy neural network for speed controller

Each node has an activation function, representative of the fuzzy system for each layer of neurons. Next, the functions of the nodes in each of the five layers of this connectionist model are described.
In the first layer, the activation functions are for adjustment of the variables values to the universe of discourse, through a linear function with saturation in the upper/lower limits, as expressed by (3).

$$O_{1,i} = \begin{cases} U_{iMAX}; x_i \geq G_{iMAX} \\ a_i x_i + b_i; G_{iMIN} < x_i < G_{iMAX} \\ U_{iMIN}; x_i \leq G_{iMIN} \end{cases} \; ; i \in \mathrm{N}^+ \qquad (3)$$

O_{1ij} represents the result of the node; the universe of discourse is represented by U where U_{iMAX} and U_{iMIN} are its limits; x_i is the input value where G_{iMAX} and G_{iMIN} are its limits and index i is the variable number. In this case the variables will be the speed error and the variation of the error.
In the second layer the fuzzification is performed. The output function of each single node is a simple membership function of the fuzzy system. The membership function used is Gaussian and it is expressed by (4).

$$O_{2(i,j)} = Exp^{-\frac{1}{2}\left(\frac{O_{1,i}-c_{i,j}}{\upsilon_{i,j}}\right)^2} \; ; i \text{ and } j \in \mathrm{N}^+ \qquad (4)$$

Where $c_{i,j}$ and $u_{i,j}$ are respectively, the gaussian function centre (or mean) and width (or variance) of the *jth* term of the *ith* input linguistic variable x_i .
In the third layer the inference mechanism is activated. It is used the *Tnorm*. The used operator is the algebraic product (5).

$$O_{3(i,j)} = O_{2,i} O_{2,j} \; ; i \text{ and } j \in \mathrm{N}^+ \qquad (5)$$

Where i and j are the nodes of the second layer associated with the input variable.
The fourth layer performs the consequent part of the rules through the expression (6). The value of the rule weight $\rho_{k(i,j)}$ is produced by the fifth layer in function of the learning algorithm.

$$O_{4(i,j)} = O_{3(i,j)} \rho_{K(i,j)} \; ; i \text{ and } j \in \mathrm{N}^+ \qquad (6)$$

The fifth layer has two kinds of nodes. The first one performs the decision signal output. These nodes act as defuzzifier and are expressed in (7).

$$O_{5out} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} O_{4(i,j)}}{\sum_{i=1}^{n}\sum_{j=1}^{n} O_{3(i,j)}} \; ; i \text{ and } j \in \mathrm{N}^+ \qquad (7)$$

The second kind of nodes performs the learning in order to minimize the error function (9) represented as input for correction in the diagram of the proposed fuzzy neural network by modifying the value of the nodes of the fourth layer through the weighted rules consequents (8).

$$\rho_{K(i,j)} = \rho_{K-1(i,j)} + \gamma \frac{\partial E}{\partial \rho_{K-1(i,j)}} \; ; i \text{ and } j \in \mathrm{N}^+ \qquad (8)$$

The learning rate γ assumes a value $\in$ [0,1] and E is a cost function to be minimized. E is a quadratic error function (9) where w_{ref} is the value of the speed reference and w_k the value of the machine speed.

$$E = \frac{1}{2}\left[w_{ref} - w_k\right]^2 \; ; \mathrm{k} \in \mathrm{N}^+ \qquad (9)$$

2.2 Error Analysis

The *MPE* and the *MAPE* error analysis were used in this study. It consists in the calculation of the error percentage average (*MPE* - mean percentage error) and the average of the absolute values of the error percentage (*MAPE*- mean absolute percentage error) expressed in (10) and (11), respectively.

$$MPE = \sum_{i=1}^{n} \frac{PE_i}{n} \quad ; i \in \mathrm{N}^+ \qquad (10)$$

$$MAPE = \sum_{i=1}^{n} \frac{|PE_i|}{n} \quad ; i \in \mathrm{N}^+ \qquad (11)$$

PE is the percentage of the speed error, given by (12).

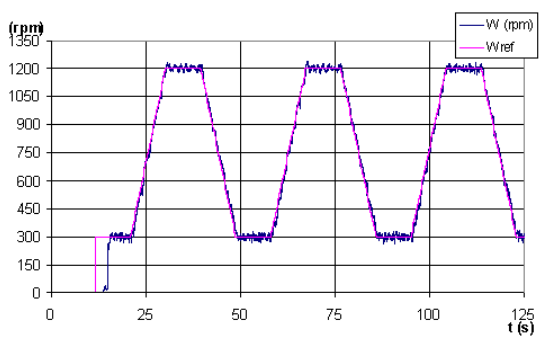
$$PE_i = \frac{(w_i - w_{ref})}{w_{ref}} 100 \quad ; i \in \mathrm{N}^+ \qquad (12)$$

The performance in the speed tracking will be better for lower values of these two error measures.

3 Behaviour of the System and Problem Definition

The laboratorial tests were set up by tracking a trapezoidal speed reference function between 300 and 1200 rpm, 500 and 1400 rpm and finally between 400 and 800 rpm, as presented in figures 2 and 3. The slopes of the first and second group of speed cycles have the angular acceleration of 1,618 rad/s^2 and in the last cycle is 0,719 rad/s^2.

Fig.2: Speed tracking with a trapezoidal reference



function

The error is presented in table 1 for a set of cycles of speed levels. It is evident that the performance presented in the interval between 400 and 800 rpm is the best. This is due to the fact that the acceleration being lower implies a better adaptation of the neuro-fuzzy system to the speed reference signal tracking.

Table 1 - Error performance analysis by speed cycles

Cycles	300 / 1200 (rpm)	500 / 1400 (rpm)	400 / 800 (rpm)
MAPE (%)	2,94	2,26	2,25
MPE (%)	0,005	0,129	0,145

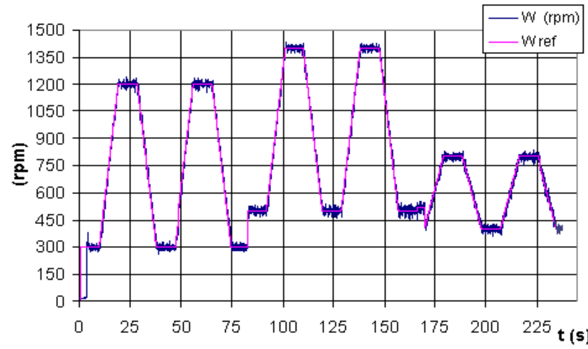


Fig.3: Speed tracking with a trapezoidal reference function for various speed

But in the cases where there is the necessity of changing the reference speed brusquely, the system presents an oscillatory behaviour while the neuro-fuzzy controller readjusts the rules consequents weights. In figure 4 it is presented the behaviour of the machine speed with evident degradation of the set performance. This is one problem that appears in these control systems due to the necessary learning time for acquire knowledge about the new functioning regimen. This necessary time will be more or less long in function of the algorithm execution, the learning rate, the neuro-fuzzy system net dimension and the type of computer used.

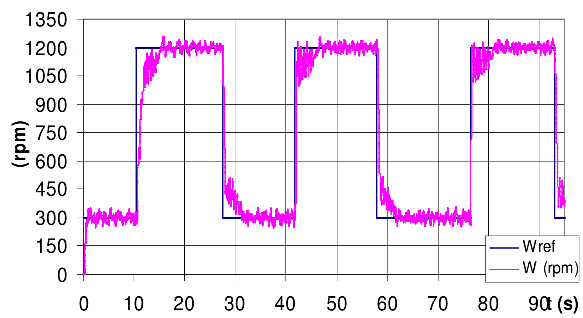


Fig.4: Speed tracking with a rectangular reference function

3.1 Slopes Reference Tracking

The tracking of the triangular speed reference function is presented in figure 5. The speed tracking error is 0,19% for *MPE* and 3,0% for *MAPE*. One evidences that it is slightly higher than the one presented with a trapezoidal function in Table 1.

The tracking of sinusoidal speed reference function is presented in figure 6. The error with *MPE* is 0,32% and with the *MAPE* is 5,13%, assuming higher values related to the previous ones, due to the increase of the slope. The acceleration for the triangular and trapezoidal functions is 9,40 rad/s2 and for the sinusoidal function is 20,06 rad/s2 which represents an increase of 213%.

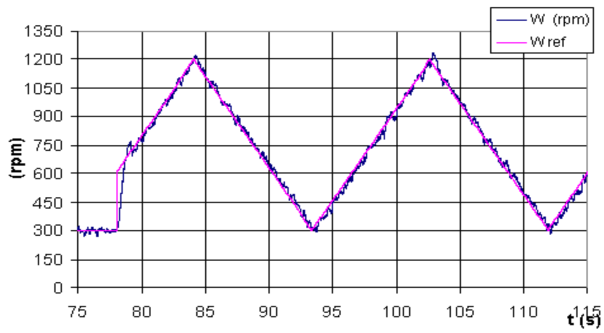


Fig. 5. Speed tracking with triangular reference function

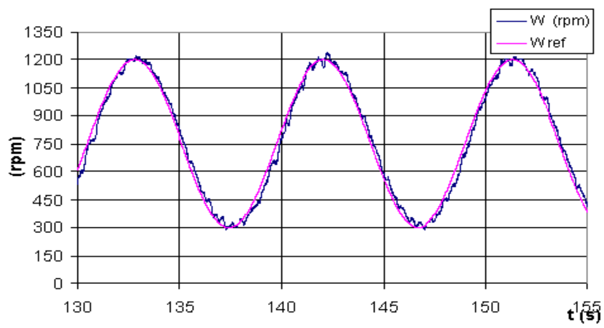


Fig. 6. Speed tracking with sinusoidal reference function

The degradation of the control capability observed by the *MAPE* value increase is inevitable consonant the reference signal variation increases in order to the time. This degradation is a consequence in part of the increasing derivative absolute value of the reference signal.
This is another problem that appears in these control systems due to the necessary learning time for acquire knowledge about the new functioning regimen. This necessary time will be more or less long in function of the algorithm execution, the learning rate, the neuro-fuzzy system net dimension and the type of computer used.

4 Proposed Solutions

In principle, it is considered that it is not possible to change the computer for another one with better processor unit performance. It is also considered that the net dimension of the neuro-fuzzy system is not changed because it will affect the good functioning of the set.

4.1 Rectangular Reference Tracking

The proposed solution for the rectangular reference tracking problem in this work, will not modify the algorithm cycle time, nor the dimension of the neuro-fuzzy net. The learning rate will remain constant, which is very common in these systems. Respecting the imposed conditions, a solution passes acting on the weights matrix of the rules consequent part level generated by the learning algorithm. Although a two dimensional matrix already exists for the effect in (8) called $\rho_{k(i,j)}$, this will not be able to store simultaneously different information of the weight of each rule related to each regimen of the set functioning. Consequently, the proposed solution consists on using one rules consequent weights matrix for each functioning regimen. This means that the system must have a three-dimensional matrix or a multilayer weights matrix. Theses weights matrix, after the learning process, has all the necessary information for each functioning regimen and they are selected depending on the decision block. In this work, this decision block is based on the speed reference value. For example it was generated one empty matrix for the learning process when is detected a new speed reference value step (old reference value is 300 and the new one is 500 rpm). In this moment the weight matrix stay linked to this speed reference value (500 rpm) and always used or call when this last one is desired.
Thus, in the learning expression (13), the rules consequent weights matrices, m, are used in function of the speed reference value (in this in case m = {1, 2}).

$$\rho_{k(m,n,o)} = \rho_{k-1(m,n,0)} + \gamma \frac{\partial E}{\partial \rho_{k-1(m,n,o)}} \qquad (13)$$

4.2 Slopes Reference Tracking

In many cases the previous solution is efficient. However there are situations like concavities, declivities very accented or very random variations of the reference signal to tracking. For these cases the system has difficulty in following exactly that reference signal or need a new structure composed for a lot of new layers of rules consequent weights.
The proposed solution for this particularity is to joint an algorithm to change the learning rate when necessary. In this work it was understood to be necessary to modify the learning rate value when the module of the error percentage value was equal or greater than the objective value. In this case it was 2%. Thus, using this rule, the learning rate value was changed becoming it more dynamic through the application of the expression (14).

$$\gamma_k = \begin{cases} \gamma_{k-1}\left(1 + \lambda Ln(PE)\right) & ; PE \geq 2\% \\ \gamma_d & ; PE < 2\% \end{cases} \quad ;\ \mathrm{k} \in \mathrm{N+} \qquad (14)$$

Where γ_k is the learning rate value used for updating the weights of the consequent rules expressed in (8), γ_d is the default learning rate value gotten and demonstrated in [6], *PE* is the error percentage gotten through the expression (12) and λ is the impulse factor of the logarithm function. The impulse factor value $\in\{0,1\}$ and it was chosen when the learning rate value result, for a maximum *PE* value considered, is such that the system is between the cushioned and the oscillatory reply. This set of conditions support the learning rate adaptation algorithm and was applied to the on-line learning controller in the tracking of reference speed function.

5 Experimental Results

For the first compensating method, the principle was applied to the on-line learning controller in the tracking of a rectangular reference speed function with 300 and 1200 rpm. That implies the use of two rules consequent weights matrices. The first one will be applied in the learning and it is always used when the speed reference is 300 rpm and the second one will be always applied when the speed reference is 1200 rpm. In figure 7 one observes, in the first step, the typical oscillations due to the learning stage. Figure 7 presents the machine speed trajectory, without the oscillations, in next cycles.

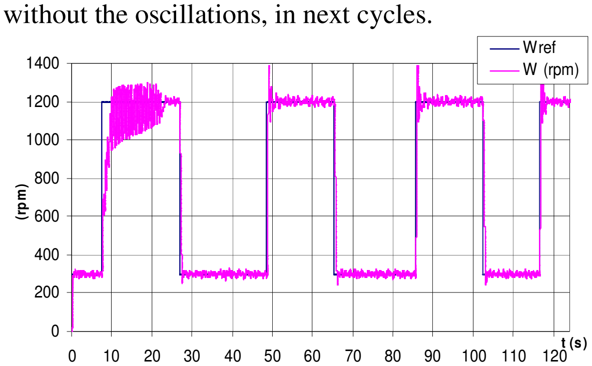


Fig.7: Speed tracking with a rectangular reference function

It is evident that the controller with one weights matrix layer presents, in the first time that the control trajectory is described, bigger easiness in learning the new regimen of functioning due to the use of a base of existing knowledge, but needing some adjustments. While that, in the same situation, the controller with two weights matrix layers delays more to learn the new regimen (compare the first rectangular cycle in figure 4 and figure 7 at 1200 rpm). This is verified due to the second matrix of the weights to be initialized at zero which means without knowledge. In the subsequent cycles the controller with one weights matrix layer continues to have to readjust and to learn the new regimen of functioning, either 300 or 1200 rpm, to the step that the same controlling with two layers of weights is faster in reply for already having acquired knowledge characterized by the stored value of the weights for the two regimes. In the case that it has load variations some small readjustments are needed to joint the already knowledge.

For the second compensating method, first, without the adaptation learning rate algorithm, the controller was submitted to a set of tests. Theses tests were developed applying a speed trapezoidal reference function between 300 and 1200 rpm showed in figure 2, with 10s, 8s, 6s, 4s and 2s of slopes duration each time.

Next the same set of tests was applied to the controller with the adaptation learning rate algorithm (14). The gotten results were summarized in the representation of the *MAPE* in function of the slopes acceleration in figure 8. In dashed-dot line, it is observed the tests results of the *MAPE* with the controller without the adaptation learning rate algorithm and in solid line the course of the *MAPE* representative of the controller performance with the adaptation learning rate algorithm.

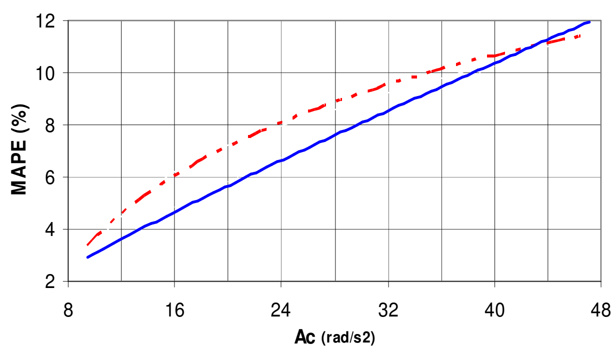


Fig. 8: MAPE factor of the experimental results of the neuro-fuzzy controller without (dashed-dot line) and with (solid line) adaptive learning rate algorithm.

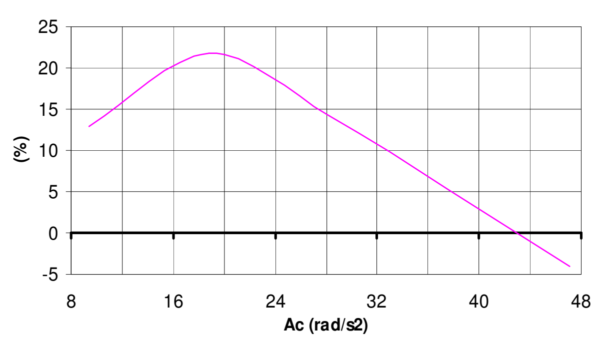


Fig. 9: Percentile improvement of the performance of the controller based on the MAPE.

The maximum improvement is placed in 22% for slopes with 18,8 rad/s^2. However it is observed that this algorithm presents a real profit in the performance of the controller until accelerations of 43 rad/s^2 approximately as is observed in figure 9. Verifying, in this last point, that figure 9 supplies the information of when the adaptation algorithm considered will have to be inhibited to update the learning rate. Better results are possible when conjugating the two functioning ways; constant learning rate versus adaptive learning rate. Above 43 rad/s^2 the performance of the controller is negative because the increases of the increment values of the adaptation learning rate algorithm provoked the sprouting of oscillations in the reply of the controller, raising the *MAPE* value. In figure 8, the course of the *MAPE* results of the controller with the learning rate adaptation is linear. It can be interpreted that the slope of this linear function represents the difficulty to improve the performance of the system materialized for the set of pertaining parameters to the machine and controller in the presented conditions of the assay.

6 Conclusions

The implementation of a neuro-fuzzy system was described. All the five layers of the connectionist model were explained. The learning algorithm, in order to minimize the error function, acts modifying the values of the rules consequents weight and was presented. The parameters of the controller were defined and also explained. Some parameters were fixed while others vary until reach a satisfactory system behavior.

Experimental results, showing the system performance, were presented along a set of various speed reference functions tracking. In this work, it was used two kinds of error measures analysis: the *MPE* and the *MAPE*. It was verified a good performance in the speed tracking of a trapezoidal reference function

It was also presented a solution and its application for the on-line learning SRM speed controller when abrupt reference speed variations is needed and it is necessary to self readjust without compromising the set good performance. This system was applied having into account some conditioning that do not warrant in favour of the controller such as not to be able to break the machine speed, as it was tested in machine functioning regimes situations more adverse, without mechanic load.

It was also presented one solution and its application for the same speed controller when the increase of the reference speed slopes is needed and it is necessary to self readjust without compromising the set good performance. It is shown that the controller, with an adaptive learning rate, presents a better behaviour in learning the new functioning regimen with increase slope rates. This is due to a more quickly learning of the regimen depending of the error percentage.

The continuation of this work will consists on in automatically determining when these two compensation methods must be on individually or simultaneously contributing for a high behaviour performance.

References:

[1] Nicolai J., *Simplified Electronics Bring the Switched Reluctance Motor to the Mass Market,* in EPE Conference'95, 1995, vol 3,pp903, 907.

[2] Miller.T.J.E., *Switched Reluctance Motors and Their Control*, Magna Phisics Publ. And Clarendon Press, Oxford, 1993.

[3] Baltazar P., Rafael S., A.J.Pires, Costa Branco P.J., Obtaining the Magnetic Characteristics of an 8/6-Switched Reluctance Machine: FEM Analysis and Experimental Tests, IEEE International Symposium on Industrial Electronics - ISIE2003, 2003, Rio de Janeiro, ISBN: 0-7803-7912-8

[4] Arabshahi P., R.J. Marks II and T.P. Caudell, *Adaptation of Fuzzy Inferencing :A survey*, Proceedings of the IEEE/Nagoya University WWW on Learning and Adaptive Systems, 1993, pp.1-9, Nagoya University,

[5] Wang L.X. and J.M.Mendel, *Generating Fuzzy Rules by Learning from Example,* IEEE Trans. Syst. Man. Cybern., vol 22, nº 6, 1992, pp 116-132, 1414-1427.

[6] Silviano R, Pires AJ and Costa Branco PJ, *Metodologia de Parametrização de um Controlador Neuro-Fuzzy de Velocidade para uma Máquina de Relutância Variável*, CBA'04 - XV Congresso Brasileiro de Automática, 2004, Gramado, Brasil.