A Fast Adaptive Memetic Algorithm for Online and Offline Control Design of PMSM Drives

Andrea Caponio, *Student Member, IEEE*, Giuseppe Leonardo Cascella, *Student Member, IEEE*, Ferrante Neri, *Student Member, IEEE*, Nadia Salvatore, *Student Member, IEEE*, and Mark Sumner, *Member, IEEE*

***Abstract*—A fast adaptive memetic algorithm (FAMA) is proposed which is used to design the optimal control system for a permanent-magnet synchronous motor. The FAMA is a memetic algorithm with a dynamic parameter setting and two local searchers adaptively launched, either one by one or simultaneously, according to the necessities of the evolution. The FAMA has been tested for both offline and online optimization. The former is based on a simulation of the whole system—control system and plant—using a model obtained through identification tests. The online optimization is model free because each fitness evaluation consists of an experimental test on the real motor drive. The proposed algorithm has been compared with other optimization approaches, and a matching analysis has been carried out offline and online. Excellent results are obtained in terms of optimality, convergence, and algorithmic efficiency. Moreover, the FAMA has given very robust results in the presence of noise in the experimental system.**

***Index Terms*—Electric drives, evolutionary algorithms, local search, memetic algorithms, synchronous motors.**

I. Introduction

According to recent studies [1], [2], the electric drives market is one of the most dynamic business sectors, particularly with respect to permanent-magnet synchronous motors (PMSM) drives, which now account for more than 25% of the total number of motors produced. PMSMs are commonly used for both high-performance applications (actuators, machine tools, and robotics) and for applications where the machine can be integrated into the mechanical load (e.g., fans and pumps). This is due to a number of advantages such as high-power density, efficiency, reduced volume and weight, and reduced maintenance and service requirements. In order to make the most of a motor performance, a very effective control system is needed. Although many possible solutions are available, e.g., nonlinear, adaptive, and intelligent control [3]–[6], the market of electric drives does not justify the expense needed to implement such sophisticated solutions in industrial drives, and the proportional-integral (PI)-based control scheme still remains the more widely adopted solution. Such a propensity is supported by the fact that, although simple, a PI-based control allows achieving of very high performances when optimally designed [7]. The control design, i.e., setting of controller parameters, aims to provide a good system behavior in terms of the response to a speed command, a load disturbance, and measurement noise. This is consist of a multiobjective problem that easily leads to a constrained optimization of a large number of parameters too complex for analytical solution [8]. Consequently, all the standard design techniques are based on a simplified model and analyze system responses to single inputs, providing just suboptimal solutions [9]–[11].

For this reason, there has been an increased interest in robust optimization techniques for electric drives over the last few years for example genetic algorithms (GAs). There are two main ways for controller design and optimization: offline and online [12]. The former is consist of simulating the whole system, control system and plant, using a model obtained from earlier tests. The optimization will then search the best offline solution. Clearly, the better the model accuracy in simulation, the better solution is obtained for the experimental system. Unfortunately, nonlinearities and uncertainties of the system are hard to be accurately modeled, and as consequence, the offline solution needs to be improved by commissioning engineers who have to hand-calibrate the drive by carrying out tedious and time-consuming tests. Online optimization overcomes this problem because it is model free, i.e., each possible solution is experimentally evaluated throughout the optimization. A complete survey about this approach is given in [13], where the suitability of online GAs for robust autotuning is highlighted. Unfortunately, during the online evolution of the control system, the controlled process can be critically stressed by poorly conditioned solutions. For this reason, the successful applications of online GAs are really limited in number.

With regard to electric drives, this approach has been described in [14]–[16]. The first of these papers has proposed the tuning of the PI speed controller of a brushless dc motor drive evaluating the response to a speed reference step through a very simple fitness function. In [15], the potential of online GAs has been better exploited because both the current and speed controllers of a dc motor drive have been optimized. The evolution process tunes the controller parameters and chooses the controller structures (by cascading elementary controllers) using a parallel algorithm. The study in [16] describes a fuzzy logic controller for a dc motor drive which has been genetically designed and compared with a proportional, integral, differential (PID) controller under variable load conditions. With regard to PMSM drives, the study in [17] describes an online GA implementation which optimizes both the gains of PIs and

Manuscript received April 30, 2005; revised August 23, 2005. This paper was recommended by Guest Editor Y. S. Ong.

A. Caponio, G. L. Cascella, F. Neri, and N. Salvatore are with the Dipartimento di Elettrotecnica ed Elettronica, Politecnico di Bari, 70125 Bari, Italy (e-mail: andrea.caponio@libero.it; leo@leocascella.it; neri@deemail.poliba.it; n.salvatore@deemail.poliba.it).

M. Sumner is with the School of Electrical and Electronic Engineering, University of Nottingham, University Park, NG7 2RD Nottingham, U.K. (e-mail: mark.sumner@nottingham.ac.uk).

Digital Object Identifier 10.1109/TSMCB.2006.883271

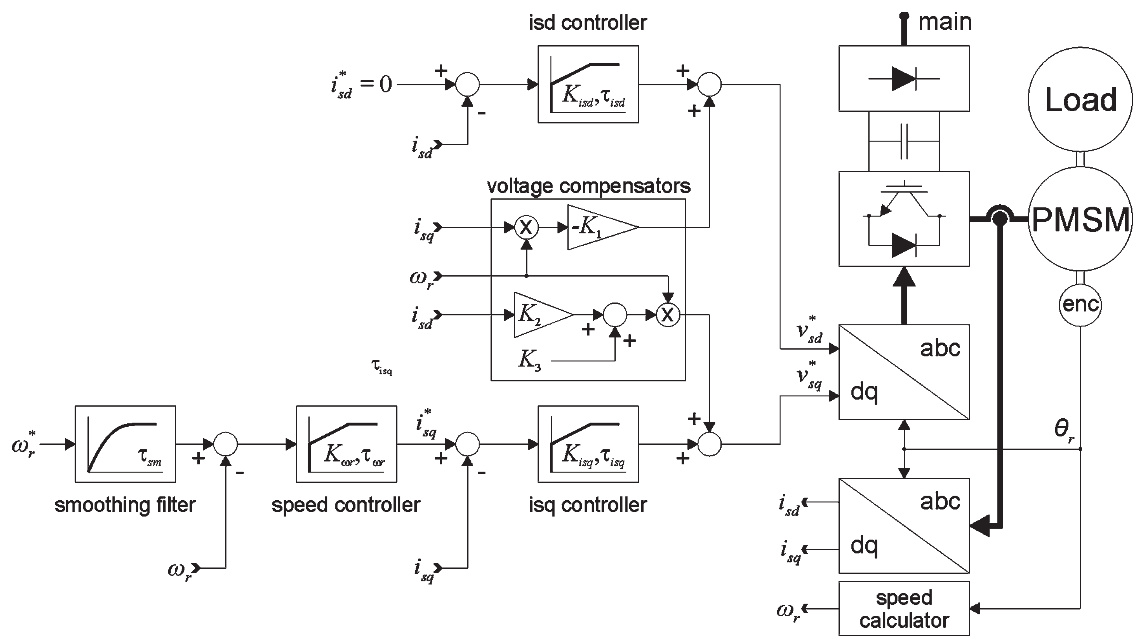


Fig. 1. Block diagram of a vector-controlled PMSM drive.

feedforward compensators of a vector-controlled PMSM drive. The study in [17] describes how to drastically reduce the critical stress due to poorly performing solutions by a real-time fitness evaluation. In [18] and [19], an online approach that combines GA and the "hill-climbing" method has been applied, following the elite-based hybrid architecture proposed in [20]. This combination further reduces the number of the evaluations the GA has to make.

It should be noted that, even if the results obtained have turned out to be satisfactory, in order to avoid the undesirable effects of the premature convergence and stagnation, a proper tuning of the algorithm's parameters such as the size of the population and the probability of the mutation is required. Unfortunately, for this class of problems, the proper parameter settings vary significantly with the state of the evolution, and therefore, a classical GA can easily converge to a suboptimal solution. In order to overcome this problem, a fast adaptive memetic algorithm (FAMA) is proposed in this paper to design the optimal control system of a PMSM. This algorithm adaptively chooses the size of the population and the probability of mutation according to a fitness-based criterion. Moreover, it adaptively "decides" whether none, one, or two heuristic local searches have to be performed. It does this by selecting between two local searchers with different characteristics of pivot rule and neighborhood generating function, choosing the one to be used for the different stages of the search process [21]–[23]. The main idea of the FAMA is that the premature convergence and the stagnation should be avoided by a control on the fitness values taken by the individuals of the population. This control has to balance dynamically the needs of exploration and exploitation, taking into account the state of the evolution [24]. The goal of this adaptive control is achieved by means of the combination of two different algorithmic philosophies: the use of adaptive conventional parameters (e.g., dynamic "aggressiveness" of the mutation probability) and the adaptive use of two local searchers with different structures.

To prove its effectiveness, the proposed FAMA optimizes the gains of the speed, d and q axis current controllers as well as the coefficients of the decoupling actions and the time constant of the smoothing filter, ensuring an optimal control of the PMSM. The objective function to be minimized results from the scaling of a multiobjective problem with several conflicting goals concerning the speed and current responses. The performance of FAMA has been compared with the performance of a standard GA and a heuristic method, and the results show that FAMA, for this class of problem, works better in terms of best solution, algorithmic efficiency, and robustness. The FAMA has outperformed not only the other methods for offline optimization (in simulation) but also the other methods for online optimization (on the real experimental setup). Regarding the latter, the hardware and software implementations are described in detail in order to show how the usual limitations are overcome and how manufacturers can embed this fully automated tool into their drives without any extra hardware.

II. Vector-Controlled PMSM Drive

In Fig. 1, the block diagram of a vector-controlled PMSM drive is shown. This control scheme is based on the dynamic equations of the motor in the rotor flux reference frame (d, q) [11]

$$v_{\mathrm{sd}} = R_s i_{\mathrm{sd}} + L_{\mathrm{sd}} \frac{di_{\mathrm{sd}}}{dt} - \omega_r L_{\mathrm{sq}} i_{\mathrm{sq}} \tag{1}$$

$$v_{\mathrm{sq}} = R_s i_{\mathrm{sq}} + L_{\mathrm{sq}} \frac{di_{\mathrm{sq}}}{dt} + \omega_r \left(L_{\mathrm{sd}} i_{\mathrm{sd}} + \Psi \right) \tag{2}$$

where v_{sd} and v_{sq} are the d and q axis stator voltages; i_{sd} and i_{sq} are the d and q axis stator currents; R_s, L_{sd}, and L_{sq} are the stator resistance, d and q axis inductances; Ψ is the flux linkage of the permanent magnet (PM); and ω_r is the rotor speed. Both the d and q axis voltage references v^*_{sd} and v^*_{sq}

TABLE I
UPPER AND LOWER BOUNDS OF THE DECISION SPACE

i	LB	UB
1	66	600
2	66	200
3	66	600
4	66	200
5	66	600
6	66	200
7	90	400
8	90	400
9	90	400
10	90	400

are provided partly by current controllers and partly by voltage compensators. The d and q axis current controllers are two PIs whose transfer functions are $K_{\text{isd}}(1 + \tau_{\text{isd}}s)/s$ and $K_{\text{isq}}(1 + \tau_{\text{isq}}s)/s$, respectively. The d and q axis voltage compensators, whose feedforward actions are $-\omega_r i_{\text{sq}} K_1$ and $\omega_r(i_{\text{sd}}K_2 + K_3)$, respectively, improve the performance of the current control loops by reducing the influence of the cross coupling terms, i.e., the last terms in (1) and (2). The reference of the q axis current is provided by the speed controller $K_{\omega r}(1 + \tau_{\omega r}s)/s$. The speed reference is prefiltered before being passed to the speed loop to reduce the overshoot and settling associated with step demands while retaining a fast disturbance rejection [10], [11], [25]. This is achieved using a smoothing filter $1/(1 + \tau_{\text{sm}}s)$ whose time constant is tuned according to the speed-loop bandwidth.

As mentioned in the introduction, a model-based design can be adopted in order to achieve a stable control system in a short time. The stable control system so obtained, although a suboptimal solution, will be proved to be useful in the next section. A previous parameter identification can be performed, and the values of K_1, K_2, and K_3 are set equal to the identified values of L_{sq}, L_{sd}, and Ψ, respectively. Moreover, the absolute value optimum (AVO) criterion can be applied to design the current regulators and the symmetrical optimum (SO) to design the speed regulator [25].

III. FORMULATION OF THE PROBLEM

The problem of the self-commissioning can be formulated as the determination of the following ten parameters which guarantee conditions of an optimal control of the PMSM. The decision space $H \subset \mathbb{R}^{10}$ is a ten-dimensional hyperrectangle given by the Cartesian product of the intervals which each parameter is limited to. The upper and lower bounds of each interval have been set as follows:

$$x_{\text{lb}}(i) = x_0(i)\,(1 - \text{LB}(i)/100) \tag{3}$$

$$x_{\text{ub}}(i) = x_0(i)\,(1 + \text{UB}(i)/100) \tag{4}$$

where $\text{LB}(i)$ and $\text{UB}(i)$ are the upper and lower bounds in terms of percentage of $x_0(i)$, and x_0 is the vector of the parameter values of the suboptimal solution obtained by the

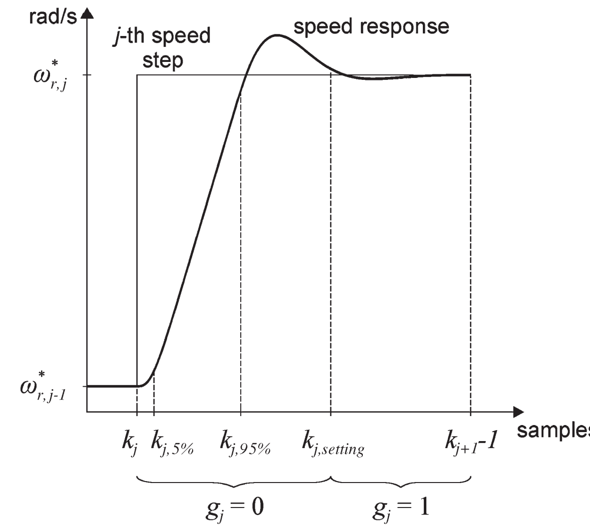


Fig. 2. jth speed step of the training.

initial commissioning calculated in Section II. Typical search ranges would be from ±30% to ±70% of pretuned control parameters [26]; but in order to analyze the problem in a more deeper way, a wider range has been chosen, and the bounds are listed in Table I.

The performance given by each solution is numerically evaluated through the cost objective function following the conventional weighted-sum approach [27]. To optimize the overall response of the drive, the objective function to be minimized is

$$f = \sum_{i=1}^{4} \left(a_i \cdot \sum_{j=1}^{n_{\text{step}}} f_{i,j} \right) \tag{5}$$

where j indicates the number of the generic speed step (the training test is a sequence of the n_{step} speed steps), i indicates the number of the performance index, and a_i is the positive normalization factor of the respective performance index $f_{i,j}$. For more clarity, the jth speed step of the training test is shown in Fig. 2.

The performance index $f_{1,j}$ measures the speed error in the settling phase

$$f_{1,j} = \sum_{k=k_j}^{k_{j+1}-1} \left|\omega_r(k) - \omega^*_{r,j}\right| g_j(k) \tag{6}$$

where $\omega_r(k)$ is the kth stored sample of the rotor speed, and k_j is the sample at which the set point changes to $\omega^*_{r,j}$. The activation function g_j returns zero when the speed reference changes and returns one after $k_{j,\text{set}}$ samples. $k_{j,\text{set}}$ is determined from the time it takes to settle to within 5% of the speed demand, when the original controller (x_0) is used, and remains fixed throughout the optimization. In this way, the optimization is directed toward solutions whose settling time is better than that provided by x_0. In fact, $f_{1,j}$ aims to take account of the steady-state speed error which depends mainly on the nonlinearities and disturbance. If some solution provides a settling slower than x_0, then $f_{1,j}$ will be affected not only by the steady-state speed error but also by the transient speed error, which is numerically bigger.

The overshoot index is

$$f_{2,j} = \left| \frac{\max\limits_{k=k_j,\ldots,k_{j,\text{set}}} \left(\omega_r(k) \cdot \text{sign}\Delta\omega^*_{r,j}\right) - \omega^*_{r,j} \cdot \text{sign}\Delta\omega^*_{r,j}}{\Delta\omega^*_{r,j}} \right| \tag{7}$$

where $\Delta\omega^*_{r,j} = \omega^*_{r,j} - \omega^*_{r,j-1}$ is the amplitude of the jth speed step. It should be noted that (7) works for positive and negative values of $\omega^*_{r,j}$ and $\Delta\omega^*_{r,j}$, and for under- and overdamped responses.

The rise time index is

$$f_{3,j} = \frac{k_{j,95\%} - k_{j,5\%}}{|\Delta\omega^*_{r,j}|} \tag{8}$$

where $k_{j,95\%}$ and $k_{j,5\%}$ are the samples between which the measured speed rises from 5% to 95% of the speed demand.

Finally, the last performance index is

$$f_{4,j} = \sum_{k=k_j}^{k_{j+1}-1} |i_{\text{sd}}(k)| \tag{9}$$

and takes account of the undesired d axis-current oscillations which increase losses and vibrations in the motor and drive.

The online optimization is consist of numerous experiments because each possible solution will be experimentally tested. Because of this, it is possible that an unstable (or highly unsatisfactory) solution may be tested and can severely stress the hardware. While the theoretical guarantees of closed-loop stability on the actual hardware remain open for further research, in this paper, we have adopted a heuristic strategy that effectively overcomes the limitations of model-based analysis. Rather than computing the objective value at the end of each experiment, as is usually done in online tuning approaches, each performance index value is monitored and updated in real time, i.e., at each sample time of each experiment. If during an experiment one of the performance indexes exceeds a predefined warning threshold, the current solution is recognized as "bad." It is immediately replaced by the stable solution x_0 obtained by the initial commissioning, and the motor is stopped. Then, a penalty factor is applied to the objective value according to the rule: "the earlier the bad solution is detected, the higher the penalty factor." In other words, if a bad solution received a fitness equal to $T_{\text{exp}} \cdot f^*/t^*$, where f^* is the fitness scored at the time t^* when the solution is recognized as bad, and T_{exp} is the entire duration of the training test. Afterward, the optimization can go on with the next experiment. The value of each warning threshold can be set between 1.5 and 3 times the value of the respective performance index given by x_0.

IV. Fast Adaptive Memetic Algorithm

In [28], adaptive systems are classified as deterministic, adaptive, or self-adaptive. Following the definition given in [28], a FAMA is proposed here to perform the minimization of the objective function f shown in (5). This FAMA is consist of the following. Initially, a pseudorandom sampling of points of the decision space H is taken. This sample points form a set of vectors x whose genes are the control parameters shown in Fig. 3. Each gene is a real number within the boundaries previously set. Subsequently, at each iteration, crossover and mutation are applied to the population of solutions. The individuals undergoing crossover are selected according to the ranking selection [29], and the blend crossover [30] is then applied. In brief, for two parent solutions x_{par1} and x_{par2}, the blend crossover generates the offspring

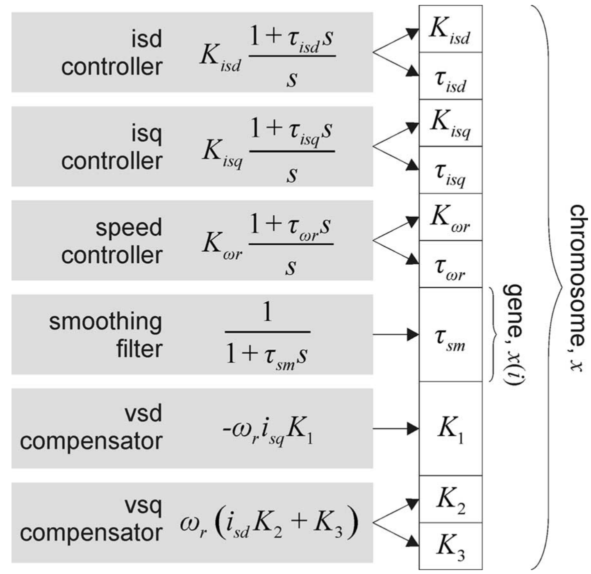


Fig. 3. Parameters to be optimized, encoded as chromosome.

$$x_{\text{off}} = \gamma x_{\text{par1}} + (1-\gamma)x_{\text{par2}} \tag{10}$$

where $\gamma = (1+2\alpha)u - \alpha$, α has been set equal to 0.5, according to the suggestions given in [30], and finally, u is a random number between 0 and 1.

The mutation occurs according to the following formula:

$$x_{\text{mut}} = x + \delta \tag{11}$$

where x is the individual before the mutation, x_{mut} is the mutated individual, and $\delta = (u - 0.5) \cdot (x_{\text{ub}} - x_{\text{lb}})$, where x_{ub} and x_{lb} are the vectors of the upper and lower bounds of the decision space H defined at the beginning of Section III, respectively. To force the x_{mut} in the decision space H, each gene value is saturated if it exceeds its range.

At the end of each iteration, the following coefficient is calculated:

$$\xi = \begin{cases} \left|\frac{f_{\text{best}}-f_{\text{avg}}}{f_{\text{best}}}\right|, & \text{if } \left|\frac{f_{\text{best}}-f_{\text{avg}}}{f_{\text{best}}}\right| \leqslant 1 \\ 1, & \text{if } \left|\frac{f_{\text{best}}-f_{\text{avg}}}{f_{\text{best}}}\right| > 1 \end{cases} \tag{12}$$

where f_{best} and f_{avg} are the best and average fitness at the iteration under study, respectively. Clearly this coefficient ξ is a measurement of the current state of convergence of the algorithm. More specifically, if $\xi = 1$, it means that there is

```
begin-FAMA
  Create initial EA population, S_pop = 200;
  Perform fitness evaluations;
  Define ξ, ξ = min{1, |(f_best − f_avg)/f_best|};
  while (Stopping conditions are not satisfied)
    Perform linear ranking selection with selection pressure of 1.8;
    Perform recombination operation using blend crossover with probability of 0.9;
    Perform mutation operation with probability p_m = 0.4(1 − ξ);
    Perform fitness evaluations on the offspring;
    Merge parents and offspring;
    if (ξ < 0.1) AND (η > 8)
      Perform Hooke-Jeeves search method on the elite individual;
      Replace the elite genotype in the population with the Hooke-Jeeves improved solution;
    end-if
    if (0.05 < ξ < 0.5) AND (η > 4)
      Perform pseudo-random selection of 11 individuals from the population;
      Perform Nelder-Mead simplex search method on the 11 individuals;
      Replace the original genotypes in the population with the Nelder-Mead simplex improved solutions;
    end-if
    Update S_pop, S_pop = S^f_pop + S^v_pop (1 − ξ);
    Survivor selection: Select only the S_pop best performing individuals to proceed in the next generation;
    Update ξ, ξ = min{1, |(f_best − f_avg)/f_best|};
  end-while
end-FAMA
```

Fig. 4. FAMA pseudocode.

a high diversity (in terms of fitness) among the individuals of the population and that the solutions are not exploited enough. On the other hand, if $\xi \approx 0$, it means that the convergence is going to happen, and since this convergence can be premature, a higher search pressure is needed.

The coefficient ξ is therefore used to set dynamically the algorithmic parameters [31], [32] in order to inhibit the premature convergence and stagnation and therefore to guarantee a more robust algorithm. The size of the population of the FAMA is therefore dynamic [33], and it is given by

$$S_{\text{pop}} = S^{f}_{\text{pop}} + S^{v}_{\text{pop}}(1 - \xi) \tag{13}$$

where S^{f}_{pop} is the minimum size of the population deterministically fixed and S^{v}_{pop} is the maximum size of the variable population. If $\xi = 1$, the population contains high diversity, and therefore, a small number of solutions need to be exploited; if $\xi \to 0$, the population is going to converge, and a bigger population size is required to increase the exploration. Following the same idea, also the probability of mutation dynamically depends on the coefficient ξ as follows:

$$p_m = 0.4(1 - \xi). \tag{14}$$

It is important to notice that the number of individuals undergoing mutation is given by round $(p_m S_{\text{pop}})$, and 0.4 is a weight chosen by the authors according to the semiempirical consideration that a mutation occurring on over 40% of the population could be a too explorative action spoiling the genotype of some good solutions.

A $(\mu + \lambda)$ strategy has been chosen. This strategy has turned out to be more efficient in terms of convergence velocity than the (μ, λ) and ordinary elitist strategies for this class of problems and for the implemented adaptation [34], [35].

The FAMA makes use also of two local searchers adaptively selected by a criterion based on the value taken by ξ. Defining η as the number of the current iteration (i.e., the number of the current generation), the conditions for the use of the local searchers are the following.

1) If $(\xi < 0.1)$ and $(\eta > 8)$, the Hooke–Jeeves method [36], [37] is applied at the best individual of the population, and the solution returned by the Hooke–Jeeves method is inserted in the population.
2) If $(0.05 < \xi < 0.5)$ and $(\eta > 4)$, 11 individuals $(=$ the number of decision space dimensions $+ 1)$ of the population are selected pseudorandomly. The Nelder–Mead Simplex [38] is applied to these 11 individuals, and the solutions are inserted in the starting population.

It is worthwhile commenting on this criterion. It is important to notice that both Hooke–Jeeves and Nelder–Mead methods are direct methods. Since our fitness function is nonlinear and does not have an explicit analytical expression depending on the variables to be optimized, an approach that makes use of information concerning the derivatives cannot be applied. It is therefore necessary to use a method which considers the function as a "black box" which can just return a value for a given set of variables. In addition, the two local searchers are structurally different. The Hooke–Jeeves method in fact follows a "more steepest ascent" pivot rule compared to the Nelder–Mead method whose pivot rule is "more greedy ascent" [39]. The two methods are different also according to the "neighborhood generating function" $n(i)$, where this function defines a set of points that can be reached by the application of some move operator to the point i [39]. The Hooke–Jeeves method is characterized by a completely deterministic $n(i)$ function, and if the first exploratory step is small enough (smaller than the width of the basin of attraction), the method definitely converges to the nearest local optimum.

TABLE II
PMSM NAMEPLATE

phase-to-phase resistance	10.4 Ω
phase-to-phase inductance	0.0087 H
voltage constant	0.35 V/(rad/s)
torque constant	0.40 Nm/A
moment of inertia	0.00012 kg · m^2
rated power	350 W
rated speed	4000 rpm

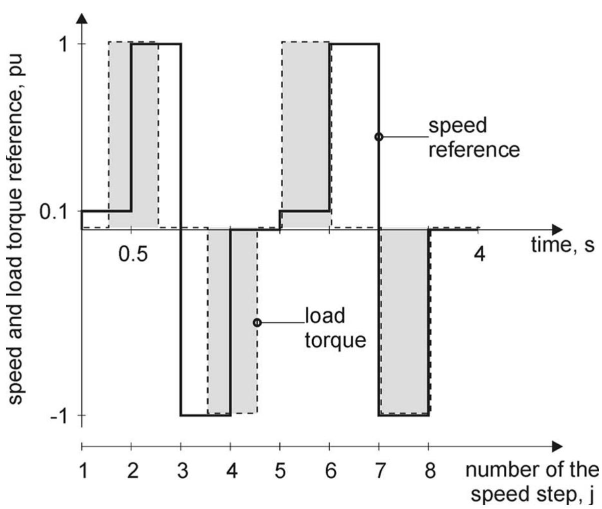


Fig. 5. Training test is a combination of speed commands (a sequence of eight speed steps, continuous line) and load torque commands (dashed line).

TABLE III
GA AND FAMA CONFIGURATION

PARAMETER	GA	FAMA
size of initial population pseudo-randomly generated	200	200
population size for subsequent iterations	200	dynamic between 40 and 160
selective pressure ranking	1.8	1.8
mutation probability	0.3	dynamic between 0 and 0.4
functional evaluations	10000	10000

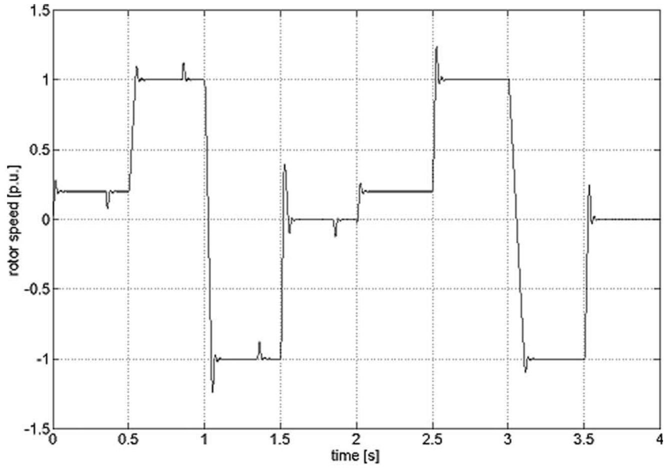
TABLE IV
SIMULATION RESULTS, VALUES OF THE CONTROL SYSTEM PARAMETERS

	Init. Comm.	Simplex	GA	FAMA
K_{isd}	3.9852	4.5624	11.076	11.494
τ_{isd}	0.0016	0.0023	0.0021	0.0022
K_{isq}	3.9852	6.7661	6.6055	6.1892
τ_{isq}	0.0016	0.0005	0.0026	0.0010
$K_{\omega r}$	0.0350	0.0950	0.2370	0.2643
$\tau_{\omega r}$	0.0146	0.0202	0.0116	0.0145
τ_{sm}	0.0037	0.0037	0.0099	0.0110
K_1	0.0025	0.0022	0.0016	0.0021
K_2	0.0025	0.0018	0.0029	0.0006
K_3	0.0750	0.0465	0.0157	0.1901

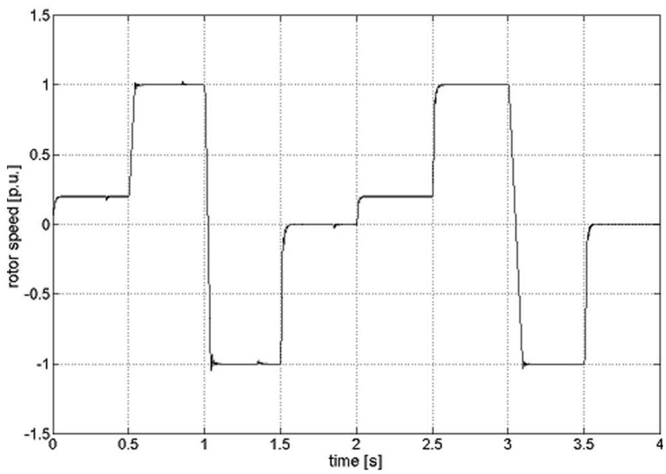
In the case of the Nelder–Mead method as described in 2), the behavior of $n(i)$ function is different because it depends on the initial sampling of the 11 individuals selected from the population. Since this selection is done at pseudorandom, the $n(i)$ function is somehow "randomized," and if the 11 initial individuals are rather spread out in the decision space, the Nelder–Mead method can be quite explorative.

Regarding 1), the idea is that the Hooke–Jeeves method, by means of its steepest ascent pivot rule, should "carefully" analyze the neighborhood of the best solution, when the population is going to converge in order to improve deterministically the results obtained by the evolutionary process. The Hooke–Jeeves should then execute a "lifetime learning" [40] in a Lamarckian logic. The application of the Hooke–Jeeves method after a certain number of iterations is due to the consideration that, during the first iterations, the evolutionary algorithm leads quickly to significant improvement of the initial population and the use of a local searcher, in this phase, would slow down the process without giving important advantages. The choice concerning the eighth iteration is empirically done for the problem under study. The first exploratory step of the Hooke–Jeeves method [41], namely h, has also been set adaptively $h = \xi(x_{\mathrm{ub}} - x_{\mathrm{lb}})$, where x_{ub} and x_{lb} are the vectors of the upper and lower bounds of the decision space H. This choice clearly means that the closer the evolutionary process is to the convergence, the smaller the exploratory steps are. In other words, the algorithm explores the closer solutions as the evolutionary algorithm hones in on the final result.

Concerning 2), the idea is that the Nelder–Mead method should "help" the evolutionary algorithm in a rather wide range of the evolution. The greedy ascent pivot rule should quickly lead to better results, and the structure of the method should bring 11 new individuals who would "refresh" (also by means of shrinking) the population, therefore avoiding the phenomenon of stagnation. The condition $\xi < 0.5$ is due to the consideration that a population with a high diversity does not need this "refresh" and the condition $\xi > 0.05$ is done because the Nelder–Mead method is not efficient anymore if applied to 11 individuals who are very similar. The choice of applying the Nelder–Mead method after a certain number of iterations (the condition $\eta > 4$) is made following the same considerations for the application of the Hooke–Jeeves method. As can be seen from 1) and 2), there is a range $0.05 < \xi < 0.1$ in which both the local searchers can be adaptively launched. If ξ is in this range, it could mean that the population of the solutions is in a basin of attraction [42] and it is probably going to converge in a certain direction. In order to avoid the convergence to a sub-optimal solution and if possible to "jump" to another basin of attraction which leads to the global optimum, the FAMA makes use of both the local searchers to explore the decision space from complementary prospectives [43]. The Hooke–Jeeves and

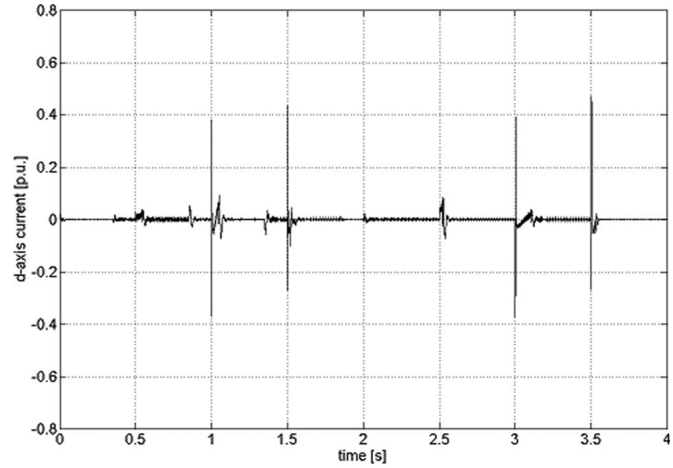


(a)

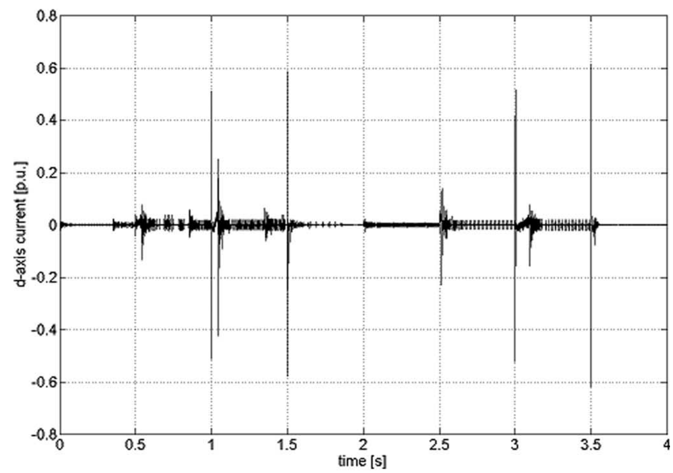


(b)

Fig. 6. Comparison of the speed responses (simulation results). (a) Speed response given by initial commissioning's solution. (b) Speed response given by FAMA's solution.



(a)



(b)

Fig. 7. Comparison of the i_{sd} responses (simulation results). (a) i_{sd} response given by initial commissioning's solution. (b) i_{sd} response given by FAMA's solution.

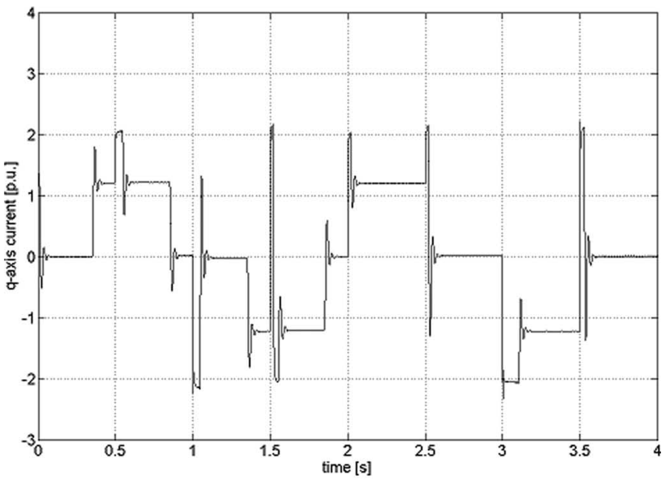
the Nelder–Mead methods are therefore supposed to compete and cooperate [44], fitting the dynamic "necessities" of the algorithm and helping the evolutionary process. The FAMA is stopped when at least one of the following conditions occurs: the number of iteration η reaches a prearranged maximum number of iteration, and the coefficient ξ is smaller than a predetermined value.

The pseudocode which shows the working principle of FAMA is reported in Fig. 4.

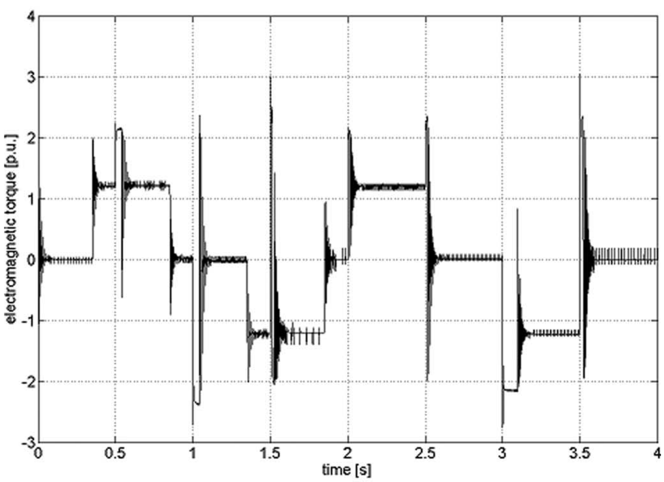
The major limit of the proposed criterion based on the coefficient ξ is that the convergence guess is fitness-based and therefore does not take into account the genotypical distance among the individuals of the population. The consequent limit is that theoretically the FAMA could converge simultaneously to several points with different genotype but the same fitness value. On the other hand, this limit is just theoretical because of the following two reasons. The first is that the fitness function is built on in a way, such that it should not have saddle points in functional landscape. In addition, it should not have several minima which take the same fitness value. The second is that, although several points take the same fitness value, after few iterations, the population would tend to "choose" just one minimum and the whole set of individuals would converge on it [45]. In this light, the ξ-based criterion has been chosen because it is very simple to implement and is suitable and reliable for this class of problems.

V. Numerical Results

For this research, the experimental test facility is constructed from a Technosoft ACPM750 three-phase inverter, a 350-W three-phase PMSM (Table II), and a 250-pulse incremental encoder. A second torque-controlled PMSM has been used for the load. The training test shown in Fig. 5 is proposed, where there are $n_{\mathrm{step}} = 8$ speed steps. It should be noted that the combination of the speed and load torque commands makes it a general-purpose test sequence. Indeed, in addition to low and high speeds, zero speed and speed reversal are considered at no-load and full-load torques. In some applications, it will not be possible to change the load directly, as in this test. However, in many applications (e.g., fan and pump), a change in load will result from a change in speed, and a series of speed changes may prove to be a sufficient training test.



(a)



(b)

Fig. 8. Comparison of the electromagnetic torque responses (simulation results). (a) Electromagnetic torque response given by initial commissioning's solution. (b) Electromagnetic torque response given by FAMA's solution.

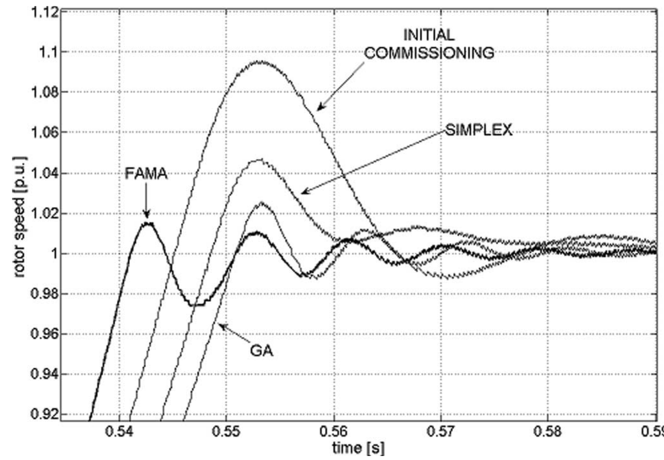


Fig. 9. Comparison of the speed responses to the second speed step (simulation results).

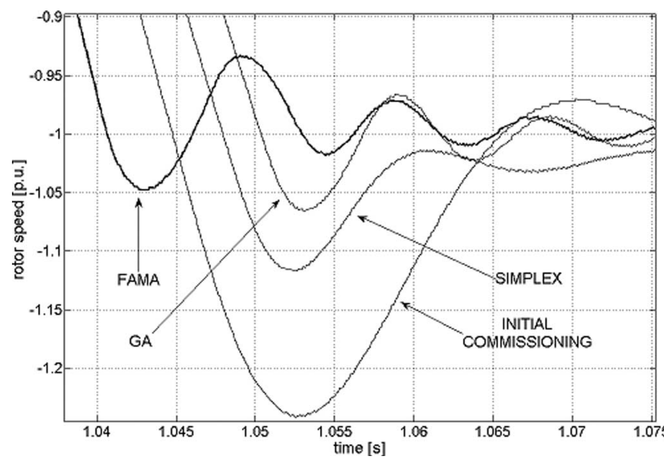


Fig. 10. Comparison of the speed responses to the no-load speed reversal (simulation results).

The FAMA has been tested on the training test described above, and its behavior has been compared with the Simplex method implemented by fminsearch, function of the Matlab Optimization Toolbox, and with a standard GA. The chosen Simplex initial guess is the test from the initial commissioning, and the maximum number of functional evaluations has been set equal to 10 000. Table III summarizes the parameter settings adopted for the GA and FAMA.

A. Offline Results

This section shows the results carried out with the offline approach (see Fig. 14). After a conventional output-error identification [46], the modeled system has been simulated in Matlab/Simulink. Initially, the control parameters has been set by applying the AVO and SO criteria, and the control system has proven to be stable but with a poor dynamic response. To improve the performance of the control system, a further calibration has been carried out following the suggestions in [25]. Essentially, we have retraced the typical steps of an engineer commissioning an industrial drive. This set of the control system parameters, i.e., the solution of the initial commissioning, is reported in the first column of Table IV. The speed response to the training test [see Fig. 6(a)] shows large overshoots, particularly for the first speed reversal. A simple way to reduce the overshoots would have consisted of decreasing the speed dynamics with a consequent deterioration of the other objectives of the speed response. The speed response given by FAMAs solution is much better because the overshoots are reduced without any deterioration of the rise and settling times, as shown in Fig. 6(b). Regarding the last objective, which is the amplitude of the d axis current oscillations, both the responses are good, as shown in Fig. 7(a) and (b). For a fair comparison, the same transient torque limits, equal to 2.2 times the rated torque, have been set, as shown by the torque responses in Fig. 8(a) and (b).

Although the initial commissioning's solution has been outperformed by other optimization techniques, our FAMA has proved to be the best, as shown in Figs. 9–12, where just the zoom details of the most critical points of the training test are shown for the sake of clarity. It has to be noted that FAMA solution ensures a better rise time in spite of a smaller overshoot and shorter settling time. These results are confirmed both when there is a change in speed or torque (see Figs. 9–11) and when the speed and torque commands are simultaneously applied, as shown in Fig. 12.

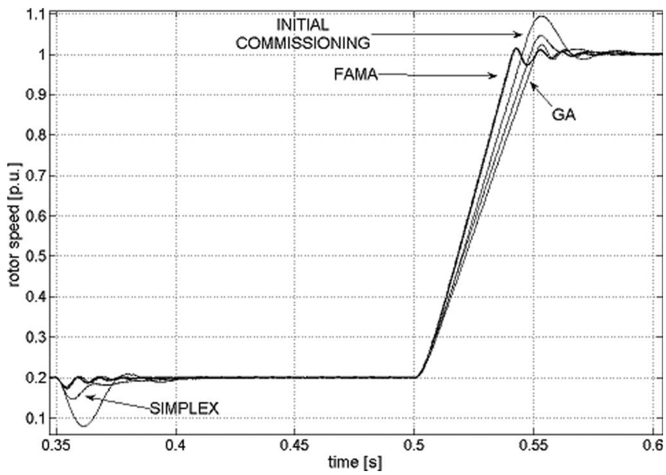


Fig. 11. Comparison of the speed responses. At low speed, the load torque is applied, and then, the motor is operated at the rated speed (simulation results).

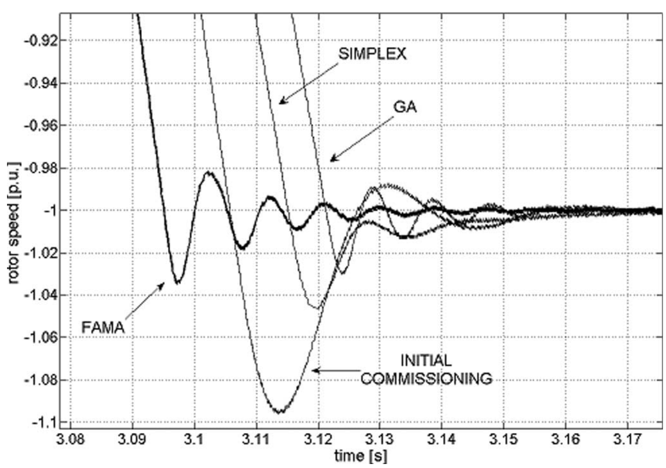


Fig. 12. Comparison of the speed responses to the speed reversal when the load torque is simultaneously applied (simulation results).

TABLE V
SIMULATION RESULTS, VALUES OF THE MULTIOBJECTIVE FUNCTION AND ITS COMPONENTS

	a_1 $[10^{-5}]$	$\cdot$ f_1 $[10^{3}]$	$+$ a_2 $[10^{2}]$	$\cdot$ f_2 $[10^{-4}]$	$+$ a_3	$\cdot$ f_3	$+$ a_4 $[10^{-4}]$	$\cdot$ f_4 $[10^{2}]$	$=$ f
Init. Comm.	5.49	·142.60	+13.16	· 3.10	+3.11	·1.940	+ 6.25	·11.27	=14.97
Sim.	5.49	· 73.18	+13.16	· 3.19	+3.11	·0.985	+ 6.25	·14.67	= 8.42
GA	5.49	· 35.41	+13.16	· 4.44	+3.11	·0.112	+ 6.25	·11.27	= 3.58
FAMA	5.49	· 32.27	+13.16	· 3.07	+3.11	·0.082	+ 6.25	·11.00	= 3.12

However, for a numerical evaluation of FAMA's superiority, Table IV reports the best results obtained by the FAMA and the other methods. It is clear that the Simplex has been "trapped" in a local minimum close to the solution of the initial commissioning approach. This proves that our objective function is multimodal, and therefore, a classical approach of optimization is inadequate.

Table V shows the values of each objective function and of the scaled multiobjective fitness for the initial commissioning and the results given by the three optimization methods. In order to compare each terms of the multiobjective fitness $a_i \cdot f_i$,

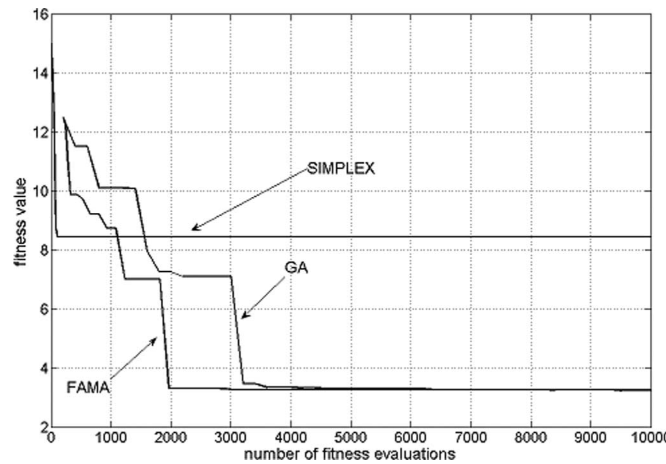


Fig. 13. Convergence of three optimization methods (simulation results).

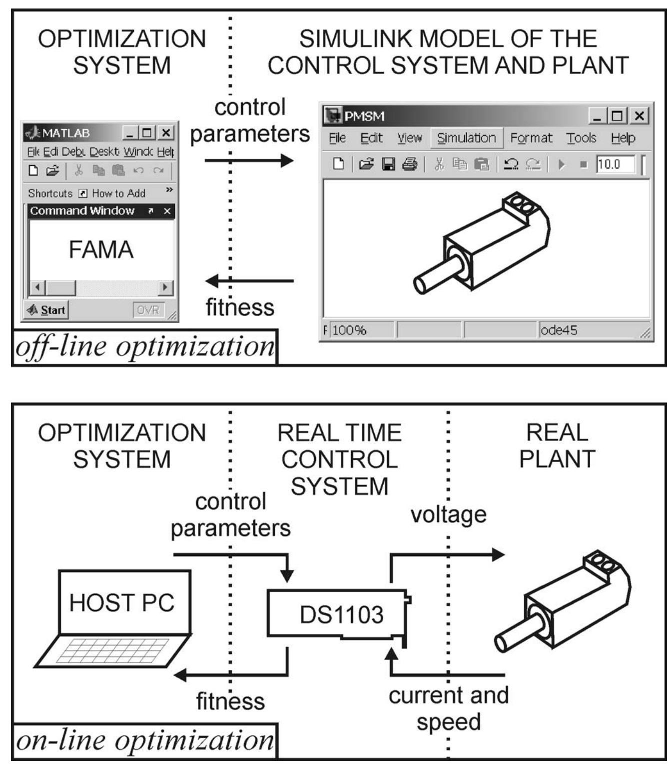


Fig. 14. Offline and online optimization.

each weight a_i has been chosen according to preliminary tests and *a priori* knowledge of the system.

Both the solution obtained by the Simplex and the GA are better than the solution obtained from initial commissioning according to the fitness f, but they do not dominate it. More specifically, for both the Simplex and the GA, the values taken by f_2 are worse than the solution of the initial commissioning, and for the Simplex, the value taken by f_4 is worse than the solution from initial commissioning. On the other hand, the solution given by the FAMA strictly dominates not only the solution from initial commissioning but also all the results given by the other methods.

Each algorithm has been run 50 times. In Fig. 13, a graphical representation of the algorithmic convergence is shown. The mean values of the best fitness per iteration have been

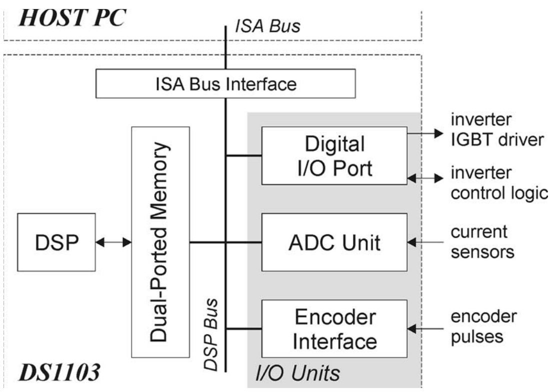


Fig. 15. Scheme of the hardware implementation.

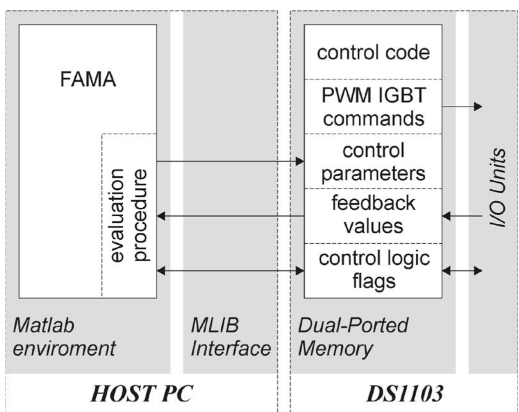


Fig. 16. Scheme of the software implementation.

TABLE VI
EXPERIMENTAL RESULTS, VALUES OF THE CONTROL SYSTEM PARAMETERS

	Init. Comm.	GA	FAMA
K_{isd}	3.9852	18.1552	21.9159
τ_{isd}	0.0016	0.0023	0.0035
K_{isq}	3.9852	23.6281	21.9391
τ_{isq}	0.0016	0.0010	0.0024
$K_{\omega r}$	0.0350	0.1518	0.1665
$\tau_{\omega r}$	0.0146	0.0093	0.0085
τ_{sm}	0.0037	0.0088	0.0089
K_1	0.0025	0.0068	0.0065
K_2	0.0025	0.0038	0.0080
K_3	0.0750	0.2027	0.0307

calculated. It has to be noted that throughout the optimization, each iteration of FAMA can take different functional evaluations. The same is for simplex, but not for GA, where each population is made up of a fixed number of individuals. For a comparison, Fig. 13 shows the mean values of the best fitness versus fitness evaluations. As highlighted above, the Simplex quickly reaches a local minimum and subsequently no further improvements are made. The trends related to the GA and the FAMA start from 200 because, for both, the initial sampling is made up of 200 individuals. Although the initial sampling for the FAMA is the same with the one for the GA, the convergence velocity of the FAMA is higher. In fact, the FAMA reaches a fitness value rather near to the final value, after

TABLE VII
EXPERIMENTAL RESULTS, VALUES OF THE MULTIOBJECTIVE FUNCTION AND ITS COMPONENTS

	a_1	· f_1	+ a_2	· f_2	+ a_3	· f_3	+ a_4	· f_4	= f
	$[10^{-5}]$	$[10^{3}]$	$[10^{2}]$	$[10^{-4}]$			$[10^{-4}]$	$[10^{2}]$	
Init. Comm.	5.49	·49.16	+13.16	· 6.12	+3.11	·2.073	+ 6.25	·20.76	=11.25
GA	5.49	·18.73	+13.16	· 7.47	+3.11	·0.190	+ 6.25	·12.91	= 3.40
FAMA	5.49	·18.55	+13.16	· 6.56	+3.11	·0.187	+ 6.25	·12.47	= 3.24

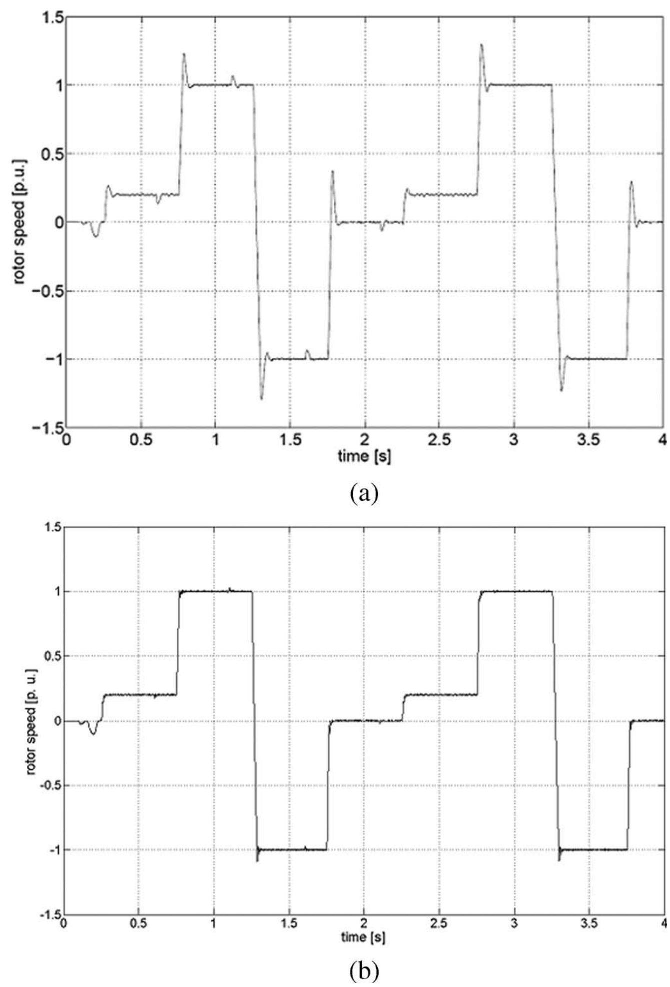


Fig. 17. Comparison of the speed responses (experimental results). (a) Speed response given by initial commissioning's solution. (b) Speed response given by FAMA's solution.

approximately 2000 iterations. The GA on the other hand needs more than 3200 iterations. After this number of iterations, the fitness values of the FAMA and GA are very similar, and the improvements are slow for both the GA and FAMA. However, for each point of the trend shown in Fig. 13, the FAMA leads to better results. The mean and best values results obtained in simulation are practically the same, as can be seen from Fig. 13 and Table V; in other words, after 10 000 evaluations, we found a percentage standard deviation lower than 0.1%.

B. Experimental Results

The implementation of the online optimization is based on a host PC and a multipurpose single-board dSpace DS1103 [47],

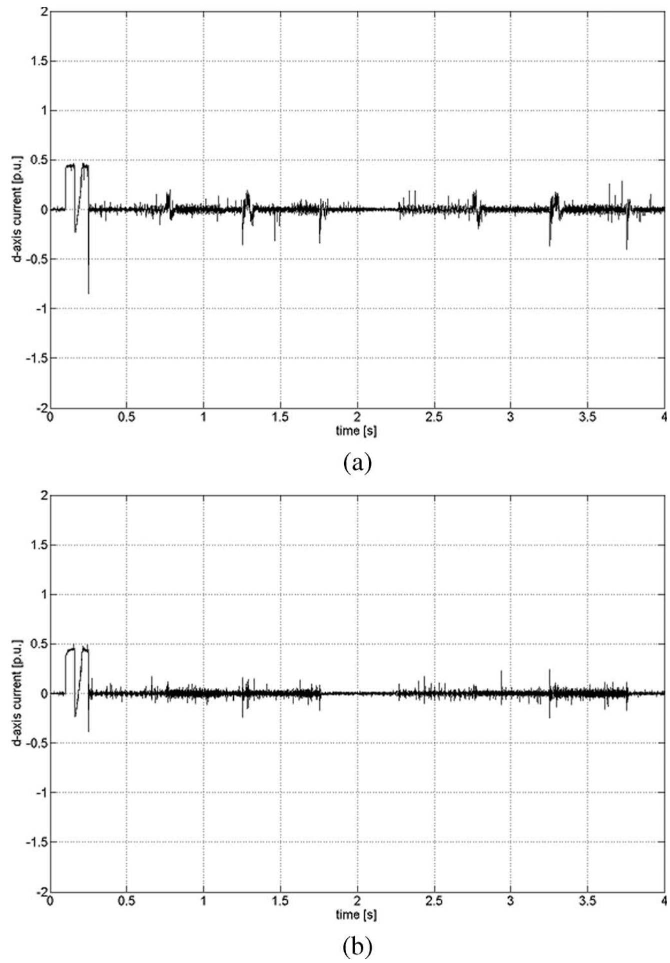


Fig. 18. Comparison of the i_{sd} responses (experimental results). (a) i_{sd} response given by online initial commissioning's solution. (b) i_{sd} response given by online FAMA's solution.

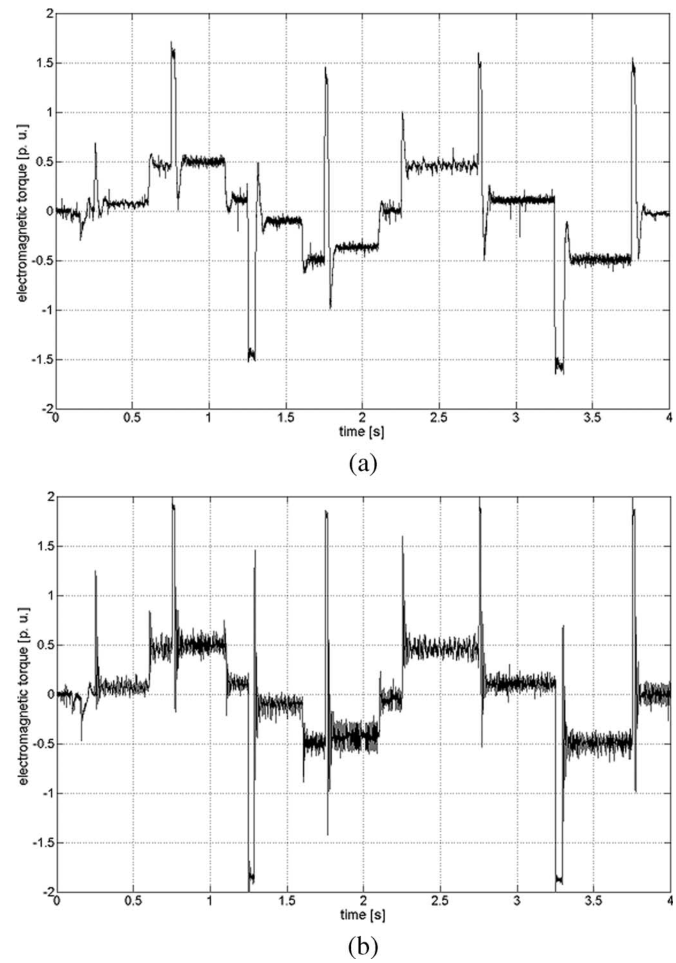


Fig. 19. Comparison of the electromagnetic torque responses (experimental results). (a) Electromagnetic torque response given by initial commissioning' s solution. (b) Online electromagnetic torque responsse given by FAMA's solution.

as shown in Fig. 14. Although the DS1103 has a floating-point PowerPC 604e microprocessor working in conjunction with a TMS320F240 fixed-point DSP, the control platform used in this project employs the fixed-point DSP only, so that the control hardware is equivalent to that of a typical industrial drive, as shown in Fig. 15. The DSP executes the code in the dual-ported memory that can also be accessed by the host PC and the I/O units through the DSP bus. The digital I/O port is consist of two eight-bit I/O configurable ports through which the pulsewidth modulation (PWM) commands and control signals are sent to the inverter. The feedback from the current sensors is converted with an A/D 14-bit converter. Finally, the feedback from the encoder is handled by the dedicated encoder interface. The vector-control program has been developed in a Simulink environment and has been compiled by the dSpace code generator. The real-time executable code is downloaded to the DSP memory, whose simplified map is shown in Fig. 16. During the normal working of the drive, the vector-control code is executed by the DSP which updates the outputs while the I/O units update the feedback values. The control parameters remain static, and the control logic flags are used to synchronize the processes and to handle the faults. During the autotuning, the evaluation procedure of the algorithms (GA and FAMA) changes the control parameters and analyzes the response to the training test set. This is performed by accessing the memory locations where the control parameters and the feedback values are stored using the dSpace MLIB Interface [48]. In this way, the autotuning procedure does not need to recompile the control code. Furthermore, the evaluation procedure uses the control flags to make the autotuning independent of the training test time and the DSP execution code time.

It should be emphasized that the dSpace board and the host PC have been used to reduce prototyping time. No extra hardware has been used for the vector-control implementation compared with an industrial drive. Moreover, the GA code has been implemented in Matlab environment by using the GA Toolbox [49], but for this application, it can be optimized to take up a small part of the DSP memory. Hence, this solution can be simply embedded into industrial drive software as a self-commissioning tool.

Table VI shows the best solutions obtained online for both the GA and FAMA. The results related to the Simplex method have not been included because they have turned out to be completely unreliable. This was because of the functional local minima, as shown in the offline section, and also because of the noise introduced by the measurement instrument. As can be seen, the online training results are rather different when

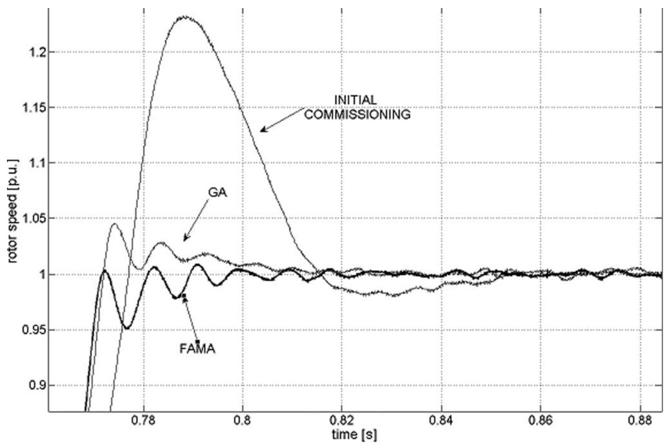


Fig. 20. Comparison of the online speed responses to the second speed step (experimental results).

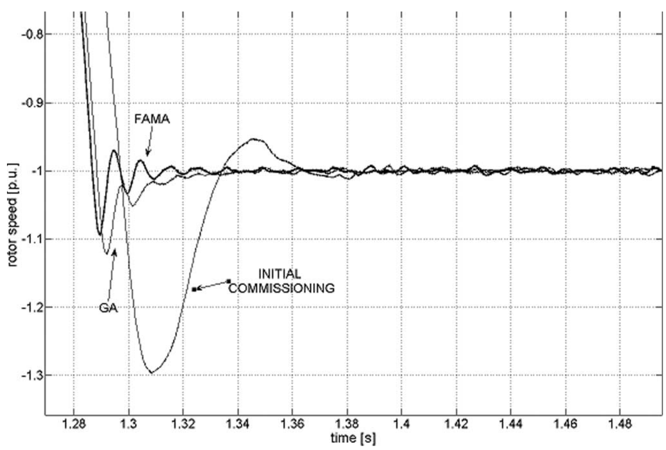


Fig. 21. Comparison of the speed responses to the no-load speed reversal (experimental results).

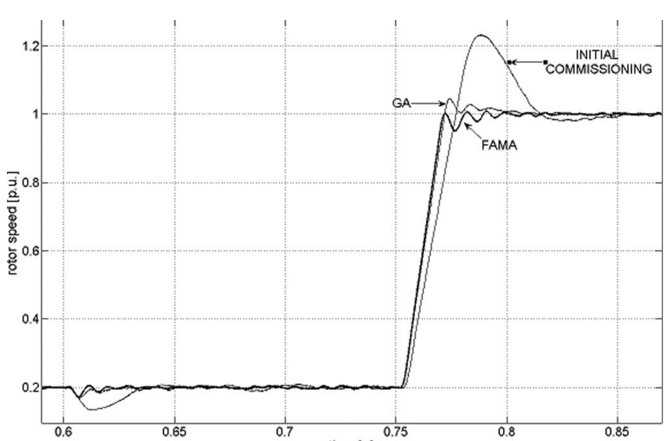


Fig. 22. Comparison of the speed responses. At lowspeed, the load torque is applied, and then, the motor is operated at the rated speed (experimental results).

compared to the offline training results. This was expected because of the unavoidable nonlinearities and uncertainties of the system which are very difficult to model [12] and therefore proves the importance of the experimental approach for this class of problem. In Table VII, the best fitness of each objective function is reported; also in this case, the solution obtained by

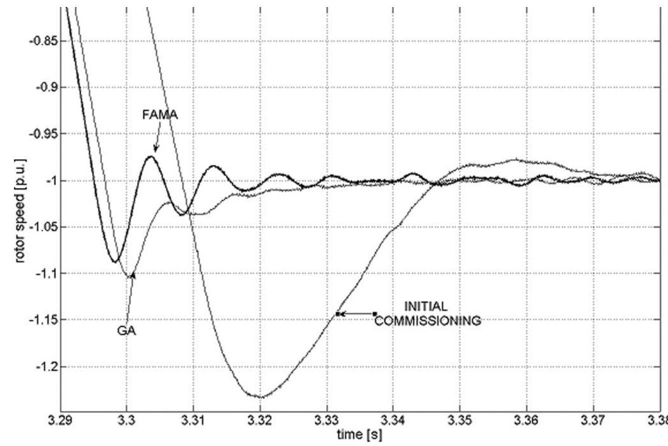


Fig. 23. Comparison of the speed responses to the speed reversal when the load torque is simultaneously applied (experimental results).

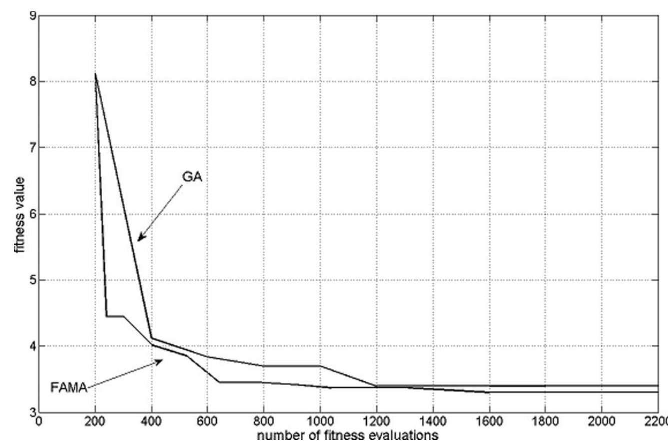


Fig. 24. Convergence of the FAMA and the GA (experimental results).

TABLE VIII
MEAN VALUES AND PERCENTAGE STANDARD DEVIATIONS OF FITNESS AMONG THE 50 EXPERIMENTAL RUNS

	Mean Value	Percentage Standard Deviation
GA	3.56	12.4%
FAMA	3.33	8.9%

FAMA again strictly dominates both the solutions obtained by the initial commissioning and the GA. The comparison of the responses is shown in Figs. 17–23.

Concerning the convergence velocity, the FAMA outperforms the GA, as shown in Fig. 24. Both the FAMA and GA have been run 50 times. Unlike simulation, the experimental fitness evaluations are affected by uncertainties due to measurement, and Table VIII shows the mean values and the standard deviations of the results obtained analyzing the 50 runs.

VI. CONCLUSION

In this paper, an original adaptive memetic algorithm is proposed. The fast adaptive MA has been successfully applied to the control design of PMSM drives. To the authors' knowledge, a memetic algorithm has never been applied to this class of problem. A comparison with other well-known approaches present in the literature has therefore been presented. This comparison shows that the FAMA overcomes the problem of the convergence to a local optimum, which is typical of the deterministic algorithms. Moreover, thanks to its adaptivity and to its two local searchers with different features, the FAMA outperforms a classical GA. Numerical results show that, after a large number of iterations, the solution given by the FAMA is slightly better than the one given by the GA, and the FAMA converged significantly faster than the classical GA.

This result is obtained for both offline and online optimizations. The latter is particularly difficult due to nonlinearities and uncertainties of the real world application. The measurement, in particular, introduces a noise in each fitness evaluation. In this noisy environment, the FAMA behaved better than the classical GA in terms of robustness of the algorithm.

REFERENCES

[1] Company News From Frost and Sullivan, Electric Drives Market Prove to be Dynamic. (Feb. 2001). [Online]. Available: http://www.manufacturingtalk.com

[2] News From Control Engineering, Economic Turnaround to Boost Europe's Electric Drives Market. (Sep. 2003). [Online]. Available: http://www.manufacturing.net/ctl

[3] K. J. Åström and B. Wittenmark, *Adaptive Control*, 2nd ed. Reading, MA: Addison-Wesley, 1995.

[4] J.-J. Slotine and W. Li, *Applied Nonlinear Control*. Englewood Cliffs, NJ: Pearson Education, 1991.

[5] F. Khorrami, P. Krishnamurthy, and H. Melkote, *Modeling and Adaptive Nonlinear Control of Electric Motors*. Heidelberg, Germany: Springer-Verlag, Aug. 2003.

[6] L. C. Jain, C. W. De Silva, and L. C. Jain, *Intelligent Adaptive Control: Industrial Applications*. Boca Raton, FL: CRC, Dec. 1998.

[7] K. J. Åström and T. Hägglund, "The future of PID control," *Control Eng. Pract.*, vol. 9, no. 11, pp. 1163–1175, Nov. 2001.

[8] H. Panagopoulos, K. J. Åström, and T. Hägglund, "Design of PID controllers based on constrained optimization," in *Proc. ACC*, San Diego, CA, Jun. 1999, pp. 3858–3862.

[9] B. Drury, *The Control Techniques Drives and Controls Handbook*, vol. 35. Cambridge, U.K.: Cambridge Univ. Press, 2001.

[10] W. Leonhard, *Control of Electrical Drives*, 2nd ed. Berlin, Germany: Springer-Verlag, 1996.

[11] R. Krishnan, *Electronic Motor Drives: Modeling, Analysis and Control*. Upper Saddle River, NJ: Prentice-Hall, Feb. 2001.

[12] P. Zanchetta, M. Sumner, F. Cupertino, M. Marinelli, and E. Mininno, "On-line and off-line control design in power electronics and drives using genetic algorithms," in *Proc. 11th IEEE CAIA*. Los Angeles, CA, Feb. 1995, pp. 277–283.

[13] P. J. Fleming and R. C. Purshouse, "Evolutionary algorithms in control systems engineering: A survey," *Control Eng. Prac.*, vol. 10, no. 11, pp. 1223–1241, Nov. 2002.

[14] W. G. da Silva, P. P. Acarnley, and J. W. Finch, "Application of genetic algorithms to the online tuning of electric drive speed controllers," *IEEE Trans. Ind. Electron.*, vol. 47, no. 1, pp. 217–219, Feb. 2000.

[15] F. Cupertino, E. Mininno, D. Naso, B. Turchiano, and L. Salvatore, "On-line genetic design of anti-windup unstructured controllers for electric drives with variable load," *IEEE Trans. Evol. Comput.*, vol. 8, no. 4, pp. 347–364, Aug. 2004.

[16] F. Cupertino, V. Giordano, D. Naso, B. Turchiano, and L. Salvatore, "On-line genetic design of fuzzy controllers for dc drives with variable load," *Electron. Lett.*, vol. 39, no. 5, pp. 479–480, Mar. 2003.

[17] M. Sumner, G. L. Cascella, N. Salvatore, and F. Cupertino, "Improvements to self-commissioning of PMSM drives with on-line genetic algorithms," *IEEE Trans. Power Electron.* submitted for publication.

[18] G. L. Cascella, "Online hybrid evolutionary algorithms for self-commissioning of electric drives," Ph.D. dissertation, Politecnico di Bari, Bari, Italy, 2005.

[19] G. L. Cascella, N. Salvatore, M. Sumner, and L. Salvatore, "On-line simplex-genetic algorithm for self-commissioning of electric drives," in *Proc. 11th EPE*, Dresden, Germany, Sep. 2005, pp. 277–283.

[20] J. Yen, J. C. Liao, B. Lee, and D. Randolph, "A hybrid approach to modeling metabolic systems using genetic algorithms and the simplex method," *IEEE Trans. Syst., Man, Cybern., B: Cybern.*, vol. 28, no. 2, pp. 112–147, Apr. 1998.

[21] N. Krasnogor and J. Smith, "A memetic algorithm with self-adaptive local search: TSP as a case study," in *Proc. GECCO*, D. Whitley, D. Goldberg, E. Cantu-Paz, L. Spector, I. Parmee, and H.-G. Beyer, Eds. Las Vegas, NV: Morgan Kaufmann, 2000, pp. 987–994.

[22] Z. Ning, Y. S. Ong, K. W. Wong, and M. H. Lim, "Choice of memes in memetic algorithm," in *Proc. 2nd Int. Conf. CIRAS, Special Session Optimization Using Genetic, Evolutionary, Social and Behavioral Algorithms*, Singapore, Dec. 2003. CD-ROM.

[23] Y. S. Ong, M. H. Lim, N. Zhu, and K. W. Wong, "Classification of adaptive memetic algorithms: A comparative study," *IEEE Trans. Syst., Man Cybern., B, Cybern.*, vol. 36, no. 1, pp. 141–152, Feb. 2006.

[24] H. P. Schwefel, *System Evolution and Optimum Seeking*. New York: Wiley, 1995.

[25] J. H. H. Gross and G. Wiegärtner, *Electrical Feed Drives in Automation: Basics, Computation, Dimensioning*. Munich, Germany: Siemens Aktiengesellschaft, 2001.

[26] P. Schroder, B. Green, N. Grum, and P. J. Fleming, "On-line evolution of robust control systems: An industrial active magnetic bearing application," *Control Eng. Pract.*, vol. 9, no. 1, pp. 37–49, Jan. 2001.

[27] K. Miettinen, *Nonlinear Multiobjective Optimization*. Berlin, Germany: Springer-Verlag, 1998.

[28] J. E. Smith and T. Fogarthy, "Operator and parameter adaptation in genetic algorithms," *Soft Comput.*, vol. 1, no. 2, pp. 81–87, Jun. 1997.

[29] D. Whitley, "The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best," in *Proc. 3rd Int. Conf. Genetic Algorithms*, 1989, pp. 116–121.

[30] L. J. Eshelman and J. D. Shaffer, "Real-coded genetic algorithms and interval-schemata," in *Foundations of Genetic Algorithms 2 (FOGA-2)*. San Mateo, CA: Morgan Kaufmann, 1993, pp. 187–202.

[31] T. Bäck, "The interaction rate of mutation rate, selection, and self-adaptation within a genetic algorithm," in *Proc. PPSN-II*, 1992, pp. 85–94.

[32] A. E. Eiben, R. Hinterding, and Z. Michaelwicz, "Parameter control," in *Evolutionary Computation, Vol. 2, Advanced Algorithms and Operators*, T. Baeck, D. B. Fogel, and Z. Michaelwicz, Eds. Goldberg, DE: Inst. Phys. Publishing, 2000, pp. 170–187.

[33] J. Arabas, Z. Michalewicz, and J. Mulawka, "Gavaps—A genetic algorithm with varying population size," in *Proc. 1st IEEE Conf. Evolutionary Computation, and IEEE World Congr. Comput. Intell.*, Jun. 1994, vol. 1, pp. 73–78.

[34] L. J. Eshelman, "The CHC adaptive search algorithm: How to have safe search when engaging in nontraditional genetic recombination," in *Foundations of Genetic Algorithms 1 (FOGA-1)*. San Mateo, CA: Morgan Kaufmann, 1991, pp. 263–283.

[35] D. K. Gehlhaar and D. B. Fogel, "Tuning evolutionary programming for conformationally flexible molecular docking," in *Proc. 5th Annu. Conf. Evolutionary Programming*, T. B. L. J. Fogel and P. J. Angeline, Eds. Cambridge, MA: MIT Press, 1996, pp. 419–429.

[36] R. Hooke and T. A. Jeeves, "Direct search solution of numerical and statistical problems," *J. ACM*, vol. 8, no. 2, pp. 212–229, Apr. 1961.

[37] F. Kaupe, Jr., "Algorithm 178: Direct search," *Commun. ACM*, vol. 6, no. 6, pp. 313–314, Jun. 1963.

[38] A. Nelder and R. Mead, "A simplex method for function optimization," *Comput. J.*, vol. 7, pp. 308–313, 1965.

[39] W. E. Hart, N. Krasnogor, and J. E. Smith, "Memetic evolutionary algorithms," in *Recent Advances in Memetic Algorithms*, W. E. Hart, N. Krasnogor, and J. E. Smith, Eds. Berlin, Germany: Springer-Verlag, 2004, pp. 3–27.

[40] A. E. Eiben and J. E. Smith, "Hybrid evolutionary algorithms," *Introduction to Evolutionary Computing, Hybridisation With Other Techniques: Memetic Algorithms*, 2003. Slides of the Lecture Notes. [Online]. Available: http://www.cems.uwe.ac.uk/~jsmith/ecbook/ecbook-course.html

[41] C. T. Kelley, *Iterative Methods of Optimization*. Philadelphia, PA: SIAM, 1999, pp. 212–229.

[42] O. Becker and M. Karplus, "The topology of multidimensional potential energy surfaces: Theory and application to peptide structure and kinetics," *J. Chem. Phys.*, vol. 106, no. 4, pp. 1495–1517, Jan. 1997.

[43] N. Krasnogor, "Toward robust memetic algorithms," in *Recent Advances in Memetic Algorithms*, W. E. Hart, N. Krasnogor, and J. E. Smith, Eds. Berlin, Germany: Springer-Verlag, 2004, pp. 185–207.

[44] Y. S. Ong and A. J. Keane, "Meta-Lamarkian learning in memetic algorithms," *IEEE Trans. Evol. Comput.*, vol. 8, no. 2, pp. 99–110, Apr. 2004.

[45] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms*. Chichester, U.K.: Wiley, 2001, pp. 147–149.

[46] L. Ljung, *System Identification: A Theory for the User*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 1992.
[47] *DS1103 PPC Controller Board Hardware Reference*. Paderborne, Germany: dSPACE GmbH, 2003.
[48] *MATLAB-dSPACE Interface and Trace Libraries*. Paderborne, Germany: dSPACE GmbH, 2003.
[49] A. Chipperfield, P. J. Fleming, H. Pohlheim, and C. Fonseca, *Genetic Algorithm Toolbox*. [Online]. Available: http://www.shef.ac.uk/acse/research/ecrg/gat.html

Andrea Caponio (S'05) received the M.Sc. degree in electrical engineering from the Technical University of Bari, Bari, Italy, in 2005.

His final project in electric drives dealt with the intelligent design of permanent-magnet synchronous motors drives. His research interests include variable-speed drives and optimization methods based on genetic and evolutionary algorithms.

Giuseppe Leonardo Cascella (S'02) received the M.Sc. degree (with honors) and Ph.D. degrees in electrical engineering from the Technical University of Bari, Bari, Italy, in 2001 and 2005, respectively.

He worked with the Getrag GmbH Systemtechnik, St. Georgen, Germany, on the automatic transmission. He is currently an Assistant Researcher with the Technical University of Bari. His research interests include artificial intelligence for electrical drives.

Ferrante Neri (S'04) received the M.Sc. degree in electrical engineering from the Technical University of Bari, Bari, Italy, in 2002, where he is currently working toward the Ph.D. degree.

In 2003, he was an Assistant Researcher with the Department of Electrotechnics and Electronics—Technical University of Bari. His research interests include memetic algorithms, evolutionary optimization in the presence of uncertainties, evolutionary computation in automation and electrical engineering, evolutionary structural and topological optimization, and evolutionary computation in medicine and biology.

Nadia Salvatore (S'03) received the M.Sc. degree (with honors) in electrical engineering from the Technical University of Bari, Bari, Italy, in 2002, where she is currently working toward the Ph.D. degree.

Since 2003, she has been working with the Electrical Machine and Drives Laboratory, Technical University of Bari. Initially, she participated in the Italian research project: "Signal Processing for Diagnostic of Electrical Machines fed by Power Converters." During 2005, she worked as a Researcher with the School of Electric and Electronic Engineering, University of Nottingham, Nottingham, U.K. Her primary research interest is intelligent modeling and diagnosis and control of induction motor drives.

Mark Sumner (M'92) received the B.Eng. degree in electrical and electronic engineering from Leeds University, Leeds, U.K., in 1986, and the Ph.D. degree from Nottingham University, Nottingham, U.K., in 1990.

He worked with Rolls Royce Ltd. in Ansty before embarking on a research work in induction motor drives at Nottingham University. After working at Nottingham as a Research Assistant, he was appointed Lecturer in October 1992 and Senior Lecturer in 2004. His research interests cover microprocessor control of power electronic systems, including advanced sensored and sensorless induction motor drives research, active filters, system identification, and the development of new converter topologies.