Prudent-Daring vs Tolerant Survivor Selection Schemes in Control Design of Electric Drives

Ferrante Neri[1], Giuseppe L. Cascella[1], Nadia Salvatore[1],
Anna V. Kononova[2], and Giuseppe Acciani[1]

[1] Dipatimento di Elettrotecnica ed Elettronica, Politecnico di Bari,
Via E. Orabona 4, Bari, 70125, Italy
neri@deemail.poliba.it, leocascella@ieee.org,
n.salvatore@deemail.poliba.it, acciani@deemail.poliba.it

[2] Laboratory of Theoretical Probabilistic Methods,
Yaroslavl' State University,
150000, Sovetskaya st. 14, Yaroslavl', Russia
annk2000@mail.ru

Abstract. This paper proposes and compares two approaches to defeat the noise due the measurement errors in control system design of electric drives. The former is based on a penalized fitness and two cooperative-competitive survivor selection schemes, the latter is based on a survivor selection scheme which makes use of the tolerance interval related to the noise distribution. These approaches use adaptive rules in parameter setting to execute both the explicit and the implicit averaging in order to obtain the noise defeating in the optimization process with a relatively low number of fitness evaluations. The results show that the two approaches differently bias the population diversity and that the first can outperform the second but requires a more accurate parameter setting.

1 Introduction and Problem Description

When an evolutionary algorithm is implemented, the individuals are explicitly or implicitly sorted according to their fitness values in order to perform a parent selection, a survivor selection or to assign a lifetime score. If the evolutionary optimization is performed in a noisy environment, the solutions can be wrongly sorted due to the fitness overestimations and underestimations (see [1] and [2]). This paper proposes and compares two different approaches to treat the noisy environment and shows an application to the control of a Permanent Magnet Synchronous Motor (PMSM) in presence of measurement errors. In Fig. 1 the block diagram of a vector-controlled PMSM drive is shown. For details concerning this control scheme see [3], [4] and [5]. The problem of the control design (self-commissioning) can be formulated as the determination of ten parameters (see Fig. 2) which solve a multi-objective optimization problem in $H \subset \mathbb{R}^{10}$. The performance given by each solution is numerically evaluated through a cost objective function f built up by means of the weighted-sum of $f_{1,j}$, $f_{2,j}$, $f_{3,j}$ and $f_{4,j}$ which respectively measure the speed error at the settling, speed overshoot,

F. Rothlauf et al. (Eds.): EvoWorkshops 2006, LNCS 3907, pp. 805–809, 2006.

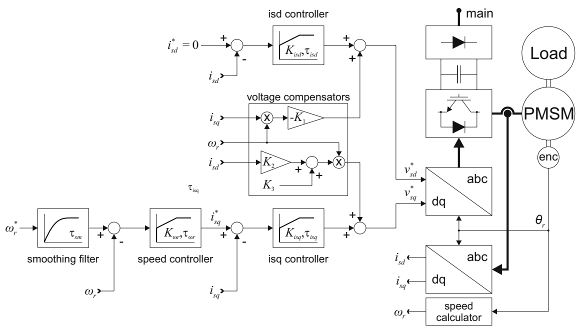

Fig. 1. Motor control system

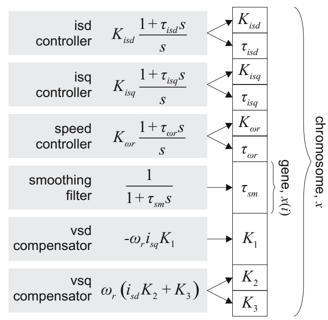

Fig. 2. Candidate solution

speed rise-time, and undesired d-axis-current oscillations for each speed step j of a training test (see [5] for details). Since each single objective function requires a set of measurements the measurement errors affect the objective function f which is therefore noisy.

2 Prudent-Daring and Tolerant Selection Schemes

The Adaptive Prudent-Daring Evolutionary Algorithm (APDEA) executes the minimization of f operating dynamically on both the population size and the number of fitness evaluations (sample size). For each individual, the fitness f is replaced with an "image" fitness function given by $\hat{f} = f_{est} + \frac{b}{n_s^i}$ where the estimated fitness f_{est} [1] is the current average fitness calculated over n_s^i samples (related to the i^{th} individual of the population) and b is a weight coefficient. $\frac{b}{n_s^i}$ is a penalty term which has a big influence for unreliable solutions (n_s^i low) and which progressively tends to have a negligible influence for reliable solutions ($n_s^i \gg 1$). Besides, a maximum number of samples n_s^{max} is established taking into account the features of the noise under examination. An initial sampling of points (see Fig.2) is done at pseudo-random. At the first generation the fitness $\hat{f}$ is calculated (with $n_s^i = 1$) for all the individuals and the coefficient $\xi = \min\left\{1, \left|\frac{\hat{f}_{best} - \hat{f}_{avg}}{\hat{f}_{best}}\right|\right\}$ is determined. $\hat{f}_{best}$ and $\hat{f}_{avg}$ are respectively the best and average fitness values among the individuals of the population. The coefficient ξ is a fitness-based index of the population diversity; it can be seen as a measurement of the state of the convergence of the algorithm (see [5]). In fact if $\xi \approx 1$ there is a high population diversity and therefore the convergence conditions are far; if $\xi \approx 0$ there is a low population diversity and therefore the convergence is approaching. At each subsequent generation μ individuals undergoing crossover are selected according to the ranking selection [6] and the blend crossover [7] is then applied. The mutation probability is then calculated by $p_m = 0.4\,(1 - \xi)$ and the mutation is executed (see for details [5]). The fitness values of the λ offspring individuals are calculated (with $n_s^i = 1$) and the population made up of both parents and offspring ($\mu + \lambda$) undergoes the following survivor selection based on two cooperative-competitive [8] schemes. a) *Prudent* Survivor Selection: the value $S_{pru} = S^f_{pru} + S^v_{pru}(1-\xi)$ is calculated and the best performing S_{pru} individuals according to $\hat{f}$ are thus selected. S^f_{pru} is the minimum size of the prudent population and S^v_{pru} is the maximum size of the variable population; b) *Daring* Survivor Selection: the value $S_{dar} = round\left(S^{max}_{dar}\xi\right)$ is calculated and the best performing S_{dar} individuals according to f_{est} are thus selected. S^{max}_{dar} is maximum size of the daring population; c) The prudent and daring populations are merged ($S_{pop} = S_{pru} + S_{dar}$). Thus, the algorithm prudently selects a part of the population taking into account the reliability of the solutions and dares give the surviving chance to those solutions which are not reliable (n^i_s low) but which are promising (f_{est} high). Moreover, the algorithmic logic is based on the idea that for $\xi \approx 1$ (S_{pru} small and S_{dar} big) the fitness values are very different among each other and a wrongly estimated individual coming from the daring survivor selection does not strongly affect the operation of sorting according to the fitness. On the contrary, for $\xi \approx 0$ (S_{pru} big and S_{dar} small) a wrongly estimated individual could significantly affect the operation of sorting according to the fitness. In the last case, to dare introduce an unreliable individual could mean to introduce a wrong direction search [1]. The newly merged population then undergoes a reevaluating cycle: for each individual the value of additional samples $n^{add}_s = round\left(n^{max}_s \frac{(1-\xi)}{n^i_s}\right)$ is calculated and n^{add}_s fitness reevaluations are executed. The values of n^i_s, f_{est} and $\hat{f}$ are then updated. Consequently, the number of reevaluations to be executed on one individual depends on the state of the whole population (i.e. ξ) and on the previous history of the individual (i.e. n^i_s). Finally, the coefficient ξ is updated at the end of each generation.

The Adaptive Tolerant Evolutionary Algorithm (ATEA) assumes that the noise is Gaussian and that its standard deviation has the same constant value in all the points of the domain H under study. Taking into account these hypotheses, the wideness w_{TI} of the Tolerance Interval related to noise has been determined as shown in [9]. The ATEA works on the fitness $\tilde{f} = f_{est}$ [1]. An initial sampling is performed at pseudo-random. The fitness values f of these individuals are determined and the coefficient $\xi = \min\left\{1, \left|\frac{f_{best}-f_{avg}}{f_{best}}\right|\right\}$ is thus calculated. In the generic k^{th} generation the following steps are executed. Selection (μ), blend corossover and mutation occur as for the APDEA. The fitness values related to the offspring newly generated (λ) are thus calculated. The $(\mu + \lambda)$ individuals undergo the Tolerant Survivor Selection consisting of the following. a) The individuals are sorted according to the fitness $\tilde{f}$; b) The population size $S_{pop} = S^f_{pop} + S^v_{pop}(1-\xi)$ is calculated; c) The individual having position S_{pop} with fitness $\tilde{f}_{S_{pop}}$ is detected and an auxiliary population made up of individuals whose fitness value falls within $\left[\tilde{f}_{S_{pop}} - \frac{w_{TI}}{2}, \tilde{f}_{S_{pop}} + \frac{w_{TI}}{2}\right]$ is created; d) For each individual of this auxiliary population the value $n^{add}_s = round\left(n^{max}_s \frac{(1-\xi)}{n^i_s}\right)$ is calculated and n^{add}_s fitness reevaluations are executed. The values of n^i_s and $\tilde{f}$ are then updated; e) The main population (made up of $(\mu + \lambda)$ individuals) is updated and resorted according to $\tilde{f}$; f) The best performing S_{pop} individuals are saved for the subsequent generation. Finally, the value of ξ is updated according to the formula $\xi = \min\left\{1, \left|\frac{\tilde{f}_{best}-\tilde{f}_{avg}}{\tilde{f}_{best}}\right|\right\}$. The main idea behind the ATEA is that if it is possible to prove that an individual, even if underestimated, is in the top part of the list, it should not be reevaluated; analogously if an individual is so bad that, even if overestimated, is in the bottom part of the list. In other words, when S_{pop} is calculated, the algorithm implicitly divides the population in three categories: individuals surely good, individuals surely bad and individuals which require a more attentive analysis. The individuals surely good or bad do not require additional evaluations; the others require a reevaluation cycle.

3 Numerical Results and Conclusion

Following the procedure described in [9] for Gaussian distribution, the 90% of the fitness samples falls in a tolerance interval with wideness $w_{TI} = 0.1702$ with a probability $\gamma = 0.99$. Both the APDEA and the ATEA have been executed in order to minimize the fitness f with $n_s^{max} = 10$. Concerning the APDEA $S_{pru} \in [40, 160]$, $S_{dar} \in [0, 20]$ and $b = 0.2$; concerning the ATEA $S_{pop} \in [40, 160]$. Also a standard Reevaluating Genetic Algorithm (RGA) employing the same crossover and mutation techniques used for the APDEA and the ATEA has been implemented. This RGA executes the averaging over time [2] with a sample size $n_s^{max} = 10$ for every evaluation and makes use of a standard survivor selection which saves at each generation the prefixed $S_{pop} = 100$ best performing individuals. Each of the three algorithms has been executed 65 times. Each execution has been stopped after 20000 fitness evaluations. Table 1 compares the best performing solutions found by the three methods and shows the best fitness values f, the average best fitness $\bar{f}$ (over the 65 experiments) and the corresponding standard deviation σ. The algorithmic performances and the behavior of ξ for the APDEA and the ATEA are shown in Fig. 3 and in Fig. 4 respectively. The numerical results show that both the APDEA and the ATEA converge faster than the RGA to solutions very similar among each other. Concerning the convergence velocity, the APDEA proved to be more performing than the ATEA. Moreover the APDEA is more general than the ATEA since the latter makes use of the assumption that the noise is Gaussian and with a constant σ in all the domain. On the other hand, the APDEA, unlike the ATEA, requires the setting of b and S_{dar}^{max}. A wrong choice of b would lead to a too strong or too weak penalization in the fitness function. Analogously, S_{dar}^{max} determines the audacity of the algorithm and its wrong setting could lead to either a wrong

Table 1. Solutions and related Fitness

	K_{isd}	τ_{isd}	K_{isq}	τ_{isq}	$K_{\omega r}$	$\tau_{\omega r}$	τ_{sm}	K_1	K_2	K_3	f	$\bar{f}$	σ
RGA	10.99	0.0023	6.66	0.0012	0.243	0.0141	0.0106	0.0019	0.0009	0.1891	0.858	0.867	0.0134
APDEA	11.49	0.0022	6.20	0.0011	0.264	0.0145	0.0109	0.0021	0.0006	0.1901	0.851	0.861	0.0158
ATEA	11.14	0.0021	6.61	0.0013	0.259	0.0132	0.0101	0.0020	0.0008	0.1957	0.854	0.860	0.0120

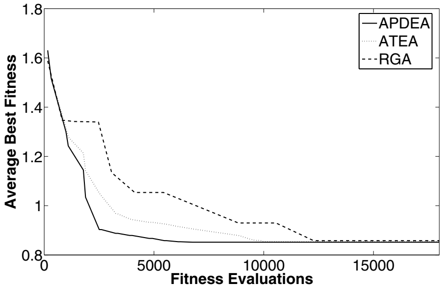

Fig. 3. Performances of the three algorithms

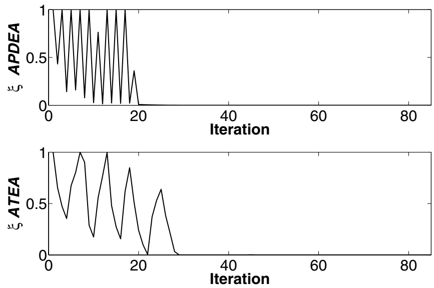

Fig. 4. Behavior of ξ for the APDEA and the ATEA

search direction or an excessive exploitation. Regarding n_s^{max}, that is a parameter common for both APDEA and ATEA, the setting is a much less critical issue. In fact, it can be set as the minimum sample size which describes a proportion of distribution with a given probability (see [9]). Fig. 4 shows that in the case of the APDEA, ξ has high-frequency oscillations before settling down to the value 0. For the ATEA, ξ has less oscillations with low-frequency. Our interpretation of this phenomenon is the following. The APDEA introduces during the daring selection some unreliable solutions before reevaluating them. This behavior leads to an abrupt increasing of the population diversity that is corrected during the survivor selection of the subsequent generation. On the contrary, the ATEA aims to properly sort the candidate solutions and to include for the subsequent generation only the solutions that are surely in the top part of the list. Consequently, the classical recombination and mutation are the ones in charge of the exploration. This logic leads to a milder variation of the population diversity.

References

1. Branke, J.: Evolutionary Optimization in Dynamic Enviroments. Kluwer (2001)
2. Jin, Y., Branke, J.: Evultionary optimization in uncertain enviroments-a survey. IEEE Trans. on Evolutionary Computation **9** (2005) 303–317
3. Krishnan, R.: Electronic Motor Drives: Modeling, Analysis and Control. Prentice Hall, Upper Saddle River, New Jersey, USA (2001)
4. Leonhard, W.: Control of Electrical Drives. 2nd edn. Springer-Verlag (1996)
5. Caponio, A., Cascella, G.L., Neri, F., Salvatore, N., Sumner, M.: A fast adaptive memetic algorithm for on-line and off-line control design of pmsm drives. To app. IEEE Trans. on System Man and Cybernetics-B, spec. issue Memetic Alg.s (2006)
6. Whitley, D.: The genitor algorithm and selection pressure: Why rank-based allocation of reproductive trials is best. In: Proc. 3rd Int. Conf. on GAs. (1989) 116–121
7. Eshelman, L.J., Shaffer, J.D.: Real-coded genetic algorithms and interval-schemata. In: Foundations of Genetic Algorithms 2, Morgan Kaufmann (1993) 187–202
8. Ong, Y.S., Keane, A.J.: Meta-lamarkian learning in memetic algorithms. IEEE Trans. on Evolutionary Computation **8** (2004) 99–110
9. NIST: e-handbook of statistical methods. (www.itl.nist.gov/div898/handbook/)