Optimization of Delayed-State Kalman-Filter-Based Algorithm via Differential Evolution for Sensorless Control of Induction Motors

Nadia Salvatore, Member, IEEE, Andrea Caponio, Student Member, IEEE, Ferrante Neri, Member, IEEE, Silvio Stasi, and Giuseppe Leonardo Cascella, Member, IEEE

Abstract—This paper proposes the employment of the differential evolution (DE) to offline optimize the covariance matrices of a new reduced delayed-state Kalman-filter (DSKF)-based algorithm which estimates the stator-flux linkage components, in the stationary reference frame, to realize sensorless control of induction motors (IMs). The DSKF-based algorithm uses the derivatives of the stator-flux components as mathematical model and the stator-voltage equations as observation model so that only a vector of four variables has to be offline optimized. Numerical results, carried out using a low-speed training test, show that the proposed DE-based approach is very promising and clearly outperforms a classical local search and three popular metaheuristics in terms of quality of the final solution for the problem considered in this paper. A novel simple stator-flux-oriented sliding-mode (SFO-SM) control scheme is online used in conjunction with the optimized DSKF-based algorithm to improve the robustness of the sensorless IM drive at low speed. The SFO-SM control scheme has closed loops of torque and stator-flux linkage without proportional-plus-integral controllers so that a minimum number of gains has to be tuned.

Index Terms—AC motor drives, algorithms, covariance matrices, evolutionary algorithms (EAs), induction-motor (IM) drives, Kalman filtering, optimization methods, parameter estimation, speed sensorless, state estimation, velocity control.

NOMENCLATURE

\( p \) Derivative operator.

\( v_{sd}, v_{sq} \) - and \( q \)-axis stator-voltage components (in volts).

\( i_{sA}, i_{sB} \) Stator phase currents (in amperes).

\( i_{sA,m}, i_{sB,m} \) Measured stator phase currents (in amperes).

\( i_{sd}, i_{sq} \) - and \( q \)-axis stator current components (in amperes).

\( \lambda_s \) Stator-flux linkage (in weber).

\( \Delta \lambda_s \) Flux linkage error (in weber).

\( \alpha_s, \beta_s \) Electrical angular speed and position of stator-flux linkage vector (in radians per second and in radians, respectively).

\( R_s \) Stator resistance (in ohms).

\( \Phi_\lambda \) Boundary layer thickness of stator flux (in weber).

\( k_{\lambda_s} \) Maximum derivative of the stator flux (in volts).

\( \tau_\lambda \) Time constant that defines the stator-flux dynamic (in seconds).

\( T_e \) Electromagnetic torque (in newton meter).

\( n_p \) Pole pairs.

\( \Delta T_e \) Electromagnetic torque error (in newton meter).

\( k_p \) Proportional gain of variable-saturation P-type regulator of electromagnetic torque.

\( g_T \) Electromagnetic torque gain.

\( v_{\text{max}} \) Maximum \( q \)-axis voltage (in volts).

\( V_{dc} \) DC-link voltage (in volts).

\( k_{i\omega} \) Proportional gain of speed proportional-plus-integral (PI) regulator (in newton meter second per radian).

\( k_i \) Integral gain of speed PI regulator (in newton meter per radian).

\( \lambda_{\alpha\beta} \) Stator-flux linkage vector in the stationary reference frame (in weber).

\( \alpha \) - and \( \beta \)-axis stator-flux linkage vector components in the stationary reference frame (in weber).

\( x(n) \) State variable vector at the \( n \)th time-step.

\( y(n) \) Measured variable vector at the \( n \)th time-step.

\( T_c \) Sample time interval (in seconds).

\( f_c \) Inverter switching frequency (in hertz).

\( F(n) \) State matrix.

\( H(n) \) Output matrix.

\( G(n) \) Kalman gain matrix.

\( Q, R \) Covariance matrices of mathematical model errors and measurement errors.

\( \omega_r \) Electrical angular rotor speed (in radians per second).

\( L_s, L_r, L_m \) Stator, rotor, and magnetizing inductances (in henrys).

\( T_r \) Rotor time constant (in seconds).

\( \sigma \) Total leakage factor.

\( R_r \) Rotor resistance (in ohms).

\( d_a, d_b, d_c \) Duty cycles of phases a, b, c.

\( d_a^*, d_b^*, d_c^* \) Duty cycles of phases a, b, c with compensation of dead time.

0278-0046/$26.00 © 2010 IEEE
DIFFERENT techniques for the speed-sensorless control of induction motors (IMs) have been proposed in these last years because of lower cost and greater reliability of drives. In [1], the sensorless control with stator-flux orientation was proposed for the first time. Computational steps and dependence on many motor parameters are very much reduced by using only the stator-flux linkages and stator currents to close the flux loop and the electromagnetic torque one. Unfortunately, drift problems rise in calculating stator-flux linkages by open integration. Moreover, when the stator voltages are small and of magnitude comparable to the resistive voltage drops, the stator-resistance variation and signal noise cause a lack of the estimation accuracy of both the stator-flux linkages and rotor speed so that dynamic operations at low speeds are very poor under stator-flux orientation. The previously described drawbacks suggest to improve the stator-flux-oriented control (SFOC) by means of sliding-mode controllers and stator-flux linkages estimator based on the Kalman filter theory. The sliding-mode controllers have good robustness in the face of parameter uncertainties and other disturbances [2]. Moreover, the stator-flux-oriented sliding-mode (SFO-SM) control scheme has structural simplicity, compared to the classical SFOC scheme, and less computational requirements. It is only necessary to pay attention to the chattering phenomenon. In [2], it was demonstrated that the SFO-SM control scheme is capable of fast torque response and smooth steady-state behavior. The proposed scheme can be considered a direct-torque-control space-vector-modulation scheme with closed loops of torque and stator flux without PI controllers. Comparison of indirect rotor field orientation and SFO-SM-controlled IM drives showed similar dynamic responses in spite of structural simplicity of the SFO-SM control scheme.

This paper presents an SFO-SM control scheme of sensorless IM drive realized by using a new algorithm based on the Kalman filter theory [3] to estimate stator-flux linkages from line currents affected by noise. The Kalman filter theory is considered to be very efficient to online estimate the state variables of IM from line currents [4], [5]. The Kalman filter needs the mathematical model of IM. The usual fourth-order electrical model is extended to include other parameters to the state vector. In [4] and [5], the rotor time constant parameter is also estimated, and fifth-order extended Kalman filters are necessary. De Santana et al. [6] uses fifth-order EKFs to estimate rotor speed. Alonge et al. [7], [8] use sixth-order EKFs to estimate speed and load torque and show the possibility to realize sensorless controls. The high-order mathematical models present the disadvantages of high computational cost and hardness of tuning of noise covariance matrices. These disadvantages can be strongly decreased by a suitable choice of reduced mathematical models. The papers [9] and [10] showed the effectiveness of the approach based on Kalman filters having reduced mathematical models. In this paper, we propose a second-order mathematical model to estimate stator-flux components used to calculate the rotor speed. The Kalman filter also needs the observation model to predict the measurement vector. The proposed observation model depends on the state variables at time \( n \) and \( (n - 1) \) so that the designed Kalman filter has the structure of a delayed-state Kalman filter (DSKF) [3].

As it is well known, superior Kalman filter performance can be obtained by offline tuning the covariance matrices. A technique of optimization of the covariance matrices is necessary to avoid trial-and-error time-consuming tuning.

In electrical and control engineering, it can often happen to face nonlinear and nondifferentiable optimization problems that must be solved through a direct-search approach. As highlighted in [11], the application of stand-alone traditional local search methods (e.g., Hooker–Jeeves and Nelder–Mead) can be inappropriate in industrial electronics, since, due to the multimodalities characterizing the optimization problems, these methods are likely to converge to local optima. Among the possible search methods that can be applied, evolutionary algorithms (EAs) can be a reasonable choice, since they often offer good performances for highly nonlinear and multimodal optimization problems. The most diffused and known EAs in literature are genetic algorithms (GAs) which have been applied in many different engineering and control problems. Recently, hybrid algorithms composed of an evolutionary framework and a set of local search methods (memetic algorithms) have been proposed in control-engineering problems in order to overcome the difficulties of the fitness landscapes (features of the objective functions) and converge to solutions with high performance.

Although the aforementioned algorithms are able to offer an excellent performance in domain-specific problems, due to their complexity and the large amount of algorithmic parameters to be tuned, they might likely suffer of a lack of robustness over similar engineering problems. In other words, a sophisticated algorithm contains several algorithmic components and (self-) adaptive systems. Due to this fact, a large amount of parameters must be set and, more generally, several implementation choices must be carried out by the user, e.g., the depth or initial exploratory radius of a local-search algorithm integrated within an evolutionary framework. In accordance with the no-free-lunch theorems (see [12]), a tailored setting/implementation is necessary in order to guarantee a high performance for a given optimization problem. In the case of optimization in control engineering, there is a further difficulty. Due to the dynamic nature of control engineering problems, mild variations in the structure or working conditions might heavily affect the main features of the optimization problem. Thus, proper values of the algorithmic parameters (and implementation choices) might significantly vary for very similar control-engineering problems.

Taking into account these considerations, this paper proposes a relatively simple and robust optimization approach for tuning the covariance matrices of the DSKF. More specifically, the proposed approach employs the structure of the differential evolution (DE).
DE (see [13] and [14]) is a reliable and versatile function optimizer and seems to be very efficient for solving real-valued problems and often able to handle nonlinearities and multimodalities. Despite its simplicity, the DE (also owing to the few parameters to be tuned) is often a reliable possibility which exhibits a very good performance for a wide range of optimization problems and can thus be employed for various real-world applications. The main advantages of DE, such as a simple and straightforward logic, compact structure, ease of use, high convergence characteristics, and robustness, make it a high-class technique for parameter optimization. Recently, the DE has been an object of interest for computer scientists, and various relevant enhancements have been proposed. For example, Brest et al. [15] proposed a simple probabilistic scheme with a self-adaptive logic for updating parameter values during the evolution. Rahnamayan et al. [16] proposed a heuristic technique, namely, opposition-based learning for population initialization and also for generating search directions, alternative to those normally generated by a DE scheme, during the evolution. Brest and Maučec [17] show that a progressive population reduction is often beneficial in terms of algorithmic performance.

Due to the fact that DE has been recently defined, it has still been seldom used in the field of control and automation, while in the field of electric drives, to the authors’ knowledge, it has only been applied to the problem of IM parameter identification [18] and to the parameter determination of a line-start interior permanent-magnet synchronous motor [19]. In this paper, which is an extended and enhanced version of the conference paper [20], we propose the use of a relatively simple DE version (the so-called DE/rand/1/bin [13]) to offline tune the DSKF-based algorithm used in the speed-sensorless SFO-SM control of IM. With respect to paper [20], we compare the performance of DE, related to the given problem, with those of the Nelder–Mead algorithm (NMA), the simulated annealing (SA), and a particle swarm optimizer (PSO). The reasons for the choice of a simple DE structure can be summarized in the following. First, the authors aim at proving the viability of the DE schemes for the application studied in this paper, since this application is novel in literature. Second, the authors aim at employing, at this stage of the research, a simple and versatile optimizer which requires the setting of only three parameters and can easily be implemented for various similar industrial electronic problems. Third, the low-dimensionality of this application makes the employment of complex DE schemes unnecessary or even inadequate, since they would display a poor performance in terms of convergence speed without leading to improvements on the final solution detected. In other words, a standard DE is usually efficient in low-dimensional problems while it may likely fail in high dimensions, i.e., suffers from the so-called curse of dimensionality. Thus, while in high-dimensional cases, a modified structure of DE can significantly improve upon the standard DE performance in low-dimensional cases (like the application problem presented in this paper), a standard DE is competitive with its complex modified versions or even superior to them (see [15]–[17]).

The rest of this paper is organized in the following way. Section II describes the details of the control system considered in this paper. Section III describes the novel DSKF algorithm proposed in this paper. Section IV describes the features of the DE employed for the offline optimization of the DSKF algorithm. Section V shows new experimental results, also in the presence of load torque, of the real-time implementation of the proposed sensorless-control scheme with the offline-optimized DSKF. Experiments show that the sensorless-control scheme is stable and effective. Section VI gives the conclusions of this paper.

II. STATOR-FLUX SLIDING-MODE CONTROLLER AND TORQUE CONTROLLER

Consider the stator equations for the induction machine, written in the \( d-q \) reference frame synchronously rotating with the stator-flux space vector

\[
\begin{aligned}
v_{sd} &= R_s i_{sd} + p \lambda_s \\
v_{sq} &= R_s i_{sq} + \omega_s \lambda_s. 
\end{aligned}
\]  

Since the \( q \)-axis component of the stator flux is zero, the flux dynamics can be controlled directly by the \( d \)-axis stator-voltage component \( v_{sd} \). The following simple control law was proposed in [2]:

\[
v_{sd}^* = k_{\lambda_s} \text{sat} \left( \frac{\Delta \lambda_s}{\Phi_s} \right) + \dot{R}_s \hat{i}_{sd} 
\]

where the flux error \( \Delta \lambda_s \) equals \( \lambda_s^* - \hat{\lambda}_s \). The sat operator is defined as follows:

\[
sat(x) = \begin{cases} 
  \text{sign}(x), & \text{if } |x| > 1 \\
  x, & \text{otherwise}
\end{cases}
\]

where \( \text{sign}() \) denotes the signum function.

The stator-flux sliding-mode dynamics is defined by a first-order system with time constant \( \tau_{\lambda_s} = \Phi_s / k_{\lambda_s} \).

It is suggested that the time constant is set to twice the sampling time to have very fast dynamics inside the boundary layer. Once this time constant is chosen, it is possible to calculate the correct value for the parameter \( k_{\lambda_s} = \Phi_s / \tau_{\lambda_s} \) that must be low enough to avoid chattering.

The electromagnetic torque developed by an IM, written as a function of the stator-flux linkage and \( q \)-axis stator current, is

\[
T_c = (3/2) n_p \lambda_s i_{sq}.
\]

The following control law for the \( q \)-axis stator voltage was proposed in [2]:

\[
v_{sq}^* = v_{sq}^{\text{max sat}} \left[ \frac{v_{sq}^{\text{max sat}} \text{sat} \left( \frac{\Delta T_c}{V_{dc}} \right) + \omega_s \lambda_s}{v_{sq}^{\text{max sat}}} \right]
\]

where \( \Phi_T \) is a positive constant, \( v_{sq}^{\text{max sat}} = \sqrt{(v_{sq}^{\text{sat}})^2 - (v_{sq}^{\text{sat}})^2} \) is the maximum voltage available along the \( q \)-axis, \( v_{sq}^{\text{sat}} \) is the rated stator voltage equal to \( V_{dc}/\sqrt{3} \), \( V_{dc} \) is the dc-bus voltage, and \( \Delta T_c \) equals \( T_c^* - \dot{T}_c \).

Parameter \( \Phi_T \) has to be chosen large enough to avoid chattering, but no fine-tuning is required to obtain good performance. We suggest setting \( \Phi_T \) equal to \( 0.5 T_c^{\text{sat}} \), where \( T_c^{\text{sat}} \) is the rated electromagnetic torque.
The torque control law is P-type with compensation for the back-EMF $\omega_{\lambda_s}\lambda_s$. To avoid steady-state torque error, the torque reference can be calculated by multiplying the desired torque value $(3/2)n_p\lambda_{sq}^2$ by a gain $g_T$

$$T_e = g_T \frac{3}{2} n_p \lambda_s^2 \nu_{sq}^2$$

(5)

with $g_T = (1 + k_p)/k_p$ and $k_p = (3/2)n_p\lambda_s^2(1/\Phi_T)(\nu_{sq}^2/R_s)$.

### III. DSKF-Based Algorithm

In the stationary $\alpha-\beta$ reference frame, the stator-flux linkage vector is $\lambda_{s\alpha\beta} = \lambda_{s\alpha}e^{j\theta_{s\alpha}},$ where $\theta_{s\alpha} = \tan^{-1}(\lambda_{s\beta}/\lambda_{s\alpha})$. The derivative of stator-flux linkage vector can be written as

$$d\lambda_{s\alpha\beta} = j\omega_{s\lambda}s\lambda_{s\alpha\beta} + \bar{e}_{s\alpha\beta}$$

(6)

where $\bar{e}_{s\alpha\beta} = e^{j\theta_{s\lambda}}p\lambda_s$ and $\omega_{s\lambda} = p\theta_{s\lambda}$. The term $\sigma_{s\alpha\beta}$ depends on the derivative of the modulus of flux linkage vector. A discrete mathematical model for the new DSKF algorithm is derived from (6) as follows:

$$x(n + 1) = F(n)x(n) + u(n) + \varepsilon(n)$$

(7)

where

$$x(n) = \begin{bmatrix} \lambda_{s\alpha}(n) \\ \lambda_{s\beta}(n) \end{bmatrix}, u(n) = \begin{bmatrix} u_{s\alpha}(n) \\ u_{s\beta}(n) \end{bmatrix}$$

(8)

$$\varepsilon(n) = \begin{bmatrix} \varepsilon_{s\alpha}(n) \\ \varepsilon_{s\beta}(n) \end{bmatrix}$$

(9)

$$F(n) = \begin{bmatrix} 1 & -\omega_{s\lambda}(n)T_c \\ \omega_{s\lambda}(n)T_c & 1 \end{bmatrix}$$

(10)

$$\omega_{s\lambda}(n) = \frac{\theta_{s\lambda}(n) - \theta_{s\lambda}(n - 1)}{T_c}$$

(11)

and $\varepsilon_{s\alpha}, \varepsilon_{s\beta}$ are the model errors. The forcing term is

$$u(n) = \begin{bmatrix} \cos(\theta_{s\lambda}(n)) \\ \sin(\theta_{s\lambda}(n)) \end{bmatrix}(\lambda_{s\alpha}(n) - \lambda_s(n - 1)).$$

(12)

The observation model is

$$y(n + 1) = H(x(n + 1) - x(n)) + \eta(n + 1)$$

(13)

where

$$y(n) = \begin{bmatrix} \nu_{s\alpha}(n) - R_s\dot{i}_{s\alpha}(n) \\ \nu_{s\beta}(n) - R_s\dot{i}_{s\beta}(n) \end{bmatrix} T_c$$

(14)

$$\eta(n) = \begin{bmatrix} \eta_{s\alpha}(n) \\ \eta_{s\beta}(n) \end{bmatrix}, H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

(15)

and $\eta_{s\alpha}, \eta_{s\beta}$ are the observation errors and represent all disturbances such as offsets, unbalances, and other errors that are contained in the measured currents and in the voltages calculated by inverter duty cycles and measured dc-link voltage $V_{dc}$. The state variables and measurements at time $(n + 1)$ are predicted by means of the mathematical model and observation one. The prediction equations are

$$\hat{x}(n + 1) = \hat{F}(n)\hat{x}(n) + \hat{u}(n)$$

$$\hat{y}(n + 1) = H(\hat{x}(n + 1) - \hat{x}(n)) \cdot$$

(16)

The predicted measurement vector has the structure of the DSKF [3]. The predicted state at time $(n + 1)$ is corrected by adding the weighted difference between the measurement vector $y(n + 1)$ and the predicted one $\hat{y}(n + 1)$ to obtain the estimated state at time $(n + 1)$. It is

$$\hat{x}(n + 1) = \hat{x}(n + 1) + G(n + 1)(y(n + 1) - \hat{y}(n + 1))$$

(17)

where the gain matrix $G$ is chosen to minimize the estimation-error variances of the state variables to be estimated. The gain matrix $G$ (also called Kalman gain matrix) is given by

$$G(n + 1) = \left(\hat{P}(n + 1) - \hat{F}(n)\hat{P}(n)\right)H^TN^{-1}(n + 1)$$

(18)

where

$$N(n + 1) = H\left(\hat{P}(n + 1) + \left(I - \hat{F}(n)\right)\hat{P}(n)\right) - \hat{P}(n)H^TN^{-1}(n + 1) + R$$

(19)

$\hat{P}$ and $\hat{P}$ are the covariance matrices of the prediction and estimation errors, respectively, and $R = \text{diag}(R(1), R(2))$ is the covariance matrix of the measurement errors.

The prediction and estimation-error covariance matrices may be computed as follows:

$$\hat{P}(n + 1) = \hat{F}(n)\hat{P}(n)\hat{F}^T(n) + Q$$

(20)

$$\hat{P}(n + 1) = \hat{P}(n + 1) - G(n + 1)N(n + 1)G^T(n + 1)$$

(21)

where $Q = \text{diag}(Q(1), Q(2))$ is the covariance matrix of the mathematical model errors.

The sufficient condition for the asymptotic stability of the proposed DSKF exists at steady state, in which the filter becomes linear, and is due to the complete observability and controllability of the process and to the bounded matrices $Q$, $R$, and $F$ [21].

The electrical angular rotor speed is obtained as

$$\hat{\omega}_r = \hat{\omega}_s - \frac{L_s\dot{i}_{sq}}{T_c(\lambda_s - \sigma L_s\dot{i}_{sd})}$$

(22)

where $T_c = (L_r/R_r)$ is the rotor time constant and $\sigma = 1 - (L_m^2/L_cL_r)$.

### IV. Offline Optimization of DSKF Via DE and Comparison With NMA, GA, SA, and PSO

An important advantage linked to the reduced number of state variables of the proposed DSKF algorithm is that the measurement and model errors covariance matrices have dimension
The voltage components regulator was designed according to the suggestions reported in [22]. The voltage components are necessary. The speed reference shown in Fig. 1 was selected to obtain the experimental results. It is a SFOC scheme.

To have optimal performance from the DSKF-based algorithm, we need to offline optimize the two diagonal elements of the decision space. To do so, the experimental results of a sensored algorithm, we need to offline optimize the two diagonal elements of the decision space. To do so, the experimental results of a sensored current model, were stored. The voltages $v_{sα}$, $v_{sβ}$, and currents $i_{sα}$, $i_{sβ}$ were used as inputs for the offline optimization of the DSKF. The outputs $\hat{ω}_r$, $\hat{i}_{sd}$, and $\hat{i}_{sq}$ estimated by DSKF were compared with $ω_r$, $i_{sd}$, and $i_{sq}$ for each possible combination of diagonal elements $R(1)$, $R(2)$ and $Q(1)$, $Q(2)$ of $R$ and $Q$ so to have a measure of the tuning quality (see Fig. 2).

Each possible combination of $R(1)$, $R(2)$, $Q(1)$, and $Q(2)$ can be represented as a vector of four variables, then the decision space $H \subseteq \mathbb{R}^4$ is a 4-D hyperrectangle given by the Cartesian product of the intervals where each parameter is limited to. According to our experience, the lower and the upper bounds for the four parameters have to be set, respectively, equal to zero and one.

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

$$f = \sum_{i=1}^{3} a_i f_i$$

where $i$ indicates the number of the performance index $f_i$ and $a_i$ is the positive normalization factor of the respective performance index. The performance index $f_i$ is given by the sum of the absolute difference between the estimated rotor speed and the measured one at each step, while $f_2$ and $f_3$ are given by the sum of the absolute difference between the estimated $q$- and $d$-axis currents and the corresponding measured currents, again at each step. In order not to spoil the optimization process considering values of electrical rotor speed too low for the DSKF-based algorithm, the performance indexes were evaluated only when $0.4 < t < 4$ and $7 < t \leq 10$, and the reference speed module was greater than $2\pi$. The condition $1 > 0.4$ was needed so that the initial settling phase was not taken into account. The resulting objective function is consequently highly nonlinear and nondifferentiable, so a direct-search-approach method is needed to find its minimum. In order to minimize the fitness function $f$ within the decision space $H$, the employment of the following DE scheme is here proposed. The DE starts from an initial population vector of $N > 4$ solutions, randomly generated within the search space $H$. Being $P_G$ the population at the $G$th generation, each solution $x_i^G$, with $i = 1, 2, \ldots, N$, can be represented as a vector $[x_{i,1}^G, x_{i,2}^G, \ldots, x_{i,m}^G]$, with $m$ equal to the cardinality of the search space. Given a point $x_i^G$ in $P_G$, the DE pseudorandomly selects other three points: $x_{r1}^G, x_{r2}^G, x_{r3}^G \in P_G$, where $i \neq r_1 \neq r_2 \neq r_3$. Then, DE generates the mutant vector $y_i^G = [y_{i,1}^G, y_{i,2}^G, \ldots, y_{i,m}^G]$ through the following:

$$y_i^G = x_{r_1}^G + (x_{r_2}^G - x_{r_3}^G) F$$

where $F$ is a real and constant factor which controls the amplification of the differential variation $F \in [0, 2]$. DE then performs the crossover on $y_1$ and $x_1$, creating the trial vector $z_i^G = [z_{i,1}^G, z_{i,2}^G, \ldots, z_{i,m}^G]$:

$$z_{i,j}^G = \begin{cases} x_{i,j}^G, & \text{if } \xi_j > CR \\ y_{i,j}^G, & \text{if } \xi_j \leq CR \end{cases}$$

where $j = 1, 2, \ldots, m$, $\xi_j$ is a real number pseudorandomly generated in $[0, 1]$, and $CR$ is the parameter which controls the crossover rate $CR \in [0, 1]$. Vector $z_i^G$ is evaluated, and if it yields a smaller objective function value than $x_i^G$, $x_{i}^{G+1}$ is set equal to $z_i^G$; otherwise, the old value $x_i^G$ is retained. When the point $x_{N}^G$ has been processed, the DE starts again from $P_{G+1}$.

The optimization process is stopped when a budget condition on the number of function evaluations is reached. The simplicity of DE is best explained via the C-style pseudocode, which is shown in Fig. 3.

It is important to underline how the only parameters for the user to set in DE are $N$, $F$, and $CR$, thus implementing and using DE is extremely easy and straightforward.
create set \( P_f \) of \( N \) pseudo-randomly generated points in \( H \); 
while \( i = 1, 2, ..., N \); 
consider solution \( x_f^i \); 
pseudo-randomly pick 3 solution \( x_f^i, x_f^1, x_f^3 \); 
generate point \( y_f^i \) according to equation 24; 
generate point \( z_f^i \) according to equation 25; 
if \( f(x_f^i) \leq f(y_f^i) \) 
\( x_f^{i+1} = x_f^i \), 
else 
\( x_f^{i+1} = z_f^i \), 
end - if 
end - for 
end - while

Fig. 3. DE pseudocode.

In order to prove the effectiveness of the DE in solving the problem under analysis, the DE has been compared with the NMA, a real-valued GA, the SA, and a PSO. For each algorithm, 30 independent runs of 4020 fitness evaluations each have been performed. 4020 evaluations per run, corresponding to 134 generations, were chosen, since it was seen that after about 4000 evaluations, the DE (and the other algorithms under analysis) could no longer improve upon the final solution. Regarding the DE, the population size has been set to 30 in accordance to the suggestion given in [23]. The remaining two parameters have been set \( F = 0.7 \) and \( CR = 0.7 \), as suggested in [24]. The NMA has been run according to its original implementation (see [25]), pseudorandomly generating the starting points within the decision space for each of the 30 runs performed. The parameter setting of the GA has been chosen after a tuning of the performance for the problem under analysis (see [20]). A ranking parent-selection scheme with stochastic uniform selection function and elite count of two has been selected. Regarding the variation operators, a scattered crossover with crossover fraction of 0.8 and a Gaussian mutation function with scale and shrink value equal to one have been set (see [26]). The population size of the GA has been set equal to 30 in order to perform fair comparison against the DE. The SA has been run, according to the implementation described in [27], with pseudorandomly generated starting solution for each independent run and hyperbolic reduction of the temperature as suggested in [28]. The initial temperature has been set equal to one. The PSO has been implemented according to the description given in [29]. A vector containing the maximum acceptable velocities over each dimension has been set; each component is equal to 20% of the width along the corresponding variable. In order to perform a fair comparison with the other population-based algorithms, the population size of PSO has been set equal to 30.

Table I displays the best overall solutions \( \{ R(1), R(2), Q(1), \) and \( Q(2) \} \) detected by the five algorithms under examination.

Table II shows the minimum \( f_{\text{min}} \), the maximum \( f_{\text{max}} \), the average \( f_{\text{mean}} \) of the final (after 4020 fitness evaluations) fitness values over the 30 runs. Table II shows also the corresponding standard-deviation values \( \sigma \). In addition, in order to execute a numerical comparison of the convergence speed performance, the following numerical test has been implemented. The average final fitness values returned by the best performing algorithm \( G \) has been considered (in our case, \( f_{\text{mean}} \) of the DE). Subsequently, the average fitness value at the beginning of the optimization process \( J \) has also been computed. The threshold value \( THR = J - 0.95(G - J) \) has then been calculated. The value \( THR \) represents 95% of the decay in the fitness value of the algorithm with the best performance. If an algorithm succeeds during a certain run to reach the value \( THR \), the run is said to be successful. For each test problem, the average amount of fitness evaluations \( \text{nie} \) required, for each algorithm, to reach \( THR \) has been computed. Subsequently, the \( Q \) test \( (Q \) stands for quality) described in [14] has been applied. For each test problem and each algorithm, the \( Q \) measure is computed as

\[
Q = \frac{\text{nie}}{R_{sr}}
\]

where the robustness \( R_{sr} \) is the percentage of successful runs. It is clear that, for each test problem, the smallest value equals the best performance in terms of convergence speed. The value \( inf \) means that \( R_{sr} = 0 \), i.e., the algorithm never reached the \( THR \). For this application and the algorithms under examination, the \( Q \) measures are listed in Table II. Numerical results listed in Table II show that DE outperforms the other algorithms considered in this paper in terms of final solution and standard deviation. Regarding the \( Q \) measure, the DE displays a very competitive performance but is slightly outperformed by the SA. This fact means that during the early stages of the optimization process, the SA has a good performance in terms of convergence speed but is subsequently outperformed by the DE for the detection of the final solution. In order to give a graphical representation of the performance of the solution detected by the DE, as shown in Fig. 4, the estimated speed obtained by the DSKf using the DE best solution in Table I is represented with a comparison with the measured speed. In order to prove statistical significance of the results, the Student’s t-test has been applied according to the description given in [30] for a confidence level of 0.95. Final values obtained by the DE have been compared to the final value returned by each algorithm used as a benchmark. Table III shows the results of the test. Indicated with “+” is the case when the DE statistically outperforms, for the corresponding test problem,
the algorithm mentioned in the column; indicated with ‘+’ is the case when pairwise comparison leads to success of the t-test, i.e., the two algorithms have the same performance; and indicated with ‘−’ is the case when the DE is outperformed. For the sake of completeness, a qualitative representation of the average algorithmic performance for all the algorithms under investigation is shown in Fig. 5. The average performance is expressed in terms of fitness values in dependence on the number of fitness evaluations. As shown in Fig. 5, the NMA is inadequate for this problem, since it tends to converge to the nearest local minimum. This fact can be seen as a confirmation of the multimodality of the fitness landscape. The applications of GA and PSO tend to lead to some improvements upon the initial sampling but the final results are clearly outperformed by both SA and DE. The comparison between DE and SA shows that the latter has a better performance in terms of convergence speed during the early stages of the optimization (see the Q measure in Table II) but is eventually outperformed by the DE.

V. REAL-TIME IMPLEMENTATION OF OPTIMIZED DSKF

The experimental setup consists of two coupled 3.70-kW Siemens 1PH7 three-phase induction servomotors, a sin/cos-type encoder, a 4-kW Danfoss VLT50006 three-phase inverter with an Interface and Protection Card IPC-VLT5000, a dSPACE DS1103 digital controller, and a 4-kW Siemens 6SE7 Simovert Master-drives Motion Control three-phase inverter.

The nameplate data and parameters of the induction servomotors are reported in Table IV. The nameplate data of the Danfoss VLT50006 three-phase inverter are reported in Table V. The inverter switching-frequency was put at the value $f_c = 10$ kHz. The IM fed by the 4-kW Danfoss VLT50006 three-phase inverter was speed controlled without sensor. The IM fed by the 4-kW Siemens 6SE7 Simovert Master-drives Motion Control three-phase inverter was torque controlled.

The Simulink model of the control system shown in Fig. 6 has been used to build the real-time code and download it onto dSPACE hardware. Two-phase currents $(i_{sA_m}, i_{sB_m})$ have been acquired by Hall-effect sensors and sampled by A/D converters at frequency of 10 kHz. The sources of errors in this process are dc offsets of the Hall-effect sensors. Values of dc offsets have to be measured prior to each system startup and subtracted from the measured current signals because, otherwise, they generate ac ripple components of fundamental frequency in the $d−q$-axes reference frame. This procedure has been implemented in the block called “dc offset eliminator” and shown in Fig. 6. The voltages $v_{s\alpha}$ and $v_{s\beta}$ were calculated by the inverters’ duty cycles and measured dc-link voltage $V_{dc}$. The phase duty cycles have been computed, as reported in [2]. The estimated rotor speed has been obtained from the optimized-DSKF algorithm using (22).

For the experiments carried out using the speed sensorless-control system based on the DSKF algorithm, the initial values of the state variables and estimation-error covariance matrix were chosen as follows:

$$\hat{x}(0) = \begin{bmatrix} 0 & 0 \end{bmatrix}^T, \quad \hat{P}(0) = 10^{-5} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{27}$$

The speed PI regulator, designed according to the suggestions reported in [22], has been used for the proposed sensorless SFO-SM control scheme. The speed PI and the sliding control parameters were set up, as reported in Table VI. Fig. 7(a) and (b) shows the actual speed response of the sensorless scheme to
a sequence of the speed steps without load torque and evidence the effectiveness of the optimized-DSKF-based algorithm.

Fig. 8 refers to the responses of the sensorless-controlled IM to a speed reversal from +20 to −20 rad/s. The test was carried out by imposing a constant torque reference to the induction machine coupled to the sensorless-controlled IM. There is evidence that the speed reversal is made easy owing to the applied torque. Fig. 8(c) shows that the $q$-axis current component variation causes a transient in the estimated stator-flux linkage that does not influence the speed reversal owing to the optimal performance of the optimized-DSKF-based algorithm. The accurate estimate of electrical angular position of the stator-flux linkage vector is shown in Fig. 8(d) and confirms the optimal characteristics of the proposed sensorless-control scheme.

VI. CONCLUSION

This paper has presented a novel sensorless SFO-SM scheme based on an offline-optimized-DSKF algorithm to estimate the stator-flux components and rotor speed in IMs. The advantages of the proposed sensorless-control scheme are a minimum number of parameters to be offline optimized, robustness to

![Fig. 6. Sensorless SFO-SM control scheme of IM drive using optimized-DSKF-based algorithm.](image)

![Fig. 7. Actual speed responses of the sensorless scheme without load torque.](image)

(a) Actual speed response using the hand-calibrated DSKF-based algorithm.
(b) Actual speed response using the new optimized-DSKF-based algorithm.

<table>
<thead>
<tr>
<th>TABLE VI</th>
<th>SPEED PI AND THE SLIDING CONTROL PARAMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>$k_{ps}$</td>
<td>0.20 Nm s/rad</td>
</tr>
<tr>
<td>$k_{i_d}$</td>
<td>0.47 Nm/rad</td>
</tr>
<tr>
<td>$k_M$</td>
<td>50 V</td>
</tr>
<tr>
<td>$\Phi_\lambda$</td>
<td>0.01 Wb</td>
</tr>
<tr>
<td>$g_r$</td>
<td>1.17</td>
</tr>
<tr>
<td>$\Phi_F$</td>
<td>0.57 $\pi f_i$ rad Nm</td>
</tr>
</tbody>
</table>
parameters uncertainties and noise, and low computational effort in real-time implementation.

The original offline optimization of the covariance matrices has been carried out by means of a DE scheme. The performance of the DE, related to the given problem, has been compared with a traditional optimization algorithm and three popular metaheuristics (GA, SA, and PSO). Numerical results show that the employment of a traditional local-search method is inadequate for this class of problems, likely due to the presence of high multimodalities. The GA and PSO succeed at improving upon their initial population but prematurely converge to suboptimal solution, thus displaying an overall mediocre performance. The SA and DE seem the most promising approaches for this problem. However, unlike the DE, the SA performance heavily depends on the sampling of the starting solution. Thus, the SA could turn out to be not robust. The DE seems to be the most promising approach for the problem under analysis.

The real-time implementation of the novel sensorless-control scheme with offline-optimized DSKF algorithm showed smooth operation at very low speed and load speed reversal, stability at zero speed, and removed both offsets and random noise. The proposed optimized DSKF is a reliable alternative to the other model-based estimators. Further applications of DE function optimizer in the field of IM drive parameter identification are under development.

**REFERENCES**


---

**Nadia Salvatore** (S’04–M’08) received the M.Sc. (with honors) and Ph.D. degrees in electrical engineering from the Technical University of Bari, Bari, Italy, in 2002 and 2006, respectively. During her Ph.D. studies, she was involved in the Italian research project “Signal Processing for Diagnostics of Electrical Machines Fed by Power Converters.”

She worked for one year on the research project entitled “Sensorless Control of Induction Motors” at the School of Electrical and Electronic Engineering, University of Nottingham, Nottingham, U.K. She is currently an Assistant Researcher at the Technical University of Bari. Her primary research interests include the intelligent motion control of electrical machines. In particular, she is interested in vector drives with sliding-mode controllers and techniques of sensorless control of ac drives.

**Andrea Caponio** (S’05) received the M.Sc. degree (with honors) in electrical engineering from the Technical University of Bari, Bari, Italy, in 2005, where he is currently working toward the Ph.D. degree.

Since 2005, he has been working in the Electrical Machine and Drives Laboratory, Technical University of Bari. In 2007, he started to actively collaborate with the University of Jyväskylä, Jyväskylä, Finland, where he spent one year of his Ph.D. degree studies. His research deals with optimization in electric engineering and his main interests include electric drives, memetic algorithms, and evolutionary optimization in the presence of uncertainties.

**Silvio Stasi** received the M.Sc. degree in electrical engineering from the University of Bari, Bari, Italy, in 1989, and the Ph.D. degree in electrical engineering from the Technical University of Bari, Bari, in 1993.

From 1990 to 1993, he was with the Electric Drives and Machines Group, Technical University of Bari, where he carried out research on control and state and parameter estimation of electrical drives and, since November 2002, has been an Associate Professor of electrical machines and drives in the Dipartimento di Elettrotecnica ed Elettronica. He has authored more than 50 scientific publications. His research interests include control of electric drives, fuzzy logic, neural networks, power electronics, motor parameter estimation, and robotics.

**Giuseppe Leonardo Cascella** (S’02–M’06) received the M.Sc. (with honors) and Ph.D. degrees in electrical engineering from the Technical University of Bari, Bari, Italy, in 2001 and 2005, respectively. He was with Getrag GmbH Systemtechnik, St. Georgen, Germany, where he worked in the area of automatic transmissions. He is currently an Assistant Researcher at the Technical University of Bari. His research interests include artificial intelligence for electrical drives.