GPU-Accelerated Parallel Coevolutionary Algorithm for Parameters Identification and Temperature Monitoring in Permanent Magnet Synchronous Machines

Zhao-Hua Liu, Xiao-Hua Li, Liang-Hong Wu, Shao-Wu Zhou, and Kan Liu

***Abstract*—A hierarchical fast parallel co-evolutionary immune particle swarm optimization (PSO) algorithm, accelerated by graphics processing unit (GPU) technique (G-PCIPSO), is proposed for multiparameter identification and temperature monitoring of permanent magnet synchronous machines (PMSM). It is composed of two levels and is developed based on compute unified device architecture (CUDA). In G-PCIPSO, the antibodies (Abs) of higher level memory are selected from the lower level swarms and improved by immune clonal-selection operator. The search information exchanges between swarms using the memory-based sharing mechanism. Moreover, an immune vaccine-enhanced operator is proposed to lead the *P*bests particles to unexplored areas. Optimized parallel implementations of G-PCIPSO algorithm is developed on GPU using CUDA, which significantly speeds up the search process. Finally, the proposed algorithm is applied to multiple parameters identification and temperature monitoring of PMSM. It can track parameter variation and achieve temperature monitoring online effectively. Compared with a CPU-based serial execution, the computational efficiency is greatly enhanced by GPU-accelerated parallel computing technique.**

***Index Terms*—Artificial immune system (AIS), condition monitoring, graphics processing unit (GPU), parallel computing, parameter identification, particle swarm optimization (PSO), permanent magnet synchronous machines (PMSMs), temperature monitoring.**

I. Introduction

IN RECENT years, permanent magnet synchronous motor (PMSM) has been widely applied in servo control, electric vehicles, and renewable energy power generation [1]–[5] for its high power density, torque response, and efficiency. Precise estimation of motor parameters is required to support system modeling, optimization control, state monitoring, and stability analysis in industrial application [6]. For instance, the temperature can be monitored according to the variation of windings resistance in the motor. However, PMSM is typically a nonlinear system, whose physical parameters are easily subjected to changes with variation in operating conditions. Therefore, technologies for parameter estimation of PMSM, especially simultaneous estimation of multiparameters, such as winding resistance, *dq*-axis inductances, and permanent magnet flux, become a challenging work in machine control [7].

Existing research mainly focuses on online estimation algorithms, such as extended Kalman filter (EKF) [8]–[11], model reference adaptive system (MRAS) [12], [13], recursive least-square (RLS) methods [14]–[16], adaptive observer [17], [18], and artificial neural network (ANN) [19], [20]. Each algorithm has some inherent drawbacks of slow convergence, unsteadiness, complex computing, system self-disturbances, and monitoring data growth. In [8], an EKF is proposed to estimate the rotor speed and position of PMSM. It suffers from unidentified PMSM parameters because of noise sensitivity. Similarly, an EKF for the estimation of the winding resistance and rotor flux linkage is proposed in [11]. The results indicate that this method also suffers from noise and instability in engineering. The MRAS estimators proposed in [12] and [13] cannot simultaneously estimate winding resistance, inductance, and rotor flux linkage accurately. The nominal values of other parameters specified in the motor manual are required to estimate individual parameters. Since the nominal value is not equal to the actual operating value, and the working condition changes frequently, those methods cannot ensure the convergence to actual parameter value. The RLS estimator was discussed in [14]. It also suffers from the noise characteristics and could not guarantee the convergence to actual values during experiments. In order to obtain the observation values, an adaptive interconnected observer for PMSM parameter estimation is discussed in [18]. The algorithm may also be unfeasible when the parameters of PMSM vary in large ranges (such as empty running and full-loaded running). Reference [19] presents an ANN to estimate parameters of PMSM. Computation is complicated due to the training of network weight. It is likely to get stuck in local minima because of too much over-fitness. As is pointed in [21], the proposed method should be capable of identifying parameters dynamically and adaptively in time-varying and nonlinear systems.

Manuscript received July 03, 2014; revised December 09, 2014 and February 04, 2015; accepted March 24, 2015. Date of publication April 16, 2015; date of current version October 02, 2015. This work was supported in part by the China Postdoctoral Science Foundation funded project under Grant 2013M540628 and Grant 2014T70767, in part by the National Natural Science Foundation of China under Grant 61174140 and Grant 51374107, and in part by the Hunan Provincial Natural Science Foundation of China under Grant 13JJ8014 and Grant 14JJ3107. Paper no. TII-14-0700.

Z.-H. Liu, X.-H. Li, L.-H. Wu, and S.-W. Zhou are with the School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan 411201, China (e-mail: zhaohualiu2009@hotmail.com).

K. Liu is with the Department of Electronic and Electrical Engineering, University of Sheffield, Sheffield S10 2TN, U.K.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2015.2424073

1551-3203 © 2015 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

Inspired by biological computing, evolutionary computation techniques are proposed to solve this problem. Genetic algorithm (GA) and particle swarm optimization (PSO) algorithm have attracted much attention in the investigation of PMSM parameter identification. It is because the evolutionary algorithms have the ability to obtain a suitable set of parameter values via optimizing objective function between the system ideal model and the actual models. GA was proposed to identify multiparameter of PMSM including stator winding resistance, *dq*-axis inductances, and rotor flux linkage. However, the accuracy and stability of these parameters estimation need to be improved, since GA tends to lose diversity and easily get trapped in local optima [22], [23] during the later evolution process when solving complex problems. Some researchers propose an intelligent system parameter identification approach by utilizing the PSO algorithm [24], [25]. Since PSO is easy to implement on the computer and superior in convergence speed, the intelligent estimator is effective in estimating the stator resistance and disturbed load torque. However, it cannot exactly estimate multiple parameters simultaneously (winding resistance, *dq*-axis inductances, and permanent magnet flux identification). The reason is that the basic PSO can be easily trapped in local minima when solving complex nonlinear problem.

Hence, several algorithms are developed to improve the solution performances of PSO. A hybrid PSO with mutation was proposed by Ahmed *et al.* in [26]. Juang *et al.* proposed another hybrid PSO algorithm associated with the genetic operator in [27]. The diversity of PSO is significantly enhanced using the GA operators. Compared with the basic PSO, a hybrid PSO combined with wavelet dynamic mutating space was proposed in [28], which can obtain dynamic optimization effect. Liang *et al.* proposed a comprehensive learning PSO (CLPSO) in [29]. All the flying directions of individuals were updated by randomly selected particles during the iteration process. Although the method in [29] was superior in keeping diversity, it did not design a scheme to help the particles jump out of the local optima when the whole population was homogeneous during the later evolution process. The same defect also exists in another improved CLPSO using different *p*bests [30]. A parameter-adaptive regulation scheme and an elitist learning strategy were introduced into the PSO by Zhan *et al.* [31], who presented an adaptive PSO (APSO) that can accelerate the convergence speed and jump out of the local optima. Nevertheless, it still needs expert judgment on the evolutionary state, which may be difficult to determine. As the spatial distribution of multiple populations was broad and outperformed a single population in terms of diversity, other researchers proposed a multipopulation scheme to improve the diversity of basic PSO [32]–[35]. For instance, Bergh *et al.* [32] proposed a multiple PSO algorithm, in which the whole population was split into many small swarms. It showed a better performance than single PSO. However, this method is redundant in iteration computing because there is no information interaction between these isolated swarms. A coevolutionary PSO was discussed in [33], where a truncated Gaussian distribution function is utilized to accelerate the convergence speed. Reference [34] proposed a dynamic multiswarm PSO embedded with an adaptively adjusted scheme. In order to solve multiple-objective problems, Zhan *et al.* [35] proposed a multiple populations PSO, which used an external shared archive and another two novel operator, including velocity modification and elitist learning strategy to enhance the performance. However, coevolutionary PSO always has mass data to analyze during machine operation. The proposed methods, executed in conventional CPU, may take considerably long time. In fact, it does not take full advantage of the inherent parallelism of population-based intelligent computing techniques. This disadvantage severely restricts the practical applications of PSO in parameters estimation and condition monitoring of PMSM. Therefore, the GPU-accelerated massively parallel computing technique becomes a major complementary way to speedup those population-based intelligent algorithms.

Motivated by coevolution theory [36], artificial immune system (AIS) [37], [38], and GPU parallel computing technique [39], a novel parallel coevolutionary immune PSO algorithm, accelerated by graphics processing unit (GPU) technique (G-PCIPSO), is proposed for multiparameter identification and temperature monitoring of PMSM in this paper. The framework of G-PCIPSO consists of one high-level memory population and several bottom normal swarms based on parallel computing model of GPUs. In G-PCIPSO, the global best individuals of normal swarms are regarded as antibody (Ab) in immune system and memorized in the high-level memory, which serves as a leader set. The memory is updated using the improved immune clonal-selection operator. Then an immune vaccine-enhanced operator is employed to accelerate the convergence speed of *P*bests. The information sharing mechanism is employed for the information exchange between memory and different swarms. Moreover, the proposed method is implemented on the GPU using the compute unified device architecture (CUDA) (NVIDIA Corporation). Finally, the proposed method is applied to PMSM multiple parameter identification and temperature monitoring. It shows that the method's performance is much better than the existing improved hybrid PSOs in simultaneously estimating multiple electric parameters of the machine system. In addition, the proposed method can not only track the varied physical parameter but also realize temperature monitoring online effectively. The experiments show that the proposed approach demonstrates high efficiency compared with a CPU-based serial execution.

Our main contributions can be summarized as follows.

1) The proposed hierarchical-based multiple population cooperative scheme can improve the population search performance since the immunity-based high-level memory can reserve valuable historical knowledge, and the excellent solution information can spread between different subswarms through the designed information sharing updated scheme of particles in bottom level.
2) Immune concepts are introduced into PSO to overcome the blindness in action of *g*Best particles (antibodies in memory) and *P*best particles stochastic evolution in solving parameters identification and temperature monitoring of PMSM. Immune clonal-selection operating with an adaptive wavelet mutation for *g*Best and an immune vaccine operator for *P*best are introduced. The proposed parameter estimator is capable of identifying parameters dynamically and adaptively for time-varying and nonlinear PMSM system.

3) High-performance computing ability of GPU is fully utilized to speedup the search process of particles and Abs. Fast and exact parameter estimation and temperature monitoring for PMSM under changing conditions is realized.

This paper is organized as follows. In Section II, a brief introduction of PMSM model is provided and challenges of designing parameter estimators for PMSM is analyzed. G-PCIPSO algorithm for parameters estimation and temperature monitoring of PMSM is proposed in Section III, where algorithm principle, mathematical model, and implementation procedure are addressed in detail. Section IV presents parameter estimation mechanism modeling based on the proposed G-PCIPSO. Experiment results are given in Section V. Finally, conclusion and future work are summarized in Section VI.

II. PMSM Model

The mathematical model of the PMSM in *dq*-axis frame is given as

$$\begin{bmatrix} U_d \\ U_q \end{bmatrix} = \begin{bmatrix} R + L_d \dfrac{d}{dt} & -\omega L_q \\ \omega L_d & R + L_q \dfrac{d}{dt} \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} + \begin{bmatrix} 0 \\ \omega\psi \end{bmatrix} \tag{1}$$

where $\dfrac{d}{dt}$ is a differential operator, ω is the electrical angular velocity and R, L_d, L_q, and ψ are the motor winding resistance, d-axis and q-axis inductances, and magnet flux, respectively. U_d, U_q, i_d, and i_q are *dq*-axis stator voltage and current. The motion equation can be expressed in discrete form as follows when the machine is on the steady-state model [4]

$$\begin{bmatrix} U_{d(k)} \\ U_{q(k)} \end{bmatrix} = \begin{bmatrix} R & -\omega(k)L_q \\ \omega(K).L_d(K) & R \end{bmatrix} \begin{bmatrix} i_{d(k)} \\ i_{q(k)} \end{bmatrix} + \begin{bmatrix} 0 \\ \omega(k).\psi(k) \end{bmatrix} \tag{2}$$

where R, L_d, L_q, and ψ are unknown parameters. From (2), it is clear that there are four parameters to be identified, while both rank numbers are two and coupled with each other [7], [8]. Generally, i_d is set to be zero for decoupling the flux and torque control and a very short time of $i_d < 0$ should be injected to obtain a full-rank reference model inspired by [7] and [22], which are shown as follows:

$$\begin{bmatrix} U_{d0(k_0)} \\ U_{q0(k_0)} \\ U_{d(k_1)} \\ U_{q(k1)} \end{bmatrix} = \begin{bmatrix} R & -\omega(k0)L_q & 0 & 0 \\ \omega(k0).L_d(k0) & R & 0 & 0 \\ 0 & 0 & R & -\omega(k1)L_q \\ 0 & 0 & \omega(k1).L_d(k1) & R \end{bmatrix} \times \begin{bmatrix} i_{d0(k0)} \\ i_{q0(k0)} \\ i_{d1(k1)} \\ iq_{1(k1)} \end{bmatrix} + \begin{bmatrix} 0 \\ \psi\omega(k_0) \\ 0 \\ \psi\omega(k_1) \end{bmatrix}. \tag{3}$$

Equation (3) is the full-rank model of PMSM and will be used for the design of objective function of multiparameter estimator. id_0 and id_1 means the *id*-axis is injected with the current equal and unequal to zero, respectively. It is a challengeable work to estimate these parameters simultaneously since the PMSM operates under various operating conditions. Noise, disturbance, and parameter variations are unavoidable and immeasurable in many applications. Therefore, it is very important to develop an efficient model parameter tracking approach for PMSM multiparameter identification.

III. GPU-Accelerated Parallel Immune Coevolutionary PSO

A. Principle of Basic PSO Algorithm

According to the social intelligent behavior of bird flocking, the particles fly toward better searching areas during the evolutionary process. Assuming in a d-dimensional solution space, each particle i is composed of two vectors, which are the velocity vector $\boldsymbol{V}_i = \{\boldsymbol{V}_{i1}, \boldsymbol{V}_{i2}, \ldots, \boldsymbol{V}_{id}\}$ and the position vector $\boldsymbol{X}_i = \{\boldsymbol{X}_{i1}, \boldsymbol{X}_{i2}, \ldots, \boldsymbol{X}_{id}\}$, the searching procedure can be given as

$$\begin{aligned} \boldsymbol{V}_{id}(t+1) = \omega\boldsymbol{V}_{id} &+ c_1 * \mathrm{rand}_1()\,(P\mathrm{best}_{id}(t) - \boldsymbol{X}_{id}(t)) \\ &+ c_2 * \mathrm{rand}_2()\,(g\mathrm{Best}_d(t) - \boldsymbol{X}_{id}(t)) \end{aligned} \tag{4}$$

$$\boldsymbol{X}_{id}(t+1) = \boldsymbol{X}_{id}(t) + \boldsymbol{V}_{id}(t+1). \tag{5}$$

In (4) and (5), $P\mathrm{best}_{id}$ represents the best position found by ith particle up to now and $g\mathrm{Best}_d$ is the best particle among the entire population. c_1 and c_2 are the acceleration coefficients, ω is the inertia weight factor decreasing linearly, and rand_1 and rand_2 are two uniformly distributed numbers generated randomly within [0, 1], respectively.

B. Principle of G-PCIPSO Algorithm

The general steps of G-PCIPSO are stated as follows.

Algorithm 1. G-PCIPSO

Step1: Initialize the normal sub-swarms P_i and antibodies memory; set up parameters for G-PCIPSO.
Step2: Transfer data from CPU to GPU.
// sub-processes in "for" are done in parallel
Step3: for $i = 1$ to I//$1 \le i \le I$, I is the number of normal sub-swarms;
Perform the process of PSO for sub-swarms *Pi*
for $j = 1$ to N//$1 \le j \le N$, N is the number of particles //sub-swarms *Pi*
update $particle_i$ velocity using the equation (4)
update $particle_i$ position using the equation (5)
Evaluate the fitness value of $particle_i$;
end for
end for
Step4: for $i = 1$ to k//$1 \le i \le k$, k is the number of *Pbests*
Perform the vaccine-enhanced for $Pbest_i$ based on the equations (9)–(11) and the Fig. 2.
end for

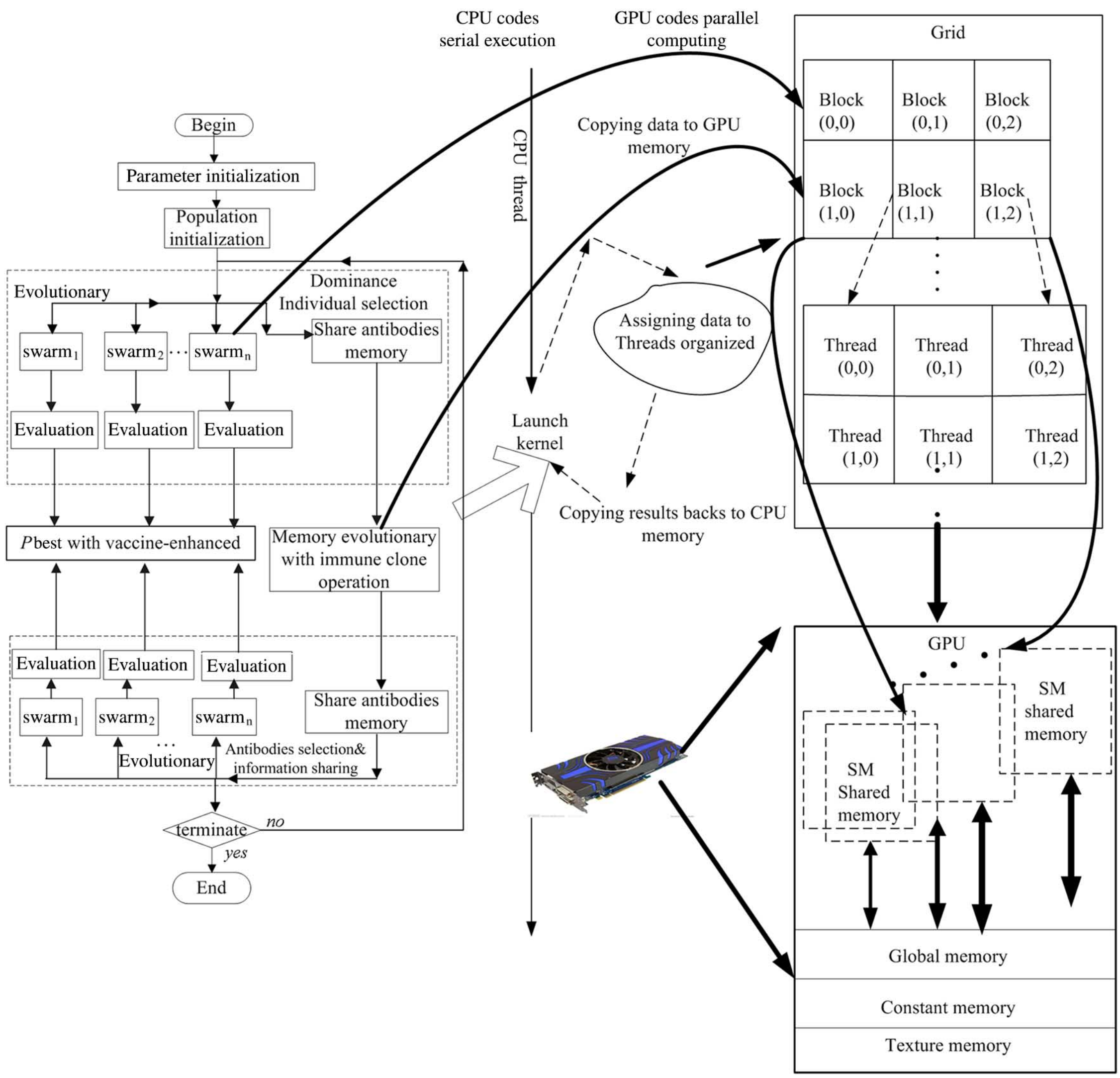

Fig. 1. Model of G-PCIPSO.

Step5: The best individuals of each normal swarm are selected into memory and replace antibodies memory.

Step6: for $i = 1$ to $k//1 \leq i \leq k$, K is the number of antibodies in //memory

Perform the process of immune clonal selection with adaptive wavelet hymutation for $antibody_i$ based on (6)–(9).

end for

Step7: Perform the process of information sharing operation based on (13)–(14);

Step8: Until a terminate-condition is met, or else, return step3.

Step9: Transfer result back to CPU and output.

The proposed G-PCIPSO method involves three key strategies.

1) The method has two-level hierarchical coevolutionary framework. The high-level memory is composed of Abs, which are selected from different swarms. The lower level is made up of multiple swarms. A memory-based information-sharing mechanism is designed to speedup information exchanging between swarms.
2) The AIS mechanisms are introduced into the coevolutionary PSO framework and an adaptive wavelet hyper-mutation-based immune clonal-selection operator is applied to improve the dynamic optimization performance of the Abs in high-level memory. Also, an immune vaccine operator is used to expand the search space of pbests.
3) GPU parallel computing technique is used to speedup the search process and then an optimized parallel G-PCIPSO algorithm using CUDA is developed.

The design process is detailed as follows.

1) The best individuals selected from the subswarms with high fitness are saved into the higher level memory, and then the memory is optimized by the improved immune clonal-selection operating with an adaptive wavelet mutation. Furthermore, an information-sharing-based updating modified the equation for the lower level subswarms to boost the excellent solution information interacting between different subswarms.
2) An immune vaccine operator is designed to expand the search space of *p*bests when the *P*best particles nearly reach a local optimum point during the evolution process.
3) To speedup the search process of the proposed method, an optimized parallel G-PCIPSO algorithm is implemented on GPU using CUDA.

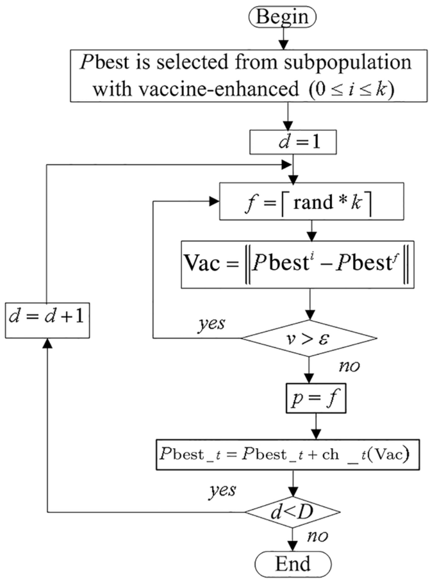

Fig. 2. Vaccine-enhanced operator for *P*bests.

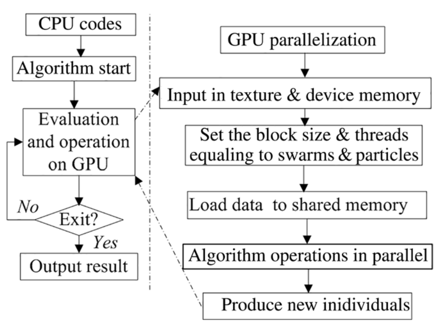

Fig. 3. PCIPSO running on GPU.

The model of G-PCIPSO and parallel implementation is shown in Fig. 1. The populations and related parameters are generated on CPU side and allocated to a grid with one dimension of blocks at GPU side. One block handles one swarm or memory population, whereas one thread handles the particle in the swarm or the Ab in memory. The data and parameters of G-PCIPSO are copied from CPU to GPU before the kernel function is launched. The data of the algorithm is copied to the global memory of GPU and then copied to the shared memory (SM), the access to which is faster. The parameters of the algorithm are copied to the constant memory, as all individuals in the algorithm can share the same utility function to evaluate their fitness, and all the threads can access to the constant memory in the grid at the same time.

C. Memory-Based Information Sharing and Evolutionary Strategy

In order to effectively use the available historical knowledge, an information exchange strategy is implemented between the swarms and higher level memory. The higher level memory is composed of the best individuals selected from the lower level swarms with high fitness and regarded as Abs set

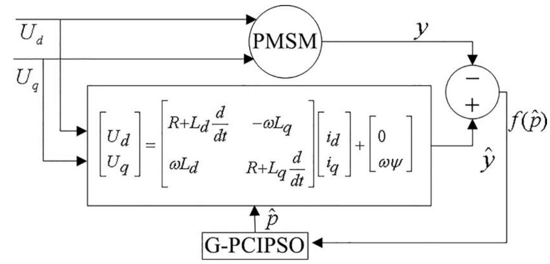

Fig. 4. Parameter identification model based on G-PCIPSO.

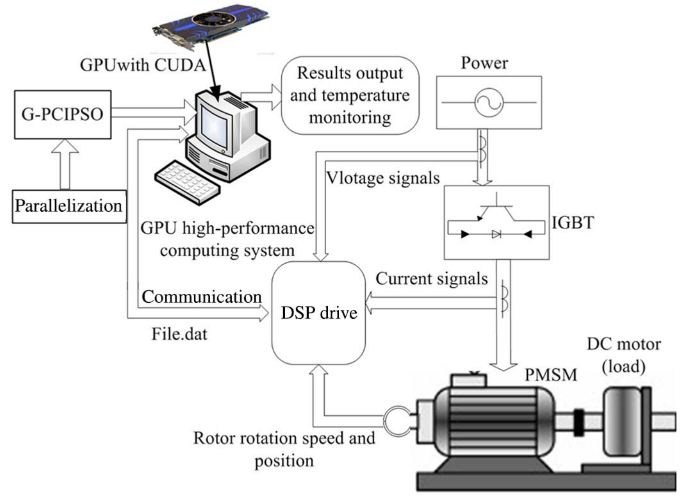

Fig. 5. Schematic diagram of identification and condition monitoring system.

TABLE I
DESIGN PARAMETERS AND SPECIFICATION OF PMSM

Rated speed	400 rpm
Rated current	4 A
DC link voltage	36 v
Nominal terminal wire resistance	0.043
Nominal self-inductance	2.91 mh
Nominal mutual inductance	–0.330 mh
Nominal *d*-axis inductance	3.24 mh
Nominal *q*-axis inductance	3.24 mh
Nominal amplitude of flux induced by magnets	77.6 mWb
Number of pole pairs	5
Nominal phase resistance (T=25°C)	0.330 Ω
Inertia	0.8e−5 kgm^2

$\mathrm{Ab} = \{\mathrm{Ab}_{1d}, \mathrm{Ab}_{2d}, \ldots, \mathrm{Ab}_{Kd}\}$ (K being the memory size). The information sharing is updated as follows:

$$\begin{aligned}\boldsymbol{V}_{id}(t+1) = \omega\boldsymbol{V}_{id} &+ c_1 * r_1\left(P\mathrm{best}_{id}(t) - \boldsymbol{X}_{id}(t)\right)\\ &+ c_2 * r_2\left(g\mathrm{Best}_d(t) - \boldsymbol{X}_{id}(t)\right)\\ &+ c_3 * r_3\left(Ab_{\varphi[d]} - \boldsymbol{X}_{id}(t)\right)\end{aligned} \tag{6}$$

$$\boldsymbol{X}_{id}(t+1) = \boldsymbol{X}_{id}(t) + \boldsymbol{V}_{id}(t+1) \tag{7}$$

where c_1, c_2, and c_3 are constant, and the term $c_3 * r_3\left(\mathrm{Ab} - \boldsymbol{X}_{id}\left(t\right)\right)$ is the sharing information from the higher level memory. The symbol φ is the randomly selected Ab and $\varphi = \lfloor \mathrm{rand} * K \rfloor$. With the help of excellent solution information in the higher level memory, the particle can take advantage of the search information not only from its own population

TABLE II
COMPARISON RESULTS AMONG SEVEN PSOS ON PMSM PARAMETER IDENTIFICATION WITH NORMAL TEMPERATURE

		HPSOM	HGAPSO	HPSOWM	CLPSO	A-CLPSO	APSO	G-PCIPSO
R (Ω)		0.354	0.338	0.367	0.376	0.402	0.367	**0.3729**
ψ (Wb)		0.0785	0.0790	0.0781	0.0781	0.0772	0.0780	**0.0779**
Ld (h)		0.00353	0.00307	0.00377	0.00379	0.00371	0.00344	**0.00290**
Lq (h)		0.00389	0.00395	0.00387	0.00390	0.00377	0.00381	**0.00383**
Fitness	Mean	0.596	2.558	0.987	2.86	2.703	4.503	**0.197**
	Std.dev	0.881	1.19	1.077	1.58	0.995	2.598	**0.06**
	Time (s)	27.23	27.859	27.91	34.94	34.58	34.31	**26.05**
	t-value	3.09	13.74	5.06	11.78	17.29	11.67	**0**

but also from other populations. The excellent Ab is selected randomly from the higher level memory to update the ith particle. The information sharing operation strategy not only uses historic information effectively but also enhances the feasible solution information interactions between the memory and different swarms. It can make a balance between extensive searching and accurate searching. In order to explore better searching regions and carry out fine search, all the individuals in memory will be optimized by the improved immune clonal-selection operating with an adaptive wavelet mutation operator, and the whole memory population will be updated with the optimized results frequently.

The evolutionary strategy of the higher level memory is stated as follows.

1) Clone: In each generation, the gBest individuals in the higher level memory population are seen as Abs in immune systems. The individuals' clonal scale is proportional to its fitness. The clonal scale is

$$N_c = \sum_{i}^{N} \text{round}\left(\frac{\beta N}{i} + C\right) \tag{8}$$

where N is the memory size and β is on the interval (0, 1), with the value of 0.8 in this paper. In addition, C is the fairness factor usually with the value larger than 1 (C is fixed to 5 in this paper). The clonal scale makes sure that the worst individual has the chances to be cloned.

2) Adaptive wavelet mutation: After the clonal expansion, the individuals constitute a temporary clonal swarm. A wavelet mutation operator is employed to improve the dynamic search performance of cloned Ab since the wavelet mutating space dynamically varies with the properties of the wavelet function. The Morlet wavelet features can be referred to [28]. Here is an example of the mother wavelet

$$\phi = \frac{1}{\sqrt{a}} e^{-\left(\frac{\lambda}{a}\right)^2 / 2} \cos\left(5\left(\frac{\lambda}{a}\right)\right) \tag{9}$$

in which a is the dilation parameter for mother wavelet and $\lambda \in [-2.5a, 2.5a]$ and λ will be randomly generated during the iterative process.

From (9), we can see that the amplitude ($\phi(a)$) of the function can be adjusted through controlling the dilation parameter a. Using this feature, we can adjust the amplitude of the wavelet function in the searching space adaptively. Therefore,

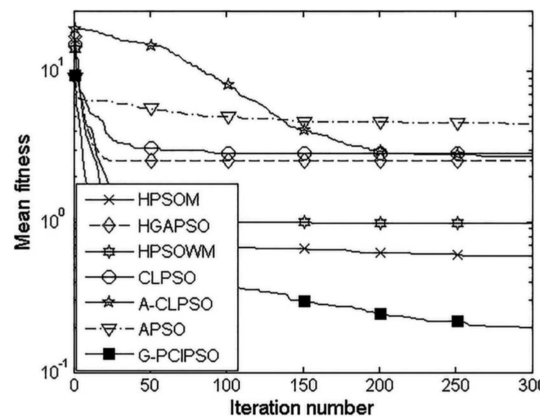

Fig. 6. Fitness convergence curve of seven PSOs on PMSM parameter identification under normal temperature.

an adaptive nonlinearly increasing strategy for the parameter a is proposed and embedded in the following equation:

$$a = a_{\max} - (a_{\max} - a_{\min})\left(\frac{T-t}{T}\right)^2 \tag{10}$$

where $a_{\max}$ and $a_{\min}$ are the upper and lower boundaries of a ($a_{\max} = 500$, $a_{\min} = 5$ in this paper). Using (9) and (10), all the temporary Ab clonal population will be operated by (11) in each generation, i.e.,

$$g\text{Best}_d{}^{\text{new}} = g\text{Best}_d + \phi * g\text{Best}_d. \tag{11}$$

3) Clonal selection: After the hyper-mutation, the newly generated individuals will be reselected according to the fitness; the details were presented in [37]. It will result in an updated higher level memory.

D. Vaccine-Enhanced for Pbests

The *P*best particles can be potentially used as the exemplars to lead the flying direction of other particles among the swarms. Hence, the search status of *P*best particles is of importance for the PSO. *P*best particles easily come closer to each other to reach a local optimum point during the evolution process. In order to accelerate the *P*best particles convergence speed and prevent the *P*best particles from deterioration, a vaccine mechanism in AIS is introduced to help *P*best push itself out to a

TABLE III
COMPARISON RESULTS AMONG SEVEN PSOS ON PMSM PARAMETER IDENTIFICATION UNDER TEMPERATURE VARIATION

		HPSOM	HGAPSO	HPSOWM	CLPSO	A-CLPSO	APSO	G-PCIPSO
$R(\Omega)$		0.478	0.529	0.578	0.479	0.493	0.476	**0.454**
$\psi(wb)$		0.0762	0.0749	0.0734	0.0764	0.0759	0.0765	**0.0769**
$Ld(h)$		0.0032	0.00356	0.00371	0.00385	0.00393	0.00387	**0.00306**
$Lq(h)$		0.00367	0.00355	0.00336	0.00378	0.00367	0.00378	**0.00376**
Fitness	Mean	0.493	2.89	2.12	2.073	1.379	3.594	**0.234**
	Std.dev	0.518	1.166	1.53	1.032	0.569	4.04	**0.063**
	Time (s)	27.63	27.859	28.69	34.22	34.34	34.28	**26.18**
	t-value	3.182	15.745	8.601	12.244	13.019	5.869	**0**

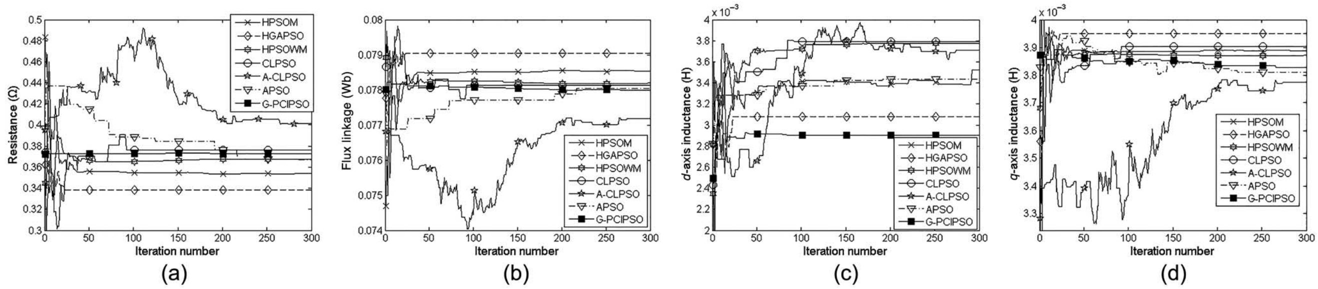

Fig. 7. Identified parameters under normal temperature condition. (a) Winding resistance. (b) Rotor flux linkage. (c) *d*-axis inductance. (d) *q*-axis inductance.

potentially better region by acquisition of population excellent prior knowledge. The vaccines can activate the Ab population to incorporate diverse antibodies and explore new landscape when introduced to the main body. The vaccine construction operator is as follows:

$$\mathrm{Vac} = (P\mathrm{best}^i - P\mathrm{best}^p). \tag{12}$$

If $\|\mathrm{Vac}\| \prec \lambda$ (λ is the threshold value on [0, 1]), the vaccine-enhanced operator will be executed as

$$P\mathrm{best\_}t = P\mathrm{best\_}t + \mathrm{ch\_}t(\mathrm{Vac}) \tag{13}$$

where $\mathrm{ch\_}t$ is the density function expressed as

$$\mathrm{ch\_}t(x) = \frac{1}{\pi}\frac{t}{t^2 + x^2}, \quad -\infty < x < \infty, t > 0 \tag{14}$$

where the subscript t is the current number of evolutionary generations, x is the input variable, and Vac is used as inputs for function $\mathrm{ch\_}t$. Compared with other random mutation operators, the ch operator performs better due to its higher probability of making a longer jump [40]. From the description above, it is quite clear that the proposed vaccine-enhanced operator gives the *P*best chance to exploit their search landscape as shown in Fig. 2.

E. Parallelization Implementation Based on GPU-CUDA

The GPU program includes two parts: 1) CPU control program that controls the process of the whole program and does the sequential work; and 2) GPU parallel program. C language is provided by NVIDIA CUDA technology to access the computing resources of multicore parallel processing GPUs. Based on the principle shown in Fig. 1, the PCIPSO is run on GPU based on the CUDA. The flowchart is demonstrated in Fig. 3. Meanwhile, the workload should be equally distributed among GPU cores. Each block corresponds to a swarm or a memory population, and each thread corresponds to a particle or an Ab in the high-level memory. It can significantly enhance the computing efficiency and speedup the PCIPSO through the CUDA.

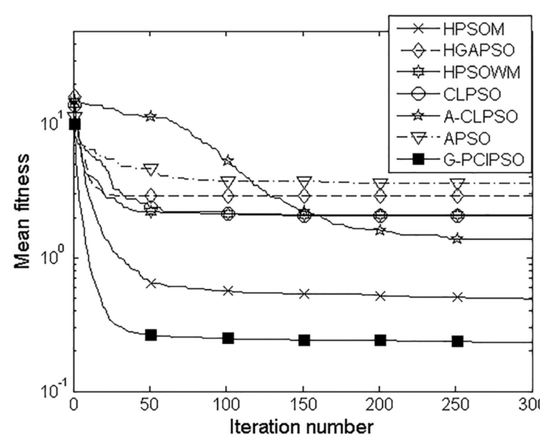

Fig. 8. Fitness convergence curve of several PSOs on PMSM parameter identification under temperature variation.

IV. PARAMETER ESTIMATION MECHANISM MODELING BASED ON G-PCIPSO

The parameter identification can be seen as a kind of systemic optimization of a system with known model structure but unknown parameters. The basic idea is to compare the real system output with the ideal mathematic model output. The

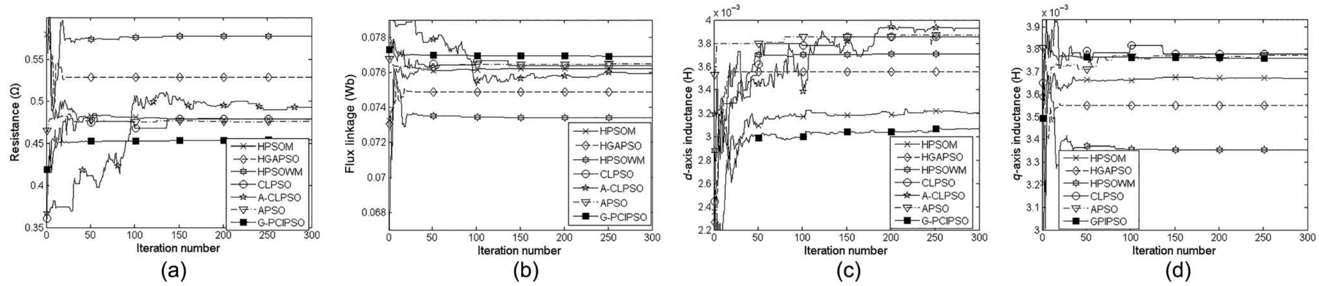

Fig. 9. Identified parameters under temperature variation. (a) Winding resistance. (b) Rotor flux linkage. (c) d-axis inductance. (d) d-axis inductance.

resultant error is minimized through optimization algorithm based on objective function. As described in Section II, the parameter vector $\boldsymbol{p} = \{R, L_d, L_q, \psi\}$ will be identified based on the PMSM model structure. The fitness function for the PMSM parameters identification is constructed as in (15). The structure of G-PCIPSO-based parameter identification can be roughly illustrated in Figs. 4. and 5.

The parameter identification could be regarded as multimodal optimization with restrictions. Therefore, the average fitness function over M runs is defined as

$$\begin{cases} \min F = \dfrac{\sum\limits_{i=1}^{M} f(p)}{M} \\ \text{s.t.} \\ R_{\min} \le R \le R_{\max} \\ L_{d\min} \le L_d \le L_{d\max} \\ L_{q\min} \le L_q \le L_{q\max} \\ \psi_{\min} \le \psi \le \psi_{\max} \end{cases} \tag{15}$$

where R is on (0, 1), L_d and L_q are on (0, 0.0512), and ψ is on (0, 0.1024). Based on (4) and Fig. 4, the fitness function is built as

$$\begin{aligned} f(p) = \rho . \sum_{k=1}^{n} (w_1 \, |u_{d0}(k) - \hat{u}_{d0}(k) + w_2| \, |u_{q0}(k) - \hat{u}_{q0}(k)| \\ + w_3 \, |u_d(k) - \hat{u}_d(k)| + w_4 \, |u_q(k) - \hat{u}_q(k)|) \end{aligned} \tag{16}$$

where ρ is constant, n is the length of the measured data, w_1, w_2, w_3, w_4 are the weight coefficients subject to $0 < w_1, w_2$, $w_3, w_4 < 1, w_1 + w_2 + w_3 + w_4 = 1$, and $wi = \frac{fi}{\sum_{i=1}^{n} fi}$, in which f_i is the ith fitness function, the variables with "^" means voltage calculated through the estimated parameters and measured currents. The actual values of electromagnetic parameter of the PMSM values can be obtained when (15) is minimized by the proposed G-PCIPSO.

V. Experimental Results

A. Hardware and Software Platform for Experiments

To perform our experiment, a PMSM parameter identification testing system is built as in Fig. 5 and Table I, where a PMSM prototype and DSP-based vector control hardware platform are used as the experimental facility. It can be seen from

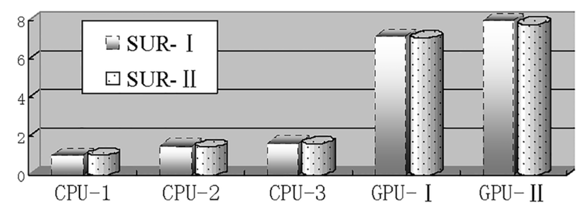

Fig. 10. SUR of G-PCIPSO for PMSM parameter identification based on multicore CPU and GPUs.

Fig. 5 that the memorized data from the electric signal are computed iteratively in host computer by the proposed G-PCIPSO, until the identification results converge to the final value.

A series of hybrid PSOs are used for comparison with the G-PCIPSO. The existing hybrid PSOs, involving HPSOM (hybrid PSO with mutation) [26], HGAPSO (hybrid PSO with GA) [27], HPSOWM (hybrid PSO with wavelet mutation) [28], CLPSO [29], A-CLPSO (an improved CLPSO) [30], and APSO [31], are usually used to achieve better performance in unimodal and multimodal functions optimization. The comparisons are made in terms of the mean results, standard deviation, and the t-test value. The basic settings of these PSOs are as follows: the maximum iteration is 300 and the number of runs is 30. All the hybrid PSOs are operated on the same platform with the same objective function and PMSM hardware and software platform. All experiments are carried out on the same computer with AMD Phenom II four-core processors, DDR III 4G 1600 memory, and NVIDIA Testal. The HPSOM, HGAPSO, HPSOWM, CLPSO, A-CLPSO and APSO are executed on CPU based on serial execution. The G-PCIPSO for PMSM multiple parameters estimation are implemented on a parallel multicore GPU with Testal k20c framework using the CUDA 5.5.

B. Parameter Estimation Under Normal Temperature Condition

The experimental results of PMSM multiparameter identification under normal temperature environment are shown in Table II, and the convergence rates of different PSOs are shown in Fig. 6. It is clear that the G-PCIPSO shows the best performances in terms of mean, standard deviations, and t-values. In addition, all the t-values are higher than 2.5, which imply that the G-PCIPSO has significantly better solution performance than other hybrid PSOs (the confidence level is 98%). From

TABLE IV
COMPUTING TIME FOR PMSM PARAMETER IDENTIFICATION BASED ON G-PCIPSO WITH DIFFERENT PLATFORMS

Operating condition	CPU-with one core		CPU-with two cores		CPU-with three cores		GPU-GTX560TI		GPU-Testal	
	Time (s)	SUR	Time (s)	SUR	Time (s)	SUR	Time (s)	SUR	**Time (s)**	**SUR**
Normal temperature	207.195	1	140.205	1.48	127.06	1.64	29.01	7.14	**26.05**	**7.95**
Variation temperature	203.02	1	141.19	1.44	125.42	1.62	28.77	7.06	**26.18**	**7.75**

Fig. 6, the convergence speed of G-PCIPSO is faster than other hybrid PSOs. The statistical results in terms of mean fitness, standard deviation, and *t*-value are shown in Table II. Moreover, as shown in Table II, the execution time of G-PCIPSO is the shortest among these seven methods. The comparison result is the same as in Table III. The reason for this fact lies in two aspects.

1) Our added operations (hierarchical-based multiple population, immune clone, and immune vaccine) have parallel features.
2) Optimization on the parallel implementation of the G-PCIPSO on GPU framework including reasonable task assignment, minimized data-dependent, and minimized communication cost is performed further.

As demonstrated in Table II, the estimated winding resistance (0.3729Ω) agrees well with the measurement value (0.373Ω) under normal temperature. It is indicated that the proposed G-PCIPSO is of high precision and the estimated parameters involving motor resistance, *dq*-axis inductances, and the rotor flux are converging to their right points rapidly. As can be seen from Fig. 7, four plots appear relatively stable when compared to other hybrid PSOs, which means that the proposed parameter identification method is feasible with high solution accuracy and good robustness when solving dynamically nonlinear problem.

C. Parameter Estimation Under Temperature Variation Condition

In order to examine whether the proposed method can track the varied parameters effectively under the working condition environment with temperature variation or not, a heater is used to heat the prototype PMSM for 20 min and identification experiment is carried out. The performance comparisons of different PSOs are shown in Table III and Fig. 8. The convergence curves of different PSOs are shown in Fig. 8. From Table III, it is evident that G-PCIPSO shows best performances in terms of mean, standard deviations, and *t*-values. Furthermore, all the *t*-values are higher than 3.0, which imply that the G-PCIPSO is significantly better than other hybrid PSOs (the confidence level is 98%). From Fig. 8, it also shown that the G-PCIPSO has a fast convergence speed compared to other hybrid PSOs. Therefore, the proposed G-PCIPSO is much better in solution qualities and can accurately track the variation of parameters. Additionally, the steadiness of the G-PCIPSO is better than other hybrid PSOs. Meanwhile, as can be seen from Table III and Fig. 9, the estimated winding resistance R, d-axis inductance L_d, q-axis inductance L_q, and rotor flux linkage ψ will vary with the changing temperature. The experimental results indicate that G-PCIPSO has a good dynamic tracking performance. Hence, the G-PCIPSO is significantly better and statistically more robust than other listed hybrid PSOs in terms of global search capacity and local search precision in our experiments.

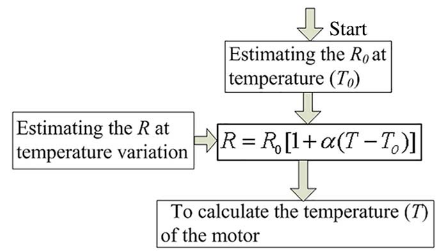

Fig. 11. Schematic diagram of Stator winding temperature monitoring.

D. Computational Efficiency Evaluation for the Parameters Estimation Using the G-PCIPSO Based on Multicore

In this section, the proposed G-PCIPSO is implemented on central processing unit (CPU) with one core (CPU-1), two cores (CPU-2), three cores (CPU-3), GPU-GTX560TI, and GPU-Testal, respectively. The speed-up ratio (SUR) is defined as $\mathrm{SUR} = T\mathrm{s}/T\mathrm{p}$, where Ts and Tp are the execution runs of the serial and parallel algorithms, respectively. In Fig. 10, the symbol SUR-I and SUR-II denotes the time cost of the G-PCIPSO for PMSM parameter identification using CPUs and GPUs under temperature normal operating condition and temperature variation operating condition.

From the result shown in Table IV and Fig. 10, it is clear that the execution time of the G-PCIPSO decreases greatly when the number of CPU cores increases. The average time required for one core, two cores, and three cores CPU under the normal temperature condition is 207.195 s, 140.205 s, and 127.06 s, respectively, whereas the average time required for GPU-GTX560TI and GPU-Testal is 29.01 and 26.05, respectively. It is clear that the SUR is greatly enhanced by the GPUs ($7.14\times$ for GPU-GTX560TI and $7.95\times$ for GPU-Testal) since the computing speed of GPU with hundreds of threads is much faster than that of CPUs.

E. Online Temperature Monitoring

In this section, the proposed G-PCIPSO is applied to monitor the temperature by estimating the resistance of the windings based on the relationship between temperature and copper windings. Except for nearby absolute zero and melting point, there is a strict linear relationship between copper resistance and temperature (as given in Fig. 11). To estimate temperature

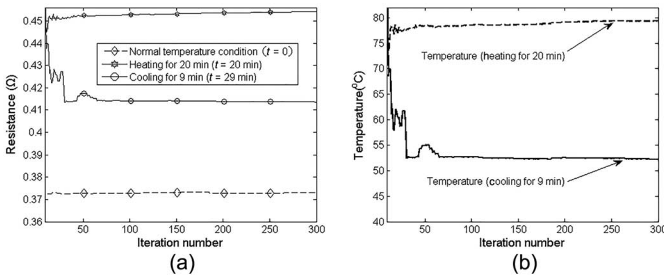

Fig. 12. PMSM stator winding temperature monitoring with G-PCIPSO. (a) Estimated resistance under temperature variation. (b) Temperatures of stator winding under variation condition.

T, steps of G-PCIPSO temperature estimation are stated as follows: 1) stator winding resistance under normal temperature (T_0) is estimated; 2) stator winding resistance under the corresponding temperature is estimated; and 3) motor temperature can be calculated according to the aforementioned principle. In Fig. 11, R is the copper resistance under the temperature T, R_0 is the metal resistance under the temperature $T_0 (T_0 \text{ is } 25°\mathrm{C})$, α is the temperature coefficient of resistance (α is 0.004). The estimation of winding resistance under various temperatures by the proposed G-PCIPSO is shown in Fig. 12. We can see that the stator winding temperature increases gradually when the heater works and it decreases gradually when the heater is cut off, which does consistent with the well-known natural law. Indeed, online temperature monitoring of PMSM can be achieved effectively using the proposed method.

VI. Conclusion

To solve the problem of PMSM multiple parameter identification and temperature monitoring, a novel parameter identification approach based on GPU-accelerated parallel hierarchical-based cooperative immunity PSO is proposed in this paper. The computational efficiency of the proposed method is greatly enhanced using the GPU parallel computing technique. In our experiments, the proposed method is used for PMSM multiparameter identification modeling and temperature monitoring, which proves that it is effective in estimating the machine *dq*-axis inductances, stator winding resistance, and rotor flux linkage. The proposed estimator can also track the parameter variation that is verified by temperature variation experiments at last. Furthermore, online temperature monitoring for PMSM can be realized using the proposed method.

In our future work, we will apply the proposed method to practical industrial PMSM drive systems, such as hybrid electric vehicles, renew energy power generation, and high-speed rail.

References

[1] M. C. Chou *et al.*, "Robust current and torque controls for PMSM driven satellite reaction wheel," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 47, no. 1, pp. 58–74, Jan. 2011.

[2] R. Nalepa and T. Orlowska-Kowalska, "Optimum trajectory control of the current vector of a nonsalient-pole PMSM in the field-weakening region," *IEEE Trans. Ind. Electron.*, vol. 59, no. 7, pp. 2867–2876, Jul. 2012.

[3] G. Pellegrino, A. Vagati, P. Guglielmi, and B. Boazzo, "Performance comparison between surface-mounted and interior PM motor drives for electric vehicle application," *IEEE Trans. Ind. Electron.*, vol. 59, no. 2, pp. 803–811, Feb. 2012.

[4] P. Zheng, J. Zhao, R. Liu, C. D. Tong, and Q. Wu, "Magnetic characteristics investigation of an axial-axial flux compound-structure PMSM used for HEVs," *IEEE Trans. Mag.*, vol. 46, no. 6, pp. 2191–2194, Jun. 2010.

[5] W. Su, H. Eichi, W. Zeng, and M.-Y. Chow, "A survey on the electrification of transportation in a smart grid environment," *IEEE Trans. Ind. Informat.*, vol. 8, no. 1, pp. 1–10, Feb. 2012.

[6] P. Sergeant, F. De Belie, and J. Melkebeek, "Rotor geometry design of interior PMSMS with and without flux barriers for more accurate sensorless control," *IEEE Trans. Ind. Electron.*, vol. 59, no. 6, pp. 2457–2465, Jun. 2012.

[7] K. Liu *et al.*, "Online multiparameter estimation of nonsalient-pole PM synchronous machines with temperature variation tracking," *IEEE Trans. Ind. Electron.*, vol. 58, no. 5, pp. 1776–1788, May 2011.

[8] Y. C. Shi, K. Sun, L. P. Huang, and Y. Li, "Online identification of permanent magnet flux based on extended Kalman filter for IPMSM drive with position sensorless control". *IEEE Trans. Ind. Electron*, vol. 59, no. 11, pp. 4169–4178, Nov. 2012.

[9] E. Monmasson *et al.*, "FPGAS in industrial control applications," *IEEE Trans. Ind. Informat.*, vol. 7, no. 2, pp. 224–243, May 2011.

[10] S. Bolognani, R. Oboe, and M. Zigliotto, "Sensorless full-digital PMSM drive with EKF estimation of speed and rotor position," *IEEE Trans. Ind. Electron.*, vol. 46, no. 1, pp. 184–191, Feb. 1999.

[11] B. N. Mobarakeh, F. Meibody-Tabar, and F. M. Sargos, "Mechanical sensorless control of PMSM with online estimation of stator resistance," *IEEE Trans. Ind. Appl.*, vol. 40, no. 2, pp. 457–471, Mar./Apr. 2004.

[12] S. Moreau, R. Kahoul, and J.-P. Louis, "Parameters estimation of permanent magnet synchronous machine without adding extra-signal as input excitation," in *Proc. IEEE Int. Symp. Ind. Electron.*, 2004, vol. 1, pp. 371–376.

[13] M. Rashed, P. F. A. Macconnell, A. F. Stronach, and P. Acarnley, "Sensorless indirect-rotor-field-orientation speed control of a permanent-magnet synchronous motor with stator resistance estimation," *IEEE Trans. Ind. Electron.*, vol. 54, no. 3, pp. 1664–1675, Mar. 2007.

[14] R. Ramakrishnan, R. Islam, M. Islam, and T. Sebastian, "Real time estimation of parameters for controlling and monitoring permanent magnet synchronous motors," in *Proc. IEEE Int. Elect. Mach. Drives Conf.*, Miami, FL, USA: IEEE, 2009, pp. 1194–1199.

[15] S. Ichikawa, M. Tomita, S. Doki, and S. Okuma, "Sensorless control of permanent-magnet synchronous motors using online parameter identification based on system identification theory," *IEEE Trans. Ind. Electron.*, vol. 53, no. 2, pp. 363–372, Apr. 2006.

[16] S. J. Underwood and I. Husain, "Online parameter estimation and adaptive control of permanent-magnet synchronous machines," *IEEE Trans. Ind. Electron*, vol. 57, no. 7, pp. 2435–2443, Jul. 2010.

[17] G. Besançon, D. Leon, and O. Huerta, "On adaptive observers for state affine systems," *Int. J. Control*, vol. 79, pp. 581–591, 2006.

[18] M. Hamida, J. De Leon, A. Glumineau, and R. Boisliveau, "An adaptive interconnected observer for sensorles control of PM synchronous motors with online parameter identification," *IEEE Trans. Ind. Electron.*, vol. 60, no. 2, pp. 739–748, Feb. 2013.

[19] M. A. Jabbar, J. Dong, and Z. Lin, "Determination of machine parameters for internal permanent magnet synchronous motors," in *Proc. Power Electron. Mach. Drives*, 2004, pp. 805–810.

[20] F. F. M. El-Sousy, "Robust wavelet-neural-network sliding-mode control system for permanent magnet synchronous motor drive," *IET Elect. Power Appl.*, vol. 5, no. 1, pp. 113–132, Jan. 2011.

[21] W. Miroslaw, K. Zbigniew, and H. A. Toliyat, "Neural-network-based parameter estimations of induction motors," *IEEE Trans. Ind. Electron.*, vol. 55, no. 4, pp. 1783–1794, Apr. 2008.

[22] K. Liu, Z. Q. Zhu, and J. Zhang, "Multi-parameter estimation of nonsalient pole permanent magnet synchronous machines by using evolutionary algorithms," in *Proc. IEEE Int. Conf. BIC-TA*, Changsha, China, 2010, pp. 766–774.

[23] D. C. Aliprantis, S. D. Sudhoff, and B. T. Kuhn, "Genetic algorithm-based parameter identification of a hysteretic brushless exciter model," *IEEE Trans. Energy Convers.*, vol. 21, no. 1, pp. 148–154, Mar. 2006.

[24] W. X. Liu, L. Liu, and D. A. Cartes, "Efforts on real-time implementation of PSO based PMSM parameter identification," in *Proc. IEEE Power Energy Soc. Gen. Meeting Convers. Delivery Elect. Energy 21st Century*, 2008, pp. 1–7.

[25] L. Liu, W. X. Liu, and D. A. Cartes, "Permanent magnet synchronous motor parameter identification using particle swarm optimization," *Int. J. Comput. Intell. Res.*, vol. 4, no. 2, pp. 211–218, Feb. 2008.
[26] A. A. E. Ahmed, L. T. Germano, and Z. C. Antonio, "A hybrid particle swarm optimization applied to loss power minimization," *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 859–866, May 2005.
[27] C. F. Juang, "A hybrid of genetic algorithm and particle swarm optimization for recurrent network design," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 34, no. 2, pp. 997–1006, Apr. 2004.
[28] S. H. Ling *et al.*, "Hybrid particle swarm optimization with wavelet mutation and its industrial applications," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 38, no. 3, pp. 743–763, Jun. 2008.
[29] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, Jun. 2006.
[30] H. Wu *et al.*, "An improved comprehensive learning particle swarm optimization and its application to the semiautomatic design of antennas," *IEEE Trans. Antennas Propag.*, vol. 57, no. 10, pp. 3018–3028, Oct. 2009.
[31] Z. H. Zhan, J. Zhang, Y. Li, and H. S. H. Chung, "Adaptive particle swarm optimization," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 39, no. 6, pp. 1362–1381. Dec. 2009.
[32] F. Van den Bergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Trans. Evol. Comput.*, vol. 8, no. 3, pp. 225–239, Jun. 2004.
[33] R. A. Krohling and L. S. Coelho, "Coevolutionary particle swarm optimization using Gaussian distribution for solving constrained optimization problems," *IEEE Trans. Syst. Man Cybern. B, Cybern.*, vol. 36, no. 6, pp. 1407–1416, Dec. 2006.
[34] G. G. Yen and W. F. Leong, "Dynamic multiple swarms in multi objective particle swarm optimization," *IEEE Trans. Syst. Man Cybern. A, Syst. Hum.*, vol. 39, no. 4, pp. 890–911, Jul. 2009.
[35] Z. H. Zhan *et al.*, "Multiple populations for multiple objectives: A coevolutionary technique for solving multiobjective optimization problems," *IEEE Trans. Cybern.*, vol. 43, no. 2, pp. 445–463, Apr. 2013.
[36] M. A. Potter and K. A. De Jong, "Cooperative coevolution: An architecture for evolving coadapted subcomponents," *Evol. Comput.*, vol. 8, no. pp. 1–29, Jan. 2000.
[37] H. W. Ge *et al.*, "An effective PSO and AIS-based hybrid intelligent algorithm for job-shop scheduling," *IEEE Trans. Syst. Man Cybern. A, Syst. Hum.*, vol. 38, no. 2, pp. 358–368. Mar. 2008.
[38] Z. H. Liu, J. Zhang, S. W. Zhou, X. H. Li, and K. Liu, "Coevolutionary particle swarm optimization using AIS and its application in multiparameter estimation of PMSM," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1921–1935. Dec. 2013.
[39] V. L. Thé, N. Melab, and E. Talbi, "GPU Computing for parallel local search metaheuristic algorithms," *IEEE Trans. Comput.*, vol. 62, no. 1, pp. 173–185, Jan. 2013.
[40] X. Yao, Y. Liu, and G. M. Lin, "Evolutionary programming made faster," *IEEE Trans. Evol. Comput.*, vol. 3, no. 2, pp. 82–102, Jul. 1999.

Zhao-Hua Liu received the M.Sc. degree in computer science and technology, and the Ph.D. degree in electrical engineering from the Hunan University, Changsha, China, in 2010 and 2012, respectively.

He is currently a Lecturer with the School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan, China. His research interests include evolutionary algorithms and their related applications, parallel computing, big data processing, machine learning technology, and condition monitoring and fault diagnosis.

Dr. Liu has published a monograph in the field of *Biological immune system inspired hybrid intelligent algorithm and its applications*, and published more than 20 research papers in refereed journals and conferences, including some well-known international journals, such as IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS. He is a regular reviewer for several international journals and conferences. He is also a member of Association for Computing Machinery.

Xiao-Hua Li received the B.Eng. degree in computer science and technology from the Hunan University of Science and Engineering, Yongzhou, China, in 2007, and the M.Sc. degree in computer science from Hunan University, Changsha, China, in 2010.

She is currently a Lecturer with the School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan, China. Her research interests include evolutionary computation.

Liang-Hong Wu received the B.S. degree in industrial automation from the Hunan University of Science and Technology, Xiangtan, China, in 2001, and the M.S. and Ph.D. degrees in control science and engineering from Hunan University, Changsha, China, in 2007 and 2011, respectively.

He is currently an Associate Professor with the Hunan University of Science and Technology. His research interests include artificial intelligence, evolutionary computation, and applications.

Shao-Wu Zhou received the M.Sc. and Ph.D. degrees in electrical engineering from Central South University, Changsha, China, and Hunan University, Changsha, China, in 1990 and 2005, respectively.

He is currently a Professor with the School of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan, China. His research interests include intelligent control and robot control.

Kan Liu received the B.Eng. and Ph.D. degrees in automation from Hunan University, Changsha, China, in 2005 and 2011 respectively.

He is currently a Research Associate with the Department of Electronics and Electrical Engineering, University of Sheffield, Sheffield, U.K. His research interests include brushless ac motor parameters estimation by control theory and nonlinearity compensation for voltage source inverter (VSI) output voltage.