IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 20, 2022, accepted February 7, 2022, date of publication February 10, 2022, date of current version February 22, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3150777

Auto-Tuning Controller Using MLPSO With K-Means Clustering and Adaptive Learning Strategy for PMSM Drives

HOANG NGOC TRAN, TY TRUNG NGUYEN, HUNG QUANG CAO, TON HOANG NGUYEN, HA XUAN NGUYEN, AND JAE WOOK JEON, (Senior Member, IEEE)
Department of Electrical and Computer Engineering, Sungkyunkwan University, Suwon 16419, South Korea

Corresponding author: Jae Wook Jeon (jwjeon@skku.edu)

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government Ministry of Science and ICT (MSIT) (2021-0-00836, AI Auto-Tuning for Motor Drivers).

ABSTRACT This paper proposes a new online auto-tuning method to improve the accuracy and reduce the tuning time of permanent magnet synchronous motor (PMSM) drives. Under varying loads, the ability to tune the controllers of PMSM drives using optimal tuning time is crucial. However, direct tuning of controller parameters using estimated parameters or conventional particle swarm optimization (PSO) methods do not satisfy the performance criteria. To solve this problem, the new method combining mechanical parameter estimation (MPE) and multi-layer particle swarm optimization (MLPSO) with K-means clustering (KMC) and an adaptive learning strategy (ALS) is proposed. First, the combination of an MPE method with a lookup table (LUT) for initial parameter selection is introduced to reduce the iteration time. Then, the MLPSO-KMCALS method is proposed as an improvement over the conventional PSO method by increasing the number of layers, grouping the swarm into several subswarms, and using the ALS for each particle to increase the population diversity and optimize the controller parameters within the shortest possible amount of time. Finally, a disturbance load torque observer is applied to compensate for the effect of external disturbances after tuning. The effectiveness of the proposed method is validated through experiments conducted under practical conditions.

INDEX TERMS PMSM drives, auto-tuning, parameter estimation, particle swarm optimization (PSO), adaptive multi-layer search.

I. INTRODUCTION

PMSM drives are widely used in many applications due to the properties of lower maintenance, speed regulation performance, and high power density. In fact, the PMSM system is not easy to control because most processes have different nonlinear levels, variable parameters and a large amount of uncertainty in the mathematical models. Thus, high-performance PMSM motor drives require fast response, high accuracy, and stable adaptability against the motor parameter and load torque variation. In general, the typical electric drive controller consists of several nested control loops for the control of current, speed, and position. In each control loop, conventional proportional-integral-derivative (PID) controllers are widely applied due to their simplicity, stability, and easy adjustment [1], [2]. However, the PID parameters must be tuned to obtain the minimum operating error that satisfies the stability criteria of motor under different operating conditions.

There have been a lot of approaches to tune the parameters for the motor PID controller. For example, parameters were tuned by using an basis adaptive rule and a speed error under various conditions [3]. A parameter modifier was proposed for the PD position controller based on a position error [4]. Although these control techniques do not require knowledge of the mechanical and electrical parameters of the motor, they are often complicated and take a long execution time. Therefore, the tuning using estimators has been developed to simplify control law based on the identified parameters such as flux linkage, disturbance, viscous frictional, and motor inertia [5], [6]. In this case, the accuracy of the estimator directly affects the ability to optimize the tuning process. For low-cost

The associate editor coordinating the review of this manuscript and approving it for publication was Shihong Ding.

motors with low-resolution encoders ($\leq$10000 $pulses/rev$), the control parameters cannot be guaranteed to be optimal because of the large estimation error.

To overcome the above disadvantages, some intelligent auto-tuning approaches have been integrated into PID controllers [7]–[13]. In [7] and [8], a parameter tuning approaches was proposed by Ziegler and Nichols based on the values of the lag and unit reaction rate in order to improve the response and accuracy of the motor speed controller. Fuzzy logic control (FLC) [9], [10] has been employed to reduce the overshoot and settling time by tuning on-line parameters under the load torque variance. Moreover, with the development of artificial intelligence, many researchers have been used an artificial neural network (ANN) technique [11]–[13] to control the PMSM system for linear and non-linear factor compensation. However, the large number of rules and the high computational burden increase the complexity, which limits the implementation of this technique in practical applications.

Among the previously studied intelligence auto-tuning controls, particle swarm optimization (PSO) [14]–[16] has a strong ability to search for the best parameters for the controllers. In addition, it is simple to implement and has low computational cost, which are important characteristics considering industrial needs. To speed up the tuning time and avoid the searching of parameters not being optimal due to falling into local optima, the conventional PSO [17] has been improved by using a comprehensive learning strategy [18], [19] or a multi-player PSO technique (ML-PSO) [20]. The experimental results of these tuning techniques show that although it is possible to optimal the parameters, the tuning time still does not satisfy the requirements of the industry standard.

Motivated by the wish to find the best parameters of the PID speed controller in the shortest tuning time as well as to improve the operating accuracy of the PMSM system with low-revolution encoder under variable load, this paper presents a new approach, called Multi-Layer PSO with k-means clustering and adaptive learning strategy (MLPSO-KMCALS). The main contributions of this article are summarized as follows:

1) Pre-tuning processing is accomplished using a combination of a MPE method and lookup table (LUT) to reduce the tuning time. A disturbance load torque observer with gain tuning is designed to compensate the load torque changes.
2) The MLPSO-KMCALS algorithm is proposed. First, the multi-layer PSO is applied to increase the convergence speed. Second, the k-means clustering and adaptive learning strategies are integrated to increase the population diversity and avoid local optima.
3) The modified fitness function for the MLPSO-KMCALS algorithm is proposed to adapt to the performance criteria.
4) Experiments are performed and the method is compared with existing auto-tuning methods to verify the effectiveness of the proposed method and its practical applicability.

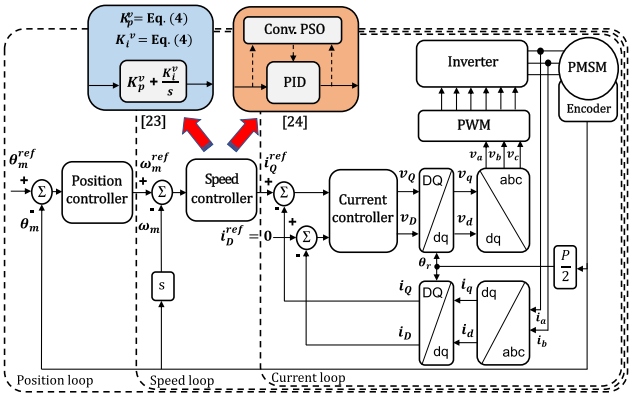

FIGURE 1. Motor control system.

The remainder of this paper is organized as follows. Section II provides the overview and reviews related works of auto-tuning for a PMSM driver. Section III describes the pre-tuning process and disturbance load torque observer. Section IV details the proposed online auto-tuning technique for the speed loop. Finally, Section V and VI present the experimental results and summarizes the conclusions of this paper, respectively.

II. OVERVIEW AND RELATED WORKS

A. MATHEMATICAL MODELING OF A PMSM

The system equations in a d-q model of a PMSM can be expressed as follows [21], [22]:

$$\begin{bmatrix} v_Q \\ v_D \end{bmatrix} = \begin{bmatrix} R_s + sL_{qs} & \omega_r L_{ds} \\ -\omega_r L_{qs} & R_s + sL_{ds} \end{bmatrix} \begin{bmatrix} i_Q \\ i_D \end{bmatrix} + \begin{bmatrix} \omega_r \lambda_m \\ 0 \end{bmatrix} \quad (1)$$

$$T_e = \frac{3}{2}\frac{P}{2}\left[\lambda_m i_Q + \left(L_{ds} - L_{qs}\right) i_Q i_D\right] \quad (2)$$

where v_Q, v_D, i_Q, i_D, L_{qs} and L_{ds} are the $q-$ and $d-axis$ voltages, current, and inductances, respectively; R_s is the resistance; λ_m is the flux linkage of permanent magnet; ω_r is the electrical angular speed of the rotor; "s" represents the Laplace operator; T_e is the electromagnetic torque; and P is the number of rotor poles.

In the linearized model, i_D can be made zero by controlling the field direction. Therefore, the electromagnetic torque can be described as:

$$T_e = K_t i_Q = J\frac{d\omega_m}{dt} + B\omega_m + T_L \quad (3)$$

where K_t is the motor torque constant; J is the inertia of the rotor (J_r) and load (J_l); B is the viscous frictional coefficient; T_L is the load torque; and ω_m is the mechanical speed of the rotor shaft.

B. RELATED WORKS

Given the effect of varying high loads, many techniques have been proposed for tuning the parameters of three controllers,

as shown in Fig. 1. Among them, two methods are most often applied, the direct auto-tuning method based on the estimated parameters [23] and the optimal auto-tuning method based on the traditional PSO [24].

In [23], Yang and Lin first estimated the resistance and inductance for current control loop tuning. The torque constant, inertia and viscous frictional coefficients for speed and position loop tuning were also estimated. When there is a significant change in the load, the inertia J changes and the parameters controllers are adjusted accordingly. The PI parameters of the current controller and speed controller were suggested by the authors as follows:

$$Current: \begin{cases} K_p^i = \omega^i L_{qs} \\ K_i^i = \omega^i R_s \end{cases} \quad Speed: \begin{cases} K_p^v = \omega^v J \\ K_i^v = \omega^v B \end{cases} \tag{4}$$

where $K_p^{i,v}$ and $K_i^{i,v}$ are the proportional and integral parameters of the current and speed controller, respectively; and ω^i,and ω^v are the cutoff frequencies of the current loop,and speed loop, respectively, which are identified based on experience.

Using the zero-cancellation method, the parameters controllers are quickly adapted, however, factors such as overshoot, settling time and especially vibration error are not considered optimally. In addition, the J and B coefficients require high accuracy under the impact of noise and high load.

Qi and Zhang used the PSO algorithm for the PID parameter tuning of the speed controller [24]. PSO is a population based search algorithm that imitates the behaviors of birds. In this algorithm, an initial population of randomly generated solutions is called a particle. Each particle has a velocity and a position vector, which are updated to find the best solution during each iteration in D-dimemsional space. The velocity v_i and position z_i vectors are updated according to (5) and (6), as follows:

$$v_i^{t+1} = Wv_i^t + c_1 r_1 \left(pbest_i^t - z_i^t\right) + c_2 r_2 \left(gbest_i^t - z_i^t\right) \tag{5}$$

$$z_i^{t+1} = z_i^t + v_i^{t+1} \tag{6}$$

where W is the inertia factor; t indicates the iterations; $i = 1, 2, \ldots, N_{size}$ is the particles index; c_1 and c_2 are two learning factors; r_1 and r_2 are two random numbers in the range $[0, 1]$; $pbest_i = (pbest_{i1}, pbest_{i2}, \ldots, pbest_{iD})$ is the local best position recorded of the ith particle; and $gbest_i = (gbest_{i1}, gbest_{i2}, \ldots, gbest_{iD})$ is the global best position of all particles. The quality of each particle is evaluated using the fitness function. In [24], each particle is endowed with a local update of the vector $\left[K_p^v, K_i^v, K_d^v\right]$. The fitness function is modified to deal with the stochastic characteristics of the motor as (7).

$$F = \left(K_p^v, K_i^v, K_d^v\right) = \frac{1}{m}\sum_{l=1}^{m}\left(\sum_{k=1}^{N} Q_1 \left|\omega_{m,k}^{ref} - \omega_{m,k}\right| + \sum_{k=1}^{N} Q_2 \left|i_{Q,k}^{ref} - i_{Q,k-1}^{ref}\right|\right) \tag{7}$$

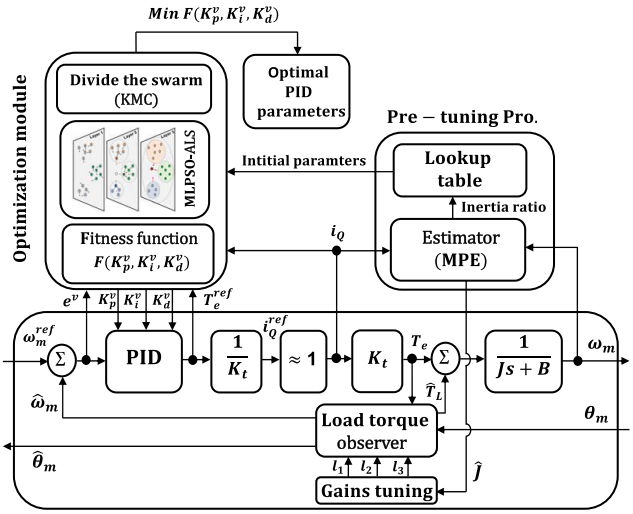

FIGURE 2. Auto-tuning procedure of the proposed method.

where m is the maximum simulation executed; l is the simulation execution index; Q_1 and Q_2 are positive weighting factors; ω_m^{ref} and i_Q^{ref} are the reference speed and current, respectively; and N is the number of samples.

The PSO algorithm is designed to optimize the set of parameters $\left[K_p^v, K_i^v, K_d^v\right]$ that minimizes $F\left(K_p^v, K_i^v, K_d^v\right)$. This method is based on the integral of the speed error and the deviation between the outputs. However when optimizing the parameters, the authors do not consider the convergence time or the iterations of the motor. In addition, the local optima problem has not been solved, leading to uncertainty in the parameters.

In this paper, a new online auto-tuning method based on MLPSO-KMCALS is proposed to overcome all the disadvantages of the previous methods. Under the impact of variable load, the speed control loop is mainly affected by the corresponding change in load inertia, so the paper focuses on the optimal speed control loop parameters. The current loop and position loop parameters were determined based on a previous paper [23]. Fig. 2 shows the auto-tuning procedure used in the proposed method. First, the MPE method is used to estimate the inertia ratio as input to the lookup table. The outputs are the initial parameters of the particles. The particles are then grouped into subwarms by the KMC method before being valued by the modified fitness function and updated by the MLPSO-ALS algorithm. In the proposed method, the speed parameters are tuned based on load change, vibration, and accuracy in the optimal tuning time.

III. PRE-TUNING PROCESS AND DISTURBANCE LOAD TORQUE OBSERVER

A. MECHANICAL PARAMETER ESTIMATION

As in the traditional PSO method, random selection of the initial positions will increase the convergence time as well as the number of iterations of the motor necessary to find the

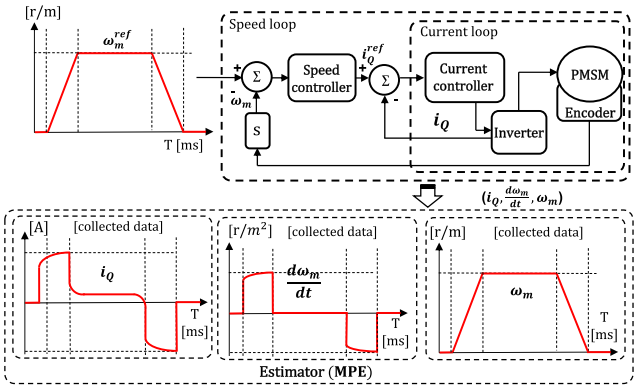

FIGURE 3. Current, acceleration and, speed waveforms used to estimate the inertia ratio.

optimal parameters. Therefore, we proposed the MPE to estimate the inertia ratio value, which is used as a reference value to narrow the initial search area. The estimation accuracy does not need to be very high to directly tune the parameters as in [23]. We have applied a simple least square method base on the torque characteristic. According to the electromagnetic torque (3), this can be expressed as follows:

$$K_t i_Q = \begin{bmatrix} \frac{d\omega_m}{dt} & \omega_m & 1 \end{bmatrix} \begin{bmatrix} B \\ J \\ T_L \end{bmatrix} \tag{8}$$

which can also be expressed by applying Euler's rule and the LSM identification [25], [26] as follows:

$$\begin{bmatrix} K_t i_{Q(k)} \\ K_t i_{Q(k+1)} \\ \cdots \\ K_t i_{Q(k+N)} \end{bmatrix} = \begin{bmatrix} \frac{\omega_{m(k)} - \omega_{m(k-1)}}{t_s} & \omega_{m(k)} & 1 \\ \frac{\omega_{m(k+1)} - \omega_{m(k)}}{t_s} & \omega_{m(k+1)} & 1 \\ \cdots & \cdots & \cdots \\ \frac{\omega_{m(k+N)} - \omega_{m(k+N-1)}}{t_s} & \omega_{m(k+N)} & 1 \end{bmatrix} \begin{bmatrix} B \\ J \\ T_L \end{bmatrix} \tag{9}$$

or

$$Y = X\theta \tag{10}$$

where N is a set volume of estimated data, t_s is a sampling time interval, $Y \in R^N$ is a vector of the current feedback of the motor, $X \in R^{3xN}$ is a vector of the angular acceleration and the motor speed feedback, and $\theta \in R^3$ is the parameters vector. By using ordinary least squares regression method (11), the estimation parameters $\hat{\theta} = \begin{bmatrix} \hat{B} & \hat{J} & \hat{T_L} \end{bmatrix}^T$ can be solved.

$$\hat{\theta} = arg\min_{\theta} \|X\theta - Y\|^2 = (X^T X)^{-1} X^T Y \tag{11}$$

Fig.3 shows the speed input waveform and current, speed and acceleration output waveform for these estimated inertia. From (11), the inertia ($\hat{J}$) is estimated to determine the inertia ratio ($\hat{IR}$) with known inertia rotor (J_r), as follows:

$$\hat{IR} \approx \frac{\hat{J}}{J_r} \tag{12}$$

TABLE 1. Random search range with LUT.

$IR = \frac{J_r + J_L}{J_r}$	Random search range K_p^v	Random search range K_i^v	Random search range K_d^v
$1 \leq IR \leq 2$	$[0, 30]$	$[0, 0.05]$	$[0, 0.05]$
$2 < IR \leq 4$	$[15, 40]$	$[0, 0.05]$	$[0, 0.05]$
$4 < IR \leq 6$	$[20, 50]$	$[0, 0.05]$	$[0, 0.05]$
$6 < IR \leq 8$	$[35, 85]$	$[0, 0.05]$	$[0, 0.05]$
$8 < IR \leq 10$	$[45, 100]$	$[0, 0.05]$	$[0, 0.05]$
$10 < IR \leq 12$	$[60, 130]$	$[0, 0.05]$	$[0, 0.05]$
$12 < IR \leq 14$	$[70, 140]$	$[0, 0.05]$	$[0, 0.05]$
$14 < IR \leq 16$	$[75, 145]$	$[0, 0.05]$	$[0, 0.05]$
$16 < IR \leq 18$	$[80, 150]$	$[0, 0.05]$	$[0, 0.05]$
$18 < IR \leq 20$	$[85, 155]$	$[0, 0.05]$	$[0, 0.05]$
$20 < IR$	$[90, 160]$	$[0, 0.05]$	$[0, 0.05]$

After estimation, the inertia ratio is used as reference input of the LUT to select the initial parameters for the MLPSO-KMCALS tuning algorithm.

B. LUT OPTIMIZATION FOR THE INITIAL PARTICLE RANGE

The LUT is constructed with the estimated inertial ratio input; the output is the initial N_{size} particles for the tuning algorithm. The main aim of this paper is to optimize the PID speed controller parameters in the shortest amount of time. The PID controller is considered in discrete time, as follows:

$$T_e^{ref} = K_p^v e_k^v + K_i^v \sum_{n=0}^{m-1} e_n^v + K_d^v \left(e_k^v - e_{k-1}^v\right) \tag{13}$$

where $k = 1, 2, 3, \ldots, m$; m is the maximum sampling number; T_e^{ref} is the output of the PID speed controller and e^v is the error between the reference speed (ω_m^{ref}) and feedback speed (ω_m).

As shown in (13), each particle consists of a spatial dimension $D = 3$ corresponding to the value of K_p^v, K_i^v and K_d^v. The LUT is built based on the experimental results. The MLPSO-KMCALS tuning algorithm was applied without the LUT to identify the optimal value of each parameter at different loads ($IR = 2.u \leq \frac{J_{limit}}{J_r}, u = 1 : 10$). Therefore, the initial value of each particle is limited to the random search range within the LUT, as shown in Table 1. The effect of the pre-tuning process on the convergence time of the MLPSO-KMCALS algorithm is evaluated in the experimental section.

C. DISTURBANCE LOAD TORQUE OBSERVER

During the pre-tuning process, for the estimator to work correctly, external influences or sudden changes in load torque are limited. However, the disturbance load torque observer is designed to estimate the $\hat{T_L}$ for compensation of load torque changes. According to the state-space model of the PMSM motion equation [27], it can be expressed as follows:

$$\begin{cases} \omega_m' = -\frac{B}{J}\omega_m - \frac{T_L}{J} + \frac{K_t}{J} i_Q \\ \theta_m' = \omega_m \\ T_L' = 0 \end{cases} \tag{14}$$

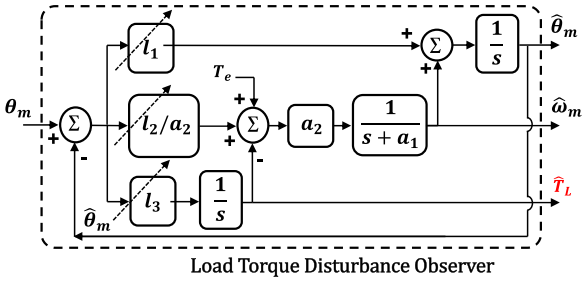

FIGURE 4. The principle block diagram of the disturbance load torque observer.

which can also be considered in linear time-invariant systems as follows:

$$\begin{cases} x' = Ax + Bu \\ y = Cx \end{cases} \tag{15}$$

where $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -a_1 & -a_2 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 & a_2 & 0 \end{bmatrix}^T, C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, a_1 = B/J, a_2 = 1/J, x = \begin{bmatrix} \theta_m & \omega_m & T_L \end{bmatrix}^T, u = T_e, y = \theta_m$; θ_m is the mechanical position.

By using the full-dimensional state observer idea, the structural formula of the disturbance load torque observer is obtained as follows:

$$\begin{cases} \widehat{x}' = A\widehat{x} + Bu + L\widetilde{y} \\ \widehat{y} = C\widehat{x} \end{cases} \tag{16}$$

$$\begin{cases} \widehat{\theta}_m' = l_1(\theta_m - \widehat{\theta}_m) + \widehat{\omega}_m \\ \widehat{\omega}_m' = l_2(\theta_m - \widehat{\theta}_m) - a_1\widehat{\omega}_m + a_2(T_e - \widehat{T}_L) \\ \widehat{T}_L' = l_3(\theta_m - \widehat{\theta}_m) \end{cases} \tag{17}$$

where $L = \begin{bmatrix} l_1 & l_2 & l_3 \end{bmatrix}^T, \widehat{x} = \begin{bmatrix} \widehat{\theta}_m & \widehat{\omega}_m & \widehat{T}_L \end{bmatrix}^T, \widetilde{y} = y - \widehat{y}$, $y = \theta_m, \widehat{y} = \widehat{\theta}_m$; $\widehat{\theta}_m, \widehat{\omega}_m$ and $\widehat{T}_L$ are the estimated position, speed of rotor and the estimated disturbance load torque, respectively; and l_1, l_2, and l_3 are the state feedback gains.

Fig. 4 shows the block diagram of the disturbance load torque observer. To satisfy the stability requirement, the state feedback gains are considered on the condition of the characteristic equation [28], [29] as follows:

$$det[sI - (A - LC)] = s^3 + (a_1 + l_1)s^2 + (a_1 l_1 + l_2)s - a_2 l_3 = 0 \tag{18}$$

For the third-order system, the stable characteristic equation with a pair of dominant complex conjugate poles $p_{1,2} = -\xi\omega_n \pm j\omega_n\sqrt{1-\xi^2}$ is expressed as:

$$s^3 + (2\xi\omega_n - b)s^2 + ({\omega_n}^2 - 2\xi\omega_n b)s - {\omega_n}^2 b = 0 \tag{19}$$

where the natural frequency $w_n \approx 1000$ and the damping factor $\xi \approx 0.707$ are assumed. The ω_n should be large to have a fast converging time, while ξ is selected to obtain a small error. Balancing the coefficients of (18) and (19) gives:

$$\begin{cases} l_1 = 2\xi\omega_n - b - a_1 \\ l_2 = {\omega_n}^2 - 2\xi\omega_n b - a_1 l_1 \\ l_3 = {\omega_n}^2 b/a_2 \end{cases} \tag{20}$$

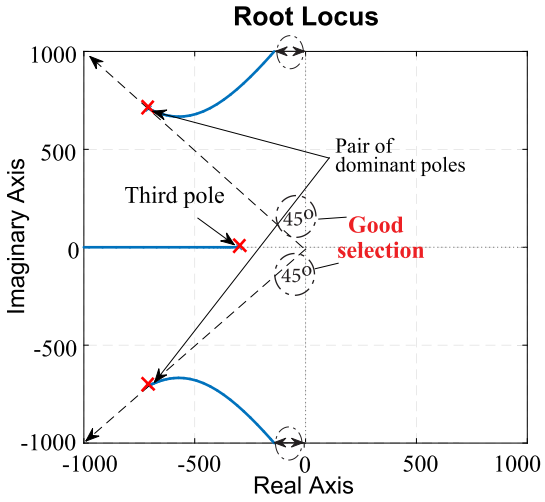

FIGURE 5. The root of the characteristic equation.

After estimating the value J, the state feedback gains $l_{1,2,3}$ are tuned to adapt to load changes following (20). To consider the stability of the response feedback gains, the roof of the characteristic equation (18) is observed in Fig. 5. The value of the third pole (b) is selected (≈ -300) away from the imaginary axis to improve the response rapidity. To have a good balance between rapidity and stationary, the pair of dominant complex conjugate poles should be located around the line of 45^o from the negative real axis.

IV. ONLINE AUTO-TUNING BASED ON A NEW MLPSO ALGORITHM

Given the effect of varying loads, solving the PID parameters tuning problem is necessary to increase the accuracy and stability of the motor. In this section, an online auto-tuning method for the PID parameters is proposed based on a new MLPSO algorithm. The multi-layer PSO is proposed in [20] to increase the convergence time by extending from the two-layers of the conventional PSO to multiple-layers and dividing the single swarm of a conventional PSO into several subswarms.

A. SUBSWARM DIVISION WITH K-MEANS CLUSTERING ALGORITHM

The K-means clustering algorithm [30], [31] is a popular data clustering method. In this paper, this method is used to divide the initial particles into subswarms. First, we select S particles as initial subswarm centers, then compute the Euclidian distance between each particle and each subswarm center and assign it to the nearest subswarm. The average of all subswarms is updated and the process is repeated until the criterion function converges. The square error criterion function is calculated as follows:

$$SE = \sum_{a=1}^{S}\sum_{b=1}^{n_a} \|x_{ab} - m_a\|^2 \tag{21}$$

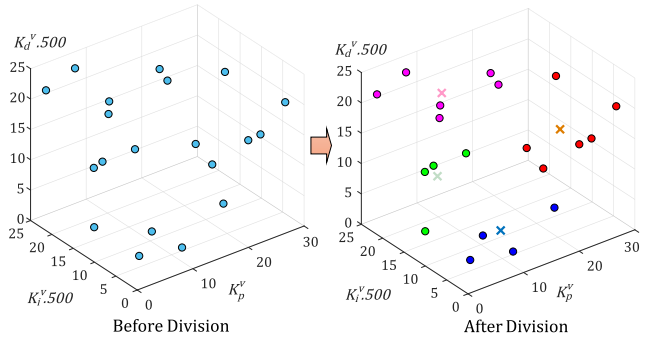

FIGURE 6. Subswarm division with K-Means clustering. x represents the subswarm center.

where x_{ab} is the particle b of a-subswarm, m_a is the center of a-subswarm, and n_a is the number of particles in a-subswarm. Fig. 6 shows an example of the subswarm division results with $S = 4$, $N_{size} = 20$, $IR = 1$, $D = 3$, $K_p^v = rand\,[0, 30]$, $K_i^v = rand\,[0, 0.05]$, and $K_d^v = rand\,[0, 0.05]$.

B. AUTO-TUNING METHOD WITH THE MLPSO-ALS ALGORITHM

As in [20], MLPSO is used to divide the swarm into several layers and the particles in each layer into several subswarms. A subswarm in the lower layer is called a swarmparticle and is controlled by a swarmparticle in an upper layer. In the conventional global MLPSO, each particle is impacted by the particle in the best position in the same swarmparticle in each layer. The velocity of a particle is updated based on the accumulated information from all layers, as follows:

$$\begin{cases} v_i^{t+1} = Wv_i^t + \sum_{j=1}^{M} c_j r\left(pbest_{ij}^t - z_i^t\right) \\ z_i^{t+1} = z_i^t + v_i^{t+1} \end{cases} \tag{22}$$

where M is the number of layers, $c_j = \frac{c}{M}$ and c is a summary of the acceleration constant, r is a random number in the range $[0, 1]$, and $pbest_j$ is the best position found in the swarmparticle containing the current particle in layer j.

In this paper, the global MLPSO is applied and improved to tune the PID parameters of the speed controller with the following settings: M = 3-players, N_{size} = 10-particles and D = 3-dimemsions of a particle corresponding to the vector $[K_p^v, K_i^v, K_d^v]$. Equation (22) is rewritten as follows:

$$\begin{cases} v_i^{t+1} = Wv_i^t + c_1 r(pbest_{i1}^t - z_i^t) \\ \qquad +c_2 r(pbest_{i2}^t - z_i^t) + c_3 r(pbest_{i3}^t - z_i^t) \\ z_i^{t+1} = z_i^t + v_i^{t+1} \end{cases} \tag{23}$$

However, in each subswarm in layer two, there are two types of particles: one is the best particle in the subswarm (best-SUB particle), while the others are the remaining particles (normal SUB particles). According to the learning strategy of the conventional global MLPSO, if the current particle is a best-SUB particle, $(pbest_{i2}^t - z_i^t)$ would be zero. This causes a loss of population diversity and tends to make the algorithm fall into local optima. Therefore, the adaptive

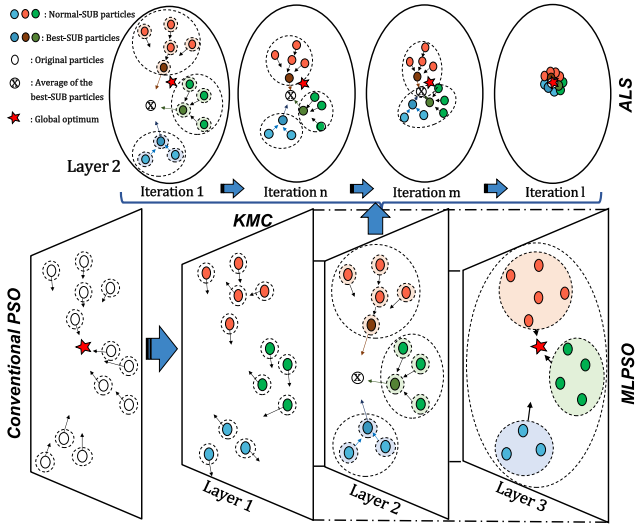

FIGURE 7. Comparison between the conventional PSO and the proposed MLPSO-KMCALS algorithm framework.

learning strategy (ALS) algorithm is proposed, in which all particles in a subswarm are identified as best-SUB particles or normal-SUB particles and the velocity is updated using two different learning strategies. Equation (23) is used to update the normal-SUB particles, while (24) used to update the best-SUB particles:

$$\begin{cases} v_i^{t+1} = Wv_i^t + c_1 r(pbest_{i1}^t - z_i^t) \\ \qquad +c_2 r\left(\frac{1}{S}\sum_{a=1}^{S} pbest_{i2a}^t - z_i^t\right) + c_3 r\left(pbest_{i3}^t - z_i^t\right) \\ z_i^{t+1} = z_i^t + v_i^{t+1} \end{cases} \tag{24}$$

where S is the number of subswarms. The social learning exemplar $pbest_{i2}^t$ in (23) is replaced by the average of the best-SUB particles in the second layer. This method not only improves population diversity but also accelerates the convergence time. Fig. 7 compares the conventional PSO and the proposed MLPSO-KMCALS algorithm framework.

Moreover, a modified fitness function is proposed to consider all tracking performance criteria in the PID speed controller, including the overshoot (O_{vs}),rise time (t_r), settling time (t_s), and steady state error (E_{ss}) of the speed controller [32]. However, after tuning the controller parameters, the steady-state error would be zero, so the mean squared error (MSE) of speed is considered. The modified fitness function is defined as follows:

$$F\left(K^v, q\right) = \frac{1}{N}\sum_{k=1}^{N}\left|\omega_{m,k}^{ref} - \omega_{m,k}\right|^2 + q_1 O_{vs} + q_2 t_s + q_3 t_r \tag{25}$$

where $K^v = [K_p^v, K_i^v, K_d^v]$ is the parameter vector of the speed controller, and $q = [q_1, q_2, q_3]$ is the weight parameter vector. In this fitness function, the MSE is the most important. To have a minimum MSE the other objects (O_{vs}, t_s, t_r) have to be at their minimum. Therefore, to minimize the fitness

function $F(K^v, q)$, we can only evaluate the *MSE*. However, in some cases, the motor is required to run with different priorities, so the four functions of the $F(K^v, q)$ are generated as follow:

$$\begin{cases} [q_1, q_2, q_3] = [0, 0, 0], & Balance \quad F \\ [q_1, q_2, q_3] = [1, 0, 0], & Maximize \; O_{vs} \\ [q_1, q_2, q_3] = [0, 1, 0], & Maximize \quad t_s \\ [q_1, q_2, q_3] = [0, 0, 1], & Maximize \quad t_r \end{cases} \tag{26}$$

To qualify for the parameters of the proposed MLPSO-KMCALS algorithm, the convergence property of particle cognition is analyzed in the next subsection.

C. CONVERGENCE ANALYSIS AND IMPLEMENTATION OF THE MLPSO-KMCALS ALGORITHM

The convergence property of the iterative process can be analyzed by considering the one-particle one-dimensional proposed algorithm with fixed $p_1 = pbest_1, p_2 = pbest_2$ or $\frac{1}{S}\sum_{a=1}^{S} pbest_{2a}$, $p_3 = pbest_3$, $\varphi_1 = c_1 r_1$, $\varphi_2 = c_2 r_2$ and $\varphi_3 = c_3 r_3$. The position and velocity updating (23) and (24) of the particle can be reduced for the analysis, as follows:

$$\begin{aligned} v^{t+1} &= Wv^t + \varphi_1(p_1 - z^t) + \varphi_2(p_2 - z^t) + \varphi_3(p_3 - z^t) \\ z^{t+1} &= z^t + v^{t+1} \end{aligned} \tag{27}$$

The basic simplified dynamic system can be expressed as

$$v^{t+1} = Wv^t - \varphi y^t; \quad y^{t+1} = Wz^t + (1 - \varphi)y^t \tag{28}$$

where $\varphi = \varphi_1 + \varphi_2 + \varphi_3$, $y^t = z^t - p$ and $p = \frac{\varphi_1 p_1 + \varphi_2 p_2 + \varphi_3 p_3}{\varphi_1 + \varphi_2 + \varphi_3}$

Let $P^t = \begin{bmatrix} v^t \\ y^t \end{bmatrix}$, and $M = \begin{bmatrix} W & -\varphi \\ W & 1 - \varphi \end{bmatrix}$ then (28) can be written in matrix form of the system as:

$$P^{t+1} = MP^t \tag{29}$$

The characteristic equation of the matrix M is defined by:

$$\lambda^2 - (W - 1 - \varphi)\lambda + W = 0 \tag{30}$$

In the convergent condition, the absolute values of both eigenvalues λ_1 and λ_2 are less than 1,as follows:

$$\lambda_{1,2} = \left| \frac{W + 1 - \varphi \pm \sqrt{\Delta}}{2} \right| < 1 \tag{31}$$

where $\Delta = (W + 1 - \varphi)^2 - 4W$. We consider two cases:

$$Case\,1 \; : \; \Delta < 0 \Leftrightarrow (W + 1 - \varphi)^2 < 4W \tag{32}$$

$$Case\,2 \; : \; \Delta \geq 0 \Leftrightarrow (W + 1 - \varphi)^2 \geq 4W \tag{33}$$

The condition $max(|\lambda_1|, |\lambda_2| < 1)$, case 1 itself and case 2 itself require:

$$Case\,1 \; : \begin{cases} 0 < W < 1 \\ 1 + W - 2\sqrt{W} < \varphi < 1 + W + 2\sqrt{W} \end{cases} \tag{34}$$

$$Case\,2 \; : \begin{cases} W \geq 0 \\ \varphi \leq 1 + W - 2\sqrt{W} \; or \; \varphi \geq 1 + W + 2\sqrt{W} \end{cases} \tag{35}$$

$$\begin{cases} \begin{cases} W < 1 \\ 0 < \varphi \leq 1 + W - 2\sqrt{W} \end{cases} if \quad \begin{cases} \varphi \leq 1 + W - 2\sqrt{W} \\ max(|\lambda_1|, |\lambda_2| < 1) \end{cases} \\ \begin{cases} W < 1 \\ 1 + W + 2\sqrt{W} \leq \varphi \leq 2W + 2 \end{cases} if \quad \begin{cases} \varphi \geq 1 + W + 2\sqrt{W} \\ max(|\lambda_1|, |\lambda_2| < 1) \end{cases} \end{cases} \tag{36}$$

The guaranteed convergent condition of the parameters are synthesized from cases 1 and 2, as follow (37). The results of the convergence coefficient analysis match the results of previous studies on convergence using the standards PSO method [33].

$$0 \leq W < 1 \; and \; 0 < \varphi < 2W + 2 \tag{37}$$

The implementation of the auto-tuning method with the proposed algorithm for the optimal parameters under the effect of varying high loads can be summarized in the following steps. S1:

1) Run the first rotation motor for inertia ratio estimation (*IR*) and input the results into the LUT to determine the range of the parameters (K_p^v, K_i^v, K_d^v).
2) Specify the number of particles N_{size}; the number of subswarms (S); the random particles vector data set (x_i) with the range following S1; and the threshold $min(SE)$. Then, apply the KMC algorithm to divide the swarm into subswarms.
3) Define the initial position $(z_i = x_{ab})$; randomly initialize the velocities of the particles; specify the parameters of the update equation satisfying the convergence criterion as (37); and specify the threshold value of the fitness function F.
4) Run each rotation of the motor with the parameters corresponding to the position of each particle and evaluate using the fitness function (F) as (25).
5) To compare the obtained value of F, the best position of a particle in layer 1 is saved as $pbest_1$, the best position of a subswarm in layer 2 is saved as $pbest_2$, and the global best position of the swarm in layer 3 is saved as $pbest_3$.
6) Update the velocity and position of the two types of particles according to (23) and (24).
7) If the value of the fitness function reaches the threshold value, go to S8. Otherwise, return to S4.
8) Obtain the optimal parameter set, $pbest_3$.

Algorithm 1 gives a more detailed explanation of the MLPSO-ALS algorithm.

V. EXPERIMENTAL

A. SYSTEM CONFIGURATION AND REAL TIME EXPERIMENT OF TUNING PROCESS

An experimental system was constructed as shown in Fig. 8. We used a PMSM model attached to an optical encoder (Fastech, Co., motor model K6LS30N2). An MCU STM32F446VCT6 was used to apply the proposed algorithm. A powder clutch (Mitsubishi, Co., model ZKG20AN) was mounted coaxially to the motor shaft to generate the external

Algorithm 1 MLPSO-ALS Algorithm

Input: N_{size}, M, D, initial (z_i, v_i), W, $c_{1,2,3,4}$, r, $q_{1,2,3}$, threshold F;
Output: $pbest_3 = [(K^v_{p\,best}, K^v_{i\,best}, K^v_{d\,best})]$

```
for h = 1 : 1 : N_size do
    z(h) = x_ab(h)
    F(h) ← F(z(h))
    pbest_1(h) ← z(h); F_pbest1(h) ← F(h)
end for
for a = 1 : 1 : S do
    j ← index for min(F) in size of a subswarm (n_a)
    pbest_2a ← z(j); F_pbest2a(h) ← F(z(j))
end for
    j ← index for min(F) of swarm
    pbest_3 ← z(j); F_pbest3(h) ← F(z(j))
while F > threshold
    for h = 1 : 1 : N_size do
        if (normal-SUB particle) then
        Update z(h), v(h) with (23)
        end if
        if (best-SUB particle) then
        Update z(h), v(h) with (24)
        end if
        F(h) ← F(z(h))
        Compare F(h) with F_pbest1; F_pbest2a, F_pbest3
        and update the best position at every layer
    end for
end while
return
```

load torque T_L for the performance evaluation of against load torque disturbance rejection. The sampling frequency for the current control loop is $20kHz$, and it is $2kHz$ for the speed and position controller. The q-axis inductance is $L_{qs} = 1.12mH$ and the resistance is $R_s = 1.4\Omega$. The cut-off frequencies for determination of the current, speed and position controller parameters are $\omega^i = 600Hz$, $\omega^v = 100Hz$, and $\omega^p = 25Hz$, respectively. The inertia rotor is $J_r = 8.6 \times 10^{-6} kgm^2$ and the viscous friction is $B = 2.66 \times 10^{-3} Nm/rad/s$. The parameters of the MLPSO-KMCALS algorithm are selected according to the convergence condition as $W = 0.7298$, $r = 1/2$, $c_1 = 0.1$, and $c_{2,3} = 0.072$.

Fig. 9 shows the speed waveform for the auto-tuning process ($\omega_m^{ref} = 1200r/min$, $d\omega_m^{ref}/dt = 200ms$ and, $IR = 12.5$). At the first rotation motor (Fig. 9 (a)), the data of speed feedback, current feedback and acceleration feedback are collected to estimate the inertia ratio ($\hat{IR}$). After the data is collected, the estimated inertia ratio is determined as ($\hat{IR}$) ≈ 11.75, which makes the estimated error is 6% in the Fig. 9 (b). In addition, based on the estimated ($\hat{IR}$), the state feedback gains of observer $l_{1,2,3}$ are adjusted according to (20) and the range of the parameters (K_p^v, K_i^v, K_d^v) are also determined based on the LUT. Then, the KMC algorithm is applied to divide the swarm into subswarms before starting

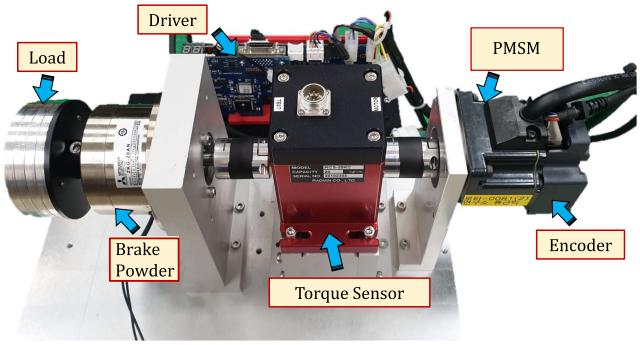

FIGURE 8. Experimental system.

the second rotation. Fig. 9 (c) shows the speed performance and speed error of the particle with the best parameters (K_p^v, K_i^v, K_d^v) in the first 10 rotations as well as in the 1^{st} iteration. By using the proposed auto-tuning technique, the particle with optimal parameter is found after 31 rotations and takes approximately 60s from the beginning to the end of tuning. Fig. 9 (d) shows the speed performance and speed error of the optimal controller parameters. The speed error has been reduced more than 2 times and the overshoot is slower than before tuning.

B. EXPERIMENTAL RESULTS COMPARED WITH DIRECT AUTO-TUNING RESULTS USING ZERO-CANCELLATION FOR THE PI METHOD

Fig. 10 and 11 compare the speed tracking control performance of the Zero Cancellation and the proposed methods. The speed reference value is $1200r/min$ in the low-load ($IR = 5$), and high-load ($IR = 19$). Under the influence of load, the speed controller with the PID parameters optimized using the Zero Cancellation method show increasing overshoot with increasing load. The cause of this error is due to the large estimation error for low-cost motors with low-resolution encoders. In addition, the speed error as well as the overshoot, rise time and settling time standards is not considered in the tuning process. Meanwhile, the controller with the parameters after tuning using the proposed method tracks the speed reference with an overshoot smaller than more than 2 times. The Zero Cancellation method has a maximum speed error ($\approx 175r/min$) at $IR = 5$ and ($\approx 225r/min$) at $IR = 19$. The proposed method achieves the best performance, with a maximum speed error ($\approx 50r/min$) at $IR = 5$ and ($\approx 55r/min$) at $IR = 19$. We also can compare the value of the current feedback at the red arrows to clearly see the difference in the impact of the both methods in Fig. 10 (e),(f) and Fig. 11 (e),(f).

Moreover, the speed and current controller's performance under multiple speed points with high-load ($IR = 19$) in a cycle are analyzed in Fig. 12. The multiple speed profile is generated that includes (600, 1200 and, 1800)r/min. As can be seen, the set parameters with the proposed method can reduce the overshoot problem and adapt better as the

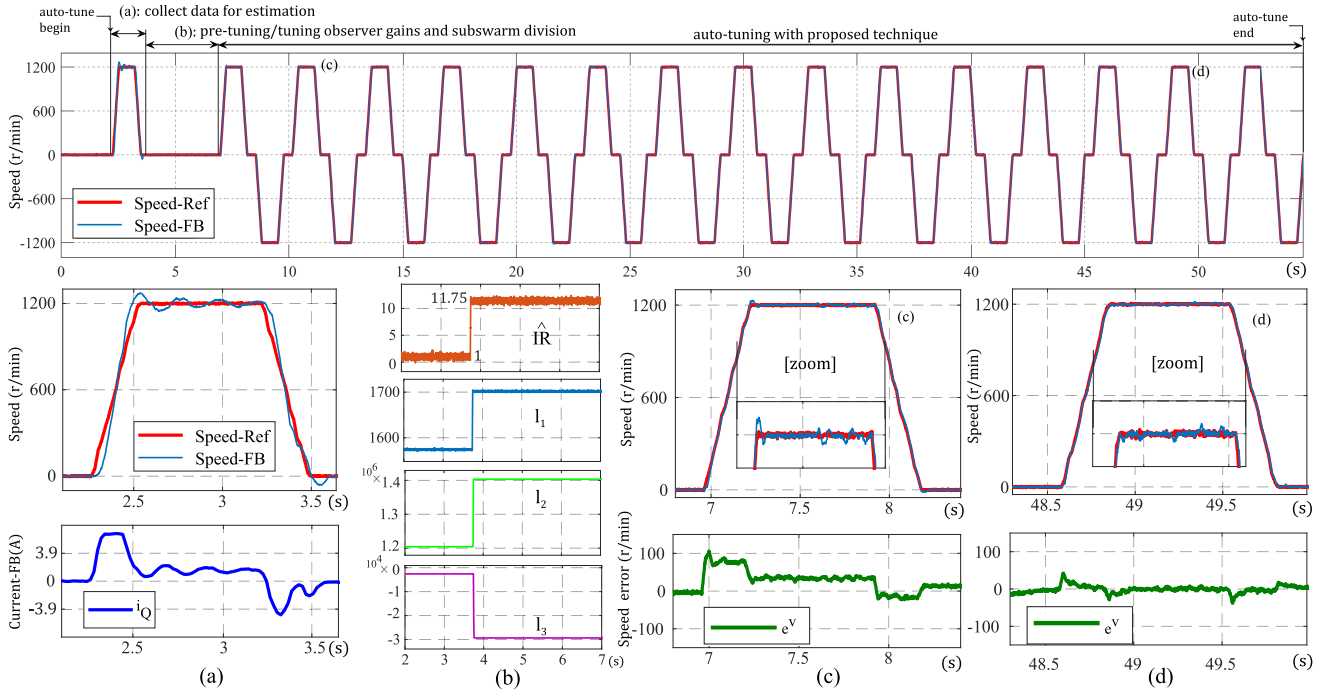

FIGURE 9. The speed waveforms for the auto-tuning process. (a) The data collection for inertia ratio estimation. (b) The estimated inertia ratio and tuned observer gains. (c) The speed performance and speed error of the best set parameters in the 1st iteration. (d) The speed performance and speed error of the best set parameters in the 3rd iteration.

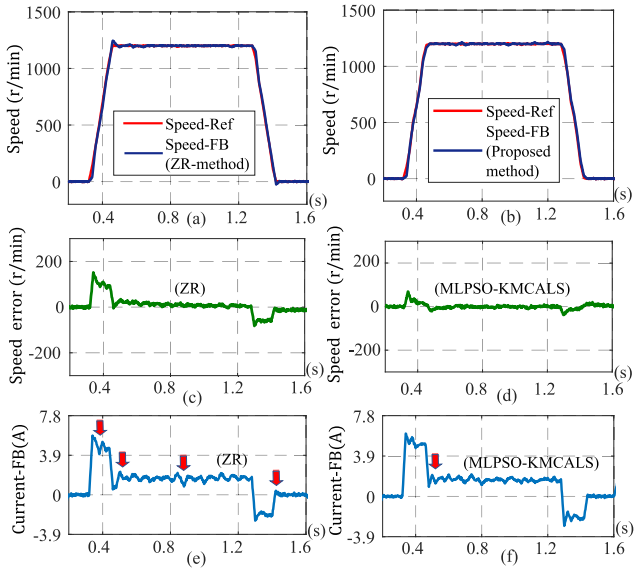

FIGURE 10. Performance comparison of the conventional Zero-Cancellation method ($K_p^v \approx 22.18, K_i^v \approx 0.01$) and the proposed method ($K_p^v \approx 27.2, K_i^v \approx 0.01, K_d^v \approx 0.0093$) under low-load condition. (a) and (b) Speed response of the ZR method and the MLPSO-KMCALS method. (c) and (d) Speed error of the ZR method and MLPSO-KMCALS method. (e) and (f) Current feedback of the ZR method and the MLPSO-KMCALS method.

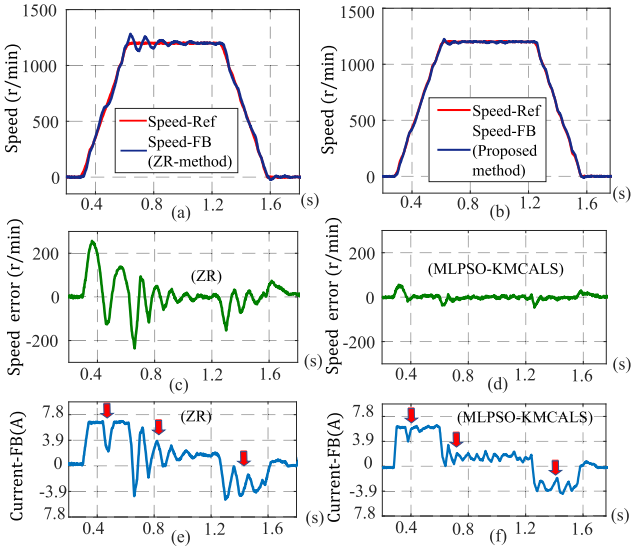

FIGURE 11. Performance comparison of the conventional Zero-Cancellation method ($K_p^v \approx 84.43, K_i^v \approx 0.02$) and the proposed method ($K_p^v \approx 110, K_i^v \approx 0.01, K_d^v \approx 0.01$) under high-load condition. (a) and (b) Speed response of the ZR method and the MLPSO-KMCALS method. (c) and (d) Speed error of the ZR method and the MLPSO-KMCALS method. (e) and (f) Current feedback of the ZR method and MLPSO-KMCALS method.

speed changes. As discuss in Section II, the K_p^v and K_i^v value of the Zero Cancellation method are tuned depending on the $\hat{J}$, $\hat{B}$ and, ω^v. However, the estimation error is larger than 5%, the viscous friction value is too small while the encoder resolution is too large and, the selected cutoff frequency of the speed loop is suboptimal. Those are the reasons for the poor response ability of the Zero Cancellation method compared to the proposed method.

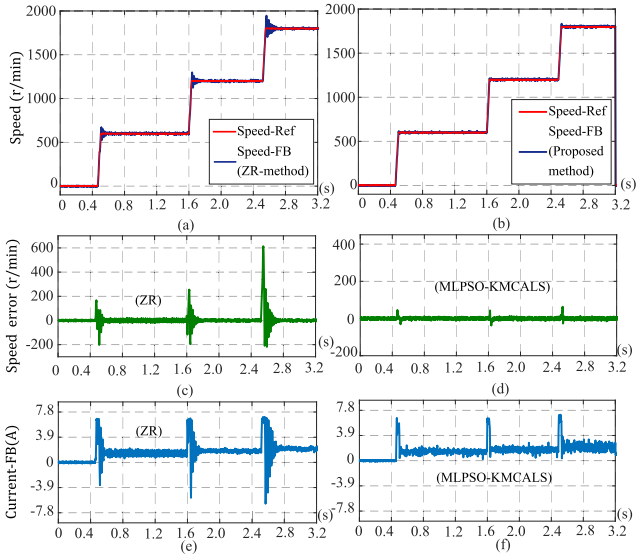

FIGURE 12. Performance comparison of the conventional Zero-Cancellation method and the proposed method under multiple speed points in a single cycle. (a) and (b) Speed response of the ZR method and the MLPSO-KMCALS method. (c) and (d) Speed error of the ZR method and the MLPSO-KMCALS method. (e) and (f) Current feedback of the ZR method and MLPSO-KMCALS method.

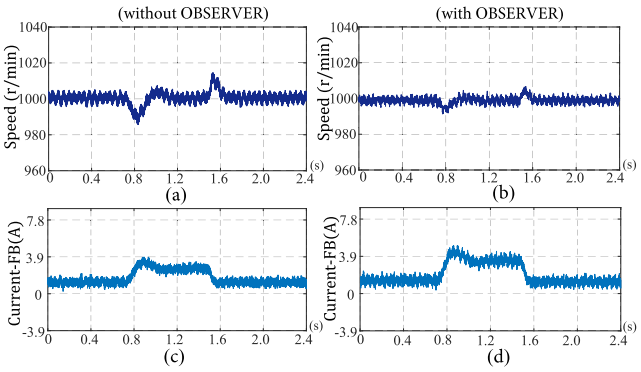

FIGURE 13. The speed response and current feedback under the load torque suddenly occur. (a) and (b) Speed response under proposed controller with observer and without observer. (c) and (d) current feedback under proposed controller with observer and without observer.

To mitigate the sudden change of load torque, we proposed the load torque disturbance observer to compensate. The experimental results are shown in the Fig. 13. An external load torque ($T_L \approx 0.15\ N.m$) is generated within $1s$ by a brake powder clutch. At the constant speed ($1000r/min$), the dynamic response of the system with the proposed observer can achieve a better disturbance rejection capability.

C. COMPARISON OF THE EXPERIMENTAL RESULTS WITH THOSE OF THE CONVENTIONAL PSO AND GLOBAL MLPSO METHODS

To evaluate the effect of the proposed method on tuning time, we compared the fitness function according to the number of iterations of the motor with the conventional PSO and global

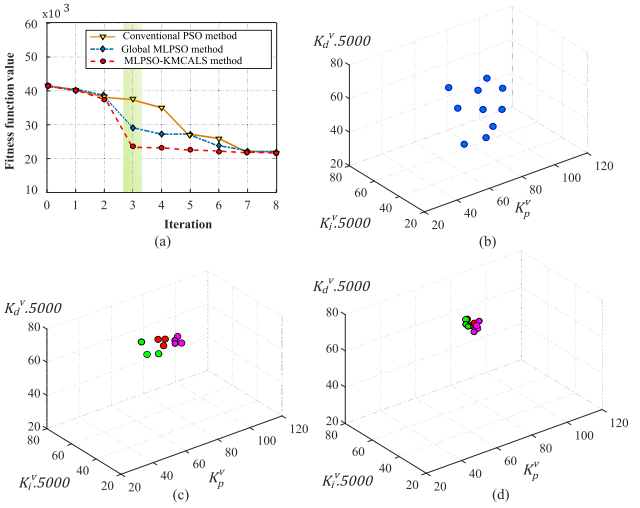

FIGURE 14. Comparison of the convergence time and converging position of the particles. (a) Convergence time according to the number of iteration of the motor. (b) Position of the particles at the 3^{rd} iteration with the conventional PSO method. (c) Position of the particles at the 3^{rd} iteration with the global MLPSO method. (d) Position of the particles at the 3^{rd} iteration with the proposed method.

MLPSO methods (Fig. 14 (a)). We can easily observe that the rate of convergence is affected by the equation updating the velocity of each particle. When comparing (5) of the conventional PSO method and (23) of MLPSO method,the equation of updating the velocity of each particle is accelerated by a different value compared to the best-SUB particle.Experimental results have proven the theory to be correct. At the 3^{rd} iteration corresponding to the 30^{th} (tuning time $\approx 60s$) rotation of the motor ($N_{size} = 10$), the fitness function value is minimized with the proposed method, while the fitness function value is not optimized in the conventional PSO and global MLPSO methods until the 80^{th} and 70^{th} rotation (tuning time $\approx 140s$) and $\approx 160s$), respectively.

We also compare the convergence quality of the three methods in Fig. 14 (b), (c), and (d). After just the 30^{th} rotation of the motor, the particles have almost converge to the best position in the proposed method. More clearly, we compare the high speed performance ($1500r/min$) with the best parameters is tuned at 3^{rd} iteration of the conventional PSO, MLPSO and MLPSO-KMCALS method in the Fig. 15. The speed error (Fig. 15 (b),(d),(f)) decreases in order conventional PSO, MLPSO and MLPSO-KMCALS method. As presented in Section IV, we can compare the velocity update equation of the conventional PSO (5), MLPSO (23) and MLPSO-KMCALS method (24). The MLPSO method considered adding a layer when compared to the conventional PSO component ($pbest_{i2}^t - z_i^t$) in (23), which improves the convergence speed. Compared with the proposed method, we have applied the pre-tuning process to reduce the limit of the initial particle value. Moreover, the KMC and ALS method are integrated into the MLPSO method to further increase the convergence speed, and avoid local optima.

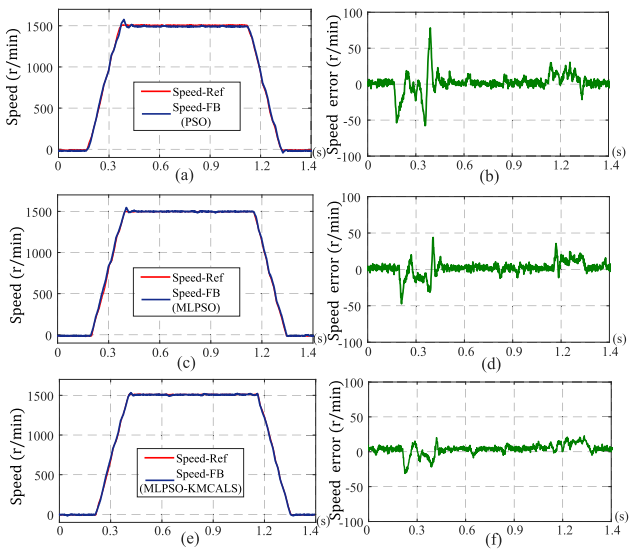

FIGURE 15. Speed performance and speed error with the best parameters tuned at the 3^{rd} iteration of the conventional PSO ((a) and (b)), MLPSO ((c) and (d)) and, MLPSO-KMCALS methods ((e) and (f)).

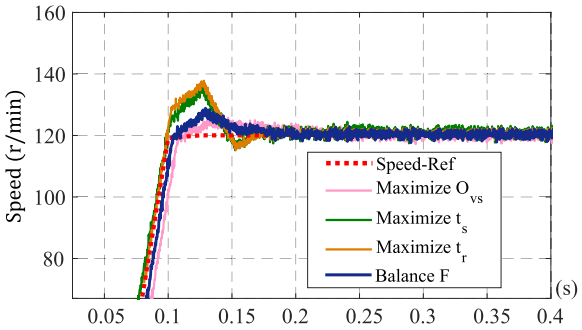

FIGURE 16. Speed performance corresponding to the four fitness function.

Experimental results have proven that the proposed method is effective. Therefore, we can conclude that our proposed method for tracking speed and optimal tuning time shows good stability, high accuracy and fast convergence time.

To compare the effects of different functions of $F(K^v, q)$, an experiment in acceleration of approximately zero is performed. Fig. 16 shows the the speed feedback corresponding to the sets of parameters tuned with different fitness functions. After the 31^{th} rotation of the tuning gains at $120r/min$, with Maximize t_s ($[q_1, q_2, q_3] = [0, 1, 0]$) and ($t_r$) ($[q_1, q_2, q_3] = [0, 0, 1]$) function, the speed feedback have a slight similarity. The overshoot of both functions is larger than the Maximize O_{vs} ($[q_1, q_2, q_3] = [1, 0, 0]$) and Balance F ($[q_1, q_2, q_3] = [0, 0, 0]$) functions, but the settling time and rise time are shorter. Depending on the specific requirements, the fitness function will be used differently.

VI. CONCLUSION

In this paper, we developed an online-auto tuning scheme for the PID controller tuning of PMSM drives that can adapt to varying load conditions. The mechanical parameter was estimated first. Then, a LUT was used to identify the initial range of the parameters for the purpose of reducing the tuning time. The speed control loop parameters are tuned automatically based on the MLPSO-KMCALS algorithm. The experimental results verified that the proposed scheme, which includes multiple-layers, KMC subswarm division, an adaptive learning strategy and an advanced fitness function, is accurate and satisfies the stability criteria. Furthermore, the population diversity is improved to overcome the local optima problem and reduce the convergence time. Therefore, this method can be widely applied to industrial applications.

REFERENCES

[1] Y. I. Son, I. H. Kim, D. S. Choi, and H. Shim, "Robust cascade control of electric motor drives using dual reduced-order PI observer," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3672–3682, Jun. 2015.
[2] H. Zhang, Y. Shi, and A. S. Mehr, "Robust $\mathcal{H}_\infty$ PID control for multivariable networked control systems with disturbance/noise attenuation," *Int. J. Robust Nonlinear Control*, vol. 22, no. 2, pp. 183–204, Jan. 2012.
[3] D. Simhachalam, C. Dey, and R. K. Mudi, "An auto-tuning PD controller for DC servo position control system," in *Proc. IEEE Power, Control Embedded Syst.*, Dec. 2012, pp. 1–6.
[4] T. Higashiyama, M. Mine, H. Ohmori, A. Sano, H. Nishida, and Y. Todaka, "Auto-tuning of motor drive system by simple adaptive control approach," in *Proc. IEEE Int. Conf. Control Appl.*, Sep. 2000, pp. 868–873.
[5] K. H. Kim, "Nonlinear speed control for a PM synchronous motor with a sequential parameter auto-tuning algorithm," *IEE Proc., Electr. Power Appl.*, vol. 152, no. 5, pp. 1253–1262, May 2005.
[6] S.-M. Yang and Y.-J. Deng, "Observer-based inertial identification for auto-tuning servo motor drives," in *Proc. Conf. Rec. 40th IAS Annu. Meeting*, Oct. 2005, pp. 968–972.
[7] J. G. Ziegler and N. B. Nichols, "Optimal settings for automatic controllers," *Trans. ASME*, vol. 64, no. 11, pp. 759–768, 1942.
[8] P. Meshram and R. G. Kanojiya, "Tuning of PID controller using Ziegler–Nichols method for speed control of DC motor," in *Proc. IEEE Int. Conf. Adv. Eng., Sci. Manage.*, 2012, pp. 117–122.
[9] K. Y. Cheng and Y. Y. Tzou, "Fuzzy optimization techniques applied to the design of a digital PMSM servo drive," *IEEE Trans. Power Electron.*, vol. 19, no. 4, pp. 1085–1099, Jul. 2004.
[10] H. H. Choi, H. M. Yun, and Y. Kim, "Implementation of evolutionary fuzzy PID speed controller for PM synchronous motor," *IEEE Trans. Ind. Informat.*, vol. 11, no. 2, pp. 540–547, Apr. 2015.
[11] J.-W. Jung, V. Q. Leu, T. D. Do, E.-K. Kim, and H. H. Choi, "Adaptive PID speed control design for permanent magnet synchronous motor drives," *IEEE Trans. Power Electron.*, vol. 30, no. 2, pp. 900–908, Feb. 2015.
[12] V. Chopra, S. K. Singla, and L. Dewan, "Comparative analysis of tuning a PID controller using intelligent methods," *ACTA Polytechnica Hungarica*, vol. 11, no. 8, pp. 235–249, 2014.
[13] H. N. Tran, K. M. Le, and J. W. Jeon, "Adaptive current controller based on neural network and double phase compensator for a stepper motor," *IEEE Trans. Power Electron.*, vol. 34, no. 8, pp. 8092–8103, Aug. 2019.
[14] S.-Z. Zhao, M. W. Iruthayarajan, S. Baskar, and P. N. Suganthan, "Multi-objective robust PID controller tuning using two lbests multi-objective particle swarm optimization," *Inf. Sci.*, vol. 181, no. 16, pp. 3323–3335, Aug. 2011.
[15] S. Bassi, M. Mishra, and E. Omizegba, "Automatic tuning of proportional-integral-derivative (PID) controller using particle swarm optimization (PSO) algorithm," *Int. J. Artif. Intell. Appl.*, vol. 2, no. 4, pp. 25–32, 2011.
[16] M. Calvini, M. Carpita, A. Formentini, and M. Marchesoni, "PSO-based self-commissioning of electrical motor drives," *IEEE Trans. Ind. Electron.*, vol. 62, no. 2, pp. 768–776, Feb. 2015.
[17] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Hum. Sci.*, 1995, pp. 39–43.
[18] 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.

[19] Y. Zhang, X. Liu, F. Bao, J. Chi, C. Zhang, and P. Liu, "Particle swarm optimization with adaptive learning strategy," *Knowl.-Based Syst.*, vol. 196, May 2020, Art. no. 105789.
[20] L. Wang, B. Yang, and Y. Chen, "Improving particle swarm optimization using multi-layer searching strategy," *Inf. Sci.*, vol. 274, pp. 70–94, Aug. 2014.
[21] J. S. Ko, J. H. Lee, S. K. Chung, and M. J. Youn, "A robust digital position control of brushless DC motor with dead beat load torque observer," *IEEE Trans. Ind. Electron.*, vol. 40, no. 5, pp. 512–520, Oct. 1993.
[22] M. Lazor and M. Stulrajter, "Modified field oriented control for smooth torque operation of a BLDC motor," in *Proc. ELEKTRO*, Rajecké Teplice, Slovakia, May 2014, pp. 180–185.
[23] S. Yang and K. Lin, "Automatic control loop tuning for permanent-magnet AC servo motor drives," *IEEE Trans. Ind. Electron.*, vol. 63, no. 3, pp. 1499–1506, Mar. 2016.
[24] Z. Qi, Q. Shi, and H. Zhang, "Tuning of digital PID controllers using particle swarm optimization algorithm for a CAN-based DC motor subject to stochastic delays," *IEEE Trans. Ind. Electron.*, vol. 67, no. 7, pp. 5637–5646, Jul. 2020.
[25] A. N. Diligenskaya, *Control Object Identification*. Samara, Russia: Publishing House of Samara State Technical Univ., 2009.
[26] S. V. Stelmashchuk, "Identification of static moment and inertia moment of electric drive with the method of least squares," in *Proc. Int. Conf. Ind. Eng., Appl. Manuf. (ICIEAM)*, May 2020, pp. 1–5.
[27] W. Lu, D. Zheng, Y. Lu, K. Lu, L. Guo, W. Yan, and J. Luo, "New sensorless vector control system with high load capacity based on improved SMO and improved FOO," *IEEE Access*, vol. 9, pp. 40716–40727, 2021.
[28] Q. Wang, H. Yu, M. Wang, and X. Qi, "An improved sliding mode control using disturbance torque observer for permanent magnet synchronous motor," *IEEE Access*, vol. 7, pp. 36691–36701, 2019.
[29] S. K. Kommuri, S. B. Lee, and K. C. Veluvolu, "Robust sensors-fault-tolerance with sliding mode estimation and control for PMSM drives," *IEEE/ASME Trans. Mechatronics*, vol. 23, no. 1, pp. 17–28, Feb. 2018.
[30] C. C. Aggarwal, "A human-computer interactive method for projected clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 4, pp. 448–460, Apr. 2004.
[31] J. Wang and X. Su, "An improved K-Means clustering algorithm," in *Proc. IEEE 3rd Int. Conf. Commun. Softw. Netw.*, Xi'an, China, May 2011, pp. 44–46.
[32] A. A. A. El-Gammal and A. A. El-Samahy, "A modified design of PID controller for DC motor drives using particle swarm optimization PSO," in *Proc. Int. Conf. Power Eng., Energy Electr. Drives*, Mar. 2009, pp. 419–424.
[33] M. Clerc and J. Kennedy, "The particle swarm—Explosion, stability, and convergence in a multidimensional complex space," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58–73, Feb. 2002.

HOANG NGOC TRAN received the B.S. degree in mechatronics engineering from the Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam, in 2015, and the Ph.D. degree in electrical and computer engineering from Sungkyunkwan University, Suwon, South Korea, in 2020.

He is currently a Postdoctoral Researcher with the College of Information and Communication Engineering, Sungkyunkwan University. His research interests include signal processing, motion control, robotics, and embedded systems.

TY TRUNG NGUYEN received the B.S. degree in automatic control engineering from Hanoi University, Hanoi, Vietnam, in 2018. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the School of Information and Communication Engineering, Sungkyunkwan University, Suwon, South Korea.

His research interests include signal processing, motion control, and robotics.

HUNG QUANG CAO received the B.S. degree in mechatronics engineering from the Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam, in 2016. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the School of Information and Communication Engineering, Sungkyunkwan University, Suwon, South Korea.

His research interests include robotics, motion control, and embedded systems.

TON HOANG NGUYEN received the B.S. degree in mechatronics engineering from the Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam, in 2016. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the School of Information and Communication Engineering, Sungkyunkwan University, Suwon, South Korea. His research interests include signal processing, motion control, robotics, and embedded systems.

HA XUAN NGUYEN received the B.S. degree in mechatronics engineering from the Ho Chi Minh City University of Technology, Ho Chi Minh City, Vietnam, in 2015, and the Ph.D. degree in electrical and computer engineering from Sungkyunkwan University, Suwon, South Korea, in 2020.

He is currently a Postdoctoral Researcher with the College of Information and Computer Engineering, Sungkyunkwan University. His research interests include signal processing, motion control, robotics, and embedded systems.

JAE WOOK JEON (Senior Member, IEEE) received the B.S. and M.S. degrees in electronics engineering from Seoul National University, Seoul, South Korea, in 1984 and 1986, respectively, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 1990.

From 1990 to 1994, he was a Senior Researcher with Samsung Electronics, Suwon, South Korea. Since 1994, he has been with Sungkyunkwan University, Suwon, where he was an Assistant Professor with the School of Electrical and Computer Engineering and is currently a Professor with the School of Information and Communication Engineering. His research interests include robotics, embedded systems, and factory automation.