International Journal of Robotics and Control Systems
Vol. 6, No. 2, 2026, pp. 1271-1291
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

Adaptive PI Controller using Puma Optimization with Kernel Extreme Learning Machine for Dynamic Response Improvements of Brushless DC Motor

Herlambang Setiadi [a,b,1], Muhammad Syahril Mubarok [c,d,e,2,*], Ananta Adhi Wardana [c,3], Yoga Uta Nugraha [c,d,e,4], R. Thirumalaivasan [f,5], Nur Vidia Laksmi B. [g,6], Habib Miftahudin Alfatah [c,7]

[a] Electrical Engineering Study Program, School of Electrical Engineering, Telkom University, Bandung 40257, Indonesia
[b] Centre of Excellence Smart Transportation and Robotics (STAR), Telkom University, Bandung 40257, Indonesia
[c] Department of Engineering, Faculty of Advanced Technology and Multidiscipline, Universitas Airlangga, Surabaya 60155, Indonesia
[d] Electrical Engineering, Renewable, and Sustainable Energy Technology Research Group, Universitas Airlangga, Surabaya 60155, Indonesia
[e] Research Center for New and Renewable Energy Engineering (RCNREE), Universitas Airlangga, Surabaya 60155, Indonesia
[f] School of Electrical Engineering, Vellore Institute of Technology, Vellore, Tamilnadu 632014, India
[g] Department of Electrical Engineering, Faculty of Vocational Studies, Universitas Negeri Surabaya, Surabaya 60231, Indonesia
[1] herlambangsetiadi@telkomuniversity.ac.id; [2] syahril.mubarok@ftmm.unair.ac.id; [3] ananta.adhi@ftmm.unair.ac.id;
[4] yoga.uta.n@ftmm.unair.ac.id; [5] thirumalai.r@vit.ac.in; [6] nurvidialaksmi@unesa.ac.id;
[7] habib.miftahudin.alfata-2022@ftmm.unair.ac.id
* Corresponding Author

ARTICLE INFO

Article history
Received February 08, 2026
Revised April 14, 2026
Accepted April 23, 2026

Keywords
Brushless DC Motor;
Speed Regulation;
Kernel Extreme Learning Machine;
Puma Optimization;
Metaheuristic

ABSTRACT

This paper presents a novel approach for controlling the speed of Brushless DC (BLDC) motors using an adaptive Proportional–Integral (PI) controller. Conventional PI controllers suffer from its performance when speed and load torque reference change due to their fixed parameter structure. To address these limitations, the proposed method integrates Hybrid Puma Optimization with Kernel Extreme Learning Machine (KELM) to enhance controller adaptability under varying conditions, including speed changes and load torque. In the proposed method, Puma Optimization is employed to optimally tune PI parameters under diverse operating conditions, generating high-quality training data. These optimized parameters are used to train a KELM model, enabling real-time adaptive adjustment of PI parameter based on reference speed and load torque variations. The effectiveness is validated through MATLAB/Simulink simulations and the results are compared with PI controllers tuned using Extreme Learning Machine (ELM) and Artificial Neural Network (ANN). Simulation results demonstrate that PI-KELM effectively adjusts to dynamic operating conditions, thereby improving the overall performance of the BLDC motor. The proposed method achieves superior dynamic performance with smallest overshoot, faster settling time, and improved damping behavior. PI-KELM significantly improves stability compared to the baseline with 25% improvement in settling time, 83.33% in overshoot, and 50% slower in rise time. Furthermore, the PI–KELM controller yields the lowest MSE of 0.567 during training and significantly reduced ITAE, IAE, and ISE indices during testing. Compared to conventional scenarios, the proposed method exhibits superior dynamic response and robustness.

1. Introduction

Brushless Direct Current (BLDC) motors have become increasingly vital in modern electromechanical systems due to their high efficiency, compact design, and superior controllability compared to conventional brushed motors [1]. The absence of brushes and mechanical commutators reduces maintenance requirements, enhances reliability, and extends operational lifespan [2]. These advantages, combined with low acoustic noise and excellent dynamic response, have positioned BLDC motors as a preferred solution in electric vehicles, industrial automation, and renewable energy systems [3]. Their ability to deliver high torque across variable speeds while maintaining energy efficiency underscores their importance in the global push toward sustainable technologies [4], [5].

Recent research emphasizes advancements in BLDC motor design, control strategies, and integration with renewable energy sources. Developments in sensorless control, fault-tolerant operation, and intelligent power electronics have expanded their applicability in demanding environments [6]. Moreover, optimization techniques for speed regulation and thermal management continue to be explored to improve performance and reliability. As industries prioritize energy-efficient and sustainable solutions, BLDC motors are expected to play a pivotal role in shaping next-generation electrical drives and mobility systems [7].

One of the key factors influencing the performance of Brushless DC (BLDC) motors is the controller. Various control strategies have been developed to regulate BLDC motor speed and torque, with Field-Oriented Control (FOC) and Vector Control being among the most widely adopted due to their ability to achieve high dynamic response and efficiency. In both methods, the controller plays a critical role in stabilizing motor operation and ensuring precise performance under varying load conditions. Among the controllers commonly employed in both industrial applications and academic research, the Proportional–Integral (PI) controller remains the most prevalent due to its simplicity, ease of implementation, and reliable performance in speed regulation and current control [8]. Recent studies continue to refine PI-based approaches, often integrating them with advanced modulation techniques or optimization algorithms to enhance robustness and adaptability in modern BLDC drive systems [9].

The application of the Proportional–Integral (PI) controller for induction motor drives has been widely reported in the literature. As demonstrated in [10], the PI controller significantly enhances the dynamic performance and stability of induction motors under varying operating conditions. A comparative study of different PI controller structures for induction motor control is presented in [11], where indirect field-oriented control (IFOC) is employed to regulate motor speed. The study further highlights the hybrid integration of PI controllers with Adaptive Neuro-Fuzzy Inference Systems (ANFIS), which improves robustness and adaptability. Extending this approach to Brushless DC (BLDC) motors, Subbarao *et al.* [12] reported a performance comparison between conventional PI controllers and ANFIS-based controllers in electric vehicle applications, demonstrating that intelligent hybrid control strategies can outperform classical methods in terms of efficiency and dynamic response.

Hybrid controller strategies have been extensively investigated to improve the performance of electric machines. In [13], a hybrid approach combining fuzzy logic and PI controllers was proposed for single-phase induction motors, where a multilevel inverter acted as the actuator and renewable energy served as the power source. Simulation results demonstrated that this hybrid strategy significantly enhanced the dynamic response and efficiency of the induction motor. Similarly, a hybrid artificial intelligence technique integrating fuzzy logic and Particle Swarm Optimization (PSO) for BLDC motor control was reported in [14]. In this study, fuzzy logic and PSO were employed to achieve self-tuning of the PI controller, resulting in notable improvements in speed regulation and overall performance. Furthermore, Naqvi *et al.* [15] introduced a swarm-based optimization method for tuning PI controller parameters in BLDC motor drives. Their findings confirmed that swarm-based tuning effectively identified optimal PI gains, thereby enhancing the stability, efficiency, and robustness of BLDC motor operation.

From the research discussed above, it is evident that the tuning of PI controller parameters plays a crucial role in determining the overall performance of BLDC motor drives. However, since the operating conditions of such systems are inherently dynamic and subject to continuous variation, a static PI controller often fails to deliver optimal control performance. This limitation highlights the necessity of employing adaptive PI controllers that can adjust their parameters in real time to maintain stability and efficiency. Among the advanced approaches proposed in recent studies, the Kernel Extreme Learning Machine (KELM) has emerged as a promising method for adaptive tuning. By leveraging kernel-based learning and fast training capabilities, KELM enables the PI controller to self-adjust under varying load and speed conditions, thereby enhancing robustness, accuracy, and dynamic response in BLDC motor applications.

The central challenge lies in identifying the most effective PI controller that can be trained using Kernel Extreme Learning Machine (KELM). Training an unoptimized PI controller would yield limited benefits, as the resulting performance would not be impactful. Therefore, optimization techniques particularly those inspired by nature offer promising solutions to this problem. For instance, [16] reports the use of Red Fox Optimization to tune PID controller parameters in a smart LED driver circuit. The results demonstrate that applying this method reduces overshoot and achieves faster settling times, thereby enhancing system stability. Similarly, [17] discusses the application of Whale Optimization Algorithms for tuning PID parameters in a DC motor system. Simulation outcomes reveal that the optimized PID controller improves the transient response, as evidenced by reduced overshoot and shorter settling times. In another study, Soni et al. [18] propose an automatic generation control scheme based on the Grey Wolf Optimizer. In this approach, the PI controller serves as the automatic generation controller, while the Grey Wolf Optimizer fine-tunes its parameters. Simulation results confirm that this method significantly enhances the frequency dynamic response of the power system. Furthermore, [19] highlights the use of gradient-based optimization for PID parameter tuning in hybrid power systems, demonstrating its effectiveness in improving system performance. Collectively, these studies underscore the potential of nature-inspired algorithms to achieve optimal parameter tuning. Among the various approaches, Puma Optimization has recently gained popularity due to its superior performance [19].

The research contribution is to proposes a novel hybrid approach that integrates Puma Optimization (PO) with Kernel Extreme Learning Machine (KELM) for the design of an adaptive PI controller to regulate the speed of Brushless DC (BLDC) motors. Unlike conventional static PI controllers, the proposed method enables adaptive tuning of controller parameters under varying operating conditions, thereby enhancing robustness, efficiency, and dynamic performance. By combining the global search capability of PO with the fast learning and nonlinear mapping strength of KELM, the hybrid controller achieves superior adaptability and precision in speed regulation.

The remainder of this paper is organized as follows: Chapter 2 presents the fundamental theory and background of BLDC motor control and adaptive PI strategies. Chapter 3 outlines the research methodology, including the design of the hybrid PO–KELM controller. Chapter 4 discusses the simulation results and performance evaluation of the proposed method. Finally, Chapter 5 provides conclusions and future recommendations, highlighting potential extensions of this work for advanced electric drive applications

2. Method

2.1. Kernel Extreme Learning Machine (KELM)

The Extreme Learning Machine (ELM) is based on a feed-forward neural network with a single hidden layer (SLFN) [20]. It is well known for its exceptionally fast ability to map input–output relationships, overcoming the limitations of conventional neural networks by minimizing empirical risk and reducing training errors [21]. Unlike traditional networks that require iterative training over many epochs, ELM eliminates iterative parameter updates, making the training process significantly faster [22].

However, ELM has drawbacks, particularly in determining the number of hidden neurons, which is often done through trial and error [23]. In addition, random initialization of input weights and biases typically requires a larger number of hidden neurons to achieve optimal performance [24]. To address these issues, kernel-based ELM variants project input data into higher-dimensional feature spaces using kernel functions [25]. This transformation linearizes nonlinear patterns and improves stability, eliminating reliance on randomly assigned parameters [26].

The mathematical model for the ELM output with Mtraining samples is defined as follows [27]:

$$y_M(x) = \sum_{i=1}^{M} \beta_i h_i(x) \tag{1}$$

where $\beta = [\beta_1, \beta_2, \dots, \beta_M]$; the output weights between the hidden layer of M neuron, and output neuron $h(x) = [h_1(x), h_2(x), \dots, h_M(x)]$; the output of hidden layer with respect to input x [28].

The ELM learning algorithm aims to simultaneously minimize the training errors and the magnitude of the output weights, as defined in [29]:

$$Min: \|H\beta - T\|, \|\beta\|. \tag{2}$$

To address the problem of simultaneously minimizing the training errors and the output weights, the Karush–Kuhn–Tucker (KKT) optimality conditions can be applied as follows [30]:

$$\beta = H^T \left(\frac{1}{C} + HH^T\right)^{-1} T \tag{3}$$

where H is the hidden layer output; C is the regularization coefficient; and T is the predicted output.

By substituting (3) into (1), the ELM output can be expressed as shown in the following equation [31]:

$$y(x) = h(x)H^T \left(\frac{1}{C} + HH^T\right)^{-1} T \tag{4}$$

If the feature mapping $h(x)$ is unknown, it can be addressed using a kernel method that satisfies Mercer's condition, as proposed by G.-B. Huang [32]. The kernel function is defined as follows [33]:

$$O = HH^T : m_{ij} = h(x_i)h(x_j) = \Omega(x_i, x_j) \tag{5}$$

The output function $y(x)$of the Kernel Extreme Learning Machine (KELM) is defined as follows [34]:

$$y(x) = [\Omega(x, x_1), \dots, \Omega(x, x_M))] \left(\frac{1}{C} + O\right)^{-1} T \tag{6}$$

where $O = HH^T$ and $\Omega(x, y)$ is the kernel function of hidden neurons of SLFN.

The support vector machine (SVM) model resembles (2) by providing closed-form expressions for the kernel coefficients [35]. To evaluate the performance of the ELM learning algorithm, the radial basis function (RBF) is employed as the kernel, as expressed in the following equation:

$$\Omega(x, y) = exp\left(\frac{\|x - y\|^2}{2\sigma^2}\right) \tag{7}$$

where σ = kernel parameter.

The selection of the regularization coefficient $C$and the kernel parameter σmust be performed carefully, as these parameters have a significant impact on the performance of KELM. The use of RBF kernel because it has ability to handle complex and nonlinear function. On the other hand, the RBF kernel function has the better numerical stability which makes RBF robust during optimization phase of training. Compared to standard ELM, the KELM learning algorithm demonstrates better stability, and it also trains faster than SVM [36]. Fig. 1 shows the network structure of the KELM algorithm.

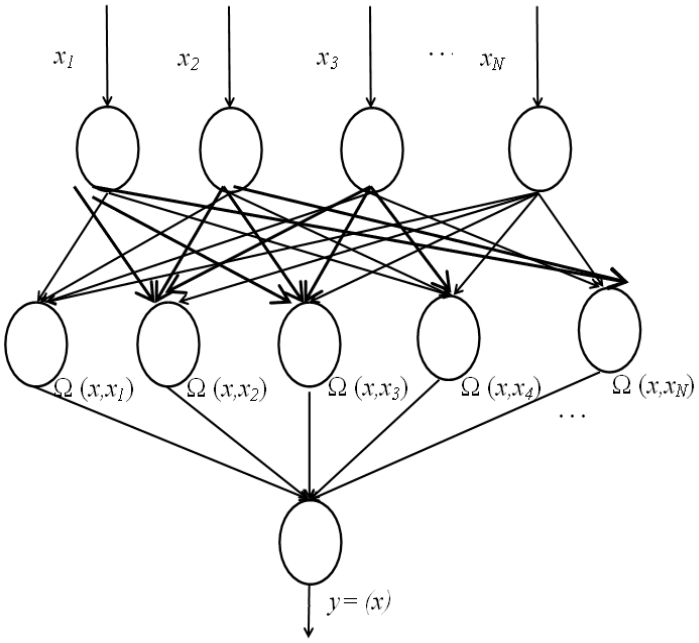

Fig. 1. Network structure of KELM algorithm

2.2. Puma Optimization

Puma Optimization (PO) was first established in 2024 by Abdollahzadeh and his collage [37]. Puma optimization is based on the intelligence and behavior of pumas, a predatory species commonly found in the America continent [38]. In the wild, pumas often roam their large territories to patrol and hunt. They conduct random searches within their territory or randomly visit the territories of other pumas [39]. Inspired by this behavior, a mechanism for generating new solutions in the exploration phase of the PO algorithm is built, as shown in the (8) [40].

$$If\ rand_1 > 0.5,\ \ Z_{i,G} = R_{Dim}.(Ub - Lb) + Lb \tag{8}$$

If (8) could not find the prey. Hence the Pumas goes to exploration step. The exploration step can be mathematically modelled as described in (9) [41].

$$\begin{aligned} Z_{i,G} = X_{a,G} + G.\left(X_{a,G} - X_{b,G}\right) + G.\left(\left(X_{a,G} - X_{b,G}\right) - \left(X_{c,G} - X_{d,G}\right)\right) \\ + G.\left(\left(X_{c,G} - X_{d,G}\right) - \left(X_{e,G} - X_{f,G}\right)\right) \end{aligned} \tag{9}$$

where $Z_{i,G}$ is the produced solution; Ub and Lb is upper and lower limit. Furthermore, R_{Dim} dan $rand_1$ are random value from 0 to 1. In addition, $X_{a,G}, X_{b,G}, X_{c,G}, X_{d,G}, X_{e,G}, X_{f,G}$ are all the population that chosen randomly. Moreover, G are the value that can be calculated using (10) [41].

$$G = 2 \cdot rand_2 - 1 \tag{10}$$

where $rand_2$ is random value from 0 to 1.

From (10) and (11), one of the equations will be used to produced new solutions. The mathematical representation of this step can be described using (11) [41].

$$\begin{cases} If\ j = j_{rand}\ \ or\ rand_3 \le U, & X_{baru} = Z_{i,G} \\ \quad if\ not, & X_{baru} = X_{a,G} \end{cases} \tag{11}$$

where X_{baru} is the new solution; j is the recent variable and j_{acak} is the random number. Furthermore, $rand_3$ is the random number from 0 to 1. Moreover, U is the static parameter from 0 to 1 that can be updated using (12). This parameter ensures a high diversity of the selected solutions. In (5), $CostX_i$ and $CostX_{baru}$ are the current fitness function and the new fitness function. Furthermore, X_i, $X_{i,baru}$, and N_{total} are current solution, new solutions and number of pumas respectively [41].

$$\begin{cases} If\ \mathrm{Cost}X_{\mathrm{new}} < \mathrm{Cost}X_i, & X_i = X_{new} \\ \quad If\ not\ U = U + \dfrac{1-U}{N_{pop}} \end{cases} \tag{12}$$

$$\begin{cases} if\ rand_4 \ge 0.5, & X_{new} = \dfrac{\frac{mean(Sol_{total})}{N_{pop}}.X_1^r - (-1)^{\beta}.X_i}{1 + (\alpha.rand_5)} \\ if\ not, if\ rand_6 \ge L, & X_{new} = Puma_{male} + (2.rand_1).exp(randn_1).X_2^r - X_i \\ if\ not, & X_{new} = (2.rand_8).\dfrac{F_1.R.X_i + F_2.(1-R).Puma_{male}}{2.rand_9 - 1 + randn_2} \end{cases} \tag{13}$$

$$if\ CostX_{new} < CostX_i,\ \ X_i = X_{new} \tag{14}$$

In the exploitation phase of the PO algorithm, two operators are used to improve the quality of the solution, inspired by two puma hunting behaviors: ambush and sprint, described in (6). The operator simulating the puma's sprinting characteristics is used if $rand_4 \ge 0.5$. Otherwise, the operator simulating the ambush is used. The first operation simulates short jumps towards other puma hunters, and the second operation simulates long jumps towards the best puma hunter [42].

In (13) and (14), $rand_4$, $rand_5$, $rand_6$, $rand_7$, $rand_8$, dan $rand_9$ are random numbers with values between 0 and 1. $randn_1$ and $randn_2$ are random numbers from a normal distribution. Sol_{total} represents the sum of all solutions. Furthermore, α and L are static parameters that must be tuned before the optimization procedure is performed. β is a randomly obtained number of 0 or 1. $Puma_{male}$ is the best solution. In addition, "*mean*" represents the mean function, and "*exp*" represents the exponential function. X_1^r is a randomly selected solution from the entire population. Finally, the parameters X_2^r, R, F_1, and F_2 are calculated by the equations found in [40].

The phase change mechanism is proposed based on other characteristics of pumas, namely: excellent memory and high intelligence. Based on these characteristics, the phase change mechanism is divided into two stages: the early stage and the experienced stage [43]. In the early stage of life, pumas do not have much experience, as a result they explore their territory and hunt at the same time. In this stage, the exploration and exploitation phases are carried out simultaneously in the first three iterations. After the early stage of life, pumas have enough experience to optimize the decision to choose to explore new areas in their territory or hunt in places where prey is often found [31]. For the fourth iteration, the decision to enter the exploration phase and the exploitation phase is made by calculating the $Score_{Exploration}$ and $Score_{Exploitation}$ using (15), (16). If the $Score_{Exploration}$ is greater than the $Score_{Exploitation}$, the exploration phase is selected and vice versa if the $Score_{Exploitation}$ is greater [44].

$$Score_{Exploration} = \left(PF_1.f_{1Exploration}\right) + \left(PF_2.f_{2Exploration}\right) \tag{15}$$

$$Score_{Exploitation} = \left(PF_1 . f_{1Exploitation}\right) + \left(PF_2 . f_{2Exploitation}\right) \quad (16)$$

In (15), (16), PF_1 and PF_2 are fixed parameters set before the optimization process with values from 0 to 1. These parameters are used to adjust the functions f_1 dan f_2. After each iteration of this stage, the $Score^t_{Exploration}$ and $Score^t_{Exploitation}$ are calculated to select the exploration or exploitation phase in the next iteration, which are calculated using (17), (18). If the $Score^t_{Exploration}$ is higher than the $Score^t_{Exploitation}$, then the exploration phase is selected, and vice versa [45].

$$Skor^t_{Exploration} = \left(\alpha^t_{Exploration} . f_1{}^t_{Exploration}\right) + \left(\alpha^t_{Exploration} . f_2{}^t_{Exploration}\right) + \left(\delta^t_{Exploration} . \left(lc . f_3{}^t_{Exploration}\right)\right) \quad (17)$$

$$Skor^t_{Exploitation} = \left(\alpha^t_{Exploitation} . f_1{}^t_{Exploitation}\right) + \left(\alpha^t_{Exploitation} . f_2{}^t_{Exploitation}\right) + \left(\delta^t_{Exploitation} . \left(lc . f_3{}^t_{Exploitation}\right)\right) \quad (18)$$

In (17), (18), t represents the current iteration value. The parameters α, f_1, f_2, f_3, lc, and δ are calculated using the equations found in [46]. Like the parameters PF_1 and PF_2 used to adjust the functions $f_1, f_2,$, the function f_3 also uses the parameter PF_3. This parameter is a fixed value chosen before the optimization process and ranges from 0 to 1 [41]. Table 1 shows the parameter of Puma Optimization used in this paper.

Table 1. Puma optimizer parameters

Index	Value
K_p *min*	0.1
K_p *max*	1
K_i *min*	0.5
K_i *max*	100
Maximum iterations	40
Population size	20
Dimension	2
α	0.01
δ	0.91

2.3. Dynamic Modelling of BLDC Motor

The modelling of BLDC is based on the d-q transformation model. BLDC is highly influenced by current controlled pulsed width modulation (PWM) [47]. The current component of the motor is divided into two (I_d, I_q) which is the component of flux and torque in the rotor based on d-q axis [48]. The mathematical representation of BLDC is described in (19)–(23) [49].

$$T_e = \frac{3}{2}\frac{P}{2}\left[\lambda_m i_q + \left(L_d - L_q\right) i_d i_q\right] \quad (19)$$

$$\frac{d(i_d)}{dt} = \frac{v_d - r_s i_d + \omega_r L_q i_q}{L_d} \quad (20)$$

$$\frac{d(i_q)}{dt} = \frac{v_q - r_s i_q + \omega_r (L_d i_d + \lambda_m)}{L_q} \quad (21)$$

$$\frac{d(\omega_m)}{dt} = \frac{T_e - T_L - B\omega_m}{J} \tag{22}$$

$$\omega_r = \frac{P}{2}\omega_m \tag{23}$$

Several parameters corresponding to electric torque (T_e), load torque (T_L), number of pole (P) and magnetic flux (λ_m) are considered [50]. Here, i_d and i_q represent stator current in d and q axis, respectively [51]. Other parameters are inductance in d axis (L_d), inductance in q axis (L_q), stator voltage in d axis (v_d), stator voltage in q axis (v_q), momen inertia (J), and friction (B). While r_s and ω_r are stator resistance per phase and electric rotor speed, respectively [52].

In this paper, the BLDC speed is controlled using PI controller. The BLDC speed response is used as the feedback of controller [53]. Then the difference between reference speed and actual speed will be used as the signal input of the PI controller. The output of PI controller is iq^* that will be used as the input reference of the PWM inverter. The mathematical representation of iq^* are described below [54]:

$$i_q^* = \left(K_p + \frac{1}{K_i s}\right) \times \left(\omega_{reff} - \Delta\omega_r\right) \tag{24}$$

$$\begin{bmatrix} i_a \\ i_b \\ i_c \end{bmatrix} \sqrt{\frac{2}{3}} = \begin{bmatrix} cos\left(\omega_r \times \frac{1}{s}\right) & -sin\left(\omega_r \times \frac{1}{s}\right) \\ cos\left(\left(\omega_r \times \frac{1}{s}\right) - \frac{2\pi}{3}\right) & -sin\left(\left(\omega_r \times \frac{1}{s}\right) - \frac{2\pi}{3}\right) \\ cos\left(\left(\omega_r \times \frac{1}{s}\right) + \frac{2\pi}{3}\right) & -sin\left(\left(\omega_r \times \frac{1}{s}\right) + \frac{2\pi}{3}\right) \end{bmatrix} \begin{bmatrix} i_d \\ i_q \end{bmatrix} \tag{25}$$

$$\begin{bmatrix} V_d \\ V_q \\ V_0 \end{bmatrix} \sqrt{\frac{2}{3}} = \begin{bmatrix} cos\left(\omega_r \times \frac{1}{s}\right) & cos\left(\left(\omega_r \times \frac{1}{s}\right) - \frac{2\pi}{3}\right) & cos\left(\left(\omega_r \times \frac{1}{s}\right) + \frac{2\pi}{3}\right) \\ -sin\left(\omega_r \times \frac{1}{s}\right) & -sin\left(\left(\omega_r \times \frac{1}{s}\right) - \frac{2\pi}{3}\right) & -sin\left(\left(\omega_r \times \frac{1}{s}\right) + \frac{2\pi}{3}\right) \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \tag{26}$$

As the inverter is based on the abc modelled hence the iq^* should be transformed into abc refence frame. The mathematical representation of current in the abc reference frame can be described using (25). PWM of the inverter will be produced by comparing the actual three phase current and reference three phase current [55]. The PWM signal will be used to control the switching mechanism of the three-phase inverter [56]. Furthermore, the three-phase inverter will be producing voltage in abc reference frame [57]. As the BLDC is modelled on dq reference frame the output of Inverter will be transformed again into the dq reference frame using equation (26) [58]. Fig. 2 shows the block diagram of BLDC with the speed controller mechanism.

2.4. Procedure of Optimizing PI Parameters Using PO

In this paper the PI controller [59] parameters are optimized using PO under different operating condition, specifically different speed reference and load torques. To get the optimal value of PI controller, Puma Optimization is used as the optimization method. The procedures are described as follows:

1. Start the Puma Algorithm by initializing the puma population, number of prey, attack parameters, defence parameters, visibility and maximum speed.

2. Initialize the population. In this step the value of K_p and K_i are represented as the Puma.
3. Create a positional randomization of each puma in the search space.

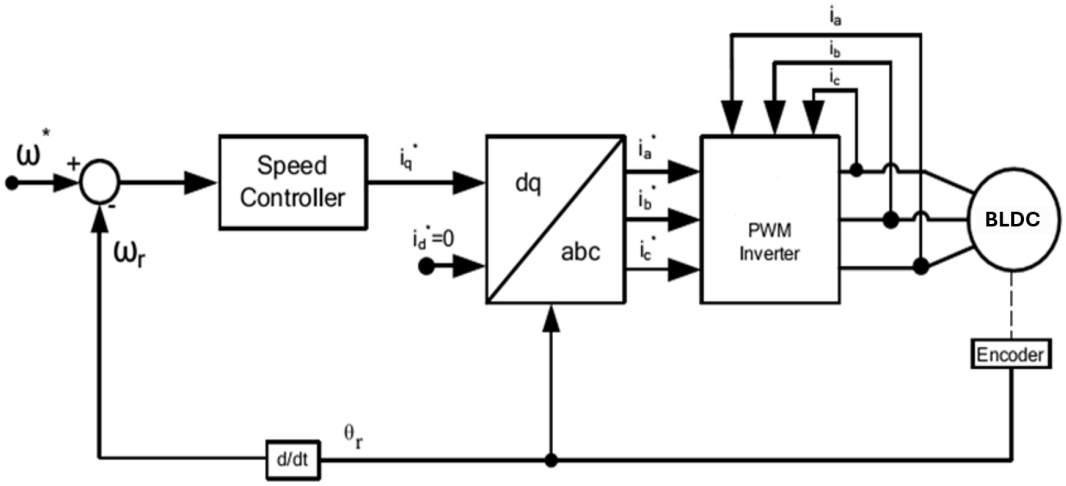

Fig. 2. Block diagram of BLDC with the speed controller

4. Evaluate fitness of each puma using (27) [60].

$$Objective = \int_0^{t_{simulation}} t|e(t)|\,dt \tag{27}$$

5. The movement of the puma uses the equation on exploration phase, stalking phase and attack phase.
6. If the distance to the prey is further than visual ability and the iteration is smaller than the maximum iteration divided by three, then use the exploration phase. If the person's visual ability is two times smaller than the prey's distance, then use the stalking phase. If the distance to the prey is smaller than the ability to see divided by two then use the attack phase. In this phase, save the best solution value and update the prey position to the best solution.
7. Evaluation of the objective function for new populations
8. If the new objective value is better than the previous value, accept the new position, update the global best puma position, and update the prey position. Otherwise, maintain the old position and perform a prey shift with a probability of defence ratio.
9. Update the control parameters of the algorithm
10. The next step is checking the stopping criteria. In this paper, max iteration is used as the stopping iteration.
11. Print the result (K_p and K_i).

Next, the optimized PI parameters are used to train the KELM model. This enables the PI controller to adapt to fluctuations in both load torque and reference speed.

2.5. Procedure of PI Controller Training Using-KELM

In this paper, the input features for the KELM model consist of the speed reference ω_{ref} and the load torque T_L. The output of the KELM model is the predicted PI controller parameters, namely $K_p$and K_i. The development of the adaptive PI controller is divided into two stages: the training phase and the testing phase. The workflow of the KELM–based adaptive PI controller during the training phase can be summarized as follows:

1. Collect the data of BLDC, inverter, and PI controller.
2. The training dataset for the KELM model is constructed using the BLDC system data. The input dataset consists of the speed reference ω_{ref} and the load torque T_L. The corresponding output dataset includes the PI controller parameters K_p and K_i, which serve as the predicted targets for the KELM model.
3. Configure the KELM network structure by defining the topology, including the number of neurons in the input and output layers as illustrated in Fig. 1. Select the kernel function according to (7).
4. Map the KELM input dataset into a high-dimensional feature space using the RBF kernel defined in (7), as described below:

$$\Omega(x,y) = exp\left(\frac{\|x-y\|^2}{2\sigma^2}\right)$$

5. Set the regularization coefficient $C$and the kernel parameter $\sigma$for the KELM model
6. Conduct the training process using (1)–(6) as described below:

$$y_M(x) = \sum_{i=1}^{M} \beta_i h_i(x)$$

$$Min: \|H\beta - T\|, \|\beta\|.$$

$$\beta = H^T\left(\frac{1}{C} + HH^T\right)^{-1} T$$

$$y(x) = h(x)H^T\left(\frac{1}{C} + HH^T\right)^{-1} T$$

$$O = HH^T : m_{ij} = h(x_i)h(x_j) = \Omega(x_i, x_j)$$

$$y(x) = [\Omega(x,x_1), \ldots, \Omega(x,x_M))]\left(\frac{1}{C} + O\right)^{-1} T$$

The forecasted parameters of PI are produced by using (6) as described below:

$$y(x) = [\Omega(x,x_1), \ldots, \Omega(x,x_M))]\left(\frac{1}{C} + O\right)^{-1} T.$$

7. Compute the accuracy of the PI parameters predicted by the KELM model. The mean square error (MSE) is used as the statistical performance indicator to evaluate the forecasting capability of KELM, as defined in (28).

$$MSE = \sum_{i=1}^{N} \frac{(\hat{y}_i - y_i)^2}{N} \quad (28)$$

where N is number of datasets; $\hat{y}_i$ is the predicted PI parameters (i.e. K_p and K_i); y_i is the real data record of PI parameters (i.e. K_p and K_i).

8. If the MSE reaches its minimum value, store the corresponding regularization coefficient $C$and kernel parameter σ obtained during the training phase for use in the testing phase, and record the predicted PI parameters K_p and K_i. Otherwise, return to Step 6 and repeat the process until the minimum MSE is achieved

After completing the training process, the next stage is the testing phase, where the KELM model is used to determine the appropriate PI controller parameters. The workflow of the KELM–based PI controller (KELM-PI) during the testing phase is outlined as follows:

1. Collect the data of BLDC, inverter and PI controller.
2. Prepare the input dataset of KELM for testing phase including the speed reference *Wreff* and load torque *TL* of the BLDC model.
3. Utilize the regularization coefficient C and kernel parameter σ of KELM for testing stage that has been obtained from training process.
4. Transform the input dataset of KELM to high-dimensional feature using kernel function in (7).

$$\Omega(x,y) = exp\left(\frac{\|x-y\|^2}{2\sigma^2}\right)$$

5. Compute the predicted parameters of PI including K_p and K_i using (6).

$$y(x) = [\Omega(x,x_1),\ldots,\Omega(x,x_M))]\left(\frac{1}{C}+O\right)^{-1}T$$

6. Print out the PI controller parameters (i.e K_p and K_i) and ITAE value.

3. Results and Discussion

3.1. Data Processing (Optimized PI Controller using PO)

The BLDC model data were collected and used to construct the training dataset, as summarized in Table 2. The input variables for the KELM model consist of the motor speed reference and the load torque. These variables represent the operating conditions under which the controller must perform, forming the basis for developing an adaptive control strategy.

Table 2. The datasets for training phase of KELM method and other techniques

Operating Condition		PI based on PO	
ω_r	T_L	K_p	K_i
1850	5.5	0.56	22.8
	3.5	0.40	24.5
1750	4	0.60	26
	3	0.30	17.5
1650	4.2	0.59	27.5
	2.2	0.58	22
1500	5	0.35	25.8
	4	0.62	32
1450	4.4	0.65	33
	1.1	0.45	18.5
1300	5.6	0.39	31.7
	1.6	0.37	17.2
1250	1	0.70	32
	2	0.50	25.9
1100	2	0.66	34
	3	0.66	38.5
1000	3.3	0.46	32.9
	1.5	0.69	38.3

The eighteen operating condition combinations in Table 2 were selected to provide coverage result of the proposed control operational range by varying load conditions and reference setpoints. This approach is to ensure the proposed method has good performance over the wide speed and load variation. For each input combination, the Puma Optimization (PO) algorithm is employed to obtain

the corresponding optimized PI controller parameters. The resulting optimal values of $K_p$and K_i are then used as the output targets for training the KELM model. This ensures that the KELM is trained to predict PI parameters that provide optimal performance across varying operating conditions.

3.2. Training Phase

In this section, the results of the training phase are presented. The primary objective is to identify the KELM parameters that yield the minimum MSE value, ensuring the highest prediction accuracy for the PI controller gains. To evaluate its effectiveness, the KELM method is compared against two other approaches: the Extreme Learning Machine (ELM) and the Artificial Neural Network (ANN).

The PI controller parameter obtained from the training process of KELM is depicted in Fig. 3. While Fig. 4 and Fig. 5 show the PI controller parameters obtained from ELM and ANN. In addition, training performance comparison between KELM, ANN, and ELM are shown in Table 3. From the MSE results it is noticeable that KELM provides better

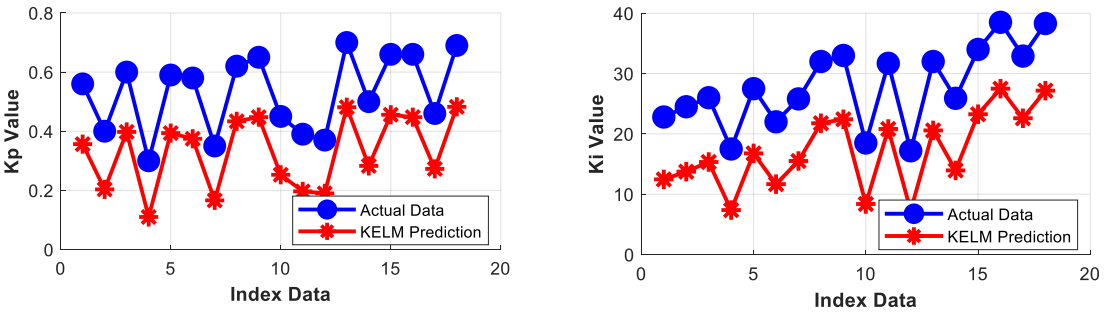

Fig. 3. K_p and K_i value based on KELM

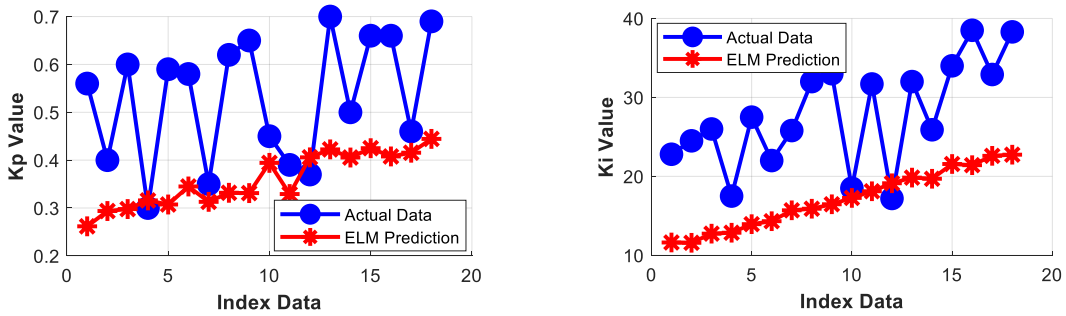

Fig. 4. K_p and $K_i$value based on ELM

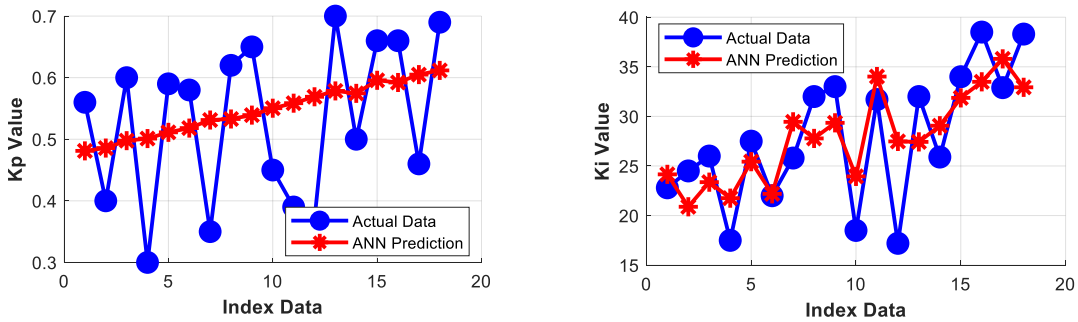

Fig. 5. K_p and $K_i$value based on ANN

Table 3. MSE comparison between KELM, ELM and ANN

Index	MSE
KELM	0.567
ELM	0.786
ANN	0.678

3.3. Testing Phase

To evaluate the effectiveness of the proposed KELM–based PI controller tuning method, the BLDC motor drive system was tested under reference speed variations and load torque disturbances. The optimal PI parameters obtained during the training phase for KELM, ELM, and ANN were applied during the testing stage. Fig. 6 illustrates the speed responses of the BLDC motor under the three control strategies, while the detailed performance indices extracted from these responses are summarized in Table 4.

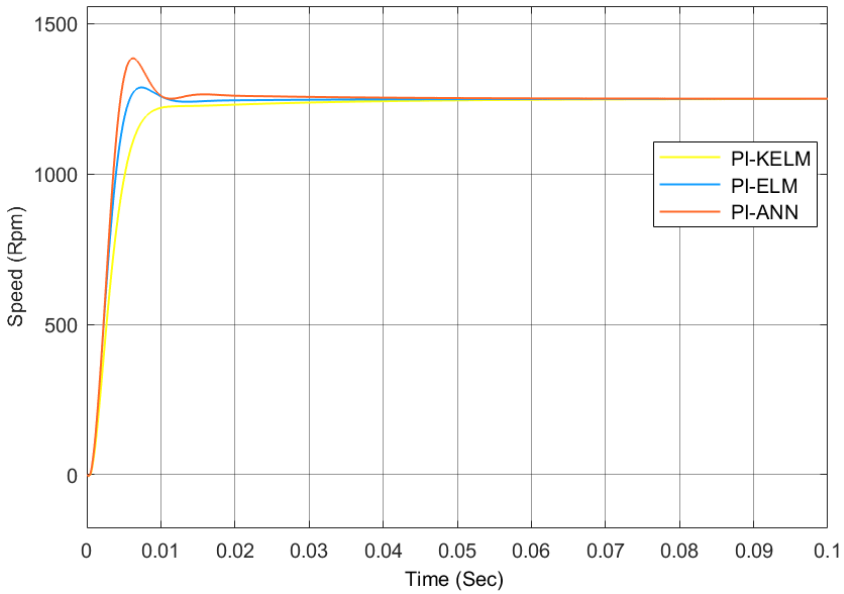

Fig. 6. Dynamic response comparison of BLDC speed

Table 4. Detailed features of Fig. 6

Methods	Rise Time (Sec)	Overshoot (%)	Settling Time (Sec)
ANN-PI	0.004	12	0,020
ELM-PI	0.005	5	0.018
KELM-PI	0.006	2	0.015

The comparative analysis shows that the PI-KELM controller provides the most favourable dynamic characteristics among the tested methods. As observed in Fig. 6, the PI-KELM configuration achieves a settling time to the reference speed while maintaining a smooth transient trajectory without any overshoot. This indicates superior damping behaviour and stability, which can be attributed to the enhanced nonlinear mapping capability of KELM in accurately determining the PI gains. In contrast, the PI-ELM controller exhibits a moderate amount of overshoot during the initial transient, while the PI-ANN controller demonstrates the largest overshoot and pronounced oscillatory behaviour, suggesting less effective generalization in identifying optimal parameters for rapid dynamic conditions.

In terms of settling time, the PI-KELM method outperforms the other two approaches, achieving the fastest transition to steady-state conditions. PI-ELM settles slightly later, whereas PI-ANN requires the longest time due to its overshoot and oscillations in the early transient phase. Despite all

controllers eventually reaching the same steady-state speed, the KELM-based controller shows markedly superior transient response quality, characterized by minimal oscillation, faster stabilization, and improved robustness against sudden changes in speed reference and load disturbances.

Fig. 7 and Fig. 8 shows the performance of proposed method during parameter variations. Fig. 7 shows the dynamic performance among three methods while motor inertia is set by +50%. As we can observed from that figure, PI KELM has faster settling time and less overshoot among three methods. In addition to that, when the stator resistance and inertia are varied by 30 %, the dynamic performance is shown in Fig. 8. As one can observe, the PI-KELM has better dynamic response compared to PI-ANN and PI-ELM. It is proved that even though the parameter has changed, the proposed controller compensates the variation.

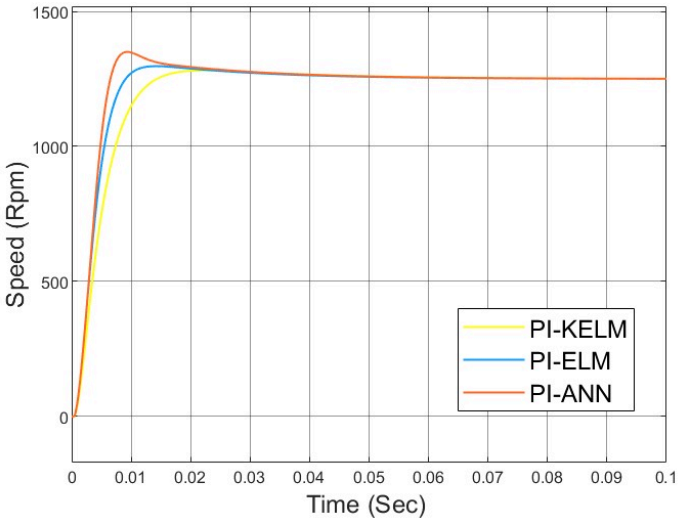

Fig. 7. Dynamic response when motor inertia is varied 50%

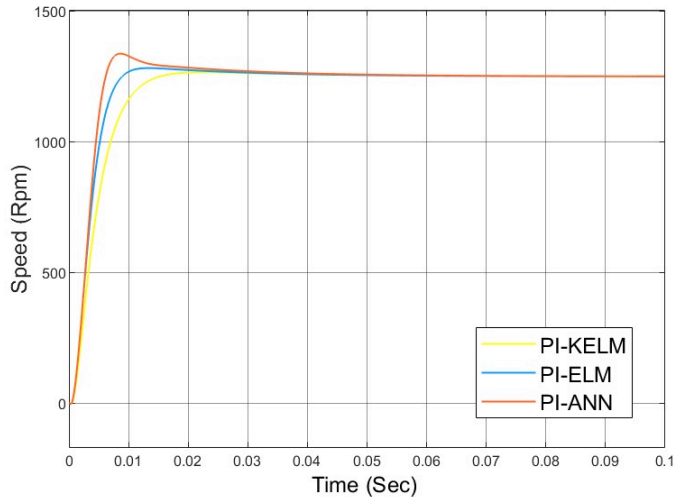

Fig. 8. Dynamic response when stator resistance and inertia are varied 30%

Overall, the results confirm that the integration of KELM for PI controller tuning significantly enhances the dynamic performance of the BLDC motor system. The improved rise time, zero overshoot, and reduced settling time collectively demonstrate the effectiveness of the proposed method, positioning PI-KELM as a more reliable and stable control strategy compared to PI-ELM and PI-ANN in handling dynamic operational conditions.

3.4. Controller Index Comparison

This section evaluates the error of the systems (reference minus feedback signal) using three widely recognized performance indices: Integral Time Absolute Error (ITAE), Integral Absolute Error (IAE), and Integral Square Error (ISE). These metrics provide a quantitative basis for assessing the quality of transient response under different control strategies. Fig. 9 presents a comparative analysis of the three scenarios: PI-ANN (blue chart), PI-ELM (brown chart), and the Proposed Method (green chart). Each chart reflects the cumulative error behavior over time, offering insight into the system's ability to dampen disturbances and restore frequency stability. For the ITAE indices, PI-KELM has better value with 0.005 compared to the other two methods, which are 0.012 and 0.022. For the IAE indices, PO-KELM has smallest value with 0.35 compared to the other two method, which are 0.38 for PI-ELM and 0.43 for PI-ANN. For the ISE index, PI-KELM has 0.30 value whether PI-ELM has 0.32 and PI-ANN has 0.38 value.

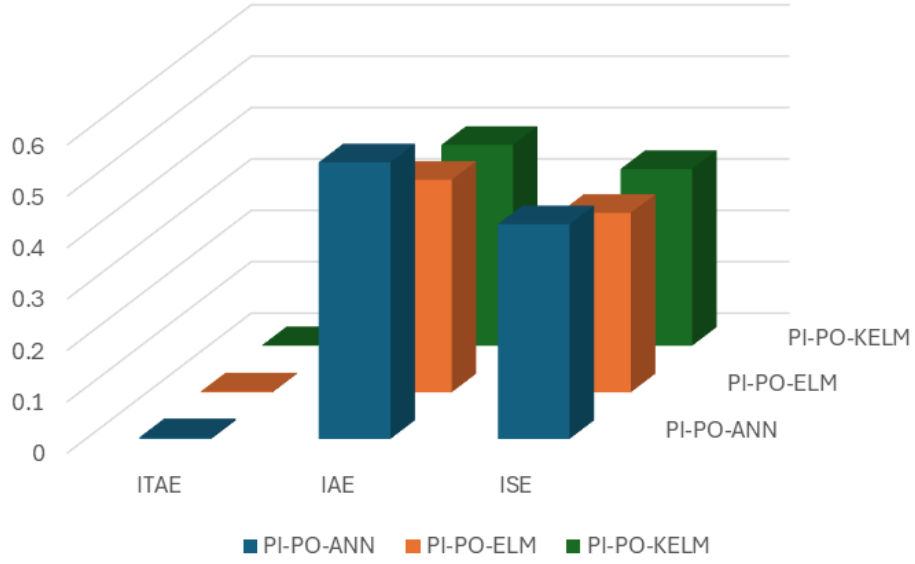

Fig. 9. Controller index comparison

The results clearly indicate that the Proposed Method yields the lowest values across all three performance indices, signifying superior transient performance. Lower ITAE, IAE, and ISE values imply reduced oscillations, faster settling time, and improved damping characteristics. In contrast, the system with PI-ANN abd PI-ELM configurations exhibit higher error magnitudes, suggesting less effective control. These findings confirm that the coordinated control strategy introduced in the Proposed Method enhances system stability and responsiveness, making it the most effective among the evaluated approaches.

4. Conclusion

This paper introduces an adaptive PI controller design using Puma Optimization (PO) and Kernel Extreme Learning Machine (KELM). The study demonstrates that KELM outperforms standard ELM and ANN in tuning PI parameters for BLDC motors, achieving the highest prediction accuracy with a low Mean Squared Error (MSE) of 0.567. Testing results indicates that the PI-KELM significantly improves stability compared to the baseline with 25% improvement in settling time, 83.33% in overshoot, and 50% slower in rise time. In addition, the proposed method has stable transient response under varying speed references and load disturbances. Future work includes implementing the proposed controller in real-time hardware such as in low-cost microcontroller or digital signal processor, exploring hybridization with other optimization algorithms, and expanding tests to diverse motor types and extreme load conditions. Investigating trade-offs between computational complexity and settling time is also recommended to further optimize performance.

Author Contribution: Conceptualization, Herlambang Setiadi and Muhammad Syahril Mubarok; methodology, Herlambang Setiadi and Muhammad Syahril Mubarok; software, Herlambang Setiadi and Muhammad Syahril Mubarok; validation, Herlambang Setiadi, Muhammad Syahril Mubarok, Ananta Adhi Wardana, Nur Vidia Laksmi; formal analysis, Herlambang Setiadi, Muhammad Syahril Mubarok, Ananta Adhi Wardana, Yoga Uta Nugraha, R. Thirumalaivasan, Nur Vidia Laksmi; investigation, Herlambang Setiadi, Muhammad Syahril Mubarok, Yoga Uta Nugraha, Habib Miftahudin Alfatah; resources, Herlambang Setiadi and Muhammad Syahril Mubarok; writing original draft preparation, Herlambang Setiadi, Muhammad Syahril Mubarok, Ananta Adhi Wardana, Yoga Uta Nugraha, R. Thirumalaivasan, Nur Vidia Laksmi; writing review and editing, Herlambang Setiadi, Muhammad Syahril Mubarok, Ananta Adhi Wardana, Yoga Uta Nugraha, R. Thirumalaivasan, Nur Vidia Laksmi; visualization, Herlambang Setiadi, Muhammad Syahril Mubarok, Nur Vidia Laksmi. All authors have read and agreed to the published version of the manuscript.

Funding: This research is funded by the Indonesian Endowment Fund for Education (LPDP) on behalf of the Indonesian Ministry of Higher Education, Science and Technology and managed under the EQUITY Program (Contract No. 4300/B3/DT.03.08/2025; No. 297/UN3/HK.07.00/2025) and International Research Collaboration by Subject Scheme No. 5549/B/UN3.LPPM/PT.01.03/2025.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix

Table 5. BLDC and inverter parameters

Index	Value
BLDC Parameter	
P	6
R_s *(Ohm)*	1.42
L_d *(H)*	0.0067
L_q *(H)*	0.0060
λ_m *(Wb)*	0.1556
B(Nm/(rad/s))	0.00039819
$J(kgm^2)$	0.00177
Inverter Parameter	
Frequency switching (KHz)	2
V (volt)	300
IGBT Ron (Ohm)	0.003

Table 6. ANN, ELM and KELM parameters

Index	Value
ANN	
Hidden Layer	8
MaxEpochs	500
Learning rate	0.01
Momentum	0.5
ELM	
Hidden Neurons	15
Activation function	sigmoid
Regularization	1

References

[1] D. Mohanraj, R. Aruldavid, R. Verma, K. Sathiyasekar, A. B. Barnawi, and B. Chokkalingam, "A review of BLDC motor: State of art, advanced control techniques, and applications," *IEEE Access*, vol. 10, pp. 54833–54869, 2022, https://doi.org/10.1109/ACCESS.2022.3175011.

[2] A. Darba, F. De Belie, P. D'haese, and J. A. Melkebeek, "Improved dynamic behavior in BLDC drives using model predictive speed and current control," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 2, pp. 728–740, Feb. 2016, https://doi.org/10.1109/TIE.2015.2477262.

[3] A. Kumar, M. V. Naik, R. Kumar, and Aman, "Design and analysis of BLDC motor speed control for electric vehicles powered by solar PV and grid supply," *Discover Electronics*, vol. 2, no. 1, p. 60, 2025, https://doi.org/10.1007/s44291-025-00097-4.

[4] N. Prabhu, R. Thirumalaivasan, and B. Ashok, "Critical review on torque ripple sources and mitigation control strategies of BLDC motors in electric vehicle applications," *IEEE Access*, vol. 11, pp. 115699–115739, 2023, https://doi.org/10.1109/ACCESS.2023.3324419.

[5] M. E. Sulistyo, D. D. Susilo, M. Nizam, and U. Ubaidillah, "A literature review: Bearing fault in BLDC motor based on vibration and thermal signals," *Journal of Electrical, Electronic, Information, and Communication Technology*, vol. 7, no. 1, pp. 10–15, 2025, https://doi.org/10.20961/jeeict.7.1.100165.

[6] A. Usman and A. Saxena, "Technical roadmaps of electric motor technology for next generation electric vehicles," *Machines*, vol. 13, no. 2, p. 156, 2025, https://doi.org/10.3390/machines13020156.

[7] M. Karthik, S. Usha, K. Venkateswaran, H. Panchal, M. Suresh, V. Priya, and K. K. Hinduja, "Evaluation of electromagnetic intrusion in brushless DC motor drive for electric vehicle applications with manifestation of mitigating the electromagnetic interference," *International Journal of Ambient Energy*, pp. 1–8, 2020, https://doi.org/10.1080/01430750.2020.1839546.

[8] P. Ramesh, A. Ranjeev, C. Santhakumar, J. Vinoth, and C. Bharatiraja, "Sensor-less field orientation control for brushless direct current motor controller for electric vehicles," *Materials Today: Proceedings*, vol. 65, pp. 277–284, 2022, https://doi.org/10.1016/j.matpr.2022.06.168.

[9] V. Kumarasamy, V. Karumanchetty Thottam Ramasamy, G. Chandrasekaran, G. Chinnaraj, P. Sivalingam, and N. S. Kumar, "A review of integer order PID and fractional order PID controllers using optimization techniques for speed control of brushless DC motor drive," *International Journal of System Assurance Engineering and Management*, vol. 14, no. 4, pp. 1139–1150, 2023, https://doi.org/10.1007/s13198-023-01952-x.

[10] C. Turkeri, S. Ekinci, D. Izci, and O. Kiselychnyk, "FGO-tuned 2-DOF PI controller for pressure regulation in centrifugal fans driven by induction motors using an experimentally validated model," *ISA Transactions*, vol. 169, pp. 436–446, 2026, https://doi.org/10.1016/j.isatra.2025.12.012.

[11] G. K. Alitasb, "Integer PI, fractional PI and fractional PI data trained ANFIS speed controllers for indirect field-oriented control of induction motor," *Heliyon*, vol. 10, no. 18, p. e37822, 2024, https://doi.org/10.1016/j.heliyon.2024.e37822.

[12] M. Subbarao, K. Dasari, S. S. Duvvuri, K. R. K. V. Prasad, B. K. Narendra, and V. B. Murali Krishna, "Design, control and performance comparison of PI and ANFIS controllers for BLDC motor driven electric vehicles," *Measurement: Sensors*, vol. 31, p. 101001, 2024, https://doi.org/10.1016/j.measen.2023.101001.

[13] A. R. Hasouna, S. A. Mahmoud, A. E. El-Sabbe, and D. S. M. Osheba, "Extendable multilevel inverter for renewable energy sources with fuzzy-PI controller for single phase induction motors," *Computers and Electrical Engineering*, vol. 123, p. 110195, 2025, https://doi.org/10.1016/j.compeleceng.2025.110195.

[14] M. F. Kethiri, O. Charrouf, A. Betka, I. E. Tibermacine, and C. Napoli, "Hybrid fuzzy-PSO based self-tuning fractional order PI controller for BLDC motors in electric vehicles: Comparative analysis and experimental validation," *Journal of Vibration and Control*, pp. 1–22, 2025, https://doi.org/10.1177/10775463251374112.

[15] S. S. A. Naqvi, H. Jamil, N. Iqbal, S. Khan, D.-I. Lee, Y. C. Park, and D. H. Kim, "Multi-objective optimization of PI controller for BLDC motor speed control and energy saving in electric vehicles: A constrained swarm-based approach," *Energy Reports*, vol. 12, pp. 402–417, 2024, https://doi.org/10.1016/j.egyr.2024.06.019.

[16] G. Murugaiyan, J. Gnanamalar, M. Narayanaperumal, and V. Muthuvel, "Red fox-based fractional order fuzzy PID controller for smart LED driver circuit," *Revue Roumaine des Sciences Techniques–Série Électrotechnique et Énergétique*, vol. 68, no. 4, pp. 395–400, 2023, https://doi.org/10.59277/RRST-EE.2023.4.12.

[17] N. Razmjooy, Z. Vahedi, V. V. Estrela, R. Padilha, and A. C. B. Monteiro, "Speed control of a DC motor using PID controller based on improved whale optimization algorithm," in *Metaheuristics and Optimization in Computer and Electrical Engineering*, N. Razmjooy, M. Ashourian, and Z. Foroozandeh, Eds. Cham, Switzerland: Springer, 2021, pp. 153–167, https://doi.org/10.1007/978-3-030-56689-0_8.

[18] V. Soni, G. Parmar, and M. Kumar, "A hybrid grey wolf optimisation and pattern search algorithm for automatic generation control of multi-area interconnected power systems," *International Journal of Advanced Intelligence Paradigms*, vol. 18, no. 3, pp. 265–293, 2021, https://doi.org/10.1504/IJAIP.2021.113323.

[19] A. A. Hossam-Eldin, E. Negm, M. Ragab, and K. M. AboRas, "A maiden robust FPIDD2 regulator for frequency-voltage enhancement in a hybrid interconnected power system using gradient-based optimizer," *Alexandria Engineering Journal*, vol. 65, pp. 103–118, 2023, https://doi.org/10.1016/j.aej.2022.10.029.

[20] E.-J. Liu, Y.-H. Huang, W.-L. Lin, C.-K. Wen, and C.-I. Lin, "Rapid, precise parameter optimization and performance prediction for multi-diode photovoltaic model using puma optimizer," *Energies*, vol. 18, no. 11, p. 2855, 2025, https://doi.org/10.3390/en18112855.

[21] B. Lei, H. Tang, Y. Su, Y. Ru, and S. Fei, "Fuzzy adaptive power optimization control of wind turbine with improved whale optimization algorithm and kernel extreme learning machine," *Expert Systems with Applications*, vol. 272, p. 126750, 2025, https://doi.org/10.1016/j.eswa.2025.126750.

[22] J. Xia, D. Yang, H. Zhou, Y. Chen, H. Zhang, T. Liu, A. A. Heidari, H. Chen, and Z. Pan, "Evolving kernel extreme learning machine for medical diagnosis via a disperse foraging sine cosine algorithm," *Computers in Biology and Medicine*, vol. 141, p. 105137, 2022, https://doi.org/10.1016/j.compbiomed.2021.105137.

[23] J. Peng, Y. Gao, L. Cai, M. Zhang, C. Sun, and H. Liu, "State of health estimation for lithium-ion batteries using electrochemical impedance spectroscopy and a multi-scale kernel extreme learning machine," *World Electric Vehicle Journal*, vol. 16, no. 4, p. 224, 2025, https://doi.org/10.3390/wevj16040224.

[24] Z. Gao and W. Yi, "Optimizing projectile aerodynamic parameter identification of kernel extreme learning machine based on improved Dung Beetle Optimizer algorithm," *Measurement*, vol. 239, p. 115473, 2025, https://doi.org/10.1016/j.measurement.2024.115473.

[25] F. Kang, X. Liu, J. Li, and H. Li, "Multi-parameter inverse analysis of concrete dams using kernel extreme learning machines-based response surface model," *Engineering Structures*, vol. 256, p. 113999, 2022, https://doi.org/10.1016/j.engstruct.2022.113999.

[26] X. Qin, L. Yuan, X. Dong, S. Zhang, and H. Shi, "Short term wind speed prediction based on CEESMDAN and improved seagull optimization kernel extreme learning machine," *Earth Science Informatics*, vol. 18, no. 1, p. 141, 2025, https://doi.org/10.1007/s12145-024-01560-8.

[27] L. Cui, G. Hu, and Y. Zhu, "Multi-strategy improved snow ablation optimizer: A case study of optimization of kernel extreme learning machine for flood prediction," *Artificial Intelligence Review*, vol. 58, no. 6, p. 181, 2025, https://doi.org/10.1007/s10462-025-11192-z.

[28] Q. Li, X. Zhang, H. Liang, A. Li, X. Ding, W. Huang, and C. Mechefske, "Variational mode feature construction-based improved kernel extreme learning machine for rotating machinery intelligent diagnosis," *IEEE Sensors Journal*, vol. 25, no. 5, pp. 8124–8133, 2025, https://doi.org/10.1109/JSEN.2025.3532798.

[29] Y. Wang, Y. Liu, W. Li, M. Deng, and K. Wang, "Online estimation method for extreme learning machine with kernels based on the multi-innovation theory and intelligent optimization strategy," *ISA Transactions*, vol. 156, pp. 142–153, 2025, https://doi.org/10.1016/j.isatra.2024.10.028.

[30] L. Gan, H. Wu, and Z. Zhong, "Fatigue life prediction considering mean stress effect based on random forests and kernel extreme learning machine," *International Journal of Fatigue*, vol. 158, p. 106761, 2022, https://doi.org/10.1016/j.ijfatigue.2022.106761.

[31] B. Liu, G. Chen, H.-C. Lin, W. Zhang, and J. Liu, "Prediction of IGBT junction temperature using improved cuckoo search-based extreme learning machine," *Microelectronics Reliability*, vol. 124, p. 114267, 2021, https://doi.org/10.1016/j.microrel.2021.114267.

[32] P. Pi and D. Lima, "Gray level co-occurrence matrix and extreme learning machine for Covid-19 diagnosis," *International Journal of Cognitive Computing in Engineering*, vol. 2, pp. 93–103, 2021, https://doi.org/10.1016/j.ijcce.2021.05.001.

[33] E. Dokur, N. Erdogan, and U. Yuzgec, "Swarm intelligence-based multi-layer kernel meta extreme learning machine for tidal current to power prediction," *Renewable Energy*, vol. 243, p. 122516, 2025, https://doi.org/10.1016/j.renene.2025.122516.

[34] H. Setiadi, R. Shah, M. R. Islam, D. A. Asfani, T. H. Nasution, M. Abdillah, P. Megantoro, and A. U. Krismanto, "An extreme learning machine based adaptive VISMA for stability enhancement of renewable rich power systems," *Electronics*, vol. 11, no. 2, p. 247, 2022, https://doi.org/10.3390/electronics11020247.

[35] S. Chaudhari, A. Thakare, and A. M. Anter, "PSOGSA: A parallel implementation model for data clustering using new hybrid swarm intelligence and improved machine learning technique," *Sustainable Computing: Informatics and Systems*, vol. 41, p. 100953, 2024, https://doi.org/10.1016/j.suscom.2023.100953.

[36] J. Cai, T. Chen, Y. Qi, S. Liu, and R. Chen, "Fibrosis and inflammatory activity diagnosis of chronic hepatitis C based on extreme learning machine," *Scientific Reports*, vol. 15, no. 1, p. 11, 2025, https://doi.org/10.1038/s41598-024-84695-4.

[37] B. Abdollahzadeh, N. Khodadadi, S. Barshandeh, P. Trojovský, F. S. Gharehchopogh, E.-S. M. El-Kenawy, L. Abualigah, and S. Mirjalili, "Puma optimizer (PO): A novel metaheuristic optimization algorithm and its application in machine learning," *Cluster Computing*, vol. 27, no. 4, pp. 5235–5283, 2024, https://doi.org/10.1007/s10586-023-04221-5.

[38] P. Maurya, P. Tiwari, and A. Pratap, "Puma optimizer technique for optimal planning of different types of distributed generation units in radial distribution network considering different load models," *Electrical Engineering*, vol. 107, no. 3, pp. 2777–2828, 2025, https://doi.org/10.1007/s00202-024-02631-1.

[39] H. Ashraf, S. O. Abdellatif, M. M. Elkholy, and A. A. El-Fergany, "Performance analysis of PEM fuel cells via Puma optimizer with the aid of practical verifications," *Neural Computing and Applications*, vol. 37, no. 29, pp. 24051–24074, 2025, https://doi.org/10.1007/s00521-025-11552-4.

[40] E.-J. Liu, R.-W. Chen, Q.-A. Wang, and W.-L. Lu, "Shuffled Puma Optimizer for parameter extraction and sensitivity analysis in photovoltaic models," *Energies*, vol. 18, no. 15, p. 4008, 2025, https://doi.org/10.3390/en18154008.

[41] T. K. Do and T. L. Duong, "Enhancing load frequency control in power systems using Puma Optimizer–Proportional Integral Derivative method," *Iranian Journal of Electrical and Electronic Engineering*, http://ijeee.iust.ac.ir/article-1-3430-en.pdf.

[42] E. S. M. El-Kenawy, A. M. Zaki, E. A. Mattar, A. A. Alhussan, D. S. Khafaga, M. M. Eid, and M. El-Seddek, Smart room occupancy detection using neural networks and the puma optimization algorithm. Scientific Reports, vol. 15, 2025, https://doi.org/10.1038/s41598-025-29938-8.

[43] M. Kmich, N. El Ghouate, A. Bencharqui, H. Karmouni, M. Sayyouri, S. S. Askar, and M. Abouhawwash, "Chaotic Puma Optimizer algorithm for controlling wheeled mobile robots," *Engineering Science and Technology, an International Journal*, vol. 63, p. 101982, 2025, https://doi.org/10.1016/j.jestch.2025.101982.

[44] W. Aribowo, L. Abualigah, D. Oliva, M. E. A. Elaziz, F. S. Gharehchopogh, H. A. Shehadeh, A. Sabo, and A. Prapanca, "Enhancing photovoltaic parameters based on modified Puma optimizer," *Bulletin of Electrical Engineering and Informatics*, vol. 14, no. 3, pp. 1667–1680, 2025, https://doi.org/10.11591/eei.v14i3.8977.

[45] Ö. Can, "The novel controller design tuned by Puma optimizer for load frequency control in two-area PV-thermal power system," *Sādhanā*, vol. 50, no. 3, pp. 1–11, 2025, https://doi.org/10.1007/s12046-025-02799-7.

[46] R. Abbassi, S. Saidi, H. Jerbi, L. Ladhar, and M. Omri, "Accurate parameters extraction of photovoltaic models using Lambert W-function collaborated with AI-based Puma optimization method," *Results in Engineering*, vol. 25, p. 104268, 2025, https://doi.org/10.1016/j.rineng.2025.104268.

[47] B. Ydir, I. Abazine, I. Choulli, D. Saadaoui, I. Antohe, G. Socol, D. B. Hmamou, E. H. Arjdal, M. Elyaqouti, and H. Lahlou, "Puma optimization algorithm for robust and accurate modeling of photovoltaic systems under diverse conditions," *Energy Conversion and Management: X*, vol. 28, p. 101320, 2025, https://doi.org/10.1016/j.ecmx.2025.101320.

[48] A. I. Abouseda, R. O. Doruk, and A. Amini, "Parameter identification and speed control of a small-scale BLDC motor: Experimental validation and real-time PI control with low-pass filtering," *Machines*, vol. 13, no. 8, p. 656, 2025, https://doi.org/10.3390/machines13080656.

[49] M. A. Khan, D.-Z. Baig, H. Ali, and F. R. Albogamy, "Optimized system identification (SI) of brushless DC (BLDC) motor using data-driven modeling methods," *Scientific Reports*, vol. 15, no. 1, p. 8497, 2025, https://doi.org/10.1038/s41598-025-93444-0.

[50] M. Dai, B. Meng, C. Tong, S. Li, and J. Ruan, "Analytical modeling and experimental validation of the churning torque based on partial oil immersion wet external rotor permanent magnet BLDC motor with an isolation sleeve," *Mechanical Systems and Signal Processing*, vol. 235, p. 112882, 2025, https://doi.org/10.1016/j.ymssp.2025.112882.

[51] D. Ziemiański, G. Chwalik-Pilszyk, and G. Dudzik, "Analysis of stator material influence on BLDC motor performance," *Materials*, vol. 18, no. 19, p. 4630, 2025, https://doi.org/10.3390/ma18194630.

[52] G. Taş and M. Özdamar, "Estimation of PID parameters of BLDC motor system by using machine learning methods," *Signal, Image and Video Processing*, vol. 19, no. 2, p. 140, 2025, https://doi.org/10.1007/s11760-024-03714-z.

[53] D. Kumpanya, S. Thaiparnat, and D. Puangdownreong, "Parameter identification of BLDC motor model via metaheuristic optimization techniques," *Procedia Manufacturing*, vol. 4, pp. 322–327, 2015, https://doi.org/10.1016/j.promfg.2015.11.047.

[54] Z. A. Al-Dabbagh and S. W. Shneen, "Design of a PID speed controller for BLDC motor with cascaded boost converter for high-efficiency industrial applications," *International Journal of Robotics and Control Systems*, vol. 5, no. 1, pp. 22–46, 2025, https://doi.org/10.31763/ijrcs.v5i1.1601.

[55] R. Baz, K. El Majdoub, F. Giri, and A. Taouni, "Self-tuning fuzzy PID speed controller for quarter electric vehicle driven by in-wheel BLDC motor and Pacejka's tire model," *IFAC-PapersOnLine*, vol. 55, no. 12, pp. 598–603, 2022, https://doi.org/10.1016/j.ifacol.2022.07.377.

[56] A. Gopalakrishnan and R. Thottungal, "Enhancing power efficiency in BLDC motor drives for drones using multiview learning with hybrid optimization algorithms," *Scientific Reports*, vol. 15, no. 1, p. 28021, 2025, https://doi.org/10.1038/s41598-025-13420-6.

[57] N. Hannan, S. K. Shib, A. Shufian, M. A. Islam, S. M. Islam Sharan, and A. D. Gupta, "Advanced regenerative braking system for EVs: Leveraging BLDC-supercapacitor technologies for optimized energy recovery, economic viability, and maintenance strategies," *Future Batteries*, vol. 7, p. 100103, 2025, https://doi.org/10.1016/j.fub.2025.100103.

[58] O. M. Neda, "Enhancing performance of BLDC: An improved fast terminal slide mode control for BLDC motor speed control," *Franklin Open*, vol. 14, p. 100510, 2026, https://doi.org/10.1016/j.fraope.2026.100510.

[59] M. Chattopadhyay, P. Roy, and S. Dutta, "Simulation modeling of BLDC motor drive and harmonic analysis of stator current, rotor speed and acceleration using discrete wavelet transform technique," *Procedia Technology*, vol. 4, pp. 666–670, 2012, https://doi.org/10.1016/j.protcy.2012.05.107.

[60] H. Setiadi, A. Swandaru, and T. A. Nugroho, "Design feedback controller of six pulse three phase rectifier based on differential evolution algorithm," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 22, no. 2, pp. 670–677, 2021, https://doi.org/10.11591/ijeecs.v22.i2.pp670-677.