International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 59, No. 7, July, 2026, pp. 1903-1917
Journal homepage: http://iieta.org/journals/jesa

Enhancing Speed Control of Separately Excited Direct Current Motors: A Novel Approach for Disturbance Rejection Using Interval Complex Neutrosophic Soft Sets

Omar Talal Mahmood, Laith A. Mohammed, Rakan Khalil Antar*

Electrical Engineering Techniques, Northern Technical University, Mosul 41001, Iraq

Corresponding Author Email: rakan.antar@ntu.edu.iq

https://doi.org/10.18280/jesa.590708

Received: 3 May 2026
Revised: 7 July 2026
Accepted: 24 July 2026
Available online: 31 July 2026

Keywords:
direct current motor, speed control, interval complex neutrosophic soft set, proportional-integral controller, dual loop control

ABSTRACT

DC motors have a wide range of applications, including electric vehicles, industrial applications, and automation systems. The speed control of these motors is a bottleneck in these applications. Researchers commonly use proportional-integral (PI), sliding mode, fuzzy, and neural controllers. However, DC motors experience several disturbances and uncertainties during operation, mainly due to variations in input voltage and fluctuations in load torque. Most controllers struggle to reject these disturbances and maintain smooth motor operation. Therefore, this study proposes a novel disturbance rejection approach to suppress these disturbances. The proposed controller is a novel fuzzy controller based on an interval complex neutrosophic soft set (ICNSS) algorithm. The novel ICNSS-based speed controller links a PI-like baseline regulator with a neutrosophic candidate-evaluation structure. At each sampling time, the controller measures the speed-tracking error and uses its discretized value to construct an uncertainty-sensitive decision space. This algorithm has not previously been applied to power electronic applications. To validate the proposed algorithm, this study compares its results with those of a two-loop well-tuned PI controller. The results for the new algorithm demonstrate better performance and robust disturbance rejection. MATLAB results demonstrated that the proposed ICNSS controller maintained the motor speed much closer to the reference value than the conventional PI controller under multiple disturbance scenarios, with smaller steady-state deviation and faster recovery in most tested cases.

1. INTRODUCTION

DC motors remain essential in present electromechanical systems for their simple structure, superior starting torque, broad speed range, and improved speed regulation techniques [1]. These aspects fulfill the requirements of DC motors' importance in industrial drives, with a usage percentage of 30.1% [2], educational laboratories [3], industrial 3D printing [4], and other adjustable-speed applications where precise and rapid speed regulation is required [5]. Because the practical performance of DC drives is strongly influenced by load changes, supply variations, and parameter uncertainty, a large body of recent work has continued to investigate both classical and intelligent speed-control strategies [6]. Although a wide range of controllers have been proposed to improve the disturbance-rejection capability of DC motor speed control or its drive systems, each method still exhibits practical limitations. Orta-Quintana et al. [7] used a robust two-stage controller to control the speed of a DC motor via a full-bridge buck inverter powered by a renewable energy source. Three disturbances were analyzed: input voltage, load resistance, and load torque. Yet the output speed still shows ripple and chatter due to the use of a sliding-mode controller, and not all the disturbances are considered. A proportional-integral (PI) controller was applied to the speed loop with online feedback disturbance estimation, as in studies [8, 9], to compensate for the torque and friction of the brushless DC motor. Although the approach reduced steady-state speed ripple, its effectiveness depended on motor parameter accuracy and showed less favorable transient behavior than the baseline proportional-integral and derivative (PID) loop.

A noise-reduction observer with anti-windup approaches was applied to a series DC motor in the study [10] to reject many types of uncertainties and disturbances. But this work relies on linear approximations and complex filter design, does not consider motor nonlinearities, and provides limited rejection over the operating range. In the study [11], a speed regulation scheme for a motor was proposed using an incremental port-Hamiltonian framework and a nonlinear disturbance observer. Yet, the disturbance observer is determined mainly by load torque, and the method is more complex than PI or fuzzy control [12]. An adaptive-fuzzy-PID controller enhanced by a disturbance observer for DC motor speed control was proposed in the study [13], where a fuzzy controller tuned the PID gains online, and the observer estimated the disturbance for compensation. However, adding the disturbance observer increased the overshoot, and disturbance validation was mainly limited to load-change conditions. Montoya-Acevedo et al. [14] proposed a continuous control-set model-predictive controller for a series-wound DC motor drive to reject the disturbances. Still, this method required a complex mathematical representation, and disturbance rejection was only applied to the load torque.

A nonlinear adaptive neural controller for a buck-converter-driven permanent magnet direct current (PMDC) motor was proposed in the study [15]. The proposed method remains mathematically complex, and its adaptive estimation process primarily focuses on load-torque uncertainty. A PID controller scaled by homogeneous state-dependent nonlinear functions is proposed to improve disturbance handling for a DC motor [16], but the method improves disturbance tolerance only through nonlinear robust feedback. Using a feedback linearizing controller with a disturbance observer, Prasanthi et al. [17] proposed a speed and torque-control system. The system is complex, and the contribution of the disturbance observer does not seem clear experimentally. The authors used a DC motor to validate data-driven state-feedback tracking with an integral action algorithm [18]. The formulation is limited to linear systems with measurable states and primarily addresses constant-matched disturbances.

Within the wider family of recently introduced uncertainty-handling approaches, neutrosophic and soft set-based models have attracted attention [19, 20]. They can represent three types of dynamic membership as truth, indeterminacy, and falsity simultaneously, while also preserving the parameterized structure of soft sets [21, 22]. Recent literature describes the complex neutrosophic soft set (CNSS) as an extension of neutrosophic soft modeling that incorporates phase information [23], and it further describes the interval complex neutrosophic soft set (ICNSS) as a generalization of CNSS in which interval-valued complex membership information is used better to represent uncertain and imprecise data [24]. Recent papers [23, 25, 26], and related authors have extended this line of work through ICNSS mappings, ICNSS relations, similarity measures, and fuzzy-parameterized ICNSS models [27]. These studies show that ICNSS-type models have mainly been explored in decision-making, pattern evaluation, and medical-diagnosis-type uncertainty problems, where they are used to compare alternatives under incomplete or ambiguous information [28].

From a control-engineering perspective, the proposed ICNSS speed controller is also related to several established lines of research on uncertainty-aware and nonlinear drive control [29]. Recent reviews and studies show that fuzzy and intelligent control remain important in mechatronics and electrical drives because they can handle nonlinear behavior, incomplete plant knowledge, and changing operating conditions more flexibly than fixed-gain linear controllers [30]. At the same time, robust nonlinear methods such as sliding-mode control and active disturbance rejection control (ADRC) are widely used in motor-drive applications because of their anti-disturbance capability, although practical trade-offs remain, including chattering, tuning burden, or the difficulty of simultaneously achieving fast response, low overshoot, and strong disturbance rejection. In parallel, model-predictive control [31] has become a major engineering framework for electrical drives, and in finite-control-set forms it evaluates a finite number of candidate control actions or switching states at each update using a cost function; this makes candidate-action selection itself a well-established engineering idea rather than a purely mathematical abstraction [32]. Therefore, the engineering motivation of the present work is to position ICNSS not only as a set-theoretic uncertainty model, but as an uncertainty-aware candidate-evaluation layer for the outer speed loop of a DC drive, where control action must be selected under disturbance, parameter variation, and nonlinear operating conditions.

Motivated by those strengths, this paper explores the use of ICNSS for speed control of a two-loop separately excited DC-motor drive, in which the outer loop generates the armature current reference using the ICNSS approach, and the inner loop regulates the armature current via the chopper converter. In such a cascaded structure, the inner current loop provides fast electrical regulation. In contrast, the outer speed loop handles slower mechanical dynamics and disturbances such as input-voltage variations, load-torque fluctuations, reference-speed variations, and field-condition changes. The proposed ICNSS-based speed controller is constructed as an uncertainty-sensitive decision mechanism that combines a PI-like baseline with neutrosophic candidate evaluation to produce a more suitable armature-current reference value in the presence of operating disturbances and uncertainties. In this manner, this work aims to connect the developed literature on DC-motor cascade control with the emerging ICNSS decision framework and to assess whether the latter can improve disturbance-rejection capability in motor-drive applications. The rest of this paper presents the methodology used to design the outer-loop ICNSS algorithm, followed by the results and discussion sections, and finally, the conclusion of this work.

2. METHODOLOGY

The proposed ICNSS-based speed controller combines a PI-like baseline regulator with a neutrosophic candidate-evaluation mechanism. At each outer-loop sampling instant, the speed-tracking error and its discrete variation are used to construct an uncertainty-sensitive decision space. A set of candidate armature-current corrections was generated according to the error magnitude. For each candidate, neutrosophic truth, indeterminacy, and falsity measures were formed using both directional agreement and an uncertainty/activity index. These amplitudes are then enriched by a phase surrogate derived from the error dynamics, yielding a complex-neutrosophic-like representation. A weighted score function is finally used to select the most suitable current correction, while rate limiting and output smoothing ensure robust, non-oscillatory reference-current generation. For fair benchmarking, the conventional PI controller used in the speed loop was tuned using the MATLAB Parameters Tuning Optimization Toolbox, specifically the PID Controller Design in the Live Editor. The obtained gains were then implemented on the same DC motor-chopper plant used for the proposed ICNSS controller, with the same inner current-loop structure, sampling times, and practical control constraints, in order to ensure a transparent and consistent comparison.

2.1 Control goal of the algorithm

MATLAB code introduces the outer-loop speed controller. Its role is to generate the reference armature current ($I_{\boldsymbol{a},\mathrm{ref}}$), from the speed tracking error:

$$e(k) = \omega_{\mathrm{ref}}(k) - \omega(k) \tag{1}$$

where, $\omega_{\mathrm{ref}}(k)$ and $\omega(k)$ are reference speed and measured motor speed in rpm, respectively. This reference current is fed to the inner current-loop PI controller, which generates the PWM signals for the chopper as described in Figure 1.

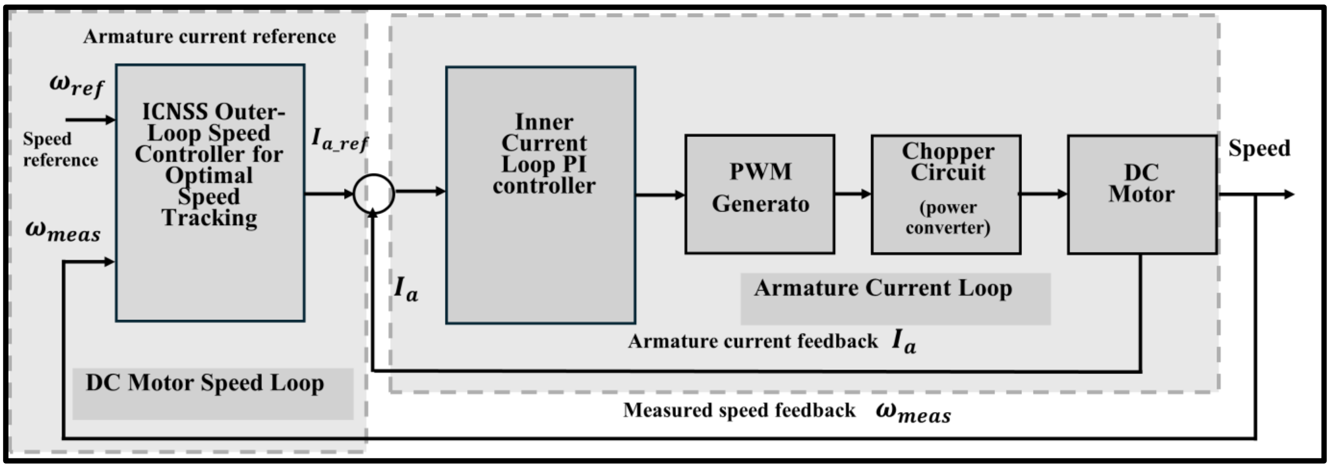

Figure 1. Block diagram of the proposed interval complex neutrosophic soft set (ICNSS)-based speed controller with the proportional-integral (PI) inner current loop

2.2 Mathematical representation of the DC motor outer-loop problem

For a separately excited DC motor, the main dynamic equations are [9]:

Electrical armature equation

$$V_a(t) = R_a i_a(t) + L_a \frac{di_a(t)}{dt} + e_b(t) \tag{2}$$

with back-emf:

$$e_b(t) = K_e \omega(t) \tag{3}$$

Electromagnetic torque

$$T_e(t) = K_t i_a(t) \tag{4}$$

Mechanical dynamics

$$J \frac{d\omega(t)}{dt} + B\omega(t) = T_e(t) - T_L(t) \tag{5}$$

where, R_a and L_a are armature resistance and inductance, J is moment of inertia, B is viscous friction, T_L is load torque disturbance, and i_a is armature current.

$$T_e(t) = K_t i_a(t) \tag{6}$$

The aim of the outer loop is to choose a suitable current reference $I_{a,\text{ref}}$ such that the actual speed ω tracks ω_{ref} despite input-voltage variation [33], load torque variation, field-voltage variation, parameter uncertainty, and reference-speed changes. The controller is not updated at every simulation step. It is updated every outer-loop sample time as $T_{\text{s,model}}=10^{-5}$ s and $T_{\text{s,speed}}=10^{-3}$ s. Thus, the number of model steps per controller update is:

$$N_{\text{update}} = \text{round}\left(\frac{T_{s,\text{speed}}}{T_{s,\text{model}}}\right)$$

So, the controller performs as a discrete-time supervisory controller for the speed loop. The steps below outline the design of the ICNSS controller for the speed loop in this proposed system.

2.3 Speed error and discrete-time error variation

At each controller instant k, $e(k)$ in Eq. (1) is updated in the discrete change as:

$$\Delta e(k) = e(k) - e(k-1) \tag{7}$$

This pair $(e, \Delta e)$ is the main information used by the ICNSS decision logic.

- $e(k)$ tells whether the speed is below or above the reference.
- $\Delta e(k)$ tells whether the speed error is improving or worsening.

2.4 Deadband mechanism

The code in the speed loop for the ICNSS algorithm first checks whether the system is already very close to the equilibrium point:

$$|e(k)| < 0.01, |\Delta e(k)| < 1.0 \tag{8}$$

If it is true, then the value of the generated reference armature current will be:

$$I_{a,\text{ref}}(k) = I_{a,\text{ref}}(k-1) \tag{9}$$

The controller will set its output near the steady state to avoid unnecessary oscillation and chattering.

2.5 Bias adaptation term

The algorithm utilizes an adaptive current bias I_{bias} approach which is updated as:

$$I_{\text{bias}}(k) = I_{\text{bias}}(k-1) + K_b e(k) T_{s,\text{speed}} \tag{10}$$

With $K_b = 0.02$, and bounded by $8 \le I_{\text{bias}}(k) \le 13$. This part will act as a slow adaptive feedforward baseline. It trains the approximate current level required to maintain the operating point under varying conditions. It helps reject uncertainties because when the motor needs more current due to a disturbance, this bias gradually shifts the control signal $(I_{a,\text{ref}}(k))$ upward. So, the controller does not depend solely on the instantaneous error; it also maintains a memory of the required current level.

2.6 Proportional-integral-like baseline component

The code builds a baseline current command as:

$$I_{a,\text{ base}}(k) = I_{\text{bias}}(k) + K_p e(k) + K_i e_{\text{int}}(k) \tag{11}$$

where,

$$e_{\text{int}}(k) = e_{\text{int}}(k-1) + e(k)T_{s,\text{ speed}}$$

and

$$K_p = 0.06, K_i = 0.15$$

with anti-windup bounds of $-500 \leq e_{\text{int}}(k) \leq 500$. This equation is the deterministic baseline regulator; each gain has the following function:

- $K_p e(k)$: immediate response to speed error
- $K_i e_{\text{int}}(k)$: slow removal of steady-state error
- $I_{\text{bias}}(k)$: operating-point adaptation

So, before ICNSS is even applied, the controller already contains a smooth PI-like core.

2.7 Candidate action set ($\mathcal{A}$)

The action of the novel ICNSS will start in this stage of the code; the ICNSS part does not directly compute a continuous control law from scratch [25]. Instead, it chooses among a finite set of candidate corrections:

$$\Delta I_a \in \mathcal{A}(k) \tag{12}$$

This step is the core of the algorithm regarding its disturbance rejection property [23]. Depending on the error magnitude, the algorithm has these three cases:

- If a large error

$$\begin{gathered} |e(k)| > 150 \\ \mathcal{A}(k) = \{-0.30, -0.15, 0, 0.15, 0.30\} \end{gathered} \tag{13}$$

- If medium error

$$\begin{gathered} 40 < |e(k)| \leq 150 \\ \mathcal{A}(k) = \{-0.12, -0.06, 0, 0.06, 0.12\} \end{gathered} \tag{14}$$

- If a small error

$$\begin{gathered} |e(k)| \leq 40 \\ \mathcal{A}(k) = \{-0.03, -0.015, 0, 0.015, 0.03\} \end{gathered} \tag{15}$$

Under heavy disturbances, the ICNSS algorithm is allowed to produce a large correction, but near the equilibrium, it only applies a small corrective action [22].

For uncertainty/activity magnitude indication, a scalar value defined $r(k) \in [0,1]$. For a small value of $r(k) \in [0,1]$, the speed dynamics are calm and low uncertainty, while for a large value of $r(k)$, the speed error is changing rapidly, indicating disturbance, transient, or uncertainty. The value of this indicator defined as:

$$r(k) = \min\left(1, \frac{|\Delta e(k)|}{150}\right) \tag{16}$$

So $r(k)$ is the algorithm's internal evaluation of how "uncertain" or "agitated" the system is? This behavior is another core way that the ICNSS algorithm responds to disturbances or uncertainties; the 150 value is used as an estimated value of uncertainty.

2.8 Phase magnitude of the membership functions

The phase of the membership is another property of the proposed ICNSS algorithm; this is the complex/phase component of the ICNSS idea. Instead of using only amplitude information $(e, \Delta e)$, the algorithm also encodes the dynamic direction/state of the error in a phase-like quantity. The code represents this as [22, 23]:

$$\phi(k) = \text{atan2}(\Delta e(k), e(k)) \tag{17}$$

If $\phi(k) < 0$, then:

$$\phi(k) \text{will be } \phi(k) + 2\pi$$

and then the upper phase bound is:

$$\phi_U(k) = 1.02\phi(k)$$

That is why the ICNSS approach is not just a classical fuzzy or PI controller, since it is applied to both amplitude-type and dynamic phase-type data.

Positive and negative control current term $A^{(j)}$ is applied in this algorithm to decide the direction of the correction signal depending on the error value $e(k)$.

$$A^{(j)}(k) = \tfrac{1}{2}\left(1 + \text{signnz}(e(k))\text{signnz}\left(\Delta I_a^{(j)}\right)\right) \tag{18}$$

Given that:

$$\text{signnz}(x) = \begin{cases} 1, & x > 0 \\ -1, & x < 0 \\ 0, & x = 0 \end{cases}$$

If the error is larger than zero ($e(k) > 0$), i.e., the speed is below reference, a positive current correction is preferred. If the error is less than zero ($e(k) < 0$), i.e., the speed is above reference, a negative correction is preferred.

2.9 ICNSS membership design for the proposed speed loop controller

The design procedure for the ICNSS membership is defined using three neutrosophic components for each candidate action j, these components are:

1) Truth amplitude:

$$T_{\text{amp}}^{(j)}(k) = \text{sat01}\left(0.82A^{(j)}(k) + 0.18(1 - r(k))\right) \tag{19}$$

2) Indeterminacy amplitude

$$I_{\text{amp}}^{(j)}(k) = \text{sat01}(0.12 + 0.75r(k)) \tag{20}$$

3) Falsity amplitude

$$F_{\text{amp}}^{(j)}(k) = \text{sat01}\left(0.60\left(1 - A^{(j)}(k)\right) + 0.40r(k)\right) \tag{21}$$

where, $\mathrm{sat01}(x) = \min(1, \max(0, x))$. The truth amplitude T is high when the action direction is correct, and the system is not highly uncertain. While the indeterminacy I is high when the activity/uncertainty index r is high. Finally, falsity F is high when the candidate action direction is wrong, or the uncertainty is high. This membership design is the method's neutrosophic core action.

The algorithm then combines the amplitude and phase using a suitable weight to form a complex-neutrosophic aggregation, such as:

$$w_1 = 0.90, w_2 = 0.10 \tag{22}$$

to get:

$$\begin{aligned}\tilde{T}^{(j)}(k) &= w_1 T_{\text{amp}}^{(j)}(k) + w_2 s\phi_U(k)\\ \tilde{I}^{(j)}(k) &= w_1 I_{\text{amp}}^{(j)}(k) + w_2 s\phi_U(k)\\ \tilde{F}^{(j)}(k) &= w_1 F_{\text{amp}}^{(j)}(k) + w_2 s\phi_U(k)\end{aligned} \tag{23}$$

This is the ICNSS algorithm decision representation in the code, so at each candidate action, the decision is not only based on static truth/indeterminacy/falsity, but also on the dynamic-phase condition of the speed error. This section of the algorithm provides the controller with additional capability to reject the disturbances under different operating conditions.

2.10 Candidate scoring rule for the control signal $I_{a,\text{ref}}(k)$

For each candidate j, the code computes [34, 35]:

$$\begin{aligned}\text{Score}^{(j)}(k) = \tilde{T}^{(j)}(k) - \lambda_I \tilde{I}^{(j)}(k) - \tilde{F}^{(j)}(k) -\\ \mu_{\text{Jump}}\left|I_{a,\text{cand}}^{(j)}(k) - I_{a,\text{prev}}(k)\right| - \mu_{\text{Mag}}\left|\Delta I_a^{(j)}\right|\end{aligned} \tag{24}$$

with:

$$\lambda_I = 0.18, \mu_{\text{Jump}} = 0.30, \mu_{\text{Mag}} = 0.15$$

And

$$I_{a,\text{cand}}^{(j)}(k) = I_{a,\text{base}}(k) + \Delta I_a^{(j)}$$

Then the optimal action is selected as:

$$j^\star = \arg\max_j \text{Score}^{(j)}(k) \tag{25}$$

and the command becomes

$$I_{a,\text{cand}}(k) = I_{a,\text{base}}(k) + \Delta I_a^{(j^*)} \tag{26}$$

This scoring tool is the actual decision engine that produces an adaptive, smooth controller action. It prefers actions that increase the truth ($\tilde{T}^{(j)}$), reduce falsity $\tilde{F}^{(j)}$, reduce indeterminacy $\tilde{I}^{(j)}$, and avoid unnecessary large corrections and abrupt jumps. The above control parameters and gains are selected to achieve optimal system performance.

To guarantee physically meaningful current reference and avoid excessive command during large disturbances, an output saturation part was added to the code of the algorithm, which will bound the control signal as:

$$7 \le I_{a,\text{cand}}(k) \le 17 \tag{27}$$

A rate limiter is another disturbance rejection added to the algorithm to prevent the outer-loop controller from commanding extremely abrupt current jumps, which could excite the inner loop and worsen overshoot. It is applied as:

$$\Delta I(k) = I_{a,\text{cand}}(k) - I_{a,\text{prev}}(k) \tag{28}$$

with maximum allowed step:

$$|\Delta I(k)| \le 0.15$$

More precisely:

$$I_{a,\text{cand}}(k) = \begin{cases} I_{a,\text{prev}}(k) + 0.15, & \Delta I(k) > 0.15\\ I_{a,\text{prev}}(k) - 0.15, & \Delta I(k) < -0.15\\ I_{a,\text{cand}}(k), & \text{otherwise}\end{cases} \tag{29}$$

2.11 Final closed-form structure of the algorithm

The full controller is summarized as:

$$\begin{aligned}I_{a,\text{ref}}(k) = \mathcal{S}\Big(\mathcal{R}\Big(I_{\text{bias}}(k) + K_p e(k) + K_i e_{\text{int}}(k) +\\ \Delta I_a^{(j^*)}(k)\Big)\Big)\end{aligned} \tag{30}$$

where, $\mathcal{R}(\cdot)$: rate-limiting operator, $\mathcal{S}(\cdot)$: smoothing and saturation operator, and $j^\star$: optimal candidate selected by the ICNSS scoring judge. The final flowchart for the proposed ICNSS controller of the speed loop is shown in Figure 2.

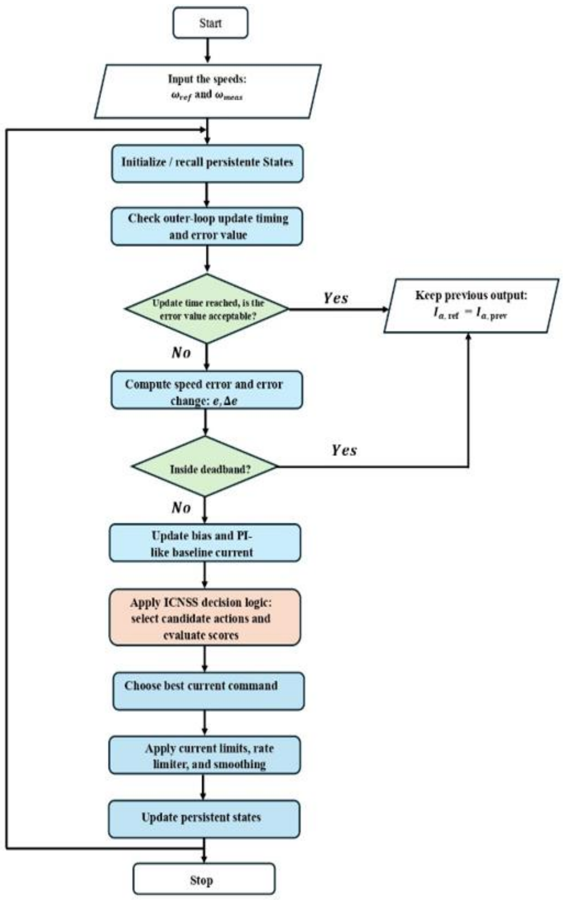

Figure 2. Flowchart of the proposed outer-loop proportional–integral and interval-valued complex neutrosophic soft set controller

2.12 Final decision comparison between the interval complex neutrosophic soft set controller and the proportional-integral controller

Table 1 summarizes the main differences between the conventional PI controller, the conventional fuzzy controller, and the proposed ICNSS-based controller. Unlike the PI controller, which generates the control signal directly from fixed gains, and unlike the fuzzy controller, which depends on predefined linguistic rules and membership functions, the proposed ICNSS controller evaluates multiple candidate control actions through a neutrosophic decision layer. This layer combines truth, indeterminacy, and falsity measures with directional logic, uncertainty evaluation, and a phase-related term, and then selects the final action using a weighted scoring rule. Therefore, the novelty of the proposed method lies in the ICNSS-based decision mechanism rather than in the PI-like baseline term alone.

Table 1. Comparison between proportional-integral (PI), fuzzy, and proposed interval complex neutrosophic soft sets (ICNSS)-based controllers

Feature	Conventional PI Controller	Conventional Fuzzy Controller	Proposed ICNSS-Based Controller
Basic control principle	Fixed PI action based on error and accumulated error	Rule-based inference using linguistic variables and membership functions	Candidate-action evaluation using truth, indeterminacy, falsity, phase term, and weighted scoring
Main inputs	Error e, integral of error $\int edt$	Error e, change of error Δe	Error e, change of error Δe, phase surrogate, uncertainty/activity index
Control output generation	Direct continuous output from fixed gains	Output selected through fuzzy rules and defuzzification	Output selected from candidate actions through ICNSS decision scoring
Adaptation to disturbance level	Limited; depends on fixed gain tuning	Moderate; depends on rule base and membership-function design	High; action magnitude changes according to error size and uncertainty level
Handling of uncertainty	Weak under parameter variation and nonlinear disturbances unless retuned	Better than PI, but depends strongly on expert rule design	Explicitly accounts for uncertainty through indeterminacy and falsity measures
Decision mechanism	Deterministic linear combination	Rule-based inference	Multi-candidate neutrosophic evaluation with directional consistency and phase-aware scoring
Need for expert rules	No	Yes	No conventional fuzzy rule base required
Use of phase/dynamic condition	No	Usually limited	Yes; phase surrogate is included in the decision process
Direction-awareness of control action	Implicit through sign of error	Implicit through rule base	Explicitly checked through directional agreement function
Behavior near equilibrium	May oscillate if gains are aggressive	Can be smooth if well designed	Small candidate actions and deadband reduce unnecessary corrections
Behavior under large disturbances	Can produce overshoot or poor rejection if gains are not retuned	Improved, but depends on rule tuning	Larger corrective actions are automatically enabled under strong disturbances
Interpretability of controller structure	High and simple	Moderate; depends on rule base complexity	Moderate; more structured than fuzzy, but more complex than PI
Computational complexity	Low	Medium	Medium
Main limitation	Weak robustness to nonlinearities and disturbances	Strong dependence on membership/rule design	More parameters to tune; startup overshoot may appear if not optimized

2.13 Summary of the control parameters used by the interval complex neutrosophic soft set controller and the proportional-integral controller

To improve reproducibility, Table 2 summarizes the main implementation parameters of both the conventional PI controller and the proposed ICNSS-based controller, including gains, candidate-action magnitudes, deadband conditions, integral bounds, output saturation limits, rate-limiter settings, and sampling parameters.

2.14 Electrical, mechanical parameters for the DC motor and the chopper components

The electrical and mechanical parameters of the separately excited DC motor and the chopper-fed drive used in this study. The motor armature parameters were set as $R_a = 5.79\ \Omega$ and $L_a = 0.06$ H, while the field parameters were $R_f = 246.7\ \Omega$ and $L_f = 52$ H. The field-armature mutual inductance was $L_{af} = 1.6$ H. The mechanical parameters were the total inertia $J = 0.012\ \text{kg}\cdot\text{m}^2$, viscous friction coefficient $B_m = 0.0204\ \text{N}\cdot\text{m}\cdot\text{s}$, and Coulomb friction torque $T_f = 0\ \text{N}\cdot\text{m}$. The chopper input voltage was 220 V, the nominal field voltage was 180 V, the reference speed was 1250 rpm, and the nominal load torque was $10\ \text{N}\cdot\text{m}$. The PWM switching frequency was 20 kHz, and the chopper filter inductance was 10×10^{-3} H. These parameters were used as the nominal operating values before introducing disturbances in input voltage, load torque, field voltage, and speed reference.

2.15 Qualitative behaviour matrices

For every disturbance scenario, the performance of the conventional PI controller and the proposed ICNSS controller is now evaluated using maximum speed deviation, recovery time, integral absolute error (IAE), integral squared error (ISE), overshoot, and steady-state error. In addition, the reference-speed variation case is reported separately as a tracking performance; below is the mathematical representation of each metric:

- Maximum speed deviation

$$\Delta\omega_{\max} = \max|\omega(t) - \omega_{\text{ref}}(t)| \quad (31)$$

Unit: rpm

- Recovery time

Time required after disturbance application until the speed re-enters and remains inside the selected tolerance band around the reference.

Table 2. Summary of the main implementation parameters of the conventional proportional-integral (PI) and interval complex neutrosophic soft set (ICNSS) controllers.

Controller Block	Parameter	Symbol/Condition	Value in Manuscript	Unit/Note
Conventional PI controller	Outer-loop speed proportional gain	$K_{p,\omega}$	1.5	unitless
Conventional PI controller	Outer-loop speed integral gain	$K_{i,\omega}$	20	unitless
Conventional PI controller	Current-loop proportional gain	$K_{p,i}$	3	unitless
Conventional PI controller	Current-loop integral gain	$K_{i,i}$	10	unitless
Conventional PI controller	Speed-loop sampling time	$T_{s,\text{ speed}}$	10^{-3}	s
Conventional PI controller	Model/simulation step	$T_{s,\text{ model}}$	10^{-5}	s
ICNSS controller	Deadband on error	$\lvert e(k)\rvert$	< 0.01	rpm
ICNSS controller	Deadband on error variation	$\lvert\Delta e(k)\rvert$	< 1.0	rpm
ICNSS controller	Bias adaptation gain	K_b	0.02	dimensionless
ICNSS controller	Bias current lower bound	$I_{\text{bias,min}}$	8	A
ICNSS controller	Bias current upper bound	$I_{\text{bias,max}}$	13	A
ICNSS controller	PI-like baseline proportional gain	K_p	0.06	unitless
ICNSS controller	PI-like baseline integral gain	K_i	0.15	unitless
ICNSS controller	Integral lower bound	$e_{\text{int, min}}$	-500	rpm s
ICNSS controller	Integral upper bound	$e_{\text{int,max}}$	500	rpm s
ICNSS controller	Large-error candidate action set	$\lvert e(k)\rvert$	> 150	rpm s
ICNSS controller	Medium-error candidate action set	$\lvert e(k)\rvert$	$40 < \lvert e(k)\rvert \leq 150$	rpm s
ICNSS controller	Small-error candidate action set	$\lvert e(k)\rvert$	$\lvert e(k)\rvert \leq 40$	rpm s
ICNSS controller	Uncertainty/activity index	$r(k)$	$\min\left(1, \frac{\lvert\Delta e(k)\rvert}{150}\right)$	dimensionless
ICNSS controller	Phase weight	w_1	0.90	dimensionless
ICNSS controller	Amplitude weight	w_2	0.10	dimensionless
ICNSS controller	Indeterminacy penalty	λ_I	0.18	dimensionless
ICNSS controller	Jump penalty	μ_{Jump}	0.30	dimensionless
ICNSS controller	Magnitude penalty	μ_{Mag}	0.15	dimensionless
ICNSS controller	Output lower limit	$I_{a,\text{ cand, min}}$	7	A
ICNSS controller	Output upper limit	$I_{a,\text{ cand, max}}$	17	A
ICNSS controller	Maximum rate-limiter step	$\lvert\Delta I(k)\rvert$	≤ 0.15	A
ICNSS controller	Speed-loop sampling time	$T_{s,\text{ speed}}$	10^{-3}	s
ICNSS controller	Model/simulation step	$T_{s,\text{ model}}$	10^{-5}	s
ICNSS controller	Update ratio	$N_{\text{update}} = \text{round}\left(T_{s,\text{ speed}}/T_{s,\text{ model}}\right)$	100	samples

Unit: s

- Integral of absolute error (IAE)

$$\text{IAE} = \int_{t_d}^{t_f} |\omega_{\text{ref}}(t) - \omega(t)| dt \quad (32)$$

Unit: rpm•s

- Integral of squared error (ISE)

$$\text{ISE} = \int_{t_d}^{t_f} (\omega_{\text{ref}}(t) - \omega(t))^2 dt \quad (33)$$

Unit: $\text{rpm}^2 \cdot \text{s}$

- Overshoot

Maximum positive excursion above the reference after the disturbance or command change.

Unit: rpm or %

- Steady-state error

$$e_{ss} = |\omega_{\text{ref}} - \omega_{ss}| \quad (34)$$

2.16 Computational complexity and implementation feasibility

The proposed ICNSS controller is implemented only in the outer speed loop. Therefore, its control decision is updated at the outer-loop sampling interval $T_{s,\text{ speed}} = 10^{-3}$ s, while the plant simulation step is $T_{s,\text{ model}} = 10^{-5}$ s. At each outer-loop update, the controller performs the following operations:

(i) error and error-difference calculation,
(ii) adaptive bias and PI-like baseline update,
(iii) generation of a candidate-action set,
(iv) evaluation of five candidate actions,
(v) computation of the directional agreement term and the truth, indeterminacy, and falsity values for each candidate,
(vi) weighted score calculation, and

(vii) final candidate selection followed by output limiting and smoothing.

Table 3. Computational structure of the proposed interval complex neutrosophic soft sets (ICNSS) speed controller

Item	Value / Description
Control location	Outer speed loop only
Plant simulation step	$T_{s,\,\text{model}} = 10^{-5}$ s
Outer-loop controller update time	$T_{s,\,\text{speed}} = 10^{-3}$ s
Number of candidate actions	5
Candidate evaluation type	Sequential scoring
Membership terms per candidate	truth, indeterminacy, falsity
Additional terms per candidate	Directional agreement, phase surrogate, weighted score
Output post-processing	Saturation, rate limiting, smoothing
Complexity per update	$O(N_c)$, with $N_c = 5$
Real-time validation	Not experimentally benchmarked in this work

In the present implementation, the number of candidate actions is fixed at five per update. Hence, the computational complexity per outer-loop update is linear in the number of candidates, i.e., $O(N_c)$ with $N_c = 5$, which remains moderate. Since the ICNSS algorithm is not executed in the inner current loop or at the PWM switching rate, its computational burden is considerably lower than that of a high-frequency intelligent current controller. The present study demonstrates feasibility in simulation; however, detailed execution-time measurement on a specific embedded processor is left for future work. Table 3 summarizes the computational structure of the proposed ICNSS speed controller.

2.17 Summary of nominal conditions and test scenarios

To present the evaluation procedure more systematically, Table 4 summarizes all test scenarios considered in this study, including the nominal values, disturbed values, percentage changes, switching instants, and disturbance durations. This table helps clarify the severity and timing of the disturbances and parameter variations applied to the motor-drive system.

Table 4. Summary of nominal conditions and test scenarios used for the proposed controller evaluation

Test Category	Parameter	Nominal Value	Disturbed Value (s)	Percentage Change	Switching Time (s)	Disturbance Duration	Purpose
Input-voltage variation	Armature supply voltage V_a	220 V	240 V, 200 V	+9.09%, -9.09%	10 s, 20 s, 30 s, 40 s	10 s each interval	Disturbance rejection
Load-torque fluctuation	Load torque T_L	10 N·m	12 N · m, 8 N · m	+20%, -20%	10 s, 20 s, 30 s, 40 s	10 s each interval	Disturbance rejection
Field-voltage fluctuation	Field voltage V_f	180 V	200 V, 160 V	+11.11%, -11.11%	10 s, 20 s, 30 s, 40 s	10 s each interval	Disturbance rejection
Reference-speed variation	Reference speed ω_{ref}	1250 rpm	1350 rpm, 1150 rpm	+8%, -8%	10 s, 20 s, 30 s, 40 s	10 s each interval	Reference-tracking test
Chopper inductance variation	Chopper inductance L_s	10 mH	20 mH, 0.1 mH	+100%, -99%	10 s, 20 s, 30 s, 40 s	10 s each interval	Parameter variation robustness
Armature resistance variation	Armature resistance R_a	5.8 Ω	6.3 Ω	+8.62%	10 s, 20 s	10 s disturbed interval	Parameter variation robustness

3. RESULTS AND DISCUSSION

The results obtained using MATLAB Simulink with code for the proposed ICNSS controller depend on the disturbance applied to both the PI and ICNSS controllers, and they also assess the motor speed response. Five types of disturbances are applied to demonstrate the performance of the new algorithm and to compare it with that of the PI controller.

3.1 Results for speed response under input voltage fluctuation disturbance

In this subsection of the results, the motor is tested under both the PI and ICNSS algorithms for the speed loop. The input voltage of the DC chopper changes from its original value of 220 V to 240 V, then back to 220 V, and finally to 200 V. The speed response is checked in this stage. A MATLAB-controlled source used with a MATLAB function, and a simple code used to generate this sequence of voltage fluctuations.

As shown in Figure 3, the voltage fluctuation disturbance was eliminated using the new algorithm at both the upper and lower bounds of the fluctuation. While this fluctuation of the input voltage cannot be eliminated using a PI controller during a 200-volt under-voltage (from t = 30 s to t = 40 s), it produces a speed below the required reference speed of 1120 rpm. The ICNSS algorithm has some overshoot at the start of the motor performance.

3.2 Results for speed response under load torque variation disturbance

In this subsection, a MATLAB function block is used to generate a variable load torque that starts from 10 N·m, then increases to 12 N·m, back to 10 N·m, reduces the torque to 8 N·m, and finally back to 10 N·m. The speed response for both the PI and ICNSS algorithms is checked.

As shown in Figure 4, the new algorithm successfully handles this type of disturbance in both the upper and lower values of the load torque. Yet, the PI controller failed to handle the upper bound of the load torque = 12 (from t = 10 s to t = 20 s), and the required speed reference of 1250 rpm was reduced to 1200 rpm. The ICNSS algorithm still exhibits overshoot and undershoot pulsation at the beginning of the disturbance action and then converges to the reference speed during the transition from the load torque values.

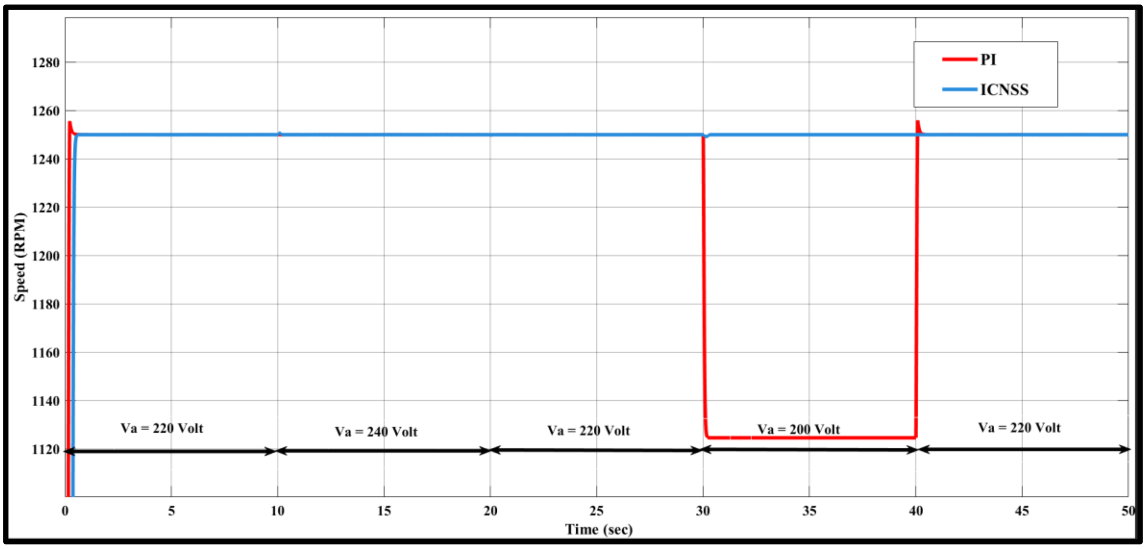

Figure 3. Speed response using both proportional-integral (PI) and interval complex neutrosophic soft sets (ICNSS) approaches with input voltage fluctuation

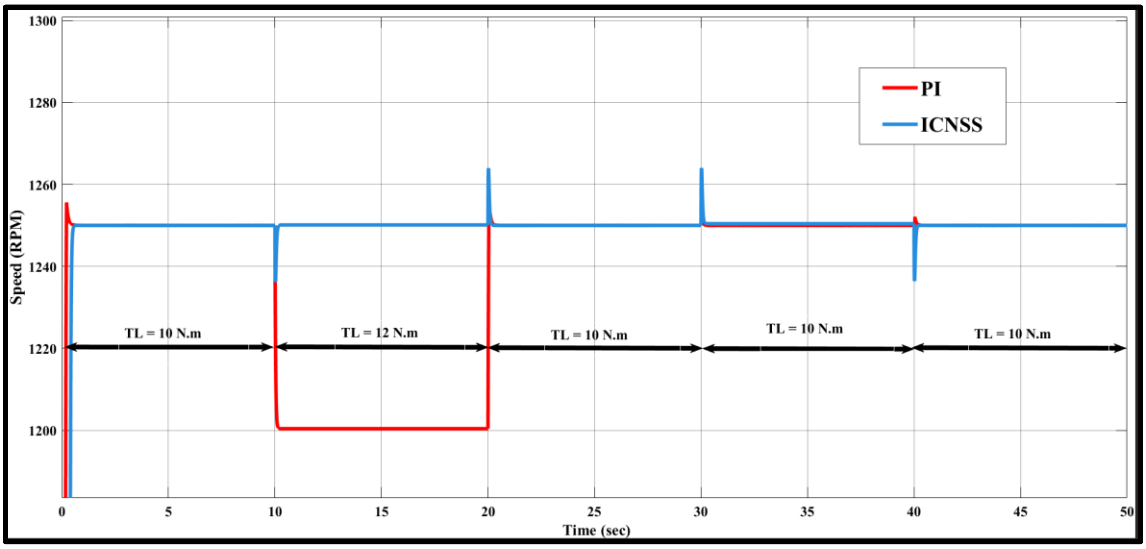

Figure 4. Speed response using both proportional-integral (PI) and interval complex neutrosophic soft sets (ICNSS) approaches with load torque fluctuation

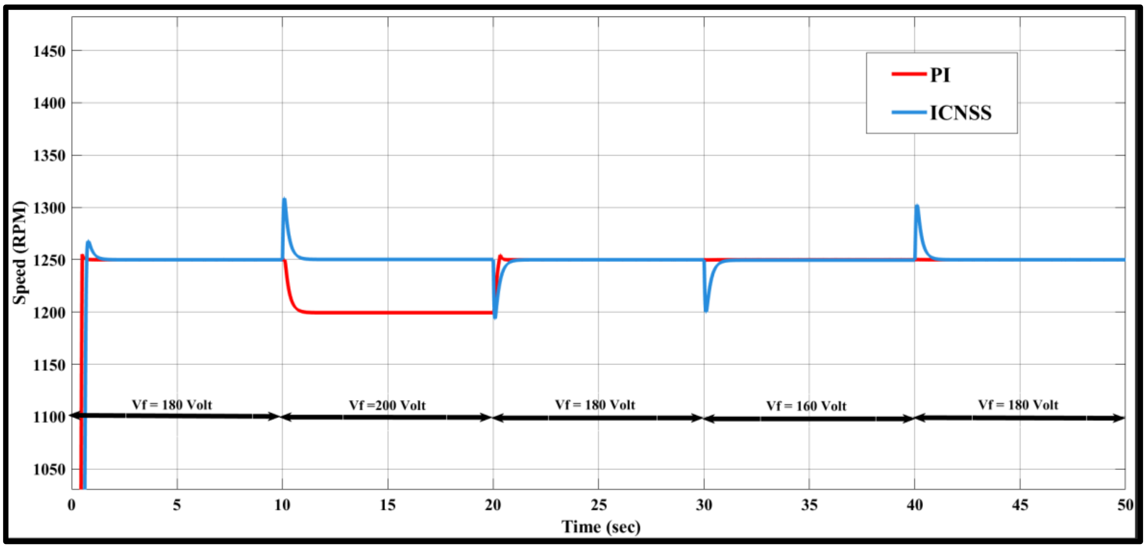

Figure 5. Speed response using both proportional-integral (PI) and interval complex neutrosophic soft sets (ICNSS) approaches with field voltage fluctuation

3.3 Results for speed response under field voltage variation disturbance

A new type of disturbance is demonstrated in this subsection, which is the field-voltage variation. Similar to the input voltage fluctuation, a controlled voltage source with a MATLAB function block used to simulate the field voltage with a MATLAB function block to simulate the voltage variations.

Figure 5 shows the speed response to field voltage variations from 180 V to 200 V, then back to 180 V, then reduced to 160 V and finally back to 180 V during t = 10 to t = 20, t = 30 s, and t = 40 s, respectively. The ICNSS algorithm eliminates this type of disturbance using the upper and lower operating bands. In contrast, the PI controller failed to handle the upper limit of this disturbance (200 V) as demonstrated from t = 10 s to t = 20 s, when the speed was below the reference value of 1200 rpm. The ICNSS algorithm still exhibits overshoot and undershoot pulsation at the beginning of the disturbance action and then converges to the reference speed during the transition from the field voltage values.

3.4 Results for reference-tracking performance test

The purpose of this test is to validate the new algorithm and compare its results with those of the PI controller. To perform this test, a MATLAB function replaces the constant block that represents the reference speed with a code block to simulate speed variation from the reference speed (1250 rpm) to other reference values of 1350 rpm and 1150 rpm. Figure 6 shows that the new ICNSS algorithm also eliminates reference speed fluctuations in both upper and lower speed bounds. The PI controller is unable to handle the upper limit of the new reference, and it sets the speed to only 1275 rpm. As shown, the PI controller displays greater overshoot during transitions between speed values.

3.5 Results for speed response under the uncertainty of the variation of the chopper inductance

During the motor or chopper operation, and because of the current values thrown through the parameters of this chopper, the value of these parameters exhibits variations. In this subsection, the effect of using the new algorithm on this type of uncertainty will be discussed. In this test, the inductor filter as a parameter varies from its original value of 10 mH to 20 mH, then is reduced to 0.1 mH. A special arrangement in MATLAB Simulink is used to perform inductor variation using a breaker block with three inductor blocks and a MATLAB function block to turn the breaker ON or OFF based on the required inductor value.

In Figure 7, the variation of the inductor affects the PI controller speed response clearly. Increasing the inductor value directly affects the speed and produces a transient spike at t = 15 s. The performance under the new ICNSS algorithm is smoother, but it still has overshoot during the starting period.

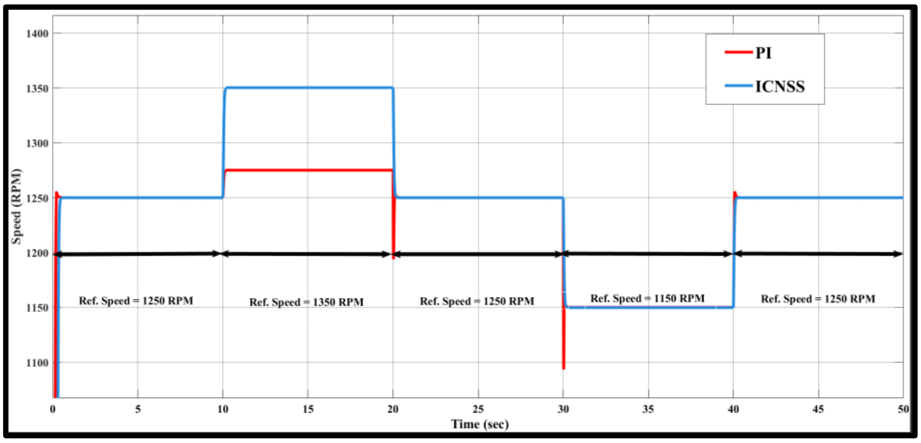

Figure 6. Speed response using both proportional-integral (PI) and interval complex neutrosophic soft sets (ICNSS) approaches using reference-tracking performance test

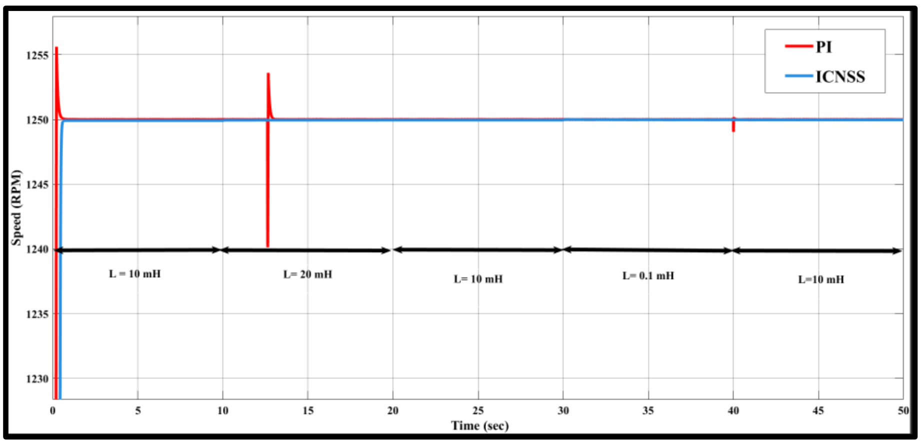

Figure 7. Speed response using both proportional-integral (PI) and interval complex neutrosophic soft sets (ICNSS) approaches under variation of the chopper inductance

3.6 Results for speed response under the uncertainty of Motor Armature Resistance

This type of uncertainty is tested using variable resistance connected in series with the original armature resistance of the DC motor. The resistance is increased to 0.5 Ω at t = 10 s, and the speed response is checked as in Figure 8. The armature resistance fluctuation directly affects the speed value in the case of a PI controller, where the system failed to reach the reference speed, settling at 1234 rpm. Meanwhile, the ICNSS approach successfully handles this disturbance with an overshoot and undershoot during the transition between reference speed variations.

Overall, the results indicate that the proposed ICNSS controller does not produce perfect disturbance rejection in the strict sense, since transient deviations still appear during startup and at some disturbance transitions; however, it consistently reduces the disturbance effect more effectively than the conventional PI controller and preserves closer tracking of the reference speed.

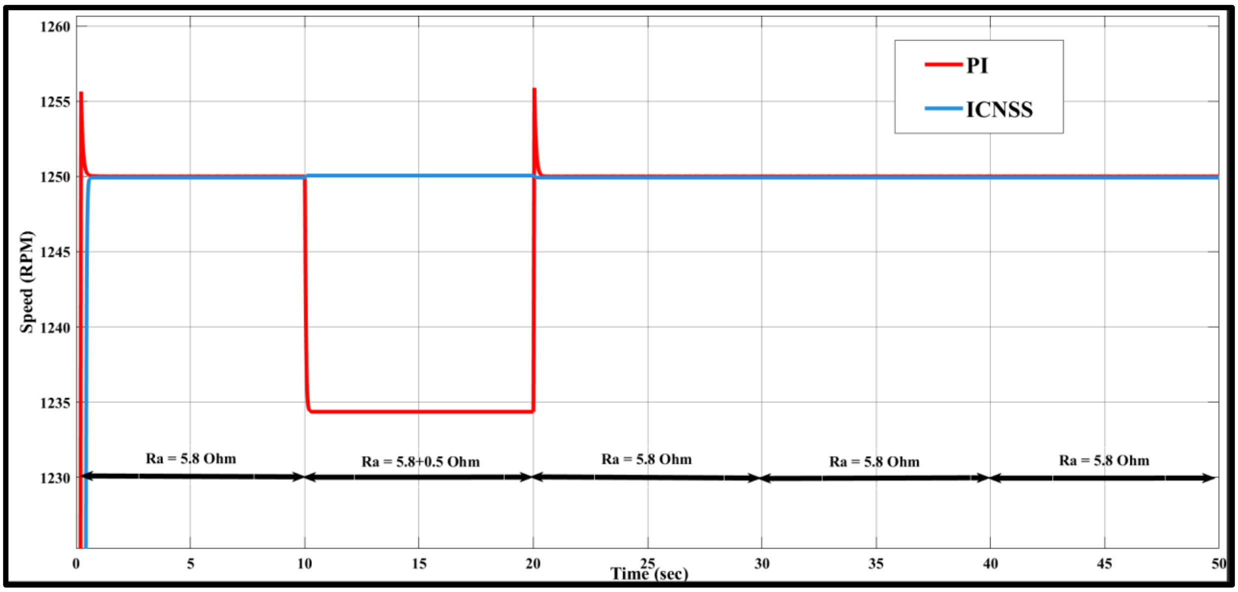

Figure 8. Speed response using both proportional-integral (PI) and interval complex neutrosophic soft sets (ICNSS) approaches with armature resistance variation uncertainty

Table 5. Quantitative performance under different variation and disturbance scenarios using proportional-integral (PI) and interval complex neutrosophic soft set (ICNSS) controllers

Scenario	Controller	Maximum Speed Deviation (rpm)	Recovery Time (s)	IAE (rpm•s)	ISE ($rpm^2 \cdot s$)	Overshoot (rpm)	Steady-State Error (rpm)
			Input Voltage Variation				
220 → 240 V	PI	≈ 0.5	≈ 0.05	≈ 0.02	≈ 0.01	≈ 0.5	≈ 0
220 → 240 V	ICNSS	≈ 0.2	≈ 0.05	≈ 0.01	≈ 0.005	≈ 0.2	≈ 0
220 → 200 V	PI	≈ 125	≈ 0.35 after 40 s return	≈ 1250	≈ 156250	≈ 5 at recovery	≈ 125 during sag
220 → 200 V	ICNSS	≈ 1.0	≈ 0.10	≈ 1.5	≈ 1.5	≈ 0.5	≈ 0
			Load Torque Fluctuation				
10 → 12 N·m	PI	≈ 50	Not recovered within 10 s	≈ 500	≈ 25000	-50	≈ 50
10 → 12 N·m	ICNSS	≈ 100	≈ 5.0	≈ 180	≈ 9000	+100	≈ 0
12 → 10 N·m	PI	≈ 50	≈ 0.3	≈ 10	≈ 300	+5	≈ 0
12 → 10 N·m	ICNSS	≈ 2	≈ 0.1	≈ 1	≈ 1	+2	≈ 0
10 → 8 N·m	PI	≈ 1	≈ 0.1	≈ 1	≈ 1	+1	≈ 0
10 → 8 N·m	ICNSS	≈ 72	≈ 1.5	≈ 65	≈ 3500	+72	≈ 0
8 → 10 N·m	PI	≈ 2	≈ 0.1	≈ 1	≈ 1	+2	≈ 0
8 → 10 N·m	ICNSS	≈ 58	≈ 1.0	≈ 55	≈ 3000	-58	≈ 0
			Field Voltage Fluctuation				
180 → 200 V	PI	≈ 50	Not recovered within 10 s	≈ 500	≈ 25000	-50	≈ 50
180 → 200 V	ICNSS	≈ 60	≈ 2.5	≈ 120	≈ 4500	+60	≈ 0
200 → 180 V	PI	≈ 5	≈ 0.2	≈ 2	≈ 5	+5	≈ 0
200 → 180 V	ICNSS	≈ 55	≈ 1.0	≈ 45	≈ 2500	-55	≈ 0
180 → 160 V	PI	≈ 1	≈ 0.1	≈ 1	≈ 1	+1	≈ 0
180 → 160 V	ICNSS	≈ 50	≈ 1.0	≈ 40	≈ 2000	-50	≈ 0
160 → 180 V	PI	≈ 5	≈ 0.2	≈ 2	≈ 5	+5	≈ 0
160 → 180 V	ICNSS	≈ 50	≈ 1.0	≈ 45	≈ 2200	+50	≈ 0

Reference-Tracking Performance Test							
Start-up to 1250 rpm	PI	≈ 6	≈ 0.25	≈ 8	≈ 20	+6	≈ 0
Start-up to 1250 rpm	ICNSS	≈ 0 to 2	≈ 0.45	≈ 20	≈ 80	≈ 0	≈ 0
1250 → 1350rpm	PI	≈ 75	Not reached within 10 s	≈ 750	≈ 56250	-75	≈ 75
1250 → 1350rpm	ICNSS	≈ 0 to 2	≈ 0.15	≈ 3	≈ 5	≈ 0	≈ 0
1350 → 1250rpm	PI	≈ 55	≈ 0.35	≈ 15	≈ 500	-55	≈ 0
1350 → 1250rpm	ICNSS	≈ 0 to 2	≈ 0.10	≈ 1	≈ 1	≈ 0	≈ 0
1250 → 1150rpm	PI	≈ 55	≈ 0.25	≈ 10	≈ 350	-55	≈ 0
1250 → 1150rpm	ICNSS	≈ 0 to 2	≈ 0.10	≈ 1	≈ 1	≈ 0	≈ 0
1150 → 1250rpm	PI	≈ 6	≈ 0.20	≈ 4	≈ 10	+6	≈ 0
1150 → 1250rpm	ICNSS	≈ 0 to 2	≈ 0.10	≈ 1	≈ 1	≈ 0	≈ 0
Variation of the Chopper Inductance							
10 → 20 mH	PI	≈ 3.5	≈ 0.20	≈ 0.6	≈ 1.5	+3.5	≈ 0
10 → 20 mH	ICNSS	≈ 0.2	≈ 0.05	≈ 0.05	≈ 0.01	≈ 0	≈ 0
20 → 10 mH	PI	≈ 0.0 to 0.2	≈ 0.05	≈ 0.01	≈ 0.001	≈ 0	≈ 0
20 → 10 mH	ICNSS	≈ 0.0 to 0.1	≈ 0.05	≈ 0.01	≈ 0.001	≈ 0	≈ 0
10 → 0.1 mH	PI	≈ 5.0	≈ 0.20	≈ 0.8	≈ 2.5	-5.0 then +3.5	≈ 0
10 → 0.1 mH	ICNSS	≈ 0.2	≈ 0.05	≈ 0.05	≈ 0.01	≈ 0	≈ 0
0.1 → 10 mH	PI	≈ 1.0	≈ 0.10	≈ 0.1	≈ 0.05	-1.0	≈ 0
0.1 → 10 mH	ICNSS	≈ 0.1	≈ 0.05	≈ 0.01	≈ 0.001	≈ 0	≈ 0
Armature Resistance Variation							
5.8 → 6.3 Ω	PI	≈ 15.5	Not recovered within 10 s	≈ 155	≈ 2400	-15.5	≈ 15.5
5.8 → 6.3 Ω	ICNSS	≈ 0.1	≈ 0.05	≈ 0.01	≈ 0.001	≈ 0	≈ 0
6.3 → 5.8 Ω	PI	≈ 5.8	≈ 0.20	≈ 1.5	≈ 12	+5.8	≈ 0
6.3 → 5.8 Ω	ICNSS	≈ 0.1	≈ 0.05	≈ 0.01	≈ 0.001	≈ 0	≈ 0

Note: PI = proportional-integral; ICNSS = interval complex neutrosophic soft set; IAE = integral of absolute error; ISE = integral of squared error.

3.7 Quantitative performance results

To support the visual comparison of Figures 3-8 with quantitative evidence, Table 5 summarizes the dynamic performance of the benchmark PI controller and the proposed ICNSS controller for each test case. The comparison includes maximum speed deviation, recovery time, IAE, ISE, overshoot, and steady-state error. These metrics provide a more objective evaluation of the disturbance-handling and reference-tracking performance of both controllers.

From Table 5, we can observe the following:

Under input-voltage variation, the performance difference between the two controllers becomes very clear, especially during the voltage drop case. When the input voltage increases from 220 V to 240 V, both controllers maintain the speed very close to the 1250 rpm reference, with only negligible deviation, very small IAE and ISE values, and almost zero steady-state error. This indicates that both controllers are sufficiently robust under moderate voltage rise. However, when the input voltage decreases from 220 V to 200 V, the PI controller experiences a severe deterioration in performance. Its maximum speed deviation reaches about 125 rpm, the steady-state error remains around 125 rpm during the voltage sag, and the IAE and ISE rise sharply to about 1250 rpm·s and 156250 $\text{rpm}^2 \cdot \text{s}$, respectively. Moreover, the PI controller does not recover during the sag interval and only recovers after the nominal voltage is restored. In contrast, the ICNSS controller keeps the speed almost unchanged, with only about 1 rpm maximum deviation, around 0.10 s recovery time, negligible overshoot, and nearly zero steady-state error. These results indicate that the proposed ICNSS controller has a significantly stronger ability to handle under-voltage conditions than the benchmark PI controller.

The load-torque fluctuation case shows a mixed performance pattern. When the load torque increases from 10 to 12 N·m, the PI controller exhibits a speed drop of about 50 rpm and fails to recover during the disturbance interval, producing a steady-state error of approximately 50 rpm and relatively large IAE and ISE values. By contrast, the ICNSS controller restores speed to the reference value, giving almost zero steady-state error, but this comes at the cost of a relatively large transient overshoot of about 100 rpm and a longer recovery time of about 5 s. This means that the ICNSS controller is more effective in final disturbance rejection under increased load, but its transient response is more aggressive. When the torque returns from 12 to 10 N·m, ICNSS performs better than PI, since its deviation is only about 2 rpm compared with about 50 rpm for PI. However, when the torque decreases from 10 to 8 N·m and later returns from 8 to 10 N·m, the PI controller behaves much more smoothly, with only very small deviations and negligible overshoot, whereas the ICNSS controller produces much larger transient excursions, with about 72 rpm overshoot in the 10 → 8 N·m case and about 58 rpm undershoot in the 8 → 10 N·m case. Therefore, under load-torque fluctuation, the ICNSS controller is advantageous when the main objective is to eliminate steady-state speed drop under heavier loading, but the PI controller is better in terms of transient smoothness when the torque is reduced or restored.

The field-voltage fluctuation case also reveals different strengths for the two controllers. When the field voltage increases from 180 V to 200 V, the PI controller again suffers a persistent speed reduction of about 50 rpm, with a corresponding steady-state error of about 50 rpm and large IAE and ISE values. In contrast, the ICNSS controller overshoots by about 60 rpm but eventually recovers to reference speed, resulting in nearly zero steady-state error and much smaller integral error indices. Hence, in this transition, ICNSS provides better final regulation than PI. On the other hand, when the field voltage returns from 200 V to 180 V, and also when it decreases from 180 V to 160 V and returns again to 180 V, the PI controller is clearly smoother. Its maximum deviation is only around 5 rpm or less, while the ICNSS controller exhibits larger undershoot and overshoot of about 50-55 rpm. Thus, the proposed controller is better at removing the steady-state drift caused by increased field voltage, but the PI controller remains superior in terms of transient smoothness for the other field-voltage transitions. This suggests that further tuning of the ICNSS transient decision parameters may be needed for this class of disturbance.

The reference-tracking test is the strongest case in favor of the ICNSS controller. At startup toward 1250 rpm, the PI controller responds faster, with a recovery time of about 0.25 s, while the ICNSS controller takes about 0.45 s. However, ICNSS produces almost no overshoot, whereas PI overshoots by about 6 rpm. The most important difference appears when the reference speed is raised from 1250 rpm to 1350 rpm. In this case, the PI controller fails to track the new command and remains around 1275 rpm, leading to about 75 rpm maximum deviation, 75 rpm steady-state error, and very large IAE and ISE values. By contrast, the ICNSS controller tracks the 1350 rpm command almost exactly, with only negligible deviation and very small error indices. The same pattern is observed when the reference returns from 1350 rpm to 1250 rpm and later changes from 1250 rpm to 1150 rpm and back to 1250 rpm. In all these transitions, the PI controller shows noticeable undershoot or overshoot, while the ICNSS controller follows the new reference much more accurately and with shorter recovery time. Therefore, although PI is faster during startup, the ICNSS controller clearly outperforms the PI controller in reference-tracking accuracy and command-following capability, especially for large speed-reference changes.

The variation of chopper inductance has only a minor influence on the speed response for both controllers. In all transitions, including $10 \rightarrow 20\ \mathrm{mH}, 20 \rightarrow 10\ \mathrm{mH}, 10 \rightarrow 0.1\ \mathrm{mH}$, and $0.1 \rightarrow 10\ \mathrm{mH}$, both controllers maintain the motor speed very close to the 1250 rpm reference, with nearly zero steady-state error and very small IAE and ISE values. The PI controller shows small transient spikes at some inductance changes, particularly at $10 \rightarrow 20\ \mathrm{mH}$ and $10 \rightarrow 0.1\ \mathrm{mH}$, while the ICNSS controller remains almost completely flat. Even so, the absolute magnitude of the deviations in both cases is very small, indicating that the system is inherently robust to chopper-inductance variation. Therefore, this scenario is not a strong discriminator between the two controllers, although the ICNSS controller can still be considered marginally smoother.

The armature-resistance variation case strongly favors the ICNSS controller. When the armature resistance increases from 5.8 Ω to 6.3 Ω, the PI controller suffers a persistent speed drop of approximately 15.5 rpm and maintains this error during the disturbed interval, which causes a nonzero steady-state error and larger integral error measures. In contrast, the ICNSS controller keeps the speed almost exactly at the reference, with negligible deviation and essentially zero steady-state error. When the resistance is restored from 6.3Ω back to 5.8Ω, the PI controller produces a small overshoot of about 5.8 rpm, whereas the ICNSS controller remains almost unchanged. These results demonstrate that the proposed ICNSS controller is far less sensitive to armature-resistance uncertainty than the conventional PI controller and provides clearly better robustness against this type of parameter variation.

3.8 Step response parameters calculations

Table 6 demonstrates the transient response of the conventional PI controller and the proposed ICNSS controller. The results are obtained using the stepinfo MATLAB function, and various motor speed parameters have been calculated.

The results indicate that the PI controller gives a faster initial response, with a rise time of 0.0089 s compared with 0.0460 s for the ICNSS controller, and it also reaches the settling region sooner, with a settling time of 0.19 s versus 0.32 s. However, the ICNSS algorithm provides slightly smoother behaviour, as shown by its lower settling maximum (1249.9 rpm versus 1255.6 rpm), lower peak speed, lower overshoot, and lower undershoots. In addition, the peak time of the ICNSS controller is 1.0994 s, compared with 0.8206 s for the PI controller, which indicates that the proposed algorithm responds more slowly. Therefore, although the PI controller offers faster response, the ICNSS controller exhibits enhanced damping and reduced transient excursions around the reference speed. This behaviour implies that the proposed ICNSS approach sacrifices some response speed to achieve smoother, less oscillatory transient performance under various types of disturbances.

Table 6. Speed step response parameters using both proportional-integral (PI) and interval complex neutrosophic soft sets (ICNSS) approaches

Parameter Name	PI Controller	ICNSS Controller	Unit
Rise Time	0.0089	0.0460	s
Settling Time, 4% criteria	0.19	0.32	s
Peak Value	1255.6	1249.9	rpm
Peak Time	0.8206	1.0994	s
Overshoot	0.448	0	%
Undershoot	11.8094	11.2272	%
Steady-state value	1250	1250	rpm

Note: A 4% settling criterion was selected because it provides a practical tolerance band for the motor-speed response under disturbance and uncertainty conditions, while avoiding an overly strict criterion that may exaggerate insignificant small residual oscillations.

4. CONCLUSIONS

The results of this work demonstrate that the proposed ICNSS-based speed controller for the speed loop provides clear advantages in rejecting different types of disturbances and uncertainties in the DC motor and its drive system. The controller demonstrated a strong ability to mitigate the effects of input-voltage variation, load-torque variation, reference-speed variation, field-voltage fluctuation, and system-parameter deviation, while maintaining the motor speed closer to the reference value than the conventional PI controller. Compared with the conventional PI controller, the proposed approach exhibited improved disturbance-handling capability and greater robustness under varying operating conditions, although transient overshoot and short-duration deviations were still present in some cases. The efficiency of the proposed method comes from numerous built-in mechanisms. First, the deadband structure reduces sensitivity to small measurement disturbances and minor fluctuations around the operating point, which helps prevent unnecessary control action. Second, the adaptive candidate-action mechanism allows the controller to apply larger corrective actions under severe disturbances, while only small actions are applied near the equilibrium point. This control provides the controller with an adaptive action magnitude based on the disturbance level. Furthermore, the decision technique estimates whether the candidate control action aligns with the expected corrective direction. Therefore, when the speed error is positive, indicating that the actual speed is below the reference, a positive current correction is selected. In contrast, when the speed error is negative, a negative current correction is selected. This directional consistency improves the control response under transient and disturbed conditions. Moreover, the ICNSS decision representation does not depend only on static truth, indeterminacy, and falsity measures, but also includes the dynamic phase condition of the speed error. This new mechanism provides the controller with an additional ability to respond differently to changing transients and different disturbance scenarios, which enhances its flexibility and disturbance-rejection performance. Despite these advantages, the proposed controller still exhibits noticeable overshoot during disturbance handling in some types of disturbances and uncertainties. Quantitative Performance show that the proposed ICNSS controller is not uniformly superior in every metric under every operating condition, but it demonstrates important advantages in the most critical cases. Its strongest benefits appear in under-voltage disturbance rejection, reference-speed tracking, and armature-resistance variation, where it maintains the speed much closer to the desired reference than the PI controller. It also improves final regulation in some load-torque and field-voltage disturbance cases, although this often comes with larger transient overshoot or undershoot. By contrast, the PI controller is generally smoother in some torque-release and field-voltage-return cases, and it is also faster during startup. Therefore, the main strength of the ICNSS controller lies in its enhanced robustness and tracking accuracy under significant disturbances and uncertainties, while its main limitation is the larger transient excursion that may appear in some transition conditions. Furthermore, the step response for the new algorithm is poor compared to the PI controller. This limitation is mainly due to the ICNSS control parameters not yet being fully optimized. Thus, future work will focus on systematically tuning the suggested controller parameters using suitable optimization methods to reduce overshoot further and improve the transient performance of the control system. Overall, the proposed ICNSS controller improved robustness and reduced the effect of several disturbances and uncertainties; however, this benefit was achieved at the expense of a slower initial transient response than the benchmark PI controller, as reflected by the rise time, settling time, and peak time results.

REFERENCES

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

[2] Montoya-Acevedo, D., Gil-González, W., Montoya, O.D., Restrepo, C., González-Castaño, C. (2024). Adaptive speed control for a DC motor using DC/DC converters: An inverse optimal control approach. IEEE Access, 12: 154503-154513. https://doi.org/10.1109/ACCESS.2024.3482982

[3] Yasar, C.F., Ziraoui, A., Jneid, A.A., Ali, M. (2025). DC motor control lab: Comparative study of model predictive and PI control using MATLAB/Simulink. IFAC-PapersOnLine, 59(7): 111-116. https://doi.org/10.1016/j.ifacol.2025.08.032

[4] Xu, Y., Deng, C. (2017). An investigation on 3D printing technology for power electronic converters. In 2017 IEEE 8th International Symposium on Power Electronics for Distributed Generation Systems (PEDG), Florianopolis, Brazil, pp. 1-6. https://doi.org/10.1109/pedg.2017.7972486

[5] Can, E. (2026). UAV DC motor behaviors control using new hybrid PSO-fuzzy logic with a dynamic error based optimization. Aerospace Science and Technology, 168: 111156. https://doi.org/10.1016/j.ast.2025.111156

[6] Hsueh, P.W., Fazdi, M.F. (2026). Parameter estimation of permanent magnet DC motor using various particle swarm optimization variants. Array, 30: 100823. https://doi.org/10.1016/j.array.2026.100823

[7] Orta-Quintana, Á.A., García-Chávez, R.E., Silva-Ortigoza, R., Marciano-Melchor, M., et al. (2025). Robust two-stage controller for the bidirectional 'full-bridge Buck inverter-DC motor' fed by renewable energy. Results in Engineering, 26: 105209. https://doi.org/10.1016/j.rineng.2025.105209

[8] Wu, W. (2011). Disturbance compensation for DC motor mechanism low speed regulation: A feedforward and feedback implementation. In IEEE Conference on Decision and Control and European Control Conference, Orlando, FL, USA, pp. 1614-1619. https://doi.org/10.1109/CDC.2011.6160408

[9] Antar, R.K. (2018). Speed/torque estimation and control of a DC machine in four quadrants operation modes. Al-Nahrain Journal for Engineering Sciences, 21(2): 238-247. https://doi.org/10.29194/NJES21020238

[10] Liceaga-Castro, J.U., Siller-Alcalá, I.I., Román, J.D.G.S., Alcántara-Ramírez, R.A. (2022). PI speed control with reverse motion of a series DC motor based on the noise reduction disturbance observer. Actuators, 11(5): 117. https://doi.org/10.3390/act11050117

[11] Acevedo, D.M., Parraguez-Garrido, I., Gil-Gonzalez, W., Montoya, O.D., Gonzalez-Castano, C. (2025). Adaptive passivity-based control for DC motor speed regulation in DC-DC converter-fed systems. IEEE Access, 13: 131957-131966. https://doi.org/10.1109/ACCESS.2025.3592594

[12] Alnaib, A.M.T.I., Sultan, N.S., Mahmood, O.T. (2018). Design a fuel cell based drive DC motor for an electric vehicle applications. International Journal of Engineering and Technology (UAE), 7(4): 2081-2087. https://doi.org/10.14419/ijet.v7i4.16308

[13] Has, Z., Muslim, A.H., Mardiyah, N.A. (2017). Adaptive-fuzzy-PID controller based disturbance observer for DC motor speed control. In 2017 4th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Yogyakarta, Indonesia, pp. 1-6. https://doi.org/10.1109/eecsi.2017.8239165

[14] Montoya-Acevedo, D., Montoya, O.D., Garcés-Ruiz, A., González-Castaño, C., Restrepo, C. (2025). Continuous control-set model-predictive control for a series-wound DC motor drive. IEEE Transactions on Industrial Electronics, 73(5): 6680-6690. https://doi.org/10.1109/TIE.2025.3637340

[15] Nizami, T.K., Gangula, S.D., Udumula, R.R., Chakravarty, A., Ahmad, F., Hosseinpour, A. (2025). Nonlinear adaptive neural control of power converter-driven DC motor system: Design and experimental validation. Engineering Reports, 7(1): 1-16. https://doi.org/10.1002/eng2.13025

[16] Luna, L., Chairez, I., Polyakov, A. (2026). Improved trajectory tracking for DC-motors via homogeneous PID controller. IEEE Transactions on Automation Science and Engineering, 23: 7544-7557. https://doi.org/10.1109/TASE.2026.3677895

[17] Prasanthi, A., Shareef, H., Asna, M. (2024). An application of feedback linearizing controller with disturbance observer for DC motor real-time speed control. Journal of Physics: Conference Series, 2689(1): 012001. https://doi.org/10.1088/1742-6596/2689/1/012001

[18] Umar, A.A., Kim, J.S. (2025). Constant reference tracking in data-driven control. International Journal of Control, Automation and Systems, 23(2): 611-619. https://doi.org/10.1007/s12555-024-0516-x

[19] Hazaymeh, A., Al-Qudah, Y., Al-Sharqi, F., Bataihah, A. (2025). A novel Q-neutrosophic soft under interval matrix setting and its applications. International Journal of Neutrosophic Science, 25(4): 156-168. https://doi.org/10.54216/IJNS.250413

[20] Sai Anu, P., Arulselvam, A., Bobin, A. (2025). Intuitionistic quadri-partitioned neutrosophic sets with extended operators. Neutrosophic Sets and Systems, 80: 443-464. https://doi.org/10.5281/zenodo.14708891

[21] Saqlain, M., Garg, H., Kumam, P., Kumam, W. (2023). Uncertainty and decision-making with multi-polar interval-valued neutrosophic hypersoft set: A distance, similarity measure and machine learning approach. Alexandria Engineering Journal, 84: 323-332. https://doi.org/10.1016/j.aej.2023.11.001

[22] Al-Saidi, M., Ballagi, Á., Hassen, O.A., Saad, S.M. (2024). Type-2 neutrosophic markov chain model for subject-independent sign language recognition: A new uncertainty-aware soft sensor paradigm. Sensors, 24(23): 7828. https://doi.org/10.3390/s24237828

[23] Arshad, M., Saeed, M., Rahman, A.U. (2023). Interval complex single-valued neutrosophic hypersoft set with application in decision making. Neutrosophic Sets and Systems, 60: 396-419. https://doi.org/10.5281/zenodo.4568742

[24] Peng, X. (2019). New multiparametric similarity measure and distance measure for interval neutrosophic set with IoT industry evaluation. IEEE Access, 7: 28258-28280. https://doi.org/10.1109/ACCESS.2019.2902148

[25] Saeed, M.M., Khan, S.U., Suriyya, F., Hamja, J.J., Mehmood, A. (2025). Interval-valued complex neutrosophic sets and complex neutrosophic soft topological spaces. International Journal of Analysis and Applications, 23: 1-35. https://doi.org/10.28924/2291-8639-23-2025-132

[26] Mohd Kamal, N.L.A., Abdullah, L., Abdullah, I., Saqlain, M. (2020). Multi-valued interval neutrosophic linguistic soft set theory and its application in knowledge management. CAAI Transactions on Intelligence Technology, 5(3): 200-208. https://doi.org/10.1049/trit.2020.0036

[27] Dat, L.Q., Thong, N.T., Son, L.H., Ali, M., Smarandache, F., Abdel-Basset, M. (2019). Linguistic approaches to interval complex neutrosophic sets in decision making. IEEE Access, 7: 38902-38917. https://doi.org/10.1109/ACCESS.2019.2902841

[28] Al-Sharqi, F., Ahmad, A.G., Al-Quran, A. (2022). Similarity measures on interval-complex neutrosophic soft sets with applications to decision making and medical diagnosis under uncertainty. Neutrosophic Sets and Systems, 51(1): 32. https://doi.org/10.5281/zenodo.7135362

[29] Precup, R.E., Nguyen, A.T., Blažič, S. (2024). A survey on fuzzy control for mechatronics applications. International Journal of Systems Science, 55(4): 771-813. https://doi.org/10.1080/00207721.2023.2293486

[30] Rodriguez, J., Garcia, C., Mora, A., Flores-Bahamonde, F., et al. (2021). Latest advances of model predictive control in electrical drives—Part I: Basic concepts and advanced strategies. IEEE Transactions on Power Electronics, 37(4): 3927-3942. https://doi.org/10.1109/TPEL.2021.3121532

[31] Li, C., Meng, Q., Shi, T. (2024). A review on model predictive control strategies for AC motor drives. IET Electric Power Applications, 18(11): 1584-1604. https://doi.org/10.1049/elp2.12510

[32] Prabhakaran, J., Thirumoorthi, P., Mathankumar, M., Mangaiyarkarasi, S.P. (2025). Development of intelligent controller for high performance electric drives with hybrid CSA and RERNN technique. Scientific Reports, 15(1): 14020. https://doi.org/10.1038/s41598-025-98704-7

[33] Yahia, T.T., Antar, R.K., Saleh, A.A. (2023). DC motor speed/torque estimation and control based on transfer function characteristics. In 2023 International Conference on Engineering, Science and Advanced Technology (ICESAT), Mosul, Iraq, pp. 156-161. https://doi.org/10.1109/ICESAT58213.2023.10347312

[34] Yolcu, A., Öztürk, T.Y. (2026). Neutrosophic hesitant fuzzy soft sets: Algebraic structure, topology, and application in decision making. Journal of Ambient Intelligence and Humanized Computing, 17: 185-203. https://doi.org/10.1007/s12652-025-05028-5

[35] Wasim, A., Saeed, M., Saeed, M.H., Deveci, M., Mohammed, M.A., Tomášková, H. (2026). Symmetrical behaviors based on order relations in advanced fuzzy soft sets for optimization and collaborative solutions. Applied Soft Computing, 197: 115180. https://doi.org/10.1016/j.asoc.2026.115180