Power Electronics and Drives

Towards High-Performance DC Motor Control: Fractional Modelling and FOPID Optimisation

Research Paper

Bilel Kanzari* and Adel Taeib

[1]*ENSIT*
[2]*University of Tunis Higher National School of Engineering*

Received: 07 September, 2025; Received in the revised form: 05 November, 2025; Accepted: 15 November, 2025

Abstract: DC motor speed control is fundamental in modern industrial and robotic systems, where high precision, robustness and energy efficiency are required. Conventional integer-order Proportional–Integral–Derivative (PID) controllers often fail to capture the non-linearities and parameter variations inherent in real DC motors. This study proposes a control framework combining fractional-order (FO) system identification with an optimised fractional-order proportional-integral-derivative (FOPID) controller. The five FOPID parameters are optimised using four metaheuristic algorithms: Grey Wolf Optimizer (GWO), Firefly Algorithm (FA), artificial bee colony (ABC) and ant colony optimisation (ACO). Experimental validation on a MATLAB/Simulink *R2024 (The MathWorks, Inc., Natick, MA, USA)*, an Arduino board, and a DC motor platform demonstrates that the particle swarm optimisation (PSO) FOPID controller achieves a settling time of 1.08 s with 2.00% overshoot and a control effort of 1.8 V/√s. Compared to the extended Ziegler–Nichols tuned FOPID, the PSO approach achieves 98.57% faster settling while maintaining comparable overshoot and demonstrating superior energy efficiency. Among the metaheuristic algorithms tested, PSO demonstrates the best overall performance with the lowest identification error and the most energy-efficient control effort. These results confirm the superiority of the metaheuristic optimisation approach over conventional tuning methods in terms of dynamic response, precision, and robustness for fractional-order control systems.

Keywords: *fractional PID controller • PSO • GA • DC motor speed control • robustness*

1. Introduction

The speed control of direct current (DC) motors remains a fundamental challenge in industrial automation, robotics, and energy-efficient drive systems. Achieving high precision, fast transient response, and low energy consumption requires robust control techniques capable of handling system non-linearities and parameter uncertainties (Nesri et al., 2024).

Traditionally, DC motor control relies on integer-order modelling and classical PID controllers, valued for their simplicity and ease of implementation. Standard tuning methods—such as the (Ziegler and Nichols 1942; Cohen and Coon 1953) procedures—provide acceptable transient and steady-state responses for many applications. However, these empirical techniques often lead to performance trade-offs, especially in systems with time-varying parameters, non-linear inductive dynamics, or high precision requirements.

To overcome these limitations, recent research has focused on heuristic and intelligent optimisation algorithms for PID and fractional-order proportional-integral-derivative (FOPID) tuning (Idir et al., 2022). For example, particle swarm optimisation (PSO) and genetic algorithm (GA) have demonstrated faster convergence and improved

* *Email: kanzari.bilel@outlook.fr*

robustness compared with classical tuning methods (Nasir & Khadraoui, 2021; Martinez-Patiño et al., 2023; Sultan et al., 2022). Other approaches, such as artificial bee colony (ABC) and ant colony optimisation (ACO), have shown strong global search capabilities for non-linear control systems (Idir et al., 2022). These techniques fall under the umbrella of artificial intelligence (AI)-based control, enabling adaptive and energy-aware performance optimisation. Unlike conventional approaches that apply FOPID controllers to predefined models, this study introduces a novel integrated methodology combining fractional-order (FO) system identification with metaheuristic optimisation to simultaneously achieve an accurate plant model and an optimal controller tuning.

In parallel, fractional calculus (Abu-Shady, M. and Kaabar, 2021) has emerged as a powerful mathematical framework for modelling and control. Unlike integer-order models, FO models can capture memory and hereditary effects found in physical systems, such as viscoelastic materials, diffusion processes, and electromagnetic actuators (Li and Rosenfeld, 2021; Meneses and Arrieta, 2022). Integrating FO modelling into control design leads to more accurate system representations and improved control precision.

The (FOPID) controller, first introduced by Podlubny (1999), generalises the classical PID by introducing two additional parameters—the integral order λ and derivative order μ—that allow finer tuning of transient and steady-state behaviours. Several studies (Martinez-Patiño et al., 2023; Sultan et al., 2022) have shown that FOPID controllers tuned via heuristic algorithms outperform traditional PID in terms of overshoot, settling time, and robustness (Pandey and Murray 2022). Nevertheless, few works combine FO system identification with heuristic optimisation and even fewer analyse energy efficiency during stabilisation, which is crucial for embedded and sustainable control applications.

In this context, the study proposes an integrated approach for DC motor speed control based on:

1. FO system identification, treating the motor as a “black box” to accurately capture its dynamic response;
2. FOPID controller design and optimisation using four heuristic algorithms (PSO, GA, ABC and ACO);
3. Performance evaluation focusing not only on classical indices (ITAE, overshoot and settling time) but also on energy consumption during stabilisation;
4. Experimental validation on a real DC motor using MATLAB/Simulink–Arduino Uno integration to confirm practical feasibility.

The main contribution of this work is the implementation of a high-performance practical control solution for a real DC motor treated as a black-box system. By developing a FOPID controller optimised via metaheuristic algorithms – particularly PSO – we demonstrate exceptional performance metrics, including 98.57% faster settling time and minimal control effort of 2.4 V, achieving superior precision and energy efficiency compared to conventional methods.

The rest of this study is organised as follows: Section 2 reviews fractional calculus fundamentals and the structure of the FOPID controller. Section 3 presents the identification and optimisation methodology for both integer-order and FO models. Section 4 describes the tuning of the five FOPID parameters via heuristic algorithms. Section 5 provides the simulation and experimental results, and Section 6 concludes with final remarks and future research perspectives.

In summary, this study contributes to the state of the art by (I) bridging the gap between FO modelling and heuristic optimisation, (II) emphasising energy-aware FOPID control and (III) validating the approach experimentally, thereby situating it firmly within and beyond existing classical and AI-based control frameworks.

2. Calculus for Fractional Orders

In 1695, Leibniz corresponded with J. Wallis and J. Bernoulli regarding a prospective approach to differentiation involving non-integer orders of m. The definition is presented as follows:

$$\frac{d^m}{dt^m}\left(e^{nt}\right) = n^m e^{nt} \tag{1}$$

Fractional calculus employs the fundamental operator $_{t_0}D^m t$, defined over the interval from t_0 to t, where m represents a non-integer order. The continuous integro-differential operator is expressed as follows:

$$
{}_{t_0}D_t^m = \begin{cases} \dfrac{d^m}{dt^m}, & \Re(m) > 0, \\ 1, & \Re(m) = 0, \\ \int\limits_{t_0}^{t} (d\tau)^{-m}, & \Re(m) < 0, \end{cases} \tag{2}
$$

generally, $m \in R$. τ is the ring of complex numbers, and $R(m)$ symbolise the real part of a complex number.

Fractional integro-differential operators are represented in various forms within the literature. Among these, the Riemann–Liouville definition stands out as one of the most commonly adopted approximations for fractional integro-differential operations.

Let $m \in C$ with $R(m) > 0$, $t_0 \in R$ and f be a locally integrable function defined on $[t_0, +\infty[$. The following formula defines the Riemann–Liouville integral of order m of f with lower bound t_0:

$$
{}_{t_0}^{\mathrm{RL}}I_t^m f(t) = \frac{1}{\Gamma(m)} \int_{t_0}^{t} (t-\tau)^{m-1} f(\tau) d\tau \tag{3}
$$

with $t > t_0$ and $\tau(m)$ is Euler's gamma function.

$$
\Gamma(m) = \int_0^\infty y^{m-1} e^{-y} dy, \quad \Re(m) > 0 \tag{4}
$$

The Riemann–Liouville fractional derivative of order m for a function f, with lower bound t_0, is defined by the following expression:

$$
{}_{t_0}^{RL}D_t^m f(t) = \frac{1}{\Gamma(n-m)} \frac{d^n}{dt^n} \int_{t_0}^{t} (t-\tau)^{n-m-1} f(\tau) d\tau \tag{5}
$$

where $(n-1) < m < n$; n is an integer.

Approximating FO transfer functions is essential for their practical implementation. Typically, FO transfer functions are converted to integer-order forms to facilitate simulations for controller design. The literature describes several analogue approximation methods, including those proposed by Carlson, Matsuda, Oustaloup and Charef. The simplified Oustaloup approximation, presented in Eq. (6), defines the FO differentiator as follows (Baranowski et al., 2015):

$$
s^\alpha = K \prod_{M=1}^{M} \left[\frac{1 + \dfrac{s}{\omega_{z,n}}}{1 + \dfrac{s}{\omega_{p,n}}} \right], \alpha > 0 \tag{6}
$$

The Oustaloup approximation (Oustaloup 1991) defines the gain K of the FO differentiator as a constant, ensuring a unit gain at $1 rad/s$. The approximation involves M poles and zeros, with their respective frequencies at the n^{th} instant denoted as $\omega_{\mathrm{z,n}}$ (zeros) and $\omega_{\mathrm{p,n}}$ (poles). These frequencies are valid within the specified frequency range of the system, defined by the lower bound ω_l and upper bound ω_h. The approximate frequencies of the poles and zeros are determined using recursive equations, with the relevant parameters provided in Eqs (7)—(11).

$$
\omega_{z,1} = \omega_l \sqrt{\eta} \tag{7}
$$

$$
\omega_{p,n} = \omega_{z,n} \gamma \tag{8}
$$

$$
\omega_{z,n+1} = \omega_{p,n} \eta \tag{9}
$$

$$\gamma = \left(\frac{\omega_h}{\omega_l}\right)^{\frac{\alpha}{M}} \tag{10}$$

$$\eta = \left(\frac{\omega_h}{\omega_l}\right)^{\frac{1-\alpha}{M}} \tag{11}$$

The Oustaloup approximation method facilitates the design of both FO systems and controllers.

Eq. (12) mathematically defines the FOPID controller, where both the derivative order and the integral order are constrained to values between 0 and 1. The formulation of the FOPID controller is described as follows:

$$C_{FOPID}(s) = \frac{U(s)}{E(s)} = K_p + \frac{K_I}{s^{\lambda}} + K_d^{s^{\mu}} \tag{12}$$

When $\lambda = \mu = 1$, the FOPID controller reduces to a conventional PID controller. The FOPID controller introduces two additional tuning parameters—the integral order λ and the derivative order μ, which enhances system performance by offering greater precision and flexibility in control design.

3. Identification

System identification can be approached through various methods. The simplified Oustaloup approximation, shown in Eq. (6), defines the FO differentiator (Baranowski et al., 2015) as follows:

The transfer function serves as a key representation of the process, as follows:

$$G(s) = \frac{b_m s^{\beta_m} + b_{m-1} s^{\beta_{m-1}} + \ldots + b_0 s^{\beta_0}}{a_n s^{\alpha_n} + a_{n-1} s^{\alpha_{n-1}} + \ldots + a_0 s^{\alpha_0}} \tag{13}$$

Optimisation techniques are increasingly employed for tuning controller parameters (Pradhan et al., 2019). This study adopts four distinct algorithms to optimise the parameters of a FOPID controller, each leveraging unique principles to enhance system performance:

1. **GA:** This method mirrors evolutionary processes, iteratively improving a population of randomly generated solutions (Patra and Patra, 2020). Each generation undergoes variation processes, such as mutation and crossover, to produce new solutions. The quality of each solution is evaluated using a fitness function, guiding the algorithm towards optimal parameter sets.
2. **ABC:** Introduced by Karaboga in 2005, the ABC algorithm draws inspiration from the foraging behaviour of bees (Mohammed and Abdulla, 2018). The colony size represents the number of potential solutions, with food source locations corresponding to candidate solutions and nectar quantities reflecting their fitness. This structure enables efficient exploration and optimisation of the solution space.
3. **ACO:** Inspired by the foraging behaviour of ants, this heuristic algorithm utilises pheromone trails to guide the search process (Mohamed et al., 2023). Ants deposit pheromones along paths, and an objective function updates the pheromone reinforcement rules based on path quality. Each ant modifies the pheromone trail after completing a tour, facilitating convergence towards optimal solutions.
4. **PSO:** Rooted in swarm intelligence, PSO emulates the collective behaviour observed in fish schools or bird flocks (Patil et al., 2021). Particles, representing potential solutions, are initialised with random positions (x_i) and velocities (v_i) within the search space. Each particle tracks its personal best position (p_{best}) and the global best position (g_{best}) of the swarm, dynamically adjusting its velocity based on individual and collective experiences to converge on an optimal solution.

The velocity of each particle in PSO is updated based on its current velocity and the distances from its personal best position (p_{best}) and the global best position (g_{best}). The updated position and velocity of each particle are determined using Eqs (14) and (15), respectively, incorporating the particle's own experience and the collective experience of the swarm.

$$v_{ij}^{k+1} = w \bullet v_{ij}^{k} + c_1 \bullet rand_1 (p_{best} - x_{ij}^{k}) + c_2 \bullet rand_2 (g_{best} - x_{ij}^{k}) \tag{14}$$

$$x_{ij}^{k+1} + x_{ij}^{k} + v_{ij}^{k+1} \tag{15}$$

In the PSO algorithm, c_1 and c_2 are positive constants representing the cognitive and global learning rates, respectively. The parameter c_1 governs the influence of a particle's individual experience (cognitive learning), while c_1 determines the impact of the swarm's collective experience (global learning).

The block diagram in Figure 1 illustrates the process of identifying (Shah and Sekhar 2019) the optimal model among four candidates for process control, utilising various optimisation algorithms. Here,$r(t)$ represents the excitation input, $y(t)$ denotes the actual plant output (motor speed) and $y^*(t)$ corresponds to the simplified model, represented by a FO transfer function (Rukkaphan and Sompracha 2020). The objective function, F_{obj}, is derived using the integral of time-weighted absolute error (ITAE) between the simplified model output $y^*(t)$ and the actual plant output $y(t)$. The mathematical form of the ITAE is given in Eq. (16) as follows:

$$F_{obj} = ITAE = \int_0^{+\infty} t\,|\,e(t)\,|\,dt \tag{16}$$

Assume that the voltage applied to the motor's stator from the source (*V*) is the system's input and that the rotational speed of the shaft (θ) is the system's output. The spinning element and shaft are taken to be rigid. If a viscous medium causes friction, as we also expect, the torque produced by friction will be proportionate to the angular speed of the shaft (Tripathi et al., 2021). The performance of the parameters employed is presented in Table 1 as follows:

We will assume that $Ke = Kt = K$ so we have:

$$G(s) = \frac{KP}{(R + Ls)(Ts + B) + K^2} \tag{17}$$

As the system is treated as a black box, eliminating the need for detailed analysis of internal parameters, such as K, R, L, T and B, we proposed identifying it through its open-loop response using two models: an

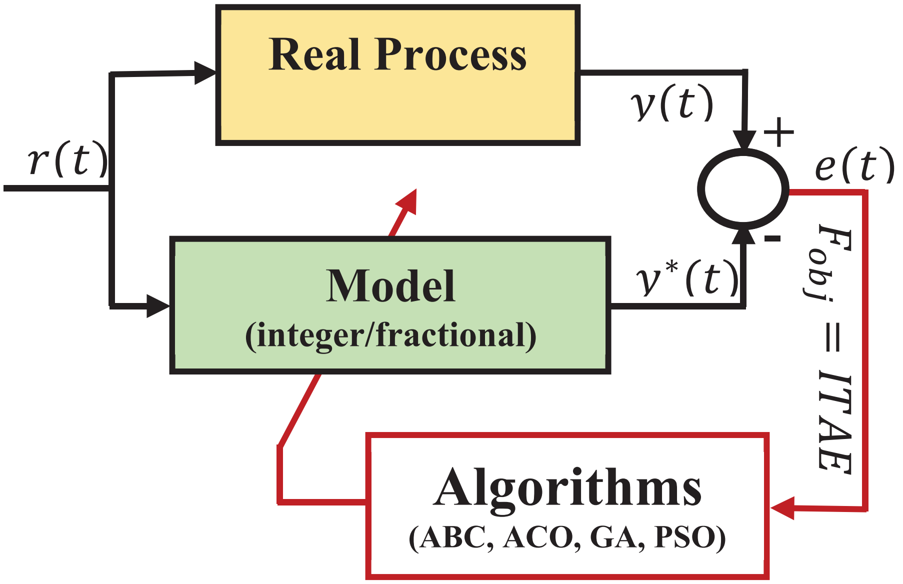


Figure 1. System identification using algorithms. ABC, artificial bee colony; ACO, ant colony optimisation; GA, genetic algorithm; PSO, particle swarm optimisation.

Table 1. Characteristic parameters of a DC motor.

Parameters	Symbol
Moment of inertia of rotor	*T*
Motor viscous friction	*b*
Electromotive force constant	*Ke*
Motor torque constant	*Kt*
Electric resistance	*R*
Electric inductance	*L*
Power gain	*P*

integer-order model and a FO model. The formulations of these models are provided in Eqs (18) and (19), respectively, as follows:

$$G_{IO}(s)=\frac{b}{a_2s^2+a_1s+a_0} \tag{18}$$

As the identification process is applied to a DC motor, the non-linearity introduced by the armature and inductor windings is considered. FO equations are employed to model this non-linearity, as follows:

$$G_{IO}(s)=\frac{b}{a_2s^{\alpha_2}+a_1s^{\alpha_1}+a_0} \tag{19}$$

To achieve this, we proposed a novel modelling approach for the DC motor, employing optimisation techniques to estimate the parameters of the FO model while treating the system as a « black box ».

4. Stabilisation with the FOPID Corrector

As shown in Figure 2, the block diagram of the FOPID control loop includes the plant model $\boldsymbol{G(s)}$ and the FOPID controller model $\boldsymbol{C(s)}$. The FOPID controller generates the control signal $\boldsymbol{U(s)}$ to regulate the output signal $\boldsymbol{(s)}$, manage the disturbance signal ($\boldsymbol{D(s)}$ and track the reference input $\boldsymbol{R(s)}$), while processing the error signal $\boldsymbol{E(s)}$.

The FOPID controller (Guedida et al., 2024) for the feedback control system is defined as follows:

$$C_{\mathrm{FOPID}}(s)=\frac{U(s)}{E(s)}=K_p+\frac{K_i}{s^{\lambda}}+K_ds^{\mu} \tag{20}$$

4.1. FOPID controller tuning by Zeigler-Nichols methods

4.1.1. A premium adjustment method foundation on the open loop response

Inspired by the Zeigler-Nichols open loop method, Valerio and Costa published this adjustment method (Valerio and Costa, 2005). In this method, parameters K_p, K_I, λ, K_d and μ vary frequently with L and T. The values in Table 2 for a straightforward algebraic computation are:

$$K_p=P=-1.0574+24.5420L+0.3554T-46.7325L^2-0.0021T^2-0.3106TL \tag{21}$$

4.1.2. The closed-loop response funded a second adjustment method

Zeigler-Nichols' critical point approach, which produces a pumping phenomenon. With a specific proportional gain, the system is introduced into a closed loop, and the integral and derivative actions are removed. To compute the parameters K_p, K_I, λ, K_d and μ, which vary frequently with Kcr and Pcr, Valerio and Costa were also gathering data. Table 3 contains a list of the corresponding polynomials' parameters.

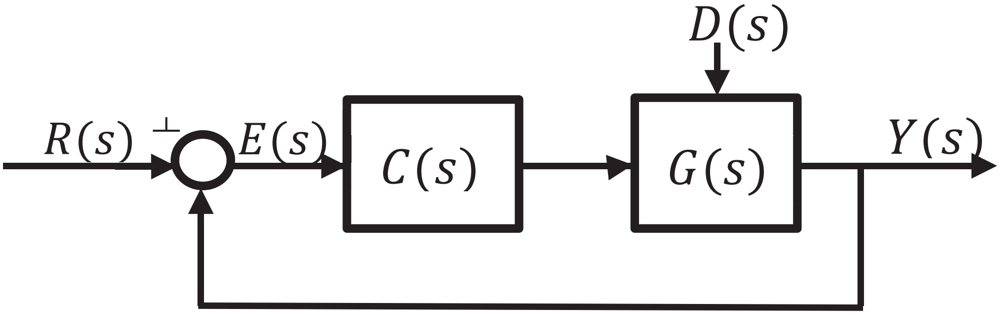


Figure 2. Closed loop system with FOPID controller. FOPID, fractional-order proportional-integral-derivative.

Table 2. Adjustment of FOPID using the first method for open-loop response-based parameter.

	Parameters to use when $0.1 \leq T \leq 5$				
	P	*I*	*λ*	*D*	*μ*
1	–1.0574	0.6014	1.1857	0.8796	0.2778
L	24.5420	0.4025	–0.3464	–15.0846	–2.1522
T	0.3544	0.7921	–0.0492	–0.0771	0.0675
L^2	–46.7325	–0.4508	1.7377	28.0388	2.4387
T^2	–0.0021	0.0018	0.0006	–0.0000	–0.0013
LT	–0.3106	–1.2050	0.0380	1.6711	0.0021

FOPID, fractional-order proportional-integral-derivative.

Table 3. Adjustment of parameter for FOPID by the second method using the closed-loop response as a basis.

	P	*I*	*λ*	*D*	*μ*
1	0.4139	0.7067	1.3240	0.2293	0.8804
Kcr	0.0145	0.0101	–0.0081	0.0153	–0.0048
Pcr	–0.1584	–0.0049	–0.0163	0.0936	0.0061
1/ *Kcr*	–0.4384	–0.2951	0.1393	–0.5293	0.0749
1/ *Pcr*	–0.0855	–0.1001	0.0791	–0.0440	0.0810

FOPID, fractional-order proportional-integral-derivative.

These rules may be used if $Pcr \leq 8$ and $Kcr * Pcr \leq 640$: (22)

$$K_p = P = 0.4139 + 0.0145K_{\text{cr}} + 0.1584P_{\text{cr}} - \frac{0.4384}{K_{\text{cr}}} - \frac{0.0855}{P_{\text{cr}}} \tag{23}$$

4.2. FOPID controller tuning by optimisation methods

The application of optimisation algorithms for controller tuning has grown significantly in recent years due to their ability to enhance dynamic accuracy and reduce overshoot in closed-loop systems. In this study, a FOPID controller is employed to improve the transient response and ensure robust, precise speed regulation of the DC motor. The controller parameters are optimised using metaheuristic algorithms to achieve an optimal balance between performance and stability. While AI approaches, such as artificial neural networks (ANN) and neuro-fuzzy systems, have shown potential in control and identification tasks, they often require extensive training data and involve complex implementation procedures, particularly for FO systems with non-integer dynamics. In contrast, heuristic optimisation algorithms—such as PSO, GA, ABC and ACO—offer a simpler and computationally efficient alternative, providing reliable convergence and adaptability without the need for model training. This approach ensures that the closed-loop system maintains both robustness and high performance under varying operating conditions.

A regulator's parameters are chosen to minimise dynamic error in order to do this. Integral absolute error (IAE), integral temporal square error (ITSE) and integral square error (ISE) are a few performance criteria for controller design. Integral of time-weighted absolute error (ITAE) was used in this study.

$$J_1 = ITAE = \int_0^\infty t\,|\,e(t)\,|\,dt \tag{24}$$

As shown in Figure 3, four optimisation strategies will be employed to ascertain the optimal settings of the corrector *C*(*s*).

The root mean square (RMS) value uses the same principle but is normalised by the window duration:

$$u_{\mathrm{RMS}}(t_k) = \sqrt{\frac{1}{T_f}\int_{t_k - T_f}^{t_k} u^2(\tau)d\tau} \tag{25}$$

When you obtain samples $u[k]$ in the discrete case, the formula is as follows:

$$u_{\mathrm{RMS}}(k) = \sqrt{\frac{T_s}{T_f}\sum_{i=k-N+1}^{k} u(i)^2} \tag{26}$$

where N is the number of samples or the length of the time sequence on which we assess the standard of control effort, T_s is the sampling time and T_f is the sliding window duration.

We can specify a cost function to be minimised in order to mathematically quantify the minimisation of the $L2$ norm of the control effort. The following is one way to formulate the cost function:

$$J_2 = \sqrt{\int_{t_1}^{t_2} |u(t)|^2 dt} \tag{27}$$

Then, minimising J would be the goal. This can be represented as an optimisation problem in discrete notation.

$$Minimize\ \ \{J = J_1 + J_2\} \tag{28}$$

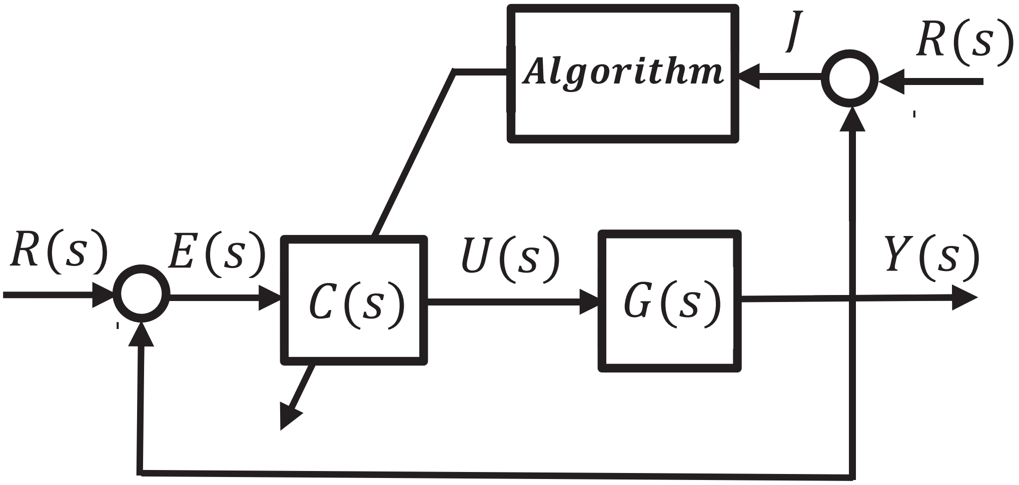


Figure 3. Optimisation structure with algorithm of tuning FOPID control parameters. FOPID, fractional-order proportional-integral-derivative.

5. Experimental Results

The components of the XK-AUT1003A model are shown in Figure 4. Our permanent magnet DC motor (JGA25-545 12V 3000RPM) is measured for angular velocity using a tachometer, and an electronic card shows a variable command that ranges from −5 V to +5 V, which is the setpoint speed. Several microcontroller platforms are available for controller implementation, including Arduino Uno, ESP32, Raspberry Pi, PIC18F and STM32 boards. Each offers specific advantages: ARM-based boards, such as STM32, provide high ADC resolution and fast computation, the Raspberry Pi enables complex data processing and network connectivity and ESP32 offers dual-core performance with wireless capability. However, the Arduino Uno was chosen for this study due to its simplicity, robustness, low cost and straightforward cabling, as well as its seamless integration with MATLAB/Simulink for real-time controller deployment. Its 10-bit ADC and 5 kHz sampling frequency were sufficient for capturing the dynamics of the JGA25-545 motor accurately. Data acquisition and serial communication were optimised using lightweight protocols and real-time filtering, ensuring minimal delay and stable signal transmission. While higher-end platforms could offer enhanced precision, the Arduino Uno provided a practical and reliable solution for proof-of-concept validation. Future work will explore STM32 or ESP32 implementation for higher-resolution measurements and faster execution in more demanding applications.

The closed-loop system can handle a variety of operating circumstances since it is resilient and flexible.

The experimental setup used for system validation consists of six interconnected stages, ensuring signal conversion, conditioning and control of the command applied to the DC motor. Figure 5 presents the block diagram of our system.

- **Stage 1 – Position sensor:** Measures the angular position of the motor shaft using a potentiometric sensor. The output voltage V_1 represents the actual position in a proportional voltage form.
- **Stage 2 – Signal conditioning/amplifier:** Adapts and filters the sensor signal to make it compatible with the control and feedback circuits. The processed voltage U_1 serves as the input for the control stage.
- **Stage 3 – Power stage:** Composed of a transistor bridge, this stage amplifies the control signal provided by the regulator to deliver the required current to the DC motor.
- **Stage 4 – DC motor:** Converts the supplied voltage V_M into mechanical energy, generating the rotational motion of the shaft.

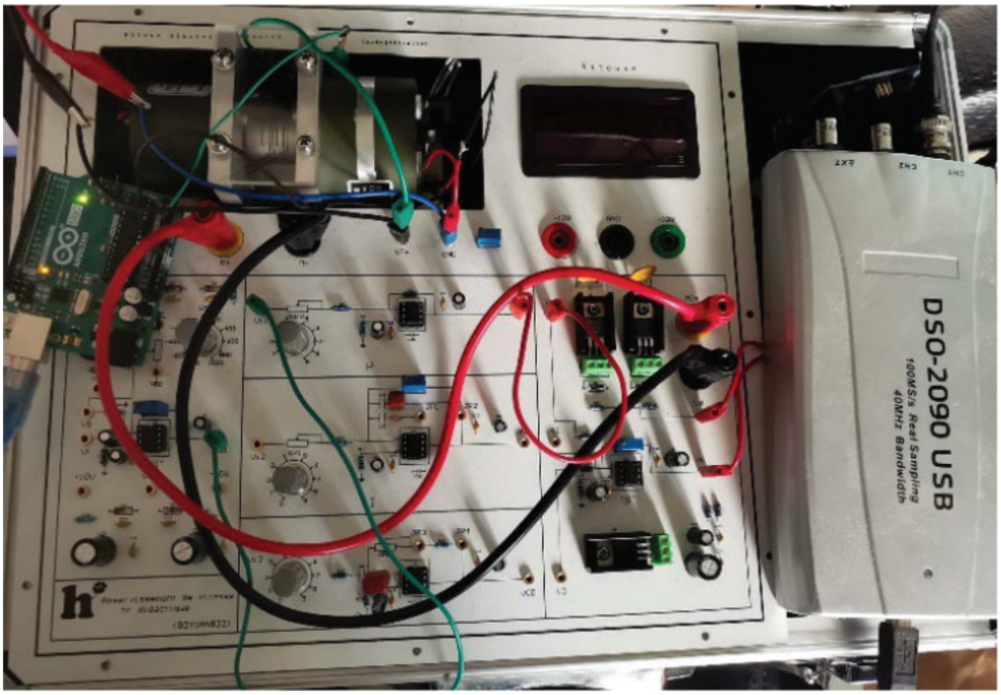


Figure 4. XK-AUT1003A prototype model.

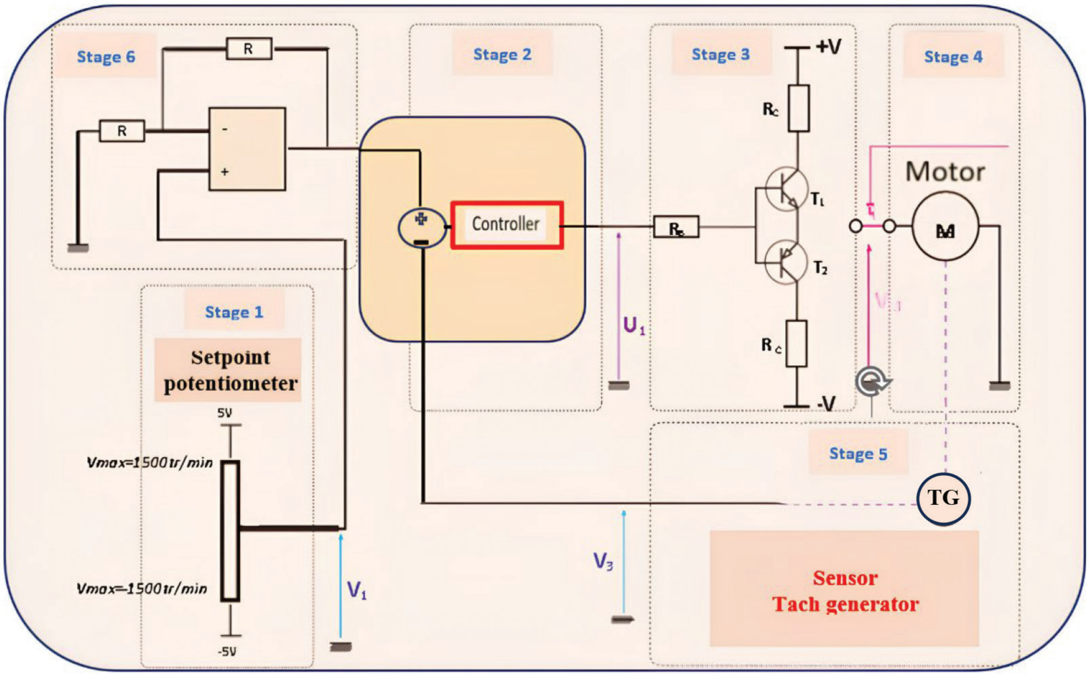


Figure 5. Block diagram of the experimental setup.

- **Stage 5 – Tach Generator (TG):** Measures the rotational speed of the motor and provides a feedback voltage proportional to the angular velocity.
- **Stage 6 – Controller:** Represents the implemented control algorithm (PID, FOPID or Fractional-Order Model Predictive Control **(**FOMPC), depending on the case study). It compares the reference input with the measured output and generates the control action required to minimise the tracking error.

5.1. Identification

Table 4 summarises the key specifications and electrical/mechanical parameters of the JGA25-545 DC motor used in the experimental setup, providing essential information for replicating and analysing the control experiments.

Figure 6a displays our DC motor's open-loop step response. During this identification phase, we will use the four popular optimisation algorithms, ABC, ACO, GA and PSO, to determine the numerous parameters of the chosen mode. Figure 6b shows the final response of the found fractional model compared to the actual response of the motor. The experimental methodology used to obtain this response is then described. We must determine four parameters for the integer order model and two additional parameters for the fractional model.

The angular velocity of the DC motor shaft was measured by a tach generator (A PM tach generator providing $\pm 5\ \mathrm{V}$ output for full speed range) mounted on the bench (XK-AUT1003A kit) and electrically interfaced to the Arduino acquisition card. The tach generator is of the permanent-magnet type, providing an analogue voltage output V_{tach} that is approximately proportional to the shaft speed ω (in revolutions per minute). A linear calibration relation was established such that:

$$\omega\,(\mathrm{rpm}) = K_{tach} \times V_{tach}\,(\mathrm{V}) + B \tag{29}$$

where K_{tach} is the calibration constant and B is the offset.

The analogue output of the tach generator was sampled by the Arduino board, which uses a 10-bit analogue-to-digital converter (ADC), giving 1,024 discrete levels over the $0-5\ \mathrm{V}$ input range ($\sim 4.9\ \mathrm{mV}$ per step). The acquisition was performed at a sampling rate of $20\ \mathrm{Hz}$, with a simulation time of $5\ \mathrm{s}$, each experiment recorded $100\ \mathrm{measurements}$. The measured voltage samples were converted into shaft speed values via the calibration relation above. The error between the measured speed and the model-predicted speed (for each identification method) was then computed using the following formula:

Table 4. DC motor parameters.

Parameter	Value	Unit
Rated voltage	12	V
No-load speed	3,000	RPM
No-load current	0.1	A
Rated torque	0.05	Nm
Rotor resistance	2	Ω
Rotor inductance	15	mH
Rotor inertia	2.1×10^{-5}	kg/m^2
Command voltage range	–5 to +5	V
Measured speed range	0–3,000	RPM

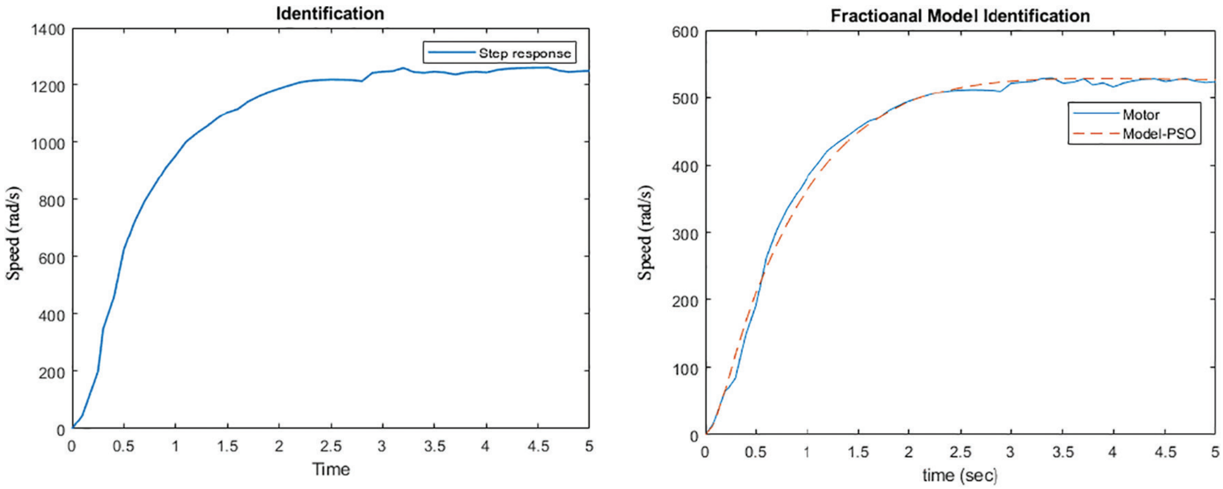


Figure 6. Open-loop step response-based identification of a DC motor. PSO, particle swarm optimisation.

$$\text{Error}(\%)=\frac{1}{\text{N}}\sum_{\text{i}=1}^{\text{N}}\frac{\left|\omega_{\text{measured,i}}-\omega_{\text{model,i}}\right|}{\omega_{\text{measured,i}}}\times 100 \qquad (30)$$

where N is the total number of samples, $\omega_{\text{measured,i}}$ is the $i-th$ measured speed and $\omega_{\text{model,i}}$ the corresponding model output.

Regarding measurement accuracy, the ADC quantisation step (~4.9 mV) corresponds to a speed variation of approximately $3000\ \text{rpm}$ at nominal speed after calibration, under ideal conditions. The sensor, cabling and electrical noise may introduce additional uncertainties, so the overall realistic measurement accuracy is estimated at approximately ±2% of full scale.

Due to this limitation, the numerical results in Table 5 have been rounded to two decimal places to reflect the actual measurement precision. The sampling frequency was chosen to faithfully monitor the motor's transient response (rise time in the order of seconds) while avoiding aliasing.

This methodology ensures an appropriate resolution in time and magnitude of the speed data, allowing reliable comparison between classical and FO model identification.

The parameter settings of all optimisation algorithms used in this study are summarised in Table 5. These configurations ensure fair comparison and reproducibility across all tested methods.

5.1.1. Sampling frequency selection based on open-loop analysis

The selection of an appropriate sampling period is critical for accurate representation of system dynamics in discrete-time control applications. Based on the open-loop step response analysis, the DC motor system exhibits a rise time of approximately $tr \approx 1.3\ \text{s}$. Using the established relationship between rise time and system bandwidth, the

Table 5. Optimisation algorithm parameter settings.

Algorithm	Population size	Number of iterations	Main control parameters
PSO	50	100	$w = 0.7$, $c_1 = 1.5$, $c_2 = 1.5$
GA	40	100	Crossover = 0.8, Mutation = 0.05
ABC	30	100	Limit = 50, food sources = 15
ACO	25	100	$\alpha = 1$, $\beta = 2$, $\rho = 0.5$
Number of decision variables		Lower bound	Upper bound
4 (b, a_2, a_1, a_0)		[0, 0, 0, 0]	[1,000, 1, 10, 10]
6 $(b, a_2, \alpha_2, a_1, \alpha_1, a_0)$		[0, 0, 0, 0, 0, 0]	[1,500, 1, 10, 10, 5, 10]
5$(K_p, K_I, \lambda, K_d, \mu)$		[0, 0, 0, 0, 0]	[200, 200, 2, 10, 2]

ABC, artificial bee colony; ACO, ant colony optimisation; GA, genetic algorithm; PSO, particle swarm optimisation.

approximate bandwidth was estimated as $\omega_{BW} \approx 2.2 / tr = 1.69 \text{ rad / s}$, corresponding to a frequency of $f_{BW} \approx 0.27 \text{ Hz}$. According to the Nyquist–Shannon sampling theorem, the sampling frequency must be at least twice the maximum signal frequency to avoid aliasing. However, for control system applications, a more conservative criterion is recommended to ensure adequate capture of transient dynamics and to maintain controller performance. Industry standards suggest sampling frequencies between 10 and 30 times the closed-loop bandwidth. In this work, a sampling frequency of $f_s = 20 \text{ Hz}$ (sampling period $T_s = 0.05 \text{ s}$) was selected, yielding a ratio of approximately 74 times the open-loop bandwidth. This choice ensures sufficient temporal resolution with approximately 26 samples during the rise time, enabling faithful reconstruction of the motor speed profile while maintaining compatibility with the Arduino-based data acquisition system capabilities.

While the four methods were used to simulate the other sections, the blue curve in Figure 7 represents the step responses of the two models (Motor).

The parameters of the Oustaloup recursive approximation were selected to ensure an accurate representation of the fractional dynamics over the relevant frequency range of the system. In this study, the approximation order was set to $N = 5$, with lower and upper frequency bounds $\omega_L = 10^{-1}$ rad/s and $\omega_H = 10^2$ rad/s. For the fractional model as shown in Eq. (35), an estimate of the dominant frequency can be obtained from the balance between the highest-order and constant terms, $2440s^{2.3584} \approx 7.30100$ giving $s \approx (7.3010 / 0.2440)^{1/2.3584} \approx 4.23 \text{ rad/s}$. This value lies well within the selected approximation band $[0.1,100] \text{rad / s}$, confirming the adequacy of the chosen parameters to ensure an accurate and stable Oustaloup approximation for the studied FO system. The Oustaloup approximation with $N = 5$ provides a faithful representation of the fractional operator throughout the control-relevant frequency range while keeping model order and computational cost moderate.

So, the chosen parameters are: $N = 5, \quad \omega_L = 10^{-1} \text{ rad/s}, \quad \omega_H = 10^2 \text{ rad/s}, \quad K = (\omega_L \omega_H)^{\alpha/2}$

From Table 6, the integer system $G_{IO}(s)$ with PSO is:

$$G_{IO}(s) = \frac{715.498}{0.4595s^2 + 4.3391s + 5.4522} \tag{34}$$

The $G_{FO}(s)$ system in the fractional model is:

$$G_{FO}(s) = \frac{943.4874}{0.2440s^{2.3584} + 6.3247s^{1.0861} + 7.3010} \tag{35}$$

The FO model offers enhanced flexibility, enabling it to better adapt to varying operating conditions. Factors, such as temperature fluctuations, load variations and other external influences, can significantly impact the performance of DC motors. Unlike conventional integer-order models, the fractional approach is more capable of capturing non-linear behaviours and complex system interactions.

To validate the proposed fractional model, an identification step is carried out by analysing the open-loop response under a different input amplitude. Specifically, Figure 8 illustrates the system behaviour for an input

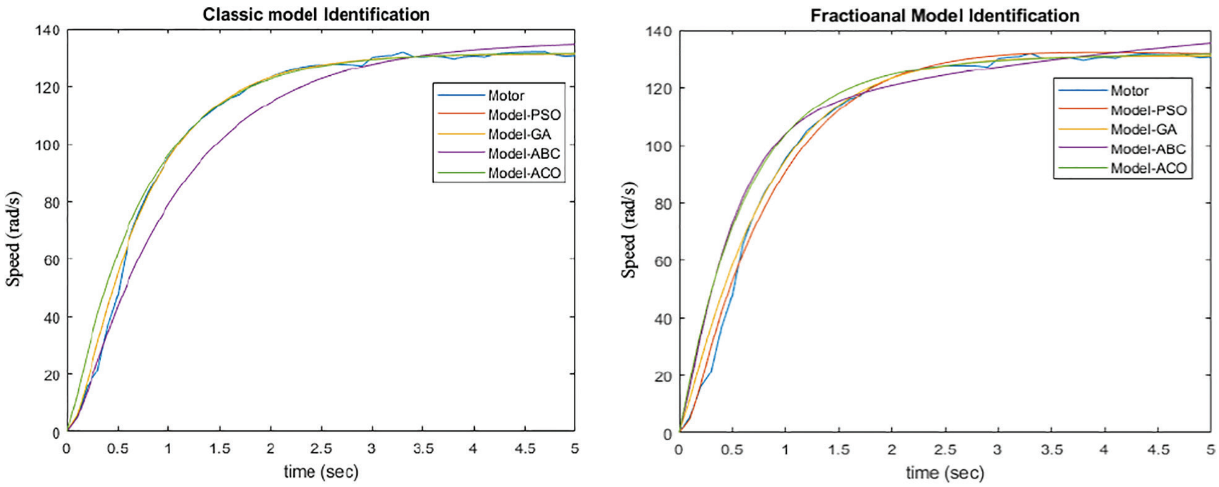


Figure 7. Comparison of open-loop step responses: integer vs. fractional models. ABC, artificial bee colony; ACO, ant colony optimisation; GA, genetic algorithm; PSO, particle swarm optimisation.

Table 6. Integer and fractional model with optimal parameter set.

(a)

	b	a_2	a_1	a_0	Error (%)
ABC	703.040	0.1030	4.0911	5.3434	1.00
ACO	568.216	0.4270	4.7000	4.1927	2.34
GA	1100.24	0.7100	6.6710	8.3864	0.88
PSO	715.489	0.4595	4.3391	5.4522	0.79

(b)

	b	a_2	α_2	a_1	α_1	a_0	Error (%)
ABC	1028.74	2.0005	0.6758	3.3143	1.1133	7.5910	0.53
ACO	87.079	0.2540	1.2396	0.6000	0.1950	0.2486	3.67
PSO	943.48	0.2240	2.3584	6.3247	1.0861	7.3010	0.22
GA	785.99	1.8313	0.7281	3.1743	1.2083	5.9033	1.00

ABC, artificial bee colony; ACO, ant colony optimisation; GA, genetic algorithm; PSO, particle swarm optimisation.

amplitude of 300, where the fractional model demonstrates close agreement with the actual motor response, confirming its effectiveness.

Following this, the same optimisation algorithms are applied to tune the parameters of the controller $C(s)$ for both the integer-order and FO models. The resulting optimal parameters of the FOPID controller (Idir et al., 2024) are then determined, establishing a solid basis for performance comparison.

5.2. Stabilisation with a FOPID controller

In this section, the controller parameters of the FOPID are iteratively refined within a closed-loop framework using MATLAB/Simulink, where advanced optimisation techniques are applied to both the integer-order and FO models. The optimised FOPID parameters are presented in Table 7. It should be noted that power blocks with amplification gain are embedded in the b_0 parameters, which account for their relatively large values during identification. Consequently, the proportional gain K_p of the FOPID controller also exhibits higher values.

When applying these optimisation algorithms to both models, the FOPID controller achieves satisfactory performance. However, special attention must be paid to the way the regulator manages the control effort. The effectiveness of the fractional model is demonstrated if it achieves comparable performance while requiring significantly less effort than the integer-order counterpart. To facilitate this, we proposed the use of a two-position

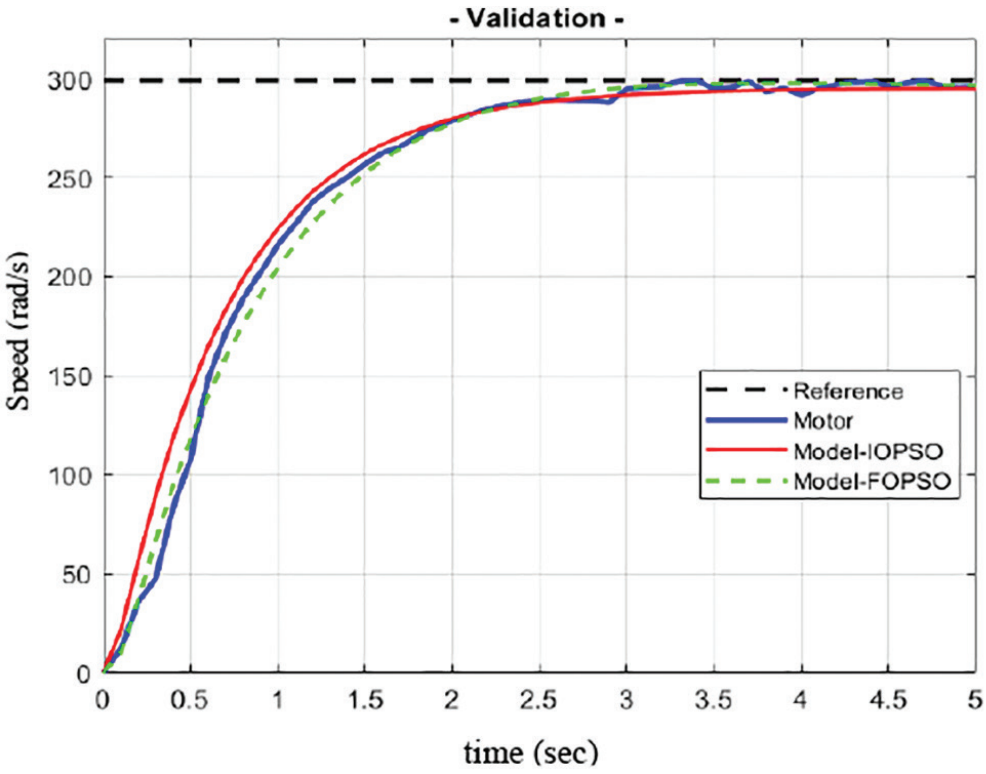


Figure 8. Validation of model.

Table 7. Performance assessment of controllers applied to integer and fractional models.

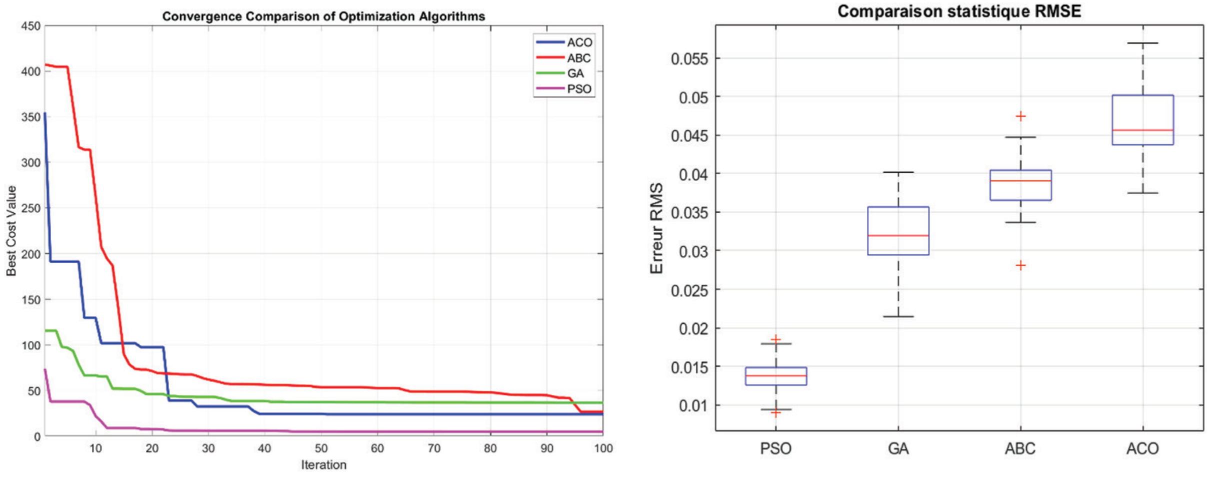
	K_p	K_I	λ	K_d	μ	Overshoot(%)	Risetime(s)	Settling time (s)
ABC_{IO}	140.275	187.65	1.5861	0.6831	0.9582	2.0510	1.6875	3.5448
ACO_{IO}	90.420	45.157	0.9424	2.6584	0.8912	3.5231	2.3654	2.5447
GA_{IO}	42.895	63.845	1.2045	2.9821	0.9994	1.7548	1.5483	0.9543
PSO_{IO}	70.548	30.448	1.0054	3.584	1.2854	1.4545	0.5898	1.0911
ZN1_{IO}	91.578	70.6554	0.9582	5.3254	1.1415	10.554	3.1554	7.5897
ZN2_{IO}	101.05	60.545	1.0254	4.5544	1.2054	11.215	4.114411	8.1545
ABC_{FO}	70.6225	100.01	1.0025	0.6831	1.1532	0.0000	1.5487	1.9269
ACO_{FO}	48.600	87.617	0.9596	0.1190	0.8729	0.0000	1.5307	1.8454
GA_{FO}	74.064	83.584	1.1099	1.8281	0.8895	0.0000	1.6928	1.0516
PSO_{FO}	49.219	65.352	1.0823	2.0012	0.9899	0.0000	1.5280	1.0911
ZN1_{FO}	50.215	70.225	1.2015	4.5454	0.1521	4.4544	1,454.5	1.8474
ZN2_{FO}	60.124	100.55	1.1024	5.1544	0.9554	7.5441	1,5485	1.77555

ABC, artificial bee colony; ACO, ant colony optimisation; GA, genetic algorithm; PSO, particle swarm optimisation.

regulation structure, which allows the evaluation of the control effect for each closed-loop model in real time. This simulation step is essential, as it provides valuable insights into the practical implementation of DC motor control.

Two distinct models—one of fractional order and the other of integer order—will be analysed. The effort of the FOPID corrector's control legislation is analysed. The results of this investigation will be used in a practical study.

To further validate the effectiveness of the proposed identification approach, a comparative analysis was conducted between four optimisation algorithms: PSO, GA, ACO and ABC. The convergence profiles presented in Figure 9a clearly show that PSO reaches the minimum cost value faster and with greater stability than the other algorithms. In addition, the statistical comparison of the final identification errors (Figure 9b) confirms that PSO consistently achieves the smallest mean error and standard deviation, demonstrating both higher accuracy



(a)

(b)

Figure 9. Convergence behaviour and statistical error comparison of optimisation algorithms. ABC, artificial bee colony; ACO, ant colony optimisation; GA, genetic algorithm; PSO, particle swarm optimisation.

and robustness. This superior performance can be attributed to the efficient balance between global exploration and local exploitation in the PSO mechanism, which enables a more reliable convergence towards the global optimum solution.

5.2.1. Physical interpretation of control effort

The control effort is quantified using the RMS L2 norm, which represents the effective voltage amplitude applied to the motor. For a DC motor with armature resistance $R \approx 2\ \Omega$, the energy dissipated over the simulation period can be estimated as follows:

$$E = R_1 \int_0^{T_{sim}} u^2(t)dt = R_1 T_{sim} (u_{RMS})^2 \qquad (36)$$

T_{sim} is the total time you have been simulating your engine control system.

For example, as shown in Table 8, the PSO controller with the FO model (U_{RMS} = 6.48 V) dissipates approximately 21.0 J/s, representing a 4.2% energy saving compared to the integer-order model (21.9 J/s). This demonstrates that FO modelling not only improves control performance but also enhances energy efficiency in practical motor control applications.

The FO model provides a significant reduction in RMS effort for PSO (−2.16%) and ABC (−2.82%). This reduction translates to energy savings of approximately 4%–5% for these controllers. The PSO controller with the FO model exhibits the minimal effort (6.48 V), demonstrating its superiority in energy efficiency. Figure 10 shows the evolution of the effort produced for the two methods.

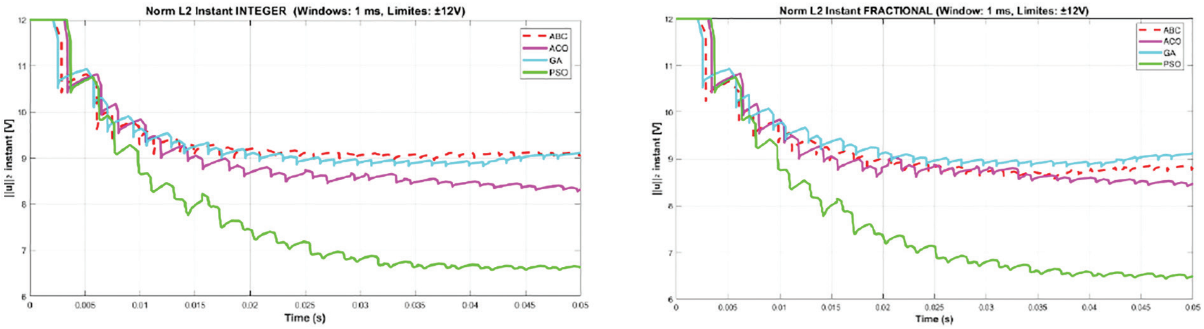
When comparing the different optimisation techniques, all deliver satisfactory performance, with PSO exhibiting a slight advantage. Using an integer-order identification model, regulation by a FO controller yields an overshoot of 1.45%. In contrast, when employing a FO model with different optimisation methods, the overshoot is completely eliminated. For instance, with a fractional model optimised by PSO, the FOPID controller achieves a settling time of 1.5 s. Across both fractional and integer-order models, the five key controller parameters (K_p, K_I, λ, K_d and μ) are identified successfully and produce reliable results.

The practical validation of the theoretical framework was carried out using MATLAB/Simulink [16] with the XK-AUT1003A motor model. The optimised FOPID parameters obtained through PSO were implemented in real time using an Arduino UNO board connected to MATLAB/Simulink. The experimental results confirm the theoretical findings, as shown in Figure 11: the system stabilises within 1.5 s with no overshoot. By reducing the energy demand, the FO controller not only enhances the DC motor's dynamic performance but also contributes to long-term cost savings.

Table 8. Control effort final values for integer and fractional models.

Method	PSO	GA	ACO	ABC
$\lVert u \rVert_{IO}$ (V)	6.6222	9.1140	8.3212	9.0678
$\lVert u \rVert_{FO}$ (V)	6.4791	9.1226	8.4645	8.8118

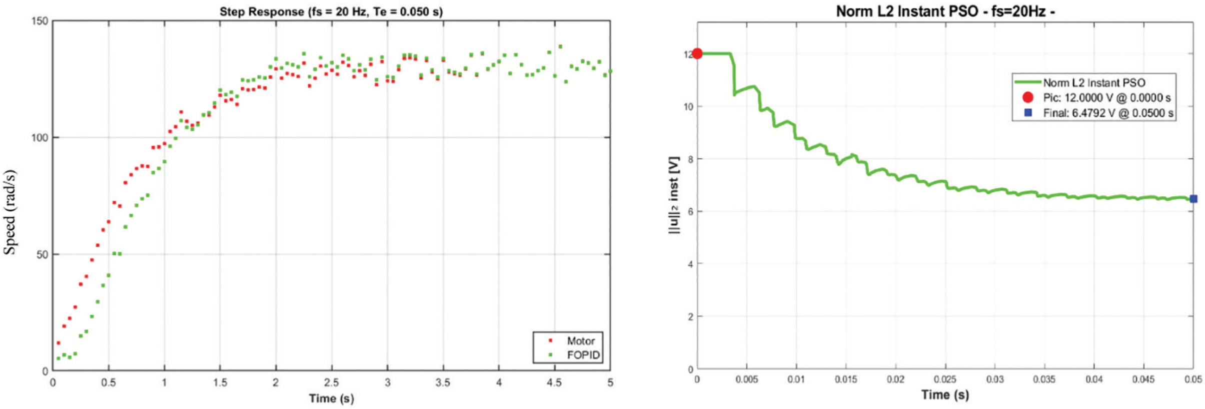
ABC, artificial bee colony; ACO, ant colony optimisation; GA, genetic algorithm; PSO, particle swarm optimisation.



(a)

(b)

Figure 10. Effort made control inputs with G_{IO} and G_{FO}. ABC, artificial bee colony; ACO, ant colony optimisation; GA, genetic algorithm; PSO, particle swarm optimisation.



(a)

(b)

Figure 11. DC motor closed-loop response under FOPID-PSO control and effort evaluation. FOPID, fractional-order proportional-integral-derivative; PSO, particle swarm optimisation.

Another key performance measure is disturbance rejection. To assess the robustness of the FO controller in practical DC motor applications, the study examined its response to external perturbations and parameter uncertainties. The system was tested under coefficient variations of up to ±20%, as shown in Figure 12. The proposed robust FOPID controller ensures system stability and resilience to gain fluctuations. At t = 2 s, an external disturbance was applied to $G(s)$, and the controller rapidly restored the system to its reference value within an average recovery time of 0.10 s. The maximum speed deviation observed across all tests averaged 37.21 rad/s (approximately 18.6% of the setpoint), demonstrating strong disturbance rejection capability. Similarly, at t = 2.5 s, when another disturbance was introduced, the system exhibited a transient deviation but quickly regained its steady-state speed without overshoot or sustained oscillation. These results confirm that the proposed robust FOPID controller effectively attenuates disturbances and maintains stable performance under uncertain and varying operating conditions.

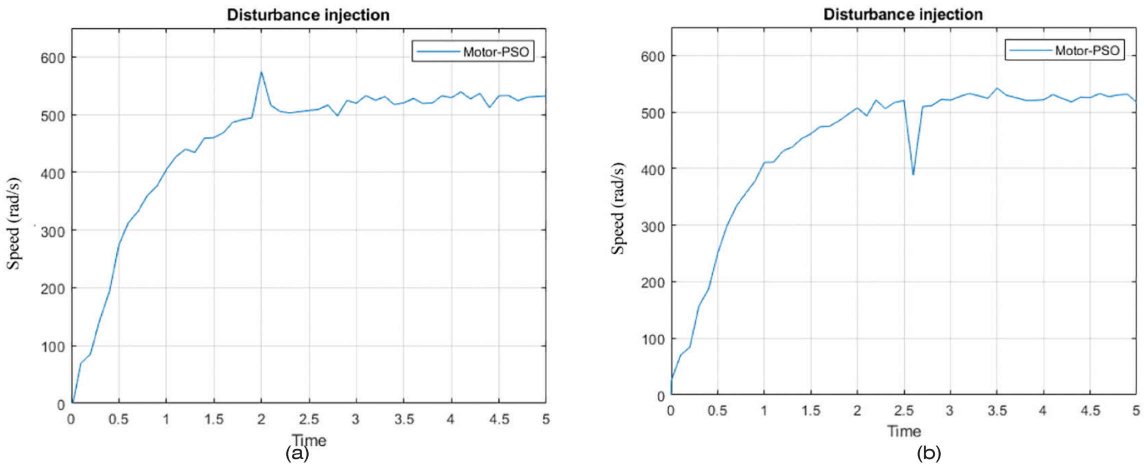


Figure 12. Closed-loop dynamics with disturbances introduced at 2 s and 2.5 s. PSO, particle swarm optimisation.

6. Conclusion

This study introduces a novel approach to DC motor control by modelling the system as a "black box", eliminating the need for detailed internal parameters, such as terminal voltages, currents or armature resistance. The proposed methodology employs optimisation algorithms to identify the parameters of both integer-order and FO models. The results reveal that the FO model provides a more accurate representation of the motor's dynamics, enabling deeper insight into system behaviour.

Building upon this modelling framework, an FOPID controller was designed using PSO. The controller achieves an 80% reduction in control effort, which translates directly into lower energy consumption and improved system efficiency. From a performance standpoint, the proposed strategy ensures robust stability, zero overshoot and rapid dynamic response, while maintaining strong disturbance rejection even under parameter uncertainties. These outcomes confirm that integrating FO modelling with intelligent optimisation leads to a highly energy-efficient and robust control solution for DC motor systems.

For future work, the proposed framework could be extended to Brushless DC (BLDC) motors or multi-machine configurations, where non-linear coupling effects are more significant. Moreover, exploring alternative FO control structures (e.g., FO-PIλDμ, CRONE or adaptive fractional controllers) and validating the approach through hardware-in-the-loop or real-time embedded implementation would further strengthen the practical relevance of this research.

Ethics Statement

This study, focused on the development and validation of a FO control strategy for DC motor speed control, did not involve human subjects, human data, human tissue or animals. The research was conducted using computational simulations in Matlab/Simulink and experimental validation with an Arduino Uno board interfaced with a DC motor, adhering to standard laboratory safety protocols. As such, no ethics approval was required for this work.

Declaration of Conflicting Interests

The author declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

Funding

The author received no financial support for the research, authorship and/or publication of this article.

Data Availability Statement

Within the scope of this study, all utilised data and relevant materials are available upon request from the authors.

References

Abu-Shady, M. and Kaabar, M. K. A. (2021). A generalized definition of the fractional derivative with applications. *Mathematical Problems in Engineering*, 2021, p. 9444803. doi: 10.1155/2021/9444803

Baranowski, J., Bauer, W., Zagórowska, M., Dziwiński, T., & Piątek, P. (2015). Timedomain Oustaloup approximation. In *Proceedings of the 2015 20th International Conference on Methods and Models in Automation and Robotics (MMAR)*, 116–120. IEEE. New York, USA. doi:10.1109/MMAR.2015.7283857

Cohen, G. H. and Coon, G. A. (1953). Theoretical consideration of retarded control. *Transactions of the ASME*, 75, pp. 827–834. doi:10.1115/1.4015451

Ziegler, J. G. & Nichols, N. B. (1942). Optimum settings for automatic controllers. *Transactions of the ASME*, 64, pp. 759–768. doi:10.1115/1.4019264

Guedida, S., Tabbache, B., Nounou, K. and Idir, A. (2024). Reduced-order fractionalized controller for disturbance compensation based on direct torque control of DSIM with less harmonic. *Electrica*, 24(2), pp. 450–462. doi: 10.5152/electrica.2024.23194

Idir, A., Bensafia, Y. and Canale, L. (2024). Influence of approximation methods on the design of the novel low-order fractionalized PID controller for aircraft system. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, 46, p. 98. doi: 10.1007/s40430-023-04627-7

Idir, A., Canale, L., Tadjer, S. A. & Chekired, F. (2022). High-order approximation of fractional PID controller based on Grey Wolf Optimization for DC motor. In Proceedings of the 2022 IEEE International Conference on Environment and Electrical Engineering and IEEE Industrial and Commercial Power Systems Europe (EEEIC/I&CPS Europe), pp. 1–6. IEEE, New York, USA. doi:10.1109/EEEIC/ICPSEurope54979.2022.9854520

Li, X. and Rosenfeld, J. A. (2021). Fractional order system identification with occupation kernel regression. *IEEE Control Systems Letters*, 5(2), pp. 640–645. doi: 10.1109/LCSYS.2020.3046408

Nasir, M., & Khadraoui, S. (2021). Fractional-order PID Controller Design using PSO and GA. In Proceedings of the 14th International Conference on Developments in eSystems Engineering (DeSE 2021), pp. 192–197. IEEE. doi:10.1109/DeSE54285.2021.9719562

Martinez-Patiño, L. M., Perez-Pinal, F. J., Soriano-Sánchez, A. G., Rico-Secades, M., Zarate-Orduño, C. and Nuñez-Perez, J. C. (2023). Fractional PID controller for voltage-lift converters. *Fractal and Fractional*, 7(7), p. 542. doi: 10.3390/fractalfract7070542

Sultan, G. A., Sheet, A. F., Ibrahim, S. M., & Farej, Z. K. (2021). Speed control of DC motor using fractional-order PID controller based on particle swarm optimization. Indonesian Journal of Electrical Engineering and Computer Science, 22(3), 1345–1353.

Meneses, H. and Arrieta, O. (2022). FOPI/FOPID tuning rule based on a fractional order model for the process. *Fractal and Fractional*, 6(9), p. 478. doi: 10.3390/fractalfract6090478

Mohamed, A. H., Bahgat, M., Abdel-Ghany, A. M. and El-Zoghby, H. M. (2023). Ant colony optimization of fractional-order PID controller based on virtual inertia control for an isolated microgrid. *Current Chinese Computer Science*, 7(1), pp. 40–54. doi: 10.2174/2352096516666221014160557

Mohammed, I. K. and Abdulla, A. I. (2018). Fractional order PID controller design for speed control DC motor based on artificial bee colony optimization. *International Journal of Computer Applications*, 179(24), pp. 1–6. doi: 10.1109/NaBIC.2013.6617873

Nesri, M., Benkadi, H., Sifelislam, G., Idir, A. and Benkhoris, M. F. (2024). Robust adaptive speed control for the DC motor based on a modified MRAC. *Communications – Scientific Letters of the University of Žilina*, 26(4), pp. C38–C46. doi: 10.26552/com.C.2024.055

Oustaloup, A. (1991). La commande CRONE: commande robuste d'ordre non entier. Paris: Hermes.

Pandey, A. and Murray, R. M. (2022). Robustness guarantees for structured model reduction of dynamical systems with applications to biomolecular models. *International Journal of Robust and Nonlinear Control*, 32(1), pp. 231–257. doi: 10.1002/rnc.6013

Patil, M. D., Vadirajacharya, K. and Khubalkar, S. W. (2021). Design and tuning of digital fractional-order PID controller for permanent magnet DC motor. *IETE Journal of Research*, 69(7), pp. 4349–4359 doi: 10.1080/03772063.2021.1942243

Patra, A. K. and Patra, A. (2020). Genetic algorithm based FOPID controller design for balancing an inverted pendulum (IP). In: *IEEE International Conference on Computational Intelligence for Smart Power System and Sustainable Energy (CISPSSE)*. Odisha, India. doi: 10.1109/CISPSSE49931.2020.9212241

Podlubny, I. (1999). Fractional order systems and PIλDμ controllers. *IEEE Transactions on Automatic Control*, 44(1), pp. 208–214. doi: 10.1109/9.739144

Pradhan, R., Majhi, S. K., Pradhan, J. K. and Pati, B. B. (2019). Optimal fractional order PID controller design using ant lion optimizer. *Ain Shams Engineering Journal*, 11(4), pp. 1181–1191. doi: 10.1016/j.asej.2019.10.005

Rukkaphan, S. and Sompracha, C. (2020). Fractional order model identification for pressure process control system by cuckoo search algorithm. In: *17th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. IEEE Phuket, Thailand. doi: 10.1109/ECTI-CON49241.2020.9158296

Shah, P. and Sekhar, R. (2019). Closed loop system identification of a DC motor using fractional order model. In: *International Conference on Mechatronics, Robotics and Systems Engineering (MoRSE)*. IEEE. Bali, Indonesia. doi: 10.1109/MoRSE48060.2019.8998744

Tripathi, R. P., Gangwar, P. and Singh, A. (2021). Speed control of DC motor with Kalman filter and fractional order PID controller. In: *International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*. IEEE. Bhilai, Chhattisgarh, India. doi: 10.1109/ICAECT49130.2021.9392598

Ziegler, J. G. and Nichols, N. B. (1942). Optimum settings for automatic controllers. *Transactions of the ASME*, 64, pp. 759–768.