scientific reports

Check for updates

OPEN

Fractional-order PID control for elevation and azimuth in a twin rotor system

Abebe Alemu Wendimu[1], Radek Matušů[1]✉, Ibrahim Shaikh[1] & Zeru Kifle Kebede[2]

This paper presents a real-time application of fractional-order PID (FOPID or PI$^{\lambda}$D$^{\mu}$) control for a twin rotor system, optimizing performance beyond conventional PID approaches. A linear model identification is first performed using a black-box approach, with a detailed examination of the system's static properties. The primary aim is to implement PI$^{\lambda}$D$^{\mu}$ control, where the fractional orders λ and μ correspond to the integral and derivative components, respectively, offering enhanced flexibility in system dynamics tuning. The proposed control strategy is validated through experiments on a laboratory-scale twin rotor benchmark. Controller parameters are optimized using advanced algorithms, including Particle Swarm Optimization (PSO), Genetic Algorithm (GA), and the Nelder-Mead (NM) method. These algorithms minimize time-domain performance metrics such as Integral of Absolute Error (IAE), Integral of Time-weighted Squared Error (ITSE), Integral of Squared Error (ISE), and Integral of Time-weighted Absolute Error (ITAE). Notably, the optimized GA-based FOPID controller achieves an IAE performance index of 180.33 for the FOPID in elevation. The GA-based FOPID tuning is particularly effective for IAE performance in the azimuth, yielding a value of 109.2, compared to the GA-based IOPID, which results in a value of 247.05. Additionally, the least performance index is observed when comparing the PSO and NM-based FOPID tuning across all performance indexes. These results demonstrate that the FOPID controller significantly enhances control precision and stability in the twin rotor system, highlighting the potential of fractional-order control (FOC) in real-time applications.

Keywords FOPID, Fractional order, IOPID, Identification, Optimization techniques

Background on fractional calculus: The mathematical foundations of fractional calculus trace their origins to a 1695 correspondence initiated by the German polymath Gottfried Wilhelm Leibniz. Leibniz, having formalized differentiation for integer orders $\frac{d^n f}{dx^n}$ (where $n \in \mathbb{N}$), was subsequently questioned by the Marquis de l'Hopital regarding the potential extension of differentiation to non-integer orders, specifically positing the case $n = \frac{1}{2}$. In his response dated September 30, 1695, Leibniz acknowledged the apparent paradox inherent in such a concept but presciently noted that "one day, useful consequences will be drawn from it"[1]. This exchange represents the conceptual inception of fractional-order differentiation. Since then, the concept has gained popularity and has been formalized as fractional-order calculus. Today, its applications span multiple disciplines, including engineering, where it is a topic of significant interest, particularly in control engineering, focusing on fractional-order control[2].

The Twin Rotor Multi-Input Multi-Output (MIMO) System (TRMS) is a highly nonlinear and complex system that exhibits significant cross-coupling between its control inputs and outputs. One of the primary challenges in controlling such a system is achieving precise and stable control of the rotor dynamics. Traditional integer-order controllers often struggle to provide optimal performance due to the complex dynamics of the TRMS. These challenges necessitate advanced control strategies capable of handling the coupled and dynamic nature of the system. FOCs have demonstrated significant advantages over traditional integer-order controllers by offering enhanced flexibility in tuning. Unlike conventional controllers, which are constrained to integer-order derivatives and integrals, FOCs utilize fractional calculus to provide a broader range of tuning parameters[3–10]. This additional degree of freedom enables better adaptation to system dynamics, leading to improved transient and steady-state performance. Moreover, fractional controllers exhibit superior memory

[1]Department of Automation and Control Engineering, Faculty of Applied Informatics, Tomas Bata University in Zlín, nám. T. G. Masaryka 5555, 760 01 Zlín, Czech Republic. [2]Faculty of Economics and Administration, Department of System Engineering and Informatics, University of Pardubice, Studentska 84, 532 10 Pardubice, Czech Republic. ✉email: rmatusu@utb.cz

effects and hereditary properties, making them highly suitable for applications where long-term system behavior influences performance. These merits make fractional-order controllers particularly attractive for the control of complex systems like the TRMS, where achieving stability and optimal control remains a significant challenge.

Literature review: FOC systems have emerged as a prominent topic in control engineering, attracting significant attention in recent research efforts[3–10]. Numerous studies have demonstrated the application of FOC in control engineering. For instance,[11] proposed a fractional integrator for various control applications, while[12] developed three generations of the CRONE controller. The TID controller was introduced by[13], and the FOPID controller was presented by[14]. The FOMCON MATLAB Toolbox, introduced by[15,16], extends the FOTF toolbox and is widely adopted for modeling, analysis, and design of fractional-order systems due to its flexibility and user-friendly interface. Additionally,[17] expanded on the FOPID concept, and[18] provided a comprehensive analysis of lead-lag fractional-order compensators, detailing their tuning methods and implementation techniques. TRMS presents significant challenges in control engineering due to its highly nonlinear dynamics, strong cross-coupling effects between elevation and azimuth, and inherent system uncertainties. These complexities arise from aerodynamics, actuator limitations, and external disturbances, making precise control difficult. This system has been extensively studied, and various advanced control strategies have been implemented. Key contributions include the work of[19], who developed a mathematical model of the TRMS and designed a fuzzy logic controller to regulate the pitch and yaw angles. Furthermore,[20] applied a Takagi-Sugeno fuzzy model-based controller to the TRMS, optimizing the membership functions using PSO. The importance of this research lies in its potential to enhance the robustness and accuracy of control systems, effectively addressing the inherent challenges associated with controlling the dynamics of modern engineering systems. In their research,[21] proposed the application of fractional algorithms in the control of a TRMS system. They derived a nonlinear mathematical equation based on Newton's second law of motion to model the system and provided an overview of the system model. The authors utilized the PSO algorithm to tune the controllers. They compared the performance of FOPID and IOPID controllers and employed the PSO algorithm to optimize the controller for minimum error. Testing with a real helicopter demonstrated that the FOPID or PID$^{\mu}$ controller outperformed the IOPID in terms of speed and error reduction. Furthermore, the time response of the PID$^{\mu}$ controller was smoother, exhibiting less overshoot and reduced input/output cross-coupling. While both controllers yielded satisfactory output responses, the IOPID required more energy to perform the same task, as suggested by the authors. In their work[22], introduced a real-time control approach for the TRMS using both IOPID and FOPID controllers. They utilized the"System Identification Toolbox"to determine the mathematical models of the TRMS and then derived the parameters of the FOPID controller based on the identified plant model. The optimization algorithm employed for tuning the controller parameters was the GA, implemented in the Matlab program, to achieve different integral performance criteria. The study focused on real-time control of both pitch and yaw positions of the TRMS, comparing the performance of the IOPID controller and the FOPID or PI$^{\lambda}$D$^{\mu}$ controller. The authors evaluated four different integral performance criteria: ISE, IAE, ITAE, and ITSE. Their findings showed that the ITSE-based controller design was the most successful in terms of response performance. Overall, they successfully demonstrated the real-time control of a nonlinear TRMS with two inputs and two outputs using a FOPID controller, offering a promising approach for controlling complex systems. In[23], the author introduced the"Fractional Order Modeling and Control of Twin Rotor Aero Dynamical System using Nelder Mead Optimization". The technique involves identifying a fractional order model of the system. The controller parameters are tuned using NM optimization, and their performance is compared with PSO techniques. The evaluation of the system's performance is done using the IAE metric. However, it is important to note that the authors did not test the controller on a real-time plant, which could limit the generalizability of the results. Linear models may not accurately represent the dynamics of real-time plants, and a linear model not identified from the plant may not accurately reflect its dynamics. Therefore, further testing and validation on a real-time plant would be necessary to confirm the practical effectiveness of the proposed controller. In[24], the authors introduced an advanced parameter estimation technique for modeling a twin rotor MIMO system (TRMS) using a fractional order transfer function with time-delay. Their primary objective was to accurately capture the dynamics of the TRMS by leveraging fractional calculus. To this end, they formulated an optimization problem and employed the NM algorithm, a robust optimization method, to derive the model parameters. The implementation was carried out using the `fminsearch` function from the MATLAB software suite. Furthermore, the authors utilized the FOMCON toolbox in MATLAB to illustrate the efficacy of fractional-order derivative operators in characterizing the TRMS system's behavior. To further strengthen the research background, recent studies on advanced optimization methods in control systems have been considered. For example, the use of artificial intelligence-based algorithms such as the Artificial Rabbits Optimization (ARO) algorithm in PID tuning for temperature control in electric furnaces demonstrates the evolving landscape of controller optimization techniques[25]. Integrating such perspectives highlights the growing relevance and diversity of intelligent optimization strategies in modern control applications.

Research gap and objective: Despite significant theoretical advancements and simulation-based validations of optimization techniques such as PSO, GA, and the NM method for fractional-order control, a critical research gap remains: the lack of comprehensive real-world validation of these controllers on complex, laboratory-scale systems. In particular, most existing studies focus heavily on performance evaluation in simulated environments, offering limited insight into the practical challenges of real-time implementation on physical platforms like the Twin Rotor System. Moreover, there is a notable absence of systematic comparative analyses that investigate how different optimization methods (PSO, GA, NM) influence the performance of fractional-order controllers during the transition from simulation to hardware deployment. These two gaps underscore the need for further research to establish the practical implementation of fractional-order controllers under real-world conditions. The research question posed is: *How can FOCs, optimized using PSO, GA, and NM techniques and integrated with FOMCON toolbox, improve the performance of elevation and azimuth angle of TRMS in real-world scenarios?*

Traditional integer-order controllers (IOPID) often struggle to achieve optimal performance due to their limited tuning flexibility. FOC, such as FOPID, offer a promising alternative. This study investigates how FOC, optimized using PSO, GA, and NM, can enhance elevation and azimuth angle of TRMS performance in real-world applications by improving stability, response time, and error minimization compared to conventional control methods. This paper extends the work presented in[26,27] by focusing on the design, implementation, and testing of a FOC for a TRMS, optimized using the aforementioned techniques.

This study aims to bridge the gap between simulation and real-world implementation in the fractional-order control of elevation and azimuth in twin rotor systems. To achieve this, both IOPID and FOPID controllers are designed and optimized using three distinct optimization algorithms PSO, GA, and the NM method. Controller parameters are tuned using the FOMCON toolbox in MATLAB and integrated with optimization algorithms. Controller performance is first validated within the MATLAB/Simulink simulation environment, followed by real-time implementation on laboratory-scale TRMS hardware. A comparative performance analysis is performed using quantitative metrics such as IAE, ITSE, ISE, ITAE, settling time, and overshoot.

Paper contributions: This research contributes to the growing body of literature on the application of fractional calculus in control systems by integrating PSO, GA, and NM optimization algorithms with the FOMCON toolbox for end-to-end fractional-order controller design. It facilitates the transition from controller optimization to real-time implementation and establishes a co-validation methodology using FOMCON for both MATLAB/Simulink simulations and laboratory-scale TRMS deployment. Additionally, it enhances the understanding of the practical applicability and implementation challenges of FOC techniques in real-world engineering scenarios.

The major contributions of this work summarized as follows:

- A complete methodology from controller optimization to hardware deployment;
- A comparative analysis of PSO, GA, and NM optimization strategies; and
- Real-time validation on a laboratory model benchmark system.

Paper Structure: The remainder of the paper is organized as follows: Section A review of tuning methods for FOPIDs provides a review of FOPID tuning techniques. Section Methodology presents the detailed methodology for implementing the fractional-order controller algorithm in both MATLAB and real-time environments. Section Results from static characteristics is dedicated to the presentation and discussion of MATLAB simulation and real-time experimental results, including a comparative analysis between integer-order and fractional-order controllers. Finally, Section Conclusion summarizes the key findings, highlights their implications for the field, and offers suggestions for future research, followed by a comprehensive list of references.

A review of tuning methods for FOPIDs

Tuning methods for fractional-order PID (FOPID) controllers are generally categorized into three main approaches: analytical, rule-based, and numerical methods. Analytical tuning involves deriving controller parameters from mathematical models to meet specified control objectives, such as achieving desired phase margin and gain crossover frequency. Rule-based methods employ heuristic techniques, trial-and-error strategies, or predefined empirical rules to determine appropriate controller gains. Numerical methods, on the other hand, utilize optimization algorithms and iterative procedures to compute optimal controller parameters based on defined performance criteria. Among numerical approaches, techniques such as GA, PSO, and the NM simplex method have been extensively applied in FOPID tuning. These methods are particularly advantageous in handling complex, nonlinear systems where accurate analytical modeling is challenging or infeasible. For example, PSO and GA are recognized for their effectiveness in exploring large solution spaces and solving multi-objective optimization problems.[28]demonstrated the capability of PSO in tuning FOPID controllers for time-delay systems, achieving enhanced robustness and stability. Likewise, several studies[29–31] have reported the successful application of GA in fractional-order control, highlighting improvements in key performance metrics such as rise time and overshoot. GA, PSO, and NM algorithms were selected based on their robustness, popularity, and established success in controller parameter tuning for nonlinear systems. A summary of previous works on fractional-order PID controllers, including controller structures, tuning techniques, number of parameters used, experimental testing setups, advantages and disadvantages, key findings, and controller comparisons, is presented in Table 1.

Methodology

The methodology employed in this paper is illustrated in Fig. 1. It consists of several key steps to ensure a systematic approach to modeling, controller design, and performance evaluation for the Twin Rotor System:

1. **System/Model Identification:** The TRMS model was identified using real experimental measurements. Parameter estimation was carried out using MATLAB's `fminsearch` function, which applies the Nelder–Mead (NM) simplex optimization algorithm. This method was selected due to its effectiveness in minimizing the error between the experimental data and the simulated model response, ensuring that the identified model accurately reflects the system dynamics. The suggested mathematical models for the decoupled subsystems are represented by the following transfer functions:

$$G_{\text{elevation}}(s) = \frac{K}{(Ts^2 + 2\eta Ts + 1)(T_1 s + 1)}, \qquad G_{\text{azimuth}}(s) = \frac{K}{Ts^2 + 2\eta Ts + 1} \tag{1}$$

Reference	Controller Type	Optimization Method	No. of Parameters	Experimental Testing System	Advantages	Disadvantages	Key Findings	Comparing Controllers
[32]	TSA-aided IT2FLC-(1+PD)-FOPID	Fire-bug Swarm Optimization (FSO), Levenberg–Marquardt (LMA), PSO, TSA	7	First-order plus time-delayed multi-area power system with deregulation	Handles time delay and deregulation, robust control for renewable integration, superior OS, US, ST performance compared to other optimizations	Processor in the loop test only, no hardware in the loop validation	TSA-FOPID outperforms TSA-PID and other optimizations with 88-100% better overshoot, 8-10% better undershoot in frequency error, 38-98% better undershoot in tie-line power error, and improved settling times	TSA-PID, TSA-FOPID, FSO, LMA, PSO controllers
[33]	Two-loop FOPID-FOPI	Levenberg–Marquardt Algorithm (LMA)	7	Inverted cart-pendulum system (ICPS)	Stabilizes an inherently unstable system, improved robustness and performance, considers nonlinear dynamics, robust to disturbances	Requires complex nonlinear modeling and tuning, no real-time experimental validation mentioned	Improved rise time, peak time, settling time, overshoot, undershoot, steady-state error compared to classical PID; stability analyzed via Riemann surface method	PID PI controller
[34]	FPID	Grey Wolf Optimization	5	DC Motor	Improvements in settling time, rise time, and maximum overshoot	limited to Simulation test (did not test the controller on a real-time plant) which could limit the generalizability of the results.	Demonstrated good transient/frequency responses, and good load disturbance reduction analyses	HGSO/PID, GWO/PID, PSO/PID, and SCA/PID
[34]	PSO/FrPID	PSO	4	Cruise Control System	Improvements in settling time, rise time, and maximum overshoot	Assumes linearity and limited to Simulation test	Demonstrated effectiveness of PSO-based FrPID	conventional PID, ASO/PID, GA/PID, and PSO/PID
[18]	FOPID	Auto-tuning	5	Servomotor	Robustness to gain variation, noise rejection	Assumes linearity, limited adaptability	Demonstrated robustness and stability in time/frequency domain	None
[35]	Complex-Order PID (COPID)	`fmincon` (MATLAB)	7	Low-pressure flowing water circuit modeled as a First Order Plus Time Delay (FOPTD) system	Handles delays and uncertainty, robustness against gain variation	Requires accurate model, time-consuming	Improved robustness over ROPID	ROPID
[36]	COPID	Rule-based (from earlier work[37])	7	DC Motor (MATLAB Sim)	Simple to apply, interpretable	Depends on prior heuristic structure	Reduced rise and peak time	PI, PID, PR, FOPID
[38]	COPI-PD (cascaded)	Graphical tuning	6	Cart-Inverted Pendulum, 2nd–3rd order unstable systems	Effective for unstable systems, robustness to uncertainty, noise rejection, steady-state error	Less systematic, tuning effort required	Improved settling time, steady-state error	PID, PI-PD, FOPID, FOPI-COPD
[39]	FOPI-PD	PSO, GA	5	Buck converter	Accurate tuning, suitable for nonlinear systems	Heuristic, risk of local optima	Improved overshoot and settling vs. PID	I-PD, PID
[40]	FOPI-PD	Vector tuning, trial-and-error	-	DC Motor (frequency domain)	Simple, flexible for practical use	Not optimized, high tuning effort	Reduced steady-state error in simulation	None
[41]	FOPI-PD	PSO	-	Magnetic Levitation System	Efficient in nonlinear dynamics	Sensitive to PSO parameters	Improved time-domain metrics	None

Continued

Reference	Controller Type	Optimization Method	No. of Parameters	Experimental Testing System	Advantages	Disadvantages	Key Findings	Comparing Controllers
[42]	FOPI-PD	Ziegler–Nichols	-	Pressure process	Familiar classical tuning + FOPID	Not ideal for complex systems	FOPI-PD gave lower ISE, overshoot	PID, FOPID, PI-PD
[37]	FOPID	Nelder-Mead	5	Quadruple tank system	Simple, gradient-free optimization	May converge slowly, local optima	Enhanced transient response in MIMO process	None
[43]	FOPID	Stability boundary locus + geometric center	6	pH control (sugar cane neutralization)	Visualizable tuning, disturbance rejection and set-point tracking, response time, robustness to uncertainty, and steady-state error.	Model-dependent, limited to specific processes	Improved tracking, rejection, robustness	PI-PD variants
[44]	COPI-PD	Trial-and-error	7	Pressure Process	Wide applicability, robustness to uncertainty, and steady-state error.	Time-consuming manual tuning	Superior steady-state and transient performance	PID, PI-PD, COPI-COPD, FOPID variants

Table 1. Summary of previous works on fractional-order PID controllers.

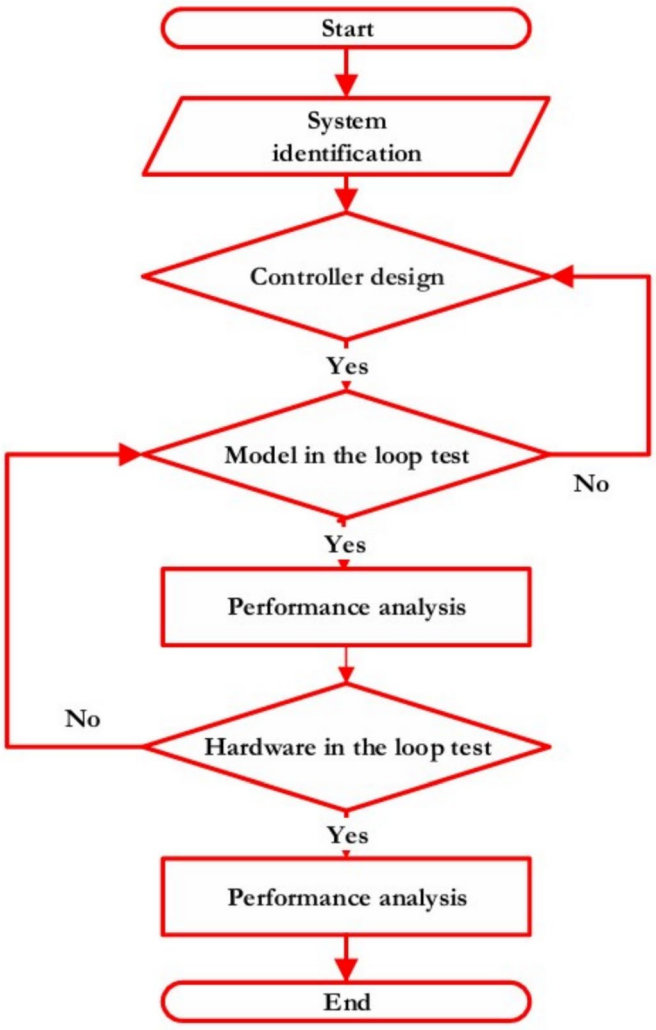

Fig. 1. Schematic representation of the methodology applied in this work.

where K, T, η, and T_1 are optimized using the NM algorithm, and the results are provided in Eq. 14 and Eq. 15 for the elevation and azimuth subsystems, respectively. Although fractional-order controllers are implemented in this study, the system identification process is based on an integer-order model. This decision is motivated by the practical availability of integer-order models and their proven effectiveness in the literature[45]. While fractional-order models may provide greater accuracy, their identification generally requires extensive experimental data and advanced fractional modeling techniques[46]. The chosen integer-order model, however, captures the dominant dynamics near the operating point of the TRMS and is therefore suitable for controller tuning and performance evaluation. The TRMS is inherently a nonlinear, coupled MIMO system. For simplicity, it was decoupled into two linear SISO subsystems corresponding to the elevation and azimuth channels. This assumption of weak coupling (particularly the limited effect of the tail rotor on elevation) enables the design and implementation of fractional-order controllers using well-established linear control methodologies. Nevertheless, this simplification has limitations in conditions with significant cross-axis interaction or strong nonlinearities. Addressing these scenarios through full MIMO fractional modeling will be part of future research. Model validation was performed by comparing the simulated step response of the identified transfer functions with the experimental data. The results confirmed that the identified models closely match the real system behavior within the operating range of interest, thereby ensuring their suitability for controller design.

2. **Controller design and parameter optimization:** The control algorithms are implemented in MATLAB where both IOPID and FOPID controllers are designed. The controller parameters are optimized using advanced optimization techniques, including PSO, GA, and the NM method. These algorithms help determine the optimal control gains that enhance system performance in terms of stability, accuracy, and response time.
3. **Simulation and testing:** The designed controllers are tested in two stages: **Model-in-the-Loop (MIL) Simulation:** The controllers are first evaluated in a MATLAB/Simulink simulation environment using the identified TRMS model. This allows for tuning and validation before hardware implementation. **Hardware-in-the-Loop (HIL) testing:** The validated controllers are then deployed on the actual TRMS hardware to assess their real-time performance.
4. **Performance evaluation and comparative analysis:** The effectiveness of the controllers is analyzed based on multiple performance metrics, including rise time, settling time, overshoot, steady-state error, and control effort. The comparison between IOPID and FOPID controllers provides insights into the advantages of fractional-order control over traditional integer-order approaches.

The subsequent subsections provide a detailed explanation of each methodological step, highlighting the procedures, tools, and key findings at each stage.

System identification

System identification is a fundamental process in control engineering, where mathematical models are developed to accurately represent the behavior of dynamic systems. This is achieved using a black-box approach, which relies on analyzing experimental input-output data without requiring detailed knowledge of the internal dynamics of the system[47]. The resulting transfer function models represent the system behavior without requiring explicit knowledge of internal mechanical or physical equations. A detailed discussion of this identification process can be found in[26,27]. To ensure reliable data collection, real-time experiments were conducted in a controlled laboratory environment. The room was sealed, and the temperature was maintained at standard room temperature to minimize external disturbances affecting the measurements. To accurately model the system dynamics, an integer-order linear plant identification approach is employed for both the elevation and azimuth control loops. This identification is performed using MATLAB's `fminsearch` function, which applies the Nelder–Mead simplex algorithm, as illustrated in Fig. 2. The identified models include a third-order integer plant for the elevation angle and a second-order integer plant for the azimuth angle. In this work, the integer-order model of the system is controlled using a fractional-order controller. Furthermore, the development of a fractional-order model for system identification is presented in[48], where a detailed methodology for fractional-order model identification of the TRMS is provided. These models offer an accurate mathematical representation of the TRMS dynamics, forming the foundation for controller design and optimization.

Controller algorithm implementation

This subsections introduce the structures of both IOPID and FOPID controllers. The IOPID controller is one of the most widely used controllers in industrial applications due to its simplicity and effectiveness in various control scenarios[49]. The standard form of the PID controller in the Laplace domain is given by[50]:

$$C(s) = K_p + K_i\left(\frac{1}{s}\right) + K_d\left(\frac{N}{1+\frac{N}{s}}\right) \tag{2}$$

Where: K_p: Proportional gain. It scales the error signal directly to control the system's output. Increasing K_p increases the system's response speed but can also cause overshoot and instability.

K_i: Integral gain. It accounts for the accumulation of past errors. It helps eliminate steady-state error but can introduce overshoot and oscillations if too high.

K_d: Derivative gain. It predicts future error based on its rate of change. It helps dampen the system's response, reducing overshoot and improving stability.

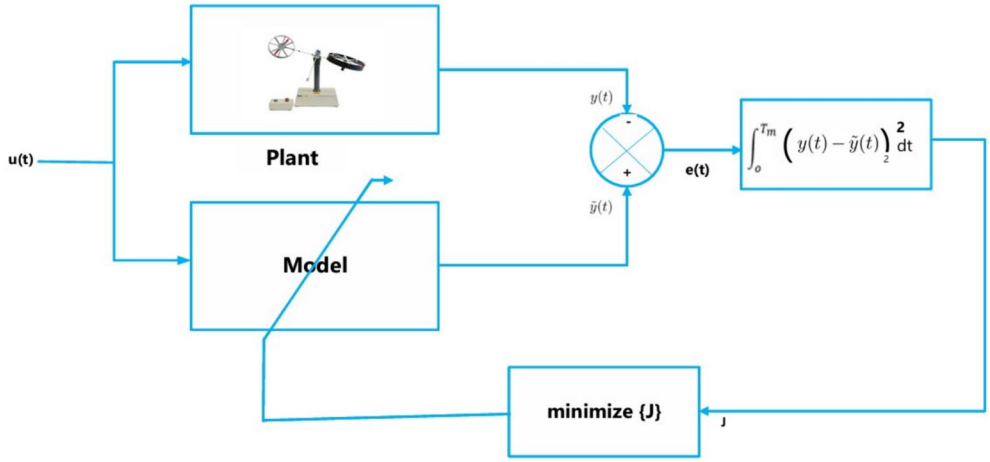

Fig. 2. Block diagram of MATLAB's `fminsearch` algorithm for plant model identification of TRMS[27].

Basically, the IOPID controller operates on the principle of combining three control actions, each addressing different aspects of the control process. The FOPID controller extends the IOPID controller by incorporating fractional calculus. This allows for more flexible and accurate control as it introduces non-integer order differentiation and integration, enhancing control reliability[51,52]. In the FOPID controller, the individual non-integer order differentiation and integration components are defined as follows:

Fractional Integral Control (I^{λ}) integrates the error signal over time with a fractional order λ:

$$u(t) = K_i D_t^{-\lambda} e(t) \tag{3}$$

where $D_t^{-\lambda}$ represents the fractional integral of order λ.

Fractional Derivative Control (D^{μ}) differentiates the error signal with a fractional order μ:

$$u(t) = K_d D_t^{\mu} e(t) \tag{4}$$

where D_t^{μ} represents the fractional derivative of order μ. Fractional-Order PID controller is represented by $PI^{\lambda}D^{\mu}$. The most commonly encountered time domain definition for the fractional-order derivative $D_t^{\lambda} e(t)$ when $\lambda > 0$ is the definition of fractional-order calculus given by the Grünwald–Letnikov approach[15]:

$$D_t^{\lambda} e(t) = \lim_{h \to 0} \frac{1}{h^{\lambda}} \sum_{r=0}^{\left\lfloor \frac{t-a}{h} \right\rfloor} (-1)^r \binom{\lambda}{r} e(t - rh) \tag{5}$$

where $\lfloor \cdot \rfloor$ denotes the integer part, and the binomial coefficient is defined as[15]:

$$\binom{\lambda}{r} = \frac{\Gamma(\lambda + 1)}{\Gamma(r + 1)\Gamma(\lambda - r + 1)} \tag{6}$$

The gamma function $\Gamma(x)$ is given by:

$$\Gamma(x) = \int_0^{\infty} t^{x-1} e^{-t}\, dt, \quad \Re(x) > 0 \tag{7}$$

Another frequency domain approximation technique employed in the literature to approximate the fractional-order differential-integral operator is the Oustaloup approximation. For a lower ω_b and upper frequency range ω_h and of order N,for γ between 0 and 1 filter for operator s^{γ} is defined as[53]:

$$G_f(s) = K \prod_{k=-N}^{N} \frac{s + \omega_k'}{s + \omega_k}, \quad K = \omega_h^{\gamma}, \quad \omega_r = \frac{\omega_h}{\omega_b},$$
$$\omega_k' = \omega_b (\omega_r)^{\frac{k+N+\frac{1}{2}(1-\gamma)}{2N+1}}, \quad \omega_k = \omega_b (\omega_r)^{\frac{k+N+\frac{1}{2}(1+\gamma)}{2N+1}}.$$

The refined Oustaloup approximation is also prevalent in frequency domain methods. It was proposed in[53] and summarized by[15].

$$s^{\gamma} \approx \left(\frac{d\omega_h}{b}\right)^{\gamma} \left(\frac{ds^2 + b\omega_h s}{d(1-\gamma)s^2 + b\omega_h s + d\gamma}\right) \prod_{k=-N}^{N} \frac{s+k}{s+k_0},$$

where

$$k = \left(\frac{b\omega_h}{d}\right)^{\frac{\gamma+2k}{2N+1}},$$
$$k_0 = \left(\frac{d\omega_h}{b}\right)^{\frac{\gamma-2k}{2N+1}}.$$

Oustaloup filter is widely used, but it remains a widely accepted and reliable standard for frequency-domain approximation of fractional-order operators, as supported by prior studies[34,53–56]. Its robustness, simplicity, and compatibility with tools such as FOMCON make it a practical choice for this study. Therefore, the fractional-order controller in our research is tuned using the Oustaloup filter approximation. In FOPID, the overall mathematical control function is given by the following equation:

$$u(t) = K_p \cdot e(t) + K_t D_t^{-\lambda} e(t) + K_d D_t^{\mu} e(t) \tag{8}$$

where $e(t)$ is the error signal. After applying the Laplace transform to Eq. (8) assuming zero initial conditions, the following equation is obtained:

$$C(s) = \frac{U(s)}{E(s)} = K_p + K_i s^{-\lambda} + K_d s^{\mu}, \quad (\lambda, \mu > 0) \tag{9}$$

Where , involving an integrator of order λ and a differentiator of order μ, where both parameters can be any positive real numbers; K_p is proportional gain, K_i is integral gain, K_d, differential gain constant. When $\lambda = 1$ and $\mu = 1$ we obatain IOPID controller. Obviously, when taking $\lambda = \mu = 1$, the result is the IOPID controller. Equation (9) represents the general form of the FOPID controller, also known as $PI^{\lambda}D^{\mu}$, which introduces additional parameters λ and μ and its block diagram scheme is given in Fig. 3. Where λ is order of integration and it determines the degree of memory of the integral term. A non-integer value allows for more precise tuning of the system's response. μ is order of differentiation and it provides a more accurate prediction of future errors.

The additional parameters λ and μ offer extra tuning flexibility beyond IOPID controllers[57]. They allows for more precise tuning and can better handle the dynamic and complex behaviors of modern control systems. As the parameter set expands from IOPID to FOPID, tuning complexity increases. Effective tuning strategies are essential to leverage the benefits of fractional-order controllers[37,58].

Parameter optimization techniques

To tune the parameters of the IOPID controller (K_p, K_i, and K_d) and the FOPID controller (K_p, K_i, K_d, λ, and μ), custom MATLAB scripts along with the FOMCON toolbox are used[15]. Three optimization algorithms are employed: PSO, GA, and the NM method, as depicted in the flowchart in Fig. 4.

The optimization process is guided by four performance indices: IAE, ITSE, ISE, and ITAE. In this work, the individual error indices such as IAE, ITSE, ISE, and ITAE were evaluated over the same simulation horizon and error signal magnitude. As such, explicit normalization was not applied. This approach assumes comparable

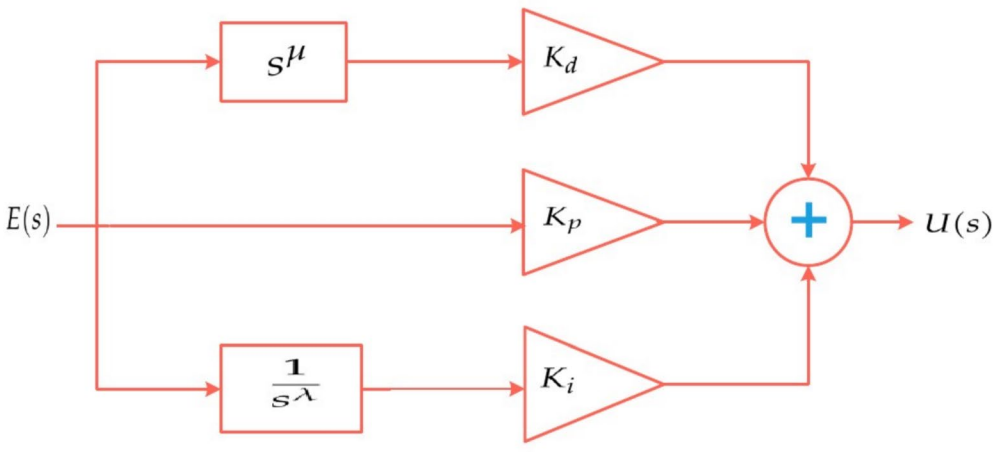

Fig. 3. Block diagram of FOPID as described in (9). When $\lambda = 1$ and $\mu = 1$, the block diagram represents an IOPID controller. Settings of $\lambda = 1$, $\mu = 0$, and $\lambda = 0$, $\mu = 1$ yield IOPI and IOPD controllers, respectively.

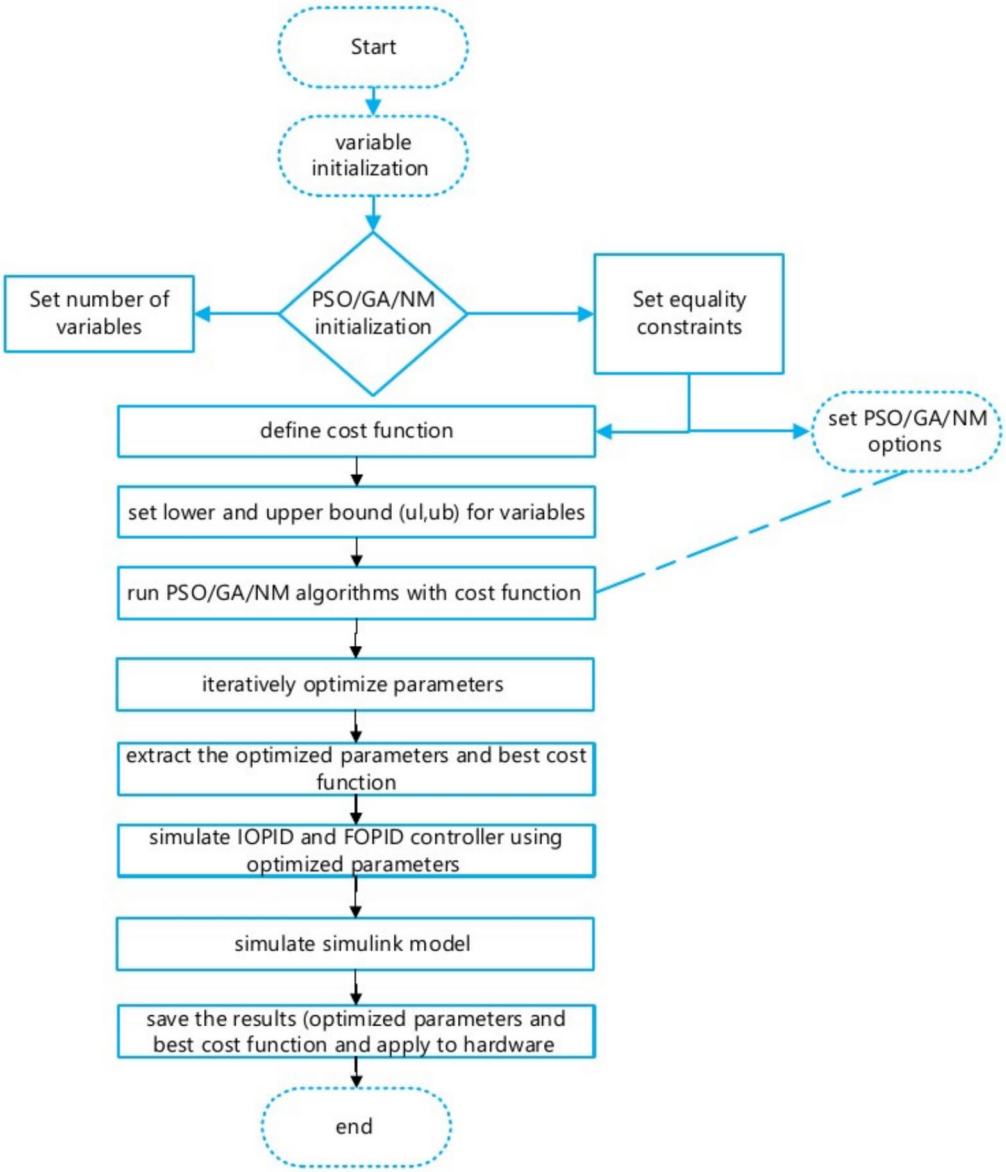

Fig. 4. Flowchart for the GA/PSO/NM-based IOPID /FOPID controller tuning technique.

scales for the indices to prevent any single metric from dominating the objective function landscape. The overall control implementation block diagram is presented in Fig. 7. Each optimization algorithm runs for a maximum of 20 iterations or generations. The simulation time is set to 60 seconds for the elevation control loop and 100 seconds for the azimuth control loop. The optimization process continues until one of the following stopping criteria is met: the stopping criteria for the optimization algorithms were set to either a maximum of 20 iterations or an error threshold of 1×10^{-6}. These values were selected based on previous studies in fractional-order control[35,37,39,41], where similar convergence behavior was observed. A maximum of 20 iterations was set to ensure a reasonable computation time, while the error tolerance was chosen to guarantee sufficient solution precision. Fig. 5 and Fig. 6 presents the convergence plot of the fitness function for azimuth and elevation control using GA optimization respectively, confirming that the selected criteria enable the optimization to effectively reach stable minima. The decreasing trend indicates that the algorithm is successfully minimizing the objective function, thereby enhancing controller performance.

If the simulation fails due to numerical instability or divergence, a penalty cost of $J = 10^3$ is assigned to discourage unfeasible solutions and ensure robust convergence.

Implementation of optimization algorithms

Optimization algorithms are essential for tuning controller parameters to achieve the desired performance in terms of stability, accuracy, and robustness. In this study, we implement and compare various optimization techniques to determine the optimal parameters of the FOPID controller. These algorithms are employed to minimize a predefined performance index derived from time-domain specifications. The following subsections

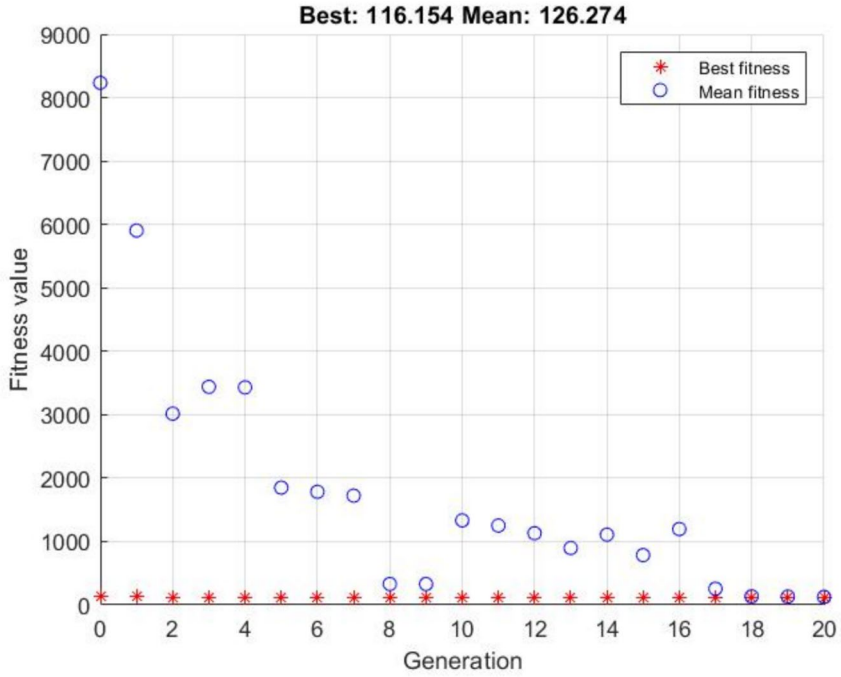

Fig. 5. Convergence plot of the objective function for azimuth control using GA optimization.

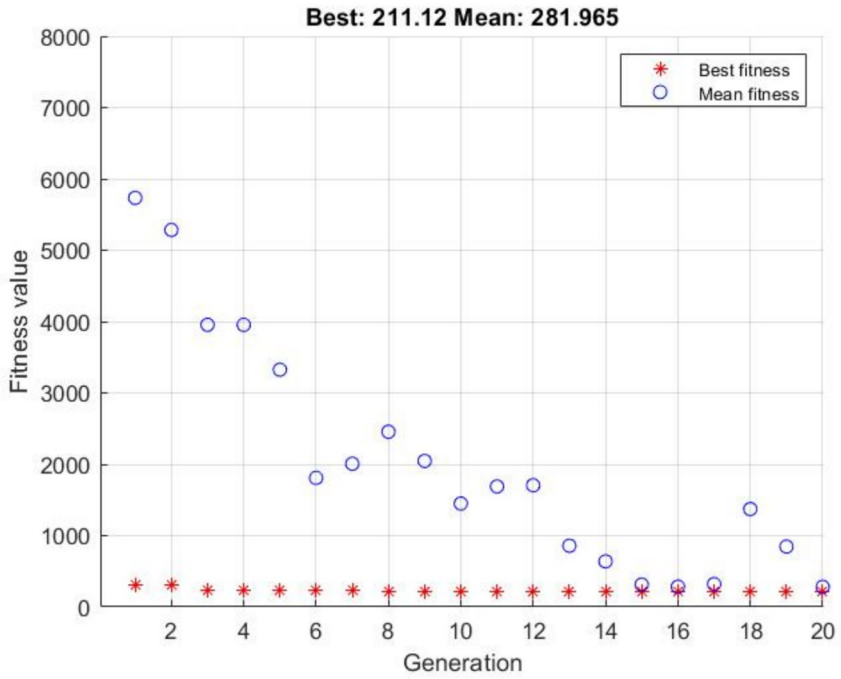

Fig. 6. Convergence plot of the objective function for elevation control using GA optimization.

present the step-by-step implementation of each algorithm, including both local (Nelder–Mead) and global (GA and PSO) optimization methods.

Particle Swarm Optimization (PSO)
Particle Swarm Optimization (PSO) is a population-based optimization algorithm inspired by the social behavior of bird flocking and fish schooling. It is widely used for controller parameter tuning due to its simplicity, fast convergence, and suitability for nonlinear and multidimensional optimization problems. In this study, PSO was implemented using MATLAB's `particleswarm` function, which iteratively updates a swarm of candidate

solutions to minimize the objective function. The optimization objective was defined using performance indices such as IAE, ITSE, ISE, or ITAE. The algorithm adjusts controller parameters within predefined bounds while balancing global and local search. For FOPID tuning, five parameters (K_p, K_i, K_d, λ, and μ) are optimized, while IOPID tuning involves only three (K_p, K_i, and K_d), resulting in lower computational complexity.

Genetic Algorithm (GA)
The Genetic Algorithm (GA) is a population-based metaheuristic optimization method inspired by the principles of natural selection and genetics. It is effective for solving complex, nonlinear, and multi-modal optimization problems. GA explores a population of candidate solutions using evolutionary operators such as selection, crossover, and mutation to evolve better solutions over generations. In this study, GA was implemented using MATLAB's `ga` function. Similar to PSO, GA optimizes controller parameters based on error-based performance indices (IAE, ITSE, ISE, ITAE) while enforcing parameter bounds. The algorithm settings, including population size, mutation rate, crossover rate, and stopping criteria, were tuned to achieve a balance between exploration and exploitation.

Nelder–Mead (NM) Optimization
The Nelder–Mead (NM) algorithm, or simplex method, is a derivative-free optimization technique that iteratively refines a simplex of candidate solutions to converge toward the optimal solution. Unlike global search methods such as PSO and GA, NM is a local search method and is computationally efficient for problems with a good initial guess. It was implemented using MATLAB's `fminsearch` function, starting from predefined initial controller parameters. The optimization objective is based on the same integral performance indices (IAE, ITSE, ISE, ITAE). NM is particularly suitable for fine-tuning controller gains once a near-optimal region is known. A summary of the optimization parameters is provided in Table 2.

Validation and performance evaluation

To assess controller performance, the integral-based error indices (IAE, ISE, ITAE, ITSE) and transient response characteristics such as rise time, settling time, and overshoot were evaluated. The optimized parameters obtained from PSO, GA, and NM were validated through MIL and HIL testing. A comparative analysis of the three optimization algorithms was also conducted, focusing on convergence speed, computational cost, and controller performance.

Integral of Absolute Error (IAE)
The IAE measures the absolute difference between the setpoint and the actual output over a specified time interval. A lower IAE value indicates improved control performance[59].

$$IAE = \int |e(t)|, dt \tag{10}$$

Integral of Squared Error (ISE)
The ISE evaluates the squared error between the setpoint and the actual output over a given time period. This criterion penalizes larger errors more heavily, making it useful for minimizing high-magnitude deviations. A smaller ISE indicates better performance[60].

$$ISE = \int e^2(t), dt \tag{11}$$

Integral of Time-weighted Absolute Error (ITAE)
The ITAE accounts for the absolute error weighted by time, emphasizing late-stage errors more significantly. Controllers optimized using ITAE tend to achieve faster settling times with reduced steady-state error. i.e., a smaller ITAE indicates better performance[61].

$$ITAE = \int t|e(t)|, dt \tag{12}$$

Parameter	PSO	GA	NM
Objective function	Error-based indices (IAE, ITSE, ISE, ITAE)	Error-based indices (IAE, ITSE, ISE, ITAE)	Error-based indices (IAE, ITSE, ISE, ITAE)
Decision variables	K_p, K_i, K_d (IOPID) or K_p, K_i, K_d, λ, μ (FOPID)	K_p, K_i, K_d (IOPID) or K_p, K_i, K_d, λ, μ (FOPID)	K_p, K_i, K_d (IOPID) or K_p, K_i, K_d, λ, μ (FOPID)
Bounds	Lower and upper limits on controller parameters as given in Table 3	Lower and upper limits on controller parameters as given in Table 3	Initial guess with feasible parameter range
Algorithm stopping criteria	1×10^{-6}	1×10^{-6}	1×10^{-6}
Search type	Global	Global	Local

Table 2. Summary of optimization parameters for PSO, GA, and NM algorithms.

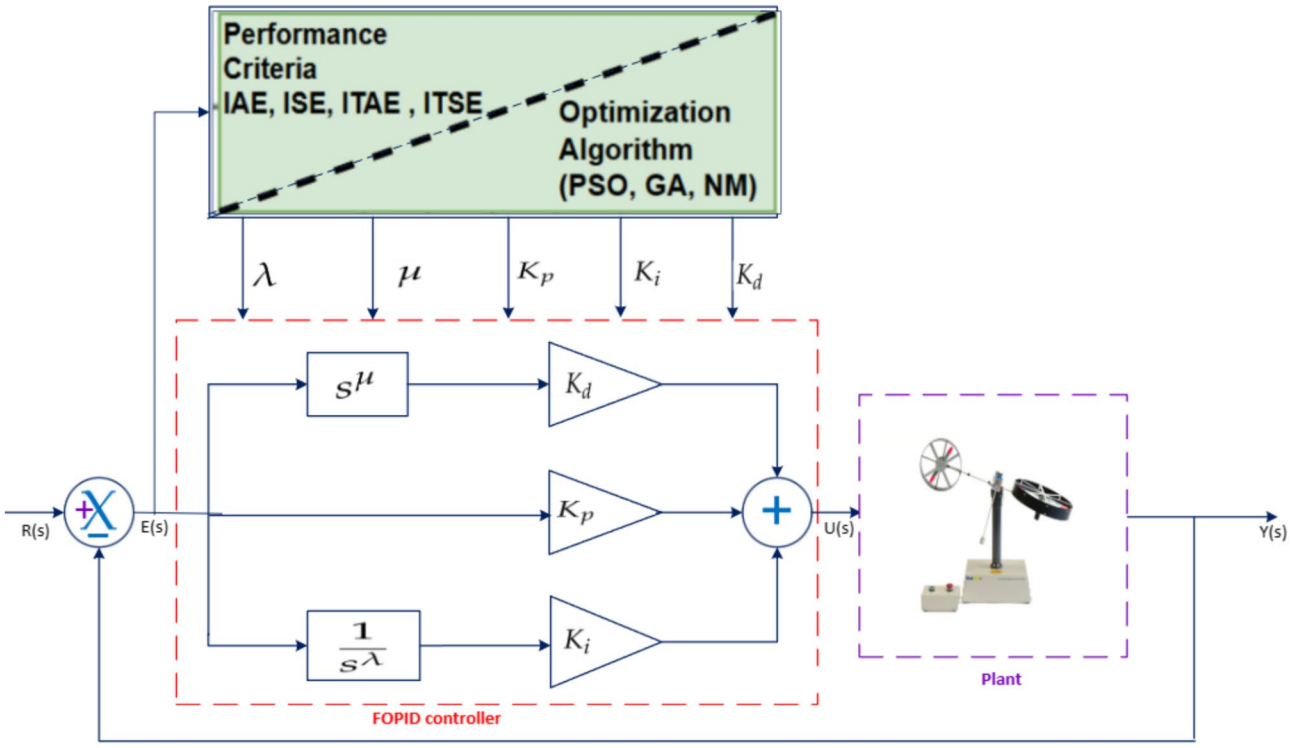

Fig. 7. Overall control system implementation block diagram.

Parameter	IOPID	FOPID	Description
K_p	0 to 0.01	0 to 0.01	Proportional Gain
K_i	0 to 0.1	0 to 0.1	Integral Gain
K_d	0 to 0.1	0 to 0.1	Derivative Gain
λ	1	0 to 2	Integral Order
μ	1	0 to 2	Differentiator Order

Table 3. Lower and upper bounds for IOPID/FOPID controller parameters in the FOMCON toolbox for both elevation and azimuth control loops.

Integral of Time-weighted Squared Error (ITSE)
The ITSE considers the squared error weighted by time, prioritizing early error reduction. It is particularly useful for balancing fast response with long-term accuracy. A smaller ITSE indicates better performance[60,61].

$$ITSE = \int te^2(t), dt \tag{13}$$

These performance indices serve as critical evaluation tools for comparing the effectiveness of IOPID and FOPID controllers. By analyzing these metrics, an informed selection of control strategies can be made based on specific application requirements.

MATLAB implementation and real-time application

The proposed algorithm implements an integer-order plant with a fractional-order controller. The control system includes an integer-order transfer function for the plant and a fractional-order transfer function for the controller, as shown in Fig. 7.

The transfer function, obtained through the linear identification of the real plant, is presented in Eq. (14) for elevation and Eq. (15) for azimuth. The FOPID controller parameters are optimized using PSO, GA, and NM techniques. The range of each controller parameter for both IOPID and FOPID are given in Table 3.

The parameter bounds presented in Table 3 were determined based on prior sensitivity analyses and empirical tuning specific to the dynamics of the TRMS. Preliminary tests showed that broader ranges often led to instability or convergence issues in the optimization process. Therefore, narrower and more practical bounds were defined to ensure effective controller tuning and maintain system stability[25]. These bounds also facilitate

the performance of the applied optimization algorithms (PSO, GA, and NM), which require well-constrained search spaces for efficient and reliable convergence[34]. For comparison, IOPID controllers with fixed fractional orders $\lambda = 1$ and $\mu = 1$ are also tuned.

Comparative analysis of IOPID and FOPID controllers for TRMS elevation and azimuth control

In this study, both IOPID and FOPID controllers are designed and optimized for the TRMS. The comparative evaluation focuses on the elevation and azimuth control loops, ensuring optimal system performance under different control strategies. The optimization of the controller parameters is performed using three optimization algorithms: PSO, GA and NM. The best-performing parameters obtained from each algorithm are selected for real-time implementation and testing. To ensure a comprehensive comparison, the controllers are evaluated based on the transient respons: The rise time, settling time, and overshoot are analyzed to assess how quickly and smoothly each controller stabilizes the system. For elevation control, the primary objective is to maintain stable vertical movement of the TRMS, ensuring minimal oscillations and steady-state error. The IOPID controller provides a simpler structure, whereas the FOPID controller introduces additional tuning flexibility, improving response characteristics such as damping and settling time. For azimuth control, the goal is to achieve precise horizontal rotation of the TRMS. Due to its additional fractional tuning parameters, the FOPID controller is expected to provide improved trajectory tracking compared to the IOPID controller. Each controller configuration is tested under identical conditions to ensure a fair comparison. The optimized controller parameters obtained from PSO, GA, and NM are validated through simulation and will be further assessed in real-time MIL and HIL experiments. The results from this analysis will help determine the most suitable control strategy for TRMS applications, balancing performance and computational feasibility.

Results and discussions

This section presents and analyzes the results obtained from the implementation of the proposed control strategies. The findings are organized to highlight the static and dynamic performance characteristics of the system, as well as the results from system identification, providing a comprehensive understanding of how the controller parameters influence system behavior. The analysis aims to validate the effectiveness of the designed controller and optimization techniques while comparing the performance of integer-order and fractional-order controllers.

Results from static characteristics

The static characteristics of elevation were obtained by applying an input control voltage to the main rotor, starting at 0V and increasing in increments of 0.5V up to 5V. Subsequently, the input voltage was decreased in increments of 0.5V from 5V back to 0V. Throughout the experiment, the resulting elevation output signal was recorded while the tail motor remained stationary[27].

Based on the displayed graph, the course of the static characteristics experiment is depicted in Fig. 8. The experiment suggests that the direction of input voltage change does not significantly affect the system's behavior. From the presented graph, the static characteristic of the system appears to be piecewise linear. Higher voltages appear to result in higher gain.

On the other hand, the static characteristics of azimuth were obtained by applying an input control voltage to the tail rotor, ranging from 1.5V to 4V for increasing azimuth and from 3.5V to 2V for decreasing azimuth. Throughout these measurements, the control input voltage for the main rotor remained at 2.5V, keeping the rotor stationary. The corresponding measurement outputs are presented in Fig. 9. The static characteristics of azimuth are nonlinear, displaying a large hysteresis curve. These nonlinearities and hysteresis significantly affect the control of the system.

Results from identification

Although the TRMS is a MIMO system, only a SISO system is considered for the identification and control of the elevation and azimuth by keeping one rotor stationary .

Linear identification of elevation using fminsearch in MATLAB considering a 3rd order system and using step response data from a 3V to 4V input, the transfer function for elevation is derived as follows[27]:

$$G_{\text{elevation}}(s) = \frac{136.84}{0.2182s^3 + 0.2785s^2 + 0.8832s + 1}\ . \tag{14}$$

The elevation step response identified using `fminsearch` is presented in Fig. 10,the comparison between the real experimental data and the simulated model yields a fitness of 82.45%. The real-time elevation response exhibits oscillations, whereas the simulated response does not, which explains the relatively lower fitness value obtained for the simulated model. While the corresponding Bode plot is shown in Fig. 11. According to the Bode plot, the frequency range extends from 0.01 rad/sec to 100 rad/sec, encompassing both the lower and upper frequencies.

Linear identification of azimuth using MATLAB's `fminsearch` function – considering a 2nd order system and the measurements of step input voltages ranging from 3V to 3.5V, the transfer function for elev is derived as follows[27]:

$$G_{\text{azimuth}}(s) = \frac{583.5}{5.128s^2 + 3.286s + 1}\ . \tag{15}$$

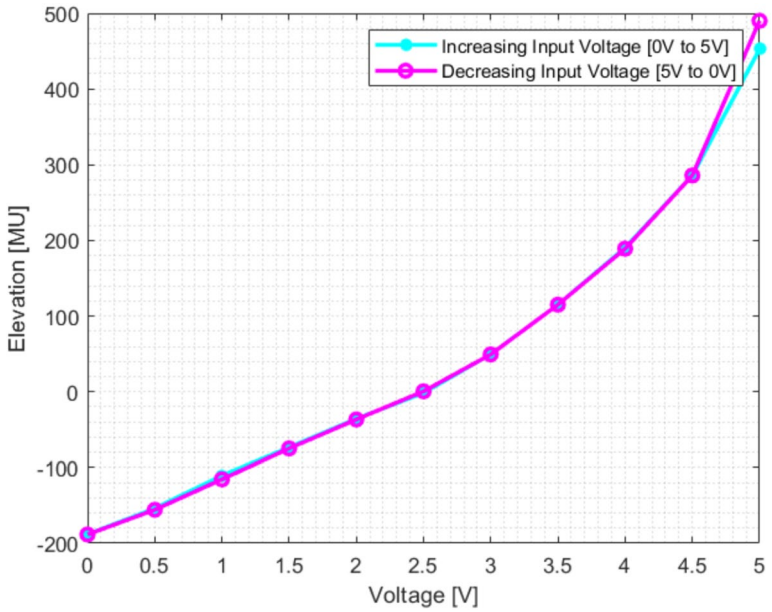

Fig. 8. Static characteristics of elevation.

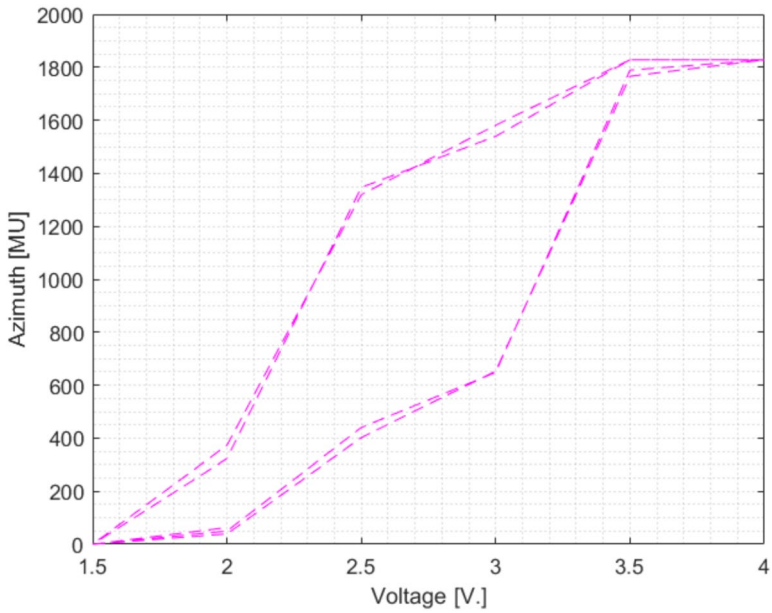

Fig. 9. Static characteristics of azimuth.

The azimuth response identified using `fminsearch` is presented in Fig. 12, the comparison between the real experimental data and the simulated model yields a fitness of 85.78%. While the corresponding Bode plot is shown in Fig. 13.

According to the Bode plot, the frequency range extends from 0.001 rad/sec to 100 rad/sec, encompassing both the lower and upper frequencies.

The results indicate that the linear plant model obtained from the `fminsearch` identification approach offers the most accurate representation of the real laboratory model of TRMS. This identified plant is used consistently throughout the entire paper.

Model validation

The identified model was validated by comparing its response with the experimental step response data. The validation was performed by simulating the identified transfer function using the same input signal as the

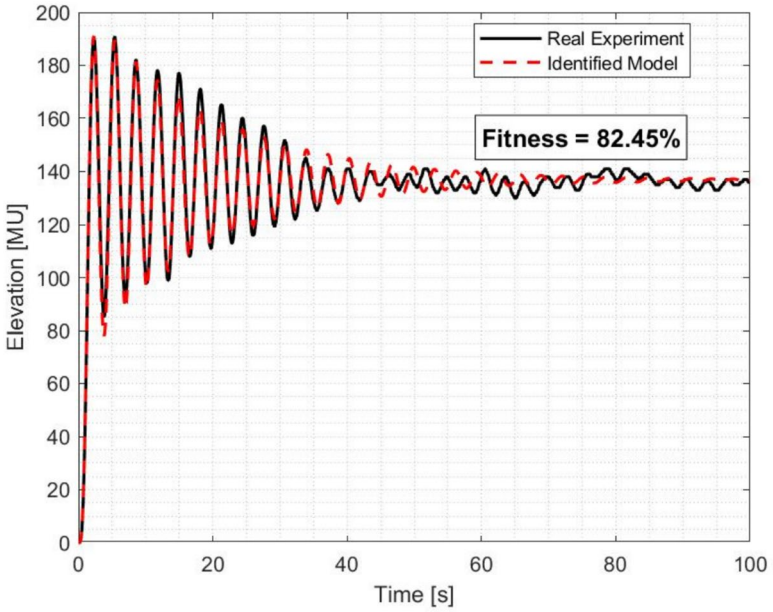

Fig. 10. Elevation step response identified using the Nelder–Mead optimization method (`fminsearch`).

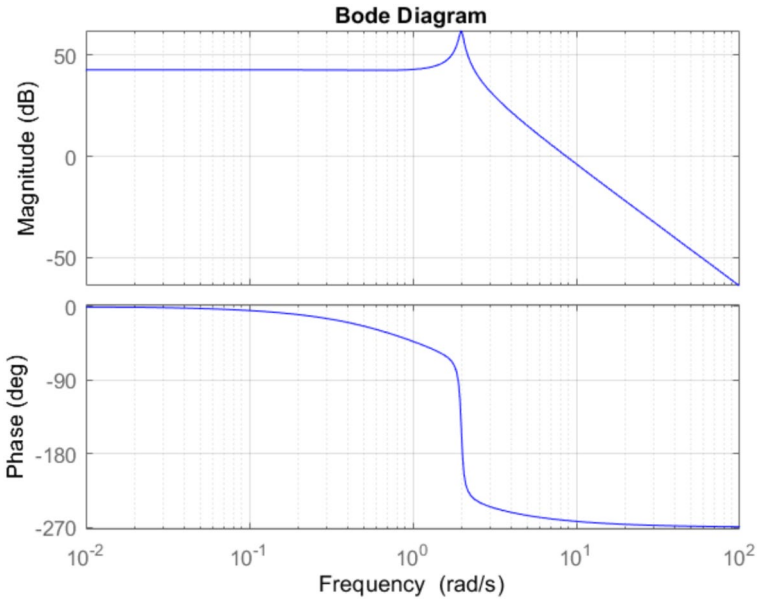

Fig. 11. Elevation Bode Plot.

experiment and calculating the fitness percentage between the measured response and the model output. The fitness was quantified using the following equation[62]:

$$\text{Fit } (\%) = 100 \times \left(1 - \frac{\|y - \hat{y}\|}{\|y - \bar{y}\|}\right) \tag{16}$$

where y is the measured output, $\hat{y}$ is the identified model output, and $\bar{y}$ is the mean of the measured output. This metric measures how well the identified model matches the experimental data, as a higher a fitness value indicates an excellent agreement between the real system and the identified model. The fitness percentage was displayed in the response plots to visually demonstrate the accuracy of the identified model as illustrated in Fig. 10 and Fig. 12.

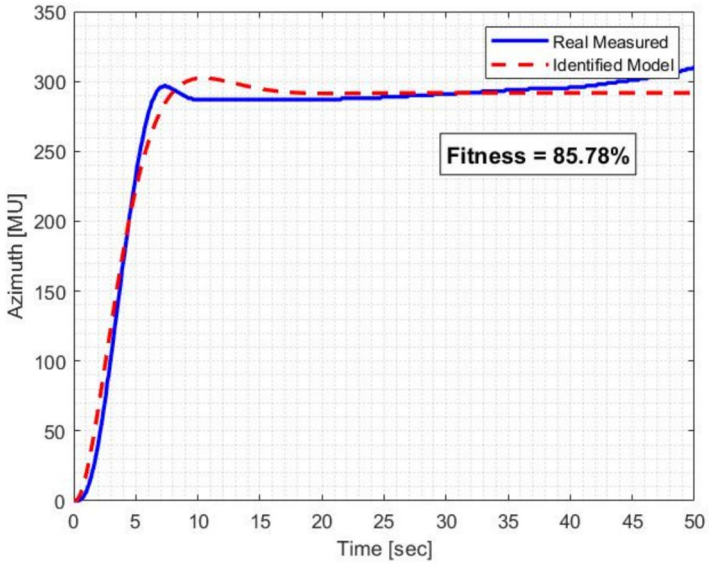

Fig. 12. Azimuth step response identified using the Nelder–Mead optimization method (`fminsearch`).

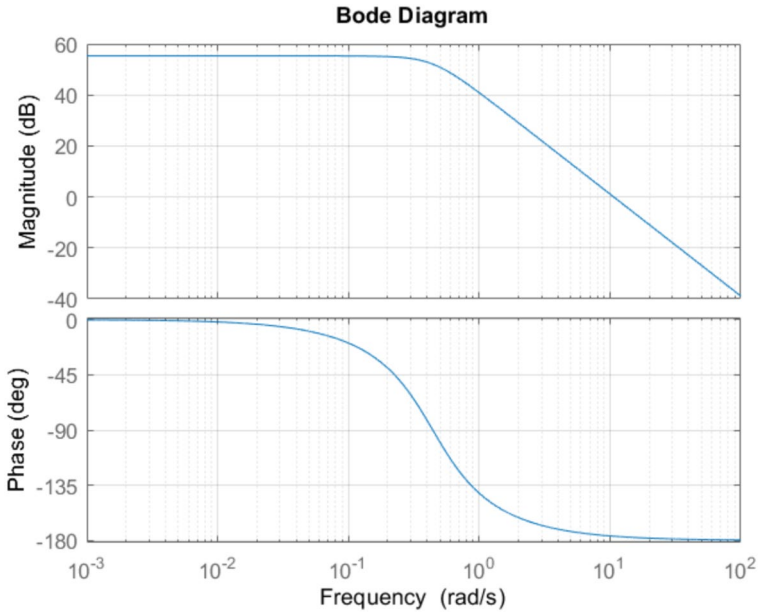

Fig. 13. Azimuth Bode Plot.

Results from controller algorithm implementation

During the laboratory experiment, two testing methods are employed. Firstly, the controller algorithm undergoes a MIL test, where it is assessed on a linear plant model obtained using MATLAB's `fminsearch` function. This initial evaluation gauges the controller's performance in controlling the TRMS system under simulated conditions.

Results of GA IOPID and FOPID controller responses for elevation linear model

Table 4 presents the parameter obtained for the IOPID controller using GA, PSO, and NM based tuning approaches for the elevation control of the TRMS. For the GA-based tuning of the IOPID controller, the ITAE performance index achieves the best overall balance, with the lowest cost function (J = 744.89) and a moderate overshoot (25.71%). The IAE-based tuning results in the highest overshoot (40.35%), suggesting that it may not be the optimal choice when stability is a priority. Meanwhile, the ITSE-based tuning provides better transient response, but its cost function remains higher than ITAE. The parameters obtained for the FOPID controller are shown in Table 5.

P.I [a]	K_p	K_i	K_d	$\%M_p$	t_r (sec.)	t_s (sec.)	J
IOPID Controller - GA based tuning - for Elevation							
ITSE	0.0018	0.0083	0.0043	20.7250	2.7133	18.9353	2.0486e+04
ITAE	0.0015	0.0093	0.0059	25.7128	2.6103	11.5748	7.4489+02
ISE	0.0037	0.0123	0.0092	37.5025	0.5672	12.2036	1.0179e+04
IAE	0.0033	0.0193	0.0109	40.3485	0.5056	12.5622	2.9531e+02
IOPID Controller - PSO based tuning - for Elevation							
ITSE	0.0011	0.0085	0.0041	18.9433	2.6248	16.2758	2.0315e+04
ITAE	1.9784e-4	0.0049	0.0020	14.4838	2.8247	16.3071	5.3337+02
ISE	1.9047e-4	0.0073	0.0025	23.7832	2.1154	15.0789	1.2007+04
IAE	6.7809e-4	0.0062	0.0032	17.3441	2.7975	15.6666	1.9209e+02
IOPID Controller - NM based tuning - for Elevation							
ITSE	3.6803e-4	0.0057	0.0020	15.0796	2.7453	17.4269	1.9209+02
ITAE	1.7278e-5	0.0044	0.0016	8.8182	2.8692	13.7324	1.9209e+02
ISE	1.9047e-4	0.0073	0.0025	24.5633	0.7483	15.8195	9.9245e+03
IAE	2.8145e-4	0.0053	0.0022	15.4363	2.7997	16.1780	1.9209e+02

Table 4. Parameter ranges for the IOPID controller (GA, PSO, NM) for elevation. [a]Performance Index.

P.I	K_p	K_i	K_d	λ	μ	J
FOPID Controller - GA based tuning - for Elevation						
ITSE	0.0806	0.0192	0.0312	0.8749	1.3736	1.7537e+04
ITAE	0.0665	0.0067	0.0322	1.0097	1.4469	636.5638
ISE	0.0870	0.00387	0.0388	0.5742	1.3610	1.0585e+04
IAE	0.0925	0.0097	0.0192	1.0284	1.6460	180.3387
FOPID Controller - PSO based tuning - for Elevation						
ITSE	0.0028	0.0127	0.0086	0.9021	1.0307	2.0838e+04
ITAE	0.0086	0.0062	0.0084	1.0251	1.2253	687.5096
ISE	0.0080	0.0944	0.0386	0.2018	1.3179	1.1215e+04
IAE	0.0099	0.1000	0.0379	0.3714	1.3501	268.1616
FOPID Controller - NM based tuning - for Elevation						
ITSE	1.8671e-4	0.0081	0.0034	1.0017	0.9368	1.9834e+04
ITAE	3.6833e-4	0.0055	0.0026	1.0057	0.9887	558.9315
ISE	0.0014	0.0150	0.0101	0.8203	0.9968	9.4019e+03
IAE	5.8010e-4	0.0059	0.0027	1.0128	0.9893	195.8456

Table 5. Parameter ranges for the FOPID controller (GA, PSO, NM) for elevation.

For GA based tuning of the FOPID controller, ITAE produces the lowest cost function (J = 636.56) and results in a well-balanced response. The IAE based GA tuning achieves the lowest J value among all tested cases (J = 180.33), highlighting its effectiveness in minimizing overall system error. However, ISE and ITSE based tuning for FOPID produce relatively high cost function values (J = 1.05e+04 and 1.75e+04, respectively), indicating that these indices might not be the best choices in terms of optimization efficiency, as illustrated in Fig. 14. In the context of elevation control, the system is designed to operate within a control input (effort) range of -2.5 V to 2.5 V. As shown in Fig. 15, certain regions of the control effort particularly in the lower voltage segments remain within or close to this specified range, making them suitable for practical implementation. Oscillations in elevation control effort likely result from a combination of more complex dynamics, controller tuning parameters (especially λ and μ terms), and aggressive performance objectives. In all cases the cost function remains relatively high. This behavior is primarily due to the complex and highly nonlinear dynamics of the twin rotor system and oscillating behaviour which makes precise control challenging. While the optimization algorithm has improved the system's response within the physical and practical limitations, a lower cost function may not be feasible without compromising system stability.

Results of PSO IOPID and FOPID controller responses for elevation linear model

In the PSO tuning algorithm, the ITAE performance index results in smaller overshoots (14.5%) and fewer oscillations compared to other indices, as shown in Fig. 16. This demonstrates the effectiveness of ITAE for better control performance in the PSO based IOPID tuning approach. The ITSE based tuning results in higher overshoot (18.94%) and a slightly increased settling time (16.28 sec). The ITAE based tuning achieves the lowest cost function value (J = 533.37), making it the most efficient among PSO based IOPID tuning approaches.

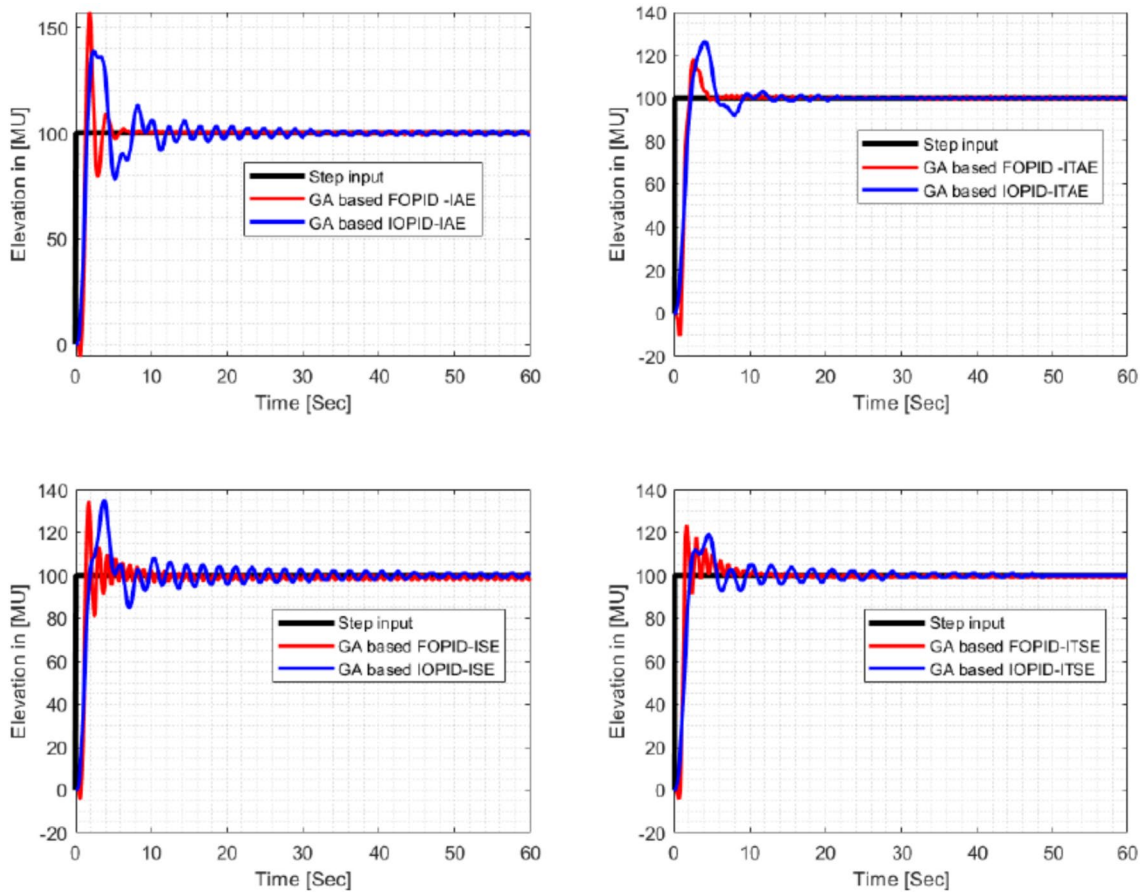

Fig. 14. GA-based IOPID/FOPID Elevation Response for various Performance Indices.

For the PSO based FOPID controller, the ITSE performance index yields smaller overshoots (22%) and fewer oscillations compared to other indices, as illustrated in Fig. 16. However, the overall cost function (J) remains high (2.08e+04), suggesting a trade-off between reduced oscillations and computational efficiency. The ITAE based PSO tuning for FOPID results in better cost function performance (J = 687.51) and provides improved stability, making it a strong candidate for tuning optimization.

Results of NM IOPID and FOPID controller responses for elevation linear model

For NM based tuning of the IOPID controller, the ITAE performance index achieves the lowest overshoot (8.82%), making it the best choice for applications where overshoot minimization is critical, as shown in Fig. 17. The corresponding cost function (J = 192.09) remains competitive, further supporting its effectiveness. The ISE based NM tuning results in a higher overshoot (24.56%) and a higher cost function (J = 9.92e+03), making it a less favorable option. The ITSE based tuning, while achieving relatively stable performance, results in a high cost function (J = 1.92e+02), suggesting it may not be the most efficient choice. For NM-based tuning of the FOPID controller, ITAE achieves the lowest cost function (J = 558.93), reinforcing its effectiveness for system stability. However, the ITSE-based NM tuning results in a higher cost function (J = 1.98e+04), indicating that it may not be the best choice for optimization. The IAE based tuning achieves a reasonable trade-off between stability and efficiency (J = 195.85) with relatively small oscillations, making it a viable alternative to ITAE.

FOPID controller comparison - real-time experiment for elevation

To ensure accuracy and practical relevance, all FOMCON-based results were validated through real-time experiments on both channels (elevation and azimuth) of the TRMS. The controller was first tested using a MATLAB simulation model and subsequently implemented on a laboratory-scale TRMS for real-time testing. This two-step approach allowed us to perform both MIL and HIL validations, effectively aligning with the V-model development process. These steps confirm the reliability and applicability of our results. In this study, ITAE was selected as the primary performance index for controller comparison. Compared to other indices such as IAE or ISE, ITAE places greater emphasis on errors occurring earlier in the response, thereby better capturing transient behavior. This made it particularly suitable for the TRMS system, where rapid stabilization of elevation and azimuth is crucial.

The second testing approach is an HIL test, where the controller is implemented in MATLAB and tested on the physical TRMS system in real-time. Identical FOPID parameters were used in both simulation and real-time experiments. The results for step and ramp inputs are compared in Fig. 18. Overall, FOPID controllers provide

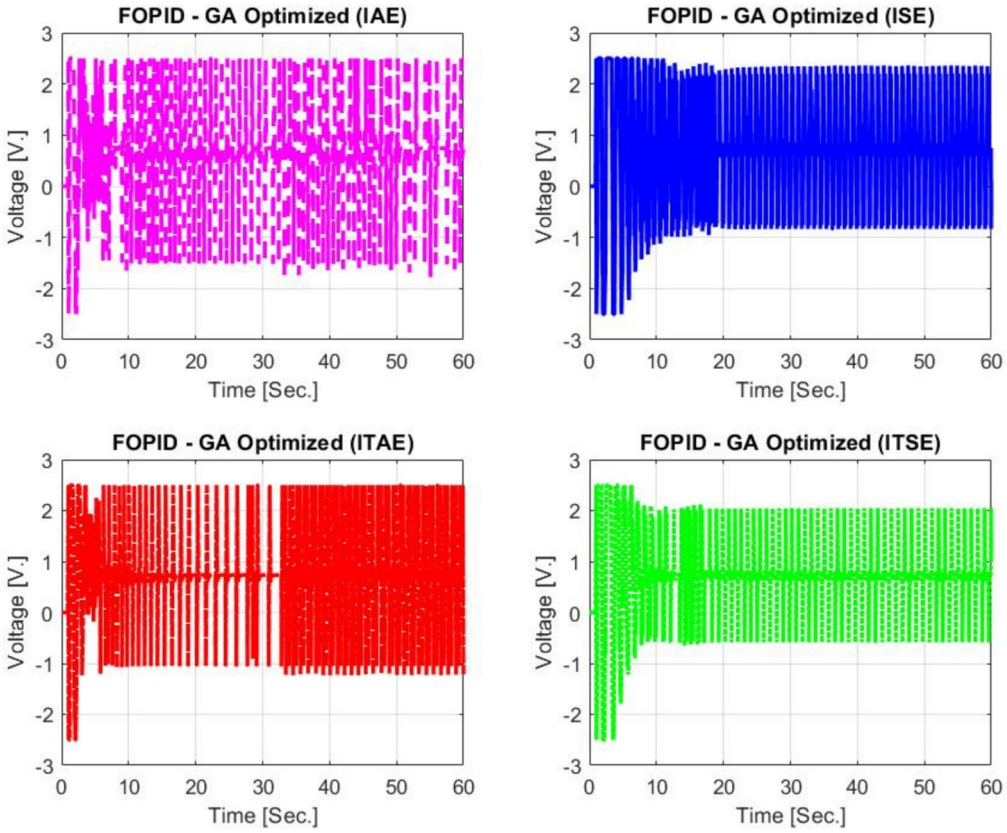

Fig. 15. Control effort of elevation GA-optimized FOPID controllers based on different performance indices.

greater tuning flexibility due to the additional parameters λ and μ, which help in refining performance trade-offs. GA based FOPID tuning appears to offer the best overall performance, while NM based IOPID tuning excels in overshoot reduction. The selection of the best controller depends on the application requirements, if minimizing overshoot is the primary concern, NM based IOPID tuning with ITAE is the best choice, whereas for an optimal balance of cost function J and system stability, GA based FOPID tuning with IAE provides the best results. In real-time experiments, the PSO based FOPID controller with the ITAE performance index shows better results compared to others.

Results of GA IOPID and FOPID controller responses for azimuth linear model
For IOPID, the IAE based tuning produced the fastest settling time (5.83s) but had a moderate overshoot (12.32%). Meanwhile, the ITAE-based tuning resulted in a lower cost function (J = 4.57e+02) and a shorter settling time (8.38s) compared to other indices, as shown in Table 6. The ITSE and ISE methods showed higher settling times above 13s (see Table 7). For FOPID, the ITAE-based tuning performed best in terms of cost function (J = 329.17), while the IAE based tuning also yielded a competitive result (J = 109.29), as illustrated in Fig. 19. Compared to IOPID, the FOPID controller had more flexibility in tuning due to the added fractional parameters λ and μ, leading to better adaptability in performance trade-offs.

Results of PSO IOPID and FOPID controller responses for azimuth linear model
According to the step response of the Azimuth for the PSO-based IOPID controller algorithm, the ITAE performance index offers the advantage of yielding smaller overshoots (1.2315%) as given in Table 6. However, this comes at the cost of a longer settling time (16.3083s), which may not be ideal for applications requiring faster response times. On the other hand, the IAE performance index provides a more balanced trade-off between overshoot (5.6498%) and a reduced settling time (10.2391s), making it a viable alternative depending on the control requirements. In addition, according to the step response of the Azimuth for the PSO-based FOPID controller algorithm, the ISE performance index offers the advantage of yielding smaller overshoots (10%) as depicted in Fig. 20. The IAE-based tuning also demonstrated efficient performance, with a low cost function (J = 116.07), indicating that FOPID allows for improved optimization of system dynamics. The fractional-order parameters λ and μ provide additional tuning flexibility, contributing to enhanced robustness and adaptability in azimuth control compared to IOPID.

Results of NM IOPID and FOPID controller responses for azimuth linear model
For IOPID, the ISE-based tuning resulted in the lowest settling time (2.38s) with minimal overshoot (2.59%). However, ITAE-based tuning produced the lowest cost function (J = 650.55) at the cost of a longer settling time

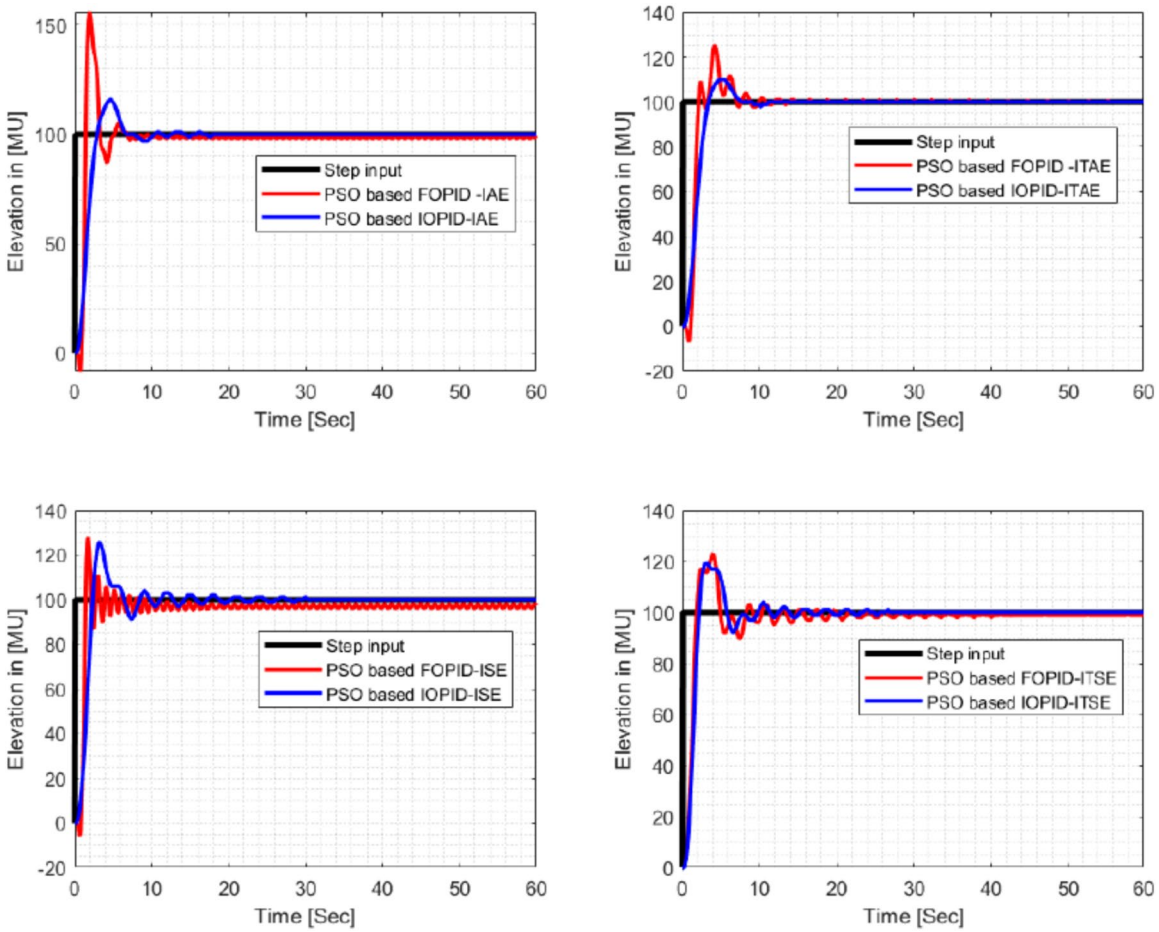

Fig. 16. PSO based IOPID/FOPID Elevation Response for various performance indices.

(14.55s). IAE-based tuning had a moderate settling time of 10.05s but a higher overshoot (14.04%), as given in Table 6. For FOPID, IAE-based tuning resulted in the lowest cost function (J = 114.05), showing better tuning efficiency. The ITAE-based tuning also achieved a lower cost function compared to IOPID. This highlights that FOPID provides a better trade-off between settling time, overshoot, and cost function compared to IOPID, especially for NM tuning. The corresponding responses are shown in Fig. 21.

In the context of azimuth control, the system is designed to operate within a control input (effort) range of -2.5 V to 2.5 V. As shown in Fig. 22, azimuth control effort remains smoother due to simpler dynamics and/or less aggressive tuning.. This indicates that the actuator functions within safe voltage limits for these intervals, reducing the risk of saturation and ensuring reliable system operation.

IOPID and FOPID controller comparison - real-time experiment for azimuth

The identical controller parameters were used for both IOPID and FOPID in both simulation and real-time experiments. The implementation approach was employed, and the measurement results are compared in Fig. 23. The inputs included both step and ramp input signals. The GA-based FOPID tuning is particularly effective for IAE performance, yielding a value of 109.2, in comparison to the GA-based IOPID, which results in a value of 247.05 due to its additional fractional-order parameters, offering better fine-tuning capabilities. Additionally, the least performance index is observed when comparing the PSO and NM-based FOPID tuning across all performance indexes. Thus, FOPID is the preferred choice for azimuth control, especially when cost function minimization is a priority.

Sensitivity analysis of fractional-order PID (FOPID) controller

To evaluate the robustness of the fractional-order PID (FOPID) controller, a sensitivity analysis was conducted for the differentiation order (μ) and integration order (λ). This analysis aimed to investigate their influence on key time-domain specifications, including overshoot, rise time, peak time, and settling time. The second-order plant obtained from azimuth identification was used for this study, while the controller parameters K_p, K_i, and K_d were kept constant across all simulations. The fractional differentiation order μ and integration order λ were varied from 0 to 2 in increments of 0.25. For each (μ, λ) combination, the closed-loop transfer function was obtained, and its step response was analyzed. The overshoot, rise time, peak time, and settling time were computed directly from the closed-loop transfer function. From Fig. 24 a), it can be inferred that the sensitivity

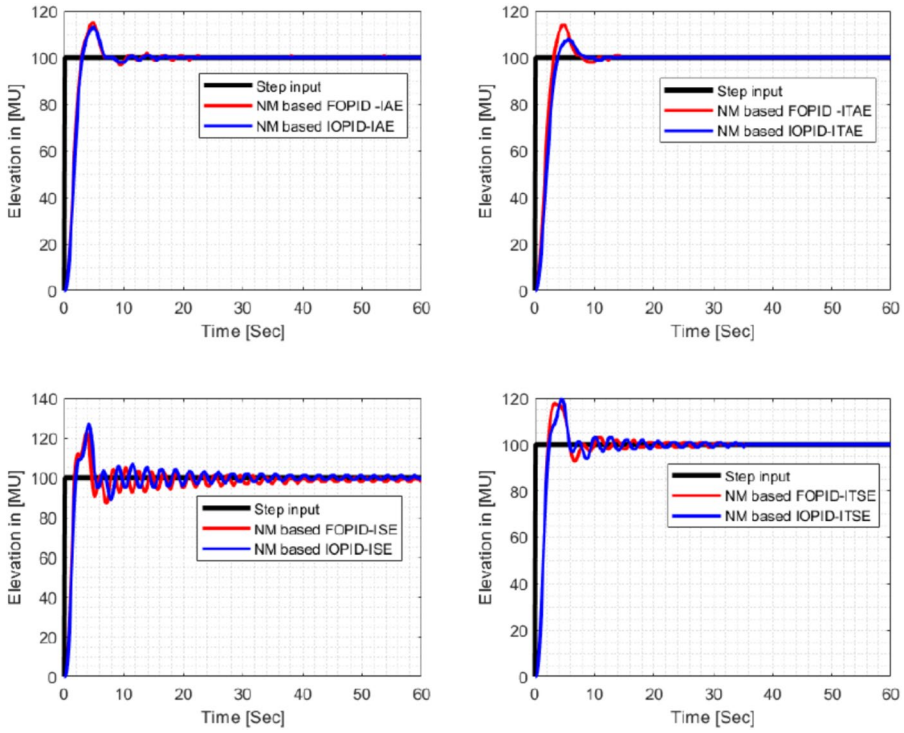

Fig. 17. NM-based IOPID/FOPID elevation response for various performance indices.

analysis of the overshoot with respect to the fractional orders reveals notable trends. As λ increases from 0 to 2, the overshoot shows an almost monotonic increase, indicating that higher integral action tends to amplify the system's overshoot. This trend becomes particularly pronounced around $\lambda \approx 2$, where a sharp gradient is observed, especially in the neighbourhood of $\mu \approx 1$. This indicates a region of high sensitivity to small changes in λ. In contrast, variation in μ does not result in a clear monotonic trend. Instead, the overshoot demonstrates a nonlinear variation with respect to μ, forming a shallow valley centred approximately at $\mu \approx 1$ across different values of λ. This suggests that while the derivative component has a significant influence, its effect on overshoot is more balanced and less abrupt. This valley region represents a potentially robust operating point for controller tuning.

The rise time surface, shown in Fig. 24b), exhibits a dominant sensitivity to the differentiation order μ. Increasing μ from 0 to 2 generally results in an increase in rise time, with higher values of μ causing a steep degradation in responsiveness. This is attributed to the increased derivative effect, which can introduce damping that slows the system's initial response. The integration order λ has a relatively weak influence on the rise time. While an increase in λ results in a slight increase in rise time, especially when μ is low, the overall effect remains subtle and does not exhibit sharp transitions or significant gradients. The fastest responses, characterized by the lowest rise times, are found in the regions of low μ, low λ, and low μ with high λ. These regions represent a favourable balance between rapid response and stability. As shown in Fig. 24c), the peak time surface reveals a more complex and nonlinear behaviour. For lower values of λ, the peak time increases gradually with increasing μ, suggesting that higher derivative orders delay the peak of the response. However, as λ approaches 2, the surface behaviour changes: the peak time increases sharply as μ rises from 0 to 1, then levels off to a flat plateau, and finally shows a slow decrease as μ increases further from 1 to 2. The influence of λ on peak time remains relatively weak across most of the domain, but becomes more significant for higher values of λ. A distinct flat region is observed around $\mu \approx 1$, $\lambda \approx 2$, representing a zone of delayed and flattened peak response due to excessive fractional action. The minimum peak time is found in the region $\mu \approx 0.01$, $\lambda \approx 0.01$, where the system exhibits the fastest response. However, this region may compromise other performance aspects, such as overshoot, and therefore may not be optimal from a holistic control perspective. As observed from Fig. 24d) the settling time surface exhibits no significant variation across the entire μ, λ parameter space. Unlike the other time-domain specifications, the settling time remains effectively constant within the tested range $\mu, \lambda \in (0, 2)$. This indicates that, under the fixed controller gains K_p, K_i, and K_d used in this analysis, the settling time is largely insensitive to changes in the fractional orders of integration and differentiation.

These findings confirm that simultaneous adjustment of μ and λ provides an additional degree of freedom compared to classical integer-order controllers, enabling improved performance optimization and enhanced robustness.

In the control system evaluation illustrated in Fig. 25, the step response of the azimuth angle using an integer-order PID controller with $\mu = 1.00$ and $\lambda = 1.00$ exhibited an overshoot (OS) of 7.64% and a settling time of 10.37 seconds. To improve the transient response without significantly increasing overshoot, a FOPID configuration was adopted. In this design, μ was maintained at 1.00 while the integral order was adjusted to

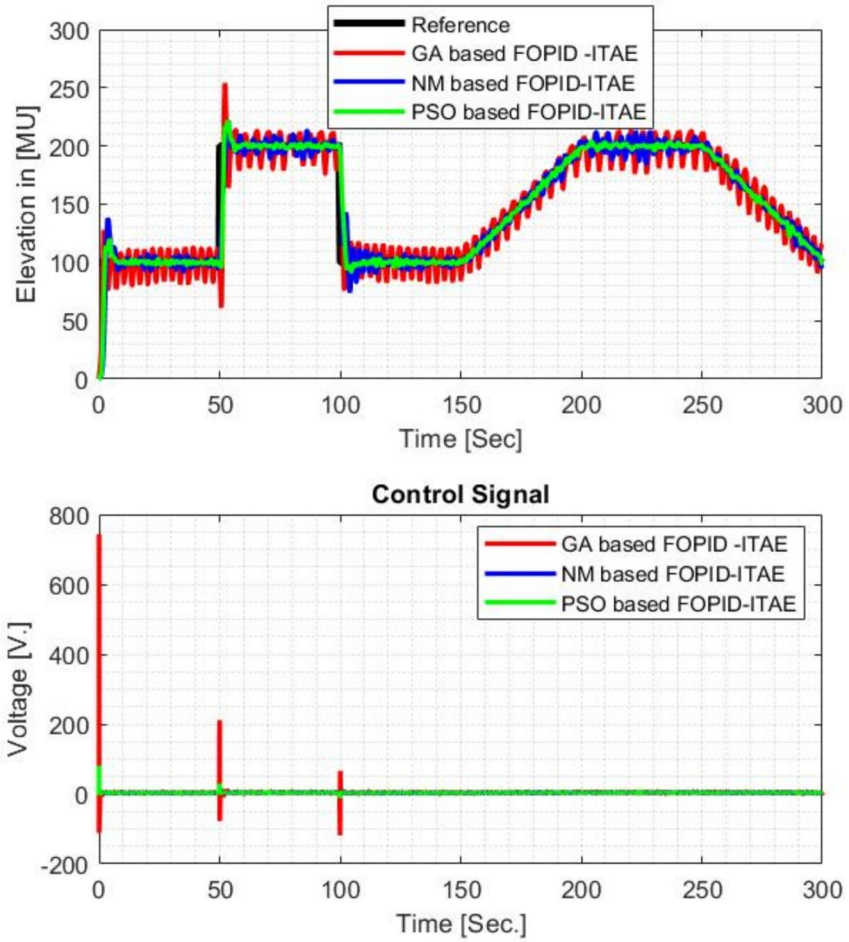

Fig. 18. FOPID controller comparison for elevation.

P.I	K_p	K_i	K_d	$\%M_p$	$t_r(sec.)$	$t_s(sec.)$	J
IOPID Controller - GA based tuning - for Azimuth							
ITSE	0.0098	0.0014	0.0059	7.7709	1.3951	13.0948	1.4666e+04
ITAE	0.0096	0.0020	0.0049	14.2635	1.3591	8.3896	4.5699e+02
ISE	0.0100	0.0012	0.0050	9.4683	1.3801	15.6546	8.8507e+03
IAE	0.0100	0.0024	0.0166	12.3214	1.2443	5.8363	2.4750e+02
IOPID Controller - PSO based tuning - for Azimuth							
ITSE	0.0099	0.0014	0.0066	5.9938	1.3898	13.1295	1.4520e+04
ITAE	0.0098	0.0012	0.0105	1.2315	1.4238	16.3083	3.0306e+02
ISE	0.0100	0.0014	0.0053	9.5381	1.3710	13.5362	8.8330e+03
IAE	0.0100	0.0017	0.0075	5.6498	1.3581	10.2391	1.4655e+02
IOPID Controller - NM based tuning - for Azimuth							
ITSE	0.0197	0.0025	0.0115	10.9792	0.8390	9.1057	1.0036e+04
ITAE	0.0073	0.0011	0.0078	1.3654	4.3986	14.5584	6.5055e+02
ISE	0.0994	0.0026	0.0337	2.5911	0.2956	2.3816	5.3225e+03
IAE	0.0255	0.0025	0.0122	14.0440	0.7217	10.0553	99.3157

Table 6. Performance and parameter values for the IOPID controller (GA, PSO, NM) for azimuth.

$\lambda = 1.25$, enhancing flexibility in shaping the system dynamics. With this adjustment, the overshoot was reduced to 6.23% and the settling time improved to 7.49 seconds. This fractional modification enables finer tuning of the low-frequency gain, thereby improving the balance between overshoot reduction and settling-

Perf. Index	K_p	K_i	K_d	λ	μ	J
FOPID Controller - GA based tuning - for Azimuth						
ITSE	0.0991	0.0019	0.0382	1.0353	1.0635	8.3742e+03
ITAE	0.0097	0.0638	0.0541	0.9039	0.4643	329.17
ISE	0.0098	0.0912	0.0962	0.2296	0.6384	1000
IAE	0.0100	0.0781	0.0992	0.3642	0.5566	109.2917
FOPID Controller - PSO based tuning - for Azimuth						
ITSE	0.0100	0.0996	0.1000	0.4036	0.6388	7.1544e+03
ITAE	0.0071	0.0200	0.0338	1.0045	0.3775	327.15
ISE	0.0100	0.100	0.100	0.2311	0.6457	5.3938e+03
IAE	0.0100	0.0956	0.0977	0.3325	0.6667	116.07
FOPID Controller - NM based tuning - for Azimuth						
ITSE	0.0100	0.0051	0.0211	1.0210	0.5201	8.5483e+03
ITAE	0.0083	0.0013	0.0077	1.0040	0.9640	326.4679
ISE	0.0099	0.0053	0.0402	0.8582	0.4286	5.5785e+03
IAE	0.0100	0.0042	0.0187	1.0371	0.5338	114.0562

Table 7. Performance and parameter values for the FOPID controller tuning (GA, PSO, NM) for azimuth.

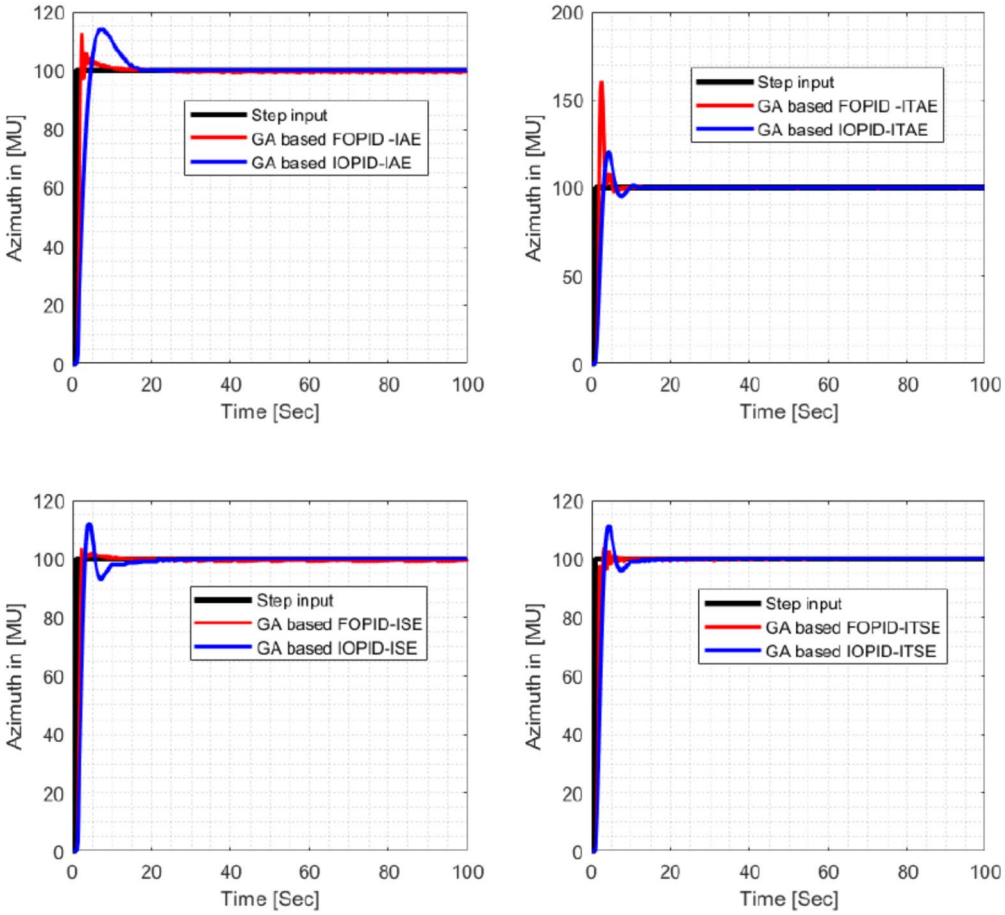

Fig. 19. GA based IOPID/FOPID Azimuth Response for various performance indices.

time performance. The comparative analysis highlights the potential advantages of fractional-order control in achieving more desirable transient behavior compared to the classical integer-order approach.

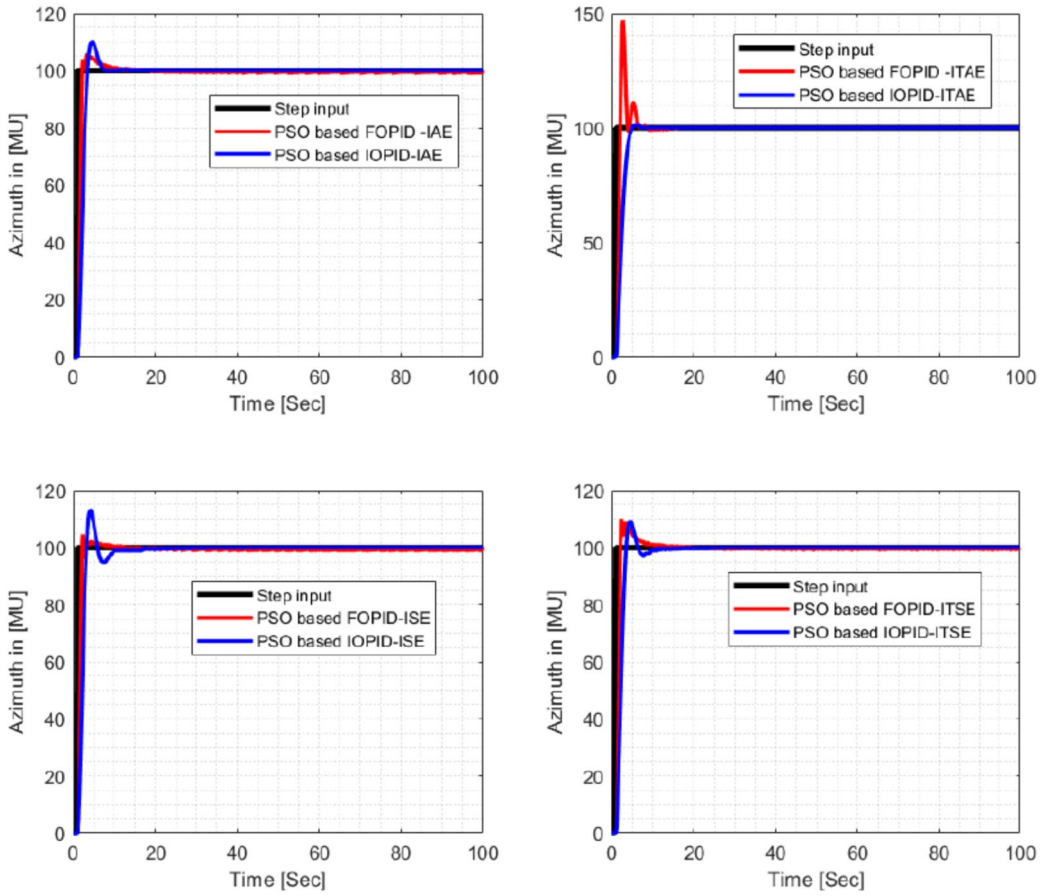

Fig. 20. PSO based IOPID/FOPID Azimuth response for various performance indices.

Conclusion

This study investigated the static characteristics of the main and tail rotors of the TRMS through real-time experiments. Additionally, a black-box system identification approach was employed. The results demonstrated that the linear plant model obtained using MATLAB's `fminsearch` function provides an accurate representation of the TRMS laboratory setup. Consequently, this approach was selected as a benchmark for testing and evaluating controller algorithms. In addition, paper investigated the real-time application of FOPID controllers for a TRMS, optimized using PSO, GA, and the NM method. The primary objective was to address the gap between simulation-based validations and real-world implementation of fractional-order control strategies. While previous works have predominantly focused on simulation environments, this study demonstrated the feasibility and effectiveness of deploying optimized FOPID controllers on physical hardware. The controllers were tuned using the FOMCON toolbox in MATLAB, and their performance was validated in both simulation and real-time experimental setups. Quantitative comparisons based on standard time-domain performance metrics (IAE, ITAE, ISE, ITSE) revealed that FOPID controllers consistently outperformed traditional integer-order PID controllers in both elevation and azimuth control. Notably, the GA-based FOPID controller yielded the best results in minimizing IAE, achieving values of 180.33 and 109.2 for elevation and azimuth respectively, compared to 247.05 for the GA-based IOPID in azimuth. Additionally, PSO and NM methods also showed significant performance improvement, further supporting the practical advantage of fractional-order control with intelligent optimization. The major contributions of this work include: (1) a complete methodology from controller optimization to hardware deployment; (2) a comparative analysis of PSO, GA, and NM optimization strategies; and (3) real-time validation on a benchmark system. These results establish that FOPID controllers are not only theoretically robust but also practically viable, with superior control precision.

Limitations and future work: Although the real-time implementation confirms the advantages of FOPID controllers, the paper does not address the sensitivity of FOPID performance to small parameter variations (e.g., λ, and μ), which is essential for robustness evaluation. Future research should focus on sensitivity and robustness analysis of the FOPID parameters under different system uncertainties and disturbances. Moreover, extending this framework to other complex real-world systems can further validate its generalizability and scalability. Additionally, future research will focus on enhancing the accuracy and robustness of the TRMS model by extending the system identification process to fractional-order models. This approach is expected to provide a more precise representation of system dynamics, capturing intricate nonlinearities and coupling effects more effectively. Moreover, the implementation of a 2-DOF control strategy will be explored to address these

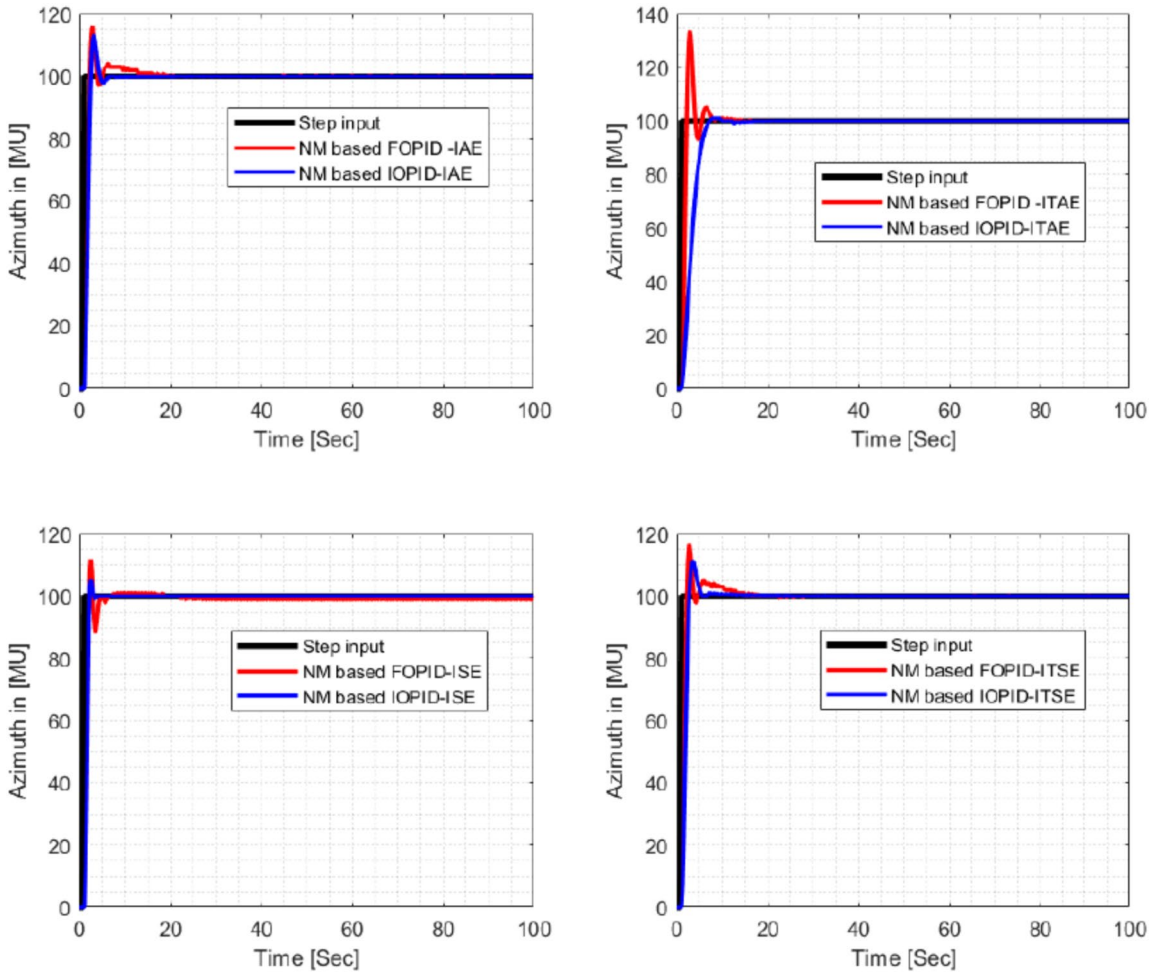

Fig. 21. NM based IOPID/FOPID Azimuth Response for various performance indices.

nonlinearities and improve overall system stability under varying operating conditions. Further investigations will be conducted into adaptive and robust control methodologies, such as adaptive FOPID controllers and H-infinity control, to enhance resilience against parameter variations and external disturbances. Lastly, additional experimental validation will be carried out on nonlinear TRMS models, incorporating aerodynamic effects and real-world uncertainties to assess the performance of the proposed control strategies under realistic conditions. These research directions will contribute to the development of more advanced and efficient control methodologies for complex dynamic systems.

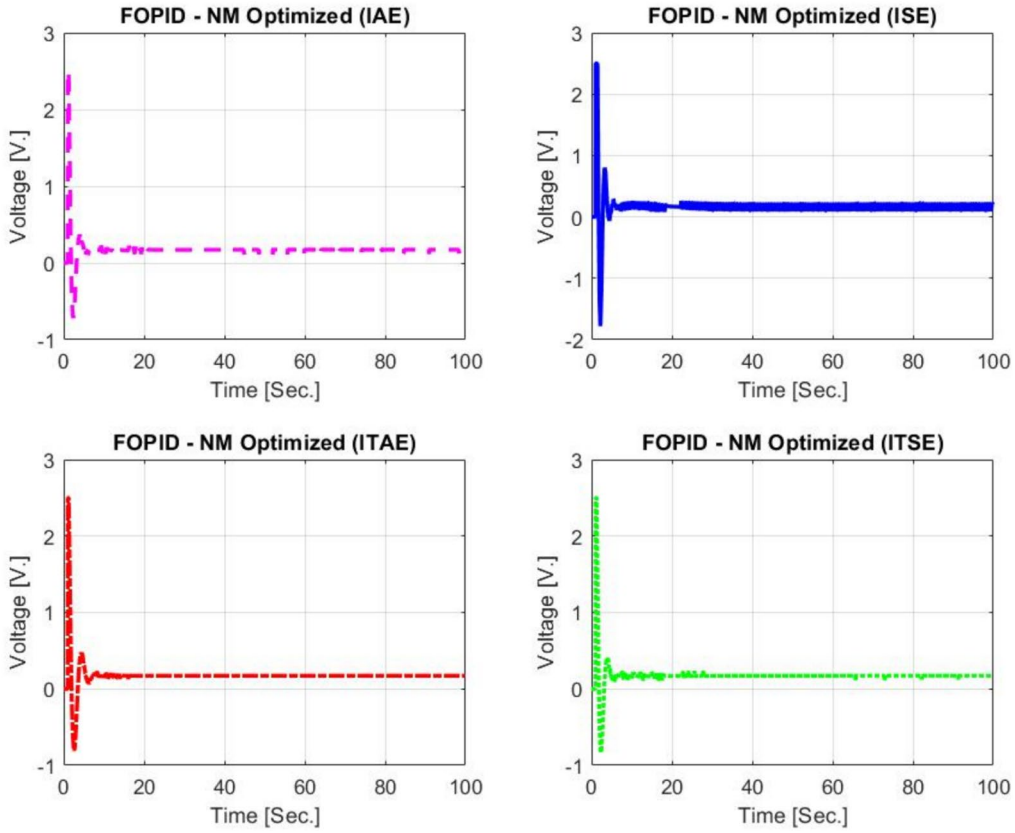

Fig. 22. Control effort of azimuth NM-optimized FOPID controllers based on different performance indices.

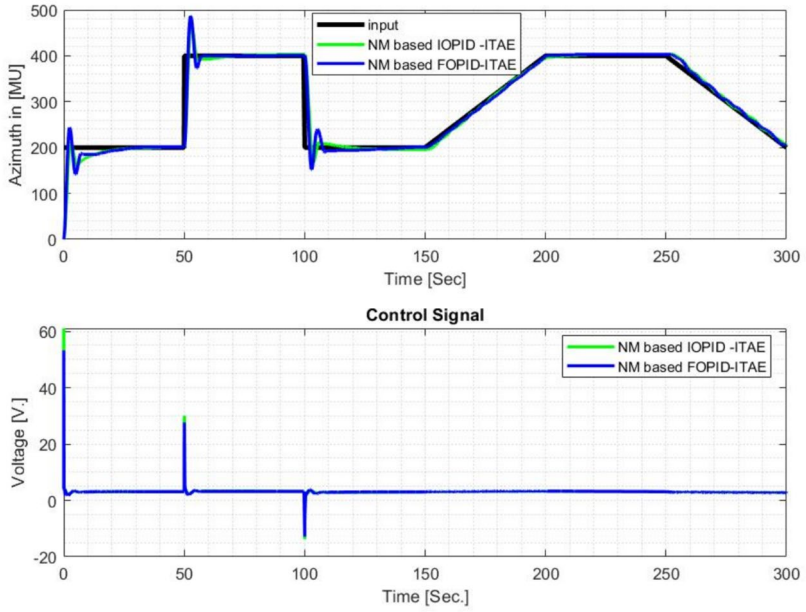

Fig. 23. IOPID/FOPID controller comparison - azimuth.

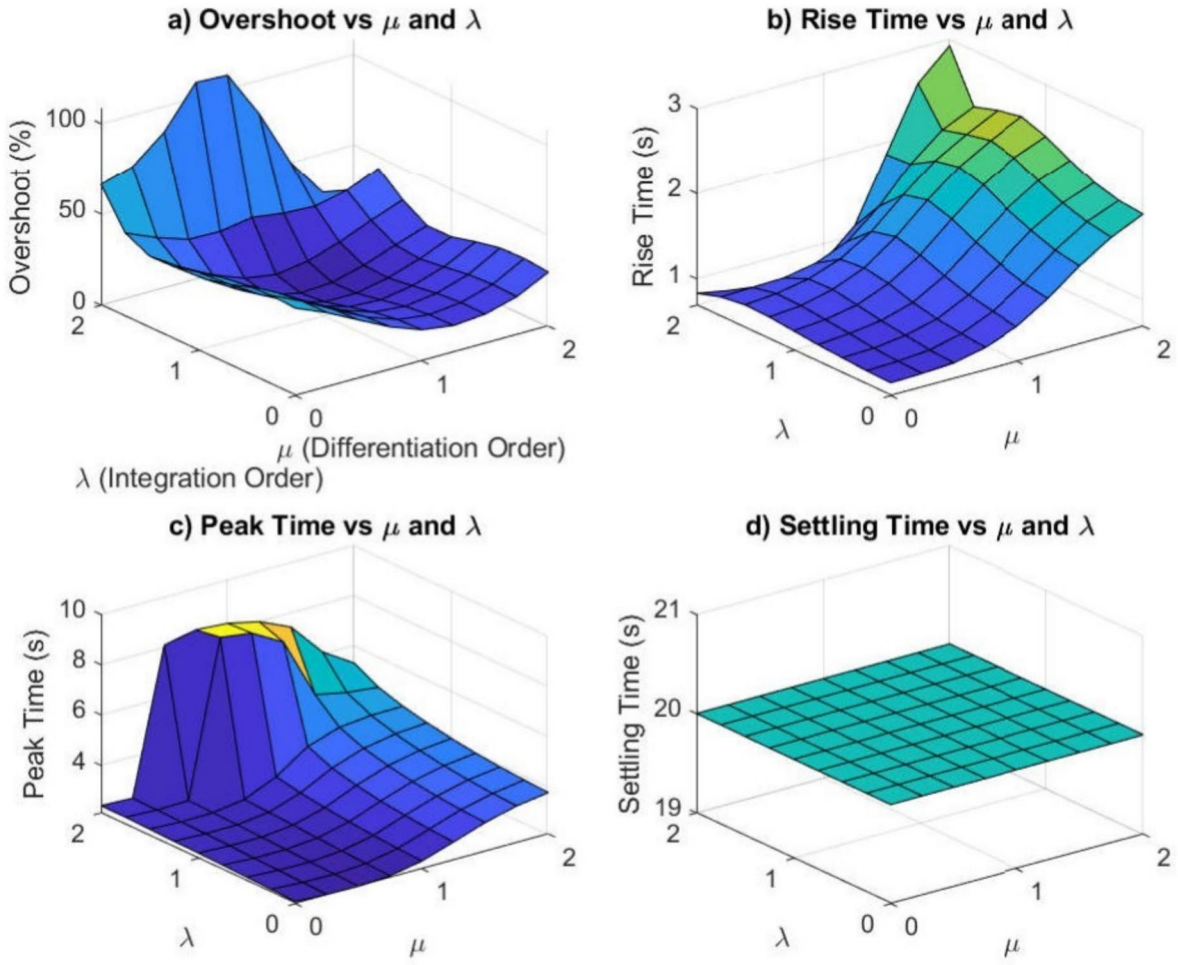

Fig. 24. Sensitivity analysis of the fractional orders μ and λ on time-domain specifications: (**a**) overshoot, (**b**) rise time, (**c**) peak time, and (**d**) settling time.

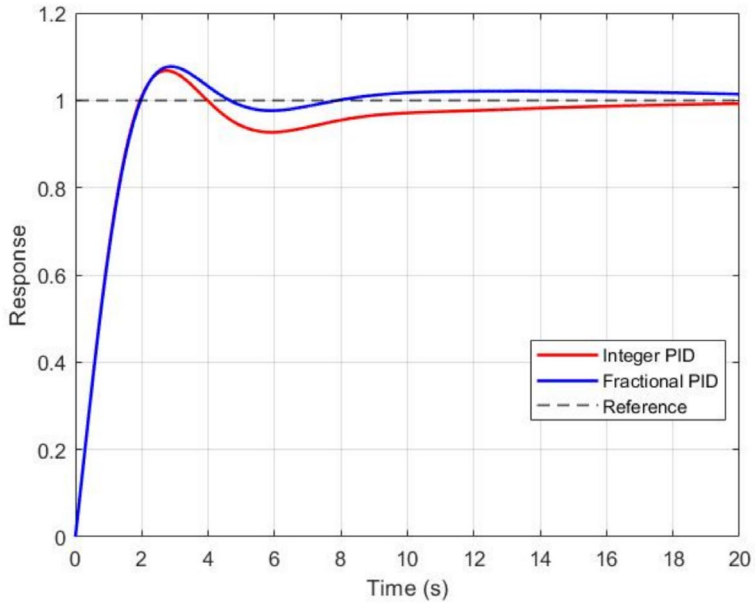

Fig. 25. Step response comparison between the Integer PID controller and the selected Fractional PID controller.

Data availability

The datasets supporting the conclusions of this study are included in the article. For further investigation, additional data will be made available upon request to the corresponding author.

Received: 5 May 2025; Accepted: 3 September 2025
Published online: 29 September 2025

References

1. Chen, Y., Petras, I. & Xue, D. Fractional order control-a tutorial. In *2009 American control conference*, 1397–1411 (IEEE, 2009).
2. Oustaloup, A., Levron, F., Mathieu, B. & Nanot, F. M. Frequency-band complex noninteger differentiator: characterization and synthesis. *IEEE Trans. Circuits Syst. I: Fundamental Theory Appl.* **47**, 25–39 (2000).
3. Ladaci, S. & Bensafia, Y. Indirect fractional order pole assignment based adaptive control. *Eng. Sci. Technol. Int. J.* **19**, 518–530 (2016).
4. Malek, H., Luo, Y. & Chen, Y. Identification and tuning fractional order proportional integral controllers for time delayed systems with a fractional pole. *Mechatronics* **23**, 746–754 (2013).
5. Razminia, A. & Torres, D. F. Control of a novel chaotic fractional order system using a state feedback technique. *Mechatronics* **23**, 755–763 (2013).
6. Yu, P., Li, J. & Li, J. The active fractional order control for maglev suspension system. *Math. Probl. Eng.* **2015**, 129129 (2015).
7. Kumar, P., Narayan, S. & Raheja, J. Optimal design of robust fractional order pid for the flight control system. *Int. J. Comput. Appl.* **128**, 31–35 (2015).
8. Fei, L., Wang, J., Zhang, L., Ge, Y. & Li, K. Fractional-order pid control of hydraulic thrust system for tunneling boring machine. In *Intelligent Robotics and Applications, 470–480 (Springer* (eds Lee, J. et al.) (Berlin Heidelberg, Berlin, Heidelberg, 2013).
9. Tejado, I., Valerio, D., Pires, P. & Martins, J. Fractional order human arm dynamics with variability analyses. *Mechatronics* **23**, 805–812 (2013).
10. Silva, M. F., Machado, J. T. & Lopes, A. Fractional order control of a hexapod robot. *Nonlinear Dyn.* **38**, 417–433 (2004).
11. Manabe, S. The non-integer integral and its application to control systems. *J. IEE Japan* **80**, 589–597 (1960).
12. Oustaloup, A., Melchior, P., Lanusse, P., Cois, O. & Dancla, F. The crone toolbox for matlab. In CACSD 2000. I*EEE International Symposium on Computer-Aided Control System Design*, 190–195 (IEEE, 2000).
13. Lurie, B. J. Three-parameter tunable tilt-integral-derivative (tid) controller. Unpublished technical report or proprietary documentation (1994).
14. Podlubny, I. Fractional-order systems and fractional-order controllers. *Inst. Exp. Physics, Slovak Acad. Sci. Kosice* **12**, 1–18 (1994).
15. Tepljakov, A., Petlenkov, E., Belikov, J. & Finajev, J. Fractional-order controller design and digital implementation using fomcon toolbox for matlab. In *2013 IEEE conference on computer aided control system design (CACSD)*, 340–345 (IEEE, 2013).
16. Tepljakov, A. & Tepljakov, A. Fomcon: fractional-order modeling and control toolbox. Fractional-order modeling and control of dynamic systems. In 107–129 (2017).
17. Dorcak, L. Numerical models for the simulation of the fractional-order control systems. *arXiv preprint* arXiv:math/0204108 (2002).
18. Monje, C. A., Vinagre, B. M., Feliu, V. & Chen, Y. Tuning and auto-tuning of fractional order controllers for industry applications. *Control Eng. Pract.* **16**, 798–812 (2008).
19. Rahideh, A., Shaheed, M. H. & Huijberts, H. J. C. Dynamic modelling of a trms using analytical and empirical approaches. *Control Eng. Pract.* **16**, 241–259 (2008).
20. Dragos, C.-A., Precup, R.-E., Preitl, S., Petriu, E. M. & Stinean, A.-I. Takagi-sugeno fuzzy control solutions for mechatronic applications. *Int. J. Artif. Intell.* **8**, 45–65 (2012).
21. Coelho, J. et al. Application of fractional algorithms in the control of a twin rotor multiple input-multiple output system. *ENOC* **2008**, 1–8 (2008).
22. Doğruer, T. & Tan, N. Real time control of twin rotor mimo system with pid and fractional order pid controller. *Mugla J. Sci. Technol.* **6**, 1–9 (2020).
23. Ijaz, S., Hamayun, M. T., Yan, L. & Mumtaz, M. F. Fractional order modeling and control of twin rotor aerodynamic system using nelder-mead optimization. *J. Electr. Eng. Technol.* **11**, 1863–1871 (2016).
24. Hore, A., Kumar, M. R. & Mishra, S. K. Parameter estimation for fractional-order with delay model of twin rotor mimo system. In *2021 IEEE 4th International Conference on Computing, Power and Communication Technologies (GUCON)*, 1–5 (IEEE, 2021).
25. Jabari, M. et al. An advanced pid tuning method for temperature control in electric furnaces using the artificial rabbits optimization algorithm. *Int. J. Dyn. Control* **13**, 1–15 (2025).
26. Wendimu, A. A. *Modeling and fractional-order control of twin rotor mimo system*. Master's thesis, Tomas Bata University in Zlín (2022).
27. Wendimu, A. A., Shaikh, I., Zerdazi, E. W. & Matúšu, R. Modeling, identification and analysis of twin rotor mimo systems. In *Computer Science On-line Conference*, 457–471 (Springer, 2024).
28. Shahri, M. E., Balochian, S., Balochian, H. & Zhang, Y. Design of fractional order pid controllers for time delay systems using differential evolution algorithm. *Indian journal Sci. Technol.* **7**, 1307–1315 (2014).
29. Rezaei Estakhrouiyeh, M., Gharaveisi, A. & Vali, M. Fractional order proportional-integral-derivative controller parameter selection based on iterative feedback tuning. case study. Ball levitation system. *Trans. Inst. Meas. Control* **40**, 1776–1787 (2018).
30. Jaiswal, S., Suresh Kumar, C., Seepana, M. M. & Babu, G. U. B. Design of fractional order pid controller using genetic algorithm optimization technique for nonlinear system. *Chem. Prod. Process Model.* **15**, 20190072 (2020).
31. Eltayeb, A., Ahmed, G., Imran, I. H., Alyazidi, N. M. & Abubaker, A. Comparative analysis: Fractional pid vs. pid controllers for robotic arm using genetic algorithm optimization. *Automation* **5**, 230–245 (2024).
32. Chakraborty, S. & Mondal, A. Tsa-aided it2flc-(1+ pd)-fopid control for regulating the first-order plus time-delayed non-conventional multi-area power system with deregulation. *Electrical Engineering* 1–21 https://doi.org/10.1007/s00202-025-03103-w (2025).
33. Mondal, A. & Chakraborty, S. Design of two-loop fopid-fopi controller for inverted cart-pendulum system. *Eng. Res. Express* **6**, 035354 (2024).
34. Idir, A. et al. Pid controller design with a new method based on fractionalized integral gain for cruise control system. In *2024 IEEE International Conference on Environment and Electrical Engineering and 2024 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I &CPS Europe)*, 1–6 (IEEE, 2024).
35. Abdulwahhab, O. W. Design of a complex fractional order pid controller for a first order plus time delay system. *ISA transactions* **99**, 154–158 (2020).
36. Shah, P., Sekhar, R., Iswanto, I. & Shah, M. Complex order pi a+ jb d c+ jd controller design for a fractional order dc motor system. *Adv. Sci. Technol. Eng. Syst. J.* **6**, 541–551 (2021).
37. Shah, P. & Agashe, S. Experimental analysis of fractional pid controller parameters on time domain specifications. *Prog.Fract. Differ. Appl.* **3**, 141–154 (2017).

38. Mondal, R. & Dey, J. A novel design methodology on cascaded fractional order (fo) pi-pd control and its real time implementation to cart-inverted pendulum system. *ISA transactions* **130**, 565–581 (2022).
39. Shanthini, C., Devi, V. K., Rajendran, S. & Jena, D. Comparative analysis of pid, i-pd and fractional order pi-pd for a dc-dc converter. In *2022 IEEE North Karnataka Subsection Flagship International Conference (NKCon)*, 1–5 (IEEE, 2022).
40. Muresan, C. I., Dulf, E. H. & Both, R. Vector-based tuning and experimental validation of fractional-order pi/pd controllers. *Nonlinear Dyn.* **84**, 179–188 (2016).
41. Roong, A. S. C., Shin-Homg, C. & Said, M. A. B. Position control of a magnetic levitation system via a pi-pd control with feedforward compensation. In *2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 73–78 (IEEE, 2017).
42. Bingi, K., Ibrahim, R., Karsiti, M. N., Hassan, S. M. & Harindran, V. R. Real-time control of pressure plant using 2dof fractional-order pid controller. *Arab. J. for Sci. Eng.* **44**, 2091–2102 (2019).
43. Ozyetkin, M. M. A simple tuning method of fractional order piλ-pdμ controllers for time delay systems. *ISA transactions* **74**, 77–87 (2018).
44. Bin Roslan, M. N., Bingi, K., Devan, P. A. M. & Ibrahim, R. Design and development of complex-order pi-pd controllers: Case studies on pressure and flow process control. *Appl. Syst. Innov.* **7**, 33 (2024).
45. Affenzeller, M. et al. White box vs. black box modeling: On the performance of deep learning, random forests, and symbolic regression in solving regression problems. In *International Conference on Computer Aided Systems Theory*, 288–295 (Springer, 2019).
46. Bingi, K., Rajanarayan Prusty, B. & Pal Singh, A. A review on fractional-order modelling and control of robotic manipulators. *Fractal Fract.* **7**, 77 (2023).
47. Krell, M. & Hergert, S. The black box approach: Analyzing modeling strategies. In *Towards a Competence-Based View on Models and Modeling in Science Education*, 147–160 (Springer, 2019).
48. Wendimu, A.A., Matušů, R., Shaikh, I., Wolde, M.K. (2025). Fractional-Order Identification and Analysis of Elevation and Azimuth Dynamics in a Twin Rotor System. In: Arai, K. (eds) Intelligent Systems and Applications. IntelliSys 2025. Lecture Notes in Networks and Systems, vol 1567. Springer, Cham. https://doi.org/10.1007/978-3-032-00071-2_37
49. Eykhoff, P. System identification? a survey. *Automatica* **7**, 123–162 (1971).
50. Åström, K. J. & Hägglund, T. *Advanced PID control* (Systems and Automation Society, ISA-The Instrumentation, 2006).
51. Podlubny, I. Fractional-order systems and pi/sup/spl lambda//d/sup/spl mu//-controllers. *IEEE Trans. Autom. Control* **44**, 208–214 (1999).
52. Faieghi, M. R. & Nemati, A. On fractional-order pid design. In *Applications of MATLAB in Science and Engineering* (IntechOpen, 2011).
53. Monje, C. A., Chen, Y., Vinagre, B. M., Xue, D. & Feliu-Batlle, V. Fractional-order systems and controls: fundamentals and applications (Springer Science & Business Media, 2010).
54. Oustaloup, A., Levron, F., Mathieu, B. & Nanot, F. M. Frequency-band complex noninteger differentiator: characterization and synthesis. *IEEE Transactions on circuits and systems I: Fundamental theory and applications* **47**, 25–39 (2002).
55. Xue, D., Zhao, C. & Chen, Y. A modified approximation method of fractional order system. In *2006 international conference on mechatronics and automation*, 1043–1048 (IEEE, 2006).
56. Idir, A., Bensafia, Y., Khettab, K. & Canale, L. Performance improvement of aircraft pitch angle control using a new reduced order fractionalized pid controller. *Asian J. Control.* **25**, 2588–2603 (2023).
57. Shah, P., Agashe, S. & Vyawahare, V. System identification with fractional-order models: A comparative study with different model structures. *Prog. Fract. Differ. Appl.* **4**, 533–552 (2019).
58. Ang, K. H., Chong, G. & Li, Y. Pid control system analysis, design, and technology. *IEEE Trans. Control Syst. Technol.* **13**, 559–576 (2005).
59. Normey-Rico, J. E. & Camacho, E. F. Control of dead-time processes (Springer, 2007).
60. Lucus, P., Webber, S. & Chew, J. PAC 2003: Proceedings of the 2003 Particle Accelerator Conference (IEEE, 2003).
61. Vilanova, R. & Visioli, A. PID control in the third millennium, vol. 75 (Springer, 2012).
62. Simpkins, A. System identification: Theory for the user, (ljung, l. 1999)[on the shelf]. *IEEE Robot. Autom. Mag.* **19**, 95–96 (2012).
63. Wendimu, A. A., Shaikh, I., Zerdazi, W., Matušů, R., Silhavy, R., Silhavy, P. Software Engineering Methods Design and Application: Proceedings of 13th Computer Science Online Conference 2024 Volume 1 Modeling Identification and Analysis of Twin Rotor MIMO Systems Springer Nature Switzerland Cham, 457–471 (2024).

Acknowledgements

This work was supported by the Internal Grant Agency of Tomas Bata University in Zlín under project number IGA/CebiaTech/2024/001. The authors would also like to thank the anonymous reviewers and editors for their valuable comments and constructive feedback, which greatly improved the quality of the manuscript.

Author contributions

A.A.W. conceptualized the study, curated the data, developed the methodology, validated the results, performed the visualizations, and prepared the original draft. R.M. supervised the research. I.Sh. and Z.K.K. contributed to reviewing and editing the manuscript. All authors reviewed the manuscript and contributed to the final version.

Declarations

Competing interests

The authors declare no competing interests.

Ethics statement

Review or approval by an ethics committee was not needed for this study because no data on human, human clinical studies, vertebrates or higher invertebrates or experimental animals was used in the article. Informed consent was not required for this study because no clinical data was produced in the review article.

Additional information

Correspondence and requests for materials should be addressed to R.M.

Reprints and permissions information is available at www.nature.com/reprints.