electronics

MDPI

Article

A Nesterov-Accelerated Hybrid PSO Optimization Approach for PID-Based Control of a Twin Rotor MIMO System

Ali Can Çabuker

Department of Electrical-Electronics Engineering, Faculty of Engineering, Artvin Coruh University, Artvin 08000, Türkiye; alicancabuker@artvin.edu.tr

Abstract

This study proposes a hybrid optimization algorithm that combines Particle Swarm Optimization (PSO) with the Nesterov Accelerated Gradient method for efficient PID controller tuning in a Twin Rotor MIMO System (TRMS). The proposed approach employs PSO to perform a global search over the six-dimensional PID parameter space and identify promising candidate solutions. When the optimization process stagnates during successive PSO iterations, a Nesterov momentum-based local search is selectively activated to refine the current best solution, reduce the risk of local minima, and accelerate convergence. The improved solution is then reintegrated into the PSO population, enabling the global search to continue from a higher-quality candidate. PID parameters are optimized using a multi-objective function that considers tracking error, overshoot, steady-state error, and control effort. The proposed hybrid method is evaluated through MATLAB 2019 simulations and compared with conventional PSO-based PID tuning. The results show that the hybrid PSO–Nesterov approach achieves a lower objective function value, improves convergence behavior, and enhances the transient response of the control system, demonstrating its effectiveness for TRMS PID parameter optimization.

Keywords: Twin Rotor MIMO System; particle swarm optimization; Nesterov acceleration; gradient-based optimization; PID tuning

1. Introduction

The Twin Rotor MIMO System (TRMS) is a challenging control benchmark characterized by two rotational degrees of freedom, namely pitch and yaw. Designing effective controllers for this system requires appropriate parameter selection while balancing multiple control-performance criteria, including reference tracking, overshoot, settling time, and steady-state error [1–3]. These requirements make PID controller tuning an important problem in TRMS control, and several conventional and optimization-based approaches have been investigated to improve closed-loop performance [4–7].

Particle Swarm Optimization (PSO) is a population-based metaheuristic that has been widely used for continuous optimization problems since its introduction by Kennedy and Eberhart [8]. Its population-based search mechanism makes it suitable for PID parameter tuning, where multiple controller gains must be selected within predefined bounds. However, PSO may experience premature convergence or stagnation when the swarm loses its ability to explore promising regions of the search space [9–11]. Although several approaches have been proposed to address stagnation and improve search efficiency [12], maintaining an effective balance between global exploration and local refinement remains an important challenge in metaheuristic optimization.

Check for updates

Academic Editor: Fabio Corti

Received: 24 September 2026
Revised: 17 October 2026
Accepted: 20 October 2026
Published: 22 October 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Gradient-based optimization methods provide an alternative mechanism for refining candidate solutions locally. Nesterov's accelerated gradient method incorporates a momentum-based update and evaluates gradient information at an extrapolated point, with convergence properties established for specific classes of optimization problems [12,13]. However, the effectiveness of incorporating a gradient-based method into a population-based metaheuristic depends on how and when the local search is activated, how its computational cost is controlled, and how the resulting candidate solution is reintegrated into the population.

To investigate these issues, this study presents a stagnation-triggered hybrid PSO–Nesterov framework for PID parameter tuning. In the proposed framework, PSO performs the population-based search over the controller parameter space. When the improvement in the global-best objective value remains below a predefined threshold for a specified number of consecutive iterations, a Nesterov-based local refinement procedure is activated. The refined candidate is subsequently evaluated and returned to the PSO population according to the prescribed acceptance and replacement rules. This design aims to coordinate global exploration and local refinement through an explicit stagnation-detection mechanism rather than applying local optimization at every iteration.

The main contributions of this study are as follows:

A stagnation-triggered hybrid PSO–Nesterov framework is formulated for PID parameter tuning, combining population-based exploration with conditional gradient-based local refinement.

A stagnation-detection mechanism is incorporated to determine when local refinement should be activated based on the improvement of the global-best objective value over consecutive iterations.

A candidate reintegration strategy is defined to allow an accepted local-search solution to influence subsequent PSO iterations through population replacement.

The proposed framework is formulated for PID parameter tuning in the pitch and yaw channels of the TRMS, with an objective function designed to account for the selected control-performance criteria.

The proposed method is intended to address stagnation during PSO-based PID tuning by coordinating local refinement with the ongoing population search. Its effectiveness must be assessed through comparisons with appropriate baseline methods under consistent computational budgets and through an evaluation of the resulting closed-loop control performance.

The methodological contribution of this study lies in the conditional coordination of global PSO exploration and Nesterov-based local refinement through a stagnation-detection mechanism and a defined population reintegration strategy. Rather than applying local search continuously or according to a fixed schedule, the proposed framework activates local refinement in response to a lack of progress in the global-best solution. The effectiveness of this coordination mechanism is evaluated by comparing the proposed method with appropriate baseline algorithms under matched objective-function evaluation budgets.

2. Materials and Methods

Metaheuristic optimization methods have been widely investigated for PID controller parameter tuning because they can explore bounded parameter spaces without requiring analytical gradient information [14]. Among these methods, Particle Swarm Optimization (PSO) is particularly relevant to controller tuning problems involving multiple gains and competing performance criteria. Nevertheless, the search performance of PSO may deteriorate when the swarm loses diversity or when the global-best solution stops improving.

These limitations motivate the investigation of hybrid optimization strategies that combine population-based exploration with local refinement.

Several studies have investigated hybrid optimization approaches to improve the search performance of PSO. Pelley et al. [15] combined PSO with Sequential Quadratic Programming (SQP) for constrained optimization problems. In this approach, PSO explores the search space, while SQP refines candidate solutions locally. This work provides an example of combining population-based search with a mathematical programming method to address optimization problems in which local refinement is beneficial.

Optimization-based approaches have also been investigated for TRMS controller design. Maiti et al. [16] applied PSO to parameter estimation and PID controller tuning for a nonlinear two-degree-of-freedom TRMS. Their study demonstrates the applicability of PSO-based optimization to TRMS control problems. Neeraj and Seema [4] investigated conventional PID and cross-PID control and considered PSO-based and hybrid optimization strategies, including a hybrid PSO–Nelder–Mead approach. These studies establish the relevance of population-based optimization and hybrid search methods to TRMS controller design.

Hybridization has also been investigated in applications beyond TRMS control. Zulu et al. [17] proposed a hybrid PSO–Gradient Descent method for estimating photovoltaic hosting capacity in distribution networks. Their work illustrates the use of gradient-based local search alongside population-based exploration. However, the suitability of a hybrid optimization strategy depends not only on the selected local search method but also on how local refinement is scheduled and how the resulting candidate solutions are incorporated into the population-based search [18].

Nesterov's accelerated gradient method introduces a momentum-based update that evaluates gradient information at an extrapolated point [12–15]. Related work has also examined the theoretical properties of accelerated gradient and heavy-ball methods [19]. These methods provide a basis for investigating gradient-based local refinement; however, their practical effectiveness within a metaheuristic framework depends on the objective function, the local-search implementation, and the criteria used to activate and accept refined solutions.

Based on these observations, the present study investigates a hybrid PSO–Nesterov framework in which local refinement is activated according to a stagnation-detection criterion. PSO continues to perform population-based exploration, while the Nesterov-based procedure is initiated when the global-best objective value fails to improve sufficiently over a predefined number of consecutive iterations. The resulting candidate is evaluated and reintegrated into the population only according to the specified acceptance and replacement rules. The intended distinction of this framework is therefore its coordination of stagnation detection, conditional local refinement, and population reintegration, rather than the use of PSO and a gradient-based method in combination alone.

The proposed framework is formulated for PID parameter tuning using the selected control-performance objective. Its effectiveness should be evaluated against suitable baseline methods using consistent objective-function evaluation budgets and repeated independent runs. These comparisons are necessary to distinguish improvements attributable to the proposed coordination mechanism from improvements that may result simply from additional optimization evaluations.

3. Twin Rotor MIMO System

The Twin Rotor MIMO System (TRMS) is an experimental control platform with two degrees of freedom (2-DOF) that exhibits dynamic characteristics similar to those of a helicopter. Unlike an actual helicopter, the TRMS is fixed to a supporting platform and

therefore does not perform free flight. The system consists of two coupled rotational axes, namely the pitch and yaw axes. The motion of each axis is actuated by DC motors driving the main and auxiliary rotors, respectively. The primary objective of the TRMS control system is to ensure that the pitch and yaw angles track their corresponding reference values either independently or simultaneously while maintaining the desired angular positions.

The TRMS can be operated under single-axis conditions, where only one of the pitch or yaw axes is controlled, or under two-axis conditions, where both the main and auxiliary motors operate simultaneously. Under two-axis operation, the dynamic interaction between the two axes introduces significant coupling effects and increases the complexity of the MIMO control problem. These characteristics make the TRMS a suitable experimental platform for investigating control strategies for nonlinear and strongly coupled systems. Figure 1 illustrates the general structure of the TRMS, including the DC motors driving the main and auxiliary rotors and the horizontal and vertical forces acting on the system [20].

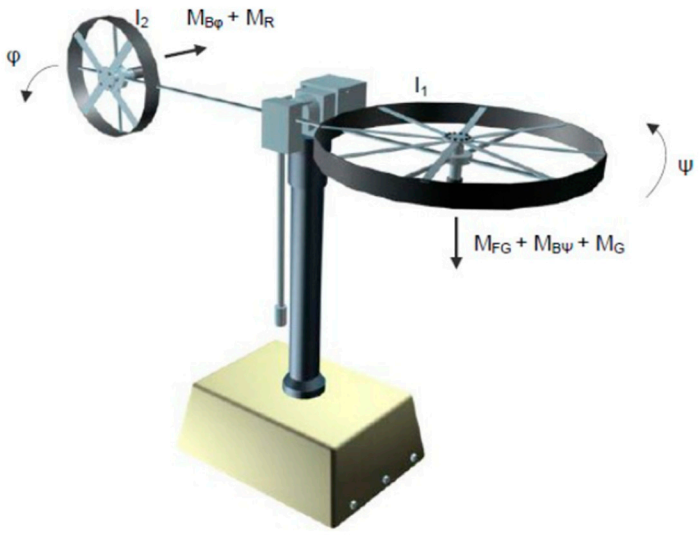


Figure 1. TRMS Laboratory Model [20].

The TRMS provides two angular position outputs, namely the pitch angle (ψ) and the yaw angle (ϕ). Its mathematical representation can be formulated by considering two interconnected subsystems: the DC motor dynamics associated with the main and tail rotors, and the mechanical dynamics governing the rotational motion of the system in the vertical and horizontal planes. The resulting rotor-generated torques directly influence the pitch and yaw motions, respectively. Accordingly, the torque equations governing the pitch and yaw dynamics are given by (1) and (2), respectively [21,22].

$$\frac{d\tau_1}{dt} = -\frac{T_{10}}{T_{11}}\tau_1 + \frac{k_1}{T_{11}}u_1 \quad (1)$$

$$\frac{d\tau_2}{dt} = -\frac{T_{20}}{T_{21}}\tau_2 + \frac{k_2}{T_{21}}u_2 \quad (2)$$

In Equations (1) and (2), T_{10}, T_{11}, T_{20}, and T_{21} denote the denominator parameters of the corresponding motor dynamics, while k_1 and k_2 represent the motor gains. The torques generated by the main and tail rotors subsequently determine the mechanical motion of the TRMS along its two rotational axes. The corresponding mechanical dynamics, describing the vertical and horizontal motions of the system, are formulated in Equations (3) and (4), respectively.

For vertical movement,

$$I_1\psi'' = M_1 - M_{FG} - M_{B\psi} - M_G \quad (3)$$

For horizontal movement,

$$I_2\phi'' = M_2 - M_{B\phi} - M_R \tag{4}$$

The horizontal and vertical expressions are provided in Equations (5)–(11).

$$M_1 = \alpha_1 \cdot \tau_1^2 + b_1 \cdot \tau_1 \tag{5}$$

$$M_2 = \alpha_2 \cdot \tau_2^2 + b_2 \cdot \tau_2 \tag{6}$$

$$M_{FG} = M_g \cdot \sin\psi \tag{7}$$

$$M_{B\psi} = B_{1\psi} \cdot \psi + \frac{0.0326}{2} \cdot \sin(2 \cdot \psi) \cdot \phi^2 \tag{8}$$

$$M_{B\phi} = B_{1\phi} \cdot \phi \tag{9}$$

$$M_G = k_{gy} \cdot M_1 \cdot \phi \cdot \cos\psi \tag{10}$$

$$M_R = \frac{kc \cdot (T_0 \cdot s + 1)}{(T_p \cdot s + 1)} \tag{11}$$

The TRMS model parameters are provided in Equations (1)–(11). I1 and I2 are moments of inertia for the vertical and horizontal movements of the rotors, respectively; M1 and M2 reflect the nonlinear static characteristics; α is momentum due to gravity; Mg is the moment of gravitational force; MBψ and MBϕ are the momentum caused by friction forces; B1ψ and B1ϕ are the friction momentum function parameters for the respective rotors; MG is the momentum from the gyroscopic effects; and kgy is the gyroscopic momentum parameter. The cross-reaction momentum between the two axes, MR, is the output of a linear system [23–26]. Table 1 defines the expressions and numerical parameter values used in the TRMS equations.

Table 1. TRMS model parameters.

Parameter	Value
I1—moment of inertia of vertical rotor	6.8×10^{-2} kg.m^2
I2—moment of inertia of horizontal rotor	2×10^{-2} kg.m^2
α1—static characteristic parameter	0.00135
b1—static characteristic parameter	0.0924
α2—static characteristic parameter	0.02
b2—static characteristic parameter	0.09
Mg—gravity momentum	0.32 N.m
B1—friction momentum function parameter	6×10^{-3} Nms/rad
B2—friction momentum function parameter	1×10^{-3} Nms^2/rad
B1—friction momentum function parameter	1×10^{-1} Nms/rad
B2—friction momentum function parameter	1×10^{-2} Nms^2/rad
Kgy—gyroscopic momentum parameter	0.05 s/rad
k1—motor 1 gain	1.1
k2—motor 2 gain	0.8
T11—motor 1 denominator parameter	1.1
T10—motor 1 denominator parameter	1
T21—motor 2 denominator parameter	1
T20—motor 2 denominator parameter	1
Tp—cross-reaction momentum parameter	2
T0—cross-reaction momentum parameter	3.5
kc—cross-reaction momentum gain	−0.2

Although the physical TRMS exhibits coupled nonlinear dynamics, the present study uses separate transfer-function models for the pitch and yaw channels. The nonlinear

coupled equations presented in the literature describe the physical behavior of the TRMS but are not directly implemented in the MATLAB simulations conducted in this study. Instead, the PID parameters are optimized using the two transfer functions specified below. Consequently, the results should be interpreted within the scope of these separate channel models and should not be considered a validation of the proposed method on a fully coupled nonlinear TRMS model. The transfer function of the TRMS, derived from the given values, is also shown in Equations (12) and (13) [23].

$$G_p = \frac{1.359}{s^3 + 0.997s^2 + 4.786s + 4.278} \tag{12}$$

$$G_y = \frac{3.6}{s^3 + 6s^2 + 5s} \tag{13}$$

4. PID Controller Structure

The TRMS is controlled using two PID controllers corresponding to the pitch and yaw channels. The control objective is to minimize the tracking error between each reference angle and its corresponding measured output. Various control strategies have been investigated for TRMS applications, including PID parameter tuning, multivariable control design, fractional-order control, and hybrid intelligent control approaches [26–30]. In the present study, the six PID controller parameters are optimized using the proposed hybrid PSO–Nesterov algorithm. The simulation model used for this optimization consists of the separate pitch and yaw transfer functions specified in Section 3.

In this study, two independent PID controllers are considered for reference tracking in the pitch and yaw channels of the Twin Rotor MIMO System (TRMS). Each controller generates a control signal based on the tracking error between the desired reference angle and the corresponding measured output. The controller parameters are optimized to improve the tracking performance of both channels.

The tracking errors for the pitch and yaw channels are defined as the differences between their respective reference signals and output angles. The corresponding PID control laws are expressed in Equations (14) and (15), while the controller implementations are described by Equations (16) and (17). The resulting control signals are applied to the main and auxiliary rotor channels, respectively.

$$e_p(t) = r_p(t) - \psi_p(t) \tag{14}$$

$$e_y(t) = r_y(t) - \phi_p(t) \tag{15}$$

PID controller structure of the system is formulated in Equations (14) and (15) and expressed as $r_p(t)$ pitch reference, $\psi_p(t)$ measured pitch angle, $r_y(t)$ yaw reference, and $\phi_p(t)$ measured yaw angle.

$$u_p(t) = K_{p,p}e_p(t) + K_{i,p}(t)\int_0^t e_p(\tau)d_\tau + K_{d,p}\frac{de_p(t)}{dt} \tag{16}$$

$$u_y(t) = K_{p,y}e_y(t) + K_{i,y}(t)\int_0^t e_y(\tau)d_\tau + K_{d,y}\frac{de_y(t)}{dt} \tag{17}$$

Here u_p and u_y, represent the control signals applied to the main and auxiliary motors, respectively. The optimization problem involves six PID parameters: the proportional, integral, and derivative gains for each channel.

$$\mathrm{K} = [K_{p,p}, K_{i,p}, K_{d,p}, K_{p,y}, K_{i,y}\ K_{d,y}] \tag{18}$$

where the subscripts (p) and (y) denote the pitch and yaw channels, respectively. The objective of the optimization procedure is to identify a suitable parameter vector that minimizes

the specified performance index while satisfying the parameter bounds. The TRMS has two controlled axes, pitch and yaw, each governed by a PID controller. Consequently, the optimization problem involves six controller parameters. Although the physical TRMS exhibits dynamic coupling between the two axes, the simulations in this study use separate transfer-function models for the pitch and yaw channels. Therefore, the decision vector is defined as:

$$K^* = argmin\, J(K) \tag{19}$$

Here, denotes the decision vector consisting of the six PID controller parameters, while represents the objective function. In order for particle swarm optimization to optimise the TRMS, we need to define an objective function. The objective function, as defined in Equation (20), is given below.

$$J = w_{ISE,p}ISE_p + w_{ISE,y}ISE_y + w_{OS,p}OS_p + w_{OS,y}OS_y + w_{SSE,p}SSE_p + w_{SSE,y}SSE_y + w_{U,p}U_p + w_{U,y}U_y \tag{20}$$

The objective function combines four performance criteria: the integral of squared error (ISE), overshoot (OS), steady-state error (SSE), and control effort. ISE penalizes accumulated tracking errors, OS penalizes excessive excursions beyond the reference value, and SSE measures the residual tracking error at the end of the simulation. The control-effort term discourages unnecessarily large control signals. The mathematical expressions for these criteria are given in Equations (21)–(28), and the contributions of the pitch and yaw channels are combined using the weighting coefficients specified in the objective function.

Although the physical TRMS exhibits dynamic coupling between the pitch and yaw axes, the simulations conducted in this study use separate transfer-function models for the two channels, as specified in Section 3. Therefore, the optimization results characterize PID tuning for the implemented channel models and should not be interpreted as validation of the proposed method on a fully coupled nonlinear TRMS model.

$$ISE_p = \int_0^T e_p^2(t)dt \tag{21}$$

$$ISE_y = \int_0^T e_p^2(t)dt \tag{22}$$

$$OS_p = \max(0, \psi_{pmax} - r_p) \tag{23}$$

$$OS_y = \max(0, \phi_{ymax} - r_y) \tag{24}$$

$$SSE_p = \frac{1}{N_s}\sum_{k=N-N_s+1}^{N}\left|e_p(k)\right| \tag{25}$$

$$SSE_y = \frac{1}{N_s}\sum_{k=N-N_s+1}^{N}\left|e_y(k)\right| \tag{26}$$

$$U_p = \int_0^T U_p^2(t)dt \tag{27}$$

$$U_y = \int_0^T U_y^2(t)dt \tag{28}$$

The control effort is included in the objective function to prevent the control signal from becoming unnecessarily large.

5. Proposed Hybrid PSO–Nesterov Optimization Algorithm

The proposed hybrid optimization algorithm combines Particle Swarm Optimization (PSO) with a Nesterov-accelerated local search to optimize the six PID controller parameters. PSO performs population-based exploration of the parameter space, whereas the Nesterov method is employed to refine the current global best solution when the PSO search stagnates [10].

Stagnation is monitored by comparing the global best objective function value before and after each PSO iteration. If the objective function does not improve by at least a prescribed threshold of 10^{-6}, the stagnation counter is incremented. Otherwise, the counter is reset to zero. When the counter reaches eight consecutive iterations, the Nesterov-based local refinement procedure is activated.

The local search starts from the current global best parameter vector and uses a numerically approximated gradient to update the candidate solution. The Nesterov momentum coefficient, learning rate, finite-difference step size, and maximum local step magnitude are set to 0.85, 0.01, 10^{-4}, and 0.5, respectively. The local refinement procedure is allowed to run for up to 100 iterations.

After local refinement, the best candidate found during the local search is compared with the current global best solution. If the candidate yields a lower objective function value, it replaces the global best solution and is inserted into the population by replacing the particle with the highest current objective function value. The replaced particle's velocity is reset to zero, and its personal best is updated to the inserted solution. Regardless of whether an improvement is obtained, the stagnation counter is reset after the local search. This mechanism enables the hybrid algorithm to alternate between population-based exploration and stagnation-triggered local refinement.

5.1. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a population-based metaheuristic optimization algorithm inspired by the collective behavior of social organisms. In the proposed framework, PSO is employed as the primary global optimization mechanism to search for the optimal PID controller parameters. Each particle represents a candidate solution in the six-dimensional PID parameter space defined by the proportional, integral, and derivative gains of the pitch and yaw controllers.

The position vector of the i-th particle is defined as

$$\mathrm{K}= \left[K_{p,p}, K_{i,p}, K_{d,p}, K_{p,y}, K_{i,y} K_{d,y}\right]^T \tag{29}$$

where the subscripts (p) and (y) denote the pitch and yaw axes, respectively. The corresponding particle velocity is represented by

$$v_i = \left[v_{i,1}, v_{i,2} \ldots ., v_{i,6}\right]^T \tag{30}$$

At each iteration, the velocity of each particle is updated according to

$$v_i^{t+1} = w(t) \cdot v_i^t + c_1 \cdot r_1 \cdot \left(p_i^t - x_i^t\right) + c_2 \cdot r_2 \cdot \left(g^t - x_i^t\right) \tag{31}$$

where p_i^t denotes the best position previously visited by particle i, while g^t represents the global best position found by the swarm up to iteration. The parameters c_1 and c_2 are the cognitive and social acceleration coefficients, respectively, and r_1 and r_2 are independent random numbers uniformly distributed in the interval ([0,1]) [8].

The particle position is subsequently updated as

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{32}$$

To facilitate a gradual transition from exploration to exploitation, a linearly decreasing inertia weight is employed [31,32].

$$w(t) = w_{max} - \frac{w_{max} - w_{min}}{T_{max}} t \tag{33}$$

where w_{max} and w_{min} denote the maximum and minimum inertia weights, respectively, and T_{max} represents the maximum number of iterations. In the proposed implementation,

$w_{max} = 0.9$ and $w_{min} = 0.2$ are employed. The cognitive and social coefficients are set to $c_1 = 1.9$ and $c_2 = 1.9$. After each position update, the particle coordinates are constrained to the prescribed lower and upper bounds of the PID parameters. The objective function defined in Section 4 is then evaluated for each candidate solution. Based on the resulting objective function values, the personal best position of each particle and the global best position of the swarm are updated whenever an improvement is obtained. This procedure is repeated until the maximum number of PSO iterations is reached.

Although PSO provides a population-based global search mechanism, its global best solution may cease to improve significantly during the optimization process. To address this limitation, the proposed hybrid algorithm monitors the improvement of the global best objective function value at each iteration. When the improvement remains below a predefined threshold for a specified number of consecutive iterations, a Nesterov-accelerated local search is activated. The stagnation detection criterion and the local refinement procedure are described in the following subsection.

5.2. Stagnation Detection Mechanism

Although PSO provides an effective population-based global search mechanism, the improvement of the global best solution may become negligible during the optimization process. This behavior can limit search progress, particularly when the population continues exploring a restricted region of the solution space. To address this issue, the proposed method explicitly monitors the progress of the global best solution using a stagnation detection mechanism [33–35].

Let ΔJ^t denote the objective function value of the global best solution at iteration. The improvement achieved between two consecutive iterations is defined as

$$\Delta J^t = J_{best}^{t-1} - J_{best}^{t} \tag{34}$$

A positive value of ΔJ^t indicates an improvement in the global best solution, whereas a value close to zero indicates that the optimization process has made little or no progress. To distinguish meaningful improvements from negligible changes, an improvement threshold is ε introduced. An iteration is considered to provide insufficient improvement when $\Delta J^t < \varepsilon$ In the proposed implementation, the improvement threshold is set to $\varepsilon = 10^{-6}$. The number of consecutive iterations satisfying this condition is tracked using a stagnation counter S_t, updated as

$$S_t = \begin{cases} S_{t-1} + 1\ , & \Delta J^t < \varepsilon \\ 0\ , & \Delta J^t \geq \varepsilon \end{cases} \tag{35}$$

Accordingly, the stagnation counter is reset whenever a meaningful improvement is obtained, whereas it increases when the global best solution fails to improve sufficiently. A stagnation condition is declared when the counter reaches a predefined limit S_{max}. In this study, the stagnation limit is set to $S_{max} = 8$.

Thus, when the global best objective function value fails to improve by at least 10^{-6} for eight consecutive iterations, the PSO search is considered to have entered a stagnation state. At this point, the standard PSO update mechanism is complemented by the Nesterov-accelerated gradient-based local search described in the following subsection. The proposed stagnation detection mechanism serves as a switching criterion between the global and local optimization stages. Importantly, the Nesterov-based local search is not continuously applied throughout the optimization process. Instead, it is activated only when the PSO search exhibits insufficient progress. This selective activation preserves the global exploration capability of PSO during productive search phases while providing an additional local exploitation mechanism when the improvement of the global best solution becomes stagnant.

5.3. Nesterov Accelerated Local Search

When the stagnation detection mechanism identifies insufficient improvement in the PSO search process, a Nesterov-accelerated gradient-based local search is activated. Its purpose is to refine the current global best solution by exploiting the local structure of the objective function. Unlike conventional gradient descent, the Nesterov approach evaluates the gradient at a look-ahead position determined by the current momentum information. This allows the local search to account for the anticipated direction of motion when updating the solution [13,32].

Let the current solution at local iteration (k) be denoted by x^k, and let v^k represent the corresponding local search velocity. The look-ahead position is calculated as

$$z^k = x^k + \mu v^k \tag{36}$$

where μ is the momentum coefficient, set to $\mu = 0.85$ in the proposed implementation. Since an analytical gradient of the simulation-based objective function is not readily available, a central finite-difference approximation is employed [33,34]. For the (j)-th PID parameter, the numerical gradient is approximated as

$$\frac{\partial J}{\partial x_j} \approx \frac{J\left(z^k + \delta e_j\right) - J\left(z^k - \delta e_j\right)}{2\delta} \tag{37}$$

where e_j is the unit vector corresponding to the (j)-th decision variable and δ is the finite-difference perturbation. The resulting numerical gradient is expressed as

$$\nabla J\left(z^k\right) = \left[\frac{\partial J}{\partial x_1}, \frac{\partial J}{\partial x_2} \cdots, \frac{\partial J}{\partial x_6}\right]^T \tag{38}$$

In the proposed implementation, the finite-difference parameter is set to $\partial = 10^{-4}$. To limit excessively large gradient values, gradient scaling is applied when the gradient norm exceeds 10^{-6}. The local search velocity is then updated according to [35]

$$v^{k+1} = \mu v^k - \eta \nabla J(z^k) \tag{39}$$

where η is the local learning rate, set to $\eta = 0.01$. The candidate solution is updated as

$$x^{k+1} = x^k + v^{k+1} \tag{40}$$

The learning rate used in the proposed implementation is $\eta = 0.01$. Because the PID parameter space is bounded, the updated solution is constrained to remain within the predefined lower and upper parameter limits. In addition, the norm of the local search velocity vector is limited to 0.5 to prevent excessively large parameter updates.

$$\|x^{k+1} - x^k\| \leq 0.5 \tag{41}$$

The local refinement procedure is repeated for a predefined maximum number of iterations. In the current MATLAB implementation, the maximum number of Nesterov iterations is set to NN = 100 per invocation. During the local refinement process, the objective function is evaluated for each candidate solution, and the best solution found is retained as the locally improved candidate.

Importantly, the Nesterov search is initialized from the current global best solution rather than from an arbitrary point in the search space. Therefore, the local search exploits a promising region already identified by PSO, using the information accumulated by the swarm to further refine the solution when the global search becomes stagnant.

5.4. Integration of Nesterov Search into PSO

The Nesterov-accelerated local search is integrated into the PSO framework through a selective feedback mechanism. The main purpose of this integration is to allow the locally refined solution obtained during stagnation to contribute to the subsequent global search process. Thus, the Nesterov mechanism does not operate as an independent optimization algorithm within the hybrid framework; instead, it serves as a local refinement operator embedded in the PSO process.

When the stagnation condition defined in Section 5.2 is satisfied, the current global best solution, g^t, is used as the initial point of the Nesterov local search. After the local refinement procedure is completed, the best solution obtained, denoted by (x_N^*), is evaluated using the objective function. Its objective function value is compared with the current global best value:

$$J(x_N^*) < J(g^t) \tag{42}$$

If this condition is satisfied, the Nesterov-refined solution is accepted as the new global best:

$$\mathrm{g^t} \leftarrow \mathrm{x_N^*} \tag{43}$$

In addition to updating the global best solution, the improved candidate is fed back into the PSO population. Specifically, the particle with the highest current objective function value is identified and replaced by the Nesterov-refined solution:

$$x_{worst}^t \leftarrow x_N^* \tag{44}$$

where x_{worst}^t denotes the position of the particle with the highest current objective function value. The velocity of the replaced particle is reset to zero, and its current position, objective function value, and personal best information are updated accordingly. This population feedback mechanism enables the locally improved solution to participate directly in subsequent PSO iterations rather than being retained only as an external candidate.

If the Nesterov local search fails to produce an objective function value lower than the current global best, the existing global best solution and particle population are retained. In either case, the stagnation counter is reset after the local search is completed, and the PSO optimization process resumes. This reset prevents the same stagnation event from immediately triggering another local search without allowing further PSO iterations.

The proposed integration combines two complementary search mechanisms. PSO performs population-based exploration of the PID parameter space, whereas the Nesterov-accelerated local search refines a promising solution when the global search exhibits insufficient progress. By selectively activating local refinement and reinserting an improved candidate into the swarm, the proposed approach aims to balance global exploration and local exploitation.

5.5. Overall Procedure of the Proposed Hybrid Algorithm

The overall procedure of the proposed hybrid PSO–Nesterov optimization algorithm is summarized in this subsection. The algorithm initially generates a population of particles within the predefined lower and upper bounds of the six PID controller parameters and initializes their velocities within the corresponding velocity limits. The objective function is then evaluated for all particles, and the personal best and global best solutions are determined.

During each subsequent iteration, the particle velocities and positions are updated according to the standard PSO equations. The objective function is evaluated for each updated particle, and the personal and global best solutions are updated whenever an improved solution is obtained. The improvement of the global best objective function

is monitored using the predefined improvement threshold. If the improvement remains below this threshold, the stagnation counter is incremented; otherwise, it is reset. When the counter reaches the predefined stagnation limit, the current global best solution is transferred to the Nesterov accelerated local search.

The Nesterov mechanism performs a predefined number of local refinement iterations using the look-ahead position and numerical gradient information. During this process, the PID parameters are updated according to the Nesterov momentum mechanism while satisfying the predefined parameter bounds and local step limitation [13]. After the local refinement, the best solution obtained is compared with the current global best solution. If the Nesterov-refined solution provides a lower objective function value, it is accepted as the new global best solution and inserted into the PSO population by replacing the particle with the worst objective function value. The stagnation counter is subsequently reset, allowing the PSO search to continue from the improved population.

If the Nesterov search does not provide an improved solution, the current global best solution and PSO population are retained. This process is repeated until the maximum number of PSO iterations is reached. Finally, the global best solution obtained at the end of the optimization process is returned as the optimized six-dimensional PID parameter vector.

The proposed optimization framework therefore establishes an iterative feedback mechanism between global exploration and local exploitation. PSO performs the primary global search, while the stagnation detection mechanism determines when Nesterov accelerated local refinement should be activated. The locally improved solution is subsequently fed back into the PSO population, allowing the refined information to contribute to subsequent global search iterations.

6. Simulation Setup and Performance Evaluation

6.1. Simulation Setup

The simulation experiments are conducted to evaluate the performance of the proposed hybrid PSO–Nesterov optimization algorithm for PID controller tuning using separate pitch and yaw transfer-function models. The simulation sampling time is set to $T_s = 0.01$ s, and the total simulation horizon is 20 s. The pitch and yaw controllers track their respective reference angles, and the six PID parameters are optimized simultaneously within predefined parameter bounds.

For the PSO component, a swarm size of 25 particles and a maximum of 100 iterations are employed. The cognitive and social acceleration coefficients are set to $c_1 = c_2 = 1.9$, while the inertia weight is linearly decreased from $w_{max} = 0.90$ to $w_{min} = 0.20$ during the optimization process. Particle positions are constrained by the predefined parameter bounds.

The Nesterov-accelerated local search is configured with a momentum coefficient of $\mu = 0.85$, a learning rate of $\eta = 0.01$, and a central finite-difference step size of $\delta = 10^{-4}$. The local search is allowed to perform up to 100 iterations per invocation, and the norm of the local search velocity is limited to 0.5. The stagnation detection mechanism uses an improvement threshold of 10^{-6} and a stagnation limit of eight consecutive iterations. When the stagnation condition is met, the current global best solution initializes the Nesterov local search. The refined solution is incorporated into the PSO population only if it improves the current global best objective function value.

The optimization vector is defined as

$$\mathrm{K} = \left[K_{p,p}, K_{i,p}, K_{d,p}, K_{p,y}, K_{i,y} K_{d,y}\right]^T \tag{45}$$

The six PID controller parameters are optimized simultaneously within the predefined parameter bounds described in Section 5. Both pitch and yaw reference angles are set to 0.5 rad.

The objective function combines the integral squared error (ISE), overshoot penalties, steady-state error penalties, and control effort for both axes. The corresponding weighting coefficients are listed in Table 2. For a fair comparison, the final comparative experiments must use the same objective function and a matched objective-function evaluation budget across all optimization methods. The simulation and optimization settings are summarized in Table 2 and The objective function weighting coefficients are provided in Table 3.

Table 2. Simulation and optimization parameters.

Parameter	Value
Sampling time	0.01 s
Simulation time	30 s
Pitch reference angle	0.5 rad
Pitch reference angle	0.5 rad
PSO population size	25
Maximum PSO iterations	100
Cognitive coefficient	1.9
Social coefficient	1.9
Initial inertia weight	0.9
Final inertia weight	0.2
Nesterov momentum coefficient	0.85
Nesterov learning rate	0.01
Finite-difference step size	10^{-4}
Maximum local-search iterations per invocation	100
Local-search velocity norm limit	0.5
Stagnation threshold	10^{-6}
Stagnation limit	8
ISE weights	1, 1
Overshoot weights	50, 50
Steady-state error weights	100, 100
Control-effort weights	0.001, 0.001
Optimization dimension	6

Table 3. Objective function weighting coefficients.

Criterion	Pitch	Yaw
ISE	1	1
Overshoot	50	50
Steady-state error	100	100
Control effort	0.001	0.001

6.2. Comparative Optimization Methods

The effectiveness of the proposed hybrid PSO–Nesterov optimization algorithm is evaluated by comparing its performance with selected benchmark optimization methods. All methods use the same TRMS model, objective function, PID parameter bounds, sampling time, and simulation duration. To ensure a fair comparison, the objective-function evaluation budget must also be matched across the methods.

Standard particle swarm optimization (PSO) is selected as the primary baseline because the proposed method extends the PSO framework with a stagnation-triggered Nesterov-accelerated local search. The comparison with standard PSO is intended to assess the contribution of the local refinement mechanism to the optimization performance.

In addition, PSO combined with gradient descent (PSO–GD) is included as a global–local optimization benchmark. This comparison enables the performance of the Nesterov-based local refinement strategy to be assessed against a conventional gradient-based local search within a PSO framework. All methods optimize the same six-dimensional PID parameter vector corresponding to the pitch and yaw controllers.

The resulting solutions are evaluated using the objective-function value and the TRMS control-performance criteria defined in the following subsection. Comparative conclusions are drawn only after the methods have been evaluated using a consistent objective-function evaluation budget and the same simulation settings.

6.3. Performance Evaluation Criteria

The performance of the investigated optimization methods is evaluated using both optimization-based and control-oriented performance metrics. The objective function value (J) is used as the primary optimization criterion. In addition, the resulting TRMS responses are assessed in terms of overshoot, settling time, root mean square error (RMSE), and control effort. These metrics provide complementary information on tracking accuracy, transient response, and control input requirements. Lower values indicate better performance for each of these metrics, provided that the metrics are calculated consistently under the same reference signals and simulation conditions.

7. Results and Discussion

7.1. Optimization Convergence Analysis

The convergence characteristics of the standard PSO, standalone Nesterov, PSO–Gradient Descent (PSO–GD), and proposed hybrid PSO–Nesterov algorithms are evaluated based on the evolution of the objective function during optimization. For a fair comparison, all methods must use the same objective function, PID parameter bounds, TRMS model, reference signals, and simulation conditions. The total number of objective-function evaluations must also be controlled to ensure that the performance differences are not caused by unequal computational budgets.

Standard PSO provides a population-based global search mechanism in which the particle positions are iteratively updated to improve the objective function. The standalone Nesterov method uses gradient-based optimization to refine candidate PID parameters, whereas PSO–GD combines population-based exploration with gradient-descent-based local refinement. The proposed hybrid PSO–Nesterov algorithm integrates PSO with a Nesterov-accelerated local search that is triggered when the global best objective function value fails to improve sufficiently for a predefined number of consecutive iterations. Figure 2 presents the objective-function convergence histories of the investigated optimization methods. The convergence curves are used to examine the rate of objective-function reduction and the search behavior of each method. For the proposed hybrid algorithm, the iterations at which the Nesterov local search is activated are also identified to illustrate the interaction between the global and local search stages.

The final objective-function values are summarized in Table 4. In addition to the best value obtained by each method, the comparison should account for the total number of objective-function evaluations. The final convergence results and the corresponding relative improvement of the proposed method over standard PSO will be reported after the comparative experiments have been repeated using a consistent evaluation budget.

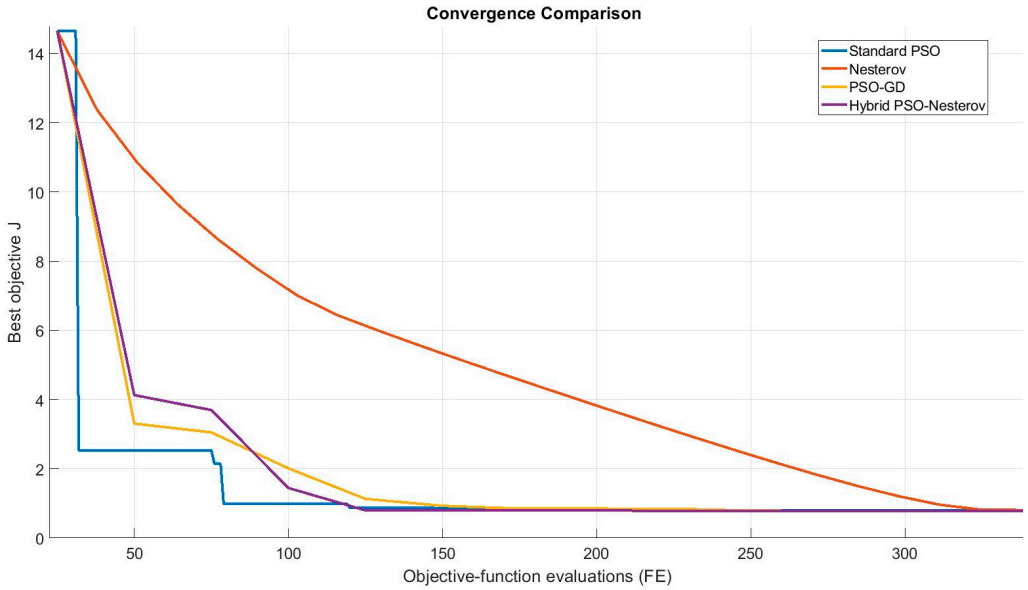


Figure 2. Convergence Comparison.

Table 4. Convergence Comparison (Best Objective Function).

Method	Best Objective Value, J	Objective-Function Evaluations
Standard PSO	0.78195	2521
Nesterov	0.74148	2521
PSO-GD	0.76041	2521
Hybrid PSO-Nesterov	0.76046	2521

To ensure a fair comparison, all four optimization methods were evaluated using the same objective-function evaluation budget of 2521 evaluations per run.

7.2. Pitch-Yaw Angles Tracking Comparison

The pitch and yaw angle tracking performances of the standard PSO, standalone Nesterov, PSO–Gradient Descent (PSO–GD), and proposed hybrid PSO–Nesterov methods are evaluated using the same reference signals and simulation conditions. The tracking responses are analyzed in terms of root mean square error (RMSE), percentage overshoot (%OS), and settling time. These indicators provide a quantitative assessment of tracking accuracy and transient response.

Figures 3 and 4 present the pitch and yaw angle responses, respectively, for the considered optimization methods. The corresponding performance indicators are summarized in Table 5, while the optimized PID controller parameters for the pitch and yaw axes are reported in Table 6. For a fair comparison, the results must be obtained using the same plant models, objective function, simulation horizon, and PID parameter bounds. The objective-function evaluation budget must also be controlled across the optimization methods.

The tracking performance of each method is assessed separately for the pitch and yaw axes. RMSE quantifies the deviation between the reference and actual angle responses, percentage overshoot characterizes the maximum excursion beyond the reference, and settling time measures the time required for the response to enter and remain within the specified tolerance band. The tolerance used to calculate settling time and the precise definition of percentage overshoot are provided to ensure reproducibility.

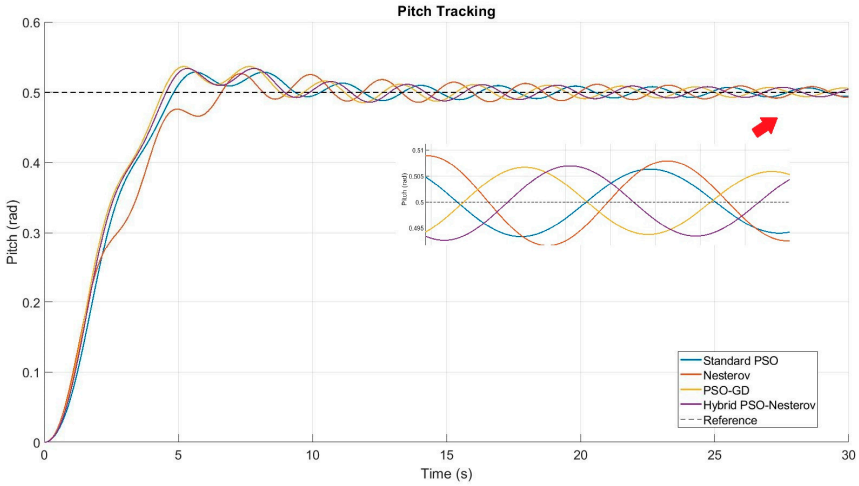


Figure 3. Pitch Angle Tracking.

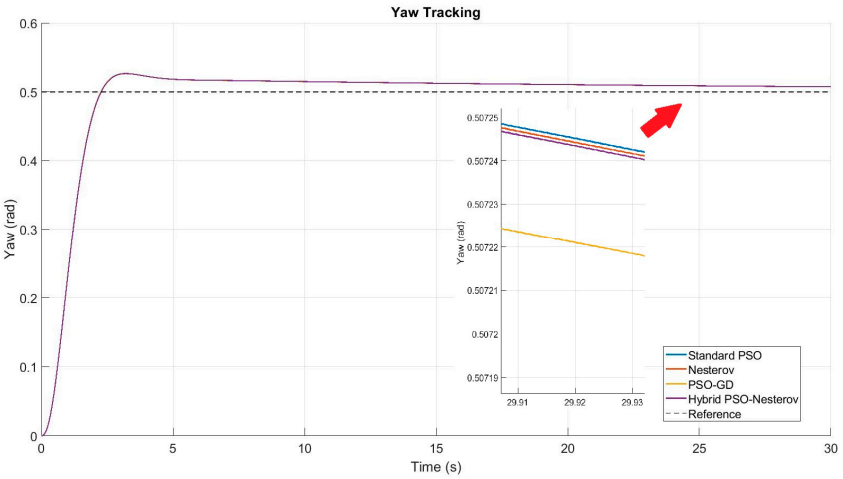


Figure 4. Yaw Angle Tracking.

Table 5. Performance Table (RMSE, %OS, Settling Time).

	RMSE	Overshoot (%)	Settling Time
Pitch (PSO)	0.11942	5.07	13.23
Yaw (PSO)	0.082926	5.2814	20.93
Pitch (Nesterov)	0.11129	6.2244	20.37
Yaw (Nesterov)	0.082924	5.2826	20.93
Pitch (PSO-GD)	0.11169	7.0795	19.02
Yaw (PSO-GD)	0.082926	5.2815	20.93
Pitch (PSO-Nesterov)	0.11128	7.2322	18.95
Yaw (PSO-Nesterov)	0.082934	5.2784	20.95

Table 6. Kp,Ki and Kd values.

Method	PSO	NESTEROV	PSO-GD	PSO-NESTEROV
Kp,p	0.2537	0.9039	0.27553	0.29478
Ki,p	1.4023	1.4405	1.6423	1.6581
Kd,p	0.19986	1.5373	0.6241	0.65747
Kp,y	2.9091	2.9095	2.9091	2.9082
Ki,y	0.1	0.1	0.1	0.1
Kd,y	2	2	2	2

The comparative results are used to assess whether the Nesterov-based local refinement improves the performance of PSO. Any conclusions regarding the relative advantages of the optimization methods are drawn from the results obtained using the final simulation model and a consistent objective-function evaluation budget. The RMSE comparison for the optimization methods is presented in Figure 5, while the overshoot in the pitch and yaw angles for each optimization method is presented in Figure 6 and settling time of pitch and yaw angles presented in Figure 7.

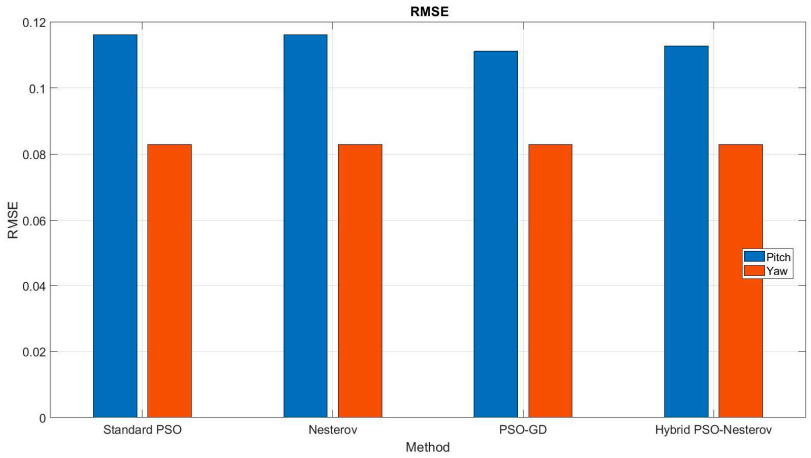


Figure 5. RMSE Comparison for Optimization Method.

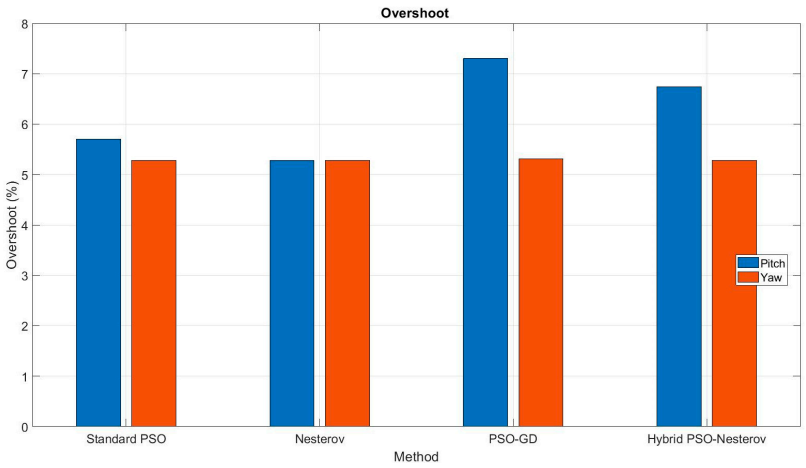


Figure 6. Overshoot in Pitch and Yaw Angles for the Optimization Method.

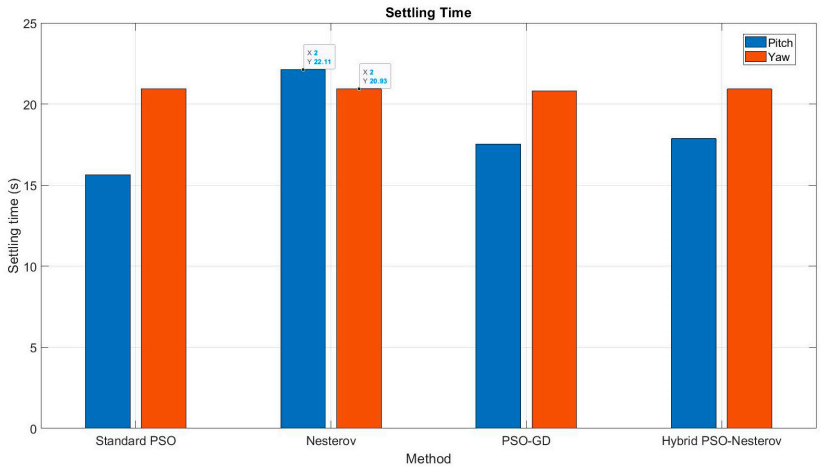


Figure 7. Settling Time of Pitch and Yaw Angles for the Optimization Method.

8. Conclusions

This study investigated a hybrid PSO–Nesterov optimization approach for PID controller tuning in a Twin Rotor MIMO System (TRMS). The proposed method combines the population-based global search capability of Particle Swarm Optimization (PSO) with Nesterov-accelerated local refinement. The local search is activated when the global best objective-function value fails to improve sufficiently over a predefined number of consecutive iterations. A candidate solution obtained through local refinement is retained only when it improves the current global best, thereby providing a selective coordination mechanism between global exploration and local exploitation.

The effectiveness of the proposed approach is evaluated by comparing its optimization performance with that of standard PSO, standalone Nesterov optimization, and PSO–Gradient Descent (PSO–GD). The comparison considers the final objective-function value and the pitch and yaw tracking performances, assessed using RMSE, overshoot, and settling time. To ensure a meaningful comparison, the methods are evaluated using consistent plant models, PID parameter bounds, reference signals, simulation conditions, and objective-function evaluation budgets. The final numerical results are reported after the comparative simulations have been repeated under these consistent conditions.

The study also highlights the importance of evaluating optimization methods using both objective-function values and axis-specific tracking indicators. A lower objective-function value alone does not necessarily imply better performance for every transient-response metric. Therefore, the relative advantages of the proposed hybrid method are assessed by considering tracking accuracy and transient behavior in both the pitch and yaw axes.

The contribution of this work lies in the stagnation-triggered coordination of PSO and Nesterov-based local refinement, together with the conditional acceptance and reinsertion of improved solutions. This mechanism is intended to refine promising PID parameter candidates when the global search stagnates, while preserving the population-based exploration process. The comparative experiments are used to determine whether this strategy provides a measurable advantage over the selected baseline methods under an equal computational budget.

A limitation of the present study is that the pitch and yaw axes are represented by separate transfer-function models. Consequently, the simulations do not establish the performance of the proposed method on a fully coupled nonlinear TRMS model. Future work will investigate the method using a coupled nonlinear model, assess its robustness to model uncertainties and disturbances, and examine its performance under more demanding operating conditions. Further studies may also investigate the sensitivity of the optimization results to algorithmic parameters and objective-function weighting coefficients.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available from the corresponding author upon reasonable request.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Ramalakshmi, P.S.; Manoharan, P.S. Non-linear modeling and PID control of twin rotor MIMO system. In Proceedings of the 2012 IEEE International Conference on Advanced Communication Control and Computing Technologies, Ramanathapuram, India, 23–25 August 2012.
2. Pandey, S.K.; Laxmi, V. Control of Twin Rotor MIMO System using PID controller with derivative filter coefficient. In Proceedings of the 2014 IEEE Students' Conference on Electrical, Electronics and Computer Science, Bhopal, India, 1–2 March 2014.

3. Choudhary, S.K. Optimal feedback control of twin rotor MIMO system with a prescribed degree of stability. *Int. J. Intell. Unmanned Syst.* **2016**, *4*, 226–238. [CrossRef]
4. Neeraj, P.J.; Seema, P.N. Control of Twin Rotor MIMO System Using Cross PID Control Technique. In Proceedings of the 2018 IEEE International Conference on Communication and Signal Processing, Singapore, 3–5 April 2018; pp. 744–750.
5. Joseph, S.B.; Dada, E.G.; Abidemi, A.; Oyewola, D.O.; Khammas, B.M. Metaheuristic algorithms for PID controller parameters tuning: Review, approaches and open problems. *Heliyon* **2022**, *8*, e09399. [CrossRef]
6. Doğruer, T.; Tan, N. Real Time Control of Twin Rotor MIMO System with PID and Fractional Order PID Controller. *Muğla J. Sci. Technol.* **2020**, *6*, 1–9. [CrossRef]
7. Doğruer, T.; Tan, N. Decoupling control of a twin rotor MIMO system using optimization method. In Proceedings of the 2019 11th International Conference on Electrical and Electronics Engineering (ELECO), Bursa, Turkey, 28–30 November 2019.
8. Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995; pp. 1942–1948. [CrossRef]
9. van den Bergh, F.; Engelbrecht, A.P. A Convergence Proof for the Particle Swarm Optimizer. *Fundam. Informaticae* **2010**, *105*, 341–374. [CrossRef]
10. Raß, A.; Schmitt, M.; Wanka, R. Explanation of Stagnation at Points that are not Local Optima in Particle Swarm Optimization by Potential Analysis. In Proceedings of the Companion Publication of the 2015 Annual Conference on Genetic and Evolutionary Computation, Madrid, Spain, 11–15 July 2015.
11. Lehre, P.K.; Witt, C. Finite First Hitting Time versus Stochastic Convergence in Particle Swarm Optimization. In *Advances in Metaheuristics*; Springer: New York, NY, USA, 2013.
12. Qi, X.; Ju, G.; Xu, S. Efficient solution to the stagnation problem of the particle swarm optimization algorithm for phase diversity. *Appl. Opt.* **2018**, *57*, 2747–2757. [CrossRef]
13. Nesterov, Y.E. A method of solving a convex programming problem with convergence rate O(1/k2). *Sov. Math. Dokl.* **1983**, *27*, 371–376.
14. Ghadimi, S.; Lan, G. Accelerated Gradient Methods for Nonconvex Nonlinear and Stochastic Programming. *arXiv* **2013**, arXiv:1310.3787.
15. Pelley, C.; Innocente, M.S.; Sienz, J. Combining Particle Swarm Optimizer with SQP Local Search for Constrained Optimization Problems. *arXiv* **2021**, arXiv:2101.10936.
16. Maiti, R.; Sharma, K.D.; Sarkar, G. PSO based parameter estimation and PID controller tuning for 2-DOF nonlinear twin rotor MIMO system. *Int. J. Autom. Control* **2018**, *12*, 582–609. [CrossRef]
17. Zulu, E.; Hara, R.; Kita, H. An efficient hybrid particle swarm and gradient descent method for the estimation of the hosting capacity of photovoltaics by distribution networks. *Energies* **2023**, *16*, 5207. [CrossRef]
18. Mojumder, M.R.H.; Roy, N.K. Review of meta-heuristic optimization algorithms to tune the PID controller parameters for automatic voltage regulator. *arXiv* **2024**, arXiv:2409.00538.
19. Hustig-Schultz, D.M.; Sanfelice, R.G. Uniting Nesterov and heavy ball methods for uniform global asymptotic stability of the set of minimizers. *Automatica* **2024**, *160*, 111389. [CrossRef]
20. Paul, P.K.; Jacob, J. On the modeling of twin rotor MIMO system using chirp inputs as test signals. *Asian J. Control* **2017**, *19*, 1731–1740. [CrossRef]
21. Abukan, Y.; Almalı, M.N. Control of a two-degree-of-freedom TRMS MIMO system using FOPID and FOSTSMC methods. *Gazi Univ. J. Sci. Part A Eng. Innov.* **2022**, *38*, 605–616. [CrossRef]
22. Atan, Ö. Fuzzy Variable Order Sliding Mode Control of Nonlinear MIMO System. *Balk. J. Electr. Comput. Eng.* **2004**, *14*, 125–133. [CrossRef]
23. Kannan, P.S.; Sheenu, P. PID control of twin rotor MIMO system. *Int. J. Adv. Res. Basic Eng. Sci. Technol.* **2017**, *3*, 1–7. [CrossRef]
24. Pathan, E.; Khan, M.H.; Aslam, M.K.; Asad, M.; Arshad, H.; Rabani, M.I. A multivariable twin-rotor system control design. *Eng. Technol. Appl. Sci. Res.* **2021**, *11*, 6626–6631. [CrossRef]
25. Sain, D.; Swain, S.K.; Saha, A.; Mishra, S.K.; Chakraborty, S. Real-Time Performance Analysis of FOI-PD Controller for Twin Rotor MIMO System. *IETE Tech. Rev.* **2019**, *36*, 547–567. [CrossRef]
26. Juang, J.-G.; Liu, W.-K.; Lin, R.-W. A hybrid intelligent controller for a twin rotor MIMO system and its hardware implementation. *ISA Trans.* **2011**, *50*, 609–619. [CrossRef]
27. Poli, R.; Kennedy, J.; Blackwell, T. Particle swarm optimization: An overview. *Swarm Intell.* **2007**, *1*, 33–57. [CrossRef]
28. Abualigah, L. Particle swarm optimization: Advances, applications, and experimental insights. *Comput. Mater. Contin.* **2025**, *82*, 1539–1592. [CrossRef]
29. Chauhan, D.; Suganthan, P.N. Learning strategies for particle swarm optimizer: A critical review and performance analysis. *Swarm Evol. Comput.* **2025**, *98*, 102048. [CrossRef]
30. Clerc, M. Stagnation analysis in particle swarm optimization or what happens when nothing happens. *HAL* **2006**, hal-00122031.

31. Su, W.; Boyd, S.; Candes, E.J. A differential equation for modeling Nesterov's accelerated gradient method: Theory and insights. *J. Mach. Learn. Res.* **2016**, *17*, 1–43.
32. Nilsen, M.M.; Stordal, A.S.; Lorentzen, R.J.; Raanes, P.N.; Eikrem, K.S. Accelerated ensemble optimization using momentum methods. *Sci. Rep.* **2024**, *14*, 25336. [CrossRef]
33. Yang, X.; Huang, Y. An adaptive dynamic multi-swarm particle swarm optimization with stagnation detection and spatial exclusion for solving continuous optimization problems. *Eng. Appl. Artif. Intell.* **2023**, *123*, 106215. [CrossRef]
34. Boresta, M.; Colombo, T.; De Santis, A.; Lucidi, S. A mixed finite differences scheme for gradient approximation. *J. Optim. Theory Appl.* **2022**, *194*, 1–24. [CrossRef]
35. Ramdass, P.; Ganesan, G.; Boulaaras, S.; Tantawy, S.S. Enhancing efficacy in breast cancer screening with Nesterov momentum optimization techniques. *Mathematics* **2024**, *12*, 3354. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.