

## Article

# High-Gain Observer-Based Backstepping Control for Real-Time Trajectory Tracking of a Twin Rotor MIMO System: Adaptive Tuning Functions Versus Metaheuristic Gain Optimization

Abderrahmane Kacimi <sup>1</sup>, Mohamed Mostefaoui <sup>2</sup>, Azeddine Beloufa <sup>2</sup> , Souaad Tahraoui <sup>3</sup> , Abdelbasset Azzouz <sup>4</sup> , Jun-Jiat Tiang <sup>5,\*</sup>  and Mehdi Houari Zaid <sup>6</sup> 

<sup>1</sup> Department of Instrumentation Maintenance, Institute of Maintenance and Industrial Safety, University of Oran 2, Oran 31000, Algeria

<sup>2</sup> Laboratory of Electrical Engineering and Renewable Energies (LGEER), Electronic Department, Hassiba Benbouali University, Chlef 02000, Algeria

<sup>3</sup> Signals, Systems and Artificial Intelligence Laboratory (2SAIL), Electronic Department, Hassiba Benbouali University, Chlef 02000, Algeria

<sup>4</sup> Laboratory of Electronics, Advanced Signal Processing, and Microwave (LESM), Department of Telecommunications, Saida University, Saida 20000, Algeria

<sup>5</sup> Centre for Wireless Technology, Centre of Excellence for Intelligent Network, Faculty of Artificial Intelligence and Engineering, Multimedia University, Cyberjaya 63100, Malaysia

<sup>6</sup> Laboratoire AVCIS (Automatisation, Vision et Contrôle Intelligents des Systèmes), Département d'Automatique, Faculté de Génie Électrique, University of Science and Technology of Oran (USTO-MB), Oran 31000, Algeria; mehdihouari.zaid@univ-usto.dz

\* Correspondence: jjtiang@mmu.edu.my

## Abstract

This paper addresses the real-time trajectory tracking problem for the Twin Rotor MIMO System (TRMS), a nonlinear, strongly coupled, open-loop unstable aerodynamic laboratory benchmark whose six-dimensional state space is only partially observable through pitch and yaw angle encoders. A High-Gain Observer (HGO) is designed to reconstruct the four unmeasured states, comprising angular velocities and rotor torques, from encoder measurements alone. Three observer-based backstepping control architectures are proposed and experimentally validated on the physical TRMS platform at a 1 kHz embedded sampling rate: (i) adaptive backstepping with tuning functions, which eliminates the over-parametrization inherent in conventional adaptive formulations through a single unified parameter update law; (ii) backstepping with online Brain Storm Optimization (BSO) of the design gains; and (iii) backstepping with online Artificial Bee Colony (ABC) gain optimization. All three architectures achieve stable 100 s trajectory tracking, whereas the conventional non-adaptive backstepping baseline diverges after 42 s due to progressive yaw-channel instability exceeding 4 rad. The BSO- and ABC-optimized controllers achieve the highest pitch-axis tracking precision (reducing pitch root-mean-square errors by 68% relative to the baseline), while the adaptive tuning functions architecture yields the best yaw-axis stability (0.2244 rad RMSE, a 91% reduction). The tuning functions architecture primarily resolves the yaw-channel instability caused by parametric over-parametrization, while the metaheuristic optimizers primarily improve pitch tracking precision through on-line gain refinement. Closed-loop stability is rigorously established via Lyapunov analysis and the nonlinear separation principle. The High-Gain Observer is directly validated on the two measured states through comparison of its pitch and yaw angle estimates against the incremental encoder signals over the full 100 s trial; the angular velocity and rotor torque estimates are only indirectly supported by the sustained stability of the closed loop, since no velocity or torque sensor is available on the rig. Comprehensive simulation and real-



Academic Editor: Guanghong Yang

Received: 12 June 2026

Revised: 15 July 2026

Accepted: 22 July 2026

Published: 27 July 2026

**Copyright:** © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

time experimental comparisons quantify the performance, robustness, and computational feasibility of each architecture under identical operating conditions.

**Keywords:** twin rotor MIMO system; backstepping control; high-gain observer; adaptive tuning functions; artificial bee colony; brain storm optimization; real-time trajectory tracking; metaheuristic gain optimization

## 1. Introduction

The rapid proliferation of unmanned aerial vehicles (UAVs) and autonomous rotary-wing platforms across defense, surveillance, and civilian sectors has imposed increasingly stringent requirements on flight control system design [1,2]. Such platforms are inherently characterized by strongly coupled multivariable dynamics, significant aerodynamic nonlinearities, and parametric uncertainties that vary substantially across the operational envelope, rendering precise and robust trajectory tracking a formidable control design challenge. To evaluate and validate advanced control strategies under controlled yet physically representative conditions prior to full-scale deployment, laboratory-scale benchmark systems that faithfully replicate these complex dynamic behaviors are widely employed [3]. Among the most extensively studied of these platforms is the Twin Rotor MIMO System (TRMS), a two-degrees-of-freedom rotary apparatus comprising independently actuated main and tail rotors whose aerodynamic interactions produce pronounced cross-coupling between the pitch and yaw axes [3–5]. This coupling, combined with open-loop instability and nonlinear rotor dynamics, renders the TRMS a rigorous and widely recognized testbed for multivariable flight control research [3–5].

Early investigations into TRMS control were grounded predominantly in classical and linear optimal frameworks. Linear Quadratic Gaussian controllers augmented with integrators and feedforward compensation were developed to achieve satisfactory regulation within localized linearized operating regions [6]. Optimization-based classical approaches, including Particle Swarm Optimization-tuned PID controllers, were subsequently proposed to extend the operational range of linear designs [7]. However, the fundamental reliance of such methods on linearized plant models renders their performance significantly degraded when the system undergoes wide-range nonlinear maneuvers, motivating the investigation of advanced nonlinear control architectures. In this context, Feedback Linearization combined with nonlinear state observers was explored as a means of explicitly compensating the nonlinear TRMS dynamics [8], while robust embedded control formulations were developed to enhance closed-loop resilience against modeling uncertainties [9]. Sliding mode control has also attracted considerable attention for TRMS applications owing to its inherent robustness against matched disturbances, with disturbance observer-augmented and integral sliding mode architectures [10,11], fuzzy logic and evolutionary optimization approaches [12], and metaheuristic-optimized variable-gain designs [13] all demonstrating improved robustness characteristics relative to classical baselines. Alternative data-driven approaches, including iterative learning control [14,15] and neural network-based adaptive control [16], have further broadened the landscape of TRMS control methodologies, though these approaches typically require either repetitive task execution or substantial offline training data, limiting their generalizability to non-repetitive real-time maneuvers.

Among the nonlinear control methodologies applied to the TRMS, backstepping has emerged as a particularly powerful recursive design framework. By systematically constructing a sequence of virtual control laws and associated Lyapunov functions, backstepping provides rigorous closed-loop stability guarantees for strict-feedback nonlinear

systems [17]. The seminal application of this methodology to the TRMS was established by Pratap and Purwar [4], who demonstrated real-time implementation of a neuro-adaptive observer-based backstepping controller, providing the foundational precedent for subsequent TRMS backstepping research. Building upon this foundation, hybrid intelligent architectures integrating backstepping with Interval Type-2 Fuzzy Logic observers have been developed for the TRMS [18], demonstrating enhanced disturbance rejection at the cost of increased design complexity. The broader applicability of backstepping to coupled nonlinear aerial platforms has been further corroborated through its successful deployment on tandem rotor helicopter systems [2], confirming its suitability as a foundational design methodology for rotor-based platforms exhibiting aerodynamic coupling. Nonlinear adaptive model predictive control formulations have also been investigated for TRMS trajectory tracking [19], though their computational demands present challenges for embedded real-time implementation. More recently, prescribed-time backstepping formulations have been reported for the TRMS to enforce a user-specified settling time independent of initial conditions [20], and observer-based neuro-adaptive sliding mode extensions combining feedback linearization with backstepping-like recursive design have been validated experimentally on the same platform [21]. Beyond the TRMS, high-gain-observer-based derivative approximation has been combined with backstepping for general high-order, fully actuated strict-feedback systems [22], confirming that the pairing of backstepping with high-gain observation remains an active and evolving research direction.

Despite these advances, the practical real-time deployment of backstepping controllers on the physical TRMS platform remains substantially more challenging than simulation results suggest. A fundamental limitation of conventional adaptive backstepping formulations is the over-parametrization problem, wherein the number of independent parameter update laws grows combinatorially with the system order, resulting in redundant estimation dynamics that increase computational burden and may compromise closed-loop stability [17]. To circumvent this limitation, the adaptive backstepping with tuning functions framework was developed by Krstic et al. [17], in which a single unified parameter update law is synthesized across all backstepping design steps, effectively eliminating over-parametrization while preserving the stability guarantees of the standard adaptive formulation. Subsequent adaptive backstepping works have extended this foundation to accommodate specific classes of actuator nonlinearity and unknown disturbance bounds [23]. Sodhi and Kar [24] applied an adaptive backstepping approach to the TRMS, demonstrating improved parameter adaptation capability; however, the over-parametrization issue and the challenges of real-time state estimation were not fully addressed in that work. The tuning function methodology of [17] thus presents a compelling and insufficiently exploited framework for resolving this bottleneck in TRMS real-time control.

A further critical impediment to real-time backstepping implementation on the TRMS is the state estimation problem. The physical TRMS laboratory setup provides direct measurement of only two output variables, namely the pitch angle  $\psi$  and the yaw angle  $\varphi$ , whereas the backstepping controller requires knowledge of all six system states, necessitating the deployment of a nonlinear state observer [3]. High-Gain Observers (HGOs) represent a well-established class of nonlinear observers capable of providing exponentially convergent state estimates for uniformly observable systems [25,26], and their integration with feedback control has been extensively investigated [27]. However, HGO-based implementations are susceptible to the peaking phenomenon during the transient phase and exhibit sensitivity to high-frequency measurement noise, both of which are exacerbated in the physical TRMS environment by mechanical vibrations and encoder quantization [28–30]. Prior investigations integrating fixed-gain HGOs with backstepping controllers on the TRMS have demonstrated satisfactory simulation-level performance;

however, a pronounced simulation-to-real-time gap persists [28], manifesting as degraded tracking accuracy and, in extreme cases, closed-loop instability during physical implementation. In prior work [28], a conventional backstepping controller integrated with an HGO was validated in simulation with satisfactory results, but real-time deployment on the TRMS hardware resulted in yaw-channel divergence beyond 4 rad, necessitating experiment termination after 42 s.

An alternative to adaptive parameter estimation for handling model uncertainty is the online optimization of controller design parameters. Metaheuristic algorithms, a class of population-based stochastic search methods inspired by natural phenomena, have been increasingly applied to tune controller gains in real time [31,32]. The Brain Storm Optimization (BSO) algorithm, introduced by Shi [33] and surveyed in [34], emulates human collaborative brainstorming through idea clustering, within-group refinement, and cross-group recombination. The Artificial Bee Colony (ABC) algorithm, proposed by Karaboga [35] and elaborated in [36], models the foraging behavior of honey bee colonies through employed, onlooker, and scout bee phases that balance exploitation of known good solutions with exploration of the search space. Both algorithms have been applied to offline controller tuning in various domains [31,32,37], but their deployment as online, real-time gain optimizers embedded within a 1 kHz control loop, where a full optimization iteration must complete within a 1 ms interrupt cycle, has received comparatively little attention. Recent developments continue to refine both algorithms for control applications: a global-best strategy combined with a chaotic difference step and opposition-based learning has been shown to accelerate BSO convergence on benchmark optimization problems [38], and an enhanced ABC variant has demonstrated improved trajectory-tracking controller tuning for a multi-degree-of-freedom robotic manipulator [39], underscoring the continued relevance of both algorithms to real-time control tuning problems.

To address the aforementioned limitations comprehensively, this study proposes and comparatively evaluates three observer-based backstepping control architectures for real-time trajectory tracking of the TRMS, each integrating a common High-Gain Observer with a distinct gain management strategy. The principal contributions of this work are summarized as follows:

1. Three observer-based backstepping control architectures are proposed for the TRMS, each integrating a common HGO for full-state reconstruction with a distinct gain management strategy: adaptive tuning functions, online BSO, and online ABC.
2. The adaptive backstepping controller with tuning functions is rigorously derived for both pitch and yaw subsystems using the framework of [17], eliminating over-parametrization through a single unified parameter update law per subsystem. Lyapunov stability is formally established.
3. The BSO and ABC algorithms are implemented as persistent-state embedded functions that perform one complete optimization iteration per 1 ms control cycle. Because every loop in both optimizers terminates after a compile-time-fixed number of iterations, the elementary-operation count per call is structurally bounded; a certified on-target worst-case execution time (WCET) was not, however, measured in the present study and is identified as a concrete item for future real-time validation.
4. All three architectures are validated through both full nonlinear simulation and 100 s real-time experiments on the physical TRMS platform at 1 kHz, providing a rigorous comparative assessment under identical conditions. The High-Gain Observer is additionally validated on the two measured states through direct comparison of its pitch and yaw angle estimates against the encoder signals over the full trial.

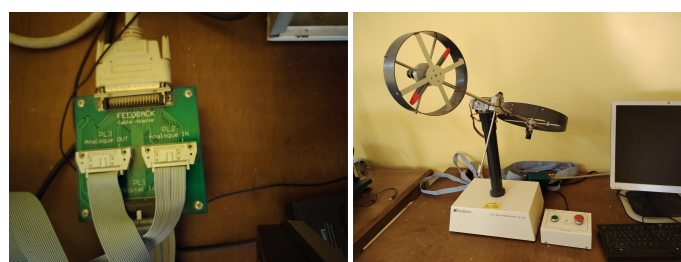
5. The distinct roles of tuning functions (resolving yaw instability) and metaheuristic optimization (improving pitch precision) are identified and quantified, providing design guidance for practitioners.

The remainder of this paper is organized as follows. Section 2 presents the TRMS mathematical model and formulates the control problem. Section 3 derives the adaptive backstepping controller with tuning functions. Section 4 designs the High-Gain Observer and establishes the observer–controller integration via the nonlinear separation principle. Section 5 develops the online BSO and ABC gain optimization algorithms. Section 6 presents simulation and real-time experimental results with a comprehensive quantitative comparison. Section 7 concludes the paper. Appendix A provides the general tuning functions formulation, and Appendix B provides the yaw-subsystem backstepping derivation and the TRMS-specific observer Jacobian in full.

## 2. TRMS Modeling and Problem Formulation

### 2.1. Plant Description

The Twin Rotor MIMO System (TRMS) is a highly coupled nonlinear aerodynamic testbed that replicates the coupled pitch-yaw aerodynamic interactions characteristic of rotary-wing aircraft, serving as a tractable laboratory analogue for helicopter and UAV flight dynamics [3,4]. As illustrated in Figure 1, which shows the physical TRMS rig used in the present study, the apparatus consists of a rigid beam pivoted on its base, affording two degrees of freedom: rotation in the vertical plane, quantified by the pitch angle  $\psi$ , driven by the main rotor; and rotation in the horizontal plane, quantified by the yaw angle  $\varphi$ , driven by the tail rotor. Consistent with the TRMS manufacturer’s notation [3],  $\psi$  denotes the pitch (elevation) angle and  $\varphi$  denotes the yaw (azimuth) angle throughout this work. The system is inherently unstable in open-loop operation and exhibits pronounced aerodynamic cross-coupling between the two axes, manifesting through gyroscopic interactions and rotor-induced reaction moments. Angular position measurements are provided by two incremental encoders with 2000 pulses per revolution, yielding a measurement resolution of approximately  $2\pi/2000 \approx 0.00314$  rad. These characteristics, combined with nonlinear aerodynamic forces and parametric uncertainties, render the TRMS a demanding and widely recognized benchmark for multivariable nonlinear control research [3,5].



**Figure 1.** The physical Twin Rotor MIMO System (TRMS) laboratory platform used in this study [3]. The main rotor (**right**) controls pitch motion; the tail rotor (**left**) controls yaw motion. Only the pitch and yaw angular positions are measured, via 2000-pulse-per-revolution incremental encoders integrated in the rig’s base.

### 2.2. Nonlinear State-Space Model

The complete physical derivation of the equations of motion, encompassing aerodynamic torques, gyroscopic effects, and friction models, follows the formulation established in [3,5]. The aerodynamic thrust moment acting on each rotor axis is approximated by a second-order polynomial function of rotor torque, a semi-empirical representation that

provides an accurate fit to experimentally measured thrust data within the operational envelope of the TRMS [3]. The state vector is defined as

$$\mathbf{x} = [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6]^T = [\psi \ \dot{\psi} \ \tau_1 \ \varphi \ \dot{\varphi} \ \tau_2]^T \quad (1)$$

where  $\tau_1$  and  $\tau_2$  denote the torques delivered by the main and tail rotors, respectively, and the control inputs  $u_1$  and  $u_2$  correspond to the supply voltages of the respective DC motors, both subject to the saturation constraint  $|u_i| \leq 2.5$  V imposed by the 12-bit digital-to-analog converter of the data acquisition card. The system constitutes a sixth-order nonlinear plant ( $n = 6$ ) with two control inputs ( $m = 2$ ) and two measured outputs ( $p = 2$ ), namely  $\mathbf{y} = [x_1, x_4]^T = [\psi, \varphi]^T$ . The four remaining states, comprising the angular velocities  $\dot{\psi}, \dot{\varphi}$  and the rotor torques  $\tau_1, \tau_2$ , are not directly accessible from the available sensors, motivating the High-Gain Observer design presented in Section 4.

The coupled nonlinear dynamics are governed by

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{1}{I_1} \left( a_1 x_3^2 + b_1 x_3 - M_g \sin(x_1) - B_{1\psi} x_2 - B_{2\psi} \text{sign}(x_2) \right. \\ \quad \left. - K_{gy} x_5 \cos(x_1) (a_1 x_3^2 + b_1 x_3) \right) \\ \dot{x}_3 = -\frac{T_{10}}{T_{11}} x_3 + \frac{K_1}{T_{11}} u_1 \\ \dot{x}_4 = x_5 \\ \dot{x}_5 = \frac{1}{I_2} \left( a_2 x_6^2 + b_2 x_6 - B_{1\varphi} x_5 - B_{2\varphi} \text{sign}(x_5) \right. \\ \quad \left. - 1.75 K_c (a_1 x_3^2 + b_1 x_3) \right) \\ \dot{x}_6 = -\frac{T_{20}}{T_{21}} x_6 + \frac{K_2}{T_{21}} u_2 \end{cases} \quad (2)$$

In Equation (2),  $I_1$  and  $I_2$  denote the moments of inertia about the pitch and yaw axes, respectively;  $a_i$  and  $b_i$  are the aerodynamic characterization parameters;  $M_g$  is the gravitational restoring moment;  $B_{1\psi}$ ,  $B_{2\psi}$ ,  $B_{1\varphi}$ , and  $B_{2\varphi}$  are the viscous and Coulomb friction coefficients; and  $T_{ij}$  and  $K_i$  are the DC motor time constants and gains, respectively. The dynamics of  $x_3$  and  $x_6$  represent the first-order electrical responses of the DC motors, where the rotor torques  $\tau_1 = x_3$  and  $\tau_2 = x_6$  evolve according to the respective motor time constants  $T_{11}$  and  $T_{21}$ , driven by the voltage control inputs  $u_1$  and  $u_2$ . The terms scaled by  $K_{gy}$  and  $K_c$  represent the gyroscopic moment and cross-reaction aerodynamic coupling between the pitch and yaw axes, respectively, and constitute the primary source of cross-channel interaction in the TRMS dynamics. The numerical values of all model parameters, as specified in the manufacturer's manual [3], are listed in Table 1.

**Table 1.** Physical parameters of the TRMS platform [3].

| Symbol | Description                  | Value                | Unit              |
|--------|------------------------------|----------------------|-------------------|
| $I_1$  | Pitch moment of inertia      | $6.8 \times 10^{-2}$ | kg m <sup>2</sup> |
| $I_2$  | Yaw moment of inertia        | $2.0 \times 10^{-2}$ | kg m <sup>2</sup> |
| $a_1$  | Main rotor aero. coefficient | 0.0135               | —                 |
| $b_1$  | Main rotor aero. coefficient | 0.0924               | —                 |
| $a_2$  | Tail rotor aero. coefficient | 0.02                 | —                 |
| $b_2$  | Tail rotor aero. coefficient | 0.09                 | —                 |
| $M_g$  | Gravitational moment         | 0.32                 | N m               |

Table 1. Cont.

| Symbol         | Description                   | Value                | Unit      |
|----------------|-------------------------------|----------------------|-----------|
| $B_{1\psi}$    | Pitch viscous friction        | $6.0 \times 10^{-3}$ | N m s/rad |
| $B_{2\psi}$    | Pitch Coulomb friction        | $1.0 \times 10^{-3}$ | N m       |
| $B_{1\varphi}$ | Yaw viscous friction          | $1.0 \times 10^{-1}$ | N m s/rad |
| $B_{2\varphi}$ | Yaw Coulomb friction          | $1.0 \times 10^{-2}$ | N m       |
| $K_{gy}$       | Gyroscopic coupling           | 0.05                 | s/rad     |
| $K_c$          | Momentum coupling             | −0.2                 | —         |
| $K_1$          | Main motor gain               | 1.1                  | —         |
| $K_2$          | Tail motor gain               | 0.8                  | —         |
| $T_{10}$       | Main motor time const. (num.) | 1.0                  | s         |
| $T_{11}$       | Main motor time const. (den.) | 1.1                  | s         |
| $T_{20}$       | Tail motor time const. (num.) | 1.0                  | s         |
| $T_{21}$       | Tail motor time const. (den.) | 1.0                  | s         |

### 2.3. Decentralized Control Structure

The measurable output vector is

$$\mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} x_1 \\ x_4 \end{bmatrix} = \begin{bmatrix} \psi \\ \varphi \end{bmatrix}. \quad (3)$$

A decentralized control strategy is adopted: the pitch subsystem  $(x_1, x_2, x_3)$  and the yaw subsystem  $(x_4, x_5, x_6)$  are treated as two independent third-order strict-feedback systems, with the cross-coupling terms treated as bounded perturbations. This decoupling is justified by the physical structure of the TRMS, where each rotor primarily drives its corresponding axis, and enables independent controller and observer design for each channel. The cross-coupling effects are subsequently handled by the robustness margins of the adaptive and optimized controllers. In the backstepping framework, the tracking strategy encompasses the regulation strategy as a special case, corresponding to a desired reference identically set to zero; the general tracking strategy is developed throughout this work.

### 2.4. Problem Statement

Given the coupled nonlinear system (2) with only the output measurements (3), design a control system  $\mathbf{u} = [u_1, u_2]^T$  such that:

1. The pitch angle  $x_1(t)$  tracks a desired reference trajectory  $x_{1d}(t)$ , and the yaw angle  $x_4(t)$  tracks  $x_{4d}(t)$ , with bounded tracking errors.
2. The four unmeasured states  $(x_2, x_3, x_5, x_6)$  are reconstructed from  $(y_1, y_2)$  with bounded estimation errors.
3. All closed-loop signals remain bounded, and the control inputs satisfy  $|u_i(t)| \leq 2.5$  V.
4. The controller operates in real time on the physical TRMS hardware at a sampling rate of  $f_s = 1$  kHz.

It is further noted that the state-space model (2) satisfies the uniform observability condition with respect to the measured output vector  $\mathbf{y} = [\psi, \varphi]^T$  [26], a property that constitutes a fundamental prerequisite for the High-Gain Observer design developed in Section 4.

## 3. Adaptive Backstepping Control with Tuning Functions

Standard adaptive backstepping formulations employ independent parameter update laws at each design step, leading to the over-parametrization problem wherein the number of estimated parameters grows combinatorially with system order, increasing computational burden and potentially compromising closed-loop stability. To circumvent this fundamental limitation, the adaptive backstepping with tuning functions framework

of [17] is adopted, in which a single unified parameter update law is synthesized across all design steps, effectively eliminating over-parametrization while preserving the stability guarantees of the standard adaptive formulation. The general theoretical formulation of this methodology is presented in Appendix A for completeness and self-containedness.

### 3.1. Nominal Backstepping Control (Baseline)

Prior to presenting the proposed adaptive architecture, a conventional decentralized backstepping controller is introduced as a performance baseline. The controller is constructed through a recursive Lyapunov-based design procedure in which the TRMS dynamics are decomposed into two independent third-order subsystems corresponding to the pitch and yaw axes, with the aerodynamic cross-coupling terms treated as bounded exogenous disturbances. The theoretical foundations and closed-loop stability analysis of this baseline formulation are thoroughly established in the authors' prior work [28], and are therefore not reproduced here in the interest of conciseness. Although this conventional controller provides mathematically rigorous stability guarantees under nominal operating conditions, its susceptibility to real-time performance degradation under physical model uncertainties and unmodeled dynamics constitutes the principal motivation for the adaptive extension developed in the subsequent subsections.

### 3.2. Pitch Subsystem Controller Design

The pitch subsystem comprises the state triplet  $(x_1, x_2, x_3)$  representing the pitch angle, pitch angular velocity, and main rotor torque, respectively, driven by the control input  $u_1$ . Following elimination of the cross-coupling terms from Equation (2), the decentralized pitch dynamics are expressed as

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \frac{1}{I_1} (a_1 x_3^2 + b_1 x_3 - M_g \sin(x_1) - B_{1\psi} x_2 - B_{2\psi} \text{sign}(x_2)) \\ \dot{x}_3 = -\frac{T_{10}}{T_{11}} x_3 + \frac{K_1}{T_{11}} u_1 \end{cases} \quad (4)$$

The aerodynamic force moment  $M_1$  is approximated by a first-order linear interpolation:

$$M_1 = a_1 x_3^2 + b_1 x_3 \approx q_{11} x_3 + q_{10}. \quad (5)$$

Accordingly, the normalized aerodynamic function is defined as

$$E(x_3) = \frac{1}{I_1} (a_1 x_3^2 + b_1 x_3) \approx \frac{1}{I_1} (q_{11} x_3 + q_{10}). \quad (6)$$

The pitch subsystem is thereby cast in the parametric strict-feedback form of Appendix A as

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = E(x_3) + \Phi_2^T(x_1, x_2) \theta \\ \dot{x}_3 = b u_1 + \psi_3(x_3) \end{cases} \quad (7)$$

where the regression vector and known functions are identified as

$$\Phi_1(x_1) = \Phi_3(x) = 0, \quad \Phi_2^T(x_1, x_2) = \begin{bmatrix} -\sin(x_1) & -x_2 & -\text{sign}(x_2) \end{bmatrix} \quad (8)$$

$$\psi_1(x_1) = \psi_2(x_1, x_2) = 0, \quad \psi_3(x_3) = -\frac{T_{10}}{T_{11}} x_3 \quad (9)$$

$$\theta = \begin{bmatrix} \frac{M_g}{I_1} & \frac{B_{1\psi}}{I_1} & \frac{B_{2\psi}}{I_1} \end{bmatrix}^T, \quad b = \frac{K_1}{T_{11}} > 0 \quad (\text{known sign}). \quad (10)$$

The three-step adaptive backstepping design procedure is elaborated as follows.

### Step 1

The first tracking error is defined as  $z_1 = x_1 - x_{1d}$ , whose time derivative is  $\dot{z}_1 = \dot{x}_1 - \dot{x}_{1d}$ . The state  $x_2$  is treated as a virtual control input, and the second error variable is defined as  $z_2 = x_2 - \alpha_1 \implies \dot{x}_2 = \dot{z}_2 + \dot{\alpha}_1$ . With the candidate Lyapunov function  $V_1 = \frac{1}{2}z_1^2$ , one obtains  $\dot{V}_1 = z_1(\dot{z}_2 + \dot{\alpha}_1 - \dot{x}_{1d})$ . The first stabilizing function is chosen as

$$\alpha_1 = -c_1 z_1 + \dot{x}_{1d}, \quad c_1 > 0, \quad (11)$$

yielding  $\dot{V}_1 = -c_1 z_1^2 + z_1 z_2$ . If  $z_2 = 0$ , then  $\dot{V}_1 < 0$ , guaranteeing asymptotic convergence of  $z_1$ .

### Step 2

Differentiating  $z_2$  gives  $\dot{z}_2 = E(x_3) + \Phi_2^T \theta + c_1 \dot{x}_2 - c_1 \dot{x}_{1d} - \ddot{x}_{1d}$ . Defining  $z_3 = E(x_3) - \alpha_2$ , the first tuning function is introduced as  $\tau_2 = \Phi_2 z_2$ . With the augmented Lyapunov function  $V_2 = V_1 + \frac{1}{2}z_2^2 + \frac{1}{2}\tilde{\theta}^T \Gamma_1^{-1} \tilde{\theta}$ , where  $\tilde{\theta} = \theta - \hat{\theta}$  and  $\Gamma_1 \in \mathbb{R}^{3 \times 3}$  is positive definite, the parameter update law is selected as

$$\dot{\hat{\theta}} = \Gamma_1 \tau_2 = \Gamma_1 \Phi_2 z_2, \quad (12)$$

and the second stabilizing function as

$$\alpha_2 = -c_2 z_2 - z_1 - \Phi_2^T \hat{\theta} - c_1 \dot{x}_2 + c_1 \dot{x}_{1d} + \ddot{x}_{1d}, \quad c_2 > 0, \quad (13)$$

which gives  $\dot{V}_2 = -c_1 z_1^2 - c_2 z_2^2 + z_2 z_3$ .

### Step 3

Differentiating  $z_3 = E(x_3) - \alpha_2$  and substituting the motor dynamics yields  $\dot{z}_3 = b_a u_1 + \alpha_3 + \ddot{x}_{1d}$ , where  $b_a = q_{11} K_1 / (I_1 T_{11}) = q_{11} b / I_1 > 0$  (known sign). The control input and adaptive update laws are

$$\begin{cases} u_1 = \hat{p} \bar{u}_1 \\ \bar{u}_1 = \alpha_3 + \ddot{x}_{1d} \\ \dot{\hat{\theta}} = \Gamma_1 \Phi_2 z_2 \\ \dot{\hat{p}} = -\gamma_1 \text{sign}(b_a) \bar{u}_1 z_3 \end{cases} \quad (14)$$

where  $\gamma_1 > 0$  and  $\hat{p}$  estimates  $p = 1/b_a$ . The explicit pitch control law takes the form

$$\begin{aligned} u_1 = \hat{p} \left( -z_2 - c_3 z_3 + \frac{q_{11} T_{10}}{I_1 T_{11}} x_3 + (-c_1 + c_1 c_2) x_2 \right. \\ \left. + \hat{\theta}_1^T (-2c_1 - c_2) \Phi_2 + (-c_1 - c_2) E + (c_1 + c_2) \dot{x}_{1d} \right. \\ \left. + \Phi_2^T \Gamma_1 \tau_2 + c_1 \dot{x}_{1d} + \ddot{x}_{1d} \right). \end{aligned} \quad (15)$$

The total Lyapunov function  $V_3 = V_2 + \frac{1}{2}z_3^2 + \frac{|b_a|}{2\gamma_1}\tilde{p}^2$  satisfies

$$\dot{V}_3 = -c_1z_1^2 - c_2z_2^2 - c_3z_3^2 = -\sum_{i=1}^3 c_i z_i^2 < 0. \quad (16)$$

By LaSalle's Invariance Theorem [40],  $z_1, z_2, z_3, \hat{\theta}, \hat{p}$  are bounded and  $z_1, z_2, z_3 \rightarrow 0$  asymptotically. Consequently  $u_1(t)$  is bounded and asymptotic pitch trajectory tracking is ensured [17].

### 3.3. Yaw Subsystem Controller Design

The yaw subsystem, comprising the state triplet  $(x_4, x_5, x_6)$  and driven by the control input  $u_2$ , is designed through an adaptive backstepping procedure that mirrors, step for step, the pitch-subsystem derivation of Section 3.2, under the substitutions  $x_1 \rightarrow x_4, x_2 \rightarrow x_5, x_3 \rightarrow x_6, c_1 \rightarrow c_4, c_2 \rightarrow c_5, c_3 \rightarrow c_6$ , the yaw-specific regression vector  $\Phi_2^T(x_4, x_5) = [-x_5 \ -\text{sign}(x_5)]$ , and the adaptation gain matrix  $\Gamma_2 \in \mathbb{R}^{2 \times 2}$  in place of  $\Gamma_1$ . The full three-step derivation is retained for reproducibility and is provided in Appendix B.1. The resulting yaw control law and adaptive update laws are given by Equations (A7) and (A8), and the corresponding total Lyapunov function satisfies  $\dot{V}_3 = -c_4z_1^2 - c_5z_2^2 - c_6z_3^2 < 0$  (with  $z_1, z_2, z_3$  now denoting the yaw-subsystem error variables), establishing boundedness of all closed-loop signals and asymptotic yaw trajectory tracking via LaSalle's Invariance Theorem [40], by the identical argument used for the pitch subsystem.

**Remark 1.** In the decentralized structure, the pitch and yaw controllers are designed independently with separate parameter vectors  $\hat{\theta}_1$  and  $\hat{\theta}_2$  and separate design gain triplets  $(c_1, c_2, c_3)$  for the pitch axis and  $(c_4, c_5, c_6)$  for the yaw axis. The combined closed-loop stability follows from the cascaded Lyapunov argument  $V = V_\psi + V_\varphi$  under the assumption that the cross-coupling perturbations are bounded, which is guaranteed by the physical constraints of the TRMS operating envelope.

## 4. Observer Design

The only measurable states of the TRMS laboratory setup are the pitch angle ( $\psi$ ) and yaw angle ( $\varphi$ ). The adaptive backstepping controller, however, requires knowledge of all six states. A nonlinear observer is therefore required to estimate the unmeasured states from the available outputs. The proposed full-state observer is a high-gain observer as in [26], of the form

$$\dot{\hat{\mathbf{x}}} = \mathbf{f}(\hat{\mathbf{x}}) + \mathbf{B}\mathbf{u} + \left( \frac{\partial \phi(\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}} \right)^{-1} \Lambda^{-1}(\theta, \delta) L (\mathbf{y} - \mathbf{C}\hat{\mathbf{x}}). \quad (17)$$

For a MIMO nonlinear system described by the state-space representation

$$\begin{cases} \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{B}\mathbf{u} \\ \mathbf{y} = \mathbf{h}(\mathbf{x}) \end{cases} \quad (18)$$

with

$$\phi(\hat{\mathbf{x}}) = \left[ L_f^0 h_1(\hat{\mathbf{x}}) \ \dots \ L_f^{\mu_1-1} h_1(\hat{\mathbf{x}}) \ \middle| \ \dots \ \middle| \ L_f^0 h_p(\hat{\mathbf{x}}) \ \dots \ L_f^{\mu_p-1} h_p(\hat{\mathbf{x}}) \right]^T, \quad (19)$$

$$\mu_1 + \mu_2 + \dots + \mu_p = n,$$

and for  $k = 1, \dots, p$ ,

$$\Lambda^{-1}(\theta, \delta) = \text{diag}(\theta^{\delta_k} \Delta_k(\theta^{\delta_k})), \quad \Delta_k(\theta) = \text{diag}(1, \theta, \dots, \theta^{\mu_k-1}), \quad (20)$$

with  $A = \text{diag}(A_k)$ ,  $A_k \in \mathbb{R}^{\mu_k \times \mu_k}$ ,  $C = \text{diag}(C_k)$ ,  $C_k = [1 \ 0 \ \dots \ 0] \in \mathbb{R}^{1 \times \mu_k}$ , and  $L \in \mathbb{R}^{n \times p}$  chosen such that the eigenvalues of  $(A - LC)$  lie in the open left half-plane. The quantity  $1/\theta^{\delta_k}$  is the high-gain parameter, acting as a time scaling for the observer dynamics.

Applied to the TRMS,  $n = 6$ ,  $p = 2$ , and  $\mu_1 = \mu_2 = 3$ . The resulting Jacobian  $\partial\phi(\hat{x})/\partial\hat{x}$ , its nonzero partial-derivative entries, and the explicit form of  $\Lambda^{-1}(\theta, \delta)$  for the TRMS are lengthy expressions that add little to the conceptual development of the observer; for conciseness, they are relegated to Appendix B.2 and are used directly in the implementation of Equation (17). For the real-time implementation, the two high-gain parameters were set to  $\theta_1 = 2$  for the pitch channel and  $\theta_2 = 1.5$  for the yaw channel, values selected empirically to balance fast observer convergence against amplification of encoder measurement noise.

### Observer–Controller Integration

Consider the adaptive backstepping with tuning functions control laws given by Equations (15) and (A7), evaluated on the observed state  $\hat{x}$  from the observer, Equation (17), instead of on the true state  $x$ ; the certainty-equivalence pitch and yaw laws then take the form

$$u_1 = \hat{p} \left( -z_2 - c_3 z_3 + \frac{q_{11} T_{10}}{I_1 T_{11}} \hat{x}_3 + (-c_1 + c_1 c_2) \hat{x}_2 + \hat{\theta}_1^\top (-2c_1 - c_2) \Phi_2 + (-c_1 - c_2) E + (c_1 + c_2) \dot{x}_{1d} + \Phi_2^\top \Gamma_1 \tau_2 + c_1 \ddot{x}_{1d} + \ddot{x}_{1d} \right), \quad (21)$$

$$u_2 = \hat{p} \left( -z_2 - c_6 z_3 + \frac{q_{21} T_{20}}{I_2 T_{21}} \hat{x}_6 + (-c_4 + c_4 c_5) \hat{x}_5 + \hat{\theta}_2^\top (-2c_4 - c_5) \Phi_2 + (-c_4 - c_5) E + (c_4 + c_5) \dot{x}_{4d} + \Phi_2^\top \Gamma_2 \tau_2 + c_4 \ddot{x}_{4d} + \ddot{x}_{4d} \right). \quad (22)$$

The stability of the combined observer–controller architecture is established by invoking the weakened nonlinear separation principle [41,42]. The underlying separation-principle results [27,43,44] state that combining an asymptotically stabilizing state-feedback controller with an exponentially convergent high-gain observer recovers the performance of the state-feedback design on the closed-loop system, provided three hypotheses hold: exponential convergence of the observer, an asymptotically stabilizing control law, and a valid Lyapunov function. The proposed adaptive backstepping design with tuning functions satisfies these prerequisites. However, because the TRMS actuators are subject to the physical saturation limit  $|u_i| \leq 2.5 \text{ V}$ , the control law cannot be globally stabilizing; the domain of attraction is necessarily bounded. The closed-loop TRMS architecture is therefore guaranteed to be semi-globally asymptotically stable, with the region of attraction determined jointly by the actuator saturation limit and by the initial condition set used at start-up, rather than the global result that would hold for an unconstrained input.

## 5. Online Metaheuristic Backstepping Gain Optimization

### 5.1. Motivation and Common Optimization Framework

The adaptive backstepping controller with tuning functions (Section 3) addresses the over-parametrization problem through online estimation of the uncertain parameter vector  $\theta$ , but leaves the design gains ( $c_1, c_2, c_3$  for pitch;  $c_4, c_5, c_6$  for yaw) fixed. This section explores an independent design axis: rather than adapting  $\hat{\theta}$ , the nominal backstepping structure of Section 3.1 is retained (with  $\theta$  held at its fixed, calibrated value), while the design gains themselves are optimized online. Structurally, the metaheuristic-optimized controllers are therefore direct extensions of the baseline architecture rather than of the

tuning-functions architecture: the fixed gains in the nominal law of Section 3.1 are replaced by the time-varying vector  $\mathbf{c}^*(t)$  produced by the algorithms below.

Although the  $c_i$  are physically interpretable as bandwidth parameters for each subsystem, their optimal fixed values are sensitive to the prevailing aerodynamic operating point, the cross-coupling intensity, and the magnitude of unmodeled disturbances, all of which vary during real-time TRMS operation. Suboptimal fixed gains result in either sluggish tracking response or excessive control effort, both of which measurably degrade closed-loop performance on the physical platform.

To evaluate online gain optimization as an alternative to parameter adaptation, two metaheuristic algorithms are proposed as gain-tuning mechanisms for the nominal backstepping structure: the Brain Storm Optimization (BSO) algorithm [33,34] and the Artificial Bee Colony (ABC) algorithm [35,36]. Both algorithms optimize an identical decision variable space and objective function under identical real-time experimental conditions, enabling a rigorous direct comparative evaluation.

The decision variable vector to be optimized is

$$\mathbf{c} = [c_1 \quad c_2 \quad c_3]^T \quad (\text{or } [c_4, c_5, c_6]^T \text{ for yaw}) \in \mathbb{R}^3, \quad (23)$$

subject to the box constraints

$$\underbrace{\begin{bmatrix} 0.1 \\ 0.1 \\ 0.01 \end{bmatrix}}_{\text{lb}} \leq \mathbf{c} \leq \underbrace{\begin{bmatrix} 10.0 \\ 5.0 \\ 1.0 \end{bmatrix}}_{\text{ub}} \quad (24)$$

where the bounds are established from the TRMS stability region identified through systematic preliminary experimentation. The same bounds are applied independently to the pitch and yaw subsystem optimizers.

The common objective function is a regularized tracking cost:

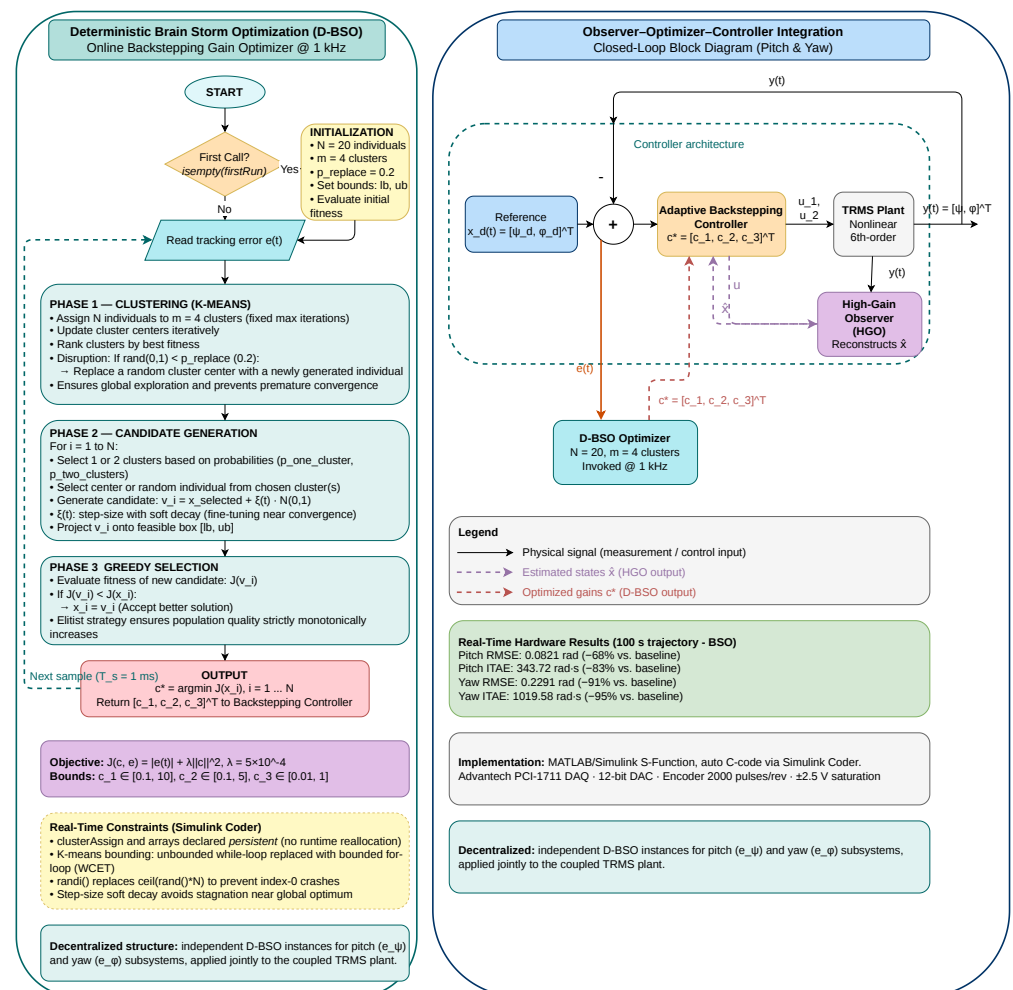
$$J(\mathbf{c}, e) = |e(t)| + \lambda \|\mathbf{c}\|^2, \quad (25)$$

where  $e(t)$  denotes the instantaneous axis tracking error and  $\lambda = 5 \times 10^{-4}$  is a regularization coefficient that penalizes large gain magnitudes, suppressing control chattering and protecting against actuator saturation without biasing the solution toward the lower bound. The use of an instantaneous tracking error rather than a windowed integral metric eliminates the memory overhead and computational latency associated with sliding-window cost evaluation, both of which are critical under the 1 ms interrupt-cycle constraint. Temporal smoothing of the gain trajectory is achieved implicitly through the elitist selection mechanism common to both algorithms, which accepts new gain vectors only when they reduce the objective value, thereby preventing abrupt gain variations between successive control cycles.

Both algorithms are executed as persistent-state embedded functions invoked at every control sample step  $T_s = 0.001$  s. The entire population is maintained in static memory across successive calls, and one complete optimization iteration is performed per invocation. The current best gain vector  $\mathbf{c}^*$  is fed directly into the nominal backstepping control law of Section 3.1 (see [28] for the complete derivation), with  $\theta$  fixed at its nominal value and the tuning-function terms of Section 3 omitted, in place of the fixed gains  $c_1$  through  $c_6$ . Independent instances of each optimizer are deployed for the pitch and yaw subsystems, each receiving its respective tracking error  $e_\psi(t) = x_1 - x_{1d}$  and  $e_\phi(t) = x_4 - x_{4d}$ .

## 5.2. Brain Storm Optimization Gain Tuning

The Brain Storm Optimization algorithm [33,34] models human collaborative problem-solving, in which participants organize their ideas into thematic clusters, refine solutions within each cluster through within-group deliberation, and recombine insights across clusters to generate qualitatively novel candidates. This dual mechanism of intra-cluster exploitation and inter-cluster exploration confers on BSO a favorable balance between local refinement and global search, particularly effective in the low-dimensional bounded continuous optimization problem defined by (25). The BSO population consists of  $N_{id} = 20$  ideas (candidate gain triplets) partitioned into  $N_c = 4$  clusters. The overall structure of the online BSO gain tuner, together with its closed-loop integration with the backstepping controller and the High-Gain Observer, is summarized in Figure 2. The complete algorithm parameters are listed in Table 2.



**Figure 2.** Flowchart of the online Brain Storm Optimization (BSO) gain tuner and its closed-loop integration with the backstepping controller and High-Gain Observer for the TRMS. The BSO optimizer delivers the optimal gain vector  $c^*$  to the respective backstepping controller once per 1 ms control cycle.

### Phase 1: Rank-Based Clustering

At each iteration, the  $N_{id}$  ideas are ranked in ascending order of objective value  $J$ . The top  $N_c$  ideas are designated as cluster centres  $\mu_c$ . Every remaining idea is assigned to its nearest centre under the squared Euclidean distance:

$$\text{cluster}(i) = \arg \min_{c \in \{1, \dots, N_c\}} \|\mathbf{x}_i - \boldsymbol{\mu}_c\|_2^2, \quad i = 1, \dots, N_{id}. \quad (26)$$

**Table 2.** BSO algorithm parameters as deployed in the embedded real-time implementation.

| Parameter             | Description                          | Value |
|-----------------------|--------------------------------------|-------|
| $N_{id}$              | Population size                      | 20    |
| $N_c$                 | Number of clusters                   | 4     |
| $p_{\text{replace}}$  | Centre replacement probability       | 0.20  |
| $p_{\text{one}}$      | Single-cluster selection probability | 0.80  |
| $p_{\text{center}}$   | Centre-based generation probability  | 0.40  |
| $p_{\text{two-c}}$    | Two-centre combination probability   | 0.50  |
| $\sigma_0$            | Initial step-size factor             | 0.10  |
| $T_{\text{decay}}$    | Decay time constant (iterations)     | 300   |
| $\sigma_{\text{min}}$ | Step-size floor                      | 0.02  |

This rank-based single-pass assignment has fixed computational cost  $\mathcal{O}(N_{id} \cdot N_c)$ , replacing the unbounded while-loop of iterative K-means with a bounded for-loop, so that the per-call operation count is bounded above by a value that does not depend on the random draws made during the call, and is compatible with  $T_s = 0.001$  s. With probability  $p_{\text{replace}} = 0.20$ , one randomly selected centre is replaced by a uniformly randomized idea, providing diversity injection against cluster stagnation.

#### Phase 2: Adaptive Step-Size Schedule

The mutation amplitude decays monotonically with the cumulative iteration count  $t$ :

$$\sigma(t) = \sigma_0 \exp\left(-\frac{t}{T_{\text{decay}}}\right) + \sigma_{\text{min}}, \quad (27)$$

where  $\sigma(t) \in (\sigma_{\text{min}}, \sigma_0 + \sigma_{\text{min}}) = (0.02, 0.12)$ , promoting broad exploration early and fine local refinement as the population converges.

#### Phase 3: Idea Generation

A new candidate is generated for each idea through a two-mode strategy parameterized by  $p_{\text{one}} = 0.80$ . In single-cluster mode (probability  $p_{\text{one}}$ ), a cluster  $c$  is selected uniformly at random and a base idea is drawn from the centre  $\boldsymbol{\mu}_c$  (probability  $p_{\text{center}} = 0.40$ ) or from a uniformly selected member, with Gaussian mutation:

$$\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{base}} + \sigma(t) \cdot \boldsymbol{\zeta} \odot (\mathbf{ub} - \mathbf{lb}), \quad \boldsymbol{\zeta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_{n_p}). \quad (28)$$

In two-cluster mode (probability  $1 - p_{\text{one}}$ ), two distinct clusters are combined at their midpoint:

$$\mathbf{x}_{\text{base}} = \frac{\mathbf{x}_{c1} + \mathbf{x}_{c2}}{2}, \quad \mathbf{x}_{\text{new}} = \mathbf{x}_{\text{base}} + \sigma(t) \cdot \boldsymbol{\zeta} \odot (\mathbf{ub} - \mathbf{lb}). \quad (29)$$

All generated candidates are projected onto the feasible box (24) prior to objective evaluation.

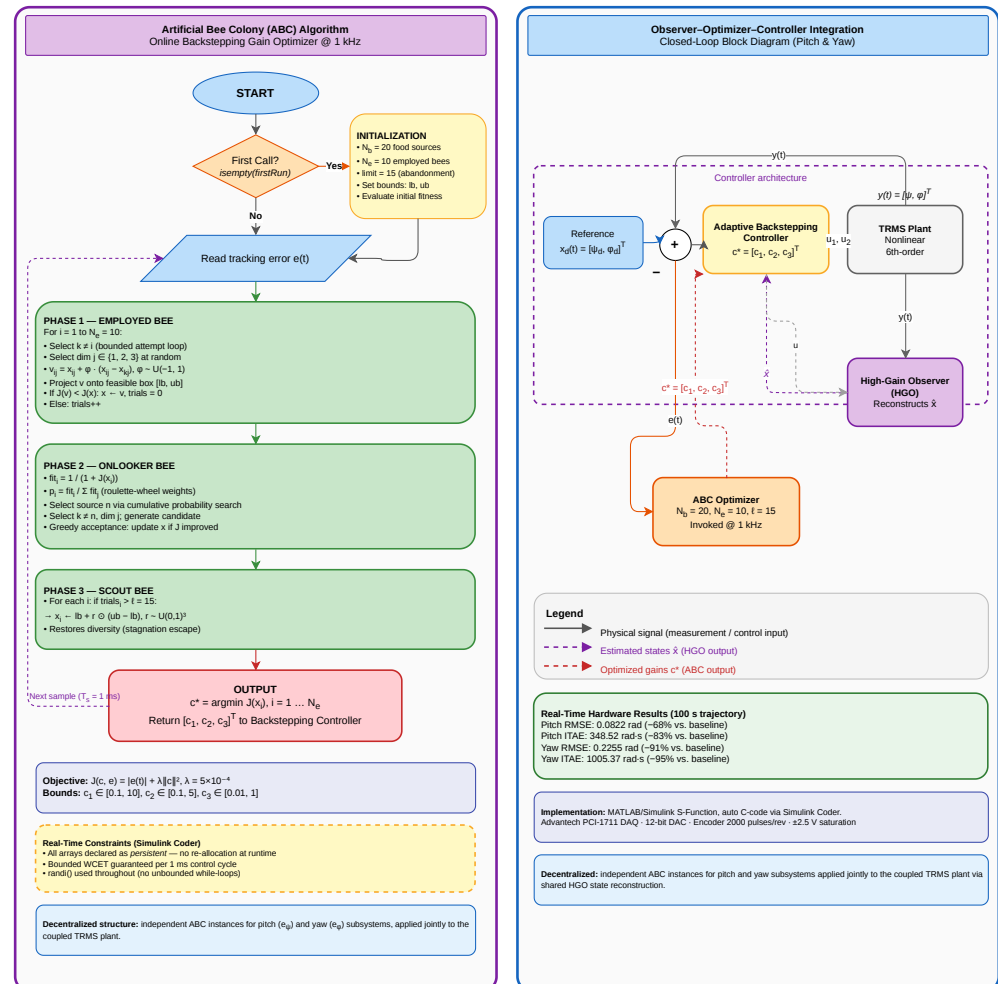
#### Phase 4: Elitist Selection

A greedy acceptance criterion is applied independently to each idea, and the best gain vector across the full population is output at each time step:

$$\mathbf{c}_{\text{BSO}}^* = \mathbf{x}_{i^*}, \quad i^* = \arg \min_i J(\mathbf{x}_i, e). \quad (30)$$

### 5.3. Artificial Bee Colony Gain Optimization

The Artificial Bee Colony algorithm [35,36] emulates the division-of-labor foraging behavior of a honey bee colony. The population of  $N_b = 20$  candidate solutions (food sources) is partitioned equally into  $N_e = 10$  employed bees and  $N_o = 10$  onlooker bees. Each food source  $\mathbf{x}_i \in \mathbb{R}^n$  represents a candidate gain triplet evaluated via (25). Figure 3 summarizes the structure of the online ABC gain optimizer and its closed-loop integration with the backstepping controller and the High-Gain Observer.



**Figure 3.** Flowchart of the online Artificial Bee Colony (ABC) gain optimizer and its closed-loop integration with the backstepping controller and High-Gain Observer. The ABC optimizer delivers the optimal gain vector  $c^*$  to the respective backstepping controller once per 1 ms control cycle.

#### Phase 1: Employed Bee Phase

Each employed bee  $i$  generates a candidate  $\mathbf{v}_i$  by perturbing its current position along a single randomly selected dimension  $j$  using a neighbor  $k \neq i$ :

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}), \quad \phi_{ij} \sim \mathcal{U}(-1, 1). \quad (31)$$

The candidate is projected onto the feasible box (24) and a greedy acceptance criterion is applied:

$$\mathbf{x}_i \leftarrow \begin{cases} \mathbf{v}_i & \text{if } J(\mathbf{v}_i, e) < J(\mathbf{x}_i, e) \\ \mathbf{x}_i & \text{otherwise} \end{cases} \quad (32)$$

with the trial counter reset to zero upon improvement and incremented otherwise.

#### Phase 2: Onlooker Bee Phase

Onlooker bees select food sources via a roulette-wheel mechanism proportional to a fitness transformation:

$$\text{fit}_i = \frac{1}{1 + J(\mathbf{x}_i, e)}, \quad p_i = \frac{\text{fit}_i}{\sum_{j=1}^{N_e} \text{fit}_j}. \quad (33)$$

The same perturbation (31) and greedy acceptance (32) are subsequently applied.

#### Phase 3: Scout Bee Phase

Any food source whose trial counter exceeds the abandonment threshold  $\ell = 15$  is replaced by a uniformly randomized new solution:

$$\mathbf{x}_i \leftarrow \mathbf{lb} + \mathbf{r} \odot (\mathbf{ub} - \mathbf{lb}), \quad \mathbf{r} \sim \mathcal{U}(0, 1)^{n_p}, \quad \text{if trials}_i > \ell. \quad (34)$$

At the 1 kHz call rate,  $\ell = 15$  corresponds to 15 ms of stagnation per food source. This intentionally aggressive abandonment schedule is appropriate for the TRMS application, where the aerodynamic operating point evolves on a timescale of seconds. The ABC algorithm parameters are summarized in Table 3.

**Table 3.** ABC algorithm parameters as deployed in the embedded real-time implementation.

| Parameter | Description                 | Value              |
|-----------|-----------------------------|--------------------|
| $N_b$     | Total population size       | 20                 |
| $N_e$     | Employed/onlooker bees each | 10                 |
| $n_p$     | Decision variable dimension | 3                  |
| $\ell$    | Abandonment threshold       | 15                 |
| $\lambda$ | Regularization coefficient  | $5 \times 10^{-4}$ |

#### 5.4. Real-Time Feasibility

A necessary condition for embedding either optimizer within the 1 kHz control loop is that every operation inside the BSO and ABC gain-tuning functions completes within the  $T_s = 0.001$  s interrupt period on every call, not merely on average. Both implementations are constructed so that every loop has a compile-time-fixed trip count (population size, cluster count, or decision-variable dimension) rather than a data-dependent loop condition, which is the property that makes the per-call operation count structurally bounded rather than merely typically small. For BSO, one call performs a sort of the  $N_{id} = 20$  fitness values and centre extraction ( $\mathcal{O}(N_{id} \log N_{id})$ ), nearest-centre assignment over  $N_{id} \times N_c = 80$  squared-distance evaluations, idea generation with a bounded double linear scan over cluster membership, and  $N_{id}$  objective re-evaluations for elitist selection, on the order of  $2 \times 10^3$  elementary operations per call. ABC admits a comparable bound given its similar population size ( $N_b = 20$ ) and three-phase structure; two previously data-dependent retry loops in the employed- and onlooker-bee neighbour-selection steps have been replaced with bounded-iteration equivalents to ensure that this property holds in the deployed implementation.

On-target execution-time logs were not preserved for the reported trials, so no measured per-call timing distribution is available for the real-time target itself. The real-time feasibility argument therefore rests on the structural bound established above: because every loop in both optimizers terminates after a compile-time-fixed number of iterations, the elementary-operation count per call is bounded above by the same figure of order  $2 \times 10^3$  regardless of the random draws made during that call, so neither optimizer can degrade into a long-running call under adverse inputs, which is the specific hazard that

data-dependent loops would introduce. A certified on-target worst-case execution time (WCET), obtained either by static timing analysis of the compiled Real-Time Windows Target executable or by dedicated low-overhead on-target instrumentation, was not performed for this study and is explicitly identified as a concrete item for future real-time validation work. The claim of “real-time feasibility” in the sequel should therefore be understood as structural (bounded operation count per call) rather than as a certified WCET guarantee.

## 6. Results and Discussion

This section presents a comprehensive evaluation of the four backstepping control architectures through both full nonlinear simulation and 100 s real-time experiments on the physical TRMS platform at a 1 kHz sampling rate. The four controllers evaluated are: (i) the conventional non-adaptive backstepping baseline [28]; (ii) adaptive backstepping with tuning functions (Section 3); (iii) BSO-optimized backstepping (Section 5.2); (iv) ABC-optimized backstepping (Section 5.3). All controllers share the common HGO designed in Section 4 and are deployed under identical operating conditions, reference trajectories, and initial conditions to ensure a rigorous and unbiased comparison.

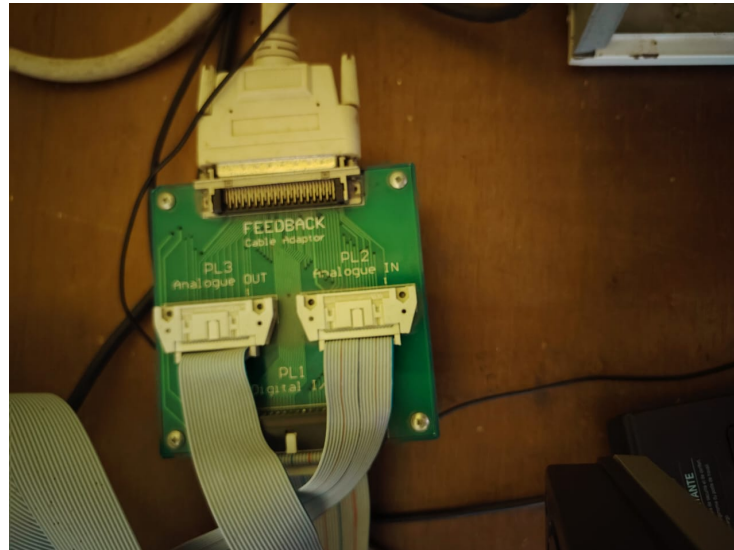
### 6.1. Experimental and Simulation Setup

Simulation studies were carried out in MATLAB/Simulink (MathWorks Inc., Natick, MA, USA, Version R2014a), integrating the full nonlinear model of Equation (2) with a fixed-step Euler (ode1) solver at a step size of 0.001 s, matched to the sampling interval of the physical rig described below. Real-time experiments were conducted on the physical TRMS platform (i.e., the rig itself, not a simulated plant model) driven by a host PC through an Advantech PCI-1711 multifunction data acquisition card. The card’s 12-bit analog output channels supply the drive voltages  $u_1, u_2$  to the main- and tail-rotor DC motor amplifiers, subject to the saturation limit  $|u_i| \leq 2.5$  V introduced in Section 2, and its high-speed digital inputs read the 2000-pulse-per-revolution incremental optical encoders on the pitch and yaw axes at the resolution of  $\approx 0.00314$  rad already stated. The PCI-1711 is connected to the TRMS base unit through the manufacturer-supplied SCSI-type multi-conductor cable and adapter that carry, in a single shielded bundle, the outbound analog voltage commands and the returning digital encoder pulse trains. The control law and both metaheuristic optimizers were compiled to the Real-Time Windows Target kernel on the host PC and executed at a fixed rate of 1 kHz ( $T_s = 0.001$  s). The pitch and yaw tracking signals shown in the figures below were logged in the Simulink data-logging workspace at the same 1 kHz rate through the trial and post-processed off-line to compute the RMSE and ITAE metrics of Table 4. This legacy MATLAB release was retained on the laboratory control computer specifically to preserve stable driver compatibility with the PCI-1711 card. Figure 4 shows the physical DAQ interface linking the controller to the TRMS rig.

**Table 4.** Quantitative comparison of real-time tracking performance for all controller architectures on the physical TRMS platform.

| Controller Architecture              | Pitch Axis    |               | Yaw Axis      |                |
|--------------------------------------|---------------|---------------|---------------|----------------|
|                                      | RMSE (rad)    | ITAE (rad·s)  | RMSE (rad)    | ITAE (rad·s)   |
| Conv. Backstepping [28] <sup>†</sup> | 0.2571        | 1990.24       | 2.5624        | 21281.99       |
| Adapt. BS + Tuning Func. + HGO       | 0.1588        | 609.55        | <b>0.2244</b> | 1021.24        |
| BS + ABC + HGO                       | 0.0822        | 348.52        | 0.2255        | <b>1005.37</b> |
| BS + BSO + HGO                       | <b>0.0821</b> | <b>343.72</b> | 0.2291        | 1019.58        |

<sup>†</sup> Terminated at  $t \approx 42$  s due to yaw divergence; metrics computed over the available 42 s interval. Bold values indicate the best performance per metric.



**Figure 4.** Physical data-acquisition interface between the host PC and the Twin Rotor MIMO System. The Advantech PCI-1711 multifunction DAQ card, installed in the host PC, connects to the TRMS base unit via the manufacturer-supplied Feedback Cable Adaptor shown here. This board routes the outbound analog motor drive voltages (via the PL3 Analogue OUT ribbon) and the returning digital encoder pulse trains (via the PL1 Digital I/O ribbon).

The reference trajectories were identical across simulation and hardware for both channels, and were selected as smooth sinusoids representative of the continuous maneuvering regime for which the TRMS is intended as a benchmark:

$$x_{1d}(t) = 0.5 \sin(0.5t) \text{ rad}, \quad x_{4d}(t) = 1.0 \sin(0.5t) \text{ rad}, \quad (35)$$

for the pitch and yaw axes, respectively.

## 6.2. Research Methodology

The overarching research strategy adopted in this study is a controlled comparative experiment, in which four backstepping-based control architectures, namely the conventional non-adaptive baseline [28], the proposed adaptive backstepping controller with tuning functions (Section 3), and the two proposed metaheuristic-optimized variants (Section 5), are evaluated under identical operating conditions to isolate the effect of the gain management strategy on closed-loop tracking performance.

The research proceeds in four steps. First, the common High-Gain Observer (Section 4) is designed and validated to provide the full state vector from the two available encoder outputs, since all four controllers rely on this reconstructed state. Second, each of the four control architectures is implemented in full nonlinear simulation to verify nominal closed-loop stability and to detect any structural design errors prior to hardware deployment. Third, all four architectures are deployed sequentially on the physical TRMS platform under the identical reference trajectories of Equation (35) and identical initial conditions, and are each run for a fixed 100 s trial, or until early termination in the case of the baseline. Fourth, the resulting tracking records are post-processed into the RMSE and ITAE metrics of Section 6.6.

The controlled factors, held identical across all four architectures, are the physical TRMS platform and its parameters (Table 1), the High-Gain Observer (Section 4), the reference trajectories of Equation (35), the sampling rate (1 kHz), and the initial conditions. The manipulated factors are the presence or absence of parameter adaptation  $\hat{\theta}$ , and the presence, absence, and choice of algorithm for online design-gain optimization. The depen-

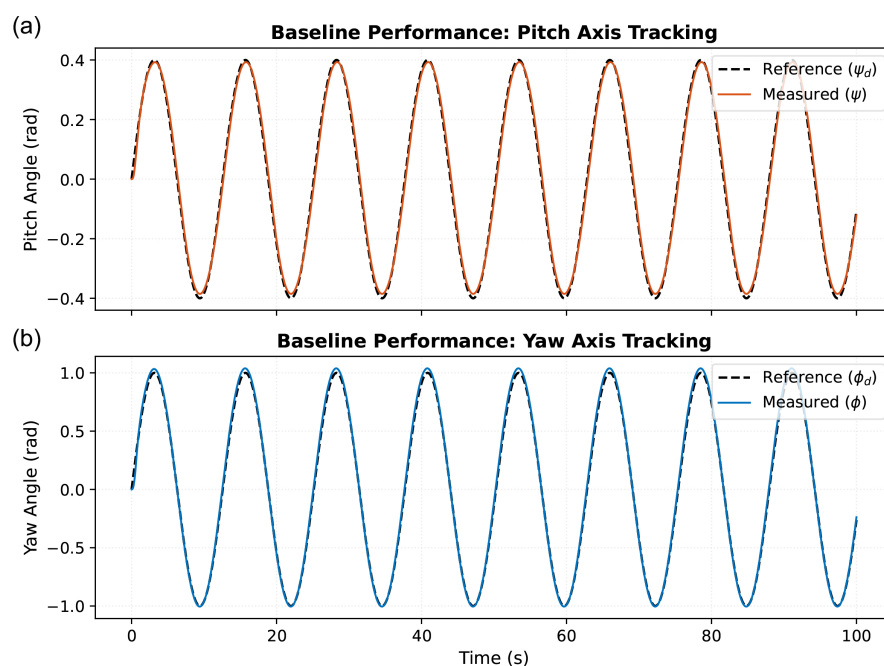
dent measures are the RMSE and ITAE of the pitch and yaw tracking error over the full 100s trial, or over the available interval in the case of the baseline. This factorial structure ensures that any observed difference in tracking performance between architectures is attributable to the gain management strategy rather than to confounding variation in the plant, the observer, the reference signal, or the operating conditions.

### 6.3. Simulation Results

The full nonlinear TRMS model (2), including the cross-coupling terms, is implemented in MATLAB/Simulink. All four controllers are evaluated under identical reference trajectories in a noise-free, ideal simulation environment to establish the nominal tracking capability of each architecture prior to real-time deployment.

#### 6.3.1. Baseline Backstepping

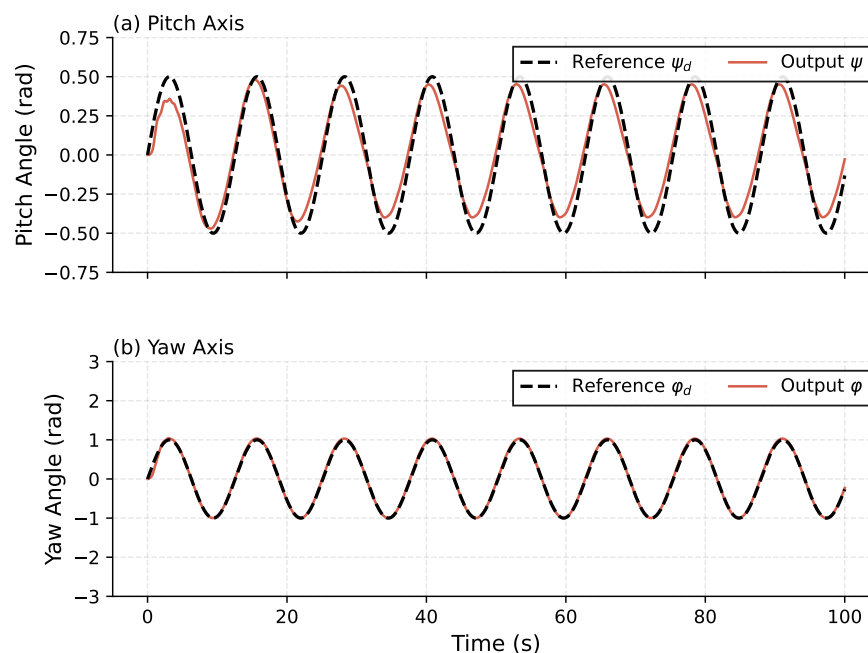
Figure 5 presents the simulation tracking performance of the conventional non-adaptive backstepping controller. In the absence of measurement noise, actuator quantization, and unmodeled physical disturbances, the baseline controller achieves satisfactory tracking of both the pitch and yaw reference trajectories, with tracking errors bounded and converging to a small neighborhood of the origin, consistent with the Lyapunov stability guarantees established in [28].



**Figure 5.** Simulation results for the conventional backstepping baseline: (a) pitch tracking; (b) yaw tracking.

#### 6.3.2. Adaptive Backstepping with Tuning Functions (Simulation)

Figure 6 presents the simulation tracking performance of the proposed adaptive backstepping controller with tuning functions. In the ideal simulation environment, the adaptive controller achieves tracking performance comparable to the baseline, with marginal improvement attributable to online parameter adaptation. The parameter estimates  $\hat{\theta}$  converge smoothly to values consistent with the nominal system parameters.



**Figure 6.** Simulation results for adaptive backstepping with tuning functions: (a) pitch tracking; (b) yaw tracking.

### 6.3.3. BSO- and ABC-Optimized Backstepping (Simulation)

Figures 7 and 8 present the simulation results for the BSO- and ABC-optimized controllers. In both cases the online gain optimization converges rapidly within the first few seconds, with the design gains settling to values that yield slightly improved transient response relative to the fixed-gain controllers. The near-identical convergence of the two optimizers is consistent with the low dimensionality of the optimization problem ( $n_p = 3$ ).

The simulation environment presented above is deliberately idealized: the reference trajectories of Equation (35) are smooth analytic signals, sensor measurements are noise-free and unquantized, and the nonlinear model (2) excludes any unmodeled dynamics, backlash, or parameter drift. This idealization is standard practice for establishing nominal controller behavior, but by itself provides no guarantee of practical validity. The baseline controller, for example, achieves satisfactory tracking under exactly these idealized conditions (Figure 5) yet diverges within 42 s once deployed on hardware (Section 6.4). Four specific gaps between the simulated and physical environments account for this discrepancy: (i) encoder quantization limits angular resolution to  $\approx 0.00314$  rad, a discrete measurement floor absent from the continuous-state simulation; (ii) the actuator saturation limit  $|u_i| \leq 2.5$  V is enforced in hardware at all times, whereas transient control demand in simulation is only implicitly bounded by the controller's own gains; (iii) mechanical friction and aerodynamic cross-coupling are modeled by the fixed nominal coefficients of Table 1, which do not capture their dependence on ambient temperature, bearing wear, or airflow recirculation in the test cell; (iv) the physical data-acquisition chain introduces small transport delays and digital-to-analog quantization absent from the continuous-time Simulink model. The real-time experiments of Section 6.4 are therefore not a redundant confirmation of the simulation results, but rather the primary evidence for the claims of this paper. The simulation results serve only to establish that each controller is nominally well-posed before this harsher test.

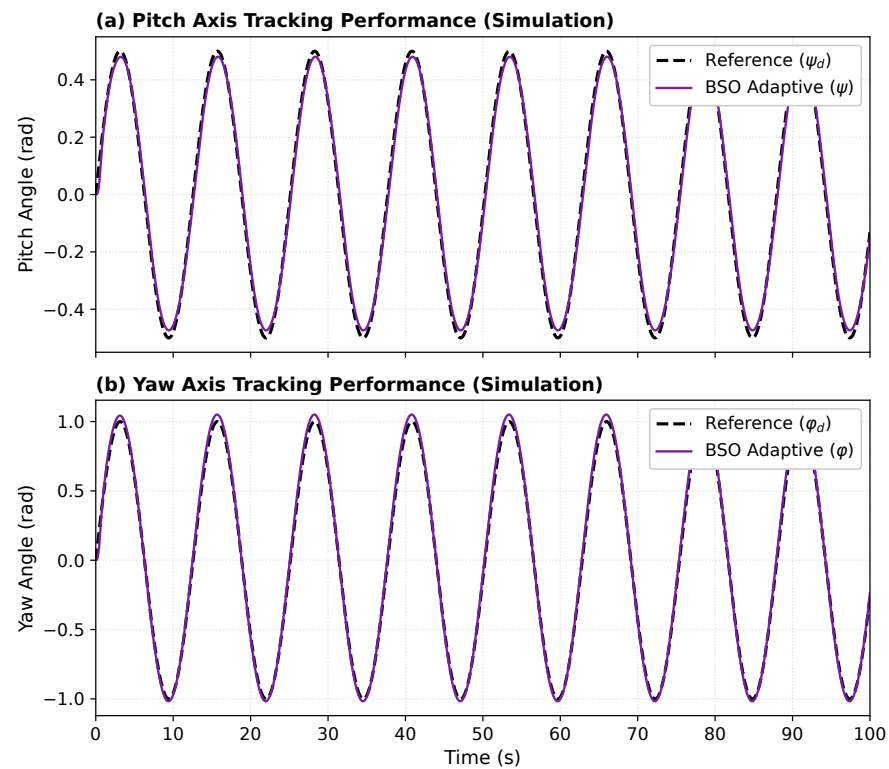


Figure 7. Simulation results for BSO-optimized backstepping: (a) pitch tracking; (b) yaw tracking.

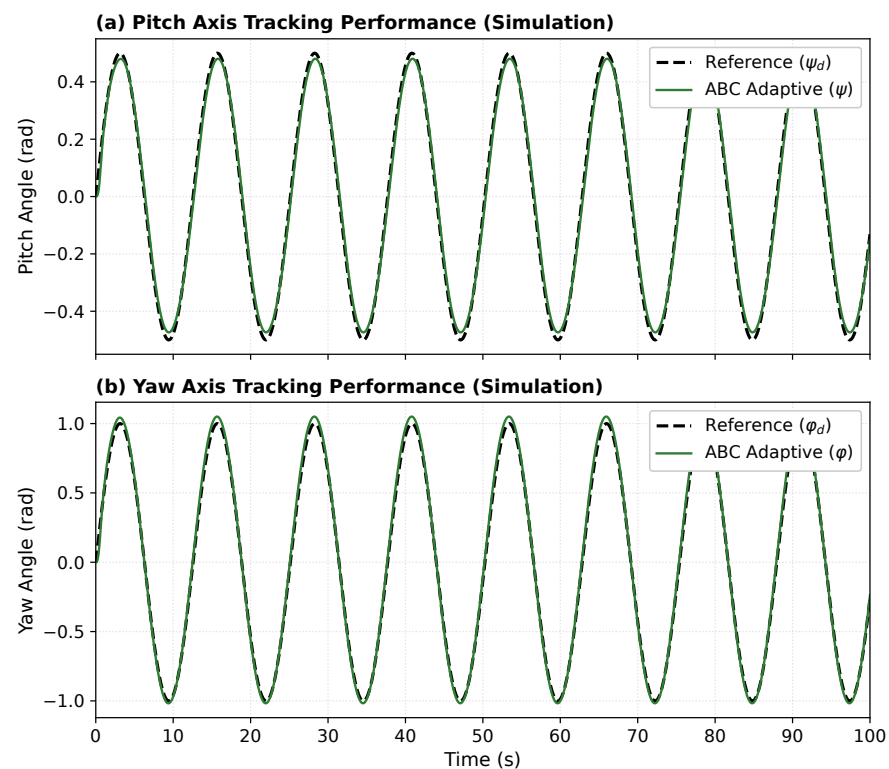


Figure 8. Simulation results for ABC-optimized backstepping: (a) pitch tracking; (b) yaw tracking.

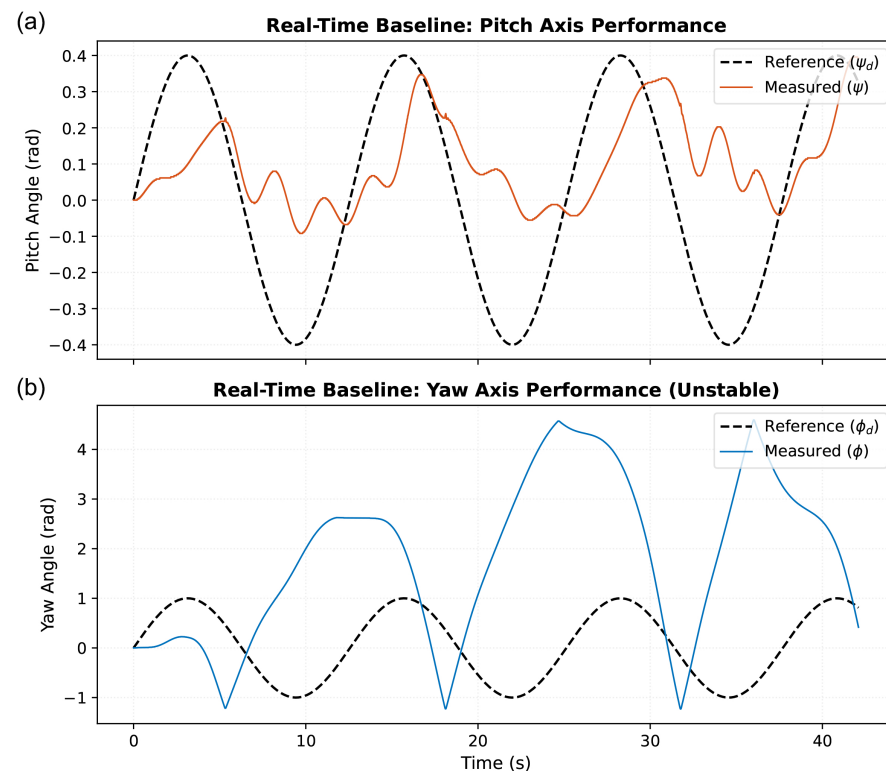
#### 6.4. Real-Time Experimental Results

The four controllers are deployed on the physical TRMS platform via MATLAB/Simulink Real-Time with a fixed-step solver at  $T_s = 1$  ms. The experimental operating conditions differ fundamentally from the simulation environment through the presence of sensor quantization (encoder resolution  $\approx 0.00314$  rad), actuator saturation ( $|u_i| \leq 2.5$  V), un-

modeled aerodynamic perturbations, mechanical friction nonlinearities, and parametric variations due to ambient conditions.

#### 6.4.1. Baseline Backstepping (Divergence)

Figure 9 presents the real-time experimental results for the conventional baseline controller. While the pitch-axis tracking initially exhibits acceptable performance, the yaw channel develops a progressive divergence that becomes clearly discernible after approximately 20 s and accelerates beyond 4 rad by  $t = 42$  s, at which point the experiment was manually terminated to prevent mechanical damage. The baseline experimental RMSE values of 0.2571 rad (pitch) and 2.5624 rad (yaw) quantify the severity of this simulation-to-reality gap.



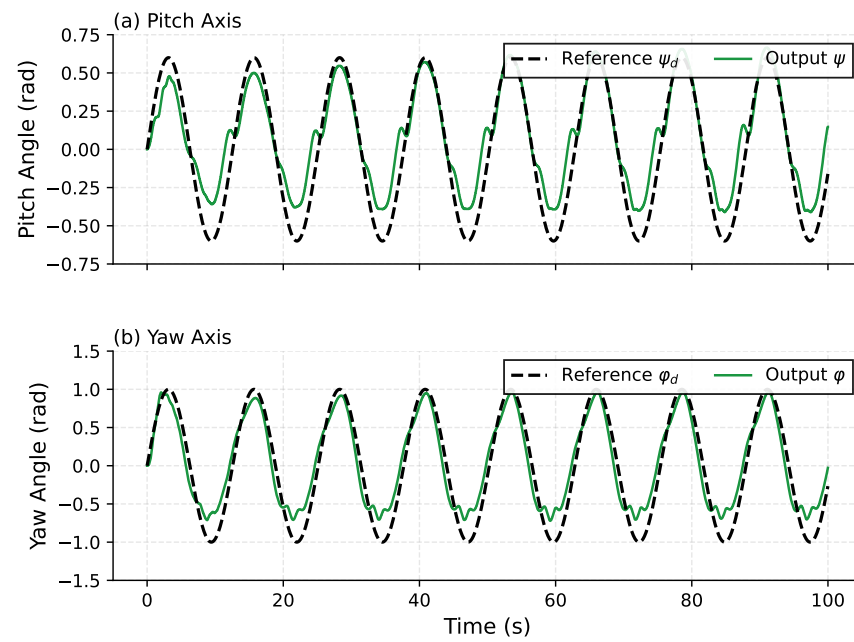
**Figure 9.** Real-time experimental results for the conventional backstepping baseline: (a) pitch tracking; (b) yaw tracking. The yaw channel exhibits progressive divergence exceeding 4 rad, necessitating experiment termination at  $t = 42$  s.

#### 6.4.2. Adaptive Backstepping with Tuning Functions

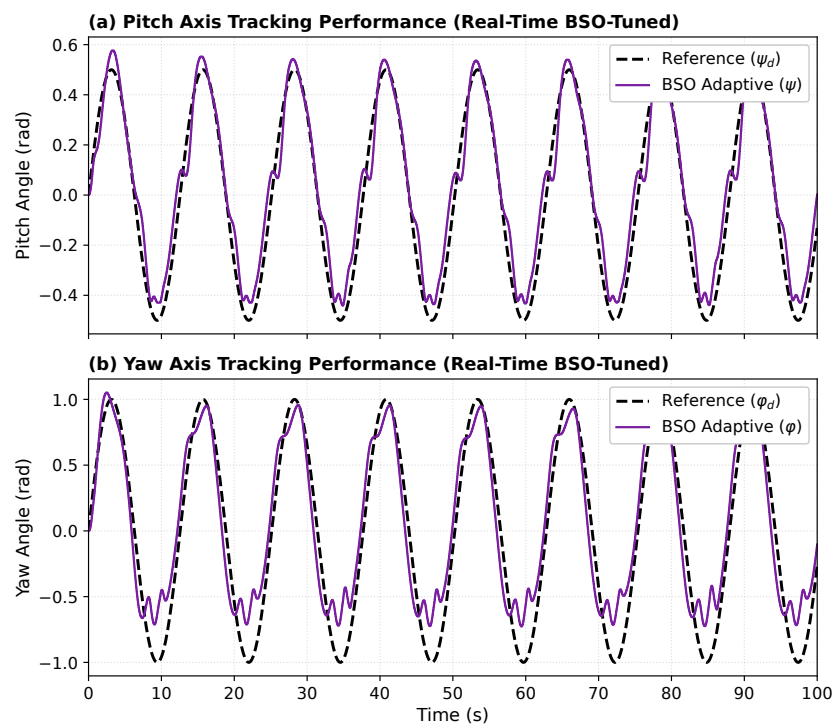
Figure 10 presents the real-time results for the proposed adaptive controller. In stark contrast to the baseline, the tuning-functions architecture maintains stable trajectory tracking over the full 100 s duration on both axes. The yaw-channel divergence is entirely eliminated, confirming that the over-parametrization problem was the root cause of the yaw instability. The pitch RMSE is reduced to 0.1588 rad (38% improvement), while the yaw RMSE is reduced to 0.2244 rad (91% improvement).

#### 6.4.3. BSO- and ABC-Optimized Backstepping

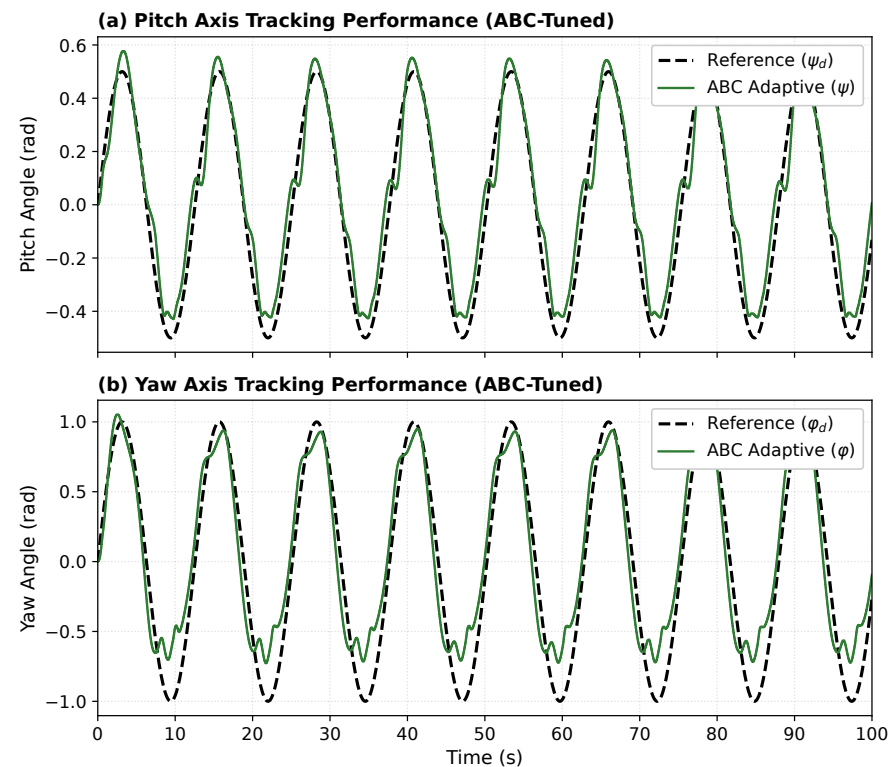
Figures 11 and 12 present the real-time results for the metaheuristic-optimized controllers. Both achieve the best overall tracking performance, particularly on the pitch axis, where the RMSE is reduced to 0.0821 rad (BSO) and 0.0822 rad (ABC), a 68% improvement over the baseline. The yaw RMSE values (0.2291 rad and 0.2255 rad, respectively) represent a 91% reduction. The near-identical performance is consistent with the low dimensionality of the optimization problem ( $n_p = 3$ ).



**Figure 10.** Real-time experimental results for adaptive backstepping with tuning functions: (a) pitch tracking; (b) yaw tracking. Stable tracking is maintained over the full 100 s duration.



**Figure 11.** Real-time experimental results for BSO-optimized backstepping: (a) pitch tracking; (b) yaw tracking.

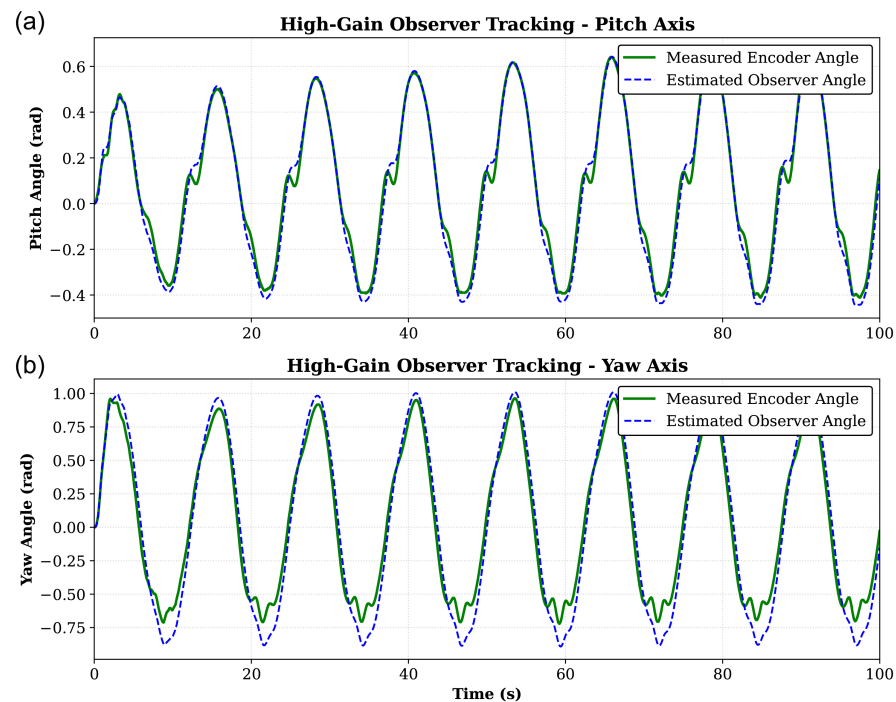


**Figure 12.** Real-time experimental results for ABC-optimized backstepping: (a) pitch tracking; (b) yaw tracking.

### 6.5. Observer Estimation Performance

To validate the High-Gain Observer directly rather than solely through closed-loop tracking, Figure 13 reports the real-time estimation performance of the observer on the physical TRMS over the full 100 s trial. The estimated pitch and yaw angles ( $\hat{x}_1$ ,  $\hat{x}_4$ ) are overlaid on the corresponding 2000-pulse-per-revolution encoder measurements, providing a direct quantitative measure of estimation accuracy for the two measured states. The observer estimates track the encoder signals closely throughout the reference maneuver, with root-mean-square estimation errors of 0.0383 rad on the pitch axis and 0.1304 rad on the yaw axis. Relative to the reference amplitudes of 0.5 rad (pitch) and 1.0 rad (yaw), these correspond to normalized estimation errors of approximately 7.7% and 13%, respectively; the larger yaw estimation error is consistent with the greater mechanical vibration and encoder-quantization sensitivity of the tail-rotor axis.

Three observations drawn from Figure 13 directly address the observer-validation concerns. (i) *Peaking*. The recorded pitch and yaw estimates track the measured angles from  $t = 0$  without any start-up excursion, because the observer states were initialized at the rig's rest condition, which coincides with the physical initial state of the plant on the two measured channels; the initial estimation error on the angle channels is therefore negligible and no peaking spike is visible in Figure 13. This behavior is consistent with the empirically selected gains ( $\theta_1 = 2$ ,  $\theta_2 = 1.5$ ). (ii) *Noise sensitivity*. The estimates remain smooth over the entire trial, with only minor ripple visible on the yaw channel near velocity reversals, where encoder quantization and mechanical vibration are most pronounced; the smaller yaw-channel gain limits this amplification. This ripple is the principal contributor to the 0.1304 rad yaw estimation error accumulated over the full run. (iii) *Sustained accuracy*. The observer maintains this estimation accuracy over the complete 100 s of closed-loop operation, confirming that it supplies the backstepping control law with reliable pitch and yaw angle estimates at the 1 kHz rate.



**Figure 13.** Real-time High-Gain Observer estimation performance on the physical TRMS, with both axes in radians: (a) estimated pitch angle  $\hat{x}_1$  (blue) versus the measured encoder angle (green); (b) estimated yaw angle  $\hat{x}_4$  (blue) versus the measured encoder angle (green). The observer estimates track the encoder measurements over the full 100 s trial with no start-up peaking and only minor ripple near yaw velocity reversals.

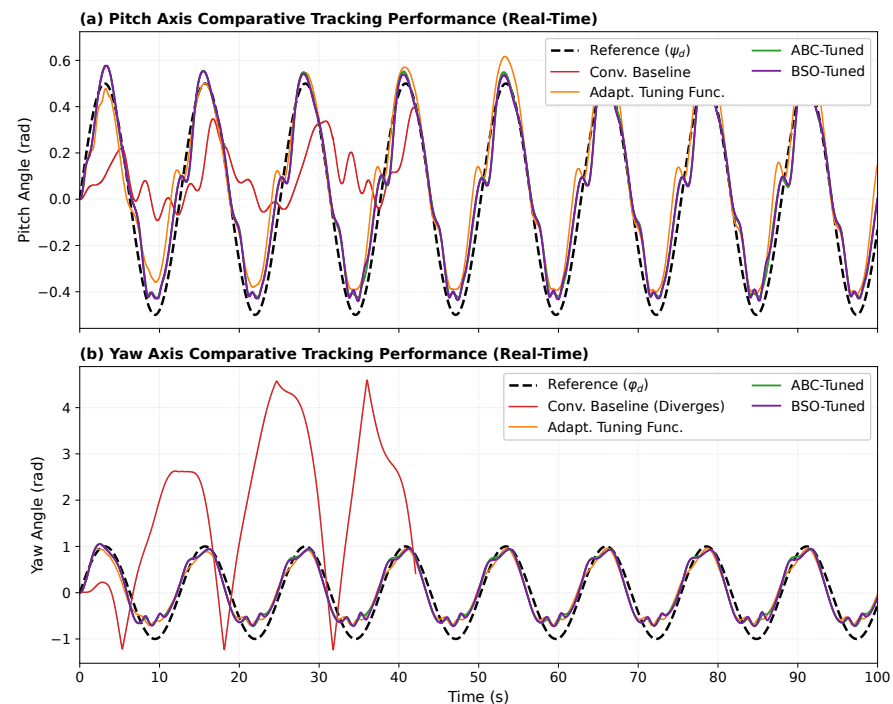
The two directly validated quantities are therefore the pitch angle estimate  $\hat{x}_1$  and the yaw angle estimate  $\hat{x}_4$ . The four remaining state estimates, namely the angular velocities ( $\hat{x}_2$ ,  $\hat{x}_5$ ) and the rotor torques ( $\hat{x}_3$ ,  $\hat{x}_6$ ), cannot be validated against ground truth because the TRMS rig used in this study is not equipped with velocity or torque sensors. Their quality is only indirectly supported: the backstepping control law depends on these estimates at every 1 kHz cycle, so any severe peaking or unbounded noise amplification on the velocity or torque channels would necessarily have driven the control signal into the 2.5 V saturation limit and destabilized the loop, which was not observed. The full-state observer is therefore best described as directly validated on the pitch and yaw angle channels and indirectly supported (through bounded and stable 100 s closed-loop operation) on the velocity and torque channels. A dedicated velocity/torque estimation study against an auxiliary reference sensor is identified as future work.

### 6.6. Comparative Analysis

To provide a definitive quantitative comparison, all four experimental tracking responses are overlaid in Figure 14. The comprehensive performance metrics are compiled in Table 4, and the percentage improvements relative to the baseline are summarized in Table 5.

**Table 5.** Percentage improvement of each proposed controller relative to the conventional baseline.

| Controller       | RMSE Reduction (%) |     | ITAE Reduction (%) |     |
|------------------|--------------------|-----|--------------------|-----|
|                  | Pitch              | Yaw | Pitch              | Yaw |
| Tuning Functions | 38                 | 91  | 69                 | 95  |
| BSO-Optimized    | 68                 | 91  | 83                 | 95  |
| ABC-Optimized    | 68                 | 91  | 82                 | 95  |



**Figure 14.** Comparative real-time experimental tracking for all four controllers overlaid with the reference trajectory: (a) pitch axis; (b) yaw axis.

### 6.7. Discussion

*Role of tuning functions.* The adaptive tuning-functions architecture resolves the fundamental yaw-channel instability that renders the conventional baseline unusable beyond 42 s. The 91% reduction in yaw RMSE and the complete elimination of divergent behavior confirm that the over-parametrization problem was the root cause of the yaw instability. The comparatively modest pitch improvement (38%) indicates that the primary benefit of tuning functions is structural stability rather than precision enhancement.

*Role of metaheuristic optimization.* The BSO and ABC controllers achieve a further 68% pitch RMSE reduction relative to the baseline, compared with 38% for tuning functions alone, attributable to online refinement of the design gains ( $c_1$  through  $c_6$ ). The yaw RMSE achieved by both metaheuristic controllers (approximately 0.23 rad) is comparable to that of the tuning-functions architecture, suggesting that yaw stability is primarily a structural property rather than a gain-optimization problem.

*BSO versus ABC.* The near-identical performance is attributable to the low dimensionality ( $n_p = 3$ ). For higher-dimensional problems, the distinct exploration mechanisms of BSO (cluster-based recombination) and ABC (scout reinitialization) may yield more differentiated performance.

*Complementary mechanisms.* Tuning functions ensure structural closed-loop stability by eliminating over-parametrization (a stability mechanism), while metaheuristic optimization refines the design gains for improved accuracy (a performance mechanism). These two mechanisms are complementary rather than competing, and their integration within a unified architecture represents a promising direction for future research.

## 7. Conclusions

This paper presented a comprehensive comparative study of three observer-based backstepping control architectures for real-time trajectory tracking of the Twin Rotor MIMO System: adaptive backstepping with tuning functions, BSO-optimized backstepping, and ABC-optimized backstepping. All three were integrated with a common High-Gain Observer for

full-state reconstruction from encoder measurements and validated through both nonlinear simulation and 100 s real-time experiments on the physical TRMS platform at 1 kHz.

The experimental results demonstrated a clear hierarchical improvement over the conventional non-adaptive baseline, which diverged after 42 s due to progressive yaw-channel instability. The adaptive tuning-functions architecture eliminated this instability entirely, reducing the yaw-axis RMSE by 91% (from 2.5624 rad to 0.2244 rad) and the pitch-axis RMSE by 38% (from 0.2571 rad to 0.1588 rad). The BSO and ABC controllers achieved a further pitch-precision improvement, reducing the pitch RMSE to 0.0821 rad and 0.0822 rad respectively (68% reduction), while maintaining comparable yaw performance. The High-Gain Observer was additionally validated on the two measured states through direct real-time comparison of its pitch and yaw angle estimates against the encoder measurements, confirming accurate angle estimation, absence of peaking, and low noise sensitivity over the full trial; the four unmeasured state estimates (angular velocities and rotor torques) are only indirectly supported by the sustained boundedness of the closed loop, since no velocity or torque reference sensor is available on the rig.

The study identified two complementary mechanisms for improving real-time back-stepping performance on coupled nonlinear platforms: tuning functions address structural closed-loop stability by eliminating over-parametrization, while metaheuristic optimization refines the design gains for enhanced tracking precision. These mechanisms operate on orthogonal aspects of the control problem and are not competing alternatives. Future research directions include the integration of tuning functions with metaheuristic gain optimization within a unified architecture, a dedicated hyperparameter sensitivity study of the BSO and ABC optimizers, a certified on-target WCET analysis of the embedded optimizers, direct validation of the velocity and torque estimates against an auxiliary reference sensor, and the extension to higher-dimensional aerial platforms under time-varying parametric perturbations and external wind disturbances.

**Author Contributions:** Conceptualization, A.K., A.B. and J.-J.T.; methodology, A.K., A.B. and S.T.; software, A.K. and M.H.Z.; validation, A.K., A.B. and M.M.; formal analysis, A.K., S.T., M.M., A.A. and M.H.Z.; investigation, A.K., M.M., S.T. and A.A.; resources, A.B., A.A. and J.-J.T.; data curation, A.K. and M.H.Z.; writing—original draft preparation, A.K.; writing—review and editing, A.K., A.B., S.T., A.A., J.-J.T. and M.H.Z.; visualization, A.K., S.T. and M.M.; supervision, A.B. and J.-J.T.; project administration, A.B. and J.-J.T.; funding acquisition, J.-J.T. All authors have read and agreed to the published version of the manuscript.

**Funding:** The Article Processing Charge (APC) was funded by Multimedia University, Malaysia.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

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

**Conflicts of Interest:** The authors declare no conflicts of interest.

## Abbreviations

|      |                                       |
|------|---------------------------------------|
| ABC  | Artificial Bee Colony                 |
| BSO  | Brain Storm Optimization              |
| DAQ  | Data Acquisition                      |
| HGO  | High-Gain Observer                    |
| ITAE | Integral Time-weighted Absolute Error |
| LQG  | Linear Quadratic Gaussian             |
| MIMO | Multiple Input Multiple Output        |

|      |                                  |
|------|----------------------------------|
| PID  | Proportional Integral Derivative |
| PSO  | Particle Swarm Optimization      |
| RMSE | Root Mean Square Error           |
| TRMS | Twin Rotor MIMO System           |
| UAV  | Unmanned Aerial Vehicle          |
| WCET | Worst-Case Execution Time        |

## Appendix A. General Tuning Functions Formulation

Consider a single-input nonlinear system in parametric strict-feedback form:

$$\begin{cases} \dot{x}_1 = x_2 + \Phi_1^T(x_1) \theta + \psi_1(x_1) \\ \dot{x}_2 = x_3 + \Phi_2^T(x_1, x_2) \theta + \psi_2(x_1, x_2) \\ \vdots \\ \dot{x}_n = b u + \Phi_n^T(x) \theta + \psi_n(x) \end{cases} \quad (\text{A1})$$

where  $x \in \mathbb{R}^n$  is the state,  $u \in \mathbb{R}$  the scalar control,  $\theta \in \mathbb{R}^p$  the unknown constant parameter vector,  $b > 0$  a known positive constant,  $\Phi_i$  the regression vectors, and  $\psi_i$  known nonlinear functions. The adaptive backstepping with tuning functions design proceeds through  $n$  recursive steps. At each step  $k$ : (1) an error variable  $z_k = x_k - \alpha_{k-1}$  is defined, with  $\alpha_0 = x_{1d}$ ; (2) a tuning function is computed as  $\tau_k = \tau_{k-1} + \Phi_k z_k$ , with  $\tau_0 = 0$ ; (3) a stabilizing function  $\alpha_k$  is designed to render  $\dot{V}_k \leq -\sum_{i=1}^k c_i z_i^2 + z_k z_{k+1}$ . The key distinction from standard adaptive backstepping is that the update law  $\hat{\theta}$  is not committed at any intermediate step; only at the final step is it defined as

$$\hat{\theta} = \Gamma \tau_n, \quad \Gamma = \Gamma^T > 0, \quad (\text{A2})$$

ensuring a single estimator per uncertain parameter regardless of system order  $n$ , thereby eliminating over-parametrization. The total Lyapunov function  $V = \frac{1}{2} \sum_{k=1}^n z_k^2 + \frac{1}{2} \tilde{\theta}^T \Gamma^{-1} \tilde{\theta}$  satisfies  $\dot{V} = -\sum_{k=1}^n c_k z_k^2 < 0$ , with all cross-terms cancelling by construction. By LaSalle's Invariance Theorem [40],  $z_k \rightarrow 0$  for all  $k$  and  $\hat{\theta}$  remains bounded.

## Appendix B. Yaw Subsystem Backstepping Derivation and Observer Jacobian

### Appendix B.1. Detailed Yaw Subsystem Backstepping Derivation

An analogous three-step adaptive backstepping design is applied to the yaw subsystem. Following elimination of the cross-coupling terms, the decentralized yaw dynamics governing  $(x_4, x_5, x_6)$  are

$$\begin{cases} \dot{x}_4 = x_5 \\ \dot{x}_5 = \frac{1}{I_2} (a_2 x_6^2 + b_2 x_6 - B_{1\varphi} x_5 - B_{2\varphi} \text{sign}(x_5)) \\ \dot{x}_6 = -\frac{T_{20}}{T_{21}} x_6 + \frac{K_2}{T_{21}} u_2. \end{cases} \quad (\text{A3})$$

The aerodynamic moment  $M_2 = a_2 x_6^2 + b_2 x_6 \approx q_{21} x_6 + q_{20}$ , with normalized function  $E(x_6) = \frac{1}{I_2} (a_2 x_6^2 + b_2 x_6)$ . The subsystem is cast in strict-feedback form with

$$\Phi_1(x_4) = \Phi_3(x) = 0, \quad \Phi_2^T(x_4, x_5) = \begin{bmatrix} -x_5 & -\text{sign}(x_5) \end{bmatrix}, \quad (\text{A4})$$

$$\theta = \begin{bmatrix} \frac{B_{1\varphi}}{I_2} & \frac{B_{2\varphi}}{I_2} \end{bmatrix}^T, \quad b = \frac{K_2}{T_{21}} > 0. \quad (\text{A5})$$

*Step 1.* With  $z_1 = x_4 - x_{4d}$ ,  $\dot{z}_1 = x_5 - \dot{x}_{4d}$ ,  $z_2 = x_5 - \alpha_1$ , and  $V_1 = \frac{1}{2}z_1^2$ , the first stabilizing function is  $\alpha_1 = -c_4 z_1 + \dot{x}_{4d}$ , giving  $\dot{V}_1 = -c_4 z_1^2 + z_1 z_2$ . Note that the error symbols  $z_1, z_2, z_3$  used in this appendix are the yaw-subsystem error variables and are distinct from the pitch-subsystem error variables of Section 3.2.

*Step 2.* The second error dynamics are  $\dot{z}_2 = z_3 + \alpha_2 + \Phi_2^T \theta + c_4 \dot{x}_5 - c_4 \dot{x}_{4d} - \ddot{x}_{4d}$ , with  $z_3 = E(x_6) - \alpha_2$ . The tuning function is  $\tau_2 = \Phi_2 z_2$ , the update law  $\dot{\hat{\theta}} = \Gamma_2 \Phi_2 z_2$  with  $\Gamma_2 \in \mathbb{R}^{2 \times 2}$  positive definite, and

$$\alpha_2 = -c_5 z_2 - z_1 - \Phi_2^T \hat{\theta} - c_4 \dot{x}_5 + c_4 \dot{x}_{4d} + \ddot{x}_{4d}, \quad (\text{A6})$$

yielding  $\dot{V}_2 = -c_4 z_1^2 - c_5 z_2^2 + z_2 z_3$ .

*Step 3.* With  $b_a = q_{21} K_2 / (I_2 T_{21}) > 0$ , the complete yaw control law is

$$u_2 = \hat{p} \left( -z_2 - c_6 z_3 + \frac{q_{21} T_{20}}{I_2 T_{21}} x_6 + (-c_4 + c_4 c_5) x_5 + \hat{\theta}_2^T (-2c_4 - c_5) \Phi_2 + (-c_4 - c_5) E + (c_4 + c_5) \dot{x}_{4d} + \Phi_2^T \Gamma_2 \tau_2 + c_4 \ddot{x}_{4d} + \ddot{x}_{4d} \right), \quad (\text{A7})$$

with adaptive update laws

$$\begin{cases} u_2 = \hat{p} \bar{u}_2 \\ \bar{u}_2 = \alpha_3 + \ddot{x}_{4d} \\ \dot{\hat{\theta}} = \Gamma_2 \Phi_2 z_2 \\ \dot{\hat{p}} = -\gamma_2 \text{sign}(b_a) \bar{u}_2 z_3. \end{cases} \quad (\text{A8})$$

The total Lyapunov function  $V_3 = V_2 + \frac{1}{2}z_3^2 + \frac{|b_a|}{2\gamma_2} \bar{p}^2$  satisfies  $\dot{V}_3 = -c_4 z_1^2 - c_5 z_2^2 - c_6 z_3^2 < 0$ , establishing boundedness of all closed-loop signals and asymptotic yaw trajectory tracking via LaSalle's Invariance Theorem [40].

## Appendix B.2. Observer Jacobian for the TRMS

Applied to the TRMS ( $n = 6, p = 2, \mu_1 = \mu_2 = 3$ ), the Jacobian of the observer output map takes the block form

$$\frac{\partial \phi(\hat{\mathbf{x}})}{\partial \hat{\mathbf{x}}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \frac{\partial L_f^2 h_1}{\partial \hat{x}_1} & \frac{\partial L_f^2 h_1}{\partial \hat{x}_2} & \frac{\partial L_f^2 h_1}{\partial \hat{x}_3} & 0 & \frac{\partial L_f^2 h_1}{\partial \hat{x}_5} & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{\partial L_f^2 h_2}{\partial \hat{x}_3} & 0 & \frac{\partial L_f^2 h_2}{\partial \hat{x}_5} & \frac{\partial L_f^2 h_2}{\partial \hat{x}_6} \end{bmatrix}, \quad (\text{A9})$$

whose nonzero partial-derivative entries follow directly from differentiating the model (2) with respect to the state ordering  $\mathbf{x} = [\psi, \dot{\psi}, \tau_1, \varphi, \dot{\varphi}, \tau_2]^T$ . In particular,

$$\begin{aligned}\frac{\partial L_f^2 h_1(\hat{\mathbf{x}})}{\partial \hat{x}_1} &= \frac{-M_g \cos(\hat{x}_1) + K_{gy} \sin(\hat{x}_1) \hat{x}_5 (a_1 \hat{x}_3^2 + b_1 \hat{x}_3)}{I_1}, \\ \frac{\partial L_f^2 h_1(\hat{\mathbf{x}})}{\partial \hat{x}_2} &= -\frac{B_{1\psi}}{I_1}, \\ \frac{\partial L_f^2 h_1(\hat{\mathbf{x}})}{\partial \hat{x}_3} &= \frac{(2a_1 \hat{x}_3 + b_1)(1 - K_{gy} \cos(\hat{x}_1) \hat{x}_5)}{I_1}, \\ \frac{\partial L_f^2 h_1(\hat{\mathbf{x}})}{\partial \hat{x}_5} &= -\frac{K_{gy} \cos(\hat{x}_1) (a_1 \hat{x}_3^2 + b_1 \hat{x}_3)}{I_1}, \\ \frac{\partial L_f^2 h_2(\hat{\mathbf{x}})}{\partial \hat{x}_3} &= -\frac{1.75 K_c (2a_1 \hat{x}_3 + b_1)}{I_2}, \\ \frac{\partial L_f^2 h_2(\hat{\mathbf{x}})}{\partial \hat{x}_5} &= -\frac{B_{1\varphi}}{I_2}, \\ \frac{\partial L_f^2 h_2(\hat{\mathbf{x}})}{\partial \hat{x}_6} &= \frac{2a_2 \hat{x}_6 + b_2}{I_2}.\end{aligned}$$

The scaling matrix is defined block-wise as  $\Lambda^{-1}(\theta, \delta) = \text{diag}(\theta_1^{\delta_1} \Delta_1, \theta_2^{\delta_2} \Delta_2)$  with  $\Delta_k = \text{diag}(1, 1/\theta_k, 1/\theta_k^2)$ .

## References

- Bouadi, H.; Bouchoucha, M.; Tadjine, M. Sliding mode control based on backstepping approach for an UAV type-quadrotor. *Int. J. Appl. Math. Comput. Sci.* **2007**, *4*, 12–17.
- Wen, L.; Xu, J.; Li, Q. Backstepping control for tandem rotor helicopter. *Aerosp. Sci. Technol.* **2019**, *93*, 105312.
- Feedback Instruments Ltd. *Twin Rotor MIMO System: Control Experiments Manual 33-949S*; Feedback Instruments: Crowborough, UK, 2006.
- Pratap, B.; Purwar, S. Neuro-adaptive observer based backstepping control for the Twin Rotor MIMO System. *IFAC Proc. Vol.* **2014**, *47*, 477–484.
- Rahideh, A.; Shaheed, M.H. Mathematical modelling and simulation of the Twin Rotor MIMO System. *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.* **2007**, *221*, 89–101. [\[CrossRef\]](#)
- Ahmad, S.M.; Chipperfield, A.J.; Tokhi, M.O. Dynamic modelling and linear-quadratic Gaussian controller design for a Twin Rotor MIMO System. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2003**, *217*, 203–227. [\[CrossRef\]](#)
- Marie, M.J.; Al-Suhail, G.A.; Latif, W.A. PSO-based optimal PID controller for twin rotor MIMO system. *Int. J. Comput. Technol.* **2015**, *14*, 5719–5730. [\[CrossRef\]](#)
- Kacimi, A.; Senoussaoui, A. Feedback Linearization Controller with Thau Observer. *J. Autom. Mob. Robot. Intell. Syst.* **2025**, *19*, 66–74. [\[CrossRef\]](#)
- Tao, C.W.; Taur, J.S.; Chen, Y.C. Design of a parallel distributed compensation controller and stability analysis for TRMS. *IEEE Trans. Fuzzy Syst.* **2010**, *18*, 442–451.
- Purwar, S. Adaptive Terminal Sliding Mode Disturbance Observer Augmented Backstepping Integral Sliding Mode Control for Twin Rotor MIMO System. In *Proceedings of the 35th Irish Signals and Systems Conference (ISSC)*; IEEE: Piscataway, NJ, USA, 2024; pp. 1–6.
- Mondal, S.; Mahanta, C. Adaptive second-order sliding mode controller for a twin rotor multi-input–multi-output system. *IET Control Theory Appl.* **2012**, *6*, 2157–2167. [\[CrossRef\]](#)
- Juang, J.G.; Lin, R.W.; Liu, W.K. Comparison of classical control and intelligent control for a TRMS. *Appl. Math. Comput.* **2011**, *205*, 778–791.
- Kacimi, A.; Beloufa, A.; Tahraoui, S.; Senoussaoui, A.; Zaid, M.H.; Azzouz, A.; Tiang, J.-J. Robust Control of Twin-Rotor MIMO Systems Under Unmodeled Dynamics: Comparative Experimental Validation of Hybrid BSMC and Online QBHO Strategies. *Actuators* **2026**, *15*, 236. [\[CrossRef\]](#)
- Mascaró Palliser, R.; Costa-Castelló, R.; Ramos, G.A. Iterative Learning Control Experimental Results in Twin-Rotor Device. *Math. Probl. Eng.* **2017**, *2017*, 6519497. [\[CrossRef\]](#)

15. Danh, H.D.; Van, C.N.; Van, Q.V. Tracking iterative learning control of TRMS using feedback linearization model with input disturbance. *J. Robot. Control* **2025**, *6*, 446–455. [\[CrossRef\]](#)
16. Bacha, A.; Chelihi, A.; Sentouh, C. Fixed-Time Adaptive Neural Tracking Control for a Helicopter-Like Twin Rotor MIMO System. In *Proceedings of the International Conference on Nonlinear Dynamics and Applications (NODYCON)*; Springer: Cham, Switzerland, 2024; pp. 163–173.
17. Krstic, M.; Kanellakopoulos, I.; Kokotovic, P.V. *Nonlinear and Adaptive Control Design*; John Wiley & Sons: New York, NY, USA, 1995.
18. Beloufa, A.; Tahraoui, S.; Kacimi, A.; Allouache, H.; Tiang, J.-J.; Azzouz, A. Experimental Validation of Robust Backstepping Control for TRMS Using an Interval Type-2 Fuzzy Observer. *Eng* **2026**, *7*, 171. [\[CrossRef\]](#)
19. Ebirim, K.U.; Lecchini-Visintini, A.; Rubagotti, M.; Prempan, E. Constrained model predictive control with integral action for twin rotor MIMO systems. *J. Dyn. Syst. Meas. Control* **2023**, *145*, 081006. [\[CrossRef\]](#)
20. Messaoudi, K.; Doudou, S. Robust Prescribed Time Backstepping for Stabilization and Tracking of a 2-DOF Twin Rotor Model. In *Proceedings of the 2024 International Conference on Advanced Electrical Engineering, Control and Optimization Techniques (ICAECOT)*; IEEE: Piscataway, NJ, USA, 2024.
21. Irfan, S.; Zhao, L.; Javaid, U.; Iqbal, J. URED Observer-Based Feedback Linearized Neuro Adaptive SMC for a Twin Rotor MIMO System: Design and Experimental Study. *Guid. Navig. Control* **2025**, *5*, 559–583. [\[CrossRef\]](#)
22. Liu, W.; Duan, G.; Hou, M.; Golestani, M.; Kong, H. Control of Uncertain High-Order Fully Actuated Strict-Feedback Systems: A Backstepping Approach with High-Gain Observer-Based Derivative Approximation. *IEEE Trans. Cybern.* **2024**, *54*, 7456–7468. [\[CrossRef\]](#) [\[PubMed\]](#)
23. Zhou, J.; Wen, C.; Zhang, Y. Adaptive backstepping control of a class of uncertain nonlinear systems with unknown backlash-like hysteresis. *IEEE Trans. Autom. Control* **2004**, *49*, 1751–1757. [\[CrossRef\]](#)
24. Sodhi, P.; Kar, I. Adaptive backstepping controller design for a Twin Rotor MIMO System. *IFAC-PapersOnLine* **2016**, *49*, 480–485.
25. Khalil, H.K. *Nonlinear Systems*, 3rd ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2002.
26. Bornard, G.; Hammouri, H. A high-gain observer for a class of uniformly observable systems. In *Proceedings of the 30th IEEE Conference on Decision and Control, Brighton, UK, 11–13 December 1991*; IEEE: Piscataway, NJ, USA, 1991; pp. 1494–1496.
27. Atassi, A.N.; Khalil, H.K. A separation principle for the stabilization of a class of nonlinear systems. *IEEE Trans. Autom. Control* **1999**, *44*, 1672–1687. [\[CrossRef\]](#)
28. Beloufa, A.; Tahraoui, S.; Kacimi, A.; Allouache, H.; Tiang, J.-J.; Azzouz, A.; Zaid, M.H. Robust Backstepping Control of a Twin Rotor MIMO System via an RBF-Tuned High-Gain Observer. *Automation* **2026**, *7*, 40. [\[CrossRef\]](#)
29. Ahrens, J.H.; Khalil, H.K. High-gain observers in the presence of measurement noise: A switched-gain approach. *Automatica* **2009**, *45*, 936–943. [\[CrossRef\]](#)
30. Dam, Q.T.; Thabet, R.E.H.; Ali, S.A.; Guerin, F. Observer design for a class of uncertain nonlinear systems with sampled-delayed output using high-gain observer and low-pass filter: Application for a quadrotor UAV. *IEEE Trans. Ind. Electron.* **2024**, *71*, 933–942. [\[CrossRef\]](#)
31. Aguilar-Ibanez, C.; Suarez-Castanon, M.S.; Mendoza-Mendoza, J.A. A survey of optimization-based control for nonlinear systems. *ISA Trans.* **2021**, *118*, 1–15. [\[CrossRef\]](#)
32. Dokeroglu, T.; Sevinc, E.; Kucukyilmaz, T.; Cosar, A. A survey on new generation metaheuristic algorithms. *Comput. Ind. Eng.* **2019**, *137*, 106040. [\[CrossRef\]](#)
33. Shi, Y. Brain storm optimization algorithm. In *Advances in Swarm Intelligence*; Springer: Berlin/Heidelberg, Germany, 2011; pp. 303–309.
34. Cheng, S.; Qin, Q.; Chen, J.; Shi, Y. Brain storm optimization algorithm: A review. *Artif. Intell. Rev.* **2016**, *46*, 445–458. [\[CrossRef\]](#)
35. Karaboga, D. *An Idea Based on Honey Bee Swarm for Numerical Optimization*; Technical Report TR06; Erciyes University: Kayseri, Turkey, 2005.
36. Karaboga, D.; Basturk, B. A powerful and efficient algorithm for numerical function optimization: Artificial Bee Colony (ABC) algorithm. *J. Glob. Optim.* **2007**, *39*, 459–471. [\[CrossRef\]](#)
37. Al-Betar, M.A.; Awadallah, M.A.; Braik, M.S. Recent advances in ABC algorithm: A comprehensive review. *Artif. Intell. Rev.* **2023**, *56*, 7005–7055.
38. Zhao, Y.; Cheng, J.; Cai, J.; Qi, B. Global-best brain storm optimization algorithm based on chaotic difference step and opposition-based learning. *Sci. Rep.* **2024**, *14*, 6432. [\[CrossRef\]](#) [\[PubMed\]](#)
39. Azeez, M.I.; Abdelhaleem, A.M.M.; Elnaggar, S.; Moustafa, K.A.F.; Atia, K.R. Optimization of PID trajectory tracking controller for a 3-DOF robotic manipulator using enhanced Artificial Bee Colony algorithm. *Sci. Rep.* **2023**, *13*, 11164. [\[CrossRef\]](#) [\[PubMed\]](#)
40. LaSalle, J.P. Some extensions of Liapunov's second method. *IRE Trans. Circuit Theory* **1960**, *7*, 520–527. [\[CrossRef\]](#)
41. Vidyasagar, M. *Nonlinear Systems Analysis*; Prentice-Hall: Englewood Cliffs, NJ, USA, 1978.
42. Khalil, H.K. High-gain observers in feedback control: Application to permanent magnet synchronous motors. *IEEE Control Syst. Mag.* **2017**, *37*, 25–41. [\[CrossRef\]](#)

43. Khelfi, M.F. Observateurs non linéaires: Application à la Commande des Robots Manipulateurs. Ph.D. Thesis, Université Henri Poincaré, Nancy, France, 2000.
44. Xia, X.-H.; Gao, W.-B. On exponential observers for nonlinear systems. *Syst. Control Lett.* **1988**, *11*, 319–325. [[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.