MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

HASSIBA BENBOUALI UNIVERSITY OF CHLEF

FACULTY OF TECHNOLOGY

ELECTRONIC DEPARTMENT

THESIS

Presented to obtain the degree of

DOCTORATE

Major: Science and Technology

Specialty: Automation and Industrial Informatics

By

AZEDDINE BELOUFA

Title

AI Techniques for Control and Observation of Nonlinear Dynamic Systems

Defended on 19 / 05 / 2026 before a jury composed of

Rachid TALEB	Professor	University of Chlef	Chairman
Souaad TAHRAOUI	MCA	University of Chlef	Supervisor
Boulanouar SAADAT	MCA	University of Chlef	Examiner
Maamar LATROCH	MCA	University of Chlef	Examiner
Jalal Eddine BENMANSOUR	MRA	Algerian Space Agency (ASAL)	Examiner
Aoued MEHARRAR	MCA	University of Tissemsilt	Examiner

Academic Year 2025 - 2026

To my dear parents,

To my family,

To all those who supported me throughout this journey.

Acknowledgements

First and foremost, I praise **Allah**, the Almighty, for His infinite grace, blessings, and for providing me with the strength and perseverance to complete this work.

I wish to express my deepest and most sincere gratitude to my thesis director, **Dr. TAHRAOUI Souaad**. Her unwavering support, invaluable guidance, and profound technical expertise have been the cornerstones of this research. She fostered an environment of intellectual curiosity and provided the constant encouragement needed to navigate the challenges of this doctoral journey. I feel truly privileged to have conducted this research under her mentorship.

I would like to express my profound respect and sincere gratitude to **Prof. TALEB Rachid** for the great honor of presiding over the jury of this thesis. His distinguished academic standing and scientific rigor make his presence at the head of this committee a true distinction for this work.

I would also like to extend my heartfelt thanks to the members of the examination committee for their time, expertise, and commitment in evaluating this work. I am sincerely grateful to **Dr. SAADAT Boulanouar** and **Dr. LATROCH Maamar**, both of the University of Chlef, for their careful reading and the valuable feedback they provided as examiners. I equally extend my appreciation to **Dr. MEHARRAR Aoued** of the University of Tissemsilt, whose participation as an examiner enriched the scientific discussion of this thesis. I also wish to sincerely thank **Mr. BENMANSOUR Jalal Eddine** from the Algerian Space Agency (ASAL), whose presence as an examiner brought a valuable bridge between academic research and real-world aerospace applications.

I would also like to express my sincere appreciation to **Dr. KACIMI Abderahmane** for his significant help and valuable support throughout the realization of this work. His contributions were essential to the success of this project.

My thanks are also due to the **Hassiba Benbouali University of Chlef** for providing the academic home and the resources that made this achievement possible.

Finally, on a more personal note, I am eternally grateful to my **family** for their unconditional love and support. Their patience, sacrifices, and constant belief in me have been my greatest source of strength and motivation throughout my life.

Abstract

This thesis confronts the gap between theory and practice in controlling the Twin Rotor MIMO System (TRMS). A conventional approach combining a backstepping controller with a fixed-gain High-Gain Observer (HGO) was first evaluated, demonstrating successful performance in simulation for both setpoint regulation and trajectory tracking. When transferred to the physical platform, the regulation performance was preserved, confirming the validity of the design under steady-state conditions. However, the same controller failed completely during real-time trajectory tracking, revealing a fundamental limitation of the fixed-gain observer architecture: its inability to simultaneously ensure fast state estimation and suppress real-world sensor noise under dynamic operating conditions. To resolve this specific failure mode, an adaptive control architecture was designed in which online-learning neural networks intelligently tune the observer gains in real time. Two distinct schemes were developed: a Feedforward Neural Network (FFNN) and a Radial Basis Function Neural Network (RBFNN), both modulating the HGO parameters based on the live observation error. Experimental validation on a TRMS testbed confirms that both proposed methods achieve robust and accurate tracking under the exact conditions that caused the conventional controller to fail. This work presents a proven solution to a documented real-world control problem and provides a direct comparative analysis of the FFNN and RBFNN approaches in this challenging application.

Keywords: *Backstepping Control, High-Gain Observer, Feedforward Neural Network, Radial Basis Function Neural Network, Adaptive Gain Tuning, Twin Rotor MIMO System, Nonlinear Systems, Real-Time Control.*

Résumé

Cette thèse aborde l'écart entre la théorie et la pratique dans le contrôle du système à double rotor multi-entrées multi-sorties (TRMS). Une approche conventionnelle combinant un contrôleur par backstepping et un observateur à grand gain à gain fixe (HGO) a d'abord été évaluée, démontrant des performances satisfaisantes en simulation pour la régulation de consigne et le suivi de trajectoire. Lors du transfert sur la plateforme physique, les performances de régulation ont été préservées, confirmant la validité de la conception en régime permanent. Cependant, le même contrôleur a échoué complètement lors du suivi de trajectoire en temps réel, révélant une limitation fondamentale de l'architecture de l'observateur à gain fixe : son incapacité à garantir simultanément une estimation rapide de l'état et la suppression du bruit des capteurs en conditions de fonctionnement dynamiques. Pour résoudre ce mode de défaillance spécifique, une architecture de contrôle adaptative a été conçue dans laquelle des réseaux de neurones à apprentissage en ligne ajustent intelligemment les gains de l'observateur en temps réel. Deux schémas distincts ont été développés : un réseau de neurones à propagation directe (FFNN) et un réseau de neurones à fonctions de base radiales (RBFNN), modulant tous deux les paramètres du HGO en fonction de l'erreur d'observation en temps réel. La validation expérimentale sur un banc d'essai TRMS confirme que les deux méthodes proposées atteignent un suivi robuste et précis dans les conditions exactes qui ont causé l'échec du contrôleur conventionnel. Ce travail présente une solution éprouvée à un problème de contrôle réel documenté et fournit une analyse comparative directe des approches FFNN et RBFNN dans cette application exigeante.

Mots-clés : *Contrôle par backstepping, observateur à grand gain, réseau de neurones à propagation directe, réseau de neurones à fonctions de base radiales, réglage adaptatif du gain, système à double rotor MIMO, systèmes non linéaires, contrôle en temps réel.*

ملخص

تتناول هذه الأطروحة الفجوة القائمة بين النظرية والتطبيق في مجال التحكم بنظام المروحة الثنائية متعدد المداخل والمخارج (TRMS). تم في البداية تقييم مقاربة تقليدية تجمع بين متحكم الخطوة العكسية ومراقب ذي كسب عالٍ ثابت (HGO)، والتي أثبتت أداءً ناجحاً في المحاكاة لكلٍّ من ضبط نقطة العمل وتتبع المسار. عند الانتقال إلى المنصة الفيزيائية الفعلية، حافظ المتحكم على أداء ضبط نقطة العمل، مما يؤكد صلاحية التصميم في ظروف الحالة المستقرة. غير أن المتحكم ذاته أخفق كلياً أثناء تتبع المسار في الزمن الحقيقي، مما كشف عن قصور جوهري في بنية المراقب ذي الكسب الثابت، يتمثل في عجزه عن ضمان سرعة تقدير متجه الحالة وقمع ضوضاء أجهزة الاستشعار في آنٍ واحد تحت ظروف التشغيل الديناميكية. لمعالجة هذا الإخفاق المحدد، صُمِّمت بنية تحكم تكيفية تستخدم شبكات عصبية اصطناعية ذات تعلم آني لضبط كسب المراقب بصورة ذكية ومستمرة. طُوِّر مخططان مستقلان، يعتمد أولهما على شبكة عصبية ذات تغذية أمامية (FFNN) والثاني على شبكة عصبية ذات دوال قاعدية شعاعية (RBFNN)، لتعديل معاملات المراقب بناءً على خطأ المراقبة الآني. يُثبت التحقق التجريبي على منصة اختبار TRMS أن كلا المخططين المقترحين يحققان تتبعاً دقيقاً وقوياً في الظروف ذاتها التي أخفق فيها المتحكم التقليدي. يُقدِّم هذا العمل حلاً مُثبَتاً لمشكلة تحكم واقعية موثقة، فضلاً عن تحليل مقارن مباشر بين مقاربتَي FFNN وRBFNN في هذا التطبيق المتحدي.

الكلمات المفتاحية: التحكم بالخطوة العكسية، المراقب ذو الكسب العالي، الشبكة العصبية ذات التغذية الأمامية، الشبكة العصبية ذات الدوال القاعدية الشعاعية، ضبط الكسب التكيفي، نظام المروحة الثنائية، الأنظمة اللاخطية، التحكم في الزمن الحقيقي.

List of Publications

During the course of this doctoral research, the following articles have been published in peer-reviewed international scientific journals and presented at international conferences.

Peer-Reviewed International Journals

1. **A. Beloufa**, S. Tahraoui, A. Kacimi, H. Allouache, J.-J. Tiang, A. Azzouz, and M. H. Zaid, "Robust Backstepping Control of a Twin Rotor MIMO System via an RBF-Tuned High-Gain Observer," *Automation*, vol. 7, no. 2, p. 40, 2026.

2. **A. Beloufa**, S. Tahraoui, A. Kacimi, H. Allouach, J.-J. Tiang, and A. Azzouz, "Experimental Validation of Robust Backstepping Control for TRMS Using an Interval Type-2 Fuzzy Observer," *Eng*, vol. 7, no. 4, p. 171, 2026.

3. A. Kacimi, **A. Beloufa**, S. Tahraoui, A. Senoussaoui, M. H. Zaid, A. Azzouz, and J.-J. Tiang, "Robust Control of Twin-Rotor MIMO Systems Under Unmodeled Dynamics: Comparative Experimental Validation of Hybrid BSMC and Online QBHO Strategies," *Actuators*, vol. 15, no. 5, p. 236, 2026.

4. A. Kacimi, **A. Beloufa**, S. Tahraoui, and A. Senoussaoui, "Optimized TS Fuzzy Logic 2 Enhanced Observer for Twin Rotor Backstepping Control," *Electrotehnica, Electronica, Automatica*, vol. 73, no. 3, 2025.

International Conferences

5. **A. Beloufa**, S. Tahraoui, A. Kacimi, and R. Taleb, "PSO-Optimized High-Gain Observer-Based Backstepping Control for TRMS Trajectory Tracking," in *9th International Conference on Artificial Intelligence in Renewable Energetic Systems (IC-AIRES2025)*, Oct. 2025.

6. **A. Beloufa**, A. Kacimi, S. Tahraoui, and A. Djarir, "Backstepping Control Enhanced by Water Cycling Algorithms Applied on TRMS," in *1st International Seminar on Mechatronics Innovation Materials, Renewable Energy and Artificial Intelligence (ISMIMREAI'24)*, Tipaza, Nov. 2024.

Contents

List of Figures

List of Tables

Nomenclature

System Variables

Symbol	Description	Unit
ψ	Pitch angle	rad
$\dot{\psi}$	Pitch angular velocity	rad/s
φ	Yaw angle	rad
$\dot{\varphi}$	Yaw angular velocity	rad/s
τ_1	Main rotor torque state	N m
τ_2	Tail rotor torque state	N m
u_1	Main motor input voltage	V
u_2	Tail motor input voltage	V
$\mathbf{x}$	State vector $[\psi, \dot{\psi}, \varphi, \dot{\varphi}, \tau_1, \tau_2]^T$	–
$\hat{\mathbf{x}}$	Estimated state vector	–
$\mathbf{y}$	Output vector $[\psi, \varphi]^T$	rad

System Parameters

Symbol	Description	Unit
I_1	Moment of inertia, pitch axis	kg m^2
I_2	Moment of inertia, yaw axis	kg m^2
$a_1,\ b_1$	Main rotor thrust coefficients	–
$a_2,\ b_2$	Tail rotor thrust coefficients	–
M_g	Gravitational moment	N m
$B_{1\psi},\ B_{2\psi}$	Pitch friction coefficients	N m s/rad
$B_{1\varphi},\ B_{2\varphi}$	Yaw friction coefficients	N m s/rad
K_{gy}	Gyroscopic coupling gain	s/rad
K_c	Cross-coupling gain	–
$K_1,\ K_2$	Motor gains	–
$T_{10},\ T_{11}$	Main motor time constants	s
$T_{20},\ T_{21}$	Tail motor time constants	s

Control and Observer Design

Symbol	Description	Unit
e_i	i-th error coordinate	–
α_i	i-th stabilizing function (virtual control)	–
k_i	i-th backstepping gain	–
V_i	i-th Lyapunov function candidate	–
θ	Observer gain parameter	–
θ_p	Pitch observer gain	–
θ_y	Yaw observer gain	–
$\mathbf{f}(\mathbf{x})$	Nonlinear drift vector	–
$\mathbf{B}$	Input distribution matrix	–
$\mathbf{C}$	Output matrix	–

Neural Network Parameters

Symbol	Description	Unit
$\mathbf{W}$	Neural network weight matrix	–
η	Learning rate	–
$\sigma(\cdot)$	Activation function	–
$\mathbf{c}_j$	RBF center vector of j-th neuron	–
σ_j	RBF spread of j-th neuron	–

List of Abbreviations

AI Artificial Intelligence

ANN Artificial Neural Network

DC Direct Current

DE Differential Evolution

DOF Degree of Freedom

FFNN Feedforward Neural Network

GA Genetic Algorithm

HGO High-Gain Observer

HIL Hardware-in-the-Loop

IIR Infinite Impulse Response

ISE Integral of Squared Error

ITAE Integral of Time-weighted Absolute Error

LQG Linear-Quadratic Gaussian

LQR Linear-Quadratic Regulator

MIMO Multi-Input Multi-Output

MLP Multilayer Perceptron

MOGA Multi-Objective Genetic Algorithm

MPC Model Predictive Control

MSE Mean Squared Error

NN Neural Network

PCI Peripheral Component Interconnect

PID Proportional-Integral-Derivative

RBF Radial Basis Function

RBFNN Radial Basis Function Neural Network

ReLU Rectified Linear Unit

RL Reinforcement Learning

RMSE Root Mean Square Error

RNN Recurrent Neural Network

SCSI Small Computer System Interface

SGD Stochastic Gradient Descent

SMC Sliding Mode Control

SPAS Semi-Global Practical Asymptotic Stability

TRMS Twin Rotor MIMO System

UAV Unmanned Aerial Vehicle

General Introduction

The capacity to control complex physical systems with precision and reliability is one of the defining objectives of modern engineering. From aerospace vehicles to industrial robots, the systems at the heart of today's most critical applications share a common mathematical character: they are nonlinear, multi-variable, and subject to uncertainties that no theoretical model can fully anticipate. This fundamental tension between the clean abstractions of control theory and the imperfect reality of physical hardware constitutes the central challenge that this thesis sets out to address.

Context and Motivation

Within the broad field of nonlinear control, the combination of backstepping control with a High-Gain Observer (HGO) represents a theoretically well-grounded and widely adopted design paradigm. Backstepping provides a systematic, Lyapunov-based procedure for stabilizing nonlinear systems of arbitrary order, while the HGO offers a principled mechanism for reconstructing unmeasured internal states from noisy output measurements. Together, they form a closed-loop architecture whose stability properties are guaranteed under idealized conditions by the classical separation theorem.

In practice, however, idealized conditions rarely hold. Real hardware introduces sensor noise, unmodeled friction, actuator nonlinearities, and aerodynamic cross-coupling effects that are absent from any simulation environment. These factors do not merely degrade performance incrementally; they can cause an otherwise theoretically sound controller to fail catastrophically the moment it is deployed on physical equipment.

This work was motivated precisely by such a failure. During the preliminary phase of this research, a conventional backstepping controller paired with a fixed-gain HGO was applied to the Twin Rotor MIMO System (TRMS), a canonical laboratory benchmark that replicates the challenging flight dynamics of a helicopter. While the controller performed satisfactorily in both regulation tasks and simulation-based trajectory tracking, it failed completely when trajectory tracking was attempted on the real physical platform. This documented, reproducible failure constitutes the starting point of the entire research effort presented here.

Identified Problem

The root cause of the failure was traced to a fundamental limitation of the fixed-gain observer architecture. A High-Gain Observer requires large gain values to achieve fast and accu-

rate state estimation, but those same large gains amplify measurement noise on real hardware, injecting corrupted signals into the feedback loop and ultimately destabilizing the tracking performance. No single fixed gain value can simultaneously satisfy the conflicting requirements of fast convergence and effective noise rejection. This irreconcilable trade-off is the core problem this thesis addresses.

Proposed Approach

To resolve this conflict, this thesis proposes an adaptive observation strategy in which the HGO gains are no longer fixed quantities determined offline, but dynamic variables tuned continuously in real time by an intelligent, data-driven mechanism. The proposed architecture integrates online-learning neural networks directly into the observer loop: when estimation errors are large, the network commands elevated gains to drive rapid state convergence; as the system approaches the desired trajectory, it progressively reduces the gains to suppress noise amplification.

Two distinct neural network topologies are investigated and compared. The first is a Feedforward Neural Network (FFNN), selected for its universal approximation capability and its suitability for learning complex nonlinear mappings. The second is a Radial Basis Function Neural Network (RBFNN), chosen for its localized activation structure, which offers potentially faster adaptation and more interpretable gain trajectories. Both architectures are trained purely online, requiring no offline dataset and imposing computational overhead compatible with real-time execution.

Contributions

This work makes three primary contributions to the field of nonlinear control and intelligent observation:

1. **Experimental characterization of the simulation-to-reality gap:** A rigorous and reproducible documentation of the real-time tracking failure of a conventional Backstepping-HGO controller on the TRMS, establishing a clear evidence base for the practical limits of fixed-gain observer architectures.

2. **Design and validation of two AI-enhanced adaptive observers:** The introduction of FFNN-based and RBFNN-based online gain tuning mechanisms that restore robust trajectory tracking on the physical TRMS under the exact conditions where the conventional controller fails.

3. **Systematic comparative analysis:** A direct, quantitative comparison of the FFNN and RBFNN tuning schemes under identical experimental conditions, characterizing their respective adaptation dynamics, noise rejection capabilities, and practical trade-offs.

Thesis Outline

This manuscript is organized into five chapters, structured to guide the reader from the identification of the problem to its experimental resolution.

Chapter 1 introduces the research background, formally states the problem, and presents the three primary contributions of this work.

Chapter 2 reviews the relevant literature, covering the TRMS as a benchmark platform, the theoretical foundations of backstepping control and High-Gain Observers, and the application of neural networks to adaptive and observer-based control systems.

Chapter 3 establishes the complete mathematical model of the TRMS from first principles. Starting from Newton's laws applied to the pitch and yaw dynamics, it derives the full nonlinear state-space representation of the system and confirms, through open-loop simulation, the unstable nature of the uncontrolled plant.

Chapter 4 presents the complete control and observation architecture. It covers the design of the decentralized backstepping controller, the formulation of the High-Gain Observer and its stability analysis, and the full derivation of the two proposed AI-enhanced adaptive observer architectures based on the FFNN and RBFNN.

Chapter 5 presents the experimental results obtained on the physical TRMS testbed, providing a direct and quantitative comparison between the failed conventional controller and the two successful adaptive controllers across both regulation and trajectory tracking scenarios.

The thesis concludes with a General Conclusion that synthesizes the findings, discusses the limitations of the proposed approach, and outlines concrete directions for future research.

Chapter 1

Introduction

1.1 Background and Motivation

Controlling nonlinear Multi-Input Multi-Output (MIMO) systems remains one of the most persistent and consequential challenges in modern control engineering. Such systems are not merely academic constructs; they constitute the standard mathematical model for a broad class of safety-critical real-world applications spanning aerospace, robotics, and industrial process control. Their behaviour is governed by a particularly demanding combination of factors: complex internal nonlinear dynamics, strong cross-coupling between control channels, and significant parametric uncertainties that vary with operating conditions. Taken together, these characteristics render conventional linear control methods fundamentally inadequate, as they rely on approximations that are only valid in a narrow neighbourhood of a single operating point.

This inadequacy has driven sustained research interest in sophisticated nonlinear control strategies capable of guaranteeing both robustness and high performance across the full operating envelope of a system. Yet formulating such a strategy is only half the challenge; it must also be subjected to rigorous experimental validation. For this purpose, the research community routinely employs canonical benchmark platforms whose dynamics are sufficiently rich to expose the weaknesses of any proposed controller. A particularly well-established example is the Twin Rotor MIMO System (TRMS), a laboratory apparatus designed to replicate the flight dynamics of a helicopter at reduced scale. Despite the apparent simplicity of its concept two motor-driven rotors controlling the pitch and yaw axes of a free beam the TRMS presents formidable control challenges, including pronounced aerodynamic cross-coupling, actuator nonlinearities, and sensor noise, making it an exacting testbed for any advanced control algorithm.

1.2 Problem Statement

For a system exhibiting the level of complexity characteristic of the TRMS, a natural and theoretically well-motivated design choice is the combination of Backstepping control, which provides a systematic, Lyapunov-based framework for handling nonlinear dynamics, with a High-Gain Observer (HGO) for reconstructing the system's unmeasured internal states. The separation theorem lends additional theoretical support to this pairing: under ideal conditions, it permits the controller and observer to be designed independently while still guaranteeing the closed-loop stability of the combined system.

Preliminary investigations conducted during this research appeared to confirm the viability of this approach. In regulation tasks where the objective is simply to drive the system to a fixed setpoint and hold it there the conventional Backstepping-HGO controller performed satisfactorily, both in simulation and on the physical TRMS hardware. Performance in simulation also remained strong for the considerably more demanding trajectory tracking objective. The critical breakdown, however, occurred at the moment the tracking controller was deployed on the physical platform: the system consistently failed to follow the desired reference trajectory with acceptable accuracy, exhibiting a pronounced loss of robustness and a dramatic degradation in performance that had not been apparent in the simulation environment.

The root cause of this failure is a fundamental and well-documented trade-off inherent to the High-Gain Observer architecture. Fast convergence of the observer's state estimates requires large gain values; however, on a real system equipped with noisy sensors, those same large gains function as wideband amplifiers for measurement noise. The observer therefore injects a distorted, noise-corrupted signal into the feedback loop, ultimately undermining the controller's ability to execute precise tracking manoeuvres. This gap between the controller's clean-simulation success and its real-hardware failure a consequence of the irreconcilable demands placed on a fixed, manually tuned observer gain constitutes the central problem that this thesis sets out to address.

1.3 Proposed Solution: AI-Enhanced Adaptive Observation

The fundamental inadequacy of the conventional fixed-gain approach motivates the central contribution of this thesis: an adaptive observer architecture in which the High-Gain Observer's gain parameters are no longer static quantities determined offline, but dynamic variables tuned continuously in real-time by an intelligent mechanism driven by Artificial Intelligence.

The core insight behind this strategy is that the conflicting demands placed on a fixed gain high enough to ensure rapid state convergence, yet low enough to suppress noise amplification can be reconciled only if the gain is allowed to vary with operating conditions. Rather

than committing to a single compromise value, the proposed architecture employs an online learning mechanism that maps the real-time observation error to an appropriate set of gain parameters. When estimation errors are large, the tuner commands elevated gains to drive rapid convergence; as the system approaches steady-state tracking, it automatically reduces the gains to attenuate the corrupting influence of measurement noise. This self-regulating behaviour directly addresses the failure mode identified in the baseline experiments.

To investigate the most effective realisation of this concept, two distinct neural network topologies are designed and compared. The first employs a Feedforward Neural Network (FFNN), a universal function approximator whose global approximation capability makes it well-suited to learning the complex, non-linear mapping from observation error to optimal observer gain. The second employs a Radial Basis Function Neural Network (RBFNN), whose localised hidden-layer activations offer a structural alternative with potentially faster adaptation and more interpretable gain trajectories. Both networks are trained entirely online using the live observation error as the sole input signal, requiring no offline pre-training and imposing minimal computational overhead compatible with real-time execution on the TRMS testbed.

1.4 Contributions and Thesis Outline

This work makes three primary contributions to the field of nonlinear control and intelligent observation. First, it provides a rigorous experimental characterisation of the real-time tracking failure of a conventional Backstepping-HGO controller on the TRMS, establishing a clear and reproducible evidence base for the gap between simulation-validated and hardware-validated performance in fixed-gain observer designs. Second, it introduces and implements an AI-enhanced adaptive observer that resolves this robustness deficiency, enabling high-performance trajectory tracking on the physical platform where the conventional design fails. Third, it offers a systematic comparative analysis of the FFNN-based and RBFNN-based tuning schemes, characterising their respective adaptation dynamics, noise rejection capabilities, and performance trade-offs so as to provide actionable insights for practitioners selecting between the two architectures in similar applications.

Chapter 2

Literature Review

2.1 The TRMS: An Enduring Benchmark for Evolving Control Strategies

The Twin Rotor MIMO System (TRMS) holds a distinct and enduring position within the control engineering literature, serving as a canonical benchmark for validating novel theories against a complex physical reality. Its design, as detailed by its manufacturer, Feedback Instruments Ltd. [1], is specifically intended to emulate the challenging flight dynamics of a helicopter. The system's value as a research platform stems from its rich set of characteristics: it is an inherently unstable, nonlinear, multi-variable system defined by strong cross-coupling between its pitch and yaw axes [2]. For this reason, the TRMS has been the subject of extensive investigation across the full spectrum of control engineering sub-disciplines.

A foundational aspect of TRMS research involves mathematical modeling. While some works have successfully applied system identification techniques to derive data-driven models [3], the majority of the literature focuses on first-principles modeling to capture the underlying physics, which is crucial for the design of model-based controllers and observers.

The primary focus of the TRMS literature, however, is on the design and implementation of control strategies, showcasing a clear evolution from classical to modern intelligent paradigms. A vast body of work is dedicated to classical and optimal methods, such as tuned Proportional-Integral-Derivative (PID) controllers [4] and the Linear-Quadratic Regulator (LQR) [5, 6, 7]. To address the system's inherent nonlinearities, more advanced paradigms have been extensively explored. These include robust methods like H-infinity control [8] and a significant body of research into Model Predictive Control (MPC) [9, 10, 11].

Given its challenging dynamics, Sliding Mode Control (SMC) has emerged as a particularly prominent and successful strategy for the TRMS. The literature documents a rich variety of SMC implementations, from foundational designs [12] to more advanced architectures that incorporate state-varying gains [13, 14], synergetic control theory [15], and disturbance observers [16]. Furthermore, to enhance performance and adaptability, researchers have suc-

cessfully integrated these robust methods with intelligent systems. This has led to the development of neuro-adaptive SMC [17], neuro-fuzzy controllers [18], and even fault-tolerant adaptive fuzzy systems designed to handle real-world hardware failures [19].

2.2 Backstepping Control and the Challenge of State Observation

To address the documented complexities of the TRMS, this thesis employs a control architecture based on two powerful nonlinear techniques: Backstepping for the controller design and a High-Gain Observer for state estimation. This section reviews the literature pertinent to both methods, establishing their theoretical strengths and, critically, their practical limitations which motivate this research.

2.2.1 Lyapunov-Based Control for the TRMS: Backstepping and SMC

The Backstepping control technique is a systematic and recursive design methodology for a broad class of nonlinear systems, particularly those that can be represented in a ”strict-feedback” form. Its primary advantage, as established in foundational works on nonlinear and adaptive control [20, 21, 22], is its rigorous foundation in Lyapunov stability theory. This provides a constructive, step-by-step path to designing a control law with guaranteed stability properties.

This method belongs to a powerful family of Lyapunov-based robust controllers whose applicability has been demonstrated across a vast range of complex engineering systems. The technique’s versatility is evident in its successful application to electromechanical systems with unknown hysteresis [23], electrohydraulic servos [24], and high-performance power electronics for solar energy [25] and grid-connected converters [26, 27]. In the domain of autonomous vehicles, backstepping has been a cornerstone for controlling quadrotor-slung load systems [28, 29], underactuated autonomous surface vehicles [30], and for improving the maneuverability of ground vehicles through torque vectoring [31].

A closely related and more widely studied member of this family is Sliding Mode Control (SMC), which has also been extensively and successfully applied to the TRMS [15, 13, 32]. The literature showcases a clear trend of integrating these two robust paradigms. Researchers have designed adaptive backstepping controllers to handle system uncertainties [33, 34], developed prescribed-time backstepping for faster convergence [35], and combined Backstepping directly with SMC to leverage the strengths of both methodologies [36, 37]. Furthermore, these techniques have been integrated with intelligent systems, such as in the development of fuzzy backstepping SMC [38, 39] and the use of neural networks to create adaptive laws [40, 41, 42]. The success of these varied and sophisticated implementations provides a

strong and undeniable precedent for using Backstepping as the core control strategy in this thesis.

2.2.2 The High-Gain Observer: A Powerful Tool with a Critical Flaw

A significant practical challenge in controlling the TRMS is that not all of its states, such as the angular velocities, are directly measured. A state observer is therefore an essential component for implementing a state-feedback controller like backstepping. The High-Gain Observer (HGO), whose theoretical foundations were established in a series of seminal works in the early 1990s [43, 44, 45], is an attractive choice. As extensively detailed in the modern literature [46, 47], its design is conceptually straightforward and guarantees rapid, exponential convergence of the estimation error, provided the observer gain is chosen to be sufficiently high. This property is remarkably powerful; it facilitates a separation principle between the controller and observer designs, which is a foundational concept for nonlinear output-feedback control schemes [48].

The utility of the HGO has been demonstrated across a wide range of applications, from power systems and fuel cells [49, 50] to the robust trajectory estimation for multirotor UAVs [51]. However, a significant body of literature is dedicated to a critical and unavoidable trade-off that plagues the HGO in practice: its inherent sensitivity to measurement noise. The very mechanism that provides its powerful convergence the high gain is also responsible for amplifying any noise present in the sensor readings. This specific problem has been a central focus of research [52, 53]. The amplification of noise corrupts the state estimates fed to the controller, leading to a sharp degradation in performance and potentially exciting the ”peaking phenomenon,” a burst of high-amplitude estimation error that can destabilize a physical system [54, 55].

The challenge of mitigating these negative effects has spurred considerable innovation. Researchers have developed numerous strategies, including improved designs that are less sensitive to disturbances [56], methods that achieve stability with lower gain values [57], and filtered or saturated observers designed to explicitly manage peaking and noise [58]. Another important research direction, and one that is highly relevant to this thesis, involves making the observer’s gain adaptive rather than fixed. The concept of an adaptive HGO, where the gain can be adjusted online in response to system conditions, has been proposed as a promising solution to balance the competing demands of fast convergence and noise rejection [59, 60]. Despite these advances, the practical application of these ideas particularly using modern AI to solve the specific real-time tracking failure on a complex plant like the TRMS remains a compelling open problem. This well-documented conflict between the HGO’s theoretical power and its practical fragility is the central issue this research sets out to resolve.

2.3 Intelligent and AI-Based Paradigms in Control

The sheer difficulty of controlling the TRMS has naturally led the research community to look beyond conventional methods and explore the realm of intelligent or AI-based control. These paradigms are especially valuable for tackling systems where a precise mathematical model is hard to come by, or where significant uncertainties and nonlinearities are a core part of the problem.

One major strategy has been to use metaheuristic algorithms as sophisticated tuning tools for trusted conventional controllers. Particle Swarm Optimization (PSO), for example, stands out as a popular choice for tuning both PID controllers [61, 62, 63] and LQR architectures. Other evolutionary methods, such as Genetic Algorithms (GA) [64, 65] and Differential Evolution (DE) [66], have also been proven effective, with real-time hardware implementations consistently validating the performance of the optimized controllers [67, 68].

Another major line of inquiry involves fuzzy logic systems. The power of fuzzy logic is its ability to translate human expert knowledge into a mathematical framework. Research in this area shows a clear progression from foundational concepts to highly sophisticated applications, such as experimentally validating adaptive fuzzy controllers against wind disturbances [69], creating fault-tolerant control architectures [19], and even implementing Type-2 fuzzy backstepping sliding mode controllers in real-time [38].

A third, and increasingly prominent, paradigm is Reinforcement Learning (RL), where an agent learns an optimal control policy through direct trial-and-error interaction with its environment. Unlike the supervised methods central to this thesis, RL is powerful for problems where an explicit ”target” is unknown, but a high-level goal, encapsulated in a reward function, can be defined [70]. The literature has seen a surge in RL applications for complex robotics tasks [71], including helicopter and UAV control, where deep RL algorithms have shown significant promise [72]. These methods have been used for everything from aggressive, champion-level aerobatic maneuvers [73] to complex recovery behaviors like autorotation [74]. While a full RL implementation is beyond the scope of this work, its growing prominence marks it as a key area in the future of autonomous control.

Building on these paradigms, Artificial Neural Networks (NNs) have become a cornerstone of modern control due to their inherent capability as universal function approximators [75, 76]. The **Feedforward Neural Network (FFNN)**, in particular, has a long history of application in control, from early works on system identification and feedforward-feedback structures [77, 78] to more recent applications in complex thermal systems [79] and power electronics [80]. Their utility is often enhanced by pairing them with optimization algorithms like PSO [81] or Ant Colony Optimization [82].

The **Radial Basis Function Network (RBFNN)** has also been extensively studied for control, often favored for its fast learning characteristics. The literature is rich with examples of RBFNNs being used to develop adaptive sliding mode controllers for a wide variety of

dynamic systems [83, 31, 84, 85]. Beyond SMC, RBFNNs have been central to the design of adaptive controllers for robotic manipulators [86], satellite attitude control [87], and even fault-tolerant observers for wind turbines [88]. Crucially for this thesis, both FFNNs and RBFNNs have been successfully paired with backstepping control for a range of uncertain nonlinear systems [89, 90, 91, 92].

Specifically for the TRMS, NNs have been used to design adaptive controllers [93, 94] and, in a series of pioneering works, to construct novel state observers from the ground up [95, 96]. However, these studies focused on using NNs to *replace* the traditional observer entirely. In parallel, the concept of making a High-Gain Observer adaptive using classical, non-NN adaptation laws has also been explored, which sets the stage for the specific research gap addressed by this work.

2.4 Synthesis and Identified Research Gap

The literature thus presents a clear narrative. The TRMS is a well-established benchmark for which powerful nonlinear controllers like Backstepping are a suitable and well-researched choice [38, 36]. The practical implementation of these controllers, however, is often compromised by the inherent noise sensitivity of the High-Gain Observers upon which they depend, a challenge noted in foundational HGO literature [52, 54].

While advanced solutions have been explored, a distinct gap emerges from a synthesis of the existing work. On one hand, researchers have successfully designed adaptive HGOs using classical adaptation laws. On the other hand, neural networks have been used to design entirely new types of observers for the system [95]. A comprehensive review of the literature indicates that these two distinct research thrusts have not yet been integrated. Specifically, there appears to be no existing work that applies modern neural network architectures, such as the widely used FFNN or RBFNN, to the specific problem of **robustifying a conventional HGO by adaptively tuning its gain online** to solve the documented real-time tracking failure.

Therefore, this thesis aims to fill this identified gap. It proposes and validates an AI-enhanced adaptive observer where an online-learning neural network is used to intelligently modulate the HGO gain, thereby solving its practical limitations and enabling high-performance, robust trajectory tracking on the physical TRMS.

Chapter 3

System Modeling of the Twin Rotor MIMO System

3.1 Introduction

Before any control strategy can be designed, a reliable mathematical description of the physical plant must be established. This chapter is dedicated entirely to that task. A first-principles model of the Twin Rotor MIMO System (TRMS) is derived, a laboratory-scale aerodynamic platform whose dynamics are characterized by strong nonlinearity, inherent instability, and significant coupling between its two axes of motion [1].

The derivation proceeds from a physical description of the hardware to a complete set of nonlinear differential equations governing the system behavior. These equations are subsequently cast in a compact state-space form that serves as the mathematical foundation for all design work presented in Chapter 4. An open-loop simulation is included to confirm that the model captures the essential unstable behavior of the real plant, thereby establishing the necessity of a closed-loop feedback strategy.

3.2 System Overview and Mathematical Modeling

3.2.1 Hardware Description

The TRMS is an educational prototype manufactured by Feedback Instruments Ltd., designed specifically for control experimentation on a system that emulates the static flight dynamics of a helicopter [1]. The system is nonlinear, multi-variable, and aerodynamic in nature, with two actuated inputs and two measurable outputs connected by strong inter-axis coupling. Control is implemented in MATLAB, and signal acquisition is managed through a dedicated data acquisition card. The TRMS continues to serve as an active benchmark platform in the control literature, as evidenced by recent experimental studies [97].

Composition of the TRMS

The main structural and functional components of the TRMS are as follows:

- **Beam and Pivot:** A central rigid beam connected to a support tower through a universal joint, permitting free rotation in both the vertical plane (pitch) and the horizontal plane (yaw).
- **Rotors:** Two DC motor-driven propellers mounted at opposite ends of the beam. The main rotor governs elevation and produces the pitch motion; the tail rotor governs the horizontal position and produces the yaw motion.
- **Counterweight:** A mass attached to a rod at the pivot point, provided to reduce mechanical vibrations during operation.
- **Base and Electronics:** A base unit housing the signal conditioning circuits, synchronization logic, filtering electronics, and motor power switches.
- **Position Sensors:** Incremental optical encoders mounted at the pivot, providing direct measurement of the horizontal and vertical angular positions.

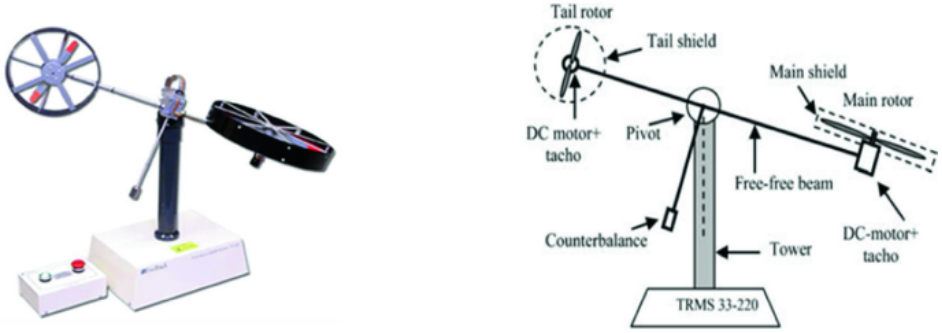

Figure 3.1: Composition of the TRMS experimental setup showing main components.

3.2.2 Operating Principle

The angle of attack of the propeller blades on the TRMS is fixed. Aerodynamic force is therefore controlled exclusively by varying the rotational speed of each rotor through the supply voltage applied to the corresponding DC motor. A change in voltage produces a change in rotational speed, which alters the aerodynamic thrust and consequently repositions the beam. The thrust forces generated by the main and tail rotors are denoted F_1 and F_2, respectively.

A defining feature of this platform is the strong aerodynamic and inertial coupling between the two axes. The rotation of the main rotor generates a reaction torque that influences the yaw dynamics, and gyroscopic effects couple the pitch and yaw motions under dynamic conditions. This coupling is what makes the TRMS particularly challenging to control and representative of real rotorcraft behavior [1].

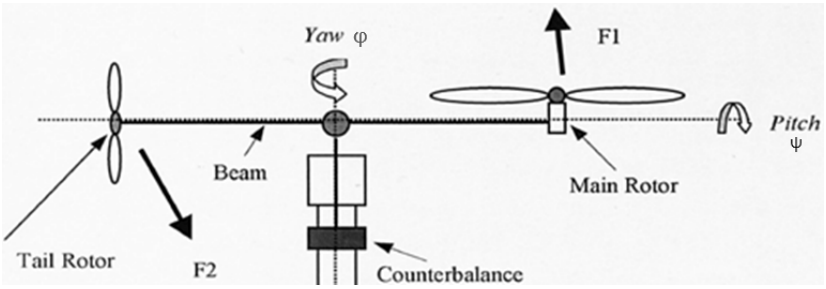

Figure 3.2: Thrust generated by the rotors.

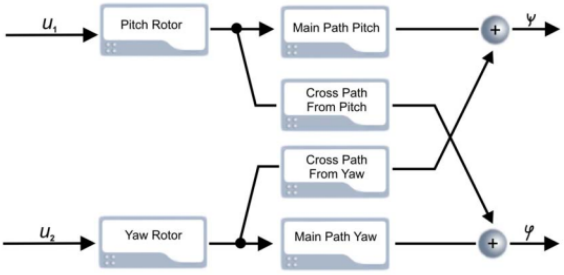

Figure 3.3: Schematic of the direct and cross-coupling channels of the TRMS.

3.2.3 Mathematical Modeling Derivation

The equations of motion are derived from Newton's second law applied independently to each axis of rotation, supplemented by first-order motor dynamics relating the actuator input voltage to the rotor torque state. The derivation follows the formulation established in the manufacturer documentation [1] and adopted throughout the TRMS control literature.

Pitch Dynamics

The motion of the beam in the vertical plane is driven by the main rotor and characterized by the pitch angle ψ. Applying Newton's second law to the rotational degree of freedom yields:

$$M_v = I_1 \frac{d^2\psi}{dt^2} \tag{3.1}$$

where I_1 is the moment of inertia of the beam assembly about the pitch axis and M_v is the net moment of all forces acting in the vertical plane:

$$M_v = M_1 - M_{FG} - M_{B\psi} - M_G \tag{3.2}$$

The four contributing moment terms are defined as follows:

- $M_1 = a_1\tau_1^2 + b_1\tau_1$: the aerodynamic thrust moment of the main rotor, modeled as a second-order polynomial in the rotor torque state τ_1.

- $M_{FG} = M_g \sin(\psi)$: the gravitational restoring moment acting on the beam.
- $M_{B\psi} = B_{1\psi}\dot{\psi} + B_{2\psi}\,\text{sign}(\dot{\psi})$: the viscous and Coulomb friction moment opposing angular velocity.
- $M_G = K_{gy} M_1 \dot{\varphi} \cos(\psi)$: the gyroscopic moment arising from the interaction between the main rotor thrust and the yaw rate, which constitutes the primary coupling term between the two axes.

The dynamic relationship between the control input voltage u_1 and the torque state τ_1 is described by a first-order motor model. In the Laplace domain:

$$\frac{\tau_1(s)}{u_1(s)} = \frac{k_1}{T_{11}s + T_{10}} \tag{3.3}$$

which corresponds to the following first-order ordinary differential equation in the time domain:

$$\dot{\tau}_1 = -\frac{T_{10}}{T_{11}}\tau_1 + \frac{k_1}{T_{11}}u_1 \tag{3.4}$$

Yaw Dynamics

The motion of the beam in the horizontal plane is driven by the tail rotor and characterized by the yaw angle φ. The governing equation is:

$$M_h = I_2 \frac{d^2\varphi}{dt^2} \tag{3.5}$$

where I_2 is the moment of inertia about the yaw axis and M_h is the net moment in the horizontal plane:

$$M_h = M_2 - M_{B\varphi} - M_R \tag{3.6}$$

The three contributing terms are:

- $M_2 = a_2\tau_2^2 + b_2\tau_2$: the aerodynamic thrust moment of the tail rotor, modeled as a second-order polynomial in the rotor torque state τ_2.
- $M_{B\varphi} = B_{1\varphi}\dot{\varphi} + B_{2\varphi}\,\text{sign}(\dot{\varphi})$: the viscous and Coulomb friction moment in the horizontal plane.
- $M_R \approx 1.75\, k_c\,(a_1\tau_1^2 + b_1\tau_1)$: the reaction torque exerted on the yaw axis by the spinning main rotor. The constant factor 1.75 results from evaluating the coupling transfer function at steady state.

The tail motor dynamics take the same first-order form as those of the main motor:

$$\frac{\tau_2(s)}{u_2(s)} = \frac{k_2}{T_{21}s + T_{20}} \tag{3.7}$$

$$\dot{\tau}_2 = -\frac{T_{20}}{T_{21}}\tau_2 + \frac{k_2}{T_{21}}u_2 \tag{3.8}$$

Complete Nonlinear Model

Combining the pitch dynamics (3.1), the yaw dynamics (3.5), and the two motor equations (3.4) and (3.8), the TRMS is fully described by a set of six coupled first-order nonlinear differential equations:

$$\begin{cases} \dfrac{d\psi}{dt} = \dot{\psi} \\ \dfrac{d\dot{\psi}}{dt} = \dfrac{1}{I_1}\Big(a_1\tau_1^2 + b_1\tau_1 - M_g \sin\psi - B_{1\psi}\dot{\psi} - B_{2\psi}\,\mathrm{sgn}(\psi) - K_{gy}\,\dot{\phi}\cos\psi\,(a_1\tau_1^2 + b_1\tau_1)\Big) \\ \dfrac{d\phi}{dt} = \dot{\phi} \\ \dfrac{d\dot{\phi}}{dt} = \dfrac{1}{I_2}\Big(a_2\tau_2^2 + b_2\tau_2 - B_{1\phi}\dot{\phi} - B_{2\phi}\,\mathrm{sgn}(\phi) - 1.75\,K_c\,(a_1\tau_1^2 + b_1\tau_1)\Big) \\ \dfrac{d\tau_1}{dt} = -\dfrac{T_{10}}{T_{11}}\tau_1 + \dfrac{K_1}{T_{11}}\,u_1 \\ \dfrac{d\tau_2}{dt} = -\dfrac{T_{20}}{T_{21}}\tau_2 + \dfrac{K_2}{T_{21}}\,u_2 \end{cases} \tag{3.9}$$

To facilitate state-space analysis and controller design, the state vector is introduced as:

$$\mathbf{x} = [x_1,\, x_2,\, x_3,\, x_4,\, x_5,\, x_6]^T = [\psi,\, \dot{\psi},\, \phi,\, \dot{\phi},\, \tau_1,\, \tau_2]^T \tag{3.10}$$

together with the control input vector $\mathbf{u} = [u_1,\, u_2]^T$. Under this substitution, the complete nonlinear model becomes (3.11):

$$
\begin{cases}
\dot{x}_1 = x_2 \\
\dot{x}_2 = \dfrac{1}{I_1}\Big(a_1 x_5^2 + b_1 x_5 - M_g \sin(x_1) - B_{1\psi}\, x_2 - B_{2\psi}\, \mathrm{sgn}(x_1) - K_{gy}\, x_4 \cos(x_1)\,(a_1 x_5^2 + b_1 x_5)\Big) \\
\dot{x}_3 = x_4 \\
\dot{x}_4 = \dfrac{1}{I_2}\Big(a_2 x_6^2 + b_2 x_6 - B_{1\phi}\, x_4 - B_{2\phi}\, \mathrm{sgn}(x_3) - 1.75\, K_c\,(a_1 x_5^2 + b_1 x_5)\Big) \\
\dot{x}_5 = -\dfrac{T_{10}}{T_{11}}\, x_5 + \dfrac{K_1}{T_{11}}\, u_1 \\
\dot{x}_6 = -\dfrac{T_{20}}{T_{21}}\, x_6 + \dfrac{K_2}{T_{21}}\, u_2
\end{cases}
\tag{3.11}
$$

Three structural observations follow directly from this model. First, the pitch acceleration $\dot{x}_2$ contains the gyroscopic coupling term $-K_{gy}\, x_4 \cos(x_1)(a_1 x_5^2 + b_1 x_5)$, which introduces a product of yaw velocity and main rotor thrust into the pitch dynamics. Second, the yaw acceleration $\dot{x}_4$ includes the reaction torque $-1.75\, K_c(a_1 x_5^2 + b_1 x_5)$ from the main rotor, making the yaw dynamics dependent on the pitch actuator state. Third, only x_1 and x_3 are directly measurable through the onboard encoders; the remaining four states must be estimated by a state observer, as developed in Chapter 4.

3.2.4 System Parameters

The model parameters were determined experimentally by the manufacturer and are reproduced in Table 3.1. These values are used without modification throughout all simulations and real-time experiments reported in this thesis.

Table 3.1: TRMS model parameters.

Parameter	**Value**
I_1	6.8×10^{-2} kg m^2
I_2	2.0×10^{-2} kg m^2
a_1	0.0135
b_1	0.0924
a_2	0.02
b_2	0.09
M_g	0.32 N m
$B_{1\psi},\ B_{1\phi}$	6.0×10^{-6} N m s/rad
$B_{2\psi},\ B_{2\phi}$	1.0×10^{-6} N m s/rad
K_{gy} (gyroscopic gain)	0.05 s/rad
$K_1,\ K_2$ (motor gains)	$1.1,\ 0.8$
$T_{10},\ T_{11}$ (main motor)	$1,\ 1.1$
$T_{20},\ T_{21}$ (tail motor)	$1,\ 1$
K_c (cross-coupling gain)	-0.2

3.2.5 State-Space Representation

For the purpose of observer design and stability analysis, the model (3.11) is expressed in the compact affine state-space form:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{B}\,\mathbf{u} \tag{3.12}$$

where the nonlinear drift vector $\mathbf{f}(\mathbf{x})$ collects all state-dependent terms:

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} x_2 \\ \dfrac{1}{I_1}\Big(a_1 x_5^2 + b_1 x_5 - M_g \sin(x_1) - B_{1\psi}\, x_2 - B_{2\psi}\, \mathrm{sgn}(x_1) - K_{gy}\, x_4 \cos(x_1)(a_1 x_5^2 + b_1 x_5)\Big) \\ x_4 \\ \dfrac{1}{I_2}\Big(a_2 x_6^2 + b_2 x_6 - B_{1\varphi}\, x_4 - B_{2\varphi}\, \mathrm{sgn}(x_3) - 1.75\, K_c\, (a_1 x_5^2 + b_1 x_5)\Big) \\ -\dfrac{T_{10}}{T_{11}}\, x_5 \\ -\dfrac{T_{20}}{T_{21}}\, x_6 \end{bmatrix} \tag{3.13}$$

and the input matrix **B** distributes the control voltages to the motor torque states:

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ \dfrac{K_1}{T_{11}} & 0 \\ 0 & \dfrac{K_2}{T_{21}} \end{bmatrix} \tag{3.14}$$

The TRMS is equipped with two incremental optical encoders that measure the pitch angle $\psi = x_1$ and the yaw angle $\varphi = x_3$ directly. The measurement equation is:

$$\mathbf{y} = \mathbf{C}\,\mathbf{x} \tag{3.15}$$

where $\mathbf{y} = [x_1,\ x_3]^T$ and the output matrix is:

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{3.16}$$

The structure of **C** confirms that four of the six states, namely $x_2 = \dot{\psi}$, $x_4 = \dot{\varphi}$, $x_5 = \tau_1$, and $x_6 = \tau_2$, are inaccessible to direct measurement. Their reconstruction through a state observer is therefore not a design choice but a strict requirement for any state-feedback control law applied to this system.

3.2.6 Open-Loop Simulation and Analysis

To characterize the inherent dynamics of the uncontrolled plant, the model is subjected to a step input of 0.5 rad for both the pitch and yaw channels. The resulting open-loop response is shown in Figure 3.4.

The pitch output exhibits a bounded but poorly damped transient that does not converge to the commanded setpoint, confirming the absence of any self-regulating behavior in this axis. The yaw output diverges without bound, which provides direct experimental evidence that the open-loop TRMS is unstable. These observations establish that a closed-loop feedback control strategy is not merely advantageous but strictly necessary for safe and reliable operation of the system.

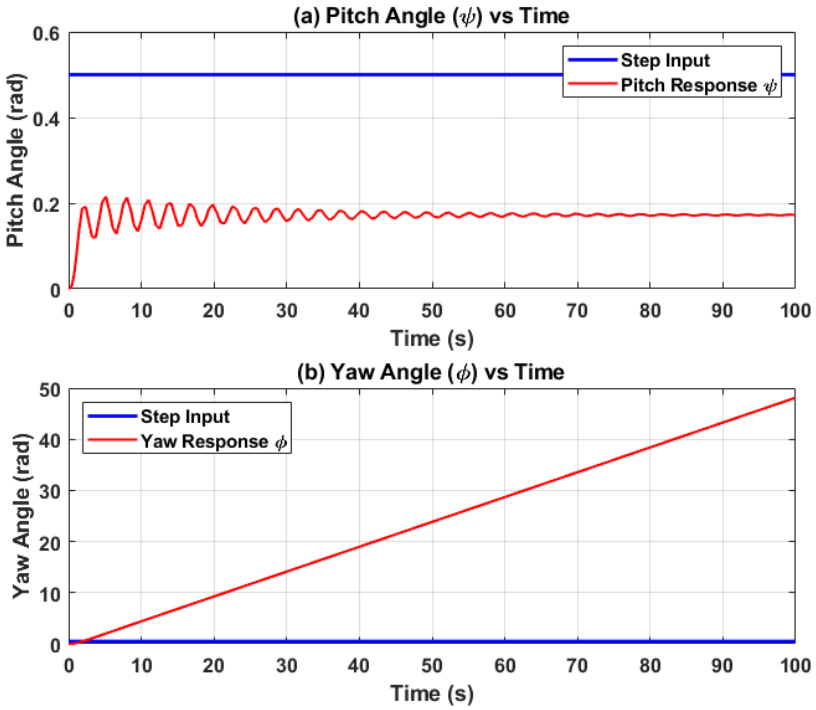

Figure 3.4: TRMS open-loop step response.

3.3 Chapter Summary

This chapter has established a complete, physics-based mathematical model of the TRMS. Starting from Newton's second law applied to the pitch and yaw axes, and accounting for aerodynamic thrust, gravitational loading, viscous and Coulomb friction, gyroscopic coupling, and cross-axis reaction torque, a system of six coupled first-order nonlinear differential equations was derived. This model was subsequently written in the compact affine state-space form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{B}\mathbf{u}$, with the output equation $\mathbf{y} = \mathbf{C}\mathbf{x}$ confirming that only the pitch angle x_1 and the yaw angle x_3 are directly measurable. The four remaining states, namely the angular velocities and the motor torques, are inaccessible to direct measurement and must be reconstructed by a state observer.

The open-loop step response demonstrated that the uncontrolled system is unstable, with the yaw output diverging under constant input. This result confirms that a feedback control strategy is not merely beneficial but strictly necessary for safe operation of the TRMS.

The model and parameters established in this chapter form the complete foundation upon which the backstepping controller, the High-Gain Observer, and the AI-based adaptive tuning schemes are designed in Chapter 4.

Chapter 4

Controller and Observer Design

4.1 Introduction

Building on the mathematical model established in Chapter 3, this chapter addresses the central engineering problem of this thesis: how to control a system whose full state vector cannot be directly measured, and how to do so robustly under real-world operating conditions.

The chapter is organised around three successive design layers, each one building upon the results of the previous.

The first layer is the **backstepping controller**. A decentralised control strategy is adopted, treating the pitch and yaw axes as two independent strict-feedback subsystems. For each axis, a three-stage Lyapunov-based derivation yields a control law with formally guaranteed asymptotic stability. Both a regulation controller, for constant setpoint holding, and a trajectory tracking controller, for time-varying references, are derived in full.

The second layer is the **High-Gain Observer (HGO)**. Since the backstepping law requires the full state vector and four of the six states are unmeasured, a state observer is an indispensable component of the architecture. The HGO is designed to estimate these states from the available encoder measurements. Its theoretical properties and its specific formulation for the TRMS are presented, followed by a rigorous stability analysis of the integrated observer-controller system based on the nonlinear separation principle.

The third layer is the **AI-enhanced adaptive observer**. The fixed-gain HGO suffers from a fundamental conflict between estimation speed and noise rejection, a limitation that leads to complete failure during real-time trajectory tracking, as documented in Chapter 5. To resolve this, two neural network-based strategies are designed to tune the observer gain online: one based on a Feedforward Neural Network (FFNN) and one based on a Radial Basis Function Neural Network (RBFNN). Both architectures, their training algorithms, and their integration into the control loop are detailed in full.

The chapter concludes with the performance evaluation framework used to compare all three architectures in Chapter 5.

4.2 Theoretical Background: Backstepping Control

Backstepping is a recursive, Lyapunov-based design methodology applicable to a broad class of nonlinear systems that can be expressed in strict-feedback form. First proposed in the early 1990s and extensively studied by Krstić, Kanellakopoulos, and Kokotović [22], the method has since been applied to a wide range of complex nonlinear systems [98, 99]. Its defining property is a constructive, step-by-step procedure in which a composite Lyapunov function is assembled recursively, yielding a control law whose stability guarantees are embedded in the design by construction rather than verified after the fact.

The method applies to systems in strict-feedback form, defined as:

$$\begin{cases} \dot{x}_1 = f_1(x_1, x_2) \\ \dot{x}_2 = f_2(x_1, x_2) + g_1(x_1, x_2)x_3 \\ \quad \vdots \\ \dot{x}_n = f_n(x_1, \ldots, x_n) + g_1(x_1, \ldots, x_n)w_n \end{cases} \tag{4.1}$$

The principle of the method is illustrated through the following example, adapted from Zhou and Wen [100].

4.2.1 Backstepping Design: An Illustrative Example

Consider the third-order system:

$$\begin{cases} \dot{x}_1 = x_2 + x_1^2 \\ \dot{x}_2 = x_3 + x_2^2 \\ \dot{x}_3 = u \end{cases} \tag{4.2}$$

where x_1, x_2, x_3 are the state variables and u is the control input. The objective is to synthesise a state-feedback control law that asymptotically stabilises the origin.

Step 1

The first error coordinate is defined as $z_1 = x_1$, whose dynamics are given by:

$$\dot{z}_1 = x_2 + x_1^2 \tag{4.3}$$

At this stage, x_2 is regarded as a virtual control input. A second error variable is introduced as the deviation of x_2 from the first stabilising function α_1:

$$z_2 = x_2 - \alpha_1 \tag{4.4}$$

so that:

$$\dot{z}_1 = \alpha_1 + x_1^2 + z_2 \tag{4.5}$$

The first Lyapunov function candidate is selected as $V_1 = \frac{1}{2}z_1^2$, whose time derivative is:

$$\dot{V}_1 = z_1(\alpha_1 + x_1^2) + z_1 z_2 \tag{4.6}$$

The virtual control law α_1 is chosen to render $\dot{V}_1$ negative definite in z_1 whenever $z_2 = 0$:

$$\begin{aligned} \alpha_1 &= -c_1 z_1 - x_1^2 \\ \dot{\alpha}_1 &= -(c_1 + 2x_1)(x_2 + x_1^2) \end{aligned} \tag{4.7}$$

with $c_1 > 0$. Substituting this choice reduces the derivative of V_1 to:

$$\dot{V}_1 = -c_1 z_1^2 + z_1 z_2 \tag{4.8}$$

It is clear that when $z_2 = 0$, the expression $\dot{V}_1 = -c_1 z_1^2$ is negative definite, thereby guaranteeing the asymptotic convergence of z_1 to zero. The residual cross-term $z_1 z_2$ is addressed in the following step.

Step 2

The objective of this step is to stabilise the error z_2. The error dynamics for $z_2 = x_2 - \alpha_1$ are:

$$\dot{z}_2 = x_3 + x_2^2 + (c_1 + 2x_1)(x_2 + x_1^2) \tag{4.9}$$

A third error variable is defined as $z_3 = x_3 - \alpha_2$, so that:

$$\dot{z}_2 = z_3 + \alpha_2 + x_2^2 + (c_1 + 2x_1)(x_2 + x_1^2) \tag{4.10}$$

The augmented Lyapunov function $V_2 = V_1 + \frac{1}{2}z_2^2$ has time derivative:

$$\dot{V}_2 = -c_1 z_1^2 + z_2\left(\alpha_2 + z_1 + x_2^2 + (c_1 + 2x_1)(x_2 + x_1^2)\right) + z_2 z_3 \tag{4.11}$$

The second virtual control law is selected to cancel the known terms and introduce damping on z_2:

$$\alpha_2 = -z_1 - c_2 z_2 - x_2^2 - (c_1 + 2x_1)(x_2 + x_1^2), \qquad c_2 > 0 \tag{4.12}$$

which yields:

$$\dot{V}_2 = -c_1 z_1^2 - c_2 z_2^2 + z_2 z_3 \tag{4.13}$$

When $z_3 = 0$, both z_1 and z_2 converge asymptotically to zero, as required. The residual term $z_2 z_3$ motivates the final design step.

Step 3

The objective of this final step is to synthesise the actual control input u such that all three error coordinates converge to zero. Differentiating $z_3 = x_3 - \alpha_2$ and substituting $\dot{x}_3 = u$ gives:

$$\dot{z}_3 = u - \frac{\partial \alpha_2}{\partial x_1}(x_2 + x_1^2) - \frac{\partial \alpha_2}{\partial x_2}(x_3 + x_2^2) \tag{4.14}$$

The final composite Lyapunov function $V_3 = V_2 + \frac{1}{2}z_3^2$ has time derivative:

$$\dot{V}_3 = -\sum_{i=1}^{2} c_i z_i^2 + z_3 \left(u + z_2 - \frac{\partial \alpha_2}{\partial x_1}(x_2 + x_1^2) - \frac{\partial \alpha_2}{\partial x_2}(x_3 + x_2^2) \right) \tag{4.15}$$

The actual control law is designed to render $\dot{V}_3$ negative definite:

$$u = -z_2 - c_3 z_3 + \frac{\partial \alpha_2}{\partial x_1}(x_2 + x_1^2) + \frac{\partial \alpha_2}{\partial x_2}(x_3 + x_2^2), \qquad c_3 > 0 \tag{4.16}$$

which gives:

$$\dot{V}_3 = -\sum_{i=1}^{3} c_i z_i^2 < 0 \tag{4.17}$$

By LaSalle's invariance principle, global uniform asymptotic stability of the origin is established, with $z_1, z_2, z_3 \to 0$ as $t \to \infty$. Since $x_1 = z_1$, and since $x_2 = z_2 + \alpha_1$ and $x_3 = z_3 + \alpha_2$, the boundedness of the full state vector follows directly.

4.3 Backstepping Control Design for the TRMS

4.3.1 Control Strategy

A decentralised control strategy is adopted in this work. The pitch axis and the yaw axis are each treated as an independent third-order strict-feedback subsystem, and a dedicated backstepping controller is synthesised for each one. The two controllers are then applied simultaneously to the full nonlinear TRMS model in simulation.

The complete TRMS dynamics do not possess a global strict-feedback form, because the gyroscopic coupling term in the pitch equation depends on the yaw rate and the cross-coupling term in the yaw equation depends on the main-rotor thrust. By neglecting these coupling terms at the design stage, each axis reduces to a three-state strict-feedback chain that satisfies the classical backstepping assumptions. The robustness of the closed-loop system in the presence of these neglected terms is evaluated as a dedicated test in Chapter 5.

Two operating modes are addressed for each axis:

- **Regulation:** The reference is a constant setpoint. Manual disturbances are injected

at steady state to evaluate disturbance rejection and robustness with respect to the neglected coupling dynamics.

- **Trajectory tracking:** The reference is a smooth time-varying signal. This mode assesses transient performance and sensitivity to model mismatch.

For each axis, the regulation controller and the trajectory tracking controller are derived separately and in full, so that the structural difference between the two formulations is made explicit at every step of the derivation.

4.4 Pitch Subsystem Controller Design

4.4.1 Simplified Model for Control Synthesis

The pitch subsystem governs the vertical motion of the TRMS and is driven by the main rotor. The full three-state equations of motion are:

$$\begin{cases} \dfrac{d\psi}{dt} = \dot{\psi} \\ \dfrac{d\dot{\psi}}{dt} = \dfrac{1}{I_1}\Big(a_1\tau_1^2 + b_1\tau_1 - M_g \sin\psi - B_{1\psi}\dot{\psi} - B_{2\psi}\,\mathrm{sgn}(\psi) - K_{gy}\,\dot{\phi}\cos\psi\,(a_1\tau_1^2 + b_1\tau_1)\Big) \\ \dfrac{d\tau_1}{dt} = -\dfrac{T_{10}}{T_{11}}\,\tau_1 + \dfrac{K_1}{T_{11}}\,U_1 \end{cases} \tag{4.18}$$

where ψ is the pitch angle and τ_1 is the main-rotor torque state.

To obtain a strict-feedback form suitable for the backstepping procedure, two simplifications are introduced. First, the gyroscopic coupling term involving $\dot{\varphi}$ is neglected, consistent with the decentralized design strategy described above. Second, the Coulomb friction term $B_{2\psi}\,\mathrm{sgn}(\psi)$ is approximated by the linear viscous term $B_{2\psi}\psi$ for the purpose of control synthesis. Introducing the state variables $x_1 = \psi$, $x_2 = \dot{\psi}$, and $x_3 = \tau_1$, the simplified model used for control synthesis is:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = \dfrac{1}{I_1}\Big(a_1 x_3^2 + b_1 x_3 - M_g \sin x_1 - B_{1\psi}\,x_2 - B_{2\psi}\,x_1\Big) \\ \dot{x}_3 = -\dfrac{T_{10}}{T_{11}}\,x_3 + \dfrac{K_1}{T_{11}}\,U_1 \end{cases} \tag{4.19}$$

This system is in strict-feedback form and therefore satisfies all the assumptions required for the backstepping procedure. For notational convenience, the full expression for $\dot{x}_2$ is denoted

dx_2 throughout the derivation:

$$dx_2 \triangleq \dot{x}_2 = \frac{1}{I_1}\Big(a_1 x_3^2 + b_1 x_3 - M_g \sin x_1 - B_{1\psi}\, x_2 - B_{2\psi}\, x_1\Big) \tag{4.20}$$

4.4.2 Pitch Regulation Controller

In regulation mode, the objective is to drive x_1 to a constant setpoint $x_{1d} = \text{const}$, so that $\dot{x}_{1d} = \ddot{x}_{1d} = x_{1d}^{(3)} = 0$ identically. The backstepping procedure is applied in three successive stages, each introducing one error coordinate and one stabilising function.

Stage 1: Position Error

The first step begins with the position tracking error:

$$e_1 = x_1 - x_{1d} \tag{4.21}$$

Since x_{1d} is constant, the time derivative of e_1 reduces to:

$$\dot{e}_1 = \dot{x}_1 = x_2 \tag{4.22}$$

Treating x_2 as a virtual control input, the first stabilising function is chosen as:

$$\alpha_1 = -k_1\, e_1, \qquad k_1 > 0 \tag{4.23}$$

A second error variable is then defined as the deviation of x_2 from this stabilising function:

$$e_2 = x_2 - \alpha_1 \tag{4.24}$$

Substituting (4.23) into (4.22) gives the closed-loop dynamics of the first error coordinate:

$$\dot{e}_1 = -k_1\, e_1 + e_2 \tag{4.25}$$

The first Lyapunov function candidate and its time derivative are:

$$V_1 = \frac{1}{2}\, e_1^2, \qquad \dot{V}_1 = -k_1\, e_1^2 + e_1\, e_2 \tag{4.26}$$

When $e_2 = 0$, the derivative $\dot{V}_1 = -k_1 e_1^2$ is negative definite, ensuring the asymptotic convergence of e_1 to zero. The residual cross-term $e_1 e_2$ will be addressed in the next stage.

Stage 2: Velocity Error

The objective of this stage is to synthesise the second stabilising function α_2 such that e_2 converges to zero. Differentiating $e_2 = x_2 - \alpha_1$ and noting that $\dot{\alpha}_1 = -k_1 \dot{e}_1 = -k_1 x_2$

yields:

$$\dot{e}_2 = \frac{1}{I_1}\Big(a_1 x_3^2 + b_1 x_3 - M_g \sin x_1 - B_{1\psi}\, x_2 - B_{2\psi}\, x_1\Big) + k_1\, x_2 \tag{4.27}$$

To facilitate the subsequent design, the torque-dependent contribution to the angular acceleration is isolated through the auxiliary variable:

$$E = \frac{1}{I_1}\big(a_1 x_3^2 + b_1 x_3 - B_{1\psi}\, x_2\big) \tag{4.28}$$

With this notation, equation (4.27) becomes:

$$\dot{e}_2 = E - \frac{M_g \sin x_1 + B_{2\psi}\, x_1}{I_1} + k_1\, x_2 \tag{4.29}$$

The variable E is treated as a new virtual control input at this stage. A third error variable $e_3 = E - \alpha_2$ is introduced, so that $E = e_3 + \alpha_2$, and substituting gives:

$$\dot{e}_2 = e_3 + \alpha_2 - \frac{M_g \sin x_1 + B_{2\psi}\, x_1}{I_1} + k_1\, x_2 \tag{4.30}$$

The second virtual control law is chosen to cancel all known terms and introduce damping on e_2:

$$\alpha_2 = -k_2\, e_2 - e_1 + \frac{M_g \sin x_1 + B_{2\psi}\, x_1}{I_1} - k_1\, x_2, \qquad k_2 > 0 \tag{4.31}$$

Substituting (4.31) into (4.30) simplifies the velocity error dynamics to:

$$\dot{e}_2 = -k_2\, e_2 - e_1 + e_3 \tag{4.32}$$

The augmented Lyapunov function and its derivative are:

$$V_2 = V_1 + \frac{1}{2}\, e_2^2, \qquad \dot{V}_2 = -k_1\, e_1^2 - k_2\, e_2^2 + e_2\, e_3 \tag{4.33}$$

When $e_3 = 0$, the derivative $\dot{V}_2$ is negative definite, confirming the asymptotic convergence of both e_1 and e_2 to zero. The residual term $e_2 e_3$ is cancelled in the following stage by the design of the actual control input U_1.

Stage 3: Control Input

At this final stage, the actual control input U_1 is derived by imposing the desired dynamics on e_3. Differentiating E from (4.28) and substituting $\dot{x}_3$ from (4.19):

$$\dot{E} = \frac{2a_1 x_3 + b_1}{I_1}\left(-\frac{T_{10}}{T_{11}}\, x_3 + \frac{K_1}{T_{11}}\, U_1\right) - \frac{B_{1\psi}}{I_1}\, dx_2 \tag{4.34}$$

The total time derivative of α_2 in regulation mode is:

$$\dot{\alpha}_2\big|_{\text{reg}} = \frac{\partial \alpha_2}{\partial x_1}\, x_2 + \frac{\partial \alpha_2}{\partial x_2}\, dx_2 \tag{4.35}$$

where the partial derivatives, computed from (4.31), are:

$$\frac{\partial \alpha_2}{\partial x_1} = -k_1 k_2 - 1 + \frac{M_g \cos x_1 + B_{2\psi}}{I_1} \tag{4.36}$$

$$\frac{\partial \alpha_2}{\partial x_2} = -(k_1 + k_2) \tag{4.37}$$

The final Lyapunov function is selected as $V_3 = V_2 + \frac{1}{2} e_3^2$. The target dynamics $\dot{e}_3 = -e_2 - k_3 e_3$ with $k_3 > 0$ are imposed to render $\dot{V}_3$ negative definite. The actual control input U_1 is obtained by solving $\dot{E} - \dot{\alpha}_2 = -e_2 - k_3 e_3$, yielding the **pitch regulation control law**:

$$\boxed{\begin{aligned} U_{1,\text{reg}} = \frac{I_1\, T_{11}}{K_1(2a_1 x_3 + b_1)} \Bigg(& - e_2 - k_3\, e_3 + \frac{T_{10}(2a_1 x_3 + b_1)}{I_1\, T_{11}}\, x_3 + \frac{B_{1\psi}}{I_1}\, dx_2 \\ & + \left(-k_1 k_2 - 1 + \frac{M_g \cos x_1 + B_{2\psi}}{I_1} \right) x_2 - (k_1 + k_2)\, dx_2 \Bigg) \end{aligned}} \tag{4.38}$$

The law is well defined provided $(2a_1 x_3 + b_1) \neq 0$. The composite Lyapunov derivative satisfies:

$$\dot{V}_3 = -k_1\, e_1^2 - k_2\, e_2^2 - k_3\, e_3^2 < 0 \tag{4.39}$$

which is negative definite, establishing local asymptotic stability of the error origin and guaranteeing $x_1 \to x_{1d}$.

4.4.3 Pitch Trajectory Tracking Controller

In tracking mode, the objective is to drive x_1 to a smooth time-varying reference $x_{1d}(t)$, whose derivatives $\dot{x}_{1d}$, $\ddot{x}_{1d}$, and $x_{1d}^{(3)}$ are assumed known and bounded at all times. The three-stage structure follows that of the regulation case, with the addition of feedforward terms at each stage to compensate for the time-varying reference trajectory.

Stage 1: Position Error

The position tracking error and its time derivative are:

$$e_1 = x_1 - x_{1d}(t), \qquad \dot{e}_1 = x_2 - \dot{x}_{1d} \tag{4.40}$$

The first virtual control law is augmented with a feedforward term to compensate for the reference velocity:

$$\alpha_1 = -k_1\, e_1 + \dot{x}_{1d}, \qquad k_1 > 0 \tag{4.41}$$

Defining the second error variable as $e_2 = x_2 - \alpha_1$ yields the closed-loop dynamics of the first error:

$$\dot{e}_1 = -k_1\, e_1 + e_2 \tag{4.42}$$

with the corresponding Lyapunov function and its derivative:

$$V_1 = \frac{1}{2}\, e_1^2, \qquad \dot{V}_1 = -k_1\, e_1^2 + e_1\, e_2 \tag{4.43}$$

When $e_2 = 0$, the derivative $\dot{V}_1 = -k_1 e_1^2$ is negative definite, confirming asymptotic convergence of e_1 to zero.

Stage 2: Velocity Error

The time derivative of α_1 in tracking mode is:

$$\dot{\alpha}_1 = -k_1(x_2 - \dot{x}_{1d}) + \ddot{x}_{1d} \tag{4.44}$$

Using the auxiliary variable E defined in (4.28), the velocity error dynamics become:

$$\dot{e}_2 = E - \frac{M_g \sin x_1 + B_{2\psi}\, x_1}{I_1} + k_1\, x_2 - k_1\, \dot{x}_{1d} - \ddot{x}_{1d} \tag{4.45}$$

The second virtual control law extends the regulation expression with feedforward terms for the reference acceleration:

$$\alpha_2 = -k_2\, e_2 - e_1 + \frac{M_g \sin x_1 + B_{2\psi}\, x_1}{I_1} - k_1\, x_2 + k_1\, \dot{x}_{1d} + \ddot{x}_{1d}, \qquad k_2 > 0 \tag{4.46}$$

The augmented Lyapunov function and its derivative are:

$$V_2 = V_1 + \frac{1}{2}\, e_2^2, \qquad \dot{V}_2 = -k_1\, e_1^2 - k_2\, e_2^2 + e_2\, e_3 \tag{4.47}$$

The structure of $\dot{V}_2$ is identical to the regulation case, confirming that the feedforward augmentation preserves the Lyapunov decrease property while accommodating the time-varying reference.

Stage 3: Control Input

The total time derivative of α_2 in tracking mode includes an explicit time component arising from the time-varying reference:

$$\dot{\alpha}_2\big|_{\text{track}} = \frac{\partial \alpha_2}{\partial x_1}\, x_2 + \frac{\partial \alpha_2}{\partial x_2}\, dx_2 + \frac{\partial \alpha_2}{\partial t} \tag{4.48}$$

The partial derivatives with respect to the state variables are identical to those of the regulation case:

$$\frac{\partial \alpha_2}{\partial x_1} = -k_1 k_2 - 1 + \frac{M_g \cos x_1 + B_{2\psi}}{I_1} \tag{4.49}$$

$$\frac{\partial \alpha_2}{\partial x_2} = -(k_1 + k_2) \tag{4.50}$$

The explicit time derivative, arising from the reference signal terms in (4.46), is:

$$\frac{\partial \alpha_2}{\partial t} = (k_1 k_2 + 1)\, \dot{x}_{1d} + (k_1 + k_2)\, \ddot{x}_{1d} + x_{1d}^{(3)} \tag{4.51}$$

Applying the same Lyapunov argument as in the regulation case and solving for U_1 yields the **pitch tracking control law**:

$$\boxed{\begin{aligned} U_{1,\text{track}} = \frac{I_1\, T_{11}}{K_1(2a_1 x_3 + b_1)} \Bigg(& - e_2 - k_3\, e_3 + \frac{T_{10}(2a_1 x_3 + b_1)}{I_1\, T_{11}}\, x_3 + \frac{B_{1\psi}}{I_1}\, dx_2 \\ & + \left(-k_1 k_2 - 1 + \frac{M_g \cos x_1 + B_{2\psi}}{I_1} \right) x_2 - (k_1 + k_2)\, dx_2 \\ & + (k_1 k_2 + 1)\, \dot{x}_{1d} + (k_1 + k_2)\, \ddot{x}_{1d} + x_{1d}^{(3)} \Bigg) \end{aligned}} \tag{4.52}$$

The composite Lyapunov derivative satisfies:

$$\dot{V}_3 = -k_1\, e_1^2 - k_2\, e_2^2 - k_3\, e_3^2 < 0 \tag{4.53}$$

This expression is negative definite, establishing local asymptotic stability of the tracking error origin and guaranteeing $x_1 \to x_{1d}(t)$ asymptotically.

Remark. It is straightforward to verify that (4.52) reduces exactly to (4.38) when $x_{1d} = \text{const}$, since all feedforward and explicit time-derivative terms vanish identically. The tracking law therefore subsumes the regulation law as a special case.

4.5 Yaw Subsystem Controller Design

4.5.1 Simplified Model for Control Synthesis

The yaw subsystem governs the horizontal rotation of the TRMS and is driven by the tail rotor. The full three-state equations of motion are:

$$\begin{cases} \dfrac{d\varphi}{dt} = \dot{\varphi} \\ \dfrac{d\dot{\varphi}}{dt} = \dfrac{1}{I_2}\Big(a_2\tau_2^2 + b_2\tau_2 - B_{1\varphi}\,\dot{\varphi} - B_{2\varphi}\,\sin\big(g(\varphi)\big) - 1.75\,K_c\big(a_1\tau_1^2 + b_1\tau_1\big)\Big) \\ \dfrac{d\tau_2}{dt} = -\dfrac{T_{20}}{T_{21}}\,\tau_2 + \dfrac{K_2}{T_{21}}\,U_2 \end{cases} \tag{4.54}$$

Applying the same decentralized argument as for the pitch axis, two simplifications are introduced. First, the cross-coupling reaction thrust from the main rotor is neglected. Second, the Coulomb friction term $B_{2\varphi}\,\mathrm{sgn}(\varphi)$ is approximated by the linear viscous term $B_{2\varphi}\varphi$ for the purpose of control synthesis. With state variables $x_4 = \varphi$, $x_5 = \dot{\varphi}$, and $x_6 = \tau_2$, the simplified model used for control synthesis is:

$$\begin{cases} \dot{x}_4 = x_5 \\ \dot{x}_5 = \dfrac{1}{I_2}\Big(a_2x_6^2 + b_2x_6 - B_{1\varphi}\,x_5 - B_{2\varphi}\,x_4\Big) \\ \dot{x}_6 = -\dfrac{T_{20}}{T_{21}}\,x_6 + \dfrac{K_2}{T_{21}}\,U_2 \end{cases} \tag{4.55}$$

It is noted that the yaw acceleration equation contains no gravity term, since the yaw axis is horizontal and the gravitational moment acts exclusively on the pitch axis. The shorthand $dx_5 \triangleq \dot{x}_5$ is used below:

$$dx_5 \;\triangleq\; \dot{x}_5 = \frac{1}{I_2}\Big(a_2x_6^2 + b_2x_6 - B_{1\varphi}\,x_5 - B_{2\varphi}\,x_4\Big) \tag{4.56}$$

4.5.2 Yaw Regulation Controller

The objective is to drive x_4 to a constant setpoint $x_{4d} = \text{const}$. Dedicated error variables e_4, e_5, e_6 and gains $k_4, k_5, k_6 > 0$ are introduced to maintain notational consistency with the yaw subsystem throughout the derivation.

Stage 1: Position Error

The position tracking error and its time derivative are:

$$e_4 = x_4 - x_{4d}, \qquad \dot{e}_4 = x_5 \tag{4.57}$$

Treating x_5 as a virtual control input, the first stabilising function is chosen as:

$$\alpha_4 = -k_4\, e_4 \tag{4.58}$$

A second error variable $e_5 = x_5 - \alpha_4$ is introduced, yielding the closed-loop dynamics:

$$\dot{e}_4 = -k_4\, e_4 + e_5 \tag{4.59}$$

with the Lyapunov function and its derivative:

$$V_4 = \frac{1}{2}\, e_4^2, \qquad \dot{V}_4 = -k_4\, e_4^2 + e_4\, e_5 \tag{4.60}$$

When $e_5 = 0$, the derivative $\dot{V}_4 = -k_4 e_4^2$ is negative definite, ensuring the asymptotic convergence of e_4 to zero.

Stage 2: Velocity Error

The objective of this stage is to synthesise α_5 such that e_5 converges to zero. Differentiating e_5 with $\dot{\alpha}_4 = -k_4 x_5$ and introducing the auxiliary variable:

$$F = \frac{1}{I_2}\big(a_2 x_6^2 + b_2 x_6 - B_{1\varphi}\, x_5\big) \tag{4.61}$$

the velocity error dynamics are:

$$\dot{e}_5 = F - \frac{B_{2\varphi}\, x_4}{I_2} + k_4\, x_5 \tag{4.62}$$

Treating F as a new virtual control input and defining $e_6 = F - \alpha_5$, the second virtual control law is chosen to cancel the known terms and add damping on e_5:

$$\alpha_5 = -k_5\, e_5 - e_4 + \frac{B_{2\varphi}\, x_4}{I_2} - k_4\, x_5, \qquad k_5 > 0 \tag{4.63}$$

The augmented Lyapunov function and its derivative are:

$$V_5 = V_4 + \frac{1}{2}\, e_5^2, \qquad \dot{V}_5 = -k_4\, e_4^2 - k_5\, e_5^2 + e_5\, e_6 \tag{4.64}$$

Stage 3: Control Input

At this final stage, the actual control input U_2 is derived. Differentiating F and substituting $\dot{x}_6$:

$$\dot{F} = \frac{2a_2x_6 + b_2}{I_2}\left(-\frac{T_{20}}{T_{21}}\,x_6 + \frac{K_2}{T_{21}}\,U_2\right) - \frac{B_{1\varphi}}{I_2}\,dx_5 \tag{4.65}$$

The partial derivatives of α_5 with respect to the state variables are:

$$\frac{\partial \alpha_5}{\partial x_4} = -k_4k_5 - 1 + \frac{B_{2\varphi}}{I_2} \tag{4.66}$$

$$\frac{\partial \alpha_5}{\partial x_5} = -(k_4 + k_5) \tag{4.67}$$

Imposing the target dynamics $\dot{e}_6 = -e_5 - k_6e_6$ and solving for U_2 yields the **yaw regulation control law**:

$$\boxed{\begin{aligned} U_{2,\text{reg}} = \frac{I_2\,T_{21}}{K_2(2a_2x_6 + b_2)}\Bigg(& -e_5 - k_6\,e_6 + \frac{T_{20}(2a_2x_6 + b_2)}{I_2\,T_{21}}\,x_6 + \frac{B_{1\varphi}}{I_2}\,dx_5 \\ & + \left(-k_4k_5 - 1 + \frac{B_{2\varphi}}{I_2}\right)x_5 - (k_4 + k_5)\,dx_5\Bigg) \end{aligned}} \tag{4.68}$$

The composite Lyapunov derivative satisfies:

$$\dot{V}_6 = -k_4\,e_4^2 - k_5\,e_5^2 - k_6\,e_6^2 < 0 \tag{4.69}$$

which is negative definite, establishing local asymptotic stability of the error origin and guaranteeing $x_4 \to x_{4d}$.

4.5.3 Yaw Trajectory Tracking Controller

The objective in tracking mode is to drive x_4 to a smooth time-varying reference $x_{4d}(t)$, with $\dot{x}_{4d}$, $\ddot{x}_{4d}$, and $x_{4d}^{(3)}$ assumed known and bounded at all times. The derivation follows the structure of the yaw regulation controller, with feedforward compensation terms introduced at each stage to account for the time-varying nature of the reference.

Stage 1: Position Error

The position tracking error and its time derivative are:

$$e_4 = x_4 - x_{4d}(t), \qquad \dot{e}_4 = x_5 - \dot{x}_{4d} \tag{4.70}$$

The first virtual control law is augmented with a feedforward term for the reference velocity:

$$\alpha_4 = -k_4\, e_4 + \dot{x}_{4d} \tag{4.71}$$

Defining $e_5 = x_5 - \alpha_4$ gives the closed-loop dynamics:

$$\dot{e}_4 = -k_4\, e_4 + e_5 \tag{4.72}$$

with the corresponding Lyapunov function and its derivative:

$$V_4 = \frac{1}{2}\, e_4^2, \qquad \dot{V}_4 = -k_4\, e_4^2 + e_4\, e_5 \tag{4.73}$$

Stage 2: Velocity Error

The time derivative of α_4 in tracking mode is:

$$\dot{\alpha}_4 = -k_4(x_5 - \dot{x}_{4d}) + \ddot{x}_{4d} \tag{4.74}$$

Using F from (4.61), the velocity error dynamics are:

$$\dot{e}_5 = F - \frac{B_{2\varphi}\, x_4}{I_2} + k_4\, x_5 - k_4\, \dot{x}_{4d} - \ddot{x}_{4d} \tag{4.75}$$

The second virtual control law is extended with feedforward compensation for the reference acceleration:

$$\alpha_5 = -k_5\, e_5 - e_4 + \frac{B_{2\varphi}\, x_4}{I_2} - k_4\, x_5 + k_4\, \dot{x}_{4d} + \ddot{x}_{4d}, \qquad k_5 > 0 \tag{4.76}$$

The augmented Lyapunov function and its derivative are:

$$V_5 = V_4 + \frac{1}{2}\, e_5^2, \qquad \dot{V}_5 = -k_4\, e_4^2 - k_5\, e_5^2 + e_5\, e_6 \tag{4.77}$$

Stage 3: Control Input

The total time derivative of α_5 in tracking mode includes an explicit time component:

$$\dot{\alpha}_5\big|_{\text{track}} = \frac{\partial \alpha_5}{\partial x_4}\, x_5 + \frac{\partial \alpha_5}{\partial x_5}\, dx_5 + \frac{\partial \alpha_5}{\partial t} \tag{4.78}$$

The partial derivatives with respect to the state variables are identical to the regulation case:

$$\frac{\partial \alpha_5}{\partial x_4} = -k_4 k_5 - 1 + \frac{B_{2\varphi}}{I_2} \tag{4.79}$$

$$\frac{\partial \alpha_5}{\partial x_5} = -(k_4 + k_5) \tag{4.80}$$

The explicit time derivative arising from the reference signal terms in (4.76) is:

$$\frac{\partial \alpha_5}{\partial t} = (k_4 k_5 + 1)\, \dot{x}_{4d} + (k_4 + k_5)\, \ddot{x}_{4d} + x_{4d}^{(3)} \tag{4.81}$$

Applying the same Lyapunov argument as in the regulation case and solving for U_2 yields the **yaw tracking control law**:

$$\boxed{\begin{aligned} U_{2,\text{track}} = \frac{I_2\, T_{21}}{K_2(2a_2 x_6 + b_2)} \Bigg(& - e_5 - k_6\, e_6 + \frac{T_{20}(2a_2 x_6 + b_2)}{I_2\, T_{21}}\, x_6 + \frac{B_{1\varphi}}{I_2}\, dx_5 \\ & + \left(-k_4 k_5 - 1 + \frac{B_{2\varphi}}{I_2} \right) x_5 - (k_4 + k_5)\, dx_5 \\ & + (k_4 k_5 + 1)\, \dot{x}_{4d} + (k_4 + k_5)\, \ddot{x}_{4d} + x_{4d}^{(3)} \Bigg) \end{aligned}} \tag{4.82}$$

The composite Lyapunov derivative satisfies:

$$\dot{V}_6 = -k_4\, e_4^2 - k_5\, e_5^2 - k_6\, e_6^2 < 0 \tag{4.83}$$

which is negative definite, guaranteeing the asymptotic convergence $x_4 \to x_{4d}(t)$.

Remark. As for the pitch axis, (4.82) reduces exactly to (4.68) when $x_{4d} = \text{const}$, since all feedforward and explicit time-derivative terms vanish identically. The tracking law therefore subsumes the regulation law as a special case.

4.6 Practical Implementation Considerations

The theoretical control laws require several modifications for implementation in the MATLAB/Simulink environment and for deployment on the physical hardware. These modifications address numerical stability, computational efficiency, and actuator protection, and are described in turn below.

Algebraic Simplification

The theoretical expressions for U_1 and U_2 contain the term $\frac{T_{i0}(2a_i x_j + b_i)}{I_i T_{i1}}\, x_j$ inside the parenthesis, multiplied at the front of each law by the inverse factor $\frac{I_i T_{i1}}{K_i(2a_i x_j + b_i)}$. These factors

cancel algebraically, leaving the simplified feedforward term $\frac{T_{i0}}{K_i} x_j$. This cancellation was exploited directly in the simulation code to reduce floating-point operations and improve real-time execution speed.

Singularity Avoidance

The control laws contain the denominator $(2a_i x_j + b_i)$, which is related to the derivative of the motor thrust characteristic with respect to the torque state. Under certain operating conditions, particularly during start-up transients, this term may approach zero, causing the control gain to diverge. To prevent numerical instability, the denominator is lower-bounded by a small threshold:

$$|2a_i x_j + b_i| \geq \varepsilon, \qquad \varepsilon = 10^{-6} \tag{4.84}$$

Whenever the magnitude falls below ε, the denominator is clamped to $\varepsilon \cdot \text{sgn}(2a_i x_j + b_i)$, protecting the system from unbounded control commands during low-thrust transients.

Actuator Saturation

The TRMS motor amplifiers are limited to the voltage range $[-2.5\,\text{V}, +2.5\,\text{V}]$. The computed control signals are saturated to this interval before being applied to the plant, preventing damage to the drive electronics and respecting the physical power limits of the actuators.

Generation of Reference Derivatives

The trajectory tracking controllers require the first three time derivatives of the reference signal. Numerical differentiation of a measured position signal was avoided, as it is highly sensitive to encoder noise and would introduce high-frequency disturbances into the feedforward channel. Instead, the reference trajectory and its analytic derivatives were generated by dedicated signal-source blocks within Simulink, providing noise-free feedforward signals. For physical real-time implementation, the same approach is retained by pre-computing the reference trajectory and its derivatives offline from a smooth path plan.

4.7 Observer-Based Control Architecture

4.7.1 The Need for State Estimation

The backstepping control laws derived in the preceding sections require the full state vector $\mathbf{x} \in \mathbb{R}^6$. In practice, the TRMS is equipped only with two incremental optical encoders, which provide direct measurements of the pitch angle x_1 and the yaw angle x_4. The angular velocities x_2 and x_5, and the motor torque states x_3 and x_6, are inaccessible to direct measurement.

A naive approach would be to estimate the velocities by numerical differentiation of the position signals. This approach is not viable for the TRMS, because differentiation is a high-pass filtering operation that strongly amplifies the measurement noise inherent in incremental encoder signals, and feeding such corrupted estimates to a high-gain controller would in all likelihood lead to closed-loop instability.

This fundamental gap between the required and available information makes a state observer a strict necessity rather than a design option. A High-Gain Observer is therefore incorporated into the architecture to reconstruct the four unmeasured states from the known control inputs and the measured outputs.

4.7.2 High-Gain Observer Theory

The High-Gain Observer is a well-established class of nonlinear observer whose theoretical foundations were laid by Bornard and Hammouri [43]. For a MIMO nonlinear system of the form:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \mathbf{B}\mathbf{u} \\ \mathbf{y} &= \mathbf{h}(\mathbf{x}) \end{aligned} \tag{4.85}$$

the general HGO equation is:

$$\dot{\hat{x}} = f(\hat{x}) + Bu + \left(\frac{\partial \phi(\hat{x})}{\partial \hat{x}}\right)^{-1} \Lambda^{-1}(\theta, \delta) L(y - C\hat{x}) \tag{4.86}$$

where $\hat{\mathbf{x}}$ is the estimated state vector and the term $(y - C\hat{x})$ is the observation error. The structured correction term is designed to drive the estimation error $\mathbf{e_x} = \mathbf{x} - \hat{\mathbf{x}}$ to zero.

The design requires the system to be uniformly observable. The observability mapping $\phi(\mathbf{x})$ is constructed from successive Lie derivatives of the output function $h(\mathbf{x})$ along the vector field $f(\mathbf{x})$. The matrix $\mathbf{\Lambda}^{-1}(\theta, \delta)$ is block diagonal and contains powers of the high-gain parameters θ_k, scaled by the power indices δ_k. These parameters effectively set the time scale of the observer dynamics; larger values of θ_k produce faster convergence of the estimation error at the cost of increased noise amplification. The constant gain matrix $\mathbf{L}$ is chosen to place the poles of the linearised error dynamics in the stable left-half plane.

4.7.3 Application and Formulation for the TRMS

For the TRMS, the state dimension is $n = 6$ and the number of outputs is $p = 2$. The observability indices are $\mu_1 = 3$ for the pitch subsystem and $\mu_2 = 3$ for the yaw subsystem, satisfying $\mu_1 + \mu_2 = n$. The observability mapping is constructed as:

$$\phi(\hat{x}) = \left[L_f^0 h_1 \; L_f h_1 \cdots L_f^{\mu_1 - 1} h_1 \mid \cdots \mid L_f^0 h_p \; L_f h_p \cdots L_f^{\mu_p - 1} h_p\right]^T \tag{4.87}$$

The observer gain matrix $\Lambda^{-1}(\theta, \delta)$ is block diagonal:

$$\Lambda^{-1}(\theta, \delta) = \begin{bmatrix} \theta^{\delta_1}\Delta_1(\theta^{\delta_1}) & 0 \\ 0 & \theta^{\delta_2}\Delta_2(\theta^{\delta_2}) \end{bmatrix} \tag{4.88}$$

with the scaling matrices:

$$\Delta_1(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/\theta_1 & 0 \\ 0 & 0 & 1/\theta_1^2 \end{bmatrix}, \quad \Delta_2(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1/\theta_2 & 0 \\ 0 & 0 & 1/\theta_2^2 \end{bmatrix} \tag{4.89}$$

The Jacobian of the observability mapping evaluated for the TRMS with $n = 6$, $p = 2$, and $\mu_1 = \mu_2 = 3$ is:

$$\frac{\partial \phi(\hat{x})}{\partial \hat{x}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \dfrac{\partial L_f^2 h_1}{\partial \hat{x}_1} & \dfrac{\partial L_f^2 h_1}{\partial \hat{x}_2} & 0 & \dfrac{\partial L_f^2 h_1}{\partial \hat{x}_4} & \dfrac{\partial L_f^2 h_1}{\partial \hat{x}_5} & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \dfrac{\partial L_f^2 h_2}{\partial \hat{x}_4} & \dfrac{\partial L_f^2 h_2}{\partial \hat{x}_5} & \dfrac{\partial L_f^2 h_2}{\partial \hat{x}_6} \end{bmatrix} \tag{4.90}$$

with the nonzero entries:

$$\begin{aligned} \frac{\partial L_f^2 h_1}{\partial \hat{x}_1} &= \frac{-M_g \cos(\hat{x}_1) + K_{gy} \sin(\hat{x}_1)\hat{x}_4(a_1\hat{x}_2^2 + b_1\hat{x}_5)}{I_1} \\ \frac{\partial L_f^2 h_1}{\partial \hat{x}_2} &= -\frac{B_{1\psi}}{I_1} \\ \frac{\partial L_f^2 h_1}{\partial \hat{x}_4} &= \frac{K_{gy} \cos(\hat{x}_1)(a_1\hat{x}_2^2 + b_1\hat{x}_5)}{I_1} \\ \frac{\partial L_f^2 h_1}{\partial \hat{x}_5} &= \frac{1}{I_1}\Big(2a_1\hat{x}_5 + b_1 + K_{gy} \sin(\hat{x}_1)\hat{x}_4(a_1\hat{x}_5^2 + b_1\hat{x}_5) \\ &\qquad - (2a_1 K_{gy} \cos(\hat{x}_1)\hat{x}_5 - b_1 K_{gy} \cos(\hat{x}_1)\hat{x}_4)\Big) \\ \frac{\partial L_f^2 h_2}{\partial \hat{x}_4} &= -\frac{B_{1\phi}}{I_2} \\ \frac{\partial L_f^2 h_2}{\partial \hat{x}_5} &= \frac{-1.75K_c(2a_1\hat{x}_5 + b_1)}{I_2} \\ \frac{\partial L_f^2 h_2}{\partial \hat{x}_6} &= \frac{2a_2\hat{x}_6 + b_2}{I_2} \end{aligned} \tag{4.91}$$

The constant gain matrix $\mathbf{L}$ is structured to produce stable Hurwitz error dynamics for each subsystem:

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_p & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_y \end{bmatrix}, \quad \mathbf{L}_p = \begin{bmatrix} 3\theta_p \\ 3\theta_p^2 \\ \theta_p^3 \end{bmatrix}, \quad \mathbf{L}_y = \begin{bmatrix} 3\theta_y \\ 3\theta_y^2 \\ \theta_y^3 \end{bmatrix} \tag{4.92}$$

The high-gain parameters θ_p and θ_y are the quantities adapted online by the neural networks in the proposed architecture. In the baseline controller, these parameters are held at fixed constant values, which is the source of the performance limitations documented in Chapter 5.

4.7.4 Stability Analysis of the Observer-Based Controller

A rigorous analysis of the integrated closed-loop system is required before implementation. The backstepping controller and the high-gain observer have been designed independently; their interaction must therefore be studied carefully. For general nonlinear systems, the classical separation principle applicable in the linear setting does not hold, and a dedicated framework is required.

The analysis relies on the **nonlinear separation principle** [21], which provides conditions under which the stability of the integrated system can be inferred from the stability of its components. For the specific combination of a globally stabilising state-feedback controller and a high-gain observer, **semi-global practical asymptotic stability (SPAS)** of the integrated system is guaranteed provided two conditions are satisfied [21, 48]:

1. **Controller stability:** The state-feedback control law $\mathbf{u} = \alpha(\mathbf{x})$, when applied with the true state, must render the closed-loop system globally asymptotically stable.

2. **Observer stability:** The estimation error dynamics must be globally asymptotically stable, with $\mathbf{e}_\mathbf{x} = \mathbf{x} - \hat{\mathbf{x}} \to 0$.

Both conditions are satisfied by construction in this work. The backstepping procedure yields a composite Lyapunov function for the entire pitch and yaw error systems whose negative-definite time derivative formally establishes global asymptotic stability of the state-feedback closed-loop system. The HGO, for a sufficiently large gain parameter θ, is known to produce globally, uniformly, and exponentially stable estimation error dynamics [47].

Since both components satisfy the required global stability properties, the nonlinear separation principle applies. The complete closed-loop system formed by the TRMS plant, the backstepping controller operating on estimated states $\mathbf{u} = \alpha(\hat{\mathbf{x}})$, and the high-gain observer achieves **semi-global practical asymptotic stability**. This result guarantees that for any bounded set of initial conditions and any desired tolerance $\epsilon > 0$, there exists a sufficiently large gain θ^* such that, for all $\theta > \theta^*$, the system trajectories ultimately converge to and remain within a ball of radius ϵ around the origin.

A practical consequence of this result is the well-known design trade-off inherent to the HGO: while a larger gain accelerates error convergence and improves closed-loop performance, it simultaneously amplifies measurement noise and may induce high-amplitude transients, referred to as the peaking phenomenon, in the control signal during the initial phase of operation. This trade-off constitutes the central motivation for the adaptive gain strategies presented in the following section.

4.8 Design of the Proposed AI-Enhanced High-Gain Observers

The preceding analysis identifies the fixed observer gain as the fundamental weakness of the baseline architecture. The value of θ that guarantees fast transient convergence is incompatible with the value that suppresses sensor noise during steady-state operation, and no single fixed setting can satisfy both requirements simultaneously under dynamic operating conditions.

This section presents the primary contribution of this thesis: a control architecture in which the HGO gains θ_p and θ_y are rendered adaptive through an intelligent tuner based on artificial neural networks. The tuner is designed to command high gains when the observation error is large, ensuring fast convergence during transients, and to reduce the gains progressively as the error diminishes, limiting noise amplification during steady-state tracking. Two distinct network topologies are designed and compared: the Feedforward Neural Network (FFNN) and the Radial Basis Function Neural Network (RBFNN).

4.8.1 Neural Network Foundations for Adaptive Control

Feedforward Neural Networks

A Feedforward Neural Network, also termed a multilayer perceptron (MLP), is characterized by a strictly unidirectional flow of information from the input layer through one or more hidden layers to the output layer, with no feedback connections or cycles. This acyclic structure distinguishes FFNNs from recurrent architectures and makes them well suited to static mapping tasks such as function approximation.

The origins of the FFNN can be traced to the McCulloch-Pitts neuron [101] and Rosenblatt's Perceptron [102]. The fundamental limitation of single-layer networks, namely their inability to solve non-linearly separable problems, was established by Minsky and Papert [103], motivating research into multilayer designs. The subsequent development of the backpropagation algorithm [104] provided an efficient method for training these networks and remains the standard optimization approach today.

The theoretical power of FFNNs rests on the Universal Approximation Theorem [105],

which guarantees that a single hidden layer with a sufficient number of neurons can approximate any continuous function to an arbitrary degree of accuracy. In each hidden neuron, a weighted sum of inputs is computed, a bias is added, and the result is passed through a nonlinear activation function such as sigmoid, tanh, or ReLU. Without this nonlinearity, the entire network collapses to a linear model irrespective of its depth.

Radial Basis Function Networks

A Radial Basis Function Network (RBFNN) is a feedforward network comprising an input layer, a single hidden layer of nonlinear RBF neurons, and a linear output layer. Its defining characteristic is a localized response: each hidden neuron responds strongly only to inputs that fall within its specific receptive field in the input space, in direct contrast to the global activation strategy of the FFNN. This locality property often enables faster training compared to traditional MLPs [106, 107] and is particularly advantageous for capturing region-dependent nonlinear behavior. The core principle of an RBFNN is to transform the input vector into a higher-dimensional space through the hidden layer, where a simple linear operation on the resulting activations suffices to produce the desired output. Training typically follows a two-stage hybrid approach: the centers and spreads of the RBF neurons are first determined through an unsupervised algorithm such as k-means clustering, after which the linear output weights are solved by least squares. This strategy avoids the nonlinear optimization required by backpropagation and is the primary source of the RBFNN's characteristic training speed.

Comparison of FFNN and RBFNN Architectures

While both architectures are feedforward universal approximators, FFNNs and RBFNNs differ fundamentally in their internal mechanics, training procedures, and generalization properties. Table 4.1 summarizes the key distinctions.

Table 4.1: Comparison of Feedforward Neural Networks (MLPs) and Radial Basis Function (RBF) Networks.

Feature	Feedforward Neural Networks (MLPs)	Radial Basis Function (RBF) Networks
Hidden Layer Activation	Global (e.g., sigmoid, tanh, ReLU)	Localized (e.g., Gaussian)
Input Processing	Weighted sum of inputs (dot product)	Euclidean distance to neuron center
Typical Training Method	Backpropagation (non-linear optimization)	Hybrid: unsupervised then linear regression
Number of Hidden Layers	Can be multiple (deep)	Typically one hidden layer
Nature of Basis Functions	Global, affects entire input space	Localized, affects specific regions
Generalization Behaviour	Interpolates between data points	Averages contributions from prototypes
Training Speed	Can be slower	Often faster

The primary distinction lies in how each network responds to an arbitrary input. In an MLP, a large proportion of the neurons across all layers contribute to the output regardless of the input location, since sigmoid and tanh activations are nonzero over the entire real line.

In an RBFNN, a given input activates only those neurons whose centers lie in its immediate vicinity; neurons with centers far from the input contribute negligibly. This localized behavior makes the RBFNN particularly effective at modelling phenomena that differ substantially across regions of the input space, such as the gain adaptation requirements of the HGO considered here.

4.8.2 FFNN Architecture, Input Preprocessing, and Mathematical Formulation

A Feedforward Neural Network was designed to function as an intelligent gain tuner, adjusting the HGO parameters online in response to the current observation error. The selection of an FFNN is justified by its proven capability as a universal function approximator, making it well suited to learn the complex nonlinear mapping between the system error and the optimal observer gains. The architecture was refined through an iterative process of empirical evaluation to ensure robust and computationally efficient real-time performance.

The final network comprises three layers: a single input neuron, a hidden layer with five neurons, and an output layer with two neurons corresponding to the two HGO gain parameters. This compact structure proved sufficient for the task, providing enough representational capacity to model the desired gain behavior without incurring significant computational overhead or the risk of overfitting to the single-dimensional input. A conceptual diagram of the architecture is presented in Figure 4.1.

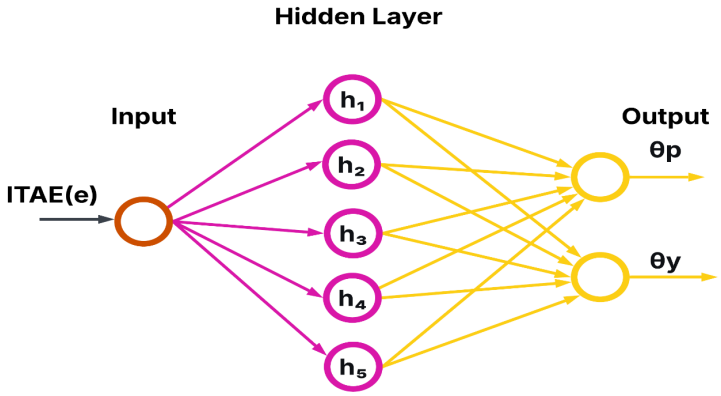

Figure 4.1: Conceptual architecture of the Feedforward Neural Network used for online HGO gain tuning.

4.8.3 Online Backpropagation Training Algorithm and TRMS Integration

Input Signal and Preprocessing. A key design decision concerned the choice of input signal for the network. While separate pitch and yaw observation errors could have been used,

preliminary experiments showed that a single aggregate performance metric yields a more stable and robust input for the learning algorithm. The Integral of Time-weighted Absolute Error (ITAE) was selected as the sole input, defined as:

$$\text{ITAE}(t) = \int_0^t \tau \left(|\psi(\tau) - \hat{\psi}(\tau)| + |\varphi(\tau) - \hat{\varphi}(\tau)| \right) d\tau \tag{4.93}$$

The raw ITAE value e is first clipped to the range $[-10, 10]$:

$$e_c = \text{clip}(e,\ -10,\ 10) \tag{4.94}$$

and then normalized via:

$$x = \tanh\left(\frac{e_c}{2}\right) \tag{4.95}$$

This two-step preprocessing ensures that the network always receives a bounded, well-scaled input $x \in (-1, 1)$, regardless of the magnitude of the raw ITAE signal.

Hidden and Output Layer Formulation. The five hidden neurons transform the normalized input x into a hidden state vector $\mathbf{h} \in \mathbb{R}^5$:

$$\mathbf{h} = \tanh(\mathbf{W}_{xh}\, x + \mathbf{b}_h) \tag{4.96}$$

where $\mathbf{W}_{xh} \in \mathbb{R}^5$ is the input-to-hidden weight vector and $\mathbf{b}_h \in \mathbb{R}^5$ is the hidden bias vector. The output layer maps this hidden state to a normalized output vector $\mathbf{y}_{net} \in \mathbb{R}^2$ through a sigmoid activation:

$$\mathbf{y}_{net} = \sigma(\mathbf{W}_{hy}\, \mathbf{h} + \mathbf{b}_y) = \frac{1}{1 + \exp(-(\mathbf{W}_{hy}\, \mathbf{h} + \mathbf{b}_y))} \tag{4.97}$$

where $\mathbf{W}_{hy} \in \mathbb{R}^{2\times 5}$ is the hidden-to-output weight matrix and $\mathbf{b}_y \in \mathbb{R}^2$ is the output bias. The sigmoid function constrains $\mathbf{y}_{net}$ to the range $(0, 1)$. The final observer gain parameters are obtained by scaling this normalized output to the experimentally determined operational ranges:

$$\theta_p = 1 + 4 \cdot y_{net,1} \tag{4.98}$$

$$\theta_y = 0.5 + 1.5 \cdot y_{net,2} \tag{4.99}$$

yielding $\theta_p \in (1, 5)$ and $\theta_y \in (0.5, 2)$. All network weights were initialized using the Xavier method.

Dynamic Target and Justification for Neural Approximation. The learning process is driven by a dynamically generated target vector $\mathbf{t}$, which encodes the desired gain behavior

in the same normalized $(0, 1)$ range as $\mathbf{y}_{net}$:

$$\mathbf{t} = \begin{bmatrix} \dfrac{1 + \tanh(e_c)}{2} \\ \dfrac{1 + \tanh(e_c)}{2} \end{bmatrix} \tag{4.100}$$

A natural question is why an FFNN is trained to approximate this heuristic rather than applying it directly. Direct application would produce severe chattering in the observer gains, since the raw error signal e contains high-frequency sensor noise. The FFNN, by contrast, functions as a learned nonlinear low-pass filter with memory. Through the dynamics of the learning rate and momentum, the network learns the underlying trend of the error and provides the smooth, physically realizable gain trajectory required for stable hardware implementation.

Loss Function and Enhanced Update Rule. The discrepancy between the network output $\mathbf{y}_{net}$ and the target $\mathbf{t}$ is quantified by a Mean Squared Error loss:

$$L = \frac{1}{2} \|\mathbf{t} - \mathbf{y}_{net}\|_2^2 \tag{4.101}$$

Since both $\mathbf{t}$ and $\mathbf{y}_{net}$ are confined to the same $(0, 1)$ range, the resulting gradients are well-scaled and numerically consistent. The update rule incorporates L2 regularization, a momentum term $\mu = 0.9$, and a decaying learning rate:

$$\mathbf{v}_{k+1} = \mu\, \mathbf{v}_k + \eta_k \frac{\partial L}{\partial \mathbf{W}_k}, \qquad \mathbf{W}_{k+1} = \mathbf{W}_k - \mathbf{v}_{k+1} \tag{4.102}$$

where $\mathbf{v}$ is the momentum vector. A first-order IIR smoothing filter with coefficient $\alpha = 0.8$ is applied as a final output step to suppress residual high-frequency fluctuations in the tuned parameters:

$$\mathbf{\Theta}(k) = \alpha\, \mathbf{\Theta}(k-1) + (1 - \alpha)\, \mathbf{\Theta}_{scaled}(k) \tag{4.103}$$

where $\mathbf{\Theta}_{scaled}(k) = [\theta_p(k),\, \theta_y(k)]^T$ is the scaled output at the current step. The complete integrated control loop is depicted in Figure 4.2, and the computational steps are formally presented in Algorithm 1.

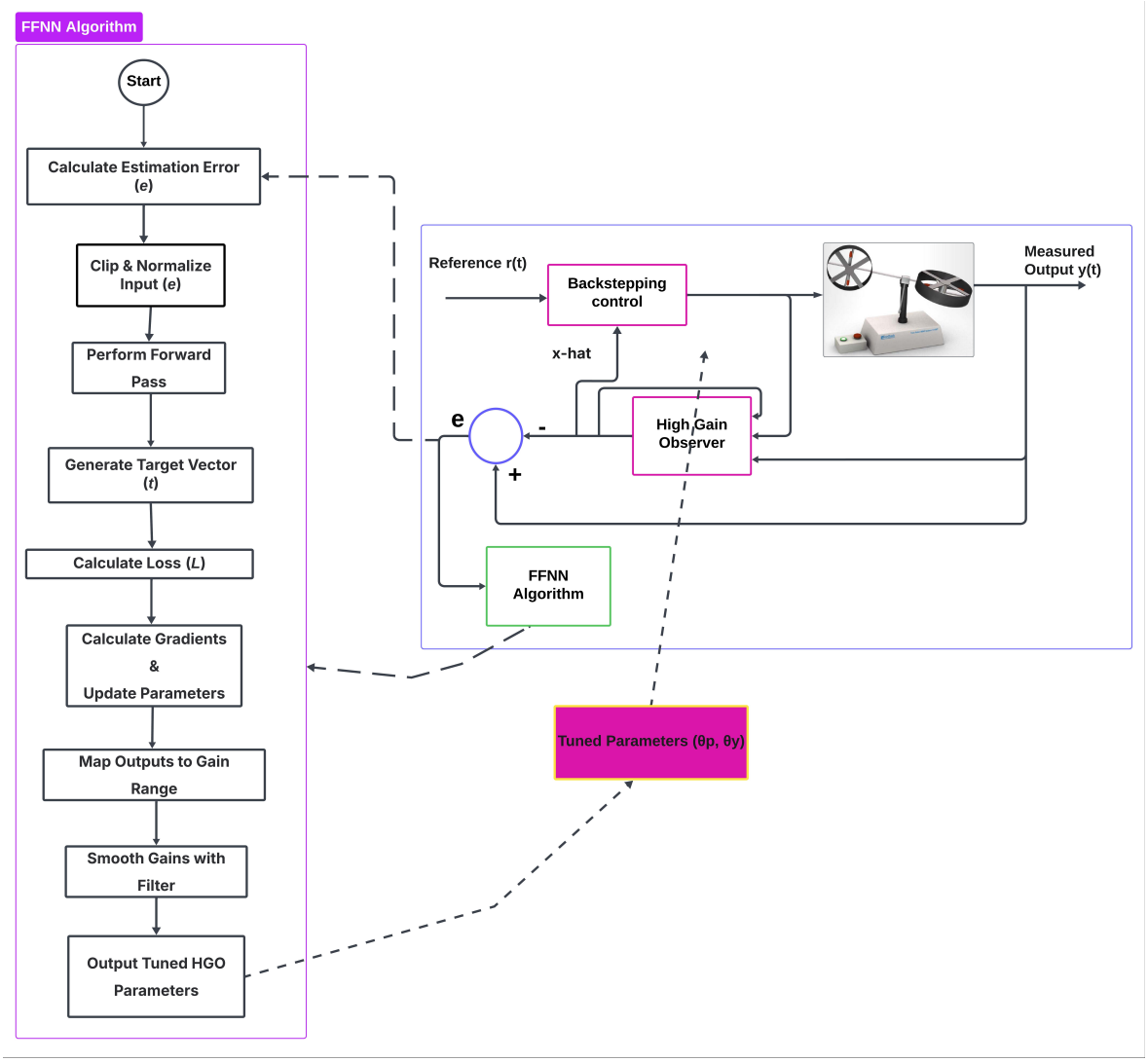

Figure 4.2: The FFNN-based adaptive control system architecture.
The diagram illustrates the complete system integration, divided into two conceptual parts. On the left, the flowchart details the algorithmic logic of the FFNN tuner, from the initial calculation of the ITAE estimation error e to the generation of the final smoothed HGO gain parameters. On the right, the block diagram depicts the overall control loop, showing the FFNN tuner receiving the observation error $e = \mathbf{y}(t) - \mathbf{C}\hat{\mathbf{x}}$, providing the tuned gains (θ_p, θ_y) to the HGO, which supplies the backstepping controller with the state estimates $\hat{\mathbf{x}}$ required for control actuation.

Algorithm 1 Online FFNN-Based HGO Tuning Strategy

Require: Raw ITAE error $e(k)$, Previous gains $\mathbf{\Theta}(k-1)$, Network parameters ($\mathbf{W}_{xh}, \mathbf{b}_h, \mathbf{W}_{hy}, \mathbf{b}_y, \mathbf{v}$), Learning rate η_k

Ensure: Tuned HGO parameters $\mathbf{\Theta}(k) = [\theta_p(k),\ \theta_y(k)]^T$

▷ *Input Pre-processing*

$e_c \leftarrow \text{clip}(e(k),\ -10,\ 10)$

$x \leftarrow \tanh(e_c/2)$

▷ *Forward Propagation*

$\mathbf{h} \leftarrow \tanh(\mathbf{W}_{xh}\, x + \mathbf{b}_h)$

$\mathbf{y}_{net} \leftarrow \sigma(\mathbf{W}_{hy}\, \mathbf{h} + \mathbf{b}_y)$

▷ *Online Learning*

$\mathbf{t} \leftarrow \left[\dfrac{1+\tanh(e_c)}{2},\ \dfrac{1+\tanh(e_c)}{2}\right]^T$

$L \leftarrow \frac{1}{2}\left\|\mathbf{t} - \mathbf{y}_{net}\right\|^2$

Compute gradients including L2 regularization

Update momentum vectors $\mathbf{v}$ and parameters using:

$$\mathbf{v}_{k+1} = \mu\, \mathbf{v}_k + \eta_k \frac{\partial L}{\partial \mathbf{W}_k}, \quad \mathbf{W}_{k+1} = \mathbf{W}_k - \mathbf{v}_{k+1}$$

▷ *Output Processing*

$\theta_p \leftarrow 1 + 4 \cdot y_{net,1}$

$\theta_y \leftarrow 0.5 + 1.5 \cdot y_{net,2}$

$\mathbf{\Theta}_{scaled}(k) \leftarrow [\theta_p,\ \theta_y]^T$

$\mathbf{\Theta}(k) \leftarrow 0.8 \cdot \mathbf{\Theta}(k-1) + 0.2 \cdot \mathbf{\Theta}_{scaled}(k)$

return $\mathbf{\Theta}(k)$

4.8.4 RBFNN Architecture, Gaussian Basis Functions, and Mathematical Formulation

The RBFNN-based tuner presented in this section constitutes an original contribution published by the author in [108]. The following develops the full implementation details and theoretical justification within the thesis context.

To provide a comprehensive comparative analysis against the FFNN-based strategy, a second intelligent tuner was developed using the Radial Basis Function Neural Network architecture. Its defining characteristic, the localized response of each hidden neuron, stands in direct contrast to the FFNN's global approximation strategy. For the TRMS, where the gain adaptation requirements differ substantially between the large-error transient phase and the small-error steady-state phase, this locality property offers a potential advantage in modeling complex, region-dependent nonlinear behavior.

Network Architecture

The network is organized as a three-layer structure: one input neuron, a hidden RBF layer with $N = 7$ Gaussian neurons, and a linear output layer with two neurons corresponding to the two HGO gain parameters θ_p and θ_y. The complete architecture is illustrated in Figure 4.3.

Input Normalization. The raw ITAE estimation error $e(k)$ is normalized before being passed to the network:

$$x_{in}(k) = \tanh\left(\frac{e(k)}{2}\right) \tag{4.104}$$

This maps the error signal into the bounded range $(-1, 1)$, which is well matched to the domain of the Gaussian receptive fields.

Hidden Layer: RBF Activations. The activation of the i-th hidden neuron is governed by a Gaussian radial basis function centered at c_i with spread width σ_i:

$$\phi_i(k) = \exp\left(-\frac{\big(x_{in}(k) - c_i\big)^2}{2\sigma_i^2}\right), \quad i = 1, \ldots, N \tag{4.105}$$

The seven centers were initialized by uniform spacing over the range $[-2, 2]$, effectively covering the most dynamic region of the normalized input signal. All widths were initialized to $\sigma_i = 0.5$ to ensure adequate overlap between adjacent receptive fields, which promotes smooth function approximation across the entire input domain.

Output Layer: Weighted Sum. The output layer is linear and computes the final network outputs as a weighted sum of the hidden activations:

$$y_{net,j}(k) = \sum_{i=1}^{N} w_{i,j}\, \phi_i(k), \quad j \in \{1, 2\} \tag{4.106}$$

where $w_{i,j}$ is the weight connecting the i-th hidden neuron to the j-th output neuron. Since the output layer is linear, $y_{net,j}$ is unbounded and requires a subsequent squashing and scaling stage, as described below.

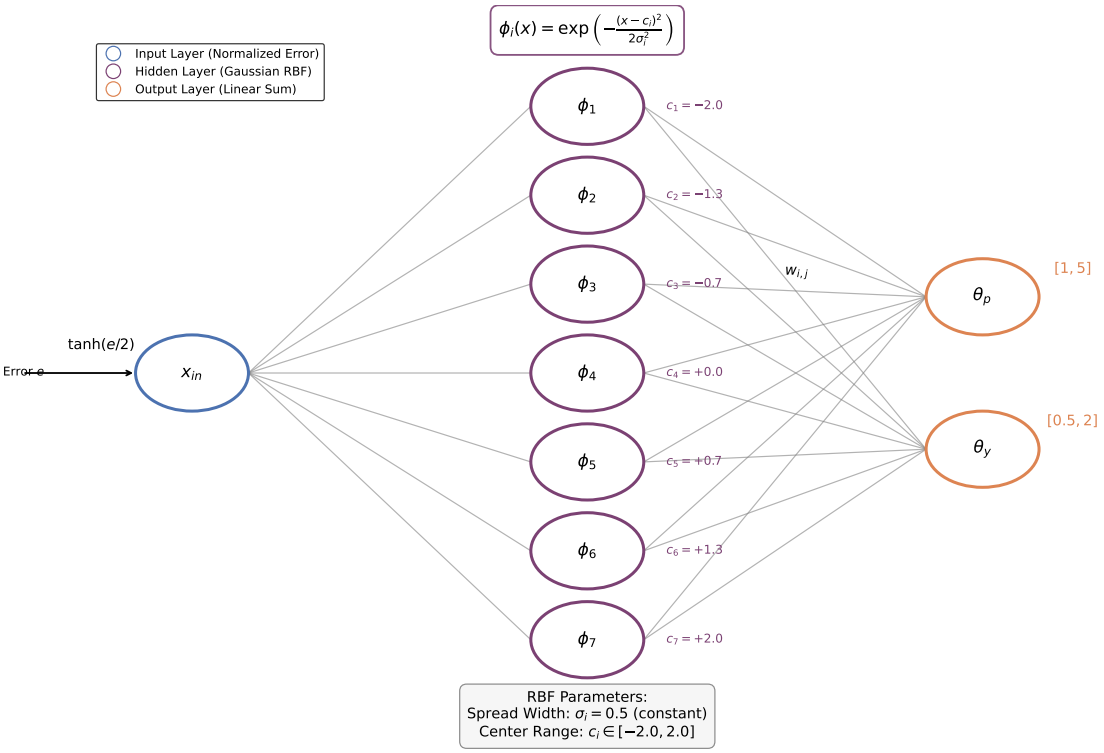

Figure 4.3: Detailed architecture of the RBF Neural Network used for online HGO gain tuning. The network receives the normalized tracking error x_{in} as its single input, processes it through $N = 7$ Gaussian hidden neurons with uniformly spaced centers, and produces the raw adaptive observer gains for pitch (θ_p) and yaw (θ_y) at the linear output layer.

Output Scaling and Smoothing

Since the output layer is linear, the raw values $y_{net,j}$ can grow without bound during online learning. To guarantee that the observer gains remain within a safe and physically meaningful range on the hardware, the raw outputs are first squashed through a tanh function and then linearly scaled to the experimentally determined operational envelopes:

$$\theta_{p,raw}(k) = 1 + 4\left(\frac{\tanh\big(y_{net,1}(k)\big) + 1}{2}\right) \in [1,\ 5] \tag{4.107}$$

$$\theta_{y,raw}(k) = 0.5 + 1.5\left(\frac{\tanh\big(y_{net,2}(k)\big) + 1}{2}\right) \in [0.5,\ 2] \tag{4.108}$$

A discrete-time first-order IIR low-pass filter is applied as the final processing step to prevent abrupt changes in the observer dynamics:

$$\boldsymbol{\Theta}(k) = 0.8\,\boldsymbol{\Theta}(k-1) + 0.2\,\boldsymbol{\Theta}_{raw}(k) \tag{4.109}$$

where $\boldsymbol{\Theta} = [\theta_p,\ \theta_y]^T$. The smoothing coefficient 0.8 was selected to adequately attenuate high-frequency gain fluctuations while preserving the network's ability to respond to genuine

changes in system dynamics.

The complete integration of the RBFNN tuner within the closed-loop system is illustrated in Figure 4.4.

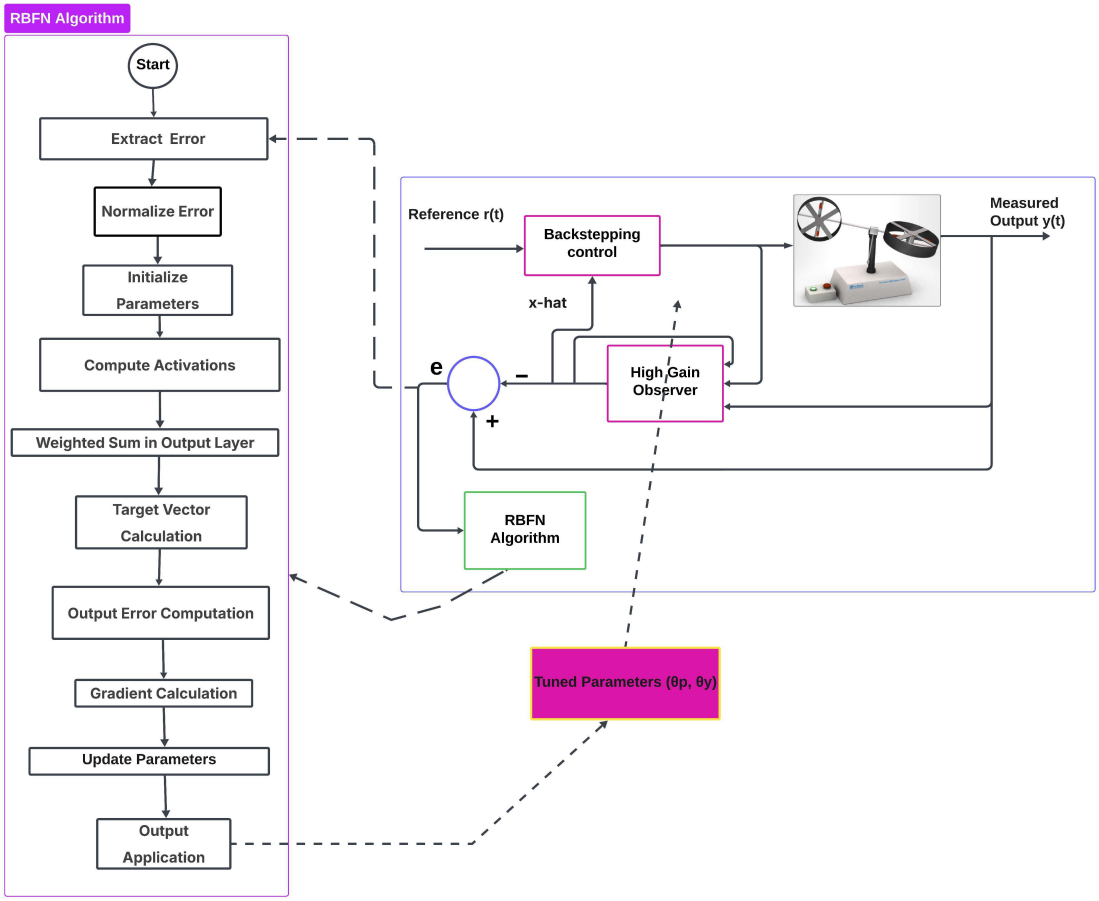

Figure 4.4: Flowchart of the RBFNN-based adaptive HGO tuning algorithm and its integration with the TRMS control loop. The signal $e = \mathbf{y}(t) - \mathbf{C}\hat{\mathbf{x}}$ at the summing junction drives the ITAE computation, and the RBFNN tuner supplies the smoothed gain parameters $(\theta_p,\ \theta_y)$ to the High-Gain Observer.

Online Learning Algorithm

In contrast to the classical hybrid training approach often associated with RBFNNs, in which centers are determined offline by clustering and only the output weights are subsequently trained, this work implements a fully online gradient-based approach. This allows continuous adaptation of all three parameter sets simultaneously: the output weights $w_{i,j}$, the neuron centers c_i, and the widths σ_i. Full online adaptation is indispensable for a real-time control application where the system operating region shifts continuously and no representative offline dataset is available.

Target Vector. The learning process is driven by a dynamically generated target vector. Unlike the FFNN tuner, whose sigmoid output layer constrains $\mathbf{y}_{net}$ to the normalized range $(0, 1)$

and therefore uses a normalized target, the RBFNN output layer is linear and unbounded. The target vector is therefore defined to reflect the desired gain magnitudes directly, scaled to be consistent with the operational envelopes in Equations (4.107) and (4.108). The design principle is the same as for the FFNN: increase the observer gain when the estimation error is large, and reduce it when the error is small:

$$\mathbf{t}(k) = \begin{bmatrix} 1 + \tanh\big(e(k)\big) \\ 0.5\big(1 + \tanh\big(e(k)\big)\big) \end{bmatrix} \tag{4.110}$$

The first component targets the pitch gain θ_p over the range $(0, 2)$, which maps to the operational envelope $[1, 5]$ through (4.107). The second component, scaled by 0.5, targets the yaw gain θ_y and maps to the narrower envelope $[0.5, 2]$ through (4.108).

Cost Function and Gradient Derivation. The instantaneous cost function is:

$$J(k) = \frac{1}{2} \sum_{j=1}^{2} \big(\mathbf{t}_j(k) - y_{net,j}(k)\big)^2 \tag{4.111}$$

Applying the chain rule yields the gradient terms for each parameter set. For the output weights:

$$\Delta w_{i,j} = \big(\mathbf{t}_j - y_{net,j}\big) \cdot \phi_i \tag{4.112}$$

For the neuron centers:

$$\Delta c_i = \left(\sum_{j=1}^{2} \big(\mathbf{t}_j - y_{net,j}\big) \cdot w_{i,j} \right) \cdot \phi_i \cdot \frac{x_{in} - c_i}{\sigma_i^2} \tag{4.113}$$

For the neuron widths:

$$\Delta \sigma_i = \left(\sum_{j=1}^{2} \big(\mathbf{t}_j - y_{net,j}\big) \cdot w_{i,j} \right) \cdot \phi_i \cdot \frac{(x_{in} - c_i)^2}{\sigma_i^3} \tag{4.114}$$

Parameter Update Rules. All parameters are updated at each time step using gradient descent with a fixed learning rate $\eta = 0.01$:

$$w_{i,j}(k+1) = w_{i,j}(k) + \eta \cdot \Delta w_{i,j} \tag{4.115}$$

$$c_i(k+1) = c_i(k) + \eta \cdot \Delta c_i \tag{4.116}$$

$$\sigma_i(k+1) = \max\big(\sigma_i(k) + \eta \cdot \Delta \sigma_i,\ 0.1\big) \tag{4.117}$$

A lower bound of 0.1 is enforced on σ_i to prevent the denominators σ_i^2 and σ_i^3 in Equations (4.113) and (4.114) from approaching zero, which would cause numerical blow-up in the gradient computation. This value is five times smaller than the initial width $\sigma_i = 0.5$, en-

suring that the constraint remains inactive under normal operating conditions and is engaged only as a safeguard against the pathological collapse of a receptive field. As the observation error e approaches zero, the gradients diminish naturally and all network parameters converge to stable values. The complete computational procedure is presented in Algorithm 2.

Algorithm 2 Online HGO Gain Tuning via RBF Neural Network

Require: Estimation error $e(k)$, Previous gains $\boldsymbol{\Theta}(k-1)$, RBF parameters $\{c_i, \sigma_i, w_{i,j}\}$, Learning rate η

Ensure: Tuned HGO parameters $\boldsymbol{\Theta}(k) = [\theta_p(k),\, \theta_y(k)]^T$

▷ *Input Pre-processing*

$x_{in} \leftarrow \tanh(e(k)/2)$

▷ *Forward Pass*

for $i = 1$ **to** N **do**

$\quad \phi_i \leftarrow \exp\!\big(-(x_{in} - c_i)^2 \,/\, (2\sigma_i^2)\big)$

end for

for $j = 1$ **to** 2 **do**

$$y_{net,j} \leftarrow \sum_{i=1}^{N} w_{i,j}\, \phi_i$$

end for

▷ *Target Generation*

$$\mathbf{t} \leftarrow \big[1 + \tanh(e(k));\quad 0.5\,(1 + \tanh(e(k)))\big]^T$$

▷ *Gradient Computation*

$E_j \leftarrow \mathbf{t}_j - y_{net,j}$ **for** $j = 1, 2$

for $i = 1$ **to** N **do**

$$\Delta w_{i,j} \leftarrow E_j \cdot \phi_i \quad \forall\, j$$

$$\Delta c_i \leftarrow \left(\sum_j E_j\, w_{i,j}\right) \cdot \phi_i \cdot (x_{in} - c_i) \,/\, \sigma_i^2$$

$$\Delta \sigma_i \leftarrow \left(\sum_j E_j\, w_{i,j}\right) \cdot \phi_i \cdot (x_{in} - c_i)^2 \,/\, \sigma_i^3$$

end for

▷ *Parameter Update*

$w_{i,j} \leftarrow w_{i,j} + \eta \cdot \Delta w_{i,j}$

$c_i \leftarrow c_i + \eta \cdot \Delta c_i$

$\sigma_i \leftarrow \max\!\big(\sigma_i + \eta \cdot \Delta\sigma_i,\ 0.1\big)$

▷ *Output Scaling and Smoothing*

$\theta_p \leftarrow 1 + 4 \cdot 0.5\,\big(\tanh(y_{net,1}) + 1\big)$

$\theta_y \leftarrow 0.5 + 1.5 \cdot 0.5\,\big(\tanh(y_{net,2}) + 1\big)$

$\boldsymbol{\Theta}(k) \leftarrow 0.8\,\boldsymbol{\Theta}(k-1) + 0.2\,[\theta_p,\, \theta_y]^T$

return $\boldsymbol{\Theta}(k)$

4.8.5 Comparative Synthesis of the Two Tuning Architectures

The two adaptive gain tuners presented in this chapter share the same control objective and the same integration point within the closed-loop architecture, yet they pursue that objective through fundamentally different internal mechanisms, as summarized in Table 4.1. The FFNN tuner employs global activation functions and adapts only its weights and biases, re-

lying on the momentum-augmented backpropagation algorithm to learn a smooth nonlinear mapping from the aggregate observation error to the observer gain parameters. Its strength lies in its generalization capability across the full input range, which makes it well suited to tracking scenarios where the error signal varies continuously and over a wide dynamic range. The RBFNN tuner, by contrast, partitions the input space into localized receptive fields and adapts three parameter sets simultaneously: the output weights, the neuron centers, and the width parameters. This localized structure allows the network to assign distinct gain behaviors to the large-error transient phase and the small-error steady-state phase independently, without one region of the input space interfering with the other. The practical consequence of this distinction is that the RBFNN is expected to respond more sharply to abrupt changes in the observation error, while the FFNN is expected to produce smoother gain trajectories over extended tracking intervals. A rigorous quantitative assessment of these theoretical predictions, under identical experimental conditions and against the full set of performance metrics defined in Section 4.8.6, is provided in Chapter 5.

4.8.6 Performance Evaluation Framework

A single performance metric is insufficient to characterize the behavior of a complex nonlinear MIMO system such as the TRMS. The evaluation framework adopted here is structured around four orthogonal assessment categories: steady-state tracking accuracy, transient response quality, control effort economy, and robustness to real-world imperfections. This multi-faceted approach enables an unbiased comparison that moves beyond simple error metrics to evaluate the practical viability of each control strategy. All quantities are computed independently for the pitch (ψ) and yaw (φ) degrees of freedom.

Steady-State Tracking Accuracy

Root-Mean-Square Error (RMSE). The RMSE quantifies the overall magnitude of the tracking error over a discrete-time sequence of length N with sampling period T_s:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\big(y[k] - r[k]\big)^2} \tag{4.118}$$

The squaring of the error term makes the RMSE particularly sensitive to large intermittent deviations, which makes it an effective indicator of overall tracking fidelity.

Integral of Time-Weighted Absolute Error (ITAE). The ITAE criterion penalizes persistent errors by weighting the absolute deviation by elapsed time:

$$\text{ITAE} = T_s \sum_{k=1}^{N} k\, T_s \left| y[k] - r[k] \right| \tag{4.119}$$

The deliberate use of the ITAE metric for both performance assessment and as the scalar feedback signal to the AI tuners ensures direct and consistent alignment between the online adaptation objective and the offline evaluation criterion.

Transient Response Quality

Settling Time ($t_{s,2\%}$). The settling time is defined as the interval between the application of a reference step and the instant at which the response error $|y(t) - y_\infty|$ enters and remains within a $\pm 2\%$ band of the final steady-state value y_∞. Measurements are repeated for both positive and negative steps and subsequently averaged to mitigate directional asymmetries in the system response.

Percentage Overshoot (%OS).

$$\%\text{OS} = \frac{\max_k \left| y[k] \right| - y_\infty}{y_\infty} \times 100\% \tag{4.120}$$

This metric quantifies the trade-off between response speed and relative stability. Excessive overshoot can be detrimental in physical systems subject to hard angle limits.

Control Effort and Efficiency

Control Effort (CE). The energy expenditure of the actuators is quantified by aggregating the squared control action over the duration of the experiment:

$$\text{CE} = T_s \sum_{k=1}^{N} \big(u[k] \big)^2 \tag{4.121}$$

A controller that achieves high tracking performance with low control effort is considered superior, as it reduces the risk of actuator saturation and limits thermal loading on the motors.

Robustness Assessment

Noise Sensitivity. To evaluate robustness to imperfect measurements, band-limited white Gaussian noise with variance $\sigma_n^2 = 0.25\ (\text{deg})^2$ is superimposed on the encoder measurements. The controller's noise rejection capability is quantified by the Noise Amplification

Factor:

$$\Gamma = \frac{\text{RMSE}_{\text{noisy}}}{\text{RMSE}_{\text{noiseless}}} \tag{4.122}$$

A value of $\Gamma \approx 1$ indicates robust noise rejection, whereas $\Gamma \gg 1$ signals that the controller detrimentally amplifies measurement noise.

Parametric Uncertainty. Key inertia and damping parameters are varied by $\pm 20\%$ from their nominal values to assess performance degradation under model mismatch. This range represents a significant but physically plausible degree of uncertainty. The worst-case percentage increase in RMSE and ITAE across the entire uncertainty envelope is reported, providing a quantitative measure of each controller's guaranteed performance boundary.

Statistical Validation

All experiments involving stochastic elements are executed over five independent Monte Carlo runs, each with a distinct random noise seed. Mean values and 95% confidence intervals are computed and reported for every metric. This procedure ensures that the conclusions drawn are statistically significant rather than artifacts of a single-run result.

A consolidated summary table presenting the aggregated results for the three controllers under comparison is provided in Chapter 5 as Table 5.1.

4.9 Chapter Summary

This chapter has presented the complete design of the control and observation architecture proposed in this thesis. Three successive design layers were developed.

At the first layer, a decentralized backstepping controller was derived for both the pitch and yaw subsystems. Four control laws were obtained in closed form: regulation and trajectory tracking for each axis. Each law is accompanied by a composite Lyapunov function whose negative-definite time derivative formally establishes local asymptotic stability. Practical implementation provisions, including singularity avoidance, actuator saturation, and the analytical generation of reference derivatives, were addressed explicitly.

At the second layer, a High-Gain Observer was designed and applied to the TRMS to reconstruct the four unmeasured states required by the backstepping law. The stability of the integrated system was established through the nonlinear separation principle, confirming semi-global practical asymptotic stability of the complete closed-loop system. The inherent noise sensitivity of the fixed-gain HGO was identified as the central practical limitation of the baseline architecture.

At the third layer, two AI-based adaptive gain tuners were designed to resolve this limitation. The FFNN-based tuner learns a smooth nonlinear mapping from the aggregate observation error to the observer gain parameters, acting as an intelligent adaptive filter. The

RBFNN-based tuner achieves the same objective through a localized approximation strategy, providing a structurally distinct alternative whose comparative performance is evaluated in Chapter 5.

The performance evaluation framework defined in Section 4.8.6 provides the rigorous, multi-criteria basis on which all three architectures are compared in the following chapter.

Chapter 5

Results and Discussion

This chapter presents the experimental results for the controllers designed in Chapter 4. The primary objective is to validate the central claim of this thesis. We will show that the proposed AI-enhanced observers overcome the real-world limitations of the conventional fixed-gain approach.

The chapter is organized to tell a clear story. First, we present the performance of the baseline backstepping controller. We show its success in simulation and its critical failure during real-time tracking experiments. This establishes the problem we need to solve. Second, we present the successful results of the FFNN-based and RBFNN-based adaptive controllers. Finally, we provide a rigorous comparative analysis of all three schemes using the performance metrics established in Chapter 4.

5.1 Baseline Performance in Simulation

The initial phase of our experimental work focused on evaluating the baseline controller in an ideal environment. The controller, combining backstepping with a fixed-gain HGO, was tested within the MATLAB/Simulink environment using the full nonlinear model of the TRMS. The objective was to verify the theoretical correctness of the control law in a noise-free context before proceeding to real-world tests. The complete Simulink diagram for this test setup is shown in Figure 5.1.

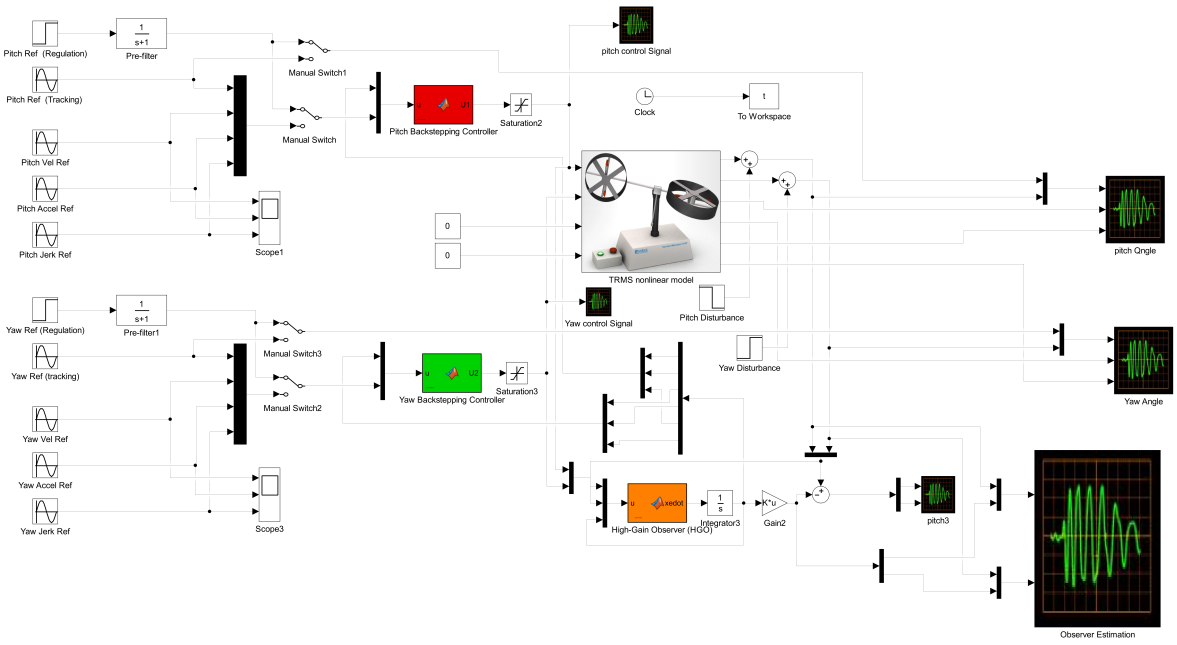

Figure 5.1: Simulink Implementation for the Regulation Test, Showing Disturbance Injection Points.

5.1.1 Performance in Regulation Mode

The first simulation scenario evaluates the controller's ability to perform setpoint regulation. Step inputs were used as the reference signals, with a target of 0.5 radians for both the pitch and yaw axes. To test the controller's robustness, simulated disturbances were manually injected at t = 40s for the pitch axis and at t = 60s for the yaw axis.

System Response and Disturbance Rejection

The primary result is the closed-loop tracking performance of the system. Figure 5.2 shows the measured pitch and yaw angles compared to their desired reference values.

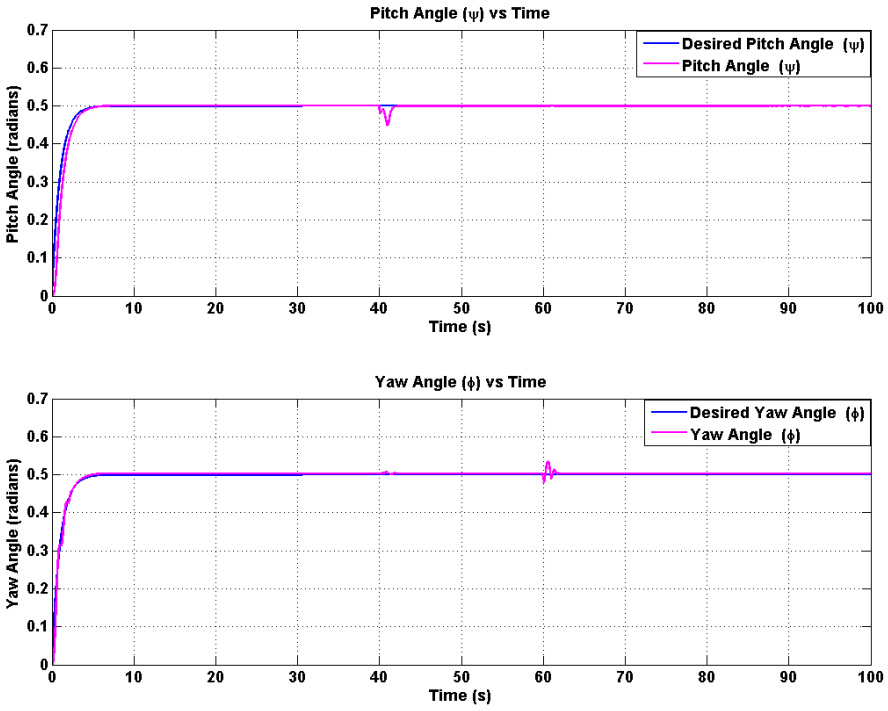

Figure 5.2: System Response in the Regulation Simulation (Top: Pitch Angle, Bottom: Yaw Angle).

The results demonstrate excellent regulation performance. The controller rapidly drives both the pitch and yaw angles to their respective setpoints. The settling time is very short, and the steady-state error is effectively zero. Crucially, the controller also shows excellent disturbance rejection. When disturbances are injected, the system exhibits a brief transient but immediately and robustly returns to the desired setpoint.

Observer Performance

For the backstepping controller to function correctly, the High-Gain Observer must provide accurate estimates of the system states. Figure 5.3 shows the performance of the observer during the regulation test.

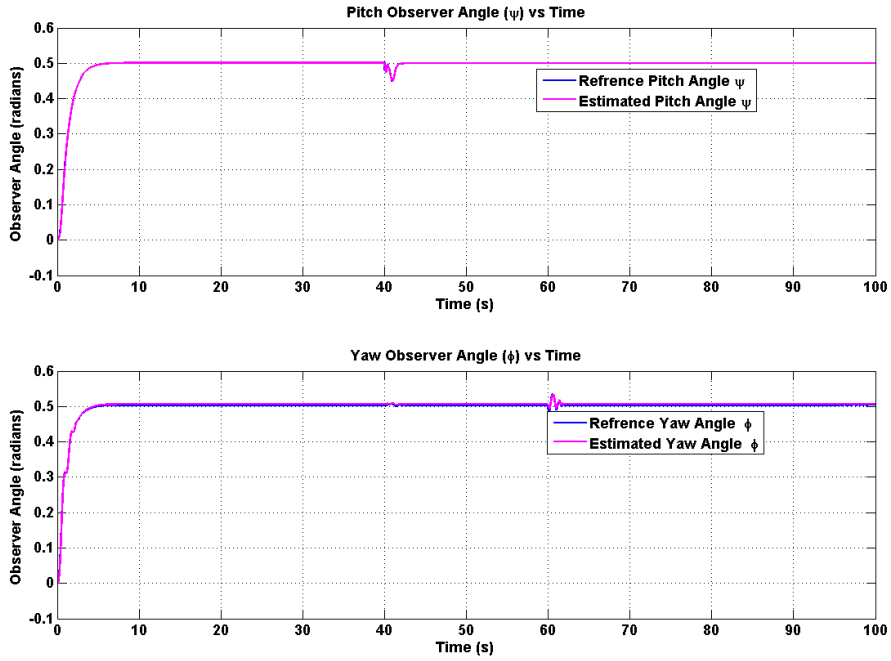

Figure 5.3: High-Gain Observer Performance in the Regulation Simulation.

The observer's performance is also nearly perfect. The estimated pitch and yaw angles converge to the true reference angles almost instantaneously. The estimation error becomes zero very quickly. This confirms that the fixed-gain HGO is effective in an ideal simulation environment.

Control Effort

Finally, we analyze the control signals generated by the backstepping controller. The control effort, shown in Figure 5.4, indicates the voltage being sent to the pitch and yaw motors.

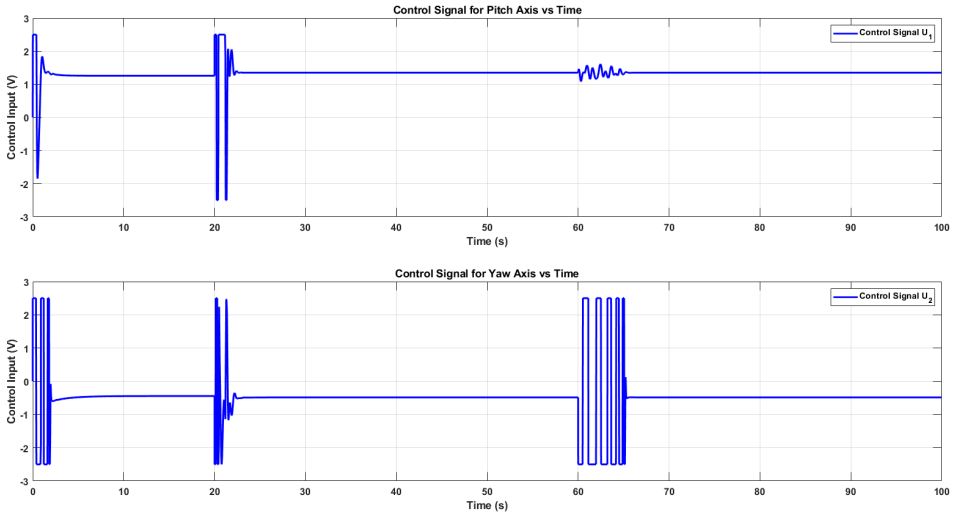

Figure 5.4: Control Signals Generated by the Backstepping Controller During the Regulation Simulation.

The control signals are well-behaved. An initial control action is required to lift the system to the setpoint. The controller then provides sharp, corrective pulses to reject the injected disturbances. In steady-state, the control effort is minimal, indicating an efficient controller. The signals are also smooth and free of chattering.

The collective results from these simulation tests validate that the baseline controller is theoretically sound and performs its function perfectly under ideal conditions.

5.1.2 Performance in Trajectory Tracking Mode

While setpoint regulation demonstrates a controller's stability, a more demanding evaluation is its ability to follow a continuously varying reference signal. This scenario, known as trajectory tracking, rigorously tests the controller's dynamic response capabilities and its ability to handle time-varying system states. To conduct this test, the Simulink model was reconfigured to switch the reference inputs from step functions to sinusoidal signal generators, as shown in the updated implementation diagram in Figure 5.5.

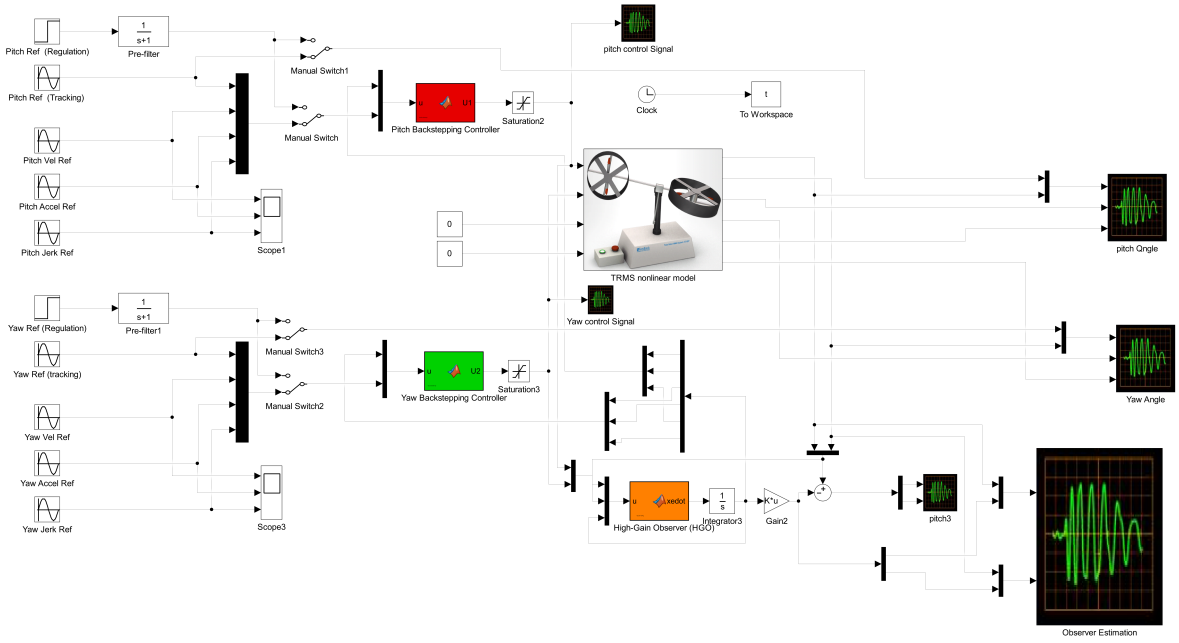

Figure 5.5: Simulink implementation for the trajectory tracking test, with sinusoidal reference signals selected.

System Response and Tracking Fidelity

The primary result for this test is a direct measure of how well the system's output follows the desired sinusoidal trajectory. Figure 5.6 presents the tracking performance for both the pitch and yaw axes, showing the reference and the system's actual response overlaid.

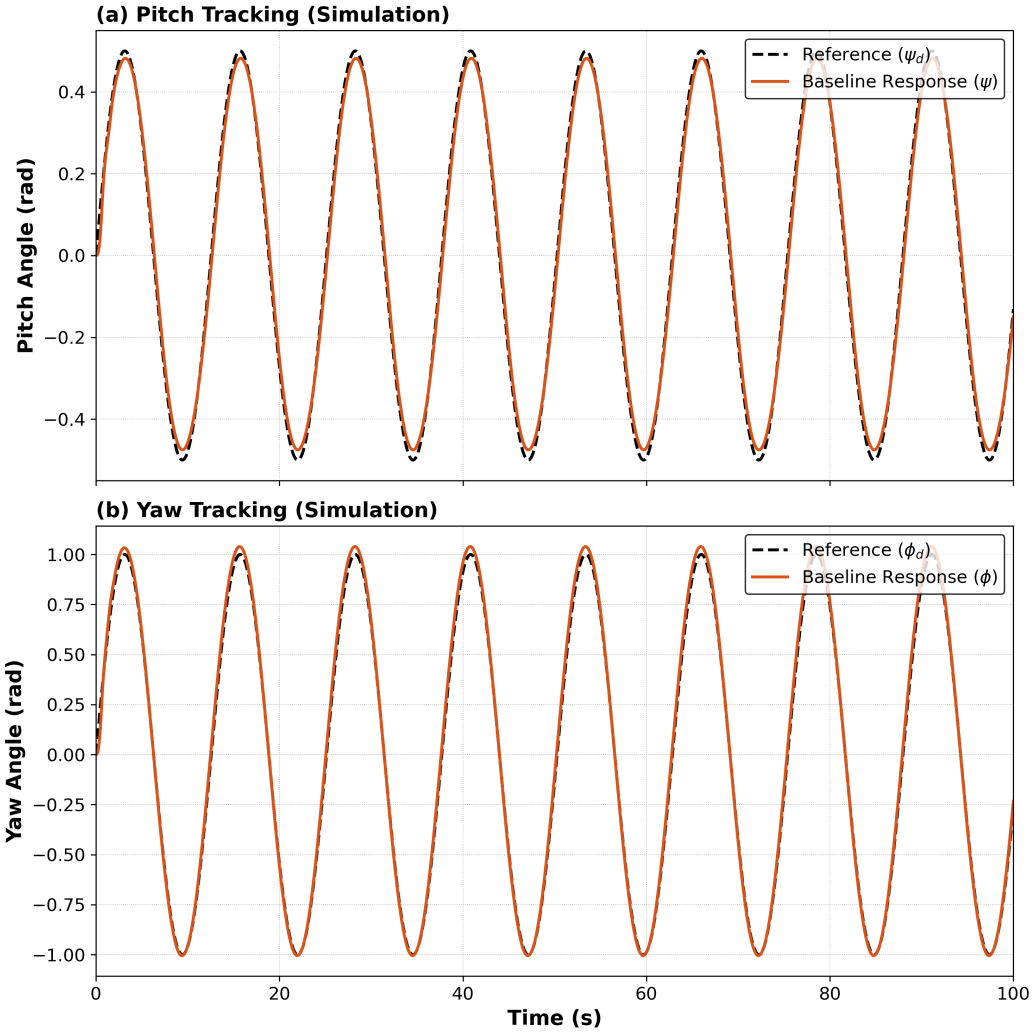

Figure 5.6: Simulation results for sinusoidal trajectory tracking.

The tracking performance achieved in the simulation is nearly flawless. For both axes, the output signal almost perfectly overlaps with its reference trajectory, indicating no discernible phase lag or amplitude attenuation. This result is significant because it demonstrates that, in a theoretical, noise-free environment, the baseline backstepping controller possesses the necessary agility and precision for high-performance dynamic tracking.

Observer Performance During Dynamic Tracking

The high-fidelity tracking seen above is critically dependent on the HGO's ability to provide accurate state estimates, not just at a fixed point, but while the system states are in constant motion. This is a non-trivial task for any observer. Figure 5.7 assesses the observer's performance during this dynamic task by comparing the true system states from the model to the observer's estimates.

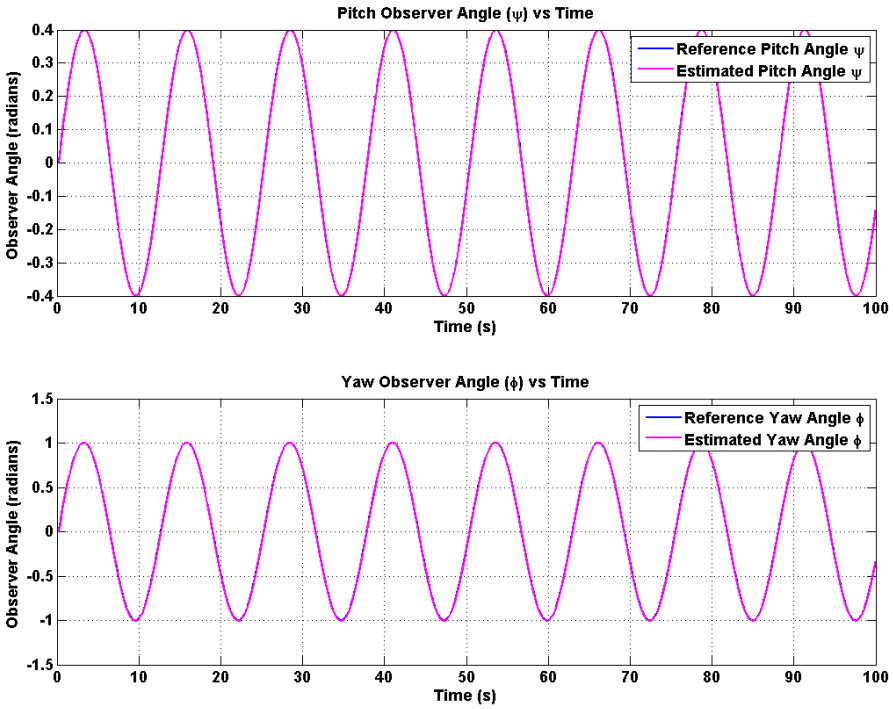

Figure 5.7: Performance of the High-Gain Observer during the trajectory tracking simulation. The estimated states (e.g., pitch and yaw angles) accurately follow the time-varying true states.

The observer continues to perform with exceptional accuracy, even under these dynamic conditions. The estimated states are shown to converge to and track the time-varying true states with negligible error. This confirms that the fixed-gain HGO, within the idealized context of the simulation, possesses the necessary dynamic bandwidth to supply the controller with the high-quality, real-time state information required for demanding tracking tasks.

Analysis of Control Effort

Finally, we examine the control signals required to achieve this precision. The control effort is a critical practical consideration, as excessively aggressive or noisy control signals can lead to actuator saturation, increased mechanical wear, and are generally undesirable for physical implementation. The control voltages applied to the pitch and yaw motors are shown in Figure 5.8.

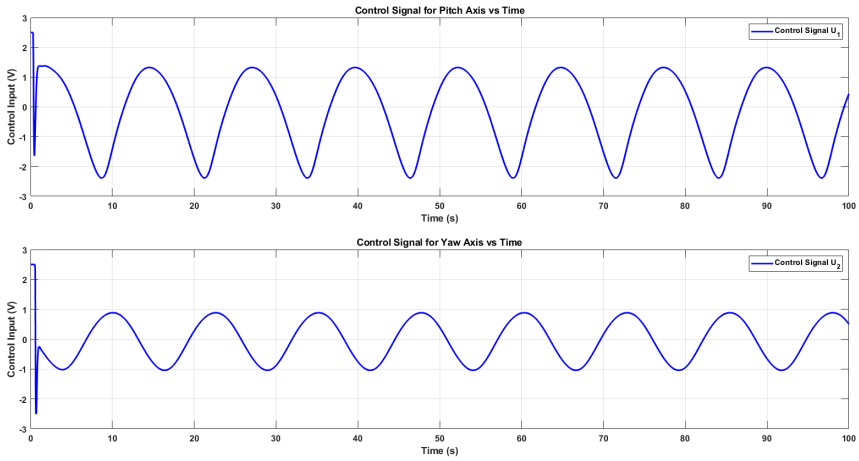

Figure 5.8: Control signals generated by the backstepping controller during the sinusoidal trajectory tracking simulation.

The resulting control signals are smooth, continuous, and sinusoidal, directly reflecting the dynamic nature of the reference trajectory. Importantly, the magnitude of the signals remains well within the typical actuator limits of the TRMS (e.g., ± 2.5V), indicating that the controller is not only effective but also efficient, operating without excessive energy expenditure. The complete absence of chattering or high-frequency oscillations in the control signals further validates that the baseline controller, in simulation, represents an optimal and well-behaved solution.

The successful outcomes from both the regulation and tracking simulations provide a strong and comprehensive validation of the controller's theoretical design. These results form a critical benchmark of ideal performance. The subsequent sections will serve as a stark contrast, presenting the results from real-world hardware experiments where this ideal behavior is fundamentally challenged.

5.2 Real-Time Experimental Setup

Having established the controller's ideal performance in simulation, the investigation now transitions to its evaluation on the physical Twin-Rotor MIMO System. This section details the hardware and software architecture used to implement the control algorithms in a real-time, hardware-in-the-loop (HIL) configuration. A clear understanding of this setup is essential for interpreting the experimental results that follow.

5.2.1 TRMS Hardware Components

A concise description of the key hardware components and the signal flow is necessary to understand the experimental arrangement.

Power On/Off Box A small hand-held enclosure carries two push-buttons: a green one that energizes the motor amplifiers and a red one that removes power completely. Its sole purpose is to protect the motors against accidental overload. The box is cabled to the rear panel of the TRMS base.

TRMS Base Panel The rear panel hosts the primary connectors for power and control signals. This includes the DIN socket for the power box, a 110/220 V selector switch, and the main power inlet. Three additional connectors route measurement and control signals:

- `CN1` (40-way ribbon): Digital position data from the two 16-bit optical encoders.
- `CN2` (20-way ribbon): Analogue voltage commands sent to the two motors.
- `CN3` (20-way ribbon): Analogue tachometer feedback received from each motor.

SCSI Adapter Box An interface box is used to adapt the low-level TRMS signals to the 68-pin SCSI cable required by the data-acquisition card. The box separates the signals into three groups, corresponding to the digital encoder outputs, the analogue motor-drive voltages, and the analogue tachometer inputs.

Data-Acquisition Hardware: Advantech PCI-1711 The link between the host PC and the TRMS hardware is an Advantech PCI-1711 data acquisition card. This is a general-purpose board that occupies a standard PCI slot. Its key features for this work are:

- Two 12-bit analogue outputs (DAC) to generate the reference voltages for the TRMS motors.
- 16 digital input channels to read the optical encoders for the pitch and yaw angles.
- 16 configurable analogue inputs (ADC) to acquire signals such as tachometer feedback.
- A maximum aggregate sampling rate of 100 kHz.

A schematic overview of the complete PC-to-TRMS connection is provided in Figure 5.9.

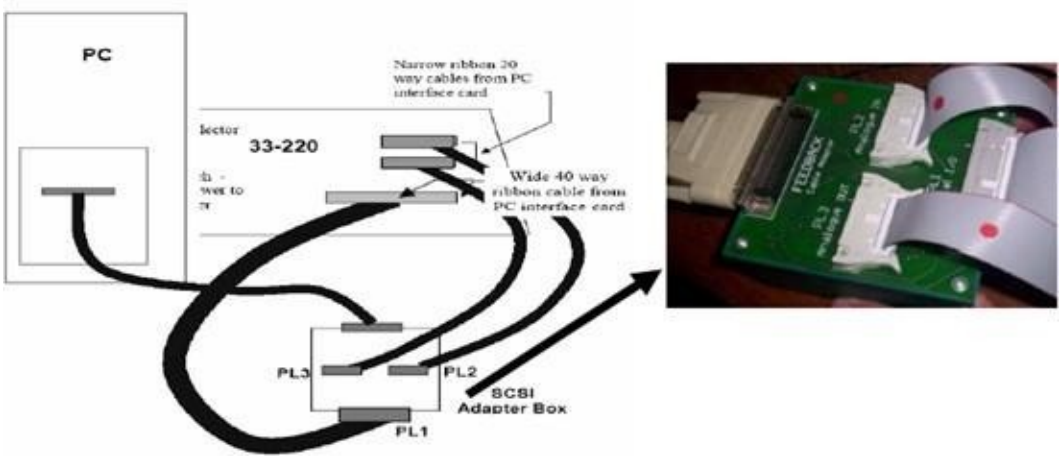

Figure 5.9: PC-TRMS connection overview, showing the role of the SCSI adapter box.

5.2.2 Real-Time Execution Model (Hardware-in-the-Loop)

The implementation follows a hardware-in-the-loop (HIL) philosophy. The controller that was validated in simulation is executed directly, interacting with the physical plant in real time. The development PC acquires sensor data via the PCI-1711, computes the control voltages, and applies them back to the motors. The transition from non-real-time simulation to real-time operation is straightforward. The Simulink model is augmented with the appropriate I/O blocks for the PCI-1711, compiled into a real-time executable, and downloaded to the target PC. Figure 5.10 illustrates the data flow for this computer-based control scheme.

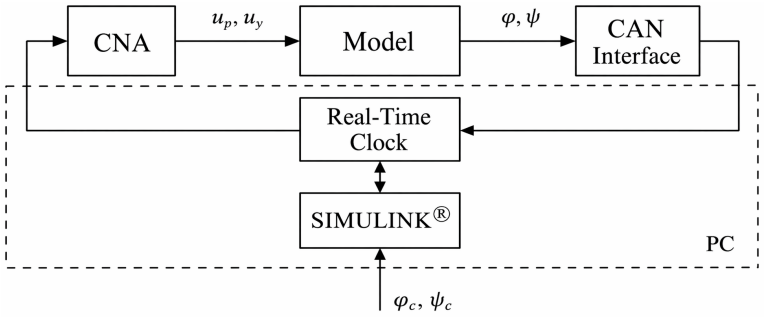

Figure 5.10: Principle of the hardware-in-the-loop control scheme.

5.3 Real-Time Experimental Performance

With the simulation benchmarks established, we moved to the decisive phase of testing: deploying the baseline controller on the physical TRMS hardware. This is the true test, where the ideal controller must confront the unmodeled dynamics, friction, and sensor noise of a real system. The experiments were run using the hardware-in-the-loop (HIL) setup as detailed in Section 5.2, with the Simulink model configured for real-time I/O (Figure 5.11).

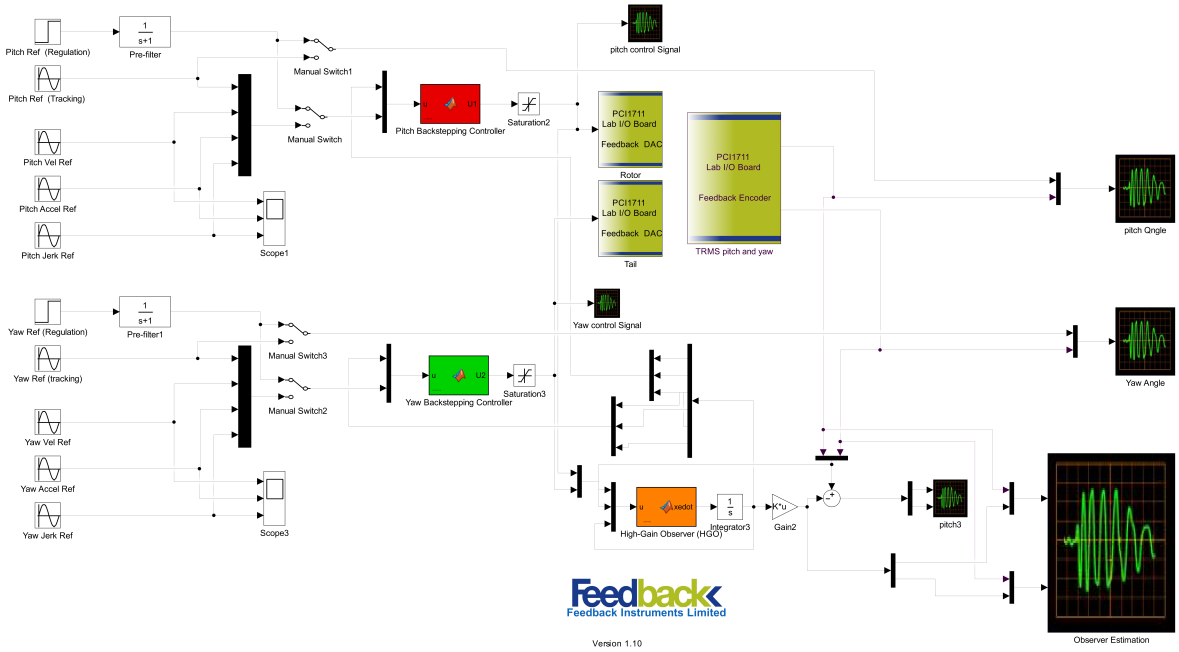

Figure 5.11: Simulink Diagram for Hardware-in-the-Loop (HIL) Real-Time Implementation.

5.3.1 Performance in Real-Time Regulation

The first hardware test was a direct replication of the regulation scenario. The goal was to confirm the controller's most basic competency: can it stabilize the physical system at a fixed setpoint and hold it there against real-world imperfections? This serves as the foundational validation for the entire experimental setup.

System Response and Attenuation of Cross-Coupling

The real-time response of the system, shown in Figure 5.12, confirms that the controller is indeed capable of this fundamental task.

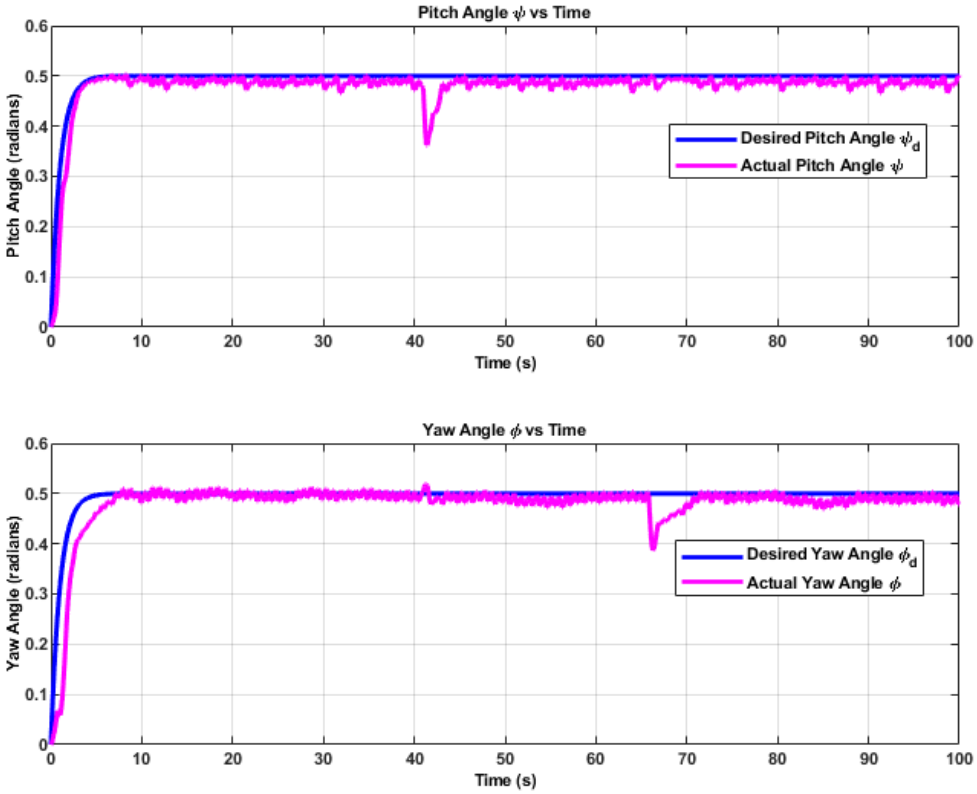

Figure 5.12: Real-Time Experimental Results for the Baseline Controller in Regulation Mode.

Both the pitch and yaw axes are brought to their setpoints with a clean, well-damped response. The controller effectively stabilizes the system, though some minor steady-state oscillation due to sensor noise and mechanical friction is now visible an expected artifact of a physical rig that was absent in the pristine simulation.

The controller's handling of disturbances in these plots provides a powerful validation of its decoupling performance. When a disturbance is applied to the pitch axis at t=40s, the corresponding reaction on the yaw axis is minimal a slight, well-damped wiggle rather than a large, destabilizing swing. This is particularly important given the physical asymmetry of the TRMS, where the coupling effect from the main rotor (pitch) onto the yaw axis is far more pronounced than the reverse. The ability of the backstepping law to significantly attenuate this dominant coupling effect is a strong demonstration of the controller's design. The reciprocal is also true: when the yaw axis is disturbed at t=65s, the pitch axis shows only a very minor transient. While not a perfect, theoretical decoupling, the controller successfully treats each axis as a largely independent system, confirming that a key theoretical advantage of the design translates effectively to the hardware.

Observer Performance in a Noisy Environment

This successful control is built entirely on the state estimates from the High-Gain Observer, whose real-world performance is shown in Figure 5.13.

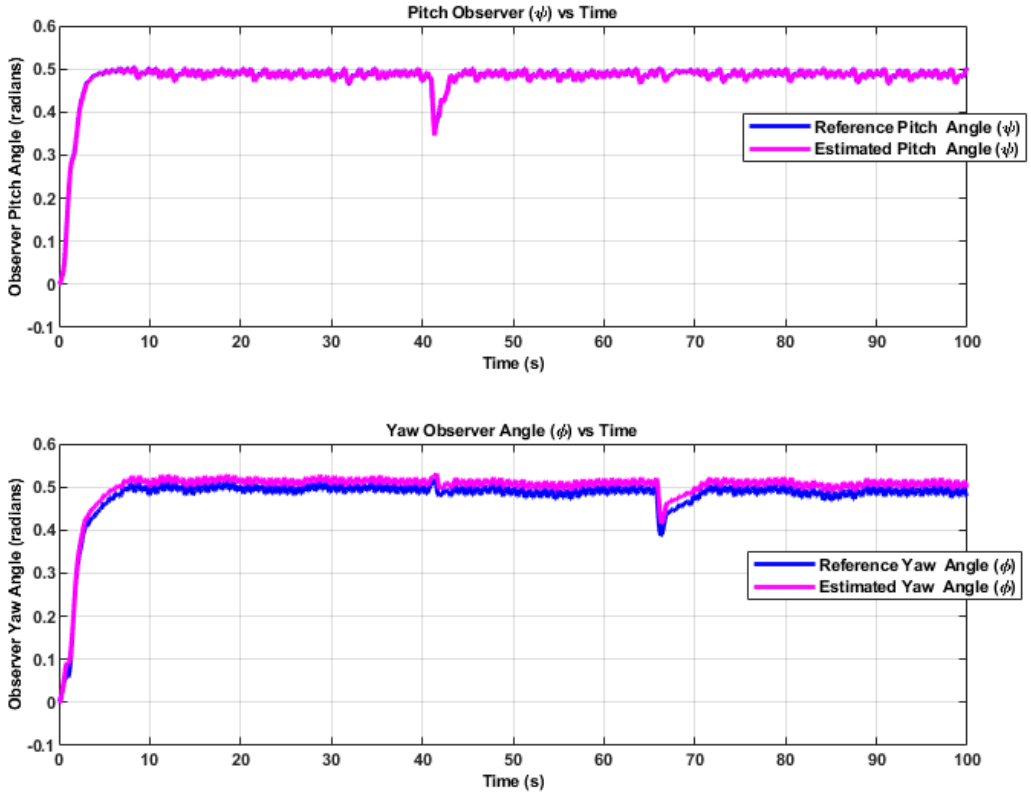

Figure 5.13: High-Gain Observer Performance During Real-Time Regulation.

Here, the inherent trade-off of the fixed-gain approach becomes starkly apparent. While the estimates (magenta) track the real measurements (blue) well enough to ensure stability, they are clearly contaminated with persistent, high-frequency noise. This is the HGO doing exactly what it is designed to do applying high gain to ensure fast convergence but in doing so, it inevitably amplifies the measurement noise from the encoders. For the relatively simple task of holding a fixed position, these noisy estimates are still usable, but they come at a cost.

Analysis of Control Effort and Decoupling Intent

The cost of using these noisy estimates is laid bare in the control effort plot, Figure 5.14. This plot also provides the most definitive proof of the controller's decoupling success by revealing its intent.

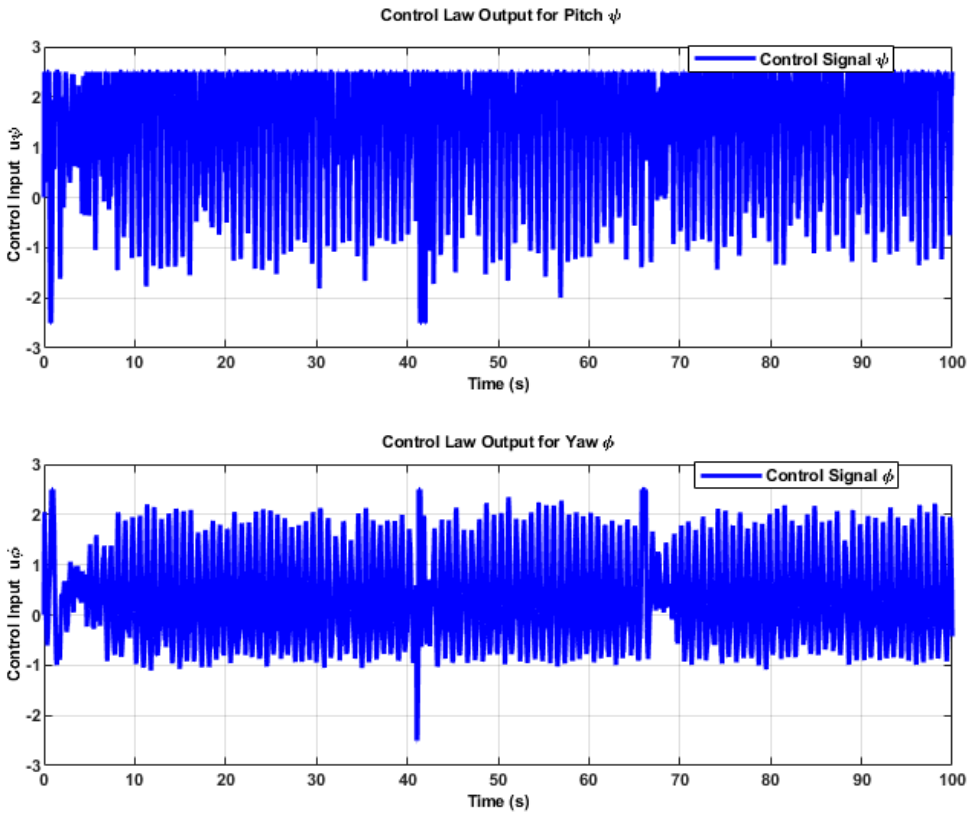

Figure 5.14: Control Effort Generated by the Baseline Controller During Real-Time Regulation.

Instead of the clean commands from the simulation, the real-world control signals are plagued by aggressive, high-frequency chattering. This is the backstepping law faithfully reacting to every noise-induced fluctuation it receives from the observer. While the system's inertia filters this effect, the action itself is inefficient and unsustainable.

More profoundly, however, this plot shows the controller's internal logic. At t=40s, the Pitch Control Law issues a large, sharp command to counteract the disturbance. At that exact moment, the Yaw Control Law shows no corresponding action; it simply continues its baseline chattering, ignorant of the event on the other axis. Likewise, at t=65s, the Yaw Control Law acts decisively, while the Pitch Control Law remains disengaged. This demonstrates that the decoupling is not an accident of physics, but an explicit function of the control algorithm itself. The controller is actively isolating the axes at the command level.

In summary, the regulation test offers a nuanced conclusion. The baseline controller is stable and effective, and its decoupling works precisely as designed, significantly attenuating the cross-couplings of the physical plant. However, its reliance on a fixed-gain HGO makes it susceptible to noise, leading to inefficient and aggressive control action. This qualified success is the necessary prelude to the final test, where the controller's fragility is exposed under the demanding conditions of real-time tracking.

5.4 Real-Time Trajectory Tracking: A Critical Performance Failure

The successful performance of the baseline controller in regulation provided a crucial, yet ultimately misleading, sense of its real-world viability. The true test and the one most central to this thesis is the controller's ability to perform high-fidelity trajectory tracking on the physical hardware. This scenario forces the controller to confront the full spectrum of real-world challenges simultaneously: persistent sensor noise, unmodeled friction, and the inevitable mismatch between the design model and the physical plant, all while the system is in continuous, dynamic motion.

The experimental setup for this test directly mirrors the tracking simulation, with the reference inputs configured as sinusoidal trajectories. The real-time Simulink implementation is shown in Figure 5.15.

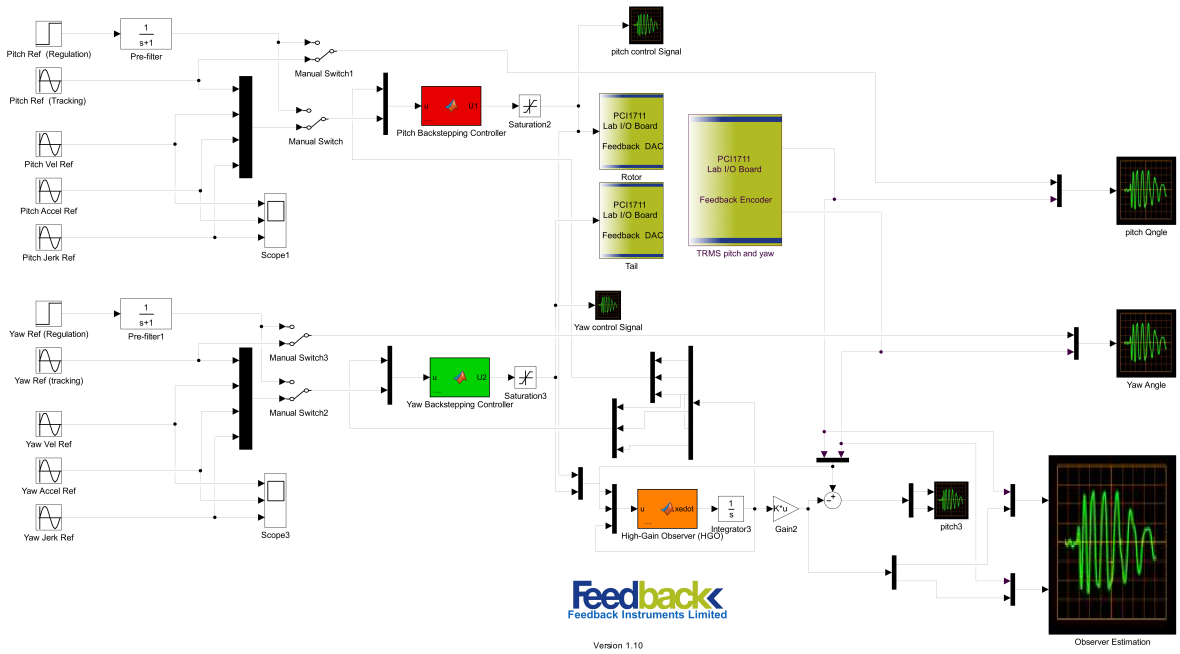

Figure 5.15: Simulink diagram for the real-time trajectory tracking experiment.

5.4.1 Analysis of the Tracking Performance Failure

The results of the real-time tracking experiment, presented in Figure 5.16, stand in stark contrast to the flawless performance seen in simulation. The outcome is an unambiguous and catastrophic failure.

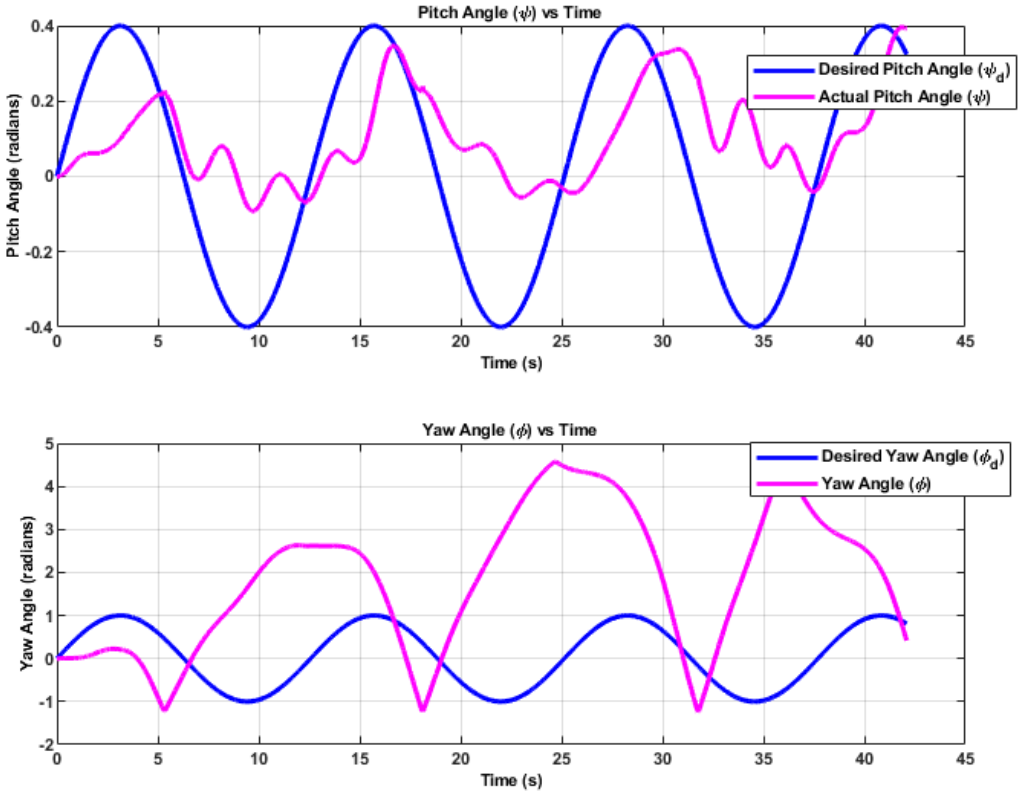

Figure 5.16: Real-time experimental results for the baseline controller in trajectory tracking mode. The system output (magenta) fails to track the reference trajectory (blue).

For the pitch axis, the system initially attempts to follow the reference but quickly deviates into violent, high-frequency oscillations of increasing amplitude, demonstrating a clear and dangerous instability that forced the termination of the experiment. The yaw axis response, while not unstable in the same divergent manner, is entirely incorrect. It fails to capture the frequency or amplitude of the reference signal, exhibiting only a slow, distorted motion that bears no resemblance to the desired trajectory. This plot is the visual embodiment of the problem statement of this thesis: a theoretically sound nonlinear controller that performs perfectly in simulation can be rendered completely ineffective and even dangerous in a practical, real-world application.

5.4.2 Diagnosing the Failure: The Observer Breakdown

The root cause of this catastrophic failure in tracking can be traced directly to the breakdown of the conventional, fixed-gain High-Gain Observer. While the observer was adequate for the quasi-static conditions of regulation, it is wholly unsuited for the dynamic, noisy environment of real-time tracking.

Figure 5.17 shows the state estimates for the pitch and yaw angles generated by the HGO during the failed experiment. This plot reveals the observer's fatal flaw.

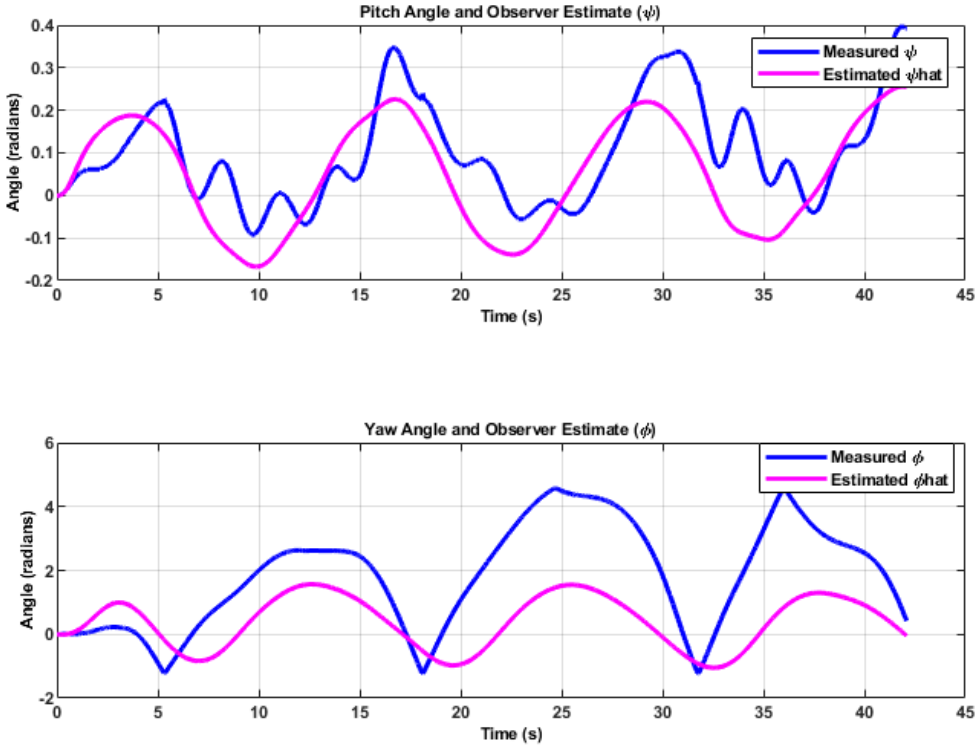

Figure 5.17: State estimates generated by the fixed-gain High-Gain Observer during the failed real-time tracking experiment.

The fixed high gain, which is necessary to provide the fast dynamics required for tracking, aggressively amplifies the measurement noise from the encoders. This interaction results in the estimated states becoming heavily corrupted by high-frequency, large-amplitude oscillations. These extremely noisy and erroneous state estimates are then fed back to the backstepping controller. The controller, having no way to distinguish between true system dynamics and this observer-induced noise, faithfully attempts to counteract these wild fluctuations.

Unstable Control Effort: The Vicious Cycle

The control signals generated during this failure, shown in Figure 5.18, confirm this diagnosis and illustrate the resulting vicious cycle.

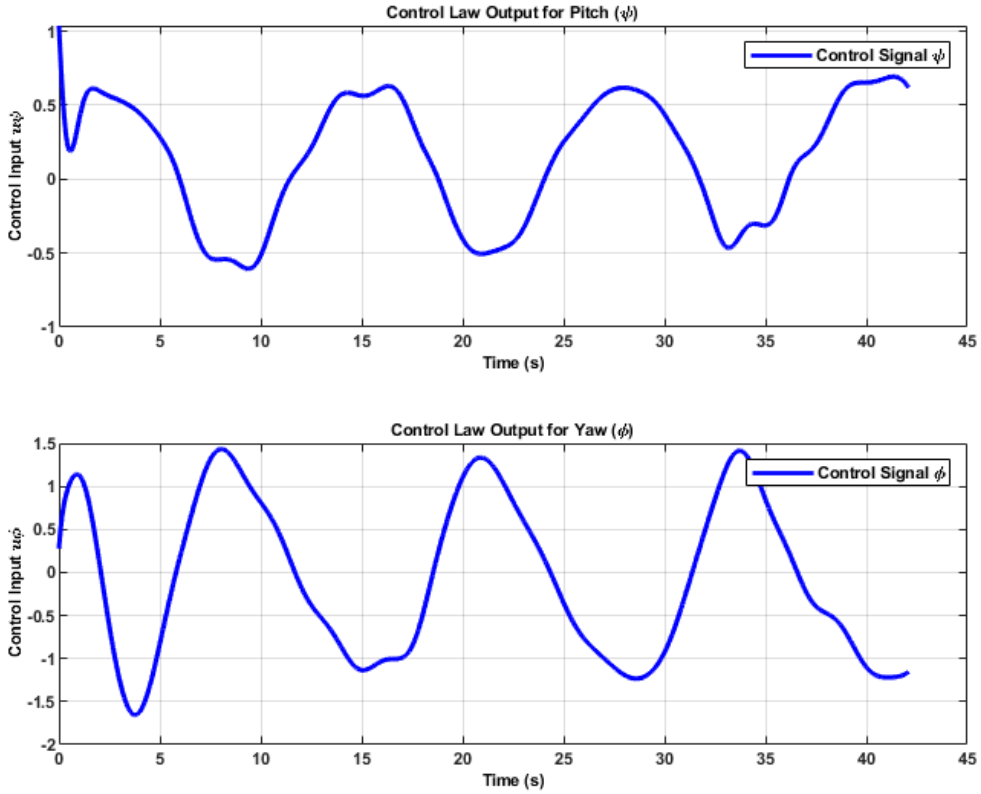

Figure 5.18: Control signals generated by the baseline controller during the failed real-time tracking experiment.

Unlike the smooth, sinusoidal signals from the simulation (Figure 5.8), the real-time control signals are erratic and heavily saturated. The controller is commanding the motors to switch between their maximum and minimum voltage limits a behavior known as ”actuator slamming” in a desperate attempt to follow the wildly oscillating state estimates. This action not only fails to control the system but actively drives it towards instability. This confirms that the fixed-gain HGO is the weak link in the control chain. Its inability to simultaneously provide fast and clean state estimates creates a destructive feedback loop of noisy estimates and aggressive, destabilizing control actions, leading to total system failure.

5.5 Summary of Baseline Controller Limitations

The experimental work presented thus far has systematically investigated the performance of a conventional backstepping-HGO control strategy, highlighting a critical performance gap between theory and practice.

Our findings confirm that in an idealized, model-perfect simulation, the controller performs flawlessly in both regulation and tracking tasks. In a real-world setting, it remains effective for the simpler task of regulation, successfully stabilizing the system and attenuating

its inherent cross-couplings. However, the demanding task of real-time trajectory tracking exposes a fundamental weakness, resulting in complete system failure.

We conclude that this failure stems from the controller's brittle sensitivity to the combination of real-world factors, primarily the measurement noise amplified by the fixed-gain observer. While the controller proved robust to some unmodeled dynamics (like the cross-couplings), it was not robust to the noisy state information that is an unavoidable consequence of using a high-gain observer on a physical system.

This practical failure of an otherwise effective nonlinear control law does not diminish its theoretical value. Instead, it precisely defines the problem that must be solved. It highlights the urgent need for a more intelligent approach to state estimation one that can adapt to the changing conditions of a real-world task. This is the primary motivation for the core contribution of this thesis. Having now empirically demonstrated the limitations of the conventional approach, we will proceed to show how an intelligent, adaptive observer based on neural networks can successfully overcome the very failure modes identified here.

5.6 Performance of the FFNN-Based Adaptive Controller

Having established the definitive failure of the conventional fixed gain controller, we now present the performance of the first proposed solution: the backstepping controller integrated with the Feedforward Neural Network (FFNN) based adaptive High Gain Observer. The first and most crucial step is to verify the stability and nominal performance of this new, more complex algorithm in the same ideal, noise free simulation environment where the baseline controller excelled. This provides a fundamental sanity check before attempting a real world hardware test. The Simulink diagram for this simulation, shown in Figure 5.19, now includes the FFNN algorithm block, which dynamically supplies the gains to the HGO.

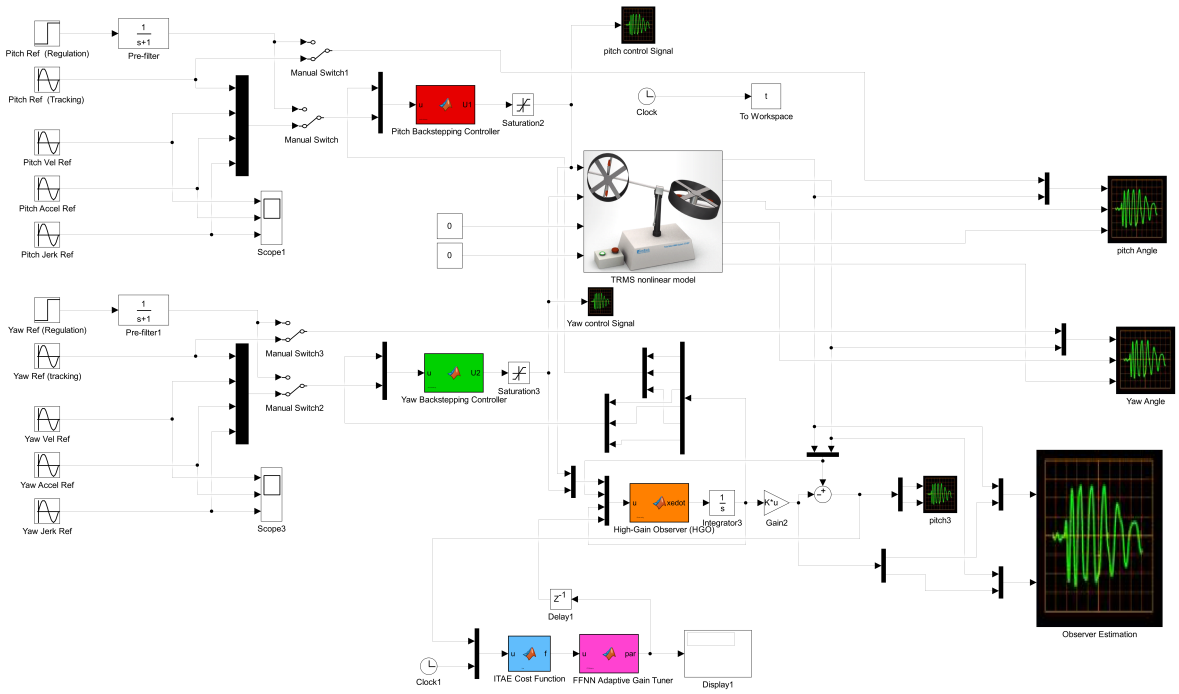

Figure 5.19: Simulink Diagram for the FFNN-Based Adaptive Controller Simulation.

5.6.1 FFNN Controller Performance in Simulation

System Response in Trajectory Tracking Mode

We subjected the adaptive controller to the same demanding sinusoidal trajectory tracking task that caused the baseline controller to fail. The system's tracking performance is shown in Figure 5.20.

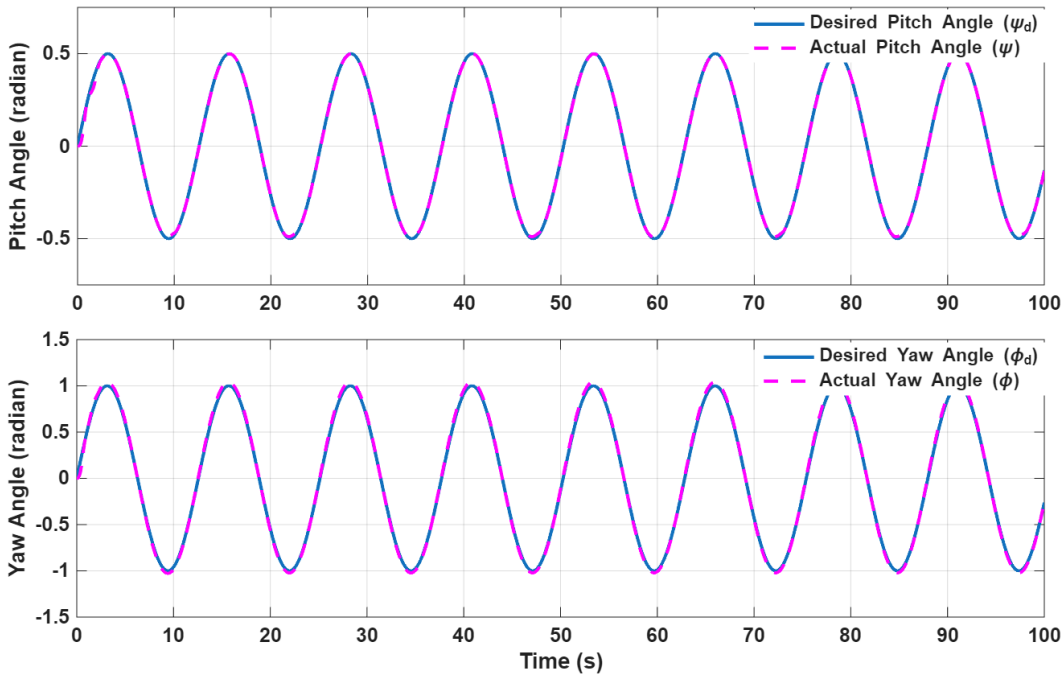

Figure 5.20: Trajectory Tracking Performance of the FFNN Controller in Simulation.

The simulation results are excellent. Following a very brief initial transient lasting approximately 10 seconds, the controller's output for both the pitch and yaw axes almost perfectly matches the reference trajectories. This result is a critical first validation: it confirms that the introduction of the FFNN based adaptive mechanism is not only stable but is capable of achieving the same optimal performance as the baseline controller in an ideal environment.

Analysis of the Adaptive Gain Mechanism

The controller's success is now predicated on the intelligent gain tuner. Figure 5.21 offers the first direct evidence of this adaptive process in action, showing the HGO gain parameters, θ_p and θ_y, as they evolve over time.

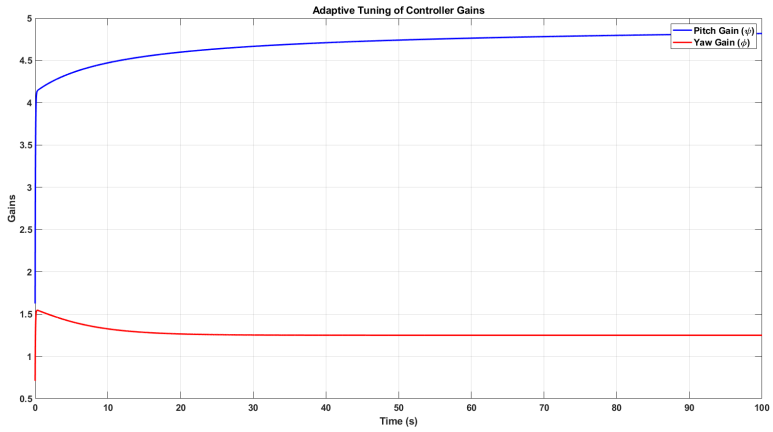

Figure 5.21: Adaptive tuning of the HGO gain parameters by the FFNN in simulation.

This plot clearly illustrates the learning process. The gains begin at their initial values and, driven by the online training algorithm, converge smoothly to new, stable steady state values. This convergence period corresponds exactly with the initial transient seen in the tracking plot. Once the gains have settled, the tracking performance becomes nearly perfect. This demonstrates that the FFNN is successfully performing its intended function: using the error signal to actively search for and find a set of observer gains that optimizes system performance for this specific task.

Observer Performance and Estimation Error

The ultimate goal of the adaptive gain mechanism is to produce fast and accurate state estimates. Figure 5.22 directly assesses the observer's performance by comparing the true system states from the model to the observer's estimates.

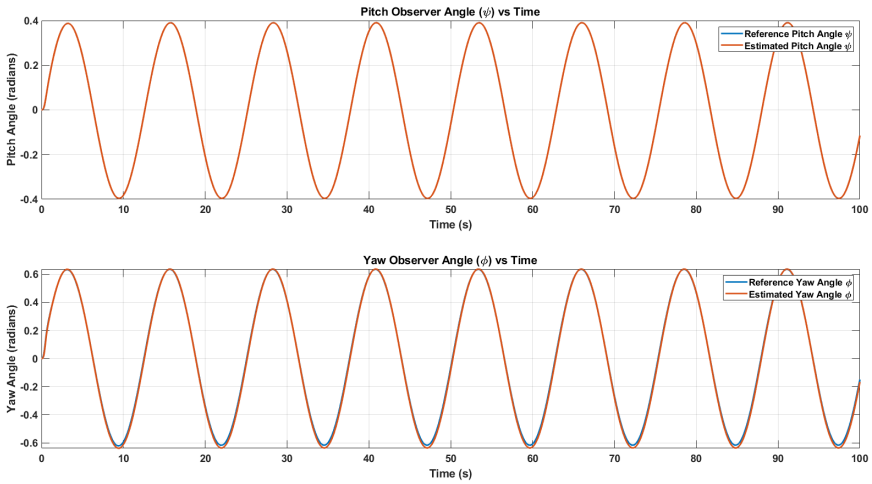

Figure 5.22: High-Gain Observer Performance with FFNN Tuner in Simulation.

The observer's performance is, again, nearly perfect. The estimated states (blue) converge to the true states (red) very quickly. After this brief initial period, the two lines are visually indistinguishable, confirming that the FFNN tuner is providing highly effective gains to the HGO.

To quantify this, Figure 5.23 presents the estimation error, which is the difference between the true state and the estimated state.

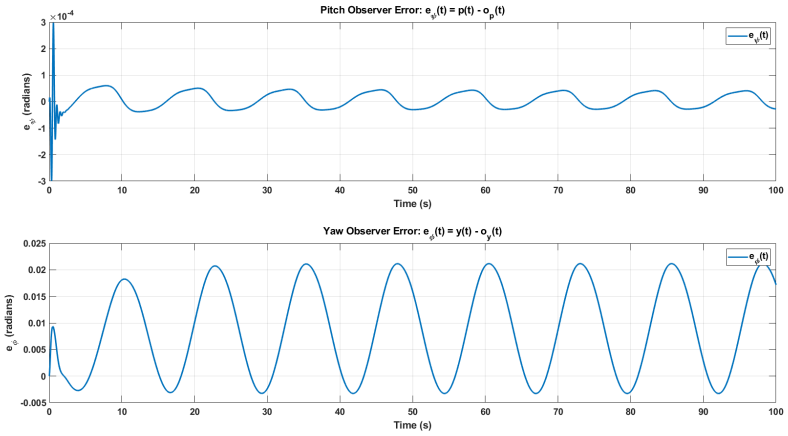

Figure 5.23: Observer estimation error for the FFNN-tuned HGO in simulation.

This plot provides quantitative proof of the adaptive observer's effectiveness. The initial estimation error is rapidly suppressed and settles into a very small, bounded oscillation. The minimal magnitude of this steady state error confirms that the observer is providing clean and accurate state estimates to the backstepping controller, which is the necessary condition for high performance tracking.

Control Effort in Simulation

Finally, we analyzed the control signals to ensure the adaptive controller remained efficient. The control effort is shown in Figure 5.24.

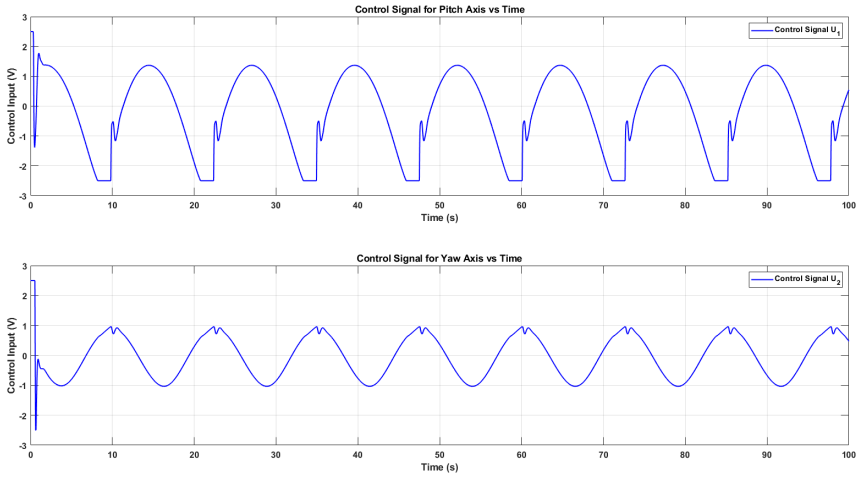

Figure 5.24: Control Effort of the FFNN-Based Controller in Simulation.

The resulting control signals are smooth, continuous, and remain well within typical actuator limits. This confirms that the added complexity of the FFNN tuner has not introduced any undesirable chattering or aggressiveness into the control law in the ideal case. Having

now successfully validated the FFNN based adaptive controller in a simulated environment, we have the necessary confidence to proceed with the far more telling real time hardware experiments.

5.6.2 FFNN Controller Performance in Real-Time Experiments

We then deployed the FFNN enhanced controller on the physical TRMS hardware for the final and most important validation of our proposed solution. This test was conducted under the exact same sinusoidal tracking scenario that induced catastrophic failure in the baseline controller. This direct comparison is designed to isolate the impact of the adaptive observer and demonstrate its ability to solve the core problem. The Simulink model for this hardware-in-the-loop experiment is shown in Figure 5.25.

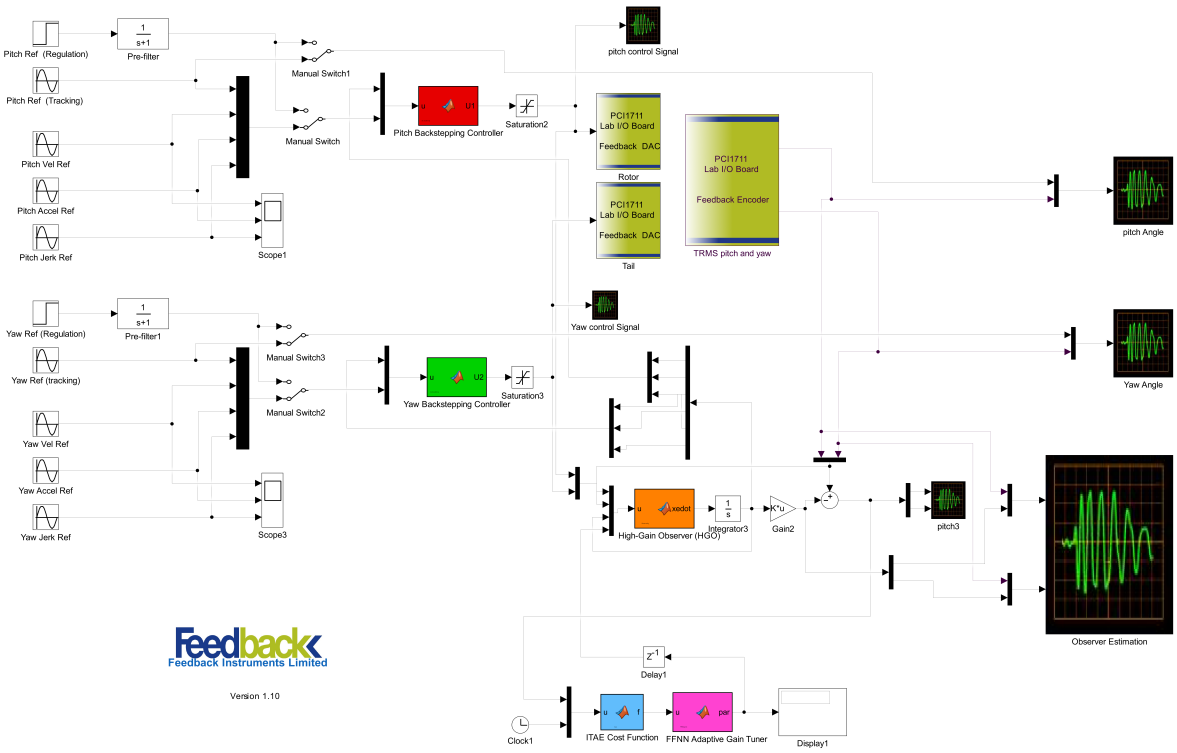

Figure 5.25: Simulink Diagram for the FFNN-based Real-Time (HIL) Experiment.

System Response: A Successful Recovery from Failure

The primary result is the real time tracking performance of the physical system, shown in Figure 5.26. The outcome is a dramatic and unambiguous success.

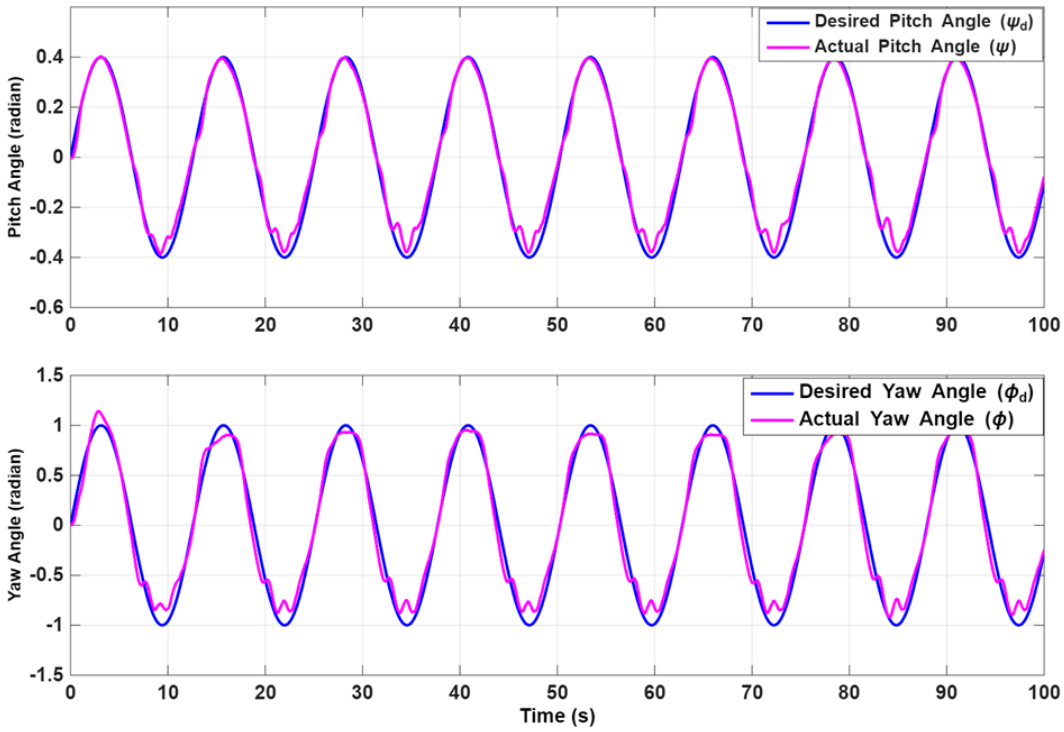

Figure 5.26: Real-time tracking performance of the FFNN-enhanced controller.

Where the baseline controller became unstable and failed completely, the FFNN enhanced system now successfully tracks the reference trajectories for both pitch and yaw. The violent oscillations and divergent behavior are entirely eliminated. The system's output follows the desired path with good fidelity, demonstrating stable and effective control. This result is the first definitive proof that the adaptive gain mechanism of the FFNN tuner directly remedies the failure mode of the conventional, fixed gain approach.

Analysis of the Real-Time Adaptive Gain Mechanism

Figure 5.27 provides direct evidence of the intelligent adaptation process at the heart of this success. It plots the HGO gain parameters, θ_p and θ_y, as they are adjusted by the FFNN during the real time experiment.

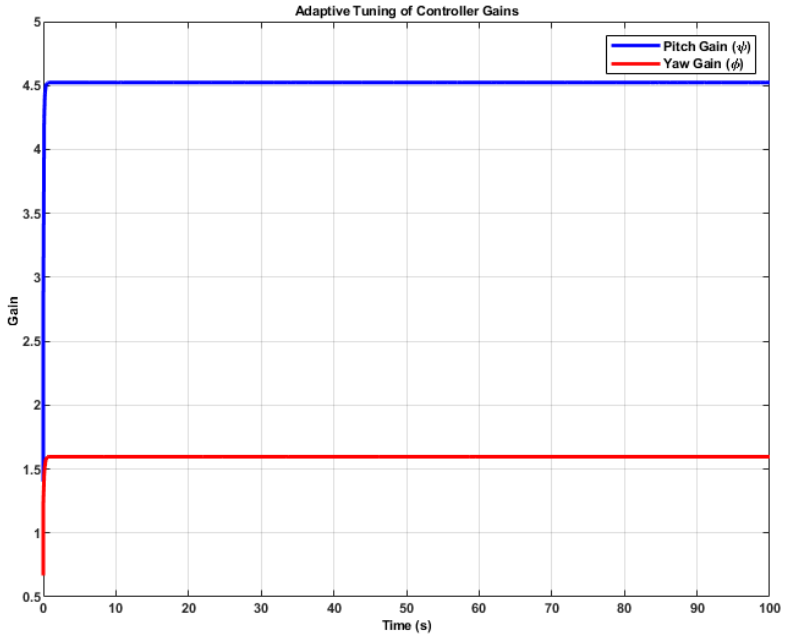

Figure 5.27: Adaptive tuning of the HGO gain parameters by the FFNN in the real-time experiment.

This plot is a powerful illustration of the system's intelligence. In stark contrast to the static gains of the baseline controller, these gains are continuously and smoothly adjusted by the FFNN in response to the live observation error. This dynamic behavior is the key to the controller's success. It allows the observer to intelligently balance the trade-off between estimation speed and noise rejection on the fly, a feat impossible for any fixed gain system.

Observer Performance: The Key to Robustness

The purpose of this adaptive mechanism is to generate clean and accurate state estimates even from noisy sensor data. Figure 5.28 shows how well the FFNN tuned observer's estimates track the measured system states.

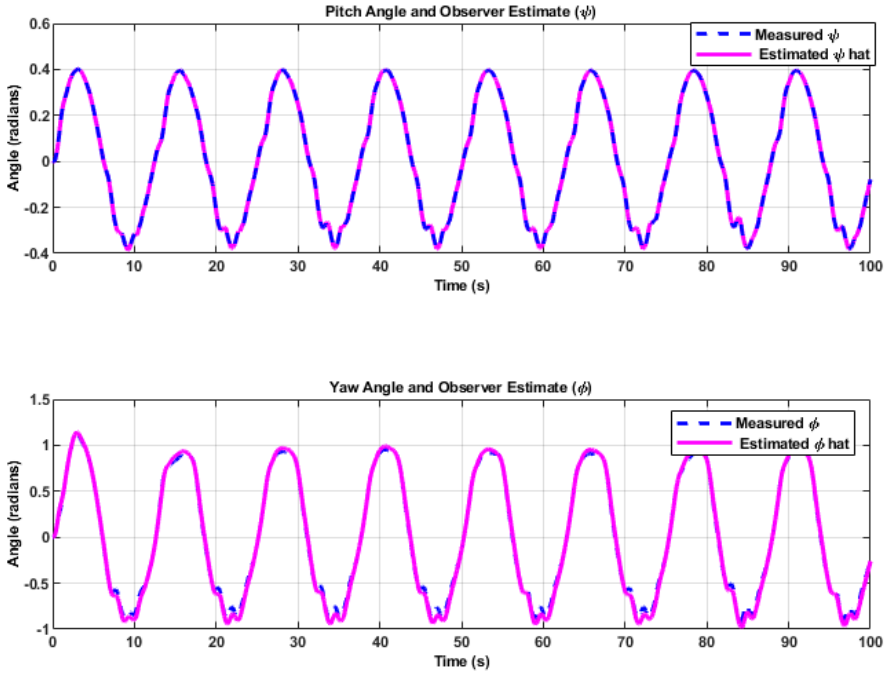

Figure 5.28: Real-time observer performance with the FFNN tuner.

The observer's performance is excellent and stands in sharp contrast to the baseline observer's failure (Figure 5.17). The estimated states now accurately follow the true system states with very little visible error, even in the presence of real world noise and system motion. The wild, divergent oscillations are gone. This proves that the FFNN tuner is successfully managing the HGO gains to reject noise while maintaining tracking accuracy.

To quantify this, Figure 5.29 shows the real time estimation error.

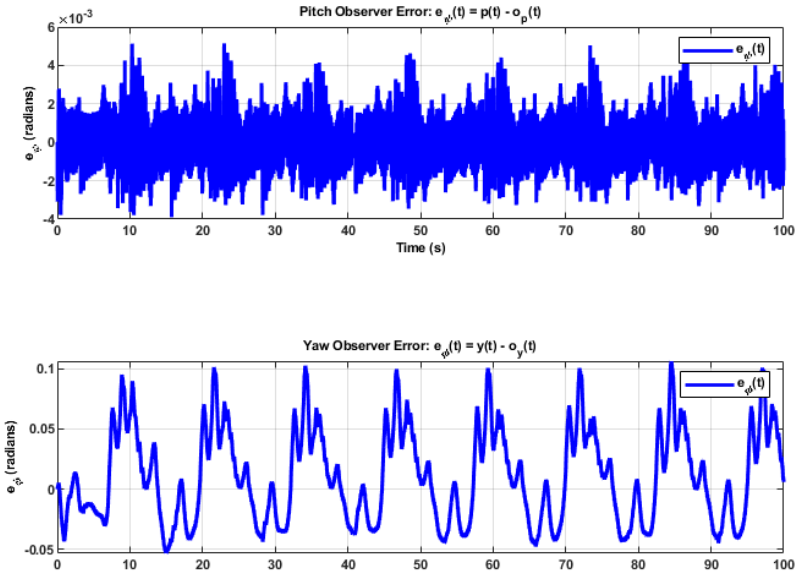

Figure 5.29: Observer estimation error for the FFNN-tuned HGO in the real-time experiment.

This plot confirms the success of the adaptive observer. The estimation error is kept small and contained within a tight, bounded range throughout the entire experiment. By preventing the error from growing and corrupting the estimates, the FFNN ensures that the backstepping controller receives clean and reliable state information.

Control Effort: The Return to Stability

Finally, we analyze the control signals from this successful experiment, shown in Figure 5.30.

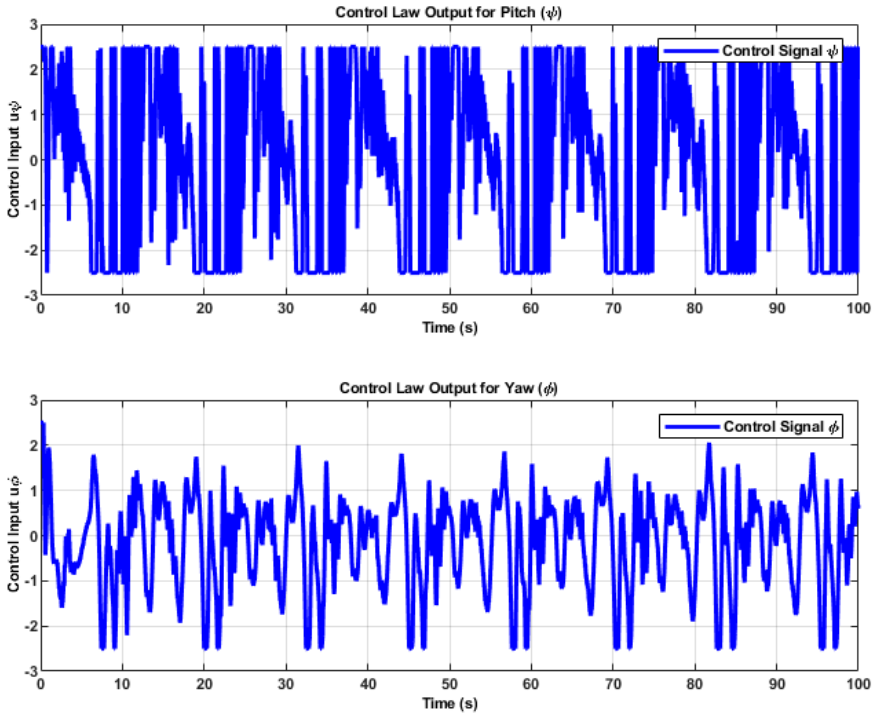

Figure 5.30: Control effort of the FFNN-based controller during real-time tracking.

The control signals tell the final part of the story. The erratic, saturated ”actuator slamming” of the failed baseline controller is gone. In its place are smooth, well behaved control signals that operate comfortably within the motor voltage limits. This plot is the ultimate proof of the system’s success: the clean state estimates provided by the FFNN tuned observer allow the backstepping controller to function exactly as it was designed to, producing efficient, stable, and effective control actions.

5.7 Performance of the RBFNN-Based Adaptive Controller

To provide a comprehensive comparative analysis, we developed a second intelligent controller using the alternative Radial Basis Function Neural Network (RBFNN) architecture. The first step, as with the FFNN, was to verify its stability and baseline performance in the ideal, noise-free simulation environment. This is a crucial validation step to ensure the RBFNN-based adaptation algorithm is theoretically sound before proceeding to hardware tests. The Simulink diagram, shown in Figure 5.31, is structurally identical to the FFNN setup, with only the neural network block itself being replaced.

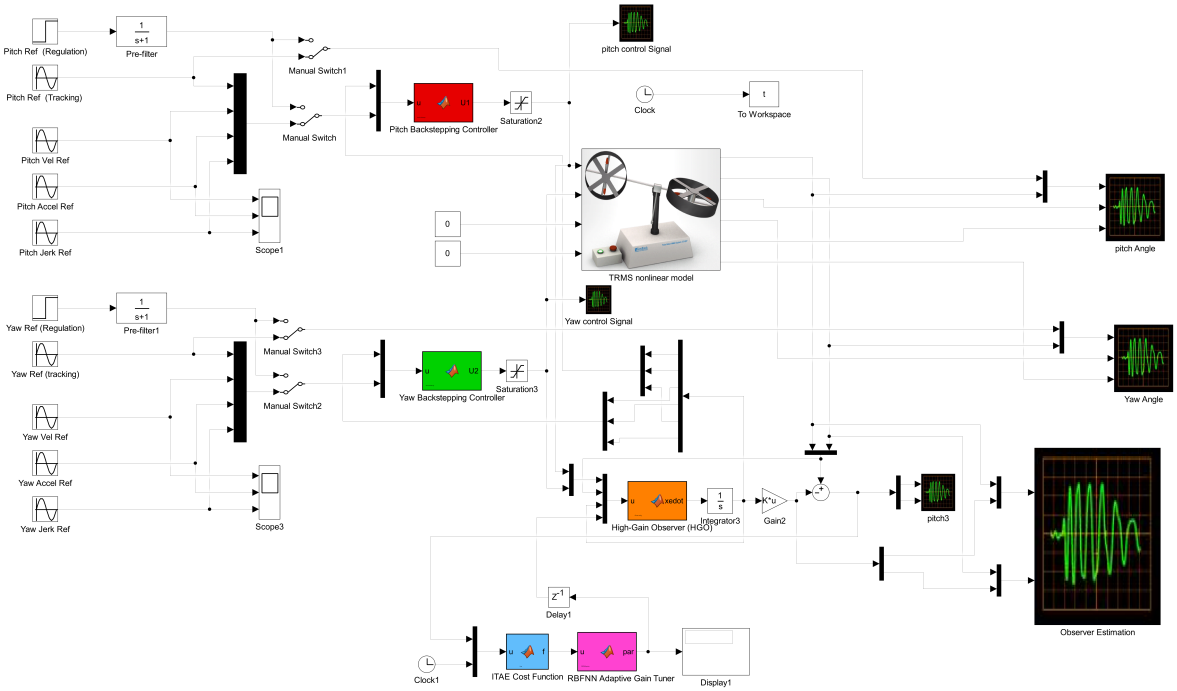

Figure 5.31: Simulink Diagram for the RBFNN-Based Adaptive Controller Simulation.

5.7.1 RBFNN Controller Performance in Simulation

System Response in Tracking Mode

The RBFNN-based controller was subjected to the same sinusoidal trajectory tracking task. The resulting system performance is shown in Figure 5.32.

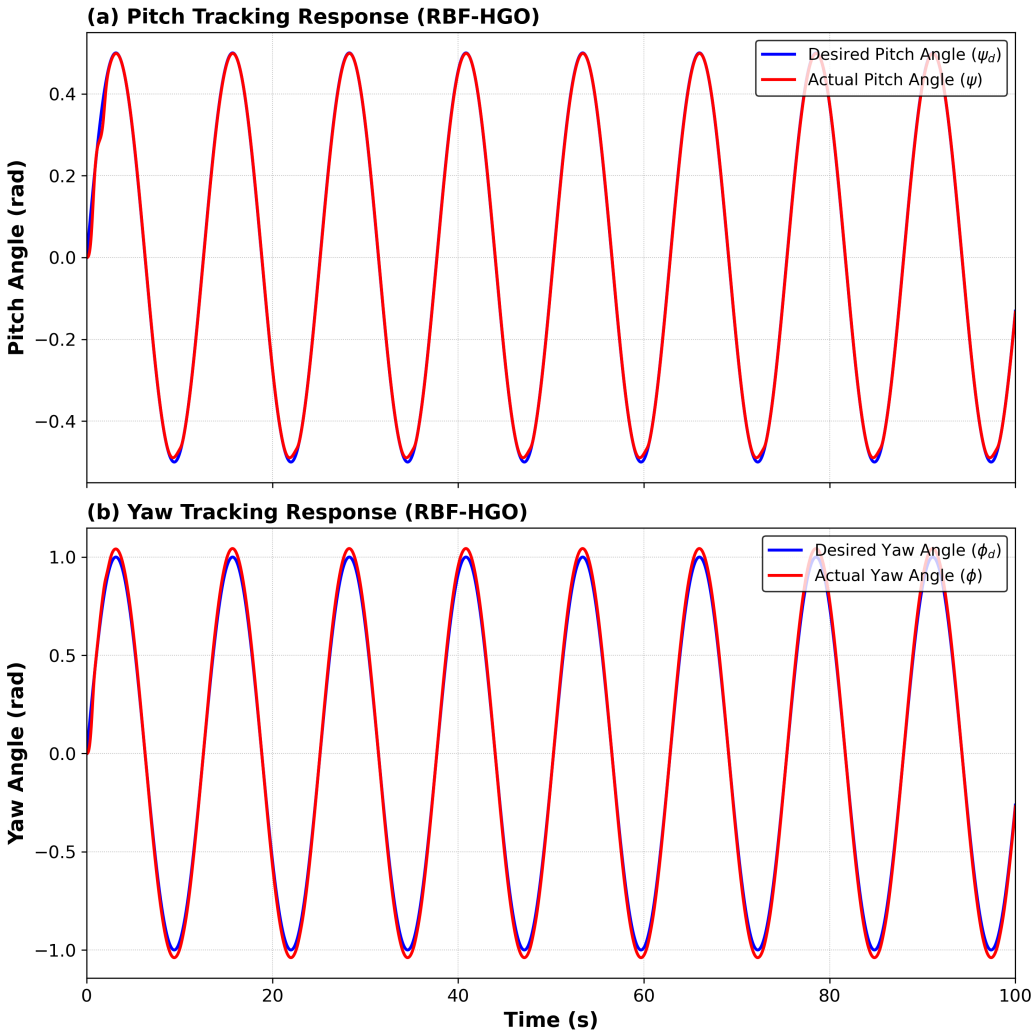

Figure 5.32: Trajectory Tracking Performance of the RBFNN Controller in Simulation.

The simulation results are again nearly perfect. Following a brief initial transient, the controller's output for both the pitch and yaw axes almost perfectly matches the reference trajectories. The tracking error becomes negligible very quickly. This result confirms that the RBFNN-based adaptive controller is also fundamentally stable and capable of achieving optimal performance in an ideal environment, providing a solid point of comparison against the FFNN.

Analysis of the RBFNN Adaptive Gain Mechanism

The controller's performance is driven by the RBFNN-based gain tuner. Figure 5.33 shows the behavior of this tuner by plotting the evolution of the HGO gain parameters, θ_p and θ_y, over time.

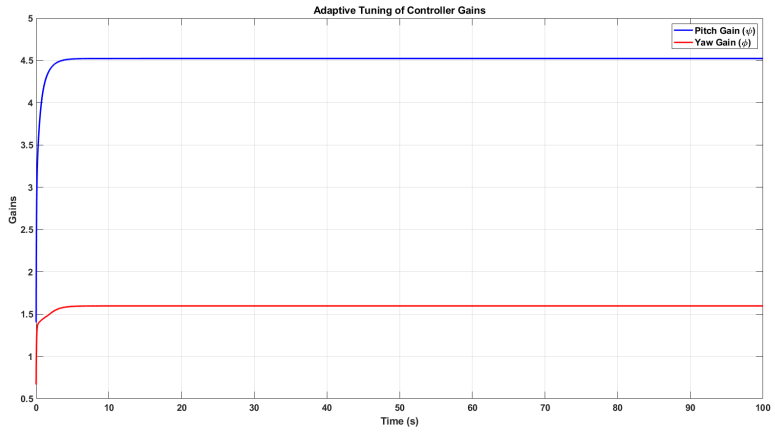

Figure 5.33: Adaptive tuning of the HGO gain parameters by the RBFNN in simulation.

The plot provides direct evidence of the RBFNN's adaptation process. The gains start at their initial values and converge rapidly to new, stable steady-state values. The convergence period mirrors the initial transient seen in the tracking plot. This demonstrates that the RBFNN, despite its different internal architecture, is also successfully optimizing the observer gains for the given task.

Observer Performance and Estimation Error

The objective of the RBFNN tuner is to produce accurate state estimates. Figure 5.34 shows the observer's performance, comparing the true system states to the observer's estimates.

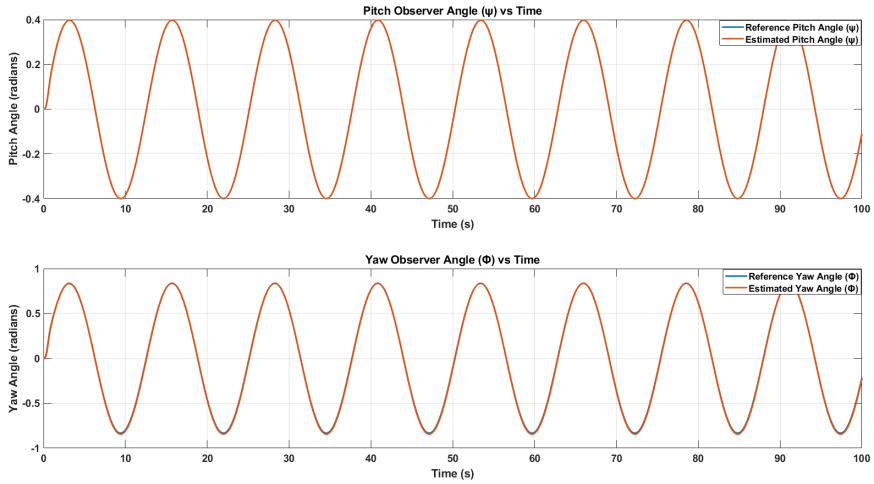

Figure 5.34: High-Gain Observer Performance with RBFNN Tuner in Simulation.

The observer's performance with the RBFNN tuner is excellent. The estimated states converge to the true states very quickly, and after this convergence, the two are visually indistinguishable. This confirms that the RBFNN tuner is also providing highly effective gains to the HGO in the simulation environment.

To quantify this, Figure 5.35 shows the resulting estimation error.

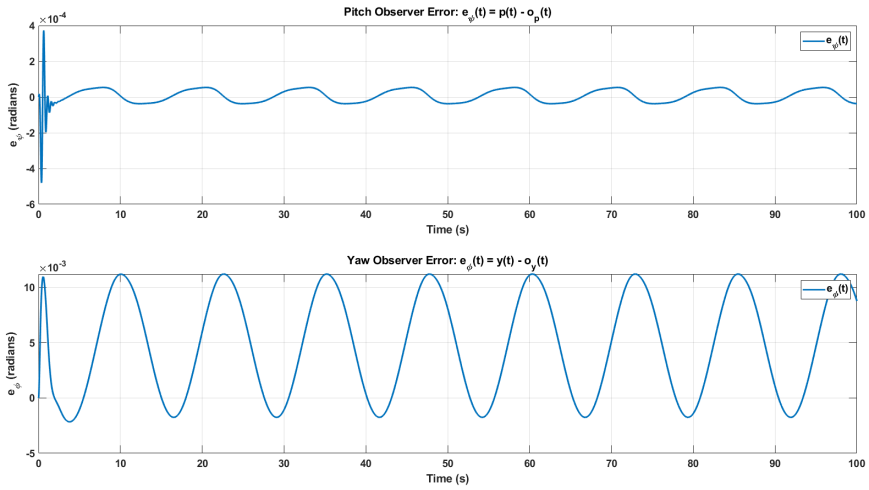

Figure 5.35: Observer estimation error for the RBFNN-tuned HGO in simulation.

This plot confirms the tuner is highly effective. The initial estimation error is suppressed almost immediately and settles into a very small, bounded oscillation. The minimal steady-state error proves that the observer is providing clean and accurate state estimates to the controller.

Control Effort in Simulation

Finally, we analyzed the control signals to ensure the RBFNN-based controller also operates efficiently. The control effort is shown in Figure 5.36.

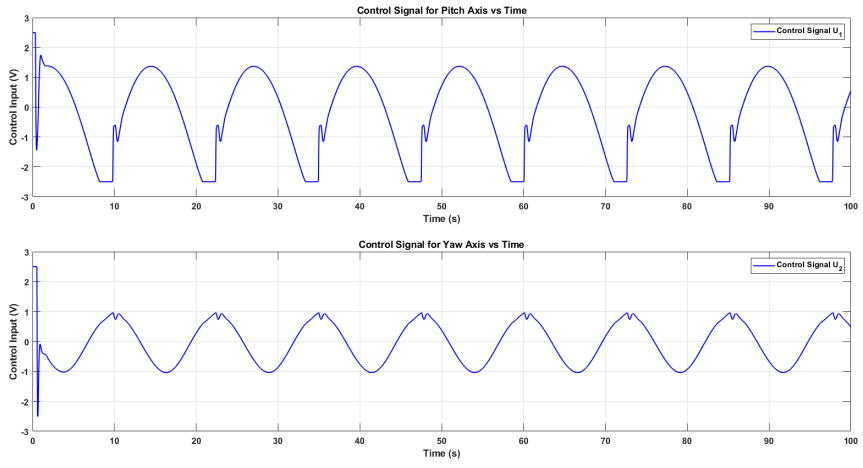

Figure 5.36: Control Effort of the RBFNN-Based Controller in Simulation.

The control signals are smooth, continuous, and remain well within the typical actuator limits, indicating that the RBFNN-based controller is both effective and efficient in simula-

tion. These successful results fully validate the RBFNN's design and provide the necessary confidence to proceed with its real-time hardware testing, where its performance can be directly compared to that of the FFNN.

5.7.2 RBFNN Controller Performance in Real-Time Experiments

Finally, we deployed the RBFNN-enhanced controller on the physical TRMS hardware to complete the validation of our proposed solutions. This test serves as the final piece of experimental evidence, conducted under the same challenging tracking scenario where the baseline controller failed. This allows for a direct, one-to-one comparison not only against the failed baseline but also against the successful FFNN controller. The Simulink model for this hardware-in-the-loop experiment is shown in Figure 5.37.

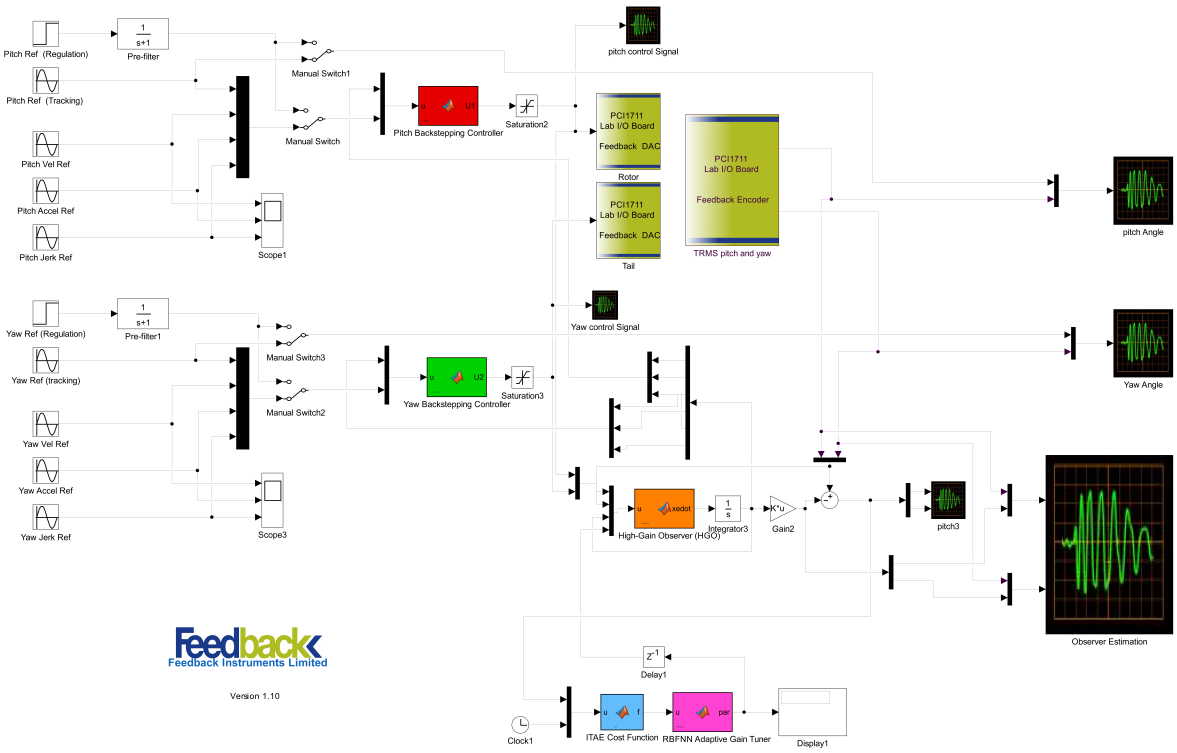

Figure 5.37: Simulink Diagram for the RBFNN-based Real-Time (HIL) Experiment.

System Response in Real-Time Tracking

The primary result is the tracking performance of the physical system, presented in Figure 5.38.

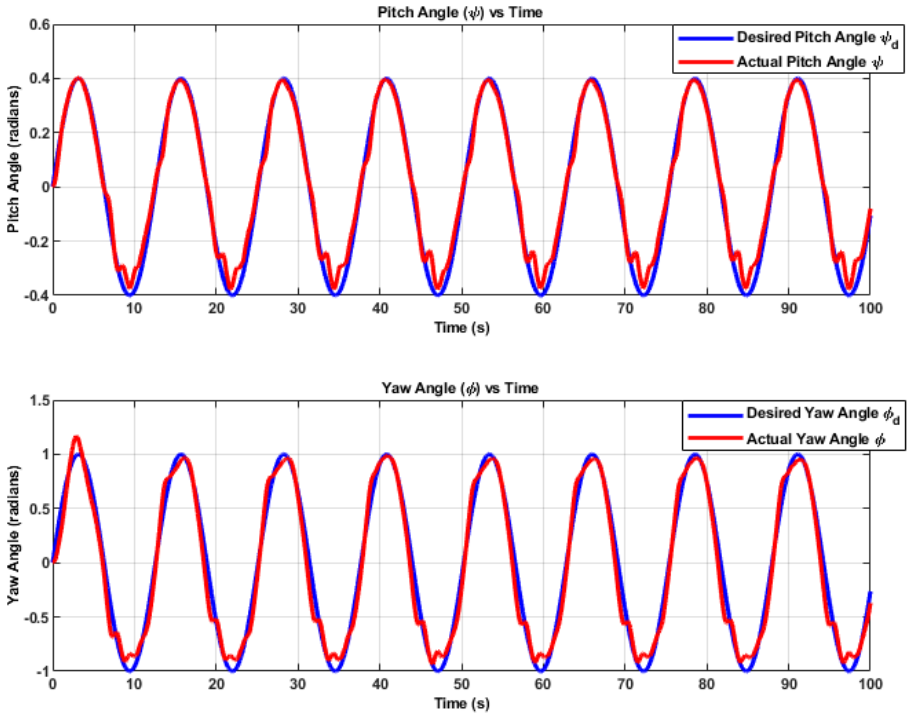

Figure 5.38: Real-time tracking performance of the RBFNN-enhanced controller.

The plot clearly shows that the RBFNN-based controller also achieves successful and stable tracking. The system's output now correctly follows the reference trajectories for both pitch and yaw, completely eliminating the instability of the baseline system. This result provides a second, independent validation of our core thesis: that an intelligent, adaptive observer is the key to robust real-world performance. It confirms that the concept is not limited to a single neural network architecture.

Analysis of the RBFNN Real-Time Adaptive Gain Mechanism

Figure 5.39 shows the behavior of the RBFNN tuner during the experiment, plotting the HGO gain parameters, θ_p and θ_y, over time.

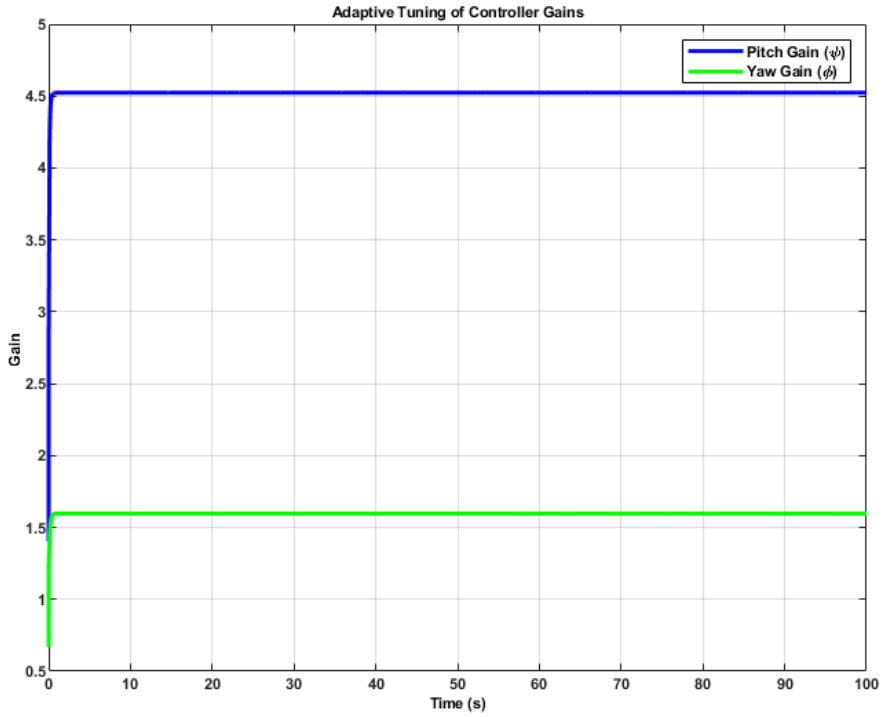

Figure 5.39: Adaptive tuning of the HGO gain parameters by the RBFNN in the real-time experiment.

This plot gives us direct insight into the RBFNN's real-time adaptation process. The gains are not static but are continuously adjusted by the RBFNN in response to the live observation error. This dynamic behavior, similar to that of the FFNN, is the essential mechanism that allows the observer to balance the conflicting demands of estimation speed and noise rejection in a challenging real-world environment.

Observer Performance and Estimation Error in Real-Time

The adaptive mechanism's success hinges on its ability to produce clean and accurate state estimates from noisy sensor data. Figure 5.40 shows how well the RBFNN-tuned observer's estimates track the measured system states.

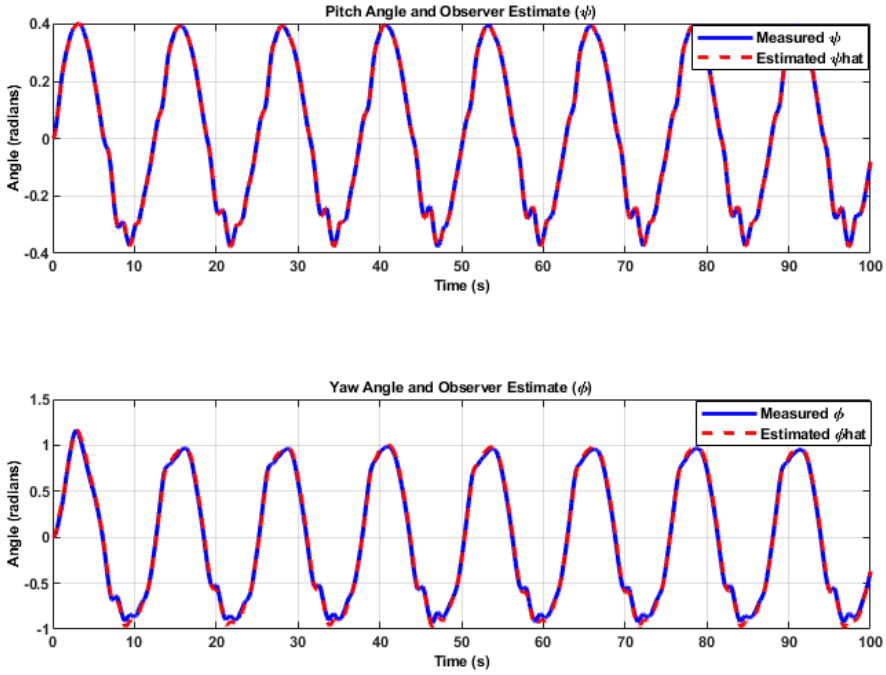

Figure 5.40: Real-time observer performance with the RBFNN tuner.

The observer's performance with the RBFNN tuner is excellent. The estimated states accurately follow the true states with very little visible error, once again demonstrating a vast improvement over the failed baseline observer. This proves that the RBFNN tuner is also highly effective at managing the HGO gains in a noisy, dynamic environment.

To quantify this, Figure 5.41 shows the real-time estimation error.

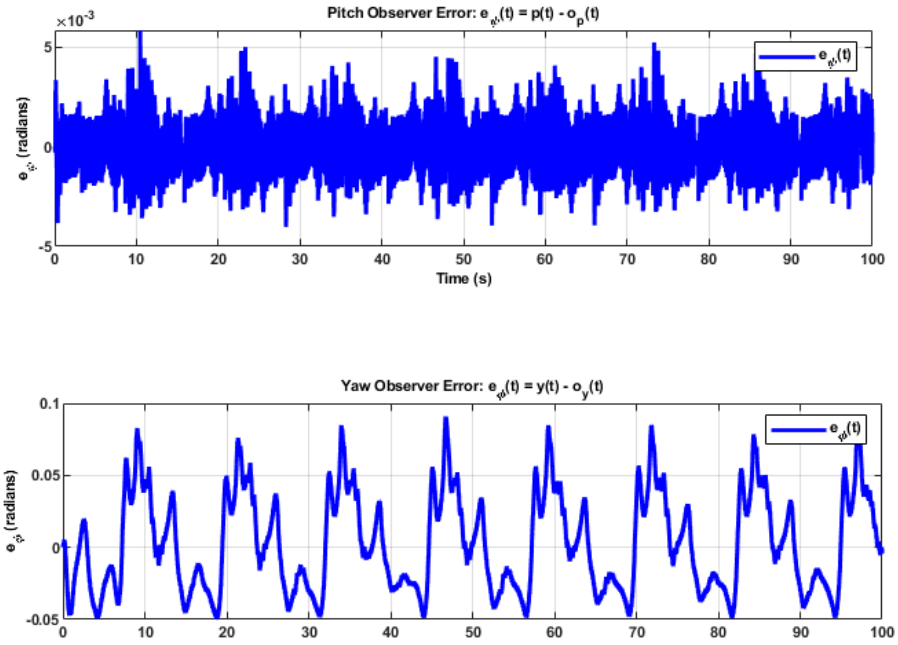

Figure 5.41: Observer estimation error for the RBFNN-tuned HGO in the real-time experiment.

This plot confirms the success of the RBFNN-based adaptive observer. The estimation error is kept small and well-contained within a tight, bounded range. By preventing the error from growing, the RBFNN, like the FFNN, ensures that the backstepping controller receives the clean and reliable state information it needs to function properly.

Control Effort in Real-Time

Finally, we analyze the control signals from this successful experiment. Figure 5.42 shows the voltage sent to the motors.

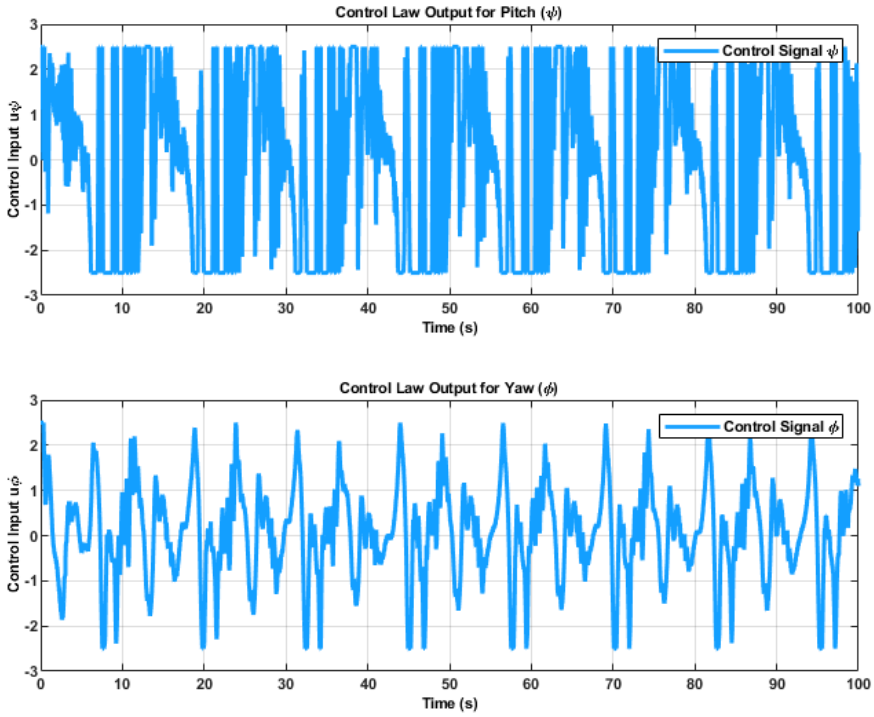

Figure 5.42: Control effort of the RBFNN-based controller during real-time tracking.

The control signals are smooth and well-behaved, operating well within the actuator limits and showing no signs of the chattering or saturation that plagued the baseline controller. This plot provides the final proof that the RBFNN-tuned observer, by providing clean state estimates, enables the backstepping controller to work as designed, producing efficient and stable control actions.

With both the FFNN and RBFNN controllers now proven to be successful in the real-time tracking task, the stage is set for the final, quantitative comparison of their performance.

5.8 Comparative Analysis and Discussion

The preceding sections have documented the performance of each controller individually. We established the theoretical success of the baseline controller in simulation and its definitive failure in real-time experiments. We then demonstrated that both the FFNN-based and RBFNN-based adaptive controllers successfully solved this real-world tracking problem.

This final section brings all of these results together for a direct, quantitative and qualitative comparison. The analysis is focused on the challenging real-time trajectory tracking experiment, as it represents the core problem this thesis set out to solve.

5.8.1 Visual Comparison of Tracking Performance

The most immediate and powerful comparison comes from a direct visual inspection of the tracking results. Figure 5.43 presents an overlay of the real-time tracking performance for the baseline controller, the FFNN-based controller, and the RBFNN-based controller.

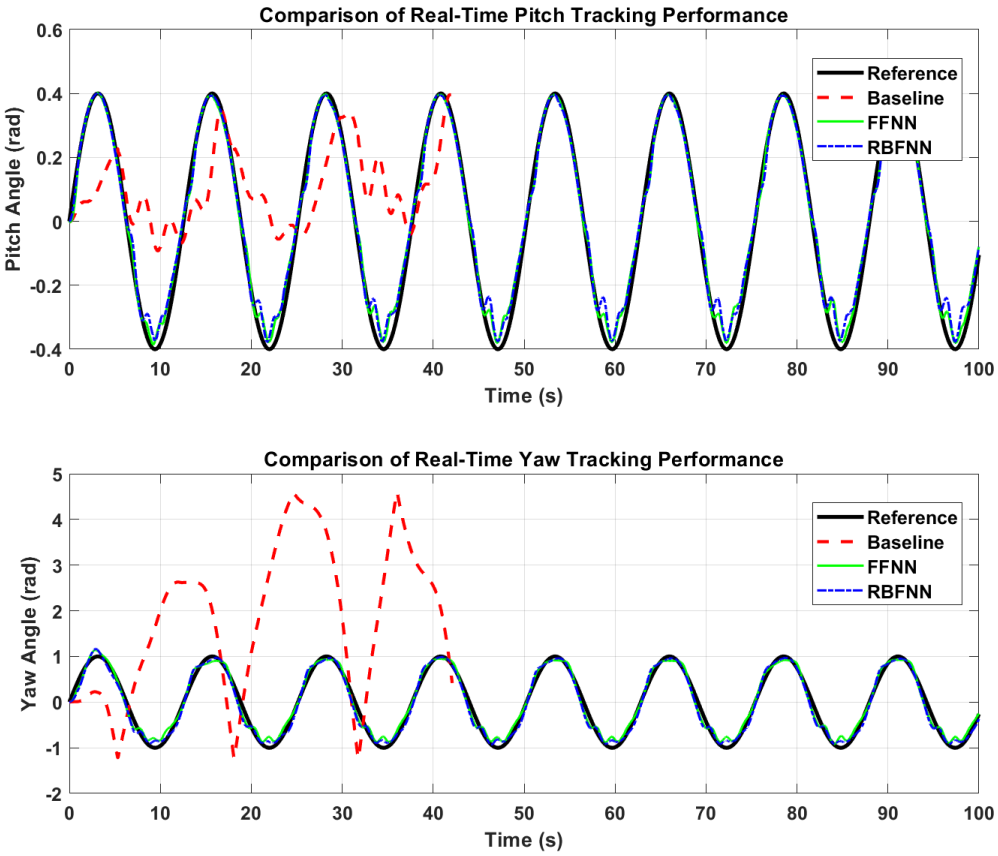

Figure 5.43: Visual comparison of real-time tracking performance for the Baseline, FFNN, and RBFNN controllers.

This single figure encapsulates the central finding of this thesis. The failure of the baseline controller (red dashed line) is stark and unambiguous. It is unable to follow the reference trajectory (black line) and is driven to instability. In stark contrast, both the FFNN (green line) and RBFNN (blue dashed line) controllers demonstrate excellent tracking performance. Their outputs closely follow the reference trajectory for both the pitch and yaw axes, confirming their success.

5.8.2 Quantitative Performance Comparison

To move beyond visual inspection to a rigorous, data-driven conclusion, we turn to the performance metrics defined in Chapter 4. Table 5.1 summarizes the calculated metrics for each of the three controllers from the real-time tracking experiments.

Table 5.1: Quantitative comparison of controller performance in real-time tracking.

Performance Metric	Baseline HGO	FFNN-HGO	RBFNN-HGO
RMSE (Pitch) [rad]	0.2571	**0.0327**	0.0385
RMSE (Yaw) [rad]	2.5624	0.1111	**0.1080**
ITAE (Pitch) [rad·s]	1990.24	**1271.62**	1486.68
ITAE (Yaw) [rad·s]	21281.99	4633.87	**4077.79**
CE (Pitch) [V²·s]	82.83	3670.23	**3636.53**
CE (Yaw) [V²·s]	352.25	**1112.00**	1119.65

The numerical data provides quantitative validation for the visual results. The RMSE and ITAE values for both AI-enhanced controllers are an order of magnitude smaller than for the baseline, quantifying their superior accuracy and their ability to rapidly eliminate persistent errors.

The table also reveals interesting trade-offs between the two successful AI methods. The FFNN controller achieved a slightly lower RMSE and a notably better ITAE on the pitch axis, suggesting its global approximation strategy may have yielded a slightly more precise and faster-settling response for this axis. Conversely, the RBFNN controller showed a slightly better RMSE and a significantly better ITAE in the more challenging yaw axis. This suggests that the RBFNN's localized response may have been more effective at modeling and compensating for the complex, coupled dynamics of the yaw channel.

The Control Effort (CE) metric is also highly informative. The deceptively low control effort of the baseline controller is not a sign of efficiency, but rather a symptom of its failure; the controller simply gave up performing useful work as it lost tracking. In contrast, the higher control effort of the AI-based controllers represents the energy required for continuous, precise adjustments. This is the necessary cost of achieving high-performance control, and both AI controllers manage this cost effectively, with very similar energy consumption profiles.

5.8.3 Analysis of the Adaptive Gain Mechanism

Figure 5.44 provides the final piece of the puzzle, explaining *why* the AI controllers succeeded. It shows the real-time behavior of the observer gain parameters as determined by the FFNN and RBFNN tuners.

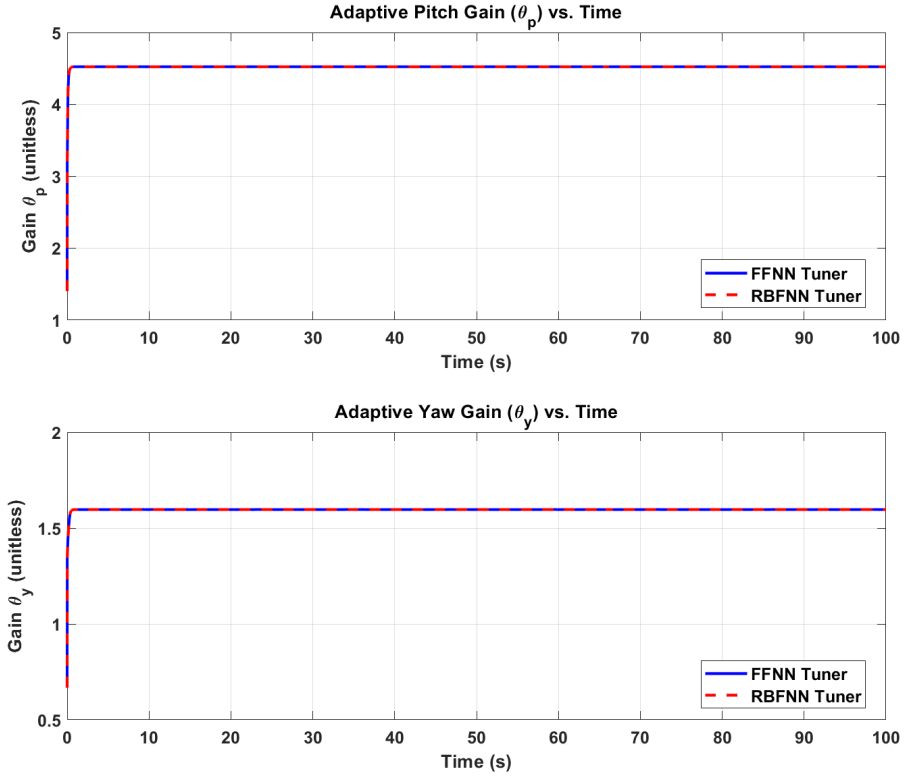

Figure 5.44: Observer gain parameters (θ_p, θ_y) during real-time tracking for the FFNN and RBFNN controllers.

Figure 5.44 confirms that neither tuner produces a static gain. Both the FFNN and RBFNN begin from their initialized values and converge rapidly to stable operating points, with the pitch gain θ_p settling near the upper boundary of its operational range and the yaw gain θ_y settling at a moderate value within its range. The two axes converge to distinct values, reflecting the different dynamic characteristics and noise levels of the pitch and yaw channels. Notably, the FFNN and RBFNN converge to the same steady-state gain for each axis, a result that is expected and discussed in detail below. This real-time adaptation is the mechanism through which both observers acquired the state estimation quality necessary for stable trajectory tracking, which the fixed-gain architecture was unable to achieve.

5.8.4 Discussion

The results presented in this chapter provide a clear and definitive validation of our thesis statement. We have shown that a conventional, fixed-gain Backstepping-HGO controller, while effective in simulation, is not robust enough for practical, real-time trajectory tracking of the TRMS. We have then proven, through direct experimental evidence, that this failure can be completely solved by augmenting the observer with an intelligent, AI-based gain tuner.

Both the FFNN and RBFNN architectures proved to be highly effective solutions. They successfully stabilized the system, achieved excellent tracking performance, and did so with

smooth and practical control signals. The quantitative data reveals subtle but important performance differences between them. The FFNN, as a global approximator, demonstrated a slight advantage on the pitch axis, while the RBFNN, as a local approximator, performed marginally better on the more complex yaw axis. This suggests that the choice between the two architectures is not arbitrary and may depend on the specific dynamics of the system being controlled a valuable insight for future work.

In conclusion, this chapter has successfully demonstrated that the gap between simulation and reality in this challenging control problem can be definitively closed. The solution was not to create an infinitely more complex plant model, but rather to embed the existing control structure with a targeted, lightweight intelligence. By giving the observer the ability to learn from its own performance and adapt to the real world in real time, we transformed a failed controller into a high-performance, practical, and robust system. This serves as a powerful case study for the value of neuro-adaptive control in bridging the gap from theory to practice.

General Conclusion

This thesis was motivated by a concrete and reproducible engineering failure. A theoretically sound nonlinear controller, combining backstepping control with a fixed-gain High-Gain Observer and validated through extensive simulation, collapsed entirely when deployed on the physical Twin Rotor MIMO System for real-time trajectory tracking. The root cause was not a flaw in the control law itself but a fundamental limitation of the fixed-gain observer architecture, which cannot simultaneously satisfy the conflicting demands of fast state estimation and effective sensor noise rejection under dynamic operating conditions. The work presented in this thesis was devoted to diagnosing this failure rigorously and resolving it through an intelligent, adaptive observation strategy grounded in online-learning neural networks.

Synthesis of Findings

The research began with a comprehensive review of the state of the art, which revealed that while adaptive High-Gain Observers and neural network-based observers had been studied independently, no prior work had combined modern neural network architectures with a conventional HGO to solve the specific real-time tracking failure on a complex MIMO plant. A complete mathematical model of the TRMS was then derived from first principles and cast in a state-space form suitable for observer and controller design. On this foundation, a decentralized backstepping controller was synthesized for both regulation and trajectory tracking, with formal Lyapunov-based stability guarantees at each design stage. The baseline controller was paired with a fixed-gain HGO and tested systematically: it performed flawlessly in simulation and adequately in real-time regulation, but failed catastrophically during real-time trajectory tracking due to the noise amplification inherent in the high-gain architecture. Two AI-enhanced adaptive observers were then designed, one based on an FFNN and one based on an RBFNN, both operating entirely online. Experimental validation on the physical TRMS confirmed that both adaptive schemes successfully restored robust trajectory tracking under the exact conditions that caused the baseline controller to fail.

Contributions

First contribution: experimental characterization of the simulation-to-reality gap. The performance failure of the conventional backstepping controller paired with a fixed-gain HGO was documented on the physical TRMS in a controlled and reproducible manner. By

establishing a clear and honest evidence base for this failure, this work makes a direct contribution to the literature: a well-characterized benchmark case that rigorously demonstrates the practical limits of fixed-gain observer architectures under real-world sensor noise and dynamic tracking demands.

Second contribution: design and experimental validation of two AI-enhanced adaptive observers. Two distinct intelligent gain tuners were designed and implemented, one based on a Feedforward Neural Network (FFNN) and one based on a Radial Basis Function Neural Network (RBFNN). Both architectures operate entirely online, requiring no offline pre-training, and both successfully restored stable and accurate trajectory tracking on the physical TRMS testbed under the exact conditions that caused the conventional controller to fail. The experimental results constitute definitive proof of the validity of the proposed adaptive observation strategy.

Third contribution: systematic comparative analysis of the two tuning schemes. The FFNN-based and RBFNN-based controllers were evaluated under identical experimental conditions and against the same set of quantitative performance metrics. This analysis characterized their respective adaptation dynamics, noise rejection capabilities, and practical trade-offs, providing actionable guidance for practitioners who must select between competing neural network topologies for similar observer-based control applications.

Broader Significance

Beyond these specific contributions, this work demonstrates a broader methodological principle. The known fragility of conventional high-gain observers in the presence of real-world sensor noise can be overcome not by abandoning the observer architecture, but by augmenting it with a lightweight online-learning mechanism. The backstepping control law itself required no modification. The intelligent adaptation was introduced exclusively at the observation level, which preserved the proven theoretical guarantees of the control law while significantly improving the practical robustness of the closed-loop system. This outcome reflects a productive and complementary relationship between classical control theory and modern machine learning: neither discipline alone was sufficient to resolve the problem, but their considered combination produced a solution that is both theoretically grounded and experimentally validated on real hardware.

Limitations

While the experimental results are conclusive, certain limitations should be acknowledged. The stability of the closed-loop adaptive system has been demonstrated empirically but has not yet been established through a formal Lyapunov-based proof that accounts for the neural network weight dynamics. Additionally, the current architectures were validated

on a single benchmark platform, and their generalization to other nonlinear MIMO systems, while expected, remains to be demonstrated experimentally.

Perspectives

Several promising research directions emerge from this work.

Formal stability proof. The most important theoretical open question is the derivation of a rigorous Lyapunov-based stability proof for the complete closed-loop adaptive system, including the neural network weight update dynamics. Such a proof would provide a mathematical guarantee of stability that complements the experimental evidence presented here and would be indispensable for deploying this framework in safety-critical applications.

Advanced network architectures. Recurrent Neural Networks and Long Short-Term Memory networks are specifically designed to exploit temporal structure in sequential data. An LSTM-based gain tuner could learn the temporal patterns of the observation error and develop a predictive, rather than purely reactive, adaptation strategy, potentially enabling preemptive responses to predictable disturbances and producing smoother gain trajectories than those achievable with the static architectures studied here.

Hybrid offline and online training. The tuners presented in this thesis were trained purely online, which demonstrated their adaptability but required an initial convergence period before full performance was reached. Pre-training a network offline on a rich dataset of simulated operating scenarios and subsequently refining it online on the physical hardware could substantially reduce this initial transient and improve closed-loop performance from the earliest moments of operation.

Generalization to other nonlinear systems. The principle of using an online neural network to robustify a conventional observer is not specific to the TRMS. Applying this framework to other complex nonlinear MIMO systems, such as multi-jointed robotic manipulators, unmanned aerial vehicles navigating in turbulent conditions, or industrial process control plants with time-varying loads, would provide compelling evidence of its value as a general-purpose tool and would constitute a meaningful extension of the contributions presented in this work.

Bibliography

[1] Feedback Instruments Ltd. *Twin Rotor MIMO System Control Experiments Manual*. Documentation for TRMS testbed. Crowborough, East Sussex: Feedback Instruments Ltd., 2006.

[2] Ching-Long Shih, Mao-Lin Chen, and Jiun-Yaw Wang. "Mathematical model setpoint stabilizing controller design of a twin rotor MIMO system". In: *Asian journal of control* 10.1 (2008), pp. 107–114.

[3] Bidyadhar Subudhi and Debashisha Jena. "Nonlinear system identification of a twin rotor MIMO system". In: *TENCON 2009-2009 IEEE Region 10 Conference*. IEEE. 2009, pp. 1–6.

[4] Ayan Saha and Sarbani Chakraborty. "Genetic algorithm based I-PD controller design for Twin Rotor MIMO system". In: *2016 2nd International Conference on Control, Instrumentation, Energy & Communication (CIEC)*. IEEE. 2016, pp. 15–19.

[5] Santosh Kumar Choudhary. "Optimal feedback control of a twin rotor MIMO system". In: *International Journal of Modelling and Simulation* 37.1 (2017), pp. 46–53.

[6] Ankesh Kumar Agrawal. "Optimal Controller Design for Twin Rotor MIMO System". Master's thesis. Rourkela-769008, India: National Institute of Technology, June 2013.

[7] Abderrahmene Senoussaoui et al. "LQGi/LTR controller with integrators and feedforward controller applied to a Twin Rotor MIMO System". In: *Przegląd Elektrotechniczny* 97 (2021).

[8] Hem Prabha and Rajul Kumar. "Robust H_∞ Control Approach for Trajectory Tracking of Twin Rotor MIMO System". In: *2020 International Conference on Emerging Trends in Communication, Control and Computing (ICONC3)*. IEEE. 2020, pp. 1–4.

[9] Abasin Ulasyar and Haris Sheh Zad. "Robust & optimal model predictive controller design for twin rotor MIMO system". In: *2015 9th International Conference on Electrical and Electronics Engineering (ELECO)*. IEEE. 2015, pp. 854–858.

[10] Kelechi Ebirim. "Linear Model Predictive Control Implementations on a Twin Rotor MIMO System". PhD thesis. University of Leicester, 2024.

[11] Lakshmi Dutta and Dushmanta Kumar Das. “Multiple-model adaptive explicit predictive control for nonlinear MIMO system”. In: *Journal of Control and Decision* 12.4 (2025), pp. 695–710.

[12] Ramy Rashad, Ayman El-Badawy, and Ahmed Aboudonia. “Sliding mode disturbance observer-based control of a twin rotor MIMO system”. In: *ISA transactions* 69 (2017), pp. 166–174.

[13] Koteswara Rao Palepogu and Subhasish Mahapatra. “Design of sliding mode control with state varying gains for a Benchmark Twin Rotor MIMO System in Horizontal Motion”. In: *European Journal of Control* 75 (2024), p. 100909.

[14] Koteswara Rao Palepogu and Subhasish Mahapatra. “Pitch orientation control of twin-rotor MIMO system using sliding mode controller with state varying gains”. In: *Journal of Control and Decision* 11.2 (2024), pp. 211–221.

[15] Nguyen Xuan Chiem, Bui Xuan Hai, and TC Phan. “Synthesis of Adaptive Sliding Mode Control for Twin Rotor MIMO System with Mass Uncertainty based on Synergetic Control Theory.” In: *International Journal of Robotics & Control Systems* 4.1 (2024).

[16] Seyyed Sajjad Moosapour, Habib Mehdipour, and Mehrdad Keramatzadeh. “Sliding Mode Disturbance Observer-based Control of a laboratory Twin Rotor MIMO System (TRMS)”. In: *IEEE Access* (2024).

[17] Saqib Irfan et al. “URED Observer-Based Feedback Linearized Neuro Adaptive SMC for a Twin Rotor MIMO System: Design and Experimental Study”. In: *Guidance, Navigation and Control* (2025), pp. 1–25.

[18] Soufyane CHEKROUN et al. “Design and analysis of robust adaptive control for twin rotor MIMO systems using neuro-fuzzy inference techniques.” In: *Przegląd Elektrotechniczny* 2025.6 (2025).

[19] Samir Zeghlache et al. “Fault Tolerant Control Based on Direct Adaptive Fuzzy Position Controller for a Helicopter-Like Twin Rotor MIMO System: Design and Real Time Experimentation”. In: *Iranian Journal of Science and Technology, Transactions of Electrical Engineering* (2025), pp. 1–25.

[20] Petar V Kokotovic. “The joy of feedback: nonlinear and adaptive”. In: *IEEE control systems magazine* 12.3 (1992), pp. 7–17.

[21] Hassan K Khalil. *Nonlinear systems*. Vol. 3. Prentice hall, 2002.

[22] Miroslav Krstić, Ioannis Kanellakopoulos, and Petar V. Kokotović. *Nonlinear and Adaptive Control Design*. Wiley, 1995.

[23] Jing Zhou, Changyun Wen, and Ying Zhang. “Adaptive backstepping control of a class of uncertain nonlinear systems with unknown backlash-like hysteresis”. In: *IEEE transactions on Automatic Control* 49.10 (2004), pp. 1751–1759.

[24] Claude Kaddissi, Jean-Pierre Kenne, and Maarouf Saad. “Identification and real-time control of an electrohydraulic servo system based on nonlinear backstepping”. In: *IEEE/ASME Transactions on Mechatronics* 12.1 (2007), pp. 12–22.

[25] Muhammad Arsalan et al. “MPPT for photovoltaic system using nonlinear backstepping controller with integral action”. In: *Solar energy* 170 (2018), pp. 192–200.

[26] Youcef Saidi et al. “A comprehensive review of LVRT capability and advanced nonlinear backstepping control of grid-connected wind-turbine-driven permanent magnet synchronous generator during voltage dips”. In: *Journal of Control, Automation and Electrical Systems* 33.6 (2022), pp. 1773–1791.

[27] Lilan Karunaratne et al. “Nonlinear backstepping control of grid-forming converters in presence of grid-following converters and synchronous generators”. In: *IEEE Transactions on Power Systems* 39.1 (2023), pp. 1948–1964.

[28] Gan Yu et al. “Nonlinear backstepping control of a quadrotor-slung load system”. In: *IEEE/ASME Transactions On Mechatronics* 24.5 (2019), pp. 2304–2315.

[29] Abdullah Baraean et al. “Optimal Nonlinear backstepping controller design of a Quadrotor-Slung load system using particle Swarm Optimization”. In: *Alexandria Engineering Journal* 68 (2023), pp. 551–560.

[30] Joel Reis et al. “Nonlinear backstepping controller for an underactuated ASV with model parametric uncertainty: Design and experimental validation”. In: *IEEE Transactions on Intelligent Vehicles* 8.3 (2022), pp. 2514–2526.

[31] Lin Zhang et al. “An adaptive backstepping sliding mode controller to improve vehicle maneuverability and stability via torque vectoring control”. In: *IEEE Transactions on Vehicular Technology* 69.3 (2020), pp. 2598–2612.

[32] Dinh-Hieu Phan et al. “Sliding Mode Control Using Genetic Algorithm for Twin Rotor MIMO System”. In: *International Conference on Sustainability and Emerging Technologies for Smart Manufacturing*. Springer. 2024, pp. 671–678.

[33] Paloma Sodhi and Indrani Kar. “Adaptive backstepping control for a twin rotor MIMO system”. In: *IFAC Proceedings Volumes* 47.1 (2014), pp. 740–747.

[34] Samiran Maiti and Achintya Das. “Design of one nonlinear controller for a MIMO system using adaptive backstepping method”. In: *International Conference on Application of Robotics in Industry using Advanced Mechanisms*. Springer. 2019, pp. 14–25.

[35] Messaoudi Khaled and Sofiane Doudou. “Robust Prescribed Time Backstepping for Stabilization and Tracking of 2-DOF Twin Rotor Model”. In: *2024 International Conference on Advances in Electrical and Communication Technologies (ICAECOT)*. IEEE. 2024, pp. 1–5.

[36] Mehran Derakhshannia, Sayed Bagher Fazeli Asl, and Seyyed Sajjad Moosapour. “Backstepping terminal sliding mode control design for a TRMS”. In: *2021 7th International Conference on Control, Instrumentation and Automation (ICCIA)*. IEEE. 2021, pp. 1–5.

[37] Sinan Yigit and Aziz Sezgin. “Trajectory tracking via backstepping controller with PID or SMC for mobile robots”. In: *Sakarya University Journal of Science* 27.1 (2023), pp. 120–134.

[38] Benyettou Loutfi et al. “Real time implementation of type-2 fuzzy backstepping sliding mode controller for twin rotor MIMO system (TRMS)”. In: *Traitement du Signal* 36.1 (2019), pp. 1–11.

[39] Saeed Amiri and Saleh Mobayen. “Optimal Adaptive Back-Stepping SMC Synthesis via the Fuzzy Model: Applications to a Two-Degree-of-Freedom Helicopter System”. In: *Optimal Control Applications and Methods* (2025).

[40] Boyan Zhu et al. “Neural network-based adaptive reinforcement learning for optimized backstepping tracking control of nonlinear systems with input delay”. In: *Applied Intelligence* 55.2 (2025), p. 129.

[41] Xiaolong Zheng et al. “Adaptive neural zeta-backstepping with predefined damping ratio. Application to DC motors”. In: *IEEE Transactions on Cybernetics* (2025).

[42] Umair Hussan et al. “Robust Maximum Power Point Tracking in PV Generation System: A Hybrid ANN-Backstepping Approach With PSO-GA Optimization”. In: *IEEE Transactions on Consumer Electronics* (2025).

[43] Guy Bornard and Hassan Hammouri. “A high gain observer for a class of uniformly observable systems”. In: *[1991] Proceedings of the 30th IEEE Conference on Decision and Control*. IEEE. 1991, pp. 1494–1496.

[44] Jean-Paul Gauthier, Hassan Hammouri, and Salah Othman. “A simple observer for nonlinear systems with application to bioreactors”. In: *IEEE Transactions on automatic control* 37.6 (1992), pp. 875–880.

[45] Antonio Tornambè. “High-gain observers for non-linear systems”. In: *International Journal of Systems Science* 23.9 (1992), pp. 1475–1489.

[46] Hassan K Khalil and Laurent Praly. “High-gain observers in nonlinear feedback control”. In: *International Journal of Robust and Nonlinear Control* 24.6 (2014), pp. 993–1015.

[47] Hassan K Khalil. *High-Gain Observers in Nonlinear Feedback Control*. Philadelphia: Society for Industrial and Applied Mathematics, 2017.

[48] AN Atassi and HK Khalil. “Separation results for the stabilization of nonlinear systems using different high-gain observer designs”. In: *Systems & Control Letters* 39.3 (2000), pp. 183–191.

[49] Jianxing Liu et al. “Sliding mode control of grid-connected neutral-point-clamped converters via high-gain observer”. In: *IEEE Transactions on Industrial Electronics* 69.4 (2021), pp. 4010–4021.

[50] Soukaina Nady et al. “High-Gain Observer in Fuel Cell and Cascaded DC-DC Boost Converter System”. In: *E3S Web of Conferences*. Vol. 601. EDP Sciences. 2025, p. 00062.

[51] Connor J Boss and Vaibhav Srivastava. “A High-Gain Observer Approach to Robust Trajectory Estimation and Tracking for a Multirotor Unmanned Aerial Vehicle”. In: *Journal of Dynamic Systems, Measurement, and Control* 147.1 (2025), p. 011005.

[52] Alexis A Prasov and Hassan K Khalil. “A nonlinear high-gain observer for systems with measurement noise in a feedback control framework”. In: *IEEE Transactions on Automatic Control* 58.3 (2012), pp. 569–580.

[53] E Busvelle and JP Gauthier. “High-gain and non-high-gain observers for nonlinear systems”. In: *Contemporary trends in nonlinear geometric control theory and its applications*. World Scientific, 2002, pp. 257–286.

[54] Seungrohk Oh and Hassan K Khalil. “Nonlinear output-feedback tracking using high-gain observer and variable structure control”. In: *Automatica* 33.10 (1997), pp. 1845–1856.

[55] Iman Dehbozorgi and Alireza Khayatian. “Impulsive high-gain observer for nonlinear systems to reduce peaking phenomenon”. In: *International Journal of Systems Science* (2025), pp. 1–15.

[56] M Farza et al. “Improved high gain observer design for a class of disturbed nonlinear systems”. In: *Nonlinear Dynamics* 106.1 (2021), pp. 631–655.

[57] Ali Zemouche et al. “High-gain nonlinear observer with lower tuning parameter”. In: *IEEE Transactions on Automatic Control* 64.8 (2018), pp. 3194–3209.

[58] M Farza et al. “Design of a Saturated Filtered High Gain Observer for a Class of Disturbed Nonlinear Systems”. In: *International Journal of Robust and Nonlinear Control* (2025).

[59] Nicolas Boizot, Eric Busvelle, and Jean-Paul Gauthier. “An adaptive high-gain observer for nonlinear systems”. In: *Automatica* 46.9 (2010), pp. 1483–1488.

[60] Eric Bullinger and Frank Allgower. “An adaptive high-gain observer for nonlinear systems”. In: *Proceedings of the 36th IEEE Conference on Decision and Control*. Vol. 5. IEEE. 1997, pp. 4348–4353.

[61] El-Sayed M Ahmed and M Mohamed. “PID controller tuning scheme for twin rotor multi-input multi-output system based particle swarm optimization approach”. In: *JES. Journal of Engineering Sciences* 37.4 (2009), pp. 955–967.

[62] Roshni Maiti, Kaushik Das Sharma, and Gautam Sarkar. “PSO based parameter estimation and PID controller tuning for 2-DOF nonlinear twin rotor MIMO system”. In: *International Journal of Automation and Control* 12.4 (2018), pp. 582–609.

[63] Ahmad Taher Azar et al. “PID controller for 2-DOFs twin rotor MIMO system tuned with particle swarm optimization”. In: *international conference on advanced intelligent systems and informatics*. Springer. 2019, pp. 229–242.

[64] Barış Çelebi and Boğaç Bilgiç. “Optimizing TRMS stability: a multi-objective genetic algorithm approach to PID controller design”. In: *Engineering Computations* 42.2 (2025), pp. 710–721.

[65] Arpit Jain, Satya Sheel, and Piyush Kuchhal. “Fuzzy logic-based real-time control for a twin-rotor MIMO system using GA-based optimization”. In: *World Journal of Engineering* 15.2 (2018), pp. 192–204.

[66] Fadi ALYOUSSEF and KAYA Ibrahim. “TRMS experimental results of new nonlinear PID tuned by DE algorithm”. In: *2019 International Conference on Applied Automation and Industrial Diagnostics (ICAAID)*. Vol. 1. IEEE. 2019, pp. 1–6.

[67] Sudarshan K Valluru, Rajul Kumar, and Rahul Kumar. “Design and Real Time implementation of fmincon, MOGA tuned IO-PID and FO-PIλDμ Controllers for Stabilization of TRMS”. In: *Procedia Computer Science* 171 (2020), pp. 1241–1250.

[68] Ali Can Cabuker and Mehmet Nuri Almalı. “Metaheuristic Algorithm-Based Proportional–Integrative–Derivative Control of a Twin Rotor Multi Input Multi Output System”. In: *Electronics* 13.16 (2024), p. 3291.

[69] Samir Zeghlache et al. “Twin rotor MIMO system experimental validation of robust adaptive fuzzy control against wind effects”. In: *IEEE Systems Journal* 16.1 (2020), pp. 409–419.

[70] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[71] W Koch, R Müller, and UD Hanebeck. “A review on reinforcement learning for robot control”. In: *IFAC-PapersOnLine* 52.13 (2019), pp. 217–224.

[72] Jemin Hwangbo et al. “Learning agile and dynamic motor skills for legged robots”. In: *Science robotics* 4.26 (2019), eaau5872.

[73] Elia Kaufmann et al. “Champion-level drone racing using deep reinforcement learning”. In: *Nature* 609.7927 (2022), pp. 494–498.

[74] Oussama Bello, Andrea L’Afflitto, and Fábio Bœing. “Deep reinforcement learning for an autorotation maneuver of a helicopter”. In: *IEEE Access* 9 (2021), pp. 111874–111885.

[75] Marek Krok et al. “Application of feedforward and recurrent neural networks for model-based control systems”. In: *Control Theory and Technology* 23.1 (2025), pp. 91–104.

[76] Paolo Pagliuca and Davide Yuri Inglese. “The Importance of Functionality over Complexity: A Preliminary Study on Feed-Forward Neural Networks”. In: *Advanced Neural Artificial Intelligence: Theories and Applications*. Springer, 2025, pp. 447–458.

[77] John G Kuschewski, Stefen Hui, and Stanislaw H Zak. “Application of feedforward neural networks to dynamical system identification and control”. In: *IEEE transactions on control systems technology* 1.1 (1993), pp. 37–49.

[78] Thomas Parisini and Riccardo Zoppoli. “Neural networks for feedback feedforward nonlinear control systems”. In: *IEEE Transactions on Neural Networks* 5.3 (1994), pp. 436–449.

[79] Wen-Qi Wang et al. “A feedforward-feedback control strategy based on artificial neural network for solar receivers”. In: *Applied Thermal Engineering* 224 (2023), p. 120069.

[80] Fayçal Mehedi et al. “Feedforward neural network-DTC of multi-phase permanent magnet synchronous motor using five-phase neural space vector pulse width modulation strategy”. In: *Journal Européen des Systèmes Automatisés* 54.2 (2021), pp. 345–354.

[81] Arief Abdurrakhman et al. “A Multilayer Perceptron Feedforward Neural Network and Particle Swarm Optimization Algorithm for Optimizing Biogas Production”. In: *Energies* 18.4 (2025), p. 1002.

[82] Hadjer Benfatma, Houari Khouidmi, and Boubaker Bessedik. “Neural network and ACO algorithm-tuned PI controller for MPPT in a hybrid battery-supercapacitor energy storage system within DC micro-grid photovoltaic installations”. In: *Journal of Energy Storage* 120 (2025), p. 116499.

[83] Juntao Fei and Hongfei Ding. “Adaptive sliding mode control of dynamic system using RBF neural network”. In: *Nonlinear Dynamics* 70.2 (2012), pp. 1563–1573.

[84] Chao Jia, Xuanyue Shangguan, and Linxin Zheng. “Dynamic boundary layer super-twisting sliding mode control algorithm based on RBF neural networks for a class of leader-follower multi-agent systems”. In: *International Journal of Robust and Nonlinear Control* 34.3 (2024), pp. 2109–2140.

[85] Ming Guan et al. “A novel RBF neural network–based sliding mode controller for a master–slave motor coordinated drive system”. In: *The International Journal of Advanced Manufacturing Technology* 133.9 (2024), pp. 4907–4921.

[86] Qiong Liu et al. “Adaptive bias RBF neural network control for a robotic manipulator”. In: *Neurocomputing* 447 (2021), pp. 213–223.

[87] Bowen Su, Fan Zhang, and Panfeng Huang. “Stability analysis and RBF neural network control of second-order nonlinear satellite system”. In: *IEEE Transactions on Aerospace and Electronic Systems* 59.4 (2023), pp. 4575–4589.

[88] Amir Veisi and Hadi Delavari. “Fuzzy-type 2 fractional fault tolerant adaptive controller for wind turbine based on adaptive RBF neural network observer”. In: *Soft Computing* 28.17 (2024), pp. 10689–10700.

[89] Yahui Li et al. “Robust and adaptive backstepping control for nonlinear systems using RBF neural networks”. In: *IEEE Transactions on Neural Networks* 15.3 (2004), pp. 693–701.

[90] Xiaoyu Shi et al. “Design of adaptive backstepping dynamic surface control method with RBF neural network for uncertain nonlinear system”. In: *Neurocomputing* 330 (2019), pp. 490–503.

[91] Xudong Zhao et al. “Adaptive neural backstepping control design for a class of nonsmooth nonlinear systems”. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 49.9 (2018), pp. 1820–1831.

[92] Son Tung Dang et al. “Adaptive backstepping hierarchical sliding mode control for 3-wheeled mobile robots based on RBF neural networks”. In: *Electronics* 12.11 (2023), p. 2345.

[93] Mohammed Zinelaabidine Ghellab et al. “Experimental validation of adaptive RBFNN global fast dynamic terminal sliding mode control for twin rotor MIMO system against wind effects”. In: *Measurement* 168 (2021), p. 108472.

[94] Senoussaoui Abderrahmene et al. “Neural network NARMA-L2 control of a Twin Rotor MIMO System”. In: *2019 International Conference on Advanced Electrical Engineering (ICAEE)*. IEEE. 2019, pp. 1–6.

[95] Ferdose Ahammad Shaik, Shubhi Purwar, and Bhanu Pratap. “Real-time implementation of Chebyshev neural network observer for twin rotor control system”. In: *Expert Systems with Applications* 38.10 (2011), pp. 13043–13049.

[96] Bhanu Pratap and Shubhi Purwar. “Real-time implementation of neuro adaptive observer-based robust backstepping controller for twin rotor control system”. In: *Journal of Control, Automation and Electrical Systems* 25.2 (2014), pp. 137–150.

[97] Lai Khac Lai et al. “The New Method to Control Twin Rotor MIMO System (TRMS)”. In: *International Conference on Engineering Research and Applications*. Springer. 2020, pp. 810–819.

[98] Tarek Madani and Abdelaziz Benallegue. “Backstepping Control for a Quadrotor Helicopter”. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2006, pp. 3255–3260. DOI: 10.1109/IROS.2006.282433.

[99] Samir Bouabdallah. “Design and Control of Quadrotors with Application to Autonomous Flying”. PhD thesis. École Polytechnique Fédérale de Lausanne (EPFL), 2007.

[100] Jie Zhou and Changyun Wen. *Adaptive Backstepping Control of Uncertain Systems*. Berlin, Heidelberg: Springer, 2008. DOI: 10.1007/978-3-540-77890-9.

[101] Warren S McCulloch and Walter Pitts. “A logical calculus of the ideas immanent in nervous activity”. In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.

[102] Frank Rosenblatt. “The perceptron: a probabilistic model for information storage and organization in the brain.” In: *Psychological review* 65.6 (1958), p. 386.

[103] Marvin Minsky and Seymour Papert. *Perceptrons: An Introduction to Computational Geometry*. MIT Press, 1969.

[104] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. “Learning representations by back-propagating errors”. In: *Nature* 323.6088 (1986), pp. 533–536.

[105] Terrence L. Fine. *Feedforward Neural Network Methodology*. Statistics for Engineering and Information Science. New York: Springer, 1999. ISBN: 0-387-98745-2.

[106] David S Broomhead and David Lowe. “Multivariable functional interpolation and adaptive networks”. In: *Complex Systems* 2 (1988), pp. 321–355.

[107] J. Moody and C.J. Darken. “Fast learning in networks of locally-tuned processing units”. In: *Neural Computation* 1.2 (1989), pp. 281–294.

[108] Azeddine Beloufa et al. “Robust Backstepping Control of a Twin Rotor MIMO System via an RBF-Tuned High-Gain Observer”. In: *Automation* 7.2 (2026), p. 40.