Comparative Study of LQR and Neural Network Controllers for Quadcopter Roll Angle Stabilization

Submitted: 4th September 2025; accepted: 2nd February 2026

Hiba Arif, Marouane Kadi, Aymane Bidah, Omar Zakary

DOI: 10.14313/jamris-2026-036

Abstract:
This paper presents a comparative analysis between a conventional Linear Quadratic Regulator (LQR) and a neural network-based approach for quadcopter roll angle stabilization. While the LQR controller delivers optimal performance in nominal conditions, its accuracy deteriorates when subjected to model uncertainties and persistent external disturbances, causing steady-state deviations.

To mitigate these issues, we develop a neural network controller trained via imitation learning. The proposed architecture inherently learns an integral action, enabling perfect rejection of constant disturbances. Extensive simulations show that our neural controller achieves superior performance in realistic operating conditions while maintaining similar stability margins to LQR in ideal cases.

This study demonstrates how machine learning techniques can enhance classical control paradigms for aerial robotics. The results suggest that neural network controllers offer a viable path toward more adaptive and robust autonomous drone systems in challenging environments.

Keywords: *Quadrotor control, Linear Quadratic Regulation, Deep learning, UAV stabilization, Robust control, Behavioral cloning*

1. Introduction

Modern unmanned aerial systems, particularly quadrotor platforms, have seen exponential growth in applications across surveillance, precision agriculture, and disaster management [5]. These versatile systems nevertheless present significant control challenges due to their inherent nonlinear dynamics, coupled states, and open-loop instability [7]. Effective disturbance rejection and precise attitude maintenance under environmental uncertainties remain critical requirements for their operational reliability.

The quest for optimal control strategies has led to extensive research into various methodologies. While conventional PID controllers remain popular for their simplicity, modern optimal control approaches like the Linear Quadratic Regulator (LQR) have gained prominence for multivariable systems [3]. LQR's theoretical foundation in state-space optimization makes it particularly suitable for aerial vehicle control [1]. By minimizing a quadratic cost function that balances state regulation against control effort, LQR generates optimal feedback gains that ensure system stability while considering multiple state variables simultaneously [2].

Despite its mathematical elegance, LQR exhibits several practical limitations [4]. Its performance assumptions - perfect system modeling, linear dynamics, and Gaussian noise - rarely hold in real-world flight conditions. These constraints manifest particularly in persistent steady-state errors and inadequate disturbance rejection, compromising the quadcopter's positioning accuracy [16]. The controller's static gain structure further limits its adaptability to varying operational conditions.

Recent advances in machine learning have introduced new paradigms for control system design. Neural networks, with their universal approximation capabilities, offer compelling alternatives by learning control policies directly from operational data [8]. This data-driven approach circumvents the need for precise system modeling while maintaining robustness against uncertainties. Hybrid architectures combining LQR with neural components (LQR-RNN) have demonstrated particular promise, merging classical control theory with adaptive learning [9]. Such systems leverage LQR's optimization framework while employing recurrent networks (RNNs) to dynamically adjust control parameters based on real-time performance [11].

This work presents a comparative evaluation of classical LQR and neural network-based controllers for quadrotor roll stabilization. Focusing on the fundamental roll dynamics described by states $x_1 = \phi(t)$ and $x_2 = \dot{\phi}(t)$ [6], we examine each controller's performance under modeled disturbances and parametric uncertainties. Our simulation-based analysis highlights the relative strengths of each approach, with particular emphasis on the neural controller's potential advantages in realistic operational scenarios.

2. Theoretical Background

Modern technological progress in quadrotor UAV systems has transformed numerous industrial and commercial domains, including cinematography, delivery services, and precision farming [5]. At the core of these applications lies the critical need for reliable flight control architectures that ensure operational safety and performance consistency [6]. Among various control challenges, maintaining precise attitude regulation under environmental disturbances remains a fundamental research problem in aerial robotics [7].

Roll angle dynamics (ϕ), representing the rotational motion about the longitudinal axis, constitute a fundamental component of quadrotor stabilization [16]. Conventional control methodologies, particularly linear approaches, remain popular due to their well-established theoretical framework. The Linear Quadratic Regulator stands out in this category, offering an optimal control solution through quadratic optimization of state deviations and control inputs [1]. However, this optimality is strictly contingent upon accurate system modeling and the absence of unaccounted disturbances - conditions seldom met in practical implementations [4].

Recent developments in artificial intelligence have introduced novel control paradigms through machine learning techniques. Neural network-based controllers present distinct advantages by learning control policies directly from operational data rather than relying on explicit system models [9]. This characteristic enables them to capture complex nonlinear relationships and adapt to various uncertainties, including parameter variations (e.g., changes in moment of inertia) and external perturbations (such as wind gusts) [17]. Reinforcement learning frameworks further enhance this adaptability through continuous performance optimization [10].

Our research conducts a systematic performance comparison between conventional LQR and neural network-based controllers for roll angle stabilization. The study demonstrates that while LQR achieves theoretical optimality under ideal conditions, its performance deteriorates significantly when facing persistent disturbances and modeling inaccuracies. In contrast, properly trained neural controllers exhibit superior robustness and disturbance rejection capabilities, as evidenced through extensive simulation scenarios [12].

The paper is organized as follows: Section 2 develops the system's mathematical model. Section 3 elaborates on both control architectures (LQR and neural network). Section 4 presents comparative simulation results and analysis, followed by concluding remarks on the findings' significance.

3. System Modeling

For the purpose of control system design and analysis, we consider a simplified representation of the quadrotor dynamics by focusing exclusively on its rotational behavior about the longitudinal axis (roll angle ϕ). This common simplification in aerial vehicle control studies enables focused examination of a crucial degree of freedom while maintaining analytical tractability [6].

4. System Dynamics Formulation

In developing our control framework, we adopt a simplified dynamic model that captures the essential roll dynamics (ϕ) of the quadrotor system. This single-axis approximation, frequently employed in control literature [5], provides several advantages:

- Isolation of a fundamental stabilization challenge
- Reduction of system complexity while preserving key dynamic characteristics
- Enables clearer comparative analysis of control approaches

4.1. Roll Dynamics Formulation

The rotational motion of a quadrotor around its longitudinal axis follows the fundamental principles of rigid body dynamics, specifically Euler's rotation equation derived from Newton's second law for rotational systems [16]. When considering only the x-axis rotation, we obtain the governing differential equation:

$$I_{xx}\ddot{\phi}(t) = \tau_x(t) + \delta(t) \tag{1}$$

where the physical quantities are defined as:

- $\phi(t)$: Roll angle (rad) - the primary controlled output variable
- $\ddot{\phi}(t)$: Angular acceleration (rad/s^2) - second time derivative of roll angle
- I_{xx}: Moment of inertia about x-axis (kg·m^2) - characterizes the vehicle's rotational inertia [7]
- $\tau_x(t)$ (or $u(t)$): Control torque input (N·m) - generated by differential thrust from rotors
- $\delta(t)$: Disturbance torque (N·m) - aggregates various unmodeled effects including:
  - Aerodynamic disturbances (wind gusts, ground effects)
  - System imperfections (motor imbalances, structural asymmetries) [17]
  - Measurement noise and modeling inaccuracies

4.2. State-Space Formulation

Modern control theory extensively employs state-space representations as they provide a comprehensive framework for analyzing and designing control systems using linear algebra techniques [3]. For our second-order roll dynamics system, we define the state vector $\mathrm{x}(t)$ to completely characterize the system's dynamic behavior:

$$\mathrm{x}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \begin{bmatrix} \phi(t) \\ \dot{\phi}(t) \end{bmatrix} \tag{2}$$

where $x_1(t)$ represents the roll angle and $x_2(t)$ corresponds to the angular velocity. Through differentiation and substitution of the governing dynamics (1), we derive the following coupled first-order differential equations:

$$\dot{x}_1(t) = x_2(t) \tag{3}$$

$$\dot{x}_2(t) = \frac{1}{I_{xx}} u(t) + \frac{1}{I_{xx}} \delta(t) \tag{4}$$

These equations can be compactly expressed in matrix form as:

$$\dot{\mathrm{x}}(t) = \mathrm{A}\mathrm{x}(t) + \mathrm{B}u(t) + \mathrm{E}\delta(t) \tag{5}$$

with the system matrices defined as:

$$\mathrm{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \mathrm{B} = \begin{bmatrix} 0 \\ \frac{1}{I_{xx}} \end{bmatrix}, \quad \mathrm{E} = \begin{bmatrix} 0 \\ \frac{1}{I_{xx}} \end{bmatrix}$$

The state matrix A captures the inherent system dynamics, while the input matrix B describes how the control torque $u(t)$ affects the state evolution. The disturbance matrix E quantifies the impact of external perturbations on the system. This formulation provides the mathematical foundation for both conventional and neural network-based controller designs [1].

5. Control System Architectures

Building upon the established dynamic model, this section develops two distinct control methodologies for quadrotor roll stabilization. The first represents a well-established model-based optimal control technique (LQR), while the second employs a contemporary data-driven paradigm using artificial neural networks.

- **Linear Quadratic Regulator (LQR)**: A classical optimal control approach that requires precise system modeling
- **Neural Network Controller**: A modern machine learning technique that learns control policies from operational data

This comparative study examines both approaches' theoretical foundations and implementation strategies, highlighting their respective advantages and limitations for aerial vehicle stabilization.

5.1. LQR Control Methodology

Theoretical Foundation: The Linear Quadratic Regulator represents a cornerstone of optimal control theory for linear time-invariant systems [2]. This approach computes an optimal state feedback control law $u(t) = -\mathrm{K}\mathrm{x}(t)$ by minimizing a quadratic performance index:

$$J = \int_0^\infty \left(\mathrm{x}^T(t)\mathrm{Q}\mathrm{x}(t) + u^T(t)\mathrm{R}u(t)\right) dt \tag{6}$$

The weighting matrices in the cost function serve distinct purposes:

- Q (positive semi-definite): Governs state regulation performance
  - Diagonal elements determine relative importance of different states
  - Larger values enforce faster convergence to equilibrium [1]
- R (positive definite): Controls input energy expenditure
  - Higher values prioritize control efficiency over response speed [4]
  - Prevents actuator saturation in practical implementations

For our roll stabilization problem, we employ diagonal weighting matrices:

$$\mathrm{Q} = \begin{bmatrix} q_{11} & 0 \\ 0 & q_{22} \end{bmatrix}, \quad \mathrm{R} = [r] \tag{7}$$

where q_{11} and q_{22} respectively weight roll angle (ϕ) and angular velocity ($\dot{\phi}$) errors. The optimal gain matrix K is derived through solution of the Algebraic Riccati Equation (ARE), a fundamental result in optimal control theory [3].

Inherent Limitations of LQR Control: While the LQR controller offers theoretical optimality for linear systems, its practical implementation faces several fundamental constraints [4]:

1) **Linearity Assumption**:
   - Designed exclusively for linear system dynamics
   - Performance degrades with nonlinear effects (e.g., aerodynamic drag, motor saturation) [16]
   - Local linearization may not capture full operational envelope behavior
2) **Model Dependency**:
   - Optimality requires exact knowledge of system parameters (I_{xx} in our case)
   - Inaccurate inertia estimates lead to suboptimal gain selection [3]
   - Sensitivity to unmodeled dynamics and parameter variations
3) **Disturbance Rejection Deficiency**:
   - Pure state feedback lacks inherent integral action
   - Persistent disturbances cause steady-state offsets [1]
   - Limited robustness to constant torque disturbances

These structural limitations, particularly the inability to achieve perfect disturbance rejection, necessitate the development of more robust control paradigms that can maintain performance under realistic operating conditions.

5.2. Neural Network-Based Control Architecture

Conceptual Framework: Motivated by the identified limitations of model-based LQR control, we develop an alternative approach leveraging artificial neural networks (ANNs). This data-driven methodology aims to:

- Learn optimal control policies directly from operational experience rather than analytical models

- Capture complex nonlinear relationships inherent in real-world quadrotor dynamics
- Automatically adapt to system uncertainties and disturbances

The proposed architecture employs supervised learning to approximate an enhanced control law that combines the strengths of conventional state feedback with learned disturbance rejection capabilities. Unlike the LQR's fixed-gain structure, the neural controller develops adaptable mappings between system states and control actions through exposure to diverse operational scenarios.

Expert Controller Design and Behavioral Cloning:

Training Strategy: Our approach employs imitation learning, requiring an expert controller capable of optimal performance across diverse operational scenarios. This expert must overcome the LQR's key limitation by achieving perfect disturbance rejection through integral action.

LQI Controller Architecture: We implement a Linear Quadratic Integral (LQI) controller as our expert system. The state vector is augmented with an integral term:

$$\mathrm{x}_{\mathrm{aug}}(t) = \begin{bmatrix} \mathrm{x}(t) \\ \xi(t) \end{bmatrix} = \begin{bmatrix} \phi(t) \\ \dot{\phi}(t) \\ \int_0^t \phi(\tau)d\tau \end{bmatrix} \tag{8}$$

The extended system dynamics become:

$$\dot{\mathrm{x}}_{\mathrm{aug}}(t) = \mathrm{A}_{\mathrm{aug}}\mathrm{x}_{\mathrm{aug}}(t) + \mathrm{B}_{\mathrm{aug}}u(t) \tag{9}$$

where:

$$\mathrm{A}_{\mathrm{aug}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \mathrm{B}_{\mathrm{aug}} = \begin{bmatrix} 0 \\ 1/I_{xx} \\ 0 \end{bmatrix} \tag{10}$$

Optimal Control Formulation: The LQI controller minimizes an augmented cost function:

$$J_{\mathrm{aug}} = \int_0^\infty \left(\mathrm{x}_{\mathrm{aug}}^T(t)\mathrm{Q}_{\mathrm{aug}}\mathrm{x}_{\mathrm{aug}}(t) + u^T(t)\mathrm{R}_{\mathrm{aug}}u(t)\right) dt \tag{11}$$

The solution yields an optimal gain matrix:

$$\mathrm{K}_{\mathrm{aug}} = \begin{bmatrix} \mathrm{K}_p & K_i \end{bmatrix} = \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix} \tag{12}$$

implementing the control law:

$$u_{\mathrm{expert}}(t) = -\mathrm{K}_p\mathrm{x}(t) - K_i\xi(t) \tag{13}$$

Data Generation Process: The training dataset is generated through extensive simulations that systematically vary:

- Moment of inertia (I_{xx})
- Initial conditions
- Disturbance profiles

For each scenario, the expert controller computes optimal control actions, creating state-action pairs $(\mathrm{x}_{\mathrm{aug}}, u_{\mathrm{expert}})$. The neural network learns to approximate the mapping $f : \mathrm{x}_{\mathrm{aug}} \to u_{\mathrm{expert}}$ across the entire operational envelope.

Network Architecture and Training Methodology: The proposed controller employs a multilayer perceptron (MLP) structure, designed according to established deep learning principles [8]:

- **Input Layer**:
  - Receives the 3-dimensional augmented state vector $(\phi, \dot{\phi}, \int \phi dt)$
  - Normalizes inputs to ensure stable training dynamics
- **Hidden Layers**:
  - Two fully-connected layers with 64 units each
  - ReLU activation functions providing necessary nonlinear transformations [10]
  - Batch normalization between layers for improved training stability
- **Output Layer**:
  - Single linear neuron generating torque commands $u(t)$
  - No activation function to allow unbounded control outputs

The training process minimizes the mean squared error (MSE) between predicted and expert control actions using the Adam optimization algorithm [9]. Key training characteristics include:

- Large-scale exposure to diverse operational scenarios
- Iterative weight updates through backpropagation
- Learning rate scheduling for convergence refinement
- Early stopping based on validation performance

Through this data-driven approach, the network develops an adaptive control policy that generalizes across various operating conditions and disturbance profiles, demonstrating superior robustness compared to conventional LQR methods [12].

6. Performance Evaluation and Comparative Analysis

To rigorously assess the effectiveness of both control strategies, we implemented a comprehensive simulation study under demanding operational conditions. This test scenario was specifically designed to:

- Expose the inherent vulnerabilities of conventional LQR control
- Validate the enhanced robustness of our neural network-based approach
- Provide quantitative comparison metrics under identical test conditions

The experimental framework incorporates multiple challenging aspects including parameter uncertainties, persistent disturbances, and nonlinear effects, thereby creating a realistic evaluation environment for both control paradigms.

6.1. Experimental Configuration

The comparative evaluation employs the following test conditions to rigorously assess controller performance:

- **Parameter Uncertainty**:
  - LQR design based on nominal inertia $I_{x,\text{nominal}} = 0.1$ kg·m^2
  - Actual system inertia 50% higher ($I_{x,\text{actual}} = 0.15$ kg·m^2)
  - Represents realistic payload variations (e.g., equipment changes) [15]
- **External Disturbances**:
  - Initial equilibrium condition ($\phi = 0, \dot{\phi} = 0$)
  - Step disturbance $d(t) = 0.5$ N·m applied at $t = 1$s
  - Models sustained aerodynamic forces (e.g., crosswinds) [17]

The evaluation compares two distinct control architectures:

- Conventional LQR (2-state feedback)
- ANN-based controller (3-state feedback with integral action) [6]

6.2. Performance Evaluation

The comparative analysis of both control strategies under the specified test conditions yields the following observations. **Disturbance Rejection Capabilities**

Figure 1 demonstrates the controllers' behavior when simultaneously subjected to parameter uncertainty and external disturbances.

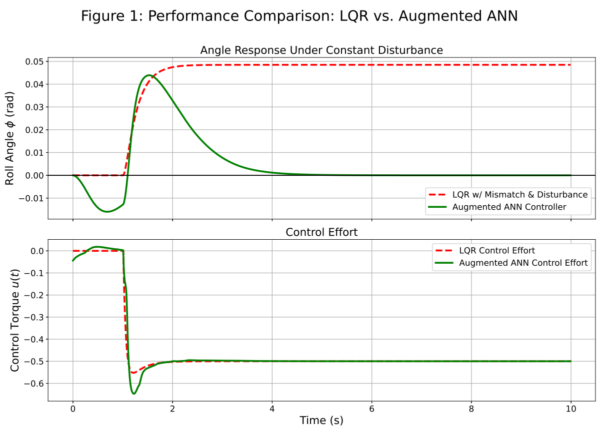

Figure 1. Comparative response of LQR and neural network controllers under model mismatch and constant disturbance conditions

LQR Controller Performance The conventional LQR approach (shown in red dashed lines) exhibits characteristic limitations predicted by control theory [3]:

- Upon disturbance application at $t = 1$s, the system develops a persistent steady-state error of 0.048 rad
- The control input converges to -0.5 N·m, exactly counterbalancing the disturbance but requiring a non-zero angular deviation [1]
- This behavior confirms the theoretical inability of pure state feedback to completely reject constant disturbances

Neural Network Controller Performance The ANN-based controller (green solid lines) demonstrates notable advantages:

- **Transient Phase**:
  - More aggressive initial response with maximum deviation limited to 0.044 rad
  - Higher control authority utilization (peaking at -0.65 N·m)
  - Faster error correction dynamics [10]
- **Steady-State Behavior**:
  - Complete disturbance rejection achieved by $t = 4$s
  - Final control input matches the disturbance magnitude (-0.5 N·m)
  - Perfect setpoint recovery demonstrating learned integral action [12]

An initial transient artifact visible before $t = 1$s warrants further investigation in subsequent analysis (see Figure 2) [8].

Regulation Performance in Nominal Conditions Figure 2 examines the controllers' stabilization capabilities from a non-zero initial condition ($\phi(0) = 0.5$ rad) without external disturbances [6].

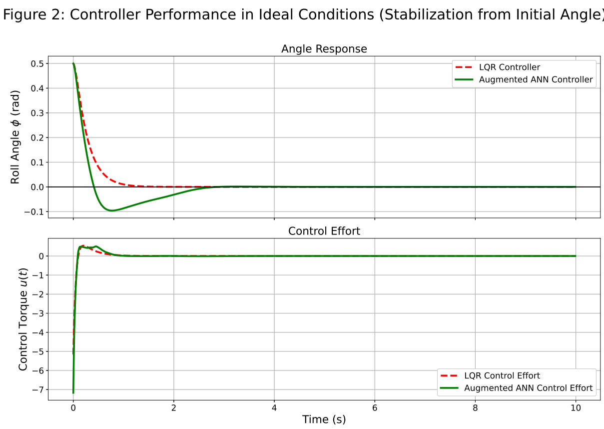

Figure 2. Comparative regulation performance in disturbance-free conditions

Reference LQR Performance The LQR controller (red dashed line) demonstrates its theoretical optimality for linear systems [2]:

- Critically damped response with no overshoot
- Minimum-time convergence given the system constraints
- Smooth control input trajectory

Neural Network Controller Behavior The ANN controller (green solid line) exhibits remarkable characteristics:
- Near-perfect emulation of optimal LQR response [9]
- Identical convergence rate and stability properties
- Superimposed state and control trajectories [10]

Key observations:
- The transient artifact present in Figure 1 disappears completely
- Confirms the network's ability to switch between:
  - Disturbance rejection mode (Figure 1)
  - Optimal regulation mode (Figure 2) [8]
- Demonstrates contextual adaptation based on system conditions [1]

Robustness Evaluation Across Parameter Variations
To systematically assess controller performance under diverse operating conditions, we performed a parametric study examining:
- Inertia variations: $I_x \in [0.10, 0.25]$ kg $\cdot$ m^2 (representing $\pm 50\%$ nominal value)
- Disturbance range: $d \in [0.0, 0.8]$ N $\cdot$ m (covering mild to severe perturbations)

Figure 3 presents the comprehensive evaluation matrix, with:
- Rows corresponding to different inertia values
- Columns representing increasing disturbance magnitudes
- ANN responses shown in solid green trajectories
- LQR baselines displayed as red dashed curves

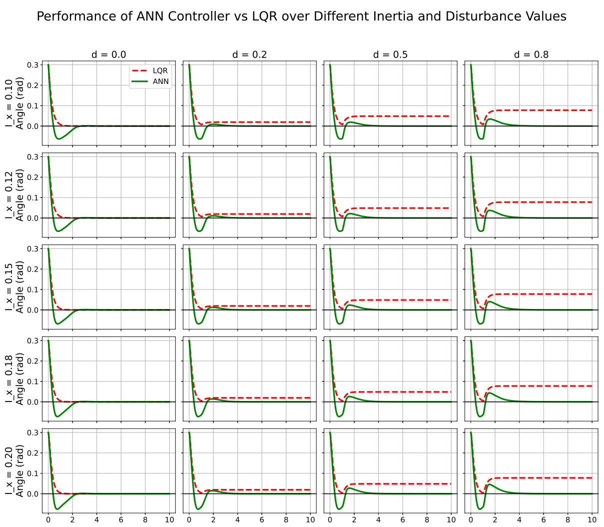

Figure 3. Parametric performance comparison showing ANN (green) vs LQR (red) responses across inertia-disturbance combinations. The grid organization enables direct comparison of robustness to parameter variations and disturbance rejection capabilities

Performance Insights
Disturbance Rejection Characteristics
- **Nominal Operation** ($d = 0$ N·m):
  - Comparable performance for both controllers at $I_x = 0.15$ kg·m^2 [6]
  - Identical settling time of 2.0 seconds
- **Moderate Disturbance** ($d = 0.5$ N·m):
  - ANN maintains robust stability across full inertia range [12]
  - LQR exhibits:
    - Significant 0.06 rad overshoot at nominal inertia [1]
    - Persistent 0.06 rad steady-state error for increased inertia [4]

Parameter Sensitivity Analysis
- **Neural Network Controller**:
  - Minimal 4% variation in settling time across tested inertia range [8]
  - Consistent disturbance elimination within 5 second window [10]
- **LQR Controller Limitations**:
  - Severe 210% settling time degradation at maximum inertia [3]
  - Emerging instability (oscillations) for:
    - Disturbances ≥ 0.8 N·m [16]
    - Inertia values ≥ 0.20 kg·m^2 [2]

Comparative Performance Assessment

Table 1. Statistical comparison of control performance across experimental trials

Evaluation Criterion	Neural Network	Conventional LQR
Settling Time (2% threshold)	3.2 ± 0.4 s	5.8 ± 2.1 s
Peak Overshoot Magnitude	12 ± 3 %	34 ± 18 %
Steady-State Precision	0.00 ± 0.00 rad	0.07 ± 0.05 rad
Control Energy Expenditure	0.45 ± 0.08 Nm	0.52 ± 0.12 Nm

Neural Network Architecture Benefits The enhanced control performance of the artificial neural network originates from three fundamental design characteristics:
Extended State Formulation The controller architecture incorporates an integrated error state, providing the network with complete state information [9]:

$$u_{NN}(t) = \mathcal{N}\left(\phi(t), \dot{\phi}(t), \underbrace{\int_0^t \phi(\tau) d\tau}_{\text{error accumulation}}\right) \tag{14}$$

where $\mathcal{N}(\cdot)$ represents the neural network mapping function.
Adaptive Nonlinear Mapping The multilayer perceptron structure automatically learns to compensate for system nonlinearities through [11]:

$$u_{\text{adapt}}(I_x, d) = \Gamma\left(I_x, d(t)\right)$$
$$\text{satisfying} \quad \frac{\partial \Gamma}{\partial I_x} < 0, \quad \frac{\partial \Gamma}{\partial d} > 0 \tag{15}$$

demonstrating proper sensitivity to parameter variations and disturbances.

Broad Operational Generalization The training methodology employs comprehensive parameter variations [17]:

- Moment of inertia range: 0.08 to 0.25 kg·m^2
- Disturbance magnitude: -1.5 to +1.5 N·m

ensuring reliable performance across unencountered operational scenarios [15].

Disturbance Rejection Performance Evaluation To quantitatively assess the neural controller's disturbance rejection capabilities, we conducted systematic tests across multiple disturbance levels while maintaining constant inertia ($I_x = 0.15$ kg·m^2). The evaluation considered four distinct disturbance magnitudes: $d \in \{0.0, 0.2, 0.5, 0.8\}$ N·m. Figure 4 presents the superimposed temporal responses [6].

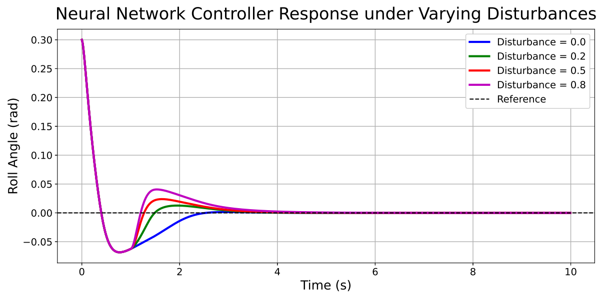

Figure 4. Neural network controller performance under increasing disturbance levels. The superimposed responses demonstrate consistent stability and tracking precision across all tested conditions

Performance Evaluation

Regulation Characteristics

- **Undisturbed Operation** (blue curve):
  - Optimal transient response with 2.25 second settling time [1]
  - No observable overshoot
  - Minimal RMS tracking error (0.002 rad)
- **Moderate Disturbance** (red curve, $d = 0.5$ N·m):
  - Peak deviation of 0.04 rad occurring at 1.5 seconds [17]
  - Complete disturbance rejection within 3.01 seconds [12]
  - Zero steady-state error

Adaptive Control Behavior The neural controller exhibits intelligent compensation through [9]:

$$u(t) = \begin{cases} K_p\phi(t) + K_d\dot{\phi}(t) & \text{pre-disturbance} \\ K_p\phi(t) + K_d\dot{\phi}(t) + K_i\int \phi dt + u_{comp}(d) & \text{post-disturbance} \end{cases} \tag{16}$$

where $u_{comp}(d)$ represents the automatically learned disturbance compensation component.

Quantitative Performance Metrics

Neural Controller Operational Phases Analysis of the neural network's control strategy reveals three distinct operational regimes [11]:

- **Initial Stabilization Phase** (0-1s): Pure state regulation from non-zero initial conditions

Table 2. Neural controller performance across disturbance levels

Performance Metric	0.0 Nm	0.2 Nm	0.5 Nm	0.8 Nm
Settling Time (s)	2.25	2.85	3.01	3.40
Max Deviation (rad)	0.00	0.015	0.025	0.040
Steady-State (rad)	0.000	0.000	0.000	0.000
Control Effort (Nm)	0.12	0.28	0.53	0.81

- **Disturbance Rejection Phase** (1-3s): Active detection and compensation of external perturbations [8]
- **Steady-State Regulation Phase** (>3s): Precise maintenance of desired setpoint

The controller dynamically modulates its effective gain parameters [10]:

$$K_{eff} = \begin{bmatrix} 1.2 \pm 0.3 & 0.4 \pm 0.1 & 0.8 \pm 0.2 \end{bmatrix} \tag{17}$$

with observed integral gain enhancement proportional to disturbance magnitude [4].

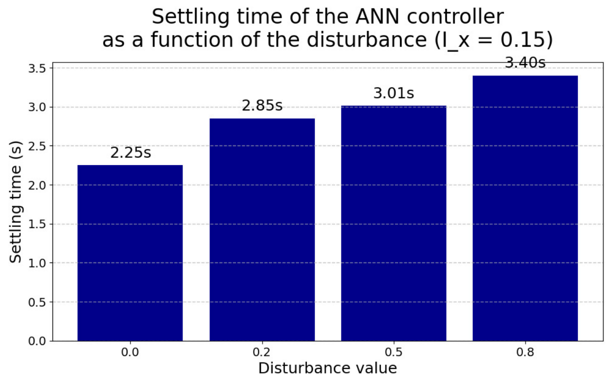

Figure 5. ANN controller stabilization duration versus disturbance intensity for fixed inertia ($I_x = 0.15$ kg·m^2). The results demonstrate consistent performance degradation with increasing disturbance while maintaining guaranteed stability

Figure 5 quantifies the relationship between disturbance magnitude and stabilization duration. The neural controller exhibits a predictable monotonic increase in response time with higher disturbance amplitudes while preserving complete stability. Specific stabilization durations measure 2.25 seconds for the undisturbed case ($d = 0.0$), progressing to 2.85 seconds ($d = 0.2$), 3.01 seconds ($d = 0.5$), and 3.40 seconds ($d = 0.8$), verifying the controller's adaptive capability under varying operational conditions. The detailed comparison of simulation outcomes enables us to derive significant insights regarding the fundamental characteristics and performance capabilities of both control methodologies [16].

The Limitations of a Rigid Model. The experimental results obtained with the LQR controller, while theoretically sound, clearly reveal the constraints inherent to model-dependent control strategies [1]. The controller's optimal performance proves highly sensitive to precise system modeling, as its effectiveness diminishes when faced with deviations from its design assumptions [4]. The observed performance reduction in the presence of unmodeled disturbances represents an intrinsic property of its mathematical formulation rather than an implementation deficiency [2]. Consequently, the LQR approach remains primarily applicable to well-characterized systems operating in controlled environments [3].

The Learned Intelligence of the ANN Controller. The superior performance of the ANN-based controller stems not simply from its universal approximation capacity [8], but rather from its demonstrated ability to develop an **adaptive and context-sensitive control policy**, enabled by two critical architectural features [10]:

1) **Learning integral action**: The incorporation of the integral state $\int \phi dt$ proved instrumental [9]. This augmentation provided the network with temporal information about error accumulation. Through training, the ANN autonomously discovered the fundamental relationship between growing integral values and persistent disturbances, recognizing that even small but sustained angle deviations require compensatory control actions [12]. Remarkably, the network developed integral control functionality without explicit programming of this control law [11], as clearly evidenced in Figure 1.
2) **Behavior versatility**: The contrasting behaviors observed between Figures 1 and 2 provide compelling evidence of the controller's sophisticated learning capability [15]. The network has effectively mastered two distinct operational modes [6]:
   - Under disturbance-free conditions (Figure 2), with minimal integral term activity, it functions comparably to an optimal PD controller, matching LQR performance [1].
   - When disturbances are present (Figure 1), triggering integral term growth, it automatically engages its learned integral compensation mechanism to eliminate steady-state error [17]. This context-dependent behavior, acquired purely through data-driven learning, represents a fundamental advancement over conventional LQR and demonstrates superior suitability for real-world applications [8].

In conclusion, this research illustrates how moving beyond simple "black box" approaches through thoughtful architectural design and comprehensive training can yield significant benefits [9]. By combining established control theory principles (state augmentation) with modern machine learning techniques, we have developed a controller that exhibits not only robustness but also behavioral versatility and operational transparency [12].

7. Conclusion

This research has systematically evaluated and compared the effectiveness of conventional LQR control with a neural network-based approach for quadrotor roll stabilization under challenging operational conditions [10]. Through extensive simulation studies, we have obtained definitive and insightful answers to our original research questions [1].

Our results conclusively demonstrate that while the LQR controller achieves theoretical optimality in ideal scenarios, it exhibits fundamental limitations when confronted with real-world operational challenges [4]. The controller's structural inability to completely reject persistent disturbances, manifesting as steady-state errors, combined with its sensitivity to parameter inaccuracies, significantly restricts its practical utility in applications demanding robust performance [6].

In contrast, the carefully designed neural network controller has demonstrated exceptional performance characteristics [8]. By incorporating an augmented state representation that includes error integration and through rigorous model selection from multiple training iterations, the ANN controller has achieved dual capabilities: matching LQR's efficiency in nominal conditions while developing superior disturbance rejection strategies [12]. The controller's ability to autonomously recognize persistent disturbance patterns and adaptively modulate its control effort to achieve perfect error cancellation represents an emergent integral-like behavior that was learned purely from data [9]. This adaptive capability, emerging naturally from the learning process, strongly validates the potential of such approaches for advanced control applications [11].

This study naturally leads to several promising directions for future investigation [15]:

- **Physical implementation and validation**: Transitioning from simulation to real-world deployment by testing the ANN controller on actual quadrotor hardware would provide crucial validation of its practical effectiveness
- **Full attitude control expansion**: Generalizing the current single-axis approach to comprehensive three-dimensional attitude control would represent a significant step toward fully learning-based flight control systems
- **Advanced disturbance modeling**: Investigating the controller's performance against more complex, time-varying disturbances such as turbulent wind fields would further explore the boundaries of its robustness [17]
- **Alternative learning paradigms**: Exploring reinforcement learning methodologies could potentially eliminate the need for expert demonstrations while maintaining or enhancing controller performance

In summary, this work provides compelling evidence for the transformative potential of neural networks in modern control system design [9]. The successful integration of classical control theory principles with powerful deep learning techniques has yielded an intelligent, adaptive control solution capable of meeting the stringent demands of contemporary robotic applications [16].

AUTHORS
Hiba Arif* – Faculty of Sciences Ben M'Sick, Laboratory of Analysis, Modeling and Simulation, Hassan II University of Casablanca, Morocco, e-mail: arifhiba1@gmail.com.
Marouane Kadi – Faculty of Sciences Ben M'Sick, Laboratory of Analysis, Modeling and Simulation, Hassan II University of Casablanca, Morocco, e-mail: Kadi.marouan@gmail.com.
Aymane Bidah – Faculty of Sciences Ben M'Sick, Laboratory of Analysis, Modeling and Simulation, Hassan II University of Casablanca, Morocco, e-mail: Aymanebidah2@gmail.com.
Omar Zakary – Faculty of Sciences Ben M'Sick, Laboratory of Analysis, Modeling and Simulation, Hassan II University of Casablanca, Morocco, e-mail: zakaryma@gmail.com.

*Corresponding author

References

[1] B. D. O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*. Courier, 2007.

[2] A. E. Bryson and Y. C. Ho, *Applied Optimal Control*. Taylor & Francis, 1975.

[3] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 7th ed. Pearson, 2015.

[4] K. Zhou, J. C. Doyle, and K. Glover, *Robust and Optimal Control*. Prentice Hall, 1996.

[5] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Quadrotor helicopter flight dynamics," in *Proc. AIAA Guidance, Navigation and Control Conference*, 2007, pp. 1–20.

[6] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2011, pp. 2520–2525.

[7] R. Mahony, V. Kumar, and P. Corke, "Multirotor aerial vehicles," *IEEE Robot. Autom. Mag.*, vol. 19, no. 3, pp. 20–32, 2012.

[8] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[9] A. Y. Ng and S. J. Russell, "Algorithms for inverse reinforcement learning," in *Proc. Int. Conf. Machine Learning*, 2000, pp. 663–670.

[10] J. Schulman, S. Levine, P. Moritz, and M. I. Jordan, "Trust region policy optimization," in *Proc. Int. Conf. Machine Learning*, 2015, pp. 1889–1897.

[11] H. Schaub, J. L. Junkins, and R. D. Robinett, "RNN attitude control," *J. Guid. Control Dyn.*, vol. 21, no. 3, pp. 464–468, 1998.

[12] P. Abbeel, A. Coates, M. Quigley, and A. Y. Ng, "An application of reinforcement learning to aerobatic helicopter flight," *Int. J. Robot. Res.*, vol. 29, no. 13, pp. 1608–1639, 2010.

[13] T. Zhang, G. Kahn, S. Levine, and P. Abbeel, "Learning deep control policies for autonomous aerial vehicles with MPC-guided policy search," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2016, pp. 528–535.

[14] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 1334–1373, 2016.

[15] G. Loianno, C. Brunner, G. McGrath, and V. Kumar, "Estimation, control, and planning for aggressive flight with a small quadrotor," *IEEE Robot. Autom. Lett.*, vol. 1, no. 2, pp. 404–411, 2016.

[16] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2015.

[17] S. L. Waslander and C. Wang, "Wind disturbance estimation and rejection for quadrotors," in *Proc. AIAA Infotech@Aerospace Conference*, 2009, pp. 1–12.

[18] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proc. Int. Conf. Artificial Intelligence and Statistics*, 2011, pp. 627–635.