Real-Time Adaptive Attitude Control of Lynx Helicopter with Hybrid LQR-Neural Network Architecture

Ahmet Kara[1] and Mehmet Önder Efe[2]

***Abstract*— This paper presents a comprehensive implementation of a multilayer Neural Network (NN) controller for trajectory tracking in a 6 Degrees of Freedom (DoF) Lynx Helicopter Model, combining a passivity-based approach with real-time weight tuning and an inner-loop Linear Quadratic Regulator (LQR) for stability augmentation. The NN controller replaces traditional integral control, minimizing steady-state errors, enhancing reference tracking accuracy, and regulating off-axis states with a robust architecture that integrates filtered error signals, passivity-based terms, and system states. Using a forward pass for control generation and a backward pass for real-time weight adjustment via gradient descent, the controller ensures stability and convergence through adaptive gains and tuned learning rates. Time-domain simulations demonstrate precise tracking performance across diverse reference commands with minimal overshoot, smooth transients, and steady neural network weight convergence, validating the hybrid controller's reliability under varying operating conditions.**

I. INTRODUCTION

Helicopter control is challenging due to nonlinear dynamics, strong axis coupling, and sensitivity to disturbances. Traditional controllers, such as PID and state-feedback methods, struggle with uncertainties, cross-axis couplings, and time-varying dynamics, particularly for precise attitude tracking [1], [2].

Neural Network (NN) controllers offer a promising alternative, capable of approximating complex nonlinearities and adapting in real-time to varying conditions [3], [4]. However, many rely on offline training, limiting adaptability during dynamic flight scenarios [5], [6].

This study proposes a hybrid control architecture integrating a Linear Quadratic Regulator (LQR) inner-loop with an outer-loop NN controller. The LQR ensures rapid stabilization and disturbance rejection, while the NN dynamically compensates for steady-state errors, tracking deviations, and cross-axis regulation using a passivity-based approach with real-time weight updates, ensuring continuous adaptation and improved tracking performance.

The paper is structured as follows: Section 2 outlines the helicopter dynamics model. Section 3 explains the LQR-based inner-loop and the outer-loop NN controller with passivity-based adaptation. Section 4 presents simulation results on tracking performance under various references. Section 5 concludes the study and suggests future research directions.

[1]Ahmet Kara is MSc. Student in Faculty of Computer Engineering, Hacettepe University, Ankara, Turkey ahmetkara@hacettepe.edu.tr

[2]Mehmet Önder Efe is Prof. Dr. in the Department of Computer Engineering, Hacettepe University, Ankara, Turkey onderefe@hacettepe.edu.tr

II. 6DoF Lynx Helicopter Model

The Lynx helicopter is a twin-engine rotorcraft, is renowned for its agility and high-speed performance, making it an ideal platform for advanced control system experiments. This section presents the mathematical representation, state and control variable definitions, trim analysis, and control input limits for the helicopter model at a trimmed forward flight speed of 140 knots.The helicopter dynamics are captured using a state-space representation, defined as:

$$\dot{x} = Ax + Bu, \quad y = Cx + Du \tag{1}$$

Where, $x \in \mathbb{R}^8$ represents the state vector, $u \in \mathbb{R}^4$ is the control input vector, and $y \in \mathbb{R}^8$ is the system output. The matrix $A \in \mathbb{R}^{8\times 8}$ defines the system dynamics, while $B \in \mathbb{R}^{8\times 4}$ characterizes how control inputs influence the system states. The output matrix $C \in \mathbb{R}^{8\times 8}$ maps the state vector to the output, and $D \in \mathbb{R}^{8\times 4}$ represents any direct input-output feedthrough. In this study, D is set to zero, reflecting no direct coupling between inputs and outputs. At a trimmed forward flight speed of 140 knots, the system matrices are defined as: [7]

$$A = \begin{bmatrix} -0.0525 & 0.0370 & 5.2710 & -9.7910 & 0 & -0.0838 & 0 & 0 \\ 0.0286 & -0.9392 & 71.6880 & 0.6160 & -0.0272 & -0.8596 & 0.6083 & 0 \\ 0.0328 & 0.0426 & -2.7297 & 0 & 0.0047 & 0.4327 & 0 & 0 \\ 0 & 0 & 0.9981 & 0 & 0 & 0 & 0 & 0.0621 \\ 0.0052 & 0.0127 & -0.1297 & -0.0383 & -0.2446 & -5.2343 & 9.7720 & -71.3836 \\ -0.0338 & 0.2396 & -1.7657 & 0 & -0.2050 & -10.1775 & 0 & -0.4394 \\ 0 & 0 & 0.0039 & 0 & 0 & 1 & 0 & -0.0629 \\ -0.0269 & 0.0406 & 0.1928 & 0 & 0.0851 & -1.7264 & 0 & -1.5264 \end{bmatrix} \tag{2}$$

$$B = \begin{bmatrix} 3.6956 & -6.8427 & 2.2599 & 0 \\ -143.5034 & -58.7853 & 0.0001 & 0 \\ 24.4192 & 32.5904 & -6.1083 & 0 \\ 0 & 0 & 0 & 0 \\ 1.5764 & -1.1831 & -9.8730 & 5.1875 \\ 38.1461 & -15.8917 & -153.8247 & -1.0381 \\ 0 & 0 & 0 & 0 \\ 21.4497 & -2.7783 & -26.1582 & -13.9821 \end{bmatrix} \tag{3}$$

A. *Eigenvalue Analysis of the Open-Loop System*

The stability of the linearized helicopter dynamics can be analyzed by evaluating the eigenvalues of the system matrix (A). These eigenvalues provide insight into the open-loop stability characteristics of the Lynx helicopter at a trimmed forward speed of 140 knots.

The corresponding eigenvalues, denoted by λ, represent the dynamic modes of the system and are listed below:

$$\begin{aligned}
&\lambda_1 = 0.3124 + 0.3172i, \quad \lambda_2 = 0.3124 - 0.3172i,\\
&\lambda_3 = -0.0430, \quad \lambda_4 = -0.3674,\\
&\lambda_5 = -0.6537 + 2.8412i, \quad \lambda_6 = -0.6537 - 2.8412i,\\
&\lambda_7 = -4.1854, \quad \lambda_8 = -10.3914.
\end{aligned} \tag{4}$$

The eigenvalues reveal key stability properties of the system. The positive real part of the first two eigenvalues ($0.3124 \pm 0.3172i$) indicates an unstable oscillatory mode (phugoid mode), which exhibits divergent behavior without control intervention. In contrast, the remaining eigenvalues have negative real parts, representing stable modes.

The presence of unstable eigenvalues highlights the need for control strategy to stabilize the Lynx helicopter's dynamics during forward flight at 140 knots. Specifically, the Stability Augmentation System (SAS) using the Linear Quadratic Regulator (LQR) and the Neural Network (NN) controller play crucial roles in suppressing these instabilities and ensuring reliable trajectory tracking.

B. State and Control Variables

The state vector x represents the key dynamic states of the Lynx helicopter:

$$x = \begin{bmatrix} u & v & w & p & q & r & \phi & \theta \end{bmatrix}^T \tag{5}$$

The parameters are defined as follows: u, v, w represent the forward, lateral, and vertical velocities (m/s), while p, q, r denote the roll, pitch, and yaw angular rates (rad/s). Additionally, θ and ϕ correspond to the pitch and roll angles (rad), respectively.

The control input vector u consists of control system commands:

$$u = \begin{bmatrix} \delta_{\text{lon}} & \delta_{\text{lat}} & \delta_{\text{coll}} & \delta_{\text{ped}} \end{bmatrix}^T \tag{6}$$

The helicopter's control system operates within predefined control limits to ensure structural integrity and predictable behavior. The collective control ranges from $0°$ to $15°$, while the longitudinal cyclic is limited to $\pm 20°$, the lateral cyclic to $\pm 15°$, and the pedals to $\pm 40°$. These limits ensure safe operation under dynamic conditions and prevent overstressing critical components.

III. Hybrid Controller Architecture

The block diagram in Fig. 1 illustrates the proposed *Hybrid Control Architecture*, integrating a *Neural Network (NN) Controller* and a *State Feedback Controller (SFC)* for the Lynx Helicopter System. This architecture leverages the adaptability of the NN Controller and the stability of the SFC to achieve precise attitude control and system stability.

The reference signal (r) represents the desired trajectory and is compared with the system output (y) to generate an error signal (e). This error signal serves as an input to the NN Controller, which dynamically adjusts its weights via real-time backpropagation, producing an adaptive control signal (u_{NN}). The NN Controller primarily minimizes tracking errors while adapting to modeling uncertainties.

Simultaneously, the State Feedback Controller (SFC) processes the system states (x) to compute a stabilizing control signal (u_{SF}). The SFC ensures inner-loop stability, effectively handling fast dynamics and rejecting external disturbances.

The outputs of both controllers are combined and passed through a saturation mechanism, which enforces actuator constraints and ensures that control commands remain within safe physical limits. The resulting control signal is applied to the helicopter dynamics, generating new system states (x) and measurable outputs (y), which are fed back to both controllers, forming a closed-loop architecture. This hybrid control architecture effectively integrates outer-loop adaptation, where the NN controller minimizes tracking errors and adapts to system uncertainties in real time, and inner-loop stability, where the SFC ensures stabilization and disturbance rejection. By combining these two controllers, the architecture effectively addresses dynamic uncertainties, mitigates disturbances, and ensures accurate and stable attitude tracking. This hybrid control approach is comparable to recent real-time NN-based helicopter control architectures proposed in the literature [10].

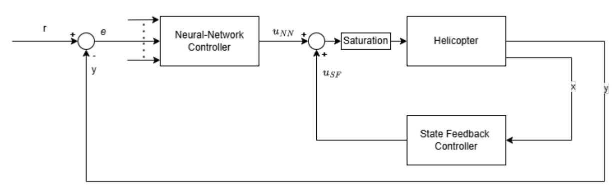

Fig. 1. General Plant-Controller Architecture

A. Stability Augmentation System (SAS)

The primary objective of the Stability Augmentation System (SAS) is to enhance the inherent stability of the Lynx helicopter, particularly during trimmed flight at high speeds. This is achieved through a State Feedback Controller (SFC), designed using the Linear Quadratic Regulator (LQR) methodology. The LQR framework ensures optimal control performance by minimizing a quadratic cost function, balancing state deviations with control effort to guarantee stability across varying operational conditions.

The state feedback control law is formulated using LQR design principles [8].

$$u_{SF} = -Kx \tag{7}$$

The optimal gain matrix K is determined by minimizing the following quadratic cost function:

$$J = \int_0^\infty \left(x^T Q x + u^T R u \right) dt \tag{8}$$

The parameters are defined as follows: J represents the performance index, balancing state deviation and control effort, while Q is the state weighting matrix, penalizing deviations in critical states. Additionally, R is the control weighting matrix, which penalizes excessive control inputs.

The weighting matrices Q and R are carefully tuned based on performance criteria, emphasizing critical states such as pitch and roll rates while preventing actuator saturation.

The state feedback gain matrix is derived as:

$$K = R^{-1} B^T P \tag{9}$$

This gain matrix optimally balances state regulation and control effort, directly contributing to the system's robustness and stability.

Key design aspects of the SAS include state prioritization, which emphasizes critical states such as pitch and roll rates through the use of the state weighting matrix Q, and control effort limitation, which prevents actuator saturation by carefully tuning the control weighting matrix R. The LQR-based SAS serves as the foundation for the outer-loop Neural Network (NN) Controller by stabilizing fast helicopter dynamics to ensure predictable responses and providing a stable baseline for the NN controller to focus on error minimization.

B. Neural-Network based Attitude Controller

The general Neural Network (NN) equation can be expressed in matrix form as:

$$y = W^T \sigma(V^T x) \tag{10}$$

The parameters are defined as follows: W represents the output layer weight matrix, connecting the hidden layer to the output layer, while V denotes the input layer weight matrix, connecting the input layer to the hidden layer. The activation function, $\sigma(\cdot)$, introduces nonlinearity into the network and can take forms such as sigmoid or hyperbolic tangent functions.

The NN controller approximates the helicopter's nonlinear dynamics using the functional estimate:

$$\hat{f}(x) = \hat{W}^T \sigma(\hat{V}^T x) \tag{11}$$

The parameters are defined as follows: $\hat{W}$ and $\hat{V}$ represent the current estimates of the ideal neural network (NN) weights, which are adaptively tuned in real-time. The input vector x encapsulates critical dynamic and control terms.

The input vector x incorporates essential dynamic states, control signals, and error terms:

$$x = [e, \dot{e}, r, \dot{r}, x, \dot{x}, u_{NN}^{previous}, \dot{u_{NN}}]^T$$

The parameters are defined as follows: e represents the tracking error, and $\dot{e}$ denotes its derivative. The tracking reference signal is r, with $\dot{r}$ as its derivative. The system states, including velocities, angular rates, and Euler angles, are represented by x, while $\dot{x}$ denotes their derivatives. Additionally, $u_{NN}^{previous}$ refers to the previous neural network (NN) control input, and $\dot{u_{NN}}$ is its derivative.

These dynamic terms collectively provide the NN with a online dataset, enabling it to approximate the nonlinear system function and adapt the weight matrices ($\hat{W}$ and $\hat{V}$) in real-time.

The NN-generated control input is mathematically defined as:

$$u_{NN} = \hat{f}(x)$$

This control law empowers the NN controller to dynamically adjust its output based on real-time system data, ensuring robust performance across varying operational conditions. A detailed representation of the Neural Network structure is shown in Fig. 2.

1) Neural-Network based Attitude Controller Metrics:: To further enhance stability and performance, the control design incorporates key metrics derived from error dynamics, filtered signals, and passivity-based terms. These metrics provide a structured representation of control dynamics, enabling the Neural Network (NN) controller to optimize its performance in real-time.

1. Error Dynamics with Filtered Terms

The error dynamics incorporate filtering gains (Λ_θ) specifically for the pitch (θ) axis:

$$r_\theta = \dot{r}_\theta - \dot{x}_\theta + \Lambda_\theta e_\theta \tag{12}$$

The parameters are defined as follows: r_θ represents the filtered error dynamics for the pitch angle, while $\dot{r}_\theta$ denotes the derivative of the pitch angle reference signals. The pitch state derivative is given by $\dot{x}_\theta$, and the filtering gain for pitch error dynamics is represented by Λ_θ. Additionally, e_θ corresponds to the pitch angle tracking error.

2. Passivity-Based Terms

Passivity terms (ζ_1 and ζ_2) ensure system stability and provide robustness against uncertainties:

$$\zeta_1 = \ddot{r}_\theta + \Lambda \dot{e}_\theta \qquad \zeta_2 = \dot{r}_\theta + \Lambda e_\theta \tag{13}$$

The parameters are defined as follows: $\ddot{r}_\theta$ represents the second derivative of the reference signal, while $\dot{e}_\theta$ denotes the derivative of the tracking error. The passivity gain is given by Λ, and the passivity terms contributing to control stability are represented by ζ_1 and ζ_2.

3. Control Metrics Derived from Errors

The error vector and its derivative are weighted to form a sliding mode metric:

$$M_1 = w_e e + w_{\dot{e}} \dot{e} \tag{14}$$

The parameters are defined as follows: w_e represents the weight for the error term, while $w_{\dot{e}}$ denotes the weight for the error derivative. The error vector is given by e, and its derivative is represented by $\dot{e}$.

4. Control Metrics Derived from Inputs

Similarly, control inputs and their derivatives are weighted to define a second control metric:

$$M_2 = w_u u + w_{\dot{u}} \dot{u} \tag{15}$$

The parameters are defined as follows: w_u represents the weight for the control input, while $w_{\dot{u}}$ denotes the weight for the control input derivative. The control input vector is given by u, and its derivative is represented by $\dot{u}$.

5. Neural Network Input Vector

The Neural Network controller combines these metrics into a comprehensive input vector:

$$X = \begin{bmatrix} r_\theta & \zeta_1 & \zeta_2 & M_1 & M_2 & x & \dot{x} & \text{sgn}(x) \end{bmatrix}^T \quad (16)$$

These metrics collectively integrate filtered error signals, passivity-based stability terms, and weighted control metrics, forming a robust input representation for the NN controller. This real-time updated dataset allows the NN controller to adapt its weights, handle uncertainties, and ensure accurate control across operating conditions..

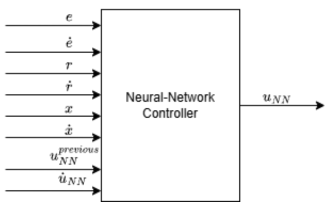

Fig. 2. Neural-Network Controller

2) Weight Update Rules:

The neural network (NN) controller employs an online weight tuning mechanism to adapt its weights in real time, minimizing tracking error and ensuring system stability. The weight update rules, based on the standard backpropagation algorithm, are formulated following recent Lyapunov-based neural adaptive control approaches [9].

$$\begin{aligned} \hat{\sigma} &= \sigma(\hat{V}^T x), \quad \hat{\sigma}' = \hat{\sigma} \odot (1 - \hat{\sigma}) \\ \dot{\hat{W}} &= F \hat{\sigma} r^T, \quad \dot{\hat{V}} = G x (\hat{\sigma}'^T \hat{W} r)^T \end{aligned} \quad (17)$$

The adaptive estimates of the ideal neural network (NN) weight matrices are represented by $\hat{W}$ and $\hat{V}$, while the positive definite design matrices F and G ensure appropriate weight adaptation gains. The activation output from the hidden layer is denoted as $\hat{\sigma}$, with $\hat{\sigma}'$ representing the derivative of the activation function. Additionally, r serves as the filtered error signal. The weight update rules are derived based on a Lyapunov stability criterion, ensuring that the derivative of the Lyapunov function $V(t)$ remains non-positive. This guarantees that both the tracking error r and the weight updates $\hat{W}$ and $\hat{V}$ converge to stable bounds over time.

The Lyapunov-based adaptation law plays a crucial role in maintaining the stability of the NN controller under varying operating conditions. It ensures that the tracking error r converges to zero while keeping the NN weights $\hat{W}$ and $\hat{V}$ bounded, even in the presence of unknown disturbances and unmodeled dynamics. This adaptation mechanism ensures the NN controller robustness and reliable performance in uncertain environments.

IV. Simulation Results

The hybrid controller's performance was evaluated in MATLAB/Simulink under varying reference signals to assess its attitude tracking capabilities. The controller's performance is analyzed under two reference scenarios: a step reference and a doublet reference. To validate weight convergence and ensure bounded adaptation, the following summed square metrics are employed:

$$W_{sq} = \sum_{t=0}^{t_{final}} WW^T, \qquad V_{sq} = \sum_{t=0}^{t_{final}} VV^T \quad (18)$$

The weighting matrices for the State Feedback Controller are chosen as:

$$Q_{\text{aug}} = \begin{bmatrix} 0.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.01 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 55 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 255500 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 50000 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix} \quad (19)$$

$$R = \begin{bmatrix} 25000 & 0 & 0 & 0 \\ 0 & 150 & 0 & 0 \\ 0 & 0 & 1000 & 0 \\ 0 & 0 & 0 & 800 \end{bmatrix} \quad (20)$$

To facilitate pitch angle tracking, the state weighting matrix Q is augmented to form Q_{aug} of size 9×9. Using these matrices, the Stability Augmentation System (SAS) gains were computed, ensuring optimized stability and tracking performance across the evaluated scenarios.

The neural network consists of a hidden layer with 10 sigmoid neurons and a linear output layer. The input and output weight matrices are $V \in \mathbb{R}^{31 \times 10}$ and $W \in \mathbb{R}^{10 \times 4}$, respectively. Weights are initialized uniformly in the range ± 0.1 to prevent large gradients during early learning. Learning rates are set to $\eta_{step} = 4 \times 10^{-4}$ and $\eta_{doublet} = 5 \times 10^{-5}$. Adaptation gains use $F \in \mathbb{R}^{10 \times 10}$ and $G \in \mathbb{R}^{31 \times 31}$, both set to 5. The passivity and filtering gains are $\Lambda = \Lambda_\theta = 45$. Error weights are $w_e = 1.5$, $w_{\dot{e}} = 10^5$, and control weights are $w_u = w_{\dot{u}} = 0.0525$.

A. Response of Hybrid Controller under Step Reference

This section presents the results of the hybrid control architecture under a step reference command for pitch angle tracking. The performance is evaluated across control commands, system response, state variables, and neural network weight adaptation. Fig. 3 illustrates the pitch angle response under a step reference input. The desired reference is tracked with minimal overshoot and rapid convergence to steady-state. The initial transient response is well-controlled, indicating the robustness of the inner-loop State Feedback Controller and the adaptive compensation provided by the NN Controller. The tracking error is minimized throughout the simulation duration, demonstrating accurate reference tracking.

The evolution of the system states during the step reference tracking is presented in Fig. 4. The state variables exhibit smooth transient behavior, with no significant oscillations or instabilities. The filtered dynamics and state feedback contributions ensure proper damping and system stability, even under abrupt changes in the reference signal.

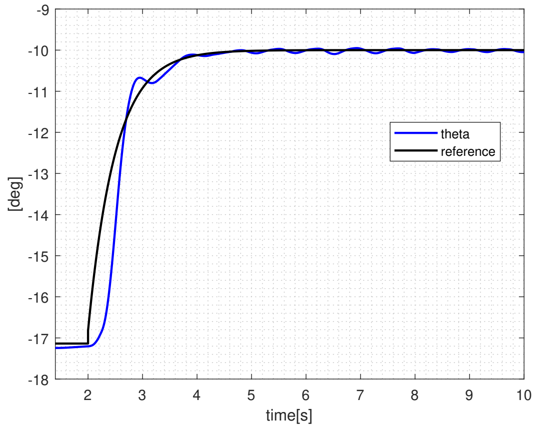

Fig. 3. Tracking Performance of Controller for Step Type Ref

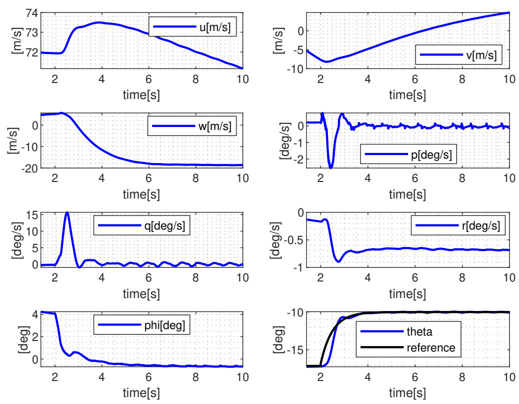

Fig. 4. Flight States during Simulation

The control commands generated by the hybrid controller are shown in Fig. 5. The State Feedback Controller (SFC) and Neural Network (NN) Controller generate control signals to stabilize and track the desired pitch angle reference. The control commands remain within predefined actuator limits, ensuring safe and stable operation. Smooth transitions and minimal control chattering demonstrate the effectiveness of the controller in avoiding abrupt system responses. The adaptation of the NN weights during the step reference tracking is shown in Fig. 6. The weight values converge over time, indicating stable adaptation and bounded behavior. This convergence ensures that the NN Controller effectively learns and compensates for system nonlinearities and dynamic uncertainties without divergence.

B. Response of Hybrid Controller under Doublet Reference

The pitch response to the doublet reference input is shown in Fig. 7. The system accurately tracks the doublet reference with minimal overshoot and rapid settling time. The transient dynamics remain well-controlled, demonstrating the hybrid controller's ability to manage abrupt changes in the reference signal while maintaining overall system stability.

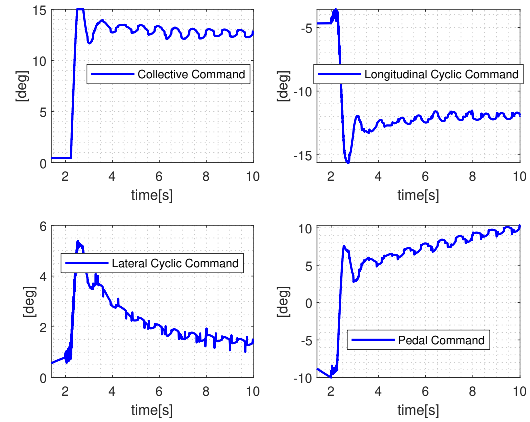

Fig. 5. Applied Control Commands for Step Type Reference

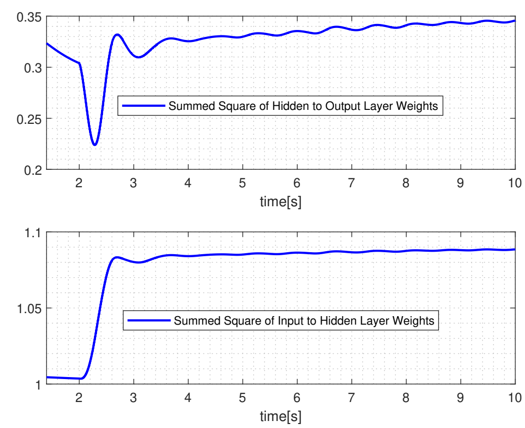

Fig. 6. Neural-Network Weights for Step Type Reference

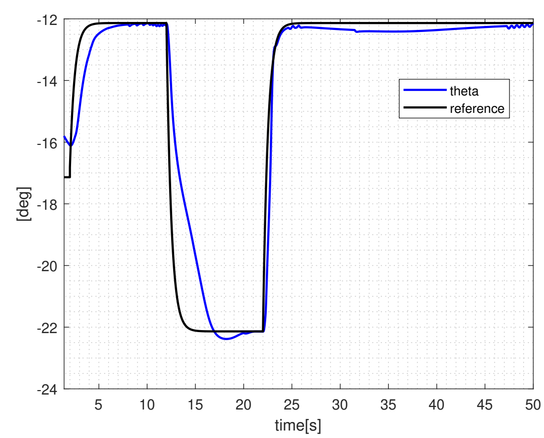

Fig. 7. Tracking Performance of Controller for Doublet Type Ref

The evolution of state variables during the doublet input is depicted in Fig. 8. The states display smooth transitions with no significant oscillations or instability. The filtering mechanisms and adaptive control gains ensure proper damping and dynamic stability throughout the maneuver. Fig. 9

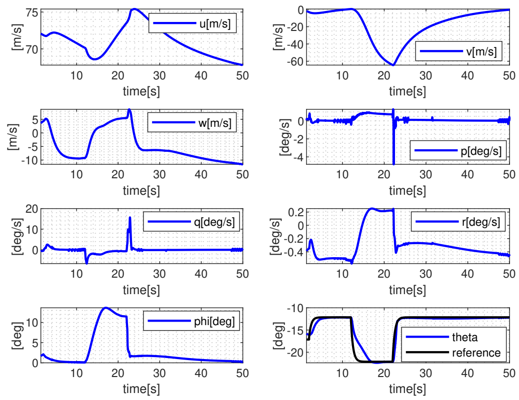

Fig. 8. Flight States during Simulation

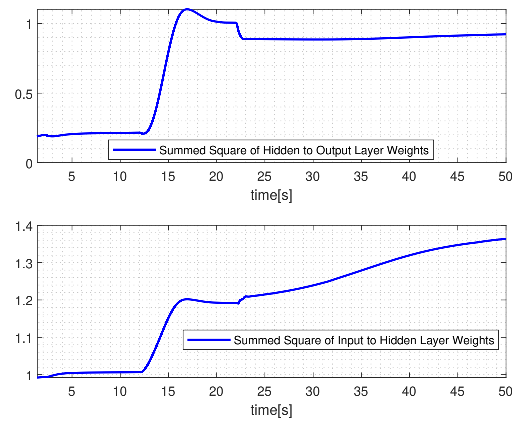

Fig. 10. Neural-Network Weights for Doublet Type Reference

presents the control commands generated by the hybrid controller in response to the doublet reference input. The State Feedback Controller (SFC) and Neural Network (NN) Controller collaboratively produce control signals that remain smooth and within actuator limits. The control commands effectively stabilize the system while adapting to rapid reference changes introduced by the doublet input. Fig. 10

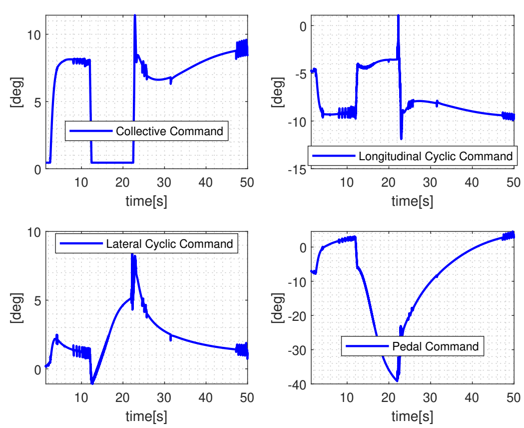

Fig. 9. Applied Control Commands for Doublet Type Reference

illustrates the evolution of the Neural Network weights during the doublet tracking. The weights converge over time without divergence, indicating stable adaptation. This behavior ensures that the NN Controller accurately compensates for system nonlinearities and uncertainties while maintaining control robustness.

V. Conclusion

This study presented a hybrid control architecture combining an inner-loop Linear Quadratic Regulator (LQR) with an outer-loop Neural Network (NN) controller for attitude tracking in a 6-DoF Lynx Helicopter Model. The LQR ensured rapid stabilization and disturbance rejection, while the NN adaptively compensated for modeling uncertainties through real-time weight updates.

Simulation results demonstrated precise attitude tracking with minimal overshoot, smooth transients, and stable weight convergence across step and doublet inputs. The combination of model-based LQR and adaptive NN control proved effective in addressing modeling inaccuracies and ensuring robust performance in dynamic environments. Future work could explore more complex flight scenarios and optimize NN hyperparameters.

References

[1] Frank L. Lewis et al., "Neural network control of robot manipulators and non-linear systems," *IEEE Control Systems*, vol. 14, no. 1, pp. 21–25, 1994.

[2] L. van Schoor Hager and Kenneth R. Uren, "Adaptive neural network control of a helicopter system with optimal observer and actor-critic design," *Neurocomputing*, vol. 302, pp. 75–90, 2018.

[3] Z. Zhao, J. Zhang, Z. Liu, C. Mu, and K. S. Hong, "Adaptive neural network control of an uncertain 2-DOF helicopter with unknown backlash-like hysteresis and output constraints," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[4] J. Li, Y. Jing, D. Wang, and D. Liu, "Neural network-based optimal adaptive output feedback control of a helicopter UAV," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 2, pp. 938–950, 2014.

[5] Z. Qu and J. Xu, "Adaptive neural network control for helicopters in vertical flight," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 3, pp. 491–500, 2008.

[6] J. Li, Y. Jing, D. Wang, and D. Liu, "Neural network-based optimal adaptive output feedback control of a helicopter UAV," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 2, pp. 938–950, 2014.

[7] Gareth D. Padfield, *Helicopter Flight Dynamics: The Theory and Application of Flying Qualities and Simulation Modelling*, 2nd ed., Wiley-Blackwell, 2007.

[8] I. Kamal, S. Azad, and A. Seuret, "LQR and neural network-based robust control for quadrotor UAVs with external disturbances," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 2, pp. 1056–1067, 2023.

[9] X. Gao, M. Fu, and D. W. C. Ho, "Adaptive neural control for uncertain nonlinear systems with time delays using Lyapunov–Krasovskii functionals," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 231–243, 2021.

[10] S. Shah, M. H. Shaheed, and A. H. Nasir, "Neural network adaptive control of a 6-DOF helicopter system: Real-time implementation and performance analysis," *Aerospace Science and Technology*, vol. 120, Art. no. 107217, 2022.