Stabilized Robust Control for Lightweight Autonomous Aircraft Mobility: A Quantum Reinforcement Learning Approach

Gyu Seon Kim†, Jaehyun Chung†, Trung Q. Duong§, Soohyun Park‡, and Joongheon Kim†
†Department of Electrical and Computer Engineering, Korea University, Seoul, Republic of Korea
§Faculty of Engineering and Applied Science, Memorial University, St. John's, NL A1C 5S7, Canada
‡Division of Computer Science, Sookmyung Women's University, Seoul, Republic of Korea
E-mails: kingdom0545@korea.ac.kr, rupang1234@korea.ac.kr, tduong@mun.ca
soohyun.park@sookmyung.ac.kr, joongheon@korea.ac.kr

***Abstract*—The stability of aircraft remains vulnerable to sudden external disturbances and unpredictable vortices. The aircraft's attitude angles undergo rapid changes due to random turbulence. Consequently, to ensure safety, it is essential to control the aircraft's control surfaces, i.e., ailerons, elevators, and rudder angles, to maintain its static stability. Although classical closed-loop control methods have been widely adopted, their limited adaptability to changing dynamics calls for more robust solutions. Reinforcement learning (RL) offers adaptive capabilities but often demands a large number of training parameters and substantial computational resources, which may be impractical for real-time lightweight aircraft applications. To overcome these limitations, this paper introduces a quantum aircraft with the quantum actor-critic networks–based aircraft control (QACN-AC) algorithm. By utilizing quantum neural networks (QNN), QACN-AC significantly reduces the number of parameters required for training, thus mitigating computational overhead while preserving robust control performance. The QACN-AC's effectiveness is validated through realistic simulations leveraging Boeing's B777 specifications. The results highlight QACN-AC's superiority over conventional RL, evidenced by a 1.25× higher control performance and a 760× reduction in the number of required parameters.**

I. Introduction

Since the inception of powered flight, aircraft has undergone constant advancements, resulting in extraordinary advantages for everyday life. The integration of sophisticated avionics, advanced navigation, and control algorithms has dramatically reduced the accident rate [1]. As a result, air travel has become one of the safest modes of transportation. However, challenges remain in sustaining the static stability of an aircraft. These challenges are particularly evident when aircrafts face *unpredictable* external disturbances and vortices. Sudden external disturbances alter the aircraft's attitude angles, and in response, the aircraft adjusts its attitude using its control surfaces. As shown in Fig. 1, the control surfaces of an aircraft, *i.e.*, ailerons,

This research was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2024-RS-2024-00436887) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation); and also by IITP grant funded by MSIT (RS-2024-00439803, SW Star Lab). The work of T. Q. Duong was supported in part by the Canada Excellence Research Chair (CERC) Program CERC-2022-00109. *(Corresponding authors: Soohyun Park, Joongheon Kim)*

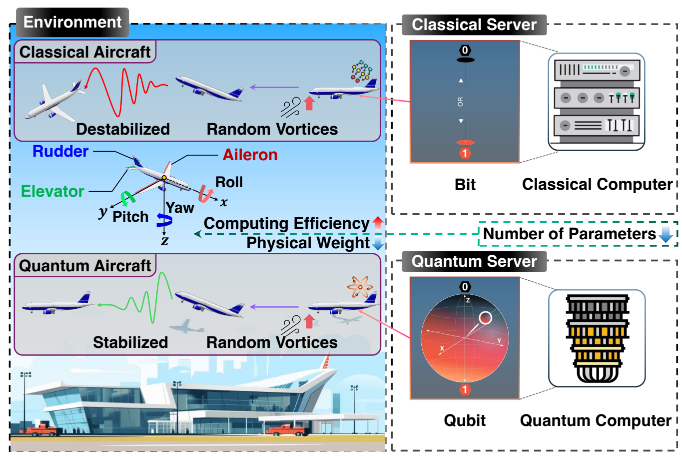


Fig. 1. Proposed aircraft control system for quantum systems.

elevators, and rudder, play a pivotal role in ensuring the stable and predictable attitude angle of the aircraft. Static stability, in particular, is a fundamental requirement: the aircraft must naturally tend to return to its *original condition* when subjected to disturbances. In practice, however, unpredictable factors such as random vortices and turbulence pose significant challenges to maintaining this stability. These vortices can cause sudden changes in the attitude angles of the aircraft, requiring prompt and effective control surface adjustments. Thus, the design of an intelligent and lightweight control system capable of adapting to random vortices is of great importance in aircraft control.

Until now, control approaches have mainly utilized model predictive control (MPC) and proportional integral derivative (PID) controllers [2]. These control techniques employ closed-loop controls that acquire feedback and make adjustments, thereby relying on the dynamic system. Classical control systems, although widely used, exhibit limited adaptability to *dynamic* and *uncertain* environments. Any change in dynamic systems requires a complete controller redesign, a process that can be both time consuming and costly. These constraints underline the demand for a more flexible and robust control paradigm. In recent years, reinforcement learning (RL) has emerged as a powerful tool to design adaptive controllers

ISBN 978-3-903176-73-7 © 2025 IFIP

capable of handling a wide range of uncertainties [3]. RL demonstrates an exceptional adaptability capacity, which requires only hyperparameter adjustments instead of a complete controller overhaul when the system dynamics change [4]. This capability proves especially advantageous for systems governed by complex dynamics that are difficult to precisely model, such as aircraft control problems, where nonlinearity or uncertainty prevails [5]. However, employing RL requires extensive parameter tuning during training, raising concerns about *(a) overfitting* and adding *(b) complexity* to the training procedure [6]. Furthermore, conventional RL algorithms typically require *a large number of training parameters*, which can be *(c) computationally expensive* and pose *(d) weight and power constraints* - critical factors in aircraft design where every kilogram must be justified. In this challenge, *quantum*-based algorithm, particularly *quantum RL (QRL)*, presents valuable solutions [7]. Therefore, this paper proposes a **quantum aircraft** with a **quantum actor-critic networks-based aircraft control (QACN-AC)** to solve these problems. Unlike conventional RL, which uses the classical neural network (NN), QACN-AC uses quantum NN (QNN) to *significantly reduce the number of training parameters* needed for effective policy training [4], [8]. This parameter reduction not only cuts computational overhead but also aligns perfectly with the need for lightweight and resource-limited onboard systems. Therefore, the proposed algorithm enables more *(i) computationally efficient* and *(ii) physically lightweight systems*. This is because less memory use reduces the computing resources that must be mounted on the aircraft. As a result, QACN-AC becomes an attractive solution for real-time control surface actuation in aircraft systems where light weight is crucial. An experimental environment is constructed using the actual aerodynamic specifications of `Boeing's B777-300X` to enhance the practical applicability of the proposed algorithm [9]. In this realistic environment, QACN-AC is evaluated in terms of static stability performance and the number of training parameters.

Contributions. The major contributions of the proposed QACN-AC can be summarized as follows. Firstly, this paper is the first study to apply QRL integrated with QNN in aircraft control, thus addressing the challenges associated with conventional RL. Secondly, this paper utilizes quantum superposition and entanglement to minimize training parameters, thereby enhancing the computational efficiency and reducing the physical weight of quantum aircraft. Thirdly, the proposed algorithm is evaluated in an experimental environment based on real aircraft, which boasts high practical application potential.

II. Preliminaries

Motivation for Quantum-based Approach. Firstly, as shown in Fig. 1, quantum aircrafts employ quantum bits (qubits) in which two states, *that is,*, 0 or 1, can *coexist*. Quantum aircrafts can encode complex state spaces using fewer qubits than the number of bits required in conventional RL, thereby reducing the number of training parameters. Secondly, *quantum parallelism* allows simultaneous evaluation of numerous possibilities, potentially reducing dependence on large parameter spaces. Thirdly, by leveraging *superposition* and *entanglement*, which are characteristics of quantum mechanics, to achieve more compact representations of the underlying state space, QACN-AR can enable faster convergence and more reliable performance under strict resource constraints [10]. This smaller set of parameters ultimately underscores the motivation to adopt QACN-AC over conventional RL for the control of aircraft surfaces. Because it directly addresses the challenges that conventional RL cannot solve, *e.g.,* overfitting, complexity, computational efficiency, and weight problems.

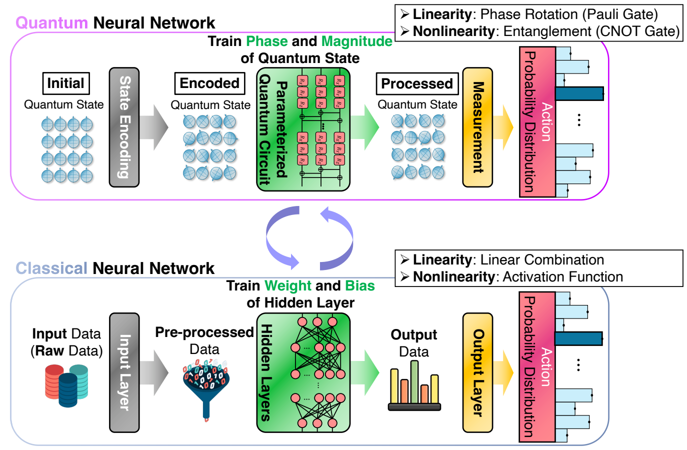


Fig. 2. Comparison of NN using classical computing and quantum computing.

Related Work. A study validated PID control designs for scale-model and full-size aircrafts by demonstrating their effectiveness in maintaining dynamic similarity [11]. Another study proposed a fixed-time adaptive fuzzy fault-tolerant control for wing micro-aerobic vehicles that flake with wing damage, ensuring convergence of the tracking error despite disturbances [12]. However, these classical controllers face the limitation of requiring a redesign under dynamics. To address this issue, methods based on artificial intelligence, such as RL, have been studied for aircraft control [13], [14]. A study proposed an anti-skid braking system for aircrafts, combining traditional methods with adaptive RL to improve efficiency and robustness under diverse conditions [15]. Despite these advances, conventional RL methods often face challenges as a result of the exponential growth of parameters. As a solution, QRL has been shown to be able to effectively reduce the number of parameters required compared to classical RL methods. One study introduced a framework based on the variational quantum circuit (VQC) for deep reinforcement learning (DRL), demonstrating that QRL can achieve performance comparable to classical methods while significantly reducing parameter counts [8]. Another study proposed a method using VQC to efficiently encode state-action mappings, thereby reducing the number of trainable parameters required in RL tasks [16].

III. Quantum Neural Network

Quantum Computing Basics. The qubit is analogous to the classical bit in traditional computing [17]. However, unlike classical bits, which are restricted to binary states of $|0\rangle$ and $|1\rangle$, qubits can exist in a superposition of both states simultaneously [18]. Mathematically, a qubit is represented as a linear combination of two basis states as $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, where α and β the probability amplitude corresponding to the states $|0\rangle$ and $|1\rangle$, which satisfies equation $|\alpha|^2 + |\beta|^2 = 1$. This superposition property enables qubits to encode exponentially more information compared to classical bits. Additionally, Bloch spheres are widely used to visualize qubit states, representing the state of a single qubit as a point on or inside a unit sphere. The general state of a qubit can be expressed in spherical coordinates as $|\psi\rangle = \cos(\frac{\theta}{2})|0\rangle + e^{i\phi}\sin(\frac{\theta}{2})|1\rangle$, where θ and ϕ are the latitude and longitude angles that determine the qubit's position on the Bloch sphere, respectively ($\theta \in [0,\pi], \phi \in [0, 2\pi]$). This geometric representation is particularly useful for understanding quantum operations, as quantum gates can be visualized as rotations around specific axes of the Bloch sphere. In addition to superposition, qubits exhibit entanglement, a key property in which the state of each qubit depends on the others, even over large distances [19].

Quantum Neural Network Structure. Although QNN shares similarities with classical NN, it differs fundamentally in design and operation. Classical NN processes input data through layers, using weights and biases to transform it into output. In contrast, QNN begins with an initial quantum state, encodes it, processes it in a parameterized quantum circuit (PQC) to train the phase and magnitude, and measures the quantum state to extract classical information. Although both networks rely on linearity and non-linearity, they achieve these differently. Classical NN implements linearity via input combinations and nonlinearity through activation functions. However, QNN achieves linearity with phase rotations, such as Pauli gates, and nonlinearity through quantum entanglement, such as CNOT gates. These distinctions highlight their differing mechanisms for similar goals. A typical QNN comprises three components: state encoding, PQC, and measurement. Fig. 2 compares classical NN and QNN. The QNN typically consists of three main components: state encoding, parameterized quantum circuit (PQC), and measurement. Firstly, the *state encoding* stage encodes classical or quantum data in the quantum state of qubits using techniques such as amplitude, angle, or basis encoding, depending on the problem and resources. Mathematically, this involves applying unitary matrices $U(\cdot)$ to prepare the quantum state as the QNN input. Secondly, the *PQC* in QNN performs computations analogous to the operations in the hidden layers of classical NN. Specifically, it facilitates both linear and nonlinear transformations over the Bloch sphere. Linear transformations are implemented using rotational gates, such as the R_Γ gates ($\Gamma \in x, y, z$), which rotate quantum states around the x, y, and z axes of the sphere. Nonlinear transformations, on the other hand, are achieved through entangling gates like the controlled-NOT (CNOT) gate, which introduces correlations between qubits by entangling their states. Illustrations of CNOT gate and rotation gate operations are shown in Fig. 3. The Hadamard gate creates superposition states, letting qubits represent multiple possibilities at once. By combining these gates, the PQC can perform complex transformations, enabling the QNN to encode intricate relationships in the data. These

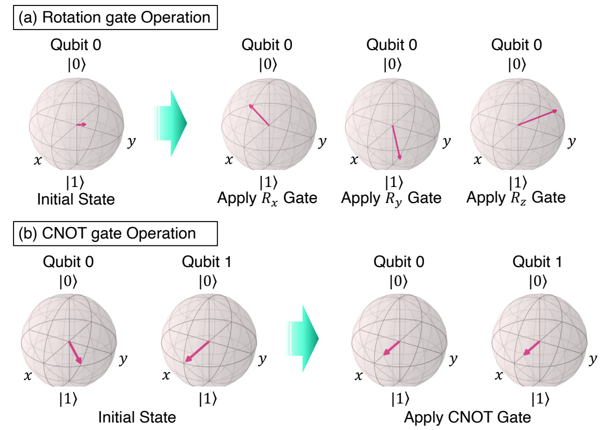


Fig. 3. Qubit state transformation through quantum gate operation.

operations are realized by strategically arranging and applying the gates in a parameterized sequence throughout the circuit. Lastly, in the *measurement* stage, the processed quantum state is measured to extract meaningful information. The measurement results are determined by the *observables*, which are operators corresponding to measurable quantities. The expectation of *observables* is calculated, providing a bridge between quantum computations and classical outputs of QNN.

Reducing the Number of Parameters. The QACN-AC algorithm introduces an innovative approach to optimizing aircraft control by integrating quantum mechanics into the RL frameworks. In this context, the agent, called the quantum aircraft, operates within a high-dimensional state and action space, which traditionally poses significant computational challenges for classical RL algorithms. Using VQC, QACN-AC reduces the number of parameters required to model and solve complex control tasks while maintaining robust performance. In classical RL, NN is commonly used to approximate policies or value functions, with parameter counts scaling quadratically as the complexity of the state and action spaces increases. This scaling leads to substantial computational and memory demands, especially for dynamic and high-dimensional systems such as aircraft control. QACN-AC addresses this challenge by employing VQC, which utilizes quantum properties such as entanglement and superposition to encode control information more efficiently. For example, while classical NN requires $2 \times n^2 + 2 \times n$ parameters for a given control problem, QACN-AC's VQC reduces this growth to a linear scale of $n \times (3 \times 2 + 1)$, significantly decreasing computational overhead [8]. The design of QACN-AC draws on the principles outlined in the VQC framework for DRL, adapting these techniques specifically for aircraft control. The algorithm comprises three key components: state encoding, PQC, and measurement. In QACN-AC, the state of the quantum aircraft is encoded using computational basis encoding, transforming classical control data into quantum states that serve as input to the VQC. These VQCs, composed of R_Γ and CNOT gates, perform both linear and non-linear transformations, enabling the circuit to effectively approximate complex control policies. The measurement phase extracts actionable information from the processed quantum state, allowing the agent to execute precise control actions. The crucial feature of QACN-AC is its iterative optimization process, which fine-tunes the parameters of the quantum circuit to minimize the RL loss function. This process not only ensures that the algorithm can learn optimal control policies, but also enhances the robustness of the quantum circuit against noise. By maintaining performance under noise while significantly reducing parameter counts, QACN-AC demonstrates its practicality for real-world applications. Through the integration of VQC and efficient parameterization, QACN-AC shows how QRL can address the scalability challenges of classical methods. By reducing computational demands without compromising performance, QACN-AC provides a robust and efficient framework for controlling quantum aircraft, paving the way for advanced control strategies in dynamic and high-dimensional environments. By significantly reducing parameter requirements and enabling efficient control in high-dimensional environments, QACN-AC demonstrates the practical advantages of applying quantum mechanics for precise aircraft control.

IV. Quantum Aircraft Control Systems

A. Coordinate System and Control Surfaces of the Aircraft

In the study of aircraft flight dynamics, the body-axis system is a fundamental coordinate framework affixed to the aircraft itself, facilitating precise analysis of its motion and control action. The body axis is defined with the aircraft's center of gravity (CG) as the origin. This coordinate system moves with the aircraft even if the aircraft changes position during flight, making it easy to analyze the forces and moments acting on the aircraft. As shown in Fig. 1, the x-axis aligns with the longitudinal axis of the fuselage, pointing forward from the tail to the nose; the y-axis lies along the lateral axis, extending out through the aircraft wings in a right-wing-positive convention; and the z-axis is oriented perpendicular to both x and y, pointing downward through the belly of the aircraft by the right rule. In the body-axis coordinate system, the rolling, pitching, and yawing moments of the quantum aircraft are closely tied to the principal control surfaces, namely the ailerons, elevators, and rudder, respectively, each producing a rotational response about the corresponding body axis. The rolling moment arises about the x-axis (which runs along the longitudinal axis of the aircraft) and is primarily governed by aileron deflections. When the ailerons are tilted asymmetrically, *e.g.*, the right aileron deflects upward while the left aileron deflects downward, the lift distribution across the wings becomes unbalanced, creating a net rolling moment that inclines the aircraft to roll toward one wing. The pitching moment occurs on the y-axis (the lateral axis) and is predominantly controlled by the elevators. Located on the horizontal stabilizer at the tail, the elevators modulate the local aerodynamic force at the tailplane. A nose-up elevator deflection increases tail download, shifting the overall pitching moment in the aircraft body frame so that the nose rises. Conversely, a nose-down deflection decreases tail download, reducing the nose-up pitching moment, and thereby pitching the aircraft downward. Finally, the yawing moment acts about the z-axis (the vertical axis), determined primarily by the deflection

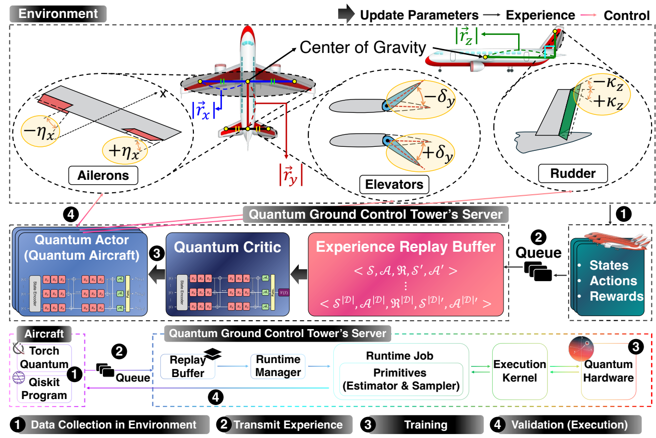


Fig. 4. Training procedure of QACN-AC for ensuring the aircraft's stability.

of the rudder on the vertical stabilizer. The yaw control surface changes the side force on the fin.

B. Attitude Angle Control of the Quantum Aircraft

This paper assumes that the aircraft coordinate system is the 3-degree-of-freedom (3-DOF) body axis. The goal of the proposed QACN-AC in quantum aircraft is to ensure the static stability of the aircraft in an environment with random vortices and turbulence. In other words, because the aircraft maneuvers on three axes, the main goal of the quantum aircraft is to ensure stability for each axis, which are the lateral (x-axis), longitudinal (y-axis), and directional (z-axis) stability. It is essential to grasp the attitude angle for each axis of the quantum aircraft at certain times to achieve this goal. At time $t+\Delta t$ on the body axis, the attitude angles for each axis of the quantum aircraft can be expressed as [20],

$$\vartheta_x(t+\Delta t) = \vartheta_x(t) + |\vec{\omega}_x(t)| \int_t^{t+\Delta t} \Delta t + \frac{1}{2}|\vec{\lambda}_x(t)| \left(\int_t^{t+\Delta t} \Delta t \right)^2, \tag{1}$$

$$\vartheta_y(t+\Delta t) = \vartheta_y(t) + |\vec{\omega}_y(t)| \int_t^{t+\Delta t} \Delta t + \frac{1}{2}|\vec{\lambda}_y(t)| \left(\int_t^{t+\Delta t} \Delta t \right)^2, \tag{2}$$

$$\vartheta_z(t+\Delta t) = \vartheta_z(t) + |\vec{\omega}_z(t)| \int_t^{t+\Delta t} \Delta t + \frac{1}{2}|\vec{\lambda}_z(t)| \left(\int_t^{t+\Delta t} \Delta t \right)^2, \tag{3}$$

where $\vartheta_x(t)$, $\vartheta_y(t)$, $\vartheta_z(t)$, $\vartheta_x(t+\Delta t)$, $\vartheta_y(t+\Delta t)$, and $\vartheta_z(t+\Delta t)$ denote the attitude angles on each axis at time t and $t+\Delta t$. Here, the variables $|\vec{\omega}_x(t)|$, $|\vec{\omega}_y(t)|$, and $|\vec{\omega}_z(t)|$ represent the angular velocities; and $|\vec{\lambda}_x(t)|$, $|\vec{\lambda}_y(t)|$, and $|\vec{\lambda}_z(t)|$ correspond to the angular accelerations, along the x-, y-, and z-axes at time t. The angular velocities for each axis can be re-expressed as,

$$|\vec{\omega}_x(t)| = |\vec{\omega}_x(t-\Delta t)| + |\vec{\lambda}_x(t-\Delta t)| \int_{t-\Delta t}^{t} \Delta t, \tag{4}$$

$$|\vec{\omega}_y(t)| = |\vec{\omega}_y(t-\Delta t)| + |\vec{\lambda}_y(t-\Delta t)| \int_{t-\Delta t}^{t} \Delta t, \tag{5}$$

$$|\vec{\omega}_z(t)| = |\vec{\omega}_z(t-\Delta t)| + |\vec{\lambda}_z(t-\Delta t)| \int_{t-\Delta t}^{t} \Delta t, \tag{6}$$

where $|\vec{\omega}_x(t-\Delta t)|$, $|\vec{\omega}_y(t-\Delta t)|$, and $|\vec{\omega}_z(t-\Delta t)|$ denote the angular velocities; and $|\vec{\lambda}_x(t-\Delta t)|$, $|\vec{\lambda}_y(t-\Delta t)|$, and $|\vec{\lambda}_z(t-\Delta t)|$ denote the angular accelerations for each axis at time $t-\Delta t$. The angular accelerations for each axis are expressed as,

$$|\vec{\lambda}_x(t)| = \frac{2|\vec{\mathcal{M}}_x(t)| - |\vec{\mathcal{M}}_x^V(t)|}{I_x} = \frac{2|\vec{r}_x||\vec{F}_x| \cdot \sin(\eta_x(t)) - \vec{r}_x^V \times \vec{F}_x^V}{I_x}, \tag{7}$$

$$|\vec{\lambda}_y(t)| = \frac{2|\vec{\mathcal{M}}_y(t)| - |\vec{\mathcal{M}}_y^V(t)|}{I_y} = \frac{2|\vec{r}_y||\vec{F}_y| \cdot \sin(\delta_y(t)) - \vec{r}_y^V \times \vec{F}_y^V}{I_y}, \tag{8}$$

$$|\vec{\lambda}_z(t)| = \frac{|\vec{\mathcal{M}}_z(t)| - |\vec{\mathcal{M}}_z^V(t)|}{I_z} = \frac{|\vec{r}_z||\vec{F}_z| \cdot \sin(\kappa_z(t)) - \vec{r}_z^V \times \vec{F}_z^V}{I_z}. \tag{9}$$

The moments $|\vec{\mathcal{M}}_x(t)|$, $|\vec{\mathcal{M}}_y(t)|$, and $|\vec{\mathcal{M}}_z(t)|$ denote the moments, *i.e.,* torques generated by the control surfaces; and $|\vec{\mathcal{M}}_x^V(t)|$, $|\vec{\mathcal{M}}_y^V(t)|$, and $|\vec{\mathcal{M}}_z^V(t)|$ denote the moments caused by random vortices and turbulence; and I_x, I_y, and I_z represent the moments of inertia, all along the x-, y- and z-axes at time t. The $|\vec{r}_x|$, $|\vec{r}_y|$, and $|\vec{r}_z|$ denote the distances (commonly called moment arms) from the CG of the aircraft to the force application points, while $|\vec{F}_x|$, $|\vec{F}_y|$, and $|\vec{F}_z|$ represent the forces generated by the control surfaces, all along the x -, y -, and z axes. This paper assumes that the aircraft operates under cruising conditions, ensuring that the forces along each axis remain constant over time. In addition, the distances from the aircraft CG to each control surface are based on `Boeing's B777-300X` [9]. Similarly, $\vec{r}_x^V$, $\vec{r}_y^V$, and $\vec{r}_z^V$ represent the distances from the CG of the aircraft to the action points of random vortices and turbulence, while $\vec{F}_x^V$, $\vec{F}_y^V$, and $\vec{F}_z^V$ are the corresponding turbulence forces, along the axes x-, y-, and z. Lastly, $\eta_x(t)$, $\delta_y(t)$, and $\kappa_z(t)$ represent the deflection angles of the aircraft control surfaces on each respective axis. Specifically, as shown in Fig. 4, they correspond to the deflection angles of the ailerons ($\eta_x(t)$), elevators ($\delta_y(t)$) and rudder ($\kappa_z(t)$), respectively, which are the actions of quantum aircraft. Changes in the deflection angles of the control surfaces for each axis alter the angular accelerations, which in turn modify the attitude angles of the quantum aircraft around each respective axis. Therefore, the quantum aircraft monitors its attitude angles in environments containing random vortices and, based on these observations, adjusts the deflection angles of the ailerons, elevators, and rudder to maintain stability.

Proof. Proof of angular accelerations for each axis. Suppose the angular momentum of an aircraft is $\vec{L}(t)$, the position vector is $\vec{r}(t)$, and the momentum is $\vec{P}(t)$ at time t. Then the angular momentum of the quantum aircraft is defined as $\vec{L}(t) = \vec{r}(t) \times \vec{P}(t)$. The time derivative of the angular momentum is denoted by, $\frac{d\vec{L}(t)}{dt} = \frac{d\vec{r}(t)}{dt} \times \vec{P}(t) + \vec{r}(t) \times \frac{d\vec{P}(t)}{dt}$. Then, the first term on the right-hand side is zero for the following physical constraint, $\frac{d\vec{r}(t)}{dt} \times \vec{P}(t) = \vec{v}(t) \times m\vec{v}(t) = m|\vec{v}(t)|^2 \sin(0^\circ) = 0$, where m and $\vec{v}(t)$ denote the mass and velocity of the quantum aircraft. Furthermore, the second term on the right-hand side is derived from Newton's second law of motion and is expressed as $\vec{r}(t) \times \frac{d\vec{P}(t)}{dt} = \vec{r}(t) \times \vec{F}(t)$, where F denotes the force of the quantum aircraft. In conclusion, $\frac{d\vec{L}(t)}{dt}$ can be expressed as $\frac{d\vec{L}(t)}{dt} = \vec{r}(t) \times \vec{F}(t) = \mathcal{M}(t)$, where $\mathcal{M}(t)$ represents the moment of the quantum aircraft. Moreover, the angular momentum of the quantum aircraft can be expressed as $\vec{L}(t) = I\vec{w}(t)$, where I denotes the moment of inertia. As mentioned above, because angular momentum is defined as $\vec{L}(t) = \vec{r}(t) \times \vec{P}(t)$, it can be re-expressed as, $\vec{L}(t) = \vec{r}(t) \times m\vec{v}(t) = \left(m|\vec{r}(t)|^2\right) \cdot \left(\frac{\vec{r}(t) \times \vec{v}(t)}{|\vec{r}(t)|^2}\right) = I\vec{\omega}(t)$, where $\vec{\omega}(t) = \frac{\vec{r}(t) \times \vec{v}(t)}{|\vec{r}(t)|^2}$ and $I = m|\vec{r}(t)|^2$. Therefore, the following physical relationship is established, $\frac{d\vec{L}(t)}{dt} = \frac{d}{dt}(I\vec{\omega}(t)) = I\frac{d}{dt}\vec{\omega}(t) = I\vec{\lambda}(t) = \vec{\mathcal{M}}(t)$, where $\vec{\lambda}(t)$ denotes the angular acceleration of quantum aircraft at time t, which can be expressed as, $\lambda(t) = \frac{d}{dt}\vec{\omega}(t) = \frac{\vec{\mathcal{M}}(t)}{I}$. Therefore, the angular accelerations about each axis of the quantum aircraft are expressed in terms of '$\lambda(t) = \frac{\vec{\mathcal{M}}(t)}{I}$'. Furthermore, unlike the rudder, the ailerons and elevators are mounted on both main wings; thus, the moments generated by the ailerons and elevators are multiplied by a factor of two compared to those produced by the rudder. □

V. Algorithm Design

Fig. 4 shows the training procedure of the proposed QACN-AC to ensure the static stability of the quantum aircraft. It is assumed that the quantum aircraft, *i.e.*, actor, is equipped with portable quantum computers. Furthermore, the quantum ground control tower, *i.e.*, critic, is also equipped with quantum computers. ① As shown in Fig. 4, the quantum aircraft controls its deflection angles of the ailerons, elevators, and rudder while collecting environmental experiences that include its states, actions, and rewards. ② The quantum aircraft then transmits its experience (experience replay buffer size $\mathcal{D}$) to the quantum ground control tower server. ③ Within the quantum ground control tower's server, a critic trains parameters for the actor by assessing the decision-making proficiency of quantum aircraft under particular states during the overall training. ④ Upon completion of training, the quantum aircraft controls its control surfaces according to the trained policy to ensure static stability.

A. QACN-AC Formulation for Quantum Aircraft

The primary objective of the quantum aircraft is to robustly maintain static stability about each axis, even when random vortices affect the aircraft. To achieve this robust stability objective, the state, action, and reward are defined as follows.

State. The state of the quantum aircraft is defined as,

$$\begin{aligned}
\mathcal{S}(t) \triangleq \{ &\bigcup_{t=0}^{t=\mathcal{T}} \{\vartheta_x(t)\}, \bigcup_{t=0}^{t=\mathcal{T}} \{\vartheta_y(t)\}, \bigcup_{t=0}^{t=\mathcal{T}} \{\vartheta_z(t)\}, \\
&\bigcup_{t=0}^{t=\mathcal{T}} \{|\vec{\omega}_x(t)|\}, \bigcup_{t=0}^{t=\mathcal{T}} \{|\vec{\omega}_y(t)|\}, \bigcup_{t=0}^{t=\mathcal{T}} \{|\vec{\omega}_z(t)|\}, \\
&\bigcup_{t=0}^{t=\mathcal{T}} \left\{|\vec{\mathcal{M}}_x^V(t)|\right\}, \bigcup_{t=0}^{t=\mathcal{T}} \left\{|\vec{\mathcal{M}}_y^V(t)|\right\}, \bigcup_{t=0}^{t=\mathcal{T}} \left\{|\vec{\mathcal{M}}_z^V(t)|\right\}, \\
&\bigcup_{t=0}^{t=\mathcal{T}} \{\eta_x(t)\}, \bigcup_{t=0}^{t=\mathcal{T}} \{\delta_y(t)|\}, \bigcup_{t=0}^{t=\mathcal{T}} \{\kappa_z(t)\}\}.
\end{aligned} \tag{10}$$

In (10), the variables, *i.e.,* $\vartheta_{(\mathcal{P})}(t)$, $|\vec{\omega}_{(\mathcal{P})}(t)|$, and $|\vec{\mathcal{M}}^{V}_{(\mathcal{P})}(t)|$ represent the attitude angles, the magnitudes of angular velocities, and the magnitudes of the moments caused by random vortices and turbulence at time t, respectively, where $(\mathcal{P})$ denotes the corresponding axis (x, y, z), *i.e.,* $(\mathcal{P}) \in \{x, y, z\}$.

Action. In dynamic and uncertain flight environments with random vortices, the quantum aircraft controls its deflection angles of the ailerons ($\eta_x(t)$), elevators ($\delta_y(t)$), and rudder ($\kappa_z(t)$) at time t to ensure static stability, which is shown in Fig. 4. Therefore, in the stable control setting of the quantum aircraft, the action of the aircraft is defined as, $\mathcal{A}(t) \triangleq \{\eta_x(t), \delta_y(t), \kappa_z(t)\}$. The deflection angles of the ailerons, elevators, and rudder are subject to constraints *i.e.* $\eta_x(t) \in [-\frac{\pi}{4}, \frac{\pi}{4}]$, $\delta_y(t) \in [-\frac{\pi}{3}, \frac{\pi}{3}]$, and $\kappa_z(t) \in [-\frac{\pi}{4}, \frac{\pi}{4}]$, respectively, whose units are in both *radians*. In essence, the QNN-equipped quantum aircraft observes the state at time t and generates continuous deflection angle values of the control surfaces to maximize the reward function defined in (11).

Reward. The quantum aircraft is designed to maintain static stability under all conditions, even when encountering random vortices during flight. To achieve this, the quantum aircraft must monitor its states, *e.g.,* attitude angles, angular velocities, etc, and control the deflection angles of the ailerons, elevators, and rudder to align its axes with those of the body-axis coordinate system. Ultimately, the quantum aircraft must ensure that, even when subjected to random vortices, its attitude angles converge rather than diverge over time, resulting in the relative angles with respect to each axis of the body-axis coordinate system approaching zero. Therefore, the reward function of the quantum aircraft is formulated in (11), which must be maximized based on its action decisions, *i.e.,* $\mathcal{A}(t) \triangleq \{\eta_x(t), \delta_y(t), \kappa_z(t)\}$.

$$\Re\left(\mathcal{S}(t), \mathcal{A}(t)\right) \triangleq \left\{\sqrt{\vartheta_x(t)^2 + \vartheta_y(t)^2 + \vartheta_z(t)^2} + \varrho\right\}^{-1}. \tag{11}$$

In (11), $\vartheta_x(t)$, $\vartheta_y(t)$, and $\vartheta_z(t)$ represent the attitude angles on each axis at time t. The ϱ denotes an adequately small constant value that ensures the reward function's divergence is avoided. Because the objective of the quantum aircraft is $\sqrt{\vartheta_x(t)^2 + \vartheta_y(t)^2 + \vartheta_z(t)^2} \rightarrow 0$, ϱ is used to prevent the denominator of the reward from becoming zero, thus ensuring a well-defined mathematical formulation of the reward.

B. Policy Training of QACN-AC

Because the action space of the quantum aircraft is defined by the deflection angles of its control surfaces, precise values for these angles are required. Even a very small angular difference in the control surfaces, such as 1°, affects the static stability of quantum aircraft. Therefore, utilizing a continuous action space is essential. The problem in a continuous action space is that the action space is near infinity, making sampling-based methods inefficient or impossible. In light of this requirement, a particular RL-based method appears to be a promising approach, capable of handling continuous action domains without segmenting them into discrete parts that undermine training performance by discarding structural information. The particular RL-based method utilizes a deterministic policy (Π) not stochastic (π). The stochastic policy implies a probability distribution over all actions, whereas a deterministic policy assigns probability to only a single action. Given that quantum aircraft equipped with QACN-AC aim to *maximize the expected cumulative reward*, referred to as the *return*, *i.e.,* $G_t(t)$, the objective function, *i.e.,* $\Xi(\zeta)$, is formulated as, $\Xi(\zeta) = \mathbb{E}_{\mu\sim\Pi_\zeta}\left[\sum_{t=0}^{\tau} G_t(t)\right]$, where μ, τ, Π_ζ, and ζ denote the training trajectory of the quantum aircraft, its trajectory horizon, the policy parameterized by ζ and the QNN parameters of the actors, respectively. Here, the return is expressed as, $G_t(t) = \Re\left(\mathcal{S}'(t+1), \mathcal{A}'(t+1)\right)_{t+1} + \gamma\left(\mathcal{S}''(t+2), \mathcal{A}''(t+2)\right)_{t+2} + \gamma^2\left(\mathcal{S}'''(t+3), \mathcal{A}'''(t+3)\right)_{t+3} + \cdots = \sum_{i=0}^{i=\tau} \gamma^i \left(\mathcal{S}(t), \mathcal{A}(t)\right)_{t+i+1}$, where γ represents the discount factor, which acts as the weighting of future rewards relative to current rewards. This discount factor reflects future uncertainty, placing greater (or lesser) emphasis on near-term rewards. Given that $\gamma \in [0, 1]$, repeated multiplication progressively reduces the reward to zero. The optimal policy (marked as $(\cdot)^*$) that maximizes the expected return is represented as, $\Pi^*_\zeta = \mathrm{argmax}_\zeta\, \mathbb{E}_{\mu\sim\Pi_\zeta}[G_t(t)]$. The proposed algorithm, *i.e.,* QACN-AC, employs a policy gradient (PG) strategy to train quantum actor-critic networks. Then, the gradient of the *actor* network's objective function $\Xi(\zeta)$ is given by,

$$\nabla_\zeta \Xi(\zeta) = \mathbb{E}_{\mathcal{S}}\Big[\sum_{t=1}^{\tau} Q_\chi(\mathcal{S}(t), \mathcal{A}(t)) \cdot \nabla_\zeta \log \Pi_\zeta(\mathcal{A}'(t)|\mathcal{S}(t); \zeta)\Big], \tag{12}$$

where $\mathcal{A}'(t+1) \sim (\mathcal{A}(t) + \mathcal{N}(0, \varsigma^2))$. The quantum aircraft's action at the next time step, *i.e.,* $\mathcal{A}'(t+1)$, is generated by adding noise $\mathcal{N}(\cdot)$ to the previous action $\mathcal{A}(t)$. Furthermore, $\mathcal{N}(0, \varsigma^2)$ represents Gaussian noise with a mean of 0 and variance ς^2, thus supporting effective exploration of quantum aircraft in the dynamic environment. Then, the gradient of the *critic* network's loss function can be expressed as,

$$\nabla_\chi \Psi(\chi) = \sum_{t=1}^{\tau} \nabla_\chi \left\|\sigma_\chi(t)\right\|^2, \tag{13}$$

where $\Psi(\chi)$, χ, and $\sigma_\chi(t)$ denote the loss function of the critic, critic QNN parameters, and temporal difference (TD) error. The TD error employed in training leverages the *Bellman optimality equation* to measure the discrepancy between the predicted and actual values (expected return $G_t(t)$). It can be expressed as,

$$\begin{aligned}\sigma_\chi(t) = {} & \Re\left(\mathcal{S}'(t+1), \mathcal{A}'(t+1)\right) \\ & + \gamma Q_\chi(\mathcal{S}(t+1), \mathcal{A}(t+1)) - Q_\chi(\mathcal{S}(t), \mathcal{A}(t)),\end{aligned} \tag{14}$$

where $\Re(\cdot)$ and $Q_\chi(\cdot)$ represent the reward function and the state-action value function. Backpropagation is applied to carry out gradient *ascent* on the *actor's objective function* and gradient *descent* on the *critic's loss function*, utilizing the derivatives with respect to actor's (ζ_k) and critic's (χ_k) k-th QNN parameters. However, the quantum circuit states that constitute the QACN-AC involve complex quantum superpositions and entanglements, making it impossible for classical backpropagation to accurately update these NN parameters. Therefore, the gradients of the actor's objective function and the critic's loss function are calculated using the parameter shift rule (PSR), which can be expressed as,

TABLE I
SYSTEM PARAMETERS FOR PERFORMANCE EVALUATION [9]

Notation	Value
Span of the B777-300X	204.4 ft
Length of the B777-300X	242.1 ft
Height of the B777-300X	60.4 ft
Distance CG-ailerons ($\lvert\vec{r}_x\rvert$)	102.2 ft
Distance CG-elevators/rudder ($\lvert\vec{r}_y\rvert, \lvert\vec{r}_z\rvert$)	157.4 ft
Constant prevent reward divergence (ϱ)	10^{-3}
Discount factor (γ) and batch size	0.98, 128
Experience replay buffer size ($\mathcal{D}$)	10^6
Learning rate of *actor* and *critic*	10^{-4}, 5×10^{-4}
Training episode	10^4
Activation function and optimizer	ReLU and Adam

TABLE II
THE NUMBER OF TRAINING PARAMETERS

	Norm. Parameters	# Num. Parameters
QACN-AC (Proposed)	**0.13%**	**56**
Comp-1	100%	42,557
Comp-2	99.69 %	42,428
Comp-3	99.69 %	42,428

$$\frac{\partial \Xi(\zeta)}{\partial \zeta_k} = \frac{\partial \Xi(\zeta)}{\partial \Pi_\zeta} \cdot \frac{\partial \Pi_\zeta}{\partial \langle \mathcal{O}_{k,\zeta}\rangle} \cdot \frac{\partial \langle \mathcal{O}_{k,\zeta}\rangle}{\partial \zeta_k}, \tag{15}$$

$$\frac{\partial \Psi(\chi)}{\partial \chi_k} = \frac{\partial \Psi(\chi)}{\partial Q_\chi} \cdot \frac{\partial Q_\chi}{\partial \langle \mathcal{O}_{k,\chi}\rangle} \cdot \frac{\partial \langle \mathcal{O}_{k,\chi}\rangle}{\partial \chi_k}, \tag{16}$$

where $\langle \mathcal{O}_{k,\zeta}\rangle$ and $\langle \mathcal{O}_{k,\chi}\rangle$ represent the *observable* of actor and critic corresponding to the k-th basis. The first and second derivatives on the right-hand side of (15)–(16) can be determined by classical backpropagation. However, the third derivative cannot be derived using classical methods because the quantum state remains unknown, *i.e.*, unobservable, until its collapse via measurement. Consequently, the PSR is used for parameter updates during QACN-AC training. When applied to the derivative of the actor's k-th parameter with the 0-th derivative, the PSR can be represented as,

$$\frac{\partial \langle \mathcal{O}_{k,\zeta}\rangle}{\partial \theta_k} = \langle \mathcal{O}_{k,\zeta+\frac{\pi}{2}e_k}\rangle - \langle \mathcal{O}_{k,\zeta-\frac{\pi}{2}e_k}\rangle. \tag{17}$$

Here, e_k denotes the k-th basis of QNN. Compared to backpropagation, the PSR provides a more straightforward, direct procedure, thereby accelerating training in QNN [10].

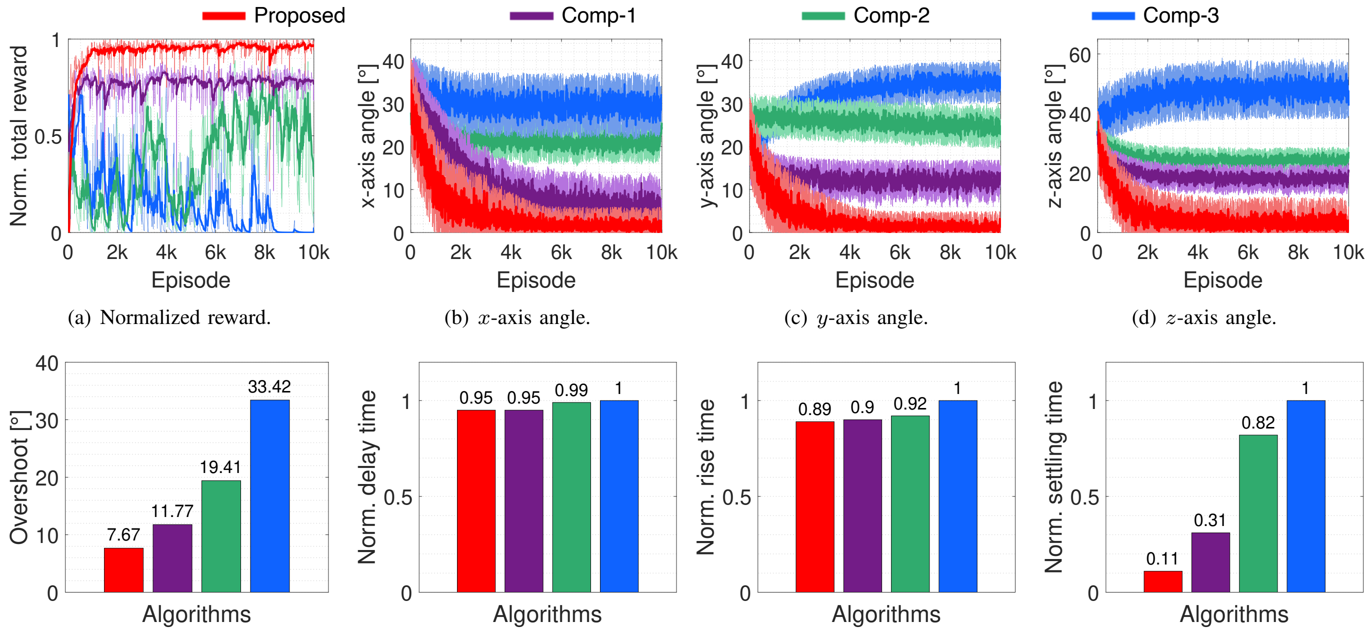
VI. PERFORMANCE EVALUATION

A. Simulation Setup

The QACN-AC is implemented using `torchquantum` Python library, which facilitates the integration of `IBM Qiskit` through interface functions. Detailed information on the parameters applied in performance evaluation is provided in Table I. Then, various *conventional RL*-based benchmarks with *classical NN* are employed, *i.e.*, *i) Comp-1: Actor-Critic*, *ii) Comp-2: Reinforce*, and *iii) Comp-3: Deep Q-network (DQN)*.

B. Result

As depicted in Fig. 5(a), the quantum aircraft utilizing the proposed QACN-AC achieves the highest reward value. This algorithm achieves a reward value approximately 1.25 times higher than Comp-1, which attained the highest rewards among conventional RL methods. With the exception of QACN-AC, the normalized reward values are highest for Comp-1, followed by Comp-2 and Comp-3. Comp-3 completely fails to converge reward values, which prevents the aircraft from achieving robust stability. Because the reward is inversely proportional to the angles about each axis of the aircraft, a high reward value indicates that the aircraft effectively reduces its tilted attitude angles and maintains stability, even when encountering random turbulence. Figs. 5(b)-(d) illustrate the changes in the absolute value of the angle for each axis throughout the training phase. It is observed that only the quantum aircraft employing the QACN-AC consistently ensures static stability by maintaining attitude angles close to zero across all axes. All axes' angles approaching zero indicate that the quantum aircraft can maintain stable and immediate attitude control, even when exposed to vortices. In contrast, aircraft equipped with conventional RL algorithms do not stabilize their pitch, roll, and yaw angles. Specifically, Comp-3 fails to stabilize the pitch and yaw angles, causing these angles to diverge as training progresses. Figs. 5(e)-(h) show the control performances of the quantum aircraft, *i.e.,* overshoot, delay time, rise time, and settling time. Overshoot refers to the extent to which a system's response exceeds the desired reference value before settling. A lower overshoot is generally preferred as it indicates that the control system avoids excessive deviation beyond the desired setpoint. The proposed algorithm shows the smallest overshoot angle, which means that the aircraft's attitude angle is reliably controlled. The delay time and rise time represent the time that the control system outputs up to 50% and the period during which the control system's output transitions from 10% to 90% of the target value. A shorter delay time is desirable as it indicates a faster reaction of the control system to changes in input. Furthermore, a shorter rise time implies a faster response of the system to reach the target value. However, excessively short rise times may lead to increased overshoot or undesirable oscillations. As shown in Figs. 5(f)-(g), the proposed algorithm shows the shortest delay time and rise time performances, balancing rapid response with stability. Settling time is the time required for the control system's output to enter and remain within a defined tolerance band around the desired reference value. The proposed algorithm has the shortest settling time performance. Lower settling times indicate that the system stabilizes more quickly after the random vortices, enhancing overall responsiveness. Table II presents the number of training parameters required. The benchmarks, which utilize classical NN for training aircraft, each require in excess of $40,000$ parameters. Specifically, both Comp-2 and Comp-3 require $42,428$ training parameters, while Comp-1 requires $42,557$ training parameters. In contrast, the QACN-AR requires only 56 parameters to train quantum aircraft. This reduction represents an approximately 760-fold decrease in training parameters compared to conventional RL-based aircraft control algorithms. QACN-AC reduces the number of parameters exponentially, enabling lightweight quantum aircraft realization.



(a) Normalized reward. (b) x-axis angle. (c) y-axis angle. (d) z-axis angle.

(e) Overshoot. (f) Delay time. (g) Rise time. (h) Settling time.

Fig. 5. Trends in the normalized total reward, variations in the absolute angle values for each axis, and control performances.

VII. Conclusion

QACN-AC has been shown to effectively mitigate aircraft instability by reducing training parameters and computational overhead. Through realistic simulations, QACN-AC outperforms conventional RL methods, demonstrating its promise for lightweight, adaptive, and robust aircraft control.

References

[1] X. Zeng, X. Zhang, and F. Wang, "Joint UAV trajectory and transceiver optimization for over-the-air computation systems," in *Proc. IEEE/IFIP International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, Singapore, August 2023, pp. 479–484.

[2] M. Milanesi, L. Consolini, G. D. Credico, N. Latronico, M. Laurini, M. Paltenghi, M. Schiavo, and A. Visioli, "Human-imitating control of depth of hypnosis combining MPC and event-based PID strategies," *IEEE Control Systems Letters*, vol. 8, pp. 580–585, May 2024.

[3] S. Lee, G. S. Kim, S. Park, and J. Kim, "Advanced taxiing path guidance using multi-agent reinforcement learning for air traffic management," in *Proc. IEEE/IFIP International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, Seoul, Korea, October 2024, pp. 305–312.

[4] G. S. Kim, J. Chung, and S. Park, "Realizing stabilized landing for computation-limited reusable rockets: A quantum reinforcement learning approach," *IEEE Transactions on Vehicular Technology*, vol. 73, no. 8, pp. 12 252–12 257, August 2024.

[5] G. S. Kim, S. Lee, T. Woo, and S. Park, "Cooperative reinforcement learning for military drones over large-scale battlefields," *IEEE Transactions on Intelligent Vehicles*, pp. 1–11, 2024 (Early Access).

[6] Z. Zhao, Y. Che, S. Luo, K. Wu, and V. C. M. Leung, "Multi-agent graph reinforcement learning based on-demand wireless energy transfer in multi-UAV-aided IoT network," in *Proc. IEEE/IFIP International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt)*, Singapore, August 2023, pp. 479–484.

[7] S. Park, J. Chung, C. Park, S. Jung, M. Choi, S. Cho, and J. Kim, "Joint quantum reinforcement learning and stabilized control for spatio-temporal coordination in metaverse," *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 12 410–12 427, December 2024.

[8] S. Y. Chen, C. H. Yang, J. Qi, P. Chen, X. Ma, and H. Goan, "Variational quantum circuits for deep reinforcement learning," *IEEE Access*, vol. 8, pp. 141 007–141 024, July 2020.

[9] P. Birtles, *Boeing 777*. Zenith Press, 1998.

[10] S. Park, G. S. Kim, Z. Han, and J. Kim, "Quantum multi-agent reinforcement learning is all you need: Coordinated global access in integrated TN/NTN cube-satellite networks," *IEEE Communications Magazine*, vol. 62, no. 10, pp. 86–92, October 2024.

[11] T. Yue, X. Zuo, L. Wang, J. Geng, and H. Zhang, "Similarity relations of PID flight control parameters of scaled-model and full-size aircraft," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 58, no. 4, pp. 2950–2960, August 2022.

[12] H. Long, P. Zhang, T. Guo, and J. Zhao, "Fixed-time adaptive fuzzy fault-tolerant control of flapping wing MAVs with wing damage," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 60, no. 5, pp. 6594–6607, October 2024.

[13] N. Kim, W. Na, D. S. Lakew, N. Dao, and S. Cho, "DQN-based directional MAC protocol in wireless ad hoc network in Internet of things," *IEEE Internet of Things Journal*, vol. 11, no. 7, pp. 12 918–12 928, April 2024.

[14] T. T. H. Pham, W. Noh, and S. Cho, "Multi-agent reinforcement learning based optimal energy sensing threshold control in distributed cognitive radio networks with directional antenna," *ICT Express*, vol. 10, no. 3, pp. 472–478, June 2024.

[15] Z. Jiao, N. Bai, D. Sun, X. Liu, J. Li, Y. Shi, Y. Hou, and C. Chen, "Aircraft anti-skid braking control based on reinforcement learning," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 59, no. 6, pp. 9219–9232, December 2023.

[16] R. Ravish, N. R. Bhat, N. Nandakumar, S. Sagar, Sunil, and P. B. Honnavalli, "Optimization of reinforcement learning using quantum computation," *IEEE Access*, vol. 12, pp. 179 396 – 179 417, November 2024.

[17] J. A. Ansere, E. Gyamfi, V. Sharma, H. Shin, O. A. Dobre, and T. Q. Duong, "Quantum deep reinforcement learning for dynamic resource allocation in mobile edge computing-based IoT systems," *IEEE Transactions on Wireless Communications*, vol. 23, no. 6, pp. 6221–6233, June 2024.

[18] S. Zhang, X. Zhou, T. Qiu, and D. O. Wu, "Quantum-inspired robust networking model with multiverse co-evolution for scale-free IoT," *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 14 085 – 14 098, December 2024.

[19] E. C. Behrman, R. E. F. Bonde, J. E. Steck, and J. F. Behrman, "On the correction of anomalous phase oscillation in entanglement witnesses using quantum neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 9, pp. 1696–1703, September 2014.

[20] F. P. Beer, E. R. Johnston, and P. J. Cornwell, *Vector Mechanics for Engineers: Dynamics*. McGraw-Hill, 2009.