TYPE Original Research
PUBLISHED 17 August 2026
DOI 10.3389/arc.2026.17000

OPEN ACCESS

*CORRESPONDENCE
Changping Du,
duchangping@zju.edu.cn

RECEIVED 24 May 2026
REVISED 02 June 2026
ACCEPTED 24 July 2026
PUBLISHED 17 August 2026

CITATION
Yang R, Du C, Yu J, Fang T and Du Z (2026) Corridor-constrained incremental TD3 for nacelle tilt scheduling in quad-tiltrotor UAV level transition.
Aerosp. Res. Commun. 4:17000.
doi: 10.3389/arc.2026.17000

Corridor-constrained incremental TD3 for nacelle tilt scheduling in quad-tiltrotor UAV level transition

Rui Yang, Changping Du*, Jiahao Yu, Tianrui Fang and Zean Du

School of Aeronautics and Astronautics, Zhejiang University, Hangzhou, Zhejiang, China

Level-flight transition of quad-tiltrotor unmanned aerial vehicles requires a nacelle tilt scheduler that adapts to closed-loop transition states while satisfying actuator-rate limits and a speed-dependent admissible transition corridor. Offline tilt schedules provide useful nominal references, but their feedback adaptability is limited under disturbances, measurement noise, and variations in low-level closed-loop response. In contrast, absolute-action reinforcement learning (RL) policies treat consecutive nacelle commands as independent setpoints, which may induce abrupt inter-sample variations and excessive nacelle-rate demand. This paper proposes an energy-aware mooth incremental tilt strategy based on a corridor-constrained incremental Twin Delayed Deep Deterministic Policy Gradient (TD3) framework. Instead of learning the absolute nacelle angle, the actor outputs a bounded nacelle-angle increment. The command is then generated through bounded integration, rate saturation, and projection onto a tightened transition corridor. This action generation mechanism converts the policy from an independent setpoint generator into a constrained nacelle-evolution generator, embedding command continuity and actuator-rate compatibility while enforcing corridor feasibility. The scheduling objective combines altitude regulation, forward-speed buildup, longitudinal smoothness, and a rotor-speed-based effort proxy under a gain-scheduled low-level control architecture. The feasibility layer provides corridor admissibility of the sampled command, and a post-training analysis characterizes the conditional closed-loop boundedness. The simulations show that the proposed scheduler generates smoother nacelle trajectories than absolute-action TD3 and act-penalty-based TD3, reducing nacelle-rate demand and longitudinal jerk, while maintaining corridor feasibility, bounded altitude response, terminal-speed buildup, and comparable rotor-speed-based effort.

KEYWORDS
nacelle tilt scheduling, quad-tiltrotor UAV, reinforcement learning, transition corridor, transition trajectory optimization

Introduction

Tiltrotor vehicles combine the vertical take-off and landing capability of rotorcraft with the efficient cruise performance of fixed-wing aircraft, making them attractive for space-constrained deployment, long-endurance missions, and multi-regime flight operations [1, 2]. During a complete mission, the vehicle must pass through rotor-borne flight, transition flight, and wing-borne flight. Among these phases, transition flight is particularly challenging because the primary lift source gradually shifts from rotor thrust to wing aerodynamic lift, while the propulsion direction, aerodynamic characteristics, and control effectiveness vary with the nacelle angle and forward speed.

As the nacelles rotate from the vertical-lift configuration toward the forward-propulsion configuration, rotor thrust is redistributed between vertical support and forward acceleration. Meanwhile, rotor wake interacts with the wing and fuselage, resulting in significant variations in lift, drag, and pitching moment characteristics [3–5]. These aerodynamic variations are coupled with thrust-vector evolution, actuator dynamics, control allocation, and longitudinal flight-state changes. Consequently, transition dynamics exhibit strong nonlinearities, time-varying control effectiveness, and rate-limited actuation characteristics [6–8]. The transition performance is therefore highly sensitive to the nacelle tilt schedule, which directly affects altitude regulation, forward-speed buildup, actuator usage, transition smoothness, and energy-related performance.

For level-flight transition, the aircraft is required to maintain constant altitude while accelerating from the rotor-borne condition to the wing-borne condition. In this process, the nacelle angle and forward speed must satisfy a feasible matching relationship to maintain sufficient lift, power margin, and control authority. This admissible relationship is commonly represented by a transition corridor in the nacelle-angle--speed plane [9–11]. Recent studies further enrich this corridor by incorporating aerodynamic variables, angle-of-attack constraints, power limits, and attitude-safety indices [12, 13]. These studies indicate that the transition corridor provides not only a feasibility boundary, but also a natural framework for formulating nacelle tilt scheduling as a high-level transition-trajectory design problem. Accordingly, the key scheduling task is to generate a nacelle command that evolves with forward speed while satisfying corridor feasibility and actuator-rate compatibility.

Existing studies on transition trajectory design can be broadly categorized into corridor-based planning, trim-based optimization, direct collocation or pseudospectral optimization, model-predictive control, and heuristic search methods. Corridor construction and trim analysis have been used to establish feasible nacelle-angle--speed envelopes and equilibrium Refs. [7, 14]. Direct collocation, pseudospectral methods, and model-based optimization have been applied to generate constrained transition trajectories under aerodynamic, power, and attitude limitations [13, 15]. Heuristic search methods, such as ant colony optimization, have also been employed to obtain feasible transition paths by considering safety, power, and control-effort indices [12]. In addition, energy-oriented transition scheduling has received increasing attention, since different acceleration and tilt profiles may lead to distinct power demands, transition durations, and mission-level energy performance [5, 16].

Although these methods provide important insights into transition-corridor construction and nominal trajectory optimization, many of them rely on prior aerodynamic models, trim computations, lookup tables, or predefined trajectory parameterizations. The resulting schedules can be highly effective under nominal conditions, but they are usually applied in an open-loop or weakly feedback-driven manner. Their adaptability may therefore degrade when model uncertainty, wind disturbances, measurement noise, or variations in the low-level closed-loop response are present. This motivates a feedback-capable nacelle scheduler that can adapt the tilt evolution according to the measured transition state while preserving the physical feasibility of the nacelle command.

Reinforcement learning (RL) offers a promising tool for adaptive decision-making in nonlinear and uncertain systems. In UAV and eVTOL control, RL has been applied to trajectory tracking, attitude control, controller-parameter tuning, energy optimization, and disturbance-aware control [17–19]. In particular, the twin delayed deep deterministic policy gradient (TD3) algorithm is well suited for continuous-control tasks because its clipped double critics, delayed policy updates, and target policy smoothing reduce overestimation bias and improve training stability [20]. Recent work has shown that TD3 can be used to generate energy-oriented pitch-angle references for tilt-rotor quadrotors in cooperation with robust low-level controllers [19]. These results indicate the potential of RL for improving transition-related performance under uncertain operating conditions.

However, most existing RL-based studies on tiltrotor vehicles focus on pitch-angle reference generation, low-level controller synthesis, or adaptive controller tuning. The nacelle transition schedule itself has received limited attention as a constrained, smooth, and multi-objective high-level scheduling variable. This distinction is important because nacelle tilt scheduling differs from generic continuous-action control. The nacelle angle is a slow physical transition variable: adjacent commands are coupled by actuator-rate limits, its admissible range depends on forward speed through the transition corridor, and its variation directly changes thrust-vector distribution, wing-lift buildup, rotor-speed demand, and longitudinal acceleration. Therefore, a learning-based nacelle scheduler should not only improve transition performance, but also generate a physically realizable command evolution that is smooth, rate-compatible, and feasible within the admissible corridor.

This requirement renders the action representation of the learning policy a central design issue. If an RL policy directly outputs the absolute nacelle angle, consecutive commands are treated as independent setpoints. Under exploration noise, critic approximation errors, or policy updates, such an absolute-action representation may generate abrupt inter-sample variations, excessive nacelle-rate demand, or degraded longitudinal smoothness. Recent studies on action-space design in deep RL have shown that action-space shaping, parameterized actions, lifted or relative action representations, and smooth-policy regularization can significantly affect learning efficiency, trajectory smoothness, and physical compatibility [21–24]. In robotic and autonomous driving tasks, lifted or relative action spaces have been used to align learned commands with the physical evolution of the controlled system [25], while smooth-policy approaches indicate that action regularity can be enforced through policy structure rather than solely through external filtering [26]. Nevertheless, these ideas have not been tailored to quad-tiltrotor nacelle scheduling, where the learned action should reflect the incremental and rate-limited evolution of the nacelle actuator.

Action-level smoothness alone, however, is insufficient. The learned command must also satisfy hard transition-feasibility constraints. Safety filtering and projection-based safe RL have been investigated as modular ways to enforce constraints after policy output. Predictive safety filters, robust MPC filters, constrained projection layers, and optimization-based output layers have been used to modify potentially unsafe actions while preserving the learning policy as the performance optimizer [27–30]. These studies motivate a separation between performance-oriented policy learning and hard constraint enforcement. For tiltrotor level-flight transition, however, the critical feasibility constraint is not merely a fixed input bound or a generic state constraint. It is a speed-dependent nacelle-angle corridor determined by aerodynamic lift capability, power margin, and transition feasibility. Therefore, a suitable learning-based nacelle scheduler should couple action representation with feasibility enforcement, so that the learned command follows the rate-limited evolution of the nacelle actuator while the executed command remains inside the speed-dependent transition corridor.

The above discussion shows that existing transition-scheduling studies and RL-based control methods have not yet provided an integrated solution for feedback-capable nacelle scheduling, physically consistent action evolution, and hard feasibility enforcement. To address this issue, this paper proposes an energy-aware smooth incremental tilt (ESIT) strategy for the level-flight transition of a quad-tiltrotor unmanned aerial vehicle. ESIT is developed under a hierarchical transition-control architecture, where the scheduled low-level controller maintains attitude and altitude tracking, and the high-level scheduler determines the nacelle evolution from measurable transition states. A Beetle Swarm (BS) optimized reference schedule is used to warm-start the actor and provide a physically reasonable initial tilt trend. Based on this warm start, the TD3 scheduler is reformulated with an incremental action representation, so that the actor outputs a bounded nacelle-angle increment rather than an independent absolute nacelle command. The resulting command is passed through a feasibility layer that performs bounded integration, actuator-rate saturation, and projection onto a tightened speed-dependent transition corridor. In this way, ESIT combines feedback adaptability, rotor-speed-based effort awareness, actuator-compatible command smoothness, and corridor-feasible transition scheduling within a unified framework. After training, the learned actor is further analyzed as a bounded nonlinear scheduler to characterize sampled-command corridor feasibility and conditional closed-loop boundedness under bounded-disturbance and local low-level stability assumptions.

The main contributions of this paper are summarized as follows.

1. A corridor-aware feedback nacelle-scheduling formulation is developed for quad-tiltrotor level-flight transition. The formulation treats the nacelle angle as a state-dependent physical transition variable constrained by actuator-rate limits and a speed-dependent admissible corridor. It integrates forward-speed buildup, altitude regulation, rotor-effort redistribution, and longitudinal smoothness into a unified high-level scheduling problem.
2. A physics-structured incremental TD3 strategy is proposed for nacelle command generation. By learning the nacelle-angle increment rather than the absolute nacelle angle, the proposed scheduler converts the RL policy from an independent setpoint generator into a constrained nacelle-evolution generator. The resulting bounded integration, rate saturation, and tightened-corridor projection provide a direct mechanism for producing smooth, rate-compatible, and corridor-feasible tilt schedules.
3. A mechanism-oriented analysis and validation framework is established for the proposed scheduler. The feasibility layer is characterized as a rate-compatible and corridor-aware command-generation mechanism, and the trained actor is analyzed as a bounded nonlinear scheduler, leading to a conditional boundedness characterization under bounded disturbances and a common Lyapunov condition for the scheduled low-level error dynamics. Simulations, including action-parameterization ablation, corridor-feasibility evaluation, Monte Carlo tests, and observation-noise evaluation, demonstrate that the proposed incremental action structure reduces nacelle-rate demand and longitudinal jerk while maintaining corridor feasibility, bounded altitude response, terminal-speed buildup, and comparable rotor-speed-based effort.

The remainder of this paper is organized as follows. Section II formulates the constrained high-level nacelle tilt-scheduling problem for quad-tiltrotor level-flight transition. Section III presents the proposed ESIT framework, including the incremental TD3 scheduler, rate-and-corridor feasibility layer, reward design, BS-guided reference generation, TD3 optimization procedure, and post-training boundedness characterization. Section IV presents comparative simulations, ablation studies, corridor evaluation, Monte Carlo tests, and observation-noise evaluation. Section V concludes this paper.

Problem formulation

This section formulates the constrained high-level nacelle tilt-scheduling problem for the level-flight transition of a quad-tiltrotor UAV. The problem is considered under a scheduled low-level flight-control architecture. Therefore, the nacelle scheduler is treated as a high-level transition-command generator, rather than as a replacement for the stabilizing attitude and altitude controller.

Longitudinal transition model

A longitudinal force-balance model [31, 32] is introduced to describe the dominant transition dynamics. The nacelle tilt angle is denoted by λ, where $\lambda = 0^\circ$ corresponds to the rotor-borne configuration and $\lambda_f = 90^\circ$ corresponds to the fully wing-borne configuration.

Let the longitudinal state vector be defined as

$$x = [\, s \;\; V_x \;\; h \;\; V_h \;\; \theta \;\; q \,]^T, \tag{1}$$

where s denotes the longitudinal position, V_x the forward velocity, h the altitude, V_h the vertical velocity, θ the pitch angle, and q the pitch rate. The resultant longitudinal-plane airspeed used only in the aerodynamic lift and drag calculations is defined as

$$V = \sqrt{V_x^2 + V_h^2}. \tag{2}$$

The longitudinal state and resultant airspeed are defined in Equations 1, 2. The low-level control input ul is defined in Equation 3.

$$u_l = \left[\,\Omega_1 \;\; \Omega_2 \;\; \Omega_3 \;\; \Omega_4 \;\; \delta_e\,\right]^T, \tag{3}$$

where Ω_i is the rotational speed of the i-th rotor and δ_e is the elevator deflection or an equivalent longitudinal control-surface input.

The dominant longitudinal force and moment balances are given by Equations 4 -9.

$$m\dot{V}_x = F_x(x, u_l, \lambda) - D(V, \alpha) + d_V, \tag{4}$$

$$m\dot{V}_h = F_z(x, u_l, \lambda) + L(V, \alpha) - mg + d_h, \tag{5}$$

$$I_y\dot{q} = M_y(x, u_l, \lambda, \delta_e) + d_q, \tag{6}$$

$$\dot{s} = V_x, \tag{7}$$

$$\dot{h} = V_h, \tag{8}$$

$$\dot{\theta} = q, \tag{9}$$

where m is the vehicle mass, I_y is the pitch-axis moment of inertia, F_x and F_z are the longitudinal and vertical components of the rotor thrust components, L and D are the aerodynamic lift and drag, M_y is the pitch moment, and d_V, d_h, d_q are bounded disturbance terms.

The rotor thrust and its longitudinal and vertical components are modeled as

$$\begin{aligned} T_{MR,i} &= \rho D_{MR}^4 C_{T_{MR}} \Omega_i^2, \\ F_x(x, u_l, \lambda) &= \sum_{i=1}^{4} \sin(\lambda) T_{MR,i}, \\ F_z(x, u_l, \lambda) &= \sum_{i=1}^{4} \cos(\lambda) T_{MR,i}, \end{aligned} \tag{10}$$

where ρ is the air density, D_{MR} is the rotor diameter, and $C_{T_{MR}}$ is the rotor thrust coefficient.

The aerodynamic lift and drag are expressed as

$$\begin{aligned} L(V, \alpha) &= C_L(\alpha) QS, \\ D(V, \alpha) &= C_D(\alpha) QS, \end{aligned} \tag{11}$$

where $Q = \frac{1}{2}\rho V^2$ is the dynamic pressure, S is the wing reference area, and $C_L(\alpha)$ and $C_D(\alpha)$ are the lift and drag coefficients, respectively.

The pitch moment is written as

$$M_y = -\left(\sum_{i=1}^{4} l_{xi} \sin(\lambda) T_{MR,i} + C_m(\alpha) QS\right), \tag{12}$$

where l_{xi} is the longitudinal moment arm from the i-th rotor to the aircraft center of gravity, and $C_m(\alpha)QS$ denotes the aerodynamic pitching-moment term under the adopted simplified longitudinal model. The rotor thrust components, aerodynamic forces, and pitch moment are expressed in Equations 10-12.

The level-transition process is dominated by the coupling among rotor thrust vectoring, wing lift buildup, forward acceleration, and pitch stabilization. The longitudinal transition dynamics can be compactly written as

$$\dot{x} = f_l(x, u_l, \lambda, d_l), \tag{13}$$

where d_l represents bounded longitudinal disturbances and unmodeled dynamics, including aerodynamic uncertainty, wind disturbance, and actuator-related perturbations. In this formulation, the low-level controller generates u_l for altitude and attitude regulation, while the high-level scheduler determines the nacelle command λ. This separation allows different nacelle tilt schedules to be compared under the same stabilizing low-level control architecture.

Nacelle command evolution and transition corridor

Let T_s denote the decision interval of the high-level scheduler and $t_k = kT_s$. The sampled nacelle command sequence is denoted by

$$\Lambda_N = \{\lambda_0, \lambda_1, \ldots, \lambda_N\}. \tag{14}$$

For a forward level-transition task, the boundary condition is specified as

$$\lambda_0 = 0^\circ, \quad \lambda_N \approx \lambda_f = 90^\circ. \tag{15}$$

The sampled nacelle command sequence and boundary conditions are given by Equations 14, 15. The nacelle angle is a physical transition variable rather than an arbitrary independent command. Its evolution affects the distribution of rotor thrust between vertical support and forward propulsion. Therefore, an overly aggressive tilt command at low forward speed may reduce the vertical rotor-thrust component before sufficient wing lift has been established. Conversely, an overly conservative tilt command at high forward speed may delay transition completion and increase rotor-speed demand. These characteristics make the nacelle schedule a key high-level variable for level-transition performance.

During level-flight transition, the forward speed and nacelle angle must satisfy a feasible coupling relationship. This relationship is represented by a transition corridor in the (V_x, λ) plane:

$$\mathcal{C} = \left\{ (V_x, \lambda) \,\middle|\, \lambda_{min}(V_x) \le \lambda \le \lambda_{max}(V_x), V_{x,0} \le V_x \le V_{x,f} \right\}, \tag{16}$$

where $V_{x,0}$ and $V_{x,f}$ are the initial and target forward speeds, respectively. The functions $\lambda_{min}(V_x)$ and $\lambda_{max}(V_x)$ denote the lower and upper admissible nacelle-angle boundaries determined by aerodynamic lift capability, power margin, actuator limitations, and attitude-safety requirements.

The above formulation shows that the high-level nacelle scheduler must address two coupled requirements. First, the nacelle command should evolve as a physically realizable transition variable rather than as a sequence of independent absolute setpoints. Second, the generated command should remain compatible with the speed-dependent admissible corridor under the scheduled low-level flight-control architecture. Therefore, the problem considered in this paper is to design a feedback-capable high-level nacelle scheduler that maps measurable transition states to a smooth, actuator-compatible, and corridor-feasible nacelle command, without replacing the stabilizing low-level attitude and altitude controller.

Proposed corridor-constrained incremental TD3 tilt strategy

This section presents the proposed energy-aware smooth incremental tilt (ESIT) strategy for solving the high-level nacelle

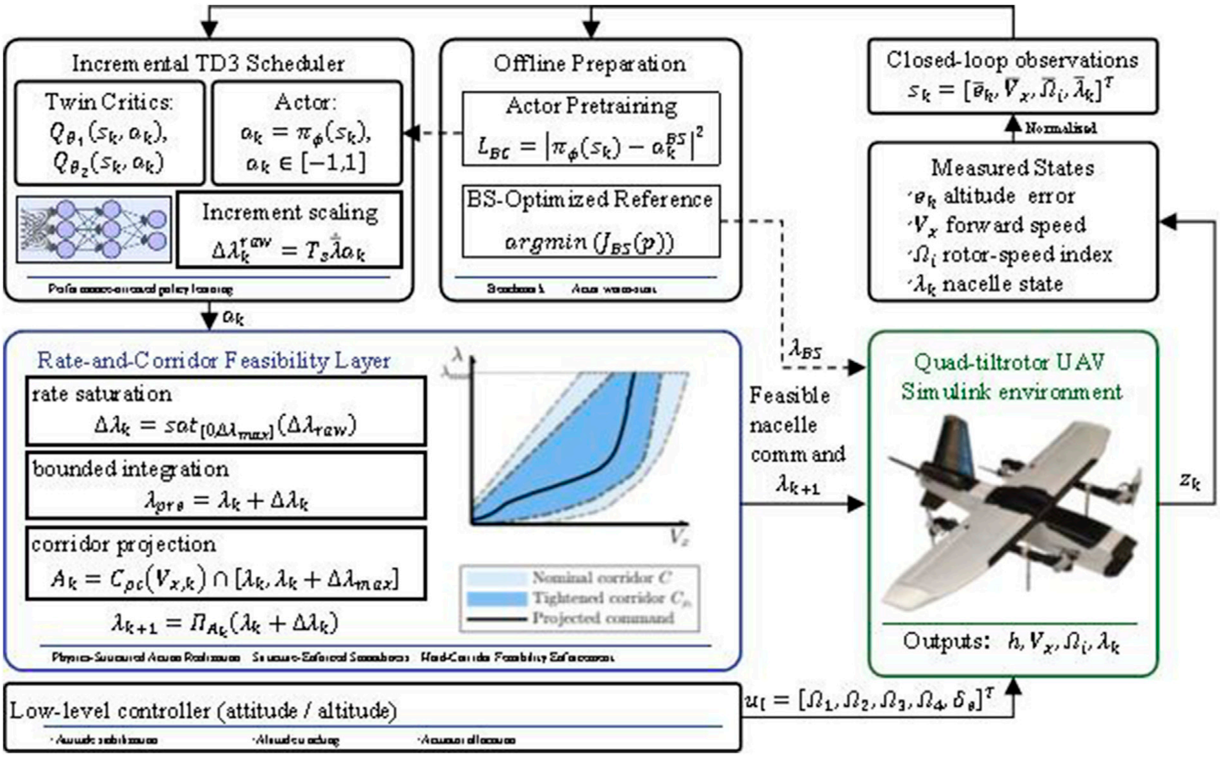

FIGURE 1
Overall framework of the proposed ESIT strategy.

scheduling problem formulated in Section *Problem formulation*. The method is developed under the same scheduled low-level flight-control architecture, so that the learned policy modifies only the nacelle-command evolution rather than the stabilizing attitude and altitude controller.

The central idea of ESIT is to formulate nacelle tilting as a constrained command-evolution process. Instead of directly learning the absolute nacelle angle at each decision instant, the actor learns a bounded nacelle-angle increment. The executable command is then obtained through bounded integration, rate saturation, and projection onto a tightened speed-dependent transition corridor. This action-generation mechanism embeds nacelle-command continuity and actuator-rate compatibility into the policy structure, while the projection layer enforces hard transition-corridor constraint.

The proposed framework is organized around five main elements: the hierarchical transition-control architecture, the scheduling state and incremental action parameterization, the rate-and-corridor feasibility layer, the multi-objective reward design, and the BS-guided reference schedule used for benchmarking and actor warm-starting. The overall framework is shown in Figure 1.

The feasibility layer is applied after the actor output and before the low-level controller. Therefore, the actor searches within the performance landscape induced by the closed-loop transition dynamics, while the final command remains compatible with the physical rate and corridor constraints. The corridor penalty in the reward is evaluated before projection, which encourages the actor to learn feasible candidate commands rather than relying entirely on the projection layer. The projection layer nevertheless remains the hard feasibility mechanism. The BS warm-start is used only to improve the initial policy quality and training efficiency. The final learned policy is evaluated as a closed-loop feedback scheduler and is not forced to track the BS reference. This distinction is essential because the objective of ESIT is not to reproduce an offline trajectory, but to learn a state-dependent nacelle-evolution policy under physical transition constraints.

Hierarchical transition-control architecture

The proposed scheduler is implemented under a hierarchical transition-control architecture. The high-level scheduler determines the evolution of the nacelle tilt angle, whereas the low-level controller tracks altitude and attitude references and allocates actuator commands to rotors and aerodynamic control surfaces. The scheduled low-level control architecture is kept unchanged for all compared tilt schedules, so that the performance differences can be attributed mainly to the high-level nacelle scheduling mechanism.

At the decision instant $t_k = kT_s$, the high-level scheduler receives measurable closed-loop transition information and generates the next nacelle command. The final command is then sent to the low-level controller, which produces the low-level actuator input $u_{l,k}$. The closed-loop information flow is summarized in Equation 17 as

$$\underbrace{z_k \xrightarrow{\mathcal{N}} s_k \xrightarrow{\pi_\phi} a_k \xrightarrow{\text{scale}} \Delta\lambda_k}_{\text{Incremental TD3 scheduler}} \xrightarrow{\text{rate/corridor}} \lambda_{k+1} \xrightarrow{\text{low-level controller}} u_{l,k} \xrightarrow{\text{plant}} x_{k+1}. \tag{17}$$

This hierarchical separation is important for the scope of this work. The RL policy is not required to stabilize the fast attitude and altitude dynamics directly. Instead, it adapts the slow nacelle transition process according to the measured closed-loop scheduling vector. The stabilizing controller provides local attitude and altitude regulation for all tested schedules, while the proposed scheduler shapes the transition behavior through the nacelle command.

Remark 1. The proposed framework evaluates nacelle tilt scheduling rather than low-level stabilization. Therefore, the same scheduled low-level control architecture, actuator model, and simulation environment are used for all compared schedules. This setting isolates the effect of action representation and feasibility enforcement at the high-level scheduling layer.

Scheduling state and incremental action parameterization

The high-level scheduler uses measurable closed-loop transition information to determine the nacelle command evolution. The physical scheduling vector is defined as

$$z_k = \left[\, e_{h,k} \;\; V_{x,k} \;\; \Omega_{\Sigma,k} \;\; \lambda_k \,\right]^T, \tag{18}$$

where $e_{h,k} = h_k - h_d$ is the altitude tracking error, $V_{x,k}$ is the forward speed, $\Omega_{\Sigma,k} = \sum_{i=1}^{4} \Omega_{i,k}$ is the total rotor-speed index, and λ_k is the current nacelle angle. The altitude error characterizes the vertical tracking margin, the forward speed represents the progress of wing-lift buildup and determines the corridor interval, the rotor-speed index provides an effort-related signal, and the nacelle state is included because the scheduling decision depends on the current nacelle configuration.

For neural-network training and policy evaluation, the physical scheduling vector is normalized as

$$s_k = \mathcal{N}(z_k) = \left[\, \bar{e}_{h,k} \;\; \bar{V}_{x,k} \;\; \bar{\Omega}_{\Sigma,k} \;\; \bar{\lambda}_k \,\right]^T \in \mathbb{R}^4, \tag{19}$$

where

$$\bar{e}_{h,k} = \frac{h_k - h_d}{e_{h,\max}}, \quad \bar{V}_{x,k} = \frac{V_{x,k}}{V_{x,f}}, \tag{20}$$

$$\bar{\Omega}_{\Sigma,k} = \frac{\sum_{i=1}^{4} \Omega_{i,k}}{\Omega_{\Sigma,\max}}, \quad \bar{\lambda}_k = \frac{\lambda_k}{\lambda_f}. \tag{21}$$

The physical and normalized scheduling vectors are defined in Equations 18-21. The normalization does not introduce a new physical scheduling state; it only maps the measurable transition variables into a numerically suitable input range for the actor and critic networks.

Rapid longitudinal acceleration is not used directly as a policy observation. This choice avoids amplifying measurement noise in the high-level scheduler. Instead, acceleration and jerk are evaluated in the reward function to guide the policy toward smooth transition behavior.

The action representation is the key difference between the proposed scheduler and an absolute-action RL scheduler. A conventional absolute-action policy directly maps the scheduling state to the complete nacelle command:

$$\lambda_{k+1}^{abs} = \lambda_f \frac{\pi_\phi^{abs}(s_k) + 1}{2}, \quad \pi_\phi^{abs}(s_k) \in [-1, 1]. \tag{22}$$

Although this representation is simple, it treats the nacelle command at each decision instant as an independent setpoint. Therefore, the physical evolution between adjacent commands is not encoded in the action itself. Exploration noise, critic approximation errors, or policy updates may consequently lead to abrupt inter-sample command variations.

In contrast, the proposed policy outputs a normalized incremental action

$$a_k = \pi_\phi(s_k), \quad a_k \in [-1, 1]. \tag{23}$$

The action a_k represents the intended change of the nacelle angle rather than the absolute nacelle command. It is then converted into a nacelle-angle increment and implemented through the discrete-time command-evolution law

$$\lambda_{k+1} = \lambda_k + \Delta\lambda_k. \tag{24}$$

Thus, the RL scheduler is converted from an absolute setpoint generator into a constrained nacelle-evolution generator. This distinction is fundamental: a smoothness penalty can discourage oscillatory actions, but it does not change the underlying action variable. By contrast, the incremental formulation directly modifies the command-generation structure itself, because the learned action is the nacelle-angle increment and the final command is produced by a constrained update.

The high-level scheduling policy is denoted by

$$\pi_\phi\colon s_k \mapsto a_k. \tag{25}$$

After action scaling, rate saturation, and corridor projection, this policy induces a physical nacelle command sequence

$$\Lambda_N = \{\lambda_0, \lambda_1, \ldots, \lambda_N\}. \tag{26}$$

The absolute-action and incremental-action parameterizations are given by Equations 22-26. A feasible scheduler should therefore generate a physically consistent nacelle evolution that respects both actuator-rate compatibility and the speed-dependent transition corridor, rather than a sequence of unrelated absolute setpoints.

Rate-and-corridor feasibility layer

The feasibility layer converts the normalized actor output into an executable nacelle command by enforcing two physical implementation requirements: nacelle-rate compatibility and speed-dependent corridor feasibility. This subsection describes the operational command-generation mechanism, whereas the post-training boundedness characterization later analyzes the resulting closed-loop scheduler under explicit assumptions.

The normalized incremental action is first scaled into a raw nacelle-angle increment:

$$\Delta\lambda_k^{raw} = T_s \bar{\dot{\lambda}} a_k, \tag{27}$$

where T_s is the high-level decision interval and $\bar{\dot{\lambda}}$ is the prescribed maximum nacelle tilt rate. For the forward level-transition task from the rotor-borne configuration to the wing-borne configuration, a monotonic rate-limited increment is adopted:

$$\Delta\lambda_{max} = T_s\bar{\dot{\lambda}}, \Delta\lambda_k = \text{sat}_{[0,\Delta\lambda_{max}]}\left(\Delta\lambda_k^{raw}\right). \tag{28}$$

Equivalently, the unidirectional saturation is expressed by Equation 29.

$$\Delta\lambda_k = \begin{cases} \Delta\lambda_{max}, & \Delta\lambda_k^{raw} > \Delta\lambda_{max}, \\ \Delta\lambda_k^{raw}, & 0 \le \Delta\lambda_k^{raw} \le \Delta\lambda_{max}, \\ 0, & \Delta\lambda_k^{raw} < 0. \end{cases} \tag{29}$$

The unidirectional saturation prevents reverse tilting and ensures that the command evolves consistently with the prescribed forward transition direction.

The pre-projection nacelle command is generated by bounded integration:

$$\lambda_{k+1}^{pre} = \lambda_k + \Delta\lambda_k. \tag{30}$$

This step embeds the inter-sample evolution of the nacelle command into the action-generation process.

For implementation, a tightened transition corridor is introduced as

$$\mathcal{C}_{\rho_c} = \left\{(V_x, \lambda) \,\middle|\, \lambda_{min}(V_x) + \rho_c \le \lambda \le \lambda_{max}(V_x) - \rho_c, V_{x,0} \le V_x \le V_{x,f}\right\}, \tag{31}$$

where $\rho_c > 0$ is a corridor margin. The margin separates the commanded trajectory from the nominal corridor boundary and provides room for sampling-induced speed variation, modeling mismatch, and low-level tracking transients.

At the current forward speed $V_{x,k}$, the tightened corridor induces the admissible nacelle-angle interval

$$\mathcal{I}_{\rho_c}(V_{x,k}) = [\lambda_{min}(V_{x,k}) + \rho_c, \lambda_{max}(V_{x,k}) - \rho_c]. \tag{32}$$

For the monotonic forward transition considered in this paper, the rate-compatible interval is

$$\mathcal{R}_k^+ = [\lambda_k, \lambda_k + \Delta\lambda_{max}]. \tag{33}$$

The final admissible update interval is the intersection

$$\mathcal{I}_a(V_{x,k}, \lambda_k) = \mathcal{I}_{\rho_c}(V_{x,k}) \cap \mathcal{R}_k^+. \tag{34}$$

When $\mathcal{I}_a(V_{x,k}, \lambda_k)$ is nonempty, the final command is obtained by scalar projection:

$$\lambda_{k+1} = \Pi_{\mathcal{I}_a(V_{x,k},\lambda_k)}\left(\lambda_{k+1}^{pre}\right), \tag{35}$$

where $\Pi_{\mathcal{I}_a}(\cdot)$ denotes the Euclidean projection onto the closed interval $\mathcal{I}_a$.

Equivalently, if

$$\mathcal{I}_a(V_{x,k}, \lambda_k) = \left[\underline{\lambda}_{a,k}, \bar{\lambda}_{a,k}\right], \tag{36}$$

with

$$\underline{\lambda}_{a,k} = \max\{\lambda_{min}(V_{x,k}) + \rho_c, \lambda_k\}, \tag{37}$$

$$\bar{\lambda}_{a,k} = \min\{\lambda_{max}(V_{x,k}) - \rho_c, \lambda_k + \Delta\lambda_{max}\}, \tag{38}$$

then the projection can be implemented as

$$\lambda_{k+1} = \min\left\{\max\left[\lambda_{k+1}^{pre}, \underline{\lambda}_{a,k}\right], \bar{\lambda}_{a,k}\right\}. \tag{39}$$

The tightened corridor, admissible intervals, and projected update are defined in Equations 31-34, 36-39. Thus, the actor output is not directly applied to the plant. Instead, it is first interpreted as an intended nacelle-angle increment and then processed by rate saturation and tightened-corridor projection. This design separates reward-level performance optimization from hard feasibility enforcement: the TD3 actor searches for a desirable transition behavior, while the feasibility layer enforces actuator-rate compatibility and practical corridor admissibility.

The transition corridor used in the feasibility layer should be interpreted as a scheduling corridor for high-level command generation, rather than as an exact representation of the full physical flight envelope. In practical tiltrotor transition, the admissible boundary may be affected by stall limits, power constraints, rotor--wing interference, attitude-safety requirements, actuator limits, and control-allocation capability. As a result, the raw transition envelope obtained from trimming, aerodynamic analysis, or offline simulation may contain piecewise segments, corner points, nonsmooth portions, or locally conservative regions.

The margin ρ_c is selected to compensate for one-step boundary movement and to avoid operating exactly on the nominal corridor boundary. Let

$$L_c = \max\{\text{Lip}(\lambda_{min}), \text{Lip}(\lambda_{max})\} \tag{40}$$

be an estimated gradient bound of the smoothed corridor boundaries over $V_x \in [V_{x,0}, V_{x,f}]$. In a tabulated implementation, L_c can be estimated from finite differences of the stored corridor table as

$$L_c = \max_i\left\{\frac{|\lambda_{min}(V_{x,i+1}) - \lambda_{min}(V_{x,i})|}{|V_{x,i+1} - V_{x,i}|}, \frac{|\lambda_{max}(V_{x,i+1}) - \lambda_{max}(V_{x,i})|}{|V_{x,i+1} - V_{x,i}|}\right\}. \tag{41}$$

If the forward-speed variation over one high-level sampling interval is bounded by

$$|V_{x,k+1} - V_{x,k}| \le \Delta V_{max}, \tag{42}$$

then a practical margin-selection rule is

$$L_c\Delta V_{max} \le \rho_c < \frac{1}{2}\min_{V_x \in [V_{x,0}, V_{x,f}]}[\lambda_{max}(V_x) - \lambda_{min}(V_x)]. \tag{43}$$

A practical corridor-margin selection rule is given by Equations 40-43. The lower bound reserves angular margin for one-step corridor-boundary movement caused by forward-speed variation, whereas the upper bound prevents the tightened corridor from becoming empty or excessively conservative. The one-step speed bound ΔV_{max} can be estimated either from an acceleration bound as $\Delta V_{max} = \bar{a}_{V_x}T_s$, or directly from the maximum one-step forward-speed variation observed in offline envelope-generation simulations.

If $\mathcal{I}_a(V_{x,k}, \lambda_k)$ becomes empty in implementation, no command can simultaneously satisfy the tightened-corridor interval and the monotonic rate interval at that decision instant. This event is recorded as a feasibility-margin event, indicating that the corridor smoothing, the margin ρ_c, the high-level sampling time T_s, the nacelle-rate limit $\Delta\lambda_{max}$, or the nominal corridor table should be redesigned. In the simulations of this paper, these design elements are selected so that the admissible update interval remains nonempty along the evaluated transition trajectories.

This feasibility layer should be distinguished from a continuous-time safety proof for detailed nacelle-servo dynamics. It specifies how the sampled high-level command is generated to be rate-compatible and corridor-aware. Detailed servo lag, overshoot, actuator faults, and online corridor adaptation require additional actuator-level analysis or experimental validation, which are beyond the scope of the present high-level scheduling design.

Multi-objective reward design

The level-transition scheduler must balance several competing objectives. Increasing the nacelle angle promotes forward acceleration and transition completion, but it reduces the vertical component of rotor thrust and may increase altitude deviation. Conversely, delaying nacelle rotation preserves vertical thrust margin but may increase transition duration and rotor-speed demand. In addition, abrupt nacelle variations may induce excessive longitudinal acceleration and jerk, which are undesirable for actuator compatibility and transition smoothness.

The reward function is designed as a normalized maximization-oriented surrogate of the scheduling trade-off considered in this work. It combines mission progress, actuator-effort regulation, and handling-qualities penalties, while leaving hard rate and corridor admissibility to the feasibility layer. The altitude-error, rotor-speed, acceleration, jerk, and increment terms correspond to the penalty terms in (Equation 56) with opposite signs, whereas the forward-speed term corresponds to the negative speed term in the cost. Additional shaping terms are introduced to improve learning efficiency and to reduce unnecessary reliance on the projection layer.

The feasibility layer enforces the actuator-rate and corridor constraints independently of the reward. Therefore, the reward function primarily shapes transition performance, while hard admissibility is imposed by the command-generation mechanism in Subsection *Rate-and-corridor feasibility layer*.

As shown in the hierarchical decomposition, the instantaneous reward is explicitly divided into three categories: mission goal reward, actuator effort penalty, and flight safety and quality penalty, which correspond to transition completion & velocity promotion, actuator operation smoothness constraint, and flight state safety maintenance, respectively. This yields the unified structural expression $r_k = \mathcal{R}_{\text{mission}} - \mathcal{P}_{\text{actuator}} - \mathcal{P}_{\text{safe,qual}}$, where $\mathcal{R}_{\text{mission}}$ denotes the mission-oriented reward item, $\mathcal{P}_{\text{actuator}}$ represents the penalty for consumption actuator operating effort and fluctuation, and $\mathcal{P}_{\text{safe,qual}}$ stands for the comprehensive penalty related to flight safety and dynamic quality.

At each decision instant, the categorized reward components are specifically defined in Equations 44-47.

$$r_k = \mathcal{R}_{\text{mission}} - \mathcal{P}_{\text{actuator}} - \mathcal{P}_{\text{safe,qual}} \tag{44}$$

$$\mathcal{R}_{\text{mission}} = c_v \bar{V}_{x,k} + c_m \bar{\lambda}_k \bar{V}_{x,k} + r_{T,k} \tag{45}$$

$$\mathcal{P}_{\text{actuator}} = c_\Omega \left(1 - \bar{\lambda}_k\right) \bar{\Omega}^2_{\Sigma,k} + c_{\Delta\lambda} \left(\frac{\Delta\lambda_k}{\Delta\lambda_{\max}}\right)^2 + c_c \Phi_c \left(V_{x,k}, \lambda^{\text{pre}}_{k+1}\right) \tag{46}$$

$$\mathcal{P}_{\text{safe,qual}} = c_h \bar{e}^2_{h,k} + c_a \rho_a \left(a_{x,k}\right) + c_j \rho_j \left(j_{x,k}\right) \tag{47}$$

in which $c_h, c_v, c_m, c_\Omega, c_a, c_j, c_{\Delta\lambda}$ and c_c are non-negative weighting coefficients. The mission goal reward encourages forward-speed buildup and accelerates the overall transition progress, where the mixed coupling term facilitates synchronous optimization of nacelle tilt angle and flight velocity. The actuator effort penalty regularizes the incremental nacelle command within the admissible rate interval (acting as a soft constraint rather than enforcing the hard rate limit, which is already handled by the feasibility layer), suppresses excessive rotor-speed demand, and guides the policy to generate candidate commands close to the feasible corridor region in advance. The safety and quality penalty mainly suppresses altitude tracking deviation, excessive longitudinal acceleration and severe jerk, so as to guarantee stable and comfortable transition flight.

The acceleration and jerk penalties are defined as

$$\rho_a \left(a_{x,k}\right) = \max\left(0, \left|a_{x,k}\right| - a_{th}\right)^2, \tag{48}$$

$$\rho_j \left(j_{x,k}\right) = \max\left(0, \left|j_{x,k}\right| - j_{th}\right)^2, \tag{49}$$

where a_{th} and j_{th} are prescribed smoothness thresholds. These terms suppress aggressive longitudinal transients during the lift-redistribution process.

The corridor-related penalty is evaluated using the pre-projection command:

$$\Phi_c \left(V_{x,k}, \lambda^{\text{pre}}_{k+1}\right) = \text{dist}^2 \left(\lambda^{\text{pre}}_{k+1}, \mathcal{I}_c \left(V_{x,k}\right)\right), \tag{50}$$

where $\text{dist}(\cdot, \mathcal{I}_c)$ is the Euclidean distance to the tightened corridor interval. This term encourages the actor to generate commands that are naturally close to the admissible region before projection, thereby reducing reliance on the feasibility layer during training.

The terminal completion reward $r_{T,k}$ is given by

$$r_{T,k} = \begin{cases} c_T, & V_{x,k} \geq V_{x,\text{tar}} \text{ and} \lambda_k \geq \lambda_f - \epsilon_\lambda, \\ 0, & \text{otherwise}, \end{cases} \tag{51}$$

where $V_{x,\text{tar}}$ is the target transition-completion speed, ϵ_λ is a small terminal nacelle-angle tolerance, and $c_T > 0$ is the positive terminal-completion bonus. The acceleration, jerk, corridor, and terminal reward terms are given by Equations 48-51.

Remark 2. The rotor-speed-related term in the reward function serves as an operation effort proxy. It reflects the variation tendency of rotor rotational speed demand during aerodynamic lift redistribution and transition scheduling, rather than establishing a high-fidelity calibrated model of electrical energy consumption, motor operating efficiency or battery power loss.

BS-guided reference schedule

A Beetle Swarm (BS) Optimization based reference schedule is used to provide a physically feasible benchmark and actor warm-start source. The BS optimization is performed offline over a finite-dimensional parameterization of the nacelle schedule. This reference is not imposed as a tracking command during final TD3 policy evaluation; it is used only to initialize the actor toward a feasible transition trend and to provide an interpretable non-learning baseline.

The nacelle schedule is parameterized by a normalized PCHIP interpolation curve. Let

$$p = [p_1, p_2, \ldots, p_M]^T \tag{52}$$

denote the free interpolation-node vector. Together with the boundary conditions

$$\lambda(0) = 0^\circ, \quad \lambda(T_{tilt}) = \lambda_f, \tag{53}$$

the parameter vector p determines a smooth baseline nacelle profile $\lambda_{BS}(t; p)$.

The offline optimization problem is written as

$$p^\star = \arg\min_{p \in \mathcal{P}} J_{BS}(p), \tag{54}$$

where $\mathcal{P}$ is the feasible parameter domain and J_{BS} (55) is a multi-objective transition-performance index including altitude error, speed buildup, rotor-speed-related effort, acceleration, jerk, and terminal completion terms. The BS schedule parameterization and offline optimization are formulated in Equations 52-54.

For a finite transition horizon $T = NT_s$, the transition-performance index is defined in Equations 55-57.

$$J_{BS} = \sum_{k=0}^{N-1} \ell(z_k, \Delta\lambda_k) + \ell_f(z_N) \tag{55}$$

$$\ell(z_k, \Delta\lambda_k) = w_h e_{h,k}^2 - w_v V_{x,k} + w_\Omega\left(1 - \frac{\lambda_k}{\lambda_f}\right)\Omega_{\Sigma,k}^2 + w_a \rho_a(a_{x,k}) + w_j \rho_j\left(j_{x,k}\right) + w_\Delta\left(\frac{\Delta\lambda_k}{\Delta\lambda_{max}}\right)^2 \tag{56}$$

$$\ell_f(z_N) = w_{vf}(V_{x,N} - V_{x,f})^2 + w_{\lambda f}(\lambda_N - \lambda_f)^2 + w_{hf} e_{h,N}^2 \tag{57}$$

where $\ell(\cdot)$ is the stage cost and $\ell_f(\cdot)$ is the terminal cost.

$w_h, w_v, w_\Omega, w_a, w_j, w_\Delta \geq 0$ are weighting coefficients. The terms in (56) penalize altitude error, rotor-speed-based effort, excessive longitudinal acceleration, jerk, and aggressive nacelle increments, while rewarding forward-speed buildup through the negative $V_{x,k}$ term. The functions $\rho_a(\cdot)$ and $\rho_j(\cdot)$ are nonnegative penalty functions used to suppress acceleration and jerk beyond desired smoothness levels. The acceleration $a_{x,k}$ and jerk $j_{x,k}$ are performance-evaluation variables rather than direct components of the scheduling state z_k. $w_{vf}, w_{\lambda f}, w_{hf} \geq 0$. This term encourages terminal speed buildup, completion of nacelle transition, and bounded terminal altitude error.

The criterion in (Equation 55) specifies the desired scheduling trade-off. The offline BS criterion follows the same scheduling trade-off as the reward design, but is expressed as a deterministic finite-horizon cost for reference generation. Compared to the reinforcement-learning reward, which acts as its normalized maximization-oriented surrogate with additional shaping terms for training efficiency and feasibility, this criterion eliminates these unnecessary implementation-related terms.

The optimized reference is used in two ways. First, it serves as a deterministic benchmark in the simulation section. Second, it provides supervised samples for actor warm-starting. Specifically, state--action pairs are extracted along the BS schedule:

$$\mathcal{D}_{BS} = \{(s_k^{BS}, a_k^{BS})\}_{k=0}^{N-1}, \tag{58}$$

where

$$a_k^{BS} = \frac{\lambda_{k+1}^{BS} - \lambda_k^{BS}}{\Delta\lambda_{max}}. \tag{59}$$

The actor can then be initialized by minimizing

$$\mathcal{L}_{pre} = \frac{1}{|\mathcal{D}_{BS}|}\sum_{(s,a)\in\mathcal{D}_{BS}} \| \pi_\phi(s) - a \|^2. \tag{60}$$

Supervised warm-start samples and the pretraining loss are given by Equations 58–60.

Remark 3. The BS reference provides an initial feasible tilt trend, but it does not constrain the final learned TD3 policy to reproduce the BS trajectory. After warm-starting, the actor is further optimized through closed-loop interaction with the environment, and the final policy is selected according to the fixed evaluation criterion described in Section *Simulation results and performance evaluation*.

TD3-based policy optimization

The proposed scheduler is trained using the twin delayed deep deterministic policy gradient algorithm. TD3 is adopted because it is suitable for continuous-action control and mitigates the overestimation bias of deterministic actor--critic learning through clipped double critics, delayed actor updates, and target policy smoothing.

The actor network is denoted by $\pi_\phi(s)$, and the two critic networks are denoted by $Q_{\theta_1}(s,a)$ and $Q_{\theta_2}(s,a)$. Their target networks are denoted by $\pi_{\phi'}$, $Q_{\theta_1'}$, and $Q_{\theta_2'}$. At each training step, a transition tuple

$$(s_k, a_k, r_k, s_{k+1}) \tag{61}$$

is stored in the replay buffer $\mathcal{B}$.

For a mini-batch sampled from $\mathcal{B}$, the target action is computed as

$$\tilde{a}_{k+1} = \mathrm{clip}\left(\pi_{\phi'}(s_{k+1}) + \epsilon, -1, 1\right), \quad \epsilon \sim \mathrm{clip}(\mathcal{N}(0, \sigma_{tar}^2), -c_\epsilon, c_\epsilon). \tag{62}$$

The target value is

$$y_k = r_k + \gamma \min_{i=1,2} Q_{\theta_i'}(s_{k+1}, \tilde{a}_{k+1}), \tag{63}$$

where γ is the discount factor. Each critic is updated by minimizing

$$\mathcal{L}(\theta_i) = \frac{1}{N_b}\sum_k [Q_{\theta_i}(s_k, a_k) - y_k]^2, \quad i = 1, 2, \tag{64}$$

where N_b is the mini-batch size.

The actor is updated less frequently according to the deterministic policy gradient:

$$\nabla_\phi J_\pi \approx \frac{1}{N_b}\sum_k \nabla_a Q_{\theta_1}(s_k, a)\Big|_{a=\pi_\phi(s_k)} \nabla_\phi \pi_\phi(s_k). \tag{65}$$

The target networks are updated by soft update:

$$\theta_i' \leftarrow \tau\theta_i + (1-\tau)\theta_i', \quad \phi' \leftarrow \tau\phi + (1-\tau)\phi', \tag{66}$$

where $\tau \in (0, 1)$ is the soft-update coefficient.

During environment interaction, the exploratory action is

$$a_k = \mathrm{clip}\left(\pi_\phi(s_k) + \eta_k, -1, 1\right), \quad \eta_k \sim \mathcal{N}\left(0, \sigma_{exp}^2\right). \tag{67}$$

The sampled action is then passed through the same scaling, rate-saturation, and corridor-projection mechanism described in Subsection *Rate-and-corridor feasibility layer*. Therefore, both training and evaluation use the same physical command-generation structure.

The training procedure is summarized in Algorithm 1.

Initialize actor π_ϕ, critics $Q_{\theta_1}, Q_{\theta_2}$, and target networks.
Initialize replay buffer $\mathcal{B}$.
Optionally warm-start π_ϕ using the BS reference dataset $\mathcal{D}_{BS}$.
For each episode **do**
  Reset the transition environment and obtain z_0; compute $s_0 = \mathcal{N}(z_0)$.
  For $k = 0, 1, \ldots, N-1$ **do**
    Generate exploratory incremental action $a_k = \mathrm{clip}(\pi_\phi(s_k) + \eta_k, -1, 1)$.
    Scale a_k to $\Delta\lambda_k^{raw}$ using (Equation 27).
    Apply rate saturation using (Equation 28).
    Generate λ_{k+1}^{pre} using (Equation 30).
    Project λ_{k+1}^{pre} onto $\mathcal{I}_a(V_{x,k}, \lambda_k)$ using (Equation 35).
    Apply λ_{k+1} to the low-level controller and simulate the plant.
    Obtain the next physical scheduling vector z_{k+1} and compute $s_{k+1} = \mathcal{N}(z_{k+1})$.
    Compute reward r_k using (Equation 47).
    Store (s_k, a_k, r_k, s_{k+1}) in $\mathcal{B}$.
    Update critics using (Equations 63, 64).
    Delayed update of the actor using (Equation 65).
    Soft-update target networks using (Equation 66).
  End For
End For

Algorithm 1. Corridor-Constrained Incremental TD3 for Nacelle Tilt Scheduling.

The preceding subsections define the command-generation and training procedure of the proposed scheduler. Before characterizing the post-training boundedness property, the scheduled low-level closed-loop representation used in the analysis is introduced below.

Low-level controller formulation

The role of the high-level scheduler is therefore to generate a feasible nacelle-angle evolution $\lambda(t)$ while the scheduled low-level controller family maintains altitude and attitude tracking over the admissible transition corridor.

For each admissible nacelle command, the low-level controller generates u_l to track the altitude and attitude references. Since the control effectiveness and aerodynamic characteristics vary significantly with the forward speed and nacelle angle, the low-level controller is represented as a scheduled controller family rather than a single fixed-parameter feedback law.

Let the scheduling variable be defined as

$$\sigma = [\, V_x \;\; \lambda \,]^T \in \mathcal{C}, \tag{68}$$

where $\mathcal{C}$ denotes the admissible transition corridor. In implementation, this admissible corridor is represented by a smoothed scheduling corridor constructed from the raw offline envelope, as detailed in Subsection *Rate-and-corridor feasibility layer*.

The low-level control law is written as

$$u_l = \kappa_{l,\sigma}(x, x_r, \lambda), \tag{69}$$

where x_r denotes the low-level reference signal, and the subscript σ indicates that the controller parameters are scheduled according to the current transition operating condition. This notation covers gain-scheduled, interpolated, or locally parameterized low-level controllers without requiring the high-level scheduler to redesign the stabilizing controller online.

Substituting (Equation 69) into (Equation 13) yields the scheduled closed-loop longitudinal dynamics by Equation 70.

$$\dot{x} = F_{l,\sigma}(x, \lambda, d_l), \tag{70}$$

where

$$F_{l,\sigma}(x, \lambda, d_l) = f_l\big(x, \kappa_{l,\sigma}(x, x_r, \lambda), \lambda, d_l\big). \tag{71}$$

With this scheduled closed-loop representation, the trained actor can now be treated as a bounded nonlinear scheduler, and the local boundedness of the resulting tracking-error dynamics can be characterized.

Post-training boundedness characterization

The preceding subsections define how the actor output is converted into an executable nacelle command through incremental action scaling, rate saturation, bounded integration, and corridor projection. This subsection provides a post-training boundedness characterization of the resulting closed-loop scheduler. The purpose is not to prove TD3 convergence, global optimality, or global nonlinear stability. Instead, the trained actor is treated as a bounded nonlinear scheduler, and the local boundedness of the scheduled low-level tracking error is characterized under explicit assumptions.

The analysis is conducted for the implemented closed-loop system after training. Suppose that the feasibility layer and the corridor-margin design described in Subsection Rate-and-corridor feasibility layer generate an implemented nacelle schedule satisfying

$$\sigma(t) = [\, V_x(t) \;\; \lambda(t) \,]^T \in \mathcal{C}, \quad t \in [0, T], \tag{72}$$

over the considered level-transition interval. This condition means that the high-level scheduling trajectory remains inside the admissible speed–nacelle-angle corridor used for the scheduled low-level control design. Although the transition task is evaluated over a finite horizon, the boundedness result below is stated in the standard UUB form. It applies when the implemented schedule remains in $\mathcal{C}$ during transition and then stays in an admissible terminal operating region after transition completion. Over the finite transition horizon, the same differential inequality directly provides an explicit local tracking-error bound. In the simulations, corridor membership is verified by evaluating the generated transition trajectories in the forward-speed–nacelle-angle plane.

The high-level scheduler determines the nacelle-angle evolution, whereas the scheduled low-level controller family stabilizes the attitude and altitude loops over the admissible transition corridor. Let the longitudinal tracking error be defined as

$$e = [\, e_h \;\; \dot{e}_h \;\; e_\theta \;\; e_q \,]^T, \tag{73}$$

where

$$e_h = h - h_d, \quad e_\theta = \theta - \theta_d, \quad e_q = q - q_d. \tag{74}$$

For the level-transition task considered in this paper, the desired altitude is constant and the nominal desired pitch motion is selected as

$$V_{hd} = 0, \quad \theta_d = 0, \quad q_d = 0. \tag{75}$$

From the scheduled closed-loop longitudinal dynamics (Equation 71), the corresponding tracking-error dynamics can be written locally as

$$\dot{e} = \Phi_{e,\sigma}(e, d_l), \tag{76}$$

where $\Phi_{e,\sigma}(\cdot)$ is the error-system vector field induced by the scheduled low-level closed-loop dynamics. The nominal zero-error manifold is assumed to be locally consistent with the scheduled low-level controller, namely

$$\Phi_{e,\sigma}(0,0) = 0, \quad \forall\sigma \in \mathcal{C}. \tag{77}$$

The tracking-error state, reference conditions, local error dynamics, and zero-error consistency condition are defined in Equations 73 -77. Any residual trim mismatch or unmodeled bias is included in the bounded disturbance term d_l. Around the nominal zero-error manifold, the scheduled tracking-error dynamics are represented by the local expansion

$$\dot{e} = A_\sigma e + B_{d,\sigma} d_l + \Delta_\sigma(e), \tag{78}$$

where A_σ denotes the Jacobian matrix of the scheduled low-level closed-loop error dynamics, $B_{d,\sigma}$ is the disturbance input matrix, and $\Delta_\sigma(e)$ collects higher-order nonlinear residuals.

In (Equation 78), the nacelle angle λ affects the tracking-error dynamics through the scheduling variable $\sigma = [V_x, \lambda]^T$ and the scheduled closed-loop vector field $F_{l,\sigma}$. Therefore, the nacelle-rate bound established by the incremental command-generation mechanism is not introduced as an independent low-level input. Instead, rate-compatible nacelle evolution supports the validity of the local scheduled model by preventing abrupt movement among operating conditions.

Assumption 1. *(Uniform local stability of the scheduled low-level controller).* For all $\sigma \in \mathcal{C}$, there exist symmetric positive definite matrices $P = P^T > 0$ and $Q = Q^T > 0$, independent of σ, such that

$$A_\sigma^T P + P A_\sigma \le -Q. \tag{79}$$

Assumption 2. *(Bounded disturbance and local residual).* The lumped longitudinal disturbance arising from aerodynamic uncertainty, external wind interference and unmodeled dynamics is bounded as

$$|d_l(t)| \le \bar{d}_l. \tag{80}$$

Herein, $\bar{d}_l$ denotes the predefined upper amplitude bound of the overall lumped system disturbance. Moreover, $B_{d,\sigma}$ is uniformly bounded over the admissible transition corridor:

$$\left|B_{d,\sigma}\right| \le \bar{B}_d, \quad \forall\sigma \in \mathcal{C}, \tag{81}$$

where $\bar{B}_d$ represents the maximum spectral norm bound of the disturbance input matrix varying with scheduling states. The nonlinear residual satisfies

$$|\Delta_\sigma(e)| \le c_\Delta |e|^2 \tag{82}$$

for all $\sigma \in \mathcal{C}$ and all e inside a compact neighborhood

$$\mathcal{D}_e = e\colon |e| \le r_e. \tag{83}$$

The common-Lyapunov and local bounding assumptions are stated in Equations 79 -83. In particular, c_Δ is the quadratic Lipschitz constant used to quantify the growth rate of high-order unmodeled nonlinear terms, and r_e stands for the maximum allowable radius of the local error convergence domain where the above nonlinear bounding condition holds.

$$\dot{e} = A_\sigma e + B_{d,\sigma} d_l + \Delta_\sigma(e), \tag{84}$$

where $\sigma = [V_x, \lambda]^T \in \mathcal{C}$. Suppose that the implemented schedule satisfies (Equation 72) during transition and remains in an admissible terminal operating region after completion. Suppose further that Assumptions 1 and 2 hold. Let

$$p_{min} = \lambda_{min}(P), \quad p_{max} = \lambda_{max}(P), \quad q_{min} = \lambda_{min}(Q). \tag{85}$$

The eigenvalue bounds used in the theorem are defined in Equation 85.

Remark 4. Assumption 1 is a uniform local stability condition for the scheduled low-level controller family over the considered transition corridor. It is not an assumption on the convergence or stability of the TD3 learning algorithm. The trained actor provides a bounded high-level scheduling command, while the scheduled low-level controller family is responsible for stabilizing the altitude and attitude tracking dynamics. The use of a common P is conservative, but it avoids additional terms associated with parameter-dependent Lyapunov functions or switching among local Lyapunov functions. This condition is used only to obtain a local boundedness characterization of the scheduled low-level error dynamics.

Theorem 1. *(Conditional uniform ultimate boundedness).* Consider the scheduled tracking-error dynamics.

Choose a local radius $r_0 \in (0, r_e]$ such that

$$2 p_{max} c_\Delta r_0 \le \frac{q_{min}}{4}. \tag{86}$$

Then, for any solution of (Equation 84) that remains in

$$\mathcal{D}_{r_0} = \{e\colon \; \| e \| \le r_0\}, \tag{87}$$

the tracking error is uniformly ultimately bounded. Specifically, define

$$\eta = \frac{q_{min}}{4 p_{max}}, \tag{88}$$

and

$$\Gamma_d = \frac{2 p_{max}^2 \bar{B}_d^2}{q_{min}} \bar{d}_l^2. \tag{89}$$

Then the Lyapunov function

$$V_e = e^T P e \tag{90}$$

satisfies

$$V_e(t) \le e^{-\eta t} V_e(0) + \frac{\Gamma_d}{\eta}\left(1 - e^{-\eta t}\right). \tag{91}$$

TABLE 1 Simulation setup and evaluation protocol.

Item	Setting
Simulation platform	MATLAB/Simulink
Solver	ode4
Step size	0.05 s
Final evaluation duration	34 s
Initial nacelle angle	0°
Terminal nacelle angle	90°
Target terminal forward speed	$V_{x,f} = 24.0$ m/s
Completion-speed threshold	$0.98V_{x,f} = 23.52$ m/s
Settling time for completion	0.5 s
Height command	Constant
Desired attitude	$\phi_d = \theta_d = \psi_d = 0$
Scheduled low-level control architecture	Identical for all strategies
Policy evaluation mode	Deterministic, without exploration noise
Agent selection rule	Best-return checkpoint
Monte Carlo runs	50
Randomized variables	Wind parameters, initial Euler angles, initial forward speed
Fairness condition	Same 50 scenarios reused for TD3 variants

Consequently,

$$\limsup_{t\to\infty} \| e(t)\| \le \sqrt{\frac{\Gamma_d}{\eta p_{min}}}. \tag{92}$$

Moreover, if there exists a constant $\bar{V}_e$ satisfying

$$V_e(0) \le \bar{V}_e, \quad \frac{\Gamma_d}{\eta} < \bar{V}_e < p_{min} r_0^2, \tag{93}$$

then the sublevel set

$$\Omega_{\bar{V}_e} = \{e: V_e(e) \le \bar{V}_e\} \tag{94}$$

is positively invariant, and the bound (Equation 92) holds for all $t \ge 0$.

Proof. Choose the Lyapunov function candidate

$$V_e = e^T P e. \tag{95}$$

Since $P = P^T > 0$, it satisfies

$$p_{min}\| e\|^2 \le V_e \le p_{max}\| e\|^2. \tag{96}$$

Taking the derivative of V_e along (Equation 84) gives

$$\dot{V}_e = e^T\left(A_\sigma^T P + PA_\sigma\right)e + 2e^T PB_{d,\sigma} d_l + 2e^T P\Delta_\sigma(e). \tag{97}$$

Using Assumption 1, one obtains

$$\dot{V}_e \le -q_{min}\| e\|^2 + 2p_{max}\bar{B}_d\| e \|\|d_l\| + 2p_{max}c_\Delta\| e\|^3. \tag{98}$$

For $e \in \mathcal{D}_{r_0}$, condition (Equation 86) implies

$$2p_{max}c_\Delta\| e\|^3 \le 2p_{max}c_\Delta r_0\| e\|^2 \le \frac{q_{min}}{4}\| e\|^2. \tag{99}$$

Substituting (Equation 99) into (Equation 98) yields

$$\dot{V}_e \le -\frac{3q_{min}}{4}\| e\|^2 + 2p_{max}\bar{B}_d\| e \|\|d_l\|. \tag{100}$$

By Young's inequality,

$$2p_{max}\bar{B}_d\| e \|\|d_l\| \le \frac{q_{min}}{2}\| e\|^2 + \frac{2p_{max}^2\bar{B}_d^2}{q_{min}}\| d_l\|^2. \tag{101}$$

Therefore,

$$\dot{V}_e \le -\frac{q_{min}}{4}\| e\|^2 + \frac{2p_{max}^2\bar{B}_d^2}{q_{min}}\| d_l\|^2. \tag{102}$$

Using $\| d_l(t)\| \le \bar{d}_l$, one obtains

$$\dot{V}_e \le -\frac{q_{min}}{4}\| e\|^2 + \Gamma_d. \tag{103}$$

From the upper quadratic bound in (Equation 96),

$$V_e \le p_{max}\| e\|^2, \tag{104}$$

$$\| e\|^2 \ge \frac{V_e}{p_{max}}. \tag{105}$$

The intermediate Lyapunov inequalities are given by Equation 100 -105. Hence,

$$\dot{V}_e \le -\frac{q_{min}}{4p_{max}}V_e + \Gamma_d = -\eta V_e + \Gamma_d. \tag{106}$$

Solving the scalar differential inequality (Equation 106) gives

$$V_e(t) \le e^{-\eta t}V_e(0) + \frac{\Gamma_d}{\eta}\left(1 - e^{-\eta t}\right), \tag{107}$$

which proves (Equation 91). Taking the limit superior on both sides gives

$$\limsup_{t\to\infty} V_e(t) \le \frac{\Gamma_d}{\eta}. \tag{108}$$

$$p_{min}\| e\|^2 \le V_e, \tag{109}$$

one obtains

$$\limsup_{t\to\infty} \| e(t)\| \le \sqrt{\frac{\Gamma_d}{\eta p_{min}}}, \tag{110}$$

which proves (92).

It remains to verify the sufficient local invariance condition. Suppose that (Equation 93) holds. On the boundary $V_e = \bar{V}_e$ (Equation 106), gives

$$\dot{V}_e \le -\eta\bar{V}_e + \Gamma_d < 0. \tag{111}$$

Therefore, the trajectory cannot leave the sublevel set $\Omega_{\bar{V}_e}$. Since $\bar{V}_e < p_{min}r_0^2$, every $e \in \Omega_{\bar{V}_e}$ satisfies

$$\| e\|^2 \le \frac{V_e}{p_{min}} \le \frac{\bar{V}_e}{p_{min}} < r_0^2. \tag{112}$$

The remaining proof steps are given by Equations 107 -112. Thus, $\Omega_{\bar{V}_e} \subset \mathcal{D}_{r_0}$, and the residual bound remains valid for all $t \ge 0$. This completes the proof.

10.3389/arc.2026.17000

TABLE 2 TD3 training parameters used in the implementation.

Parameter	Stage 1	Stage 2
Sampling time	0.1 s	0.1 s
Training horizon	31 s	31 s
Maximum episodes	150	325
Discount factor	0.98	0.98
Replay buffer length	2×10^5	Inherited
Mini-batch size	256	256
Warm-start steps	10,000	5,000
Critic learning rate	1×10^{-5}	1×10^{-6}
Actor learning rate	1×10^{-8}	2×10^{-7}
Target-network soft-update factor	5×10^{-4}	5×10^{-4}
Policy update frequency	4	2
Target update frequency	2	2
Exploration noise std.	0.008	0.005
Minimum exploration std.	0.002	0.001
Exploration decay rate	1×10^{-5}	1×10^{-5}
Target policy noise std.	0.010	0.008
Target policy noise bound	$[-0.02, 0.02]$	$[-0.015, 0.015]$

Remark 5. Theorem 1 establishes a local and conditional boundedness result for the scheduled closed-loop tracking-error system. The conclusion depends on three conditions: the implemented nacelle schedule remains inside the admissible transition corridor, the scheduled low-level controller family satisfies the common Lyapunov condition over the corridor, and the disturbance and nonlinear residual remain bounded in the local operating region. The result does not prove TD3 convergence, global stability, or global optimality of the learned scheduler.

Remark 6. The nacelle-rate bound does not appear as an independent input term in (Equation 84) because the compact closed-loop model uses λ as a scheduling variable and represents the low-level controller as a σ-scheduled controller family. The role of the incremental action parameterization is to keep the evolution of $\sigma = [V_x, \lambda]^T$ inside the admissible corridor and compatible with the nacelle actuator rate. This prevents abrupt movement among scheduled operating conditions and supports the validity of the uniform local stability condition in Assumption 1.

In summary, the boundedness characterization complements the feasibility layer. The feasibility layer defines how the actor output is converted into a corridor-aware and rate-compatible sampled command, whereas Theorem 1 characterizes the local tracking-error behavior once the implemented schedule evolves inside the admissible corridor. This result supports the mechanism of ESIT without assigning stabilizing responsibility to the TD3 actor itself.

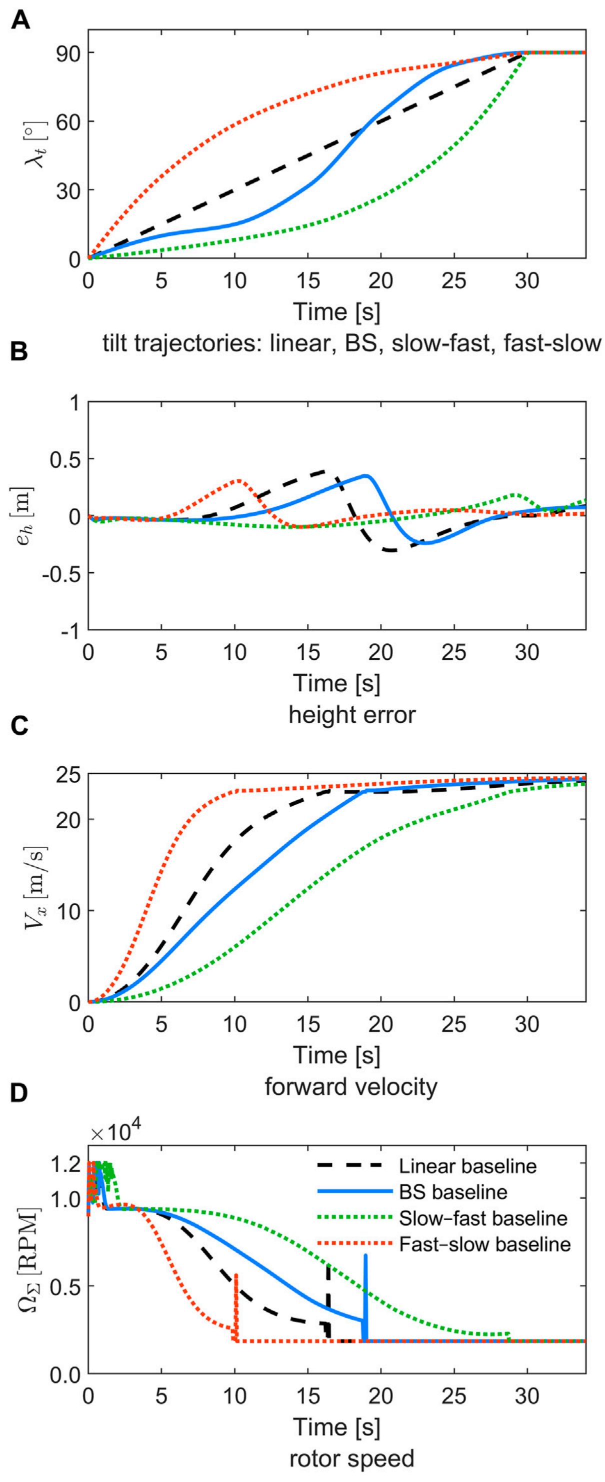

FIGURE 2
Baseline tilt schedules and closed-loop responses: **(A)** normalized nacelle-angle trajectories, **(B)** height-error responses, **(C)** forward-velocity responses, and **(D)** total rotor-speed responses.

Simulation results and performance evaluation

This section evaluates the proposed ESIT scheduler using a nonlinear quad-tiltrotor UAV model in MATLAB/Simulink.

TABLE 3 Low-level tracking boundedness under different tilt schedules.

Strategy	Max $|\theta|$ [deg]	RMS θ [deg]	Max $|e_h|$ [m]
Linear baseline	1.0981	0.3378	0.3908
Optimized baseline	0.8753	0.2633	0.3479
Slow–fast baseline	0.6977	0.3925	0.1793
Fast–slow baseline	0.6023	0.2456	0.3036
TD3	0.4526	0.1434	1.1480
TD3 + penalty	0.4526	0.1506	1.4060
TD3 + penalty + noise	0.4526	0.1422	1.0349
TD3-incremental	0.4526	0.1303	0.7738
TD3-incremental + noise	0.4526	0.1314	0.8198

Baseline schedules are first used to examine the sensitivity of transition response to nacelle timing, followed by TD3 action-parameterization ablation, corridor-feasibility verification, Monte Carlo smoothness evaluation, closed-loop performance comparison, and observation-noise assessment.

Simulation setup and evaluation protocol

The simulations were conducted in MATLAB/Simulink using the nonlinear quad-tiltrotor UAV model. All tilt-scheduling strategies were evaluated with the same scheduled low-level attitude/altitude control architecture, so that performance differences mainly reflect the high-level nacelle scheduling strategy.

The task is a forward level-flight transition from the rotor-borne configuration to the wing-borne configuration. The nacelle angle starts from 0° and reaches 90° at the terminal configuration. The height command is constant, and the desired attitude is set to zero during transition. The recorded variables include height error, forward velocity, nacelle angle, nacelle rate, longitudinal acceleration, longitudinal jerk, and rotor-speed-related quantities.

For deterministic policy evaluation, exploration noise was disabled. For the reinforcement-learning-based strategies, the saved agent with the highest training return was used as the evaluation policy. For TD3-incremental, the BS-generated reference was used only for actor warm-starting and did not constrain the final learned policy.

The transition corridor was constructed from offline envelope analysis in the forward-speed--nacelle-angle plane. One boundary was obtained from trim-based admissibility evaluation in Simulink, and the other was determined by the rotor-thrust limitation. The same corridor table was used for all safety-filtered policies and for corridor-feasibility verification.

The BS reference optimization and TD3 training used a 31 s horizon, whereas all final deterministic evaluations, baseline comparisons, TD3-variant comparisons, corridor evaluations, Monte Carlo tests, and observation-noise tests used a unified 34 s simulation duration. Unless otherwise stated, all reported performance indices correspond to this final evaluation horizon.

The main simulation settings are summarized in Table 1.

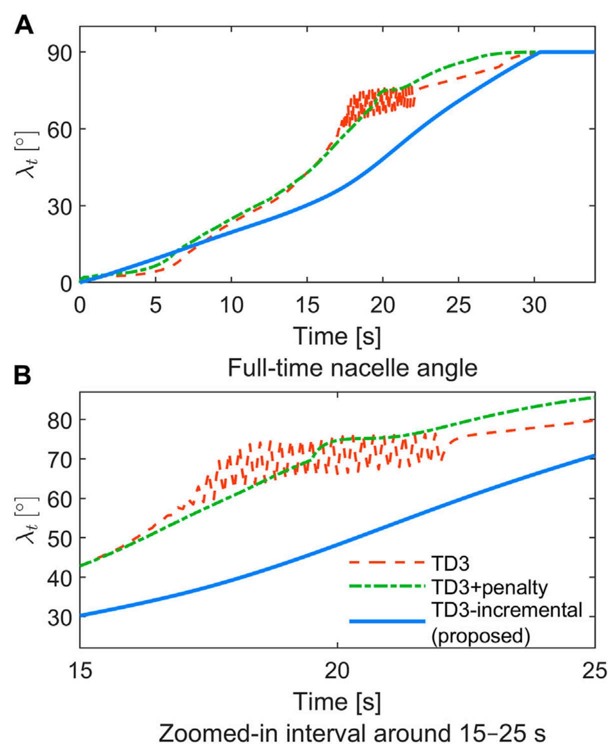

FIGURE 3
Nacelle-angle responses of TD3 variants: **(A)** full transition response and **(B)** zoomed-in view of the oscillation-prone interval.

TD3 training was conducted in two stages: a critic burn-in stage with an almost frozen actor, followed by actor fine-tuning with a small learning rate. The main training parameters are listed in Table 2.

For each simulation, the following performance indices were calculated:

$$\mathrm{RMSE}(e_h) = \sqrt{\frac{1}{T}\int_0^T e_h^2(t)\,dt} \tag{113}$$

$$\mathrm{RMS}(x) = \sqrt{\frac{1}{T}\int_0^T x^2(t)\,dt}, x \in \{a_x, j_x, \dot{\lambda}\} \tag{114}$$

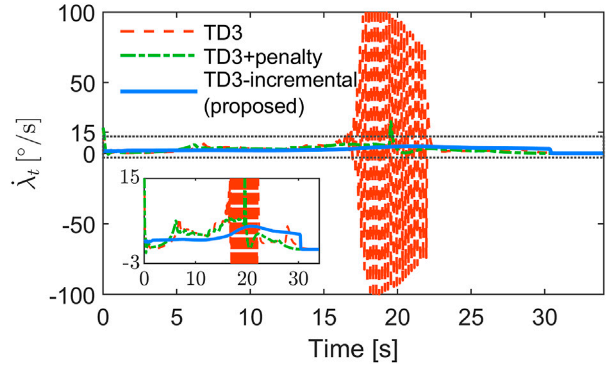

FIGURE 4
Nacelle-rate responses of TD3 variants. The proposed TD3-incremental strategy suppresses abrupt rate variations compared with the absolute-action policies.

$$J_{\Omega} = \int_{0}^{T} \left(\sum_{i=1}^{4} \Omega_i (t) \right)^2 dt \tag{115}$$

where e_h is the height tracking error, V_x is the forward velocity, a_x is the longitudinal acceleration, $j_x = \dot{a}_x$ is the longitudinal jerk, and J_{Ω} is the rotor-speed-based effort proxy. The performance indices used in the simulations are defined in Equations 113 -115.

In addition, $V_x(T)$, $\max|e_h(t)|$, and $\max|\dot{\lambda}(t)|$ were recorded.

The transition completion time is defined as the first time when $V_x(t)$ reaches $0.98V_{x,f} = 23.52\,\text{m/s}$ and remains above this threshold for at least 0.5 s. This avoids declaring completion from a transient velocity crossing.

Baseline tilt schedules and low-level tracking boundedness

Representative non-learning tilt schedules are first evaluated under the same scheduled low-level control architecture. This comparison serves two purposes: it shows the sensitivity of level-flight transition to nacelle timing, and it verifies that the shared low-level controller maintains bounded pitch and altitude responses under feasible nacelle schedules.

The evaluated schedules include a linear schedule, a slow–fast diagnostic schedule, a fast–slow diagnostic schedule, and a BS-optimized reference schedule. The predefined schedules were generated by normalized PCHIP interpolation with five free nodes between $\lambda(0) = 0^\circ$ and $\lambda(T_{tilt}) = 90^\circ$, where $T_{tilt} = 30\,\text{s}$. The command was held at the terminal value after T_{tilt}. The BS-optimized reference was generated using the 31 s optimization horizon, and all schedules were evaluated under the unified 34 s deterministic horizon.

The normalized free-node vectors are as followed.

The linear baseline used

$$p_{lin} = [0.1667, 0.3333, 0.5000, 0.6667, 0.8333].$$

The slow--fast diagnostic schedule used

$$p_{SF} = [0.04, 0.09, 0.16, 0.30, 0.55].$$

The fast--slow diagnostic schedule used

$$p_{FS} = [0.40, 0.65, 0.80, 0.90, 0.95].$$

The BS-optimized baseline used

$$p_{opt} = [0.110, 0.166, 0.351, 0.709, 0.940].$$

Figure 2 shows the baseline tilt schedules and the corresponding closed-loop responses. The fast--slow schedule advances most of the nacelle rotation to the early transition stage, whereas the slow--fast schedule delays the main tilting process. These different timing patterns produce different altitude-error, forward-velocity, and total-rotor-speed responses. The result confirms that the nacelle schedule is an active high-level variable that affects altitude regulation, speed buildup, and rotor-effort redistribution during level-flight transition.

The low-level pitch and height tracking bounds under different schedules are summarized in Table 3. The maximum absolute pitch angle remains below 1.10° for all evaluated strategies. For the TD3-based strategies, the maximum absolute pitch angle is below 0.46°, and the RMS pitch angle remains small. These results indicate that the shared low-level controller remains well behaved during the evaluated transitions.

Table 3 is not intended to rank all schedules by tracking accuracy or to prove global nonlinear stability. It only verifies that the subsequent TD3 comparisons are not caused by low-level instability. The repeated maximum pitch value observed for several TD3-based cases mainly reflects a common low-level transient under the shared stabilization loop; the corresponding height-error bound,

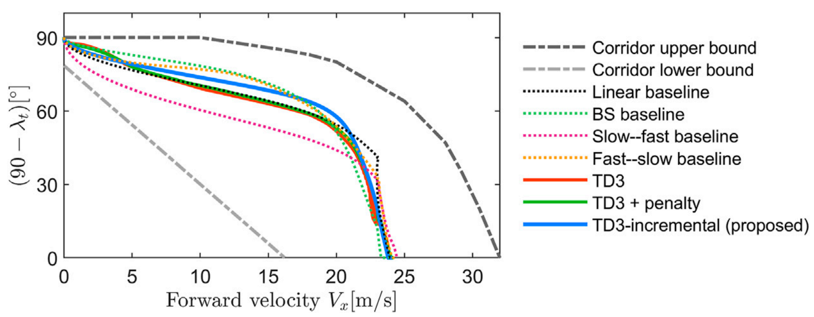

FIGURE 5
Transition trajectories in the forward-speed–converted-nacelle-angle plane, where the plotted angle is $90^\circ - \lambda_t$. All compared strategies remain inside the prescribed transition corridor during level-flight transition.

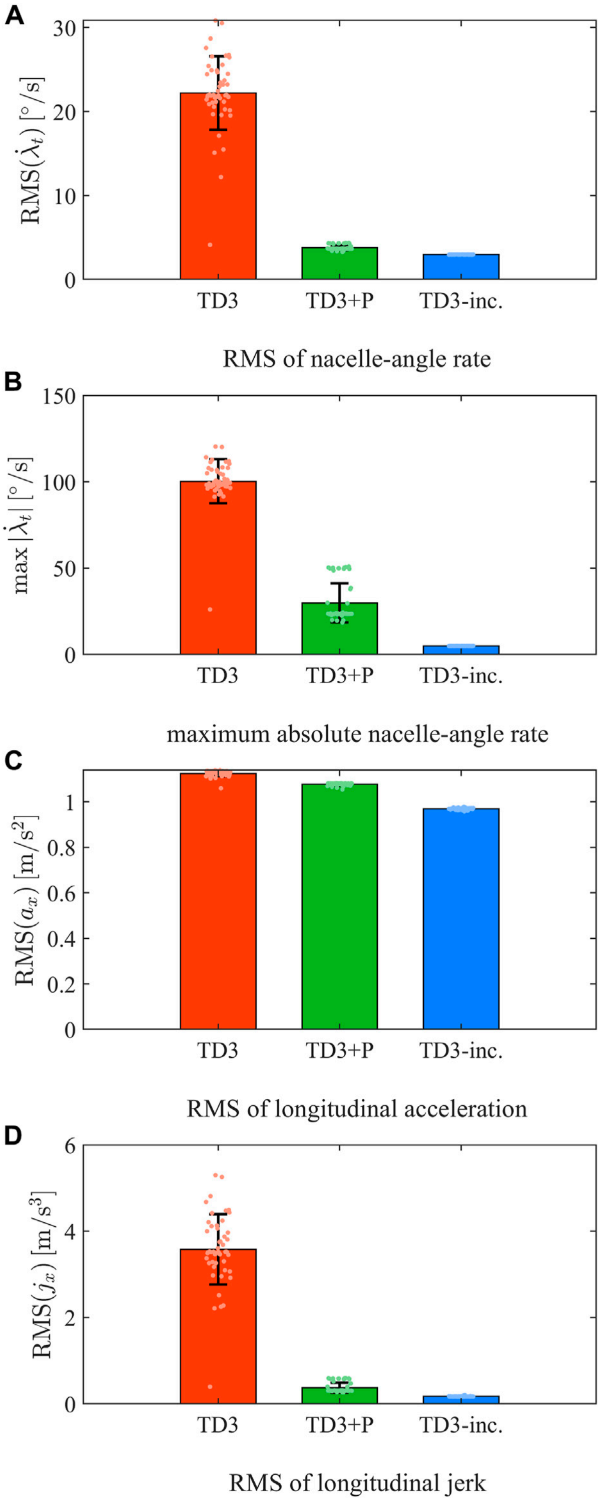

FIGURE 6
Monte Carlo smoothness statistics of TD3 variants under 50 randomized scenarios. The same random scenarios are reused for all three TD3-based strategies. **(A)** RMS nacelle-angle rate; **(B)** maximum nacelle-angle rate; **(C)** RMS longitudinal acceleration; (D) RMS longitudinal jerk.

nacelle-rate demand, jerk response, and corridor trajectory still differ across high-level schedulers.

Action-parameterization ablation of TD3 tilt scheduling

This subsection isolates the effect of action parameterization on TD3-based nacelle scheduling. Three policies are compared under the same observation variables, scheduled low-level control architecture, reward components, and deterministic evaluation protocol: absolute-action TD3, penalty-based absolute-action TD3, and the proposed incremental-action TD3.

The absolute-action policy directly outputs the complete nacelle angle, so adjacent commands are not structurally coupled. The penalty-based policy retains this absolute-action representation but adds an oscillation penalty. In contrast, the proposed policy outputs a nacelle-angle increment before command filtering. Therefore, the comparison separates action-level reparameterization from reward-level smoothness shaping.

Figure 3 compares the nacelle-angle responses of the three TD3 variants. All policies reach the terminal nacelle configuration, but their transient command evolution is different. The absolute-action TD3 policy exhibits local oscillations during the middle-to-late transition stage, especially when the aircraft shifts from rotor-dominant support to wing-lift-supported flight. TD3 + penalty attenuates this fluctuation, but local irregularities remain. By contrast, TD3-incremental produces a smoother and nearly monotonic nacelle-angle trajectory.

The nacelle-rate responses in Figure 4 further highlight the difference among the action representations. Absolute-action TD3 generates large and rapidly alternating rate peaks. TD3 + penalty reduces the peak level but does not remove local rate fluctuations. TD3-incremental keeps the nacelle-rate response within a narrow range throughout the transition, indicating better compatibility with rate-limited nacelle actuation.

These results indicate that the oscillatory behavior of absolute-action TD3 is mainly associated with the mismatch between independent absolute-angle commands and the physical continuity requirement of nacelle rotation. Reward-level oscillation penalties can mitigate this behavior, but the improvement remains tuning-dependent. The incremental action representation modifies the command structure itself and therefore provides a more direct mechanism for smooth nacelle evolution.

Corridor feasibility evaluation

Command smoothness alone does not guarantee transition feasibility. A nacelle trajectory may be smooth but still leave the admissible speed--nacelle-angle corridor. Conversely, a corridor-feasible trajectory may still impose undesirable nacelle-rate demand. Therefore, the generated schedules are examined in the forward-speed--nacelle-angle plane.

Figure 5 shows the transition trajectories of the predefined baselines, the BS-optimized baseline, and the TD3-based

TABLE 4 Monte Carlo smoothness and transition-performance statistics.

Metric	TD3	TD3 + penalty	TD3-incremental
$\mathrm{RMS}(\dot{\lambda})$ [deg/s]	22.2302 ± 4.3734	3.8009 ± 0.2985	**2.9804 ± 0.0152**
$\max\lvert\dot{\lambda}\rvert$ [deg/s]	100.2860 ± 12.7518	29.8180 ± 11.3357	**4.9256 ± 0.0445**
$\mathrm{RMS}(a_x)$ [m/s 2]	1.1246 ± 0.0130	1.0769 ± 0.0053	**0.9696 ± 0.0037**
$\mathrm{RMS}(j_x)$ [m/s 3]	3.5789 ± 0.8168	0.3750 ± 0.1170	**0.1720 ± 0.0071**
$\mathrm{RMSE}(e_h)$ [m]	0.4993 ± 0.0526	0.5728 ± 0.0467	**0.4094 ± 0.0252**
$V_x(T)$ [m/s]	24.1495 ± 0.0273	24.1819 ± 0.0118	**24.0909 ± 0.0207**
J_Ω	$1.0595 \times 10^9 \pm 7.6869 \times 10^7$	$\mathbf{9.8821 \times 10^8 \pm 6.5735 \times 10^7}$	$1.0460 \times 10^9 \pm 5.3674 \times 10^7$

Boldface denotes the best-performing value among the three TD3 variants for each metric.

schedulers. To match the plotting convention of the generated corridor map, the vertical axis is displayed as the converted nacelle angle $90^\circ - \lambda_t$. This conversion is used only for visualization and does not change the feasibility criterion defined in (Equation 16). All compared strategies remain inside the prescribed transition corridor during the evaluated level-flight transition.

The corridor trajectories show that feasibility and smoothness are distinct properties. The absolute-action TD3 trajectory satisfies the corridor constraint but still contains local irregularities, consistent with the oscillatory nacelle response in Figure 3. The penalty-based variant reduces these irregularities, whereas TD3-incremental produces the smoothest feasible path among the TD3-based schedulers. Thus, the smoothness advantage of the proposed scheduler is achieved within the same admissible speed--nacelle-angle region, rather than by relaxing or leaving the prescribed transition corridor.

Monte Carlo smoothness evaluation

Monte Carlo simulations are conducted to examine the consistency of the smoothness improvement under randomized transition conditions. Each TD3-based strategy was evaluated over 50 scenarios, and the same random scenarios were reused for absolute-action TD3, TD3 + penalty, and TD3-incremental. The randomized quantities include wind amplitude, frequency, phase, initial Euler angles, and initial forward speed.

The evaluation focuses on nacelle-rate demand, longitudinal acceleration, longitudinal jerk, altitude response, terminal speed, and rotor-speed-based effort. Figure 6 shows the smoothness statistics, and Table 4 summarizes the numerical results. Absolute-action TD3 exhibits the largest nacelle-rate demand and dispersion. TD3 + penalty reduces the mean oscillation level, but its maximum rate and jerk remain more sensitive to scenario variations. TD3-incremental yields the smallest nacelle-rate demand and the most consistent smoothness metrics among the three TD3 variants.

Compared with absolute-action TD3, TD3-incremental reduces $\mathrm{RMS}(\dot{\lambda})$ from 22.2302 deg/s to 2.9804 deg/s, and reduces $\max\lvert\dot{\lambda}\rvert$ from 100.2860 deg/s to 4.9256 deg/s. Compared with TD3 + penalty, it further reduces the maximum nacelle-rate demand from 29.8180 deg/s to 4.9256 deg/s. The standard deviations of both nacelle-rate metrics are also smaller for TD3-incremental, indicating that the smoother command behavior is less sensitive to the tested random variations.

The same trend appears in the longitudinal smoothness metrics. TD3-incremental reduces $\mathrm{RMS}(j_x)$ from 3.5789 m/s^3 for absolute-action TD3 and 0.3750 m/s^3 for TD3 + penalty to 0.1720 m/s^3. Meanwhile, all three TD3 variants maintain terminal forward speeds close to the target value. Thus, the smoother response of TD3-incremental is not obtained by failing to complete the transition, but by changing the nacelle command evolution mechanism.

These Monte Carlo results do not constitute a global robustness guarantee. They show that, under the tested randomized scenarios, the proposed incremental action structure provides more consistent nacelle-rate and jerk reduction than absolute-action TD3 and reward-penalized absolute-action TD3.

Closed-loop transition performance and observation-noise evaluation

This subsection compares the overall closed-loop transition performance of representative strategies and further evaluates the sensitivity of TD3-incremental to observation noise.

Closed-loop transition performance

The compared strategies include the linear baseline, the BS-optimized baseline, TD3 + penalty, and TD3-incremental. They represent a simple predefined schedule, an offline optimized schedule, a reward-smoothed absolute-action RL policy, and the proposed incremental-action RL scheduler, respectively.

Figure 7 shows the height-error, forward-velocity, longitudinal-acceleration, and total-rotor-speed responses. All compared strategies complete the level-flight transition without low-level instability. The linear baseline provides a simple reference response, whereas the BS-optimized baseline achieves a strong nominal profile through offline optimization. TD3 + penalty improves the smoothness of absolute-action RL, but still inherits the limitations of absolute command generation. TD3-incremental produces a smoother nacelle-driven transition while maintaining bounded height response and terminal forward-speed buildup.

The quantitative indices are summarized in Table 5. TD3-incremental reduces the RMS jerk from 0.3150 m/s^3 for TD3 +

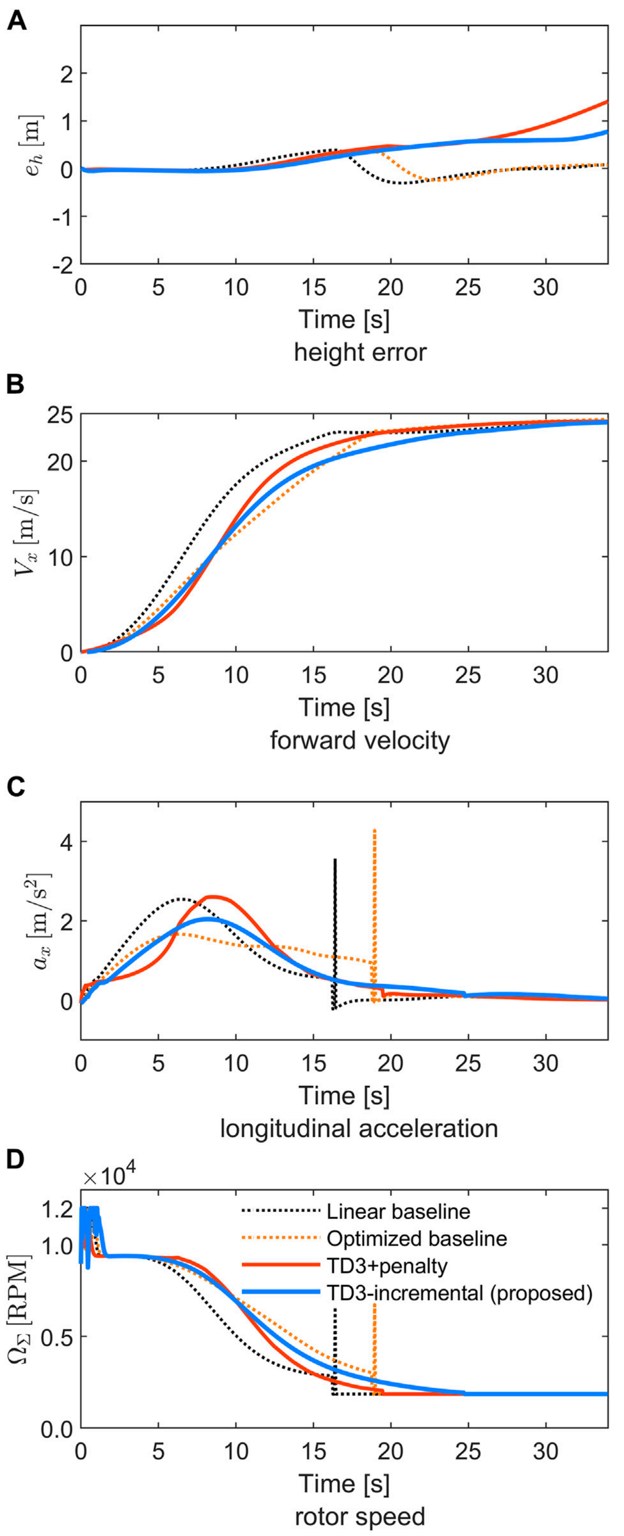

FIGURE 7
Closed-loop transition responses of representative strategies: **(A)** height error, **(B)** forward velocity, **(C)** longitudinal acceleration, and **(D)** total rotor speed.

penalty to 0.1774 m/s^3, and lowers the maximum height error from 1.4060 m to 0.7738 m. Its terminal forward speed remains close to the target value, indicating that the transition is completed. The main trade-off is a longer completion time, which reflects the more conservative nacelle evolution induced by the incremental action structure.

These results show that TD3-incremental should not be interpreted as minimizing every scalar metric independently. Its advantage lies in producing a rate-compatible and low-jerk nacelle command while maintaining bounded altitude response, corridor feasibility, terminal speed buildup, and comparable rotor-speed-based effort. The BS-optimized baseline remains a high-quality offline reference, but it is not a feedback policy. In contrast, TD3-incremental generates the nacelle increment from measured transition states and therefore provides a feedback-capable implementation of corridor-aware nacelle scheduling.

Observation-noise evaluation

Observation noise is injected into the policy inputs to examine whether perturbed high-level measurements induce undesirable nacelle-command oscillations. The noise is added to altitude error, forward speed, and total rotor-speed index before normalization, while the low-level control architecture and evaluation protocol are kept unchanged. This test evaluates input-side sensitivity of the learned scheduler rather than observer or sensor-fusion design.

Figure 8 compares the noise-free and noisy responses of TD3-incremental. The observation noise is modeled as zero-mean Gaussian perturbations with variances of 0.01 for e_h, 2 for V_x, and 2.0×10^6 for Ω_Σ. The generated nacelle command remains smooth under noisy inputs. The height-error response remains bounded, the forward-speed buildup is preserved, and the total rotor-speed response remains comparable to the noise-free case.

The bounded response under observation noise is consistent with the incremental command-generation mechanism. Since the actor output is interpreted as a bounded increment and then processed by rate saturation and corridor projection, measurement perturbations cannot directly produce arbitrarily large changes in the executable nacelle command. The low-level tracking statistics in Table 3 further show that the maximum height error increases only moderately under observation noise, while the RMS pitch angle remains almost unchanged.

Discussion

The simulation results support the main mechanism of the proposed ESIT framework. The baseline comparison confirms that nacelle timing is a dominant high-level factor affecting altitude response, speed buildup, rotor-speed demand, and longitudinal smoothness. The TD3 ablation shows that the incremental action representation provides a structural

TABLE 5 Closed-loop transition performance.

Strategy	RMSE e_h [m]	Max $\|e_h\|$ [m]	$V_x(T)$ [m/s]	RMS a_x [m/s 2]	RMS j_x [m/s 3]	J_Ω	Completion time [s]
Linear baseline	0.1682	0.3908	24.165	1.1168	2.0975	8.8493×10^8	26.90
Optimized baseline	0.1363	0.3479	24.344	0.9637	2.4089	1.0869×10^9	22.35
TD3 + penalty	0.5425	1.4060	24.173	1.0729	0.3150	1.0234×10^9	23.10
TD3-incremental	0.3881	0.7738	24.052	0.9685	0.1774	1.0827×10^9	28.35

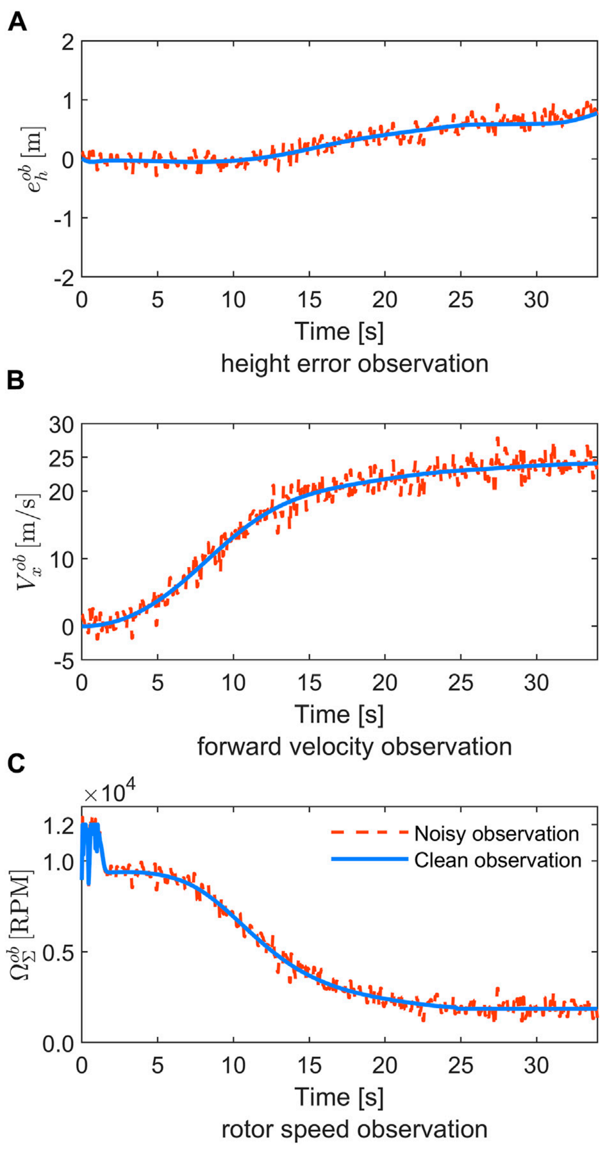

FIGURE 8
Observation-noise evaluation of the TD3-incremental scheduler: noise-free and noisy responses. **(A)** height error; **(B)** forward velocity; **(C)** total rotor speed. Noise-free and noisy responses are compared in each panel.

mechanism for smooth command generation beyond reward-level oscillation penalties. The corridor evaluation verifies that this smoothness improvement is achieved inside the prescribed speed--nacelle-angle corridor. The Monte Carlo and observation-noise evaluations further show that the smoothness advantage remains consistent under the tested randomized scenarios and perturbed policy inputs.

These findings are consistent with the command-generation mechanism developed in Section *Proposed corridor-constrained incremental TD3 tilt strategy*. By learning a bounded nacelle-angle increment instead of an absolute nacelle command, the scheduler limits inter-sample command variation before the command reaches the low-level controller. The feasibility layer then enforces rate compatibility and sampled-command corridor admissibility. As a result, the learned scheduler improves nacelle-command smoothness without requiring changes to the scheduled low-level attitude/altitude control architecture.

The results should nevertheless be interpreted within the simulation scope of this paper. The rotor-speed-based index J_Ω is used as an effort proxy rather than as a calibrated propulsion-power or battery-energy model. The transition corridor is constructed from the considered vehicle model and offline envelope analysis, so the feasibility conclusion depends on the fidelity of the corridor table. In addition, the Monte Carlo and observation-noise tests evaluate representative variations in wind parameters, initial conditions, and policy inputs, but they do not constitute a global robustness guarantee.

Conclusion

This paper investigated high-level nacelle tilt scheduling for level-flight transition of a quad-tiltrotor UAV. The main conclusion is that the smoothness and feasibility of reinforcement-learning-based transition scheduling depend not only on reward design, but also on the physical structure of the action representation. By treating the nacelle angle as a rate-limited transition variable rather than as an independently generated absolute setpoint, the proposed ESIT framework reformulated the TD3 action as a bounded nacelle-angle increment. The executable command was then generated through bounded integration, rate saturation, and projection onto a tightened speed-dependent transition corridor.

The proposed framework provides a mechanism for combining feedback learning with physical transition constraints. The incremental action representation embeds nacelle-command continuity and actuator-rate compatibility into the command-generation process, while the feasibility layer enforces sampled-command corridor admissibility. A BS-optimized reference schedule was used to provide a physically feasible warm-start and a non-learning benchmark, rather than as a globally optimal trajectory. After training, the learned actor was interpreted as a bounded nonlinear scheduler, and the post-training analysis characterized corridor feasibility and conditional closed-loop boundedness under bounded disturbances and local scheduled low-level stability assumptions. These results clarify the implemented scheduler's feasibility and boundedness properties without relying on a convergence or global-optimality claim for the TD3 learning process.

The simulation results support the proposed mechanism. Compared with absolute-action TD3 and penalty-based absolute-action TD3, the TD3-incremental scheduler generated smoother nacelle trajectories, reduced nacelle-rate demand and longitudinal jerk, and maintained bounded altitude response, terminal forward-speed buildup, and comparable rotor-speed-based effort. The corridor evaluation verified that the smoothness improvement was obtained inside the admissible speed--nacelle-angle region. The Monte Carlo and observation-noise tests further showed that the generated nacelle commands remained smooth under the tested randomized transition scenarios and perturbed policy inputs. These results indicate that physically structured action parameterization can improve the implementation compatibility of learning-based nacelle scheduling.

The proposed framework is useful for transition tasks in which a nominal corridor or reference schedule is available, but the final nacelle command must remain feedback-capable, smooth, rate-compatible, and corridor-feasible. Its current scope is nevertheless limited to monotonic forward level-flight transition under the considered vehicle model, corridor table, and scheduled low-level control architecture. In addition, the energy-related objective is represented by a rotor-speed-based effort proxy rather than by a calibrated propulsion-power or battery-energy model. Future work will therefore focus on incorporating calibrated propulsion-energy models, updating the transition corridor using higher-fidelity aerodynamic or experimental data, and extending the incremental action-generation framework to reconversion, three-dimensional transition, and coupled tilt-scheduling/control-allocation problems.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

RY contributed to the conceptualization, methodology, investigation, formal analysis, and writing of the original draft. JY, TF, and ZD contributed to investigation, validation, data curation, and visualization. CD provided overall supervision, project administration, resources, and guidance for the research group. All authors contributed to the article and approved the submitted version.

Funding

The author(s) declared that financial support was received for this work and/or its publication. This work was supported by an internal research project of the authors' research group, under the project title "Frontier Exploratory Research on Tilt-Rotor Unmanned Aerial Vehicles".

Acknowledgements

The authors would like to acknowledge the support provided by their research group throughout the development of this work.

Conflict of interest

The author(s) declared that this work was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The author(s) declared that generative AI was not used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

References

1. Xi Y, Wang Y, Sun S, Lu Y. A review of research on fight control and transition strategies fortilt-rotor unmanned aerial vehicles. In: *2025 2nd International Conference on Electrical Technology Andautomation Engineering (ETAE) (IEEE)* (2025). p. 382–90.

2. Su J, Huang H, Zhang H, Wang Y, Wang FY. Evtol performance analysis: a review from control perspectives. *IEEE Trans Intell Vehicles* (2024) 9:4877–89. doi: 10.1109/tiv.2024.3387405

3. Hernandez-Garcia RG, Rodriguez-Cortes H. Transition flight control of a cyclic tiltrotor uav based on the gain-scheduling strategy. In: *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE (2015). p. 951–6.

4. Wu Z, Li C, Cao Y. Numerical simulation of rotor-wing transient interaction for a tiltrotor in the transition mode. *Mathematics* (2019) 7:116. doi: 10.3390/math7020116

5. Liu N, Wang Y, Zhao J, Cheng X, He X, Wu K. A new inclining flight mode and oblique takeoff technique for a tiltrotor aircraft. *Aerospace Sci Technology* (2023) 139: 108370. doi: 10.1016/j.ast.2023.108370

6. Yuksek B, Vuruskan A, Ozdemir U, Yukselen M, Inalhan G. Transition flight modeling of a fixed-wing vtol uav. *J Intell and Robotic Syst* (2016) 84:83–105. doi: 10.1007/s10846-015-0325-9

7. Zhao G, Cui Z, Zhao Q, Chen X, Li P. Investigations on trimming strategy and unsteady aerodynamic characteristics of tiltrotor in conversion procedure. *Aerospace* (2024) 11:632. doi: 10.3390/aerospace11080632

8. Flores G, Lugo I, Lozano R. 6-dof hovering controller design of the quad tiltrotor aircraft: simulations and experiments. In: *53rd IEEE Conference on Decision and Control (IEEE)* (2014). p. 6123–8.

9. Ma T, Wang X, Fu J, Hao S, Xue P. Three-dimensional flight envelope for V/stol aircraft with multiple flight modes. *Aerospace* (2022) 9:691. doi: 10.3390/aerospace9110691

10. Appleton W. A, Filippone Bojdo N. "Interaction effects on the conversion corridor of tiltrotor aircraft." The Aeronautical Journal. 125.1294 (2021):2065–2086. doi: 10.1017/aer.2021.33

11. Fan Y, Wang X, Hu Z, Zhang K. Nonlinear modeling and transition corridor calculation of a tiltrotor without cyclic pitch. *MATEC Web Conf., MATEC Web of Conferences* (Les Ulis, France: EDP Sciences) (2022) 355, 01022. doi: 10.1051/matecconf/202235501022

12. Zhang Z, Tian J, Chen H, Sheng S. Optimal transition trajectory search for tilt-rotor aircraft based on ant colony algorithm. In: *2023 2nd International Conference on Artificial Intelligence and Computer Information Technology (AICIT) (IEEE)* (2023). p. 1–8.

13. Xue L, Zhang J, Zhang N, Chen Q. *Optimization Method for Tilting Transition Trajectory of Tiltrotor International Conference on Guidance, Navigation and Control*. Springer (2024). p. 465–73.

14. Zhao H, Wang B, Shen Y, Zhang Y, Li N, Gao Z. Development of multimode flight transition strategy for tilt-rotor vtol uavs. *Drones* (2023) 7:580. doi: 10.3390/drones7090580

15. Eskandarpour A, Mehrandezh M, Gupta K, Ramirez-Serrano A, Soltanshah M. A constrained robust switching mpc structure for tilt-rotor uav trajectory tracking problem. *Nonlinear Dyn* (2023) 111:17247–75. doi: 10.1007/s11071-023-08787-y

16. Kang N, Lu L, Whidborne J. Energy optimization strategies for automatic tiltrotor electric vertical takeoff and landing aircraft. *J Guidance, Control Dyn* (2025) 48: 1196–200. doi: 10.2514/1.g008466

17. Bohn E, Coates EM, Moe S, Johansen TA. Deep reinforcement learning attitude control of fixed-wing uavs using proximal policy optimization. In: *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE (2019). p. 523–33.

18. Romero A, Aljalbout E, Song Y, Scaramuzza D. Actor-critic model predictive control: differentiable optimization meets reinforcement learning for agile flight. *IEEE Trans Robotics* (2025) 42:673–92. doi: 10.1109/tro.2025.3644945

19. Eliker K, Le TN, Bessaad N. Reinforcement learning-based optimal pitch-angle generation for energy-efficient tilt-rotor quadrotors. *Control Eng Pract* (2026) 171: 106853. doi: 10.1016/j.conengprac.2026.106853

20. Fujimoto S, Hoof H, Meger D. Addressing function approximation error in actor-critic methods. In: *International Conference on Machine Learning (PMLR)* (2018). p. 1587–96.

21. Hausknecht M, Stone P. Deep reinforcement learning in parameterized action space. *arXiv Preprint arXiv:1511.04143* (2015).

22. Kanervisto A, Scheller C, Hautamaki V. Action space shaping in deep reinforcement learning. In: *2020 IEEE Conference on Games (CoG) (IEEE)* (2020). p. 479–86.

23. Zhu J, Wu F, Zhao J. An overview of the action space for deep reinforcement learning. In: *Proceedings of the 2021 4th International Conference on Algorithms, Computing and Artificial Intelligence* (2021). p. 1–10.

24. Eßer J, Margolis GB, Urbann O, Kerner S, Agrawal P. Action space design in reinforcement learning for robot motor skills. In: *8th Annual Conference on Robot Learning* (2024).

25. Chisari E, Liniger A, Rupenyan A, Van Gool L, Lygeros J. Learning from simulation, racing in reality. In: *2021 IEEE International Conference on Robotics and Automation (ICRA) (IEEE)* (2021). p. 8046–52.

26. Chen Z, He X, Wang YJ, Liao Q, Ze Y, Li Z, et al. Learning smooth humanoid locomotion through Lipschitz-constrained policies. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE (2025). p. 4743–50.

27. Wabersich KP, Zeilinger MN. Safe exploration of nonlinear dynamical systems: a predictive safety filter for reinforcement learning. *arXiv preprint arXiv:1812.05506*. (2018)

28. Zanon M, Gros S. Safe reinforcement learning using robust MPC. *IEEE Transactions on Automatic Control* (2020) 66.8: 3638-3652.

29. Pham TH, De Magistris G, Tachibana R. Optlayer-practical constrained optimization for deep reinforcement learning in the real world. In: *2018 IEEE International Conference on Robotics and Automation (ICRA) (IEEE)* (2018). p. 6236–43.

30. Lin S, Wang H, Chen Z, Kan Z. Projection-based fast and safe policy optimization for reinforcement learning. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE (2024). p. 7426–32.

31. Papachristos C, Alexis K, Tzes A. Model predictive hovering-translation control of an unmanned tri-tiltrotor. In: *2013 IEEE International Conference on Robotics and Automation* (EEE) (2013). p. 5425–32.

32. Papachristos C, Alexis K, Tzes A. Hybrid model predictive flight mode conversion control of unmanned quad-tiltrotors. In: *2013 European Control Conference (ECC)*. IEEE (2013). p. 1793–8.