Adaptive Robust Servo LQR Control for Aircraft under a Wide Range of Icing Conditions

Amin Rabiei Beheshti[1], Yoonsoo Kim[1], Rho Shin Myong[1,*]

[1]*Graduate School of Mechanical and Aerospace Engineering and ACTRC, Gyeongsang National University, Jinju, Gyeongnam 52828, Republic of Korea*

**Corresponding author: myong@gnu.ac.kr (R. S. Myong)*

Abstract

Icing can significantly affect aircraft stability and control derivatives, compromising flight safety and performance. Aircraft may encounter a wide range of icing conditions, from moderate to severe. To address this, autopilots must handle aircraft recovery in icing conditions. In this study, we propose an Adaptive Robust Servo Linear Quadratic Regulator (ARS-LQR) approach to simultaneously control the aircraft's altitude and forward velocity under a wide range of icing conditions. The ARS-LQR approach, which obtains linear models by linearizing around the local trajectory, may seem classical, but it introduces a practical element by incorporating an exponential multiplier into the cost function. This element ensures local stability with a relatively large stability margin, even in severe icing conditions, making it particularly valuable for aircraft control engineers dealing with certification requirements. This theoretical support distinguishes it from many recent techniques, including AI-based ones, lacking such backing. A comprehensive comparison study validates the theoretical advantage of ARS-LQR over Adaptive Neural Nonlinear Dynamic Inversion (AN-NDI) in tracking altitude and forward velocity references in severe icing conditions. Existing control techniques do not address the full spectrum of aircraft icing conditions. Computer simulations emphasizing longitudinal motion provide an efficient and stable solution for severe icing challenges.

Keywords: Aircraft icing, Adaptive control, Linear Quadratic Regulator, Local stability, Neural Dynamic Inversion, Severe icing

1 Introduction

1.1 Motivation

Since 1948, aircraft icing has been a subject of significant attention in aviation research, as supported by the data in Tables 17, 18, and 36 from [1]. Notably, the Air Safety Foundation's analysis of over 3000 aircraft accidents revealed that icing played a critical role in approximately 13 % of these incidents [2, 3]. Furthermore, NASA's comprehensive examination of statistical studies, encompassing 16 years from 1988 to 2003, highlighted trends in incidents associated with icing [4], focusing on loss of control in icing conditions [5]. As a result, the number of studies on aircraft icing has increased significantly over the last 50 years [6]. Ice buildup on aircraft surfaces significantly endangers flight safety, impacting aerodynamic performance and dynamic response. Ice on wings and tails can reduce lift coefficients, causing abrupt downward pitches and increasing drag, necessitating more engine power [7–12].

Additionally, aircraft can encounter a broad spectrum of icing conditions in real-world scenarios [2]. Lampton and Valasek proposed a method for assessing icing effects on airplane dynamics, stability, and control [11]. Aerospace engineers aim to enhance flight safety in icing conditions by developing ice prevention and removal techniques [13]. While effective, ice Protection Systems (IPS) have limitations, such as incomplete coverage and high energy demands, making total ice elimination impractical [14]. Consequently, developing autonomous flight control systems to prevent control loss is paramount [15].

Extensive research, particularly under NASA's Tailplane Icing Program (TIP), focused on the effects of ice accumulation on the wings and tails of aircraft, notably the DH-6 Twin Otter, to ensure flight safety and control [16–21]. These findings highlighted the critical influence of ice accretion on aerodynamic properties and aircraft stability. In [11, 22], a six-degree-of-freedom tool was used to assess the dynamical response, stability, and control specifications of a Cessna 208 Caravan aircraft during various flight phases in icing conditions. Deiler [23, 24] introduced an incremental model extension (Δ model) to analyze the longitudinal and lateral dynamical response of an aircraft with degraded stability, and control derivatives in icing conditions. New approaches presented in [25, 26] to estimating the flight envelope of a helicopter in icing conditions based on zonotopic Kalman filtering and reachability analysis, followed by Linear Quadratic Regulator (LQR) and H_∞ control to stabilize the attitude dynamics.

Despite the tremendous development of aerodynamic modeling for various aircraft in icing conditions, there has been little research on aircraft control in icing conditions. In [27], a robust control technique combined with reinforcement learning (RL) was introduced to recover aircraft from high angular rate upsets. The H_2 state feedback control was used in [28] as a control law in icing conditions. Hossain et al. [29] modified the pitch attitude hold autopilot of a DH-6 Twin Otter, utilizing a standard Proportional-Integral-Derivative (PID) controller to account for icing effects. However, this controller is not adaptive, and its coefficients remain constant, which is not ideal for icing conditions,

requiring a specified fixed linear state-space model. Hartley [30] applied predictive control to a large Airbus aircraft in icing conditions in combination with the Extended Kalman Filter to estimate the lift and pitching moment derivatives. Bugajski et al. Morton et al. [31] presented results of global stability for a nonlinear aircraft pitch-axis model using Nonlinear Dynamic Inversion (NDI). However, NDI has some drawbacks. The stability analysis for NDI-based control is often restricted, and NDI can be highly sensitive to inversion errors and modeling uncertainties.

The weaknesses mentioned above of NDI have prompted researchers to explore a combination of robust and adaptive controls, which has proven successful in aerospace applications. For example, Wei et al. [32] employed adaptive neural network-based nonlinear dynamic inversion (AN-NDI) to control aircraft flight in icing conditions. However, AN-NDI, an online adaptive gain scheduling tool, necessitates numerous time-consuming simulations and may not guarantee stability [33]. Padhi et al. [33] combined NDI control with neural adaptive control to enable high-performance aircraft to follow a desired path while rejecting uncertainties. Bradtke et al. suggest that combining LQR with solid performance characteristics, stability margin, and adaptive control techniques can enhance robustness [34, 35].

Furthermore, although LQR is originally designed for linear time-invariant (LTI) systems, it can be extended to handle specific nonlinear systems by linearizing around the local trajectory [36, 37]. This approach provides a local approximation of the optimal control solution for the nonlinear system, allowing LQR to be applicable in a broader range of control scenarios [38, 39]. Robust LQR is a critical flight control approach due to its ability to balance aircraft performance and stability under uncertainties and disturbances [34, 40]. Notably, it demonstrates promise in mitigating the negative impact of icing conditions on aircraft dynamics. Elkhatem et al. [41] designed an LQR-PI controller, a fusion of LQR and proportional-integral control, achieving zero steady-state error. However, these approaches are based on linear time-invariant (LTI) systems, with state-space matrices assumed to be constant. Further works on adaptive robust techniques in the presence of uncertainties are explored in [42, 43].

1.2 Related Work

The proposed ARS-LQR method in this paper might resemble classical control techniques. Yet, it introduces a distinctive and valuable practical component. Theoretically, it integrates an exponential multiplier into the cost function, ensuring local stability even in severe icing conditions with substantial margins (6 dB gain margin and 60-degree phase margin) [34]. This theoretical innovation holds significant interest for aircraft control engineers concerned with stringent certification demands, as recent adaptive or AI-based techniques lack comparable theoretical support.

Compared to other robust control methods, such as Nonlinear Model Predictive Control (NMPC) [44], L1 Adaptive Control [45], Optimal Deep Learning Control, and Model Reference Adaptive Control (MRAC) [34], Adaptive Robust Servo LQR (ARS-LQR) offers significant advantages in terms of computational complexity and ease of implementation. The ARS-LQR method involves fewer computational calculations, typically on the order of $O(n^3)$, thanks to its use

of simple linear models. In contrast, NMPC, with its nonlinear nature and the need to solve an optimization problem at each time step, can be computationally expensive, often requiring $O(n^4)$ computations. NMPC is also sensitive to uncertainties and disturbances, which can lead to suboptimal performance or instability. Moreover, NMPC requires complex and time-consuming parameter tuning, including the prediction horizon, cost function weighting factors, and uncertainty estimation. L_1 Adaptive Control has a more complex control formulation than ARS-LQR and involves estimating states, uncertainties, and adaptation law. It requires the rough tuning of multiple parameters, including the L_1 adaptation gain, estimation gain, weights used in cost functions, and filter cutoff frequency. Optimal Deep Learning Control relies on training a neural network, making it computationally expensive and challenging in terms of robustness and stability. The computational complexity of Optimal Deep Learning Control depends on the neural network architecture, typically scaling with $O(n^m)$, where m denotes the number of neurons per layer. Optimal Deep Learning Control requires rough tuning of hyperparameters, such as the number of layers and neurons, learning rate, and regularization parameters. On the other hand, ARS-LQR offers proven local stability, simplicity in terms of linear models, and a reduced number of tuning parameters. It primarily requires tuning the weighting factors Q and R in the cost function. The limitations of ARS-LQR include its applicability, its mainly suited to slow-varying parameter dynamics systems, and the requirement for disturbance estimation in nonlinear equations. The choice of control method depends on the specific application and involves trade-offs between computational complexity, model complexity, stability, robustness, and tuning requirements. Detailed comparisons of these control methods are summarized in Table 1.

This paper is organized as follows: Section 2 focuses on aircraft flight dynamics and the applied icing model. A detailed discussion of the formulation of the control issue follows this. The AN-NDI technique is elaborated on first in Section 3.1, followed by an explanation of the ARS-LQR method in Section 3.2. Section 4 presents the simulation results, and Section 5 summarizes the research's findings. The present study employs the workflow described in Fig.1 to construct the proposed autopilot for aircraft operating in icing conditions.

Table 1 Comparison of Control Methods with ARS-LQR

Items	ARS-LQR	NMPC	L1 Adaptive Control	Optimal Deep Learning Control	AN-NDI
Properties	• Less computational calculations ($O(n^3)$) due to the use of linear models • Simple linear models. • Proven local stability with a robustness margin (significantly important for aircraft certification) • Fewer tuning parameters (only weighting factors Q and R)	• Computational complexity ($O(n^4)$) due to its nonlinear nature and the need to solve an optimization problem at each time step • Rough estimates of uncertainties • More sensitive to uncertainties and disturbances, dependent on the accuracy of the model, the chosen optimization algorithm, and the effectiveness of the tuning process. • Several parameters to adjust (often time-consuming)	• Need for estimation of states and uncertainties, and adaptation law, adding to the complexity • Rough (No systematic) tuning of multiple parameters, such as L_1 adaptation gain k_a, estimation gain k_e, weights W used in cost functions, and filter cutoff frequency ω_c	• Computational complexity heavily depending on neural network architecture ($O(n^m)$, where m is the number of neurons per layer) • Being formulated as either model-free (offline) or model-based, with different requirements • Little evidence of robustness and stability • Need for tuning of hyperparameters	• Utilizing RBFs to estimate uncertainties to take advantage of global approximator • Making use of Lyapunov function in control design for stability guarantee
Limitations	• Applicable to slowly varying parameter dynamics systems • Requires disturbance estimation in nonlinear equations	• Real-time implementation: processing power and time constraints, especially for large-scale systems • Limited robustness due to model uncertainties and disturbances • Challenging: fast-varying dynamics systems.	• Robustness may come into question in some scenarios • Performance limitations align with hardware constraints	• Real-time implementation problems due to robustness and stability • Computational capacity source problems for training large amounts of data	• Control matrix must have a matrix inverse • Uncertainties should be bounded
Assumptions	• Assuming a linear model at each time step • Bounded norm of A and B matrices • Changes of system parameters within a range of 200–300 percent from their nominal values • Full State Availability • Controllability of A and B, and observability of A and C	• Known System Dynamics • State Observability • Continuity and Differentiability of cost function and model	• Boundedness of parameter estimation • Boundedness of partial derivatives	• Availability of sufficient training data • In model-based algorithms: requires the availability of exact and known system dynamics	• Assuming there is an inverse matrix of the control matrix • Assuming equations can be written in affine form • Ensuring boundedness of parameter estimation • Dealing with the challenge of requiring numerous tuning parameters

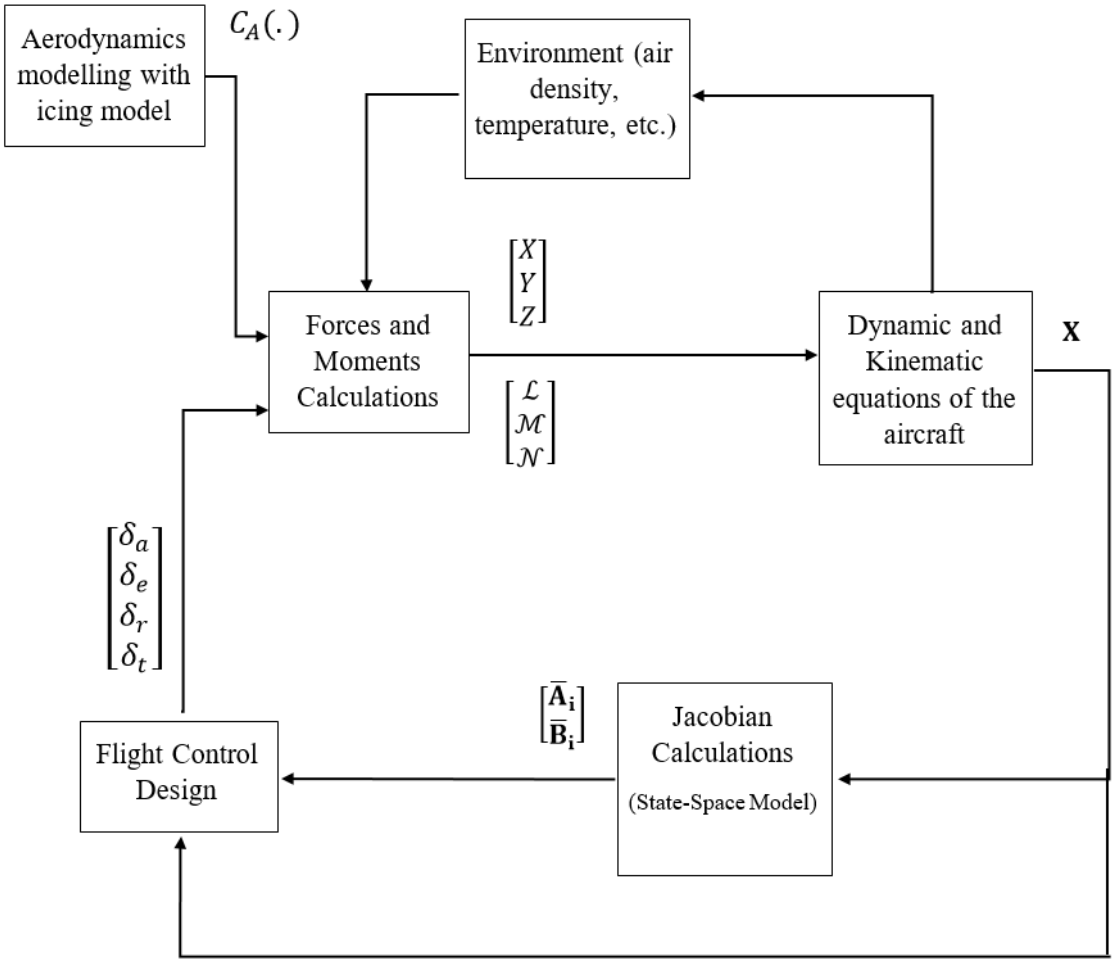


Fig. 1 Control system architecture for aircraft in icing condition

2 Aircraft Flight Dynamics

2.1 Fully nonlinear equations of motion

Assuming the aircraft is a point mass concentrated at its center of gravity, its performance and the fully nonlinear model of an aircraft is derived from Newton's second law in an axis system (body-axis) fixed to a center of gravity. According to Figure 1.10 of reference [46], x_b is along the body towards the nose of the aircraft, y_b is toward the right wing, and z_b is perpendicular to the $x_b - y_b$ plane downward. In the wind axis system, the aircraft's velocity vector, which points in the direction that the aircraft is moving, is aligned with the x-axis (x_w), and like the body axis, y_w is oriented toward the right wing.

The general nonlinear equations of an aircraft's motion come in the form of

$$\dot{\mathbf{x}} = \mathbf{f(x,u)} \tag{1}$$

where **u** is the inputs from the control surfaces and the throttle, and **x** represents all states. The full nonlinear equations are shown below.

$$X - \mathrm{m}g\sin\theta = \mathrm{m}(\dot{u} + qw - rv) \tag{2}$$

$$Y + \mathrm{m}g\cos\theta\sin\phi = \mathrm{m}(\dot{v} + ru - pw) \tag{3}$$

$$Z + \mathrm{m}g\cos\theta\cos\phi = \mathrm{m}(\dot{w} + pv - qu) \tag{4}$$

$$\mathcal{L} = I_{xz}\dot{p} - I_{xz}\dot{r} + qr(I_z - I_y) - I_{xz}pq \tag{5}$$

$$\mathcal{M} = I_y\dot{q} + rq(I_x - I_z) + I_{xz}(p^2 - r^2) \tag{6}$$

$$\mathcal{N} = -I_{xz}\dot{p} + I_z\dot{r} + pq(I_y - I_x) + I_{xz}qr \tag{7}$$

$$p = \dot{\phi} - \dot{\psi}\sin\theta \tag{8}$$

$$q = \dot{\theta}\cos\phi + \dot{\psi}\cos\theta\sin\phi \tag{9}$$

$$r = \dot{\psi}\cos\theta\cos\varphi - \dot{\theta}\sin\phi \tag{10}$$

$$\dot{\theta} = q\cos\phi - r\sin\phi \tag{11}$$

$$\dot{\phi} = p + q\sin\phi\tan\theta + r\cos\varphi\tan\theta \tag{12}$$

$$\dot{\psi} = (q\sin\varphi + r\cos\phi)\sec\theta \tag{13}$$

where m and g represent aircraft mass and gravity acceleration, respectively. u, v, w are velocity components in the x_b, y_b, and z_b directions, and p, q, r represent angular rates (roll, pitch, and yaw rate), respectively. ϕ, θ, and ψ represent the Euler angles, and I_x, I_y, I_z, and I_{xz} denote the moment/product of inertia of the aircraft around the body axis. I_{xy} and I_{yz} are taken to be zero since the aircraft is assumed to be symmetric about the $x_b - z_b$ plane. X, Y, Z, and $\mathcal{L}, \mathcal{M}, \mathcal{N}$ are forces (longitudinal and side forces) and moments around x_b, y_b, and z_b (roll, pitch, yaw), respectively. Gravity, aerodynamics, and thrust are the three main forces acting on the aircraft. Aerodynamic forces have components acting along the wind axis, which must be transferred to the body axis using the rotation matrix shown in Eq. (14).

$$\mathbf{R}_{\mathrm{O}} = \begin{pmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{pmatrix} \tag{14}$$

The following are the longitudinal forces and moments.

$$L = 0.5\,\rho\,V_t^2 S(C_{L_0} + C_{L_\alpha}\alpha + C_{L_{\delta_\mathrm{e}}}\delta_\mathrm{e} + C_{L_q}\frac{qc}{2V_t}) \tag{15}$$

$$D = 0.5\,\rho\,V_t^2 S(C_{D_0} + C_{D_\alpha}\alpha) \tag{16}$$

$$\mathcal{M} = 0.5\,\rho\,V_t^2 Sc(C_{m_0} + C_{m_\alpha}\alpha + C_{m_{\delta_\mathrm{e}}}\delta_\mathrm{e} + C_{m_q}\frac{qc}{2V_t}) \tag{17}$$

L and D, commonly expressed as the wind axis, represent lift and drag, respectively, while $\mathcal{M}$ represents pitching moment about the y_b axis. It should be noted that the y_b axis for both the velocity and body axis are the same and so the moment vector does not need to be rotated. Air density, angle of attack, chord length, total velocity, wing surface, and elevator deflections are denoted by ρ, α, c, V_t, S, and δ_e. Other variables are associated with stability and control

derivatives. To have a visual understanding of the effects of icing on flight dynamics as shown in Fig.2, it can be seen that in the above graph (icing conditions), when an aircraft flies through clouds containing supercooled water droplets (existing below 0 degrees Celsius), icing occurs and can alter the aircraft's state. In clean conditions, however, there is no degradation in performance. The two graphs on the right also show the relationship between aerodynamic coefficients and angle of attack for both conditions. When ice accumulates, the stall angle of attack decreases, and for the same lift, drag increases. Therefore, designing a robust flight control system that accounts for these conditions is necessary.

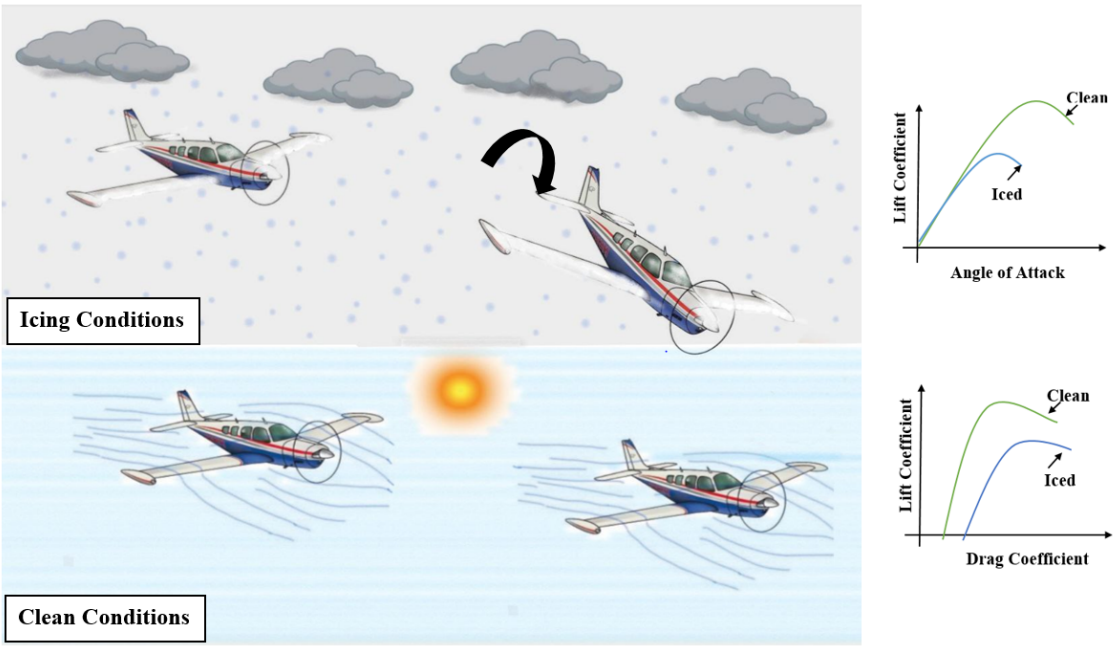


Fig. 2 The Impact of Icing on Aircraft Flight Dynamics

2.2 Aircraft equations in affine form

All equations are written in an affine form as shown in Eq. (18), and this form is utilized in the following section for the AN-NDI control technique.

$$\dot{\mathbf{x}}_i = \mathbf{f}_i(\mathbf{x}, \boldsymbol{\theta}) + \mathbf{g}_i(\mathbf{x}, \boldsymbol{\theta})\mathbf{u}, \quad i = 1, \ldots, 12 \quad \text{(number of states)} \tag{18}$$

$$\mathbf{x} = [x, y, h, u, v, w, \phi, \theta, \psi, p, q, r]^{\mathrm{T}} \tag{19}$$

$$\mathbf{u} = [\delta_{\mathrm{e}}, \delta_{\mathrm{a}}, \delta_{\mathrm{r}}, \delta_t]^{\mathrm{T}}$$

where $\mathbf{f}_i(\cdot)$ and $\mathbf{g}_i(\cdot)$ are vector-valued functions of states as well as aircraft parameters ($\boldsymbol{\theta}$), such as mass, aerodynamic derivatives, and moment of inertia. A different affine form model is employed for the relevant states depending on the control objective. States are represented as $\mathbf{x}$, which includes positions, velocities, Euler angles, and Euler rates of the

aircraft. The control input **u** also includes elevator deflection, aileron, rudder, and throttle, of which, for longitudinal motion, only elevator and throttle are considered. More details on the derivation procedure can be found in [33].

2.3 Icing model

When an aircraft encounters icing, the aerodynamic stability and control derivatives change over time, affecting forces and moments. It is also important to note that control has yet to be tried to control aircraft that experience a wide range of icing conditions, unlike other works [20, 32]. The icing effects model aims to create a straightforward yet physically accurate representation of how ice impacts aircraft flight mechanics. This model is critical in characterizing and simulating the effects, which is essential for developing the Smart Icing System. The foundation of this model, which elucidates the aerodynamic changes under icing conditions, is shown in the following equation based on Bragg's model [47].

$$C_{A_{\text{iced}}} = (I + \eta_{\text{ice}} \cdot K'_{C_A}) \cdot C_A \tag{20}$$

In this case, η_{ice} indicates the icing severity parameter, which reflects the extent and intensity of icing encounters. It is important to note that this parameter is solely influenced by atmospheric conditions rather than the aircraft itself. The term K'_{C_A} refers to aircraft-specific factors, accounting for their potential for icing effects. The parameter C_A includes any performance, stability, or control parameter affected by ice accretion. In this formulation, the weighting factor K'_{C_A} is determined by a combination of parameters, which can be summarized as $K'_{C_A} = f(\text{IPS, aircraft geometry and configuration, icing conditions})$. Similarly, the icing severity parameter η_{ice} is defined through a function that considers the freezing fraction, accumulation parameter, and collection efficiency. These interconnected parameters collectively assess the magnitude of icing's influence on the aircraft's flight mechanics [48]. As for this research, we know the values of C_A and $C_{A_{\text{iced}}}$, but have no access to the values of K'_{C_A} and η_{ice}. Therefore, the following equation, which represents linear interpolation between icing and clean conditions based on [32] to represent ice growth during time, is used in this work to model the changes in aerodynamic derivatives during the flight at a specified time. Additionally, a coefficient κ is introduced to increase the severity of icing for this research artificially:

$$C_A(t) = C_{A_{\text{clean}}} + \frac{\kappa C_{A_{\text{ice}}} - C_{A_{\text{clean}}}}{\Delta t}(t - t_1) \tag{21}$$

Here, C_A denotes a stability or control derivative affected by ice accretion. $C_{A_{\text{clean}}}$ is the value of the derivative in a clean (no-ice) condition, and $C_{A_{\text{ice}}}$ is the corresponding value in icing conditions, available from the NASA database [47]. This study introduces κ as an icing severity parameter ranging from 1 to 3. The time at which ice starts to grow is t_1, and Δt represents the period of ice growth. The proposed model is included in the aforementioned affine form

equations.

3 Flight Control Design

3.1 Adaptive Neural Nonlinear Dynamic Inversion (AN-NDI)

This section aims to design a neural-adaptive flight control system with nonlinear dynamic inversion that enables the aircraft to attain a pre-defined altitude and maintain constant forward velocity in icing conditions. The AN-NDI approach relies on full knowledge of the aircraft's dynamics; therefore, the system's performance may deteriorate in the presence of uncertainties such as icing. In order to resolve this matter, control mechanisms need to be introduced to counteract the uncertainties. In this approach, a structured Radial Basis Function (RBF) network is employed to anticipate uncertainties, with the weights updated using a Lyapunov function to ensure robustness. Specifically, the RBF network is designed to have a predetermined structure, and its weights are learned online to provide adaptation to changing conditions. The Lyapunov function is used to update the weights and guarantee stability, with the resulting controller providing robust performance in the face of uncertainties. The flowchart for the neuro-adaptive flight control method adopted in this study is depicted in Figure 3. It consists of four components, which will be explained in detail in the subsequent subsections.

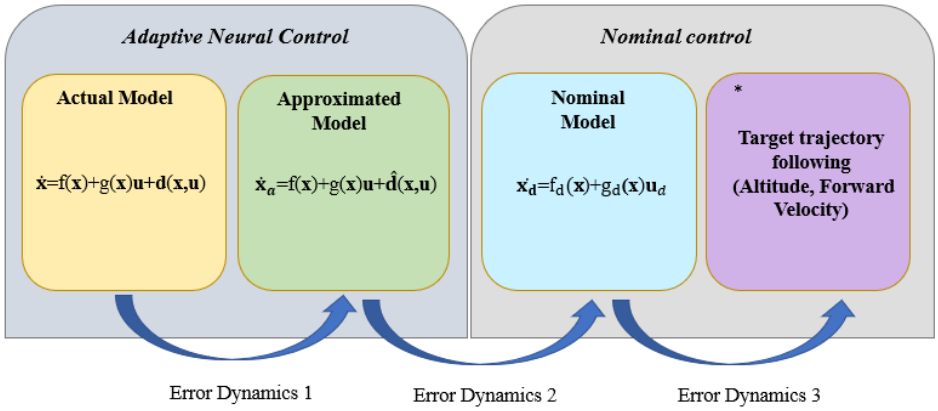


Fig. 3 The process of neuro-adaptive control for aircraft flight in icing conditions

By Figure 3, we initially formulate a nominal control design under the assumption of the absence of icing. This design is constructed by defining three sequential error dynamics. First, Error Dynamics 1 represents the error between the actual and approximated states error. Subsequently, Error Dynamics 2 characterizes the error between the nominal state and the approximated state, aiming to ensure that the approximated state closely tracks the nominal state. Finally, Error Dynamics 3 is introduced to align the desired target forward velocity and altitude. This sequence of error dynamics is established to guide the actual system toward the desired values. The nominal control design is implemented through

conventional nonlinear dynamic inversion [49]. In the subsequent subsection, the neural flight control process is elucidated, emphasizing how Error Dynamics (2 & 3) contribute to adaptive neural flight control of the aircraft in the presence of model uncertainties arising from icing[32, 33].The pseudocode algorithm for neuro-adaptive flight control design is shown in Table 2, and its architecture is shown in Figure 4. More details are provided in the following section.

Table 2 Pseudocode algorithm for neuro-adaptive flight control design

1.	- Initialize states, aircraft parameters, and aerodynamic derivatives values in clean conditions. - Determine constant values of ς_i, γ_i, σ_i, $\mathbf{k}_{a_i}$, and $\mathbf{k}_{b_i}$ - Determine the final aerodynamic derivatives in icing conditions. - At $t = 0$: (no icing, $e_{a_i} = 0$). - Initialize RBF centers
2.	Define the reference target (altitude and forward velocity)
3.	For $i = 1$ to $T_{\text{simulation}}$ / dt: 3.1. obtain $\mathbf{u}_{\text{nominal}}$ by employing conventional Nonlinear Dynamic Inversion flight control 3.2. Icing model (refer to Section 2.3). 3.3. Calculate $\mathbf{u}$ for nominal model using conventional NDI. 3.4. Real model: $\dot{\mathbf{x}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u} + \mathbf{d}_i(\mathbf{x}, \mathbf{u})$ 3.5. Get state and control inputs data from 3.3 (longitudinal motion), $\mathcal{X} = \left[u; w; q; \theta; \delta_e; \delta_t\right]$. 3.6. Approximated model (with icing): 3.6.1. Estimate $\widehat{\mathbf{d}}(\mathbf{x}, \mathbf{u})$ using Radial Basis function with one hidden layer: $\Phi_i(\mathcal{X}) = \exp\left(-\frac{1}{2}\left\|\mathcal{X} - c_i\right\|^2 / \varsigma_i^2\right)$ 3.6.2. c_i obtained by K-Means clustering. 3.6.3. Weight update: $\dot{\widehat{W}}_i = \gamma_i P_i e_{a_i} \Phi_i - \gamma_i \sigma_i \widehat{W}_i$ (P_i: Positive definite matrices). 3.7. Calculate e_{a_i}. 3.8. Calculating control inputs: $\mathbf{u} = (\mathbf{g}_i(\mathbf{x}, \theta))^{-1}[\dot{\mathbf{x}}_{d_i} - \mathbf{k}_{a_i}(\mathbf{x}_i - \mathbf{x}_{a_i}) - \hat{\mathbf{d}}_i(\mathbf{x}, \mathbf{u}) - \mathbf{f}_i(\mathbf{x}, \theta) - \mathbf{k}_{b_i}(\mathbf{x}_{a_i} - \mathbf{x}_{d_i})]$ 3.8. Go to step 3.1 3.9 End.

3.1.1 Neuro-adaptive flight control design

To design a neuro-adaptive control system based on Figure 3, it is essential to first design nominal control. Subsequently, the actual system must track the approximated system, which, in turn, follows the nominal control to attain the desired control inputs. To achieve this objective, the actual system can be represented as:

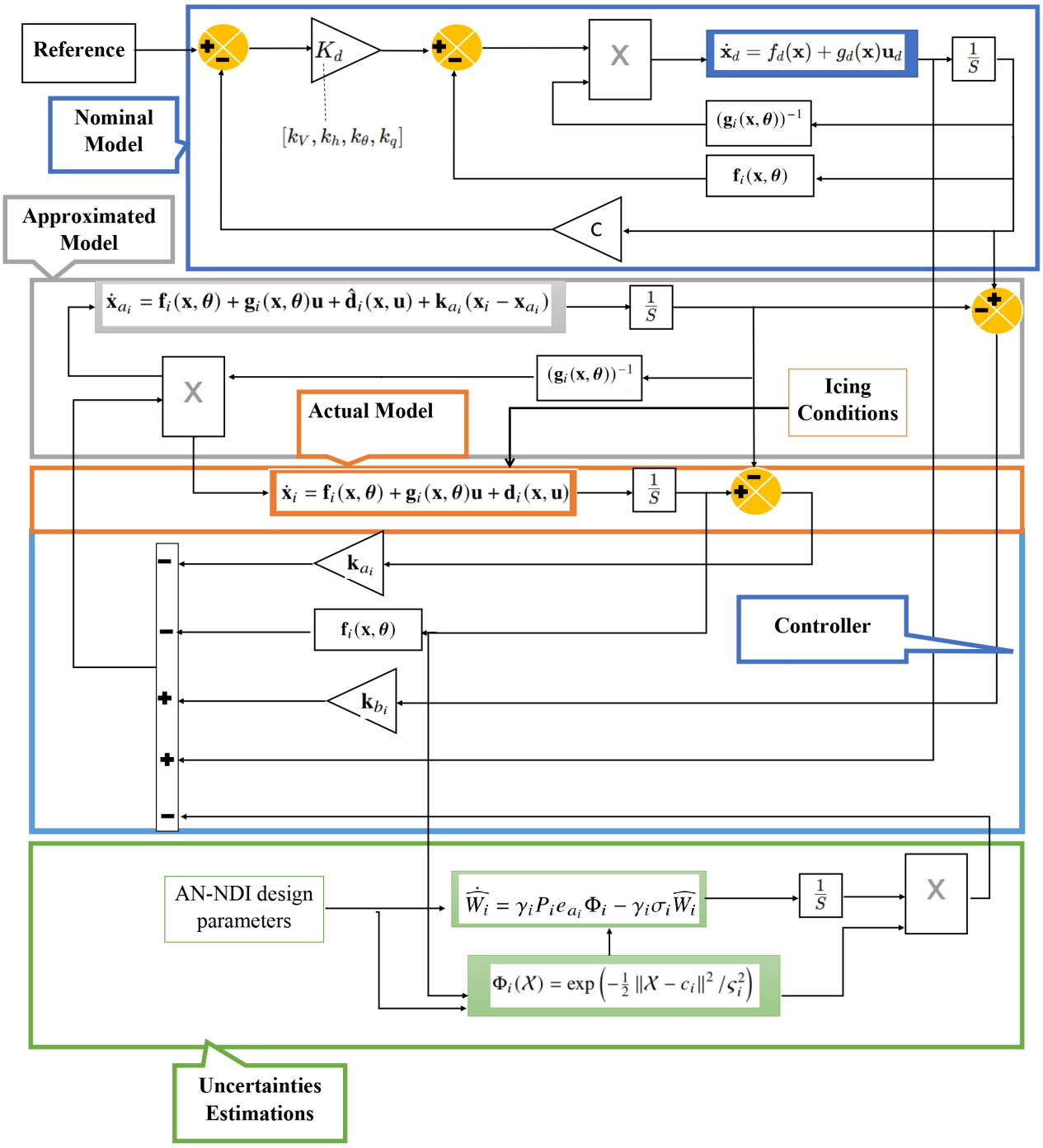


Fig. 4 AN-NDI architecture

$$\dot{\mathbf{x}}_i = \mathbf{f}_i(\mathbf{x}, \boldsymbol{\theta}) + \mathbf{g}_i(\mathbf{x}, \boldsymbol{\theta})\mathbf{u} + \mathbf{d}_i(\mathbf{x}, \mathbf{u}) \tag{22}$$

Here, the term $\mathbf{d}_i$ represents the uncertainty caused by icing. By estimating $\mathbf{d}_i$, the actual system dynamics can be approximated as:

$$\dot{\mathbf{x}}_{a_i} = \mathbf{f}_i(\mathbf{x}, \boldsymbol{\theta}) + \mathbf{g}_i(\mathbf{x}, \boldsymbol{\theta})\mathbf{u} + \hat{\mathbf{d}}_i(\mathbf{x}, \mathbf{u}) + \mathbf{k}_{a_i}(\mathbf{x}_i - \mathbf{x}_{a_i}) \tag{23}$$

This equation represents the dynamics of a 'disturbance observer' [33]. The term $\hat{\mathbf{d}}_i(\mathbf{x}, \mathbf{u})$ denotes an estimation of $\mathbf{d}_i$, where $\hat{\mathbf{d}}_i(\mathbf{x}, \mathbf{u})$ is the adaptation term, and $\mathbf{k}_{a_i}$ is a positive definite gain matrix chosen by the designer and tuned for optimal performance. The objective is to simultaneously ensure $\mathbf{x} \to \mathbf{x}_{a_i}$ and $\mathbf{d}_i(\mathbf{x}) \to \hat{\mathbf{d}}_i(\mathbf{x})$. Once this objective is achieved, the identified observer dynamics can closely represent the actual system dynamics. In order to calculate $\hat{\mathbf{d}}_i(\mathbf{x}, \mathbf{u})$ that guarantees the stability of system, first we subtract the actual and approximated system. By introducing the variable $\mathbf{e}_{a_i}$, defined as the difference between $\mathbf{x}_i$ and $\mathbf{x}_{a_i}$, we can describe the error dynamics as follows:

$$\dot{\mathbf{e}}_{a_i} = \mathbf{d}_i(\mathbf{x}, \mathbf{u}) - \hat{\mathbf{d}}_i(\mathbf{x}, \mathbf{u}) - \mathbf{k}_{a_i}(\mathbf{x}_i - \mathbf{x}_{a_i}) \tag{24}$$

To calculate $\hat{\mathbf{d}}_i(\mathbf{x}, \mathbf{u})$, several nonlinear function approximators can be employed, such as polynomials, the Kronecker product of certain functions, and neural networks. One of the most well-known and valuable functions approximators that can capture the uncertainties online is the Radial Basis Function (RBF) [50], notably for adaptive flight control, which is used in our study (Figure 5). A Radial Basis Function Network (RBFN) is a type of artificial neural network that uses radial basis functions as its activation functions. In this study, The network's output is a combination of these radial basis functions based on the inputs and the neuron parameters. An RBF network consists of three layers: an input layer, a hidden layer with a nonlinear radial basis activation function, and a linear output layer. The hidden layer performs a nonlinear mapping from the input space to a higher-dimensional space using a Gaussian function. The output layer then produces the final result by performing a weighted sum with a linear output. More details are explained in Section 3.1.1 and Appendix B.2. The following is how we expressed the uncertainties for the actual and approximated uncertainties:

$$\hat{\mathbf{d}}_i(\mathbf{x}, \mathbf{u}) = \hat{W}_i^T \Phi_i(\mathcal{X}) \tag{25}$$

$$\mathbf{d}_i(\mathbf{x}, \mathbf{u}) = W_i^T \Phi_i(\mathcal{X}) + \epsilon \tag{26}$$

where $\mathbf{d}_i(\mathbf{x}, \mathbf{u})$ represents the actual parameter uncertainty, W_i is the weight matrix of each output, $\Phi_i(\mathcal{X})$ is the basis function for each channel, and ϵ is the error term related to the network. The variable $\mathcal{X}$, as the input vector of an RBF network, is characterized as a tuple comprising the state vector $\mathbf{x}$ and the input vector $\mathbf{u}$. The output of the RBF network

will be denoted as $\hat{\mathbf{d}}_i(\mathbf{x}, \mathbf{u})$. Although there are multiple possible forms for the function $\Phi_i(X)$, this study has arrived at a specific expression after conducting numerous trials and errors. The expression is based on the square of the Euclidean distance between the input vector and the estimated online centers vector, formulated as follows:

$$\Phi_i(X) = \begin{bmatrix} e^{-1/2(||X-c_1||^2/100)} \\ e^{-1/2(||X-c_2||^2/10)} \\ e^{-1/2(||X-c_3||^2/1)} \\ e^{-1/2(||X-c_4||^2/0.1)} \end{bmatrix} \tag{27}$$

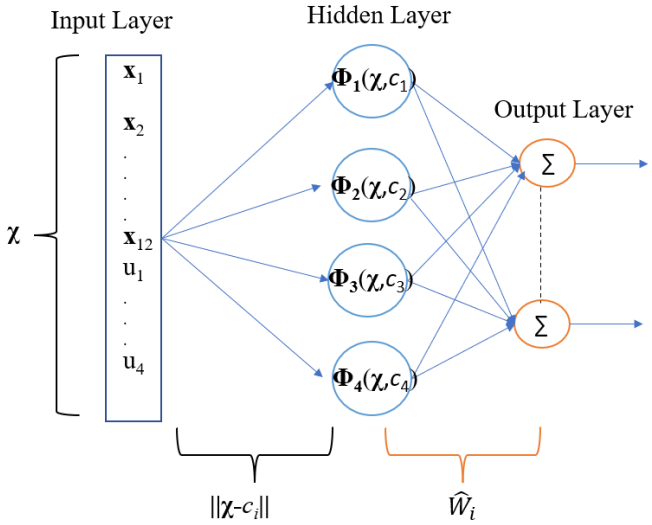


Fig. 5 Radial Basis Function Network (One hidden layer)

The online k-means algorithm is employed to identify the values of the c_i as centroids [51]. Before feeding the dataset into an activation function, the dataset is normalized using the z-score approach. It is important to note that the values of c_i are updated at each time step as new data points come into the algorithm. To derive the weight update rule in adaptive control problems, various methods exist for calculating unknown weights to estimate uncertainties. Here, we compute these unknown weights using a Lyapunov function instead of the typical gradient descent approach to calculate weights in neural networks. This choice is made to ensure system stability. To achieve this, we define the following Lyapunov function (LF):

$$V_i = \frac{1}{2}\left[e_{a_i}^T P_i e_{a_i} + \widetilde{W}_i^T \gamma_i^{-1} \widetilde{W}_i\right] \tag{28}$$

The positive definite LF, denoted by V_i, consists of two terms: the error term $e_{a_i}^T P_i e_{a_i}$ and the term $\widetilde{W}_i^T \gamma_i^{-1} \widetilde{W}_i$, where $\widetilde{W}_i = W_i - \hat{W}_i$, and W_i and $\hat{W}_i$ denote the actual and estimated weights, respectively. γ_i and P_i are positive

definite matrices. Finally, the weight update rule is obtained by enforcing the derivative of the LF to be negative, as detailed in the mathematical procedure and proof (omitted here but available in reference [52]).

$$\dot{\widehat{W_i}} = \gamma_i P_i e_{a_i} \Phi_i - \gamma_i \sigma_i \widehat{W_i} \tag{29}$$

The positive scalar parameter σ_i controls the speed of weight adaptation by dictating how quickly the weights are adapted during the learning process. As the static and dynamic derivatives of an aircraft's aerodynamic coefficients are impacted by icing, any icing-related uncertainty will only be present in the dynamic equations involving velocities and angular rates. Thus, for six-degree-of-freedom (6 DOF) flight dynamics, e_{a_i} will have a total of 6 elements, which correspond to the errors in the six states: 3 velocity values (u, v, w), and three angular rate values (p, q, r), and the number of Φ_i is equal to the number of prespecified clusters.

To construct an effective neuro-adaptive flight control system, it is essential to ensure that the following first-order error dynamics are satisfied, allowing the estimated states to converge to their nominal values over time. This can be expressed mathematically as:

$$(\dot{\mathbf{x}}_{a_i} - \dot{\mathbf{x}}_{d_i}) + \mathbf{k}_{b_i}(\mathbf{x}_{a_i} - \mathbf{x}_{d_i}) = 0 \tag{30}$$

Here, $\mathbf{x}_{a_i}$ and $\mathbf{x}_{d_i}$ are vectors representing the estimated and nominal states, respectively. It is also assumed that all states can be measured. The diagonal positive matrix of $\boldsymbol{k}_b$ represents a set of gains used to tune the control system. Given that the flight dynamics equations are affine, Eq.(23) can be utilized to rephrase Eq.(30) and determine the necessary control inputs. Consequently, utilizing neuro-adaptive control can yield the following control inputs:

$$\begin{aligned} \mathbf{u} = &(\mathbf{g}_i(\mathbf{x}, \boldsymbol{\theta}))^{-1} [\dot{\mathbf{x}}_{d_i} - \mathbf{k}_{a_i}(\mathbf{x}_i - \mathbf{x}_{a_i}) \\ &- \hat{\mathbf{d}}_i(\mathbf{x}, \mathbf{u}) - \mathbf{f}_i(\mathbf{x}, \boldsymbol{\theta}) - \mathbf{k}_{b_i}(\mathbf{x}_{a_i} - \mathbf{x}_{d_i})] \end{aligned} \tag{31}$$

It is crucial to emphasize that the neural-adaptive control design incorporates nominal and approximated dynamics, including uncertainty estimation. The design can achieve robustness and ensure reliable performance by accurately determining the uncertainties. However, selecting numerous parameters in the AN-NDI approach to achieve optimal flight performance to follow the target altitude and velocity can be time-consuming and computationally expensive. Moreover, the singularity of the control matrix ($\mathbf{g}_i(\mathbf{x}, \boldsymbol{\theta})$) can pose additional challenges. The ARS-LQR technique is an alternative approach to overcome these problems and ensure local stability.

3.2 Adaptive Robust Servo Linear Quadratic Regulator (ARS-LQR)

In this section, we develop a flight control system for the altitude/velocity tracking problem utilizing a Robust Servo Linear Quadratic Regulator strategy for a linear time-varying (LTV) system, considering that the LTV system was produced from the aircraft's nonlinear equations of motion. To accomplish this control system design, for each fixed-time step (frozen time) along the local trajectory of the aircraft, the LTV model was first built online and subsequently controlled using the optimal control strategy, the so-called Adaptive Robust Servo Linear Quadratic Regulator. This methodology has recently received much interest and has been proven effective in robotics, quadrotors, and generally for nonlinear systems [38]. In particular, the linear quadratic regulator (LQR) is one of the best techniques in optimal control for stabilizing LTI systems to the origin, with a robustness guarantee of 6 dB gain margin and 60° phase margin [34]. Moreover, when coupled with a servomechanism, the system tracks the desired reference with the most minor tracking error. The servomechanism entails either changing the primary states and control inputs used in the cost function formula by creating new state-space models or adding/multiplying some functions related to the aims of the problem, as discussed in the following sections.

Although LQR is most commonly associated with LTI systems [34], it can also be applied to nonlinear systems to provide a local approximation of the optimal control solution [36, 38]. It should be noted that this work aims to control the aircraft's altitude and velocity in icing situations. Two methods are applied. First, the entire state-space model (at each fixed-time step) is selected for the ARS-LQR design if the reference is equivalent to the initial settings, indicating that the aircraft is on cruise. When the reference is changed, and the aircraft enters a climb or descent phase, the obtained state-space model is divided into three parts, one of which is used for velocity control using the ARS-LQR method, and the other two, known as the inner and outer loops (sequential closure loops), are served for another ARS-LQR design to control the aircraft's altitude or pitch angle (a slow dynamic state). The subsequent sections go into more detailed explanations.

3.2.1 Local approximation of nonlinear equations of motion

The linear time-varying (LTV) model is approximated by linearizing the nonlinear aircraft dynamics along the aircraft's trajectory [53]. Although the system matrices in the state-space model are produced by linearizing at each 0.3-second time, this differs from the conventional concept of time-varying system matrices for LTV systems. Nevertheless, since we compute these matrices at every time step, we treat them as components of the LTV model. As previously mentioned, this approach is utilized to implement a time-varying LQR to achieve the desired command tracking. Figure 6 illustrates the conceptual representation of nonlinear approximations at different operating points. Multiple LTI models are built to constitute the LTV model for the nonlinear aircraft dynamics.

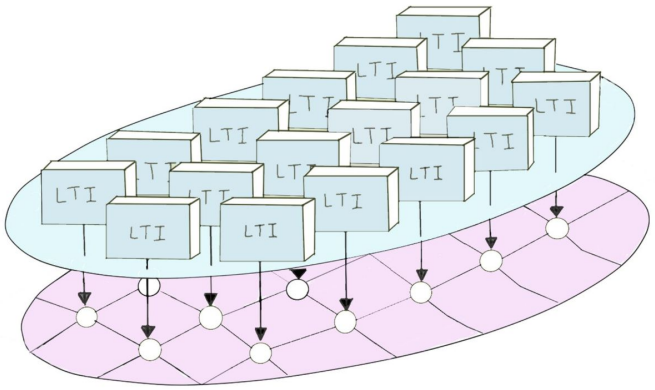


Fig. 6 Establishing LTI models at various operating points

To create an LTV model in Figure 6, the relative coordinate system needs to be defined first, as follows:

$$\bar{\mathbf{x}}_i = \mathbf{x}_i(t) - \mathbf{x}_{0_i}, \qquad i = 0, 1, 2, \ldots, n, \quad (n\text{: number of fixed-time steps})$$
$$\bar{\mathbf{u}}_i = \mathbf{u}_i(t) - \mathbf{u}_{0_i} \tag{32}$$

where states and control inputs are defined as

$$\mathbf{x} = \left[x, y, z, u, v, w, \phi, \theta, \psi, p, q, r\right]^{\mathrm{T}}$$
$$\mathbf{u} = \left[\delta_{\mathrm{e}}, \delta_{\mathrm{a}}, \delta_{\mathrm{r}}, \delta_t\right]^{\mathrm{T}} \tag{33}$$

Here, $\bar{\mathbf{x}}_{\mathrm{i}}$ shows the deviation of the aircraft's states at each fixed-time step (i^{th}) from the current trajectory ($\mathbf{x}_{0_{\mathrm{i}}}$) of the aircraft, and $\bar{\mathbf{u}}_{\mathrm{i}}$ represents the deviation in the aircraft's control input from the current control inputs ($\mathbf{u}_{0_{\mathrm{i}}}$) at each fixed-time step. An operating point is represented by subscript '0', and the current index of the operating point is indicated by subscript 'i'. The time-varying linearization of the nonlinear equations of motion around a nominal trajectory is then applied. Eq. (32) yields the following:

$$\dot{\mathbf{x}}_{\mathrm{i}} = \mathbf{f}(\mathbf{x}_{\mathrm{i}}, \mathbf{u}_{\mathrm{i}})$$
$$\dot{\bar{\mathbf{x}}}_{\mathrm{i}} = \dot{\mathbf{x}}_{\mathrm{i}} - \dot{\mathbf{x}}_{0_{\mathrm{i}}} = \mathbf{f}(\mathbf{x}_{\mathrm{i}}, \mathbf{u}_{\mathrm{i}}) - \mathbf{f}(\mathbf{x}_{0_{\mathrm{i}}}, \mathbf{u}_{0_{\mathrm{i}}}) \tag{34}$$

Then, using the first-order Taylor expansion,

$$\dot{\bar{\mathbf{x}}}_{\mathrm{i}}(t) \approx \underbrace{\mathbf{f}(\mathbf{x}_{0_{\mathrm{i}}}, \mathbf{u}_{0_{\mathrm{i}}}) + \frac{\partial \mathbf{f}(\mathbf{x}_{0_{\mathrm{i}}}, \mathbf{u}_{0_{\mathrm{i}}})}{\partial \mathbf{x}_{\mathrm{i}}}(\mathbf{x}_{\mathrm{i}}(t) - \mathbf{x}_{0_{\mathrm{i}}})}_{\mathbf{A}_{\mathrm{i}}}$$
$$+ \underbrace{\frac{\partial \mathbf{f}(\mathbf{x}_{0_{\mathrm{i}}}, \mathbf{u}_{0_{\mathrm{i}}})}{\partial \mathbf{u}_{\mathrm{i}}}(\mathbf{u}_{\mathrm{i}}(t) - \mathbf{u}_{0_{\mathrm{i}}})}_{\mathbf{B}_{\mathrm{i}}} - \mathbf{f}(\mathbf{x}_{0_{\mathrm{i}}}, \mathbf{u}_{0_{\mathrm{i}}}) \tag{35}$$
$$= \mathbf{A}_{\mathrm{i}}\bar{\mathbf{x}}_{\mathrm{i}}(t) + \mathbf{B}_{\mathrm{i}}\bar{\mathbf{u}}_{\mathrm{i}}(t)$$

It is evident from Eq. (35) that the LQR designed for this linearized model is quite similar to that for a typical LTI model. However, there are some significant distinctions. Since the coordinate system moves with the trajectory, the linearization to provide **A** and **B** matrices (Jacobian in Eqs. (36) and (37)) at each fixed time step is time-varying and applicable for any state along the trajectory, not just the fixed points.

$$\mathbf{A}_{\mathrm{i}} = \left[\begin{pmatrix} \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_1} & \cdots & \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_j} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_1} & \cdots & \frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_j} \end{pmatrix}\right] \quad \begin{matrix} i = 1, 2, \ldots, 12 \\ \\ j = 1, 2, \ldots, 12 \end{matrix} \tag{36}$$

$$\mathbf{B}_{\mathrm{i}} = \left[\begin{pmatrix} \frac{\partial \mathbf{f}_1}{\partial \mathbf{u}_1} & \cdots & \frac{\partial \mathbf{f}_1}{\partial \mathbf{u}_{\mathrm{k}}} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{f}_i}{\partial \mathbf{u}_1} & \cdots & \frac{\partial \mathbf{f}_i}{\partial \mathbf{u}_{\mathrm{k}}} \end{pmatrix}\right] \quad \mathrm{k} = 1, 2, \ldots, 4 \tag{37}$$

Thus, the optimal control obtained from Eq. (35) is $\bar{\mathbf{u}}_{\mathrm{i}}^*$, with the following final control input $\mathbf{u}_{\mathrm{i}}^*$:

$$\mathbf{u}_{\mathrm{i}}^* = \mathbf{u}_{0_{\mathrm{i}}} + \bar{\mathbf{u}}_{\mathrm{i}}^* \tag{38}$$

In the section that follows, the ARS-LQR technique of computing $\mathbf{u}_{\mathrm{i}}^*$ is covered.

3.2.2 ARS-LQR design

Figure 7 shows the generalized Robust Sevo LQR design employed at each fixed-time step. The time-varying $\mathbf{K}_{\mathbf{x}}$ coefficients, which correspond to the aircraft's full states, and the time-varying $\mathbf{K}_{\tilde{\mathbf{y}}}$ coefficients associated with output measurement are calculated using the ARS-LQR technique.

The conventional LQR procedure needs to be supplemented with an integral term for command tracking. Eq. (39)

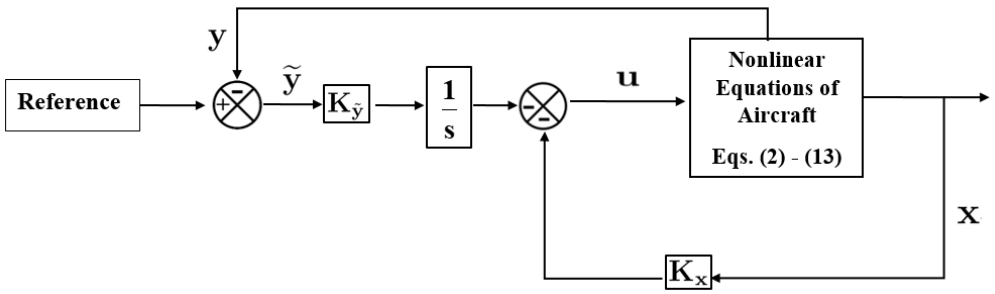


Fig. 7 Architecture of the ARS-LQR

is adopted as a typical LTI model for each fixed-time step in this section [34].

$$\dot{\bar{\mathbf{x}}}_i(t) = \mathbf{A}_i\bar{\mathbf{x}}_i(t) + \mathbf{B}_i\bar{\mathbf{u}}_i(t), \quad i = 0, 1, 2, \ldots$$
$$\bar{\mathbf{y}}_i = \mathbf{C}\bar{\mathbf{x}}_i + \mathbf{D}\bar{\mathbf{u}}_i \tag{39}$$

where $\bar{\mathbf{x}}_i \in \mathbb{R}^{\mathbf{n}_x}$, $\bar{\mathbf{u}}_i \in \mathbb{R}^{\mathbf{n}_u}$, and $\bar{\mathbf{y}}_i \in \mathbb{R}^{\mathbf{n}_y}$ stand for the state, control, and output, respectively. The matrices $\mathbf{A}_i$ and $\mathbf{B}_i$ are updated at each fixed-time step. The following conditions must be fulfilled to apply the ARS-LQR technique:

1) The model must be controllable and observable at each fixed-time step.
2) The number of control inputs must be equal to or less than the number of reference tracking, i.e. $n_{ref} \leq n_y$.

The reference command for the ARS-LQR technique can be constant, ramp, or sinusoidal, which will determine the number of integrators employed in the technique. Ref. [34] contains additional information. ARS-LQR is designed for an augmented state-space model with variables including error and state derivatives ($\dot{\bar{\mathbf{x}}}$), given as follows:

$$\tilde{\mathbf{y}} = \mathbf{y} - \mathbf{y}_{ref} \tag{40}$$

where $\mathbf{y}_{ref}$ is the reference for altitude and velocity.

$$\dot{\bar{\mathbf{z}}}_i = \bar{\mathbf{A}}_i\bar{\mathbf{z}}_i(t) + \bar{\mathbf{B}}_i\mu_i(t) \tag{41}$$

$$\bar{\mathbf{A}}_i = \begin{bmatrix} \mathbf{0} & \mathbf{C} \\ \mathbf{0} & \mathbf{A}_i \end{bmatrix}, \quad \bar{\mathbf{B}}_i = \begin{bmatrix} \mathbf{D} \\ \mathbf{B}_i \end{bmatrix} \tag{42}$$

$$\bar{\mathbf{z}} = [\tilde{\mathbf{y}}, \dot{\bar{\mathbf{x}}}^T]^T \tag{43}$$

$$\mu_i = \dot{\tilde{\mathbf{u}}}_i \tag{44}$$

ARS-LQR works by applying the LQR design procedure to Eq. (41), which lets $\bar{\mathbf{z}}_i$ converge to zero, implying that $\tilde{\mathbf{y}}$ converges to zero, or the tracking of the reference command input is met at a steady state. A cost function needs to be defined to use the infinite-horizon LQR technique for optimal control. Optimal control is achieved by minimizing the following quadratic cost function S as follows:

$$S = \int_0^\infty \frac{1}{2}(\bar{\mathbf{z}}_i^T \mathbf{Q}_i \bar{\mathbf{z}}_i + \mu_i^T \mathbf{R}_i \mu_i)dt \tag{45}$$

where $\mathbf{Q}_i \in \mathbb{R}^{(n_x+n_{ref})\times(n_x+n_{ref})}$ and $\mathbf{R}_i \in \mathbb{R}^{n_u \times n_u}$ are symmetric weight matrices for the augmented states and control derivatives, respectively. The values n_x, n_{ref}, and n_u represent the numbers of states, references, and control inputs. The following standard assumptions are also taken into account:

1) $\mathbf{Q}_i = \mathbf{Q}_i{}^T \succeq 0$; positive semi-definite
2) $\mathbf{R}_i = \mathbf{R}_i^T \succ 0$; positive definite
3) The pair $(\bar{\mathbf{A}}_i, \bar{\mathbf{B}}_i)$ is stabilizable and $(\bar{\mathbf{A}}_i, \mathbf{Q}_i^{\frac{1}{2}})$ detectable.

To obtain optimal control, the following Hamiltonian Jacobian Bellman equation (HJB) is minimized:

$$\min_{\mu_i}\left[\bar{\mathbf{z}}_i^T \mathbf{Q}_i \bar{\mathbf{z}}_i + \mu_i^T \mathbf{R}_i \mu_i + (\frac{\partial S}{\partial \bar{\mathbf{z}}_i})^T(\bar{\mathbf{A}}_i \bar{\mathbf{z}}_i + \bar{\mathbf{B}}_i \mu_i)\right] \tag{46}$$

To solve Eq. (46), we consider $S = \bar{\mathbf{z}}_i^T \mathbf{P}_i \bar{\mathbf{z}}_i$, $\mathbf{P}_i = \mathbf{P}_i^T \succ 0$ which leads to the following algebraic Riccati equation:

$$\mathbf{P}_i \bar{\mathbf{A}}_i + \bar{\mathbf{A}}_i^T \mathbf{P}_i - \mathbf{P}_i \bar{\mathbf{B}}_i \mathbf{R}_i^{-1} \bar{\mathbf{B}}_i^T \mathbf{P}_i + \mathbf{Q}_i = 0 \tag{47}$$

and the corresponding optimal control is obtained, as

$$\mu_i = -\underbrace{\mathbf{R}_i^{-1} \bar{\mathbf{B}}_i^T \mathbf{P}_i}_{\mathbf{K}_i} \bar{\mathbf{z}}_i \tag{48}$$

The optimal control is then derived by integrating Eq. (48) at each fixed-time step, considering the operating point as the initial condition. Then, this control is applied to the nonlinear equations of motion. As ice grows gradually throughout time, a time-varying artificial stability coefficient Λ is introduced to maintain the stability of the closed-loop system at each fixed-time step [35], which modifies Eq. (45) as follows:

$$S_1 = \frac{1}{2}\int_0^\infty (e^{2\Lambda_i t})(\bar{\mathbf{z}}_i^T \mathbf{Q}_i \bar{\mathbf{z}}_i + \mu_i^T \mathbf{R}_i \mu_i)dt, \quad \Lambda_i \succeq 0 \tag{49}$$

Afterward, a new state-space model, Eq. (52), can be produced by considering Eqs. (50)-(51):

$$\check{\mathbf{z}}_i = e^{\Lambda_i t}\bar{\mathbf{z}}_i \tag{50}$$

$$\bar{\mu}_i = e^{\Lambda_i t}\mu_i \tag{51}$$

$$\dot{\check{\mathbf{z}}}_i = (\bar{\mathbf{A}}_i + \Lambda_i \mathbf{I})\check{\mathbf{z}}_i + \bar{\mathbf{B}}_i\bar{\mu}_i \tag{52}$$

This state-space model is the central system for the ARS-LQR-designed aircraft flight dynamics. It is worth noting that the artificial stability coefficient Λ appears in the Riccati equation (Eq. (47)), where $\bar{\mathbf{A}}_i$ is substituted by $\bar{\mathbf{A}}_i + \Lambda_i \mathrm{I}$. This coefficient guarantees a certain degree of stability or ensures that all the eigenvalues of a linear system are located on the left side of a line at a distance of Λ from the imaginary axis [54]. Λ is defined as a variable, being proportional to the degree of stability degradation due to icing, e.g.

$$\Lambda_i = 1.1(\bar{\lambda}(\bar{\mathbf{A}}_i) - \bar{\lambda}(\bar{\mathbf{A}}_0)) \tag{53}$$

Where $\bar{\lambda}(\bar{\mathbf{A}}_i)$ represents the eigenvalue of $\bar{\mathbf{A}}_i$ being the nearest to the imaginary axis along the real axis, and subscript '0' shows the values in the clean condition. The system is vulnerable to instability because the closed-loop eigenvalues approach the imaginary axis as the ice grows. Therefore, as shown in Eq. (53), Λ_i is employed in such a way that the degree of stability is guaranteed, equivalent to the stability degradation caused by icing, which brings uncertainties to the system. As a result, the Riccati equation can be written as follows:

$$\mathbf{P}'_i(\bar{\mathbf{A}}_i + \Lambda_i \mathrm{I}) + (\bar{\mathbf{A}}_i + \Lambda_i \mathbf{I})^{\mathrm{T}}\mathbf{P}'_i - \mathbf{P}'_i\bar{\mathbf{B}}_i\mathbf{R}_i^{-1}\bar{\mathbf{B}}_i^{\mathrm{T}}\mathbf{P}'_i + \mathbf{Q}_i = 0 \tag{54}$$

Then by considering Eqs. (48),(50), and (51),

$$\bar{\mu}_i = -\underbrace{\mathbf{R}_i^{-1}\bar{\mathbf{B}}_i^{\mathrm{T}}\mathbf{P}'_i}_{\bar{\mathbf{K}}_i}\check{\mathbf{z}}_i \tag{55}$$

By looking at Eqs. (50), (51), and (55), we observe that Eq. (55) is the same as (48). Therefore, by considering Λ_i, only Riccati equation (54) is modified to handle icing uncertainties. The sequential loop closure based on Eq. (52) that is used for the climb flight phase is discussed in the following section.

Remark 1 *It can be shown that the integral in Eq.* (49) *is finite with the choice of* Λ_i *in Eq.* (53)*, which is proven in Appendix A.*

Remark 2 *One of the challenging aspects of the ARS-LQR approach is tuning the* **Q** *and* **R** *matrices. In the present*

paper, they are tuned in such a way to achieve the following objectives:

1) Reaching the target altitude and forward velocity while maintaining aircraft performance in icing conditions such as climb rate,

2) Considering all state limitations (maximum and minimum values),

3) Minimizing fluctuations in other states, particularly the angle of attack and pitch angle, and

4) Reducing energy consumption while avoiding control input saturation and fluctuations.

Therefore, these matrices were adjusted through a trial-and-error process tailored to each operational phase, including both the cruise and climbing phases, to address all the mentioned items.

3.2.3 Sequential loop closure control design

As previously mentioned, the entire set of nonlinear equations is adaptively linearized to create an augmented state-space model, which is used for the ARS-LQR technique and lets the aircraft successfully track desired states (altitude and velocity). In the climb phase, finding the appropriate $\mathbf{Q}_i$ and $\mathbf{R}_i$ for the ARS-LQR design can be somewhat tricky when looking at the entire state-space model for tracking, although it is possible. Accordingly, it was discovered that we needed more power to control q as the fastest state first. Therefore, considering the speed of dynamics associated with the characteristics of state variables, the large (5^{th}-order) state model is divided into three smaller state models (two 2^{nd}-order and one 1^{st}-order state model).

Consider the following LTI state-space model for the longitudinal flight dynamics of an aircraft, which is obtained by linearization around an operating point. Jacobian matrices are used to calculate each array of $\mathbf{A}_i$ and $\mathbf{B}_i$.

$$\begin{bmatrix} \dot{\bar{h}} \\ \dot{\bar{w}} \\ \dot{\bar{q}} \\ \dot{\bar{\theta}} \\ \dot{\bar{u}} \end{bmatrix} = \underbrace{\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \end{bmatrix}}_{\mathbf{A}_i} \begin{bmatrix} \bar{h} \\ \bar{w} \\ \bar{q} \\ \bar{\theta} \\ \bar{u} \end{bmatrix} + \underbrace{\begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \\ b_{51} & b_{52} \end{bmatrix}}_{\mathbf{B}_i} \begin{bmatrix} \bar{\delta}_e \\ \bar{\delta}_t \end{bmatrix} \tag{56}$$

The above system can be simplified and divided into three parts for tracking velocity ($\bar{u}$) in the x_b direction and altitude ($\bar{h}$) commands. The independency between the velocity and the remaining state variables in Eq. (56) is a valid presumption that can be considered for flight control designs under most flying scenarios [40, 55, 56]. Consequently,

the velocity dynamics is expressed as follows:

$$\dot{\bar{u}} = a_{55}\bar{u} + b_{52}\bar{\delta}_t \tag{57}$$

After that, the velocity dynamics is subjected to the same procedure as the ARS-LQR design in the previous Section. The remaining flight dynamics of Eq. (56) are considered for altitude tracking, described by

$$\begin{bmatrix} \dot{\bar{h}} \\ \dot{\bar{w}} \\ \dot{\bar{q}} \\ \dot{\bar{\theta}} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} \bar{h} \\ \bar{w} \\ \bar{q} \\ \bar{\theta} \end{bmatrix} + \begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \\ b_{41} \end{bmatrix} \begin{bmatrix} \bar{\delta}_e \end{bmatrix} \tag{58}$$

As demonstrated in Eq. (58), the elevator is only used to control slow dynamics, such as altitude or pitch angle, whereas the throttle is solely used to control velocity, as shown in Eq. (57). The system shown in Eq. (58) has both short and long (phugoid) period dynamics, which prompts us to partition it into two phases, inner and outer loops, for the architecture of the flight control system. The short-period dynamics, which involves q and w, are thus covered by the following definition:

$$\begin{bmatrix} \dot{\bar{w}} \\ \dot{\bar{q}} \end{bmatrix} = \underbrace{\begin{bmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{bmatrix}}_{\mathbf{A}_{i1}} \begin{bmatrix} \bar{w} \\ \bar{q} \end{bmatrix} + \underbrace{\begin{bmatrix} b_{21} \\ b_{31} \end{bmatrix}}_{\mathbf{B}_{i1}} \begin{bmatrix} \bar{\delta}_e \end{bmatrix} \tag{59}$$

And the phugoid dynamics is defined as

$$\begin{bmatrix} \dot{\bar{h}} \\ \dot{\bar{\theta}} \end{bmatrix} = \underbrace{\begin{bmatrix} a_{11} & a_{14} \\ a_{41} & a_{44} \end{bmatrix}}_{\mathbf{A}_{i2}} \begin{bmatrix} \bar{h} \\ \bar{\theta} \end{bmatrix} + \underbrace{\begin{bmatrix} a_{12} & a_{13} \\ a_{42} & a_{43} \end{bmatrix}}_{\mathbf{B}_{i2}} \begin{bmatrix} \bar{w} \\ \bar{q} \end{bmatrix} \tag{60}$$

Both Eqs. (59) and (60) are adopted for altitude tracking. First, the ARS-LQR is used for the outer loop in Eq. (60) to provide the necessary reference command for the inner loop in Eq. (59). Second, another ARS-LQR is applied to Eq. (59) to generate the proper elevator input, and the associated closed-loop system is demonstrated to be stable and to follow commands in icing conditions perfectly. Figure 8 shows the architecture of the ARS-LQR flight control based on the sequential loop closure, which is utilized for the climb phase.

The suggested flight control system is completed once the altitude loop is closed. The following simulation section will show the successful working of the proposed control strategy discussed.

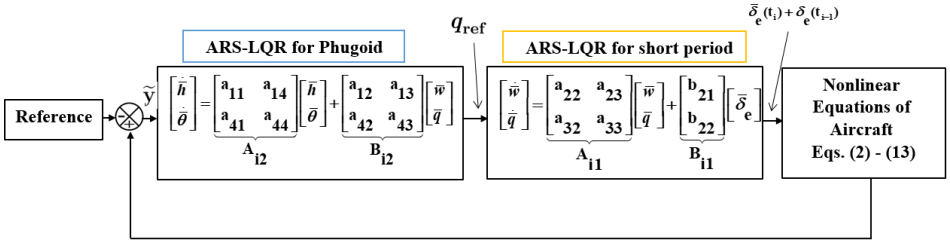


Fig. 8 Sequential loop closure architecture

4 Simulation Results

This section focuses on simulation results achieved with Simulink in MATLAB. It includes comparing open-loop and closed-loop dynamic responses, with a focus on the longitudinal motion of the nonlinear 6-DOF model of the modified NASA Lewis DeHavilland DH-6 Twin Otter Icing Research Aircraft in clean and icing conditions[18, Fig.1]. It possesses all of the stability and control derivatives of the clean and icing conditions. Before comparing the two designed flight control systems, AN-NDI and ARS-LQR, the open-loop dynamic response of the DH-6 Twin Otter was validated in the clean condition, based on flight data from [57]. It was confirmed that the response matched the actual flight data well. Table 3 provides the control input limitations, and Table 4 provides the specifications of the DH-6 Twin Otter.

Table 3 Control Input Limitations for DH-6 Twin Otter

Control Surface	Limitations (deg)
Elevator	$[-25, 25]$
Aileron	$[-17.5, 17.5]$
Rudder	$[-20, 20]$

Table 4 Specifications of the DH-6 Twin Otter

Parameters	Value
Wing span	19.81 m
Wing area	39.02 m^2
Aspect ratio (AR)	10.06
Mean Aerodynamic Chord (MAC)	1.98
Center of gravity (from the nose)	0.572 m
Weight (W)	45,100 N
Maximum thrust per engine	24,230 N

The performance of the two flight control designs (AN-NDI and ARS-LQR) for the DH-6 Twin Otter was investigated under two assumptions: 1) the longitudinal and lateral flight dynamics are independent, and 2) the sideslip angle is zero. In a clean condition, the aircraft is considered to be flying at 1,713 m at 57.25 m/s. To examine the open-loop dynamic response, the appropriate trim control inputs are determined based on the minimization of the quadratic cost function, which is the square sum of forces in the x_b, z_b axes and, pitching moment about the y_b axis. Trim condition values are

shown in Table 5.

Table 5 Trim values in the clean condition

Variables	α (deg)	u	w	θ (deg)	δ_e (rad)	δ_t	δ_r	δ_a
Trim value	2.49	57.249	2.484	2.486	-0.028	0.056	0	0

The initial altitude and speed were set to 1,713 m and 57.25 m/s, respectively. The aircraft's open-loop dynamic responses were then evaluated using the following scenarios:

1) From 0 to 100 s, the aircraft is flying in clean condition, with control inputs set to constant trim values.
2) The aircraft is flying in clean condition from 0 to 1 s, and then ice begins to form on the wing or (and) tail (wing only for the first setting, tail only for the second setting, both wing and tail for the third setting) during 1 to 100 s, which are simulated through all stability and control derivatives varying linearly over time.

It should be noted that most ice accumulations last several minutes; however, in these simulations, the severity parameter is assumed to be from 1 to 3, so some stability and control derivatives vary by more than 500%. The flight test data for the DH-6 Twin Otter regarding iced stability and control derivatives are associated with $\kappa = 1$ [6]. However, κ is set to 3 in this paper, which simulates much more severe or extreme situations due to icing. Table 6 presents the stability and control derivatives changes for both severity levels.

Table 6 Changes in stability and control derivatives for two distinct levels of severity

Severity Coefficient	ΔC_{z_0}	ΔC_{z_α}	ΔC_{z_q}	$\Delta C_{z_{\delta_e}}$	ΔC_{x_0}	ΔC_{m_0}	ΔC_{m_α}	ΔC_{m_q}	$\Delta C_{m_{\delta_e}}$
$\kappa = 1$	0	10%	1.35%	9.53%	51.21%	0	9.92%	3.5%	10%
$\kappa = 3$	0	30%	4.05%	28.61%	153.65%	0	29.77%	10.52%	30%

Figure 9 indicates changes in h, u, V_t, α, θ, and q for the two defined scenarios. Blue, red, green, and black colors indicate clean, iced wing and tail, iced wing, and iced tail conditions, respectively. According to Figure 9, the highest altitude loss occurs when both wing and tail are affected by ice, in which case the altitude decreases by about 310 m during 100 s. Figure 9-(c) demonstrates that the iced tail causes V_t to increase by about 1.75 m/s, whereas the iced wing results in a total velocity change of about 0.3 m/s. Figure 9-(d) shows that when both wing and tail or only wing is affected by ice, the angle of attack increases by 0.17 deg, whereas the angle of attack decreases when only the tail is affected by ice. Therefore, it is evident that in severe icing conditions, an iced wing may induce a stall as the angle of attack becomes undesirably large. When both wing and tail are affected by ice, as shown in Figure 9-(f), the pitch rate fluctuation reaches the maximum amplitude—roughly twice as much as when either wing or tail only is affected by ice. The following simulation results depict the worst-case scenario of icing on both the wing and tail of the aircraft, exhibiting the largest variations in altitude and pitch rate. These results provide insights into the effects of icing on the aircraft's performance and can inform decisions regarding safe operating conditions in icy environments.

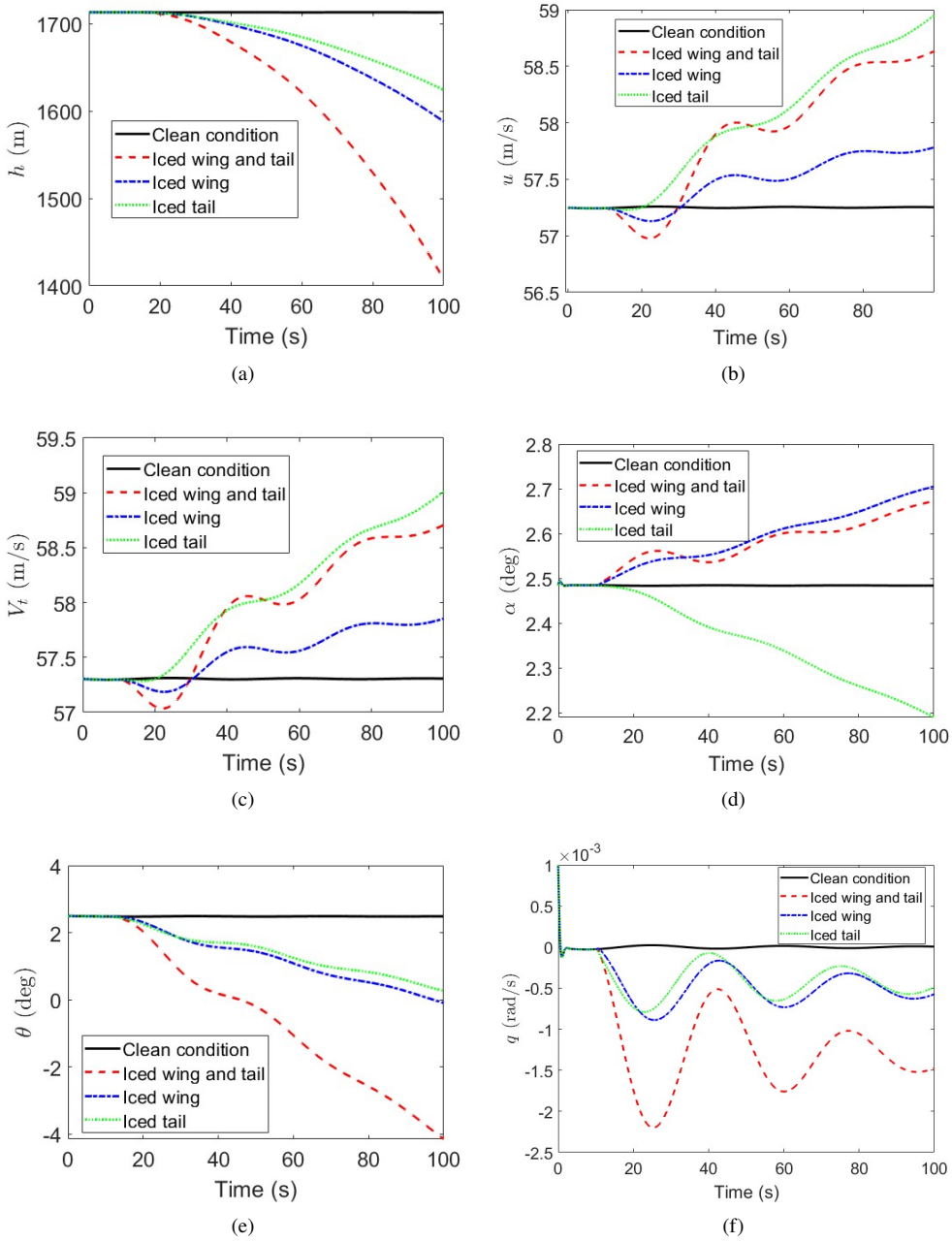


Fig. 9 States changes in the DH-6 Twin Otter for both clean and icing conditions

In this study, we applied the AN-NDI and ARS-LQR flight control techniques to the DH-6 Twin Otter aircraft under icing conditions. Two different flight control objectives were considered to evaluate the effectiveness of the flight control techniques in challenging icing conditions. The first objective was to maintain a steady altitude of 1,173 m and a forward velocity of 57.25 m/s during the cruise phase, and the second objective was to climb from an initial altitude of 1,723 m to a target altitude of 1,800 m while maintaining the same cruise velocity of 57.25 m/s. These flight control objectives are designed to test the ability of the flight control techniques to achieve specific performance goals in the presence of ice accretion on the aircraft's wings and tail. In order to achieve the best performance during various flight phases, it is crucial to select optimal time constants for error dynamics in Eqs.(24) and (30), constant parameters for weight update rule, number of clusters for RBF, and positive matrices in the defined Lyapunov function for the AN-NDI flight control technique (Table 7). This process requires a lot of trial and error to fine-tune the control parameters. In contrast, for the ARS-LQR technique, the key parameters to be selected are the weighting matrices **Q** and **R** (Table 8), and the coefficient in Eq. (53), which determine the optimal elevator deflection and throttle input for minimizing energy consumption, while keeping the non-controlled states (all but altitude and forward velocity) stable. Overall, the AN-NDI technique involves more parameters and can be more challenging to optimize than the ARS-LQR technique.

Table 7 Parameter values of AN-NDI technique and matrix P from Eq. (28)

Parameter	k_V	k_h	k_θ	k_q	k_{a_1}	k_{a_2}	k_{a_3}	k_{a_4}	k_{a_5}
Value	10	0.05	0.2	0.33	1.25	0.3	1.25	0.875	0.5
Parameter	k_{a_6}	k_{b_1}	k_{b_2}	k_{b_3}	k_{b_4}	k_{b_5}	k_{b_6}	$\sigma_{(i=1:6)}$	$\gamma_{(i=1:6)}$
Value	0.5	0.015	0.015	0.015	0.06	0.06	0.06	1×10^{-6}	60
Parameter	P from Eq.(28)								
Value	$\mathrm{diag}([0.7, 0.8, 0.7, 0.075, 0.075, 0.075])$								

Table 8 Parameter values for matrices Q and R in climbing and cruising phases of ARS-LQR approach

Climbing Phase	
$\mathbf{Q}_{\text{Climb}}$	$\mathrm{diag}\left([5 \times 10^{-7}, 5 \times 10^{-8}, 0.2, 6.5, 0.2, 0.2, 5 \times 10^{-7}]\right)$
$\mathbf{R}_{\text{Climb}}$	$\mathrm{diag}\left([1, 80]\right)$
Cruising Phase	
$\mathbf{Q}_{\text{Cruise}}$	$\mathrm{diag}\left([1 \times 10^{-11}, 5 \times 10^{-13}, 0.2, 40, 0.2, 0.2, 1 \times 10^{-6}]\right)$
$\mathbf{R}_{\text{Cruise}}$	$\mathrm{diag}\left([40, 4000]\right)$

To establish the credibility of the AN-NDI approach, we first demonstrate its accuracy of estimated states produced from estimated dynamics. This is accomplished by directly comparing the actual states, as shown in Figs.10 (a-d). The Radial Basis Function (RBF) is employed to estimate icing uncertainties. As previously discussed, These uncertainties are subsequently incorporated into the estimated dynamics. The specifications of RBF networks used in this research, along with the weight updates throughout time, are detailed in Appendix B.2. Overall, Fig.B2 highlights the adaptability

of the RBF network in handling uncertainties in flight control due to ice accretion, and the time series plots demonstrate the network's dynamic adjustments.

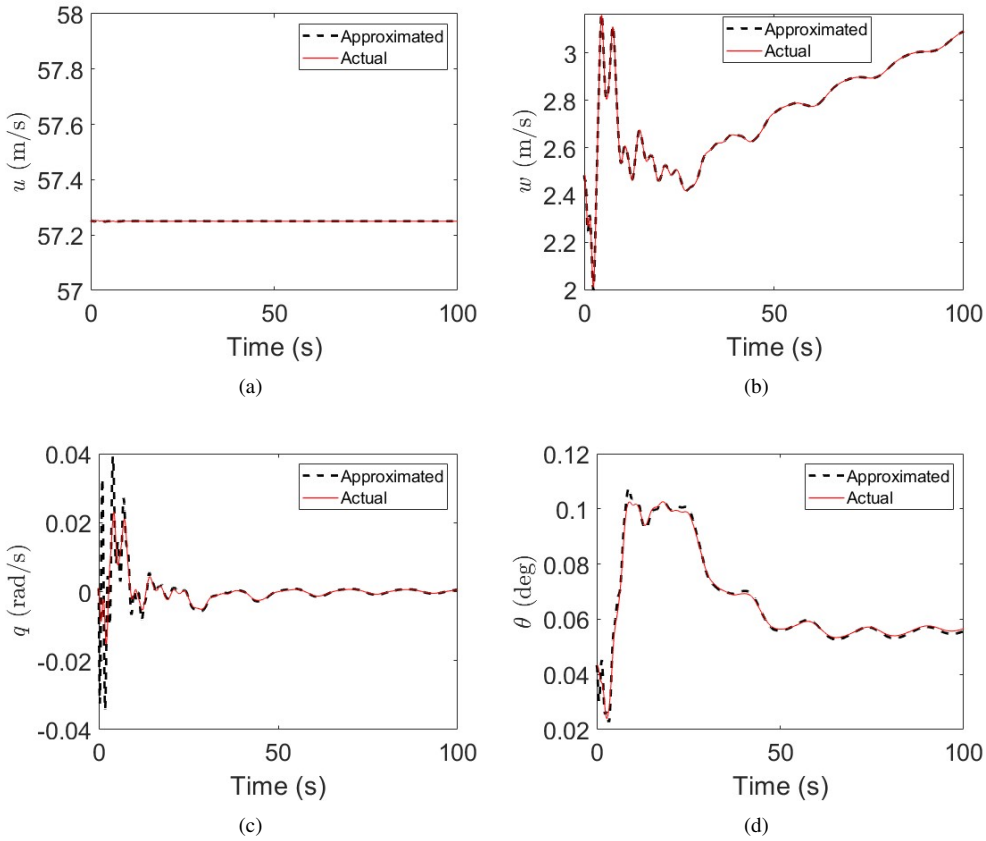


Fig. 10 Evaluation of the effectiveness of an RBF neural network using AN-NDI approach

In the case of ARS-LQR, the simulation step time is 0.1 seconds, and linearization of the nonlinear equations of motion is performed at each 0.3-second interval. The size of the time interval has been thoroughly studied to determine how it affects the controller's performance. Various time step sizes were tested during the design process, as shown in Table 9. The cost function, calculated based on the sum of the squares of the difference of states from reference values and control inputs (similar to the LQR cost function), indicated an increase in cost values for five steps or more. This increase leads to aircraft performance degradation under these time intervals. As nonlinearity increases, a minor time step may be required to capture the system's dynamics accurately. However, based on Figures B1(a) and (b), which show the forward velocity (u) and altitude (h) at different time steps, alongside the cost analysis detailed in Table 9, it is apparent that time steps of 1, 2, and 3 exhibit suitability for this application. Within this investigation, a time step of 3 is chosen for this study.

A comparison is made between the performance of the ARS-LQR and AN-NDI techniques by evaluating their closed-loop simulation results for the climb and cruise phases under severe icing conditions. Figures 11-12 show a

comparison where the reference values for altitude and forward velocity were set to 1,800 m and 57.25 m/s in the climb phase and 1,713 m and 57.25 m/s in the cruise phase. Since the DH-6 Twin Otter can climb at a maximum rate of about 8 m/s, this was considered when tuning the ARS-LQR parameters and for AN-NDI. Both methods enable the aircraft to follow the reference, but ARS-LQR exhibited much better performance in fewer oscillations in the angle of attack, total velocity, pitch rate, throttle, and elevator deflections, with smoother actuator inputs. Figure 12 presents the closed-loop responses of the aircraft in the cruise phase. The AN-NDI technique showed considerable fluctuations in states and control inputs, including $\theta, \delta_t, \delta_e, q, \alpha, V_t$, compared to the ARS-LQR technique. Although the AN-NDI technique can adapt to uncertainties and reduce these fluctuations over time, it consumes more energy and may not be desirable. The ARS-LQR technique, on the other hand, outperformed AN-NDI by achieving smoother control inputs with fewer oscillations, resulting in more efficient energy consumption.

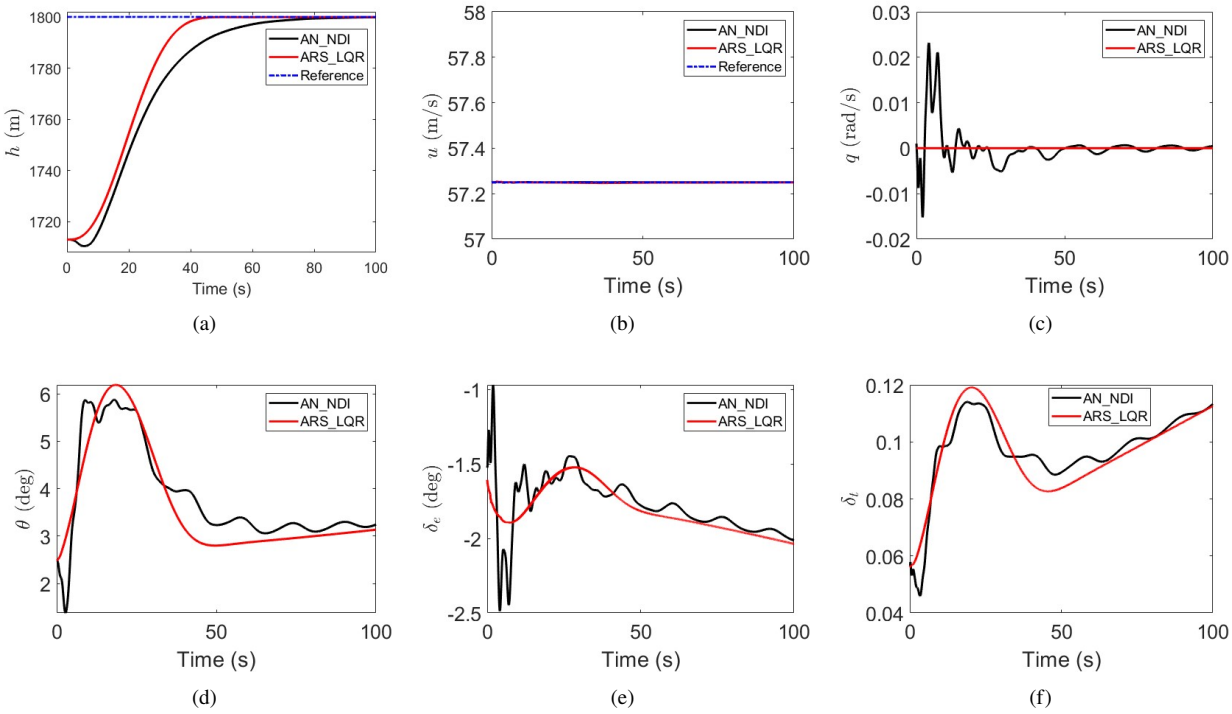


Fig. 11 Closed-loop responses of the aircraft in the climb phase under icing conditions using the ARS-LQR and AN-NDI techniques

Finally, the closed-loop stability was analyzed for ARS-LQR. Unlike AN-NDI, ARS-LQR is based on a sequence of LTI state-space models, which permits an eigenvalue analysis. As the state-space model is LTI at each time step, we consider whether all the eigenvalues of each of the closed-loop systems are located in the left half of the complex plane or not, i.e., $\bar{\lambda}(\bar{\mathbf{A}}_\mathrm{i} - \bar{\mathbf{B}}_\mathrm{i}\bar{\mathbf{K}}_\mathrm{i})$[1]. Figure 13 compares $\bar{\lambda}(\bar{\mathbf{A}}_\mathrm{i})$ and $\bar{\lambda}(\bar{\mathbf{A}}_\mathrm{i} - \bar{\mathbf{B}}_\mathrm{i}\bar{\mathbf{K}}_\mathrm{i})$, i.e., the open-loop and closed-loop eigenvalues

[1] In fact, this eigenvalue analysis only guarantees local stability around each of linearization points, rather than global stability covering the whole flight envelope. However, thanks to the guaranteed stability margin inherited from the LQR design, the local stability analysis is practically accepted and believed to cover the whole flight envelope [58]

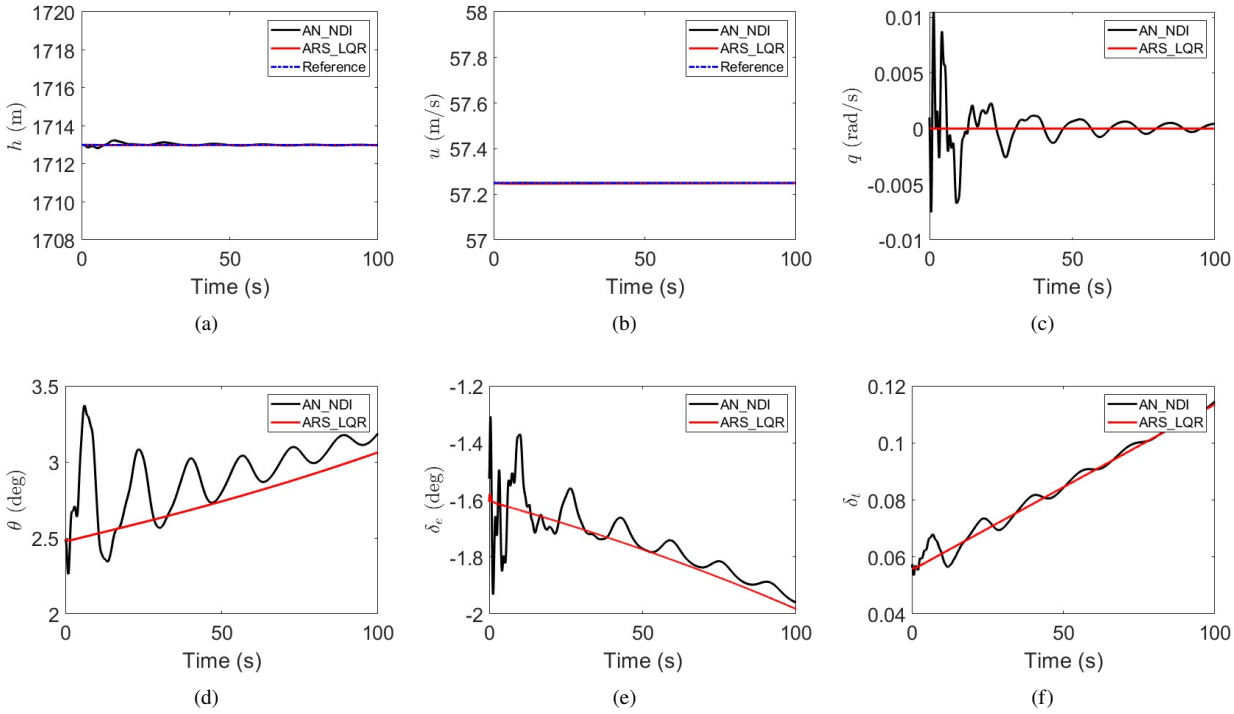


Fig. 12 Closed-loop responses of the aircraft in the cruise phase under icing conditions using the ARS-LQR and AN-NDI techniques

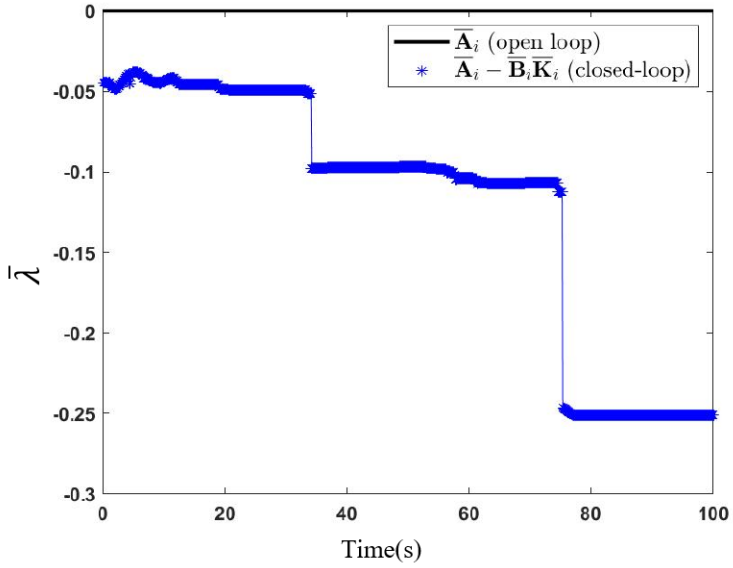


Fig. 13 $\bar{\lambda}(\bar{\mathbf{A}}_\mathbf{i})$ and $\bar{\lambda}(\bar{\mathbf{A}}_\mathbf{i} - \bar{\mathbf{B}}_\mathbf{i}\bar{\mathbf{K}}_\mathbf{i})$ before and after control

being the nearest to the imaginary axis. Notably, in this analysis, $\bar{\lambda}$ specifically denotes the maximum eigenvalue of the closed-loop at each three-time step, evaluated across 100 different initial conditions. This approach demonstrates that it has zero eigenvalues before applying control, but the eigenvalues are relocated to the left half of the complex plane after control at all times. These extensive simulations show that the autopilot can achieve the desired altitude and forward

velocity in icing conditions. Therefore, together with the robustness property of LQR, we logically conclude that the aircraft exhibits local stability even in the presence of icing, and ARS-LQR is acceptable for use in practical applications.

5 Conclusion

The primary objective of this study was to investigate the performance of the Adaptive Robust Servo Linear Quadratic Regulator (ARS-LQR) when implemented as the core of a flight control system. Additionally, we employed the Adaptive Neural Nonlinear Dynamic Inversion (AN-NDI) for comparison. An online Radial Basis Function (RBF) network was utilized to predict uncertainties, and a Lyapunov function was defined to build the adaptive control law in the AN-NDI. The ARS-LQR was designed by constructing an online Linear Time-Varying system generated by linearizing the nonlinear aircraft equations around the local trajectory of the aircraft at each fixed-time step. This allowed the design of an LQR at each time step using new system matrices. Although it may resemble classical control techniques, the proposed ARS-LQR introduces a distinctive and highly practical element. Theoretically, it incorporates an exponential multiplier into the cost function to ensure local stability, even under severe icing conditions, with substantial margins (6 dB gain margin and 60-degree phase margin). This theoretical merit is of significant interest to aircraft control engineers. To validate this, a comprehensive comparative study demonstrates ARS-LQR's superior performance in tracking altitude and forward velocity references during severe icing. Importantly, as far as the authors' knowledge extends, no control technique has been proposed for aircraft encountering such a wide range of icing conditions (from moderate to severe). The simulation results indicated that in icing conditions, ARS-LQR outperformed AN-NDI by enabling the aircraft to follow altitude and forward velocity references, and oscillations in states that were present in AN-NDI were removed. Furthermore, the number of parameters required for tuning in ARS-LQR was at least half of those needed for AN-NDI, making it more computationally efficient. Moreover, simulations conducted under 100 different initial conditions and the analysis of the maximum eigenvalue at each three-time step affirm the local stability of the designed flight control system in icing conditions.

Future research could include input constraints in the Model Predictive Control (MPC) framework and consider integrating ARS-LQR with Reinforcement Learning and Control Barrier Functions for more safety-critical situations.

Appendix A. Proof of Remark 1

Proof. Algebraic Riccati equation according to Eq. (54) is as follows:

$$\mathbf{P}_i'(\bar{\mathbf{A}}_i + \Lambda_i\mathbf{I}) + (\bar{\mathbf{A}}_i + \Lambda_i\mathbf{I})^{\mathrm{T}}\mathbf{P}_i' - \mathbf{P}_i'\bar{\mathbf{B}}_i\mathbf{R}_i^{-1}\bar{\mathbf{B}}_i^{\mathrm{T}}\mathbf{P}_i' + \mathbf{Q}_i = 0 \tag{A.1}$$

According to the standard LQR theory [34], all eigenvalues of $(\bar{\mathbf{A}}_i + \Lambda_i\mathbf{I} - \bar{\mathbf{B}}_i\bar{\mathbf{K}}_i)$ shall be located on the left side of the complex plane, i.e., $\bar{\mathbf{A}}_i + \Lambda_i\mathbf{I} - \bar{\mathbf{B}}_i\bar{\mathbf{K}}_i$ is Hurwitz.

$$S_1 = \frac{1}{2}\int_0^{\infty}(e^{2\Lambda_i t})(\bar{\mathbf{z}}_i^{\mathrm{T}}\mathbf{Q}_i\bar{\mathbf{z}}_i + \mu_i^{\mathrm{T}}\mathbf{R}_i\mu_i)dt, \quad \Lambda_i \succeq 0 \tag{A.2}$$

$$\mu_i = -\underbrace{\mathbf{R}_i^{-1}\bar{\mathbf{B}}_i^{\mathrm{T}}\mathbf{P}_i'}_{\bar{\mathbf{K}}_i}\ \bar{\mathbf{z}}_i \tag{A.3}$$

substituting μ_i from (A.3) into (A.2) yields the following:

$$S_1 = \frac{1}{2}\int_0^{\infty}(e^{2\Lambda_i t})[\bar{\mathbf{z}}_i^{\mathrm{T}}\underbrace{(\mathbf{Q}_i + \mathbf{P}_i'\bar{\mathbf{B}}_i\mathbf{R}_i^{-1}\bar{\mathbf{B}}_i^{\mathrm{T}}\mathbf{P}_i')}_{\bar{\mathbf{Q}}}\bar{\mathbf{z}}_i]dt, \quad \Lambda_i \succeq 0 \tag{A.4}$$

where $\bar{\mathbf{Q}}$ is positive semidefinite. For the state model below

$$\dot{\bar{\mathbf{z}}}_i = \bar{\mathbf{A}}_i\bar{\mathbf{z}}_i + \bar{\mathbf{B}}_i\mu_i \tag{A.5}$$

substituting μ_i from (A.3) into (A.5) yields the following:

$$\dot{\bar{\mathbf{z}}}_i = (\bar{\mathbf{A}}_i - \bar{\mathbf{B}}_i\bar{\mathbf{K}}_i)\bar{\mathbf{z}}_i \tag{A.6}$$

Clearly, the solution of (A.6) is given as

$$\bar{\mathbf{z}}_i = e^{(\bar{\mathbf{A}}_i - \bar{\mathbf{B}}_i\bar{\mathbf{K}}_i)t}\ \bar{\mathbf{z}}_0 \tag{A.7}$$

where $\bar{\mathbf{z}}_0$ is the initial state of $\bar{\mathbf{z}}_i$. By substituting (A.7) into (A.4), the following equation is obtained

$$S_1 = \frac{1}{2}\int_0^{\infty}(e^{2\Lambda_i t})\bar{\mathbf{z}}_0^{\mathrm{T}}[e^{(\bar{\mathbf{A}}_i - \bar{\mathbf{B}}_i\bar{\mathbf{K}}_i)^{\mathrm{T}}t}\bar{\mathbf{Q}}e^{(\bar{\mathbf{A}}_i - \bar{\mathbf{B}}_i\bar{\mathbf{K}}_i)t}]\bar{\mathbf{z}}_0 dt, \quad \Lambda_i \succeq 0 \tag{A.8}$$

As $e^{2\Lambda_i t}$ is a scalar function, then (A.8) can be written as follows:

$$S_1 = \frac{1}{2}\int_0^{\infty} \bar{\mathbf{z}}_0^T [e^{(\bar{\mathbf{A}}_i+\Lambda_i \mathbf{I}-\bar{\mathbf{B}}_i\bar{\mathbf{K}}_i)^T t}\bar{\mathbf{Q}}e^{(\bar{\mathbf{A}}_i+\Lambda_i \mathbf{I}-\bar{\mathbf{B}}_i\bar{\mathbf{K}}_i)t}]\bar{\mathbf{z}}_0 dt, \quad \Lambda_i \succeq 0 \tag{A.9}$$

With a vector defined as

$$y = \bar{\mathbf{Q}}^{\frac{1}{2}} e^{(\bar{\mathbf{A}}_i+\Lambda_i \mathbf{I}-\bar{\mathbf{B}}_i\bar{\mathbf{K}}_i)t}\bar{\mathbf{z}}_0 \tag{A.10}$$

the cost function can be written as

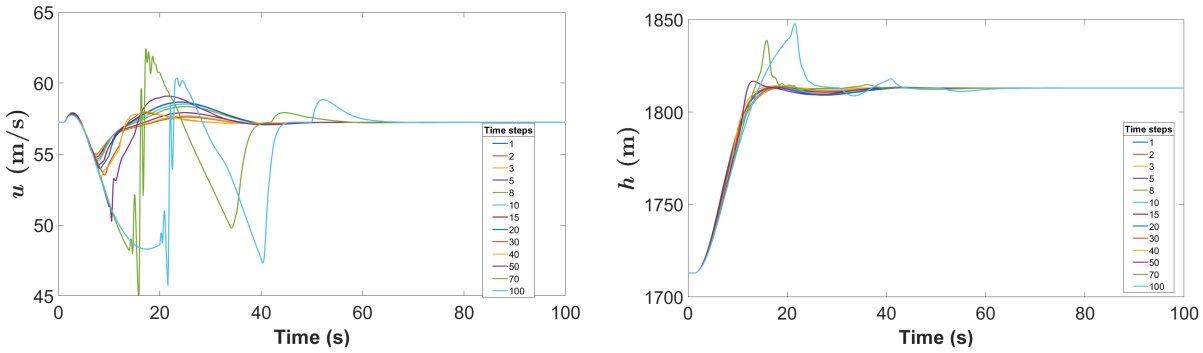
$$S_1 = \frac{1}{2}\int_0^{\infty} ||y||^2 dt \tag{A.11}$$

where $\|y\|$ denotes the 2-norm of y, and it becomes zero as t tends to infinity because $(\bar{\mathbf{A}}_i + \Lambda_i \mathbf{I} - \bar{\mathbf{B}}_i\bar{\mathbf{K}}_i)$ is Hurwitz. Consequently, S_1 is finite with a sensible choice of Λ_i, such as the one in Eq. (53).

Appendix B.

Appendix B.1: Effects of Time Steps on Controller's Performance

Effects of time steps on the controller's performance are shown in Figures B1(a) and (b), and cost values calculated for different time steps are presented in Table 9.



(a) (b)

Fig. B1 Comparison of forward velocity and altitude variations at different time steps

Table 9 Comparison of cost function values based on different linearized models at different time steps

Time Steps	1	2	3	5	8	10
Cost Values	61742	59904	59900	60218	60811	61436

	15	20	30	40	50	70	100
	60510	61249	62377	62843	64145	68051	71772

Appendix B.2: Weight Update of RBFN

RBFN weights are updated to capture uncertainties brought by icing, as shown in Fig. B2, and RBF network specifications are shown in Table 10.

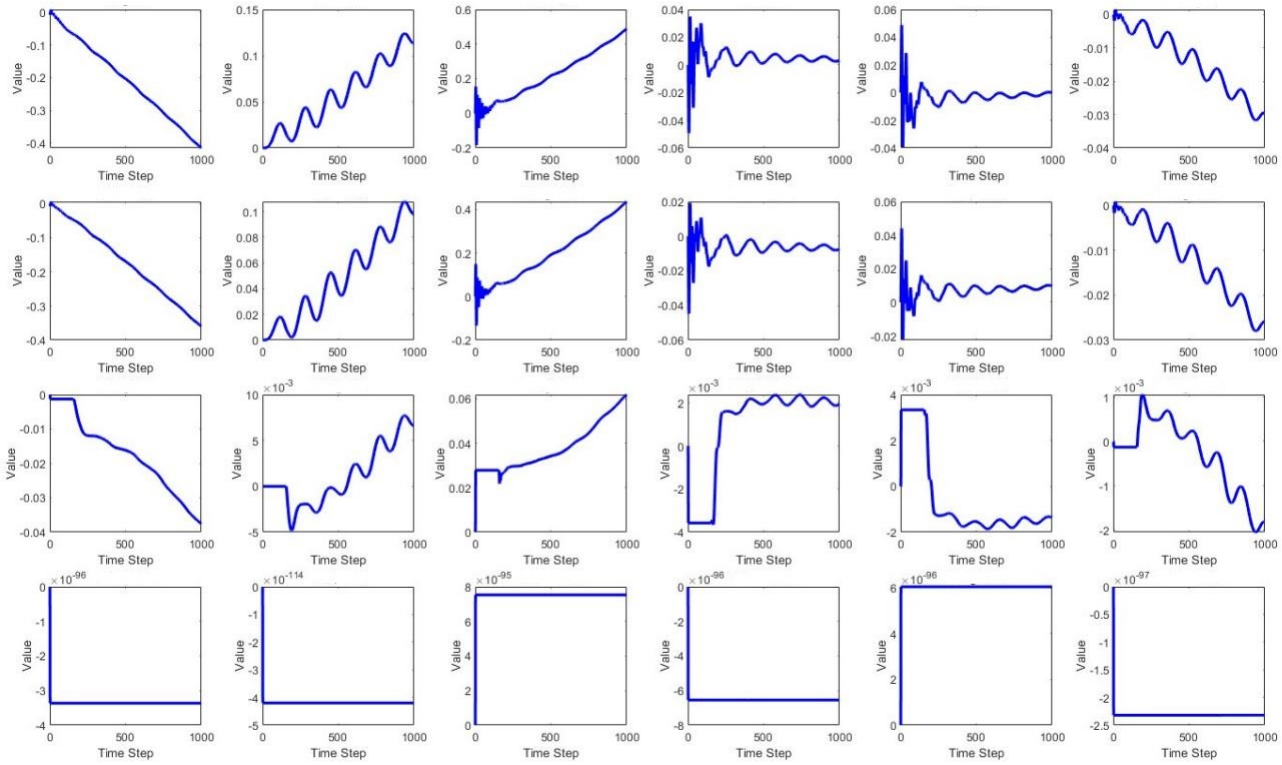


Fig. B2 Weight Update of RBF Networks Used in Neuro-Adaptive Flight Control

Table 10 RBFN Specifications

Parameter	Description
Network Type	Radial Basis Function Network (RBFN)
Number of Hidden Layers	1
Input Dimensions	13 (9 states, 4 control inputs)
Centers of Basis Functions	Calculated using online k-means clustering
Activation Function (Hidden Layer)	Gaussian (Radial Basis Functions)
Output Activation Function	Linear
Number of Neurons (Hidden Layer)	4 (same as number of centers for clustering)
Number of Neurons (Output Layer)	6 (Number of uncertainties in $\dot{u}, \dot{v}, \dot{w}, \dot{p}, \dot{q}, \dot{r}$ equations)
Weight Dimensions	4×6 (4: number of clustering centers, 6: number of uncertainties in equations)
Training Algorithm	Lyapunov function-based weight updates

Acknowledgments

This research was supported by the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT, Republic of Korea (RS-2024-00397400). Some parts of this article have been presented at the 2023 AIAA SciTech Forum (Paper Number 1636), Maryland, USA.

Competing interests

The authors declare none.

References

[1] Reveley, M. S., Briggs, J. L., Evans, J. K., Sandifer, C. E., and Jones, S. M., "Causal factors and adverse conditions of aviation accidents and incidents related to integrated resilient aircraft control," Tech. rep., 2010.

[2] Heinrich, A., Ross, R., Zumwait, G., Provorse, J., Padmanabhan, V., Thompson, J., and Riley, J., *Aircraft Icing Handbook*, Vol. 1 of 3, FAA Technical Monitor, E.E. Schlatter, 1991.

[3] 1959-2008, W. O., "Statistical Summary of Commercial Jet Airplane Accidents (BOEING)," Tech. rep., 2009.

[4] Board, N. T. S., "Aircraft Accident Report: Loss of Control on Approach," Operating as Continental Connection Flight 3407, February 2009. Bombardier DHC-8-400, N200WQ.

[5] Gingras, D. R., Barnhart, B. P., Ranaudo, R. J., Ratvasky, T. P., and Morelli, E. A., "Envelope Protection for In-Flight Ice Contamination," NASA Tech. Memo NASA/TM—2010-216072, Bihrle Applied Res. Inc.; UTSI; NASA Glenn RC; NASA Langley RC, Feb 2010. http://doi.org/10.2514/6.2009-1458.

[6] Deiler, C., and Kilian, T., "Dynamic aircraft simulation model covering local icing effects," *CEAS Aeronautical Journal*, Vol. 9, No. 3, 2018, pp. 429–444. http://doi.org/10.1007/s13272-018-0291-6.

[7] Lynch, F. T., and Khodadoust, A., "Effects of ice accretions on aircraft aerodynamics," *Progress in Aerospace Sciences*, Vol. 37, No. 8, 2001, pp. 669–767. http://doi.org/10.1016/S0376-0421(01)00018-5.

[8] Raj, L. P., Lee, L., and Myong, R. S., "Ice accretion and aerodynamic effects on a multi-element airfoil under SLD icing conditions," *Aerospace Science and Technology*, Vol. 85, 2019, pp. 320–333. http://doi.org/10.1016/j.ast.2018.12.017.

[9] Raj, L. P., Yee, K., and Myong, R. S., "Sensitivity of ice accretion and aerodynamic performance degradation to critical physical and modeling parameters affecting airfoil icing," *Aerospace Science and Technology*, Vol. 98, 2020, p. 105659. http://doi.org/10.1016/j.ast.2019.105659.

[10] Sengupta, B., Prince Raj, L., Cho, M. Y., Son, C., Yoon, T., Yee, K., and Myong, R. S., "Computational simulation of ice accretion and shedding trajectory of a rotorcraft in forward flight with strong rotor wakes," *Aerospace Science and Technology*, Vol. 119, 2021, p. 107140. http://doi.org/10.1016/j.ast.2021.107140.

[11] Lampton, A., and Valasek, J., "Prediction of icing effects on the lateral/directional stability and control of light airplanes," *Aerospace Science and Technology*, Vol. 23, 2012, pp. 305–311. http://doi.org/10.1016/j.ast.2011.08.005.

[12] Cao, Y., and Yuan, K., "Aircraft flight characteristics in conditions of windshear and icing," *The Aeronautical Journal*, Vol. 111, 2007, pp. 41–49. http://doi.org/10.1017/S0001924000001743.

[13] Cao, Y., Tan, W., and Wu, Z., "Aircraft icing: An ongoing threat to aviation safety," *Aerospace Science and Technology*, Vol. 75, 2018, pp. 353–385. http://doi.org/10.1016/j.ast.2017.12.028.

[14] Sharma, V., Voulgaris, P. G., and Frazzoli, E., "Aircraft autopilot analysis and envelope protection for operation under icing conditions," *Journal of Guidance, Control, and Dynamics*, Vol. 27, No. 3, 2004, pp. 454–465. http://doi.org/10.2514/1.1214.

[15] Jacobson, S. R., "Aircraft loss of control causal factors and mitigation challenges," *AIAA Guidance, Navigation, and Control Conference, Torotno, Ontario, Canada*, 2010. http://doi.org/10.2514/6.2010-8007.

[16] Ratvasky, T. P., Barnhart, B. P., and Lee, S., "Current methods modeling and simulating icing effects on aircraft performance, stability, control," *Journal of Aircraft*, Vol. 47, No. 1, 2010, p. 201–211. http://doi.org/10.2514/1.44650.

[17] Ratvasky, T., Van Zante, J., and Riley, J., "NASA/FAA tailplane icing program overview," *AIAA Paper No. 99–0370, Reno, NV, USA*, 1999, pp. 11–14. http://doi.org/10.2514/6.1999-370.

[18] Ranaudo, R., Reehorst, A., Bond, T., Batterson, J., and O'Mara, T., "Determination of longitudinal aerodynamic derivatives using flight data from an icing research aircraft," *AlAA 27th Aerospace Sciences Meeting, Reno, NV, USA*, 1989. http://doi.org/10.2514/6.1989-754.

[19] Ratvasky, T., and Ranaudo, R., "Icing effects on aircraft stability and control determined from flight data-Preliminary results," *NASA Technical Memorandum 105977, AlAA 31st Aerospace Sciences Meeting, Reno, NV, USA*, 1993. http://doi.org/10.2514/6.1993-398.

[20] Reehorst, A., Chung, J., Potapczuk, M., and Choo, Y., "Study of icing effects on performance and controllability of an accident aircraft," *Journal of Aircraft*, Vol. 37, No. 2, 2000, pp. 253–259. http://doi.org/10.2514/2.2588.

[21] Whalen, E., and Bragg, M., "Aircraft Characterization in Icing Using Flight Test Data," *Journal of Aircraft*, Vol. 42, No. 3, 2005. http://doi.org/10.2514/1.11198.

[22] Lampton, A., and Valasek, J., "Prediction of Icing Effects on the Stability and Control of Light Airplanes," *AIAA Atmospheric Flight Mechanics Conference and Exhibit, San Francisco, California*, 2005. http://doi.org/10.2514/6.2005-6219.

[23] Deiler, C., "Aerodynamic modeling, system identification, and analysis of iced aircraft configurations," *Journal of Aircraft*, Vol. 55, No. 1, 2018, pp. 145–161. http://doi.org/10.2514/1.C034390.

[24] Deiler, C., "Flight Characteristics with Different Supercooled Large Droplet Ice Configurations," *The Aeronautical Journal*, Vol. 126, 2022, pp. 848–865. http://doi.org/10.1017/aer.2021.98.

[25] Harno, H. G., Kim, Y., Moon, J., and Bang, H., "Zonotopic Kalman filtering for stability augmentation and flight envelope estimation," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 235, No. 15, 2021, pp. 2288–2298. http://doi.org/10.1177/0954410021995.

[26] Harno, H. G., and Kim, Y., "Flight envelope estimation for helicopters under icing conditions via the zonotopic reachability analysis," *Aerospace Science and Technology*, Vol. 102, 2020, p. 105859. http://doi.org/10.1016/j.ast.2020.105859.

[27] Dutoi, B., Richards, N., Gandhi, N., Ward, D., and Leonard, J., "Hybrid robust control and reinforcement learning for optimal upset recovery," *AIAA Guidance, Navigation and Control Conference and Exhibit, Honolulu, Hawaii*, 2008. http://doi.org/10.2514/6.2008-6502.

[28] Ying, S.-b., Ge, T., and Ai, J.-l., "H_∞ parameter identification and H_2 feedback control synthesizing for in-flight aircraft icing," *Journal of Shanghai Jiaotong University (Science)*, Vol. 18, No. 3, 2013, pp. 317–325. http://doi.org/10.1007/s12204-013-1401-6.

[29] Hossain, K., Sharma, V., Bragg, M., and Voulgaris, P., "Envelope protection and control adaptation in icing encounters," *41st Aerospace Sciences Meeting and Exhibit, Reno, Nevada*, 2003. http://doi.org/10.2514/6.2003-25.

[30] Hartley, E., "Predictive control with parameter adaptation to achieve α-protection in the RECONFIGURE benchmark in the presence of icing," *ScienceDirect, IFAC-Papers Online*, Vol. 48, No. 23, 2015, pp. 172–177. http://doi.org/10.1016/j.ifacol.2015.11.279.

[31] Morton, B., Enns, D., and Zhang, B.-Y., "Stability of dynamic inversion control laws applied to non-linear aircraft pitch-axis models," *International Journal of Control*, Vol. 63, No. 1, 1996, pp. 1–25. http://doi.org/10.1080/00207179608921828.

[32] Wei, Y., Xu, H., and Xue, Y., "Adaptive neural networks-based dynamic inversion applied to reconfigurable flight control and envelope protection under icing conditions," *IEEE Access*, Vol. 8, 2020, pp. 11577–11594. http://doi.org/10.1109/ACCESS.2020.2964728.

[33] Padhi, R., Rao, N., Goyal, S., and Tripathi, A., "A model-following neuro-adaptive approach for robust control of high performance aircrafts," *International Conference on Advances in Control and Optimization of Dynamical Systems, Bangalore, India*, 2010.

[34] Lavretsky, E., and Wise, K. A., *Robust and Adaptive Control*, Springer, New York, 2012. http://doi.org/10.1007/978-1-4471-4396-3.

[35] Anderson, B. D., and Moore, J. B., *Optimal Control*, Department of Systems Engineering, Australian National University, Canberra, 1989.

[36] Azimi, V., Farzan, S., and Hutchinson, S., "A Robust Time-Varying Riccati-Based Control for Uncertain Nonlinear Dynamical Systems," *Journal of Dynamic Systems, Measurement, and Control*, Vol. 144, No. 10, 2022, pp. 101001–1. http://doi.org/10.1115/1.4054884.

[37] Farrara, A. D., Linares, R., and Doerr, B., "An Iterative LQR Method for Addressing Model Uncertainty in the Planetary Entry Problem," *Proceedings of the 73rd International Astronautical Congress (IAC)*, International Astronautical Federation (IAF), 2022, pp. 1–10.

[38] Tedrake, R., *Underactuated robotics: Algorithms for walking, running, swimming, flying, and manipulation (MIT course)*, 2021.

[39] Beheshti.Rabiei, A., Kim, Y., and Myong, R., "Adaptive Robust Servo LQR Control of Aircraft in Icing Condition," *AIAA 2023-1636, SciTech Forum*, 2023. http://doi.org/10.2514/6.2023-1636.

[40] Beard, R. W., and McLain, T. W., *Small Unmanned Aircraft: Theory and Practice*, Princeton University Press, 2012. http://doi.org/10.2514/1.61067.

[41] Elkhatem, A. S., and Engin, S. N., "Robust LQR and LQR-PI control strategies based on adaptive weighting matrix selection for a UAV position and attitude tracking control," *Alexandria Engineering Journal*, Vol. 61, No. 8, 2022, pp. 6275–6292. http://doi.org/10.1016/j.aej.2021.11.057.

[42] Dougl, J., and Athans, M., "Robust LQR Control for the Benchmark Problem," *Proceedings of the 1991 American Control Conference*, Boston, MA, USA, 1991. http://doi.org/10.1016/j.isatra.2017.03.023.

[43] Sun, H., Chen, Y.-H., and Zhao, H., "Adaptive robust control methodology for active roll control system with uncertainty," *Nonlinear Dynamics*, Vol. 92, 2018, pp. 359–371. http://doi.org/10.1007/s11071-018-4060-1.

[44] Rawlings, J. B., and Mayne, D. Q., "Model predictive control: theory, computation, and design," *Nob Hill Publishing*, 2017.

[45] Hovakimyan, N., Cao, C., and Hu, Z., *L1 Adaptive Control Theory: Guaranteed Robustness with Fast Adaptation*, SIAM, 2010. http://doi.org/10.1109/MCS.2011.941837.

[46] Nelson, R. C., *Flight Stability and Automatic Control*, WCB/McGraw Hill, New York, 1998. http://doi.org/10.1017/S0001924000022120.

[47] Pokhariyal, D., Bragg, M., Hutchison, T., and Merret, J., "Aircraft flight dynamics with simulated ice accretion," *39th Aerospace Sciences Meeting and Exhibit, Reno, NV, USA*, 2001. http://doi.org/10.2514/6.2001-541.

[48] Bragg, M., Perkins, W., Sarter, N., Basar, T., Voulgaris, P., Gurbacki, H., Melody, J., and McCray, S., "An interdisciplinary approach to inflight aircraft icing safety," *AIAA Paper No. 98–0095, Reno, NV, USA*, 1998, pp. 12–15. http://doi.org/10.2514/6.1998-95.

[49] Rabiei Beheshti, A., Kim, Y., and Myong, R. S., "Adaptive Neural Nonlinear Dynamic Inversion Control of Aircraft in Icing Conditions," *23rd International Conference on Control, Automation and Systems (ICCAS)*, IEEE, 2023. http://doi.org/10.23919/ICCAS59377.2023.10316917.

[50] Sundararajan, N., Saratchandran, P., and Li, Y., "Fully Tuned Radial Basis function Neural Networks For Flight Control," 2001. http://doi.org/10.1007/978-1-4757-5286-1.

[51] Cohen-Addad, V., Guedj, B., Kanade, V., and Rom, G., "Online k-means Clustering," *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics (AISTATS)*, Vol. PMLR 130, 2021, pp. 1126–1134. http://doi.org/10.48550/arXiv.1909.06861.

[52] Padhi, R., Unnikrishnan, N., and Balakrishnan, S., "Model following Neuro-Adaptive control design for non square, non affine nonlinear systems," *IET Control Theory and Applications*, Vol. 1, 2007, pp. 1650–1661. http://doi.org/10.1049/iet-cta:20060364.

[53] Plessen, M., and Bemporad, A., "Reference trajectory planning under constraints and path tracking using linear time-varying model predictive control for agricultural machines," *Biosystems Engineering*, Vol. 153, 2017, pp. 28–41. http://doi.org/10.1016/j.biosystemseng.2016.10.019.

[54] Padhi, R., "Optimal Control, Guidance and Estimation, 'Linear Quadratic Regulator (LQR) Design - 3'," , 2013. `https://nptel.ac.in/courses/101108057`.

[55] Wiese, D. P., Annaswamy, A. M., Muse, J. A., and Bolender, M. A., "Adaptive control of a generic hypersonic vehicle," *AIAA Guidance, Navigation, and Control (GNC) Conference, Boston, MA*, 2013. http://doi.org/10.2514/6.2013-4514.

[56] Wiese, D. P., Annaswamy, A. M., Muse, J. A., Bolender, M. A., and Lavretsky, E., "Sequential loop closure based adaptive autopilot design for a hypersonic vehicle," *AIAA Guidance, Navigation, and Control Conference, San Diego, California, USA*, 2016. http://doi.org/10.2514/6.2016-1379.

[57] MacDonald, R., Garelick, M., and Haas, J., "Real-Time Simulation Program for De Havilland (Canada) 'Buffalo' and 'Twin Otter STOL Transports," Tech. rep., Technical note, Report No.Dot-TSC-FAA, 1972.

[58] Mukherjee, J., and Pieper, J., "Adaptive LQR gain sheduling applied to an experimental OHS aircraft," *AIAA Guidance, Navigation, and Control Conference and Exhibit, Dever, CO, USA*, 2000. http://doi.org/10.2514/6.2000-3944.