Event-Triggered Adaptive Dynamic Programming for an Aeroservoelastic System An Experimental Study

Wang, Xuerui; Sun, Bo; Sodja, Jurij; van Kampen, Erik Jan

DOI
10.2514/1.G009257

Publication date
2025

Document Version
Final published version

Published in
Journal of Guidance, Control, and Dynamics

JOURNAL OF GUIDANCE, CONTROL, AND DYNAMICS
Vol. 48, No. 12, December 2025

Technical Notes

Event-Triggered Adaptive Dynamic Programming for an Aeroservoelastic System: An Experimental Study

Xuerui Wang,* Bo Sun,† Jurij Sodja,‡ and Erik-Jan van Kampen§
Delft University of Technology, 2629 HS Delft, The Netherlands

https://doi.org/10.2514/1.G009257

I. Introduction

AEROSERVOELASTICITY, an interdisciplinary field, delves into the interactions among aerodynamic forces, elastic structural deformations, and the control systems of aircraft and other air-operating structures [1,2]. This field has attracted increasing interest, as it is one of the key pieces of the puzzle of maximizing structural air-operating efficiency without compromising safety. Overlooking aeroservoelasticity can lead to designs that have efficient appearance (high aspect-ratio) but are actually sensitive to atmospheric disturbances, such as gusts and turbulence. Such oversights may also trigger destructive structural vibrations, culminating in catastrophic instabilities and failures [3–5].

Aeroservoelastic systems are inherently nonlinear, time varying, and underactuated. The nonlinearities arise from factors such as structural geometry, aerodynamics, and mechanical imperfections [2,6,7]. Accurately modeling these dynamics often requires the use of costly computational fluid dynamics coupled with computational structural dynamics. In the literature, it is common to simplify the dynamics into a lower-order linear time-invariant (LTI) or linear parameter-varying system [8–10]. However, even after being augmented with linear robustification techniques (such as $\mathcal{H}_\infty$), the resulting performance and stability robustness can become either overly or marginally conservative [11]. A recent study also reveals that the classical linear quadratic Gaussian approach, which is designed using a lower-order LTI model, is unable to stabilize the original nonlinear aeroservoelastic dynamics [12]. Although nonlinear control methods do exist, most of them rely on the inversion of the control effectiveness matrix as a critical step [13,14]. When applied to underactuated aeroservoelastic systems, this inversion step makes the closed-loop stability heavily dependent on the inherent stability of the internal dynamics [15]. This reliance renders them less suitable for addressing instability issues of aeroservoelastic systems such as flutter.

Received 10 April 2025; accepted for publication 27 June 2025; published online 30 July 2025.

*Assistant Professor, Faculty of Aerospace Engineering; X.Wang-6@tudelft.nl. Senior Member AIAA (Corresponding Author).
†Ph.D., Faculty of Aerospace Engineering; hello.bo.sun@outlook.com.
‡Assistant Professor, Faculty of Aerospace Engineering; J.Sodja@tudelft.nl. Member AIAA.
§Associate Professor, Faculty of Aerospace Engineering; E.vanKampen@tudelft.nl. Member AIAA.

Nonlinear optimal control is a promising approach for addressing the challenges in aeroservoelastic system control design. An essential component of nonlinear optimal control involves solving the Hamilton–Jacobi–Bellman (HJB) equation. Nonetheless, deriving an analytical solution becomes particularly challenging in the context of nonlinear, time-varying systems. In contrast, adaptive dynamic programming (ADP), which hybridizes principles from control theory with reinforcement learning (RL), can generate numerical solutions for the HJB equation [16]. Moreover, ADP addresses the "curse of dimensionality" in conventional RL, enabling it to guide an agent toward optimal and adaptive behavior, even in the absence of precise models for the agent's dynamics or its environment [17,18]. Furthermore, for real-world continuous time systems, ADP with a single critic network (SCN) architecture can avoid the actor network-induced approximation error in conventional RL and reduce computational loads [19–21].

Although ADP is inherently a continuous-time control methodology well suited for the continuous-time nature of aeroservoelastic systems, in practice, the control law still needs to be processed by a digital computer. Furthermore, the digital computer normally communicates with servos in predetermined and fixed time steps. This is referred to as the time-triggered approach. However, it is less efficient for the disturbance rejection challenge of aeroservoelastic systems for the following reasons:

1) The high-dimensional (theoretically infinite-dimensional) nature of the system imposes prohibitive computational demands if updating at every time step.

2) A fixed time-triggered approach struggles to provide timely and adequate responses to sudden disturbances such as gusts, impairing system performance and stability.

3) After a sudden gust has passed the system, the persistence in fixed time-triggered control results in unnecessary data transmissions (from the controller to actuators) and control actions, thereby wasting computational, network, and servo resources.

In recognition of these challenges, our previous research in [22] pioneered the concept of event-triggered ADP for addressing nonlinear, uncertain, and underactuated aeroservoelastic control problems. Event-triggered ADP is a control framework that integrates adaptive dynamic programming with event-triggered mechanisms to reduce computational and communication load by updating control policies only when necessary, while maintaining stability and performance [23,24]. The aforementioned challenges associated with time-triggered ADP are fundamentally addressed by optimizing control signal updates based on specific events defined through triggering conditions. Moreover, the Zeno phenomenon, a scenario in which an infinite number of events or actions occur within a finite time interval [25–27], is theoretically avoided in our algorithm design. Furthermore, the input constraints are theoretically handled by employing a nonquadratic cost function with the corresponding triggering condition derived without relying on the Lipschitz assumption for the inverse hyperbolic tangent function, thereby addressing the limitations present in [25,28]. Last but not least, many existing event-triggered ADP algorithms in the literature rely on an initially admissible control policy, meaning that at the onset of the control horizon, a control law must be available that satisfies all predefined state and control constraints while ensuring that the system's state trajectories remain within the allowable bounds throughout the control process. This dependency is also theoretically relaxed in our previous work based on the Lyapunov theory.

Despite the theoretical contributions of [22], many real-world challenges are overlooked, including but not limited to, delay, measurement noise, external disturbances, and various hardware constraints. In particular, whether the algorithm can still avoid the Zeno phenomenon in the presence of real-world disturbances and

noise remains an open question in the field. Furthermore, only a single stabilization cycle is evaluated through numerical simulations in [22], whereas the robustness of the algorithm to consecutive excitations caused by unanticipated, real-world external disturbances remains unknown. This is a critical aspect for demonstrating the algorithm's ability to adapt and learn effectively from past experiences online. Last but not least, the tradeoff between rapid response to sudden gusts and improving resource efficiency when gusts gradually dissipate has not been addressed in [22] or elsewhere in the literature.

This Technical Note presents real-world wind-tunnel experimental investigation of an event-triggered ADP algorithm tailored to addressing the challenges of stabilization and consecutive external disturbance rejection in nonlinear, uncertain, and underactuated aeroservoelastic systems.

As a preliminary attempt, the algorithm presented in [22] was directly applied to a real-world aeroservoelastic system. However, it was observed that although the nonlinear hyperbolic tangent function (also adopted in [25,28]) can theoretically guarantee input constraints, it also increases the algorithm's sensitivity to parameter tuning and reduces robustness to input delays. This is a phenomenon that has not been exposed previously. Therefore, in this Note, the algorithm is adapted by removing the nonlinear hyperbolic tangent function and instead incorporating a hard input constraint using a nonlinear saturation function that is decoupled from the online adaptation loop. This modification was found to effectively enhance the practicality of the event-triggered ADP algorithm. The theoretical proof and analysis of the algorithm are adapted accordingly in this Note.

Subsequently, the practical capabilities of the adapted algorithm in terms of online adaptation, rejection of consecutive disturbances, and avoidance of the Zeno phenomenon are thoroughly evaluated under real-world experiments in the presence of mechanical and measurement imperfections. Moreover, the effectiveness of the event-triggering mechanism is quantitatively assessed through experimental comparisons between the proposed algorithm and its conventional time-triggered counterpart. Finally, the Note discusses the tradeoff between rapid responsiveness to sudden gusts and improved resource efficiency as gusts gradually dissipate.

In the remainder of this Note, the control challenge is formulated in Sec. II. The algorithm is developed in Sec. III. Section IV presents the real-world experiment setup, followed by the experiment results and discussions in Sec. V. Section VI concludes this Note.

II. Problem Description

During the aircraft conceptual design phase, to facilitate efficient analysis and iterations on structural integrity, aerodynamic performance, and control effectiveness, the three-dimensional wing is normally abstracted to an aeroservoelastic typical wing section, as illustrated in Fig. 1. Structural, inertia, and aerodynamic properties are defined for a cross-sectional airfoil with heave and pitch degrees of freedom. It is also equipped with trailing-edge active control devices, such as an aileron and/or a spoiler. Considering unsteady aerodynamics and nonlinearities, an aeroservoelastic typical wing section is modeled as follows:

$$\dot{x}(t) = f(x(t)) + g(x(t))u(t) + \mathrm{d}(t) \tag{1}$$

where $x = [\dot{h}, h, \dot{\theta}, \theta, \dot{\beta}, \beta, z_1, \ldots, z_k]^\top \in \Omega \subset \mathbb{R}^n$ is the state vector containing the structural degrees of freedom and k unsteady aerodynamic lag states. $u(t) = \beta_{\mathrm{cmd}} \in \Omega_u \subset \mathbb{R}^m$ is the trailing-edge control surface command. Regarding an aeroservoelastic system, $f(\cdot)$ is generally Lipschitz continuous in Ω with an equilibrium point $x(t) = 0$. $\mathrm{d}(t) \in \mathbb{R}^n$ is the unknown external disturbance perturbation.

The main control objective is gust disturbance rejection, in other words, using the control input β_{cmd} to alleviate the system oscillations in the presence of unknown gusts without knowing an accurate system dynamic model, while reducing the computational and communication burden.

Because the number of inputs is less than the number of degrees of freedom to be controlled, the considered problem is underactuated. System nonlinearities can originate from both aerodynamic and structural effects. Strong gust disturbances can induce aerodynamic flow separation and vortex shedding, leading to nonlinear aerodynamic forces. The structural dynamics, modeled using spring-based suspension elements for plunge and pitch motions, can exhibit nonlinear stiffness under large motions. In addition, common mechanical imperfections such as actuator dead zones, backlash, and friction further introduce nonlinearity.

Uncertainty arises from discrepancies between the control-oriented model and the physical system. Although key parameters such as mass, stiffness, and geometry can be estimated, they are often subject to modeling errors. Unmodeled dynamics, such as flow separation, actuator nonlinearities, and aerodynamic interactions between control surfaces, also contribute to the overall uncertainty of the system.

III. Control Algorithm Development

A. Optimal Control Design

For system (1), an infinite-horizon cost function is defined as

$$J(x) = \int_t^\infty (x(\tau)^\top Q x(\tau) + u(x(\tau))^\top R u(x(\tau)))\,\mathrm{d}\tau \tag{2}$$

where $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{m\times m}$ are positive semidefinite matrices. Denote $U(x, u(x)) = x^\top Q x + u^\top R u$ as the utility function.

Select an admissible law $u(x) \in \mathcal{A}(\Omega)$. Consequently, the Hamiltonian is expressed as

$$H(x, u(x), \nabla J(x)) = \nabla J^\top(x)[g(x)u(x) + f(x)] + U(x, u(x)) \tag{3}$$

When satisfying $\min_{u(x)\in\mathcal{A}(\Omega)} H(x, u(x), \nabla J^*(x)) = 0$, the cost in Eq. (2) reaches its optimal value as

$$J^*(x) = \min_{u(x)\in\mathcal{A}(\Omega)} \int_t^\infty U(x(\tau), u(x(\tau)))\,\mathrm{d}\tau \tag{4}$$

By satisfying $\partial H(x, u(x), \nabla J(x))/\partial u(x) = 0$, the corresponding optimal solution is

$$u^*(x) = \arg\min_{u(x)\in\mathcal{A}(\Omega)} H(x, u(x), \nabla J^*(x)) = -\frac{1}{2} R^{-1} g^\top(x) \nabla J^*(x) \tag{5}$$

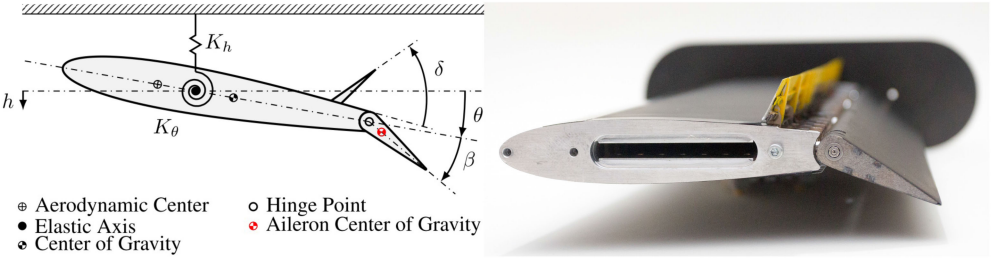

Fig. 1 A typical aeroservoelastic wing section.

Substituting Eq. (5) into the HJB equation produces

$$H(x,u^*(x),\nabla J^*(x)) = -\frac{1}{4}\nabla J^{*\top}(x)g(x)R^{-1}g^\top(x)\nabla J^*(x) + x^\top Qx + \nabla J^{*\top}(x)f(x) = 0 \tag{6}$$

When $J^*(0)=0$, this results in $H(x,u^*(x),\nabla J^*(x))=0$.

B. Event Triggering

To design the event-triggered mechanism, a sequence of triggering instants $\{s_k\}_{k=0}^{\infty}$ is defined, where $s_k < s_{k+1}$ for $k \in \mathbb{N}$. For all $t \in [s_k, s_{k+1})$, the sampled-data output is $x(s_k) \triangleq x_k$, leading to the following gap function:

$$e_k(t) = x_k - x, \quad \forall\, t \in [s_k, s_{k+1}) \tag{7}$$

Let $e_k(t)$ be denoted simply as e_k from now on. When a specific triggering condition is met, then the algorithm updates the event-triggered state vector while resetting e_k to zero. At each triggering moment (rather than at every time sample), the algorithm updates the state-feedback law $u(x(s_k)) = u(x_k)$ accordingly. By implementing a zero-order hold, the control sequence $\{u(x_k)\}_{k=0}^{\infty}$ effectively becomes a piecewise constant signal that remains unchanged over the interval $[s_k, s_{k+1})$ for all $k \in \mathbb{N}$. Given the control signal $u(x_k)$, the system in Eq. (1) can then be represented as

$$\dot{x} = g(x)u(x+e_k) + f(x), \quad \forall\, t \in [s_k, s_{k+1}) \tag{8}$$

In the event-triggered scheme, the law in Eq. (5) becomes

$$u^*(x_k) = -\frac{1}{2}R^{-1}g^\top(x_k)\nabla J^*(x_k) \tag{9}$$

Regarding system (1), with the infinite-horizon cost function (2), the triggering condition is defined as $\|e_k\|^2 > \|e_T\|^2$. The threshold $\|e_T\|^2$ will be determined later.

C. Adaptive Critic Learning

Because the HJB equation is nonlinear (6), finding an analytical solution is challenging. Instead, this Note designs an ADP algorithm with an SCN architecture. Reconstruct the optimal cost function as $J^*(x) = w_c^\top \sigma_c(x) + \varepsilon_c(x)$, approximated with l_c neurons. $w_c \in \mathbb{R}^{l_c}$ is the optimal weight vector in the ideal case. The activation function is $\sigma(x)_c \in \mathbb{R}^{l_c}$ with the approximation error $\varepsilon_c(x) \in \mathbb{R}$. As a result, the optimal cost gradient is

$$\nabla J^*(x) = \nabla \varepsilon_c(x) + \nabla \sigma_c^\top(x) w_c \tag{10}$$

In reality, the ideal value of w_c is unknown, thus it is replaced by its estimated value $\hat{w}_c \in \mathbb{R}^{l_c}$ as

$$\hat{J}^*(x) = \hat{w}_c^\top \sigma_c(x), \quad \nabla \hat{J}^*(x) = \nabla \sigma_c^\top(x)\hat{w}_c \tag{11}$$

Considering Eq. (10), the event-triggered optimal policy (9) becomes

$$u^*(x_k) = -\frac{1}{2}R^{-1}g^\top(x_k)(\nabla \sigma_c^\top(x) w_c + \nabla \varepsilon_c(x)) \tag{12}$$

Accordingly, its approximation becomes

$$\hat{u}(x_k) = -\frac{1}{2}R^{-1}g^\top(x_k)\nabla \sigma_c^\top(x)\hat{w}_c \tag{13}$$

Substituting Eq. (12) into the Hamiltonian (3) leads to

$$H(x,u^*(x_k),w_c) = w_c^\top \nabla \sigma_c(x)[g(x)u^*(x_k)+f(x)] + x^\top Qx + u^{*\top}(x_k)Ru^*(x_k) \triangleq e_{cH} \tag{14}$$

where $e_{cH} = -\nabla \varepsilon_c^\top(x)[g(x)u^*(x_k)+f(x)]$ is the corresponding artificial neural network (ANN) approximation residual. As its counterpart, using Eq. (13), the Hamiltonian approximate is

$$\hat{H}(x,u^*(x_k),\hat{w}_c) = \hat{w}_c^\top \nabla \sigma_c(x)[g(x)u^*(x_k)+f(x)] + x^\top Qx + u^{*\top}(x_k)Ru^*(x_k) \triangleq e_c \tag{15}$$

Define the critic error as $\tilde{w}_c = w_c - \hat{w}_c$ and combining Eq. (14) with Eq. (15), we obtain

$$e_c = e_{cH} - \tilde{w}_c^\top \nabla \sigma_c(x)[g(x)u^*(x_k)+f(x)] \tag{16}$$

To train the critic network, the objective is to minimize $E_c = (1/2)e_c^\top e_c$. In [28], this minimization is achieved by gradient descent:

$$\dot{\hat{w}}_{c,1} = -\eta_c \frac{\partial e_c}{\partial \hat{w}_c} e_c = -\eta_c \phi e_c \tag{17}$$

where $\phi = \nabla \sigma_c(x)(g(x)\hat{u}(x_k)+f(x))$ and with $\eta_c > 0$ representing the learning rate.

To relax the requirement on initial admissible control, inspired by [25], a stabilization term is added to enhance the updating of ANN weights.

Assumption 1 [29]: Consider system (1) using the event-triggered optimal algorithm in Eq. (9) with the cost function (2) and a continuously differentiable $J_s(x)$ satisfying

$$\dot{J}_s^*(x) = \nabla J_s^\top(x)[g(x)u^*(x_k)+f(x)] < 0 \tag{18}$$

Then, the following inequality holds with a positive definite matrix $M \in \mathbb{R}^{n\times n}$:

$$\dot{J}_s^*(x) = -\nabla J_s^\top(x) M \nabla J_s(x) \le -\underline{\lambda}(M)\|\nabla J_s(x)\|^2 \tag{19}$$

Enforcing the time derivative of the cost to be negative, the updated law is designed as

$$\dot{\hat{w}}_{c,2} = -\eta_s \frac{\partial \nabla J_s^\top(x)[g(x)\hat{u}(x_k)+f(x)]}{\partial \hat{w}_c} = \frac{1}{2}\eta_s \nabla \sigma(x_k) g(x_k) R^{-1} g^\top(x) \nabla J_s(x) \tag{20}$$

where $\eta_s > 0$ is the designed learning rate.

Furthermore, to relax the dependency on the initial admissible control, an extra stabilizing term is added as

$$\dot{\hat{w}}_c = \dot{\hat{w}}_{c,1} + \Xi(x,\hat{u}(x_k))\dot{\hat{w}}_{c,2} \tag{21}$$

where $\Xi(x,\hat{u}(x_k))$ is a sign function defined as

$$\Xi(x,\hat{u}(x_k)) = \begin{cases} 0, & \text{when } \dot{J}_s(x) < 0, \\ 1, & \text{elsewhere} \end{cases} \tag{22}$$

This term switches off the reinforcement term $\dot{\hat{w}}_{c,2}$ when the system is already stable. It can be observed from the derivation that input constraints are not explicitly incorporated into the development of the control law and triggering conditions. This is because, although input constraints can theoretically be handled using a nonquadratic cost function and a hyperbolic barrier function [22,25,28], our experiments revealed that this approach reduces the robustness margin and increases sensitivity to parameter tuning. Therefore, in this work, input constraints are enforced separately from the online adaptation loop. This decoupled enforcement was found to enhance the practical applicability of event-triggered ADP. Furthermore, with appropriate parameter tuning of the algorithm, hard input saturation can be bypassed in practice.

The proposed algorithm is summarized in Fig. 2.

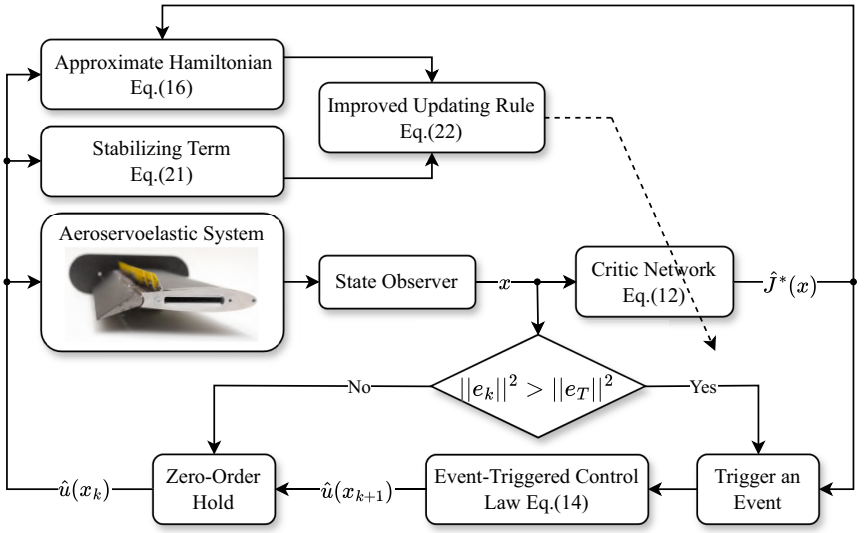

Fig. 2 Block diagram of the proposed event-triggered adaptive dynamic programming (ET-ADP) algorithm.

D. Closed-Loop Stability Analysis

Assumption 2: The closed-loop system is under persistent excitation (PE).

The persistent excitation (PE) condition is essential in adaptive control and reinforcement learning to guarantee the identifiability of system dynamics and the convergence of learning algorithms. It requires that the control input sufficiently excites all relevant modes of the system so that the collected data is informative for parameter estimation or value function approximation.

To satisfy the PE condition, a zero-mean Gaussian noise signal is added to the control input during the learning phase. This probing signal is energy bounded and spectrally rich, ensuring that the system is excited across a broad frequency range. Consequently, the regressor matrix remains full rank over time, enabling stable and convergent learning.

In addition to the designed probing noise, external disturbances and sensor noise present in the experimental environment provide further excitation. These naturally occurring variations enhance the richness of the input–output data and contribute to satisfying the PE condition in practice.

Assumption 3: The function $g(x)$ in Eq. (1) is Lipschitz continuous ($\|g(x) - g(x_k)\| \leq L_g\|e_k\|$). It also has an upper bound ($\|g(x)\| \leq b_g$).

Assumption 4: $\nabla\sigma_c(x)$ is Lipschitz continuous ($\|\nabla\sigma_c(x) - \nabla\sigma_c(x_k)\| \leq L_{\nabla\sigma_c}\|e_k\|$). Upper bounds exist such that $\|\nabla\sigma_c(x)\| \leq b_{\nabla\sigma_c}$, $\|\nabla\epsilon(x)\| \leq b_{\nabla\epsilon_c}$, $\|\varepsilon_{u_k^*}\| \leq b_{\varepsilon_{u^*}}$, and $|e_{cH}| \leq b_{e_{cH}}$.

Define the critic training error as $\tilde{w}_c = w_c - \hat{w}_c$.

Theorem 1: Under Assumptions 1–4, using Eq. (13) and the critic networking updating law (21), the closed-loop system is asymptotically stable with ultimately uniformly bounded weight error dynamics if

$$\|e_k\|^2 \leq \frac{(1-\eta)\underline{\lambda}(Q)\|x\|^2 + \underline{\lambda}(R)\|u^*(x_k)\|^2}{\mathcal{C}_1\|\hat{w}_c\|^2} \triangleq \|\hat{e}_T\|^2 \tag{23}$$

and $\|\tilde{w}_c\|^2 \geq \mathcal{W}$. $\mathcal{C}_1$ and $\mathcal{W}$ are positive definite constants; $\eta \in (0,1)$.

Proof: Using Eqs. (14), (17), and (21), the critic network error dynamics is

$$\dot{\tilde{w}}_c = -\frac{1}{2}\eta_s\Xi(x,\hat{u}(x_k))\nabla\sigma(x_k)g(x_k)R^{-1}g^\top(x)\nabla J_s(x) - \eta_c\phi(\phi^\top\tilde{w}_c - e_{cH}) \tag{24}$$

Under Assumption 2, it can be proved that $\underline{\lambda}(\phi\phi^\top) > 0$ [30].

The Lyapunov function candidate is constructed as $\mathcal{L} = L_x + L_{x_k} + L_{\tilde{w}_c} + L_{J_s}$, where $L_x = J^*(x)$, $L_{x_k} = J^*(x_k)$, $L_{\tilde{w}_c} = (1/2)\tilde{w}_c^\top\tilde{w}_c$, and $L_{J_s} = \eta_s J_s(x)$.

Situation 1: $\forall t \in [s_k, s_{k+1})$. Analogous to the derivations in our previous research in [22], it can be obtained that

If $\Xi(x,\hat{u}(x_k)) = 0$, then $\dot{\mathcal{L}} < 0$, $\forall x \neq 0$ when $\|e_k\|^2 < \|\hat{e}_T\|^2$ and $\|\tilde{w}_c\|^2 \geq \mathcal{W}_1$, where $\mathcal{W}_1$ is a positive constant.

Otherwise, if $\Xi(x,\hat{u}(x_k)) = 1$, then $\dot{\mathcal{L}} < 0$, $\forall x \neq 0$ when $\|e_k\|^2 < \|\hat{e}_T\|^2$ and $\|\tilde{w}_c\|^2 \geq \mathcal{W}_2$, where $\mathcal{W}_2$ is another positive constant. Denote $\mathcal{W} \triangleq \max\{\mathcal{W}_1, \mathcal{W}_2\}$.

Situation 2: $\forall t = s_{k+1}$. The candidate Lyapunov function is differed by $\Delta\mathcal{L} = \Delta L_x + \Delta L_{x_k} + \Delta L_{\tilde{w}_c} + \Delta L_{J_s}$. Because of the property of the limits, $\Delta L_x + \Delta L_{\tilde{w}_c} + \Delta L_{J_s} < 0$ [22]. Consequently, $\Delta\mathcal{L} < \Delta L_{x_k} \leq -\kappa(\|e_{k+1} - e_k\|)$ with a class-κ function $\kappa(\cdot)$. Therefore, the time derivative of $\mathcal{L}$ is still negative $\forall t = s_{k+1}$. This completes the proof. □

It can be seen from Eq. (23) that the event-triggered threshold $\|e_T\|^2$ is adaptive and updated online, rather than being predefined. The only tunable parameter in Eq. (23) is η, which lies in the range 0–1. This parameter introduces a tradeoff between performance and computational efficiency. When η is set to 1, the first term in the numerator becomes zero, resulting in a lower threshold. Consequently, triggering occurs more frequently, and the system's performance approaches that of the continuous counterpart of the proposed event-triggered algorithm.

E. Zeno Phenomenon Analysis

Zeno behavior refers to the occurrence of infinitely many triggering events within a finite time interval, leading to ill-posed system trajectories and rendering practical implementation infeasible. In practice, factors such as sensor and actuator delays, limited sampling resolution, and hardware constraints can alleviate the accumulation of infinite events, effectively mitigating the risk of Zeno behavior [31].

To ensure theoretical soundness and practical implementability, a positive minimum interevent time will be enforced in the triggering condition. This dwell-time constraint guarantees that successive events are separated by at least a fixed duration, thereby eliminating the possibility of Zeno behavior and ensuring the robustness of the event-triggered mechanism.

Denote the interexecution time as $\Delta s = s_{k+1} - s_k$, then $\Delta s_{\min} = \min_{k\in\mathbb{N}}\{s_{k+1} - s_k\}$ must be guaranteed to be larger than zero to avoid the Zeno phenomenon.

Theorem 2: Using control law in Eq. (13), the k-th interexecution time Δs of the closed-loop form of Eq. (1) determined by Eq. (23) is lower bounded as

$$\Delta s \geq \frac{1}{\kappa_1}\ell_n(1+\Gamma_{k,\min}) > 0 \tag{25}$$

where $\Gamma_{k,\min} = \min_{k\in\mathbb{N}}(\|e_T^{k+1}\|/(\|\hat{x}_k\|+\kappa_2)) > 0$, $\|e_T^{k+1}\| = e_k(s_{k+1}^-)$, $e_k(s_{k+1}^-) = \lim_{\vartheta\to 0} e_k(s_{k+1}-\vartheta)$, and κ_1 and κ_2 are positive constants.

Proof: Using Theorem 1, $\tilde{\omega}_c$ is upper bounded by a constant $b_{\tilde{\omega}_c}$. Using the fact that the optimal weight is bounded as $\|\omega_c\| \leq b_{\omega_c}$, then the weight estimate is bounded as $\|\hat{\omega}_c\| = \|\omega_c - \tilde{\omega}_c\| = b_{\omega_c} + b_{\tilde{\omega}_c}$.

Based on Eq. (13) and Assumptions 3 and 4, it can be derived

$$\|\dot{x}\| = \|g(x)\hat{u}(x_k) + f(x)\| \leq \kappa_1\|x\| + \kappa_2 \tag{26}$$

where κ_1 is a positive constant resulting from the Lipschitz continuity while $\kappa_2 = (1/2)\underline{\lambda}(R)b_g^2 b_{\nabla\varepsilon_c}(b_{\omega_c} + b_{\tilde{\omega}_c})/\kappa_1$. Therefore

$$\|\dot{e}_k\| \leq \kappa_1\|e_k(t)\| + \kappa_1(\|\hat{x}_k\| + \kappa_2), \quad \forall\, t \in [s_k, s_{k+1}) \tag{27}$$

Because $\hat{x}_k$ remains unchanged during $t \in [s_k, s_{k+1})$, according to [15], the following equation holds:

$$\begin{aligned}\|e_k\| &\leq \int_{s_k}^{t} \exp(\kappa_1(t-s))\kappa_1(\|\hat{x}_k\| + \kappa_2)\,\mathrm{d}s \\ &= (\|\hat{x}_k\| + \kappa_2)(\exp(\kappa_1(t-s_k)) - 1), \forall\, t \in [s_k, s_{k+1})\end{aligned} \tag{28}$$

Define $\|e_T(s_{k+1}^-)\| = e_T^{k+1}$, then from here, analogous to the proof in [32], it takes a minimal intersample of $\Delta s \geq (1/\kappa_1)\,\ell_n(1+\Gamma_{k,\min}) > 0$ for the term $\{\|e_T\|/(\|\hat{x}_k\|+\kappa_2)\}$ to grow from 0 to the minimum positive $\Gamma_{k,\min} = \min_{k\in\mathbb{N}}(\|e_T^{k+1}\|/(\|\hat{x}_k\|+\kappa_2))$. This holds for all $t \in [\lambda_k, \lambda_{k+1}), k \in \mathbb{N}$. The proof is then completed because $\Gamma_{k,\min} > 0$, then $\Delta s_{\min} > 0$ for any $x(t) \neq 0$. □

IV. Experiment Setup

A. The Aeroservoelastic Apparatus in a Wind Tunnel

Figure 3 shows an overview of the experimental setup. The aeroservoelastic apparatus is mounted in an open-circuit wind tunnel of Delft University of Technology [33]. A gust generator with two vanes is used to generate 1-cos gusts upstream, in the range of 0.5–12 Hz. The aeroservoelastic wing is designed with both pitch and heave degrees of freedom and incorporates a trailing-edge aileron and spoiler as active control devices. The actuator servos are identical for aileron and spoiler. The actuator dynamics are identified as a second-order low-pass filter with $\xi = 0.91$ and $\omega_n = 18.93$. Additionally, the position and rate are constrained within ±20 deg and ±750 deg/s, respectively. An MPU-9250 inertial measurement unit is embedded within the structure of the aeroservoelastic wing. Moreover, the linear and rotational motions of the aeroservoelastic system are captured by noninvasive linear and rotary variable differential transformers. These devices are read by a 12-bit analog-to-digital converter, sampling at 200 Hz. Overall, the measurement signals used for control are h, $\ddot{h}$, θ, $\dot{\theta}$, and β. The remaining states are observed from a Kalman filter. More details on the aeroservoelastic apparatus can be found in [34].

To design a critic network for estimating the cost function, the approximating accuracy improves as the activation function's nonlinearity and the number of neurons increase. However, using more highly nonlinear neurons can also raise computational demands and risk overfitting, which undermines control robustness. In this setup, the neural network employs basis functions composed primarily of low-order polynomial terms (up to second order), optionally combined with tanh activations. This structure balances nonlinearity and computational efficiency, especially for systems with moderate state dimensions (12 in this case), where a second-order polynomial expansion leads to one single hidden layer with in total 132 neurons. The number of neurons in the hidden layer is selected based on empirical performance and model complexity tradeoffs. Starting from a minimal configuration, the architecture is incrementally expanded until no significant gain is observed on a validation set.

To enhance generalization and avoid overfitting, regularization techniques including L_2 weight penalties, dropout, and early stopping are applied during training. The network is trained using standard backpropagation and stochastic gradient descent. During online learning, robustness and generalization are enhanced by constraining value function updates within predefined bounds and enforcing input constraints to prevent drift into poorly trained regions of the state space. In the cost function, Q is chosen as an identity matrix and R is tuned to be 0.1 as a tradeoff between performance and control efforts.

B. Offline Pretraining

Safety is of primary importance in aerospace. Directly applying a learning algorithm online has a lower probability of being certified by authorities such as the European Union Aviation Safety Agency. Therefore, this experiment adopts a combined offline and online learning approach. This approach aligns with the philosophy of initially training a human pilot on ground flight simulators before advancing to real flight training. For the offline training process, a linear quadratic regulator controller is selected as a baseline. The proposed ADP algorithm imitates the behavior of the linear quadratic regulator controller in an offline simulation environment for 10,000 training steps.

The choice of 10,000 training steps was made based on the observed convergence behavior, as illustrated in Fig. 4. A balance was sought between achieving sufficient training for policy convergence and preventing overfitting, which may result from

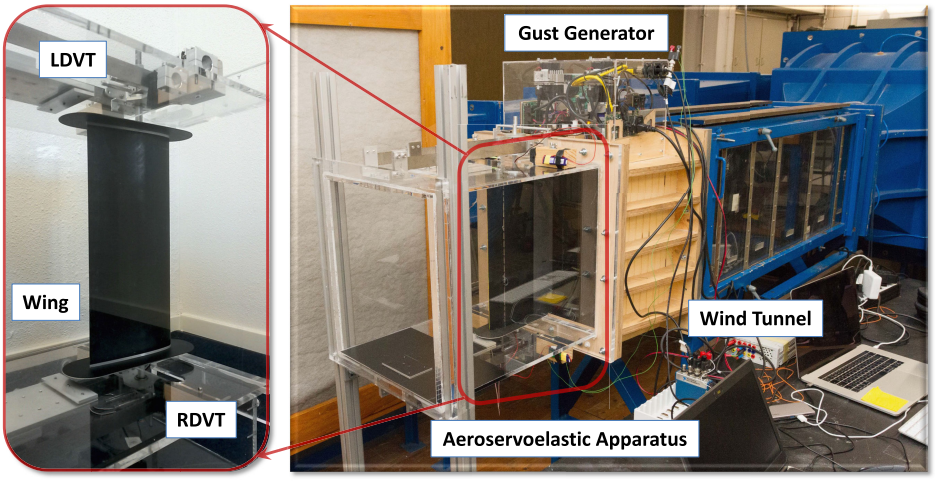

Fig. 3 An overview of the experiment setup.

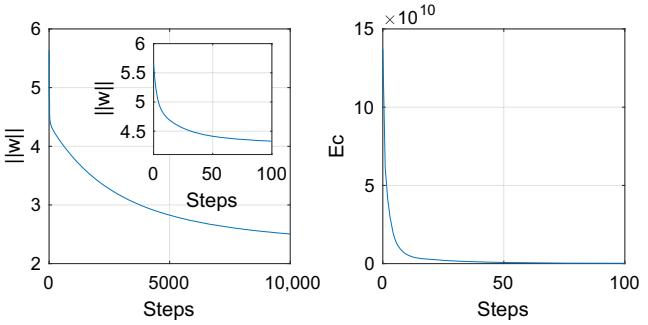

Fig. 4 Evolution of the critic network weights norm $\|w\|$ and the training cost E_c.

excessive training. Furthermore, this number was selected to provide a reasonable tradeoff between performance and computational cost. To mitigate overfitting and improve generalization, probing noise was introduced during the offline training process.

It is worth noting that the baseline controller is used solely to train the initial weights of the ADP to achieve an admissible initial design. The baseline controller becomes suboptimal in the presence of real-world nonlinearities, time delays, and non-Gaussian external disturbances (including the 1-cos gust required by certification regulations). The gap between simulation and reality will be bridged by the continuous online learning capability of the proposed ADP algorithm. Additionally, the onboard computational load will be reduced by the proposed event-triggering algorithm.

Figure 4 illustrates the evolution of norm value of the critic weights and the training cost. As observed, both values exhibit convergence. Moreover, the norm value of the critic weights experience a significant drop during the initial 100 steps, followed by a more gradual decline. In contrast, the cost demonstrates a much higher rate of convergence. The weight values at the final offline training step are used as the initial weights for the real-world experiment.

V. Results and Discussion

A. Real-World Performance of the Event-Triggered (ET)-ADP

The real-world effectiveness of the designed ET-ADP algorithm under the excitation of a 7-Hz (close to its first resonance) gust when the inflow velocity is 12.5 m/s is presented in this subsection. Robustness to a broader range of gust frequency and inflow velocity will be presented in Sec. V.C. It is noteworthy that the external atmospheric disturbances (including 1-cos gusts required by certification and continuous turbulence) are unforeseen by the controller both during the offline training process and the real-world online experiment.

Figure 5 illustrates that the aeroservoelastic modes have been excited by the gust, leading to an oscillatory behavior that persists

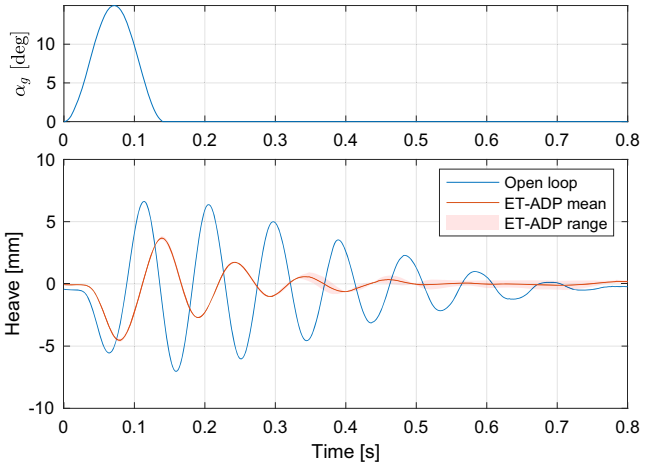

Fig. 5 Gust angle of attack α_g and the corresponding responses (shade indicates the data range).

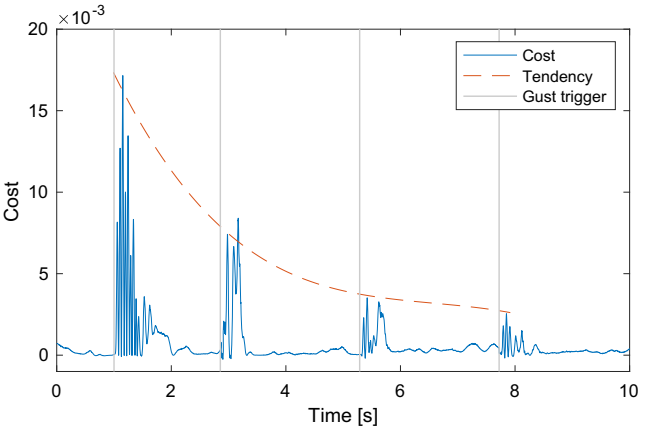

Fig. 6 Evolution of the critic cost function under four repetitive gusts.

even after the gust has passed the system. Without control, it takes more than 0.8 s for the system to dampen out. On the contrary, the ET-ADP controller increases the damping in the system and attenuates the motion within 0.5 s. With the ET-ADP controller, the peak is reduced by 35.41%, whereas the root mean square (rms) value of the motion is reduced by 54.45%. Moreover, the shading in the figure indicates that the results of 10 different tests are predominantly consistent.

The critic cost function evolution is illustrated in Fig. 6. Four 1-cos wind gusts in the same pattern strike the system one after another. It can be observed that when the system is hit by one single gust, the critic cost first increases, then quickly descends and converges. Furthermore, as the number of gusts experienced by the system increases, the overall cost of the algorithm decreases, indicating the online learning capability of the ET-ADP.

B. Effectiveness of Event Triggering

The trajectory of the triggering threshold $\|e_T\|^2$ is displayed in Fig. 7. When a single gust occurs, $\|e_T\|^2$ experiences an initial sudden increase and then shows a trend of converging to zero, along with the event error $\|e_k\|^2$. Overall, as the algorithm becomes increasingly experienced, both $\|e_T\|^2$ and $\|e_k\|^2$ decline.

Figure 8 presents interexecution time evolution. It can be observed that the ET-ADP algorithm can extend the interexecution time from 0.002 s to up to 1.755 s (877.5 times). This enhancement is also reflected in Fig. 9, which compares the control command of ET-ADP with its time-based ADP counterpart. The latter communicates with the actuator at a fixed time interval of 0.002 s. Figure 9 demonstrates that the time-based ADP not only responds to wind gusts but also continues to react to measurement noise and low-magnitude turbulence in the flow after the gust has passed and the majority of energy in the system has been dissipated. This continuous reaction after the main dissipation actually contributes

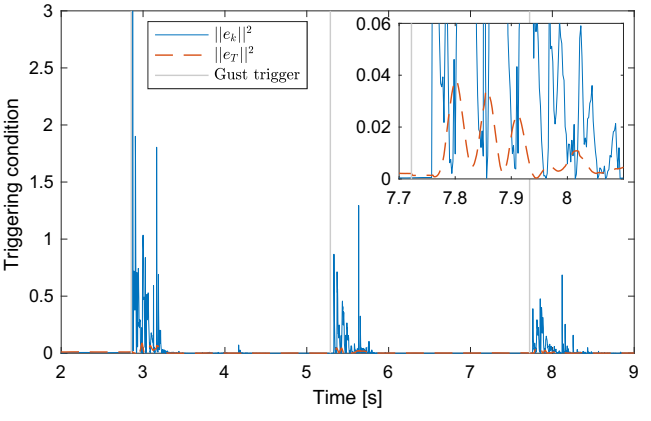

Fig. 7 The event-triggering condition evolution.

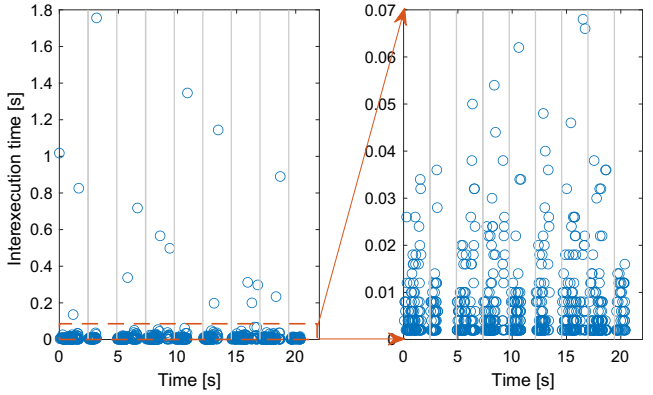

Fig. 8 Evolution of the interexecution time (with a zoomed-in subplot).

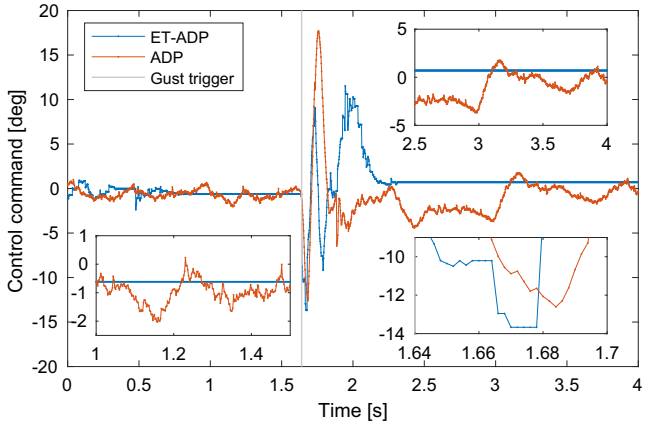

Fig. 9 Comparison between the control commands of ET-ADP and ADP.

marginally to gust load alleviation performance but increases communication costs. In contrast, ET-ADP primarily issues control commands during the transient phase and ceases updating when not necessary. This significantly improves communication and resource utilization efficiency.

As proved in Theorem 2, the Zeno phenomenon is avoided in the proposed algorithm. This is further verified in Fig. 8, in which the interexecution time not only encompasses values of 0.002, but it also includes a broad spectrum of values greater than 0.002.

The number of control command updates is presented in Fig. 10. Over a duration of 25 s, the aeroservoelastic system experienced nine consecutive gusts. The time-triggered ADP updates a total of 12,500 times, whereas the ET-ADP updates only 3462 times, resulting in a 72.30% improvement in efficiency. Furthermore, the zoomed-in plot of Fig. 10 reveals that the number of control command updates for ET-ADP displays a stair-step appearance. This indicates that the number of control command updates rapidly increases following the impact of a gust and then stabilizes after the gust has passed. Figure 8 also corroborates this phenomenon, showing that samples of shorter interexecution times cluster around the periods shortly after a gust impact. Once the gust has passed, the interexecution times generally lengthen.

In conventional time-triggered implementations, the neural network is updated at every time step, requiring continuous backpropagation, large-scale matrix computations, and frequent communication with actuators. This results in high computational demand and increased actuator activity, which can lead to excessive energy consumption and accelerated hardware degradation. By contrast, the event-triggered strategy reduces the frequency of neural network updates and actuator communications by allowing updates only when certain conditions are met. This approach lowers the computational load and decreases the energy usage associated with actuator movement.

The additional implementation cost of even-triggering is small. Compared to the standard approach, the only extra operations are the computation of a time-varying threshold and the evaluation of a triggering condition. These operations are relatively lightweight, whereas the reduction in network updates and actuator usage leads to more significant overall savings.

C. Robustness, Adaptation, and Efficiency

The robustness of the controller to external disturbances and the adaptation of the controller to various gust frequencies (ranging from 3 to 8 Hz) and inflow velocities (ranging from 11.5 to 12.5 m/s) are presented in this subsection.

Figure 11 presents a comparison between the ET-ADP and its time-based ADP counterpart across a total of 15 test conditions (as illustrated on the horizontal axes). Under each test condition, the algorithm repeatability encounters one specific type of external perturbation (containing both in-flow and cross-flow components). In the context of wind-tunnel experiments, in-flow disturbances refer to perturbations that occur along the direction of the mean flow (streamwise). In our case, such disturbances are naturally present due to inherent fluctuations in the wind-tunnel speed. In contrast, cross-flow disturbances refer to perturbations that occur perpendicular to the main flow direction, such as vertical or lateral gusts. In this experiment, a set of gust generators was installed upstream of the aeroelastic apparatus to introduce prescribed cross-flow vertical gusts.

During the intervals between gust encounters, two performance metrics are utilized: 1) the reduction percentage in the rms of the heave motion compared to the open-loop response, and 2) the reduction percentage in the maximum amplitude of the heave motion compared to the open-loop response.

It can be observed from Fig. 11 that both algorithms can effectively attenuate the aeroservoelastic oscillations. On average, the rms of the heave motion is reduced by 40.51%, whereas the peak is reduced by 24.68%. The results of different tests are also predominantly consistent. Holding the control commands after the

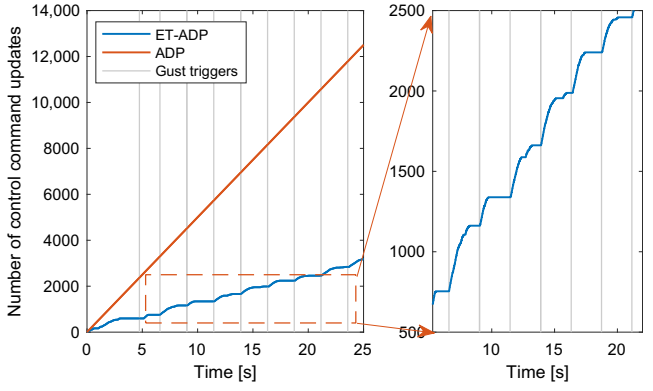

Fig. 10 Comparison between the number of control command updates (with a zoomed-in subplot).

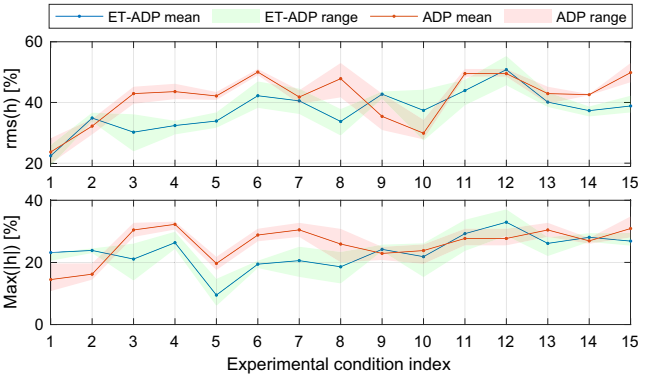

Fig. 11 The motion attenuation ratio of ET-ADP and ADP in various experimental conditions.

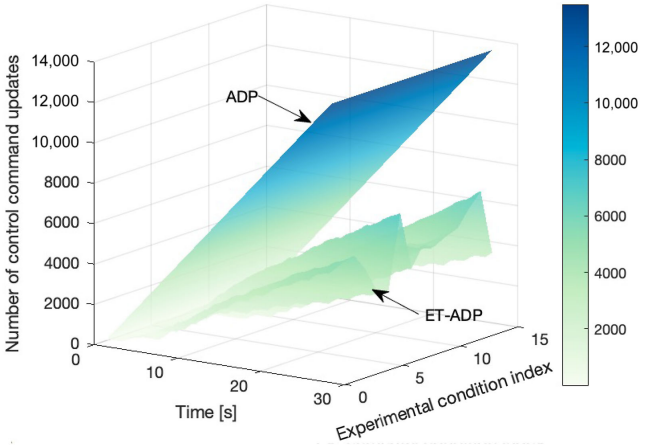

Fig. 12 Evolution of the number of command updates in various experimental conditions.

majority of the gust effects have dissipated does not significantly enhance the performance, as Fig. 11 demonstrates that the performances of ET-ADP and time-triggered ADP are comparable. Nonetheless, event-triggering significantly reduces control command update numbers. This is further verified in Fig. 12, which shows that the update times for ET-ADP are lower throughout the entire time history and across all experimental conditions. On average, ET-ADP reduces the number of updates by 62.98%, contributing to improving both computational and communication resource utilization.

VI. Conclusions

In this Technical Note, we focus on the real-world experimental investigation of gust disturbance rejection in a nonlinear, underactuated, and uncertain aeroservoelastic system using the event-triggered adaptive dynamic programming (ET-ADP) approach. Real-world experiments reveal that the nonlinear hyperbolic tangent function used in existing ET-ADP algorithms can reduce robustness to input delays and increase sensitivity to parameter tuning. To address these issues, an alternative algorithm following the ET-ADP framework is presented and theoretically proved in this Note. In real-world experiments, this algorithm suppresses the oscillatory motion of the system in the presence of unknown external gusts and turbulence, without exhibiting the Zeno phenomenon. When a single gust repeatedly impacts the system, the algorithm also demonstrates the ability of learning from past experience online. Moreover, it automatically adapts to various inflow and crossflow conditions, achieving an average reduction of 40.51% in the root mean square of system motion. Furthermore, the experimental results show that the algorithm dynamically adjusts the control command update frequency: it increases the update rate during sudden gusts for timely suppression, and reduces it after the gusts subside, effectively balancing rapid responsiveness and resource efficiency. As a result, compared to its time-triggered counterpart, the ET-ADP algorithm reduces the number of control updates by 62.98% and extends the interexecution interval by up to 877.5 times. These findings demonstrate the effectiveness of the proposed method in reducing computational and communication loads without compromising performance. Validation of the effectiveness under irregular or multimodal gust conditions is recommended as future work.

References

[1] Iannelli, A., Marcos, A., and Lowenberg, M., "Nonlinear Robust Approaches to Study Stability and Postcritical Behavior of an Aeroelastic Plant," *IEEE Transactions on Control Systems Technology*, Vol. 27, No. 2, 2018, pp. 703–716. https://doi.org/10.1109/TCST.2017.2779110

[2] Baldelli, D. H., Lind, R., and Brenner, M., "Nonlinear Aeroelastic/Aeroservoelastic Modeling by Block-Oriented Identification," *Journal of Guidance, Control, and Dynamics*, Vol. 28, No. 5, 2005, pp. 1056–1064. https://doi.org/10.2514/1.11792

[3] Tewari, A., *Aeroservoelasticity*, Springer-Verlag, Berlin, 2015, pp. 1–11. https://doi.org/10.1007/978-1-4939-2368-7

[4] Iannelli, A., Marcos, A., and Lowenberg, M., "Study of Flexible Aircraft Body Freedom Flutter with Robustness Tools," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 5, 2018, pp. 1083–1094. https://doi.org/10.2514/1.G003165

[5] Theis, J., Pfifer, H., and Seiler, P., "Robust Modal Damping Control for Active Flutter Suppression," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 6, 2020, pp. 1056–1068. https://doi.org/10.2514/1.G004846

[6] Qi, P., Zhao, X., Wang, Y., Palacios, R., and Wynn, A., "Aeroelastic and Trajectory Control of High Altitude Long Endurance Aircraft," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 54, No. 6, 2018, pp. 2992–3003. https://doi.org/10.1109/TAES.2018.2836598

[7] Poussot-Vassal, C., Demourant, F., Lepage, A., and Le Bihan, D., "Gust Load Alleviation: Identification, Control, and Wind Tunnel Testing of a 2-D Aeroelastic Airfoil," *IEEE Transactions on Control Systems Technology*, Vol. 25, No. 5, 2016, pp. 1736–1749. https://doi.org/10.1109/TCST.2016.2630505

[8] Livne, E., "Aircraft Active Flutter Suppression: State of the Art and Technology Maturation Needs," *Journal of Aircraft*, Vol. 55, No. 1, 2018, pp. 410–452. https://doi.org/10.2514/1.C034442

[9] Cook, R. G., Palacios, R., and Goulart, P., "Robust Gust Alleviation and Stabilization of Very Flexible Aircraft," *AIAA Journal*, Vol. 51, No. 2, 2013, pp. 330–340. https://doi.org/10.2514/1.J051697

[10] Van Wingerden, J., Houtzager, I., Felici, F., and Verhaegen, M., "Closed-Loop LPV Identification of the Time-Varying Dynamics of a Variable-Speed Wind Turbine," *International Journal of Robust and Nonlinear Control: IFAC-Affiliated Journal*, Vol. 19, No. 1, 2009, pp. 4–21. https://doi.org/10.1002/rnc.1320

[11] Hodel, A., Whorton, M., and Zhu, J., "Stability Metrics for Simulation and Flight-Software Assessment and Monitoring of Adaptive Control Assist Compensators," *AIAA Guidance, Navigation and Control Conference and Exhibit*, AIAA Paper 2008-7005, 2008. https://doi.org/10.2514/6.2008-7005

[12] Duessler, S., Mylvaganam, T., and Palacios, R., "LQG-Based Gust Load Alleviation Systems for Very Flexible Aircraft," *AIAA Scitech 2023 Forum*, AIAA Paper 2023-2571, 2023. https://doi.org/10.2514/6.2023-2571

[13] Wang, X., Mkhoyan, T., and De Breuker, R., "Nonlinear Incremental Control for Flexible Aircraft Trajectory Tracking and Load Alleviation," *Journal of Guidance, Control, and Dynamics*, Vol. 45, No. 1, 2022, pp. 39–57. https://doi.org/10.2514/1.G005921

[14] Wang, X., Van Kampen, E., Chu, Q., and De Breuker, R., "Flexible Aircraft Gust Load Alleviation with Incremental Nonlinear Dynamic Inversion," *Journal of Guidance, Control, and Dynamics*, Vol. 42, No. 7, 2019, pp. 1519–1536. https://doi.org/10.2514/1.G003980

[15] Khalil, H. K., *Control of Nonlinear Systems*, Prentice Hall, Upper Saddle River, NJ, 2002, pp. 505–551.

[16] Meng, Q., Yang, H., and Jiang, B., "Fault-Tolerant Optimal Spacecraft Attitude Maneuver: An Incremental Model Approach," *Journal of Guidance, Control, and Dynamics*, Vol. 45, No. 9, 2022, pp. 1676–1691. https://doi.org/10.2514/1.G006417

[17] Wang, D., Gao, N., Liu, D., Li, J., and Lewis, F. L., "Recent Progress in Reinforcement Learning and Adaptive Dynamic Programming for Advanced Control Applications," *IEEE/CAA Journal of Automatica Sinica*, Vol. 11, No. 1, 2024, pp. 18–36. https://doi.org/10.1109/JAS.2023.123843

[18] Liu, D., Shan, X., Zhao, B., Luo, B., and Wei, Q., "Adaptive Dynamic Programming for Control: A Survey and Recent Advances," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 51, No. 1, 2021, pp. 142–160. https://doi.org/10.1109/TSMC.2020.3042876

[19] Heydari, A., and Balakrishnan, S. N., "Finite-Horizon Control-Constrained Nonlinear Optimal Control Using Single Network Adaptive Critics," *IEEE Transactions on Neural Networks and Learning Systems*, Vol. 24, No. 1, 2013, pp. 145–157. https://doi.org/10.1109/TNNLS.2012.2227339

[20] Dong, H., Zhao, X., and Luo, B., "Optimal Tracking Control for Uncertain Nonlinear Systems with Prescribed Performance via Critic-Only ADP," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 52, No. 1, 2020, pp. 561–573. https://doi.org/10.1109/TSMC.2020.3003797

[21] Dong, H., Zhao, X., and Luo, B., "Optimal Tracking Control for Uncertain Nonlinear Systems with Prescribed Performance via Critic-Only ADP," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 52, No. 1, 2022, pp. 561–573. https://doi.org/10.1109/TSMC.2020.3003797

[22] Sun, B., Wang, X., and Van Kampen, E.-J., "Event-Triggered Intelligent Critic Control with Input Constraints Applied to a Nonlinear Aeroelastic System," *Aerospace Science and Technology*, Vol. 120, Jan. 2022, Paper 107279. https://doi.org/10.1016/j.ast.2021.107279

[23] Ge, X., Han, Q.-L., Zhang, X.-M., and Ding, D., "Dynamic Event-Triggered Control and Estimation: A Survey," *Machine Intelligence Research*, Vol. 18, No. 6, 2021, pp. 857–886. https://doi.org/10.1007/s11633-021-1306-z

[24] Wu, Q., Zhao, B., Liu, D., and Polycarpou, M. M., "Event-Triggered Adaptive Dynamic Programming for Decentralized Tracking Control of Input-Constrained Unknown Nonlinear Interconnected Systems," *Neural Networks*, Vol. 157, Jan. 2023, pp. 336–349. https://doi.org/10.1016/j.neunet.2022.10.025

[25] Xue, S., Luo, B., Liu, D., and Li, Y., "Adaptive Dynamic Programming Based Event-Triggered Control for Unknown Continuous-Time Nonlinear Systems with Input Constraints," *Neurocomputing*, Vol. 396, July 2020, pp. 191–200. https://doi.org/10.1016/j.neucom.2018.09.097

[26] Chaudhary, S., Dubey, R., Tripathy, N. S., and Shah, S. V., "Data-Driven Event-Triggered Predictive Post-Impact Control of Space Robot with Uncertainties," *Journal of Guidance, Control, and Dynamics*, Vol. 48, No. 2, 2025, pp. 1–16. https://doi.org/10.2514/1.G008115

[27] Geller, D. K., Rose, M. B., and Woffinden, D. C., "Event Triggers in Linear Covariance Analysis with Applications to Orbital Rendezvous," *Journal of Guidance, Control, and Dynamics*, Vol. 32, No. 1, 2009, pp. 102–111. https://doi.org/10.2514/1.36834

[28] Wang, D., Mu, C., Yang, X., and Liu, D., "Event-Based Constrained Robust Control of Affine Systems Incorporating an Adaptive Critic Mechanism," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 47, No. 7, 2017, pp. 1602–1612. https://doi.org/10.1109/TSMC.2016.2642118

[29] Wang, D., He, H., and Liu, D., "Intelligent Optimal Control with Critic Learning for a Nonlinear Overhead Crane System," *IEEE Transactions on Industrial Informatics*, Vol. 14, No. 7, 2018, pp. 2932–2940. https://doi.org/10.1109/TII.2017.2771256

[30] Vamvoudakis, K. G., "Event-Triggered Optimal Adaptive Control Algorithm for Continuous-Time Nonlinear Systems," *IEEE/CAA Journal of Automatica Sinica*, Vol. 1, No. 3, 2014, pp. 282–293. https://doi.org/10.1109/JAS.2014.7004686

[31] Wang, X., and Lemmon, M. D., "Event-Triggering in Distributed Networked Control Systems," *IEEE Transactions on Automatic Control*, Vol. 56, No. 3, 2011, pp. 586–601. https://doi.org/10.1109/TAC.2010.2057951

[32] Zhang, Q., Zhao, D., and Zhu, Y., "Event-Triggered H Infinity Control for Continuous-Time Nonlinear System via Concurrent Learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, Vol. 47, No. 7, 2016, pp. 1071–1081. https://doi.org/10.1109/TSMC.2016.2531680

[33] Mkhoyan, T., Ruland, O., De Breuker, R., and Wang, X., "On-Line Black-Box Aerodynamic Performance Optimization for a Morphing Wing with Distributed Sensing and Control," *IEEE Transactions on Control Systems Technology*, Vol. 31, No. 3, 2022, pp. 1063–1077. https://doi.org/10.1109/TCST.2022.3210164

[34] Schildkamp, R., Wang, X., Chang, J., De Breuker, R., and Sodja, J., "Development of an Active Aeroelastic Parametric Wing Apparatus," *International Forum on Aeroelasticity and Structural Dynamics 2022*, Organization Committee of IFASD 2021 POD, Curran Assoc., 2022, https://www.proceedings.com/68435.html.

V. Mukhopadhyay
Associate Editor