Nonlinear Optimal Control for Active Suppression of Airfoil Flutter via a Novel Neural-Network-Based Controller*

Difan Tang, Lei Chen, Zhao F. Tian and Eric Hu

Abstract— **This paper proposes a novel nonlinear controller based on neural networks (NNs) for active suppression of airfoil flutter (ASAF). Aeroelastic flutter can damage airfoils if not properly controlled. Existing optimal controllers for ASAF are sensitive to modeling errors while other controllers less prone to uncertainties do not provide optimal control. This study, thus, focuses on solving these problems by deriving a new intelligent model-based control scheme capable of synthesizing nonlinear near-optimal control laws in real time according to a known model and online updated system dynamics. A four-degrees-of-freedom aeroelastic model that has nonlinear translational and torsional stiffness and employs leading/trailing-edge control surfaces as control inputs is considered. Optimal control laws for the nonlinear aeroelastic system is synthesized by solving the Hamiltonian-Jacobi-Bellman equation through NN-based value function approximation (VFA) and synchronous policy iteration in a Critic-Actor configuration. A systematic approach based on linear matrix inequalities (LMIs) is proposed for the design of a scheduled parameter matrix for the VFA. An NN-based identifier is also derived to capture un-modeled dynamics online. Extended Kalman filters are employed to tune NN parameters. The proposed controller was tested in wind-tunnel experiments. Comparisons drawn with a linear-parameter-varying optimal controller confirms the effectiveness and validity of the proposed control scheme.**

I. INTRODUCTION

Airfoil flutter is a type of self-feeding oscillation due to the interaction between aerodynamic loads and non-rigid airfoil structures, occurring at and above a certain airflow velocity (i.e. flutter boundary) depending on the structural characteristics of the airfoil [1]. Aeroelastic systems in reality are subjected to various nonlinearities and are generally prone to the instability known as limit-cycle oscillation (LCO), which can cause serious damage to the airfoil. To actively suppress airfoil flutter, existing airfoil control surfaces can be utilized, as proven effective by extensive studies.

In terms of control methods for suppressing LCOs, various algorithms apply [2], [3]. Due to the time-varying nature and nonlinear characteristics of an aeroelastic system and the increasing demand on wider operation range beyond the flutter boundary, adaptive and robust control has become a major focus in recent studies for active suppression of airfoil flutter (ASAF). These advanced methods include but are not limited to: online updated linear quadratic regulator [4], [5], linear-parameter-varying techniques [6], [7], feedback linearization [8], [9], model reference adaptive control [10], back-stepping-based adaptive output feedback control [11], [12], robust output feedback control [13], modular adaptive control [14], [15], modified filtered-X least-mean-square control [16], $\mathcal{L}_1$ adaptive control [17], sliding-mode control [18], finite-time H_∞ adaptive fault-tolerant control [19], [20] and neural network based adaptive control [21]–[23], etc.

However, optimal controllers among the mentioned methods are sensitive to modeling errors, which means suboptimal or unsatisfactory performance may result, in the presence of uncertainties or faults. Though some other controllers are designed to be more adaptive and robust to the changing environments and tolerant to un-modeled dynamics, these methods do not provide nonlinear optimal control. These two problems, although of significance to improving ASAF performance, have nevertheless not been addressed.

Therefore, the study in this paper proposes a new method of nonlinear control for ASAF from the optimal control perspective, aiming to reduce the impact of the aforementioned two problems. Contributions include:

(i) A neural network (NN) based algorithm capable of synthesizing optimal control laws online for nonlinear systems is derived, which has a compact configuration suitable for real-time implementation without jeopardizing closed-loop stability.

(ii) The compact NN controller relies on a modified form of value function approximation (VFA) [24], which however, cannot be directly implemented in its original form for ASAF applications. This is because the closed-loop stability is not guaranteed for the aeroelastic dynamics that change with different airspeeds. Therefore, a scheduled parameter matrix synthesized via linear matrix inequalities (LMIs) is proposed for the VFA so that the controller remains effective across a wide airspeed range of interest beyond the flutter boundary.

(iii) Wind-tunnel experiments were conducted to validate the proposed method. To the best of our knowledge, it is the first experimentally validated approach in this regard.

II. AEROELASTIC SYSTEM

The study in this paper is based on a typical rigid airfoil section featuring two-dimensional vibration modes (i.e. the first plunge and first pitch mode oscillations) under linear unsteady aerodynamic loads in subsonic flow [9], [10], [25]. Leading-edge (LE) and trailing-edge (TE) control surfaces are used to suppress flutter. Nonlinear translational and torsional stiffness is introduced in a polynomial form up to the second order. To better capture the control delay due to servo dynamics, a four-degrees-of-freedom (4-DOF)

*This work was supported by the Sir Ross and Sir Keith Smith Fund.

The authors are with the School of Mechanical Engineering, The University of Adelaide, Adelaide, SA 5005, Australia lei.chen@adelaide.edu.au

aeroelastic system as in [26] is considered (details of the model not repeated herein due to space limit).

The aeroelastic system can be expressed in a state-space control-affine form as:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t), U_\infty) + \boldsymbol{g}(\boldsymbol{x}(t))\boldsymbol{u}(t); \quad \boldsymbol{x}(0) = \boldsymbol{x}_0, \quad (1)$$

where $\boldsymbol{x}(t) \in \mathbb{R}^{n_x}$ denotes n_x system states; $\boldsymbol{u}(t) \in \mathbb{R}^{n_u}$ refers to n_u control inputs; $\boldsymbol{f}(\boldsymbol{x}(t), U_\infty) \in \mathbb{R}^{n_x}$ describes system internal dynamics that are dependent on the airflow velocity U_∞; $\boldsymbol{g}(\boldsymbol{x}(t)) \in \mathbb{R}^{n_x \times n_u}$ are control input dynamics.

III. Proposed Controller

A. Continuous-Time HJB Equation and Policy Iteration

For a fixed velocity U_∞, equation (1) can be reduced to:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{f}(\boldsymbol{x}(t))|_{U_\infty = U} + \boldsymbol{g}(\boldsymbol{x}(t))\boldsymbol{u}(t); \quad \boldsymbol{x}(0) = \boldsymbol{x}_0, \quad (2)$$

which can be written in a compact form:

$$\dot{\boldsymbol{x}}(t) = \mathcal{F}(\boldsymbol{x}(t), \boldsymbol{u}(t))|_{U_\infty = U}; \quad \boldsymbol{x}(0) = \boldsymbol{x}_0, \quad (3)$$

where $U \in \mathbb{R}^+$ is any valid value of U_∞.

For convenience in discussion, the dynamics associated with a constant velocity U is hereafter written in a simpler form by omitting the notation of $U_\infty = U$.

The control problem is to find a control law/policy $\boldsymbol{u}(t)$ to minimize the following performance index (cost function):

$$V(\boldsymbol{x}_0) = \int_0^\infty [\bar{Q}(\boldsymbol{x}(\tau)) + \bar{R}(\boldsymbol{u}(\tau))] d\tau, \quad (4)$$

with $\bar{Q}(\boldsymbol{x}(t))$ and $\bar{R}(\boldsymbol{u}(t)) = \boldsymbol{u}^{\mathrm{T}}(t)\boldsymbol{R}\boldsymbol{u}(t)$ being positive-definite functions, in which $\boldsymbol{R} \in \mathbb{R}^{n_u \times n_u}$ is a positive-definite weighting matrix.

Differentiating (4) yields its infinitesimal version that is a nonlinear Lyapunov equation [27], written as:

$$\boldsymbol{V}_x^{\mathrm{T}}(\boldsymbol{x})\mathcal{F}(\boldsymbol{x}, \boldsymbol{u}) + \bar{Q}(\boldsymbol{x}) + \bar{R}(\boldsymbol{u}) = 0; \quad V(0) = 0. \quad (5)$$

Let $V^*(\boldsymbol{x})$ denote the optimal (minimal) cost function, named as the 'value function', and let $\boldsymbol{V}_x^*(\boldsymbol{x}) \triangleq \partial V^*(\boldsymbol{x})/\partial \boldsymbol{x}$ denote its derivative with respect to $\boldsymbol{x}$, The corresponding optimal control policy is then given by:

$$\boldsymbol{u}^*(\boldsymbol{x}) = -\frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{V}_x^*(\boldsymbol{x}), \quad (6)$$

which satisfies the Hamilton-Jacobi-Bellman (HJB) equation based on (5):

$$-\frac{1}{4}[\boldsymbol{V}_x^*(\boldsymbol{x})]^{\mathrm{T}}\boldsymbol{g}(\boldsymbol{x})\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{V}_x^*(\boldsymbol{x}) + \bar{Q}(\boldsymbol{x}) + [\boldsymbol{V}_x^*(\boldsymbol{x})]^{\mathrm{T}}\boldsymbol{f}(\boldsymbol{x}) = 0; \quad V^*(0) = 0. \quad (7)$$

That is, by solving (7) for $V^*(\boldsymbol{x})$, the optimal control can then be obtained as in (6), given that the system internal dynamics $\boldsymbol{f}(\boldsymbol{x})$ and control input dynamics $\boldsymbol{g}(\boldsymbol{x})$ are known.

The HJB equation is nonlinear and difficult to solve directly. Instead, it can be solved recursively through the successive approximation method introduced in [28], which is generally recognized as a policy-iteration approach [29]. Despite different forms of realization, the policy-iteration algorithm basically involves two steps – policy evaluation (using (5) as a 'critic') and policy improvement (using (6) as an 'actor', with $\boldsymbol{V}_x^*(\boldsymbol{x})$ substituted by $\boldsymbol{V}_x^{\boldsymbol{u}^{(i)}}(\boldsymbol{x})$ associated with the control policy $\boldsymbol{u}^{(i)}(\boldsymbol{x})$ at an iteration step i). Starting with an initial admissible control policy $\boldsymbol{u}^{(0)}(\boldsymbol{x})$, the algorithm proceeds until convergence is reached at $V^*(\boldsymbol{x})$ and $\boldsymbol{u}^*(\boldsymbol{x})$. The iterative use of (5) and (6) thus forms the 'critic-actor' algorithm structure. In the case of synchronous policy iteration, the iterative use of (5) and (6) is performed simultaneously and continuously, with the superscript 'i' in $\boldsymbol{V}_x^{\boldsymbol{u}^{(i)}}(\boldsymbol{x})$ and $\boldsymbol{u}^{(i)}(\boldsymbol{x})$ representing an infinitesimal time step.

B. NN-based Value Function Approximation (VFA)

Note that solving for $V^{\boldsymbol{u}^{(i)}}(\boldsymbol{x})$ in a direct way is difficult. To implement the policy iteration, an appropriately structured representation of $V^*(\boldsymbol{x})$ is necessary, which can be a neural-network (NN) approximation. For this purpose, a modified form of value function approximation (VFA) [24] is used:

$$V^*(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x} + \boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{x}) + \varepsilon(\boldsymbol{x}), \quad (8)$$

where $\boldsymbol{\Phi}(\cdot) = [\phi_1(\boldsymbol{x}), \cdots, \phi_N(\boldsymbol{x})]^{\mathrm{T}} : \mathbb{R}^n \to \mathbb{R}^N$ contains N hidden-layer neurons, each of which is a nonlinear activation function; $\boldsymbol{W} \in \mathbb{R}^N$ is a vector of ideal NN weights; $\boldsymbol{P} \in \mathbb{R}^{n_x \times n_x}$ is a diagonal positive-definite matrix; $\varepsilon(\boldsymbol{x}) \in \mathbb{R}$ is the approximation error.

Remark 1: This VFA allows a compact critic-actor configuration more suitable for online implementation by eliminating the actor NN tuning loop and critic NN logic switch mechanism [24]. The activation functions $\boldsymbol{\Phi}(\boldsymbol{x})$ can be selected according to the high-order Weierstrass approximation theorem [30] so that $\varepsilon(\boldsymbol{x})$ is sufficiently small [27].

The derivative of $V^*(\boldsymbol{x})$ with respect to $\boldsymbol{x}$ is given by:

$$\boldsymbol{V}_x^*(\boldsymbol{x}) \triangleq \frac{\partial V^*(\boldsymbol{x})}{\partial \boldsymbol{x}} = \boldsymbol{P}\boldsymbol{x} + \nabla\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{W} + \nabla\boldsymbol{\varepsilon}^{\mathrm{T}}(\boldsymbol{x}), \quad (9)$$

where $\nabla\boldsymbol{\Phi}(\boldsymbol{x}) \triangleq \left[\frac{\partial \boldsymbol{\Phi}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right]^{\mathrm{T}}$ is the gradient of $\boldsymbol{\Phi}(\boldsymbol{x})$, and $\nabla\boldsymbol{\varepsilon}(\boldsymbol{x}) \triangleq \left[\frac{\partial \varepsilon(\boldsymbol{x})}{\partial \boldsymbol{x}}\right]^{\mathrm{T}}$ is the gradient of $\varepsilon(\boldsymbol{x})$.

Due to the approximation error $\varepsilon(\boldsymbol{x})$, the associated control law is near optimal:

$$\boldsymbol{u}(\boldsymbol{x}) = -\frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}(\boldsymbol{x})\left[\boldsymbol{P}\boldsymbol{x} + \nabla\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{x})\boldsymbol{W}\right]. \quad (10)$$

This near-optimal control results in:

$$\varepsilon_H = \left[\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}} + \hat{\boldsymbol{W}}^{\mathrm{T}}\nabla\boldsymbol{\Phi}(\boldsymbol{x})\right][\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{u}(\boldsymbol{x})] + \bar{Q}(\boldsymbol{x}) + \bar{R}(\boldsymbol{u}), \quad (11)$$

where ε_H is the HJB approximation error due to $\varepsilon(\boldsymbol{x})$ in (8).

Recall that (8) contains a double-layer NN (i.e. $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{x})$), which is nonlinear in the hidden layer $\boldsymbol{\Phi}(\boldsymbol{x})$ but linear in the output layer weights $\boldsymbol{W}$. Let $\hat{\boldsymbol{W}}$ be the estimate of the ideal weights. To implement policy iteration, $\hat{\boldsymbol{W}}$ need to be tuned dynamically so that $\hat{\boldsymbol{W}} \to \boldsymbol{W}$ and thus (8) approximates a target value function. In this case,

$$\hat{V}(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}\boldsymbol{x} + \hat{\boldsymbol{W}}^{\mathrm{T}}\boldsymbol{\Phi}(\boldsymbol{x}), \quad (12)$$

$$\hat{\boldsymbol{u}}(\boldsymbol{x}) = -\frac{1}{2}\boldsymbol{R}^{-1}\boldsymbol{g}^{\mathrm{T}}(\boldsymbol{x})\left[\boldsymbol{P}\boldsymbol{x} + \nabla\boldsymbol{\Phi}^{\mathrm{T}}(\boldsymbol{x})\hat{\boldsymbol{W}}\right]. \tag{13}$$

The resulting nonlinear Lyapunov equation then becomes:

$$\left[\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}} + \hat{\boldsymbol{W}}^{\mathrm{T}}\nabla\boldsymbol{\Phi}(\boldsymbol{x})\right][\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\hat{\boldsymbol{u}}(\boldsymbol{x})] + \bar{Q}(\boldsymbol{x}) + \bar{R}(\hat{\boldsymbol{u}}) = \varepsilon_H + \xi, \tag{14}$$

where ξ is the error of weights estimation in a tuning process.

For uniform convergence of $\hat{\boldsymbol{W}}$ to the ideal $\boldsymbol{W}$ so that ξ is minimized, an extended Kalman filter (EKF) can be used. Since $\hat{\boldsymbol{W}}$ is the parameter vector to be estimated, (14) can be rearranged in the following form:

$$\begin{cases} \dot{\hat{\boldsymbol{W}}} = \boldsymbol{0} + \boldsymbol{w} \\ y = h(\boldsymbol{x}, \hat{\boldsymbol{W}}) - \varepsilon_H - \xi + v \end{cases}, \tag{15}$$

with $y = [\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\hat{\boldsymbol{u}}(\boldsymbol{x})] - \bar{Q}(\boldsymbol{x}) - \bar{R}(\hat{\boldsymbol{u}})$, and

$$h(\boldsymbol{x}, \hat{\boldsymbol{W}}) = \left[\boldsymbol{x}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}} + \hat{\boldsymbol{W}}^{\mathrm{T}}\nabla\boldsymbol{\Phi}(\boldsymbol{x})\right][\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\hat{\boldsymbol{u}}(\boldsymbol{x})],$$

where $\boldsymbol{0}$ is a null matrix (since $\hat{\boldsymbol{W}}$ is a constant vector), $\boldsymbol{w}$ and v are white-noise inputs with covariance matrix $\boldsymbol{Q}_f \succ 0$ and $\boldsymbol{R}_f \succ 0$, respectively.

Remark 2: In the system described by (15), $\hat{\boldsymbol{W}}$ are system states, and there are no drift dynamics for $\hat{\boldsymbol{W}}$. However, nonlinearities are present in the output dynamics, which are associated with $\nabla\boldsymbol{\Phi}(\boldsymbol{x})$ as well as $\dot{\boldsymbol{x}}$. Thus a nonlinear observer is needed to estimate the system states $\hat{\boldsymbol{W}}$.

Remark 3: White-noise inputs $\boldsymbol{w}$ and v in fact do not physically exist in the system of (15). Therefore, the corresponding covariance matrices $\boldsymbol{Q}_f$ and $\boldsymbol{R}_f$ have no physical implication. $\boldsymbol{w}$ and v is given in (15) purely in support of the use of $\boldsymbol{Q}_f$ and $\boldsymbol{R}_f$ for EKF implementation [31].

Introducing an EKF into the system of (15) yields:

$$\begin{cases} \dot{\hat{\boldsymbol{W}}} = \boldsymbol{K}_f(y - \hat{y}) \\ \hat{y} = h(\boldsymbol{x}, \hat{\boldsymbol{W}}) \end{cases}, \tag{16}$$

where $\hat{y}$ denotes the estimated output and $\boldsymbol{K}_f \in \mathbb{R}^{N\times 1}$ is the EKF gain.

The EKF gain $\boldsymbol{K}_f$ can be computed from:

$$\boldsymbol{K}_f = \boldsymbol{S}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}_f^{-1}, \tag{17}$$

$$\boldsymbol{H}^{\mathrm{T}} = \frac{\partial h(\boldsymbol{x}, \hat{\boldsymbol{W}})}{\partial \hat{\boldsymbol{W}}} = \nabla\boldsymbol{\Phi}(\boldsymbol{x})[\boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\hat{\boldsymbol{u}}(\boldsymbol{x})], \tag{18}$$

$$\dot{\boldsymbol{S}} = \boldsymbol{Q}_f - \boldsymbol{S}\boldsymbol{H}^{\mathrm{T}}\boldsymbol{R}_f^{-1}\boldsymbol{H}\boldsymbol{S}, \tag{19}$$

where $\boldsymbol{H} \in \mathbb{R}^{N\times 1}$ is defined as in (18), and $\boldsymbol{S} \in \mathbb{R}^{N\times N}$ is a symmetrical positive-definite matrix with initial state $\boldsymbol{S}(0) = \boldsymbol{S}(0)^{\mathrm{T}} \succeq 0$.

Note that the matrix $\boldsymbol{P}$ proposed and discussed in [24] is constant and hence does not suit a wide flight envelop with varying airspeed U_∞. To overcome this problem, we propose a systematic approach in the following for the selection of $\boldsymbol{P}$ to cope with U_∞ dependent dynamics as in (1).

Linearizing (1) about $\boldsymbol{x} = 0$ gives:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}_p(U_\infty)\boldsymbol{x} + \boldsymbol{B}_p\boldsymbol{u} + \boldsymbol{w}_p, \tag{20}$$

where

$$\boldsymbol{A}_p(U_\infty) \triangleq \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x}, U_\infty)}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=0}, \quad \boldsymbol{B}_p \triangleq \left.\frac{\partial \boldsymbol{g}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=0},$$

and $\boldsymbol{w}_p$ is unit white-noise input.

With performance output $\boldsymbol{z}$ considered, there is:

$$\begin{bmatrix} \dot{\boldsymbol{x}} \\ \boldsymbol{z} \\ \boldsymbol{y} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_p(U_\infty) & \boldsymbol{I} & \boldsymbol{B}_p \\ \boldsymbol{C}_z & 0 & \boldsymbol{E}_z \\ \boldsymbol{C}_p & 0 & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{w} \\ \boldsymbol{u} \end{bmatrix}, \tag{21}$$

where $\boldsymbol{C}_p = \boldsymbol{I}$ for full-state feedback, $\boldsymbol{C}_z = [\boldsymbol{Q}^{\frac{1}{2}} \quad 0]^{\mathrm{T}}$, and $\boldsymbol{E}_z = [0 \quad \boldsymbol{R}^{\frac{1}{2}}]^{\mathrm{T}}$.

Let $\boldsymbol{P}(U_\infty)$ be a matrix scheduled with U_∞. A Lyapunov matrix $\boldsymbol{X}(U_\infty) = \boldsymbol{X}^{\mathrm{T}}(U_\infty) \succ 0$ and an auxiliary parameter-dependent performance variable $\boldsymbol{Z}(U_\infty)$ are introduced to form the following linear matrix inequalities (LMIs):

$$\begin{bmatrix} \dot{\boldsymbol{X}} + \boldsymbol{A}_c^{\mathrm{T}}\boldsymbol{X} + \boldsymbol{X}\boldsymbol{A}_c & \boldsymbol{X} \\ \boldsymbol{X} & -\nu\boldsymbol{I} \end{bmatrix} \prec 0, \tag{22}$$

$$\begin{bmatrix} \boldsymbol{X} & \boldsymbol{C}_{cz}^{\mathrm{T}} \\ \boldsymbol{C}_{cz} & \boldsymbol{Z} \end{bmatrix} \succ 0, \tag{23}$$

and

$$\mathrm{Tr}(\boldsymbol{Z}) < \nu, \tag{24}$$

where

$$\boldsymbol{A}_c \triangleq \boldsymbol{A}_p(U_\infty) - \frac{1}{2}\boldsymbol{B}_p\boldsymbol{R}^{-1}\boldsymbol{B}_p^{\mathrm{T}}\boldsymbol{P}(U_\infty),$$

$$\boldsymbol{C}_{cz} \triangleq \boldsymbol{C}_z - \frac{1}{2}\boldsymbol{E}_z\boldsymbol{R}^{-1}\boldsymbol{B}_p^{\mathrm{T}}\boldsymbol{P}(U_\infty),$$

and ν is a performance index.

Let $\bar{\boldsymbol{B}}_c \triangleq \frac{1}{2}\boldsymbol{B}_p\boldsymbol{R}^{-1}\boldsymbol{B}_p^{\mathrm{T}}$, $\boldsymbol{G}(U_\infty) = \boldsymbol{P}(U_\infty)\boldsymbol{X}^{-1}(U_\infty)$, $\boldsymbol{Y}(U_\infty) = \boldsymbol{X}^{-1}(U_\infty)$, then (22) and (23) can be transformed into:

$$-\dot{\boldsymbol{Y}} + \boldsymbol{A}_p\boldsymbol{Y} + \boldsymbol{Y}\boldsymbol{A}_p^{\mathrm{T}} + \bar{\boldsymbol{B}}_c\boldsymbol{G} + \boldsymbol{G}^{\mathrm{T}}\bar{\boldsymbol{B}}_c^{\mathrm{T}} \prec 0, \tag{25}$$

$$\begin{bmatrix} \boldsymbol{Y} & (\boldsymbol{C}_z\boldsymbol{Y} + \boldsymbol{E}_z\boldsymbol{G})^{\mathrm{T}} \\ \boldsymbol{C}_z\boldsymbol{Y} + \boldsymbol{E}_z\boldsymbol{G} & \boldsymbol{Z} \end{bmatrix} \succ 0. \tag{26}$$

$\boldsymbol{A}(U_\infty)$ can be structured as:

$$\boldsymbol{A}(U_\infty) = \boldsymbol{A}_1 + \boldsymbol{A}_2U_\infty + \boldsymbol{A}_3U_\infty^2. \tag{27}$$

Thus $\boldsymbol{Y}(U_\infty)$ and $\boldsymbol{G}(U_\infty)$ take the same structure as:

$$\boldsymbol{Y}(U_\infty) = \boldsymbol{Y}_1 + \boldsymbol{Y}_2U_\infty + \boldsymbol{Y}_3U_\infty^2, \tag{28}$$

and

$$\boldsymbol{G}(U_\infty) = \boldsymbol{G}_1 + \boldsymbol{G}_2U_\infty + \boldsymbol{G}_3U_\infty^2. \tag{29}$$

Solving for $\boldsymbol{Y}(U_\infty)$ and $\boldsymbol{G}(U_\infty)$ through (24) to (26) gives $\boldsymbol{P}(U_\infty)$ in the form of:

$$\boldsymbol{P}(U_\infty) = \boldsymbol{G}(U_\infty)\boldsymbol{Y}^{-1}(U_\infty). \tag{30}$$

C. Identifier NN

The information of $\boldsymbol{f}(\boldsymbol{x}, U_\infty)$ and $\boldsymbol{g}(\boldsymbol{x})$ is required for the real-time synthesis of nonlinear optimal control laws. Although the knowledge of $\boldsymbol{f}(\boldsymbol{x}, U_\infty)$ and $\boldsymbol{g}(\boldsymbol{x})$ is analytically available, the presence of un-modeled dynamics or uncertainties can degrade controller performance as discussed in Section I. To mitigate this problem, an NN-based identifier is proposed in the following form:

$$\dot{\boldsymbol{x}} = \boldsymbol{W}_s^{\mathrm{T}} \boldsymbol{\Phi}_s(\boldsymbol{x}, \boldsymbol{u}) + \boldsymbol{\varepsilon}_s, \tag{31}$$

where $\boldsymbol{W}_s^{\mathrm{T}} \in \mathbb{R}^{n_{ws} \times n_x}$ and $\boldsymbol{\Phi}_s(\boldsymbol{x}, \boldsymbol{u}) \in \mathbb{R}^{n_{ws}}$ are the ideal NN weights and nonlinear activation functions, respectively.

In light of [32], the system states $\boldsymbol{x}$ can be filtered as:

$$\boldsymbol{x} = \boldsymbol{W}_s^{\mathrm{T}} \boldsymbol{\eta}(\boldsymbol{x}) + \boldsymbol{A}\boldsymbol{\mu}(\boldsymbol{x}) + \boldsymbol{\varepsilon}_x, \tag{32}$$

$$\dot{\boldsymbol{\eta}}(\boldsymbol{x}) = -\boldsymbol{A}\boldsymbol{\eta}(\boldsymbol{x}) + \boldsymbol{\Phi}(\boldsymbol{x}, \boldsymbol{u}), \quad \boldsymbol{\eta}(\boldsymbol{x}_0) = 0, \tag{33}$$

$$\dot{\boldsymbol{\mu}}(\boldsymbol{x}) = -\boldsymbol{A}\boldsymbol{\mu}(\boldsymbol{x}) + \boldsymbol{x}, \quad \boldsymbol{\mu}(\boldsymbol{x}_0) = 0, \tag{34}$$

where $\boldsymbol{\eta}(\boldsymbol{x}) \in \mathbb{R}^{n_{ws}}$ and $\boldsymbol{\mu}(\boldsymbol{x}) \in \mathbb{R}^{n_x}$ are auxiliary regressors, $\boldsymbol{A} = \sigma \boldsymbol{I}_{n_x \times n_x}$ with $\sigma \in \mathbb{R}^+$, and $\boldsymbol{\varepsilon}_x = e^{-\boldsymbol{A}t}\boldsymbol{x}_0 + \int_0^t e^{-\boldsymbol{A}(t-\tau)}\varepsilon_s d\tau$.

Denote the estimate of $\boldsymbol{x}$ by $\hat{\boldsymbol{x}}$. For fast estimation of $\hat{\boldsymbol{W}}_s$ towards $\boldsymbol{W}_s$, the EKF is considered for online tuning. In this study, multiple EKFs in a parallel configuration instead of a single EKF are employed, based on the fact that the columns of $\boldsymbol{W}_s$ are exclusively associated with the respective single state of $\boldsymbol{x}$. By doing so, the computational expense is significantly less than using only one multi-input multi-output EKF [31]. On this basis, we have:

$$\begin{cases} \dot{\hat{\boldsymbol{W}}}_{s(i)} = \boldsymbol{K}_{s(i)}(\boldsymbol{x}_{(i)} - \hat{\boldsymbol{x}}_{(i)}) \\ \hat{\boldsymbol{x}}_{(i)} = \hat{\boldsymbol{W}}_{s(i)}\boldsymbol{\eta}(\boldsymbol{x}) + \boldsymbol{A}\boldsymbol{\mu}(\boldsymbol{x}) \end{cases}, \tag{35}$$

where $\boldsymbol{K}_s \in \mathbb{R}^{n_w s}$ is the EKF gain and subscript (i) restricts the parameters to the i^{th} decoupled EKF.

Each EKF gain vector $\boldsymbol{K}_{s(i)}$ can be computed by:

$$\boldsymbol{K}_{s(i)} = \boldsymbol{S}_{s(i)} \boldsymbol{H}_{s(i)}^{\mathrm{T}} R_s^{-1}, \tag{36}$$

$$\boldsymbol{H}_{s(i)}^{\mathrm{T}} = \frac{\partial \hat{\boldsymbol{x}}_{(i)}}{\partial \hat{\boldsymbol{W}}_{s(i)}} = \boldsymbol{\eta}(\boldsymbol{x}), \tag{37}$$

$$\dot{\boldsymbol{S}}_{s(i)} = \boldsymbol{Q}_s - \boldsymbol{S}_{s(i)} \boldsymbol{H}_{s(i)}{}^{\mathrm{T}} R_s^{-1} \boldsymbol{H}_{s(i)} \boldsymbol{S}_{s(i)}. \tag{38}$$

where $\boldsymbol{Q}_s$ and R_s are defined the same as $\boldsymbol{Q}_f$ and $\boldsymbol{R}_f$.

As can be seen from (37), $\boldsymbol{H}_{s(i)}$ is the same among the individual EKFs in the parallel configuration. It is also to be noted $\boldsymbol{H}_{s(i)}$ is not constant but state-dependent, indicating the nonlinearities involved in system identification, which justifies the use of the EKF instead of linear observers.

The internal dynamics $\boldsymbol{f}(\boldsymbol{x}, U_\infty)$ can be given by the NN identifier in an indirect manner, since the estimated derivative of system states used by (16) can be obtained through (31), with $\boldsymbol{W}_s$ and $\boldsymbol{u}$ replaced by $\hat{\boldsymbol{W}}_s$ and $\hat{\boldsymbol{u}}$, respectively.

The input dynamics $\boldsymbol{g}(\boldsymbol{x})$ can then be obtained as:

$$\hat{\boldsymbol{g}}(\boldsymbol{x}) = \frac{\partial \hat{\mathcal{F}}(\boldsymbol{x}, \hat{\boldsymbol{u}})}{\partial \hat{\boldsymbol{u}}} = \frac{\partial \hat{\boldsymbol{W}}_s^{\mathrm{T}} \boldsymbol{\Phi}_s(\boldsymbol{x}, \hat{\boldsymbol{u}})}{\partial \hat{\boldsymbol{u}}} = \frac{\partial \boldsymbol{\Phi}_s(\boldsymbol{x}, \hat{\boldsymbol{u}})}{\partial \hat{\boldsymbol{u}}} \hat{\boldsymbol{W}}_s. \tag{39}$$

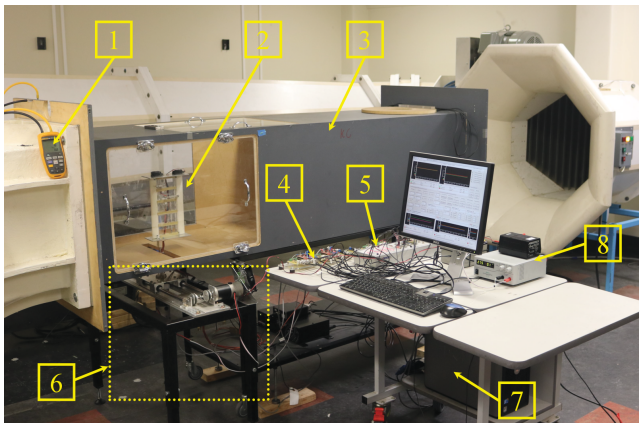

Fig. 1. Wind tunnel experiment setup (1, pressure transducer; 2, airfoil section; 3, wind tunnel test duct; 4, custom I/O board; 5, dSPACE® DS1104 R&D controller board; 6, virtual spring-damper system; 7, controller PC; 8, power source)

Remark 4: The use of the Identifier NN is mainly for updating the knowledge of system dynamics accessed by the real-time controller synthesis scheme. The known dynamics (i.e. known analytical model) are embedded into the Identifier NN in the form of the initial values of $\boldsymbol{W}_s$ obtained via pre-training the NN using known dynamics. Therefore, any mismatch between the actual dynamics and the analytical model can be captured and used to update $\boldsymbol{W}_s$ in real time.

IV. Wind-Tunnel Experiments

Experiments were performed in a temperature-regulated closed-loop wind tunnel at the University of Adelaide, Australia, and the setup is shown in Fig. 1. The wind-tunnel has a 0.5×0.5 m testing duct, and is able to generate up to 30 m/s smooth airflow with 0.5% turbulence intensity. The leading- and trailing-edge control surfaces of the airfoil section are each driven by a servo motor, with the corresponding deployment angle fed back via an optical encoder. Instead of using physical springs for plunge and pitch stiffness, a virtual-spring-damper system (VSDS) was developed in this study, where two electric motors were used to mimic the elastic force/torque using force feedback from a 6-axis force/torque transducer. The parameters of the overall aeroelastic system are listed in Table I. With the stiffness and damping setting of the VSDS as in Table I, the system had a flutter boundary around 14.6 m/s. Flutter was initiated by means of giving the VSDS a pulse signal along the plunge DOF.

In consideration of the 3^{rd} order nonlinear plunge and pitch stiffness in polynomial form, a power series of activation functions containing the powers of 8 system states up to 4^{th} order and 2 control inputs limited to 1^{st} order were used for the Identifier NN in accordance with the high-order Weierstrass approximation theorem [30]. This renders 135 activation functions for $\boldsymbol{\Phi}_s(\boldsymbol{x})$. Initial weights were determined via simulation-based training for 14.6 m/s airspeed. Similarly, $\boldsymbol{\Phi}(\boldsymbol{x})$ of the Critic NN contains the powers of 8 system states only (no control inputs) up to 4^{th} order. This gives 65 activation functions. $\boldsymbol{M}(U_\infty)$ and

TABLE I

PARAMETERS OF THE EXPERIMENTAL AEROELASTIC SYSTEM

Prm.[a]	Values	Prm.[a]	Values
ρ	1.225 kg/m^3	c_h	14 kg/s
b	0.0753 m	c_α	0.042 kg·m^2/s
m_α	0.851 kg	$c_{\beta_{\text{servo}}}$	4.231×10^{-4} kg·m^2/s
m_β	0.030 kg	$c_{\gamma_{\text{servo}}}$	4.327×10^{-4} kg·m^2/s
m_γ	0.058 kg	k_h	$50 + 300h^2$ N/m
S	0.26 m	k_α	$0.3 + 30\alpha^2$ Nm/rad
$r_{3c/4}$	0.0805 m	$k_{\beta_{\text{servo}}}$	4.57×10^{-3} Nm/rad
r_α	0.0329 m	$k_{\gamma_{\text{servo}}}$	4.70×10^{-3} Nm/rad
r_β	0 m	$\hat{I}_\alpha$	2.431×10^{-3} kg·m^2
r_γ	0 m	$\hat{I}_\beta$	2.307×10^{-6} kg·m^2
$C_{l\alpha}$	6.573	$\hat{I}_\gamma$	4.791×10^{-6} kg·m^2
$C_{l\beta}$	3.472	L_β	0.0875 m
$C_{l\gamma}$	-0.1453	L_γ	-0.01 m
$C_{m_{\alpha,\text{eff.}}}$	-0.4505	$C_{m_{\gamma,\text{eff.}}}$	0.2066
$C_{m_{\beta,\text{eff.}}}$	-1.4993		

[a] Refer to [26] for the meaning of symbols.

$\boldsymbol{G}(U_\infty)$ were designed using the parameters in Table I for airspeeds from 14.6 m/s to 20 m/s with a griding of 50 evenly spaced points. $\boldsymbol{P}(U_\infty)$ was calculated in real time using (30). $\bar{\boldsymbol{Q}}$ in (4) was structured as $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{x}$, with $\boldsymbol{Q} = \text{diag}(1,\ 1,\ 10^{-4},\ 10^{-4},\ 0.1,\ 0.1,\ 10^{-4},\ 10^{-4})$. Other weightings are $\boldsymbol{R} = 100\boldsymbol{I}$, $\boldsymbol{Q}_f = 1000\boldsymbol{I}$, $\boldsymbol{R}_f = \boldsymbol{I}$, $\boldsymbol{Q}_{s(i)} = 1 \times 10^5\boldsymbol{I}$, and $R_{s(i)} = 1$.

Tests were conducted at two different airspeeds, and flutter was allowed to develop to reach LCO before the controller under testing was turned on. To ensure consistent initial conditions $\boldsymbol{x}(t_c)$ throughout all tests under the same settings, where t_c is the time when the controller is switched on, the controller was configured to be triggered when α crossed zero immediately after 15 seconds. This means $t_c > 15$ s. For comparison, an linear-parameter-varying (LPV) controller in the form of linear-quadratic-regulator (LQR) synthesized by means of LMIs [6] was reconstructed for the 4-DOF model as in [26] with the parameters in Table I and the weighting $\boldsymbol{Q}$ and $\boldsymbol{R}$ same as those used by the proposed NN controller.

Plunge and pitch responses as well as control surfaces deflections of the airfoil section in the wind-tunnel tests under the proposed NN controller and the LPV-LQR controller at different airspeeds are plotted in Fig. 2 for 14.8 m/s and Fig. 4 for 18 m/s. Higher airspeeds were not tested due to the torque output limit of the VSDS motors. For elegance of presentation and ease of reading, t_c is offset to zero in each plot and LCOs before controllers are activated are presented in dotted curves. Since both controllers are off before $t = t_c$, only the full trajectories of control surfaces deflections under the proposed controller are shown for illustration purpose. The trajectories of NNs weights are plotted in Figs. 3 and 5.

At 14.8 m/s, the flutter was effectively suppressed within 1.5 seconds under the proposed NN controller, with only mild demands on the deflection of control surfaces. By comparing Figs. 2 and 3, it can be seen the Identifier NN has higher rate of convergence than that of the Critic NN, which means the latter is able to access updated and more accurate system dynamics for control law improvement. The Critic NN also settles 1 second before the flutter is fully suppressed,

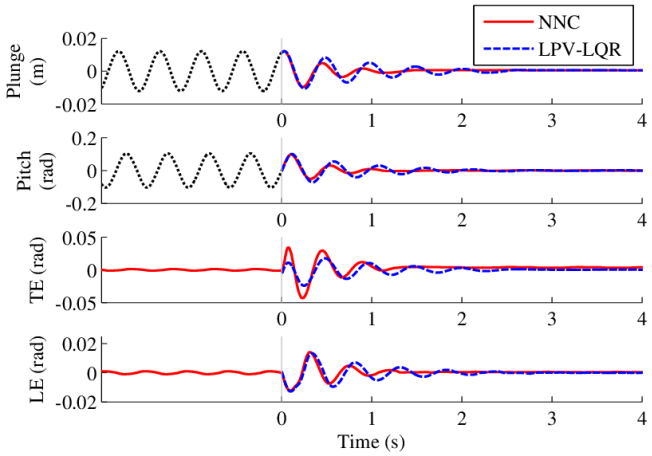

Fig. 2. Suppressing developed flutter at 14.8 m/s airflow speed using an LPV-LQR controller and the proposed NN controller (NNC)

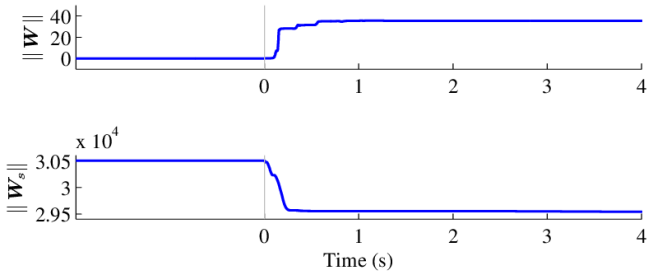

Fig. 3. Convergence trajectories of the Critic and Identifier NNs weights of the proposed controller at 14.8 m/s airflow speed

indicating satisfactory parameter convergence. This validates the selection of the activation functions sets for both the Identifier NN and the Critic NN, and also indicates that near-optimal control is obtained. In comparison, it takes longer for the LPV-LQR controller to fully suppress the flutter. Similar phenomena can be observed for 18 m/s, as shown in Figs. 4 and 5, where however, relatively larger differences between the the responses under the two controllers can be observed.

To better capture the difference between the two controllers, performance cost is evaluated for $t = 0 \to 4$ s using the experiment data with discrete approximation. Costs are each calculated and averaged from 4 tests under the same settings for data consistency. It is found that that the proposed NN control suppresses the flutter better with lower costs at both airspeeds (4.893 for 14.8 m/s; 0.545 for 18 m/s), compared with those of the LPV-LQR control (5.372 for 14.8 m/s; 0.627 for 18 m/s).

V. CONCLUSIONS

As validated in wind-tunnel experiments, the proposed controller successfully mitigates the impact of modeling uncertainties and improves ASAF from the optimal control perspective. The proposed LMI-based synthesis for the design of a scheduled parameter matrix to generalize the modified VFA and adapt it to the ASAF application is shown to be effective. Experiments also confirm that the proposed controller is suitable for real-time implementation.

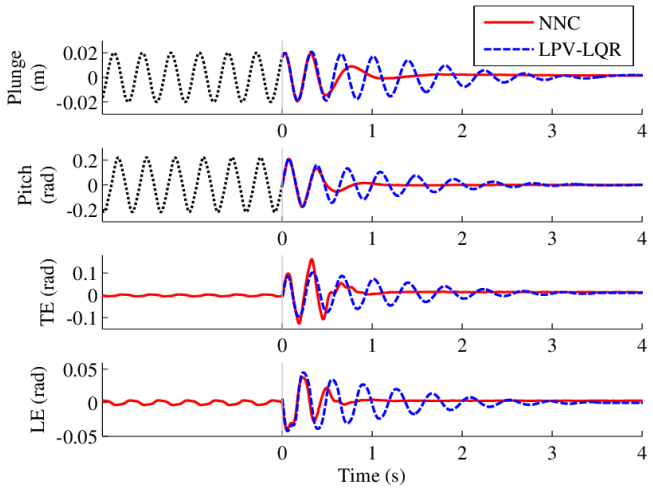

Fig. 4. Suppressing developed flutter at 18 m/s airflow speed using an LPV-LQR controller and the proposed NN controller (NNC)

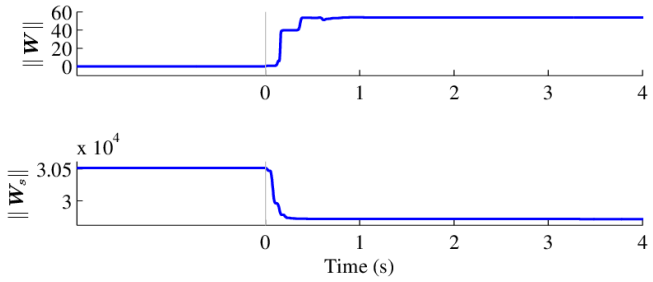

Fig. 5. Convergence trajectories of the Critic and Identifier NNs weights of the proposed controller at 18 m/s airflow speed

REFERENCES

[1] A. V. Balakrishnan, *Aeroelasticity: The Continuum Theory*. New York, NY: Springer, 2012.

[2] L. Chen, F. He, and K. Sammut, "Vibration suppression of a principal parametric resonance," *Journal of Vibration and Control*, vol. 15, no. 3, pp. 439–463, Feb 2009.

[3] M. G. Soto and H. Adeli, "Recent advances in control algorithms for smart structures and machines," *Expert Systems*, vol. 34, no. 2, pp. 1–14, Apr 2017.

[4] C.-G. Pak, P. P. Friedmann, and E. Livne, "Digital adaptive flutter suppression and simulation using approximate transonic aerodynamics," *Journal of Vibration and Control*, vol. 1, no. 4, pp. 363–388, Jan 1995.

[5] P. P. Friedmann, D. Guillot, and E. Presente, "Adaptive control of aeroelastic instabilities in transonic flow and its scaling," *Journal of Guidance, Control, and Dynamics*, vol. 20, no. 6, pp. 1190–1199, Nov-Dec 1997.

[6] Z. Prime, B. Cazzolato, C. Doolan, and T. Strganac, "Linear-parameter-varying control of an improved three-degree-of-freedom aeroelastic model," *Journal of Guidance, Control, and Dynamics*, vol. 33, no. 2, pp. 615–619, Mar-Apr 2010.

[7] G. Chen, J. Sun, and Y.-M. Li, "Adaptive reduced-order-model-based control-law design for active flutter suppression," *Journal of Aircraft*, vol. 49, no. 4, pp. 973–980, Jul-Aug 2012.

[8] J. Ko, T. W. Strganac, and A. J. Kurdila, "Adaptive feedback linearization for the control of a typical wing section with structural nonlinearity," *Nonlinear Dynamics*, vol. 18, no. 3, pp. 289–301, 1999.

[9] G. Platanitis and T. W. Strganac, "Control of a nonlinear wing section using leading- and trailing-edge surfaces," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 1, pp. 52–58, Jan-Feb 2004.

[10] J. Ko, T. W. Strganac, J. L. Junkins, M. R. Akella, and A. J. Kurdila, "Structured model reference adaptive control for a wing section with structural nonlinearity," *Journal of Vibration and Control*, vol. 8, no. 5, pp. 553–573, Jul 2002.

[11] W. Xing and S. N. Singh, "Adaptive output feedback control of a nonlinear aeroelastic structure," *Journal of Guidance, Control, and Dynamics*, vol. 23, no. 6, pp. 1109–1116, Nov-Dec 2000.

[12] S. N. Singh and L. Wang, "Output feedback form and adaptive stabilization of a nonlinear aeroelastic system," *Journal of Guidance, Control, and Dynamics*, vol. 25, no. 4, pp. 725–732, Jul-Aug 2002.

[13] K. Zhang and A. Behal, "Continuous robust control for aeroelastic vibration control of a 2-D airfoil under unsteady flow," *Journal of Vibration and Control*, vol. 22, no. 12, pp. 2841–2860, 2016.

[14] N. Bhoir and S. N. Singh, "Output feedback modular adaptive control of a nonlinear prototypical wing section," *Nonlinear Dynamics*, vol. 37, no. 4, pp. 357–373, 2004.

[15] V. M. Rao, A. Behal, P. Marzocca, and C. M. Rubillo, "Adaptive aeroelastic vibration suppression of a supersonic airfoil with flap," *Aerospace Science and Technology*, vol. 10, no. 4, pp. 309–315, 2006.

[16] J. J. Carnahan and C. M. Richards, "A modification to filtered-X LMS control for airfoil vibration and flutter suppression," *Journal of Vibration and Control*, vol. 14, no. 6, pp. 831–848, 2008.

[17] K. W. Lee and S. N. Singh, "$\mathcal{L}_1$ adaptive control of a nonlinear aeroelastic system despite gust load," *Journal of Vibration and Control*, vol. 19, no. 12, pp. 1807–1821, 2013.

[18] S. Dilmi and B. Bouzouia, "Improving performance for nonlinear aeroelastic systems via sliding mode controller," *Arabian Journal for Science and Engineering*, vol. 41, no. 9, pp. 3739–3748, Mar 2016.

[19] M.-Z. Gao and G.-P. Cai, "Finite-time fault-tolerant control for flutter involving control delay," *Journal of the Franklin Institute*, vol. 353, no. 9, pp. 2009–2029, Mar 2016.

[20] M.-Z. Gao, G.-P. Cai, and Y. Nan, "Finite-time fault-tolerant control for flutter of wing," *Control Engineering Practice*, vol. 51, pp. 26–47, Mar 2016.

[21] S. Gujjula, S. N. Singh, and W. Yim, "Adaptive and neural control of a wing section using leading- and trailing-edge surfaces," *Aerospace Science and Technology*, vol. 9, no. 2, pp. 161–171, Mar 2005.

[22] Z. Wang, A. Behal, and P. Marzocca, "Model-free control design for multi-input multi-output aeroelastic system subject to external disturbance," *Journal of Guidance, Control, and Dynamics*, vol. 34, no. 2, pp. 446–458, Mar-Apr 2011.

[23] C. Brillante and A. Mannarino, "Improvement of aeroelastic vehicles performance through recurrent neural network controllers," *Nonlinear Dynamics*, vol. 84, no. 3, pp. 1479–1495, Jan 2016.

[24] D. Tang, L. Chen, and Z. F. Tian, "Neural-network based online policy iteration for continuous-time infinite-horizon optimal control of nonlinear systems," in *Proceedings of the 3rd IEEE China Summit and International Conference on Signal and Information Processing*, Chengdu, China, Jul 2015.

[25] T. O'Neil and T. W. Strganac, "Aeroelastic response of a rigid wing supported by nonlinear springs," *Journal of Aircraft*, vol. 35, no. 4, pp. 616–622, Jul-Aug 1998.

[26] Z. D. Prime, "Robust scheduling control of aeroelasticity," Ph.D. Thesis, School of Mechanical Engineering, The University of Adelaide, Adelaide, Australia, 2010.

[27] M. Abu-Khalaf and F. L. Lewis, "Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach," *Automatica*, vol. 41, no. 5, pp. 779–791, May 2005.

[28] G. N. Saridis and C.-S. G. Lee, "An approximation theory of optimal control for trainable manipulators," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 9, no. 3, pp. 152–159, Mar 1979.

[29] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press, 1998.

[30] B. A. Finlayson, *The method of weighted residuals and variational principles: With application in fluid mechanics, heat and mass transfer*. New York, NY: Academic Press, 1972.

[31] D. Simon, "Training radial basis neural networks with the extended Kalman filter," *Neurocomputing*, vol. 48, no. 1-4, pp. 455–475, Oct 2002.

[32] H. Modares, F. L. Lewis, and M.-B. Naghibi-Sistani, "Adaptive optimal control of unknown constrained-input systems using policy iteration and neural networks," *Neural Networks and Learning Systems, IEEE Transactions on*, vol. 24, no. 10, pp. 1513–1525, Oct 2013.