Flutter Control of a Two-dimensional Airfoil based on Adaptive Dynamic Programming

Su Jia, Jian Qiao Sun, Qian Ding

School of Mechanics, Tianjin University, Tianjin 300072, China;
School of Engineering, University of California, Merced, CA, 95343, USA

Abstract. The severe aeroelastic vibrations can lead to structural damage and threaten the flight safety. It is an important topic to develop active controllers for aeroelastic stabilization and flutter suppression. The traditional control methods need the clearly established mathematic model, whereas establishing an accurate airfoil model with structural and aerodynamic nonlinearities is a difficult task. So a novel data-driven optimal design scheme, adaptive dynamic programming (ADP), is proposed in this paper for a nonlinear airfoil system with one trailing-edge control surface. The proposed ADP algorithm iteratively updates the control policy by using the state and input data without knowing the system dynamical model. Calculation shows that the control policy is convergent after several iterations. Finally, the superiority and feasibility of the control method are verified by simulation.

1. Introduction
Flutter is a nonlinear dynamic aeroelastic phenomenon in many areas, especially in aircraft structures. Under certain conditions, the airfoil experiences flutter or self-excitation behavior in which aerodynamic loads coupled with elastic and inertial nature of the structure [1]. Flutter instability can both degenerate the aircraft performance and lead to catastrophic failure of the structure. It is well known that the structure nonlinearity and aerodynamic effects can lead to complex unstable behaviors.

Investigations on dynamics of aeroelastic systems have been carried out on the flutter characteristics of folding control fins with free play [3] and reduced order aerodynamic model [2, 4] through both numerical simulation and flutter wind tunnel tests. The nonlinear dynamic behavior such as limit cycle oscillation (LCO), bifurcation or chaos in transonic air flow has been demonstrated systematically in these studies. In recent years, much attentions have been focused on how to suppress the airfoil flutter. Lots of passive methods are suggested to avoid airfoil flutter by modifying the structure. Guo [5] applied the harmonic balance method to study the influence of different nonlinear energy sink (NES) structural parameters on the limit cycle flutter of a typical 2-DOF wing. Both linear and nonlinear tuned vibration absorbers were applied in [6] to increase the flutter speed of the aeroelastic system with two degrees of freedom. In [7] the influence of nonlinear tuned vibration absorber on the airfoil flutter was also investigated. The result shows the ability of the nonlinear tuned vibration absorber can avoid its possible subcritical and reduce the LCO amplitude. However, those approaches will inevitably lead to weight increase and degrade performance for the airfoil.

The extensive studies prove that the airfoil control surfaces can be used to actively suppress the flutter effectively [11]. The use of active flutter suppression can increase the critical flutter velocity by activating the control surfaces according to the bending and torsion of the wing [8-10].
surfaces are mainly driven by hydraulic actuators or electric motors. A delayed feedback controller designed by the sliding mode control method was used to the airfoil control surface [8]. Xue [9] considered the nonlinear pitch spring stiffness then an adaptive backstepping sliding mode controller was developed to eliminate the limit cycle response. Partial feedback linearization was utilized to design the nonlinear aeroelastic controller [10]. However, most active control designs need to be based on an accurate airfoil model. There are limited researches on the model-free control design. Although some control methods can be more adaptive to the unmodeled dynamics, they are not the optimal control.

Adaptive Dynamic Programming (ADP) [12] is becoming a popular and effective data-driven approach, especially by incorporating the idea of reinforcement learning [13] to get rid of the limitations of developing the model. By achieving online approximation of the cost function, the ADP method can be applied without the knowledge of the system dynamics [14]. As the most commonly used structure in ADP, the critic NN is used to assess the control strategy, while the actor NN is used to improve the control strategy by iteration. In [15] Vrabie and Lewis proved the stability and convergence of the online adaptive dynamic programming algorithm. Based on lots of process data the ADP was applied to single cell flotation industrial process [16]. Yang [17] presented a novel robust ADP method which only needs partial system model and illustrated the applicability through a real power system. Inspired by these researches, the ADP method is used to design the data-driven active controller for the LCO flutter.

This paper is organized as follows. In Section 2 we briefly introduce the two-dimensional typical airfoil model. In Section 3 we develop the data-driven adaptive optimal control method. In Section 4 we apply the proposed approach to the airfoil system and present the simulation results. Finally, in section 5 we provide the conclusions.

2. Nonlinear aeroelastic model
The dynamical model of a typical two-dimensional airfoil is shown in Figure 1 [9,10]. The system has two degrees of freedom $h, \alpha$ and has one trailing edge flap control surface $\beta$. Using the Lagrange theorem the governing equation of the wing section is deduced as,

$$\begin{align*}
\frac{m}{h} + mx_{\alpha}b\ddot{\alpha} + c_h h + k_h h = -L \\
I\ddot{\alpha} + mx_{\alpha}b\dot{h} + c_{\alpha} \alpha + k_{\alpha}(\alpha)\alpha = M
\end{align*}$$

where $b$ is the semichord of the wing, $x_{\alpha}$ the dimensionless distance between the center of mass and the elastic axis, $m$ and $I$ the mass and the moment of inertia of the wing, $L$ and $M$ the aerodynamic lift and moment, $c_h$ and $c_{\alpha}$ the damping coefficients of the plunge displacement and the pitch angle. In this model $k_h$ is the spring stiffness coefficient of the plunge displacement and $k_{\alpha}$ is the nonlinear spring stiffness coefficient of the pitch angle.

![Figure 1. The dynamical model of the two-dimensional typical airfoil.](image)

The quasi-steady aerodynamic model [10] is adopted, which includes the aerodynamic lift and pitch moment with the flap angle effect,
\[
\begin{aligned}
L &= -\rho V^2 b C^\alpha_x \left[ \alpha + \frac{b}{u} + \frac{(1-a)ab}{u^2} \right] - \rho U^2 b C^\beta_x \beta, \\
M &= -\left( \frac{1}{2} + a \right) b L - \frac{1}{2} \pi \rho U b^3 \alpha + 2 \rho U^2 b^2 C^\beta_m \beta.
\end{aligned}
\] (2)

where \( \rho \) represents the air density, \( a \) the dimensionless distance between the elastic axis and the midchord, and \( U \) the flow velocity.

The dynamics equation (2) can be transformed into a state space expression (3) by defining state variables as \( x_1 = h, x_2 = \alpha, x_3 = \dot{h}, x_4 = \dot{\alpha} \).

\[
\begin{aligned}
\dot{x}_1 &= x_2, \\
\dot{x}_2 &= x_3, \\
\dot{x}_3 &= -k_1 x_1 - [k_2 U^2 + p(x_2)] x_2 - c_1 x_3 - c_2 x_4 + g_3 U^2 u, \\
\dot{x}_4 &= -k_3 x_1 - [k_4 U^2 + q(x_2)] x_2 - c_3 x_3 - c_4 x_4 + g_4 U^2 u.
\end{aligned}
\] (3)

where the parameters are defined as

\[
\begin{align*}
d &= m_T l_a - m_w x_a b^2, \\
k_1 &= l_a k_h / d, \\
k_2 &= (l_a \rho b c_{ia} + m_w x_a \rhoUb^3 c_{ma}) / d, \\
k_3 &= -m_w x_a b k_h / d, \\
p(x) &= -m_w x_a b k_a(x) / d, \\
q(x) &= m_T k_a(x) / d, \\
c_1 &= [l_a (c_h + \rhoUb c_{ia}) + m_w x_a \rhoUb^3 c_{ma}] / d, \\
c_2 &= [l_a \rhoUb^2 c_{ia} \left( \frac{1}{2} - a \right) - m_w x_a b c_{ia} + m_w x_a \rhoUb^3 c_{ma} \left( \frac{1}{2} - a \right)] / d, \\
c_3 &= -[m_w x_a b (c_h + \rhoUb c_{ia}) + m_T \rhoUb^2 c_{ma}] / d, \\
c_4 &= \left[ m_T \left( c_a - \rhoUb^3 c_{ma} \left( \frac{1}{2} - a \right) \right) - m_w x_a \rhoUb^3 c_{ia} \left( \frac{1}{2} - a \right) \right] / d, \\
g_3 &= -(l_a \rho b c_{i\beta} + m_w x_a b^2 \rho c_{\beta}) / d, \\
g_2 &= (m_w x_a \rho b^2 c_{i\beta} + m_T \rho b^2 c_{\beta}) / d.
\end{align*}
\]

The nonlinear pitch spring stiffness is determined by [10]

\[
k_a(\alpha) = 6.833 + 9.967 \alpha + 667.685 \alpha^2 + 26.569 \alpha^3 - 5087.931 \alpha^4 \text{(Nm/rad)}.
\]

3. ADP for optimal control

3.1. Problem description

For simplification, the nonlinear dynamic equation of the airfoil can be written as

\[
\dot{x} = f(x) + g(x)u,
\] (4)

where \( x(t) \subset \mathbb{R}^4 \) is the state variable vector, \( u(t) \subset \mathbb{R} \) is the control variable. We assume that the system functions \( f(\cdot) \in \mathbb{R}^4 \) and \( g(\cdot) \in \mathbb{R}^{4 \times 1} \) are unknown matrices and are differentiable in the arguments satisfying \( f(0) = 0 \).

By defining the infinite horizon cost function as
\[ V(x(t_0)) = \int_{t_0}^{\infty} r(x(t), u(x(t))) dt, \] (5)

where \( r(x(t), u(x(t))) = x^T Q x + u^T Ru \), we can get the Hamiltonian function

\[ H(x, u, V) = r\left(x(t), u(x(t))\right) + (\nabla V)^T [f(x) + g(x)u]. \] (6)

\( \nabla V \) denotes the gradient of the cost function \( V(x) \) with respect to \( x(t) \). Then substituting the optimal cost function \( V^*(x) \) into equation (6), the HJB equation can be obtained

\[ 0 = \min_{u(x(t))} r\left(x(t), u(x(t))\right) + (\nabla V^*)^T [f(x) + g(x)u], \] (7)

and we can solve \( u(x(t)) \) by the equation

\[ \frac{\partial H}{\partial u} = 0. \] (8)

The optimal feedback control policy is determined by

\[ u^*(x(t)) = -\frac{1}{2} R^{-1} g^T(x) \nabla V^*. \] (9)

Substituting the optimal control policy into (6) and then combing with (7), we can get the expression of the HJB equation

\[ (\nabla V^*)^T f(x) + Q(x) - \frac{1}{2} (\nabla V^*)^T g(x) R^{-1} g^T(x) \nabla V^* = 0, \] (10)

with \( V^*(0) = 0 \). By solving the HJB equation, we get the optimal control. However, it is difficult to find its analytical solution.

### 3.2. Date-based Policy Iteration

The policy iterative algorithm is adopted to get the solution of HJB equation. We apply a data-driven policy iteration method by using the system input and output data to learn the solution of the HJB equation. The control policy is continuously optimized and eventually converges to the optimal by alternately using policy evaluation and policy improvement.

In order to make it clear we write the system as

\[ \dot{x} = f + gu^{(i)} + g\left[u - u^{(i)}\right], \] (11)

and derive the cost function \( V^{(i+1)}(x) \) versus time

\[ \frac{dV^{(i+1)}(x)}{dt} = (\nabla V^{(i+1)})^T (f + gu^{(i)}) + (\nabla V^{(i+1)})^T g(u - u^{(i)}) \]

\[ = r(x, u^{(i)}) + 2[u^{(i+1)}] R [u^{(i)} - u]. \] (12)

The following equation (13) can be obtained by integrating both sides of the equation (12) in the interval \([t, t + \Delta t]\).

\[ V^{(i+1)}(x(t)) - V^{(i+1)}(x(t + \Delta t)) = -2 \int_t^{t+\Delta t} [u^{(i+1)}(x(\tau))] R [u^{(i)}(x(\tau)) - u(\tau)] d\tau + \int_t^{t+\Delta t} r(x, u^{(i)}) d\tau. \] (13)

### 3.3. Implementation of the ADP method

By using the data-based ADP method [18] without depending on the system model we apply actor-critic structure to solve equation (13). Actor NNs are used to approximate the cost function and critic NNs are used to approximate the control policy as follows.
\[ \mathcal{V}^{(i)}(x) = \sum_{j=1}^{k_v} \theta^{(i)}_{v,j} \phi_j(x) = \phi^T(x) \theta^{(i)}_v, \]

\[ \mathcal{U}^{(i)}(x) = \sum_{k=1}^{k_u} \theta^{(i)}_{u,k} \psi_k(x) = \psi^T(x) \theta^{(i)}_u, \]

where \( j \) and \( k \) denote the hide layer neuron number. \( \theta^{(i)}_v \) and \( \theta^{(i)}_u \) are weight vectors to be determined. By substituting the actor NNs and critic NNs into equation (13), the following residual error form can be obtained.

\[ \sigma^{(i)}(x(t), u(t), x(t + \Delta t)) \]

\[ = [\phi(x(t)) - \phi(x(t + \Delta t))]^T \theta^{(i+1)}_v + 2 \int_t^{t+\Delta t} [u^{(i)}(x(\tau)) - u(\tau)]^T R u^{(i+1)}(x(\tau)) \, d\tau \]

\[ - \int_t^{t+\Delta t} r(x(\tau), u^{(i)}(\tau)) \, d\tau \]

\[ = [\phi(x(t)) - \phi(x(t + \Delta t))]^T \theta^{(i+1)}_v + 2 \int_t^{t+\Delta t} [\psi^T(x(\tau)) \theta^{(i)}_u - u] R \psi^T(x(\tau)) \theta^{(i+1)}_u \, d\tau \]

\[ - \int_t^{t+\Delta t} Q(x(\tau)) \, d\tau - \int_t^{t+\Delta t} [\psi^T(x(\tau)) \theta^{(i)}_u]^T R [\psi^T(x(\tau)) \theta^{(i)}_u]. \] (14)

In order to simplify the residual error of (14), we define the following

\[ \rho_{\Delta \phi}(x(t), x(t + \Delta t)) = \phi(x(t)) - \phi(x(t + \Delta t)) \]

\[ \rho_Q(x(t)) = \int_t^{t+\Delta t} Q(x(\tau)) \, d\tau \]

\[ \rho_\psi(x(t)) = \int_t^{t+\Delta t} \psi(x(t)) \psi^T(x(t)) \, d\tau \]

\[ \rho_{u\psi}(x(t), u(t)) = \int_t^{t+\Delta t} u(\tau) \psi^T(x(\tau)) \, d\tau \]

Then expression (14) is rewritten as

\[ \sigma^{(i)}(x(t), u(t), x(t + \Delta t)) \]

\[ = \rho_{\Delta \phi}(x(t), x(t + \Delta t)) \theta^{(i+1)}_v + 2 R (\theta^{(i)}_u)^T \rho_\psi(x(t)) \theta^{(i+1)}_u - 2 R \rho_{u\psi}(x(t), u(t)) \theta^{(i+1)}_u \]

\[ - \rho_Q(x(t)) - R (\theta^{(i)}_u)^T \rho_\psi(x(t)) \theta^{(i)}_u \]

\[ = \bar{\sigma}^{(i)}(x(t), u(t), x(t + \Delta t)) + \pi^{(i)}(x(t)) \] (15)

where \( \theta^{(i+1)} = [\theta^{(i+1)}_v \theta^{(i+1)}_u]^T, \pi^{(i)}(x(t)) = \rho_Q(x(t)) + R (\theta^{(i)}_u)^T \rho_\psi(x(t)) \theta^{(i)}_u, \)

\[ \bar{\sigma}^{(i)}(x(t), u(t), x(t + \Delta t)) = \rho_{\Delta \phi}(x(t), x(t + \Delta t)) 2 R (\theta^{(i)}_u)^T \rho_\psi(x(t)) - 2 R \rho_{u\psi}(x(t), u(t)). \]

We use the method of weighted residuals [18] to update the weights. The unknown weight vector can be determined by making projection of the residual error \( \bar{\sigma}^{(i)} \) onto the term \( \frac{\partial \sigma^{(i)}}{\partial \theta^{(i+1)}} \) be zero,
\[ \langle \frac{\partial \sigma^{(i)}}{\partial \theta^{(i+1)}}, \sigma^{(i)} \rangle_D = 0. \]  
\[ (16) \]

Substituting (15) into (16) we have
\[ \langle \hat{\rho}^{(i)}, \hat{\rho}^{(i)} \rangle_{\theta^{(i+1)}} - \langle \hat{\rho}^{(i)}, \pi^{(i)} \rangle_D = 0. \]
\[ (17) \]

We select \( M \) sample points on \( D \), then define
\[ X = \left( \left( \hat{\rho}^{(i)}(x_1, u_1) \right)^T \cdots \left( \hat{\rho}^{(i)}(x_M, u_M) \right)^T \right). \]
\[ Y = \left[ \pi^{(i)}(x_1) \cdots \pi^{(i)}(x_M) \right]^T. \]

For calculation, make sure to select a sufficiently large number of sample points that the matrix \( X \) is full column rank, and then approximately compute the inner product, (17) follows
\[ (X^T X) \theta^{(i+1)} - X^T Y = 0, \]
\[ (18) \]
then yields
\[ \theta^{(i+1)} = (X^T X)^{-1} X^T Y. \]
\[ (19) \]

As we can see, the critic and actor weight vectors \( \theta^{(i+1)} \) can be obtained at the same time. When the actor NN weight sequence finally converges to constant we can get the approximate optimal controller.

### 4. Simulation

In this section, we will verify the accuracy and effectiveness of the proposed data-based control method by numerical simulation. In the data-driven approach, the activation function of the critic NN is selected as \( \phi(x) = [x_1^2 \ x_2^2 \ x_3^2 \ x_4^2 \ x_1 x_2 \ x_1 x_3 \ x_1 x_4 \ x_2 x_3 \ x_2 x_4 \ x_3 x_4]^T \) with \( L_u = 10 \), and the actor NN activation function vector is selected as \( \psi(x) = [x_1 \ x_2 \ x_3 \ x_4 \ x_1 x_2 \ x_1 x_3 \ x_1 x_4 \ x_2 x_3 \ x_2 x_4 \ x_3 x_4]^T \) with the size of \( L_u = 10 \). In this simulation we choose the initial state \( x_0 = [0.01 \ 0.1 \ 0.0]^T \) and the sampling period \( \Delta t = 0.01s \). Meanwhile, we set the initial critic and actor NN weight vectors
\[ \theta_v^{(0)} = [0; 0; 0; 0; 0; 0; 0; 0; 0; 0; 0] \]
\[ \theta_u^{(0)} = [-0.05; -0.02; -0.03; -0.01; -0.05; -0.05; -0.05; -0.05; 0; 0] \]
respectively. We first collect the input and output data of the system, and then the optimal control strategy can be obtained by offline policy iteration method. The weight vectors of the actor NNs and critic NNs eventually converges to \( \theta_v = [9.25; 88.15; 83.01; -4.54; 0.56; -4.49; 0.13; 0.05; -0.23; 0.97] \) and \( \theta_u = [-14.45; 2.41; -0.10; 0.00; 3.71; -1.55; 0.33; 0.21; -0.01; 0.03] \) respectively, under the condition of convergence criterion is \( \epsilon = 10^{-5} \). The convergence procedures are shown in Figure 2 and Figure 3.
Figure 2. Four representative actor NN weights $\theta_{u,2}, \theta_{u,7}, \theta_{u,8}, \theta_{u,9}$

Figure 3. Four representative actor NN weights $\theta_{v,2}, \theta_{v,7}, \theta_{v,8}, \theta_{v,9}$

It can be seen from the simulation in Figure 4, after applying the approximate optimal controller the trajectories of pitch angle and plunge motions can be stabilized within a few seconds.
5. Conclusions
The airfoil flutter control is investigated in this paper, focusing on how to get the controller independent on a precise physical model. A data-based adaptive dynamic programming method is used to suppress the limit cycle flutter of 2DOF airfoil with considering the structural nonlinearity by designing an optimal controller. This method can learn from the state data and iteratively improve the controller. First, we develop a model of the 2-DOF airfoil with pitch nonlinearity to collect data. Through the given explore noise we get a series of state data. Based on the offline data-driven method we update the critic NN and actor NN iteratively. Finally, after several iterations the critic and actor NN weights convergence to constant. Through the simulation, we see that the controller can quickly and effectively suppress the flutter of the airfoil.

Acknowledgments
This work is supported by the National Natural Science Foundation of China (51575378 and 11572215).

References
[1] Asjes D, Diwadkar A, Vaidya U, et al. 2014 Proc. American Control Conference. pp 1162-67.
[2] Dowell E H, Thomas J P, Hall K C. 2004 J. Fluids and Structures. 19 17-27.
[3] Yang N, Wang N, Zhang X, Liu W. 2016 J. Chinese Journal of Aeronautics 29 144-159.
[4] He S, Yang Z, Gu Y. 2017 J. Nonlinear Dynamics. 4 2099-2125.
[5] Guo H, Chen Y, Yang T. 2013 J. Applied Mathematics and Mechanics 34 1277-90.
[6] Verstraalen E, Gaëtan K, and Grigoris D. 2017 Conf. Proceeding of the Society for Experimental Mechanics Series vol 9 pp 301-313.
[7] Malher A, Touzé C, Doaré O, et al. 2017 J. Computational and Nonlinear Dynamics. 12 051016.
[8] Luo M, Gao M, Cai G. 2016 J. Fluids and Structures. 61 262-273.
[9] Xue W T, Li J Z. 2016 Conf. Industrial Electronics Society. IEEE pp 6764-69.
[10] Sriganac T W, Ko J, Thompson D E. 2000 J. Guidance Control and Dynamics. 23 1127-33.
[11] Xiang J, Yan Y, Li D. 2014 J. Chinese Journal of Aeronautics. 27 12-22.
[12] Sutton R S, Barto A G, Bach F. 1998 Reinforcement learning: An introduction (MIT press).
[13] Werbos P J. 1974 J. Ph.D Dissertation Harvard University.
[14] Lewis F L, Vrabie D, Syrmos V L. 2012 Optimal control (John Wiley & Sons).
[15] Vrabie D, Lewis F. 2009 J. Neural Networks. 22 237-246.
[16] Jiang Y, Fan J, Chai T, et al. 2018 J. IEEE Transactions on Industrial Informatics. 14 1974-89.
[17] Yang X, He H, Zhong X. 2018 J. IEEE Transactions on Industrial Electronic. 65 5722-32.
[18] Luo B, Wu H N, Huang T, et al. 2014 J. Automatica. 50 3281-90.