


RESEARCH ARTICLE

Research on model-free adaptive active flutter suppression based on ridge regression

J. Yu^{1,2}, H. Qi^{1,2} , J. Du^{1,2}, K. Wang^{1,2} and J. Guo^{1,2}

¹College of Aerospace and Civil Engineering, Harbin Engineering University, Harbin, 150001, China and ²Key Laboratory of Vehicle Transmedia Technology, Harbin Engineering University, Harbin, 150001, China

Corresponding author: Hui Qi; Email: qihui@hrbeu.edu.cn

Received: 7 April 2023; **Revised:** 5 August 2023; **Accepted:** 22 August 2023

Keywords: ridge regression; active flutter suppression; data drive

Abstract

Traditional active flutter suppression controllers are designed based on model. However, as the aircraft becomes more and more powerful, the modeling of aeroelastic system becomes difficult and the model-free requirement of controller design becomes more and more urgent. The complexity of industrial processes has brought about massive operational data generated online. Aviation industry development has entered the era of big data. Breaking through the traditional theoretical framework, mining the correlation, evolution and dynamic characteristics of the system from the data is the inevitable choice to meet this demand. In this paper, a data-driven model-free controller is designed, which relies on ridge regression of the input and output variation at each operating point of the closed-loop controlled system to recursively derive the iterative format of the control signals and ensure the numerical stability of the signals. The controller can only use the real-time measurement of the system's online input and output data for continuous correction, to achieve the purpose of flutter suppression. Then flutter suppression of a three-degree-of-freedom binary wing with a control surface is studied, and the superiority of model-free controller is demonstrated by comparing it with the optimal controller.

Nomenclature

AFS	active flutter suppression
AFW	active flexible wing
AAW	active aeroelastic wing
ADRC	active disturbance rejection control
BACT	benchmark active control technology
MIMO	multiple input multiple output
m	total mass of the wing
I_α	rotational inertia of the wing to the elastic axis
$s_{h\alpha}$	mass static moment of the wing to the elastic axis
$s_{h\beta}$	mass static moment of the control surface to its rotation axis
I_β	moment of inertia of the control surface to its rotation axis
h	sinking and floating displacement
α	pitching angle
β	control surface deflection angle
d_h	sinking and floating damping coefficient
d_α	pitching damping coefficient
k_h	sinking and floating stiffness coefficient
k_α	pitching stiffness coefficient
s_p	wing span
L	aerodynamic lift

T_α	aerodynamic moment to the elastic axis
δ	control surface deflection instruction
k	proportional coefficient
ω	natural frequency
ζ	damping ratio
Φ	pseudo partial derivative
η, ε	step length
λ, μ	regularisation coefficient
T_s	sampling period
R	weighting coefficient
Q	weighting matrix

1.0 Introduction

Active flutter suppression (AFS) is one of the main active control technologies applied in aircraft design at present. It belongs to the comprehensive category of aeroservoelasticity and is one of the research hotspots in the field of aeroservoelasticity. It focuses on increasing the critical flutter velocity without significantly increasing the structural weight of the aircraft [1, 2]. Active flexible wing (AFW) was proposed in the United States in the 1980s, and since then, active flutter suppression has been continuously invested in research and development. In this century, this concept has been developed into active aeroelastic wing (AAW) technology, which has promoted the aeroservoelasticity comprehensive technology to a higher theoretical and technical level. It is a new concept of modern aircraft design.

In recent years, important progress has been made in the research of AFS technology. Zeng et al. [3] developed a feedback control framework based on a wind tunnel experimental model to suppress flutter using robust control laws. Schmidt [4] aimed at Lockheed Martin's BFF06 whole aircraft model, using the method named "identically located force and acceleration" to construct a double loop system, and increase the model damping, and achieved good results in flutter suppression. Huang et al. [5] conducted a wind tunnel experimental study on closed-loop flutter modal identification and AFS of a 3D small aspect ratio wing model. The linear feedback controller designed through pole assignment optimisation effectively suppressed flutter, and explored the on-line closed-loop flutter identification of the wing model. Theis et al. [6] proposed a systematic robust control design method for active flutter suppression. It extends the standard four-block mixed sensitivity formulation by a means that targets specific dynamic modes and adds damping. This enables a control design to augment the damping of critical flutter modes with minimal impact on the rigid-body autopilots. Vepa et al. [7] used the linearised aeroelastic model based on the doublet lattice method to initially design the control law family for the active flutter suppression system. Using these preliminary control laws and the nonlinear transonic small disturbance theory, the approximate optimal control law was selected in the transonic domain, and the flutter of a typical wing was successfully suppressed. Yang et al. applied active disturbance rejection control (ADRC) theory to the design of aeroelastic control law and studied the transonic AFS problem of benchmark active control technology (BACT) wing [8] and 3D elastic wing [9] considering parameter uncertainty and measurement noise interference. Although AFS technology has been widely used in aircraft design, it still faces the following difficulties.

First, the existing flutter suppression controller structure is complex: The complex structure of the mathematical model determines the complex structure of the controller, and the complex high-order nonlinear system model inevitably leads to the complex high-order nonlinear controller, and the problems of simplification and reduction of the controller and robustness become insurmountable design problems.

Second, there is no systematic design method for multiple input multiple output (MIMO) systems: For flutter suppression, it is impossible to find a general mode to design the controller in the existing control theories and methods. For different aeroservoelastic systems, it can only be analysed on a case-by-case basis, which undoubtedly requires designers to have strong expert experience and knowledge, and cannot be quickly designed in a process, increasing the design cycle.

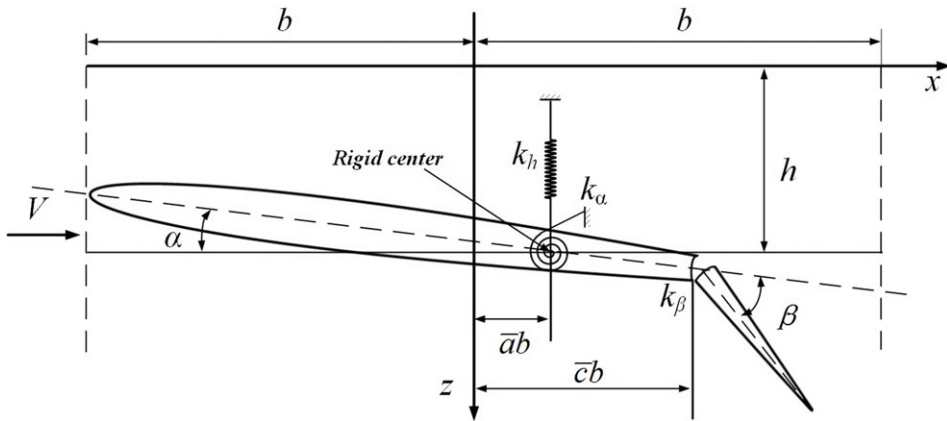


Figure 1. Three-degree-of-freedom binary wing dynamic model.

Third, it is still necessary to rely on the model design control law: Many problems in aviation engineering, such as high-fidelity unsteady aerodynamic model, nonlinear time-variant aircraft model and various uncertain disturbances, need a lot of expert experience to establish the system mechanism model and global mathematical model to solve, and the cost is huge. In addition, for complex systems, due to the complexity of the system itself and various interferences, it is impossible to establish a global mathematical model of the system, even if the local model is not very accurate. Therefore, although the model-based control theory has rich results, it is weak in solving practical complex aeronautical engineering problems.

To solve the above problems, a model-free adaptive control method based on ridge regression is proposed in this paper. The active flutter suppression controller designed by using this method can achieve the purpose of flutter suppression only by using real-time measurement of the online input and output data of the closed-loop controlled system for continuous correction.

2.0 Modeling of aeroelastic system

Figure 1 shows a three-degree-of-freedom binary wing dynamic model with control surface. The equation of motion of the binary wing can be written as

$$\begin{bmatrix} m & s_{h\alpha} \\ s_{h\alpha} & I_\alpha \end{bmatrix} \begin{Bmatrix} \ddot{h} \\ \ddot{\alpha} \end{Bmatrix} + \begin{bmatrix} d_h & 0 \\ 0 & d_\alpha \end{bmatrix} \begin{Bmatrix} \dot{h} \\ \dot{\alpha} \end{Bmatrix} + \begin{bmatrix} k_h & 0 \\ 0 & k_\alpha \end{bmatrix} \begin{Bmatrix} h \\ \alpha \end{Bmatrix} = - \begin{Bmatrix} s_{h\beta} \\ s_{\alpha\beta} \end{Bmatrix} \ddot{\beta} + \begin{Bmatrix} -L \\ T_\alpha \end{Bmatrix} \quad (1)$$

where m is the total mass of the wing; I_α is the rotational inertia to the elastic axis; $s_{h\alpha}$ is the mass static moment of the wing to the elastic axis; $s_{h\beta}$ is the mass static moment of the control surface to its rotation axis; $s_{\alpha\beta} = (\bar{c} - \bar{a})bs_{h\beta} + I_\beta$, among them, I_β is the moment of inertia of the control surface to its rotation axis; h is sinking and floating displacement; α is pitching displacement; β is the control surface deflection angle; d_h is sinking and floating damping coefficient; d_α is pitching damping coefficient; k_h is sinking and floating stiffness coefficient; k_α is pitching stiffness coefficient; the aerodynamic forces acting on the wing with span of s_p are calculated using Theodorsen theory, among them, L is aerodynamic lift, T_α is the aerodynamic moment to the elastic axis, the specific algorithm is shown in Appendix.

The actuating system reflects the relationship between the control surface deflection instruction and the actual deflection angle of the control surface, which is described by the following second-order system

$$\ddot{\beta} + 2\zeta\omega\dot{\beta} + \omega^2\beta = k\omega^2\delta \quad (2)$$

where δ represents control surface deflection instruction; k is the proportional coefficient; ω represents the natural frequency of the system; ζ represents the damping ratio. According to Equations (1) and (2), the motion equation of the wing including the actuating system can be written as

$$M_s \ddot{q}_s + D_s \dot{q}_s + K_s q_s = F_{ae} + G_s \delta \tag{3}$$

where $M_s = \begin{bmatrix} m & s_{h\alpha} & s_{h\beta} \\ s_{h\alpha} & I_\alpha & s_{\alpha\beta} \\ 0 & 0 & 1 \end{bmatrix}$, $D_s = \begin{bmatrix} d_h & 0 & 0 \\ 0 & d_\alpha & 0 \\ 0 & 0 & d_\beta \end{bmatrix}$, $K_s = \begin{bmatrix} k_h & 0 & 0 \\ 0 & k_\alpha & 0 \\ 0 & 0 & k_\beta \end{bmatrix}$, $F_{ae} = \begin{bmatrix} -L \\ T_\alpha \\ 0 \end{bmatrix}$,
 $G_s = \begin{bmatrix} 0 \\ 0 \\ g_0 \end{bmatrix}$, $q_s = [h \ \alpha \ \beta]^T$, $d_\beta = 2\zeta\omega$, $k_\beta = \omega^2$, $g_0 = k\omega^2$.

In order to design the optimal controller for flutter suppression, it is necessary to express the open-loop aeroelastic model in time-domain state space. At this time, in addition to the state variables related to the structure, there are also the constructed aerodynamic state variables. The state space realisation of the aeroelastic equation can be written as

$$\dot{X} = AX + B\delta \tag{4}$$

where $A = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 2} \\ -M^{-1}K & -M^{-1}D & -V^2 M^{-1}E_c \\ VK_a & D_a & VA_a \end{bmatrix}$, $B = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ M^{-1}G_s \\ \mathbf{0}_{2 \times 1} \end{bmatrix}$, $X = [h \ \alpha \ \beta \ \dot{h} \ \dot{\alpha} \ \dot{\beta} \ x_{a1} \ x_{a2}]^T$,

please refer to Appendix for the meaning of specific parameters and the calculation process. For the model-based optimal controller design, measurable output of the controlled object should be used to provide feedback information, while the aerodynamic state variable is unmeasurable, so the outputs selected are sinking and floating displacement of the wing, pitching angle of the wing and deflection angle of the control surface. The state equation and output equation of the controlled system can be written as

$$\begin{cases} \dot{X} = AX + B\delta \\ Y = CX \end{cases} \tag{5}$$

where $C = [I_{3 \times 3} \ \mathbf{0}_{3 \times 5}]$.

3.0 Design of adaptive filter for structural characteristics of discrete system

For MIMO discrete time system, it can be written in the form of Equation (6)

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{g}(\mathbf{x}(k), \mathbf{u}(k)) \\ \mathbf{y}(k+1) = \mathbf{h}(\mathbf{x}(k), \mathbf{u}(k)) \end{cases} \tag{6}$$

where \mathbf{g} , \mathbf{h} represent the unknown state equation and output equation, respectively, $\mathbf{y}(k+1) = [y_1(k+1) \ \dots \ y_q(k+1)]^T$ is the system output, $\mathbf{u}(k) = [u_1(k) \ \dots \ u_p(k)]^T$ is the system input, p and q are the input and output dimensions of the system, respectively. We do not know the internal structure of the system, and can only obtain the online input data \mathbf{u} and output data \mathbf{y} of the system through sensor measurement. Equation (6) indicates that the output of the system at the next moment is not only related to the input of the current moment, but also related to the state of the current moment. However, we cannot measure the system state variable. Our idea is to carry out linear regression at each working point of the system according to the existing input and output data and then estimate some structural characteristics of the system. Therefore, we set up a linear regression model of input and output variation at each working point, as shown in Equation (7). Some works have shown that Equations (6) and (7) are equivalent under very broad conditions [10, 11, 12].

$$\Delta y_n(k+1) = \Phi_n^T(k) \Delta \mathbf{u}(k) \tag{7}$$

where $\Delta y_n(k+1) = y_n(k+1) - y_n(k)$, $\Delta \mathbf{u}(k) = \mathbf{u}(k) - \mathbf{u}(k-1)$, $1 \leq n \leq q$, $\Phi_n(k) = [\phi_{n1} \ \dots \ \phi_{np}]^T$ represents the structural characteristics of the system corresponding to the

nth output $y_n(k + 1)$. Since it is obtained by linear regression using online input and output data at each operating point, we can call it pseudo partial derivative of the system. Observing Equation (7), we can see that the pseudo partial derivative $\Phi_n(k)$ depends on the system input variation $\Delta \mathbf{u}(k)$ at the current time k and the nth system output variation $\Delta y_n(k + 1)$ at the next time $k + 1$, both of which can be measured online, and $\mathbf{u}(k)$ needs to be designed by us. Next we need to construct a recursive format for the pseudo partial derivative of the system.

The estimation error is defined as $e_n(k) = \Delta y_n(k + 1) - \Phi_n^T(k) \Delta \mathbf{u}(k)$, and the estimation loss is measured by Equation (8).

$$f(\Phi_n(k)) = \|e_n(k)\|^2 + \mu \|\Phi_n(k)\|^2 \tag{8}$$

We need to use Newton method to minimise Equation (8), that is

$$\min_{\Phi} \|e_n(k)\|^2 + \mu \|\Phi_n(k)\|^2 \tag{9}$$

And then, we can obtain Equation (10).

$$f(\hat{\Phi}_n(k + 1)) = f(\hat{\Phi}_n(k) + \Delta) = f(\hat{\Phi}_n(k)) + J_{f(\Phi)}^T(\hat{\Phi}_n(k)) \times \Delta + \frac{1}{2} \Delta^T H_{f(\Phi)}(\hat{\Phi}_n(k)) \Delta \tag{10}$$

where $\Delta = \varepsilon H_{f(\Phi)}^{-1}(\hat{\Phi}_n(k)) J_{f(\Phi)}(\hat{\Phi}_n(k))$, $J_{f(\Phi)}$ is the Jacobi matrix about Φ , $H_{f(\Phi)}$ is the Hessian matrix about Φ . We can further obtain Equation (11).

$$\hat{\Phi}_n(k + 1) = \hat{\Phi}_n(k) + \varepsilon H_{f(\Phi)}^{-1}(\hat{\Phi}_n(k)) J_{f(\Phi)}(\hat{\Phi}_n(k)) \tag{11}$$

where $J_{f(\Phi)}(\hat{\Phi}_n(k)) = -2\Delta \mathbf{u}(k) e_n(k)$, $H_{f(\Phi)}(\hat{\Phi}_n(k)) = 2\Delta \mathbf{u}(k) \Delta \mathbf{u}^T(k) + 2\mu \mathbf{I}$, ε is the search step length of Newton method.

Observing Equation (8), the form is not the traditional mean square error function as shown in Equation (12). Instead, the term $\mu \|\Phi_n(k)\|^2$ is added to Equation (12).

$$f(\Phi_n(k)) = \|e_n(k)\|^2 \tag{12}$$

This is because if Equation (12) is used as the loss function, the Hessian matrix in Equation (10) will change into $H_{f(\Phi)}(\hat{\Phi}_n(k)) = 2\Delta \mathbf{u}(k) \Delta \mathbf{u}^T(k)$. At this point, every row and column of the Hessian matrix are linearly correlated, and its rank is 1. The inverse operation of the matrix will result in singularity. The loss function in the form of Equation (8) is actually a ridge regression technique, by adding regularisation term $\mu \|\Phi_n(k)\|^2$ to the objective function, so that the inverse of Hessian matrix can be established.

By comparing the Hessian matrix forms obtained by the two loss functions, it can be seen that through ridge regression, the diagonal of the Hessian matrix obtained by Equation (12) is numerically loaded with a disturbance, which not only makes the Hessian matrix become full rank and be inversely solved, but also makes the numerical stability of the inverse operation good due to the existence of penalty of regularisation term. By substituting Hessian matrix obtained by Equation (8) into Equation (11), an adaptive filter can be obtained, as shown in Equation (13).

$$\hat{\Phi}_n(k + 1) = \hat{\Phi}_n(k) + \eta (\Delta \mathbf{u}(k) \Delta \mathbf{u}^T(k) + \mu \mathbf{I})^{-1} e_n(k) \Delta \mathbf{u}^T(k) \tag{13}$$

where $\eta = -4\varepsilon$ is the filter iteration step length. And then, we need to use the formula

$$(\mathbf{A} + \mathbf{BCD})^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1} \mathbf{B} (\mathbf{D} \mathbf{A}^{-1} \mathbf{B} + \mathbf{C}^{-1})^{-1} \mathbf{D} \mathbf{A}^{-1} \tag{14}$$

to deal with the inverse operation. Making $\mathbf{A} = \mu \mathbf{I}$, $\mathbf{B} = \Delta \mathbf{u}(k)$, $\mathbf{C} = 1$, $\mathbf{D} = \Delta \mathbf{u}^T(k)$, we can export more concise filtering device, as shown Equation (15).

$$\hat{\Phi}_n(k + 1) = \hat{\Phi}_n(k) + \eta \frac{\Delta \mathbf{u}^T(k)}{\mu + \|\Delta \mathbf{u}(k)\|^2} e_n(k) \tag{15}$$

4.0 Design of data driven control law

Continuing the ridge regression thought in the previous section, the recursion of the control law is also the solution of one optimisation problem. We design the loss function of the optimisation problem to be the function of controlling signal variation, as shown in Equation (16). In order to achieve the purpose of tracking target signals, the loss function must be minimised.

$$l(\Delta \mathbf{u}(k)) = \|y_n^*(k+1) - y_n(k+1)\|^2 + \lambda \|\Delta \mathbf{u}(k)\|^2 \tag{16}$$

where y_n^* is the tracking value of the n th output, $\mathbf{y}^*(k+1) = [y_1^*(k+1) \ \dots \ y_q^*(k+1)]^T$. Defining control error corresponding to the n th output as $e_n^*(k+1) = y_n^*(k+1) - y_n(k+1)$, we can obtain Equation (17) by substituting Equation (7) into it.

$$e_n^*(k+1) = y_n^*(k+1) - y_n(k) - \hat{\boldsymbol{\Phi}}_n^T(k) \Delta \mathbf{u}(k) \tag{17}$$

The loss function can be further written as

$$l(\Delta \mathbf{u}(k)) = \|y_n^*(k+1) - y_n(k)\|^2 - 2(y_n^*(k+1) - y_n(k)) \hat{\boldsymbol{\Phi}}_n^T(k) \Delta \mathbf{u}(k) + \Delta \mathbf{u}^T(k) \hat{\boldsymbol{\Phi}}_n(k) \hat{\boldsymbol{\Phi}}_n^T(k) \Delta \mathbf{u}(k) + \lambda \|\Delta \mathbf{u}(k)\|^2 \tag{18}$$

where $\lambda \|\Delta \mathbf{u}(k)\|^2$ is still a regularisation term, to avoid the happening of singular and numerical instability. The gradient of the loss function to the control signal is the Jacobi matrix

$$J_{l(\Delta \mathbf{u})}(\Delta \mathbf{u}(k)) = -2(y_n^*(k+1) - y_n(k)) \hat{\boldsymbol{\Phi}}_n^T(k) + 2\hat{\boldsymbol{\Phi}}_n(k) \hat{\boldsymbol{\Phi}}_n^T(k) + 2\lambda \Delta \mathbf{u}(k) \tag{19}$$

The optimal control signal gradient should be 0, and then we can obtain Equation (20).

$$\Delta \mathbf{u}(k) = (y_n^*(k+1) - y_n(k)) (\hat{\boldsymbol{\Phi}}_n(k) \hat{\boldsymbol{\Phi}}_n^T(k) + \lambda \mathbf{I})^{-1} \hat{\boldsymbol{\Phi}}_n(k) \tag{20}$$

Otherwise, $\mathbf{u}(k) = \mathbf{u}(k-1) + \Delta \mathbf{u}(k)$. Using Formula (14) to deal with the inversion operation and making $\mathbf{A} = \lambda \mathbf{I}$, $\mathbf{B} = \hat{\boldsymbol{\Phi}}_n(k)$, $\mathbf{C} = 1$, $\mathbf{D} = \hat{\boldsymbol{\Phi}}_n^T(k)$, we can export more simple control signal recursive format as Equation (21).

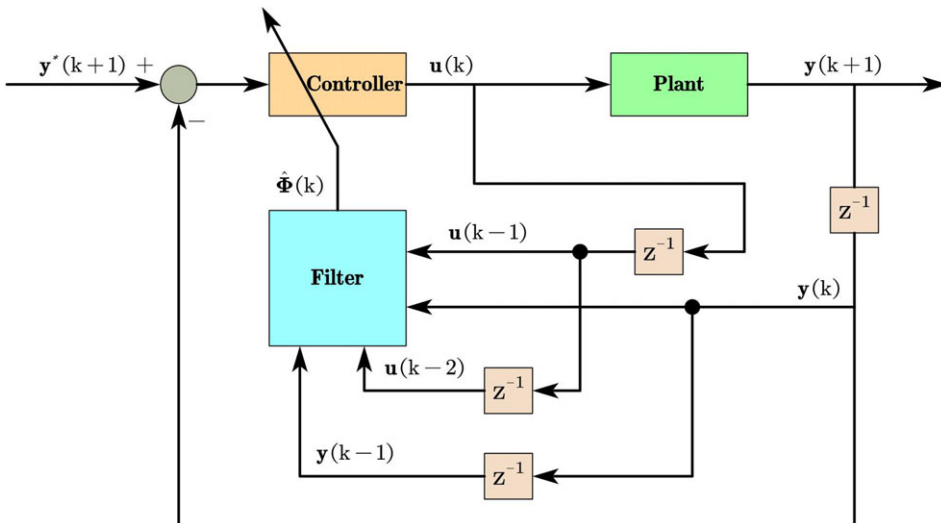


Figure 2. Data-driven control system.

Table 1. Binary wing model and actuator parameters

Parameters	Value
m	1.85 kg
\bar{a}	-0.5
b	0.1 m
\bar{c}	0.5
s_p	0.3 m
d_h	27.43 Ns/m
d_α	0.02 Ns
k_h	2542 N/m
k_α	2.512 Nm/rad
I_α	3.142×10^{-3} kgm ²
$s_{h\alpha}$	0.0309 kgm
$s_{h\beta}$	8.608×10^{-4} kgm
$s_{\alpha\beta}$	1.215×10^{-4} kgm
ρ_a	1.225 kg/m ³
ω	357.07 rad/s
ζ	0.598
k	0.9715

Table 2. Optimal controller parameters

Project	Weighting coefficient	Weighting matrix
Parameters	$R = 100$	$Q = \text{diag}(100, 100, 100, 50, 50, 50, 50, 50)$

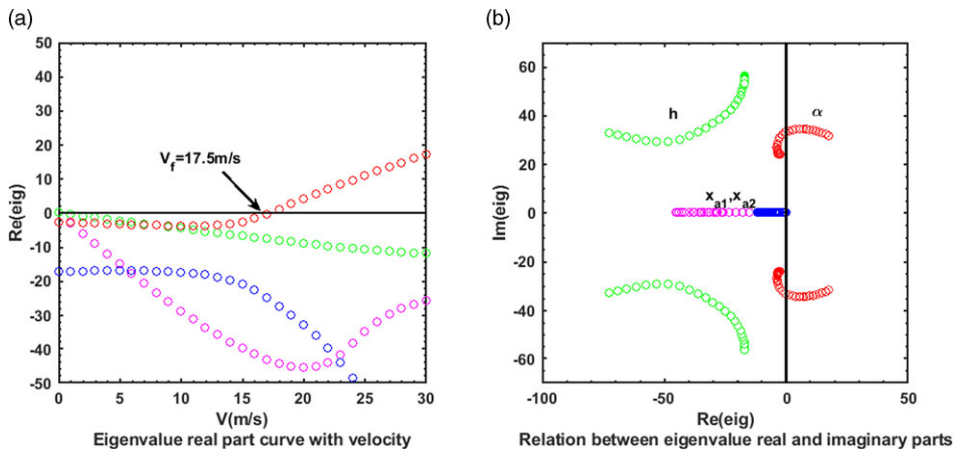


Figure 3. Root locus analysis of the uncontrolled system.

$$u(k) = u(k - 1) + \frac{\hat{\Phi}_n(k)}{\lambda + \hat{\Phi}_n^T(k) \hat{\Phi}_n(k)} (y_n^*(k + 1) - y_n(k)) \tag{21}$$

Observing Equation (21), we can see that the control signal $u(k)$ at the current time k depends on the system pseudo partial derivative estimation $\hat{\Phi}_n(k)$ and system output $y_n(k)$ at the current time k , the target tracking value $y_n^*(k + 1)$ at the next time $k + 1$, and the control signal $u(k - 1)$ at the previous

Table 3. Data-driven controller parameters

Project	Step length	Regularisation coefficient		Filter initial value	Tracking signal
Parameters	$\eta = 1$	$\mu = 1$	$\lambda = 0.5$	$\hat{\Phi}(k) = [1 \ 2 \ 2.5]^T$	$y^* = [0 \ 0 \ 0]^T$

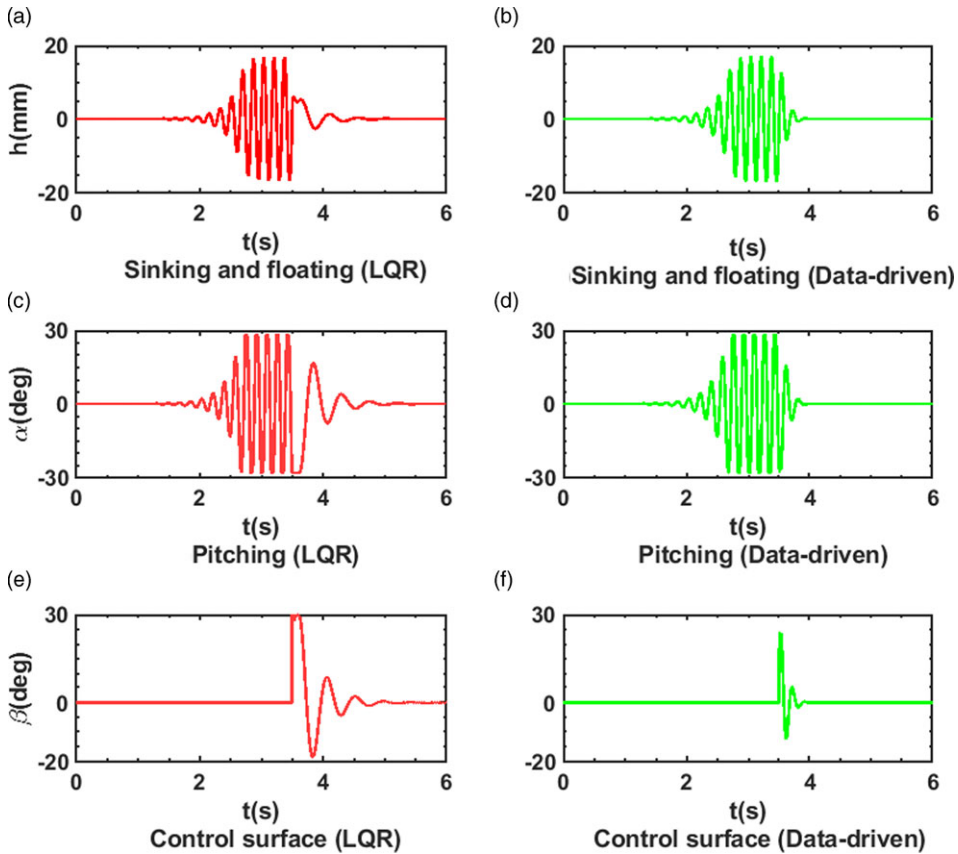


Figure 4. System response ($V = 20$ m/s).

time $k - 1$. Among them, $y_n(k)$ can be measured online, $y_n^*(k + 1)$ is known, and $\hat{\Phi}_n(k)$ can be estimated by the filter, which is a known quantity. It is necessary to point out that Equations (13) and (20) are the more essential forms, while Equations (15) and (21) are just more suitable for programming computation.

For convenience, the filter iteration step length corresponding to every output is equal, and the same to regularisation coefficient. Theoretically, q calculation results of recursive scheme should be exactly the same, but there will always be some disturbance in the measurement process. Therefore, the control signal can be averaged once, as shown in Equation (22).

$$u(k) = u(k - 1) + \frac{1}{q} \sum_{n=1}^q \frac{\hat{\Phi}_n(k)}{\lambda + \hat{\Phi}_n^T(k) \hat{\Phi}_n(k)} (y_n^*(k + 1) - y_n(k)) \tag{22}$$

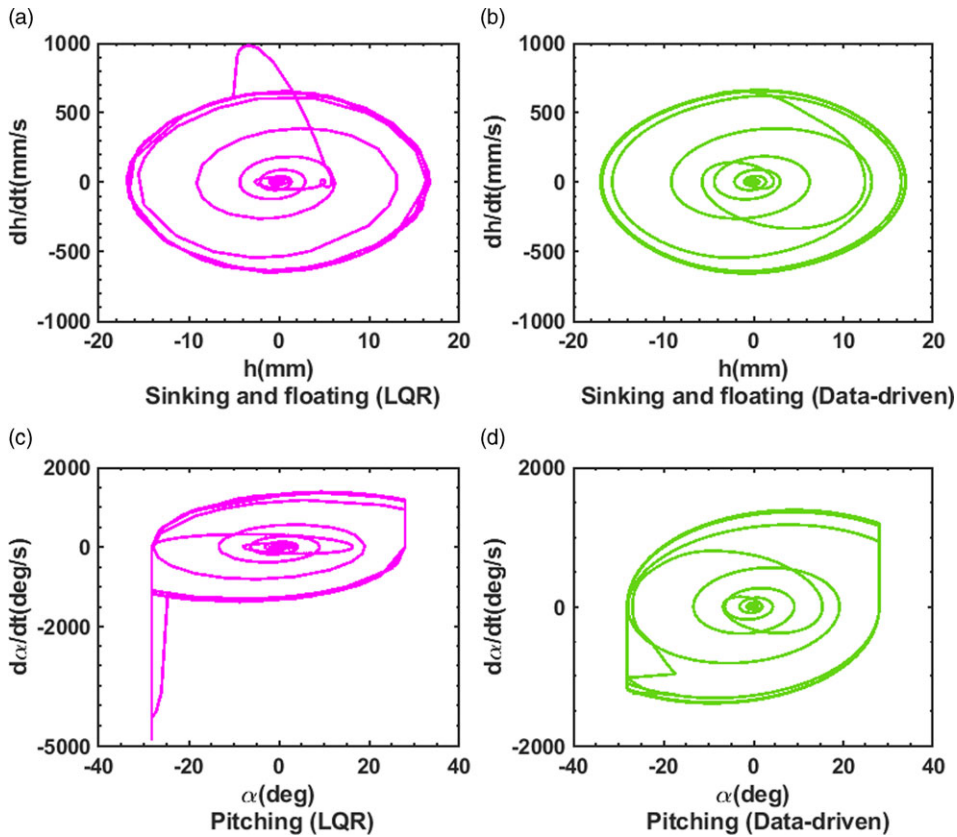


Figure 5. System state phase locus ($V = 20$ m/s).

The model-free control scheme of MIMO discrete time system can be obtained by combining filter algorithm and control algorithm, as shown in Equation (23).

$$\begin{cases} \hat{\boldsymbol{\phi}}_n(k) = \hat{\boldsymbol{\phi}}_n(k-1) + \eta \frac{e_n(k-1)\Delta\mathbf{u}^T(k-1)}{\mu + \|\Delta\mathbf{u}(k-1)\|^2} \\ \mathbf{u}(k) = \mathbf{u}(k-1) + \frac{1}{q} \sum_{n=1}^q \frac{\hat{\boldsymbol{\phi}}_{n(k)}}{\lambda + \hat{\boldsymbol{\phi}}_{n(k)}^T \hat{\boldsymbol{\phi}}_{n(k)}} (y_n^*(k+1) - y_n(k)) \end{cases} \quad (23)$$

where $\Delta\mathbf{u}(-1) = \mathbf{0}$. Figure 2 shows the operation principle of the whole control system, in which $\hat{\boldsymbol{\phi}}(k) = [\hat{\boldsymbol{\phi}}_1(k) \quad \dots \quad \hat{\boldsymbol{\phi}}_q(k)]^T$.

5.0 Numerical simulation and discussion

For an aeroelastic system, when the flow velocity exceeds the critical flutter velocity, the system will have divergent motion, which leads to the failure of the structure. Therefore, in order to prevent divergent motion, stiffness of flutter system is often designed as hard characteristic in wind tunnel test. In this simulation example the limit spring on the pitching direction is used to limit the size of the pitch angle ($-28^\circ \leq \alpha \leq 28^\circ$), and the control surface is saturated with deflection ($-30^\circ \leq \beta \leq 30^\circ$). The micro-vibration of the wing near the equilibrium position is linear vibration, and large vibration is generated after the flutter. When the pitch angle exceeds the given threshold, the limiting spring will limit the pitching angle to increase further. Therefore, the wing exhibits limit cycle motion with finite amplitude when flutter occurs. The numerical simulation parameters are shown in Table 1.

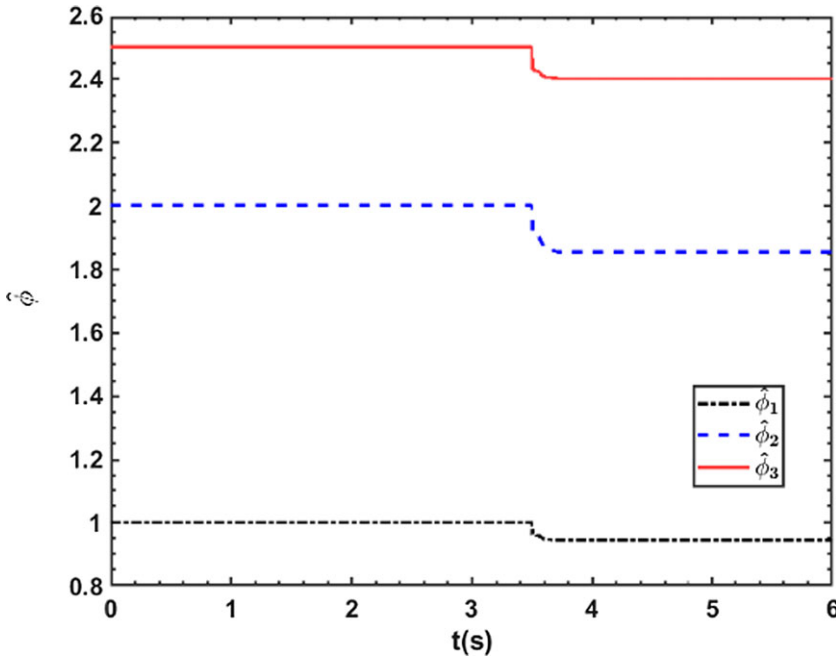


Figure 6. Evaluation curve of system pseudo partial derivative ($V = 20$ m/s).

The critical flutter velocity of the wing model can be determined by investigating the change of the eigenvalue of system matrix A of the open-loop system in Equation (4) with the flow velocity, as shown in Fig. 3. Flutter critical velocity is $V_f = 17.5$ m/s. The motion of the aeroelastic system under this wind speed is constant amplitude oscillation. At the critical flutter velocity, the motion of the system does not diverge due to the existence of the limiting spring, but behaves as a limit cycle.

The optimal controller is designed using LQR method in the research work of Darabseh et al. [13]. The optimal control law is to seek a constant gain output feedback control law

$$\delta = -K_{con}Y = -K_{con}CX \tag{24}$$

to minimise the following performance index.

$$J = \frac{1}{2} \int_0^\infty (X^T Q X + R \delta^2) dt \tag{25}$$

where Q is the positive definite weighting matrix of the state variable, and $R > 0$ is the weighting coefficient of the control variable. This objective function takes into account the requirements of both system response and control. According to the linear quadratic optimal control theory, if the performance index J is to be minimised, the control signal should be

$$\delta = -R^{-1}BPX \tag{26}$$

where $K_{con}C = R^{-1}BP$, P is a symmetric matrix. The matrix satisfies the Riccati equation

$$PA + A^T P - PBR^{-1}B^T P + Q = 0 \tag{27}$$

where Q is the given weighted matrix, R is the given weighting coefficient.

For discrete system signal sampling, according to Nyquist-Shannon sampling theorem, sampling period is set as $T_s = 0.001$ s. The related parameter settings of optimal controller and data-driven controller are shown in Tables 2 and 3, respectively.

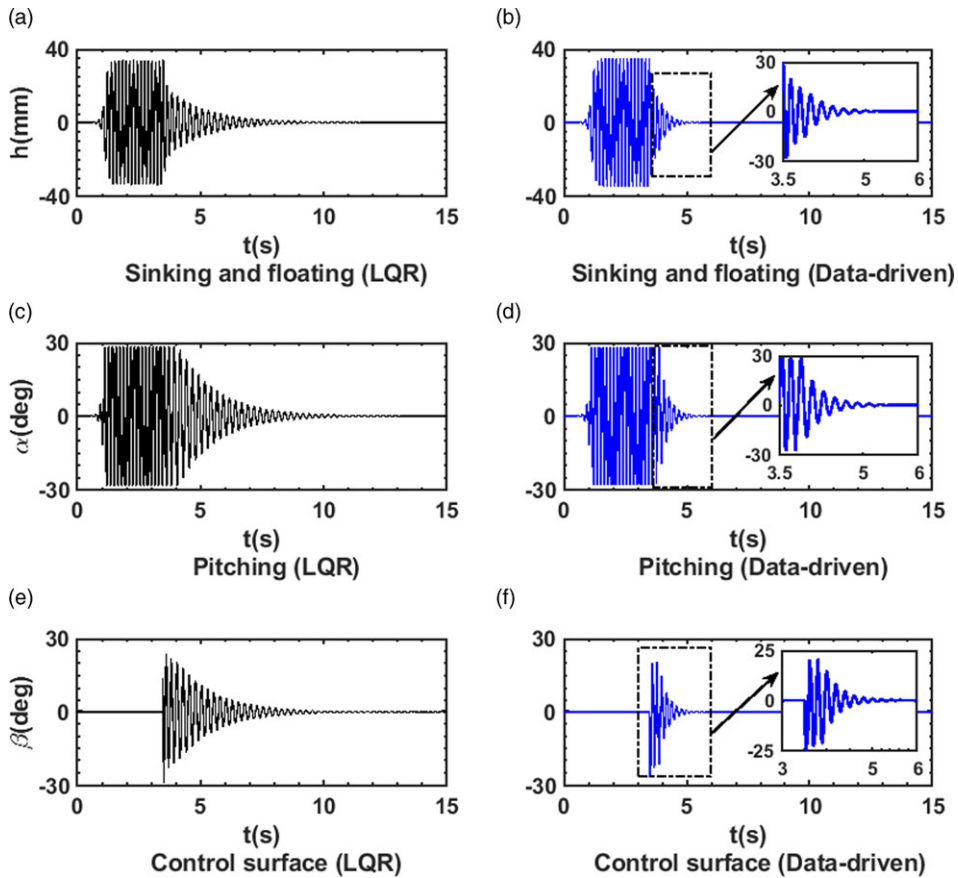


Figure 7. System response ($V = 24$ m/s).

Take different velocities to explore the ability of two controllers. Under the speed of $V = 20$ m/s, control was implemented at 3.5 s respectively. The time history of the system motion before and after the control of the two controllers is shown in Fig. 4.

It can be seen from Fig. 4 that both controllers have good flutter suppression effect when the flow velocity just exceeds the critical velocity, and the response can be attenuated to 0 by only a few deflections of the control surface, but the deflection number of the optimal controller is more than that of the data-driven controller. The saturation of the control surface has a certain influence on the effect of the optimal controller, and the control surface reaches the upper limit of saturation in the initial stage. This is because the optimal controller has no adaptive ability, and the control strategy tends to be 'drastic' in the initial stage, resulting in the control surface easily reaching the upper limit of saturation. In contrast, data-driven controllers are much more 'restrained'. The phase locus of the system before and after control is shown in Fig. 5. The phase locus shows that although both controllers can make the system state return to the equilibrium point quickly, the regression path of the optimal controller is too sharp, while that of the data-driven controller is relatively smooth. Figure 6 shows the evaluation curve of the filter on the system pseudo partial derivative. It can be seen that the estimation value can quickly converge to the real value after the controller starts to work.

Under the speed of $V = 24$ m/s, control was also implemented at 3.5 s respectively. The time history of the system motion before and after the control of the two controllers is shown in Fig. 7.

As shown in Fig. 7, the flutter of the wing becomes more intense with the increase of flow velocity. The flutter suppression effect of the optimal controller decreases significantly, and it takes a long

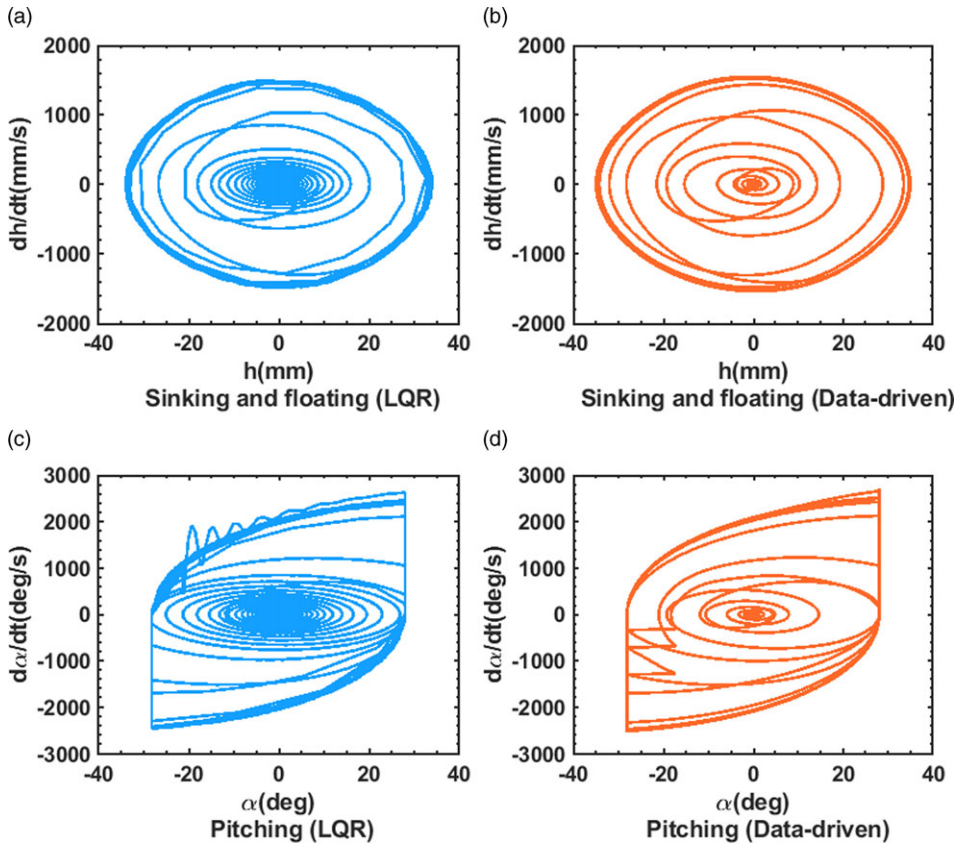


Figure 8. System state phase locus ($V = 24$ m/s).

time to attenuate the response, while the data-driven controller is obviously better. Due to its powerful adaptive ability, the data-driven controller can adjust the parameters in time according to the size of the system output, the amplitude of the control signal can change rapidly, the filter can master the system characteristics throughout the process, and the control cost can be accurately adjusted to avoid excessive control inertia. On the contrary, the optimal controller has no adaptive ability, the control signal has large inertia and the amplitude adjustment is slow, resulting in that the additional aerodynamic force generated by every deflection of the control surface is either too large or too small and the optimal controller needs to repeatedly adjust for a long time, which is the defect of traditional model-based optimal control. We can regard the reciprocal of the deflection number of the control surface as the capability margin of the controller in a certain sense. With the increase of the deflection number, the capability margin of the controller keeps approaching 0, which means that the controller is getting closer and closer to failure. Obviously, the data-driven controller still has a large margin space.

The design of optimal controller depends on the modeling of aeroelastic system, which is complicated for designers, while the design of data-driven controller avoids the modeling process. So from the practical point of view, the data-driven controller is much cheaper, which has a systematic design process, and it only needs to set a few initial parameters. In addition, for flutter, designers always want the response to decay quickly, otherwise prolonged oscillations increase the risk of fatigue in the wing structure. So the data-driven controller is also better in terms of the time it takes for the response to decay.

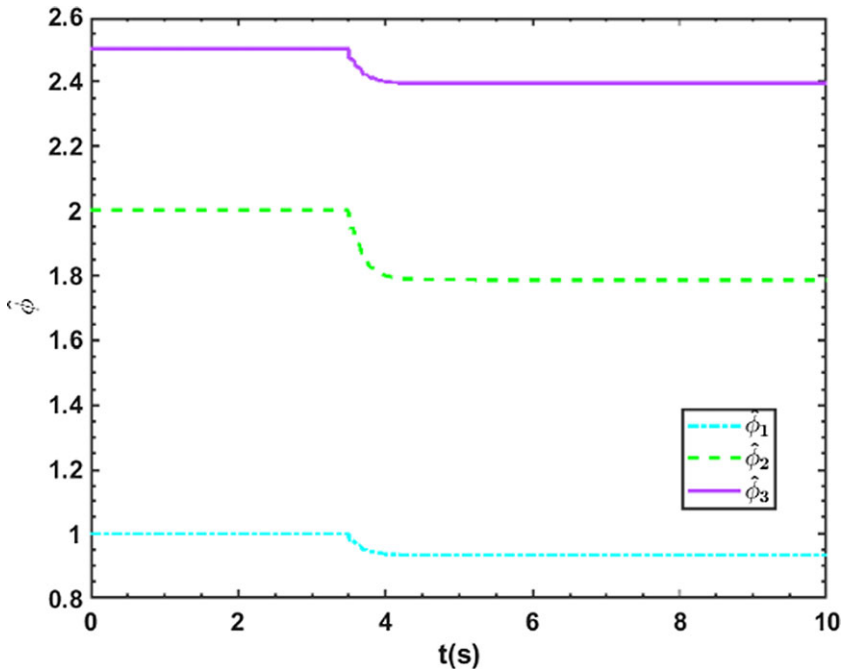


Figure 9. Evaluation curve of system pseudo partial derivative ($V = 24$ m/s).

The phase locus of the system before and after control is shown in Fig. 8. At this time, the phase locus shows that when the wind speed increases, the equilibrium point regression paths of the two controllers become longer, while the optimal controller obviously increases more, and the sharp problem of its regression path still exists. Under the action of data-driven controller, the system state will find a relatively short path for equilibrium point regression. However, under the action of the optimal controller, the system state will be attracted by the limit cycle and can only return to the equilibrium point after continuous circling around the periphery. Figure 9 shows the evaluation curve of the filter on the system pseudo partial derivative under the speed of $V = 24$ m/s. It can be seen that the estimation value can still quickly converge to the real value after the controller starts to work, and the structural characteristics of the system at this time have changed due to the change of the flow velocity, so the pseudo partial derivative estimated by the filter is also different from that under the speed of $V = 20$ m/s.

6.0 Conclusion

In this paper, an adaptive control method based on ridge regression is proposed and applied to active flutter suppression, and the following conclusions are obtained:

- (1) The controller designed by this method only uses the real-time measured online input and output data of the closed-loop controlled system, which can realise the system correction without identifying the system structure. Compared with the traditional feedback control, which requires model-based design, the design based on ridge regression is model-free.
- (2) The controller structure designed by this method is relatively simple, easy to modify and program. The calculation amount is acceptable. It is very suitable for the digital control system widely used in the aviation industry.

- (3) The focus of this method is to establish a regression model at each working point of the system. In essence, this is a kind of dynamic linearisation. Different from the traditional linearisation in a small range near the equilibrium point, this kind of dynamic linearisation is accurate.
- (4) The cores of this method are inner product operation and regularisation. Inner product operation establishes projective information between data, and regularisation eliminates singularity through diagonal loading. It is worth pointing out that the method proposed in this paper is the foundation, and appropriate operations for these two cores can be used to extend the method.

Competing of interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

[1] Qian, W., Huang, R., Hu, H. and Zhao, Y. New method of modeling uncertainty for robust flutter suppression, *J. Aircraft*, 2013, **50**, (3), pp 994–999. Retrieved from <Go to ISI>://WOS:000320075800034. doi: [10.2514/1.C031987](https://doi.org/10.2514/1.C031987)

[2] Huang, R., Hu, H. and Zhao, Y. Single-input/single-output adaptive flutter suppression of a three-dimensional aeroelastic system, *J. Guid. Control Dyn.*, 2012, **35**, (2), pp 659–665. Retrieved from <Go to ISI>://WOS:000301631100030. doi: [10.2514/1.55746](https://doi.org/10.2514/1.55746)

[3] Zeng, J., Kukreja, S.L. and Moulin, B. Experimental model-based aeroelastic control for flutter suppression and gust-load alleviation, *J. Guid. Control Dyn.*, 2012, **35**, (5), pp 1377–1390. Retrieved from <Go to ISI>://WOS:000314378900001. doi: [10.2514/1.56790](https://doi.org/10.2514/1.56790)

[4] Schmidt, D.K. Stability augmentation and active flutter suppression of a flexible flying-wing drone, *J. Guid. Control Dyn.*, 2016, **39**, (3), pp 409–422. Retrieved from <Go to ISI>://WOS:000382522000001. doi: [10.2514/1.G001484](https://doi.org/10.2514/1.G001484)

[5] Huang, R., Zhao, Y. and Hu, H. Wind-tunnel tests for active flutter control and closed-loop flutter identification, *AIAA J.*, 2016, **54**, (7), pp 2089–2099. Retrieved from <Go to ISI>://WOS:000378521600007. doi: [10.2514/1.J054649](https://doi.org/10.2514/1.J054649)

[6] Theis, J., Pfifer, H. and Seiler, P. Robust modal damping control for active flutter suppression, *J. Guid. Control Dyn.*, 2020, **43**, (6), pp 1056–1068. Retrieved from <Go to ISI>://WOS:000537277700002. doi: [10.2514/1.G004846](https://doi.org/10.2514/1.G004846)

[7] Vepa, R. and Kwon, J.R. Synthesis of an active flutter suppression system in the transonic domain using a computational model, *Aeronaut. J.*, 2021, **125**, (1293), pp 2002–2020. Retrieved from <Go to ISI>://WOS:000721293500007. doi: [10.1017/aer.2021.38](https://doi.org/10.1017/aer.2021.38)

[8] Yang, Z., Huang, R., Zhao, Y. and Hu, H. Design of an active disturbance rejection control for transonic flutter suppression, *J. Guid. Control Dyn.*, 2017, **40**, (11), pp 2905–2916. Retrieved from <Go to ISI>://WOS:000412938800013. doi: [10.2514/1.G002690](https://doi.org/10.2514/1.G002690)

[9] Yang, Z., Huang, R., Zhao, Y. and Hu, H. Transonic flutter suppression for a three-dimensional elastic wing via active disturbance rejection control, *J. Sound Vibr.*, 2019, **445**, pp 168–187. Retrieved from <Go to ISI>://WOS:000457867400013. doi: [10.1016/j.jsv.2019.01.006](https://doi.org/10.1016/j.jsv.2019.01.006)

[10] Hou, Z. and Bu, X. Model free adaptive control with data dropouts, *Expert Syst. Appl.*, 2011, **38**, (8), pp 10709–10717. Retrieved from <Go to ISI>://WOS:000290237500189. doi: [10.1016/j.eswa.2011.01.158](https://doi.org/10.1016/j.eswa.2011.01.158)

[11] Bu, X., Hou, Z., Yu, F. and Fu, Z. Model free adaptive control with disturbance observer, *Control Eng. Appl. Inf.*, 2012, **14**, (4), pp 42–49. Retrieved from <Go to ISI>://WOS:000313377700006.

[12] Hou, Z. and Zhu, Y. Controller-dynamic-linearization-based model free adaptive control for discrete-time nonlinear systems, *IEEE Trans. Ind. Inf.*, 2013, **9**, (4), pp 2301–2309. Retrieved from <Go to ISI>://WOS:000326113700049. doi: [10.1109/tii.2013.2257806](https://doi.org/10.1109/tii.2013.2257806)

[13] Darabseh, T.T., Tarabulsi, A.M. and Mourad, A.H.I. Active flutter suppression of a two-dimensional wing using linear quadratic gaussian optimal control, *Int. J. Struct. Stab. Dyn.*, 2022, **22**, (14). Retrieved from <Go to ISI>://WOS:000861439000011. doi: [10.1142/s0219455422501577](https://doi.org/10.1142/s0219455422501577)

Appendix

For subsonic incompressible flows, according to Theodorson theory, the aerodynamic lift force acting on a binary wing with span of s_p and the aerodynamic torque against the elastic axis can be written as

$$L = \pi \rho_a b^2 s_p \left[\ddot{h} + V\dot{\alpha} - b\bar{a}\ddot{\alpha} - (V/\pi) T_4 \dot{\beta} - (b/\pi) T_1 \ddot{\beta} \right] + 2\pi \rho_a V b s_p C(k) \left[V\alpha + \dot{h} + b(0.5 - \bar{a}) \dot{\alpha} + (V/\pi) T_{10} \beta + (b/2\pi) T_{11} \dot{\beta} \right] \tag{A.1}$$

$$T_\alpha = \pi \rho_a b^2 s_p \left\{ b\bar{a}\ddot{h} - Vb(0.5 - \bar{a}) \dot{\alpha} - b^2(1/8 + \bar{a}^2) \ddot{\alpha} - (V^2/\pi) (T_4 + T_{10}) \beta + (Vb/\pi) [-T_1 + T_8 + (\bar{c} - \bar{a}) T_4 - 0.5T_{11}] \dot{\beta} + (b^2/\pi) [T_7 + (\bar{c} - \bar{a}) T_1] \ddot{\beta} \right\} + 2\pi \rho_a V b^2 s_p (\bar{a} + 0.5) C(k) \left\{ V\alpha + \dot{h} + b(0.5 - \bar{a}) \dot{\alpha} + (V/\pi) T_{10} \beta + (b/2\pi) T_{11} \dot{\beta} \right\} \tag{A.2}$$

where T_i ($i = 1, 4, 7, 8, 10, 11$) is the Theodorson constant dependent on the position of the elastic axis and the position of the hinge line of the control surface, the specific algorithm is as follows; Theodorson function $C(k)$ is a complex function.

$$T_1 = -\frac{2 + \bar{c}^2}{3} \sqrt{1 - \bar{c}^2} + \bar{c} \arccos \bar{c} \tag{A.3}$$

$$T_4 = \bar{c} \sqrt{1 - \bar{c}^2} - \arccos \bar{c} \tag{A.4}$$

$$T_7 = \bar{c} \frac{7 + 2\bar{c}^2}{8} \sqrt{1 - \bar{c}^2} - \left(\frac{1}{8} + \bar{c}^2 \right) \arccos \bar{c} \tag{A.5}$$

$$T_8 = -\frac{1}{3} (1 + 2\bar{c}^2) \sqrt{1 - \bar{c}^2} + \bar{c} \arccos \bar{c} \tag{A.6}$$

$$T_{10} = \sqrt{1 - \bar{c}^2} + \arccos \bar{c} \tag{A.7}$$

$$T_{11} = (2 - \bar{c}) \sqrt{1 - \bar{c}^2} + (1 - 2\bar{c}) \arccos \bar{c} \tag{A.8}$$

Jones form of Theodorson function is conducive to the time-domain state-space modeling of aeroelastic systems, which can be written as

$$C(s) = 1 - \frac{0.165s}{s + 0.0455\frac{V}{b}} - \frac{0.335s}{s + 0.3\frac{V}{b}} = \frac{1}{2} + \frac{0.0075\frac{V}{b}}{s + 0.0455\frac{V}{b}} + \frac{0.10055\frac{V}{b}}{s + 0.3\frac{V}{b}} = \frac{1}{2} + \frac{z_1\frac{V}{b}}{s + p_1\frac{V}{b}} + \frac{z_2\frac{V}{b}}{s + p_2\frac{V}{b}} \tag{A.9}$$

where $z_1 = 0.0075$, $z_2 = 0.10055$, $p_1 = 0.0455$, $p_2 = 0.3$, and s is Laplace variable. If $C(s)$ is regarded as the transfer function of a certain system, the corresponding state-space equation of the system can be written as

$$\begin{cases} \begin{bmatrix} \dot{x}_{a1} \\ \dot{x}_{a2} \end{bmatrix} = \frac{V}{b} \begin{bmatrix} -p_1 & 0 \\ 0 & -p_2 \end{bmatrix} \begin{bmatrix} x_{a1} \\ x_{a2} \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} r = \mathbf{V}\mathbf{A}_a\mathbf{x}_a + \mathbf{V}\mathbf{K}_a\mathbf{q}_s + \mathbf{D}_a\dot{\mathbf{q}}_s \\ y_a = \frac{V}{b} \begin{bmatrix} z_1 & z_2 \end{bmatrix} \begin{bmatrix} x_{a1} \\ x_{a2} \end{bmatrix} + \frac{1}{2}r \end{cases} \tag{A.10}$$

$$r = V\alpha + \dot{h} + b(0.5 - \bar{a})\dot{\alpha} + (V/\pi)T_{10}\dot{\beta} + (b/2\pi)T_{11}\dot{\beta} = \mathbf{S}_1\dot{\mathbf{q}}_s + \mathbf{V}\mathbf{S}_2\mathbf{q}_s \tag{A.11}$$

where r is the system input, $\mathbf{S}_1 = [1 \quad b(0.5 - \bar{a}) \quad (b/2\pi)T_{11}]$, $\mathbf{S}_2 = [0 \quad 1 \quad T_{10}/\pi]$; x_{a1} and x_{a2} are the aerodynamic state variables, $\mathbf{x}_a = [x_{a1} \quad x_{a2}]^T$; $\mathbf{A}_a = \frac{1}{b} \begin{bmatrix} -p_1 & 0 \\ 0 & -p_2 \end{bmatrix}$, $\mathbf{K}_a = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, \mathbf{S}_2 , $\mathbf{D}_a = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ \mathbf{S}_1 . The calculation formula of aerodynamic force can be rewritten as

$$-L = -\pi \rho_a b^2 s_p [V\dot{\alpha} - (V/\pi)T_4\dot{\beta} - (b/\pi)T_1\ddot{\beta}] - \pi \rho_a b^2 s_p [\ddot{h} - b\bar{a}\ddot{\alpha}] - 2\pi \rho_a V b s_p y_a \tag{A.12}$$

$$T_\alpha = \pi \rho_a b^2 s_p \left\{ \begin{aligned} & -Vb(0.5 - \bar{a})\dot{\alpha} - (V^2/\pi)(T_4 + T_{10})\dot{\beta} \\ & + (Vb/\pi)[-T_1 + T_8 + (\bar{c} - \bar{a})T_4 - 0.5T_{11}]\dot{\beta} + (b^2/\pi)[T_7 + (\bar{c} - \bar{a})T_{11}]\ddot{\beta} \end{aligned} \right\} \\ + \pi \rho_a V b^3 s_p [\bar{a}\ddot{h} - b(1/8 + \bar{a}^2)\ddot{\alpha}] + 2\pi \rho_a V b^2 s_p (\bar{a} + 0.5)y_a \tag{A.13}$$

The aerodynamic force vector \mathbf{F}_{ae} is written as the sum of the non-cyclic part and the cyclic part, i.e

$$\mathbf{F}_{ae} = \mathbf{F}_{nc} + \mathbf{F}_c \tag{A.14}$$

The non-cyclic part F_{nc} can be written as

$$F_{nc} = -M_{nc}\ddot{q}_s - VD_{nc}\dot{q}_s - V^2K_{nc}q_s \tag{A.15}$$

where

$$M_{nc} = \rho_a b^2 s_p \begin{bmatrix} \pi & -\pi b \bar{a} & -b T_1 \\ -\pi b \bar{a} & \pi b^2 (1/8 + \bar{a}^2) & -b^2 (T_7 + (\bar{c} - \bar{a}) T_1) \\ 0 & 0 & 0 \end{bmatrix}$$

$$D_{nc} = \rho_a b^2 s_p \begin{bmatrix} 0 & \pi & -T_4 \\ 0 & \pi b (0.5 - \bar{a}) & -b (-T_1 + T_8 + (\bar{c} - \bar{a}) T_4 - 0.5 T_{11}) \\ 0 & 0 & 0 \end{bmatrix}$$

$$K_{nc} = \rho_a b^2 s_p \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & T_4 + T_{10} \\ 0 & 0 & 0 \end{bmatrix}$$

The cyclic part F_c can be written as

$$F_c = -\rho_a V b s_p R_c \left(\frac{V}{b} [z_1 \ z_2] \begin{bmatrix} x_{a1} \\ x_{a2} \end{bmatrix} + \frac{1}{2} (S_1 \dot{q}_s + V S_2 q_s) \right) \tag{A.16}$$

where $R_c = [2\pi \ -2\pi b (0.5 + \bar{a}) \ 0]^T$. The cyclic part F_c can be further written as

$$F_c = -VD_c \dot{q}_s - V^2 K_c q_s - V^2 E_c x_a \tag{A.17}$$

where $D_c = \frac{1}{2} \rho_a b s_p R_c S_1$, $K_c = \frac{1}{2} \rho_a b s_p R_c S_2$, $E_c = \rho_a s_p R_c [z_1 \ z_2]$. In summary, the original wing motion equation can be written as

$$(M_s + M_{nc}) \ddot{q}_s + (D_s + VD_{nc} + VD_c) \dot{q}_s + (K_s + V^2 K_{nc} + V^2 K_c) q_s = -V^2 E_c x_a + G_s \delta \tag{A.18}$$

The state space expression of the aeroelastic equation of a binary wing can be obtained by combining the wing motion equation with the aerodynamic state equation.

$$\dot{X} = AX + B\delta \tag{A.19}$$

where $A = \begin{bmatrix} \mathbf{0}_{3 \times 3} & \mathbf{I}_{3 \times 3} & \mathbf{0}_{3 \times 2} \\ -M^{-1}K & -M^{-1}D & -V^2 M^{-1}E_c \\ VK_a & D_a & VA_a \end{bmatrix}$, $B = \begin{bmatrix} \mathbf{0}_{3 \times 1} \\ M^{-1}G_s \\ \mathbf{0}_{2 \times 1} \end{bmatrix}$, $X = [h \ \alpha \ \beta \ \dot{h} \ \dot{\alpha} \ \dot{\beta} \ x_{a1} \ x_{a2}]^T$, $M = M_s + M_{nc}$, $D = D_s + VD_{nc} + VD_c$, $K = K_s + V^2 K_{nc} + V^2 K_c$.