Smart wing load alleviation through optical fiber sensing, load identification, and deep reinforcement learning

Daichi Wada 1,∗, Masato Tamayama 1 and Hideaki Murayama 2

1 Aeronautical Technology Directorate, Japan Aerospace Exploration Agency, Osawa, Mitaka-shi, Tokyo 181-0015, Japan
2 Department of Ocean Technology, The University of Tokyo, Kashiwanoha, Kashiwa-shi, Chiba 277-8561, Japan
∗ Author to whom any correspondence should be addressed.
E-mail: wada.daichi@jaxa.jp

Keywords: deep reinforcement learning, Fiber Bragg grating, load alleviation, load identification, optical fiber sensor, structural monitoring, aircraft wing

Abstract

The use of optical fiber sensors has been considered to realize smart structures, which can sense and respond to different environments [1]. Many animals and plants use their nerve endings to sense, nerves to convey signals, brains to process, and muscles and hormones to react to their environments. In a similar manner, optical fiber sensors can be used to measure strain distributions on the wing surface. Considering the strains, a group of neural networks determine the wing load distributions and angle of attacks. This information is fed into a controller that drives multiple flaps to re-distribute the loads. The controller is trained via a deep reinforcement learning technique. The wind tunnel experiments demonstrated that the proposed closed-loop control could alleviate the bending moment by 56.6% on average over the test duration from the initial state while the total load variations could be maintained within a range of ±5 N for 87.1% of the test duration. The proposed approach was also applicable to another scenario involving variations in the target loads, and the results indicated the generalized applicability of the neural-network-based controller trained via deep reinforcement learning.

Nomenclature

\( \alpha \) Angle of attack at wing root
\( \hat{\alpha} \) Identified angle of attack at wing root
\( \delta \) Flap angles
\( e \) Strain distribution
\( F \) Wing load distribution
\( \hat{F} \) Identified wing load distribution
\( F_{\text{total}} \) Total wing load \( \sum F \)
\( F_{\text{ref}} \) Reference of total wing load
\( M \) Bending moment at wing root
\( M_{\text{ref}} \) Reference of bending moment at wing root
\( f_{\text{env}} \) Environment function to calculate load distribution \( F = f_{\text{env}}(\delta, \alpha) \)

1. Introduction

The use of optical fiber sensors has been considered to realize smart structures, which can sense and respond to different environments [1]. Many animals and plants use their nerve endings to sense, nerves to convey signals, brains to process, and muscles and hormones to react to their environments. In a similar manner, optical fiber
sensors act as the nerve endings and data links for smart structures. This concept is particularly valuable in aviation, to develop smart aircraft that can sense their structural states and provide a suitable response by adjusting the flight performance and envelopes.

Optical fiber sensing technique has been actively studied for application in aircraft structural monitoring. The low weight, thin form, immunity to electromagnetic fields, and distributed sensing capability of such sensors makes their installation minimally invasive while allowing the collection of spatially high-density data. Several researchers have monitored the wings of unmanned aerial vehicles (UAVs) by using multiplexed fiber Bragg gratings (FBGs) [2–4], to investigate the time histories of the measured strains and wing deformations in flight. Passenger aircraft monitoring has also been realized by fully distributed sensing techniques. Specifically, a vertical tail [5], fuselage [6] and main wings [7, 8] were monitored during flight, and a correlation between the strain distribution histories and aircraft maneuvers was observed. The fully distributed sensing realized observations and understandings of the overall structural responses of the wings and the fuselage to flight conditions. With a high spatial resolution, local stress concentrations were also monitored, which would be utilized to detect damages [6, 7]. Optical fiber distributed sensing has demonstrated promising applicability for structural health and usage monitoring.

To enhance the aircraft performance, it is necessary to reduce the loads on the aircraft. Reducing the loads experienced by an aircraft during operation can help reduce the structural weight, thereby improving the energy efficiency. This aspect is especially beneficial in the case of wings with challenging structural designs. For example, although high-aspect ratio wings produce less induced drag, they tend to suffer larger bending moments at the wing roots. Dynamic and static load alleviation can help ensure positive margins of the structural integrity, thereby enhancing the feasibility of aggressive designs and allowing the exploitation of the advantages of such designs. Wing tip folding can facilitate ground handling owing to the reduction of the span length and can act as a passive load alleviation technique [9, 10]. Furthermore, the use of continuous trailing-edge flaps can enhance the drag reduction effect of flexible wings by changing the airfoil distributions in the span–wise direction [11]. This type of active control of the aerodynamic performance can also lead to load alleviation and flutter suppression. It has been noted that the active control of multiple flaps enables a redistribution of the aerodynamic load, and thus, the structural loads on the wings reduce over the complete mission profile [12]. In this regard, optical fiber sensors have been used to determine the wing deflections, based on which, a proportional controller was used to actuate the flaps.

To alleviate loads in a direct and adaptive manner, that is, to control loads in real time against a feedback of the actual applied loads, the actual loads must be monitored. It is not feasible to directly measure aerodynamic loads, because complex wing structures and weight limitations inhibit the installation of several pressure sensors and tubes. In this case, the loads can be estimated based on structural responses such as strains, although the obtained solutions may be unstable because the inverse problems to solve loads from strains tend to be ill-conditioned specifically when loads are distributed loads [13–15]. Consequently, several techniques to obtain stable solutions have been proposed, such as by assuming polynomial or Fourier series terms for the load distribution functions [16, 17] and conducting finite-element-based inverse analyses [18]. In addition, some researchers attempted to train neural networks to represent the strain–load relationships [19–21]. The feasibility of neural networks to identify the wing load distributions when fed with distributed optical fiber strain sensing information has been demonstrated [22].

Recent advances in deep reinforcement learning techniques provide promising opportunities to realize load control. In such techniques, a reward function can be designed to reflect the desirable behavior, which could be multi-objective, such as alleviating wing moments while maintaining the lift loads. The controller learns to develop an optimum control law that can be as nonlinear as necessary to maximize the expected reward. Once trained, the neural network controllers output the optimum actions for every state feedback without the need of iterative optimization, which helps in realizing real-time and adaptive control. The powerful task solving capabilities of deep reinforcement learning have been demonstrated in the game playing domain [23, 24] and has thus been extended to the aviation domain. In a simulation analysis, the attitude control of a quadrotor was realized using deep reinforcement learning [25]. The neural–network–based controller outperformed conventional proportional, integral, and derivative (PID) control systems in terms of the accuracy and agility. Furthermore, aerobatic maneuvers and attitude control for fixed-wing UAVs were demonstrated in simulation analyses [26, 27]. A control strategy for shape-memory alloy actuators was investigated using deep reinforcement learning to change airfoils [28]. Nevertheless, the application of deep reinforcement learning in this domain and its role in load alleviation control have not been extensively investigated yet. In this regard, experimental investigations must also be performed to assess the applicability of this technique.

Considering this background, in this study, a load alleviation technique was developed by integrating optical fiber sensing, load identification, and deep-reinforcement-learning–based control technique in a closed-loop manner, to demonstrate the framework for a smart wing, which can sense, process, and respond adaptively. A high-aspect-ratio wing with multiple flaps was considered. Optical fibers with multiplexed FBGs measured the
strain distributions on the wing surface, based on which, the load distributions and angle of attacks were identified. The flaps were maneuvered on the basis of the identified data. The controller was trained using a deep reinforcement learning technique, and the integrated system was evaluated in wind tunnel tests.

2. Method

2.1. Objective

The objective of this study was to alleviate structural loads by integrating sensing, identification, and control techniques. A wing was designed to monitor the flight state through structural sensing and identification techniques, and this information was used to adaptively controls the control surfaces.

Figure 1 shows the control diagram. Static load control was assumed, and a wing with multiple flaps and optical fiber sensors was developed. Multiplexed FBGs were used to enable effective and minimally invasive sensing of the distributed strains \( \varepsilon \) on the wing surface. Based on the sensing data and the flap angles \( \delta \), the spanwise distributed wing loads \( \tilde{F} \) and angle of attack \( \alpha \) were determined. To realize the state identification, a group of neural networks developed in a previous study [22] was employed. The flaps were controlled by another neural network, whose inputs included the identified states \( \tilde{F} \) and \( \alpha \). This neural-network-based controller was trained through a deep reinforcement learning approach, in which the controller objective was to minimize the bending moment at the wing root \( M = \tilde{F} \cdot x \), where \( x \) is the array of the distances from the wing root to the corresponding \( F \) elements. \( \tilde{F} = (f_1, f_2, \ldots, f_n) \) represents the discrete loads corresponding to the out-of-plane wing load distribution along the wingspan. To maintain stable level flight while reducing the bending moment, the total lift load was required to be maintained at a constant value. For simplicity, it was assumed that the total out-of-plane wing load \( F_{\text{total}} = \sum f_i \) remained constant. Consequently, the neural-network-based controller was trained considering the criteria of a higher reward corresponding to a larger moment alleviation \( M_{\text{ref}} = M \) (when \( M_{\text{ref}} > 0, M > 0 \)) and smaller load variation \( |F_{\text{ref}} - F_{\text{total}}| \). Here, \( M_{\text{ref}} \) and \( F_{\text{ref}} \) denote the bending moment and total out-of-plane load at a reference state, respectively.

Instead of directly feeding the FBG and servo signals to the controller, the load identification process was incorporated in the system to decrease the number of inputs to the controller. In general, the values of \( \varepsilon \) tend to be more numerous than those of \( \tilde{F} \). For example, in this study, \( \varepsilon \) and \( \tilde{F} \) involved 60 and 8 data points, respectively. The use of fewer controller inputs reduces both the computation time for the control during the application and the training time during reinforcement learning. A decrease in the number of neurons in the input layer leads to reduced durations for the gradient calculation and neural network update. \( \tilde{F} \) was considered to be an effective parameter for load alleviation control, because the strain values are considerably influenced by the wing structures. In particular, the stiffeners and joints may cause stress concentrations, leading to abrupt strain variations in the local region. By using load values, the training of the controller can be accelerated with easier convergence; however, these aspects were not examined in this work. Furthermore, the identification process was also incorporated as the structural load histories could be recorded, thereby helping understand the reason behind the flap maneuver. This further clarified the relationship between the load alleviation and the applied load distribution profiles. Moreover, the identification process facilitated the ease of modeling. In the performed deep reinforcement learning, the controller and flap shown in figure 1 corresponded to the agent, and the remaining domain was the environment. The environment was represented as a function, to output \( F \) from the inputs of \( \delta \) and \( \alpha \), as \( F = f_{\text{env}}(\delta, \alpha) \). (In experiments, identified loads \( \tilde{F} \) and angle of attack \( \alpha \) were input to the controller instead of theoretical loads \( F \) and angle of attack \( \alpha \) in training. Figure 1 represented the experimental data flow.) In particular, \( f_{\text{env}} \) represented the aerodynamic relationship, and the structural relationship between the strains and loads was not modeled explicitly. Therefore, the modeling of the
environment did not require a knowledge of the sensing method. In other words, the number and locations of the sensors as well as the load identification method did not influence $f_{\text{env}}$. Under the same wing, the reinforcement learning could be performed independent of any changes in the sensing and identification aspects. This feature is particularly beneficial for system designers to modify the sensing and identification methods without repeating the training process.

In summary, in the experiment the group of neural networks, which will be described in detail in section 2.3, identified the loads and angle of attack. These data were input to the neural-network-based controller. In the reinforcement learning, the neural-network-based controller was trained through interactions with the environment function $f_{\text{env}}$, which calculated the loads as a response to the flap maneuver by the controller. $f_{\text{env}}$ was also represented by a neural network, which will be described in section 2.4.

2.2. Experimental: Wing and wind tunnel
A high-aspect-ratio wing with a semi-span length of 3.6 m and the same configuration as that of the wing used in the previous study [22] was employed. The wing schematic is shown in figure 2. The chord lengths at the wing tip and root were 149 mm and 302 mm, respectively. The main and aft spars inside the wing were made of balsa wood, which were covered by woven carbon and glass fiber-reinforced plastic fabric. The wing skin was woven carbon-fiber-reinforced plastic fabric. A 400-mm-long stainless-steel shaft was inserted in the main spar at the wing root to fix the wing to the wind tunnel base. The airfoil was originally designed. Distributions of pressure coefficients with respect to the angle of attack $\alpha$ were also shown in figure 2.

Eight flaps were installed at the trailing edge and numbered from the wing root (Flaps 1–8). The flaps were used to change the load distribution profiles along the wingspan. Servomotors (BLS177SV, Futaba) were used to vary the flap angles between $-15^\circ$ and $15^\circ$. Positive angles indicated downward action, which resulted in an increase in the lift force. At the center of each flap in the spanwise direction, static-pressure-port arrays were installed (red bands in figure 2). Fifteen and eight pressure ports were located on the upper and lower sides of each array, respectively. The out-of-plane components of the aerodynamic loads at eight individual flap sections $F = (f_1, f_2, \ldots, f_8)$ were calculated by integrating the measured static pressures around the airfoil, and the out-of-plane loads were simply referred to as the wing loads. Silicon tubes were used to connect the pressure ports and an interrogator (DTC Initium, Pressure Systems).

Two optical fiber lines were bonded on the upper side of the wing along the main and aft spars. An epoxy adhesive was used to fully cover the optical fibers with the minimum possible thickness. Thirty FBGs with different Bragg wavelengths were inscribed in intervals of 120 mm in each optical fiber, covering the entire

![Figure 2. Schematic of the wing.](image-url)
The 60 Bragg wavelengths were measured in a wavelength division multiplexing manner by using an interrogator (I-MON 512 USB, Ibsen). The Bragg wavelength shifts were linearly correlated to the strains $\varepsilon$. To realize the signal processing, the Bragg wavelength shifts were directly used without being converted to strain values. The shift was calculated considering a reference state in which the wing was stationary in the absence of winds. The measurement data and control commands were processed in a Simulink (MathWorks) environment by using a computer. The pressure sensor tubes, optical fibers, and lead wires of the servomotors were routed to the wing root and connected to the interrogators and the computer, which were set beneath the wind tunnel floor.

Furthermore, the wing was set in a wind tunnel with a 6.5 m high and 5.5 m wide test section. The angle of attack was varied by rotating the turn table on which the wing was fixed. The turn table was moved within the angle of attack range of $1^\circ$ to $5^\circ$ with the speed of $0.145^\circ$ s$^{-1}$. For simplicity, the wind speed was set at a constant value of $14$ m s$^{-1}$, and the dynamic pressure was $118.7$ Pa. The Reynolds number was $2 \times 10^5$. To control the static load conditions and eliminate dynamic effects, the measurements and control were performed in a static manner at a rate of 5 Hz. Visual inspections were performed to ensure that the wing did not vibrate during most of the test duration.

### 2.3. Algorithm: Load and angle of attack identification

The load distribution and angle of attack were key parameters to be input to the controller in real-time. However, it was assumed that these parameters could not be directly obtained using sensors, considering a practical scenario in which several pressure sensors cannot be installed to realize the load estimation. Therefore, these parameters were computed and identified from other observables, specifically, the optical fiber sensing data and flap angles. Although the angle of attack can be measured using a probe, it was included as an identification target. In future work, the angle of attack distribution along the span can be identified and utilized as more detailed information to enhance the control. In this study, only the angle of attack at the wing root was identified.

We utilized a load and angle of attack identification technique developed in the previous study [22]. Specifically, a group of two neural networks was used to identify the load and angle of attack. Figure 3 shows the architecture of the neural network group. The first neural network had a hidden layer with 12 neurons and the hyperbolic tangent sigmoid activation function. The FBG signal $\varepsilon$ and eight flap angles $\delta$ were input, and the load distribution $\hat{F}$ was output. The second neural network had a hidden layer with 10 neurons and the sigmoid activation function. The identified load distribution and flap angles were input, and the angle of attack $\dot{\alpha}$ was output. Thus, the identification was performed by using two neural networks to solve the mechanical inverse problem and aerodynamic parameter estimation problem.

The two neural networks were trained through a priori supervised learning. The neural networks to identify the load and angle of attack were trained sequentially, using the same training data, which were collected experimentally via the wind tunnel test. The wind speed, dynamic pressure, and Reynolds number were the same as described in the previous subsection. Different combinations of the angle of attack and flap angles were established. The angle of attack was varied from $-4.0^\circ$ to $10.0^\circ$ in intervals of $0.5^\circ$. The flap angles were set pseudo-randomly and in unison for 290 and 271 cases, respectively. In the latter cases, all the flaps were aligned, or a single flap was moved while the others remained at neutral angles. In total, 561 training data were collected. All the measurements were conducted in static conditions. A detailed description regarding the training data and hyperparameters applied in the supervised learning can be found in the previous study [22]. In the neural-network-based approach, the errors (standard deviation) in the load and angle of attack for the training data were 0.14 N and 0.15°, respectively.

### 2.4. Algorithm: Deep reinforcement learning to optimize the flap angles

A deep reinforcement learning algorithm based on the deep Q-network (DQN) approach [23, 24], which is a model-free and off-policy algorithm, was employed. The agent considered an action-value function $Q(s, a)$ to
select an action to interact with the environment. Specifically, \( Q(s, a) \) output values for the action choices \( a \) at a given state \( s \). By optimizing \( Q(s, a) \), the best performance could be achieved by choosing the action with the best value. Figure 4 shows the architecture of the neural network representing \( Q(s, a) \). The input state \( s \) consisted of the total load error \( \Delta F = F_{\text{ref}} - \sum F \), load distribution \( F \), flap angle \( \delta \) and angle of attack \( \alpha \). The output was a vector with 24 elements for the flap shifts \( \Delta \delta \), which could be interpreted as a \( 3 \times 8 \) matrix of the action choices. For the eight individual flaps, the angle could be increased by \(-1 \), 0, or 1° in each time step. All the flaps selected an angle increment with the highest value among the three options and updated the flap angles. The angles were considered to be saturated when they were beyond the working range. In this manner, all the eight flaps performed a shift (or chose to remain still) simultaneously. The environment \( f_{\text{env}} \) emulated the wing load distribution \( F \) as an updated state. The agent repeated the following steps: observe state \( s \) of the environment, select an action \( a \) to update the environment, and receive a reward \( r \) as a consequence. Using the Bellman equation, \( Q(s, a) \) was iteratively targeted to be \( r + \gamma \max_a Q(s', a') \), in which \( s' \) and \( a' \) denote the state and the action in the next time step, respectively. To update \( Q(s, a) \), the loss function \( L \) was defined as

\[
L = \left( r + \gamma \max_a Q(s', a') - Q(s, a) \right)^2,
\]

where \( \gamma = 0.95 \) is the discount factor. \( Q(s, a) \) was updated in a supervised manner by using the Levenberg–Marquardt algorithm as an optimizer [29]. The weights and biases were initialized using the Nguyen–Widrow initialization method [30]. The learning parameters were derived from the previous study [29].

The reward \( r \) was defined as

\[
r = \begin{cases} 
-3n_s, & \text{if } 1.2M_{\text{ref}} - M < 0 \text{ or } |F_{\text{ref}} - \sum F| > 5N \\
(1.2M_{\text{ref}} - M) \left( 1 - \left| \frac{F_{\text{ref}} - \sum F}{F_{\text{ref}}} \right| \right)^4 - 3n_s, & \text{otherwise}
\end{cases}
\]

The reference moment \( M_{\text{ref}} \) and total load \( F_{\text{ref}} \) were set equal to the initial values of the individual training episodes. When the moment \( M \) at a time step was larger than 120% the reference, and/or the total load varied more than \( \pm 5N \) from the reference, only the penalty was assigned. The penalty coefficient was empirically chosen. The term \( n_s \) represented the number of flaps actuated (\( \Delta \delta = 0 \)) at the time step, and it was used to suppress excessive flap maneuvering with smaller contributions. In the case in which a reward was assigned, the reward was expressed as the product of the moment alleviation contribution and the load variation suppression. The coefficient of \( M_{\text{ref}}(1.2) \) was always larger than \( M \), and the 4th power index for the load variation term was empirically selected.

Each training episode involved 600 time steps. The schedule for the angle of attack \( \alpha \) variation was designed with different random seeds for the individual episodes as an external disturbance, defined based on the Ornstein–Uhlenbeck process with the following parameters: mean \( \mu = 3^\circ \), volatility \( \sigma = 0.3 \), and reverting rate toward the mean \( \theta = 0.1 \) [31]. This process generated a noise that was correlated to the previous noise, thereby producing \( \alpha \) variation histories that drifted in the same direction for a longer duration instead of oscillating around the mean value. The initial \( \alpha \) at the beginning of each episode was randomly set within \( 2.5^\circ \)–\( 3.5^\circ \) with a uniform probability. Consequently, as shown in figure 5, the \( \alpha \) variation ranged mostly from \( 1^\circ \)–\( 5^\circ \), in which, theoretically, the flaps could be controlled to obtain a total load that had a variation of less than \( \pm 5N \) from the initial value. In each time step, the agent observed \( s \) and calculated the action-value \( Q \). The agent used an \( \epsilon \)-greedy strategy, in which the best action was selected with a probability of \( 1 - \epsilon \) and random actions were selected otherwise. \( \epsilon \) decayed from 1 to 0.05 with a decay rate of 0.05%, and this value was multiplied with the \( \epsilon \) at the end of the episodes. When an action was performed, the state was updated, and the reward was given. This

**Figure 4.** Neural network architecture for the action-value function in deep reinforcement learning.
transition was recorded in an experience storage that included the last 6000 time steps. At the end of the time step, $\alpha$ was updated to the next schedule, and eventually, the load $F$ was updated. These aspects were observed by the agent in the next step. The neural network $Q(s, a)$ was updated at the end of the episodes using a 3000 mini-batch randomly chosen from the storage.

The $\alpha$ range in this study was limited to a linear region; however, in generalized cases, the relationship between the angle of attack and load is nonlinear, as in the case in the previous study, in which a range of $-4^\circ \leq \alpha \leq 10^\circ$ was considered [22]. Moreover, the relative angles between the adjacent flaps could have a correlated effect on the loads, which cannot be modeled through a summation of the independent single flap effects. To represent these nonlinear relationships, the environment $f_{\text{env}}$ was represented by another neural network having a hidden layer with 10 neurons and the sigmoid activation function. The neural network was trained a priori using the same training data and conditions as those for the load identification neural network. As a result of the training, the standard deviation of the error of the neural network was 0.32 N. Figure 6 illustrates the applied load distribution profiles of the wing and their estimates. All the flaps were set at neutral angles. ‘Applied’ denotes the values measured using the pressure sensors. ‘Environment’ represents the values estimated using the environment function as $F = f_{\text{env}}(\delta, \alpha)$ where all the elements of $\delta$ were 0° and $\alpha = -4^\circ$, 0° and 10°. The load identification results, that is, the $F$ estimated using the neural network in figure 3 and the experimental FBG signals, are also shown for reference as ‘Identification’. The load profiles were successfully reproduced using the neural networks for the environment ($f_{\text{env}}$) and identification.

![Figure 5. Sample schedule of the angle of attack.](image1)

![Figure 6. Load distribution along the wingspan for the angle of attack of $-4^\circ$, 0°, and 10°.](image2)
In summary, the neural-network-based-controller was trained via reinforcement learning, in which another neural network was used to represent the environment $f_{\text{env}}$. In the wind tunnel experiments, the FBG sensors and neural network groups identified the loads $\tilde{F}$ and angle of attack $\tilde{\alpha}$, which were fed to the controller.

3. Results

Figure 7 shows the training curves corresponding to the reinforcement learning. To examine the repeatability, three training programs were conducted with different random seeds. In all the cases, the reward steadily increased as the training proceeded to 500 episodes and then saturated, indicating successful training. In some cases, the reward decreased after 1000 episodes, which likely implies the occurrence of overfitting. The neural network controller with the best reward at 950 episodes was applied to the wind tunnel test.

Figure 8 shows the experimental results. The control started at 5 s. As shown in figure 8(a), the step variations of the angle of attack were applied as $3^\circ - 4^\circ - 5^\circ - 4^\circ - 3^\circ - 2^\circ - 1^\circ - 2^\circ - 3^\circ$ by rotating the turn table. The red line shows the angle of attack estimated using the FBG signal. Certain steps, such as 205–222 s, involved offset errors, with an average error of 0.50°. The accuracy is expected to be improved by including more data to train the identification neural network group. The original training data did not include the data set in which the flap angles were arranged to alleviate the wing moment.

The reference load $F_{\text{ref}}$ and moment $M_{\text{ref}}$ were set as 57.7 N and 92.5 Nm, respectively, based on the initial state at 0 s where all the flaps were set at neutral angles and the angle of attack was $3^\circ$. In figure 8(b), the target load range, in which the reward was assigned as described in (2), is shaded in gray. The applied and estimated loads were determined based on the pressure profiles and FBG signals, respectively. The estimated loads exhibited a similar history with the applied loads, and the average offset error was 2.53 N. In the moment alleviation task, under the angle of attack variation, the applied and estimated loads successfully remained within the target load range for 75.0% and 87.1% of the test duration, respectively. The estimated load was monitored and included in the closed-loop control, and therefore, the estimated loads remained longer in the target range than the applied loads. As a reference, the theoretical controller performance was simulated by using the same controller and the applied angle of attack history in the experiment. In this way, the identification error of the angle of attack, which existed in the experiment, was excluded. The load distributions were calculated by the environment function $f_{\text{env}}$. The simulated total load history is indicated in magenta in figure 8(b). The simulated load remained in the target range for 99.1% of the test duration, and this indicated the true controller performance. This performance can potentially be achieved in experiments by improving the sensing and identification accuracy.

Figure 8(c) shows the moment history. The applied and estimated moments exhibited a good agreement, with a standard deviation error of 1.34 Nm. The moment decreased rapidly to the first plateau between 5 and 10 s. This 5 s duration, which corresponded to 25 time steps, was sufficient to change the flap angles to the optimum combinations, while maintaining the load to be as constant as possible. The estimated moment was reduced in a range of 43.7% to 68.0%, with an average reduction of 56.6% from the initial moment value over a duration of 10 to 250 s, thereby indicating the effectiveness of the proposed approach. A similar moment
alleviation was also observed in the simulation, in which the moment reduce in a range of 44.7% to 65.5%, with an average reduction of 52.3%.

Figure 8(d) shows the flap angle histories. The solid and dotted lines indicate the experimental and simulated results, respectively, and the colored regions indicate the individual flaps. In both the experimental and simulation cases, Flaps 1 and 2, located at the wing root, were maintained at $15^\circ$, whereas Flaps 6–8, located at the wing tip, were maintained at $-15^\circ$. This was a reasonable strategy that produced loads at the wing root and mitigated the moments by inducing smaller or negative loads at the wing tip. Flaps 3–5 balanced the load and moment with respect to the angle of attack. Between 10 and 40 s, the flap angles reached a steady state, and the angle of attack was constant at $3^\circ$. The simulated flaps remained constant; however, the experimental results exhibited a transition from one steady state to another at 22 s. A shift from one local optimum to another during a steady state is unreasonable. Around this transition timing, the estimated angle of attack, load, and moment changed. The angle of Flap 6 changed from $-15^\circ$ to $-14^\circ$ and then changed to $-15^\circ$. The trigger for this transition could not be clarified conclusively. This observation highlighted the fact that it is difficult to predict the controller behavior in an actual environment before its actual application. The load remained in the target range before and after the transition. During the experiment, undamped wing vibration occurred when the angle of attack was $4^\circ$. This phenomenon was also observed in the previous study [22]. It was surmised that the vortex around the wing caused resonance. The vibration likely led to oscillations in the estimated load distributions and moments, resulting in the unstable behaviors of Flaps 4 and 5 in the 50–70 s and 100–120 s periods.

During reinforcement learning, the reward was assigned at each time step. This approach encouraged the controller to continue to maximize the reward even in transition phases when step-like angle of attack schedules were applied. This led the trajectory dependence, that is, the flap angle combinations differed at the same angle of attack if the history of the flap angles was different. Figures 9(a) and (b) show the flap angles when the angle of attack was different.
attack was 3° and the load distribution profiles, respectively. The load distributions were observed to correlate with the flap angle combinations. At 15 s, the flap angles in the experiment agreed well with the simulated result. The angle for only Flap 3 was different, likely because of the error in the estimated angle of attack. At 30 s, the flap angles shifted to another combination in the experiment. After the angle of attack variations, the flap angle combinations at 240 s were noted to be different than those at 15 s, which indicated the trajectory dependence. The optimum flap angles that maximized the theoretical reward are represented by the black line in figure 9(a).

This value did not include the flap maneuver penalty. The other flap angles were set to be equivalent to those in the experiment. The angle of attack was 3°.

Figure 9. Flap angles and load distributions for an applied angle of attack of 3°.

Figure 10. Theoretical reward with respect to the Flap 4 angle. The angles of the remaining flap were set equal to those at 240 s in the experiment. The angle of attack was 3°.
was no incentive to maintain the angle of Flap 4 as larger than 10° in theory; however, the simulated controller did not move the flap from 12°, as shown in figure 10(a). This proved that the training could not fully optimize the controller. The same reasoning could be applied to the experimental results, in which Flap 4 angle remained 4°. The slope of the reward function over 1°–10° was 1.6 [deg], which was smaller than the penalty coefficient. A larger reward could be obtained by not moving the flap in a single-time-step perspective, which would have generated a local optimum; however, the training must be conducted to maximize the future reward. In this sense, there is room to improve the training conditions to further optimize the controller performance.

4. Discussion: target load control

The controller was trained to minimize the wing moment as well as the load variation \( |F_{\text{ref}} - F_{\text{total}}| \). Because the intent was to maintain a constant load, \( F_{\text{ref}} \) was set constant in training; however, the configuration of the closed-loop control shown in figure 1 can be applied to \( F_{\text{ref}} \) variations in principle. To investigate the validity of this generalized application, another wind tunnel test was conducted using the same controller, with target load variations. Figure 11 shows the results of this test. The angle of attack was maintained at 3°, and the target load was changed in a step manner with increments of +10, +20, +10, 0, −10, −20, −10 and 0 N from the initial load. After the controller was activated at 5 s, wing and flap angle vibrations were observed, which were damped in 10 s when the flap angles reached a stable state. The angle of attack estimation suffered offset errors with the maximum value of 0.38° over 140–160 s, as shown in figure 11(a). The error amplitude was equivalent to that in the previous case with the angle of attack variations. As shown in figure 11(b), the estimated load successfully followed the changing \( F_{\text{ref}} \) and remained in the target range for 79.7% of the test duration. The average load error

![Figure 11](image-url)
was 3.1 N, and the applied load remained in the target range for 70.0% of the test duration. Spike signals of the load and other parameters were observed at certain transitions of the target load. Step variations of 10 N in the target load were sufficiently large to instantaneously lose the rewards specified in (2). The step variations triggered the simultaneous movement of multiple flaps to realize a rapid maneuver, which resulted in overshoots. Although the scenario was not experienced in training, the controller successfully controlled the flaps to manipulate the loads to remain within the target range. The applied and estimated moments agreed well, as shown in figure 11(c), with a standard deviation error of 1.44 Nm. The moment correlated strongly with the target loads, which reflected the mechanical principle of larger total loads inevitably yielding larger moments. The moment alleviation ratio from the initial moment was 55.3% on average when the target load was equal to the initial load. This was equivalent to the previous result, which indicated that the controller performance was consistent. The moment alleviation ratio was 35.8% for the smallest target load, and an alleviation of 88.3% was attained even when the target load was more than +20 N the initial value. As shown in figure 11(d), Flaps 3–5 balanced the load and moment. This behavior was reasonable and in agreement with the strategy identified from the previous results shown in figure 8(d). In addition, Flap 6 behaved specifically when it was necessary to rapidly increase the loads.

The flap control configuration in figure 1 and the training approach were applicable to a scenario with varying target loads. In this study, the out-of-plane loads were assumed to be controlled. However, the lift loads for aircraft are controlled by specifying relevant target loads in accordance with the attitude. Consequently, it is necessary to estimate and control the drag forces as well, which is a part of future work.

5. Conclusions

We demonstrated a smart wing framework in which the wing sensed, processed, and reacted to the environment. Two lines of optical fibers with multiplexed FBGs were used to measure the strain distributions on the wing surface. Using the sensor information, a group of neural networks identified the load distributions and angle of attacks. Based on this information, the neural-network-based controller maneuvered the flaps to re-distribute the loads to ensure that the bending moment at the wing root was alleviated while the total load remained constant. The controller was trained via deep reinforcement learning, in which the environment was modeled using a neural network that simulated the wing load distributions considering the flap angles and angle of attack. This modeling approach is beneficial as deep reinforcement learning can be performed independent of the structural sensing and identification methods.

The wind tunnel experiments demonstrated that the proposed approach could alleviate the static wing moment by 56.6% on average over the test duration from the initial state, and the total load variations could be maintained within a range of ±5 N for 87.1% of the test duration. The theoretical performance of maintaining the load variation was better in the simulations, which indicated that the control performance could be further enhanced by improving the sensing and identification accuracy. The post analysis on the flap behavior clarified that the controller could be further optimized via reinforcement learning. The controller attempted to unexpectedly drive the flaps in a stationary state. Although the target load range was not violated, this behavior could not be predicted a priori. This facet highlighted that it is difficult to predict and fully evaluate the controller behavior in an actual environment before its actual application. Therefore, an experimental demonstration must be conducted.

The proposed approach was also applicable to another scenario involving variations in the target loads. The results indicated the generalized applicability of the neural-network-based controller trained via deep reinforcement learning. By including the information pertaining to the drag forces and aircraft attitude, the proposed approach can be effectively extended to aircraft control applications.

Acknowledgments

A part of this research was funded by JSPS KAKENHI through grant number JP19K04850.

ORCID iDs

Daichi Wada @ https://orcid.org/0000-0002-1577-0185

References

[1] Udd E 1996 Fiber optic smart structures Proc. of the IEEE 84 60–7
[2] Kressel I et al 2014 High-speed, in-flight structural health monitoring system for medium altitude long endurance unmanned air vehicle 7th European Workshop on Structural Health Monitoring pp 274–80
[3] Kim J H, Park Y, Kim Y Y, Shrestha P and Kim C G 2015 Aircraft health and usage monitoring system for in-flight strain measurement of a wing structure Smart Mater. Struct. 24 1–12
[4] Richards L, Parker A R, Piazza A, Chan P, Harmory P and Pena F 2014 NASA armstrong flight research center (AFRC) fiber optic sensing system (FOSS) technology NASA Technical Reports Server NASA 2015000314
[5] Saito N, Yari T and Enomoto K 2014 Flight demonstration testing with distributed optical fiber sensor 7th European Workshop on Structural Health Monitoring pp 281–8
[6] Wada D et al 2018 Flight demonstration of aircraft fuselage and bulkhead monitoring using optical fiber distributed sensing system Smart Mater. Struct. 27 1–13
[7] Wada D, Igawa H, Tamayama M, Kasai T, Arizono H and Murayama H 2019 Flight demonstration of aircraft wing monitoring using optical fiber distributed sensing system Smart Mater. Struct. 28 1–12
[8] Wada D et al 2018 Fiber-optic simultaneous distributed monitoring of strain and temperature for an aircraft wing during flight Appl. Opt. 57 10458–65
[9] Castrichini A, Siddaramaiah V H, Calderon D E, Cooper J E, Wilson T and Lemmens Y 2016 Nonlinear folding wing tips for gust loads alleviation J. Aircr. 53 1391–9
[10] Castrichini A, Hodigere Siddaramaiah V, Calderon D E, Cooper J E, Wilson T and Lemmens Y 2017 Preliminary investigation of use of flexible folding wing tips for static and dynamic load alleviation Aeronaut. J. 121 73–94
[11] Ting E, Chaparro D, Nguyen N and Fujiwara G E C 2018 Optimization of variable-camber continuous trailing-edge flap configuration for drag reduction J. Aircr. 55 2217–39
[12] Pena F, Martins B L, Diego S, Diego S and Richards W L 2018 Active in-flight load redistribution utilizing fiber-optic shape sensing and multiple control surfaces NASA Technical Reports Server 20180002147
[13] Tarantola A 2005 Inverse Problem Theory and Methods for Model Parameter Estimation (Philadelphia PA: SIAM)
[14] Hansen P C 2010 Discrete Inverse Problems: Insight and Algorithms (Philadelphia PA: SIAM)
[15] Aster R C, Borchers B and Thurber C H 2012 Parameter Estimation and Inverse Problems 2nd edn (Cambridge MA: Academic)
[16] Shkarayev S, Krashantisa R and Tessler A 2001 An inverse interpolation method utilising in-flight strain measurements for determining loads and structural response of aerospace vehicles Third Int. Workshop on Structural Health Monitoring pp 336–43
[17] Coates C W and Thamburaj P 2008 Inverse method using finite strain measurements to determine flight load distribution functions J. Aircr. 45 366–70
[18] Nakamura T, Igawa H and Kanda A 2012 Inverse identification of distributed loads using strain data Aerosp. Sci. Technol. 23 75–84
[19] Cao X, Sugiyama Y and Mitsui Y 1998 Application of artificial neural networks to load identification Comput. Struct. 69 63–78
[20] Trivalio P M and Carn C I L 2006 The inverse determination of aerodynamic loading from structural response data using neural networks Inverse Prob. Sci. Eng. 14 379–95
[21] Wada D, Sugimoto Y, Murayama H, Igawa H and Nakamura T 2019 Investigation of inverse analysis and neural network approaches for identifying distributed load using distributed strains Trans. Japan Soc. Aeronaut. Space Sci. 62 151–61
[22] Wada D and Tamayama M 2019 Wing load and angle of attack identification by integrating optical fiber sensing and neural network approach in wind tunnel test Applied Sciences 9 1–15
[23] Mnih V et al 2013 Playing atari with deep reinforcement learning arXiv:1312.5602
[24] Mnih V et al 2015 Human-level control through deep reinforcement learning Nature 518 529–33
[25] Koch W, Mancuso R, West R and Bestavros A 2018 Reinforcement learning for uav attitude control arXiv:1804.04154
[26] Bohn E, Coates E M, Moe S and Johansen T A 2019 Deep reinforcement learning attitude control of fixed-wing uavs using proximal policy optimization International Conf. on Unmanned Aircraft Systems (IUCAS) (Atlanta, GA, USA) pp 523–33
[27] Clarke S G and Hwang I 2020 Deep reinforcement learning control for aerobatic maneuvering of agile fixed-wing aircraft AIAA SciTech Forum (Orlando, FL, USA)
[28] Goecks V G, Leal P B C, White T D, Valasek J and Hartl DJ 2018 Control of morphing wing shapes with deep reinforcement learning AIAA Information Systems-AIAA Infotech at Aerospace 2018 209989
[29] Hagan M T and Menhaj M B 1994 Training feedforward networks with the marquardt algorithm IEEE Trans. Neural Networks 5 989–93
[30] Nguyen D and Widrow B 1990 Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights IJCNN. International Joint Conf. on Neural Networks pp 21–6
[31] Uhlenbeck G E and Ornstein L S 1930 On the theory of the brownian motion Phys. Rev. 36 823–41