Artificial Neural Network-Based Flight Control Using Distributed Sensors on Fixed-Wing Unmanned Aerial Vehicles

Sergio A. Araujo-Estrada*, and Shane P. Windsor†
Department of Aerospace Engineering, University of Bristol, UK

Conventional control systems for autonomous aircraft use a small number of precise sensors in combination with classical control laws to maintain flight. The sensing systems encode center of mass motion and generally are set-up for flight regimes where rigid body assumptions and linear flight dynamics models are valid. Gain scheduling is used to overcome some of the limitations from these assumptions, taking advantage of well-tuned controllers over a range of design points. In contrast, flying animals achieve efficient and robust flight control by taking advantage of highly non-linear structural dynamics and aerodynamics. It has been suggested that the distributed arrays of flow and force sensors found in flying animals could be behind their remarkable flight control.

Using a wind tunnel aircraft model instrumented with distributed arrays of load and flow sensors, we developed Artificial Neural Network flight control algorithms that use signals from the sensing array as well as the signals available in conventional sensing suites to control angle-of-attack. These controllers were trained to match the response from a conventional controller, achieving a level of performance similar to the conventional controller over a wide range of angle-of-attack and wind speed values. Wind tunnel testing showed that by using an ANN-based controller in combination with signals from a distributed array of pressure and strain sensors on a wing, it was possible to control angle-of-attack. The End-to-End learning approach used here was able to control angle-of-attack by directly learning the mapping between control inputs and system outputs without explicitly estimating or being given the angle-of-attack.

Nomenclature

Roman Symbols

b	Wing span, m
c	Wing mean aerodynamic chord, m
C_P	Pressure coefficient
e_α	Angle of attack tracking error, °
q	Wing model pitch rate, °/s
S	Wing model reference surface, m^2
t_{rs}	rising time, s
V	Wind speed, m/s

Greek Symbols

α	Angle of attack, °
α_d	Angle of attack demand, °
δ_e	Elevator deflection, °
ρ	Air density, kg/m^3

*Research Associate, s.araujoestrada@bristol.ac.uk.
†Senior Lecturer in Aerodynamics.

I. Introduction

Conventional flight control systems for autonomous aircraft traditionally utilize a small number of very precise sensors which give the control system information about the motion of the aircraft's center of mass. For instance, inertial measurement units (IMUs) are used to measure linear acceleration and angular rates of rotation, while a pitot tube is used to measure airspeed. This information is then normally used alongside sensors such as GPS to provide the control system with measurements of the aircraft rigid body motion [1]. This conventional range of sensor input works well when the assumptions of rigid body motion and linearized flight dynamics and aerodynamics apply, but may be limiting when these assumptions are no longer valid; for example, with highly flexible airframes or at high angles of attack. In these cases, the dynamics involved can become highly non-linear and conventional sensors suites may not be sufficient to characterize the dynamic state of the aircraft.

Biological flyers such as birds, bats and insects all have highly flexible wings and bodies, where numerous joints and aeroelastic deformation means that the shape of their wings and the related aerodynamic forces they produce is constantly changing [2]. In addition, they fly in airflows where the magnitude of gusts can be the same order as their airspeed, meaning that they can experience very large unpredictable changes in angle of attack [3]. Rather than being limited by these departures from rigid body motion and liner aerodynamics it appears that biological flyers take advantage of these. For instance, wing flexibility helps reduce the flight costs of locusts [4], flapping motion generates additional lift through leading edge vortex generation in bats [5], and birds appear to use both spatial [6] and temporal variation [7] in the wind to reduce their flight costs.

One aspect all biological flyers appear to have in common, and in contrast to aircraft, is that they have many different sensors distributed on their bodies and aerodynamic surfaces, as well as having localized sensory organs such as eyes, ears and inertial sensors. Insects have many flow sensitive hairs distributed about their bodies, as well as strain sensitive campaniform sensile distributed throughout their wings [8]. Birds have thousands of mechanoreceptors throughout their wings [9] which are thought to encode airflow information [10], while also having proprioceptive muscle spindles which give them information about the positions of the different parts of their body and the forces acting on these [11]. Similarly, bats have flow sensitive hairs distributed over their wings [12], as well as very similar proprioceptive organs to birds. Overall, these types of distributed sensors provide animals with a rich suite of sensory input about the spatial distribution of airflow and loading over their bodies and wings and it would appear that these distributed sensory systems play an important role in their robust and efficient flight control.

The development of distributed airflow sensor systems is a growing research area. Distributed flow sensors have been used in Flush Air Data Systems (FADS) in a number of high performance aircraft, such as the X-33 [13] and the space shuttle [14], where traditional airdata booms were impractical. However, these distributed airflow sensors are also increasingly being developed for small unmanned air vehicles (UAVs) where there is a need for more agile flight control when flying in cluttered turbulent environments [15–17]. These systems have utilized a number of different types of sensors such as diaphragm based pressure sensors [18], hot film sensors [19] and artificial hair sensors [20]. Studies of these systems in wind tunnel testing [21], as well as flight testing [22], have shown the potential of these systems for measuring a range of aerodynamic parameters.

The use of distributed force or strain measurement for flight control is a less well investigated area than distributed flow sensing. Initial work has shown the potential for this type of sensor information to improve flight control, with the potential for faster responses [23] and the use of a physics based control approach [24]. Previous studies have looked at the advantages of each system in isolation in separate aircraft [25], the potential advantages of utilizing both distributed airflow and load information in combination with Artificial Neural Networks (ANN) to obtain estimates of the aerodynamic variables and loads [21], as well as the use of pressure-based distributed control systems for flight control [26] and gust alleviation [27].

There have been some efforts to apply machine learning for flight control purposes, for instance Supervised Learning has been used to obtain an ANN capable of representing a family of linear flight controllers designed at various operating points and validated in a simulation environment [28]. Another application example is the problem of helicopter flight control, which has been addressed using techniques such as Apprenticeship Learning. Using Apprenticeship Learning, an autonomous helicopter flight controller was inferred from example maneuvers from a human pilot, the resulting controller was validated in flight tests and for some maneuvers it outperformed the human pilot [29]. However, overall machine learning has not been applied extensively to flight control and there are significant opportunities to explore how machine learning can be used in this field, with particular opportunities for the flight control of non-conventional platforms or when using unconventional sensor modalities.

In this paper results of research investigating the potential application of bio-inspired distributed sensing in flight control of small scale fixed wing UAVs are presented. This paper is organized as follows: Section II presents the

experimental methods and platforms used for this research and their main characteristics. Section III presents results from wind tunnel tests carried out to characterize the aircraft model for control law design purposes. Section IV presents wind tunnel closed loop control results using a conventional controller and various distributed sensing-based controllers. Section V discusses the information presented in this paper as well as areas of potential application for flight control using distributed sensing. Section VI summarizes the information presented in this paper.

II. Experimental Platform Description

A. Aircraft Wind Tunnel Model

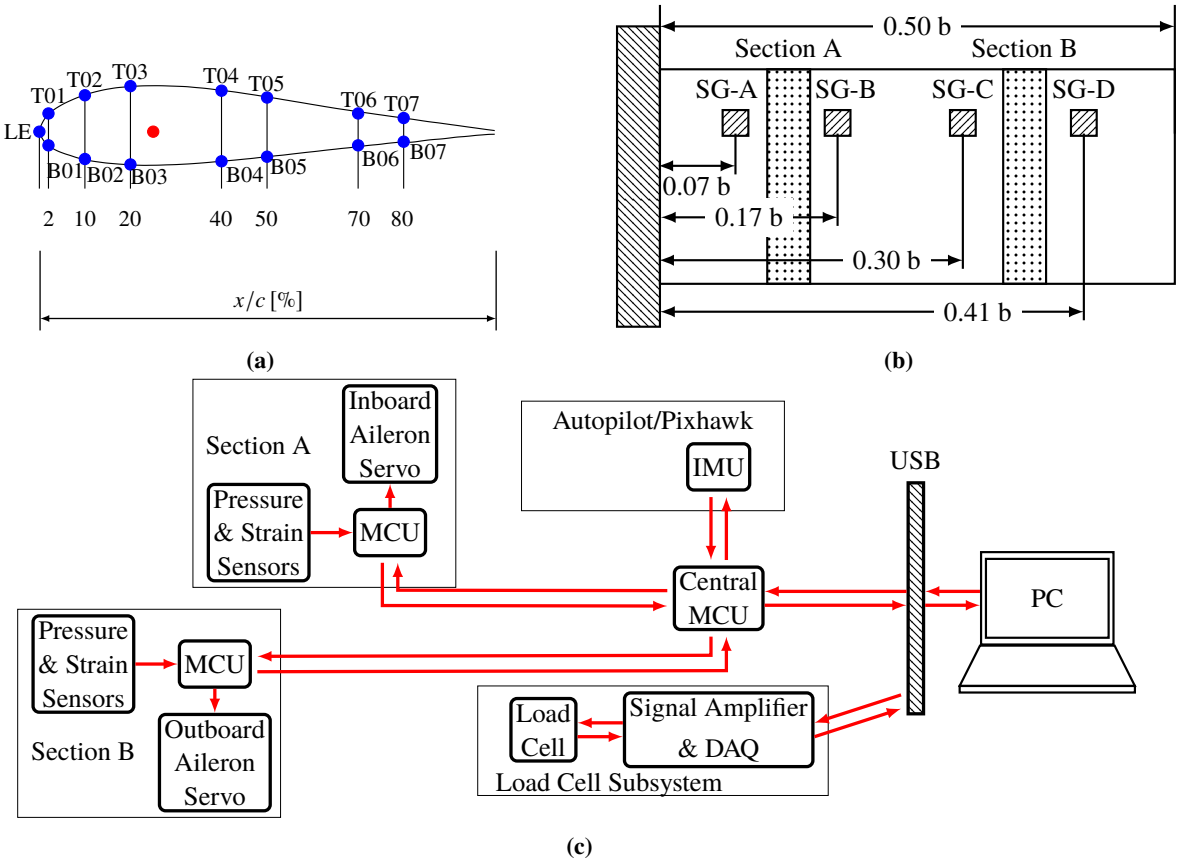

Figure 1 Experimental platform: (a) chord-wise pressure array distribution, (b) span-wise strain array distribution and (c) data acquisition system block diagram.

A wind tunnel model with a distributed array sensors was designed and integrated. The model was constructed using the starboard side of a radio control aircraft model (WOT4 Foam-E Mk2+, Ripmax, Enfield, England). The wing was of rectangular shape, with a wingspan $b = 1.205$ m and a chord $c = 0.25$ m. Only half of the wing was used for the model, i.e. the wind tunnel model wingspan was 0.602 m. A total of 30 pressure sensors were connected to pressure ports distributed along the wing chord at two different locations along the span using 3D-printed inserts. The chord-wise location of the pressure ports is shown in Figure 1a, while the span-wise location of the 3-D inserts is shown in Figure 1b. This arrangement provided information on the chord-wise pressure distribution of the wing at two locations along the span. An array of 16 strain gauges was installed on the wing spar, an aluminum beam located at the quarter chord of the wing. The strain gauges were divided into four sub-arrays, with each sub-array able to measure vertical bending and torsional moments. Figure 1b shows the span-wise location of each strain gauge sub-array. The sensor technical specifications are given in Table 1. Higher range pressure sensors were used in parallel with lower range sensors for some ports where there was potential for the sensors to saturate, with the signals fused in

the data processing stage. The experimental platform was originally designed as a wing-only model. It was used to characterize the airflow and load signals from the distributed array and the signals were then used to obtain estimates of the aerodynamic variables and loads in a low turbulence wind tunnel [21]. The platform was modified to be used in a larger 2.13 m × 1.52 m (7 ft × 5 ft) wind tunnel, which allowed a larger α range (at least ±20°). The wind tunnel model was fitted with three servo motors to drive its motion: one to drive the elevator, one to drive the flap and one to drive the aileron.

Table 1 Wing model experimental platform sensor technical specifications.

Feature	Sensors			
	Static pressure	Static pressure	Vertical bending	Torsion
Model	SDP36	MPXV7002	SGD-2/350-DY13	SGD-2/350-RY53
Manufacturer	Sensirion	Freescale Semiconductor	Omega Engineering	Omega Engineering
Measurement range	±500 Pa	±2 kPa	±30 000 µm/m	±30 000 µm/m

B. 1-DOF Pitch Rig

The wind tunnel model was mounted on a 1 Degree-Of-Freedom (DOF) pitch motion wind tunnel rig, depicted in Figure 2. A custom built pitch mechanism provided the rig with two motion configurations: one with the model free-to-pitch, where the aircraft was free to pitch; and a forced-pitch configuration, where the motion was driven by a servo motor (LXM32MD30M2, Schneider Electric, Rueil-Malmaison, France). The aerodynamic loads were measured using a load cell (Mini 45, ATI, Apex, North Carolina, USA) mounted at the interface between the wing support and the free-to-pitch mechanism. Note that in the free-to-pitch configuration pitching moment measurements were not available. α was obtained via the autopilot IMU (see Figure 1c). In the forced-pitch configuration, a magnetic rotary encoder (RMB20SC, Renishaw, Wotton-under-Edge, Gloucestershire, England), provided additional α measurements. The rotary encoder was connected to a timing pulley-belt array, this array was connected to the rig's shaft which was used to transmit the model's motion. The rig's shaft was connected to the servo motor through a elastomer insert coupling (EKH-60-B-24-28, R+W, Klingenberg, Germany).

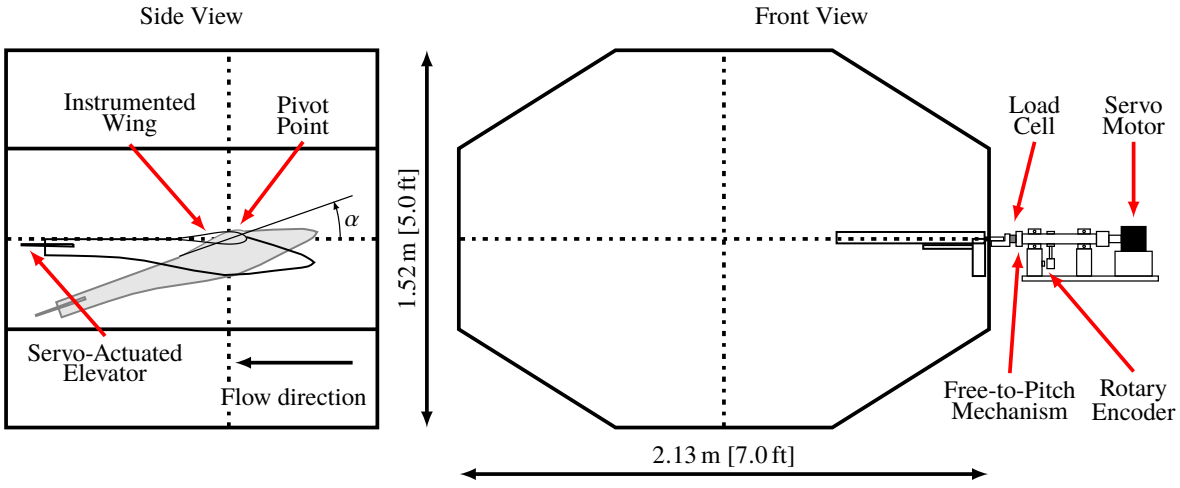

Figure 2 Experimental platform block diagram.

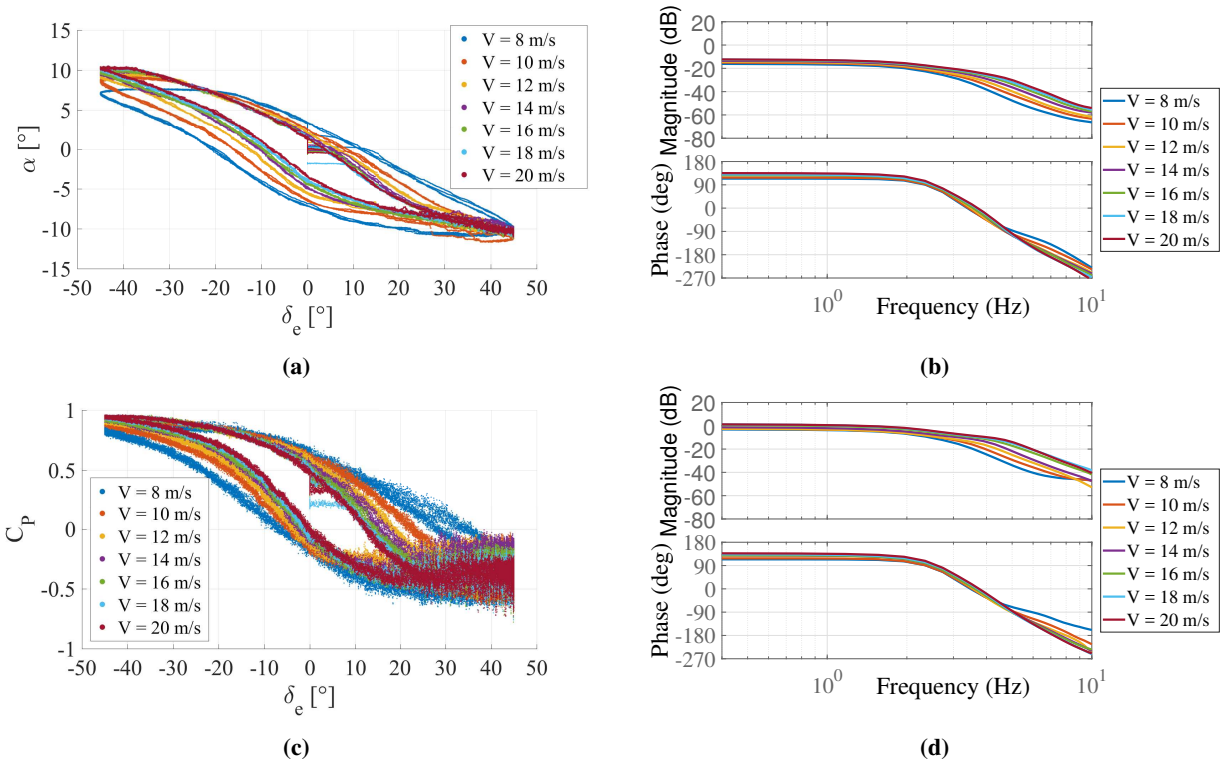

Figure 3 Response to δ_e inputs: (a) α response to slow-ramp, (b) α response to frequency sweep, (c) $C_{P_{B01}}$ response to slow-ramp and (d) $C_{P_{B01}}$ response to frequency sweep.

C. Data Acquisition System and Control Action Computation

A custom data acquisition (DAQ) system, including some proprietary hardware, was used to collect pressure, strain, aerodynamic loads and aerodynamic variables data. Figure 1c shows a block diagram of the DAQ system. The DAQ system was divided into five Micro-Controller Unit (MCU) based subsystems: the Section A subsystem acquired the signals from the pressure sensors in Section A insert and the SG-A and SG-B strain gauge arrays, it also received and stored data from the rig control electronics and controled the inboard servo motor; the Section B subsystem acquired the signals from the pressure sensors in Section B insert and the SG-B and SG-C strain gauge arrays and controlled the outboard servo motor; the central MCU subsystem received time stamped data acquired by Section A and Section B subsystems and transmitted these to a PC via USB communication. Additionally, the central MCU subsystem controlled the model's elevator. This subsystem could also be used to transmit commands generated by the PC to the other subsystems. The load cell subsystem measured, acquired, conditioned and transmitted load signals to a PC via USB using a multi-function I/O device (USB-6218, NI, Austin, Texas, USA); Lastly the PC collected and stored pressure, strain, aerodynamic loads and aerodynamic variables data, and then used these to compute an appropriate control action.

All acquired data was sampled at 200 Hz by the appropriate subsystem and received by the PC. It was then routed using User Datagram Protocol (UDP) messages and stored on disk by a custom graphical user interface (GUI) written in Python. A Simple Network Time Protocol (SNTP) was used to synchronize the clock signal of the MCU subsystems with the PC clock signal. The SNTP implementation achieves a maximum time offset of less than 1.0 ms between clock signals. The closed loop control action was computed using MATLAB Version 9.2 and Simulink Version 8.9, Release 2017a (The MathWorks, Inc., Natick, Massachusetts, United States), Results from a series of characterization experiments for flight control design are presented in the next section.

III. System Characterization and Flight Control Design

Using the wind tunnel model and testing rig described in Section II, a series of characterization experiments were carried out in the University of Bristol 2.13 m × 1.52 m (7 ft × 5 ft) low speed wind tunnel. These experiments were

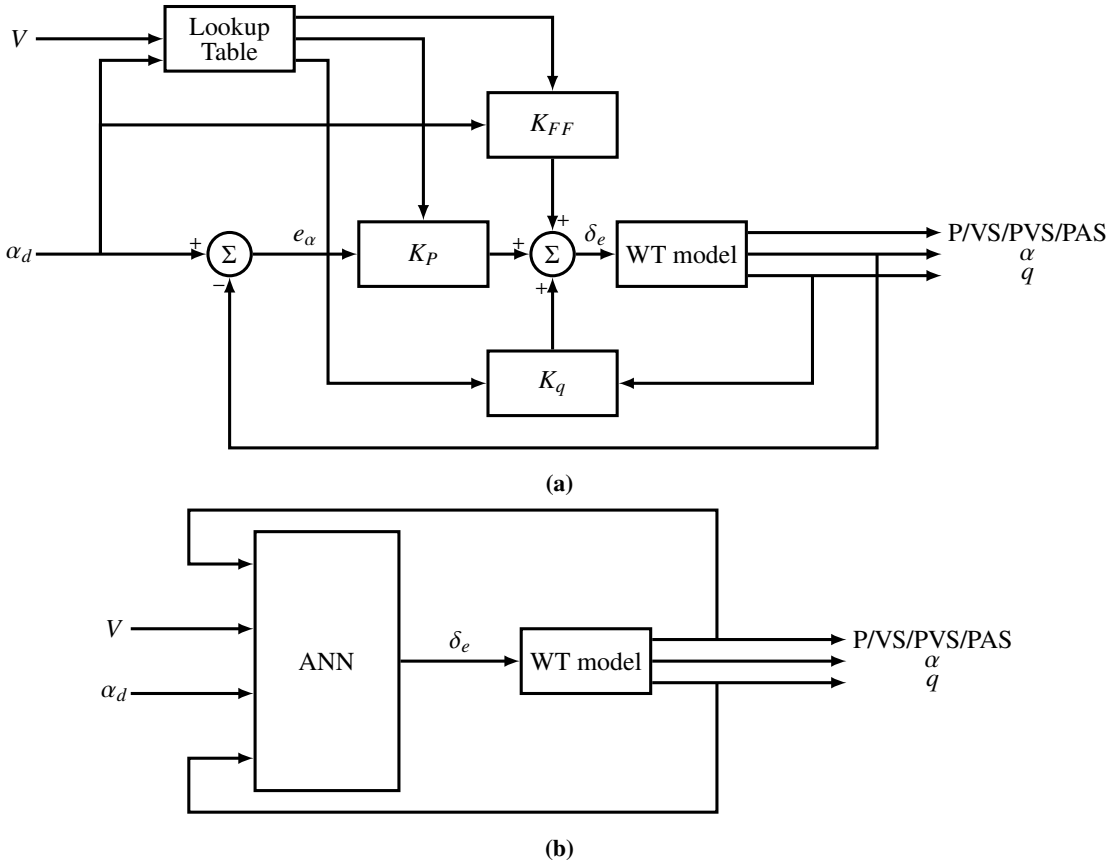

Figure 4 General α controller structure: (a) conventional gain scheduling and (b) ANN-based.

carried out to characterize the response of dynamic state, aerodynamic loads, pressure and strain signals to δ_e commands. These tests consisted of quasi-static as well as dynamic tests performed at air speeds $V = [8, 10, 12, 14, 16, 18, 20]$ m/s. For the quasi-static tests, slow δ_e ramps were used to drive α sweeps. The dynamic tests where carried out using δ_e frequency sweeps ranging from 0.1 Hz to 10.0 Hz.

Figures 3a and 3c show α and the response of one pressure sensor $C_{P_{B01}}$ to δ_e slow-ramp inputs at various V, respectively. Both α and $C_{P_{B01}}$ exhibit a nonlinear response in the form of hysteretic behavior as function of the pitch motion direction. This behavior is likely to be an aero-servo elastic phenomenon, arising from the interaction between the servo and the elevator linkage as well as the flow interaction between the wing and the horizontal tail. This is likely translate in significant time lag in closed loop response to commanded motions. Another important feature is the observed reduction in δ_e control effectiveness for $V < 12$ m/s, i.e. higher control gains are required as V decreases.

Figures 3b and 3d show α and $C_{P_{B01}}$ response to δ_e frequency sweep inputs at various V, respectively. Both α and $C_{P_{B01}}$ exhibit similar magnitude response and practically the same phase response. As observed for the δ_e slow-ramp input case, both α and $C_{P_{B01}}$ display magnitude dependence on V, with the magnitude increasing as V increased. Based on this experiment the aircraft open loop cut-off frequency was found to be ≈ 1.2 Hz and its input-output time delay ≈ 0.34 s.

The results from the characterization experiments were used to design a conventional Gain Scheduling (GS) α controller. Figure 4a shows the structure of the conventional GS α controller. To form the GS controller, a series of linear controllers providing closed-loop stability around various equilibrium points were tuned, i.e. values for the feed-forward gain K_{FF}, the proportional gain K_P and the damping term K_q were obtained for various V and α_d. The resulting set of gains were collected to form a lookup table, which in turn was used to produce and interpolate the controller gains. The GS controller takes e_α, q as well as α_d as inputs and returns δ_e command, with the angle of attack tracking error defined as:

$$e_\alpha = \alpha_d - \alpha. \tag{1}$$

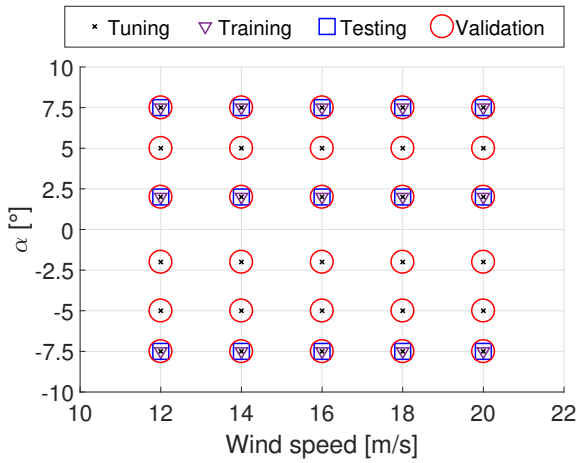

Figure 5 Controller tuning, training, testing and validation space.

Figure 5 shows the V and α points where the GS controller was tuned. Once the controller was tuned, a data set comprised of 10 step commands was collected at each point for training and testing of the ANN-based controller, with the distributed array signals recorded during these closed-loop control experiments.

Figure 4b shows the basic structure of the distributed sensing-based α controller. The controller is based on an ANN in which α_d, q, V as well as the signals from the distributed sensing array are fed and the control action δ_e is returned. Note that in this implementation α is not fed into the controller directly. A total of five different controllers were trained using this architecture. The basic input to each of the controllers is formed by α_d, q and V, then based on the amount and type of additional input information, each controller was defined as: Feed-Forward (FF), which only used the basic input; Pressure (P), using the basic input and all the pressure information; Vertical Strain (VS), using the basic input and the vertical strain information; Pressure and Vertical Strain (PVS), using the basic input, all the pressure and the vertical strain information; and lastly, Pressure and All Strain (PAS), using the basic input, all the pressure and the all strain information.

All the trained ANNs consisted of three main layers: the input, hidden and output layers. The input layer handled the inputs to the ANN, it conditioned the input signals to improve the likelihood of successfully training an ANN by scaling them using the mean value and range of the expected inputs. The number of neurons in the input layer depended on the number of inputs for each controller class, with 3 for FF, 33 for P, 7 for VS, 37 for PVS and 40 for PAS. The hidden layer consisted of interconnected artificial neurons. The neurons in the hidden layers used hyperbolic tangent sigmoid functions as activation functions. A total of 9 neurons were used in the hidden layer for each controller. Lastly, the output layer took the signals produced by the hidden layer and scaled it to match the magnitude of the target function. The Mean-Square-Error between the δ_e output from the GS controller and the δ_e output from the ANN-based controller was used as the target function to be minimized. The ANNs were trained using MATLAB Version 9.2 and Neural Network Toolbox Version 10.0, Release 2017a (The MathWorks, Inc., Natick, Massachusetts, United States), employing the Levenberg-Marquardt back-propagation algorithm as the training function.

Each ANN-based controller was trained using the tuned GS controller output as the target function. Figure 5 shows the V and α points that were used for training and testing of the ANN-based controller. A subset of the tuned GS data set at each test condition was employed to trained the ANN-based controller offline, using 70% of the data (7 step commands). The remaining 30% of the data (3 step commands) was used for offline testing of the trained ANN-based controller. The resulting controllers are non-linear controllers providing closed-loop stability throughout a continuum of operating points.

In the next section, representative experiments demonstrating the performance of the controllers presented above are presented.

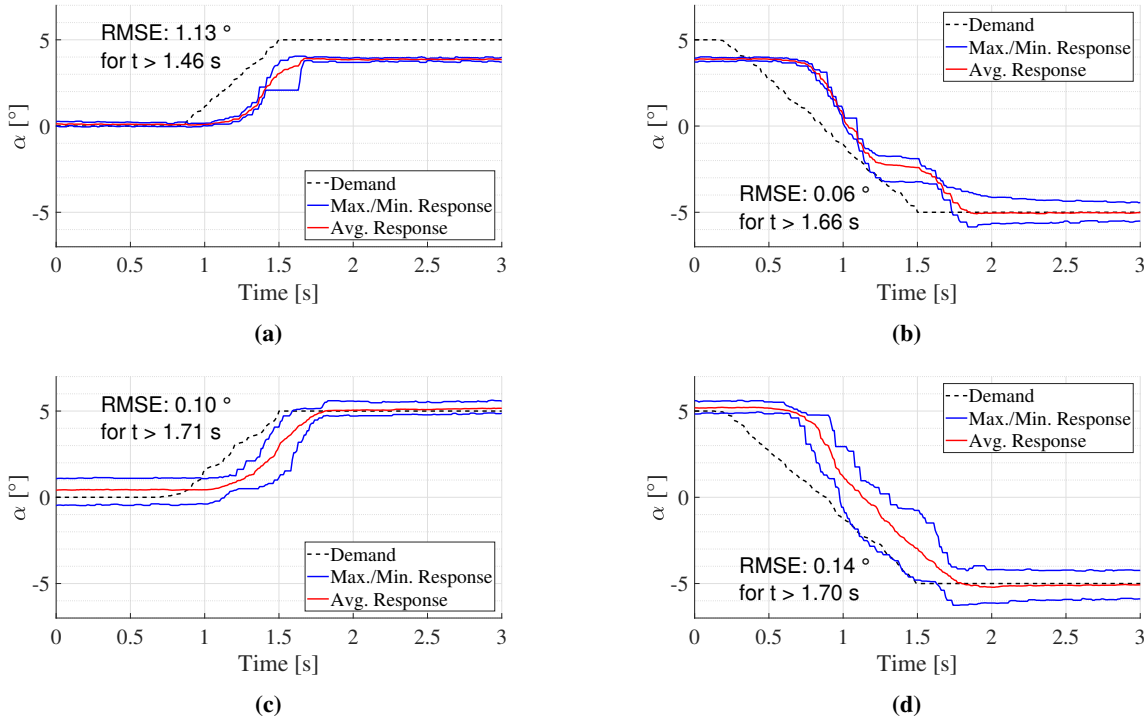

Figure 6 Controller response for validation: (a) GS positive α_d, (b) GS negative α_d, (c) VS positive α_d and (d) VS negative α_d.

IV. Experimental Results

A series of validation experiments, consisting of α doublets, were carried out in the 2.13 m × 1.52 m (7 ft × 5 ft) low speed wind tunnel to study the performance of the controllers. Figure 5 shows the V and α points where closed loop validation tests were performed. Each tests consisted of α doublets as commands at each point in the validation space.

Figures 6a and 6b, show the GS controller average response for positive and negative α_d with $V = 14$ m/s. Only the negative demand case reaches α_d, with the positive demand case displaying a steady state error of $\approx 1.0°$. Note that data for this V and α combination was not part of the training set for the ANN-based controllers. Figures 6c and 6d, show the VS controller average response for positive and negative α_d with $V = 14$ m/s. Both cases display small Root-Mean-Square-Error (RMSE) values (less than 0.15°) and a rise time similar to that of the GS controller. For this particular example, the error band for the positive α demand case is narrower when compared to the negative case, with $\approx 1.0°$ against $\approx 2.0°$. Similar response characteristics where observed for all the other distributed sensing-based controllers.

From tests like the ones shown in Figure 6 the α tracking error was computed, but only for times where α_d was constant. This process was performed for all points in the validation set space. The resulting data was used to compute step response characteristics for every controller, i.e. the rise time t_{rs}, the α RMSE, as well as the tracking error minimum, median, maximum, first quartile Q_1 and third quartile Q_3 values.

Figure 7 shows the α tracking error characteristics for the PVS controller. Panel (a) shows a box-plot of the tracking error across the α demand range. The plot shows the minimum, median, Q_1 and Q_3, as well as maximum tracking error values, for each α demand value. It can be observed that the average absolute median tracking error remains below 0.50° for $-5.0° \leq \alpha \leq 5.0°$. However, the absolute median α error for $\alpha_d = 7.5°$ is greater than that of the $\alpha_d = -7.5°$ case, with 1.56° and −0.58°. This behavior was also present in the tracking error of the GS controller used for training, as a consequence the performance of all the ANN-based controllers for $\alpha_d = \pm 7.5°$ was degraded. One way to improve the ANN-based controllers performance would be with better training data, another alternative is to use a different machine learning technique such as reinforcement learning to minimize the tracking error. Panel (b) shows a box-plot of the tracking error across the V demand range. In general the controller performance is satisfactory across V with

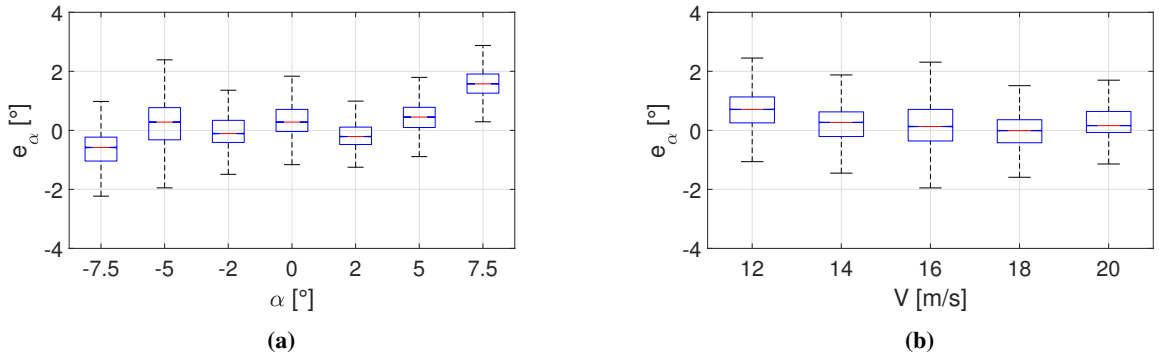

Figure 7 Pressure-Vertical Strain (PVS) controller tracking error characteristics for: (a) α demand range and (b) V range.

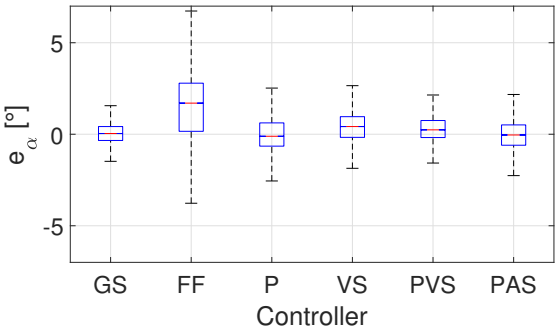

Figure 8 Controller α tracking error performance.

the average absolute median tracking error below 0.70°. It can also be observed that the median α error decreases as V increases, particularly for $V \geq 14\,\mathrm{m/s}$, with the absolute median tracking error below 0.20°. Additionally, the tracking error band decreases for $V \geq 18\,\mathrm{m/s}$, with the distance between the minimum and maximum values less than 3.20°. Similar α tracking error characteristics where observed for the other distributed sensing-based controllers.

Figure 8 shows a box-plot of the overall tracking error for all the controllers, including the GS and FF controllers. The GS controller performs the best among the controllers with a median tracking error of 0.04° (RMSE = 0.79°). It displays the narrowest error band ≈ 3.0°. The FF controller performs the worst among the controllers with a median tracking error of 1.70° (RMSE = 2.54°), it displays an error band greater than 10.0°. All of the distributed sensing-based controllers perform well with an average median tracking error of 0.13° (average RMSE = 0.96°). The error band boundary for these controllers is ≈ 4.5°. The controllers step response characteristics are summarized in Table 2. The rising times for the GS and distributed sensing-based controllers are very similar, with an overall $t_{rs} = 0.77\,\mathrm{s}$ for the GS controller against $t_{rs} = 0.75\,\mathrm{s}$ average for the distributed sensing-based controllers. The FF controller displays a slower response with $t_{rs} = 1.10\,\mathrm{s}$.

V. Discussion

In this paper, 1-DOF closed loop control wind tunnel experiments for α regulation using a conventional gain scheduling and various distributed sensing-based controllers were presented. The conventional controller was tuned experimentally using signals commonly available in traditional sensor suites. The response from the tuned conventional gain scheduling controller was then used to train various ANN-based controller using different combinations of the signals from a distributed array of sensors measuring the loads (strain) and flow distribution (pressure) acting over the wing surface of an instrumented wind tunnel aircraft model.

A series of validation experiments consisting of α steps were carried out to assess the performance of the ANN-based controllers. In general, the performance of the controllers using distributed array signals was very similar to the

Table 2 Controllers step response characteristics.

Variable	Controller architecture					
	GS	FF	P	VS	PVS	PAS
Rising time (t_{rs}) [s]	0.77	1.10	0.73	0.76	0.76	0.76
Median e_α[°]	0.04	1.70	−0.11	0.42	0.24	−0.040
Maximum e_α[°]	1.56	6.73	2.52	2.65	2.15	2.17
Minimum e_α[°]	−1.48	−3.77	−2.55	−1.86	−1.58	−2.26
e_α Q_1 [°]	−0.34	0.16	−0.65	−0.17	−0.18	−0.60
e_α Q_3 [°]	0.42	2.79	0.62	0.96	0.75	0.51
RMSE [°]	0.79	2.54	0.97	0.99	1.02	0.88

conventional controller, with an average t_{rs} of 0.77 s for the conventional controller against 0.75 s for the combined average of the distributed sensing-based controllers. It was found that the worst performing controller was the FF controller with a α RMSE of 2.54°, against 0.79° and 0.96° for the conventional and the distributed sensing-based controllers, respectively. Additionally, the tracking error statistics indicate that the VS controller performance is comparable to that of the P, PVS or PAS controllers, indicating that the strain sensor signals are similarly informative in relation α as the pressure sensor signals, at least over the range tested.

These results suggest that pressure and strain sensing can be used in combination with ANN-based controllers to improve flight control performance, by making use of flow state information embedded in the sensing array. Moreover, α is normally only available through an additional specific sensor for conventional control, while here α has been estimated as part of the ANN-based controller. In contrast with previous research [26], the approach presented here uses all the information from the distributed array, increasing robustness to individual sensor failure and/or useful measurement range.

The approach to learn the control action presented in this paper is known as supervised learning in the machine learning community, with this particular approach known as End-to-End learning ([30]). It produces a compact controller with improved performance by directly learning the mappings between control inputs and system outputs, as shown in Figure 9. This control strategy has the potential to reduce the overall control error as there is no estimation error introduced in the control loop. When using the conventional controller, gain scheduling needs to be implemented in order to provide control over a range of different flight speeds, this requires the design of several controllers at various operating points as well as careful selection of scheduling functions. This design process is simplified when using the distributed sensing-based controller, as a the output from a few tuned conventional controllers can be used to train it. The results presented here suggest that this controller is capable of handling changes in parameters without compromising the overall performance. A potential limitation from this approach is that the controller learns to minimize error between δ_e returned by conventional controller and its own δ_e output, which might not produce the best α controller, as strictly speaking the α tracking error is not minimized. A possible solution is to use other machine learning techniques such as persistent self-supervised learning, reinforcement learning or other non-linear control strategies whose main objective is to minimize the α tracking error.

In order to develop this approach further there are a number of areas of potential development. A factor that has not been considered so far in the design of the ANN-based controller is the effect that other control surface deflections have on its behavior. For the system to provide robust control, it must be capable of adapting to changes in other control inputs. One way of achieving this would be for an ANN to learn the correlation between relevant control inputs and the expected system response and to then change its output accordingly. This ANN would only then provide an appropriate control signal. It has been suggested that a similar adaptive filter approach is part of the function of the cerebellum in the mammalian nervous system [31], with other animal groups having equivalent systems involved with their motor control. Another factor that has not been addressed is that of in-flight tuning/training. For this control approach to be successfully used in a real-world environment, a strategy that takes advantage of the behavior learned under controlled conditions needs to be developed. A possible solution is to use persistent self-supervised learning [32], where the system uses previously learned input-output mappings to recognize additional, complementary information.

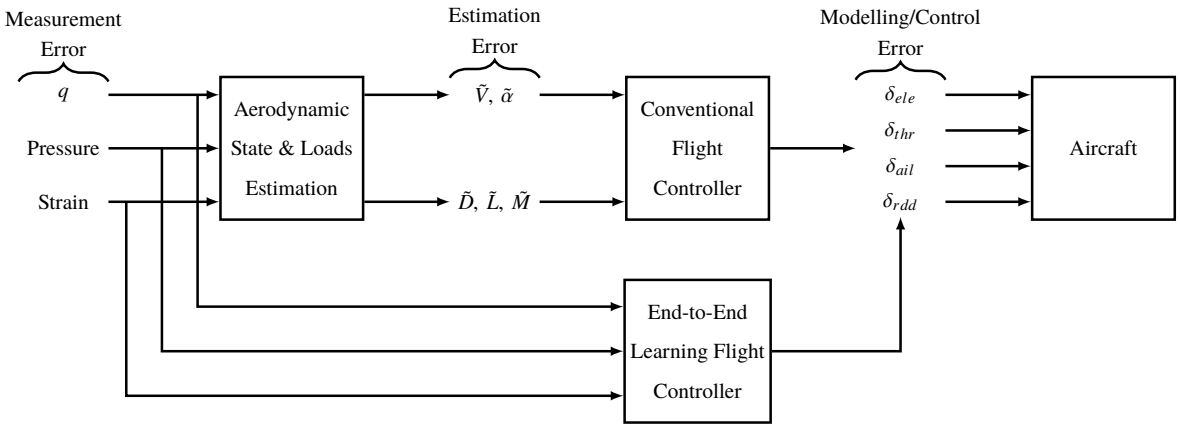

Figure 9 Conventional and End-to-End flight controller strategies.

VI. Concluding Remarks

Wind tunnel testing showed that using an ANN-based controller in combination with signals from a distributed array of pressure and strain sensors on a wing, it is possible to control α without directly feeding it to the controller. The use of the distributed sensing-based controller allowed α control, with performance similar to a conventional controller over a wide range of α and V values. It was found that the distributed array information is essential to provide accurate control and a narrow α tracking error band.

Overall, these results indicate that bio-inspired sensors, sensory processing and neural inspired control structures offer an alternative way to thinking about aircraft flight control. Future flight control applications based on distributed sensing could include control systems that are trained to directly minimize the tracking error and/or make use of multiple control surfaces, allowing for improved control and increase maneuverability. This could be particularly useful for unmanned aerial vehicles with high degrees of freedom such as highly flexible or morphing wings, where the airflow and loads vary dynamically across the wing depending on the configuration of the wing as well as the current aerodynamic conditions.

Acknowledgments

The authors would like to thank Mr Lee Winter from the University of Bristol Wind Tunnel Laboratory, for his invaluable support and work during the assembly of the pressure ports in the pressure sensing platform used to carry out the experiments presented in this paper.

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 679355).

References

[1] Mohamed, A., Clothier, R., Watkins, S., Sabatini, R., and Abdulrahim, M., "Fixed-wing MAV attitude stability in atmospheric turbulence Part 1: Suitability of conventional sensors," *Progress in Aerospace Sciences*, Vol. 70, 2014, pp. 69–82. DOI: 10.1016/j.paerosci.2014.06.001.

[2] Taylor, G. K., Carruthers, A. C., Hubel, T. Y., and Walker, S. M., "Wing morphing in insects, birds and bats: mechanism and function," *Morphing Aerospace vehicles and structures*, 2012, pp. 11–40.

[3] Watkins, S., Milbank, J., Loxton, B. J., and Melbourne, W. H., "Atmospheric winds and their implications for microair vehicles," *AIAA Journal*, Vol. 44, No. 11, 2006, pp. 2591–2600.

[4] Young, J., Walker, S. M., Bomphrey, R. J., Taylor, G. K., and Thomas, A. L., "Details of insect wing design and deformation enhance aerodynamic function and flight efficiency," *Science*, Vol. 325, No. 5947, 2009, pp. 1549–1552.

[5] Muijres, F., Johansson, L. C., Barfield, R., Wolf, M., Spedding, G., and Hedenström, A., "Leading-edge vortex improves lift in slow-flying bats," *Science*, Vol. 319, No. 5867, 2008, pp. 1250–1253.

[6] Shepard, E. L., Williamson, C., and Windsor, S. P., "Fine-scale flight strategies of gulls in urban airflows indicate risk and reward in city living," *Philosophical Transactions of the Royal Society B*, Vol. 371, No. 1704, 2016, p. 20150394.

[7] Mallon, J. M., Bildstein, K. L., and Katzner, T. E., "In-flight turbulence benefits soaring birds," *The Auk*, Vol. 133, No. 1, 2015, pp. 79–85.

[8] Taylor, G. K., and Krapp, H. G., "Sensory Systems and Flight Stability: What do Insects Measure and Why?" *Insect Mechanics and Control*, Advances in Insect Physiology, Vol. 34, edited by J. Casas and S. Simpson, Academic Press, 2007, pp. 231–316.

[9] Hörster, W., "Histological and electrophysiological investigations on the vibration-sensitive receptors (Herbst corpuscles) in the wing of the pigeon (Columba livia)," *Journal of Comparative Physiology A*, Vol. 166, No. 5, 1990, pp. 663–673.

[10] Brown, R. E., and Fedde, M. R., "Air-flow sensors in the avian wing," *Journal of Experimental Biology*, Vol. 179, 1993, pp. 13–30.

[11] Altshuler, D. L., Bahlman, J. W., Dakin, R., Gaede, A. H., Goller, B., Lentink, D., Segre, P. S., and Skandalis, D. A., "The biophysics of bird flight: functional relationships integrate aerodynamics, morphology, kinematics, muscles, and sensors 1," *Canadian Journal of Zoology*, Vol. 93, No. 12, 2015, pp. 961–975.

[12] Sterbing-D'Angelo, S., Chadha, M., Chiu, C., Falk, B., Xian, W., Barcelo, J., Zook, J. M., and Moss, C. F., "Bat wing sensors support flight control," *Proceedings of the National Academy of Sciences*, Vol. 108, No. 27, 2011, pp. 11291–11296.

[13] Whitmore, S., Cobleigh, B., and Haering, E., Jr, "Design and calibration of the X-33 flush airdata sensing (FADS) system," *36th AIAA Aerospace Sciences Meeting and Exhibit*, 1998, p. 201.

[14] Pruett, C., Wolf, H., Heck, M., and SIEMERS III, P., "Innovative air data system for the space shuttle orbiter," *Journal of Spacecraft and Rockets*, Vol. 20, No. 1, 1983, pp. 61–69.

[15] Quindlen, J., and Langelaan, J., "Flush Air Data Sensing for Soaring-Capable UAVs," *51st AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, AIAA, Grapevine (Dallas/Ft. Worth Region), Texas, 2013.

[16] Samy, I., Postlethwaite, I., Gu, D.-W., and Green, J., "Neural-Network-Based Flush Air Data Sensing System Demonstrated on a Mini Air Vehicle," *Journal of Aircraft*, Vol. 47, No. 1, 2010, pp. 18–31.

[17] Mark, A., Xu, Y., and Dickinson, B. T., "Review of Microscale Flow-Sensor-Enabled Mechanosensing in Small Unmanned Aerial Vehicles," *Journal of Aircraft*, 2019, pp. 1–12.

[18] Shen, H., Xu, Y., and Remeikas, C., "Pitch Control of a Micro Air Vehicle with Micropressure Sensors," *Journal of Aircraft*, Vol. 50, No. 1, 2013, pp. 239–248. 10.2514/1.C031894.

[19] Fei, H., Zhu, R., Zhou, Z., and Wang, J., "Aircraft flight parameter detection based on a neural network using multiple hot-film flow speed sensors," *Smart Materials and Structures*, Vol. 16, No. 4, 2007, p. 1239.

[20] Magar, K. T., Reich, G. W., Kondash, C., Slinker, K., Pankonien, A. M., Baur, J. W., and Smyers, B., "Aerodynamic parameters from distributed heterogeneous CNT hair sensors with a feedforward neural network," *Bioinspiration & biomimetics*, Vol. 11, No. 6, 2016, p. 066006.

[21] Araujo-Estrada, S. A., and Windsor, S. P., "Aerodynamic Variables and Loads Estimation Using Bio-Inspired Distributed Sensing," *AIAA Scitech 2019 Forum*, American Institute of Aeronautics and Astronautics, San Diego, CA, 2019.

[22] Yeo, D., Atkins, E. M., Bernal, L. P., and Shyy, W., "Experimental Validation of an Aerodynamic Sensing Scheme for Post-Stall Aerodynamic Moment Characterization," *AIAA Atmospheric Flight Mechanics (AFM) Conference*, AIAA, Boston, Massachusetts, 2013.

[23] Thompson, R. A., Evers, J., and Stewart, K., "Attitude control augmentation using wing load sensing-A biologically motivated strategy," *Atmospheric Flight Mechanics Conference*, AIAA, Toronto, Canada, 2010.

[24] Castano, L., Airoldi, S., McKenna, T., and Humbert, J. S., "Gust rejection using force adaptive feedback for roll," *14th AIAA Aviation Technology, Integration, and Operations Conference*, AIAA, Atlanta, Georgia, 2014.

[25] Araujo-Estrada, S. A., Salama, F., Greatwood, C. M., Wood, K. T., Richardson, T. S., and Windsor, S. P., "Bio-Inspired Distributed Strain and Airflow Sensing For Small Unmanned Air Vehicle Flight Control," *AIAA Guidance, Navigation, and Control Conference*, 2017, p. 1487.

[26] Wood, K. T., Araujo-Estrada, S., Richardson, T., and Windsor, S., "Distributed Pressure Sensing–Based Flight Control for Small Fixed-Wing Unmanned Aerial Systems," *Journal of Aircraft*, 2019, pp. 1–10.

[27] Yeo, D. W., Rehm, N., Bradley, J. M., and Chopra, I., "A Cyber-Physical Approach to Stall Avoidance and Turbulence Mitigation for Fixed-Wing Unmanned Aerial Systems," *AIAA Scitech 2019 Forum*, American Institute of Aeronautics and Astronautics, San Diego, CA, 2019.

[28] Ferrari, S., and Stengel, R. F., "Classical/Neural Synthesis of Nonlinear Control Systems," *Journal of Guidance, Control, and Dynamics*, Vol. 25, No. 3, 2002, pp. 442–448.

[29] Coates, A., Abbeel, P., and Ng, A. Y., "Apprenticeship Learning for Helicopter Control," *Communications of the ACM*, Vol. 52, No. 7, 2009, pp. 97–105.

[30] Bojarski, M., Testa, D. D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D., Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., and Zieba, K., "End to End Learning for Self-Driving Cars," *CoRR*, Vol. abs/1604.07316, 2016.

[31] Dean, P., Porrill, J., Ekerot, C.-F., and Jorntell, H., "The cerebellar microcircuit as an adaptive filter: experimental and computational evidence," *Nature Reviews Neuroscience*, Vol. 11, No. 1, 2010, pp. 30–43.

[32] van Hecke, K., de Croon, G., van der Maaten, L., Hennes, D., and Izzo, D., "Persistent self-supervised learning: From stereo to monocular vision for obstacle avoidance," *International Journal of Micro Air Vehicles*, 2018.