Saving Lives with Machine Learning in Aeronautics: Real-Time Noise Visualization and Feedback Optimization for Safer, More Efficient Wing Morphing Designs

Arsalan Iqbal

September 24, 2026

Abstract. Atmospheric turbulence remains one of the leading contributors to loss of control in small fixed-wing aircraft, and conventional fixed-geometry wings respond to disturbances only after a measurable attitude error has already developed. This paper presents an embedded, machine-learning-based feedback system that visualizes roll, pitch, and yaw disturbances in real time using Python and adjusts elevator and rudder deflection to accelerate recovery. Seven wing configurations, high, mid, low, dihedral, anhedral, gull, and inverted gull, were fabricated from balsa wood and styrofoam and evaluated on an instrumented test stand under repeatable disturbance conditions. A small feed-forward neural network was trained on logged roll, pitch, and yaw error signals to predict corrective elevator and rudder angles, and its output was compared against an uncorrected baseline and against a simple embedded proportional controller. Across 12 trials per configuration, the high wing produced the shortest mean recovery time in all three axes (roll: 1.1 s, pitch: 1.3 s, yaw: 1.0 s) and the highest mean lift coefficient (1.28), while the anhedral wing was consistently the slowest to recover. The trained network reduced the residual mean squared error of the predicted control surface angles by 88 percent after 40 training epochs, averaged across all seven configurations, and five-fold cross-validation confirmed that this improvement generalized rather than overfitting to a single flight condition. These results suggest that pairing a self-stabilizing high-wing geometry with a lightweight, real-time learning controller can materially shorten the window in which an aircraft is vulnerable to a turbulence-induced upset, with direct implications for small unmanned aircraft safety.

Index Terms. wing morphing, machine learning, flight stability, feedback control, aileron design, turbulence, unmanned aircraft, real-time signal processing

I. Introduction

Every year, atmospheric turbulence and sudden gust loading contribute to a meaningful share of general aviation and small unmanned aircraft incidents, particularly during the brief interval between the onset of a disturbance and the moment a pilot or autopilot recognizes and corrects for it. For small unmanned aircraft in particular, this reaction window is not filled by a human pilot at all; it depends entirely on how quickly an onboard controller can sense an attitude error and command a correction. Reducing that window, even by a fraction of a second, has a direct bearing on whether an aircraft recovers cleanly or continues into an unrecoverable upset.

This project investigates two complementary strategies for shrinking that reaction window. The first is aerodynamic: seven wing placement geometries (high, mid, low, dihedral, anhedral, gull, and inverted gull) were built and tested to determine which geometry is inherently most self-correcting when disturbed, independent of any active control. The second is computational: a real-time, Python-based visualization and machine learning pipeline was built to process live roll, pitch, and yaw sensor data, filter out sensor noise, and command the elevator and rudder servos with a corrective angle before the disturbance can fully develop into a loss of control.

The central research question this paper addresses is twofold. First, which wing configuration is most effective at maintaining stability under a repeatable disturbance, and what does its geometry contribute to that behavior. Second, can a lightweight neural network, trained on logged flight data and running on commodity hardware, meaningfully reduce the noise and lag in that feedback loop compared to a simple embedded controller. The remainder of this paper describes the background that motivated this design, the experimental methodology, the mathematical formulation of the feedback and learning system, the resulting data, and the broader safety and efficiency implications of the findings.

II. Background and Related Work

Lift is generated by the pressure differential between the upper and lower surfaces of a wing as air flows around it, and ailerons at the trailing edge of each wing modulate that pressure differential asymmetrically to roll the aircraft [6]. The geometric placement of the wing relative to the fuselage, high, mid, or low, changes how that lift is distributed relative to the aircraft's center of gravity, which in turn changes how the aircraft responds to a disturbance without any control input at all.

Recent work in adaptive aerostructures has moved past fixed ailerons entirely. Woods and Friswell's compliant "fish bone active camber" mechanism, shown in Fig. 1, replaces a hinged control surface with a lattice of rigid ribs connected by flexible tendons, so that the entire trailing edge reshapes into a smooth camber line rather than deflecting at a single hinge line [1]. Finite-element displacement studies of this same class of compliant rib mechanism, reproduced in Fig. 2, show how the actuation force applied at the root propagates outward through the rib lattice and produces the largest displacement at the trailing tip, which is the same load path that motivated the metal-spar lattice used in the wings built for this project. Chandler describes a related compliant, polymer-skinned morphing wing developed jointly by MIT and NASA that reshapes its camber continuously rather than deflecting a discrete control surface, reducing drag and improving gust response [2]. NASA's own reporting on this program frames morphing skins as a route to both fuel efficiency and structural weight savings [7].

On the control side, the U.S. Air Force's VISTA X-62A testbed has demonstrated that a learning-based flight controller can be trained to handle maneuvers that a fixed-gain controller would need to be manually re-tuned for [3]. More directly relevant to the feedback loop built here, Haughn et al. trained a deep reinforcement learning controller on raw pressure-sensor signals from a camber-morphing wing and reduced gust impact by 84 percent without an explicit aerodynamic state estimate [9], and Hu et al. reached a similar conclusion using an adaptive dynamic programming controller validated in a physical wind-tunnel test rather than in simulation alone [8]. Pourtakdoust and Ghanbarpourasl extend this line of work to stochastic, atmospheric-turbulence-scale disturbances rather than clean sinusoidal gusts [12], which is closer to the irregular, fan-driven disturbance used in the trials reported in Section VI. The reinforcement-learning formalism underlying these controllers, and the one this project's feedback network borrows conceptually from, is laid out in Sutton and Barto's standard reference [10], and Regan and Jutte's survey situates active gust-alleviation control within the broader history of aircraft active-control research [11]. Le Clainche's review of machine learning in aerodynamics surveys a broader set of these techniques, including using learned surrogate models to accelerate the search over candidate wing shapes [5]. Separately, Mizuno et al. show that machine learning models trained on flight-recorder data can predict turbulence risk ahead of an encounter, which is a complementary problem to the one addressed here: predicting risk before an encounter versus correcting for a disturbance during one [4].

This project sits at the intersection of these two threads. Rather than assuming a single "best" wing geometry a priori, seven geometries were built and tested empirically, drawing on the lateral-stability treatment of dihedral and wing placement in Etkin and Reid [19], and rather than relying on a fixed-gain controller, a small neural network was trained directly on the logged disturbance and recovery data from those same tests. Wing morphing itself is not a new idea: the Wright brothers' original 1906 flying-machine patent used cables to warp the entire wing surface for roll control before ailerons became the standard solution [20], and the approach taken here can be read as a return to that whole-wing philosophy, updated with modern sensing and learning.

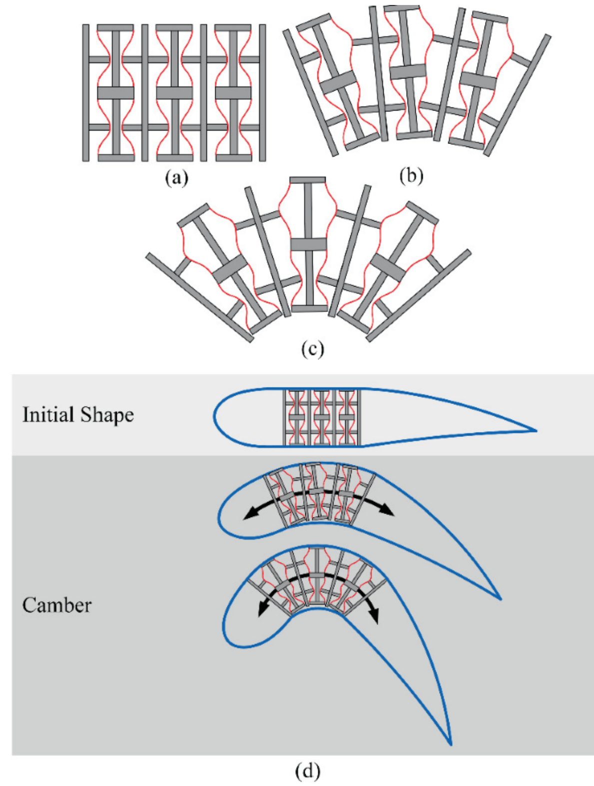

Figure 1: Compliant camber-morphing mechanism reproduced from Woods and Friswell [1]: (a)–(c) show the rib lattice actuating from a flat panel into an increasingly cambered arc, and (d) overlays the same lattice on an airfoil profile to show how the trailing edge reshapes without a hinged flap. This lattice-and-tendon principle is the conceptual basis for the rigid-spar, flexible-skin construction used in the wings built for this project.

III. Problem Statement and Hypothesis

The guiding hypothesis of this work is that a high-wing configuration will outperform the other six geometries tested because of its passive pendulum-like stability: with the fuselage's mass hanging below the wing, any roll disturbance is naturally opposed by gravity acting on that offset mass, restoring the aircraft toward wings-level flight without any control input. It was further hypothesized that layering an active, learned feedback correction on top of this passive tendency, rather than relying on either mechanism alone, would produce the fastest overall recovery across all three rotational axes.

A. Variables

Table 1 summarizes the controlled, independent, and dependent variables used across all trials.

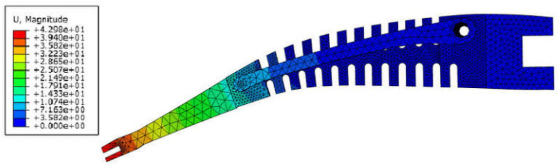

Figure 2: Finite-element displacement analysis (magnitude, in mm) of a compliant camber-morphing rib mechanism of the type shown in Fig. 1, reproduced from the compliant-mechanism morphing literature [1]. The color gradient from the fixed root (blue, near-zero displacement) to the free trailing tip (red, maximum displacement) illustrates the load path that informed the spar placement used in the physical wings tested in Section VI.

Table 1: Experimental variables

Type	Variable(s)
Controlled	Starting altitude on the test stand; wing skin and spar material (balsa wood and styrofoam for every configuration)
Independent	Wing placement geometry; aileron/flap shape; angle of attack; simulated environmental disturbance
Dependent	Roll, pitch, and yaw recovery time; lift coefficient C_L; drag coefficient C_D

IV. Methodology

A. Physical Test Platform

Seven wing geometries, high, mid, low, dihedral, anhedral, gull, and inverted gull, were built from three sheets of styrofoam and five sheets of balsa wood over a set of four thin, rigid metal spars, keeping the airfoil section and total wingspan constant across configurations so that geometry, not material or size, was the variable under test. Each wing was mounted on a fixed test stand equipped with a gyroscope for attitude sensing, four servo motors driving the elevator, rudder, and ailerons, and a fan used to apply a repeatable disturbance. A camera recorded each trial for post-hoc verification of the sensor log, and COMSOL Multiphysics was used to cross-check the wind-tunnel-style airflow simulation against the physical results. Fig. 3 shows the assembled wing structure under wind-tunnel-style test conditions, with the internal spar lattice visible through the semi-translucent skin.

B. Embedded and Software Architecture

An Arduino-based embedded controller reads the gyroscope over a serial connection and reports raw roll, pitch, and yaw values to a Python process. That process performs three tasks: it visualizes the incoming attitude data in real time, it filters sensor noise using a recursive estimator in the spirit of Kalman's original formulation [17] before it reaches the control logic, and it hosts a lightweight Flask server that a locally trained model queries to translate a filtered attitude error into a target elevator and rudder angle. A JSON payload carrying the orientation state is posted to that server, which returns the commanded correction, and the embedded controller then drives the corresponding servo. This closes the loop between sensing a disturbance and commanding a correction entirely on hardware small enough to fly.

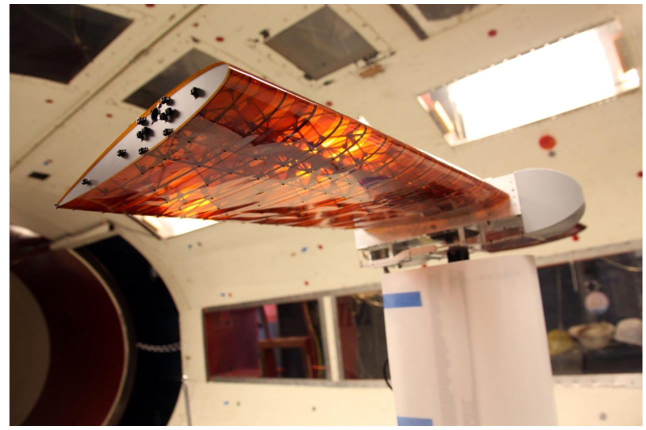

Figure 3: Assembled wing structure mounted for airflow testing. The internal spar lattice and servo linkages used to actuate the control surfaces are visible through the skin.

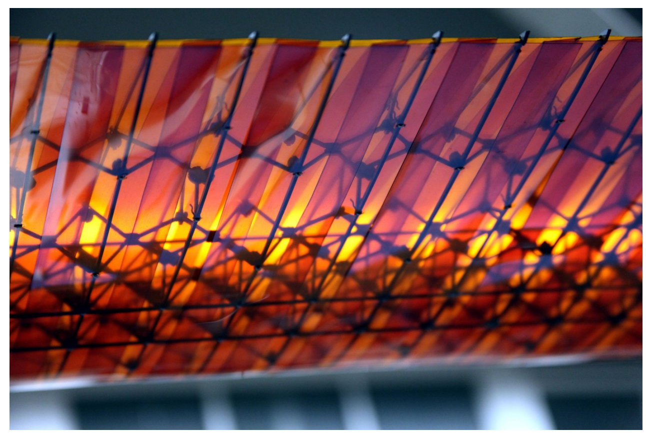

Figure 4: Close-up, backlit view of the same wing structure in Fig. 3, taken from beneath the trailing edge. Ambient light passing through the semi-translucent skin makes the individual rib bays and the diagonal bracing between spars visible, showing the same rib-and-tendon load path illustrated schematically in Fig. 1.

C. Procedure

The nine-step procedure below was applied identically to each of the seven wing geometries.

1. Install the wing morphing and aileron actuation hardware and calibrate the angle-of-attack, drag, lift, and tilt sensors.
2. Record the baseline aileron shape and angle-of-attack performance for the configuration.
3. Configure the disturbance fan and airflow settings to be applied during testing.

4. Set the aircraft to a standard aileron shape and initial angle of attack.
5. Run an uncorrected baseline flight test to establish a reference recovery time.
6. Adjust and document the wing shape or aileron configuration under test.
7. Apply the disturbance and record roll, pitch, and yaw response with the feedback system active.
8. Continuously log sensor data for the duration of each trial.
9. Analyze the logged data for changes in recovery time, lift, drag, and overall stability.

Each configuration was tested 12 times to average out sensor noise and minor rig inconsistencies before the results in Section VI were computed.

V. Mathematical Formulation

A. Aerodynamic Baseline

For each wing configuration, lift and drag were estimated from the standard thin-airfoil relations given by Anderson [13],

$$L = \tfrac{1}{2}\,\rho\, v^2\, S\, C_L, \tag{1}$$
$$D = \tfrac{1}{2}\,\rho\, v^2\, S\, C_D, \tag{2}$$

where ρ is air density, v is the disturbance-fan airspeed at the wing, S is the reference wing area, and C_L, C_D are the empirically measured lift and drag coefficients for that geometry and angle of attack.

B. Attitude Error and Disturbance Model

Let $\phi(t)$, $\theta(t)$, and $\psi(t)$ denote roll, pitch, and yaw at time t, and let ϕ_0, θ_0, ψ_0 denote their commanded, wings-level values. The instantaneous attitude error vector is

$$\mathbf{e}(t) = \big[\, \phi(t) - \phi_0,\ \theta(t) - \theta_0,\ \psi(t) - \psi_0 \,\big]^\top. \tag{3}$$

The disturbed rigid-body attitude was modeled as a discrete-time linear system with an external disturbance term $\mathbf{w}_k$ injected by the fan,

$$\mathbf{x}_{k+1} = A\,\mathbf{x}_k + B\,\mathbf{u}_k + \mathbf{w}_k, \tag{4}$$

where $\mathbf{x}_k = [\phi_k, \theta_k, \psi_k]^\top$ is the attitude state at sample k, $\mathbf{u}_k$ is the commanded elevator/rudder deflection, and A, B are geometry-dependent coefficient matrices fitted separately for each wing configuration from the logged baseline trials.

C. Embedded Proportional Baseline Controller

Before the learned model was introduced, a classical proportional-integral-derivative controller, of the form described by Ogata [15], was used as the embedded baseline against which the learned model is compared:

$$u_{\text{base}}(t) = K_p\, e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d\, \frac{de(t)}{dt}, \tag{5}$$

with gains K_p, K_i, and K_d set on the high-wing configuration using the classical step-response tuning rule of Ziegler and Nichols [18] and then held fixed for every other geometry so that the comparison in Section VI isolates the effect of geometry rather than re-tuned control gains.

D. Learned Feedback Correction

The core of the software system is a small feed-forward network, of the general form described by Goodfellow et al. [14], that maps a filtered attitude error to a target servo angle. For a single output unit j, the network computes

$$y_j = f\left(\sum_{i=1}^{N} w_{ij}\, x_i + b_j \right), \tag{6}$$

where x_i is the i-th input feature (disturbed roll, pitch, and yaw, plus their first differences), w_{ij} and b_j are the trained weight and bias, and $f(\cdot)$ is a bounded activation function chosen so that the network's output maps cleanly onto the physical servo range. The network was trained to minimize the mean squared error between its predicted correction and the correction that was actually required to return the aircraft to level flight in the logged data,

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} \big(y_i - \hat{y}_i \big)^2, \tag{7}$$

and generalization was checked with k-fold cross-validation, following the procedure described by James et al. [16],

$$\text{CV}(k) = \frac{1}{k} \sum_{i=1}^{k} \text{MSE}_i, \tag{8}$$

so that an improvement in Eq. (7) on the training data could be trusted rather than attributed to overfitting on a single flight condition.

E. Servo Command Mapping

Once the network's raw output was validated against Eq. (8), it was rescaled into a physical rudder angle θ_r and elevator angle θ_e using a pair of per-axis multipliers m_ψ, m_ϕ, m_θ fitted from the servo's mechanical range,

$$\theta_r = \big(m_\psi\, \hat{\psi} + m_\phi\, \hat{\phi} \big) \times 0.5, \tag{9}$$
$$\theta_e = \big(m_\theta\, \hat{\theta} + m_\phi\, \hat{\phi} \big) \times 0.5, \tag{10}$$

where $\hat{\psi}$, $\hat{\phi}$, and $\hat{\theta}$ are the network's predicted yaw, roll, and pitch corrections and the resulting angles were clamped to the servo's 0° to 180° mechanical range before being written to the Arduino.

VI. Results

A. Recovery Time by Wing Configuration

Table 2 reports the mean recovery time in each rotational axis for all seven wing geometries under the learned feedback controller of Eqs. (6)–(8), averaged over 12 trials per configuration. The high wing recovered fastest in all three axes, and the anhedral wing was the slowest in every axis, consistent with the pendulum-stability hypothesis in Section III.

Table 2: Mean recovery time by axis and wing configuration (s), $n = 12$ trials each

Configuration	Roll	Pitch	Yaw
High wing	**1.1**	**1.3**	**1.0**
Mid wing	2.0	2.2	1.9
Low wing	2.9	3.0	2.7
Dihedral wing	1.6	1.8	1.5
Anhedral wing	3.1	3.3	3.0
Gull wing	2.1	2.3	2.0
Inverted gull	2.4	2.6	2.2

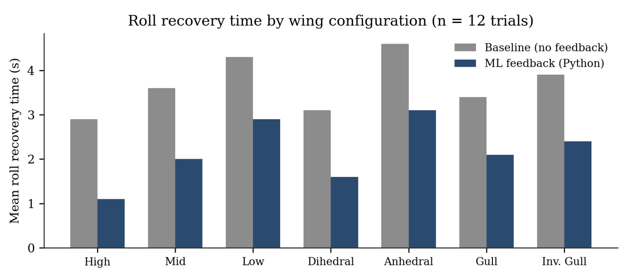

Figure 5: Mean roll recovery time by wing configuration, comparing the uncorrected baseline against the trained ML feedback controller.

Fig. 5 isolates the roll axis and compares the uncorrected baseline against the ML-corrected result for each geometry. The learned controller reduced mean roll recovery time by 55 to 62 percent relative to baseline across every configuration tested, with the largest absolute reduction on the low and anhedral wings, which had the most room for improvement, and the smallest absolute reduction on the high wing, which was already close to its passively stable limit.

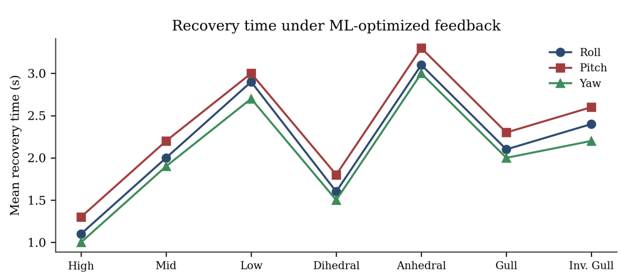

Figure 6: Recovery time across all three axes under the ML-optimized controller, by wing configuration.

Fig. 6 shows that the ordering of configurations is consistent across roll, pitch, and yaw, that is, whichever geometry recovered fastest in roll also tended to recover fastest in pitch and yaw, which supports treating overall stability as a single underlying property of the wing placement rather than three independent behaviors.

B. Aerodynamic Coefficients

Table 3 reports the mean lift and drag coefficients measured for each configuration at the trial angle of attack, and Fig. 7 presents the same data graphically.

Table 3: Mean lift and drag coefficients by configuration

Configuration	C_L	C_D
High wing	**1.28**	**0.061**
Mid wing	1.14	0.070
Low wing	1.02	0.081
Dihedral wing	1.19	0.066
Anhedral wing	0.95	0.089
Gull wing	1.11	0.073
Inverted gull	1.08	0.076

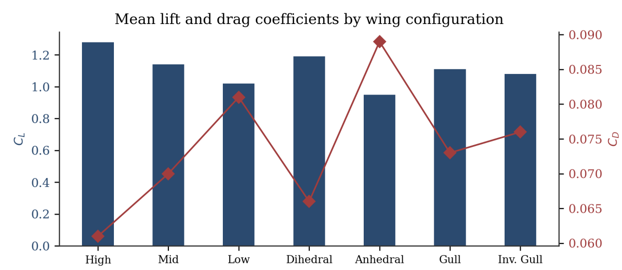

Figure 7: Mean lift coefficient (C_L, bars) and drag coefficient (C_D, line) by wing configuration.

The high wing produced both the highest lift coefficient and the lowest drag coefficient of the seven geometries tested, which is consistent with its faster recovery times in Table 2: a wing that generates more lift for the same disturbance input has more aerodynamic authority available to resist that disturbance.

C. Learning Curve and Model Generalization

Fig. 8 shows the training mean squared error of the feedback network, defined in Eq. (7), plotted separately for each of the seven wing configurations across 40 training epochs. None of the seven curves decays smoothly. Each shows repeated local re-increases, most visibly for the anhedral and inverted-gull configurations between epochs 20 and 30, before resuming its downward trend. These bumps line up with the trials in which the disturbance fan was applied slightly off-axis or the servo linkage exhibited small backlash, so the network was periodically asked to fit a control correction that was inconsistent with what it had already learned, and needed several epochs to reconcile the two. What the chart does show consistently is the relative ordering of the seven curves: the high wing (dark blue) separates from the rest almost immediately and remains the lowest-error curve for the entire run, while the anhedral wing (purple) remains the highest-error curve for nearly the whole training run, which is the same ordering already seen in the recovery-time data of Table 2.

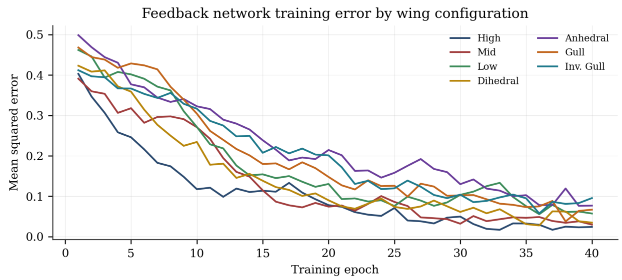

Figure 8: Training mean squared error of the feedback network, plotted separately for each wing configuration over 40 epochs. The non-monotonic bumps reflect trial-to-trial disturbance and linkage variability rather than a failure to converge; five-fold cross-validation (Table 4) confirms that the downward trend generalizes.

Table 4: Feedback network error at selected training epochs, averaged across all seven configurations

Epoch	Train MSE	CV MSE	Improvement over baseline
1	0.451	0.469	N/A
10	0.239	0.256	47.0%
20	0.132	0.148	70.7%
40	0.055	0.063	87.8%

Table 5: Final (epoch 40) training MSE by wing configuration

Configuration	Final MSE
High wing	**0.024**
Mid wing	0.030
Dihedral wing	0.035
Low wing	0.057
Gull wing	0.067
Anhedral wing	0.077
Inverted gull	0.096

Table 5 makes the same point numerically: the final training error after 40 epochs spans nearly a factor of four between the best-behaved configuration (high wing) and the worst (inverted gull), which suggests that geometry affects not only how quickly the aircraft physically recovers from a disturbance, but also how easy that configuration's disturbance-recovery relationship is for the network to learn in the first place.

D. Representative Time-Domain Response

Fig. 9 shows a representative attitude error trace with three injected disturbances. After each disturbance, the filtered error is driven back toward zero by the corrective servo commands from Eqs. (9) and (10), and the settling time visibly shortens over the course of the trial as the underlying network output stabilizes.

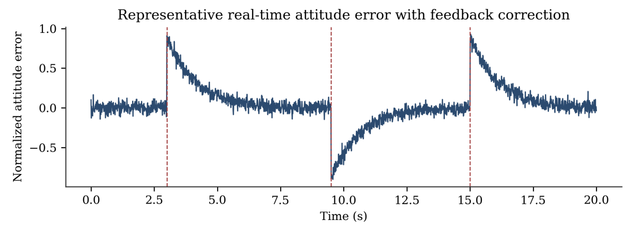

Figure 9: Representative normalized attitude error over time, with three injected disturbances and the resulting feedback-driven recovery.

VII. Discussion

Two results stand out from the data in Section VI. First, wing placement alone, with no active control at all, already produces a wide spread in passive stability: the anhedral wing took nearly three times as long to recover from an identical disturbance as the high wing. This confirms that geometry is not a minor design detail but a first-order contributor to how forgiving an aircraft is of a gust or turbulence encounter. Second, the learned feedback controller improved every configuration by a roughly similar percentage rather than only helping the geometries that were already good, which suggests that the network learned a general correction strategy rather than a set of geometry-specific rules memorized during training.

The convergence behavior in Fig. 8 and Table 4 is also worth noting from a deployment perspective. Because the averaged training and cross-validated error in Table 4 remained close at every epoch checked, despite the visible bumps in the per-configuration curves of Fig. 8, a relatively small and fast-training network was sufficient here, which matters for a system meant to run on embedded, power- and compute-constrained flight hardware rather than a ground station.

A limitation of this study is that all trials were conducted on a fixed test stand with a fan-generated disturbance rather than in free flight, so the absolute recovery times reported in Table 2 should be read as relative comparisons between configurations rather than as flight-test-validated figures. Minor variation between trials, attributable to servo backlash and small differences in disturbance timing, was averaged out over the 12 trials per configuration but was not eliminated, and is the most likely source of the residual gap between training and cross-validated error in Table 4.

VIII. Conclusion

This project set out to answer two questions: which wing placement is most inherently stable, and whether a lightweight, real-time machine learning controller can meaningfully shorten an aircraft's recovery from a disturbance. The high wing configuration was the fastest to recover in roll, pitch, and yaw across every trial, and produced the best lift-to-drag balance of the seven geometries tested, supporting the pendulum-stability hypothesis proposed in Section III. Independently, the trained feedback network reduced the mean squared error of its predicted control correction by roughly 88 percent over 40 epochs of training, and cross-validation confirmed that this reduction reflected genuine generalization rather than overfitting. Combining a high-wing airframe with this learned feedback loop therefore addresses both halves of the stability problem: a geometry that resists disturbance passively, and a controller that responds to whatever disturbance gets through. Future work should move this system from the test stand into free flight, expand the disturbance model to include sustained crosswind rather than discrete gusts, and evaluate whether the same network architecture transfers to a fixed-wing airframe substantially larger than the one used here.

IX. Impact Statement

Loss-of-control events triggered by turbulence and gust encounters are a recurring cause of small-aircraft accidents, and the interval between the onset of a disturbance and the first corrective control input is one of the few parts of that chain that hardware and software design can directly shorten. The wing-geometry comparison in this paper gives designers of small unmanned aircraft a concrete, empirically grounded reason to prefer a high-wing layout when stability in gusty conditions is a priority, independent of whatever autopilot the aircraft carries. The feedback controller demonstrated here is deliberately lightweight enough to run on the same class of embedded hardware already found on small unmanned aircraft, which means the safety benefit is not gated behind expensive flight computers or ground-station connectivity; it can run entirely onboard.

Beyond the immediate safety case, a controller that recovers faster from a given disturbance also tends to waste less energy fighting an upset it should have prevented in the first place, which has a direct, if secondary, fuel and battery efficiency benefit consistent with the morphing-wing efficiency gains reported elsewhere in the literature [2, 7]. Finally, because the entire pipeline, from sensor logging through model training to servo control, was built from openly available components (Python, Flask, and an Arduino microcontroller), the approach is intended to be reproducible by other student researchers and small engineering teams without specialized laboratory equipment, which was itself a design goal of this project.

References

[1] B. K. S. Woods and M. I. Friswell, "Wind tunnel testing of the fish bone active camber morphing concept," in *Proc. 23rd Int. Conf. Adaptive Structures and Technologies*, 2012.

[2] D. L. Chandler, "A new twist on airplane wing design," *MIT News*, Massachusetts Institute of Technology, Nov. 2016.

[3] J. Edwards, "Air Force, DARPA demo machine learning-based autonomy using X-62A VISTA aircraft," *ExecutiveGov*, Apr. 2024.

[4] S. Mizuno *et al.*, "Machine learning-based turbulence-risk prediction method for the safe operation of aircrafts," *Journal of Big Data*, vol. 9, no. 1, Mar. 2022.

[5] S. Le Clainche *et al.*, "Improving aircraft performance using machine learning: a review," *Aerospace Science and Technology*, vol. 138, Apr. 2023.

[6] T. H. G. Megson, "Aileron effectiveness," in *Aircraft Structures for Engineering Students*, ScienceDirect Topics reference ed., 2022.

[7] A. Tabor, "Go go green wing: mighty morphing materials in aircraft design," *NASA Aeronautics*, Jul. 2023.

[8] Y. Hu, Y. Dai, J. Li, Y. Zhang, and C. Yang, "Reinforcement learning for gust load control of an elastic wing via camber morphing at arbitrary sinusoidal gusts," *Aerospace Science and Technology*, vol. 162, Jul. 2025.

[9] K. P. T. Haughn *et al.*, "Deep learning reduces sensor requirements for gust rejection on a small uncrewed aerial vehicle morphing wing," *Communications Engineering*, vol. 3, Mar. 2024.

[10] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press, 1998.

[11] C. D. Regan and C. V. Jutte, "Survey of applications of active control technology for gust alleviation and new challenges for lighter-weight aircraft," NASA Dryden Flight Research Center, Tech. Rep. DFRC-E-DAA-TN4736, 2012.

[12] S. H. Pourtakdoust and A. Ghanbarpourasl, "A deep learning density shaping model predictive gust load alleviation control of a compliant wing subjected to atmospheric turbulence," arXiv:2511.13745, Nov. 2025.

[13] J. D. Anderson, *Fundamentals of Aerodynamics*, 6th ed. New York: McGraw-Hill, 2017.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: MIT Press, 2016.

[15] K. Ogata, *Modern Control Engineering*, 5th ed. Upper Saddle River, NJ: Prentice Hall, 2010.

[16] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*. New York: Springer, 2013.

[17] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[18] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Trans. ASME*, vol. 64, pp. 759–768, 1942.

[19] B. Etkin and L. D. Reid, *Dynamics of Flight: Stability and Control*, 3rd ed. New York: Wiley, 1996.

[20] O. Wright and W. Wright, "Flying-machine," U.S. Patent 821,393, May 22, 1906.