VINÍCIUS NUNES AMARAL

ARTIFICIAL NEURAL NETWORKS FOR CRITICAL FLUTTER SPEED PREDICTION IN SEMI-SPAN FLAT-PLATE WINGS WITH GEOMETRIC VARIATIONS



UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE ENGENHARIA MECÂNICA
2025

VINÍCIUS NUNES AMARAL

ARTIFICIAL NEURAL NETWORKS FOR CRITICAL FLUTTER SPEED PREDICTION IN SEMI-SPAN FLAT-PLATE WINGS WITH GEOMETRIC VARIATIONS

Final Paper submitted to the Faculty of Mechanical Engineering from Federal University of Uberlândia in partial fulfillment of the requirements for the degree of Aeronautical Engineer. *FINAL VERSION*

Advisor: Prof. Dr. Odenir de Almeida
Co-advisor: Prof. Dr. Tobias Souza Morais

Uberlândia - MG
2025

This work is dedicated to my parents, and to all the professors, mentors, and friends who supported me.

ACKNOWLEDGEMENTS

I would like to thank everyone who supported me throughout this work, especially Prof. Dr. Odenir de Almeida and Prof. Dr. Tobias Souza Morais for their patience and essential guidance on the content, methods, and experiments presented here. I also thank my parents, Flaviano and Rose, for always supporting me and doing everything possible, and sometimes seemingly impossible, to help me achieve my goals. Finally, I am grateful to all members of the 22nd cohort of Aeronautical Engineering at the Federal University of Uberlândia for walking this undergraduate journey with me in unity and collaboration.

"We make our world significant by the courage of our questions and the depth of our answers"
(Carl Sagan)

ABSTRACT

AMARAL, V. N. **Artificial Neural Networks for Critical Flutter Speed Prediction in Semi-Span Flat-Plate Wings with Geometric Variations**. 2025. 56 p. Trabalho de Conclusão de Curso (Graduação em Engenharia Aeronáutica) – Faculdade de Engenharia Mecânica, Universidade Federal de Uberlândia, Uberlândia - MG, 2025.

This thesis presents a data-driven method to predict modal damping and the critical flutter speed of semi-span flat-plate wings. A linear FE model (FEMAP/MSC Nastran SOL 145, PKNL with Doublet–Lattice aerodynamics) was calibrated with impact-hammer modal tests and used to generate mode-by-mode $g(V)$ and $f(V)$ curves for spans 0.35–0.50 m. From these runs we built a supervised dataset with inputs $\{S, V, f_1, \ldots, f_4\}$ and targets $\{g_1, \ldots, g_4\}$ and trained a regularized MLP that predicts damping across modes; flutter speed is inferred as the first zero-crossing of the predicted $g(V)$. On held-out simulation cases the network achieved low errors and reproduced the span-dependence of flutter speed; comparisons with wind-tunnel data (limited to $\sim$25 m/s) showed agreement within the experimental resolution. The approach provides near-instant estimates suitable for preliminary design, while remaining limited by the scope of the parametric sweep and the tunnel envelope. Future work includes broadening geometry variation (e.g., dihedral, sweep), expanding numerical and experimental coverage, and exploring physics-guided or unsupervised models that learn stability directly from raw time-series.

Keywords: Aeroelasticity; Flutter; Semi-span wing; Modal damping; Finite Element Method (FEM); MSC Nastran SOL 145; Doublet–Lattice Method (DLM); PK/PKNL flutter methods; Machine Learning; Multi-Layer Perceptron (MLP).

LIST OF FIGURES

Figure 1 – Northwest 710 Accident 12
Figure 2 – Collar's Triangle 15
Figure 3 – Binary (two–DOF) airfoil "typical section" 17
Figure 4 – Classical identification of U_f via V–g and V–f diagrams 19
Figure 5 – Schematic workflow of a numerical flutter simulation 22
Figure 6 – Schematic of an MLP 25
Figure 7 – Semi-span wing view and mounting 30
Figure 8 – Structural mesh and fixed root boundary in FEMAP (example semi-span 0.41 m) 32
Figure 9 – DLM panel layout (CAERO1) and SPLINE1 coupling to the structural mesh 33
Figure 10 – Mode shape illustration (0.41 m semi-span example) 35
Figure 11 – Semi-span wing experimental mounting and positioning 37
Figure 12 – Acquisition system 38
Figure 13 – Experimental and Simulation V-g-f Comparison (S = 0.41 m) 46
Figure 14 – Experimental and Simulation V-g-f Comparison (S = 0.44 m) 47
Figure 15 – Experimental and Simulation V-g-f Comparison (S = 0.47 m) 48
Figure 16 – Training and validation loss versus epoch for the final MLP. 50

LIST OF TABLES

Table 1 – Summary of flutter solution methods available in SOL 145 24
Table 2 – Semi-span wing configurations used throughout this study 30
Table 3 – Experimental modal frequencies obtained 31
Table 4 – MSC Nastran SOL 145 flutter analysis settings (applied to all semi-span cases). 34
Table 5 – First lines (samples) of the constructed dataframe 41
Table 6 – Neural network training parameters (final configuration). 43
Table 7 – Simulation vs. experiment at flutter (Mode responsible indicated; errors relative to experimental). 49
Table 8 – Aggregate errors for the final model (simulation test set). 50
Table 9 – Per-mode errors on the simulation test set (physical damping units). . . 51
Table 10 – ANN flutter speed prediction ($U_{f_{ANN}}$) versus SOL 145 (PKNL) and experimental data 52

CONTENTS

1 INTRODUCTION 11
1.1 Motivation and Objectives 11

2 THEORETICAL BACKGROUND AND LITERATURE REVIEW . . 14
2.1 Aeroelasticity Fundamentals 14
2.2 Flutter Phenomenon Theory 16
2.2.1 Classical binary model and eigenvalue viewpoint *17*
2.2.2 Energy/Work perspective and definition of flutter speed *18*
2.3 Structural Dynamics and Modal Damping Concepts 19
2.3.1 Modal properties from a finite–element model *20*
2.3.2 Structural vs aerodynamic (effective) modal damping *20*
2.3.3 Damping from complex eigenvalues *21*
2.3.4 Mode coupling and coalescence *21*
2.3.5 The challenge in prediction *22*
2.4 Numerical Flutter Analysis Methods (Finite Element Model (FEM) & Solver Approaches) 22
2.4.1 SOL 145 in MSC NASTRAN Software *23*
2.4.1.1 K–method (frequency sweep). *23*
2.4.1.2 PK–method (iterative complex–eigenvalue solve). *23*
2.4.1.3 PKNL—"PK, No Loop". *23*
2.4.2 Aerodynamic model and outputs *24*
2.5 Machine Learning in Engineering Applications 24
2.5.1 MLP essentials for regression in engineering *24*
2.5.2 MLP for flutter-related regression *26*
2.6 Literature Comparison 26

3 METHODOLOGY 29
3.1 Semi-Span Wing Model Description 29
3.1.1 Geometry and structure *29*
3.2 Experimental Modal Evaluation 31
3.3 Finite Element Modeling and Flutter Simulation 31
3.4 Wind Tunnel Experimental Procedure 35
3.5 Data Preparation and Feature Engineering for the Neural Network . 39

3.6 Neural Network Architecture and Training 41
3.7 Neural Model Evaluation Strategy and Performance Metrics 44

4 RESULTS AND ANALYSIS 45
4.1 Simulation Results and Experimental Data Comparison 45
4.2 Neural Network Training Performance 49
4.3 Prediction of Critical Flutter Speed using the Neural Network . . . 51

5 CONCLUSION 53

BIBLIOGRAPHY 55

CHAPTER

1

INTRODUCTION

1.1 Motivation and Objectives

The aeroelastic phenomenon known as flutter is characterized by the coupled self-excitation of two or more vibratory modes of a structure, sustained and re-energized by the surrounding fluid flow. Among all aeroelastic phenomena, flutter stands out as the most consequential and the most difficult to predict (Collar (1978), Garrick and Reed (1981)). Such self-excited vibrations can lead to complete structural failure in virtually any structure, which underscores the importance of analyzing and understanding this phenomenon. In aeronautical and aerospace contexts, flutter primarily affects aircraft wings and control surfaces, as these components govern the local flow field and are most directly impacted by the resulting aerodynamic loads.

On March 17, 1960, Northwest Flight 710, operated by a Lockheed L-188C Electra, experienced an in-flight rupture and separation of the right wing near Cannelton/Tell City (Indiana) following a structural failure, as can be seen in the Figure 1. The investigation by the Civil Aeronautics Board (CAB), the former U.S. federal agency responsible for aircraft accident investigations, concluded that one of the causes of the rupture was flutter induced by “whirl-mode” oscillations of the aircraft’s outer nacelles. This severe accident claimed the lives of all 63 people on board. Consequently, predicting the structural behavior of wings and semi-span wings with respect to flutter is a critical aspect of aircraft design to ensure safety and prevent tragedies such as Northwest Flight 710.

Figure 1 – Northwest 710 Accident



Source: Tadeu (2022).

As a nonlinear phenomenon involving fluid–structure interaction, the prediction of flutter and the determination of the critical flutter speed, the lowest flow speed at which the aeroelastic system becomes neutrally stable and begins to admit self-excited oscillations, are highly challenging. Accordingly, semi-span wing models (i.e., half-wings) rigidly mounted to a lateral wall or immediately downstream of a wind-tunnel exit are widely used as representative experimental configurations for investigating flutter because they simplify instrumentation and enable exploration of different wing and semi-span layouts. Even so, wind-tunnel testing is logistically difficult and costly, requiring many hours of preparation and tunnel time and, above all, posing operational risks, since flutter is intrinsically destructive and difficult to control, capable of intensifying abruptly. This thesis set out to develop and evaluate a neural–network–based methodology for predicting modal damping and the critical flutter speed of flat–plate semi–span wings. Moreover, this methodology introduces additional challenges: the root boundary condition is not a perfectly rigid clamp, and small clearances can alter the natural frequencies and the damping-versus-speed curves of the structure.

In this context, numerical analyses using software such as FEMAP/NASTRAN become essential tools. NASTRAN's flutter solution (SOL 145), with PK and PKNL methods coupled to unsteady aerodynamic models (e.g., the doublet-lattice method), provides estimates of modal frequencies and damping as functions of speed for different geometric configurations (such as span variations) under ideal boundary conditions (a clamped root). These simulations enable rapid sweeps of structural parameters (EI, GJ), mass, and inertia, thereby identifying trends and margins. Nevertheless, the computational model relies on assumptions that simplify reality, such as perfect clamping, idealized pressure distributions, absence of clearances, and linearity for small amplitudes. Small discrepancies in

local stiffness, mass added by wiring and sensors, or support damping can significantly shift the estimated critical speed.

Conversely, wind-tunnel tests provide the empirical reference needed to validate these predictions. With accelerations and displacements measured along the span and incremental sweeps of speed, it is possible to obtain v–f (frequency versus speed) and v–g (damping versus speed) diagrams directly, thereby identifying the point at which damping crosses zero, which characterizes the critical speed. However, this approach entails substantial costs and risks: wind-tunnel hours are expensive; instrumentation and preparation are time-consuming; and operating near flutter requires safety procedures (fine speed increments, abort criteria, relief devices, and protections for the test section) to avoid catastrophic model failure, tunnel damage, and operational hazards. In addition, Reynolds-number limitations (especially at reduced scales) and measurement noise in low-damping regimes impose further uncertainties.

Thus, a discrepancy between simulation and experiment often arises, where simulations offer broad parametric reach but depend on assumptions that may not fully capture real behavior, whereas tests and experiments, when properly conducted, yield results closer to reality but are costly, time-consuming, and potentially hazardous in the vicinity of flutter. The objective of this work is to combine numerical results obtained with FEMAP/NASTRAN (SOL 145) and wind-tunnel tests on flat-plate semi-span wings to assemble a coherent dataset. This dataset will then be used to develop a predictive methodology based on Machine Learning (ML) using neural-network models, with emphasis on a Multilayer Perceptron (MLP) architecture trained on simulation-generated data. Subsequently, the neural model's predictions will be compared with and validated against experimental measurements to estimate modal damping and the critical flutter speed for different spans.

This chapter presents the overall aeroelastic context, defines the flutter phenomenon and its relevance, delineates the semi-span problem, discusses the prediction challenge (simulation versus experiment), and states the objectives, contributions, and scope of the work. The next chapter provides the theoretical review, including classical flutter models, semi-span configurations, and definitions related to the SOL 145 solution and the doublet-lattice method, as well as essential ML concepts. Chapter 3 details the numerical and experimental methods, specifying the geometry, properties, FEMAP/NASTRAN modeling, wind-tunnel setup, signal acquisition and processing, and also describes the machine-learning methodology (dataset construction, features, MLP, training/validation, and metrics). Chapter 4 presents and discusses the results (simulation and experimental v–g and v–f diagrams, critical-speed estimates, performance, and uncertainties). Finally, Chapter 5 concludes the work with its limitations and directions for future research.

CHAPTER

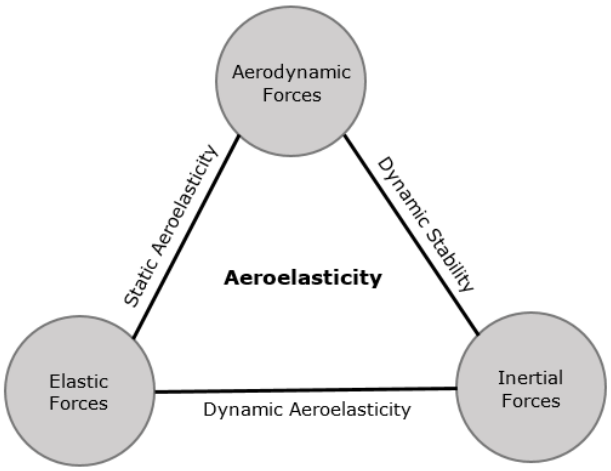
2

THEORETICAL BACKGROUND AND LITERATURE REVIEW

2.1 Aeroelasticity Fundamentals

Aeroelasticity examines the mutual interaction among aerodynamic loads, structural elasticity, and inertia. In flight, the air loads acting on a lifting surface depend on its instantaneous motion and deformation. In turn, the structure deforms under those loads (Collar (1978), Fung (2008), Wright and Cooper (2015), Bismarck-Nasr (1999)). A compact way to organize these interactions is the diagram known as *Collar's triangle*, in which aerodynamics, elasticity, and inertia occupy the vertices. Problems confined to a single edge describe two-way couplings, *static aeroelasticity* on the aerodynamic, elastic edge and *dynamic stability/flight dynamics* on the aerodynamic, inertial edge, whereas phenomena such as flutter reside in the interior, requiring the simultaneous action of all three mechanisms. Fig. 2 summarizes this viewpoint (adapted from Leishman (2023)). The remainder of this section establishes the language and concepts used throughout the work, leading from this conceptual picture to the linear structural model adopted.

Figure 2 – Collar's Triangle



Source: Leishman (2023).

From the structural viewpoint, the semi–span wing is modeled as a linear, small–motion elastic system. Let $q(t)$ represent the physical degrees of freedom, in vacuo, the dynamics are governed as shown in Eq. 2.1.

$$\mathbf{M}\,\ddot{\mathbf{q}}(t) + \mathbf{C}_s\,\dot{\mathbf{q}}(t) + \mathbf{K}_s\,\mathbf{q}(t) = \mathbf{0}, \tag{2.1}$$

where M, C_s and K_s denote the mass, structural damping, and stiffness matrices. Modal analysis provides a convenient reduced description. Writing $q(t) = \Phi\,\eta(t)$, where the columns of Φ are mass–normalized mode shapes and η the modal coordinates, each retained mode behaves approximately as a single–degree–of–freedom oscillator, as shown in Eq. 2.11

$$\ddot{\eta}_r(t) + 2\,\zeta_r\,\omega_r\,\dot{\eta}_r(t) + \omega_r^2\,\eta_r(t) = 0, \tag{2.2}$$

with natural frequency ω_r and damping ratio ζ_r. For lifting surfaces, the first bending and the first torsional modes are particularly influential, their relative frequencies, nodal patterns, and the offsets among the elastic axis, mass center, and aerodynamic center determine how strongly they couple in flow (Wright and Cooper (2015)).

Coupling the structure to a uniform stream of speed U introduces unsteady aerodynamic forces that depend on both deformation and velocity. In linear small–disturbance theory, this is represented compactly as demonstrated in Eq. 2.3.

$$\mathbf{M}\,\ddot{\mathbf{q}} + \mathbf{C}_s\,\dot{\mathbf{q}} + \mathbf{K}_s\,\mathbf{q} = \mathbf{Q}_a(\mathbf{q}, \dot{\mathbf{q}}, U)\,, \tag{2.3}$$

where Q_a is supplied by an unsteady aerodynamic model appropriate to the regime (e.g., Theodorsen theory for airfoils or doublet–lattice aerodynamics for finite wings). Projecting (2.3) onto the structural modes yields a modal form in which the flow appears as effective, velocity–dependent aerodynamic damping and stiffness, as shown in Eq. 2.4

$$\mathbf{M}_m \ddot{\boldsymbol{\eta}} + \big(\mathbf{C}_s + \mathbf{C}_a(U)\big)\,\dot{\boldsymbol{\eta}} + \big(\mathbf{K}_s + \mathbf{K}_a(U)\big)\,\boldsymbol{\eta} = \mathbf{0}, \tag{2.4}$$

whose entries vary with reduced frequency $k = \omega b/U$ and with flight condition. The matrices C_a and K_a embody phase lags and magnitude changes in lift and moment that are central to aeroelastic behavior (Fung (2008)). As the airspeed increases, these flow-induced terms may partially cancel the structural stiffness and damping or, conversely, reinforce them, thereby altering the apparent modal frequencies and damping ratios.

It is useful to distinguish outcomes by their time scale. When the flow merely shifts the static equilibrium and the modified stiffness remains positive, the problem is one of static aeroelasticity. If, beyond a certain speed, the aerodynamic pitching moment about the elastic axis overwhelms the torsional restoring moment, the structure cannot find a bounded equilibrium and diverges in twist, which defines the divergence speed in the classical sense. When time dependence and aerodynamic phase lags play a decisive role, the problem is dynamic aeroelasticity.

The perspective adopted follows the standard linear small–disturbance framework. The semi–span wing is represented by a finite–element structural model and a linear unsteady aerodynamic approximation; the coupled system is reduced to a handful of dominant modes, typically the first bending and the first torsion, whose frequencies and damping evolve with airspeed.

2.2 Flutter Phenomenon Theory

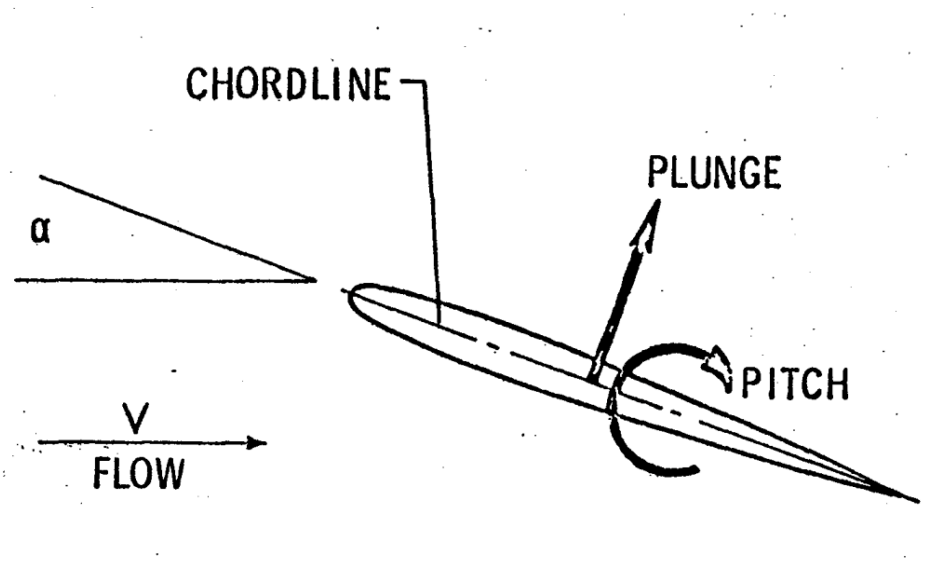
As previously mentioned, flutter is a self–excited aeroelastic instability that emerges from the interaction between the unsteady aerodynamic forces and the elastic–inertial response of a structure, typically when two or more modes couple so that the flow feeds energy into the motion. Above a critical flight speed, the effective damping of the combined system vanishes and then becomes negative, leading to exponentially growing oscillations. Classical expositions (e.g., Fung (2008), Wright and Cooper (2015), Bismarck-Nasr (1999), Theodorsen (1934)) frame flutter both in the time–domain (energy/work balance) and in the frequency–domain (eigenvalue criteria). In this thesis, the neural network will learn to predict precisely the *flutter speed* U_f (the lowest airspeed at which the linearized sys-

tem becomes neutrally stable) and the modal damping $g(V)$ as functions of speed and geometry.

2.2.1 Classical binary model and eigenvalue viewpoint

A minimal theoretical description of bending–torsion flutter is the two–degree–of–freedom (two-DOF) airfoil model (typical section) with plunge h and pitch α about the elastic axis. Denote by m the section mass, by I_α the pitch mass moment of inertia, by S_α the static unbalance, and by k_h, k_α the linear stiffnesses (viscous structural damping c_h, c_α can be included). The equations of motion, written for small motions and immersed in a uniform stream of speed U, are as demonstrated in Eq. 2.5 an Eq 2.6,

Figure 3 – Binary (two–DOF) airfoil "typical section"



Source: Tischler and Bencic (1982).

$$m\ddot{h} + S_\alpha \ddot{\alpha} + c_h \dot{h} + k_h h = -L(h, \alpha, \dot{h}, \dot{\alpha}; U), \tag{2.5}$$

$$S_\alpha \ddot{h} + I_\alpha \ddot{\alpha} + c_\alpha \dot{\alpha} + k_\alpha \alpha = M_a(h, \alpha, \dot{h}, \dot{\alpha}; U), \tag{2.6}$$

where L and M_a are the (unsteady) lift and aerodynamic moment. In linear small—disturbance theory for thin airfoils, these aerodynamic actions are linear functionals of $(h, \alpha, \dot{h}, \dot{\alpha})$ with a frequency-dependent phase lag encapsulated, for attached flow, by Theodorsen's complex function $C(k)$ with reduced frequency $k = \omega b/U$ (with b the semi-

chord) (Theodorsen (1934), Fung (2008)). Seeking time-harmonic responses $h(t), \alpha(t) \propto e^{i\omega t}$ reduces Eq. 2.5–2.6 to the nonlinear (in ω) eigenproblem showed in Eq. 2.7.

$$\left[-\omega^2 M + i\omega C_a(k,U) + K + K_a(k,U) \right] \hat{q} = 0, \tag{2.7}$$

with $\hat{q} = [\hat{h} \ \hat{\alpha}]^T$, structural matrices M, K and aerodynamic operators C_a, K_a coming from the chosen unsteady model (e.g. Theodorsen or, for finite wings, doublet–lattice). In the p–k formulation, one writes $p = \sigma + i\omega$ and seeks, for a given k, the roots p of the flutter determinant. A flutter point occurs when $\sigma(k,U) = 0$ admits a nontrivial solution and the associated airspeed U is the *flutter speed* U_f, while $\omega(k,U_f)$ gives the *flutter frequency* $f_f = \omega/2\pi$ (Wright and Cooper (2015), Fung (2008)).

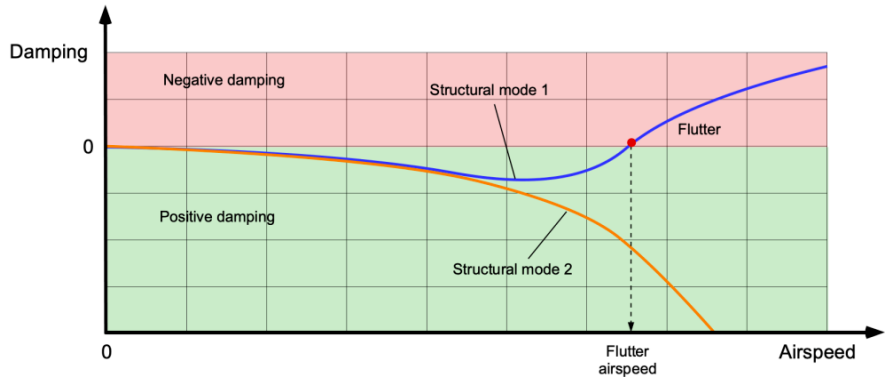
2.2.2 Energy/Work perspective and definition of flutter speed

The same onset condition can be viewed via work/energy. Over one cycle of a periodic motion with period $T = 2\pi/\omega$, the net aerodynamic work on the generalized coordinates $q(t)$ can be written as shown in Eq. 2.8.

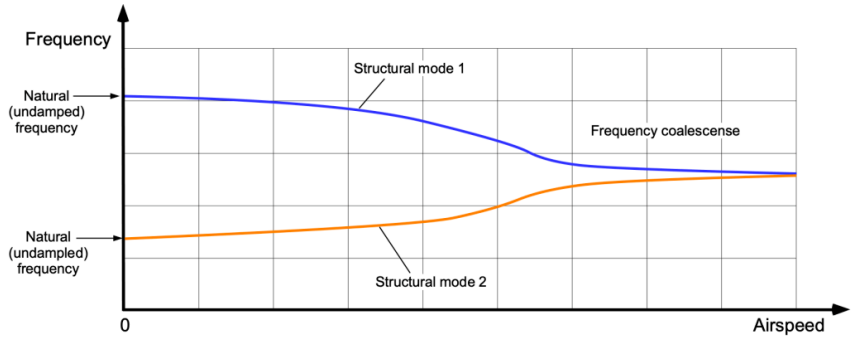
$$W = \int_0^T \dot{q}^\top(t)\, Q_a\big(q, \dot{q}; U\big)\, dt, \tag{2.8}$$

where Q_a collects the generalized aerodynamic forces. In the linear regime, W changes sign with airspeed: below U_f the flow extracts energy (positive effective damping), at U_f the net work per cycle vanishes ($W = 0$), and above U_f the flow injects energy (negative effective damping), causing amplitude growth. Thus, *flutter speed* U_f is the smallest U such that the real part of some eigenvalue $p = \sigma + i\omega$ crosses zero (i.e., $\sigma(U_f) = 0$), as shown in Fig. 4, and *flutter frequency* is the corresponding $\omega(U_f)$.

Figure 4 – Classical identification of U_f via V–g and V–f diagrams



(a) V–g (damping vs. velocity) for flexible modes



(b) V–f (frequency vs. velocity) for the same modes

Source: Inc. (2022).

2.3 Structural Dynamics and Modal Damping Concepts

Structural dynamics provides the basic vocabulary needed throughout this thesis. A linearized finite-element model of the semi-span wing, considered in vacuo, yields a set of free-vibration modes characterized by their natural frequencies and mode shapes. With small structural dissipation, each mode behaves as a lightly damped single–degree–of–freedom oscillator. When the wing is immersed in airflow, the unsteady aerodynamic loads couple with the structural motion and alter the apparent stiffness and damping of these modes. The resulting modal damping ratio, interpreted as the rate at which a modal response decays or grows with time, is positive for stable decay, vanishes at neutral stability, and becomes negative when the flow injects energy into the structure, a hallmark of flutter. In what follows, these ideas are formalized so that the outputs of a flutter analysis (modal frequencies and modal damping as functions of speed) have clear physical meaning and can later be used consistently as inputs and targets for the learning framework.

2.3.1 Modal properties from a finite–element model

For a linear elastic structure with assembled mass matrix M and stiffness matrix K, the undamped free–vibration problem is the generalized eigenvalue problem in Eq. 2.9.

$$K\,\phi_r \;=\; \omega_r^2\,M\,\phi_r, \tag{2.9}$$

whose solutions (ω_r, ϕ_r) define the natural (circular) frequencies ω_r and mode shapes ϕ_r. With mass normalization $\phi_r^{\top} M \phi_s = \delta_{rs}$, the physical coordinates $q(t)$ can be expanded as demonstrated in Eq. 2.10.

$$q(t) \;=\; \Phi\,\eta(t), \qquad \Phi = \begin{bmatrix}\phi_1 & \phi_2 & \cdots\end{bmatrix}, \tag{2.10}$$

so that, neglecting modal coupling, each retained mode behaves approximately as a single degree of freedom (SDOF), guided by Eq. 2.11.

$$\ddot{\eta}_r(t) + 2\,\zeta_{s,r}\,\omega_r\,\dot{\eta}_r(t) + \omega_r^2\,\eta_r(t) \;=\; 0, \tag{2.11}$$

where $\zeta_{s,r}$ is the structural (material/added) damping ratio assigned to mode r. In the time domain, a lightly damped SDOF exhibits an exponentially decaying oscillation, as shown in Eq. 2.12.

$$x(t) \;=\; A\,\mathrm{e}^{-\zeta\omega t}\cos\!\big(\omega_d\,t + \varphi\big), \qquad \omega_d = \omega\sqrt{1-\zeta^2}, \tag{2.12}$$

which fixes the physical meaning of a positive damping ratio: amplitudes decay in time.

2.3.2 Structural vs aerodynamic (effective) modal damping

When immersed in a flow of speed U, the structure experiences unsteady aerodynamic forces that depend on deformation and rate, adding velocity and frequency–dependent terms to the modal equations. Projecting the coupled equations onto the structural modes yields, schematically, in the Eq. 2.13.

$$\ddot{\eta} + \big(2\,\Omega\,\zeta_s + C_a(k,U)\big)\,\dot{\eta} + \big(\Omega^2 + K_a(k,U)\big)\,\eta \;=\; 0, \tag{2.13}$$

where $\Omega = \mathrm{diag}(\omega_1, \omega_2, \ldots)$, $k = \omega b/U$ is the reduced frequency, and C_a, K_a are aerodynamic operators provided by the chosen unsteady model (e.g., Theodorsen for airfoils;

doublet–lattice for finite wings (Fung (2008), Wright and Cooper (2015))). For each mode r, one can define an effective damping ratio

$$\zeta_r(U) \;=\; \zeta_{s,r} \;+\; \zeta_{a,r}(U), \tag{2.14}$$

where $\zeta_{a,r}(U)$ represents the aerodynamic contribution. As U increases, $\zeta_{a,r}(U)$ can become sufficiently negative to drive $\zeta_r(U)$ to zero and then below zero: this is the classical onset of flutter for that coupled set of modes.

2.3.3 Damping from complex eigenvalues

In practical flutter solvers (e.g., NASTRAN PK/PKNL), the stability at a flight point is obtained from the complex eigenvalues of the Eq. 2.15,

$$p_r(U) \;=\; \sigma_r(U) \;+\; i\,\omega_r(U), \tag{2.15}$$

where the real part σ_r is the modal growth/decay rate and the imaginary part ω_r is the (circular) frequency. A convenient nondimensional damping measure used in V–g (damping versus speed) diagrams is the one from Eq. 2.16,

$$g_r(U) \;=\; -\,\frac{\sigma_r(U)}{\omega_r(U)}\,, \tag{2.16}$$

which, for lightly damped motions, is numerically close to the damping ratio ζ_r. The flutter point for a mode (or a coupled pair) is identified when $g_r(U_f) = 0$ (equivalently, $\sigma_r(U_f) = 0$). Any further decrease of g_r below zero indicates exponentially growing oscillations. The associated flutter frequency is $f_f = \omega_r(U_f)/(2\pi)$.

2.3.4 Mode coupling and coalescence

In lifting surfaces, the first bending and first torsional modes often lie closest in frequency and exhibit strong aerodynamic cross–coupling. With increasing airspeed, the effective aerodynamic stiffness and damping shift the modal eigenvalues until a pair of complex–conjugate roots coalesces and the real parts pass through zero ("coalescence flutter"). The shapes at onset display a characteristic mixture of bending and twist, consistent with the typical section abstraction in Fig. 3. For finite wings, this coupling is captured, for example, by doublet–lattice aerodynamics in frequency–domain solvers and by the p–k method used in industry (Wright and Cooper (2015)).

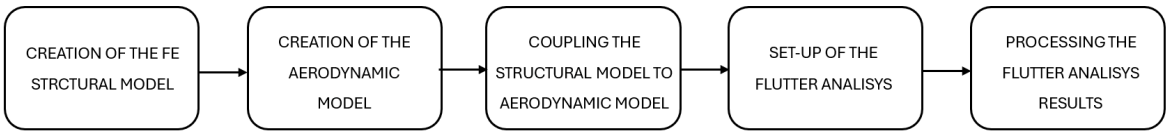
2.3.5 The challenge in prediction

Accurate U_f prediction depends on several interacting factors: unsteady aerodynamics with frequency–dependent phase lags, structural modeling of mass and stiffness distributions and boundary conditions, and the proximity of multiple modes. Analytical models become cumbersome beyond the binary case. Hence, finite–element (finite–wing) solvers (e.g., NASTRAN SOL 145 with doublet–lattice aerodynamics and the p–k method) are standard, and experiments remain the ultimate validation. In this work, the neural model targets the same physical observables used by classical flutter analysis, modal damping as a function of V and the crossing to define U_f, with supervised learning on simulation data and validation against wind–tunnel measurements.

2.4 Numerical Flutter Analysis Methods (Finite Element Model (FEM) & Solver Approaches)

Because closed–form solutions are impractical for real wings, flutter prediction in engineering practice relies on *numerical* aeroelastic analysis: a finite–element (FE) structural model is coupled to an unsteady aerodynamic model to form a linear, parameter–dependent eigenvalue problem whose roots give modal frequency and damping as functions of the flight condition (Mach, dynamic pressure). In the subsonic, small–disturbance regime adopted here, the most common aerodynamic idealization is the *doublet–lattice method* (DLM), which delivers generalized aerodynamic forces (GAFs) as complex, reduced–frequency–dependent matrices that, when projected onto structural modes, act as effective aerodynamic stiffness and aerodynamic damping (Wright and Cooper (2015), Fung (2008), Bismarck-Nasr (1999)). Modern solvers (such as MSC NASTRAN SOL 145) assemble all of this and march over a set of flight points to produce the classical V–g and V–f diagrams used to identify the flutter point.

Figure 5 – Schematic workflow of a numerical flutter simulation



Source – Elaborated by the author

2.4.1 SOL 145 in MSC NASTRAN Software

2.4.1.1 K–method (frequency sweep).

Historically, the K–method evaluates GAFs on a prescribed grid of reduced frequencies k, assembles Eq. 2.7 with real ω and searches for a sign change in the computed damping. It effectively treats k as an input and scans for the locus of minimum damping while the flight condition is varied. It is straightforward but can be computationally heavier if a fine k–grid is needed and may require careful interpolation of GAFs (Wright and Cooper (2015)).

2.4.1.2 PK–method (iterative complex–eigenvalue solve).

The PK method assumes harmonic motion and reformulates the flutter problem as a complex eigenvalue problem in $p = \sigma + i\omega$. Because the aerodynamic operators depend on $k = \omega b/U$, the solver iterates: starting from an initial guess of ω (thus k), it computes C_a and K_a, solves for p, updates k from the new ω, and repeats until convergence of p (typically both real and imaginary parts). The result at each flight point is a set of complex roots whose real parts (damping) and imaginary parts (frequency) feed the V–g and V–f plots. PK is the workhorse for subsonic flutter with DLM and is well documented in the aeroelastic literature and vendor manuals Wright and Cooper (2015), Bismarck-Nasr (1999).

2.4.1.3 PKNL—"PK, No Loop".

Large parametric sweeps (e.g., many spans or mass/stiffness variations) benefit from a non–iterative alternative. The PKNL variant in SOL 145 evaluates the aerodynamic operators at a single reduced frequency per flight point, typically taken from the previous speed step or from a user–supplied reference, and solves a single complex eigenproblem without re–evaluating C_a and K_a after the eigenvalue update (i.e., no internal loop on frequency). In other words, PKNL trades the strict k–consistency of PK for speed and monotonic, grid–like sampling of the flight envelope. This makes it particularly useful when the goal is to map stability at prescribed speeds and generate smooth V–g curves to feed downstream tasks (as in this work). The cost is a small loss of accuracy near rapid k–variation. In those cases one can refine the flight grid or, if needed, spot–check with full PK at a few critical points.

Table 1 – Summary of flutter solution methods available in SOL 145

Method	Approach	Typical use
K	Frequency sweep; evaluates GAFs on a grid of k and scans damping	Simple conceptually; robust. May require dense k–grid and interpolation; less convenient for large parametric studies.
PK	Complex–eigenvalue solve with iteration on k until $p = \sigma + i\omega$ converges	High fidelity and k–consistent; computationally costlier per point; preferred for final verification.
PKNL	"PK, no loop": one evaluation of GAFs per flight point (no k iteration)	Very fast and monotonic sweep of prescribed speeds; ideal to generate V–g/V–f datasets. Slight accuracy loss when k varies sharply; refine grid or spot–check with PK.

Source – Elaborated by the author

2.4.2 Aerodynamic model and outputs

For the subsonic cases considered, the DLM provides the GAFs on a lattice of aerodynamic panels, parameterized by Mach and reduced frequency. In SOL 145 the aeroelastic assembly uses the mode–shapes selected for the flutter set, forming small, dense matrices for each flight point. The solver reports, for each tracked mode, the damping $g(V)$ and the frequency $f(V)$. Plotting these curves yields the classical V–g and V–f diagrams from which the flutter speed U_f is read where g crosses zero and the flutter frequency is read from the corresponding f (Kehoe (1988)).

2.5 Machine Learning in Engineering Applications

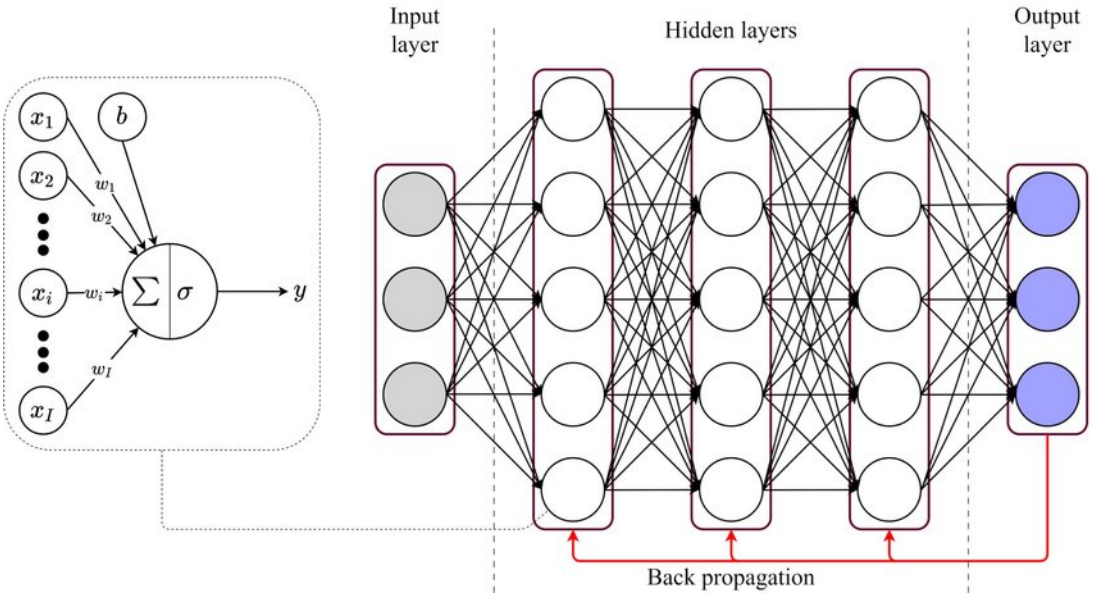
Data-driven methods are increasingly used in aerospace to complement physics-based models, especially when fast surrogate predictions are needed. In this thesis, we adopt a supervised learning approach with a multi-layer perceptron (MLP) to learn mappings from structural and flight-condition features to modal damping and to the flutter speed indicator. The goal is to provide a fast, low-risk predictor consistent with classical flutter analysis outputs (e.g., V–g and V–f curves), while remaining grounded in the standard definitions of features and targets reviewed earlier.

2.5.1 MLP essentials for regression in engineering

An MLP is a stack of layers of simple computing units (neurons), each performing an affine transform followed by a nonlinear activation (e.g., ReLU, sigmoid, tanh). Layers

are arranged as input layer, followed by a certain number of hidden layers and then the output layer, and the composition of these nonlinear transforms enables flexible function approximation in multiple dimensions. In supervised regression, we supply input vectors (features) and desired outputs (targets), and the network parameters (weights and biases) are adjusted to minimize a loss function, a single scalar that quantifies prediction error over the training set.

Figure 6 – Schematic of an MLP



Source: Wang and Wu (2020).

A convenient choice of loss function for continuous targets is the mean squared error (MSE). With targets y and predictions $\hat{y}$, the empirical MSE over N samples is as demonstrated in Eq. 2.17.

$$\mathscr{L}_{\text{MSE}} = \frac{1}{N}\sum_{i=1}^{N} \left\| y^{(i)} - \hat{y}^{(i)} \right\|^2. \tag{2.17}$$

Parameters are optimized by backpropagation, which is the learning procedure that adjusts a neural network's weights and biases by using the chain rule to distribute the prediction error from the output layer back through the hidden layers. Training proceeds in two passes: a forward pass computes the model's predictions and a scalar loss that measures the mismatch to the targets; a backward pass then computes the gradients of this loss with respect to every parameter by propagating "error signals" layer by layer. These gradients tell us how each parameter should change to reduce the loss, and an optimizer (e.g., SGD or Adam) applies small updates across many epochs until the loss stops improving. In practice, early stopping on a validation set and regularization (such as

weight decay or dropout) are used to curb overfitting and promote generalization (Géron (2019)).

2.5.2 MLP for flutter-related regression

Classical flutter solvers output frequency and damping as functions of speed. The mapping from inputs (geometry, mass/stiffness surrogates, modal frequencies at a given V) to outputs (modal damping at that V) is smooth but nonlinear. An MLP is well suited to learn such relations directly from curated datasets. Once trained, inference is essentially a sequence of matrix–vector operations, enabling very fast evaluation over many speeds and configurations, which is valuable for screening designs and for integrating with uncertainty studies.

2.6 Literature Comparison

Mohammadi-Amin and Ghadiri (2013) present a supervised multilayer perceptron (MLP) metamodel that learns the mapping from geometric/structural descriptors to aeroelastic stability outcomes, with the goal of replacing repeated coupled aeroelastic analyses during preliminary design. Using a curated set of analytical and experimental cases (including legacy tunnel data) as training exemplars, the authors adopt a 14–12–6 architecture with hyperparameters chosen for stable convergence and generalization. Their inputs include span, chord, sweep, elastic axis and center-of-gravity positions, mass per unit length, bending/torsional stiffnesses EI and GJ, radii of gyration, and selected modal properties. The outputs include flutter Mach, flutter speed, flutter frequency, and divergence speed. The key finding is that, once trained, the MLP yields near-instantaneous predictions that agree closely with analytical codes and test data across systematic variations of sweep and mass ratios Mohammadi-Amin and Ghadiri (2013).

From a methodological perspective, the contribution lies in shifting the prediction target to global stability scalars (flutter speed and frequency) and in demonstrating that a purely data-driven surrogate can reproduce classical parametric trends with modest network size Mohammadi-Amin and Ghadiri (2013). The study implicitly assumes linear structural dynamics and potential-flow aerodynamics in the training labels, which is aligned with common panel-method practice in preliminary studies. The metamodel therefore inherits the validity envelope of the labels, a point the authors acknowledge when discussing future extensions.

Relative to the present dissertation, the overlap is the use of neural networks to accelerate aeroelastic inference. The divergence is in the learned observable. While Mohammadi-Amin and Ghadiri (2013) regress directly to a scalar flutter speed/frequency from static descriptors, the present work learns the *modal* damping surfaces $g_m(V)$ per

mode and infers flutter through the first zero-crossing of the most critical g_m. This choice retains diagnostic value, it is possible to inspect how each mode's damping evolves with airspeed, and integrates seamlessly with frequency-domain FE+DLM (MSC Nastran SOL 145/PKNL) for label generation and with experimental modal identification for model updating. In short, both pipelines target rapid, parametric exploration, but the present approach provides mode-resolved interpretability while still delivering a scalar flutter estimate at negligible inference cost.

In contrast to panel-based linear aerodynamics, Ramos and Ribeiro (2018) implement a bidirectional fluid–structure interaction (FSI) scheme to identify flutter on the AGARD 445.6 standard wing. By coupling a finite-element structural model to an unsteady Reynolds-averaged Navier–Stokes (URANS) solver with turbulence modeling, the authors advance in time while increasing airspeed and extract stability trends from the temporal response, identifying modal coalescence and damping decay consistent with flutter onset. Comparisons against the canonical AGARD wind-tunnel measurements compiled by Yates E. C. (1988) show an agreement in subsonic/transonic regimes, though the authors discuss rising uncertainty and cost at higher Mach numbers, an expected trade-off in two-way FSI. Their study also underscores practical aspects often absent from reduced-order pipelines: mesh motion, structural–aerodynamic data exchange, and the sensitivity of results to turbulence modeling and coupling parameters.

The present work occupies a complementary niche. It leverages a linear frequency-domain chain, with the FE structure and Doublet-Lattice Method within MSC Nastran SOL 145/PKNL, to compute V–g and V–f surfaces efficiently and then trains an MLP to predict $g_m(V)$ without resorting to time-accurate CFD. This is appropriate for preliminary design and for controlled parametric sweeps, and it readily accommodates hybridization with experimental modal testing and selected tunnel points. In the broader literature, independent AGARD 445.6 studies (e.g., FOI — Swedish Defence Research Agency (2007)) reinforce the value of that configuration as a benchmark: it is complex enough to stress modeling assumptions yet sufficiently well documented to anchor validation. Thus, FSI provides the high-fidelity reference when viscous and nonlinear aerodynamic effects are paramount, whereas the present surrogate preserves speed and interpretability over the parameter space of interest.

Mohammed (2017) analyzes three high-aspect-ratio wings (two sailplanes and a transport configuration) using CALFUN, a solver based on the Dynamic Stiffness Method (DSM) and the Wittrick–Williams algorithm for eigenvalue counting. The aerodynamic loads assume classical unsteady potential theory of Theodorsen (1934). After validating natural frequencies and modes, the author conducts a clean parametric study varying bending and torsional rigidity. A central outcome is the dominant influence of torsional stiffness GJ on flutter speed (order of $\geq$9% change for $\pm$10% variation in GJ), in contrast to

much smaller sensitivity to bending rigidity EI (sub-percent changes) across the examined cases Mohammed (2017). The dissertation further notes when geometric nonlinearity and large deflections challenge linear flutter predictions for very slender, flexible wings.

For the present work, two implications are immediate. First, the experimental/simulation program that varies semi-span in a family of rectangular semi-wings naturally modulates effective torsional flexibility. The learned $g_m(V)$ surfaces capture the expected reduction of critical speed with increasing flexibility, aligning with the sensitivity trends reported by Mohammed (2017). Second, future dataset expansions could explicitly sweep GJ (via thickness, spar geometry, or materials) so that the learned mapping generalizes across torsional design spaces. The broader DSM literature surveyed by Banerjee (2019) supports this emphasis on torsion for slender lifting surfaces and provides an analytical framework should additional high-fidelity label generation be desired.

Guo *et al.* (2025) build a back-propagation network trained with Levenberg–Marquardt to predict flutter velocity for highly flexible wings spanning multiple materials (e.g., Al 7050, 15-5PH stainless steel, Ti–6Al–4V) and geometric downscaling ratios. Their input vector aggregates geometric dimensions, structural properties, and parameters that characterize the flutter condition, whereas the output is a single scalar U_f. The authors report good accuracy over a suite of simulated cases and discuss network-depth/width choices, highlighting data sufficiency and sensitivity to the composition of training features. An interesting observation is that prediction difficulty can concentrate around low-order modes and near coalescence regions, where small modeling differences substantially affect the inferred flutter point.

Conceptually, both the present work and Guo *et al.* (2025) share the philosophy of fast aeroelastic prediction from data. The approaches differ chiefly in target definition and interpretability: Guo *et al.* (2025) learn U_f directly, whereas this dissertation learns $g_m(V)$ and infers U_f by zero-crossing of the most critical mode. The latter choice preserves a direct link to the mode-by-mode stability picture, which can be advantageous for design diagnosis and for fusing simulation with experimental damping estimates (e.g., from half-power bandwidth). The present workflow also anchors labels in an industrial solver (MSC Nastran SOL 145/PKNL with DLM) and uses experimental modal testing and wind-tunnel checks to bound modeling error, complementing the plate-based simulation dataset used by Guo *et al.* (2025).

CHAPTER

3

METHODOLOGY

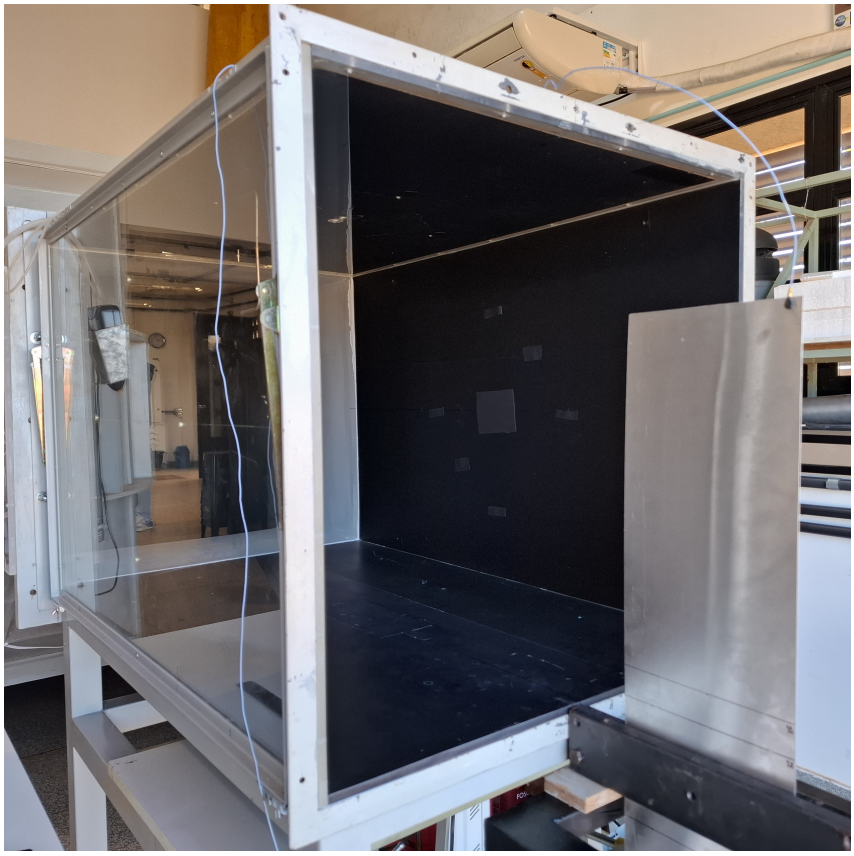
3.1 Semi-Span Wing Model Description

This work uses flat-plate semi-span wing designed to support both repeatable numerical flutter analyses and wind-tunnel testing. The model is deliberately simple, a rectangular planform with constant chord, so that changes in span directly modulate bending and torsional flexibility while keeping geometry and mounting consistent. The semi-span is bolted to a rigid root adapter that enforces a clamped boundary condition representative of a fuselage attachment.

3.1.1 Geometry and structure

The wing planform is rectangular with a constant chord of $c = 0.18$ m and zero sweep and dihedral. A family of semi-span lengths was studied to vary the flexibility and aeroelastic coupling. The structural representation is a flat plate of uniform thickness $t \approx 0.8$ mm, built from an isotropic material characterized by Young's modulus $E \approx 65$ GPa, Poisson's ratio $\nu = 0.30$, and mass density $\rho \approx 2780$ kg/m^3. The root boundary is enforced as fully clamped by constraining all six degrees of freedom on the root-grid set. The experimental mounting and positioning for collecting all the data is shown in Fig. 7.

Figure 7 – Semi-span wing view and mounting



Source – Elaborated by the author

Table 2 lists the span set used consistently in both simulations and experiments. All other geometric and material properties remained fixed across configurations, and only the semi-span length S was varied to produce systematic changes in stiffness and modal frequencies.

Table 2 – Semi-span wing configurations used throughout this study

Semi-span S [m]	**Planform [m]**	**Notes**
0.35	Rectangular, $c = 0.18$	Baseline, high stiffness
0.38	Rectangular, $c = 0.18$	Intermediate small semi-span
0.41	Rectangular, $c = 0.18$	Intermediate medium semi-span
0.44	Rectangular, $c = 0.18$	Intermediate medium semi-span
0.47	Rectangular, $c = 0.18$	Intermediate high semi-span
0.50	Rectangular, $c = 0.18$	Higher semi-span, most flexible of the set

Source – Elaborated by the author

3.2 Experimental Modal Evaluation

This stage was conducted first to anchor the numerical model in measured dynamics. For each semi–span configuration, vibration responses to a light, localized impact were acquired and the dominant natural frequencies (and basic mode indicators). These experimental modal quantities were then used to update the FE model before running the flutter simulations.

A lightweight accelerometer was affixed at a selected position along the span, to capture bending and torsional content. An instant excitation (impact) was applied within a single point near the tip. The mass of sensors and cabling was kept minimal to avoid shifting the low–order frequencies. For the purpose of FE updating, peak–picking in the FFT spectra proved sufficient to identify the first 4 modes frequencies for each span.

Table 3 – Experimental modal frequencies obtained

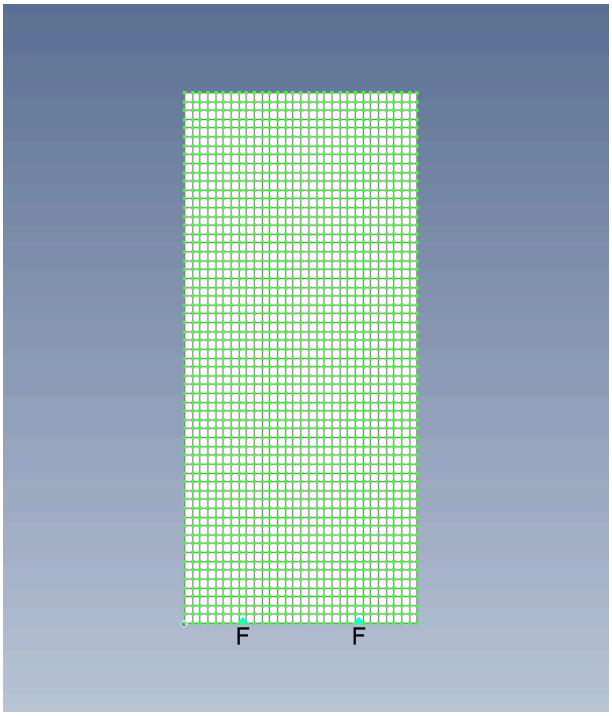
Semi-span (m)	Mode 1 (Hz)	Mode 2 (Hz)	Mode 3 (Hz)	Mode 4 (Hz)
0.35	4.2	19.7	29.9	62.5
0.38	3.9	17.8	25.0	57.9
0.41	3.2	16.5	21.9	52.3
0.44	2.8	14.5	18.5	46.5
0.47	2.3	13.3	15.5	41.4
0.50	2.0	12.9	14.1	38.9

Source – Elaborated by the author

3.3 Finite Element Modeling and Flutter Simulation

The wing is modeled as a flat rectangular plate with constant chord $c = 0.18$ m and semi-span $S \in \{0.35, 0.38, 0.41, 0.44, 0.47, 0.50\}$ m, clamped at the root. The skin is meshed primarily with four-node shell elements (CQUAD4) using a PSHELL property of uniform thickness $t = 0.0008$ m. Material is defined with MAT1 (E, ν, ρ) and updated to match experimental modal frequencies prior to flutter runs (see Sec. 3.1.1 for mechanical properties and material values used after calibration). The root boundary condition is enforced as fully fixed root nodes.

Figure 8 – Structural mesh and fixed root boundary in FEMAP (example semi-span 0.41 m)



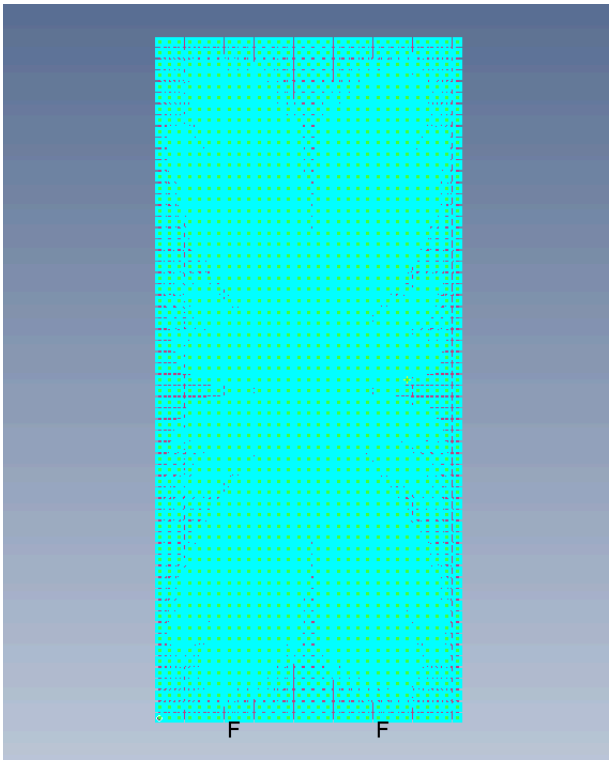
Source – Elaborated by the author

As previously mentioned, the unsteady subsonic aerodynamics are represented with the Doublet-Lattice Method (DLM). The lifting surface is defined with CAERO1 entries covering the planform of the semi-span. The semichord is $b = c/2$. Structural-to-aerodynamic coupling is performed with surface splines (SPLINE1), mapping the structural grid on the plate to the CAERO1 boxes. The aerodynamic control entries (AERO, MKAERO1) define the flow condition grid (Mach numbers) at which the generalized aerodynamic forces (GAFs) are evaluated. Reduced frequency follows the standard definition, explicited in Eq. 3.1,

$$k = \frac{\omega b}{U} \tag{3.1}$$

where ω is the circular frequency, b is the semichord, and U is the flow speed.

Figure 9 – DLM panel layout (CAERO1) and SPLINE1 coupling to the structural mesh



Source – Elaborated by the author

Flutter was computed with MSC Nastran solution sequence SOL 145. The modal basis supplied to the flutter assembly was extracted with EIGRL to include the first few modes that dominate the aeroelastic response (in this study, the first four modes were tracked consistently across spans, covering the first bending and the first torsion). The FLUTTER case control used the PKNL method ("PK, no loop") to evaluate stability at discrete, pre-specified flight points without internal iteration on frequency. This choice yields fast, monotonic V–g/V–f sweeps well suited to dataset generation. The flow conditions matched the wind-tunnel regime. For each span, a uniform grid of speeds was prescribed to bracket the first zero crossing of modal damping, and the step size was chosen small enough to resolve the sign change.

For each flight point and tracked mode, SOL 145 returns the complex aeroelastic eigenvalue $p = \sigma + i\omega$. The modal damping measure $g = -\sigma/\omega$ and the frequency $f = \omega/(2\pi)$ are assembled into V–g and V–f curves. The flutter speed U_f is identified where $g(V)$ first crosses zero (from nonpositive to positive) for any mode, and the corresponding $f(V)$ at that speed is the flutter frequency.

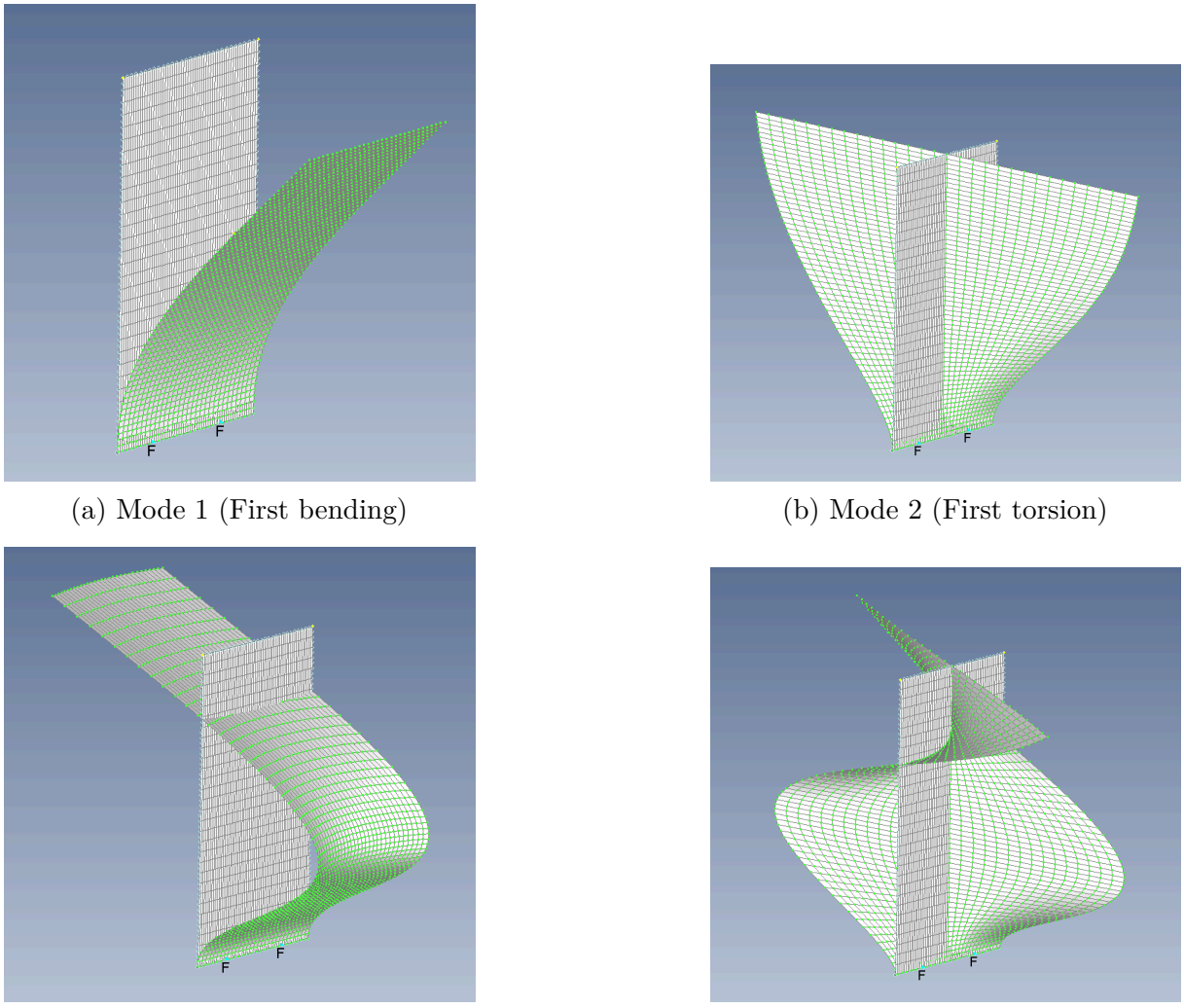
Table 4 – MSC Nastran SOL 145 flutter analysis settings (applied to all semi-span cases).

Parameter	**Setting**	**Notes**
Solver	SOL 145 (Flutter)	–
Method	PKNL (no frequency iteration)	Eigenvalues computed at each prescribed flight point
Structural model	Half-wing FE model; clamped root	Wall acts as symmetry; same workflow for all spans
Aerodynamic model	Doublet-Lattice Method (CAERO1/PAERO1)	Subsonic unsteady aerodynamics
Coupling	SPLINE1 surface spline	Transfers aerodynamic pressures to the structural grid
AERO reference	Global rectangular CSYS; $V_{\text{ref}} = 30$ m/s; ref. length $c_{\text{ref}} = 0.18$ m; ref. density $\rho_{\text{ref}} = 1.225$ kg/m^3	Matches tunnel conditions
Flight conditions	Density ratio $= 1.0$; Mach $= 0.08746$ (constant)	–
Velocity sweep	$V = \{5, 7, 9, \ldots, 57\}$ m/s (step 2 m/s)	Same grid for all spans
Modal basis (EIGRL)	First 4 modes (bending and torsion included)	Modes tracked consistently across spans
Outputs	$g(V)$ and $f(V)$ per mode	g from real part of eigenvalue; f from imaginary part

Source – Elaborated by the author

Only the semi-span is modeled, and the wall/adapter acts as a rigid clamp (no symmetry plane is imposed in the aerodynamics). The plate representation neglects small geometric features not essential to the targeted modes. Material properties are those obtained after the experimental–numerical update described earlier, and any added sensor mass during tunnel tests is excluded from the simulation baselines. Sensitivity checks (mesh refinement, number of CAERO boxes, spline support) were performed to ensure stable V–g/V–f trends across spans.

Figure 10 – Mode shape illustration (0.41 m semi-span example)



(a) Mode 1 (First bending)

(b) Mode 2 (First torsion)

(c) Mode 3 (Second bending)

(d) Mode 4 (Second torsion)

Source – Elaborated by the author

The output files data were extracted, for each span and velocity, gathering the modal damping and frequency of the first four modes. These data populate the V–g and V–f diagrams used to read U_f and serve as inputs/targets for the learning pipeline in later sections. Identical procedures were applied across all spans $S \in [0.35, 0.50]$ m in steps of 0.03 m, ensuring consistent simulation datasets for comparison with wind-tunnel measurements.

3.4 Wind Tunnel Experimental Procedure

For the flutter experimental tests, the operational envelope of the wind tunnel was limited to approximately 25 m/s, so then experimental campaigns focused on the semi–span configurations whose simulated flutter speeds fell within this range. The semi–span

model was a single flat plate whose effective span was changed by sliding the plate up or down in the root clamp, thus altering the fixed length while keeping the planform unchanged. The rig was installed immediately downstream of the tunnel exit, in the potential–core region of the jet to ensure uniform inflow. The semi–span was mounted vertically and clamped rigidly at the root to a fixed frame attached to the tunnel exit, and the wall provided an effective symmetry plane consistent with the numerical model.

For experimental validation, these wind–tunnel data were collected only for the semi–spans $S = 0.41, 0.44$, and $0.47\,\mathrm{m}$. This selection follows directly from the SOL 145 predictions: these three configurations had estimated flutter speeds within the tunnel's operational envelope (approximately $25\,\mathrm{m/s}$), which made it feasible to approach the near–critical region safely and to obtain V–g and V–f points sufficiently close to the $g = 0$ crossing. Restricting tests to these spans improves the consistency and coherence of the three–way comparison (simulation, artificial neural network prediction, and experiment), because the experimental curves for $S = 0.41, 0.44$, and $0.47\,\mathrm{m}$ actually sample the portion of the stability map that governs the onset. Conversely, attempting to test spans whose simulated flutter lies outside the facility envelope would yield measurements far from the critical neighborhood, providing little information about the true crossing and, therefore, an inadequate basis for comparing with numerical results or the artificial neural network's inferred U_f. Focusing the campaign where the onset is experimentally accessible thus reduces risk and ensures that the validation targets the physically relevant regime.

One accelerometer was bonded near the upper tip, close to the trailing edge, for all runs so as to capture both bending and torsional content. Signals were acquired at 2048 samples/s for 10 s at every condition. The same sensor location and acquisition settings were used for all spans and speeds. Airspeed was increased in steps of 2 m/s, starting at 5 m/s. The critical flutter speed in testing is identified when the effective damping approaches zero and sustained growth is observed. Quantitative values and comparisons appear in Chapter 4.

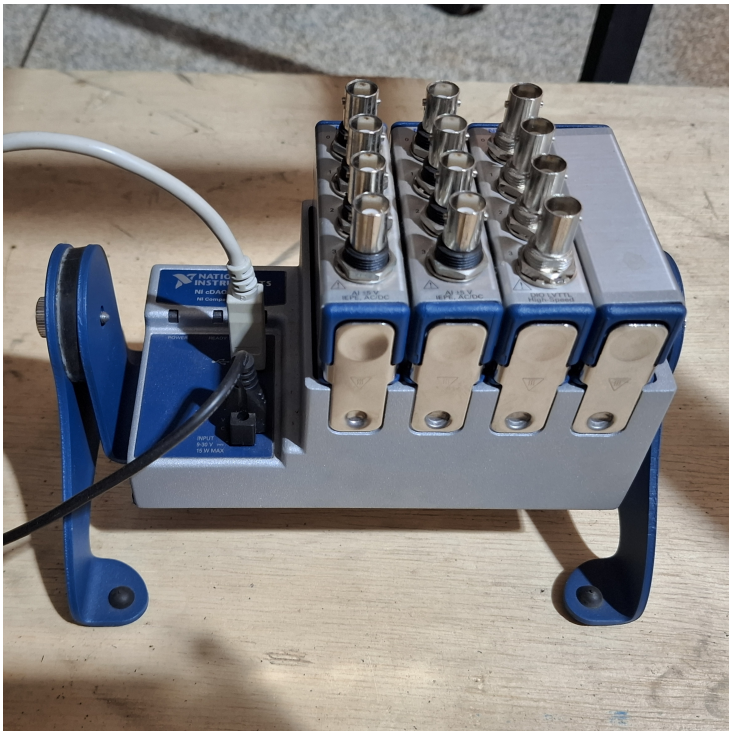
Figure 11 – Semi-span wing experimental mounting and positioning



Source – Elaborated by the author

The acquisition system consisted of a four–slot USB chassis NI cDAQ-9174 equipped with an NI 9215 BNC module for analog input. The NI 9215 provides four differential channels with 16-bit A/D conversion and BNC connectors, suitable for low-frequency modal measurements with good amplitude resolution. As already mentioned, all channels were synchronously sampled at 2048 samples/s for 10 s per operating point, with records triggered via software and stored for post-processing. Short coaxial cables with proper grounding/shielding were used to minimize noise and ensure signal integrity.

Figure 12 – Acquisition system



Source – Elaborated by the author

The vibration sensor employed was a PCB Piezotronics piezoelectric accelerometer (model 352A59), IEPE type (voltage output), appropriate for structural modal testing. The transducer was mounted lightly and rigidly near the semi-span tip, close to the trailing edge, oriented to primarily capture out-of-plane accelerations of the plate and, by positioning, also sensitive to torsional content. In addition to the mounting details, the following specifications were observed from the factory calibration sheet and datasheet: nominal sensitivity $S = 10$ mV/g, measurement range ± 500 g, broadband noise density $\approx$ 0.05 mg/$\sqrt{\text{Hz}}$ (typical, inferred from 0.005 g_{RMS} over 1–10,000 Hz), transverse sensitivity $\leq 5\%$, sensor mass 0.9 g, mounted resonance frequency ≥ 65 kHz with a usable flat bandwidth well above the highest modal frequency measured in this work, and operating temperature range -54 a $+121$ °C. As an IEPE device, it was powered with a constant-current source (ICP®) in the range of 2–20 mA and a compliance voltage of 18–30 V, yielding an output bias of approximately 7–12 V. The same position, orientation, and measurement chain were maintained across all span and velocity combinations to ensure repeatability among tests.

Data were processed with a MATLAB pipeline to extract, for each mode, the damping coefficient g (SOL 145 convention) and the natural frequency f_n, and then to assemble the experimental $V-g$ and $V-f$ diagrams. Each record is cleaned (NaN/Inf removal), de-trended, and band-pass filtered in a broad band using a zero-phase Butterworth filter.

For mode $m = 1 \ldots N$ the script uses a seed frequency f_0 (from the span–specific table measured - Table 3) and designs a narrow band–pass window around f_0 (initially $\pm 25\%$). If energy or numerical conditioning is insufficient, the window is expanded progressively. On the filtered signal $x_f(t)$, an auto-regressive model of order 2 (AR2) is identified (Burg, and if it fails, a Yule–Walker estimate with ridge regularization is used). From the AR(2) polynomial roots $z = re^{i\theta}$, the continuous–time decay rate and damped frequency are recovered as shown in Eq 3.2 and Eq. 3.3.

$$\sigma = \frac{\ln r}{\Delta t} \tag{3.2}$$

$$\omega_d = \frac{\theta}{\Delta t} \tag{3.3}$$

The damping parameter is obtained through Eq. 3.4.

$$g = \frac{\sigma}{\omega_d} \tag{3.4}$$

The outputs stored are g (stable if $g < 0$) and $f_d = \omega_d / 2\pi$. This is repeated for all speeds and the first N modes. If AR(2) fails for a mode/speed, a Welch PSD is computed, around the peak near f_0 the -3 dB points f_1, f_2 are located and $\zeta \approx (f_2 - f_1)/(2f_d)$ provides $g = -\frac{\zeta}{\sqrt{1-\zeta^2}}$.

For each mode, $g(V)$ is fit by polynomials of degree 1–4 using a polyfit with centering/scaling. The degree is chosen by the highest R^2 (ties resolved by lower degree). Choosing by R^2 is appropriate here because R^2 measures the fraction of the sample variance of g explained by the model, providing a scale-invariant criterion to compare fits across degrees and velocity ranges. Moreover, the model with the largest R^2 generally captures the trend with minimal residuals. To limit overfitting, the polynomial degree is capped and ties favor the simpler model.

3.5 Data Preparation and Feature Engineering for the Neural Network

This subsection explains how the simulation outputs (and later the experimental points) were transformed into a machine–learning dataset suitable for training the MLP model that predicts modal damping values.

From each SOL 145/PKNL run (Section 3.1.3), for every semi–span S and for each prescribed speed, the first $N = 4$ modes were tracked, yielding for every (S, V) pair

the set: $\{f_1, f_2, f_3, f_4\}$ and $\{g_1, g_2, g_3, g_4\}$, where f_m is the modal frequency (Hz) and $g_m = -\sigma/\omega$ is the SOL 145 damping measure of mode m. Each dataset row is thus:

$$x = [S,\, V,\, f_1, \ldots, f_4] \;\mapsto\; y = [g_1, \ldots, g_4],$$

i.e., features are span, speed, and modal frequencies, and targets are the corresponding modal dampings. This multi–output arrangement keeps the inter–mode correlations available to the regressor and mirrors how flutter solvers report results mode–by–mode. Simulation tables were consolidated into a single frame (one row per (S,V)), sorted by (S,V), and de–duplicated. Units were kept consistent (m, m/s, Hz). Rows with missing or non–finite values were discarded (this did not affect the regular grid produced by SOL 145).

Only simulation data were used to train and tune the network. A held–out test subset (∼10% of the samples) was separated first, and the remaining data were split into training and validation subsets (stratified by span) to monitor generalization during model selection. The experimental measurements collected in the wind tunnel (Sec. 3.1.4) were not used for training, and where reserved for independent validation and comparison in Chapter 4.

To stabilize optimization and balance feature scales, a standard score (z–score) transform was applied feature–wise on the training split and then reused for validation/test:

$$x'_j \;=\; \frac{x_j - \mu_j}{\sigma_j}, \tag{3.5}$$

with (μ_j, σ_j) computed on training data only. Because the network predicts multiple dampings at once, the same transform was also applied target–wise.

$$g'_m \;=\; \frac{g_m - \mu_m^{(g)}}{\sigma_m^{(g)}}, \qquad m = 1, \ldots, 4, \tag{3.6}$$

and later inverted to report metrics in physical units.

To increase sample diversity without altering the underlying physics, a light augmentation was applied to the *training* split:

- small jitter in V around each recorded speed (within a fraction of the global standard deviation of V);
- for each mode, $f_m(V)$ and $g_m(V)$ at the jittered speed were obtained by linear interpolation (with linear extrapolation only at the edges) along the original (V, f_m) and (V, g_m) curves of the same span;

- small mode–dependent Gaussian noise was added to f_m and g_m, scaled by local residual spreads so that higher–curvature regions receive proportionally less perturbation;

This preserves smoothness across V and avoids generating nonphysical sign changes far from the flutter region.

The network's primary outputs are the modal dampings $[g_1, \ldots, g_4]$ at the queried $(S, V, f_1, \ldots, f_4)$. The critical flutter speed for a given span is then obtained *post hoc* by evaluating the trained network over a dense speed grid and finding, for each mode, the first zero of $g_m(V)$ (from $g \leq 0$ to $g \geq 0$). The span's flutter speed is the minimum such crossing among the tracked modes. This mirrors the engineering practice used with V–g curves (and aligns the ML output with the SOL 145 definition).

The prepared dataset pairs $(S, V, f_1, \ldots, f_4)$ with $(g_1, \ldots, g_4)$ over a uniform speed grid for each span, using standardized features/targets, physics-aware augmentation on the training split, and strict separation of experimental measurements for independent validation.

Table 5 – First lines (samples) of the constructed dataframe

span	velocity	freq_1	freq_2	freq_3	freq_4	damp_1	damp_2	damp_3	damp_4
0.35	5.0	4.646770	19.553400	29.065100	64.640600	-0.079577	-0.020399	-0.010700	-0.005023
0.35	7.0	4.637790	19.398900	29.034800	64.493200	-0.115584	-0.028960	-0.015199	-0.007328
0.35	9.0	4.623800	19.189100	29.044400	64.358000	-0.154246	-0.037595	-0.019721	-0.009635
0.35	11.0	4.604530	18.920300	29.057600	64.235300	-0.196467	-0.046228	-0.024279	-0.011942
0.35	13.0	4.579210	18.588300	29.073800	64.125100	-0.243536	-0.054696	-0.028866	-0.014244
...	...	...	...	...	...	...	...	...	...

Source – Elaborated by the author

3.6 Neural Network Architecture and Training

As previously presented, the regression model is a fully connected multi–layer perceptron (MLP) implemented in Python with Keras/TensorFlow. Each input vector contains the standardized features $[S, V, f_1, f_2, f_3, f_4]$, and the network outputs the four standardized modal damping values $[g_1, g_2, g_3, g_4]$ at that operating point. Hidden layers apply affine transforms followed by a LeakyReLU nonlinearity. The output layer is linear to represent continuous targets. After a series of manual trials comparing depths, widths, and activations, the final architecture adopted three hidden layers with widths of 480, 176, 480 neurons, respectively, which provided stable training and good generalization without unnecessary complexity for the dataset size. Dropout is inserted after each hidden layer and ℓ_2 penalties are applied to the dense weights, and both acted as the main regularizers during the manual tuning.

To encode a simple physical prior, a differentiable adjustment is applied to the raw outputs inside the network so that damping remains strictly negative below a fixed threshold speed $V_0 = 10$ m/s. Let $\mathbf{g} = (g_1, \ldots, g_4)$ be the raw predictions in physical units (obtained by inverting the target standardization within the layer). The adjusted outputs $\tilde{g}_m$ are defined by Eq. 3.7.

$$\tilde{g}_m(V) = \begin{cases} -|g_m(V)| - \varepsilon, & V \le V_0, \\ g_m(V), & V > V_0, \end{cases} \qquad m = 1, \ldots, 4, \tag{3.7}$$

with a small margin $\varepsilon > 0$ to keep values strictly negative. The adjusted outputs are then re-standardized so that training remains numerically well-conditioned. Training minimizes the mean–squared error (MSE) between standardized predictions and targets, augmented by ℓ_2 penalties on the weights.

All inputs and targets are standardized using statistics computed on the training split only, and the same transforms are applied to validation and test sets. Optimization uses Adam, a learning rate of 10^{-3} was found to work robustly in preliminary sweeps and was reduced automatically on plateaus during the final runs. Models were trained for up to 10000 epochs with early stopping based on validation MSE (patience sufficiently large to avoid premature termination), mini–batch size 128, and dropout rates between 0.10 and 0.40 chosen by manual ablation.

Manual hyperparameter selection proceeded iteratively, starting from a shallow baseline, depths from two to four layers, widths from 32 to 512 neurons, LeakyReLU slopes in $[0.01, 0.30]$, and several dropout and ℓ_2 levels were tested. The final configuration above was chosen because it consistently yielded the lowest validation MSE while maintaining smooth $V-g$ curves and stable extrapolation near the flutter crossing. The checkpoint with the minimum validation loss is retained as the final model, which is then evaluated on the held–out simulation test set.

Table 6 – Neural network training parameters (final configuration).

Parameter	**Setting**
Inputs	$\{S, V, f_1, f_2, f_3, f_4\}$ (standardized)
Outputs	$\{g_1, g_2, g_3, g_4\}$ (standardized; linear activation)
Hidden layers	3 (Dense: 480, 176, 480 neurons)
Activations	LeakyReLU in hidden layers (slope 0.10)
Dropout	0.35 after each hidden layer
Regularization	ℓ_2 weight decay 3.81×10^{-6} on Dense weights
Optimizer	Adam
Learning rate	7.28×10^{-4}
Loss	mean squared error (MSE)
Batch size	128
Epochs	up to 10,000 with early stopping on validation MSE
LR scheduling	reduce-on-plateau (validation loss)
Preprocessing	standardization fit on training split (inputs and targets)
Physics prior	clamp $g(V) < 0$ for $V \leq V_0$ inside the model head

Source – Elaborated by the author

Finally, the function `predict_flutter_speeds` converts network outputs into a predicted flutter speed for each semi–span. It first builds a common velocity grid from the original simulation dataframe and assembles a frequency cube $F(\text{span}, V, \text{mode})$ by interpolation along V so that all spans share the same velocity samples. For spans not present in the dataframe, the frequencies are interpolated between neighboring spans at each velocity. For each queried span, the routine forms input rows $X = [S,\ V,\ f_1(V), \ldots, f_N(V)]$ across the grid, applies the same feature scaling used during training, and calls the trained MLP to obtain predicted modal dampings $\widehat{g}_m(V)$ in physical units via the inverse target scaling. The critical airspeed is then identified as the first zero–crossing of the predicted damping curve for any tracked mode, consistent with the classical flutter criterion.

$$g_m(V_f) = 0 \quad \text{with} \quad g_m(V) < 0 \text{ for } V < V_f. \tag{3.8}$$

Numerically, for each mode m the algorithm finds the earliest adjacent pair (V_k, V_{k+1}) such that $\widehat{g}_m(V_k) < 0$ and $\widehat{g}_m(V_{k+1}) \geq 0$, and computes an interpolated root

$$V_{f,m} \;=\; V_k + \frac{0 - \widehat{g}_m(V_k)}{\widehat{g}_m(V_{k+1}) - \widehat{g}_m(V_k)} \left(V_{k+1} - V_k\right), \tag{3.9}$$

returning as the span's flutter speed the minimum finite $V_{f,m}$ across modes. If no zero–crossing is present for a mode on the sampled grid, that mode is ignored for the minimum. If none of the modes cross zero, the routine reports `NaN` for that span. This post–processing

ensures that the network's continuous damping predictions are mapped to a single scalar flutter–speed estimate in the same way that V–g curves are interpreted in classical flutter analysis.

3.7 Neural Model Evaluation Strategy and Performance Metrics

The trained network is evaluated in two complementary ways: prediction accuracy for modal damping over the held–out simulation set and fidelity of the derived flutter–speed estimate. All metrics are computed in physical units (after inverse–scaling), while the training and validation curves are monitored in the standardized space, as shown later in the loss plot. Let $g_{i,m}$ denote the true modal damping for sample i and mode m, and $\widehat{g}_{i,m}$ the corresponding network prediction; the index i runs over all span–velocity test points and $m = 1, \ldots, N_{\text{modes}}$.

The mean–squared error (MSE) over the test set aggregates samples and modes:

$$\text{MSE} \;=\; \frac{1}{N}\sum_{i,m}\left(\widehat{g}_{i,m} - g_{i,m}\right)^2, \tag{3.10}$$

with N the total number of evaluated pairs (i,m). Reporting the square–root gives the root–mean–squared error (RMSE).

$$\text{RMSE} \;=\; \sqrt{\text{MSE}}, \tag{3.11}$$

and the mean–absolute error (MAE) complements it with a linear scale.

$$\text{MAE} \;=\; \frac{1}{N}\sum_{i,m}\left|\widehat{g}_{i,m} - g_{i,m}\right|. \tag{3.12}$$

In addition to the overall scores, the same formulas are applied per mode to reveal how accuracy distributes across the first bending/torsion families. Because stability hinges on the sign of damping at low speed, we also report a sign–consistency metric for a user–defined threshold V_0 (here $V_0 = 8\,\text{m/s}$). Let $\mathscr{I}_{\text{low}}$ be the subset of test samples with $V \le V_0$, the sign accuracy is shown in Eq. 3.13.

$$\text{Acc}_{\text{sign}} \;=\; \frac{1}{|\mathscr{I}_{\text{low}}|}\sum_{(i,m)\in\mathscr{I}_{\text{low}}} \mathbf{1}\left[\text{sign}\left(\widehat{g}_{i,m}\right) = \text{sign}\left(g_{i,m}\right)\right]. \tag{3.13}$$

CHAPTER

4

RESULTS AND ANALYSIS

4.1 Simulation Results and Experimental Data Comparison

This section confronts the SOL 145/PKNL simulations with wind-tunnel measurements for the semi–spans actually tested in the tunnel, namely $S \in \{0.41, 0.44, 0.47\}\,\mathrm{m}$. For each configuration the simulation delivered, at discrete speeds, the complex aeroelastic roots $p = \sigma + i\omega$ and the derived modal damping and frequency curves $g(V) = -\sigma/\omega$ and $f(V) = \omega/2\pi$. Experimentally, accelerometer signals acquired at $2048\,\mathrm{Hz}$ for $10\,\mathrm{s}$ were processed with the AR(2) pipeline of the previously chapter to estimate, at each tested speed, the same observables $g(V)$ and $f(V)$ for the first four modes. The flutter speed U_f is identified consistently as the first speed at which the modal damping of any tracked mode crosses zero.

At low speeds, the simulated natural frequencies agree well with the experimental modal seeds used to update the FE model (Sec. 3.1.2), and the trends of $f(V)$ remain close throughout the subcritical range. In all three tested spans the simulation predicts that Mode 2 (first torsion) is the leading instability, with Mode 1 (first bending) remaining stable up to U_f. The experimental $V-g$ points, obtained by AR(2) at each speed increment, follow the simulated curves closely and bracket the zero crossing within the tunnel's $2\,\mathrm{m/s}$ step.

To quantify the differences we report absolute and relative errors for the flutter speed. Given a experimental reference $U_{f_{exp}}$ and an simulated estimate $U_{f_{sim}}$, the relative error in percent is obtained by Eq. 4.1.

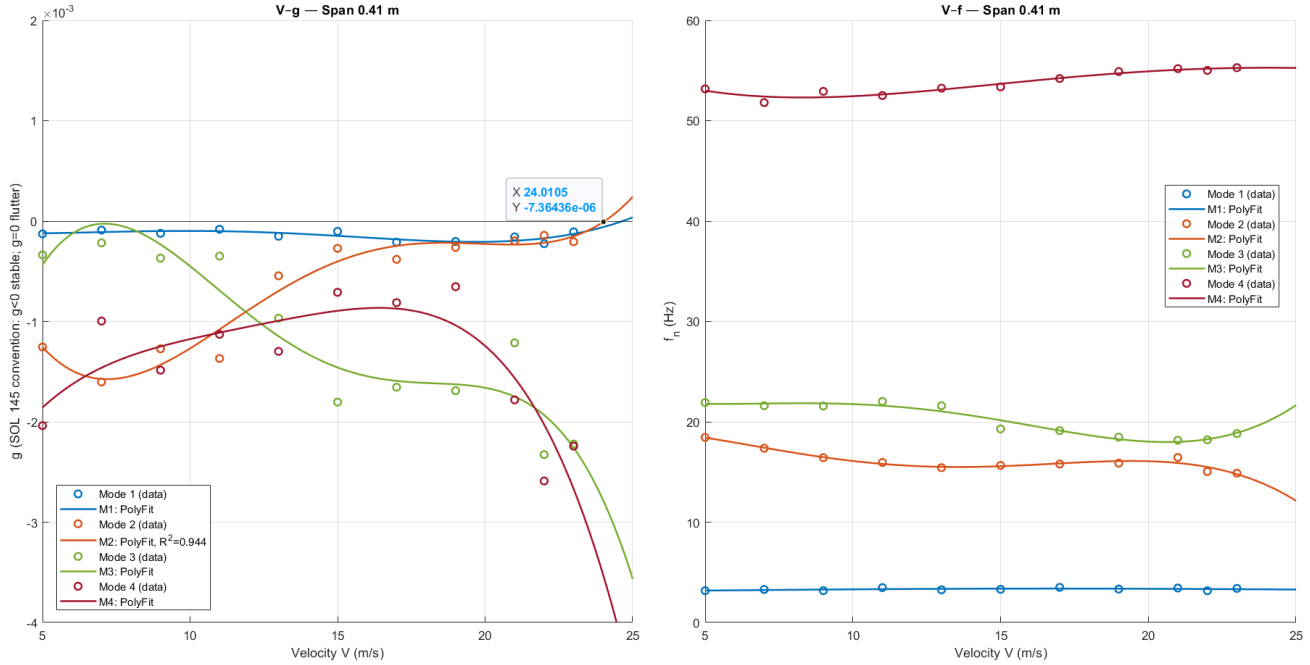
$$\Delta U_f\ [\%] \;=\; 100\,\frac{|U_{f_{sim}} - U_{f_{exp}}|}{U_{f_{exp}}}. \tag{4.1}$$

Because the experiment advances airspeed in finite increments ($\Delta V = 2\,\mathrm{m/s}$), $U_{f_{exp}}$

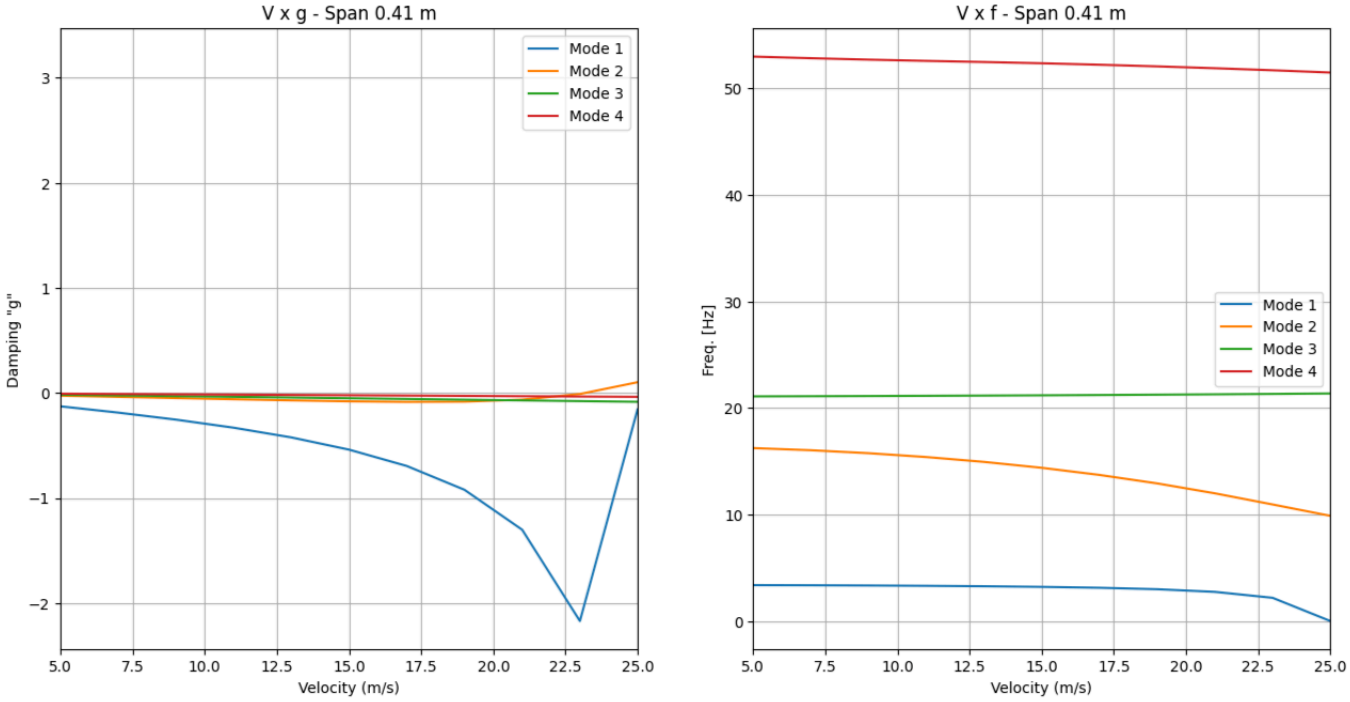
is taken as the first speed where the estimated damping becomes nonnegative.

Figures 13, 14 and 15 compare the simulated V-g-f diagrams for $S = 0.41\,\mathrm{m}$, $S = 0.44\,\mathrm{m}$ and $S = 0.47\,\mathrm{m}$ with the experimental ones.

Figure 13 – Experimental and Simulation V-g-f Comparison (S = 0.41 m)



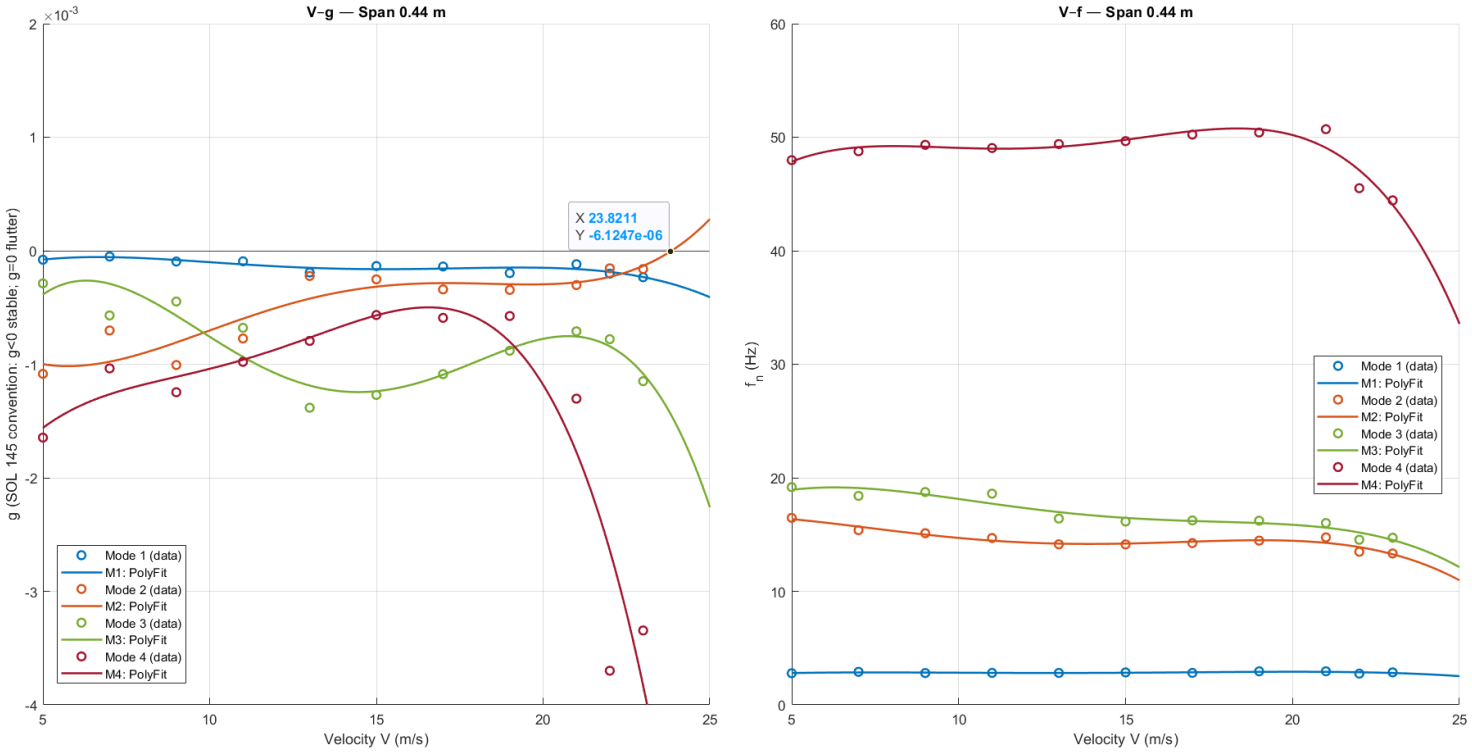
(a) Experimental V-g-f obtained



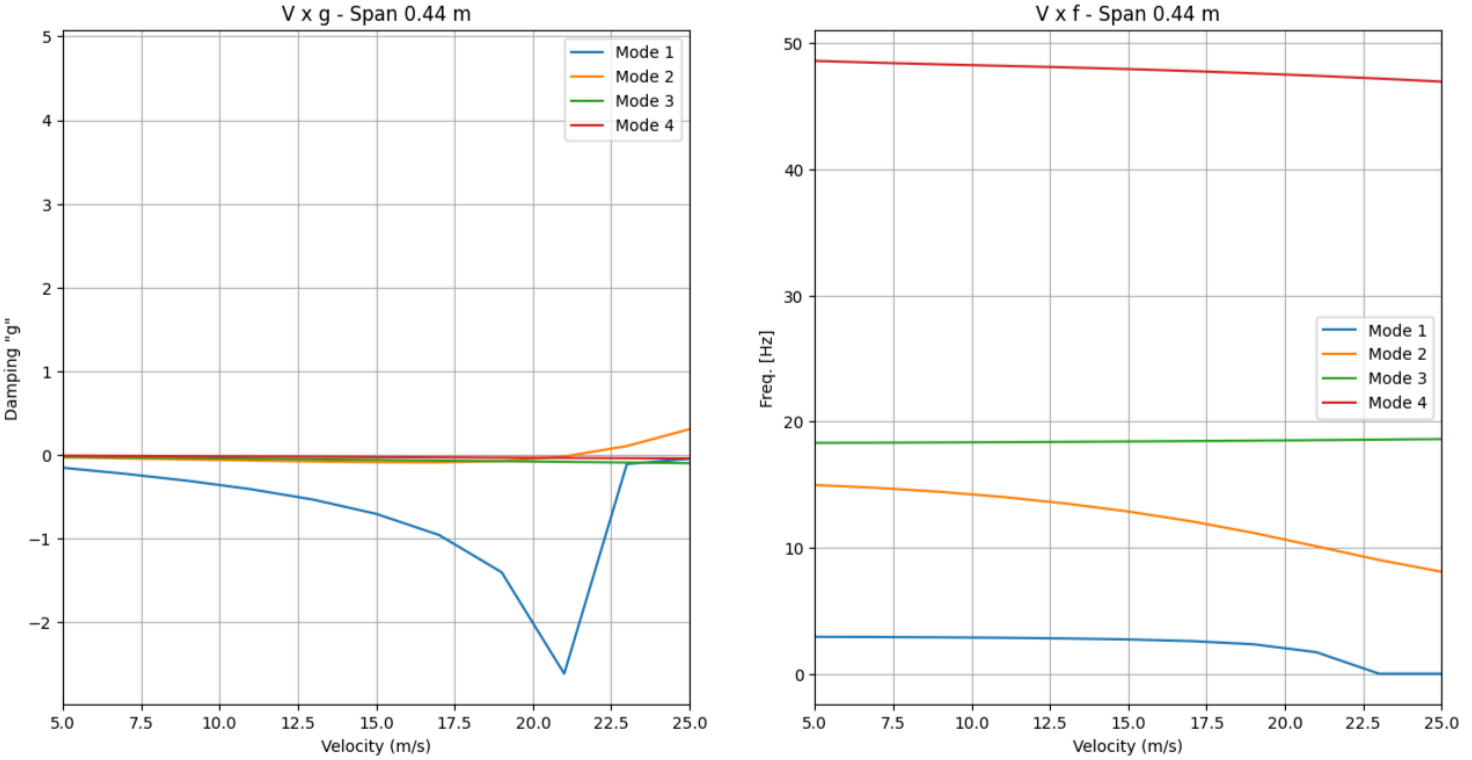
(b) Simulation V-g-f obtained

Source – Elaborated by the author

Figure 14 – Experimental and Simulation V-g-f Comparison (S = 0.44 m)



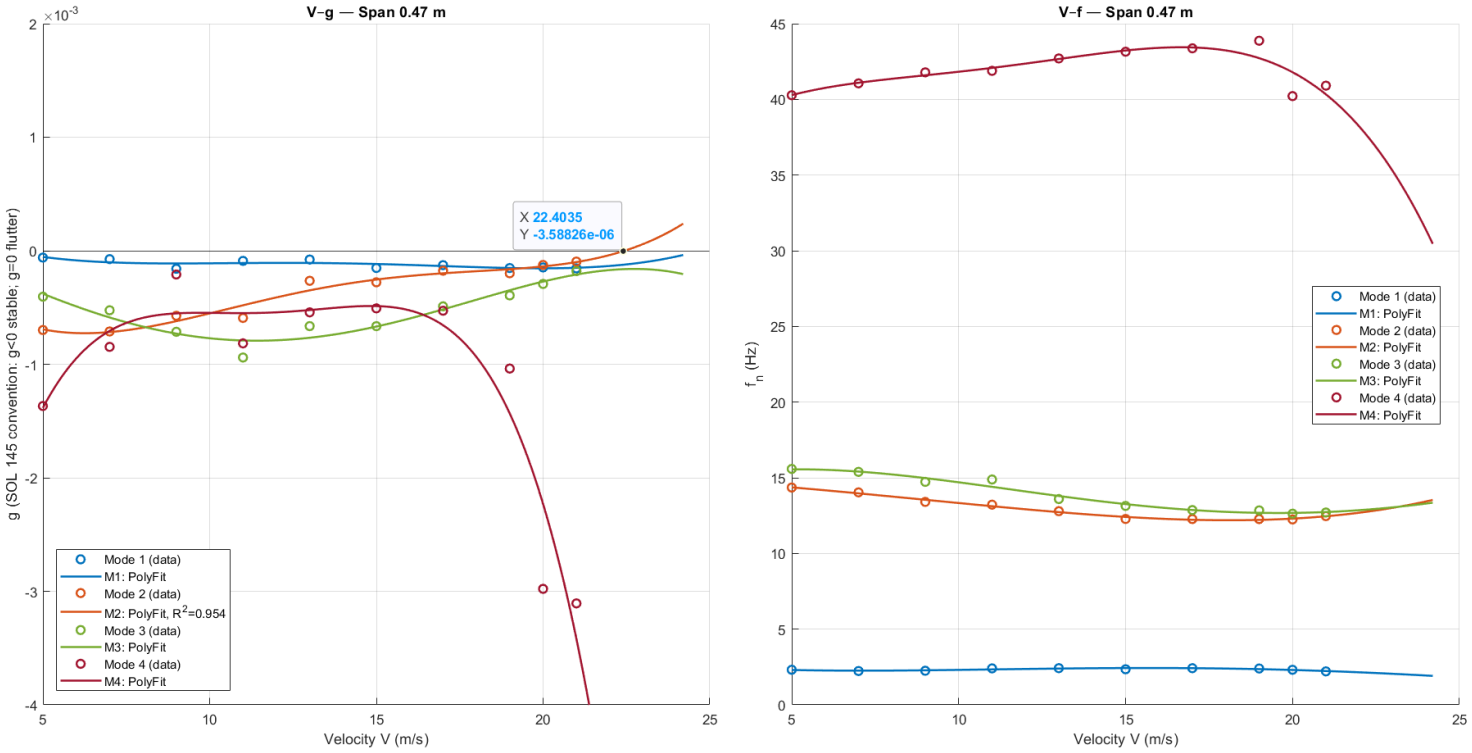
(a) Experimental V-g-f obtained



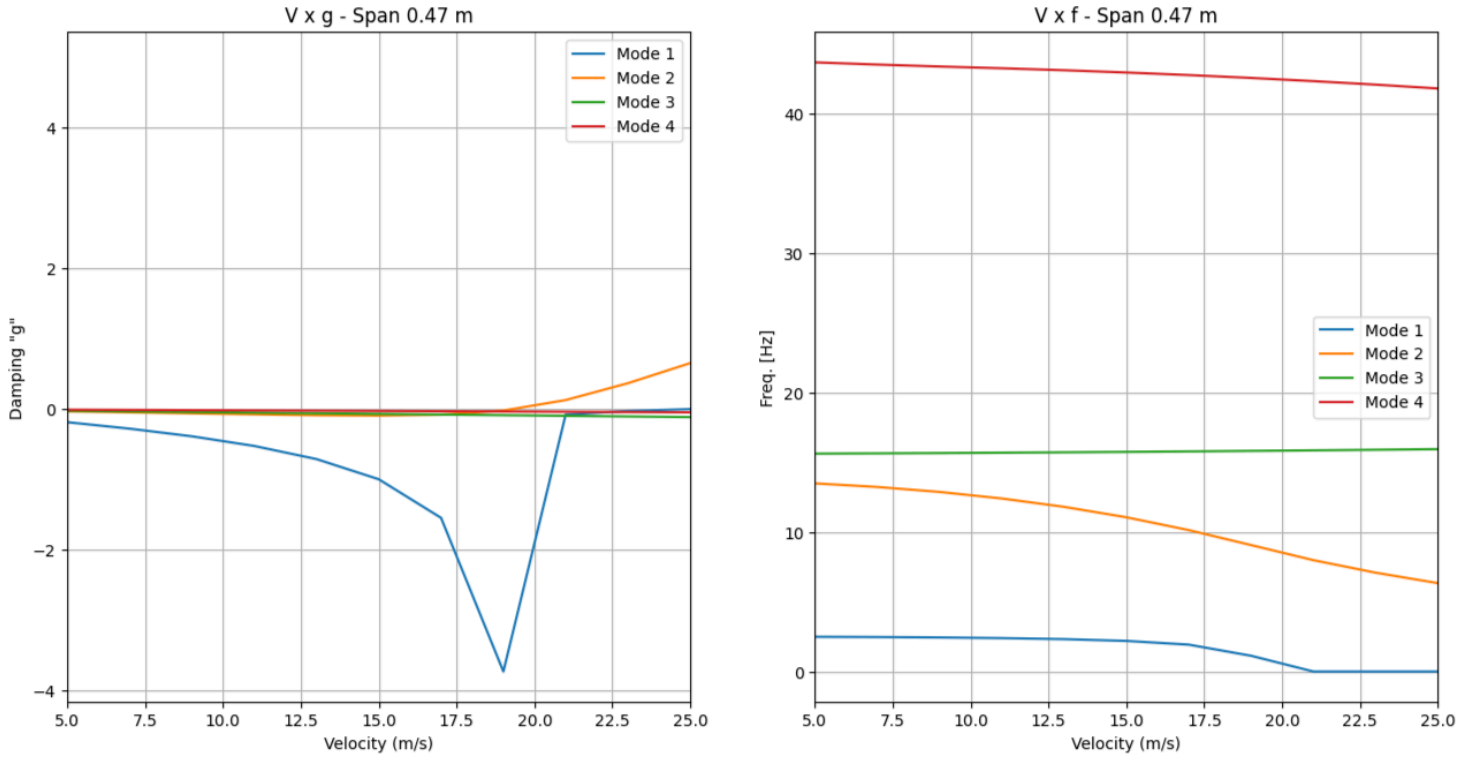
(b) Simulation V-g-f obtained

Source – Elaborated by the author

Figure 15 – Experimental and Simulation V-g-f Comparison (S = 0.47 m)



(a) Experimental V-g-f obtained



(b) Simulation V-g-f obtained

Source – Elaborated by the author

Table 7 summarizes the quantitative comparison at flutter for each tested span.

Table 7 – Simulation vs. experiment at flutter (Mode responsible indicated; errors relative to experimental).

Semi–span (m)	Flutter speed U_f (m/s)		Error [%]	Flutter Mode
	Sim	Exp		
0.41	23.18	24.01	3.45	2
0.44	21.29	23.82	10.62	2
0.47	19.27	22.40	13.97	2

Source – Elaborated by the author

Overall, the comparisons indicate that the calibrated FE model together with DLM/PKNL captures the salient subcritical trends and the onset of instability to within a relatively small percent differences, less then 15 % for the three spans tested. This level of agreement validates using the simulation dataset as the primary source for training the neural network in Chapter 3, while the experimental measurements are retained for out-of-sample verification of the learned damping maps and the derived flutter-speed predictions in Chapter 4.

4.2 Neural Network Training Performance

Figure 16 shows the learning curves for the final MLP, with a rapid initial decrease of both losses followed by a long, nearly flat plateau. The minimum training loss reached was MSE = 0.008925, while the best validation loss was MSE = 0.096932. The persistent gap indicates mild overfitting despite dropout and weight decay. However, the validation curve stabilizes without divergence, and the test performance reported below remains acceptable for the intended use.

To quantify accuracy, mean squared error, root mean squared error, and mean absolute error are used over the held-out samples, as defined previously. For targets g_i and predictions $\hat{g}_i$ across N samples (aggregating all output modes unless stated otherwise).

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{g}_i - g_i\right)^2, \tag{4.2}$$

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}}, \qquad \mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|\hat{g}_i - g_i|. \tag{4.3}$$

On the reserved test set (simulation only), the errors in the scaled space used for training were MAE = 0.060362, RMSE = 0.137973, and MSE = 0.019037. After undoing

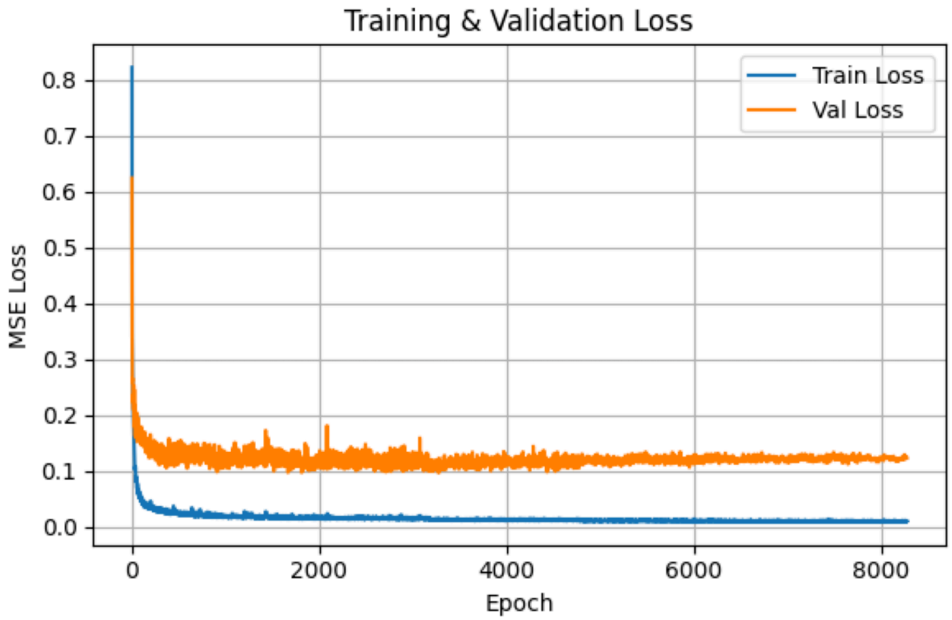
the standardization to physical damping units, the aggregate test errors were MAE = 0.040083, RMSE = 0.146016, and MSE = 0.021321. A sign-consistency check below the enforced low-speed threshold reports perfect compliance: $\mathrm{Acc}_{\mathrm{sign}} = 1.000$ for $V < 10\,\mathrm{m/s}$.

$$\mathrm{Acc}_{\mathrm{sign}} = \frac{1}{N_{<V_0}} \sum_{i:V_i<V_0} \mathbf{1}\{\hat{g}_i < 0\}. \tag{4.4}$$

Mode-wise errors in physical units show that most of the test-set discrepancy concentrates in the first mode, whereas the higher modes are learned with much smaller absolute errors. The concentration of error in Mode 1 is consistent with the larger dynamic range and stronger nonlinearity of bending-dominated damping near the onset region. The small errors in Modes 2–4 support using the network's $g(V)$ maps to extract flutter speeds robustly by zero-crossing.

Table 8 summarizes the aggregate metrics; Table 9 reports the per-mode values in physical units.

Figure 16 – Training and validation loss versus epoch for the final MLP.



Source – Elaborated by the author

Table 8 – Aggregate errors for the final model (simulation test set).

Set / Space	MAE	RMSE	MSE
Train (minimum)	–	–	0.008925
Validation (best)	–	–	0.096932
Test (scaled)	0.060362	0.137973	0.019037
Test (physical)	0.040083	0.146016	0.021321

Source – Elaborated by the author

Table 9 – Per-mode errors on the simulation test set (physical damping units).

Mode	MAE	RMSE	MSE
1	0.123272	0.285610	0.082088
2	0.003085	0.056297	0.003169
3	0.003922	0.004833	0.000023
4	0.001052	0.001306	0.000002

Source – Elaborated by the author

In summary, the network converges smoothly and generalizes adequately to unseen simulated cases. The slight train–validation gap reflects the limited size of the dataset and the sharp change of target behavior near the flutter threshold. Nevertheless, the physical-space accuracy in Modes 2–4, together with strict low-speed sign consistency, supports using the predicted $g(V)$ curves to locate flutter via zero-crossing.

4.3 Prediction of Critical Flutter Speed using the Neural Network

The trained MLP does not output flutter speed directly. Instead, it predicts the mode–by–mode damping values $\hat{g}_m$ as functions of span S, velocity U, and modal frequencies. As explained in Sec. 3.6, the critical speed is obtained by scanning the same velocity grid used in the simulations and taking the earliest zero–crossing of the predicted damping among the tracked modes.

The zero crossing is located by linear interpolation between consecutive grid points. This mirrors the classical V–g criterion and makes the artificial neural network (ANN) a fast surrogate for SOL 145 post–processing.

Table 10 compares the ANN–predicted flutter speeds with the Nastran PKNL results for the original semi–span set $\{0.35, 0.38, 0.41, 0.44, 0.47, 0.50\}$ m. The percentage error is computed as shown in Eq. 4.5.

$$\Delta U_f\ [\%] \;=\; 100\,\frac{|U_{f_{ANN}} - U_{f_{sim}}|}{U_{f_{sim}}}. \tag{4.5}$$

Table 10 – ANN flutter speed prediction ($U_{f_{ANN}}$) versus SOL 145 (PKNL) and experimental data

Semi–span (m)	$U_{f_{exp}}$ (m/s)	$U_{f_{sim}}$ (m/s)	$U_{f_{ANN}}$ (m/s)	$Error_{Sim.}$ (%)	$Error_{Exp.}$ (%)
0.35	—	28.65	29.160	1.78	—
0.38	—	25.40	25.509	0.43	—
0.41	24.01	23.18	22.996	0.80	4.24
0.44	23.82	21.29	20.577	3.35	13.64
0.47	22.40	19.27	18.278	5.15	18.34
0.50	—	17.65	16.912	4.18	—

Source – Elaborated by the author

Across the six spans, the mean absolute relative error is about 2.6%, with the largest deviation occurring at the most flexible end ($S = 0.47\,\text{m}$, 5.15%). The ANN reproduces the expected monotonic trend of decreasing flutter speed with increasing span and closely tracks the SOL 145 curve in the mid–range (0.35–0.41 m). For spans not explicitly used to generate the simulation set, the model provides smooth interpolations. For example, it predicts $U_f \approx 27.29\,\text{m/s}$ at $S = 0.36\,\text{m}$, $U_f \approx 21.89\,\text{m/s}$ at $S = 0.42\,\text{m}$ and $U_f \approx 16.83\,\text{m/s}$ at $S = 0.49\,\text{m}$, consistent with the physical trend.

For the spans with wind–tunnel data (0.41, 0.44, 0.47 m), Section 4.1 showed that SOL 145 slightly underestimates the measured flutter speed. Because the ANN is trained on simulation–based labels and we compute $\hat{U}_f$ from the ANN's predicted $V-g$ curves, the ANN largely reflects that bias. Nevertheless, its additional error relative to the simulation reference remains small. In practical terms, this makes the network a rapid evaluator that can deliver flutter speed estimates within a few percent of the high–fidelity solver while preserving the correct trend across span. From a runtime standpoint, the advantage is substantial: in our pipeline, a single SOL 145 sweep per semi–span typically requires on the order of 10–15 minutes to finish, after which the results still need to be analyzed and the $V-g$ diagram plotted to read off the flutter speed. By contrast, once trained, the network produces the mode-wise $g(V)$ curve and its zero–crossing (U_f) for a new semi–span essentially instantaneously via a single forward pass, enabling interactive and rapid exploration and batch evaluation of many spans without additional post–processing.

CHAPTER

5

CONCLUSION

The approach adopted combined calibrated FEMAP/NASTRAN flutter simulations (SOL 145 with DLM aerodynamics and PKNL sequencing) with a supervised multi–output MLP trained on simulation data and verified against wind–tunnel measurements. The proposed pipeline reproduced classical aeroelastic observables, $g(V)$ and $f(V)$ curves per mode, and used the learned $g(V)$ maps to infer the flutter speed by zero–crossing, thus remaining fully consistent with engineering practice.

The main results showed that the calibrated numerical model captured the subcritical trends and the onset of instability for the tested spans, enabling its use as the primary source for training. The final MLP achieved stable convergence, small aggregate errors in physical damping units on the held–out simulation set, and strict sign consistency at low speeds. When converted to flutter speed, the network reproduced the expected monotonic decrease of U_f with increasing span and matched the SOL 145 references with mean absolute relative errors on the order of a few percent across the span set, while remaining faithful to the experimental trend where tunnel data were available. In practical terms, once trained, the network can deliver near–instant predictions over dense velocity grids and for new spans obtained by interpolation of modal frequencies, providing a fast surrogate to support preliminary aeroelastic assessments.

Two limitations constrained the scope and breadth of the dataset. On the simulation side, the lack of access to a compute cluster limited the number of high–fidelity runs and the extent of parametric sweeps that could be performed within a reasonable time. This motivated the use of the non–iterative PKNL option to generate smooth V–grids efficiently, with spot checks and experimental validation mitigating the slight loss of k–consistency near rapid frequency variation. On the experimental side, the wind tunnel's operational envelope up to approximately 25 m/s restricted the spans that could be tested to those predicted to flutter within that range. As a consequence, only three semi–spans were measured, and the resolution in speed was limited by the 2 m/s step. These

constraints naturally narrow the domain on which the learned model is both trained and validated, and they recommend caution when extrapolating far from the studied configurations.

Despite these constraints, the thesis demonstrates that integrating numerical aeroelastic analysis with a lightweight neural regressor is a viable pathway to accelerate flutter prediction. The method delivers engineering–grade estimates of damping and flutter speed at a fraction of the computational and operational cost of full analyses or additional tunnel campaigns, which is especially attractive during early design studies and "what–if" tradeoffs across span.

Future work should prioritize expanding both numerical and experimental coverage and broadening the space of geometric and physical parameters. On the numerical front, greater computing resources would enable denser flight grids, larger span sets, and multidimensional sweeps involving stiffness, mass distribution, and boundary–condition compliance, as well as systematic PK spot–verification at critical points. On the experimental front, extending the facility envelope or employing alternative rigs would allow testing beyond 25 m/s and collecting higher–density data near the onset, improving the fidelity of validation and offering richer targets for learning.

A natural next step is to move beyond span–only variations to include other geometric features, such as dihedral and sweep, and to explore thickness or chord changes that alter $\boldsymbol{EI}$ and $\boldsymbol{GJ}$ in a controlled way. Incorporating these parameters into the feature vector would let the model learn higher–dimensional stability trends and support preliminary optimization loops over geometry. In parallel, model–development avenues include physics–guided regularization and uncertainty quantification, for example via ensembles or Bayesian layers, to attach credible intervals to $g(V)$ and U_f predictions, and domain–adaptation strategies that reduce any residual simulation–to–experiment shift by leveraging the limited tunnel data.

Finally, an interesting direction is to reduce reliance on hand–engineered modal features by learning directly from raw experimental time series. Unsupervised or self–supervised models, such as autoencoders and sequence encoders, could discover latent coordinates that track stability and, when coupled to a light supervised head, infer $g(V)$ and U_f without explicit modal identification. Such models might also detect the emergence of limit–cycle oscillations or other nonlinear signatures earlier than classical pipelines. Together with expanded datasets and broader geometric variation, these advances would strengthen the robustness and applicability of data–driven flutter prediction and help integrate it into modern, rapid design workflows.

BIBLIOGRAPHY

BANERJEE, J. R. Review of the dynamic stiffness method for free-vibration analysis of beams. **Transportation Safety and Environment**, v. 1, n. 2, p. 106–116, 2019. Citation on page 28.

BISMARCK-NASR, M. N. **Structural Dynamics in Aeronautical Engineering**. Reston, VA: American Institute of Aeronautics and Astronautics, 1999. (AIAA Education Series). Citations on pages 14, 16, 22, and 23.

COLLAR, A. R. The first fifty years of aeroelasticity. **Aerospace**, v. 5, n. 2, p. 12–20, 1978. Citations on pages 11 and 14.

FOI — Swedish Defence Research Agency. **Dynamic Aeroelastic Simulation of a Slender Wing (AGARD 445.6)**. Linköping, Sweden, 2007. Available: <https://foilibrary.foi.se/report-files/FOI-R--2259--SE.pdf>. Citation on page 27.

FUNG, Y. C. **An Introduction to the Theory of Aeroelasticity**. Mineola, NY: Dover Publications, 2008. Reimpressão da edição original de 1955 (John Wiley & Sons). Citations on pages 14, 16, 18, 21, and 22.

GARRICK, I. E.; REED, W. H. I. Historical development of aircraft flutter. **Journal of Aircraft**, v. 18, n. 11, p. 897–912, 1981. Citation on page 11.

GÉRON, A. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems**. 2. ed. Sebastopol, CA: O'Reilly Media, 2019. Citation on page 26.

GUO, Q.; LI, X.; ZHOU, Z.; MA, D.; WANG, Y. Neural network-based aeroelastic system identification for predicting flutter of high flexibility wings. **Scientific Reports**, v. 15, p. 623, 2025. Citation on page 28.

INC., A. E. **Flutter Analysis Setup Tips & Tricks**. Troy, MI: [s.n.], 2022. Altair Community — OptiStruct. Post author: usuário "pstang". Inclui link para *Tip_Trick_OptiStruct_Flutter_Analysis_2022.pdf*. Available: <https://community.altair.com/discussion/39769/flutter-analysis-setup-tips-tricks/p1>. Accessed: 3 set. 2025. Citation on page 19.

KEHOE, M. W. **Frequency and Damping Estimation During Flight Flutter Testing**. Hampton, VA, 1988. Available: <https://ntrs.nasa.gov/api/citations/19890014723/downloads/19890014723.pdf>. Citation on page 24.

LEISHMAN, J. G. Unsteady aerodynamics, aeroelasticity, & flutter. In: **Introduction to Aerospace Flight Vehicles**. Daytona Beach, FL: Embry-Riddle Aeronautical University, 2023. Available: <https://eaglepubs.erau.edu/introductiontoaerospaceflightvehicles/chapter/unsteady-aerodynamics-flutter/>. Accessed: 04 set. 2025. Citations on pages 14 and 15.

MOHAMMADI-AMIN, M.; GHADIRI, B. An aeroelastic metamodel based on experimental data for flutter prediction of swept rectangular wings. **Journal of Applied Fluid Mechanics**, v. 6, n. 1, p. 115–120, 2013. Citation on page 26.

MOHAMMED, A. **Free Vibration and Flutter Analysis of High Aspect Ratio Wings**. Bachelor's Thesis (Graduação (BEng Aeronautical Engineering)) — City, University of London, London, 2017. Citations on pages 27 and 28.

RAMOS, B. P.; RIBEIRO, J. A. G. **Identificação do fenômeno aeroelástico de *flutter* a partir de simulações FSI**. Bachelor's Thesis (Graduação) — Universidade Federal de Uberlândia, Uberlândia, 2018. Citation on page 27.

TADEU, J. **Aconteceu em 17 de março de 1960: Sabotagem? A queda do voo 710 da Northwest Orient Airlines em Indiana (EUA)**. 2022. Blog *Notícias e Histórias sobre Aviação* (Desastres Aéreos). Available: <https://desastresaereosnews.blogspot.com/2022/03/aconteceu-em-17-de-marco-de-1960.html>. Accessed: 02 set. 2025. Citation on page 12.

THEODORSEN, T. **General Theory of Aerodynamic Instability and the Mechanism of Flutter**. Washington, DC, 1934. Available: <https://ntrs.nasa.gov/api/citations/19930090935/downloads/19930090935.pdf>. Citations on pages 16, 18, and 27.

TISCHLER, M. B.; BENCIC, T. J. **Frequency- and Time-Domain Identification of a Mathematical Model of an Airfoil Undergoing Flutter**. Moffett Field, CA, 1982. Available: <https://ntrs.nasa.gov/api/citations/19830005618/downloads/19830005618.pdf>. Citation on page 17.

WANG, H.; WU, T. Knowledge-enhanced deep learning for wind-induced nonlinear structural dynamic analysis. **Journal of Structural Engineering**, American Society of Civil Engineers, v. 146, n. 11, p. 04020235, 2020. ISSN 0733-9445. Citation on page 25.

WRIGHT, J. R.; COOPER, J. E. **Introduction to Aircraft Aeroelasticity and Loads**. 2. ed. Chichester: John Wiley & Sons, 2015. (Aerospace Series). Citations on pages 14, 15, 16, 18, 21, 22, and 23.

YATES E. C., J. **AGARD Standard Aeroelastic Configurations for Dynamic Response. Volume I: Wing 445.6**. Neuilly-sur-Seine, France, 1988. Available: <https://apps.dtic.mil/sti/citations/ADA204617>. Citation on page 27.