International Forum on Aeroelasticity and Structural Dynamics
IFASD 2026
16-19 June 2026, Göttingen, Germany

VOLTERRA KERNELS FOR FLUTTER ANALYSIS

Marcello Righi[1] **and Gabriele Immordino**[1]

[1]School of Engineering
Zurich University of Applied Sciences
8401 Winterthur, Switzerland
rigm@zhaw.ch

Keywords: Structural dynamics, Aeroelasticity, Reduced Order Model, Surrogate Model, Machine Learning

Abstract:

This study presents an analysis and reflection on the use of data-driven nonlinear, parametric aerodynamic reduced-order models for flutter analysis, with particular emphasis on Volterra kernels. While errors are inevitably introduced during model generation/identification and interpolation across the parameter space (where applicable), the approach significantly reduces overall computational costs for parametric studies such as Mach or angle of attack sweeps. Perhaps more importantly, it also reduces the cost of additional analyses such as uncertainty propagation. The development of parametric reduced-order models stands to benefit considerably from the increasingly widespread and continually improving machine learning methodologies and tools. This abstract demonstrates several machine learning applications; we plan to present additional examples in the final manuscript.

NOMENCLATURE

BO	=	Bayesian optimization
$BSCW$	=	Benchmark Super Critical Wing
$FCNN$	=	fully–connected neural network
ML	=	machine–learning
MSE	=	mean squared error
ROM	=	reduced–order model

1 INTRODUCTION AND MOTIVATION

The idea for this paper arises from our 2025 study [1], in which we employed machine learning techniques—Gaussian processes and artificial neural networks—to interpolate the coefficients of first- and second-order Volterra kernels across a Mach number and angle of attack parameter space and demonstrated that the interpolation error is acceptable. We also showed that artificial neural networks, provided that hyperparameters are appropriately optimized, outperform Gaussian processes. In this study, we present several flutter analyses (two of which are already included in this abstract).

This abstract does not provide a comprehensive review of the most relevant literature (which will of course be included in the paper); however, we note a growing interest in the Volterra

kernel approach in aeroelasticity and cite several recent publications by our colleagues at RMIT [2–7]. We also note the emergence of recent studies applying machine learning to aeroelastic problems, such as [8].

This abstract is structured as follows: the previously developed methodology is presented in Section 2, and results obtained from two test cases are shown in Section 3. Note that one of the test cases, the Benchmark Super Critical Wing (BSCW), is also among the aeroelastic systems being studied in the Fourth Aeroelastic Prediction Workshop, scheduled for the week prior to IFASD. The authors are participating and plan to present results. The present paper is also expected to complement the Workshop by providing additional insights into the methodologies and developments enabled by machine learning. In the final manuscript, we plan to improve and complete the test case documentation, extend the range of results, and enhance the generation/identification process. As of this abstract submission, we are experimenting with entirely machine learning-generated parametric reduced-order models, which show considerable promise.

2 METHODOLOGY

2.1 Volterra series for SISO systems

Volterra series is a powerful technique to solve nonlinear integral equations since it enables us to represent the solution as a power series. By doing so, we can approximate the solution to any desired level of precision. It can be applied to model the relation between input $u(t)$ and output $y(t)$ of a nonlinear time–invariant system with memory, as the unsteady aerodynamic loads resulting from a step change in the AoA. The key idea behind the Volterra series is to substitute the unknown function $y(t)$ with a power series, and then use this substitution to convert the nonlinear integral equation into a set of linear equations for the coefficients of the power series. To determine the coefficients of the Volterra series, we need to solve the associated Volterra integral equations, which can be expressed as a matrix of linear equations. Once we have obtained the coefficients of the series, they can be used to estimate the solution of the original nonlinear integral equation.

The output function $y(t)$ of a continuous–time system in response to an input $u(t)$ can be modeled using the Volterra series:

$$y(t) = \int_0^t h(\tau)u(t-\tau)d\tau \tag{1}$$

where $h()$ are the convolution kernels. Its discrete–time formulation may be expressed as a sum of nonlinear kernels convoluted with inputs and products of inputs, yielding to Equation 2.

$$y(t) = \sum_{n=1}^{N} H_n[u(t)] \tag{2}$$

This polynomial approximates a nonlinear system to any desired precision if the order N is made sufficiently large. In general, the kernel can be expressed as:

$$
\begin{aligned}
H_1[u(t)] &= \sum_{i=1}^{M} h_1(m_i) \cdot u(t - m_i) \\
H_2[u(t)] &= \sum_{i=1}^{M} \sum_{j=1}^{M} h_2(m_i, m_j) \cdot u(t - m_i) u(t - m_j) \\
H_n[u(t)] &= \sum_{i=1}^{M} ... \sum_{...}^{M} h_n(m_i, ..., m_{...}) \cdot \prod_{j=1}^{N} u(t - m_j)
\end{aligned} \tag{3}
$$

where m_j are the memory lag terms. $u(t - m_j)$ is the input to the system, shifted in time. $H_1[u(t)]$ is the linear convolution term. The higher–order terms make the system nonlinear. The more terms are used, the more accurate the approximation will be, but it will also be more computationally expensive. For this reason, it is not common to use Volterra series beyond second term. To illustrate, considering a scenario where the memory length M is 10 and the kernel degree is 10 and taking into account the symmetry in the coefficients, we would need to estimate around $2 \cdot 10^5$ parameters [9]. To overcome this limitation, we adopt the simplifications proposed by Dowell [10] and references therein. We assume that the kernels are symmetrical, the diagonal kernels are sufficient to represent the system and the nonlinear kernels can be expanded respect to time as a combination of Laguerre polynomials, which are a series of orthogonal polynomials that can be used to reduce the number of coefficients required to describe a Volterra kernel. These three assumptions permit to simplify the formulation of the Volterra series [10].

Considering that high–order kernels can be represented by a product of the base functions as they are separable [11, 12], we obtain:

$$
h_n(m_i, ..., m...) = \sum_{r=1}^{R} \theta_r \prod_{i=1}^{N} L_r(m_i) \tag{4}
$$

where θ_r denote the weights of the Laguerre polynomials.

Substituting Equation 4 in Equation 2, we derive the generalized response of a dynamical system with an arbitrary order of the Volterra series:

$$
y(t) = [U][B]\{\theta\} \tag{5}
$$

where $[U]$ is the vector that combines inputs and product of the inputs of the system, $[B]$ is a block matrix of Laguerre values and $\{\theta\}$ is the vector of Laguerre coefficients which need to be determined. In our case, the order of the Laguerre polynomials has been set to 17, after a careful assessment aimed at finding the minimum order of the polynomial that better represents the solution. The reader is referred to Dowell et al. [10] for the complete formulation.

For obtaining the vector of unknowns $\{\theta\}$ it is necessary to decompose and invert the $[U][B]$ matrix which can be done using singular value decomposition (SVD).

$$svd\left([U][B]\right) = [U][\Sigma][V]^T$$
$$\{\theta\} = [V][\Sigma]^{-1}[U]^T y(t) \tag{6}$$

Now, we define $C_{L_{NL}}$ as:

$$C_{L_{NL}} = C_{L_{big}} - C_{L_{small}} \cdot factor \tag{7}$$

where $C_{L_{big}}$ refers to a pitch input angle of 2 [deg], $C_{L_{small}}$ to a pitch input angle of 1 [deg] and the $factor$ term is equal to 2. In this way, the difference between the two step–responses represents the nonlinear component.

Applying Equation 6 to Equation 7 we get the nonlinear Volterra kernels $\{\theta_{NL}\}$. The same procedure here explained can be applied for the reconstruction of the nonlinear component of the pitching moment C_M.

2.2 Machine Learning to Interpolate on Mach - Angle of Attack parameter space

Machine learning is used in the test cases presented to interpolate the kernels coefficient for a specific combination of Mach number and angle of attack. The kernels are identified for a sufficiently lagre number of samples.

We applied the concept of the deep–learning framework reported in Immordino et al. [13] which results extremely useful for optimizing the architecture of the neural network for this specific task. We trained two different neural networks, one for the linear kernels and one for the non-linear ones. We used Mach, AoA and static value of the lift coefficient and pitching moment coefficient as inputs for our models. Table 1 compares the optimal hyperparameters for the linear and nonlinear Volterra kernels identification. The reader might observe that the algorithm deepened the network in order to capture the nonlinear behaviour of the kernels.

Hyperparameter	**Optimal Value**	
	Linear Kernels	NL kernels
Learning rate	10^{-4}	10^{-4}
Number of Hidden Layers	5	7
Activation function	PReLu	PReLu
Number of parameters	144,502	259,890

Table 1: Optimal hyperparameters for the linear and nonlinear Volterra kernels identification.

2.3 State-space Model with the Linear Volterra Kernel

A linear aeroelastic system is often presented in the form:

$$M_s\ddot{x} + C_s\dot{x} + K_s q = f \tag{8}$$

for a two-degree-of-freedom system (e.g. typical section) for in instance, the matrices can be:

$$q = \begin{pmatrix} h \\ \alpha \end{pmatrix}, \quad M_s = \begin{bmatrix} m & S_\alpha \\ S_\alpha & I_\alpha \end{bmatrix}, \quad C_s = \begin{bmatrix} c_h & 0 \\ 0 & c_\alpha \end{bmatrix}, \quad K_s = \begin{bmatrix} k_h & 0 \\ 0 & k_\alpha \end{bmatrix}, \quad f = \begin{pmatrix} -L \\ M \end{pmatrix}$$

The system can be recast into the first order form:

$$\begin{cases} \dot{q} = Aq + Bu \\ y = Cq + Du \end{cases} \tag{9}$$

Since the output variables coincided with the state ones, C is a identity matrix and D is zero, whereas q, A and B assume the following expressions:

$$q = \begin{pmatrix} h \\ \alpha \\ \dot{h} \\ \dot{\alpha} \end{pmatrix}, \qquad A = \begin{bmatrix} 0 & I \\ -M_s^{-1}K_s & -M_s^{-1}C_s \end{bmatrix}, \qquad B = \begin{pmatrix} 0 \\ M_s^{-1}f \end{pmatrix}$$

Further the continuous–time system is converted into discrete–time:

$$q^{n+1} = A_D\, q^n + B_D\, f^n, \tag{10}$$

where:

$$A_D = (I - \Delta t \beta A)^{-1}\,(I + \Delta t(1-\beta)A), \qquad B_D = (I - \Delta t \beta A)^{-1}\, B$$

The linear Volterra kernels may now be included into the state–space system:

$$L^n = H_L\, \alpha_{\text{eff}}^n, \qquad M^n = H_M\, \alpha_{\text{eff}}^n$$

where α_{eff}^n is the vector of the m effective angle of attack values used in the model for the n^{th} time instance. In the case of the typical section in incompressible flow, the effective angle of attack is:

$$\alpha_{\text{eff}}^n = \begin{bmatrix} 0 & 1 & 1/V & 0 \end{bmatrix} q^n,$$

The degrees of freedom now include m values of the effective angle of attack; however, only $m-1$ are added to the vector of the degrees of freedom, as α_{eff}^n is expressed as a function of $\dot{h}^n$ and α^n:

$$x^n = \begin{pmatrix} h^n \\ \alpha^n \\ \dot{h}^n \\ \dot{\alpha}^n \\ \alpha_{\text{eff}}^n \\ \alpha_{\text{eff}}^{n-1} \\ \cdots \\ \alpha_{\text{eff}}^{n-m+1} \end{pmatrix}$$

Now, the system response at $n+1$ can be estimated with:

$$x^{n+1} = \mathbf{A}\, x^n + \{f\}^n \tag{11}$$

where $\{f\}^n$ represents the matrix including external forces and nonlinear Volterra kernels at n^{th} timestep and the state matrix $\mathbf{A}$ is defined as:

$$\mathbf{A} = \begin{bmatrix} A_D & & \mathbf{0} & \\ & & B_D\, H_L & \\ & & B_D\, H_M & \\ & 0 & & \mathbf{0} \\ & 1 & \ddots & \\ \mathbf{0} & & \ddots & 0 \\ & & & 1 \\ & \mathbf{0} & & \end{bmatrix}$$

The dynamic stability of the system can now be assessed on the basis of the eigenvalues of the state matrix.

3 RESULTS

3.1 Isogai Wing

The linear Volterra kernel (Eq. 2) can be conveniently added to a state-space dynamic model and used for aeroelastic analysis, as shown in the previous section.

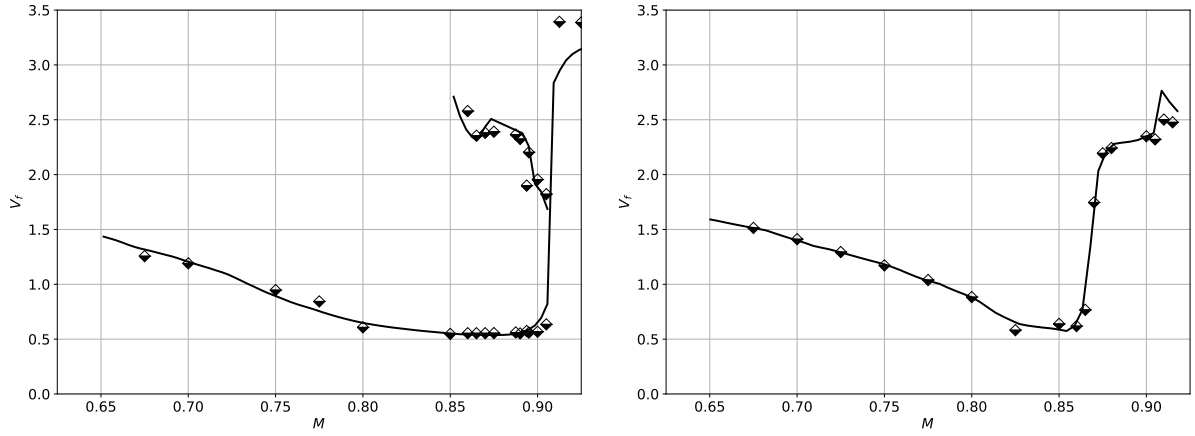
It has been used here to assess flutter speed in the test case proposed by Isogai [14]. The test case consists in a "pitch and plunge" typical section with a NACA64a010 airfoil immersed in a transonic flow.

Since the airfoil is symmetrical and the angle of attack is zero, the problem does not require a nonlinear static aeroelastic solution. Flutter analysis is carried out by identifying the linear kernel from CFD indicial responses at approximately 20 different Mach numbers, using both RANS and Euler equations, in two distinct analysis. The kernels are subsequently reconstructed over about 100 Mach values using a fully-connected neural network (FCNN). For each of these Mach values, a state matrix is built, including the two equations of motion of the typical section

and Eq. 2. Dynamic stability is then assessed based on the eigenvalue analysis. For consistence with the Isogai's paper, flutter analysis uses the flutter index:

$$V_f = \frac{U_f}{b\omega\sqrt{\mu}},$$

where U_f is flutter speed, b the semi-chord, ω the reference natural frequency and $\mu = \dfrac{m}{\pi\rho b^2}$ the airfoil mass ratio and m is the airfoil mass. The flutter index obtained from RANS and Euler simulations is presented in Fig. 1.



(a) Euler (b) RANS

Figure 1: Flutter index NACA 64a010 (Isogai). The continuous line indicates the values from reconstructed kernels, the symbols are test values.

The results obtained from Euler simulations show a transonic dip, which is qualitatively similar to those published by [15, 16] including the "hump mode" behaviour between $M \simeq 0.85$ and $M \simeq 0.91$. The results from RANS simulations are more difficult to assess as no RANS analysis is available to the authors knowledge. The solution obtained with thin layer Navier-Stokes in Ref. [17] presents a similar behaviour.

3.2 Benchmark Super Critical Wing Flutter Analysis

3.2.1 Computational Fluid Dynamics Solver

Our test case revolves around the BSCW, which is a transonic rigid semi–span wing. The BSCW features a rectangular planform and utilizes a supercritical airfoil shape sourced from the AIAA Aeroelastic Prediction Workshop [1]. This wing is mounted on a flexible system with two degrees of freedom, allowing movement in both pitch and plunge. It has been specifically designed for flutter analysis and exhibits three types of nonlinearity: shock wave motion, shock-induced boundary-layer separation, and interactions between the shock wave and detached boundary layer. These nonlinear characteristics pose significant challenges for accurate predictions using reduced-order models.

For generating the CFD step–responses, we employed the Unsteady-RANS (URANS) formulation with SU2 v7.2.1. SU2 is a software that employs a cell-centered finite volume method

[1] `https://nescacademy.nasa.gov/workshops/AePW3/public/wg/highangle`

to convert partial differential equations into ordinary differential equations. The one-equation Spalart-Allmaras turbulence model was used to close the RANS equations. A 1v multigrid scheme was utilized to accelerate the convergence of CFD simulations. The convective flows were discretized using the JST central scheme with artificial dissipation, and the gradients of flow variables were calculated using the Green Gauss method. The linear solver employed was the biconjugate gradient stabilization with the ILU preconditioner applied. All URANS simulations were computed by restarting the solution from a steady-state configuration. The simulations utilized a non-dimensional timestep of $\Delta\tau = 0.029$ and were carried out for a total non-dimensional time of $\tau = 27.2$ to ensure the full development of the flow.

A mixed-type grid consisting of $15.6{\cdot}10^6$ elements and 130,816 surface elements was generated. This grid was structured on the wing surface and in the initial layers of the boundary layer, while voxel elements were used in the rest of the computational domain. A $y^+ = 1$ value was adopted based on a preliminary mesh convergence study, which ensured adequate resolution of the boundary layer and shock wave. The computational domain extended 100 chords from the solid wall to the farfield. Figure 2 provides an impression of the grid employed in the simulations.

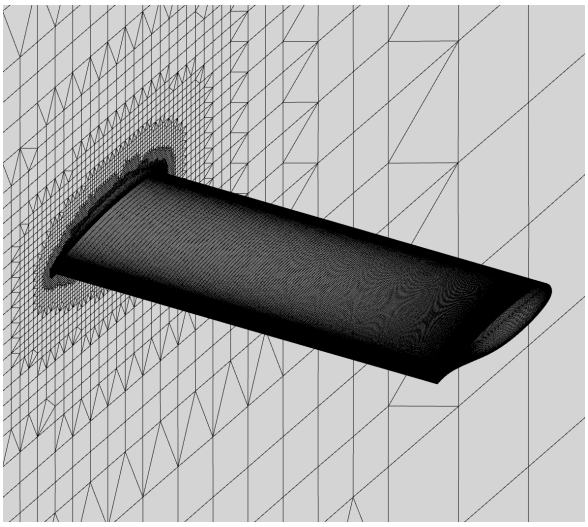


Figure 2: Impression of the BSCW CFD grid.

For the ROM, we chose angle of attack (AoA) and Mach number as the two independent parameters. The selected ranges for these parameters are $[0, 5]$ [deg] for AoA and $[0.70, 0.84]$ for Mach number. These ranges were chosen to encompass the complexity of the physics involved in the problem. In the transonic regime, shock wave formation occurs on the wing, while high angles of attack lead to boundary layer separation.

To obtain the necessary samples for AoA and Mach, we utilized Latin hypercube sampling (LHS) with a total of 70 points, as illustrated in Figure 3. Among these samples, 60% (40 flight conditions with grey squares) were selected for training the ROMs, 20% (15 flight conditions with black circles) were allocated for testing, and the remaining 20% (15 flight conditions with red diamonds) were reserved for validation purposes. Notably, the three highlighted regions in Figure 3 represent increasing complexity in the observed physics phenomena, specifically in terms of boundary layer separation and shock wave formation on both sides of the wing.

The excitation signal for the step responses is an exponential input of the pitch angle around the aerofoil mid–chord:

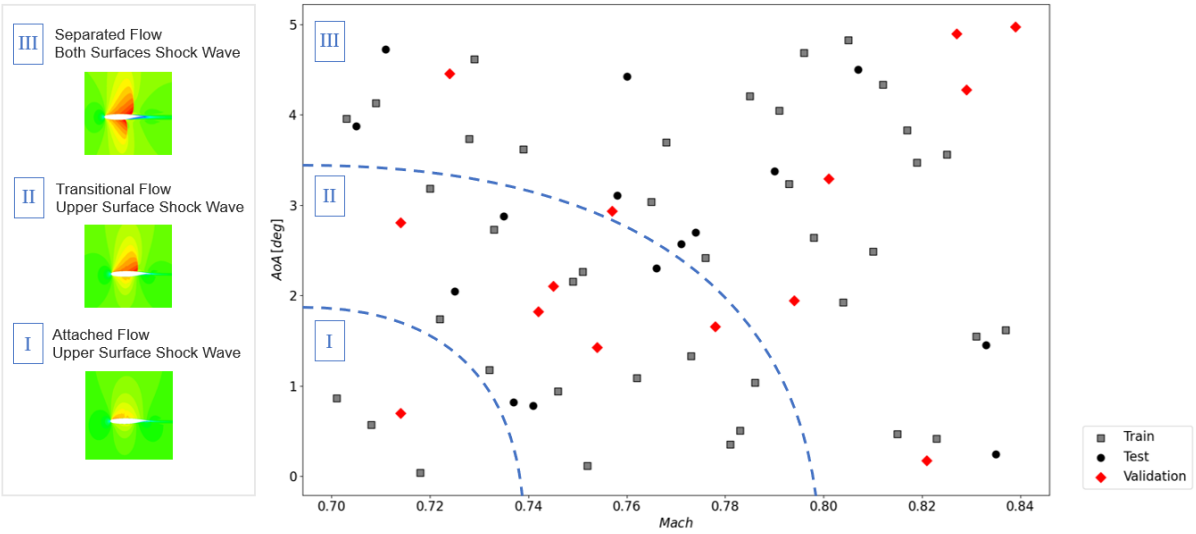


Figure 3: Training, test and validation samples for Mach number and angle of attack. Three regions define the increasing complexity of the physics phenomena to be captured.

$$\alpha(\tau) = \alpha_0 \left(1 - e^{-\tau/\tau_{ref}}\right) \tag{12}$$

where τ is the non–dimensional time and τ_{ref} is equals to 0.6, value chosen after a careful assessment. We generated pitch step–responses of 1 deg and 2 deg in order to cover the linear and nonlinear dynamics (as explained later in the document in Equation 7).

Within the low Mach-AoA region (as depicted in Figure 3), the system response tends to converge to a steady-state value without exhibiting significant oscillations (blue line in Figure 4). Conversely, when both the Mach number exceeds 0.80 and the AoA surpasses 3 deg, the response displays the characteristic behavior of dynamic stall. This phenomenon is marked by an initial increase in the lift coefficient followed by a subsequent decrease, eventually settling down to the steady–state value (orange line in Figure 4).

3.2.2 Flutter Analysis

The developed Volterra–FCNN model is interfaced in a structural solver for fast computation of aeroelastic analyses and flutter prediction. Before delving into the equation describing the wing motion, we will briefly cover the concepts of Limit cycle oscillation (LCO) and flutter and how they are determined in our work.

LCO is defined as a type of periodic motion that occurs in dynamic systems when they exhibit a stable self-sustaining oscillation. The integration of the motion equations in discrete–time and the analysis of the time domain responses at different dynamic pressures helps in determining LCO. Flutter, a closely related phenomenon, is characterized by rapid and uncontrolled oscillations of the wing that can lead to its failure. It is a specific case of LCO, where the wing oscillations become unstable and their amplitude increases until failure occurs. Modal analysis is used to predict flutter. This method involves modeling the wing as a system of coupled oscillators and determining its stability by linearizing the equations of motion of the system and by calculating its eigenvalues. When the real part of one or more eigenvalues is positive, it means that the corresponding natural frequency is growing over time, indicating that the wing is becoming unstable and is prone to rapid, uncontrolled oscillations.

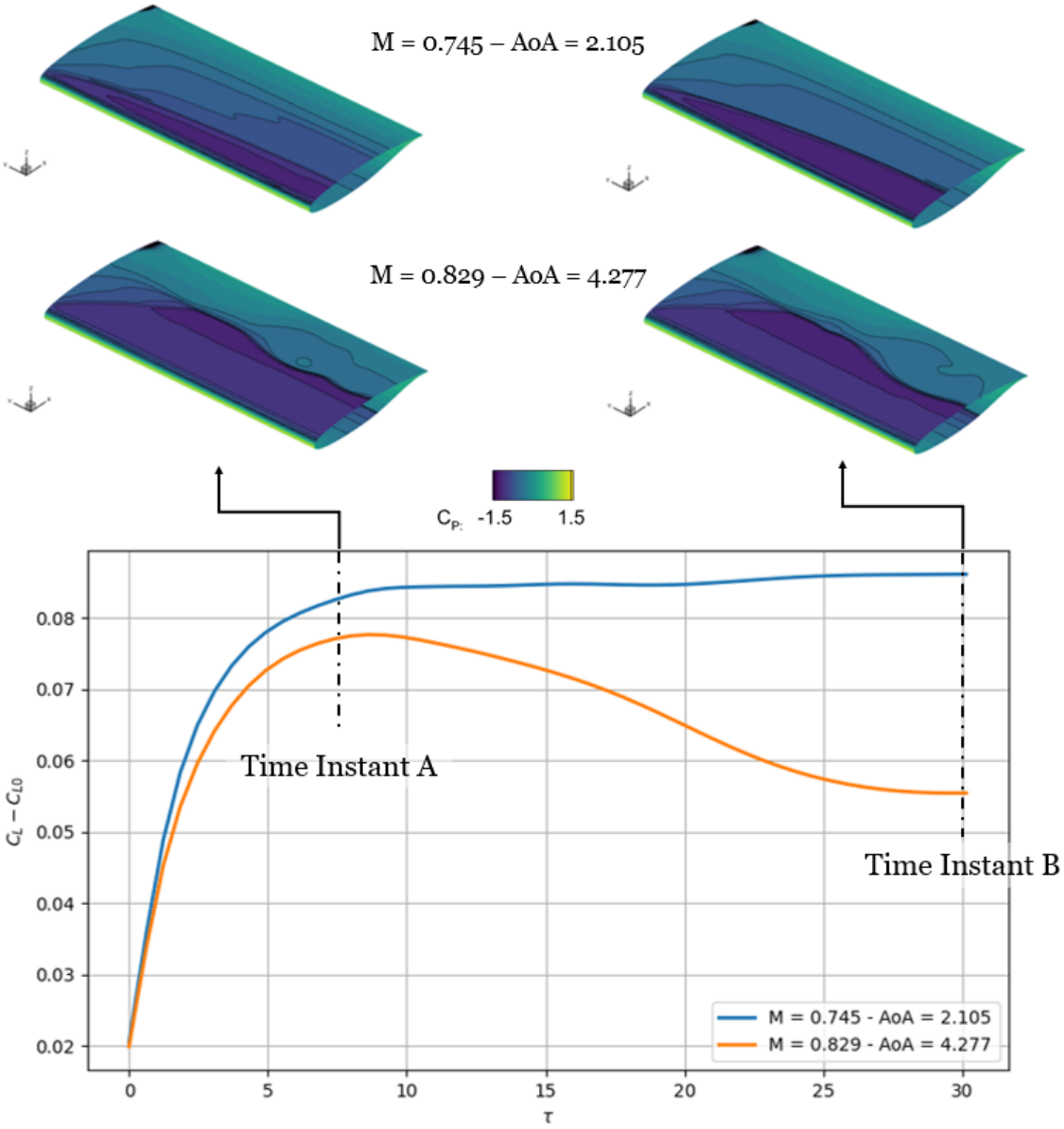


Figure 4: Lift coefficient responses at two different flight conditions with pitch input angle of 1 [deg]. The steady–state values were subtracted. Pressure coefficient contours are also added at $\tau = 7$ denoted as $Time\ Instant\ A$, and $\tau = 30$ denoted as $Time\ Instant\ B$.

The motion of the elastically suspended wing with two degree of freedom can be described as:

$$\begin{cases} m\ddot{h} + S_\alpha \ddot{\alpha} + c_h \dot{h} + K_h h = -L \\ S_\alpha \ddot{\alpha} + I_\alpha \ddot{\alpha} + c_h \dot{\alpha} + K_\alpha \alpha = M \end{cases} \tag{13}$$

where α denotes pitch and h indicates plunge displacement (see Figure 5), m, S_α and I_α are the wing mass and first and second moment of inertia respectively, K_h and K_α the springs constants, c_h and c_α the structural damping coefficients (normally set to zero). The aerodynamic lift and moment are referred to the hinge axis, placed at 50% of the chord. The positions of the elastic axis and the centre of gravity are coincident at the mid–chord.

The structural proprieties of the BSCW wing test case [18–20] are reported in Table 2. Here,

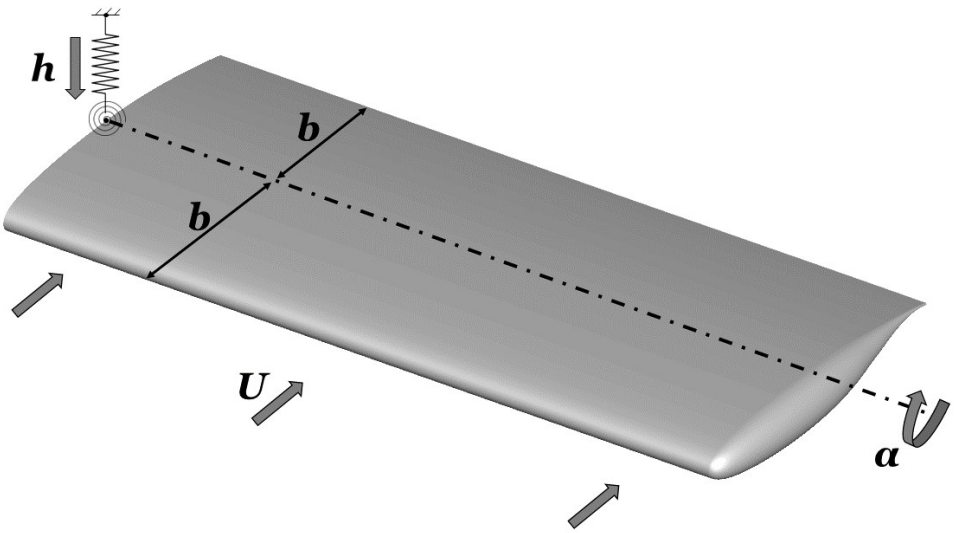


Figure 5: Schematic of the half-span BSCW subject to pitch and plunge motion.

the structural damping is neglected and the positions of the elastic axis and the centre of gravity are coincident at the mid–chord.

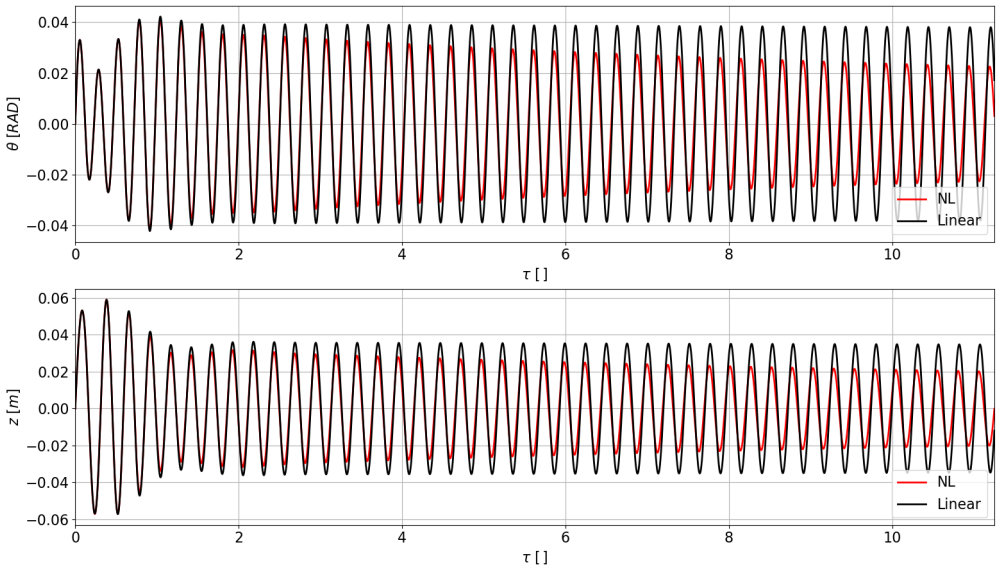
BSCW properties	
K_α	$4018.64\,\mathrm{Nm\,rad^{-1}}$
K_h	$38484.12\,\mathrm{N\,m^{-1}}$
m	$87.91\,\mathrm{kg}$
I	$3.765\,\mathrm{kg\,m^2}$
c	$0.4064\,\mathrm{m}$

Table 2: Structural properties of the BSCW model.

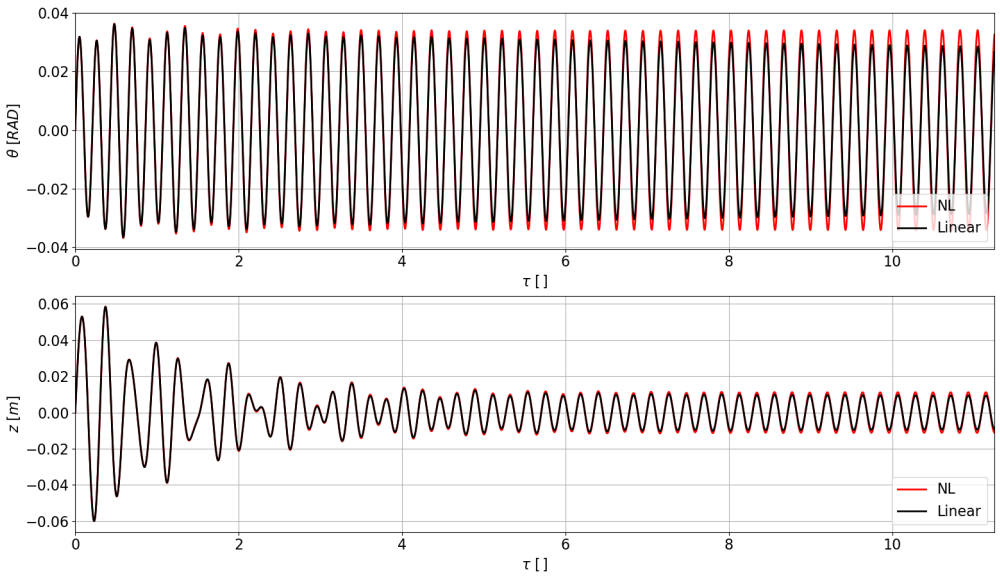
The flutter response of the BSCW system using linear and nonlinear Volterra kernels during flutter is highlighted in Figure 6 at two different flight conditions. Flutter mechanism varies based on flight conditions. Figure 6(a) shows that at low Mach number and AoA, flutter occurs earlier with linear behaviour, in accordance with flutter theory. However, beyond $M = 0.80$ and $AoA = 3$ deg, flutter occurs first when the nonlinear behaviour is considered (as depicted in Figure 6(b)). We suspect that this is related to the presense of a subcritical Limit Cyclo Oscillation (LCO), which brings the system to be unstable at flow velocities below the linear flutter speed. A possible explanation of this phenomenon is the nonlinearity that arises from a shock–induced separation bubble, which strongly influences the shock wave position by pushing it towards the leading edge.

By analyzing the stability of the wing at different dynamic pressures in the Mach-AoA design space, a 3D surface can be created displaying the effect of different flight conditions on the flutter occurrence (Figure 8). The flutter dynamic pressure usually rises with increasing Mach and AoA, but beyond $M = 0.80$ and $AoA = 3$ deg, flutter prediction became extremely sensitive to the flight conditions and more challenging to predict cause of the complex unsteady phenomena taking place. In fact, when the boundary layer detaches, aerodynamic damping decreases significantly, making it challenging to determine the flutter speed.

The dynamic stability in the analyzed region is impacted by both the magnitude and phase of C_L and C_M. Flutter can occur at low reduced frequencies, typically around 0.05. In attached



(a) $M = 0.745 \; - \; AoA = 2.105\, deg \; - \; q = 180\, psf$

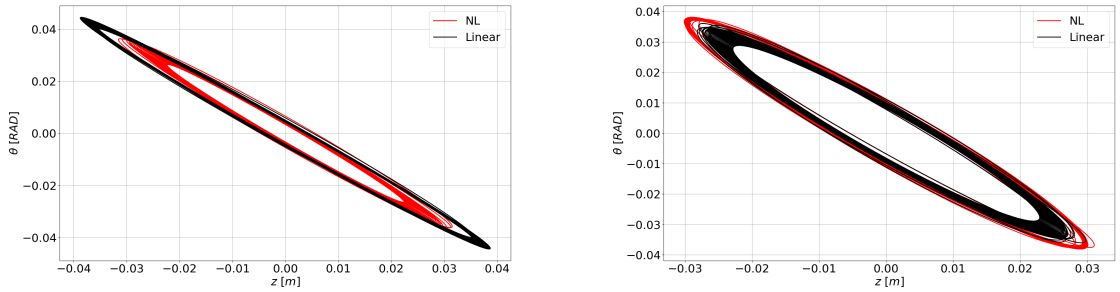


(b) $M = 0.829 \; - \; AoA = 4.277\, deg \; - \; q = 303\, psf$

Figure 6: Pitch and plunge responses during flutter at two different flight conditions for BSCW wing.

flow regions, C_L and C_M are in phase. However, in the ”dynamic stall” region, the long-term response becomes more important than the initial slope. The correlation between C_L and $\dot{\theta}$ becomes stronger as separated flow appears, which results in C_L moving ahead of C_M, in line with the ”dynamic stall” condition.

To improve the accuracy of flutter speed prediction, a data fusion approach has been utilized, integrating predictions from multiple sources, including both low-fidelity and high-fidelity evaluations. By combining these data sources, it is possible to gain a more complete understanding of the flutter speed boundary. Low-fidelity estimates can provide valuable information about



(a) $M = 0.745 \; - \; AoA = 2.105\, deg \; - \; q = 180\, psf$ (b) $M = 0.829 \; - \; AoA = 4.277\, deg \; - \; q = 303\, psf$

Figure 7: Post flutter responses at two different flight conditions for BSCW wing.

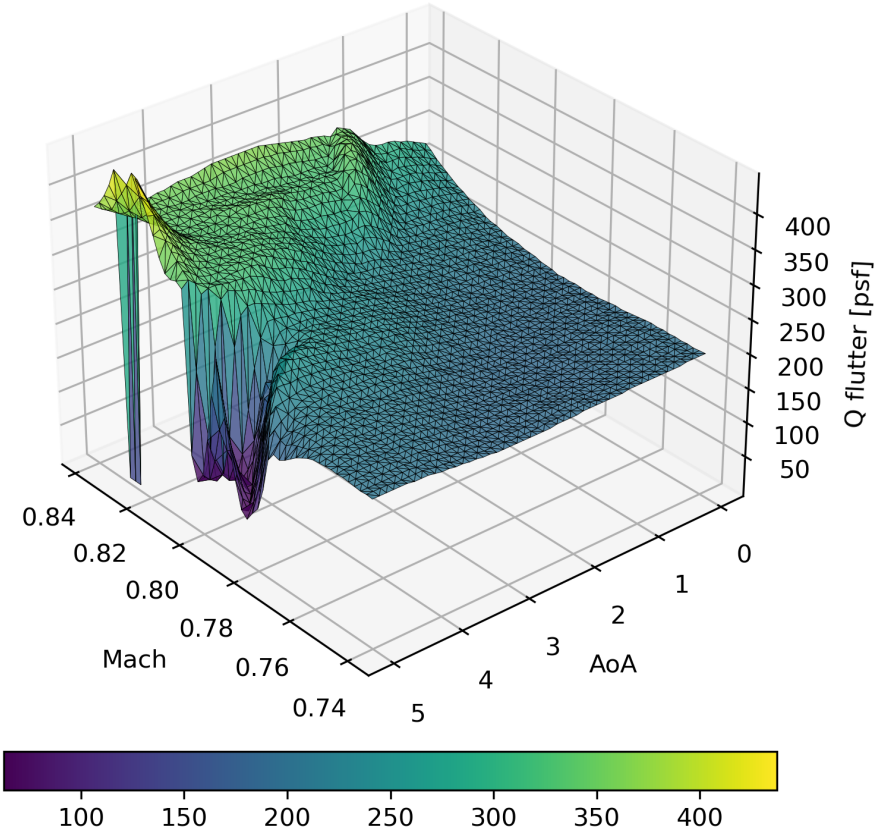


Figure 8: Flutter boundary for BSCW wing.

the general trends of the target function, while high-fidelity samples offer more quantitative information. However, high-fidelity samples are typically only available at a limited number of locations, while low-fidelity estimates may be available at many more locations. To combine the cheap and expensive data, a CoKriging function $\hat{\eta}$ is computed from the samples of the cheap evaluations, as described by Da Ronch et al. [21]. This function is then evaluated at the locations where expensive predictions are available. The resulting vector of input parameters at the expensive samples, denoted as x_i, is augmented with the evaluation of the CoKriging function for the cheap samples. The augmented vector has a dimension of three, with $x_i^{aug} = [AoA_i, M_i, \hat{\eta}(x_i)]$. The corresponding flutter speeds for each of the n_i sample points are denoted as $y_i = y(x_i)$.

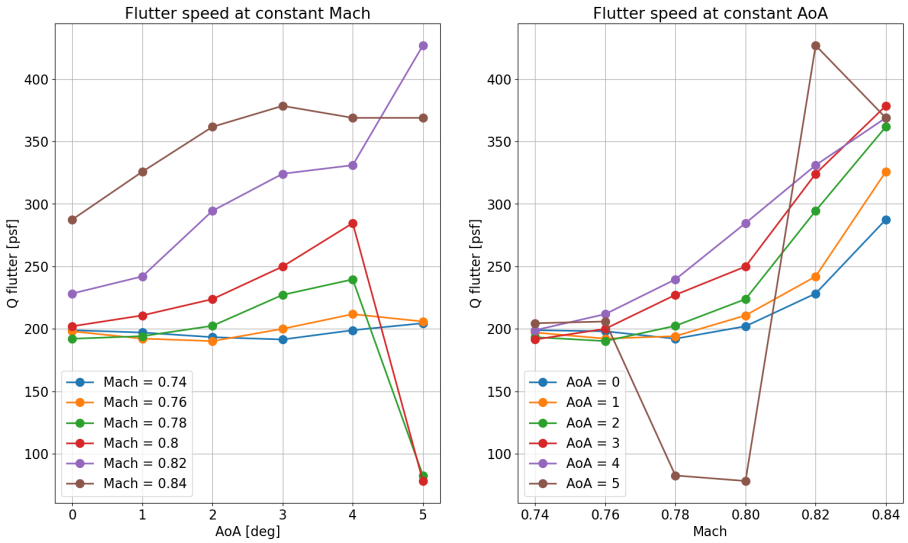


Figure 9: Flutter speed at constant Mach and constant AoA for BSCW wing.

Next, a CoKriging function is calculated for these augmented samples, $\hat{\eta}(x_i^{aug})$, with the additional component providing information for the correlation calculation from the cheap samples. This allows for a more accurate prediction of flutter speed, by combining the strengths of both high- and low-fidelity evaluations. Figure 10 shows the new flutter boundary obtained with the data fusion technique proposed, which results shift towards lower dynamic pressures while preserving its figure. Red circles denote experimental data from Dansberry et al. [22].

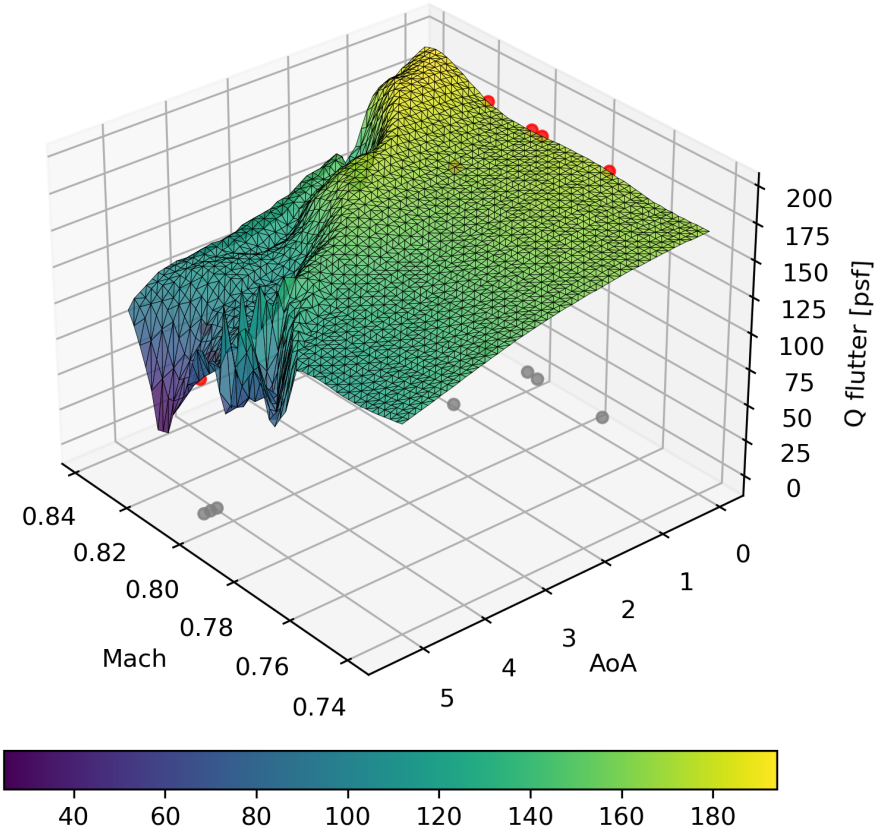


Figure 10: Flutter boundary for BSCW wing with data fusion. Red circles denote experimental data from Dansberry et al. [22]. Grey circles indicate the projection of the experimental data on the Mach-AoA plane.

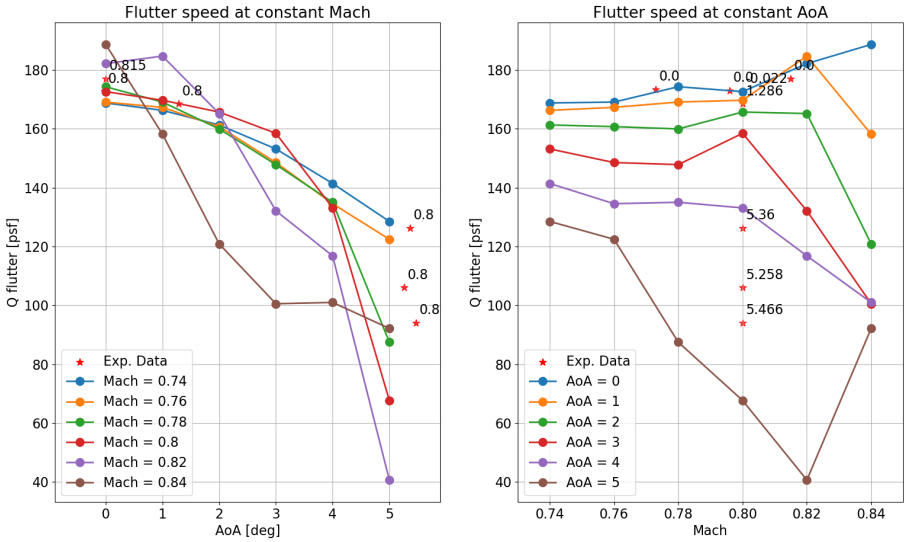


Figure 11: Flutter speed at constant Mach and constant AoA for BSCW wing with data fusion. Red stars denote experimental data from Dansberry et al. [22].

4 REFERENCES

[1] Immordino, G., Da Ronch, A., and Righi, M. (2025). Parametric nonlinear volterra series via machine learning: Transonic aerodynamics. *Journal of Aircraft*, 1–18.

[2] Candon, M., Hale, E., Delgado-Gutiérrez, A., et al. (2024). Optimal sparsity in nonlinear nonparametric reduced order models for transonic aeroelastic systems. *AIAA Journal*, 62(10), 3841–3856.

[3] Candon, M., Marzocca, P., and Dowell, E. H. (2025). Aeroelastic reduced-order model differential equations in transonic buffeting flow. *arXiv preprint arXiv:2510.22216*.

[4] Candon, M., Delgado-Gutiérrez, A., Marzocca, P., et al. (2025). Nonlinear aeroelastic reduced order modeling with optimized sparse multi-input volterra kernels. *AIAA Journal*, 1–16.

[5] Candon, M. J., Delgado-Gutiérrez, A. J., Marzocca, P., et al. (2025). Stall flutter of the benchmark supercritical wing using aeroelastic model reduction and tunable turbulence parameters. In *AIAA SCITECH 2025 Forum*. p. 0816.

[6] Candon, M. J., Marzocca, P., and Dowell, E. (2026). Sparse identification of aeroelastic reduced-order models in transonic buffeting flow. In *AIAA SCITECH 2026 Forum*. p. 0812.

[7] Dawson, S. T., Zayas, J. P., and Lopez-Doriga, B. (2025). Sparsity-promoting methods for isolating dominant linear amplification mechanisms in wall-bounded flows. *International Journal of Heat and Fluid Flow*, 115, 109872.

[8] Gazit, T. and Freydin, M. (2026). Reduced-order modeling of unsteady aerodynamics using constrained optimization and random input. In *AIAA SCITECH 2026 Forum*. p. 0185.

[9] Dodd, T. J. and Harrison, R. F. (2002). A new solution to volterra series estimation. *IFAC Proceedings Volumes*, 35(1), 67–72.

[10] Levin, D., Bastos, K. K., and Dowell, E. H. (2022). Convolution and volterra series approach to reduced-order modeling of unsteady aerodynamic loads. *AIAA Journal*, 60(3), 1663–1678.

[11] Israelsen, B. W. and Smith, D. A. (2014). Generalized laguerre reduction of the volterra kernel for practical identification of nonlinear dynamic systems. *arXiv preprint arXiv:1410.0741*.

[12] Zheng, Q. and Zafiriou, E. (2004). Volterra- laguerre models for nonlinear process identification with application to a fluid catalytic cracking unit. *Industrial & engineering chemistry research*, 43(2), 340–348.

[13] Immordino, G., Da Ronch, A., and Righi, M. (2023). Deep–learning framework for aircraft aerodynamics prediction. In *AIAA AVIATION 2023 Forum*. p. 0179.

[14] Isogai, K. (1979). On the transonic-dip mechanism of flutter of a sweptback wing. *AIAA journal*, 17(7), 793–795.

[15] Alonso, J. and Jameson, A. (1994). Fully-implicit time-marching aeroelastic solutions. In *32nd Aerospace Sciences Meeting and Exhibit*. p. 56.

[16] Jacobson, K., Stanford, B., Kiviaho, J. F., et al. (2021). Multiscale mesh adaptation for transonic aeroelastic flutter problems. In *AIAA Aviation 2021 Forum*. p. 2700.

[17] Prananta, B., Hounjet, M., and Zwaan, R. (1998). Two-dimensional transonic aeroelastic analysis using thin-layer navier–stokes method. *Journal of Fluids and Structures*, 12(6), 655–676.

[18] Heeg, J., Chwalowski, P., Raveh, D. E., et al. (2015). Plans and example results for the 2nd aiaa aeroelastic prediction workshop. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. p. 0437.

[19] Heeg, J., Chwalowski, P., Schuster, D., et al. (2013). Overview of the Aeroelastic Prediction Workshop. In *51st AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics Inc, AIAA. doi:10.2514/6.2013-783.

[20] Raveh, D. E., Yossef, Y. M., and Levy, Y. (2018). Analyses for the second aeroelastic prediction workshop using the EZNSS code. In *AIAA Journal*, vol. 56. American Institute of Aeronautics and Astronautics Inc. ISSN 00011452, pp. 387–402. doi:10.2514/1.J055960.

[21] Da Ronch, A., Ghoreyshi, M., and Badcock, K. (2011). On the generation of flight dynamics aerodynamic tables by computational fluid dynamics. *Progress in Aerospace Sciences*, 47(8), 597–620.

[22] Dansberry, B., Durham, M., Bennett, R., et al. (1993). Experimental unsteady pressures at flutter on the supercritical wing benchmark model. In *34th Structures, structural dynamics and materials conference*. p. 1592.