# Adaptive sampling for interpolation of reduced-order aeroelastic systems*

Norberto Goizueta†, Andrew Wynn‡ and Rafael Palacios§
*Department of Aeronautics, Imperial College London*

**A new strategy for the interpolation of parametric reduced-order models of dynamic aeroelastic systems is introduced. Its aim is to accelerate the numerical exploration of geometrically-nonlinear aeroelastic systems over large design spaces or multiple flight conditions. The parametric reduced-order models are obtained from high-dimensional models by Krylov subspace projection. They are subsequently interpolated to acquire realizations inexpensively anywhere in the parameter space, where having the state-space as opposed to interpolating an output metric permits the use of linear analysis tools. The interpolation scheme is heavily conditioned by the available training points, thus a novel methodology based on adaptive sampling and a combinatorial use of the available true-systems knowledge is presented, whereby we reuse all known data of the true models as different combinations of training and testing data to build statistical surrogates of the interpolation error across the parameter space and refine the sampling in those regions that need it. This minimizes the number of costly-to-evaluate functions calls and ensures that parameter space regions are sampled according to the underlying system dynamics. The initial implementation of this adaptive sampling strategy is demonstrated on a very flexible wing with a complex stability envelope.**

## I. Introduction

Aircraft certification agencies demand, among other requirements, knowledge of structural loads and stability characteristics within all flyable conditions in the flight envelope. Effectively, this means evaluating the aircraft system at hundreds of thousands of operational points [1], which will be combinations of, for example, flight velocity, altitude, orientation, payload and atmospheric conditions. To reduce the number of evaluations, engineering judgment based on past experience on similar aircraft models is typically employed [2] although, clearly, this can be challenging for new aircraft concepts. In this paper, we will address this issue using state-space interpolation methods of the aeroelastic system to efficiently explore the design space. Although most methods presented in this paper are not

*This paper is based on those originally presented as Paper 2021-1798 at the AIAA 2021 SciTech Forum, Virtual event, 11-15 & 19-21 January 2021; and Paper 2022-1345 at the AIAA 2022 SciTech Forum, San Diego, CA & Virtual, 3-7 January 2022.

†PhD Student (norberto.goizueta13@imperial.ac.uk), AIAA Student Member
‡Reader in Control and Optimization (a.wynn@imperial.ac.uk)
§Professor in Computational Aeroelasticity (r.palacios@imperial.ac.uk), AIAA Associate Fellow

model specific, we will focus on slender, light-weight wings that are capable of achieving deformations comparable to their semi-span. Such large deformations may and will affect both the structural and aerodynamic characteristics and therefore nonlinear aeroelastic models are required to capture the interaction between the two. These high aspect ratio wings are typically found in new and unconventional aircraft designed for perpetual flight that use solar energy to fulfill their power-constrained mission.

The aeroelastic models, even though typically simplified in their underlying physical modeling (structures are modeled using beams and aerodynamics may be modeled using potential flow panel methods [3]), are nonlinear and high-dimensional. This large model size is usually a contribution of the unsteady aerodynamic models which tend to require fine mesh discretisation in order to converge to steady-state results and have sufficient wake lengths to capture significant unsteady flow histories. Converged models of order $10^4 - 10^5$ states are the norm [4]. These systems may be described by flight control inputs (thrust and control surface deflections) and external environmental inputs (gust disturbances), parametrised in terms of operational conditions (payload, altitude) and the outputs be aircraft velocities, structural loads or wing deformations. Although the geometric and design characteristics of slender flexible wings allow for simpler physical models without significant tradeoffs in accuracy, the resulting systems of the governing nonlinear equations can still be very large in size, and for the purpose of design and loads evaluation across the entire flight envelope, simply unfeasible. Across the field, much research is focused on speeding up the analysis of aeroelastic systems [5].

Therefore, in this work we present a roadmap on efficient methodologies for the loads and design evaluation of aeroelastic systems. Starting from the nonlinear equations, we linearize the system in order to benefit from linear analysis tools, such as frequency domain analysis that does not require time marching of the equations to compute dynamic loads. However, linearisation does not resolve the issue of large system dimension and model reduction methods are needed in order to successfully use these tools. In addition, as previously mentioned, these vehicles are capable of undergoing large deformations which will change their shape and structural and aerodynamic properties, with significant couplings and hence the nonlinear formulations. Thus, we aim to use state-space interpolation to obtain, at any point in the $p$-dimensional aircraft parametric design space, $\boldsymbol{p} \in \mathbb{R}^p$, a reduced linear state-space realization

$$\hat{\Sigma}(\boldsymbol{p}) := \begin{cases} \dot{\boldsymbol{x}}(t) = \boldsymbol{A}(\boldsymbol{p})\boldsymbol{x}(t) + \boldsymbol{B}(\boldsymbol{p})\boldsymbol{u}(t) \\ \boldsymbol{y}(t) = \boldsymbol{C}(\boldsymbol{p})\boldsymbol{x}(t) + \boldsymbol{D}(\boldsymbol{p})\boldsymbol{u}(t) \end{cases} \tag{1}$$

that is locally accurate. With efficiency as a prime constraint, we seek to be able to interpolate the reduced-order state-space matrices $(\boldsymbol{A}(\boldsymbol{p}), \boldsymbol{B}(\boldsymbol{p}), \boldsymbol{C}(\boldsymbol{p}), \boldsymbol{D}(\boldsymbol{p}))$ with minimal training points as, despite them being small, they are costly to derive from the nonlinear equations and subsequent linearisation and reduction. Therefore, we introduce a novel Bayesian sampling method based on statistical surrogates and data-recycling to obtain said training points for an

optimal performance of the interpolator which can offer adequate visibility into the design-spaces of flexible aircraft for fractions of the cost of classical evaluations.

### A. Low speed aeroelasticity

The multidisciplinary requirement of these models brings together different fields, most notably structural dynamics and aerodynamics, and for each discipline high fidelity methods are widely used in a reliable manner, albeit often at great computational expense. Coupling these together, namely full 3D finite element models (FEM) with boundary-layer resolving computational fluid dynamics (CFD) solvers is not a trivial procedure and computationally demanding problem. Thus, usually, for high aspect ratio aircraft the first simplifying assumption comes on the structural side, employing one-dimensional beam models for slender wings which can be more easily coupled with the aerodynamic model [6, 7]. Beam models capable of capturing the nonlinear effects resulting from large deformations and finite rotations are now standard use in aeroelasticity, with several models for geometrically exact beam formulations available, notably intrinsic [8], displacement-based [9], or strain-based [10], but also others such as the modal rotation method [11, 12]. Likewise, simplifications of the Navier-Stokes equations governing the fluid can be made thanks to the design characteristics of typical high altitude long endurance aircraft, where slender, thin lifting surfaces prevail and the fluid is mostly attached, therefore viscous effects can be assumed confined to thin boundary layers and neglected for dynamic aeroelastic purposes. Thus, the assumptions of inviscid, incompressible and irrotational flow become appropriate and lead to the potential flow equations and solution methods such as the Unsteady Vortex Lattice Method (UVLM) [13] or the Doublet Lattice Method (DLM) [14], which is expressed in frequency-domain and of standard use in the aeroelastic analysis of transonic aircraft [15]. The aeroelastic models using these medium fidelity methods are usually sufficiently accurate to capture the dynamic behavior of low speed, high aspect ratio aircraft and a further review of these models can be found in [3, 16].

### B. Model reduction methods

Despite linear model order reduction being a well-developed, even classical, methodology, the performance of such techniques is model specific, especially if very high-dimensional systems are studied. The type of system, the system characteristics, the system size and the desired properties of the ROM make certain methods more appropriate than others. A review of common model reduction methods for linear systems can be found in [17]. Model order reduction of nonlinear systems is still an area of active research [18, 19]. For linear systems, most techniques can be categorized as either SVD methods or Krylov methods. The former include techniques such as proper-orthogonal decomposition, likely the most common type of ROM used for aerodynamic systems [20–23], or balanced truncation methods, for which a novel and efficient approach using frequency-limited truncation tailored to the UVLM was presented recently [24, 25]. SVD-based methods tend to preserve stability if the high-order model is stable but are also usually more expensive

to compute. On the other hand, Krylov-based methods [26–28] are very well suited to tackle very large systems and perform well if a particular frequency spectrum is of interest (unsurprisingly, the Arnoldi iteration used to construct Krylov subspaces is used extensively in the computing process of eigenvalues of large matrices [29]). Most model order reduction methods though require the input-output dimensionality to be small, be it in balanced realizations [30, 31] or Krylov-based since these are based on transfer function interpolation and a large number of transfer functions may yield the ROM process futile. Therefore, in this work, to reduce a large dimensional UVLM ($10^4$ states and $10^3$ inputs/outputs), we project it onto the structural modal space that reduces the inputs and outputs considerably onto a few structural modes and enables an efficient reduction of the projected UVLM linear system. In addition, this projection also simplifies the aeroelastic coupling, as the structural system will be expressed in the same modal coordinates. Projection of the aerodynamic input/output channels onto a lower dimensional subspace by means of other methods would be equally suited; however, as the objective is to couple them with structure, the vibration modes make for an ideal candidate. This approach has been used for the applications in [4, 32, 33] successfully and the Krylov reduction process for these MIMO systems is outlined in detail in this paper. The Krylov-reduced linear UVLM is then coupled to a linearised structural model reduced by modal truncation [34] resulting in a linear, parametric reduced order aeroelastic state-space.

The process of obtaining said aeroelastic models is not computationally cheap for refined meshed models: first, the static nonlinear equations are solved, the large UVLM is linearised analytically [24] and then reduced using the Krylov subspace method. The process of doing so at every point in the operational envelope is possible, although not quick (as done in [4] for two parameters), but obviously may become intractable for higher dimensional parameter spaces. Therefore, we seek with state-space interpolation the ability to obtain an aeroelastic state-space realization anywhere in the parameter space with only a few training points (computed using the process described). Interpolation between aerodynamic reduced order models has been performed in [25, 35]. The interpolation between reduced order state-space systems [36] is attractive as it is computationally efficient and we retain full system information and simulation capabilities (frequency response and eigenvalues). In order to interpolate the reduced order systems, they need to be adjusted such that the interpolation is performed between a compatible coordinate space [37], where the dynamics of the systems change smoothly between them.

### C. Parameter space sampling methods

The performance of any multi-dimensional interpolation scheme is highly dependent on the training points available. Regular grids soon suffer from the so-called “curse of dimensionality” thus other methods are sought. Non-regular, but also non-adaptive, grid sampling such as Latin Hypercube Sampling (LHS), may fail to sample with sufficient resolution those areas of the parameter space where dynamics change the most and are a driver of the interpolation error. In the case of LHS, for instance, a large number of points may be required to sample with sufficient resolution the

parameter space and capture the changing dynamics [35]. Therefore, we seek novel methods to sample the parameter space adaptively. To do so, we employ statistical surrogate models of an underlying interpolation error function in order to determine where more resolution in the training dataset is required without incurring prohibitive costs calling physical based models.

The use of statistical surrogates is common in various disciplines and often found in multi-disciplinary design optimization (MDO) to efficiently explore the optimization domain, for instance, using design of experiments methods, such as LHS [38]. The number of sampling cases, which is a key-driver of the MDO's cost, can be reduced using adaptive sampling to refine the surrogate models based on residual error, information gain or weighted-information gain [39]. Other applications of adaptive sampling can be found in surrogate-based optimization [40], which uses optimal LHS to obtain adequate distributions of training points in the design space; or information-gain-based approaches used in global optimization [41] and optimal experiment design. For the latter, we find cases most similar to our application such as seeking an optimal experimental set-up by placing sensors at places of maximum expected information gain [42] or by finding, using statistical models, the sets of experiments that provide most information about a target set of parameters [43].

At a time where data-driven technologies are increasing in popularity for numerous applications across all disciplines — of particular relevance to this paper in aeroelasticity [44–47] — and even certification agencies have begun to plan towards integrating them into aircraft certification processes [48], we consider our approach as hybrid between data-driven and physics-based modeling. This scheme is built on multiple training physical models as opposed to collections of input-output data more typically used in "classical" data-driven approaches. The use of novel machine learning data-driven methods can even hopefully be used to replace the need for linearisation and reduction of the nonlinear aeroelastic system by a data-driven model, and recent work has successfully acquired reduced data-driven unsteady aerodynamic models from the UVLM using aircraft simulation data on multiple trajectories [47].

In this paper we will first give an overview of typical aeroelastic models of very flexible wings, starting from the nonlinear formulation to the linearised form in Sec. II.A, followed by the reduction process of the aerodynamic system using Krylov subspaces in Sec. II.B. Then, we will detail how the systems must be pre-processed for interpolation (Sec. III.A) before showing a novel sampling scheme to optimally acquire the optimal training points for the interpolation method (Sec. III.B). In addition, we will highlight the limitations and lines of further research to use interpolation methods to robustly speed up the analysis and/or optimization process of aeroelastic systems. This approach will be demonstrated in Sec. IV on the analysis of the Pazy wing [49], a very flexible wing designed for wind tunnel flutter experiments, whose stability envelope and flutter characteristics have been previously analyzed in [4]. The nonlinearities present in the Pazy wing model require the linearization and eigenvalue evaluation at each point of interest in the parameter design space in order to determine the flutter solution, which is a costly process. Thus, we seek with this approach to significantly reduce the computational time in which the flutter envelope is obtained, while also paving the

way for loads analysis in an analogous manner. The framework is developed in `SHARPy` (Simulation of High Aspect Ratio aeroplanes and wind turbines in Python) [50], our in-house open-source, nonlinear aeroelastic simulation toolbox.

## II. Aeroelastic solver

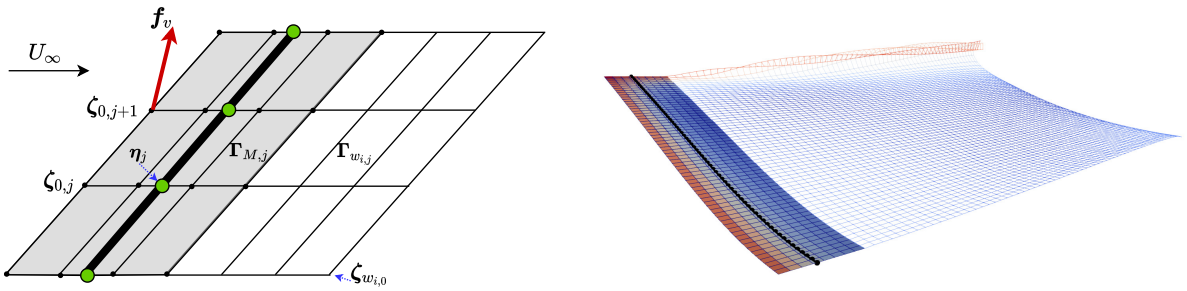
### A. Full Order Model

The structural model is based on a geometrically-exact composite beam formulation, with linear constitutive relations and nonlinear kinematic relations. The formulation is parametrized in displacements and rotations (using the Cartesian rotation vector) and implemented by discretizing the beams in quadratic (3-node) finite elements [9, 51]. The discretized structural equations [52, 53], that result from applying Hamilton's principle include — if necessary — the rigid body equations of motion, which are solved simultaneously about an arbitrary reference node (not the center of gravity).

The aerodynamics are solved using the Unsteady Vortex Lattice Method [13, 54], based on the inviscid, incompressible and irrotational assumptions of potential flow. These are appropriate as for the mostly attached nature of the flow over the thin lifting surfaces, viscous effects can be assumed to be confined to thin boundary layers and are neglected. The mesh consists of a lattice of vortex rings projected over the lifting surfaces of the structure, with the spanwise coordinates of the panels coincident with the structural nodes. The wake is modeled by a discretized sheet of vortex panels that can adopt any arbitrary shape, including wake roll-up effects. The circulation strength of the vortex panels is solved enforcing a non-penetration boundary condition on the lifting surfaces. With the circulation of the vortices solved for, the steady and unsteady aerodynamic forces can be computed [53, 55, 56]. A typical wing and wake vortex lattice is shown in Fig. 1.

A linearized aeroelastic model can then be built around an equilibrium reference condition under the assumption of small perturbations to the degrees of freedom about a, possibly, significantly deformed reference state. On the aerodynamic side, the linearization of the UVLM is performed analytically as in [24] assuming constant aerodynamic influence coefficients and a frozen, yet arbitrary, wake shape and results in a discrete, linear time invariant (DLTI) system. The state vector corresponds to perturbations to the bound and wake circulations, $\mathbf{\Gamma} \in \mathbb{R}^{K_\Gamma}$ and $\mathbf{\Gamma}_w \in \mathbb{R}^{K_{\Gamma_w}}$, circulation time derivative, $\Delta t \dot{\mathbf{\Gamma}} \in \mathbb{R}^{K_\Gamma}$, and circulation at the previous time step $\mathbf{\Gamma}^{n-1} \in \mathbb{R}^{K_\Gamma}$. The inputs are the lattice grid displacements and velocities, $\boldsymbol{\zeta} \in \mathbb{R}^{3K_\zeta}$ and $\dot{\boldsymbol{\zeta}} \in \mathbb{R}^{3K_\zeta}$ and external fluid disturbances (i.e. gusts) $\boldsymbol{w} \in \mathbb{R}^{3K_\zeta}$. Finally, the output is the vector of normalized aerodynamic forces at the lattice vertices $\boldsymbol{f}_v \in \mathbb{R}^{3K_\zeta}$. The size of these vectors is dependent on the number of UVLM bound panels, $K_\Gamma$, the number of wake panels, $K_{\Gamma_w}$ and the number of panel vertices, $K_\zeta$. Additional inputs can be introduced to the input vector, such as control surfaces deflections and deflection rates, $\delta_e$ and $\dot{\delta}_e$ and prescribed profiles for external gust disturbances.

The structural subsystem linearization is more straight-forward: the tangent mass, damping and stiffness matrices [31] at the reference state are used to construct a first-order discrete-time system using a Newmark-$\beta$ integration scheme. This




**(a) Schematic of the vortex lattice, beam and relevant variables.**

**(b) Example of vortex lattice discretisation and beam on a very flexible wing with wake roll-up effects.**

**Fig. 1 Representation of the vortex lattice method and geometrically exact beam model.**

is then coupled with the DLTI UVLM. It is useful to project the structural equations, parametrised by the displacements and rotations of the finite element nodes, $\boldsymbol{\eta} \in \mathbb{R}^{6(n_{nodes}-1)}$, onto modal coordinates, $\boldsymbol{\eta} = \boldsymbol{\Phi}\boldsymbol{q}$, where $\boldsymbol{\Phi} \in \mathbb{R}^{6(n_{nodes}-1)\times n_q}$ is the rectangular matrix retaining $n_q$ structural mode shapes and $\boldsymbol{q} \in \mathbb{R}^{n_q}$ is the modal coordinate space. This reduces the dimension of the structural state-space and can the aerodynamic model inputs and outputs can also be projected onto this modal space. Once the structural state-space is assembled, the states, $\boldsymbol{x}_s$, and outputs, $\boldsymbol{y}_s$, are the modal displacements and velocities, $\boldsymbol{y}_s = \boldsymbol{x}_s = [\boldsymbol{q}^\top, \dot{\boldsymbol{q}}^\top]^\top \in \mathbb{R}^{2n_q}$ and the inputs correspond to the modal forces $\boldsymbol{Q} \in \mathbb{R}^{n_q}$.

The number of panels is typically large since refined discretizations are often required in order to achieve convergence for flutter analyses [24, 25, 57]. Thus, the resulting linearized UVLM systems may be of the order of $10^4 - 10^5$ states and $10^3$ inputs, yet the input/output space dimension can be substantially reduced by expressing them in terms of structural modes. However, the large dimension of the aerodynamic state prevents the use of traditional linear analysis (eigenvalues or frequency response) on typical desktop hardware, therefore the need to turn to model reduction techniques, which are described next.

### B. Krylov-based model reduction of the DLTI UVLM and reduced aeroelastic model

The large dimension of the UVLM system (typically around $10^4 - 10^5$ states) makes the system analysis or interpolation between models impractical on modern hardware, thus the need to employ model order reduction techniques to reduce the system to a manageable size. Krylov subspace, or moment-matching, methods provide a suitable reduction process for two reasons: computational efficiency and the ability to match the system's transfer function at a desired frequency [58]. This will be the lower end of the spectrum given the underlying assumptions of the UVLM method which make it valid for low reduced frequencies. The Krylov-based reduction of the linear UVLM subsystem of the aeroelastic model has been described in detail in [33], and has been hitherto applied as "black box" reduction method [4, 32, 33]. This section will therefore summarize the basics of the method, and will use the numerical example to showcase the performance of the Krylov method applied to the UVLM system, as well as supporting the UVLM-specific algorithmic choices made based on the system and reduction method properties.

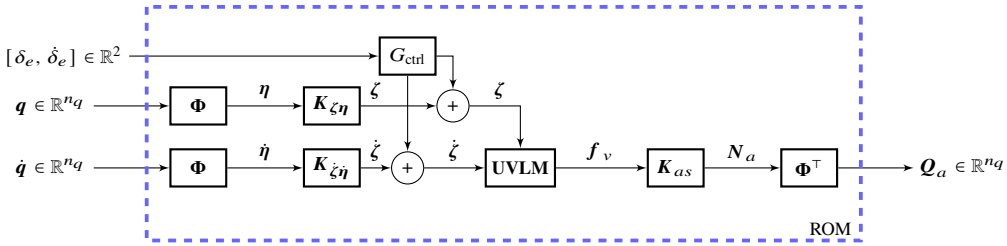



**Fig. 2 Block diagram description of the reduced aerodynamic system projected onto modal coordinates with control surface inputs.**

The reduction process of the linear UVLM is a two-step approach. The first is that the input/output dimension of the transfer function matrix $\boldsymbol{G}(s) = \boldsymbol{C}(\boldsymbol{I}s - \boldsymbol{A})^{-1}\boldsymbol{B} + \boldsymbol{D}$ for $s \in \mathbb{C}$ must be reduced. The nature of the Krylov reduction method — interpolating transfer functions — requires this as otherwise the reduction will be ineffective. Therefore, we leverage the aeroelastic nature of the problem to project the linear UVLM input (bound lattice vertex displacements and velocities) and outputs (forces at bound lattice vertices) onto the structural modal space, given by the eigenvector matrix and modal coordinates $(\boldsymbol{\Phi}, \boldsymbol{q})$, translating the inputs into modal displacements and velocities and the outputs to modal forces, while also reducing the structural system. Typically the number of retained structural modes is small and a reduction in the number of inputs from typically $10^3$ to approximately 20 is common.

With the UVLM expressed in reduced input/output spaces, the Krylov-subspace method can now be used effectively. A Krylov subspace of order $i$ associated with two matrices $\boldsymbol{F} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{g} \in \mathbb{R}^{n\times m}$ is defined as

$$\mathcal{K}_i(\boldsymbol{F}, \boldsymbol{g}) := (\boldsymbol{g},\ \boldsymbol{F}\boldsymbol{g},\ \ldots,\ \boldsymbol{F}^{i-1}\boldsymbol{g}). \tag{2}$$

This method is a projection-based method (either Galerkin or Petrov-Galerkin), where the projectors, $\boldsymbol{V}$ and $\boldsymbol{Z}$, are chosen to span the controllability or observability Krylov subspaces of the system. For linear models obtained by linearizing UVLM, the particular Krylov subspaces of interest for model reduction have the form

$$\mathcal{K}_{r_b}\left((\sigma\boldsymbol{I} - \boldsymbol{A})^{-1}, (\sigma\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{B}\right) \subseteq \mathcal{V} \text{ and} \tag{3a}$$

$$\mathcal{K}_{r_c}\left((\sigma\boldsymbol{I} - \boldsymbol{A})^{-\top}, (\sigma\boldsymbol{I} - \boldsymbol{A})^{-\top}\boldsymbol{C}^\top\right) \subseteq \mathcal{Z}. \tag{3b}$$

The reduced order model is then assembled by projection onto these subspaces, where the orthogonal projectors are chosen to span the images of the above subspaces

$$\mathcal{V} = \text{span}(\boldsymbol{V}) \text{ and } \mathcal{Z} = \text{span}(\boldsymbol{Z}), \tag{4}$$

with the additional constraint in the selection that $\boldsymbol{Z}^\top\boldsymbol{V} = \boldsymbol{I}$ for Petrov-Galerkin projections. If only a single Krylov-

subspace is used the projection matrices are orthogonal by nature of the assembly algorithm. Note that the reduced order bases can span multiple Krylov subspaces that are built about different points in the complex plane in order to improve the matching on a broader frequency spectrum provided the orthogonality conditions are still satisfied. Thence, the state can be approximated by $\boldsymbol{x} \approx \boldsymbol{V}\hat{\boldsymbol{x}}$ and the resulting reduced order model by projection onto the Krylov subspaces will satisfy the moment-matching condition at $\sigma$, i.e. the reduced order model transfer function will match $r$ (Krylov subspace order) derivatives at that point, provided $\sigma$ is not an eigenvalue of $\boldsymbol{A}$ [59, Ch. 11].

In terms of the actual assembly of the reduced order bases, the projection matrices that span the Krylov subspaces are constructed using the Arnoldi iteration with a Modified Gram-Schmidt orthogonalization method [33]. In a nutshell, the $\boldsymbol{V}$ and $\boldsymbol{Z}$ matrices are assembled by sequentially concatenating vectors of the form

$$v_i^+ = (\sigma \boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{b}_i \text{ or } z_j^+ = (\sigma \boldsymbol{I} - \boldsymbol{A})^{-\top}\boldsymbol{c}_j^\top \tag{5}$$

where $\boldsymbol{b_i}$ is the $i$-th column vector of $\boldsymbol{B}$; or, for the observability subspace, $\boldsymbol{c}_j$ is the $j$-th row vector of $\boldsymbol{C}$. The majority of the computational cost comes from finding $(\sigma \boldsymbol{I} - \boldsymbol{A})^{-1}$, however, this is not done explicitly but rather (5) is solved using an LU decomposition and the LU factors of $(\sigma \boldsymbol{I} - \boldsymbol{A})$ stored in memory. Thus, for subsequent vectors appended to the projector matrices, only matrix-vector multiplications are employed. The new vectors $v_i^+$ or $z_j^+$ are then orthogonalized with respect to the assembled projection matrix up to that iteration using the Modified Gram-Schmidt method. If any new vector is linearly dependent to those already part of the projection matrix they are not added. For more details on the assembly of Krylov subspaces the reader is referred to [33, 58, 60].

Although Krylov subspace methods have the disadvantage of not offering an *a priori* error bound nor preserve stability, previous numerical results [33, 60] and the ones shown in Sec. IV.A suggest that by just using the observability subspace (as opposed to two-sided moment matching) and interpolation about the steady-state gain, the resulting reduced systems by Galerkin projection preserve the stability of the aerodynamic system, which was also observed in the UVLM model order reduction process using frequency-limited balanced truncation in [25]. Applying the single-sided Krylov reduction using the output space of the DLTI UVLM (3b) offers the additional advantage that the size of the resulting ROM will be dependent on the number of outputs, $n_p$, only (which is half the number of inputs, $n_m$, as seen on Fig. 2) and the number of moments matched $r$ which, if there are no linear dependencies between the columns, will be $k_a = rn_p$. This projector $\boldsymbol{Z} \in \mathbb{R}^{n \times k_a}$ is orthonormal ($\boldsymbol{Z}^\top \boldsymbol{Z} = \boldsymbol{I}$), spans the observability Krylov subspace defined at $\sigma$ and can be used to assemble the reduced order DLTI UVLM system with $m$ inputs, $p$ outputs and the newly defined

reduced state $\boldsymbol{x}_{\mathrm{a}} \approx \boldsymbol{Z}\hat{\boldsymbol{x}}_{\mathrm{a}}$ as

$$\hat{\Sigma}_a := \left(\begin{array}{c|c} \hat{\boldsymbol{A}}_a & \hat{\boldsymbol{B}}_a \\ \hline \hat{\boldsymbol{C}}_a & \hat{\boldsymbol{D}}_a \end{array}\right) \text{with } \begin{cases} \hat{\boldsymbol{A}}_a = \boldsymbol{Z}^\top \boldsymbol{A}_a \boldsymbol{Z} \in \mathbb{R}^{k\times k} \\ \hat{\boldsymbol{B}} = \boldsymbol{Z}^\top \boldsymbol{B}_a \in \mathbb{R}^{k_a \times n_m} \\ \hat{\boldsymbol{C}} = \boldsymbol{C}_a \boldsymbol{Z} \in \mathbb{R}^{n_p \times k_a} \\ \hat{\boldsymbol{D}} = \boldsymbol{D}_a \in \mathbb{R}^{n_p \times n_m} \end{cases} . \tag{6}$$

The choice of $\sigma \in \mathbb{C}$ is arbitrary and can be chosen according to the physical properties of the system. In the case of the DLTI UVLM, the frequencies of interest lie in the low frequency end of the spectrum given the characteristics of the UVLM. In addition, the choice of a real value for $\sigma$ can be considered to maintain $\boldsymbol{Z}$ purely real; however, work has been done with $\sigma \in \mathrm{i}\mathbb{R}$ with successful outcome (i.e. the ROM matches at the desired frequency), despite the system matrices having complex terms with a consequent increase in the required memory for storage and increased likelihood of unstable realizations of the reduced order model. Therefore the choice of $\sigma = 0$ as the ROM interpolation point is the prime requirement to achieve in practice stable realizations, with the additional benefit of keeping the matrices real.

Since the reduced aerodynamic system is already projected onto the structural modal space, the coupling with the linearised, reduced structural subsystem by modal truncation is trivial, and the reduced order aeroelastic state becomes $\boldsymbol{x}_{ae} = [\hat{\boldsymbol{x}}_a^\top, \boldsymbol{q}^\top, \dot{\boldsymbol{q}}^\top]^\top \in \mathbb{R}^k$, where the size $k = k_a + 2n_q$. This process: finding the static nonlinear aeroelastic equilibrium, linearisation and reduction is schematically depicted in Fig. 3 and can be used parametrically to obtain a reduced model at any point $\boldsymbol{p}$ in the $p$-dimensional parameter design space, which can include angle of attack, velocity, payload, etc., and it is a function of the parameter $\boldsymbol{x}_{ae}(\boldsymbol{p})$, since the aeroelastic state has been reduced by projection of the bases defined at a particular point in the parameter space.

It is convenient to transform the system to continuous time system using a bi-linear transformation [59] to avoid interpolating between discrete-time systems with different time steps. Once transformed into continuous time, artificial dynamics may appear above the Nyquist frequency, $\omega_N = \pi/\Delta t$, given by the temporal discretisation $\Delta t$, not present in the original discrete-time system. Although the Nyquist frequency provides an upper limit to the frequency features that can be captured by the DLTI, features above one half of its value can still be distorted by aliasing effects, therefore, features above one half of the Nyquist frequency are not considered.

This reduction completes the overall model order reduction process of the aeroelastic system ($\bullet_{\mathrm{ae}}$), which can be a function of a parameter $\boldsymbol{p}$ and is expressed in continuous-time as

$$\hat{\Sigma}_{ae}(\boldsymbol{p}) := \begin{cases} \dot{\boldsymbol{x}}_{ae}(t) = \boldsymbol{A}_{\mathrm{ae}}(\boldsymbol{p})\boldsymbol{x}_{ae}(\boldsymbol{p}, t) + \boldsymbol{B}_{\mathrm{ae}}(\boldsymbol{p})\boldsymbol{u}_{ae}(t) \\ \boldsymbol{y}_{ae}(t) = \boldsymbol{C}_{\mathrm{ae}}(\boldsymbol{p})\boldsymbol{x}_{ae}(\boldsymbol{p}, t) + \boldsymbol{D}_{\mathrm{ae}}(\boldsymbol{p})\boldsymbol{u}_{ae}(t) \end{cases} . \tag{7}$$

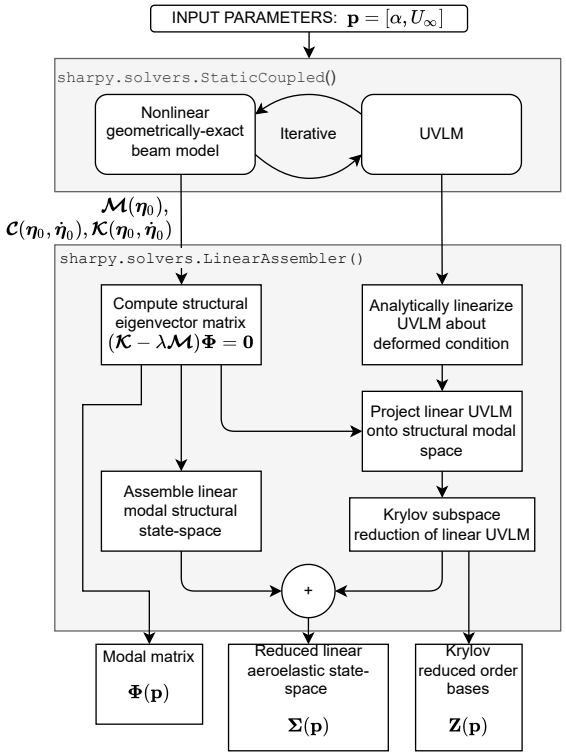



**Fig. 3 Parametric linear system assembly process for use in state-space interpolation schemes using SHARPy.**

## III. State-space interpolation of parametric reduced order models

Interpolation between state-spaces can be an effective way to explore the design space of a dynamical system, provided the changes in the dynamics occur smoothly. This avoids computing full order models across it, which would be an expensive endeavor. The advantage of interpolating state-spaces as opposed to output metrics, for example root loads, is that the whole array of linear analysis tools (frequency response, PSDs, etc.) can be subsequently used for any interpolated point in the design space. If the state-spaces are reduced and small in size, these tools can be used at very low computational expense and yield valuable system information at a fraction of the time required to obtain the same model following the entire physical derivation process.

The process of state-space interpolation can be divided in three parts: i) interpolating functions; ii) selection the interpolation source or *training* points, at which the true systems are known; and iii) evaluation of the quality of the interpolation scheme by means of *testing* points. Additionally, given that we employ reduced order systems with state vectors obtained by projection of the reduced order bases, the reduced states derived at each point $\boldsymbol{p}_i$ may differ from one another, and therefore a compatibilization step is necessary to ensure that the state-spaces are interpolated between the same coordinates.

The process described in Sec. II can be used to obtain a library of reduced order aeroelastic models at different conditions across the operational envelope or design space, which will serve as the source points of the interpolation

method. This is schematically shown in Fig. 3. For each point a six-element tuple is stored: the four reduced order aeroelastic state-space matrices and the two reduced order bases (Krylov and modal matrices); the latter two are necessary to project all the systems in the library onto the compatible coordinate space:

$$(\boldsymbol{A}_{\text{ae}}(\boldsymbol{p}), \boldsymbol{B}_{\text{ae}}(\boldsymbol{p}), \boldsymbol{C}_{\text{ae}}(\boldsymbol{p}), \boldsymbol{D}_{\text{ae}}(\boldsymbol{p}), \boldsymbol{Z}(\boldsymbol{p}), \boldsymbol{\Phi}(\boldsymbol{p}))_i, \tag{8}$$

which is calculated at the desired operating point $\boldsymbol{p} \in \mathbb{R}^p$. For the purposes of interpolation, we can focus on the assembled aeroelastic system, in the form of (7), or interpolate the aerodynamic and structural systems independently, with the coupling made after the interpolation, in which case the relevant subsystem's matrices and reduced order bases are stored.

### A. Interpolation Scheme

The multi-dimensional interpolation is performed using a linear, nearest-neighbors scheme. In particular, the domain is normalized and tessellated into triangles employing a Delaunay triangulation, based on the Qhull implementation [61]. Thence, a linear barycentric interpolation is performed to obtain the relevant interpolation weights $w_i(\boldsymbol{p})$ at the new desired operating point, $\boldsymbol{p}_{N_K+1}$, that scale each entry in the library of $N_K$ reduced order systems. Due to the nature of this interpolation scheme, no extrapolation is possible. Naturally, errors in the prediction of the flutter boundary will arise as a result of the interpolation. However, it is worth noting that, for the interpolation of parameter-varying control systems, additional methods can be introduced to ensure the stability of the closed-loop system [62]. Although in the case of the open-loop system, stability-preserving algorithms could interfere with the prediction of physically occurring instabilities such as flutter.

The interpolated system at the desired operating point $N_K + 1$ has its matrices computed directly by

$$\boldsymbol{A}_{N_K+1} = \sum_{i=1}^{N_K} w_i(\boldsymbol{p}_{N_K+1}) \boldsymbol{A}_i(\boldsymbol{p}_i), \tag{9}$$

with the $\boldsymbol{B}$, $\boldsymbol{C}$ and $\boldsymbol{D}$ matrices calculated in an analogous manner [25, 36, 63].

However, the weighted interpolation process in (9) cannot be applied directly on the reduced systems matrices. Since the source models have been reduced by projection independently of one another, the reduced order bases ($\boldsymbol{Z}_i$, $\boldsymbol{\Phi}_i$) can span different subspaces and, therefore, it may be the case that the reduced state vectors $\boldsymbol{x}_{ae}(\boldsymbol{p}_i)$ represent a different linear combination of the full order model state, leading to an erroneous interpolation between systems with significantly different dynamics [64].

Consequently, it is necessary to find a common set of bases such that the reduced state vectors are compatible for interpolation. In this case, we follow the approach in [64], which is in nature equivalent to Step A in [36]. At each

sampling point, $\boldsymbol{p}_i$, the local reduced state vector, $\boldsymbol{x}_{ae}(\boldsymbol{p}_i)$ that was defined by

$$\begin{bmatrix} \boldsymbol{x}_a \\ \boldsymbol{\eta} \\ \dot{\boldsymbol{\eta}} \end{bmatrix} \approx \begin{bmatrix} \boldsymbol{Z}(\boldsymbol{p}_i) & & \\ & \boldsymbol{\Phi}(\boldsymbol{p}_i) & \\ & & \boldsymbol{\Phi}(\boldsymbol{p}_i) \end{bmatrix} \boldsymbol{x}_{ae}(\boldsymbol{p}_i) = \boldsymbol{W}(\boldsymbol{p}_i)\boldsymbol{x}_{ae}(\boldsymbol{p}_i) \tag{10}$$

can be transformed by a matrix $\boldsymbol{T}(\boldsymbol{p}_i)^{-1}$ to a new coordinate system $\boldsymbol{x}_{ae}(\boldsymbol{p}_i) = \boldsymbol{T}(\boldsymbol{p}_i)^{-1}\boldsymbol{x}_i^*$. These transformed coordinates are chosen as

$$\boldsymbol{x}_1^* = \boldsymbol{x}_2^* = \cdots = \boldsymbol{x}^*. \tag{11}$$

An appropriate selection of $\boldsymbol{T}(\boldsymbol{p}_i)$ is one that ensures that the state vectors are compatible with a common subspace, $\mathcal{R}$, spanned by the columns of an orthogonal matrix $\mathcal{R} = \text{span}(\boldsymbol{R}) \subset \mathbb{R}^{n\times k}$ such that

$$\text{span}(\boldsymbol{R}^\top \boldsymbol{W}(\boldsymbol{p}_1)) = \cdots = \text{span}(\boldsymbol{R}^\top \boldsymbol{W}(\boldsymbol{p}_{N_K})) \subset \mathbb{R}^k \tag{12}$$

which, in turn, implies that the full order states implied by each ROM $\boldsymbol{R}\boldsymbol{R}^\top \boldsymbol{W}(\boldsymbol{p}_i)\boldsymbol{x}_{ae}(\boldsymbol{p}_i)$ all evolve in the same subspace of $\mathbb{R}^n$. From these definitions, it follows that $\boldsymbol{T}(\boldsymbol{p}_i) = \boldsymbol{R}^\top \boldsymbol{W}(\boldsymbol{p}_i)$, where the transformation $\boldsymbol{T}(\boldsymbol{p}_i)$ describes permutations, distortions and rotations of the bases $\boldsymbol{W}(\boldsymbol{p}_i)$ and is non-singular.

The choice of $\boldsymbol{R} \in \mathbb{R}^{n\times k}$ is made to include the $k$ directions that capture the most relevant dynamics of all the models in the library. This is done using an economy version of the SVD

$$\boldsymbol{W}_{\text{all}} = [\boldsymbol{W}(\boldsymbol{p}_1), \ldots, \boldsymbol{W}(\boldsymbol{p}_{N_K})] = \boldsymbol{U}\Sigma\boldsymbol{V}^\top, \text{ where } \boldsymbol{U}, \boldsymbol{V} \in \mathbb{R}^{n\times kN_K} \text{ and } \boldsymbol{\Sigma} \in \mathbb{R}^{kN_K\times kN_K} \tag{13}$$

and $\boldsymbol{R}$ is chosen to be the first $k$ columns of $\boldsymbol{U}$ which contain the most important directions in $\boldsymbol{W}_{\text{all}} \in \mathbb{R}^{n\times kN_K}$.

This approach is based on the fulfillment of the modal assurance criterion (MAC) [37]. The MAC is a statistical indicator [65] that shows whether two sets of vectors are correlated by assigning a numerical value between 1 and 0 to each vector pair, where the maximum denotes that the vectors are in strong correlation. In this case it applies as the corresponding vectors of the new right reduced order bases $\boldsymbol{W}(\boldsymbol{p}_i)\boldsymbol{T}(\boldsymbol{p}_i)^{-1}$ and $\boldsymbol{R}$ must be in good correlation, which can be assured by the MAC. Since the bases can be normalized, the MAC between the $j$-th vector of $\boldsymbol{W}(\boldsymbol{p}_i)\boldsymbol{T}(\boldsymbol{p}_i)^{-1}$ and the $k$-th vector of $\boldsymbol{R}$ can be computed as

$$\text{MAC}((\boldsymbol{W}(\boldsymbol{p}_i)\boldsymbol{W}(\boldsymbol{p}_i)^{-1})_j, \boldsymbol{R}_k) = |(\boldsymbol{W}(\boldsymbol{p}_i)\boldsymbol{T}(\boldsymbol{p}_i)^{-1})_j^\top \boldsymbol{R}_k|^2. \tag{14}$$

A strong fulfillment of the MAC (enforcing a value of 1 between corresponding vector pairs) can be achieved between

the corresponding vectors of $\boldsymbol{W}(\boldsymbol{p}_i)\boldsymbol{T}(\boldsymbol{p}_i)^{-1}$ and $\boldsymbol{R}$ by explicitly setting them to the maximum value of 1 in order for

$$\boldsymbol{T}(\boldsymbol{p}_i)^{-\top}\boldsymbol{W}(\boldsymbol{p}_i)^{\top}\boldsymbol{R} := \boldsymbol{I} \tag{15}$$

and, therefore, $\boldsymbol{T}(\boldsymbol{p}_i)$ is defined by

$$\boldsymbol{T}(\boldsymbol{p}_i) = \boldsymbol{R}^{\top}\boldsymbol{W}(\boldsymbol{p}_i). \tag{16}$$

An alternative approach to build $\boldsymbol{T}(\boldsymbol{p}_i)$ is to use a weaker fulfillment of the MAC, whereby the sum of the diagonal elements of $\boldsymbol{W}(\boldsymbol{p}_i)\boldsymbol{T}(\boldsymbol{p}_i)^{-1}$ is maximized [17, 36]. This maximization problem is a case of an orthogonal Procrustes problem which can be solved analytically using an SVD [37], but this will not be explored further in this work.

The transformation onto the congruent set of generalized coordinates is then simply applied on each of the reduced aeroelastic systems using right and left multiplication, such that a local reduced order model now takes the form of

$$\Sigma_r(\boldsymbol{p}_i) = \left(\begin{array}{c|c} \boldsymbol{T}(\boldsymbol{p}_i)\boldsymbol{A}_{ae}(\boldsymbol{p}_i)\boldsymbol{T}(\boldsymbol{p}_i)^{-1} & \boldsymbol{T}(\boldsymbol{p}_i)\boldsymbol{B}_{ae}(\boldsymbol{p}_i) \\ \hline \boldsymbol{C}_{ae}(\boldsymbol{p}_i)\boldsymbol{T}(\boldsymbol{p}_i)^{-1} & \boldsymbol{D}_{ae}(\boldsymbol{p}_i) \end{array}\right). \tag{17}$$

Although this methodology ensures that all the state vectors between which the interpolation is performed are congruent and represent the same dynamics, the modal formulation (required for the reduction of the aerodynamic system) does introduce a significant assumption. Each of the source reduced order systems has its inputs and outputs defined by the local modal matrix, therefore if the deformation of the structure varies significantly between adjacent local models an issue may arise where the interpolation may be performed between very different structural modes. However, performing a projection onto a generalized set of coordinates in the same way that has been done for the state-vector would mean that (since the reduction of the aerodynamic model depends on them) it would be necessary to know *a priori* all modal matrices in the library in order to assemble a congruent space, as shown in (13). Therefore, considering the feature to add or modify the database of models *a posteriori* necessary, it is assumed that the modes are compatible between adjacent local models and that any evolution is smooth. This may result in a finer grid sampling of local models but the overhead will be considerably smaller compared to that associated with recomputing the entire library.

Having the local reduced order model expressed in the same set of generalized coordinates, the weighted interpolation at the new operating point can be performed directly as per (9). Contrarily to [63] and [36] where the interpolation is performed in a tangent manifold, direct interpolation is preferred in this case to avoid numerical issues that arise when computing the matrix logarithm required for the tangential method. For reference, the interpolation can be performed on different manifolds depending on the properties of the system matrices (for structural systems [66]) or more generic approaches can be found in [36, 63, 67].

This approach is based on the interpolation of the aeroelastic state-spaces. Additionally, obtaining the reduced

order bases $\boldsymbol{W}(\boldsymbol{p})$ by interpolation is also possible [63], however, it requires computing a new full order model at each operating condition where we would seek the ROM and, therefore, not feasible for this application. This approach is useful when calculating reduced order bases is significantly more expensive than computing the full-order system.

### B. Sampling Strategies

The interpolation framework is not complete without a defined parameter space sampling strategy, and indeed, without adequate sampling the results from the interpolation can be either physically meaningless or computationally intractable for a large parameter space. The examples shown in [68] illustrate the significant sensitivity of the interpolation scheme to the choice of training sets that has inspired the search for algorithms select training points in the parameter space that minimize the interpolation error.

#### *1. Definitions*

We introduce the notation and definition for the different system properties, sets of points used as interpolation training and testing points and error metrics. Bold notation is employed for vector variables and capitalized bold for matrices. Capitalized, non-bold symbols refer to sets. Then, the $p$-dimensional parameter space $P \subset \mathbb{R}^p$ is defined by

$$P = [l_1, u_1] \times [l_2, u_2] \times \cdots \times [l_p, u_p] \tag{18}$$

where $p$ is the number of parameters and the $i$-th parameter is assumed to be bounded between $[l_i, u_i]$, and $\boldsymbol{p} \in P$. Then, we define the set $P_0 \subset P$ as a set of points in the parameter space at which a true system is known and is locally accurate in the vicinity of $\boldsymbol{p} \in P_0$. For each of these true systems, acquired using `SHARPy` following the procedure in Sec. II and II.B, the transfer function is known and denoted as $\boldsymbol{G}[\boldsymbol{p}](s)$, which represents the true transfer function at the parameter point $\boldsymbol{p} \in P_0$. For simplicity of the notation the Laplace variable $s \in \mathbb{C}$ is omitted, resulting in $\boldsymbol{G}[\boldsymbol{p}]$.

In addition, we define the transfer function of an interpolated system at a point $\boldsymbol{p} \in P$ as $\boldsymbol{G}[\boldsymbol{p}; Z]$, which has been computed using the set of training points $Z \subseteq P_0$ in the interpolation framework, where $Z$ is a collection of $N_K$ known systems in the form of (17). Then, by definition of the interpolation method, we can state that

$$\boldsymbol{G}[z; Z] = \boldsymbol{G}[z], z \in Z \tag{19a}$$

$$\boldsymbol{G}[\boldsymbol{p}; Z] \neq \boldsymbol{G}[\boldsymbol{p}], \boldsymbol{p} \in P \setminus Z. \tag{19b}$$

In other words, (19a) is stating that the interpolated system is equal to the true system at a point where the system is known and (19b) states that there may be an error in the approximation of a transfer function at a point not part of the training set of the interpolation framework. This is what we seek to quantify next: the error in the approximation to a

true system using the proposed interpolation framework.

We quantify the error in the approximation by means of a cost function, $\varepsilon = h(\cdot, \cdot)$, which is positive, scalar-valued and possibly nonlinear, and may take the transfer functions of true and interpolated systems as

$$\varepsilon := h(\boldsymbol{G}[\boldsymbol{p}], \boldsymbol{G}[\boldsymbol{p}; Z]) \in \mathbb{R}. \tag{20}$$

The error function (20) can also be expressed as $\varepsilon[\boldsymbol{p}; Z]$, denoting the interpolation error at a point $\boldsymbol{p} \in P$ given a training set $Z$. Then, the choice of cost function is influenced by the scope of the problem. Several error metrics can be used, and that will be part of the specifics of each problem. In Sec. IV, we will discuss two possible metrics using the stability or root loads of the wing.

*2. Adaptive multi-purpose data Bayesian sampling scheme*

Of the elements that conform the interpolation framework, all that is left is to choose the training set $Z \subset P_0$ that will be the base of the interpolation framework. The simplest choice for $Z$ is a regular grid, for which points are chosen at regular intervals across the parameter space. However, these are cursed by dimensionality, and soon become intractable if a refined sampling is chosen in a large dimensional parameter space. This is a classic design of experiments problem and several options exist which offer adequate visibility into the parameter space without incurring in the cost of structured grids.

A notable approach to sample the parameter space is a Latin Hypercube Sampling (LHS), which presents a distributed option for the choice of $Z$ in particular when the dimensions of the parameter space are large. This sampling results in an unstructured grid which can be then tessellated using a Delaunay triangulation as previously described and used by the interpolation scheme.

Depending on the nature of the aeroelastic problem, these methods may be overly simplistic and may not offer a training set $Z$ that performs well across the entire parameter space or it may at the cost of over-sampling regions that could be equally well captured with fewer points. Therefore, we seek an adaptive method to sample the parameter space where regions where dynamics change more notably are more finely sampled than those where the system barely changes. This would depend on knowing how well the interpolation performs across the space, in order to sample where the discrepancies are largest. However, the underlying cost function $\varepsilon[\boldsymbol{p}; Z]$ $\boldsymbol{p} \in P$ cannot be computed for all $\boldsymbol{p}$ since it would require $P_0 = P$ and defeat the purpose of this exercise. Therefore, we initially turned to Bayesian optimization (BO) [69] to find, without computing the actual underlying cost function, the location of the maximum interpolation error. This would be performed sequentially, such that after each iteration $i$ a new point, $\boldsymbol{p}^+$, would be added to the training set $Z$ until a computational budget was expended.

As a brief summary, Bayesian optimization is a sequential optimization algorithm that does not assume any functional

forms of the underlying true functions and does not require computing gradients, therefore it is normally used to maximize expensive-to-evaluate functions. BO samples the domain sequentially to build a Gaussian process (GP) statistical surrogate that results in a posterior mean that is updated as new samples are taken. The location of these new samples are chosen attending to an acquisition function, which normally offers a tradeoff in exploration of the domain versus minimizing the number of calls to the expensive-to-evaluate function; some of commonly used acquisition functions are probability of improvement, expected improvement or upper confidence bounds, among others [69].

The current BO scheme is initialized with a training set, $Z_0$, that includes the edges of the parameter space such that the entire domain $P$ falls within the convex hull of $Z_0$ in order to not require extrapolation, as stated in the interpolation scheme description. The BO posterior probability distribution given by

$$\mu_\varepsilon[\boldsymbol{p}; Z_0] = \mathbb{E}[\varepsilon[\boldsymbol{p}; Z_0] \,|\, \varepsilon[P_s; Z_0]], \;\; \boldsymbol{p} \in P \tag{21}$$

estimates the interpolation error anywhere in the domain. In other words, $\mu_\varepsilon[\boldsymbol{p}; Z_0]$ is the expected error of an interpolation at $\boldsymbol{p} \in P$ using $Z_0$ as training data set and given that the true error $\varepsilon$ is known at $P_s$, where $P_s$ corresponds to a set of testing points at which the scheme has computed the true system in order to inform its prediction. At these points, $P_s$, it can compute the error $\varepsilon[\boldsymbol{p}_s; Z_0], \;\; \boldsymbol{p}_s \in P_s$ and use this result to inform the Gaussian process (21). This is an iterative process, and points $\boldsymbol{p}_s$ are sampled according to an expected improvement function. The output of the optimization is the maximum cost and the location at which it occurs, $\boldsymbol{p}^+$, and this becomes the next best point to add to the training set for the next iteration $Z_1 = \{\boldsymbol{p}^+\} \cup Z_0$. The number of evaluations during the optimization, which determine the size of $P_s$, depends on the problem. The larger the testing set, the more accurate the prediction of the cost function is likely to be, despite an increased computational cost.

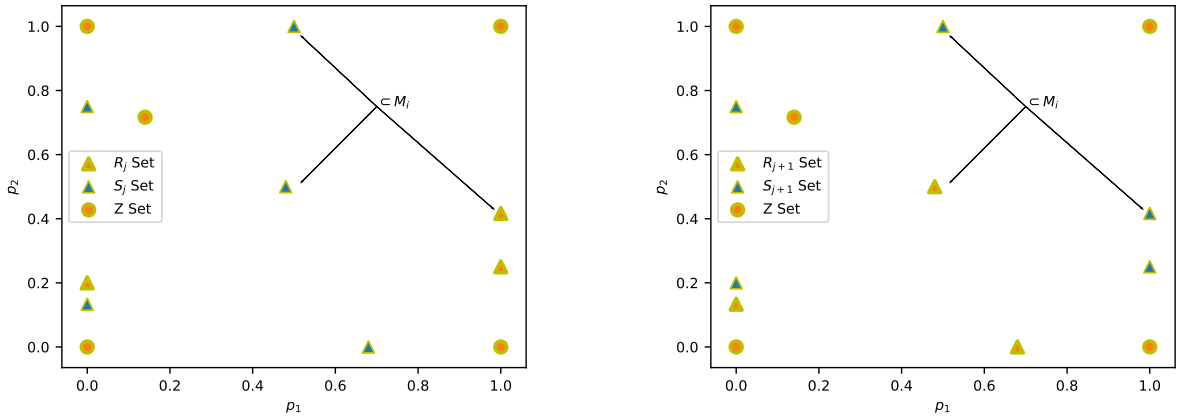
The limiting factor of this scheme is the large number of evaluations of the true system (which are the ones that this entire work aims to minimize) that are used as testing data only in order to generate $\mu_\varepsilon$, not making it onto the training set and thus not being used to improve the interpolation accuracy. As with any regression that depends on testing and training data, it is clear that the sweet-spot lies in a careful balance between the two and in the applications tested with the BO scheme [68], a large number of testing points are required in $P_s$ to find $Z_0$ that are then not later used, thus the scheme is effective albeit inefficient. Consequently, the large number of unused data points has inspired and encouraged the development of our proposed "data-green" sampling strategy, which aims to reuse and recycle a set of the computed points as training and testing sets in order to obtain a more informed Gaussian regression with the same number of full system evaluations.

For this novel adaptive sampling approach, referred to as adaptive multi-purpose Bayesian sampling, we use a pool of known true systems (recall these are the ones expensive to evaluate) from which we extract multiple different combinations of training and testing sets to, without computing any more, find with reasonable accuracy the location at

which the maximum error occurs by combining the predicted mean error from each combination.

Before delving into how the expected error is used to find the worst case condition in the parameter space, it is convenient to define two sets: a training-only set $Z_i$, which contains points that will only be used as interpolation training points; and a multipurpose set, $M_i$, which contains points that may be used as testing or training points in different combinations. The $(\bullet)_i$ subscript marks the current iteration. The following applies to these two sets: $Z_i \cup M_i = X_i$ and $Z_i \cap M_i = \emptyset$, i.e. all points for which the true system is known at every iteration, $P_{0,i}$, belong to either $M_i$ or $Z_i$.

The training-only set, $Z_i$, is simple. It is fixed at each iteration and is initialized with the parameter space edge points such that all evaluation points fall within the convex hull of the training data, as the current interpolation schemes do not support extrapolation. The multipurpose set, $M_i$, is slightly more complex. From $M_i$ we desire to obtain two sets, the *tRaining* set $R_j$ and the *teSting* set $S_j$, such that $S_j \cup R_j = M_i$. Numerous splitting options exist and in total $J$ different sets can be obtained by selecting different partitions $M_i$. To avoid a large number of possible combinations $J$ of sets in $M_i$, the sets are constrained in size such that $||S_j| - |R_j \cup Z_i|| \leq 3$. In essence, the testing and training data sets may differ in size up to a maximum mismatch of 3 members. This not only ensures that the amount of training and testing data is balanced (it will be explained later why this is required) but also limits the number of combinations $J$, which could otherwise become too large to handle if all combinations are allowed. To further clarify the concept behind the different sets, a few subsets of the $j$ possible combinations are shown in Fig. 4, which shows an example parameter space with training and testing points. For instance, this example has $|Z| = 5$, $|R| = 3$ and $|S| = 5$, which translates to 5 testing points and 8 training points which is a reasonably balanced split. Note how the points belonging to $Z$ are fixed in both cases, whereas those in $M_i$ can change their function, like the one in the vicinity of $(0.5, 0.5)$ which goes from being a testing point in $S_j$ to a training point in $R_{j+1}$.




**(a) Example combination j**

**(b) Example combination j+1**

**Fig. 4 Examples of different combinations of training and testing datasets using the training-only and multipurpose sets on a two-dimensional parameter space. The sets correspond to the training-only $Z$ set, and two arbitrary partitions ($j$ and $j+1$) of the multipurpose set into $S_j$ and $R_j$.**

Therefore, at each iteration there are available $J$ different $S_j$ and $R_j$ sets and a single, fixed $Z_i$ set. Then, for every combination $j \in \{1, \ldots, J\}$, it is possible to find the expected error function, $\mu_{\varepsilon,j}[\boldsymbol{p}; R_j \cup Z_i]$, for the interpolation framework that uses training points $R_j \cup Z_i$ at every point in the parameter space, $\boldsymbol{p} \in P$, given the known value of the error function, $\varepsilon[S_j; R_j \cup Z_i]$, at the testing data points in $S_j$ as

$$\mu_{\varepsilon,j}[\boldsymbol{p}; R_j \cup Z_i] = \mathbb{E}[\varepsilon[\boldsymbol{p}; R_j \cup Z_i] \mid \varepsilon[S_j; R_j \cup Z_i]], j \in \{1, \ldots, J\}, \text{ for any } \boldsymbol{p} \in P. \tag{22}$$

This posterior mean of the expected error is obtained using a Gaussian process regression, as done for the BO, and thus in the same form as (22), the expected variance $\sigma^2_{\varepsilon,j}(\boldsymbol{p}; R_j \cup Z_i)$ for any point $\boldsymbol{p} \in P$ is also known. Then, given the posterior mean and the variance of the Gaussian surrogates from each one of the $j$-th sets, all that remains is to define functions that aggregate them to locate two new points at the end of the iteration. These scalar-valued functions, $a : \mathbb{R}^{J+1} \to \mathbb{R}$ and $b : \mathbb{R}^{J+1} \to \mathbb{R}$, of the form

$$\mu_i(\boldsymbol{p}) = a(\mu_{\varepsilon,0}(\boldsymbol{p}), \mu_{\varepsilon,j}(\boldsymbol{p}), \ldots, \mu_{\varepsilon,J}(\boldsymbol{p})) \tag{23a}$$

$$\sigma_i(\boldsymbol{p}) = b(\sigma^2_{\varepsilon,0}(\boldsymbol{p}), \sigma^2_{\varepsilon,j}(\boldsymbol{p}), \ldots, \sigma^2_{\varepsilon,J}(\boldsymbol{p})) \tag{23b}$$

should be chosen such that the maximum of the mean error based function (23a) gives the location of the worst interpolation performance and thus identifies next point, $\boldsymbol{p}^+_{\mu,i}$, to add to the training-only dataset for the subsequent iteration, $Z_{i+1} = \{\boldsymbol{p}^+_{\mu,i}\} \cup Z_i$. The variance based function (23b), in turn, relates to the uncertainty of the model and output the point, $\boldsymbol{p}^+_{\sigma,i}$, to be added to the multipurpose set for the subsequent iteration, $M_{i+1} = \{\boldsymbol{p}^+_{\sigma,i}\} \cup M_i$ and not leave regions unexplored. Consequently, for every iteration two new points are computed, one that is added to the training-only set at the location of the worst error and the other to the multipurpose set at the location of the maximum uncertainty. The former ensures areas that have high error sensitivity are finely sampled and the latter that undersampled areas do not remain unexplored (equivalent to the expected improvement function in the standard BO). To illustrate the process described, a flowchart is presented in Fig. 5, where `SHARPy` refers to the process in Fig. 3. These functions may be adapted to the peculiarities of the problem and thus will be further explained along the numerical example. A note on these functions is that they should avoid producing an output point that is already in $Z_i$ or $M_i$, as that would reduce the number of available points to make combinations in the subsequent iterations and the algorithm's viability become compromised.

Some decisions need further discussion. First, the advantage of having training-only set, $Z_i$, as opposed to basing the training data solely on the combinations of $R_j$. The rationale behind this is that if we were to rely solely on $M_i$ without a fixed training set, we could encounter the situation where the error at one point in one of the sets $j$ is orders of magnitude larger than in other sets. If that point were added then to $M_{i+1}$ in the next iteration, the interpolations that do

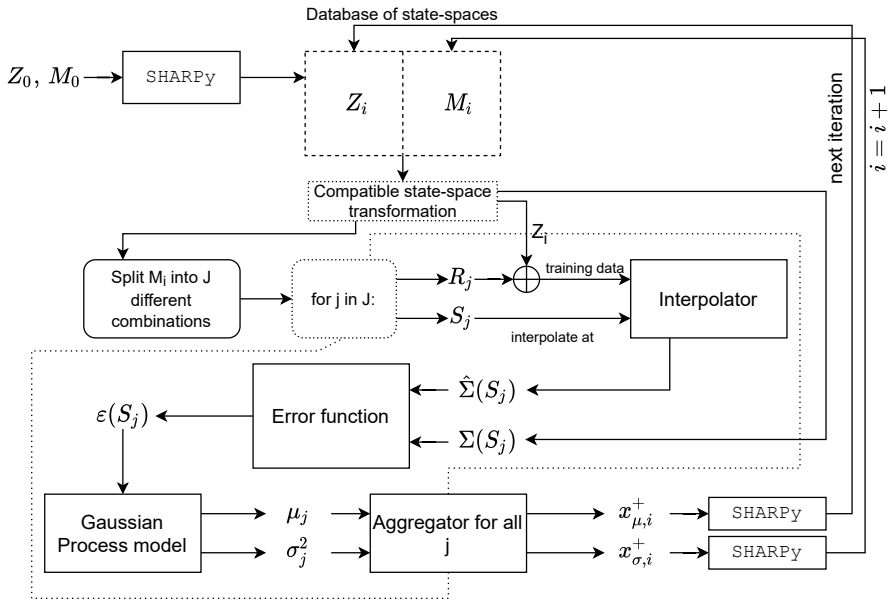



**Fig. 5 Multipurpose Bayesian sampling flow**

not include that point in their respective training sets $R_i$ would see again that same error orders of magnitude larger than elsewhere in the parameter space. Thus, the resulting "worst case" condition for that iteration could be the same as in the previous one, rendering the iteration futile. Therefore, by having a training-only set, we ensure that points in the parameter space that cause large deviations in the interpolated versus true data are always considered, allowing subsequent iterations to continue sampling the parameter space at other locations for additional "worst case" points.

Another clarification would come from the need of having another point computed from the outcome of (23b) added to $M_{i+1}$. This point is necessary to make the optimal sampling scheme independent of the initial conditions, in this case the points $P_0$, $Z_0$ and $M_0 = P_0 \setminus Z_0$. Effectively, by adding a new point to be used as training or testing in areas of the parameter space where there is large variance (uncertainty), permits the optimization to increase its knowledge on areas of potentially high error which otherwise could be ignored.

## IV. Numerical Example

A numerical demonstration of the methods described above is included in this section. The full-order system evaluations and the Krylov subspace projections have been implemented in the `SHARPy` package [50]. The nearest-neighbour interpolation scheme uses the algorithms in the `scipy` Python library, and the BO algorithms are those of `GPyOpt` [70]. The test case is the Pazy wing, a very flexible wing, designed and developed at the Technion as part of the 3rd Aeroelastic Prediction workshop, to be used as benchmark for numerical and experimental demonstrations [49]. The objective for the interpolation strategy is to predict both the flutter boundaries and root loads for this wing, which are strongly dependent on its (geometrically-nonlinear) aeroelastic equilibrium and, therefore, expensive to compute.

The Pazy wing is 550 mm in span and 100 mm in chord. It consists of an aluminum spar within a Nylon 3D printed chassis with a NACA0018 airfoil profile ribs to which an Oralight film skin gives its aerodynamic shape. The computer model also includes a simulated flap across 50% of the chord is added in the outboard 20% of span to the UVLM system to model system inputs. Then, a one-dimensional beam model of the wing has been derived by Riso et al. in [71], where a 3D finite element model has been reduced to sectional coefficients, and the vortex lattice is projected over this reference line. A representation of the beam and vortex lattice (bound and wake) of the deformed Pazy wing is shown in Fig. 1b. The Pazy wing is capable of undergoing large deformations (approximately 50% of span) and displays a complex stability envelope driven by its deformation [57]. It exhibits an initial instability characterized by a 1st flutter mode that becomes stable again as the deformation increases (via angle of attack or free stream velocity increase), leading to a region of stability before a 2nd flutter mode onset at higher speeds.

The drawback is that producing such envelope is a computationally expensive process. First, obtaining a linearized system around each nonlinear aeroelastic equilibrium of a highly discretized model (needed for flutter convergence) and subsequent model reduction to evaluate the eigenvalues is in itself costly, as previously described in Sec. II-Sec. II.B, and which will be explored in more detail in this section. Moreover, this system is influenced by many nonlinearities, with structural mode switches and large deformations which affect both the structural and aerodynamic subsystems, and requires of linearization and reduction at every angle of attack and free stream velocity pair. Thus, on a desktop system, the analysis in [4] in which the parameter space was sampled in 1 m/s increments and $\alpha \in \{0, 0.25, 0.5, 1, 2, 3, 4, 5\}^\circ$ (totaling 560 system evaluations) took approximately 60 hours to run. Moreover, adding more parameters to the exploration would yield the problem unfeasible.

Therefore, we turn to interpolation methods to reduce that time considerably. The objective is to produce a minimal library of training data that can be used as part of an interpolation scheme to sample the design parameter space efficiently and accurately. One of the many benefits of interpolating the underlying dynamical systems using this framework, as opposed to interpolating the flutter speed directly, for example, is that the resulting state-space realizations can be used to compute loads or even for controller designs (such as flutter suppression systems). Note that the methods described herein are analogous for aircraft with rigid-body motion, as shown in [60].

First, we will discuss the performance of the Krylov subspace model order reduction on the aerodynamic model of the Pazy wing about a deformed configuration. Then, the focus will shift to the 2-dimensional parameter space of the Pazy wing (angle of attack, $\alpha$ and free stream velocity, $U_\infty$) with the final goal of achieving similar results to those presented in [4] at a fraction of the cost.

### A. Krylov Subspace Reduction Analysis

The performance of the Krylov subspace reduction on the Pazy wing UVLM linearised about a deformed equilibrium condition is first explored, at $U_\infty = 50\,\mathrm{m/s}$ and the wing clamped at an angle of attack $\alpha = 5^\circ$. As detailed in Sec. II.B,

the model reduction process is heavily obstructed if the number of transfer functions is large and therefore the use of the projection of the aerodynamic FOM onto the structural modal coordinates to reduce the input/output dimensionality. The first 8 structural modes (at each nonlinear equilibrium) are retained (i.e. frequencies below 370 Hz). At the current temporal discretisation where $\Delta t = 1.25 \times 10^{-4}$ s the Nyquist frequency $f_N = 4000$ Hz and therefore there are no risks of aliasing and dynamics not being captured. The time step selection is such that ensures a $CFL = 1$ condition (whereby the time step is chosen to represent the time taken for the circulation to convect downstream by one panel length) and thus $\Delta t = c/M/U_\infty$, where $c$ is the chord, $M$ the number of chordwise bound panels and $U_\infty$ the free stream velocity; in the current Pazy example, $M = 16$ and $U_\infty = 50\,\mathrm{m/s}$, which ensures a small enough time step, yet it could be the case that for larger chord and lower speed conditions that the system misses important dynamics as a result of aliasing. Therefore, the Nyquist frequency and structural frequencies should always be contrasted. The full order model (FOM) of the Pazy linear UVLM that is subject of reduction is shown in Tab. 1; the large size of the wake states is a result of the required 20 chord wake length for numerical convergence, where each wake panel is the same in size as a bound vortex panel. However, a convergence study has been explored in [72] using different sized wake vortex panels.

**Table 1 Pazy discrete-time linear UVLM full order model (FOM) output, states and inputs channel description and size.**

| Outputs | | States | | Inputs | |
|---|---|---|---|---|---|
| Modal forces $\boldsymbol{Q}^n$ | 8 | Bound circulations $\boldsymbol{\Gamma}^n$ | 960 | Modal displacements $\boldsymbol{q}^n$ | 8 |
| | | Wake circulations $\boldsymbol{\Gamma}_w^n$ | 19200 | Modal velocities $\dot{\boldsymbol{q}}^n$ | 8 |
| | | Bound circulation derivatives $(\Delta t\dot{\boldsymbol{\Gamma}})^n$ | 960 | Flap deflection $\delta^n$ | 1 |
| | | Previous circulation $\boldsymbol{\Gamma}^{n-1}$ | 960 | Flap deflection rate $\dot{\delta}^n$ | 1 |
| Total | 8 | | 20160 | | 18 |

The performance of the Krylov-built ROM is assessed by comparison with the full order model, which in the following plots will be shown as a thick grey line ( ). The ROMs are constructed using interpolation about the steady-state gain (zero frequency, $\sigma = 0$) given the interest in that band of the spectrum and where potential flow aerodynamic assumptions hold best. The independent variable in this analysis is the Krylov subspace order, $r$, and models will be built using the controllability (3a), observability (3b), or dual-sided which is the union of both subspaces (removing any linear dependencies that may arise from the combination). This comparison is exploratory, since by the nature of the linear UVLM and its input and output space dimensions the observability subspace is *a priori* more efficient in performing the reduction given the smaller size of the output matrix $\boldsymbol{C}$.

A subset of the magnitude of the frequency response between the first three modal displacement inputs and the first three modal force outputs of the observability Krylov ROM is shown in Fig. 6, where these first three displacement modes correspond to the first out-of-plane bending (OOP1), second out-of-plane bending (OOP2) and the first torsional mode (T1). The underlying transfer functions are smooth and the magnitude of the response tends to infinity as

the frequency increases. This is due to the nature of the linear aerodynamics that do not include saturation, thus as the frequency of the displacement excitation increases, so does the output force. However, at those frequencies the assumptions of potential-flow theory no longer hold so that band of the spectrum is not considered anyway. It can be seen that as the observability Krylov subspace order $r_c \geq 3$, the ROM captures the shape of the transfer function and increasing further the Krylov order shows a less pronounced improvement.

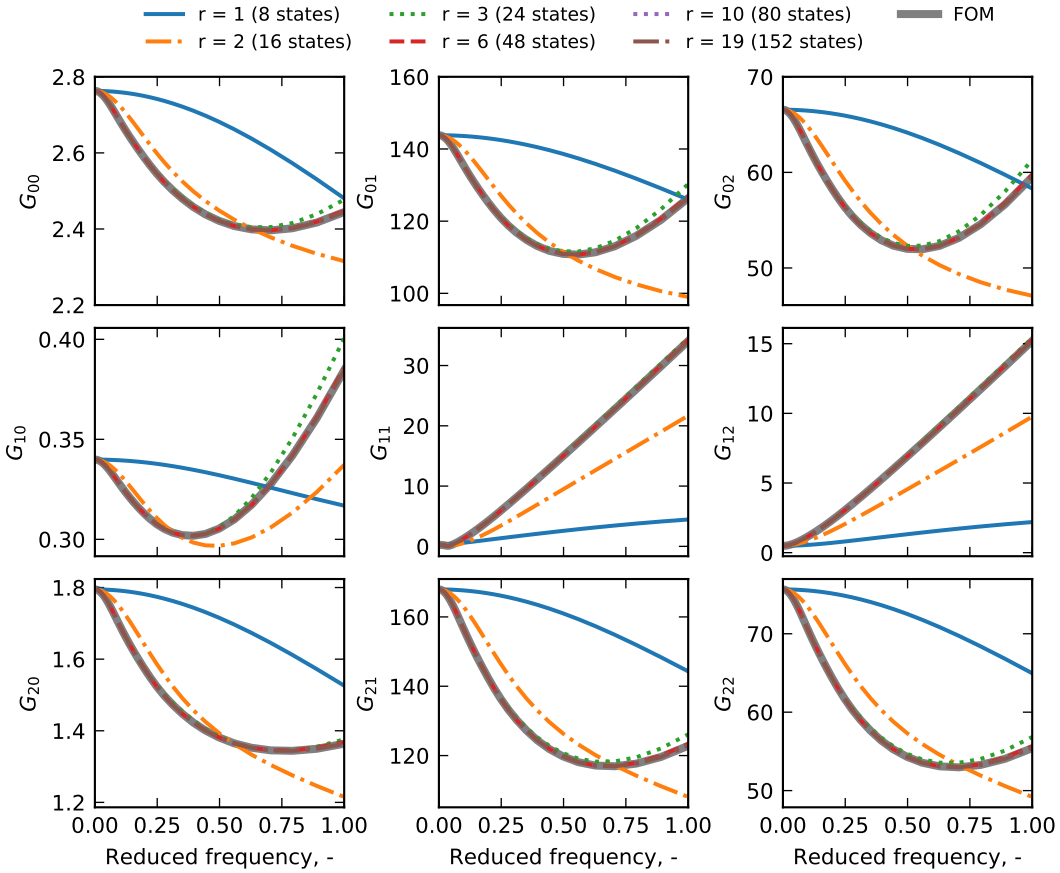



**Fig. 6 Frequency response magnitude ||G|| comparison of different sized ROMs constructed using the observability subspace and the FOM.**

In order to assess the performance in more detail, Fig. 7 plots the magnitude of the error system $\Sigma_\epsilon = \boldsymbol{G}_\mathrm{r} - \boldsymbol{G}_\mathrm{f}$ for the same subset of transfer functions at different reduced frequencies. In these plots, errors are shown at discrete reduced frequency intervals. The previous statement about the shape being captured beyond $r_c = 3$ is again trivial to see, with an orders of magnitude drop in the error at this point across all frequencies. Then, the properties of the Krylov subspace reduction with a single interpolation point ($\sigma = 0$) become readily visible, as the error increases the further away from zero frequency. As the ROM grows and more moments are matched, the accuracy of the reduced model increases further away from this point.

The reduction can also be performed assembling the projectors that span only the controllability subspace (3a). In

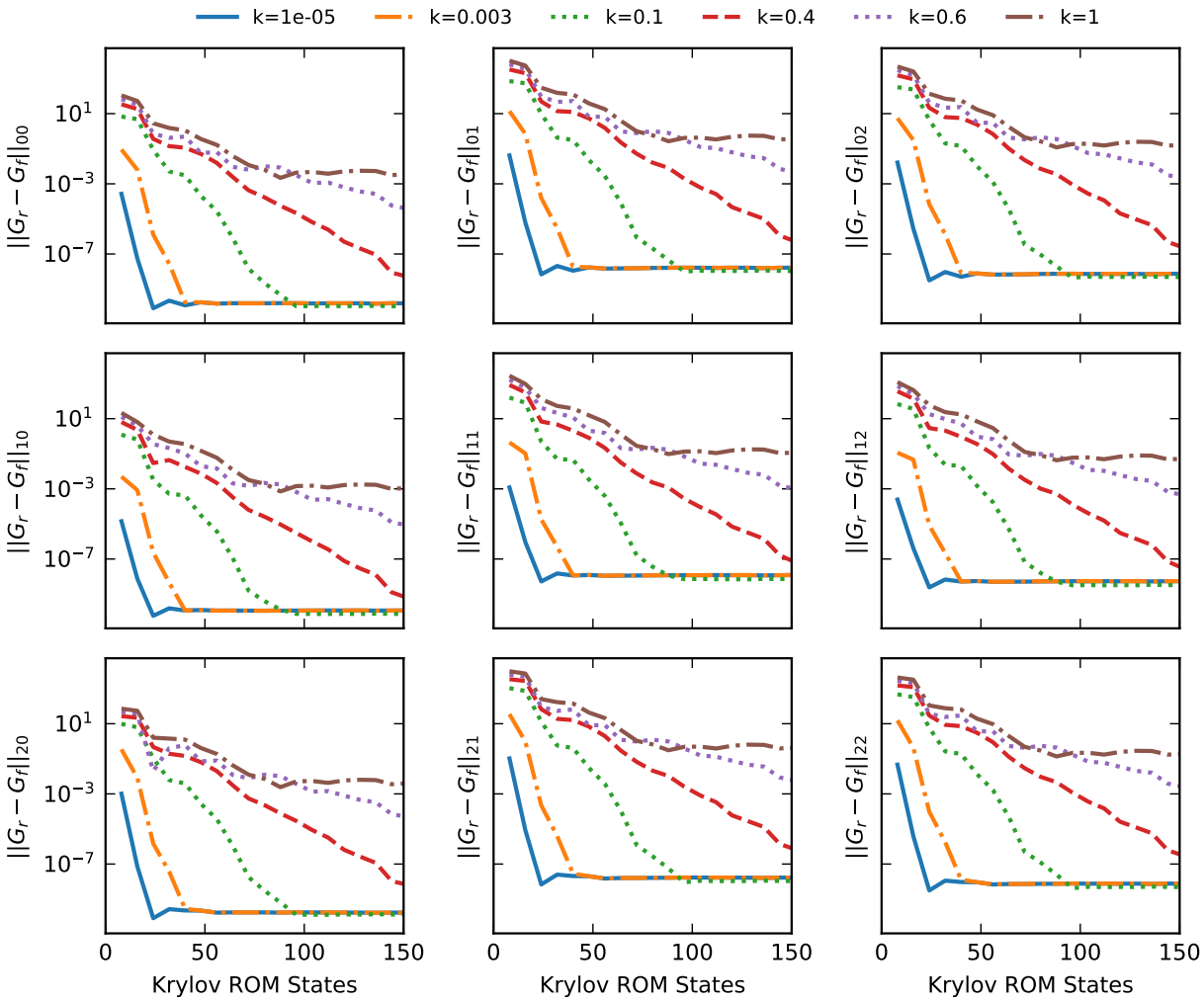


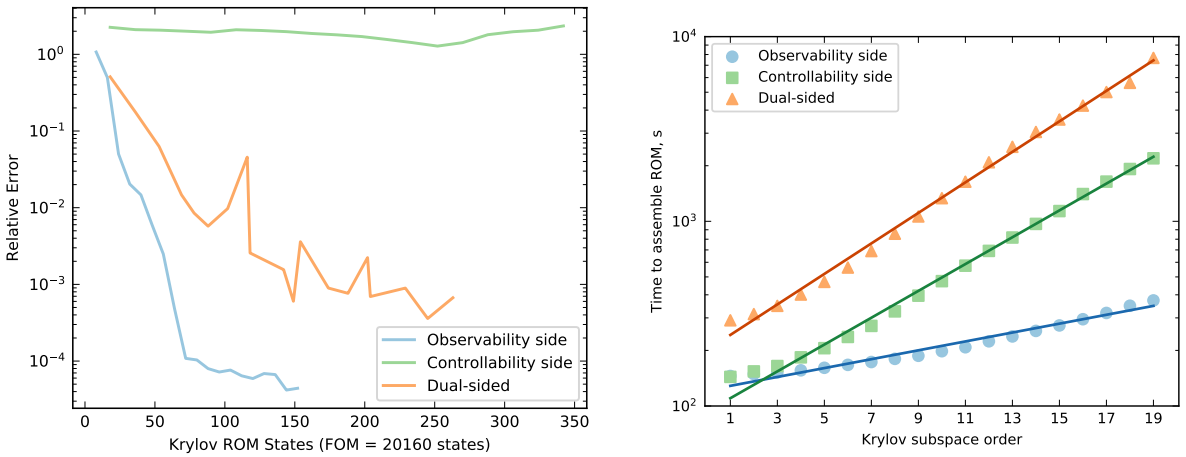

**Fig. 7 Magnitude of the error system at various reduced frequencies**

this particular case for the UVLM system, the performance of the ROM was heavily downgraded and the ROMs did not converge as the Krylov order was augmented. Given the excellent performance of the observability Krylov subspace ROMs, the reasons behind this behavior using the controllability matrices have been left outside the scope of this work. Finally, using the dual-sided Krylov subspace (built by Petrov-Galerkin projection using both (3a) and (3b)) resolves the issues with the controllability Krylov ROMs. The resulting transfer functions are similar to those as in Fig. 6 and are omitted for brevity. To visualize the ROM effectiveness, we can evaluate the performance of all these methods by means of a relative error across all transfer functions and taking the worst case condition as

$$\varepsilon(\boldsymbol{G}_r, \boldsymbol{G}_f) = \max_{i,j} \left( \frac{\sup_{0\leq k\leq 1} ||\boldsymbol{G}_{\epsilon,i,j}(k)||}{\sup_{0\leq k\leq 1} ||\boldsymbol{G}_{f,i,j}(k)||} \right) \text{ where } \boldsymbol{G}_\epsilon(k) = \boldsymbol{G}_r(k) - \boldsymbol{G}_f(k), \text{ for } i \in [0, \ldots, m],\ j \in [0, \ldots, p], \tag{24}$$

where $k$ denotes reduced frequency. With this metric, the performance across of all ROMs can be easily contrasted as

shown in Fig. 8a. The observability Krylov ROM offers the best performance with a relative error approaching $1 \times 10^{-4}$ with 60 states (the FOM being 20160 states) in the frequency range $0 \leq k \leq 1$ using a single interpolation point at $\sigma = 0$. The controllability Krylov ROM offers no convergence with increased Krylov order although this is fixed in the dual-sided ROM. The dual-sided ROM offers convergence at a slower rate than the observability ROM and it shows some regions of increased error despite using a larger Krylov bases. This occurs at points where vectors have been "deflated" as a result of linear dependencies between observability and controllability subspaces.




**(a) Relative error comparison between ROM assembly methods**

**(b) Time to assemble Krylov ROMs based on order and method with an exponential fit**

**Fig. 8 Krylov ROM assembly characteristics.**

Despite Krylov reduction methods not always preserving stability of the system, in our experience [4, 33] both the controllability and observability Krylov realizations of the UVLM system have been stable; however, the dual-sided ROM is more susceptible to including non-physical unstable modes that are not present in the full-order model.

In terms of computational cost, Fig. 8b shows the time taken to assemble each of the ROMs. All times have been taken on the same Intel i7 4.2GHz 64GB RAM workstation and with minimal external processes running, given that most functions used in the presented algorithms are paralellized. An exponential fit is adequate for all three assembly methods, yet the exponential growth rate of the Krylov observability subspace is significantly slower.

A final note must be made on the choice of a single interpolation point about $\sigma = 0$, since the Krylov reduction method allows for reduced order bases that combine multiple interpolation points across the complex plane. Although, matching at various frequencies improves the performance across the spectrum with lower Krylov orders $r$ (i.e. matching less derivatives) the resulting reduced order bases in general do not produce stable realizations. In addition, there is a significant performance cost with multiple interpolation points, as the heaviest computation is the LU factorization of $(\sigma \boldsymbol{I} - \boldsymbol{A})$, which needs to be performed for each $\sigma$. However, once these factors are computed, finding higher order Krylov vectors by increasing $r$ is relatively inexpensive since it only requires further matrix-vector multiplications,

however, for low Krylov subspace dimensions as seen in Fig. 8b. Plus, as seen from the results, good accuracy can still be achieved in the desired spectrum using simply higher Krylov orders. In particular since the intent is to use this framework as part of an adaptive sampling scheme where ROMs are computed online and performance is an important metric to consider. An additional advantage of using the $\sigma = 0$ interpolation point only is that the terms in the reduced order bases, which are "tall" matrices of dimensions $n \times k$, as well as the LU factors in particular employed in the Krylov algorithm are real. If one were to use complex interpolation points, these matrices would involve complex terms that do not impact the numerics of the scheme, but do significantly increase memory requirements, since effectively the memory needed doubles. This is a problem since, already on a 64GB RAM desktop computer, the current implementation of the scheme with real only numbers and using a single interpolation point, memory limits are exceeded during the LU factorization with FOMs larger than $\approx 70 \times 10^4$ states. For all these reasons, i) tendency for stable realizations, ii) single LU decomposition, iii) real terms for memory saving and iv) lowest computational cost, the use of single interpolation, observability-based Krylov reduction is the method of choice for the discrete-time linear UVLM system.

### B. State-space interpolation based on an optimal set of training points

The successful use of the Krylov-subspace method on the observability side of the aerodynamic system permits the creation of a library of parametric reduced order models among which to interpolate. We will use the reduced order models of the Pazy wing to explore the advantages and limitations of the proposed multi-purpose Bayesian sampling of the parameter space. To demonstrate the inner workings of the algorithm, we initially present results on a small subset of the parameter space defined as

$$P = [\alpha = 4.5^\circ,\ \alpha = 5^\circ] \times [U_\infty = 35\,\mathrm{m/s},\ U_\infty = 45\,\mathrm{m/s}] \in \mathbb{R}^2. \tag{25}$$

This domain is discretized in $0.05^\circ$ and 1 m/s increments in order to evaluate the performance of the algorithm, and is also enforced on any sampled points for training or testing purposes. Our reason for studying this small subset of the parameter space is that the changes between the real systems are very smooth, such that the inner workings of the algorithm are unobstructed by otherwise possibly large change in dynamics. We will show how, at each of the iterations within the sampling scheme, the knowledge the model has on the underlying error function is updated and new points are sequentially added to the training dataset. For this initial numerical example we use a relatively coarse aerodynamic mesh which estimates flutter within 2 m/s of the flutter results presented in [4]. However, the flutter prediction results on which this framework is applied in Sec. IV.C.2 do use the full discretisation required to achieve converged flutter results. The initial mesh results in a 2,100-state UVLM, which is reduced using the presented Krylov subspace method matching 6 moments at the steady-state gain, $\sigma = 0$, resulting in a 48 state reduced-order aerodynamic model that has a relative error below $10^{-4}$ computed by (24) in frequencies below $k = 0.5$, sufficient to capture the highest-frequency unstable

mode at $k \approx 0.31$ [4]. The structural system is reduced by modal projection using 8 modes, given by convergence on the root loads computing process, from 378 states to 16 states and these are the chosen modes to project the UVLM prior to the Krylov reduction. The reduced-order aeroelastic system contains thus 64 states in total.

This example uses the relative error in the steady-state root loads as the interpolation error function. To compute this, a manipulation is first applied to the system transfer function to modify the original modal force output $Q$, to the 3 wing root shear and 3 wing root moments. Then, the value of the loads against a control surface deflection input is used to find the maximum relative error as

$$\varepsilon^{\text{loads}}[\boldsymbol{p}; Z] = \max\left(\frac{G_{\text{shear}}[\boldsymbol{p}](0) - G_{\text{shear}}[\boldsymbol{p}; Z](0)}{G_{\text{shear}}[\boldsymbol{p}](0)}, \frac{G_{\text{moments}}[\boldsymbol{p}](0) - G_{\text{moments}}[\boldsymbol{p}; Z](0)}{G_{\text{moments}}[\boldsymbol{p}](0)}\right). \tag{26}$$

The following acquisition functions for the selection of the next training-only point and for the next point added to the multipurpose set, respectively, are chosen as

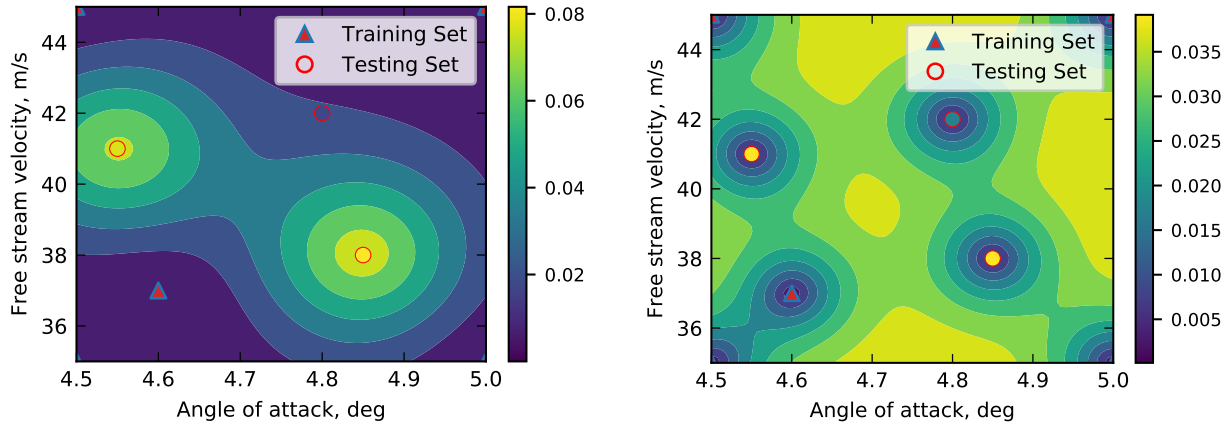
$$\mu_i(\boldsymbol{p}) = \max\left(\sum_{j\in\{0,\ldots,J\}} \mu_j(\boldsymbol{p})\right)\left[\gamma\frac{\sum_{j\in\{0,\ldots,J\}}\mu_j(\boldsymbol{p})}{\max(\sum_{j\in\{0,\ldots,J\}}\mu_j(\boldsymbol{p}))} + (1-\gamma)\frac{\sum_{j\in\{0,\ldots,J\}}\sigma_j(\boldsymbol{p})}{\max(\sum_{j\in\{0,\ldots,J\}}\sigma_j(\boldsymbol{p}))}\right], \tag{27a}$$

$$\sigma_i(\boldsymbol{p}) = \max\left(\sum_{j\in\{0,\ldots,J\}} \sigma_j(\boldsymbol{p})\right)\left[(1-\gamma)\frac{\sum\mu_j(\boldsymbol{p})}{\max(\sum_{j\in\{0,\ldots,J\}}\mu_j(\boldsymbol{p}))} + \gamma\frac{\sum_{j\in\{0,\ldots,J\}}\sigma_j(\boldsymbol{p})}{\max(\sum_{j\in\{0,\ldots,J\}}\sigma_j(\boldsymbol{p}))}\right], \tag{27b}$$

where $0 < \gamma < 1$ is a factor to change the weights of the acquisition functions from mean to variance and $J$ the number of combinations of points in the multipurpose set, which varies in each iteration. In the example to follow, $\gamma = 0.8$. Note these equations correspond to the functions $a : \mathbb{R}^{J+1} \to \mathbb{R}$ and $b : \mathbb{R}^{J+1} \to \mathbb{R}$ in (23a) and (23b), respectively.

The adaptive sampling scheme commences from the initial conditions where the training only set $Z_0$ is the parameter space edges and the initial multipurpose set $M_0$ consists of four randomly chosen points by means of Latin hypercube sampling. Using different combinations of the points in the multipurpose set as training or testing data, multiple Gaussian model surrogates of the error function can be constructed which aggregated by (27) will be used to predict the mean error and variance across the parameter space. An example of an individual set is shown in Fig. 9 illustrating the mean and variance of the Gaussian regression model. In this particular example, of the points in the multipurpose set $M_0$, the point at (4.6, 37) is used as training whereas the others are used as testing in order to generate the statistical surrogate. Using the different combinations in the multipurpose set we arrive at the combined sum of means and variances shown in the first row of Fig. 10, that shows the end result of the first iteration. The first column shows the underlying initial true error the sampling points used to generate the interpolation (the corner points). Of course, this error is unknown to the optimizer and the contours have been constructed computing all points in the parameter space, which is a very costly operation but necessary for the evaluation of the proposed scheme. Then, the center column shows the sum of the mean error across the individual Gaussian models for each of the combinations of subsets and the

third column, in a similar way, shows the sum of the variance in the underlying Gaussian models.




**(a) Posterior mean** **(b) Variance**

**Fig. 9 Gaussian regression model of the multipurpose sets of iteration $i = 0$. Current case shows the multipurpose point (4.6, 37) as part of the training data set.**

Ideally, the shape of the function in estimated error would be the same as that in actual error, meaning that with the few points known a reasonable estimation can be made. Note that the absolute value of each of them is not intended to be similar and the interest only lies on the maximum, since the estimate is a sum across all sets in an effort to avoid biases due to the possibility of individual Gaussian model predictions becoming flat surfaces with zero mean error, as will be shown next. In this iteration, the prediction of the error given the current knowledge is a decent approximation to the actual error, considering how few samples are available. Of the two prominent contours of estimate error, the point (4.6, 40) is highlighted at the maximum of the acquisition function (27a) and this will be the point added to the training-only set in the next iteration $Z_1$. The third column shows the sum of the variance and therefore, where uncertainty in the surrogate model is greatest, which is easily visible by means of the contours in the vicinity of the known points, be them training-only or multipurpose points. At the region of highest uncertainty attending to (27b), the point (4.7, 35) is added to the multipurpose for the next iteration, $M_1$.

The subsequent iteration thus counts with modified sets, $Z_1$ and $M_1$, and the results can be seen on the second row of Fig.(10). The new point in $Z_1$ confines the high error region to the parameter space boundaries at the minimum and maximum angles of attack, reducing the error in the center of the parameter space. In terms of the predicted error, the point (4.6, 40) in the training set (marked (0)) has lowered the error in that region and now the estimated maximum is on the right hand side, in the area $\alpha \geq 4.6°$. The lack of testing points in the edges of minimum (4.5°) and maximum angle of attack (5°), unfortunately causes the statistical model to be oblivious to any error in those region along the parameter space borders. The issue is that the actual error function gradient is quite pronounced, therefore the mean error is not picked by the nearby points whereas the points are close enough to the edges for the variance in that region

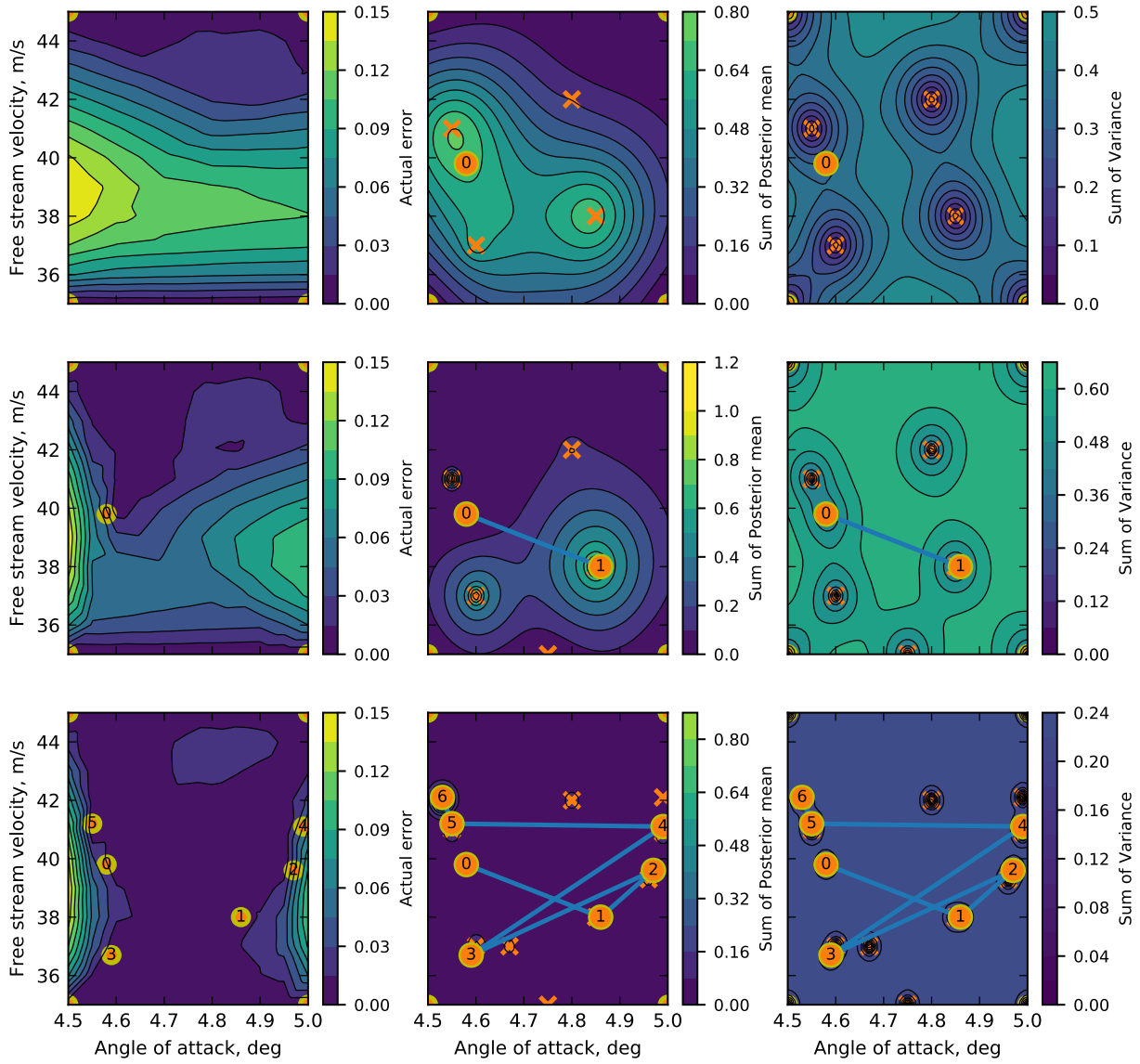



**Fig. 10 Multipurpose Bayesian sampling scheme evolution. Left to right: actual error in the interpolation; sum of individual Gaussian models expected mean error; sum of individual Gaussian models expected variance. Top to bottom: initial iteration, first iteration, seventh iteration. Orange circles denote training data (Z), where the number shows when they where added. Crosses denote multipurpose data (M).**

to not be significant

This example may illustrate the current need for the bias in each of the acquisition functions, which for the error acquisition function is clearly visible in the mean error of the initial iteration, where the added point is displaced from the maximum. To guarantee the viability of the scheme, ideally, we would like to avoid finding as next points to be added to $Z$ or $M$ points that are already known. Given that the scheme relies on using different combinations of the points in $M$ to produce training and testing data splits, the algorithm would not be viable once the number of points in $M$ is too small compared to that in $Z$ such that no balanced combination is possible. This will occur if the number of

points is significantly different to that in $Z$. This can happen as, due to the small number of testing points, and to keep it as low as possible, the error functions will be highly influenced by the actual error at these points. Therefore, if after each iteration the maximum is found at points already known it would result in a transfer of points from $M$ to $Z$, and if the point of maximum uncertainty (27b) is already in $M$ (as at the end of iteration 6) the scheme may soon become ill-conditioned. Therefore, the bias in (27) is introduced in hopes that the maximum error and variance gets shifted slightly by the other quantity. This is of particular importance given that the parameter space is forced to be discrete to avoid computing multiple points that differ from one another by small decimals.

The reason for the training-only set, $Z$, and that, unlike $M$, it can only grow in the number of points, is to avoid large error regions to dominate and hinder the performance of the scheme. This would happen, for instance, when an error region is significantly larger than others. Then, the scheme may add a point in such area but if it is significant, the individual sets that do not have that point as training but rather testing, will cause the sum of the predicted errors to still show that area as a region to sample more finely. Thus, if the point in that region is taken care of by keeping it as training only, the scheme can focus on other regions of the parameter space.

The last row of Fig. 10 shows the result after a total of 7 new points have been added to the initial training set that was the 4 corners of the parameter space. In terms of the actual error it can be seen that the absolute value of the maximum error is considerably reduced compared to that at the start across the majority of the parameter space, however, the parameter space edges do still show a non-negligible error. The expected mean error unfortunately does not capture these error regions, as for the internal error model the error is mostly uniform and a flat surface; at this point, the convergence of the algorithm is significantly reduced, as it is looking for a maximum on a flat surface. The flat surface phenomena is an issue encountered in the current implementation where the Gaussian process regression for the individual sets becomes a flat surface through zero. This occurs normally in sets where the number of training points starts being considerable, points at which the Gaussian model "knows" that the error is zero, and, therefore, if the number of known zeros is large when building the model compared to the testing data, it may be the case that that optimization (of building the GP model) determines that a flat surface through zero as the optimal fit, and assumes the testing data values are noise in the signal. In these cases, the testing error values appear as very thin "spikes" in the data, as opposed to larger perturbations as shown in the initial iterations in Fig. 10. These flat surfaces in the underlying GP models of the individual sets hinder the performance of the scheme, since they are futile in predicting a large error region in the parameter space as a whole. A mitigating method is to take the overall error as the sum of the individual sets, which would effectively ignore these "zero-error" predictions as long as other sets do include proper fits to the training and testing data. However, if a large number of sets exhibits this behavior the algorithm is compromised. At this stage in the development, this is observed in particular as the number of training points starts being considerable.

Figure 10 also illustrates the path of the newly added training points, where the first one is marked as "0". It can be seen that the large error region first identified on the left hand side is not fully resolved. The reason for this delayed

convergence on this high error region is because of the large gradient in the error function, which grows significantly from the $\alpha = 4.6°$ to $\alpha = 4.5°$ slice. To the eyes of the scheme, the 4.5 degrees section is taken care of with the training point "0", "3" and "5" and the nearby multipurpose points, which cause the variance to be low. Thus, the smaller (or narrower in this case) the error regions the harder it will be to capture them due to the nature of the Gaussian process with limited information, analogous to aliasing of high frequency dynamics with discrete temporal sampling.

A final point to address is whether this scheme offers any computational benefit to a traditional sampling technique as after the 7 iterations shown, in total 21 new points have been computed (including the parameter space corners), of which 10 are training-only points and the remaining are used as multipurpose points to construct the error surrogates. A refined LHS grid may be adequate for some cases, as in general if refined enough critical (dominant) spots in the domain will very likely be sampled. However, very refined sampling may be necessary to capture a region of changing dynamics resulting in others being overly sampled. Thus, this method, as will be shown now, is targeted at those underlying systems where dynamics change in specific regions of the parameter space and for larger dimensional parameter spaces which would highly benefit from targeted sampling. The next section will address these questions and compare the adaptive Bayesian scheme to a regular grid-based interpolation scheme.

### C. Multipurpose Bayesian Sampling applied to the Pazy Wing

The previous subsection focused on showing, in detail, the inner workings of the multi-purpose Bayesian sampling. Therefore, now the proposed algorithm is applied to the full two-dimensional parameter space of the Pazy wing described as

$$P = [\alpha = 0°, \alpha = 5°] \times [U_\infty = 30\,\mathrm{m/s}, U_\infty = 90\,\mathrm{m/s}] \in \mathbb{R}^2, \tag{28}$$

discretized in $0.5°$ and $2\,\mathrm{m/s}$ increments. This is a complex design space with two instability regions which will be used to judge the feasibility of the interpolation methods with optimal sampling. The end goal of the optimal sampling scheme is to produce an effective set of training points that can be used to produce an accurate interpolation across the entire parameter space to identify the flutter boundaries and compute the wing root loads. The costliest computational part is finding these points, as once these points are given the interpolation between reduced state-spaces across the entire space can be produced inexpensively.

We will first explore intricacies of the proposed scheme when looking at such a large parameter space and explaining the use of different error metrics that may affect the resulting sampling. Then, we will use the fully refined mesh of the Pazy wing to compare the process against standard sampling methods.

#### *1. Error function comparison*

A design routine would aim to produce root loads and gather information on the stability in an efficient manner such that it could be incorporated, for instance, in a global design optimization routine. Therefore, we explore the use of two

error functions: the eigenvalue-based function and the steady root loads-based function.

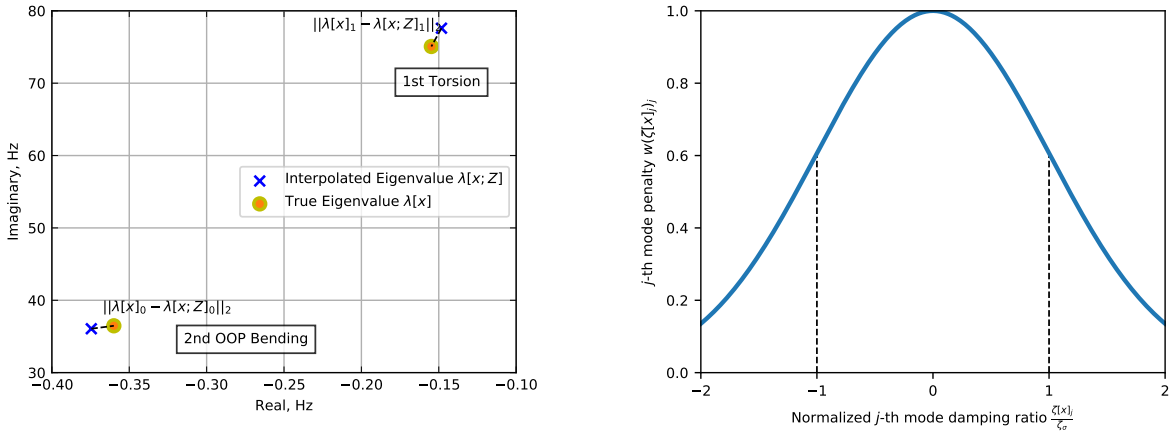
The eigenvalue-based cost function has been conceived for an interpolation scheme used to find the stability boundary of a system, i.e. where the eigenvalues have zero real part. For this case, the interpolation cost function takes the $N$ eigenvalues of the interpolated and true systems, $\lambda[\boldsymbol{p}; Z]$ and $\lambda[\boldsymbol{p}]$ respectively, and finds the Euclidean distance between corresponding eigenvalues, weighted based on the distance to the imaginary axes as

$$\varepsilon^{\text{flutter}}[\boldsymbol{p}; Z] = \sum_{j=1}^{N_{\text{eigs}}} w(\zeta[\boldsymbol{p}]_j) ||\lambda[\boldsymbol{p}, Z]_j - \lambda[\boldsymbol{p}]_j||_2, \text{ for } \boldsymbol{p} \in P_0 \tag{29}$$

where the comparison is performed across all aeroelastic eigenvalues ($N_{\text{eigs}} = 64$) and the weight $w(\cdot)$ is given by a Gaussian shaped function parametrised by the damping ratio of the current mode, $\zeta[\boldsymbol{p}]_j$,

$$w(\zeta[\boldsymbol{p}]_j) = \exp\left(\frac{-\zeta[\boldsymbol{p}]_j^2}{2\zeta_\sigma^2}\right). \tag{30}$$

This weighting function highlights errors closest to the stability boundary, as modes that are highly damped (or have already crossed the imaginary axes in the case of multiple unstable modes, hence the symmetric shape) do not drive the cost function. The parameter $\zeta_\sigma$ is user-defined to mark the damping ratio range between which modes are most penalized. For a clearer representation, Fig. 11, illustrates the cost function and weighted-averaging method.




**(a) Example of euclidean distance based metric between eigenvalues shown on Argand diagram close-up of the Pazy wing.**

**(b) $\zeta_\sigma$-normalized damping-ratio penalty function to weigh eigenvalue distance.**

**Fig. 11 Graphical representation of the cost function based on distance between true and interpolated eigenvalues. Reprinted from [68] with permission. Copyright 2022 by the authors.**

For the steady root loads error metric, the previously described function (26) is used. Recall that any cost function can only be evaluated at those points in the parameter space for which an instance of the true system exists, i.e. $\boldsymbol{p} \in P_0$.

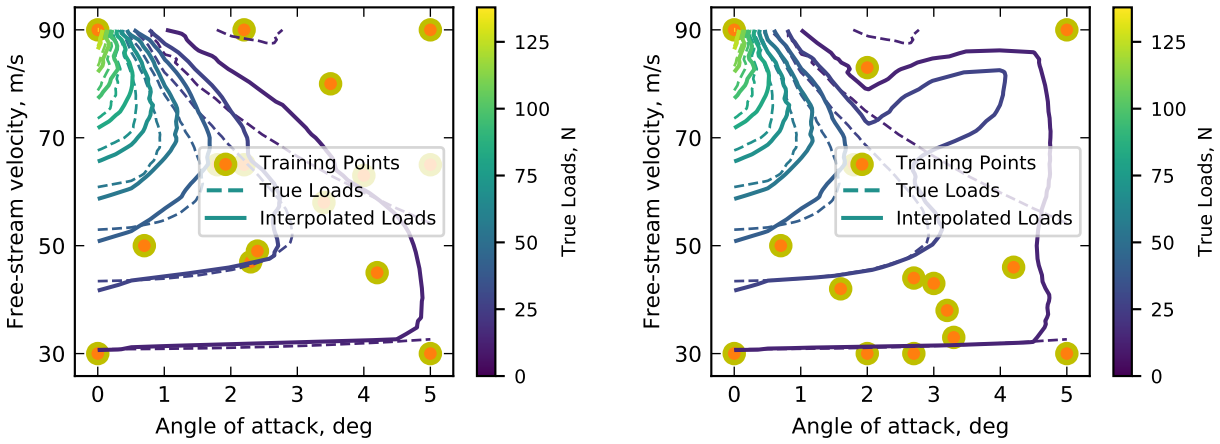
To compare the proposed different error functions, we proceed with two simulations. The resulting loads and stability plots are shown in Figs. 12 and 13. Starting with the comparison in the resulting steady root loads (Fig. 12), the resulting sampling points from the scheme using the error in load as metric in Fig. 12a produces a better approximation, i.e. there is no large excursion as in the high-$\alpha$ high-$U_\infty$ region as in Fig. 12b. In the former case, the points are more evenly distributed across the space, with a bit more emphasis on the center right region of the plot. Contrarily, the eigenvalue metric scheme has placed most of the points in the $U_\infty < 50\,\mathrm{m/s}$ region.

The stability plot of the Pazy wing is shown in Fig. 13 and compares the true underlying envelope with the interpolated predictions. The shaded regions indicate where the actual wing is unstable, the one at lower velocity being the first flutter mode, followed by a region of stability and then a second flutter mode at higher velocities. The location where the interpolated systems cross the imaginary axis is shown in $0.5^\circ$ increments. From these results the best performance comes from the scheme using the error in the loads as metric. Across both schemes, the first flutter (low speed) instability boundary is captured well, yet the errors appear in the stabilization and second flutter boundaries. Neither scheme captures any eigenvalue crossing of the imaginary axis in the region below $\alpha = 2^\circ$ and for high speeds, which is *a priori* the toughest region to interpolate given that the stability window is narrowest. In the loads metric scheme in Fig. 13a the stabilization boundary is well approximated in the $2^\circ \leq \alpha \leq 4^\circ$ region, yet it becomes less so as it nears the parameter edge. As seen in the previous example (Fig. 10), the interpolation tends to suffer at the edges of the parameter space. The second flutter boundary in general is not captured with sufficient consistency in the eigenvalue metric scheme (Fig. 13b). In this case, the abundance of sampling points in the lower velocity region appears to be responsible. However, in the loads error metric where the differing area is where most points are located the reason is less clear, unless capturing those eigenvalue crossings correctly requires of even a finer sampling.

To shed light on the reason the eigenvalue-metric simulation focuses on the lower speed range of the parameter space, Fig. 14 shows the actual and expected errors as the iterations advance in the scheme. The scheme initially does not identify the high error region in the upper velocity, low angle of attack area, and only estimates large errors at lower speed, and focuses its efforts on that region. The expected error is reduced in the zone it is targeted yet the issue again lies in the "flat surface" phenomena already described, as the statistical surrogate of the error function becomes a flat surface in the latter iterations, where testing data can be seen as very thin spikes. Again note that the value of the expected error is not to be compared directly against the actual error since the former is a sum of the expected error across all sets.

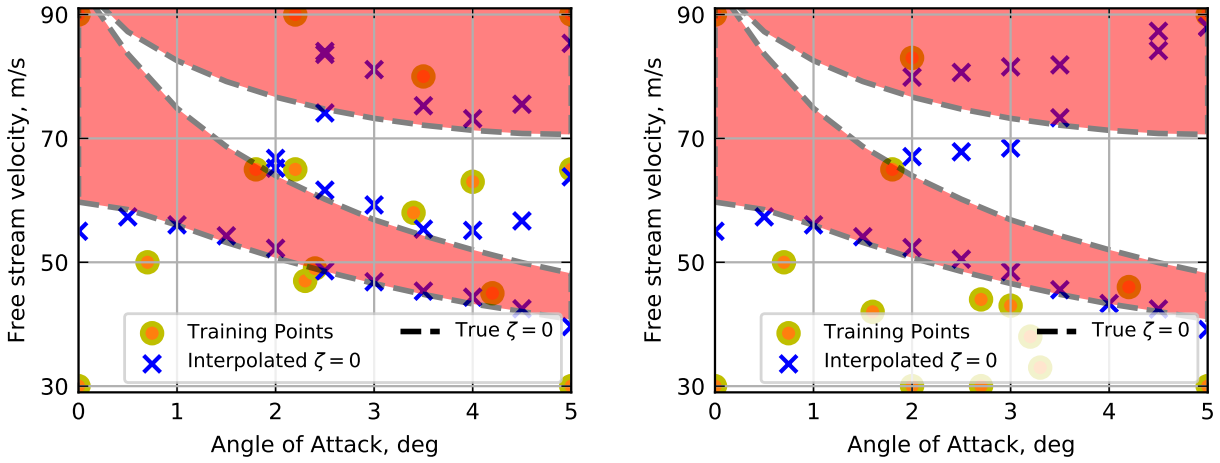
*2. High dimensional models and computational constraints*

The previous exploration has used a coarse discretization of the UVLM grid for the Pazy wing to identify the novel sampling scheme qualities, and now we are in the condition to assess the computational improvements in exploring the Pazy wing design space using this adaptive scheme scheme with respect to a regular grid. For this, we use a refined




**(a) Root loads error function** **(b) Eigenvalues error function**

**Fig. 12 Out-of-plane root loads comparison**

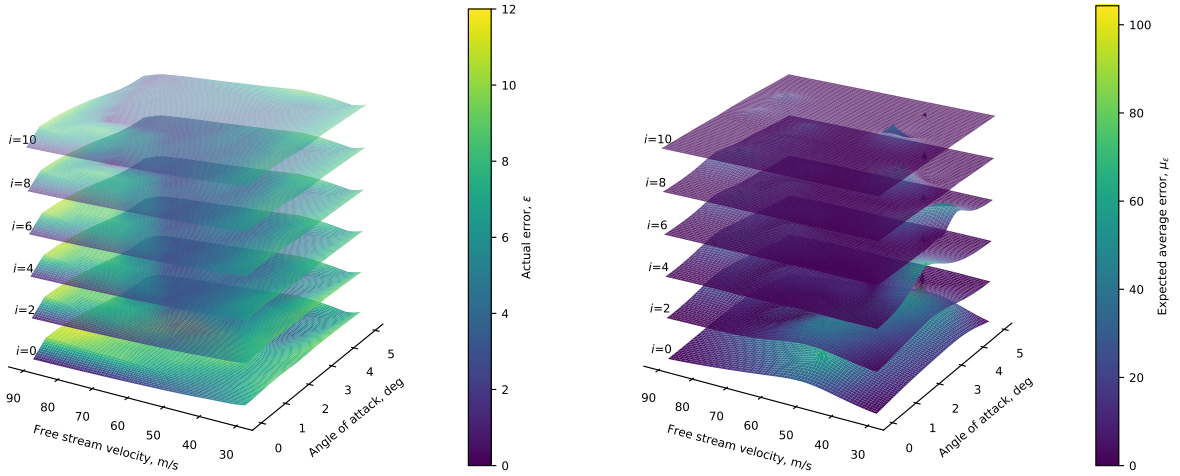



**(a) Root loads error function** **(b) Eigenvalues error function**

**Fig. 13 Stability envelope comparison**

mesh used for flutter analysis which is the mesh employed as case study for the Krylov reduction (Sec. IV.A) (the same discretisation employed in [4] to produce the wing's stability envelope). The models are reduced through the reduction process detailed in Sec. II.B from 22,458 states to 64 states in total. The first 8 structural modes are retained, where the 8th mode has a natural frequency of 272 Hz and convergence in the root loads computing process has been achieved. This leads to 16 structural modal states. The structural modes are used in the Krylov reduction on the aerodynamic system, which matches 6 moments about the steady-state gain, $\sigma = 0$, reducing the aerodynamic system from 22,080 to 48 states which, as in the previous case, has a relative error below $10^{-4}$ computed using (24) across the frequency spectrum below $k = 0.5$ and has enough margin to capture the highest-frequency unstable mode at $k \approx 0.31$ [4].

The resulting stability plots acquired by state-space interpolation with the different sampling schemes are shown in




**(a) Actual error** **(b) Expected error**

**Fig. 14 Evolution of the actual and expected error across the iterations in the eigenvalue error metric case.**

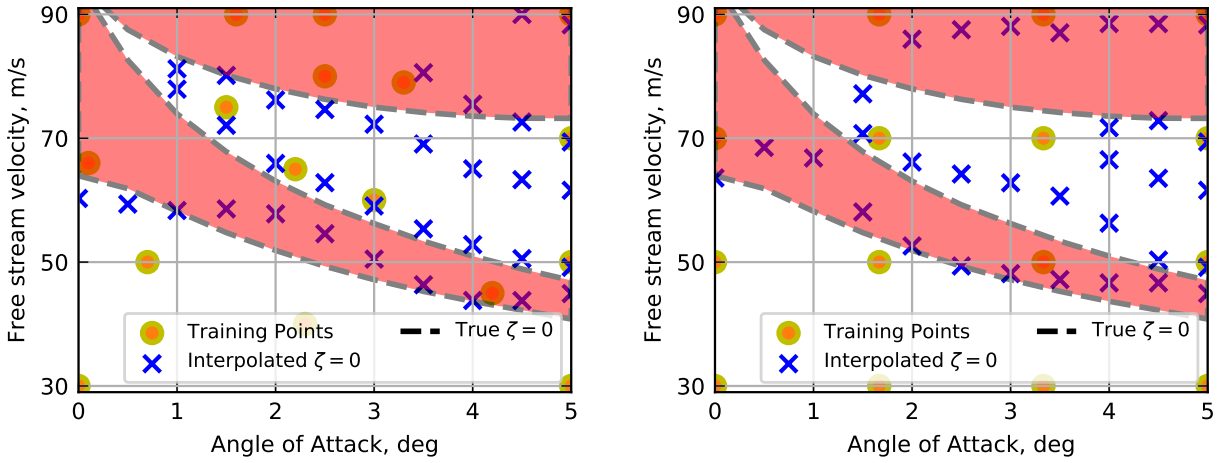
Fig. 15. The schemes use the same number of sampling points, 16, given by the limit of the multi-purpose Bayesian sampling scheme in reaching a "flat surface" estimate error prediction. The scheme is stopped at that point given that any point added subsequently does not satisfy to the specified acquisition criteria.

In general terms, there is an improvement with the adaptive sampling over the regular sampling, in particular in capturing the stabilization of boundary of the first instability and a better approximation to the instability offset of the second flutter mode. These two boundaries are the most complex to approximate, given that the instability region due to the underlying "humped" mode is very narrow and the unstable eigenvalue stays always close to the imaginary axis [4], thus the interpolation is very sensitive in this region. Consequently, the adaptive sampling scheme has opted for a more refined sampling in said region ($\alpha < 3°$ and $U_\infty > 60\,\mathrm{m/s}$) which renders a plausible approximation to this complex region of the underlying dynamics. Contrarily, the regular grid scheme (Fig. 15b) is designed with no knowledge of the system, thus the upper-left region does not benefit from the tailored sampling and fails to capture the second flutter mode instability boundary completely as it tends to reach the upper free stream velocity limit.

The second flutter boundary approximation in the adaptive scheme worsens at higher angles of attack, as the scheme predicts a relatively constant drop in the second flutter speed with angle of attack whereas in the real system the second flutter boundary flattens. However, it is visible how that area of the parameter space is not very finely sampled, yet, with very few samples it is able to already identify the general trend. In the regular grid case where this region is also undersampled, an additional crossing of the imaginary axes is predicted (indicated by the new line of blue crosses). This is of course a limitation of any interpolation with unknown underlying dynamics, unless there is a sample of that region (be it for testing or training) one cannot be sure whether the physical dynamics of the system exhibit a new change in the stability properties. However, these methods are exploring a large parameter space and therefore, if necessary, one

should focus with a more exhaustive sampling in areas of interest.

The time taken to produce these results is then what will determine whether the loss in accuracy can be acceptable provided they can be ran in less time than a refined evaluation of the true system. The time taken to run a single SHARPy case (following the process in Fig. 3 takes, on average, approximately 7 minutes (larger deflection cases take longer to converge). The multipurpose Bayesian sampling computes 16 of these points with an additional 16, which belong to the multipurpose set and are used to build the estimated error. In total, the entire adaptive scheme (including computing these 32 points) takes approximately 5 hours to run. The regular grid sampling only requires of 16 points, which can be found after 2 hours, albeit without any testing data to gauge the quality of the interpolation. Therefore, should there be an equally sized training set, the overhead due to the adaptive sampling optimization between the two schemes would only account for 1 hour of computing time. This approximation does save a significant amount of time compared to the 60 hours needed for the analysis in [4].




**(a) Adaptive multipurpose Bayesian sampling**

**(b) 4x4 regular grid**

**Fig. 15 Comparison of the stability envelope derived through state-space interpolation using different sampling schemes.**

## V. Concluding remarks

The results presented in the previous section do offer an initial proof of the benefits that can be obtained from an adaptive sampling scheme, where the data-recycling method introduced by the multipurpose set combinatorial method can save multiple costly full-order-system evaluations. This is not only applicable to this particular state-space interpolation, as it could be applied to higher-fidelity aeroelastic models that use CFD based ROMs, or implemented in design optimization problems, where any point at which the system is known becomes useful. This initial implementation has offered results that approximate the underlying complex dynamical system of a highly flexible wing, in approximately 10% of the time required to find the true system across the entire parameter space. While this example is restricted to

two dimensions, the scheme is scalable to higher dimensional spaces, where regular or non-adaptive grids are intractable or where some parameters drive the problem more than others.

In an effort for interpolation to alleviate the burden of computing thousands of operating conditions by interpolating among a reduced number of known "true" systems, several design choices have been made, that ultimately will determine the performance of the approach. Up to know we have based the performance evaluation of the sampling and interpolation method on the use of our knowledge of the entire parameter space (precisely what would be unknown in a real scenario) to compare between interpolated and "true" results. Further work, could focus on a unique and bound metric that can guarantee, up to certain confidence interval, based on a limited set of testing data, that all interpolated points fall within said confidence interval.

We have focused on "physical" metrics that are easily implemented for large MIMO systems. The use of classical error norms, such as $\mathcal{H}_2$ or $\mathcal{H}_\infty$, can not always be computed, in particular due to the nature of the linearised UVLM that lacks saturation and therefore, its frequency response of the loads to displacements is not bounded for very large frequencies, where the aerodynamic force grows unbounded with the rates and accelerations ($\boldsymbol{D}_{\text{uvlm}} \neq 0$). Whether these metrics currently employed are suited or better options are available becomes an important question (such as the Frobenius norm used in [35]), and whether it can be linked to evaluating the performance of the scheme across the entire parameter space as previously mentioned.

The current implementation based on Delaunay triangulation of the domain offers acceptable results, as shown in the previous section, but does not always offer convergence to underlying real functions as new points are added. This might be the case if the triangulation is modified by the addition of new points and this new tessellation is less favorable to the interpolation. The results presented herein use a linear barycentric interpolation, which thus far have offered the best performance, ahead of other tested methods such as a Clough-Tocher interpolation for two-dimensional parameter spaces. This scheme is based on fitting cubic polynomials on the tessellated domain, yet proved significantly more sensitive to changes in the underlying sampling points than the linear interpolation scheme. Alternative strategies based on the use of radial basis functions can be considered for the interpolation [35], and could further be optimized using machine learning techniques

A finding of particular importance in this work when using Gaussian surrogates of the error function function is the "flat surface" issue for large enough training sets. Once the number of training points becomes significant, the GP model starts fitting a flat surface through zero mean error, where the testing data points appear as localized, thin spikes that do not lead to the generation a significant error contour. If this occurs in a majority of the individual sets, the acquisition functions become ill-conditioned as it becomes search for a non-existent maximum. At this stage there is no benefit in continuing the scheme and the current implementation is then stopped. Work on the underlying surrogate models (which may not be restricted to Gaussian process) may alleviate this issue, allowing the scheme to be limited by a computational budget or an error metric.

The multipurpose Bayesian sampling method applied herein can be implemented in other interpolation or applications as part of the training process of a statistical surrogate. The data recycling scheme offers immediately an enhanced knowledge of the underlying *expensive-to-evaluate* function when the testing data is limited. The recent work in [35] focusing on interpolation methods for optimization of highly flexible transport aircraft has already shown promising results from the interpolation scheme using a dense LHS, which has been identified as over refined in certain regions of the parameter space. Thus, this adaptive sampling method would help overcome said inefficiencies in the generation of training data.

## Acknowledgements

Norberto Goizueta's research is sponsored by Department of Aeronautics at Imperial College under EPSRC grant EP/R513052/1. For the purpose of open access, the authors have applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising.

## References

[1] Lomax, T. L., *Structural Loads Analysis for Commercial Transport Aircraft*, 1st ed., American Institute of Aeronautics and Astronautics, 1996.

[2] Khodaparast, H. H., Georgiou, G., Cooper, J. E., Riccobene, L., Ricci, S., Vio, G. A., and Denner, P., "Efficient Worst Case "1-Cosine" Gust Loads Prediction," *ASDJournal*, Vol. 2, No. 3, 2012, pp. 33–54. https://doi.org/10.3293/asdj.2012.17.

[3] Palacios, R., Murua, J., and Cook, R. G., "Structural and Aerodynamic Models in Nonlinear Flight Dynamics of Very Flexible Aircraft," *AIAA Journal*, Vol. 48, No. 11, 2010, pp. 2648–2659. https://doi.org/10.2514/1.J050513.

[4] Goizueta, N., Drachinsky, A., Wynn, A., Raveh, D. E., and Palacios, R., "Flutter predictions for very flexible wing wind tunnel test," *Journal of Aircraft*, 2022. https://doi.org/10.2514/1.C036710, Article in Advance.

[5] Mallik, W., Schetz, J. A., and Kapania, R. K., "Rapid transonic flutter analysis for aircraft conceptual design applications," *AIAA Journal*, Vol. 56, No. 6, 2018, pp. 2389–2402. https://doi.org/10.2514/1.J056218.

[6] Venkatesan-Crome, C., and Palacios, R., "A discrete adjoint solver for time-domain fluid-structure interaction problems with large deformations," *AIAA SciTech Forum*, 2020. https://doi.org/10.2514/6.2020-0406.

[7] Lahooti, M., Palacios, R., and Sherwin, S. J., "Thick strip method for efficient large-eddy simulations of flexible wings in stall," *AIAA SciTech Forum*, 2021. https://doi.org/10.2514/6.2021-0363.

[8] Hodges, D. H., "Geometrically Exact, Intrinsic Theory for Dynamics of Curved and Twisted Anisotropic Beams," *AIAA Journal*, Vol. 41, No. 6, 2003, pp. 1131–1137. https://doi.org/10.2514/2.2054.

[9] Geradin, M., and Cardona, A., *Flexible multibody dynamics: a finite element approach*, John Wyley and Sons, Chichester, 2001.

[10] Su, W., and Cesnik, C. E., "Strain-based analysis for geometrically nonlinear beams: A modal approach," *Journal of Aircraft*, Vol. 51, No. 3, 2014, pp. 890–903. https://doi.org/10.2514/1.C032477.

[11] Drachinsky, A., and Raveh, D. E., "Modal rotations: A modal-based method for large structural deformations of slender bodies," *AIAA Journal*, Vol. 58, No. 7, 2020, pp. 3159–3173. https://doi.org/10.2514/1.J058899.

[12] Drachinsky, A., and Raveh, D. E., "Nonlinear aeroelastic analysis of highly flexible wings using the modal rotation method," *AIAA Journal*, Vol. 60, No. 5, 2022, pp. 3122–3134. https://doi.org/10.2514/1.j061065.

[13] Katz, J., and Plotkin, A., "Unsteady Incompressible Potential Flow," *Low Speed Aerodynamics*, Cambridge University Press, 2001, Chap. 13, 2nd ed., pp. 369–433. https://doi.org/10.1017/CBO9780511810329.

[14] Albano, E., and Hodden, W. P., "A doublet-lattice method for calculating lift distributions on oscillating surfaces in subsonic flows," *AIAA Journal*, Vol. 7, No. 2, 1969, pp. 279–285. https://doi.org/10.2514/3.5086.

[15] Kim, T., "Flutter prediction methodology based on dynamic eigen decomposition and frequency-domain stability," *Journal of Fluids and Structures*, Vol. 86, 2019, pp. 354–367. https://doi.org/10.1016/j.jfluidstructs.2019.01.022.

[16] Afonso, F., Vale, J., Oliveira, E., Lau, F., and Suleman, A., "A review on non-linear aeroelasticity of high aspect-ratio wings," *Progress in Aerospace Sciences*, Vol. 89, 2017, pp. 40–57. https://doi.org/10.1016/j.paerosci.2016.12.004.

[17] Benner, P., Gugercin, S., and Willcox, K. E., "A survey of model reduction methods for parametric systems," *MPI Magdeburg Preprints*, Vol. MPIMD, No. 13-14, 2013, pp. 1–36. URL http://www2.mpi-magdeburg.mpg.de/preprints/2013/MPIMD13-14.pdf.

[18] Scarciotti, G., and Astolfi, A., "Nonlinear model reduction by moment matching," *Foundations and Trends in Systems and Control*, Vol. 4, No. 3-4, 2017, pp. 224–409. https://doi.org/10.1561/2600000012.

[19] Grimberg, S., Farhat, C., and Youkilis, N., *On the stability of projection-based model order reduction for convection-dominated laminar and turbulent flows*, Vol. 419, 2020. https://doi.org/10.1016/j.jcp.2020.109681.

[20] Kumar, M., Rajagopal, K., Balakrishnan, S. N., and Nguyen, N. T., "Proper orthogonal decomposition technique for near-optimal control of flexible aircraft wings," *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013. https://doi.org/10.2514/6.2013-4935.

[21] Wang, B., Ma, X., Tian, K., Hao, P., Zhou, Y., and Quan, D., "Concurrent patch optimization of hybrid composite plates based on proper orthogonal decomposition," *AIAA Journal*, Vol. 57, No. 11, 2019, pp. 4915–4926. https://doi.org/10.2514/1.J058064.

[22] Kramer, B., and Willcox, K. E., "Nonlinear model order reduction via lifting transformations and proper orthogonal decomposition," *AIAA Journal*, Vol. 57, No. 6, 2019, pp. 2297–2307. https://doi.org/10.2514/1.J057791.

[23] Kim, Y., Kang, S.-H., Cho, H., and Shin, S., "Improved Nonlinear Analysis of a Propeller Blade Based on Hyper-Reduction," *AIAA Journal*, Vol. 60, No. 3, 2022, pp. 1909–1922. https://doi.org/10.2514/1.j060742.

[24] Maraniello, S., and Palacios, R., "State-Space Realizations and Internal Balancing in Potential-Flow Aerodynamics with Arbitrary Kinematics," *AIAA Journal*, Vol. 57, No. 6, 2019, pp. 2308–2321. https://doi.org/10.2514/1.J058153.

[25] Maraniello, S., and Palacios, R., "Parametric Reduced-Order Modeling of the Unsteady Vortex-Lattice Method," *AIAA Journal*, Vol. 58, No. 5, 2020, pp. 2206–2220. https://doi.org/10.2514/1.j058894.

[26] Grimme, E. J., "Krylov Projection Methods for Model Reduction," Ph.D. thesis, University of Illinois, Urbana-Champaign, 1997.

[27] Gugercin, S., "Projection Methods for Model Reduction of Large-Scale Dynamical Systems," Ph.D. thesis, Rice University, 2003. URL https://hdl.handle.net/1911/18536.

[28] Chu, C. C., Lai, M. H., and Feng, W. S., "Model-order reductions for MIMO systems using global Krylov subspace methods," *Mathematics and Computers in Simulation*, Vol. 79, No. 2008, 2007, pp. 1153–1164. https://doi.org/10.1016/j.matcom.2007.09.007.

[29] Ruhe, A., "Rational Krylov Algorithms for Nonsymmetric Eigenvalue Problems," *Linear Algebra and its Applications*, Vol. 197, No. 198, 1994, pp. 283–295. https://doi.org/10.1016/0024-3795(94)90492-8.

[30] Willcox, K. E., and Peraire, J., "Balanced Model Reduction via the Proper Orthogonal Decomposition," *AIAA Journal*, Vol. 40, No. 11, 2002, pp. 2323–2330. https://doi.org/10.2514/2.1570.

[31] Hesse, H., and Palacios, R., "Reduced-Order Aeroelastic Models for Dynamics of Maneuvering Flexible Aircraft," *AIAA Journal*, Vol. 52, No. 8, 2014, pp. 1717–1732. https://doi.org/10.2514/1.J052684.

[32] Artola, M., Goizueta, N., Wynn, A., and Palacios, R., "Proof of Concept for a Hardware-in-the-Loop Nonlinear Control Framework for Very Flexible Aircraft," *AIAA SciTech Forum*, AIAA Paper 2021-1392, 2021. https://doi.org/10.2514/6.2021-1392.

[33] Artola, M., Goizueta, N., Wynn, A., and Palacios, R., "Aeroelastic Control and Estimation with a Minimal Nonlinear Modal Description," *AIAA Journal*, Vol. 59, No. 7, 2021, pp. 2697–2713. https://doi.org/10.2514/1.J060018.

[34] Hesse, H., Palacios, R., and Murua, J., "Consistent structural linearisation in flexible aircraft dynamics with large rigid-body motion," *AIAA Journal*, Vol. 52, No. 3, 2014, pp. 528–538. https://doi.org/10.2514/1.J052316.

[35] Cea, A., and Palacios, R., "Parametric Reduced Order Models for Aeroelastic Design of Very Flexible Aircraft," *AIAA SciTech Forum*, 2022.

[36] Amsallem, D., and Farhat, C., "An Online Method for interpolating linear parametric reduced order models," *SIAM Journal on Scientific Computing*, Vol. 33, No. 5, 2011, pp. 2169–2198. https://doi.org/10.1137/100813051.

[37] Geuss, M., Panzer, H., and Lohmann, B., "On parametric model order reduction by matrix interpolation," *European Control Conference*, 2013, pp. 3433–3438. https://doi.org/10.23919/ecc.2013.6669829.

[38] Mavris, D. N., Bandte, O., and Delaurentis, D. A., "Robust design simulation: A probabilistic approach to multidisciplinary design," *Journal of Aircraft*, Vol. 36, No. 1, 1999, pp. 298–307. https://doi.org/10.2514/2.2437.

[39] Chaudhuri, A., Lam, R., and Willcox, K. E., "Multifidelity uncertainty propagation via adaptive surrogates in coupled multidisciplinary systems," *AIAA Journal*, Vol. 56, No. 1, 2018, pp. 235–249. https://doi.org/10.2514/1.J055678.

[40] Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., and Kevin Tucker, P., "Surrogate-based analysis and optimization," *Progress in Aerospace Sciences*, Vol. 41, No. 1, 2005, pp. 1–28. https://doi.org/10.1016/j.paerosci.2005.02.001.

[41] Villemonteix, J., Vazquez, E., and Walter, E., "An informational approach to the global optimization of expensive-to-evaluate functions," *Journal of Global Optimization*, Vol. 44, No. 4, 2009, pp. 509–534. https://doi.org/10.1007/s10898-008-9354-2.

[42] Villanueva, D., and Smarslok, B. P., "Using expected information gain to design aerothermal model calibration experiments," *17th AIAA Non-Deterministic Approaches Conference*, 2015. https://doi.org/10.2514/6.2015-0660.

[43] Huan, X., and Marzouk, Y. M., "Simulation-based optimal Bayesian experimental design for nonlinear systems," *Journal of Computational Physics*, Vol. 232, No. 1, 2013, pp. 288–317. https://doi.org/10.1016/j.jcp.2012.08.013.

[44] Riso, C., Ghadami, A., Cesnik, C. E., and Epureanu, B. I., "Data-driven forecasting of postflutter responses of geometrically nonlinear wings," *AIAA Journal*, Vol. 58, No. 6, 2020, pp. 2726–2736. https://doi.org/10.2514/1.J059024.

[45] Qi, P., and Zhao, X., "Flight control for very flexible aircraft using model-free adaptive control," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 3, 2020, pp. 608–619. https://doi.org/10.2514/1.G004761.

[46] Singh, V., and Willcox, K. E., "Decision-making under uncertainty for a digital thread-enabled design process," *Journal of Mechanical Design, Transactions of the ASME*, Vol. 143, No. 9, 2021, pp. 1–12. https://doi.org/10.1115/1.4050108.

[47] Wynn, A., Artola, M., and Palacios, R., "Nonlinear optimal control for gust load alleviation with a physics-constrained data-driven internal model," *AIAA SciTech Forum*, 2022. https://doi.org/10.2514/6.2022-0442.

[48] European Union Aviation Safety Agency, "EASA Concept Paper : First usable guidance for Level 1 machine learning applications A deliverable of the EASA AI Roadmap," Tech. Rep. 1, EASA, 2021. URL https://www.easa.europa.eu/downloads/126648/en.

[49] Avin, O., Raveh, D. E., Drachinsky, A., Ben-Shmuel, Y., and Tur, M., "An Experimental Benchmark of a Very Flexible Wing," *AIAA SciTech Forum*, 2021. https://doi.org/10.2514/6.2021-1709.

[50] del Carre, A., Muñoz-Simón, A., Goizueta, N., and Palacios, R., "SHARPy : A dynamic aeroelastic simulation toolbox for very flexible aircraft and wind turbines," *Journal of Open Source Software*, Vol. 4, No. 44, 2019, p. 1885. https://doi.org/10.21105/joss.01885.

[51] Simpson, R. J. S., and Palacios, R., "Numerical aspects of nonlinear flexible aircraft flight dynamics modeling," *54th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2013. https://doi.org/10.2514/6.2013-1634.

[52] Hesse, H., "Consistent Aeroelastic Linearisation and Reduced-Order Modelling in the Dynamics of Manoeuvring Flexible Aircraft," Ph.D. thesis, Imperial College London, 2013. https://doi.org/10.25560/12258.

[53] del Carre, A., Teixeira, P. C., Palacios, R., and Cesnik, C. E., "Nonlinear response of a very flexible aircraft under lateral gust," *International Forum on Aeroelasticity and Structural Dynamics 2019, IFASD 2019*, 2019, pp. 1–27.

[54] Murua, J., Palacios, R., and Graham, J. M. R., "Applications of the unsteady vortex-lattice method in aircraft aeroelasticity and flight dynamics," *Progress in Aerospace Sciences*, Vol. 55, 2012, pp. 46–72. https://doi.org/10.1016/j.paerosci.2012.06.001.

[55] Simpson, R. J. S., Palacios, R., and Murua, J., "Induced-Drag Calculations in the Unsteady Vortex Lattice Method," *AIAA Journal*, Vol. 51, No. 7, 2013, pp. 1775–1779. https://doi.org/10.2514/1.j052136.

[56] Lambert, T., and Dimitriadis, G., "Induced Drag Calculations with the Unsteady Vortex Lattice Method for Cambered Wings," *AIAA Journal*, Vol. 55, No. 2, 2016, pp. 668–672. https://doi.org/10.2514/1.j055135.

[57] Goizueta, N., Drachinsky, A., Wynn, A., Raveh, D. E., and Palacios, R., "Flutter prediction for a very flexible wing wind tunnel test," *AIAA SciTech Forum*, 2021. https://doi.org/10.2514/6.2021-1711.

[58] Antoulas, A. C., "Model Reduction using Krylov Methods," *Approximation to Large Scale Dynamical Systems*, SIAM - Advances in Design and Control, 2005, Chap. 11, pp. 343–361.

[59] Antoulas, A. C., *Approximation of Large-Scale Dynamical Systems*, 1st ed., SIAM - Advances in Design and Control, 2005.

[60] Goizueta, N., Wynn, A., and Palacios, R., "Parametric Krylov-based order reduction of aircraft aeroelastic models," *AIAA SciTech Forum*, 2021. https://doi.org/10.2514/6.2021-1798.

[61] Barber, C. B., Dobkin, D. P., and Huhdanpaa, H. T., "The Quickhull algorithm for convex hulls," *ACM Trans. on Mathematical Software*, Vol. 22, No. 4, 1996, pp. 469–483. URL http://www.qhull.org.

[62] Stilwell, D. J., and Rugh, W. J., "Stability preserving interpolation methods for the synthesis of gain scheduled controllers," *Automatica*, Vol. 36, No. 5, 2000, pp. 665–671. https://doi.org/10.1016/S0005-1098(99)00193-4.

[63] Amsallem, D., and Farhat, C., "Interpolation method for adapting reduced-order models and application to aeroelasticity," *AIAA Journal*, Vol. 46, No. 7, 2008, pp. 1803–1813. https://doi.org/10.2514/1.35374.

[64] Panzer, H., Mohring, J., Eid, R., and Lohmann, B., "Parametrische ordnungsreduktion mittels matrixinterpolation," *At-Automatisierungstechnik*, Vol. 58, No. 8, 2010, pp. 475–484. https://doi.org/10.1524/auto.2010.0863.

[65] Allemang, R. J., "The modal assurance criterion - Twenty years of use and abuse," *Sound and Vibration*, Vol. 37, No. 8, 2003, pp. 14–21.

[66] Amsallem, D., Cortial, J., Carlberg, K., and Farhat, C., "A method for interpolating on manifolds structural dynamics reduced-order models," *International Journal for Numerical Methods in Engineering*, Vol. 80, 2009, pp. 1241–1288. https://doi.org/10.1002/nme.2681.

[67] Degroote, J., Vierendeels, J., and Willcox, K. E., "Interpolation among reduced-order matrices to obtain parameterized models for design, optimization and probabilistic analysis," *International Journal for Numerical Methods in Fluids*, Vol. 63, 2010, pp. 207–229. https://doi.org/10.1002/fld.2089.

[68] Goizueta, N., Wynn, A., and Palacios, R., "Fast flutter evaluation of very flexible wing using interpolation on an optimal training dataset," *AIAA SciTech Forum*, AIAA Paper 2022-1345, 2022. https://doi.org/10.2514/6.2022-1345.

[69] Mockus, J., *Bayesian Approach to Global Optimization*, 1st ed., Springer, Dordrecht, 1989.

[70] The GPyOpt Authors, "GPyOpt: A Bayesian Optimization framework in Python," http://github.com/SheffieldML/GPyOpt, 2016.

[71] Riso, C., and Cesnik, C. E., "Correlations Between UM/NAST Nonlinear Aeroelastic Simulations and the Pre-Pazy Wing Experiment," *AIAA SciTech Forum*, AIAA Paper 2021-1712, 2021. https://doi.org/10.2514/6.2021-1712.

[72] Düssler, S., Goizueta, N., Muñoz-Simón, A., and Palacios, R., "Modelling and Numerical Enhancements on a UVLM for Nonlinear Aeroelastic Simulation," *AIAA SciTech Forum*, AIAA Paper 2022-2455, 2022. https://doi.org/10.2514/6.2022-2455.