Parametric Reduced Order Models for Aeroelastic Design of Very Flexible Aircraft

Alvaro Cea* and Rafael Palacios†

Department of Aeronautics, Imperial College, London SW7 2AZ, United Kingdom

A description for flutter and dynamic loads prediction is laid out based on state-space systems around a nonlinear equilibrium. In order to reduce the computational times associated with building and manipulating these systems, a new methodology is proposed to build parametric reduced-order models of the unsteady aerodynamic systems from optimal Latin-Hypercube design of experiments, frequency-limited balancing techniques and interpolation on the transfer functions of the resulting reduced systems. Optimization of the kernels encapsulating the parametric dependence is also studied. The developed tools are employed for the aeroelastic design optimization of a very flexible strut-braced wing with flutter constraints. Results show a promising approach to be deployed in design problems where dynamics and stability are to be included in the analysis and geometrically nonlinear effects are to be accounted for.

I. Introduction

Next-generation passenger aircraft will need to meet much stricter emissions and fuel efficiency targets. This will undoubtedly accelerate the push towards, 1) increased use of advanced lighter materials to reduce operating weight and configurations with very high aspect ratio wings (HARWs) to reduce induced drag (given the loss of lift and increase of drag due to tip effects) in low-aspect-ratio wings). The inevitable increase of flexibility may pose some challenges regarding the interaction of disciplines traditionally decoupled and the appearance of nonlinear effects. As a consequence, aeroelastic analysis is set to dominate many aspects of the design and some of the current methods based on linear analysis may need a fundamental revision [1].

Adding new features into the design process also requires consideration about the complexity and size of the actual models, and the vast amount of certification exercises that include mass cases, manoeuvres, gust loads at different velocities and altitudes, which combined accounts for over 100,000s simulations [2] to construct the design envelopes. Thus any effort to introduce new physics into the analysis will have to be accompanied of novel techniques to reduce the computational overhead related to them. Reduced Order Models (ROMs) techniques combined with more detailed analysis tools have potential to solve the aforementioned issues and consequently are receiving strong attention [3].

We propose a method for fast evaluation of the dynamic loads and flutter constraints in parametric aircraft designs that

*Research associate, Room CAGB 309, South Kensington Campus

†Professor of Computational Aeroelasticity, CAGB 338, South Kensington Campus, Associate Fellow AIAA (email: r.palacios@imperial.ac.uk).

builds on the frequency-limited balancing of the unsteady aerodynamics of [4]. The goal is to build a framework that 1) is robust, 2) can produce aeroelastic studies that account for large excursions of the wings, and 3) is fast such that it can be used in aircraft design optimization. The aerodynamic solution is based on the Unsteady Vortex Lattice Method (UVLM) [5], a time domain numerical solution of the potential Laplace equation using vortex ring singularities. The method allows linearisation around arbitrary geometries and a free wake, which will be exploited here to obtain efficient ROMs suitable for HARWs aeroelastic analysis. For structural modelling, geometrically nonlinear composite beams have been extensively used to describe HARWs with acceptable precision (see, for instance, the review by [6]). While methodologies based on detailed models are available [7, 8], this work focuses on the construction of reduced models that will be used on preliminary design studies and optimization, therefore a beam-based solution is considered as the best compromise between fidelity and efficiency. We performed an optimization of a highly flexible strut-braced wing using a couple solution with GEMSEO, an optimization library built for multidisciplinary design optimization (MDO) [9].

The structure of this paper is as follows. Sec. II presents the aeroelastic formulation that serves as the theoretical backbone of this work; sec. III introduces the parametric model reduction techniques used to improve the performance of the aerodynamic solution; numerical results are shown in sec. IV, including the coupling with GEMSEO to built an aeroelastic design optimization framework; finally sec. V concludes with a summary of the methodology and results, highlighting the next steps to further expand the tools developed in this work.

II. Aeroelastic formulation

The full aeroelastic system is formed by coupling a geometrically-exact composite beam model (GEBM) formulation [10, 11] with the Unsteady Vortex Lattice Method (UVLM) [5] and linearising around arbitrary configurations. This has been implemented into the code SHARPy [12], a modular framework for aeroelastic simulations.

This aerodynamic panel method solves the inviscid, incompressible flow equations, which can be cast into the form of the Laplace potential equation. Discretization into bound and wake panels of vorticity $\mathbf{\Gamma}_b$ and $\mathbf{\Gamma}_w$ respectively, leads to the equations for the non-penetration boundary condition at the bound lattice, and the wake propagation by enforcing the Kutta-Joukowski condition, written respectively as

$$\begin{aligned}\mathcal{A}_{bb}\mathbf{\Gamma}_b^{n+1} + \mathcal{A}_{bw}\mathbf{\Gamma}_w^{n+1} &= \mathcal{B}\mathbf{u}_b^{n+1} \\ \mathbf{\Gamma}_w^{n+1} + \mathcal{A}_{ww}\mathbf{\Gamma}_w^n + \mathcal{A}_{wb}\mathbf{\Gamma}_b^n &= \mathbf{0}\end{aligned}\tag{1}$$

where $\mathbf{u}_b$ is the input vector containing the velocities at the corners of each panel from structural deflections and rigid body motions, and possibly atmospheric disturbances and deployment of control surfaces. The $\mathcal{A}$ and $\mathcal{B}$ matrices depend on the instant geometry of both bound and wake panels, giving the nonlinear nature to the solution. Note that

the equations are defined in discrete time which facilitates the solution and integration with the structural model. A linearization of these equation is sought around a frozen geometry and wake shape as in [4], which yields a discrete, linear time invariant (DLTI) system in state-space form:

$$\begin{aligned}\mathbf{x}_a^{n+1} &= \mathbf{A}_a \mathbf{x}_a^n + \mathbf{B}_a \mathbf{u}^n \\ \mathbf{y}_a^{n+1} &= \mathbf{C}_a \mathbf{x}_a^n + \mathbf{D}_a \mathbf{u}^n\end{aligned}\tag{2}$$

where the aerodynamic state vector has been assembled with the perturbations of the vorticity and its derivative, $\mathbf{x}_a^n = [\Delta\Gamma^n; \Delta\Gamma_w^n; dt\Delta\Gamma'^n; \Delta\Gamma^{n-1}]^\top$. The aerodynamic forces, given at each vertex are concatenated in the output vector $\mathbf{y}_a$. This will be the system that we aim to reduce using projection based methods.

The nonlinear structural dynamics of the GEBM are solved after a finite-element discretization [13]. The main variables are displacements and rotations (parameterized using the Cartesian Rotation Vector (CRV)) at each node, which are concatenated in the vector $\boldsymbol{\eta}$; and the linear and angular velocities of a reference point with respect to an inertial frame of reference, $\boldsymbol{\beta} = [\mathbf{v} \ \boldsymbol{\omega}]$, which describe the rigid body dynamics. Similar to the aerodynamic description, the structural equations are linearized around a reference equilibrium given by $\boldsymbol{\eta}_0, \boldsymbol{\beta}_0$ [14]:

$$\mathcal{M}(\boldsymbol{\eta}_0) \begin{bmatrix} \Delta\boldsymbol{\eta}'' \\ \Delta\boldsymbol{\beta}' \end{bmatrix} + \mathcal{C}(\boldsymbol{\eta}_0, \dot{\boldsymbol{\eta}}_0, \boldsymbol{\beta}_0) \begin{bmatrix} \Delta\boldsymbol{\eta}' \\ \Delta\boldsymbol{\beta} \end{bmatrix} + \mathcal{K}(\boldsymbol{\eta}_0, \dot{\boldsymbol{\eta}}_0, \boldsymbol{\beta}_0) \begin{bmatrix} \Delta\boldsymbol{\eta} \\ \mathbf{0} \end{bmatrix} = \frac{\rho_\infty \bar{u}_\infty^2}{2} \begin{bmatrix} \Delta\mathbf{N}_\eta \\ \Delta\mathbf{N}_\beta \end{bmatrix}\tag{3}$$

with $\Delta\mathbf{N}_\eta$ and $\Delta\mathbf{N}_\beta$ the corresponding incremental external forces on the structure and the reference point. These equations are cast in non-dimensionalized form with a reference length $\bar{b}$, velocity $\bar{u}_\infty$, density $\bar{\rho}_\infty$. Note that projection onto the modes of the structure at equilibrium is also possible and reduces significantly the size of the structural system. Realization into discrete-time state space system is attained using a Newmark- β time integration scheme such that inputs are the external forces, $\mathbf{u}_s = [\Delta\mathbf{N}_\eta, \Delta\mathbf{N}_\beta]$, and the outputs $\mathbf{y}_s = [\Delta\boldsymbol{\eta}, \Delta\boldsymbol{\eta}', \Delta\boldsymbol{\beta}]$. The coupling with the aerodynamic system is then achieved via mapping of forces and displacements between the beam nodes and the aerodynamic grid, and addition of aeroelastic terms from the equilibrium condition as described in [14]. The outcome is an aeroelastic system that is very large but also sparse, and projection-based ROMs can provide a good solution to reduce its size. For design applications, however, the reduction process itself might be too costly and Parametric Model Order Reduction (PMORs), with an offline stage where the parametric dependence of the ROMS is built into a metamodel, and a fast online stage, can be a good compromise and will be explored in this work. Even though Parametric Model Order Reduction (PMORs) systems can be built for the coupled aeroelastic system (as done for instance in [14]), we restrict its application to the aerodynamic description for two main reasons: 1) the most expensive part and the larger matrices are found in the aerodynamics due to the propagation of the wake; 2) the ultimate objective of this methodology is to perform design and optimisation analysis, and having the structural description part of the ROM would require adding

the structural variables to the design parameters, which would create an extremely large parameterization space. The strategy we propose is depicted in Fig.1. A set of design parameters determine our aircraft, of which a subset defines the aerodynamic main surfaces and those are fed into the PMOR library together with the equilibrium deformations from a trim analysis (which is achieved using static solvers at moderately low cost). This library returns a reduced a linearized aerodynamic system, which coupled with a linearization of the structural equations around the trim state yields a linear dynamic system suitable for flutter and load analysis which will be described next.

Fig. 1 Schematic of the proposed aeroelastic solution

A. Flutter analysis and continuous turbulence loads

Calculation of the flutter boundaries is an essential part of the aircraft design process. Different techniques do exist [15], yet the linearisation of the aeroelastic system, in the time domain and with respect to a nonlinear trim configuration offers a two fold advantage: flutter points can be obtained directly from standard eigenvalue analysis, and the same system is used to compute the dynamic loads on the aircraft components. Fig. 2 shows the various state space systems and steps involved in the evaluation of flutter and dynamic loads in the current approach. For the flutter calculation only the state-space labeled "System nonlinear equilibrium" in the figure is needed. The approach has been implemented and validated in [16], including a wind tunnel test with a highly flexible wing: after linearization around a set of operating conditions, a sweep across flying speeds is carried out to find the point of zero damping at which the system will become unstable. Thanks to the nondimensional casting of the equations, this only involves updating u_∞ in the structural system in Eqs. (3), which is usually significantly smaller than the aerodynamic system in Eqs. (1). In order to have this methodology embedded into an optimization problem, obtaining the velocity at which the system turns unstable needs to be automatic and efficient, so Algorithm 1 has been implemented combining a stepping procedure where the interval on which flutter occurs is found and a secant method for updating the new evaluation. While this has proven an efficient algorithm, in some instances of the optimization in Sec. IV.B the secant method would take many iterations to converge.

A new algorithm has been implemented that triggers a bisection approach when the secant method struggles because of the flat slope in the eigenvalue evolution. The two combined achieve a good performance and do not suffer from convergence issues in the optimization problems tested.

Fig. 2 Flutter and dynamic loads solution process

Once the aeroelastic system of the aircraft has been determined to be stable under the specified operating conditions, dynamic atmospheric loads that the aircraft must withstand need to be calculated. Spectral turbulence generation methods are encouraged by aircraft certification agencies. The statistical properties of the random vector variables $\mathbf{X}$ under analysis follow a significant number of assumptions. Ergodicity of the process allow to obtain ensemble averages as time averages,

$$\bar{\mathbf{X}} = E[\mathbf{X}(t)] = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T \mathbf{X}(t) dt \quad (4)$$

which together with the assumption of a stationary process yields the correlation function as $\phi_{\mathbf{X}}(\tau) = E[\mathbf{X}(t - \tau) \otimes \mathbf{X}(t)]$. The autocorrelation matrix is then defined as $\Sigma_{\mathbf{X}} = \phi_{\mathbf{X}}(0)$. A **Gaussian** process gives the probability density function of a normal distribution

$$f(\mathbf{X}) = \frac{1}{(2\pi)^{n/2} |\Sigma_{\mathbf{X}}|^{1/2}} \exp \left(-\frac{1}{2} (\mathbf{X} - \bar{\mathbf{X}})^T \Sigma_{\mathbf{X}}^{-1} (\mathbf{X} - \bar{\mathbf{X}}) \right) \quad (5)$$

with n the number of dimensions of the variable $\mathbf{X}$. At high altitudes, the turbulence field can be considered **isotropic**,

and the power spectral density for unit, root-mean-square gust intensity is given by the **Von Kármán spectrum**:

$$\Phi_{w_g}(\kappa) = \frac{2L_w}{\pi} \frac{1 + \frac{8}{3}(2.678L_w\kappa)^2}{(1 + (2.678L_w\kappa)^2)^{\frac{11}{6}}} \quad (6)$$

with L_w a scale of turbulence and κ the ratio between frequency of encounter and the aircraft flight speed. Eq. (6) can be approximated by a rational function approximation, thus building a synthetic turbulence filter realizable in state-space form with white noise input and the external atmospheric disturbance as output. As outlined in Fig. 2, the goal is then to build a system that takes white noise as input and process the loads at given stations of the structure. Lyapunov's equation can be used for calculating the covariance matrices of the resulting aeroelastic system [17], $\Sigma := [\mathbf{A}_e, \mathbf{B}_e; \mathbf{C}_e, \mathbf{D}_e]$:

$$\begin{aligned} \mathbf{A}_e \Sigma_x + \Sigma_x \mathbf{A}_e^\top + \mathbf{B}_e \Sigma_n \mathbf{B}_e^\top &= \mathbf{0} \\ \Sigma_y &= \mathbf{C}_e \Sigma_x \mathbf{C}_e^\top \end{aligned} \quad (7)$$

Note that the general random variable $\mathbf{X}$ in the equations above is here the output loads vector, which is usually taken for certification in pairs such as torsion and bending, i.e. of dimension two. In this case Eq. (5) yields equiprobable ellipses on which to evaluate envelopes of critical loads. The covariance matrix of the loads has the form,

$$\Sigma_y = \begin{bmatrix} \sigma_1^2 & \bar{\rho}\sigma_1\sigma_2 \\ \bar{\rho}\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix} \quad (8)$$

With this values obtained from either the Lyapunov equation or spectral analysis of the frequency response, the EASA-CS25 defines 8 points on the ellipse of the loads around a static equilibrium with load $\mathbf{P}_g = [P_{g1}, P_{g2}]$ and a predefined level of design gust intensity u_σ :

$$\mathbf{P} = \begin{bmatrix} P_{g1} & P_{g1} \\ P_{g2} & P_{g2} \end{bmatrix} \pm \begin{bmatrix} \sigma_2 u_\sigma & 0 \\ 0 & \sigma_1 u_\sigma \end{bmatrix} \begin{bmatrix} ((1 - \bar{\rho})/2)^{0.5} & ((1 + \bar{\rho})/2)^{0.5} \\ ((1 - \bar{\rho})/2)^{0.5} & ((1 + \bar{\rho})/2)^{0.5} \end{bmatrix} \quad (9)$$

Therefore from the state-space system outlined in Fig. 2 both critical dynamic loads and flutter stability boundaries can be assessed. Reduction of this system for efficiency will be the subject of the next section.

Algorithm 1: Flutter calculation

input : aeroelastic state-space system, $ss_aeroelastic$, step to find flutter interval, h , initialize velocity, u_0 , eigenvalue solver, EIG , tolerance factor, ϵ_u , damping interval around 0, ϵ_damp .

begin

$u_old, u_new \leftarrow u_0$ // Search interval where pre and post flutter occurs

while $damping_old * damping_new > 0$. **do**

$ss_aeroelastic = ss_aeroelastic.update(u_new)$

$eigs, eigenvectors = EIG(ss_aeroelastic.A)$

$damping_old = damping_new$

$damping_new = \max(eigs.real/abs(eigs))$

if $damping_new > \epsilon_damp$ **then**

$u_old \leftarrow u_new$

$u_new \leftarrow u_new - h$

else if $damping_new < \epsilon_damp$ **then**

$u_old \leftarrow u_new$

$u_new \leftarrow u_new + h$

else

$u_old \leftarrow u_new$

break

while $abs(u_new - u_old) > \epsilon_u$ // Apply secant method

do

$ddamping = (damping_new - damping_old)/(u_new - u_old)$

$du = -damping_old/ddamping$

$u_secant = u_old + du$

$ss_aeroelastic = ss_aeroelastic.update(u_secant)$

$eigs, eigenvectors = EIG(ss_aeroelastic.A)$

$damping_secant = \max(eigs.real/abs(eigs))$

if $damping_secant > \epsilon_damp$ **then**

if $damping_new > 0$. **then**

$u_new = u_secant$

$damping_new = damping_secant$

else if $damping_old > 0$. **then**

$u_old = u_secant$

$damping_old = damping_secant$

else if $damping_secant < \epsilon_damp$ **then**

if $damping_new < 0$. **then**

$u_new = u_secant$

$damping_new = damping_secant$

else if $damping_old < 0$. **then**

$u_old = u_secant$

$damping_old = damping_secant$

else

$u_new = u_secant$

break

$u_flutter = u_new$

III. Parametric model order reduction techniques for unsteady aerodynamics

Even in its linearised form, the dynamic aeroelastic model presented above is expensive and impractical for design applications. The large UVLM matrices can be reduced using methods of model reduction for dynamical systems. An important consideration is the highly sparse nature of these matrices, which will allow a considerable level of reduction and possibly even an affine representation of the parametric dependence. The first step in the construction of the parametric library is to select a representative space of the design variables as is highlighted in the next section.

A. Selection of a rich parametric design space

The first challenge to build a library of PMORs is to select a subspace of parameters that is a good representation of the bounded full domain, also known as Design of Experiments (DoE). Structured grids provide a good solution for low dimensional spaces. However, the number of points grow very fast with the number of dimensions and they become infeasibly large very quickly. Unstructured DoE can be used to reduce the total number points while maintaining a reasonably good distribution, and thus are a better option when dealing with moderately large dimensional spaces ($N \approx 10$). For the generation of unstructured distributions, Latin Hyper-Cube (LHC) techniques are a good compromise between the number of sample points and their distribution. Yet the standard implementation of this approach, which distributes the points randomly, can lead to undesirable cluster of points that leaves significant empty areas within the domain. Optimal Latin Hyper-Cube (OLHC) designs tackle this issue by searching an optimal uniform grid of points. A permutation Genetic Algorithm (PermGA) is used herein as in [18]. It treats each design point as a negative charge and optimises to minimise the overall energy potential, U , as a function of the distance between points $d()$:

$$U = \sum_{i=1}^{N_p} \sum_{j=i+1}^{N_p} \frac{1}{d(x_i, x_j)} \quad (10)$$

As the dimensions grow, the effect of the optimization becomes more important since the likelihood of subspaces without points also increases. It is also harder to visualize this as shown in representation for a 3D space given in Fig. 3 with the convex-hull of the points plotted in orange solid lines.

B. Projection-based balancing methods

After identification of a representative parametric space, the next step is to reduce the aerodynamic linearised system, which will be attained using projection-based techniques of model reduction. Given a general linear system dependent of $\mathbf{p}$ parameters with $\mathbf{x}$ states, $\mathbf{u}$ inputs and $\mathbf{y}$ outputs, the dynamics in continuous time are described as,

$$\begin{aligned} \mathbf{E}(\mathbf{p})\dot{\mathbf{x}}(\mathbf{p}, t) &= \mathbf{A}(\mathbf{p})\mathbf{x}(\mathbf{p}, t) + \mathbf{B}(\mathbf{p})\mathbf{u}(\mathbf{p}, t) \\ \mathbf{y}(\mathbf{p}, t) &= \mathbf{C}(\mathbf{p})\mathbf{x}(\mathbf{p}, t) + \mathbf{D}(\mathbf{p})\mathbf{u}(\mathbf{p}, t) \end{aligned} \quad (11)$$

Fig. 3 Design of experiments in a 3D space with a total of 70 points

Projection methods seek to find a set of basis T_v and T_w such as the states are projected $\mathbf{x} = T_v \hat{\mathbf{x}}$, and the first equation in (11) is multiplied by $T_w^\top$. In the UVLM linear system E is the identity matrix and $T_w = T_v^{-1}$, though we keep T_w in the notation. The reduced linear system is represented by $\hat{\Sigma}$ (dependency on $\mathbf{p}$ is dropped for convenience):

$$\hat{\Sigma} := \left[\begin{array}{c|c} \hat{A} & \hat{B} \\ \hline \hat{C} & \hat{D} \end{array} \right] = \left[\begin{array}{c|c} T_w^\top A T_v & T_w^\top B \\ \hline C T_v & D \end{array} \right] \quad (12)$$

A common approach to find T_v is to solve for the system Grammians, $\mathcal{W}_c$ and $\mathcal{W}_o$, using the Lyapunov equation,

$$\begin{aligned} A \mathcal{W}_c A^\top - \mathcal{W}_c &= -B B^\top \\ A^\top \mathcal{W}_o A - \mathcal{W}_o &= -C^\top C \end{aligned} \quad (13)$$

followed by a balancing of the system which consists of finding T_v and T_w such that

$$\bar{\mathcal{W}}_c = T_w \mathcal{W}_c T_w^\top = \Sigma = T_v^\top \mathcal{W}_o T_v = \bar{\mathcal{W}}_o \quad (14)$$

The solution is attained starting from the Cholesky factorization of the Gramians,

$$\mathcal{W}_c = Z_c Z_c^\top \quad \text{and} \quad \mathcal{W}_o = Z_o Z_o^\top \quad (15)$$

then performing a Singular Value Decomposition (SVD), $Z_o^\top Z_c = U \Sigma V^\top$, with $\Sigma = \text{diag}(\sigma_i)$ and σ the Hankell Singular Values (HSV), and U and V unitary matrices containing the left and right singular vectors.

$$\mathbf{T}_v = \mathbf{Z}_c \mathbf{V} \mathbf{\Sigma}^{-1/2} \quad \text{and} \quad \mathbf{T}_w = \mathbf{\Sigma}^{-1/2} \mathbf{U}^\top \mathbf{Z}_o^\top \quad (16)$$

Up to this, $\mathbf{T}_v$ and $\mathbf{T}_w$ do not reduce the system but provide a balanced realization that facilitates the construction of the ROMs by truncation of the smallest HSV and subsequent removal of the columns in $\mathbf{U}$ and $\mathbf{V}$, i.e. discarding the less observable/controllable states, or by residualisation of these states.

Solving the Lyapunov equation, Eq. (13) can become too expensive, so an alternative is derived in [4] which consists of finding the frequency-limited Gramians by integration among reduced frequencies, k , the dynamics of the linear system,

$$\mathbf{W}_c = \frac{1}{2k_N} \int_{-k_c}^{k_c} \mathbf{\Phi}_c(k) \mathbf{\Phi}_c^*(k) dk \quad ; \quad \mathbf{W}_o = \frac{1}{2k_N} \int_{-k_o}^{k_o} \mathbf{\Phi}_o(k) \mathbf{\Phi}_o^*(k) dk \quad (17)$$

with $()^*$ the complex conjugate and $\mathbf{\Phi}_c(k) = (e^{jk\Delta t} \mathbf{I}_{N_x} - \mathbf{A})^{-1} \mathbf{B}$, $\mathbf{\Phi}_o(k) = ((e^{jk\Delta t} \mathbf{I}_{N_x} - \mathbf{A})^{-1})^* \mathbf{C}^\top$. Frequency-limited balanced truncation (FLBT) outperforms internal balancing in the low-frequency range, and it allows a novel interpolation among parametric transfer functions as described next.

C. Interpolation and approximation of reduced order models

The reduction of the linear dynamical system speeds-up and facilitates the aeroelastic calculations but the process itself is costly and therefore may not be suitable for design applications. A good compromise can be achieved by interpolating or approximating the projected ROMs, thereby building parametric ROMs, which is a subject of ongoing research [19]. The most straight forward technique for building such a parametric state-space models is using standard interpolation methods. For structured grids, Lagrange polynomial techniques or similar offer a relatively simple and accurate solution whereby firstly the interval on which the particular evaluation point lies is determined, and then the interpolation of the required quantity is carried out using only the points within that interval (a structured 2D grid is shown in Fig. 4 with the limiting interval given by the red points and the added white points increasing the order of the interpolation). Let $\mathbf{Y}$ be a tensor of any order that has been evaluated in a number N_i of interpolation points as a function of the vector variable $\mathbf{p}$ with dimension N_d . The matrix $\mathbf{P} \in \mathbb{R}^{N_d \times N_i}$ is the concatenation of $\mathbf{p}_i$ in columns, and $\mathbf{P}^*$ the non-repeating coordinate of the points of $\mathbf{P}$ across dimensions, which defines the order of the polynomial for each of the dimensions, N_o . ϵ is defined as,

$$\epsilon_{ijk} = \begin{cases} 0 & \text{if } P_{ki} = P_{kj}^* \\ 1 & \text{if } P_{ki} \neq P_{kj}^* \end{cases} \quad (18)$$

Fig. 4 2D structured grid with 8 interpolation points

and the continuous expression for Y within the interpolation interval is,

$$Y(\mathbf{p}) = \sum_{i=1}^{N_i} Y_i \prod_{k=1}^{N_d} \prod_{j=1}^{N_o(k)} \left(\frac{p_k - P_{kj}^*}{P_{ki} - P_{kj}^*} \right)^{\epsilon_{ijk}} = \sum_{i=1}^{N_i} L_i(\mathbf{p}) Y_i \quad (19)$$

For instance, in Fig. 4 there are eight interpolation points, $N_i = 8$, two dimensions, $N_d = 2$, three distinct points in each dimension, $N_o = [3, 3]$, and $\mathbf{P}^*$ can be written as, $\mathbf{P}^* = \begin{pmatrix} P_{11} & P_{12} & P_{13} \\ P_{21} & P_{28} & P_{27} \end{pmatrix}$ (that is, P_{11} and P_{13} are the x -coordinate of points 1 and 3, P_{27} the y -coordinate of point 7 and so on). The parametric dependence is thus encapsulated in the functions $L(\mathbf{p})$, in this case polynomial of order N_o for each dimension of $\mathbf{p}$. These $L(\mathbf{p})$ functions conform the the kernel of the approximation of Y . In high dimensional spaces, however, structure grids become infeasible due to the exponential growth of the number of sampling points with number of dimensions. Focusing now on unstructured grids, interpolation via radial basis functions represent a good alternative for the interpolation. Given the continuous functional Y that we want to approximate,

$$Y(\mathbf{p}) = \sum_{i=1}^{N_i} \varphi_i(\mathbf{p} - \mathbf{p}_i) \Xi_i \quad (20)$$

with φ_i the radial basis function for each point i in the evaluated design space. The tensor weights, Ξ_i , are found solving the linear system,

$$\begin{bmatrix} \varphi_{11} & \dots & \varphi_{1n_i} \\ \vdots & \ddots & \vdots \\ \varphi_{n_i 1} & \dots & \varphi_{n_i n_i} \end{bmatrix} \begin{bmatrix} \text{row}(\Xi_1) \\ \vdots \\ \text{row}(\Xi_{N_i}) \end{bmatrix} = \begin{bmatrix} \text{row}(Y_1) \\ \vdots \\ \text{row}(Y_{N_i}) \end{bmatrix} \quad (21)$$

with $\varphi_{n_i n_i} = \varphi_i(\mathbf{p}_j - \mathbf{p}_i)$ the elements of a matrix Φ (whose inverse is guaranteed via the properties of these functions [20]); and $\text{row}()$ concatenates every element of the input tensor into a single row in a consistent order (for matrices

Fortran or C ordering are the two options). The explicit dependence on the calculated values Y_j is

$$Y(\mathbf{p}) = \sum_{i=1}^{N_i} \varphi_i(\mathbf{p} - \mathbf{p}_i) \sum_{j=1}^{N_i} \Phi_{ij}^{-1} Y_j = \sum_{i=1}^{N_i} \sum_{j=1}^{N_i} \varphi_i(\mathbf{p} - \mathbf{p}_i) \Phi_{ij}^{-1} Y_j = \sum_{j=1}^{N_i} L_j(\mathbf{p}) Y_j \quad (22)$$

Here the kernel functions in $\mathbf{L}$ providing the parametric dependence are linear combination of the radial basis functions of choice. Note that the interpolation in Eqs. (19)-(21) is general for any dimension of $\mathbf{p}$. Having identified the functions $\mathbf{L}(\mathbf{p})$ either from Lagrangian polynomials or radial basis functions, a relevant question arises: How can these functions be optimal to approximate the aimed quantity? An optimization problem can be defined by adding a new set of training points, $\mathbf{p}^t$, and minimising an error metric of $Y(\mathbf{p}^t)$ with respect to the initially constant parameters in the kernel $\mathbf{L}$. That is, training the functionals in $\mathbf{L}$ starting from the values provided by the interpolation, which could provide relevant improvements when the distance between points is not very small. For instance, in the interpolation the following radial bases functions have been considered:

$$\begin{aligned} \text{multiquadric} &\rightarrow a((\|\mathbf{p} - \mathbf{p}_i\|/b)^2 + c)^{0.5} \\ \text{inverse} &\rightarrow a((\|\mathbf{p} - \mathbf{p}_i\|/b)^2 + c)^{-0.5} \\ \text{gaussian} &\rightarrow ae^{-(\|\mathbf{p} - \mathbf{p}_i\|/b + c)^2} \end{aligned} \quad (23)$$

For the interpolation problem, a equals one, b is a constant parameter dependent on the distributions of points, and c is 0 for the Gaussian and one for the other two. For example, for an optimization of the Gaussian kernel functions of number N_i and a number N_t of training points, Eq. (21) becomes,

$$\begin{bmatrix} a_1 e^{-(\|\mathbf{p}_1^t - \mathbf{p}_1\|/b_1 + c_1)^2} & \dots & a_{N_i} e^{-(\|\mathbf{p}_1^t - \mathbf{p}_{N_i}\|/b_{N_i} + c_{N_i})^2} \\ \vdots & \ddots & \vdots \\ a_1 e^{-(\|\mathbf{p}_{N_t}^t - \mathbf{p}_1\|/b_1 + c_1)^2} & \dots & a_{N_i} e^{-(\|\mathbf{p}_{N_t}^t - \mathbf{p}_{N_i}\|/b_{N_i} + c_{N_i})^2} \end{bmatrix} \begin{bmatrix} \text{row}(\mathbf{\Xi}_1) \\ \vdots \\ \text{row}(\mathbf{\Xi}_{N_i}) \end{bmatrix} = \begin{bmatrix} \text{row}(\mathbf{Y}_1) \\ \vdots \\ \text{row}(\mathbf{Y}_{N_t}) \end{bmatrix} \quad (24)$$

with $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^{N_i}$ the variables to optimize based on an error metric of Y at the training points.

Choosing the actual quantity on which the parametric dependence is built will be also important and three options are outlined next.

1. Gramians factors interpolation

In this method the Gramians factors, $\mathbf{Z}_c$ and $\mathbf{Z}_o$ in Eq. (15), are the quantities interpolated:

$$\mathbf{Z}_{(c,o)}(\mathbf{p}) = \sum_{i=1}^{N_i} L_i(\mathbf{p}) \mathbf{Z}_{(c,o)i} \quad (25)$$

The values of $\mathbf{Z}_{(c,o)i} = \mathbf{Z}_{(c,o)}(\mathbf{p}_i)$ are obtained by numerically solving the integral in Eq. (17) [21]. Applying the corresponding SVD to $\mathbf{Z}_o^\top(\mathbf{p})\mathbf{Z}_c(\mathbf{p})$, the matrices $\mathbf{V}$, $\mathbf{\Sigma}$ and $\mathbf{U}$ are obtained and the projection matrices that balance the system are calculated as in Eq. (16). And the reduced order model is obtained from $\mathbf{T}_v$ after selection of the number of HSV in $\mathbf{\Sigma}$ to be kept in the approximation.

2. Transfer function interpolation

The transfer function of the system Σ is,

$$\mathbf{G}(s) = \mathbf{C}(s - \mathbf{A})^{-1} - \mathbf{B} \quad (26)$$

an interpolation of $\mathbf{G}$ is carried out,

$$\mathbf{G}(\mathbf{p}) = \sum_{i=1}^{N_i} L_i(\mathbf{p})\mathbf{G}_i \quad (27)$$

and a balanced realization of this system is simply obtained concatenating the states, $\mathbf{x} = \text{col}(\mathbf{x}_i)$ [22]:

$$\bar{\Sigma} := \left[\begin{array}{c|c} \text{diag}(\mathbf{A}_i) & \text{col}(\mathbf{B}_i) \\ \hline \text{row}(L(\mathbf{p}_i)\mathbf{C}_i) & \sum_{i=1}^{N_i} L(\mathbf{p}_i)\mathbf{D}_i \end{array} \right] \quad (28)$$

As it is, this system can grow very large, especially if working with interpolation of radial basis functions which consider a large data set of points. A smaller system is attained if the square root factors of the Gramians of $\mathbf{G}$ are put as in [21],

$$\begin{aligned} \mathbf{Z}_c &= \text{col}(\mathbf{T}_{wi}^\top \mathbf{Z}_{ci}) = \text{col}(\mathbf{\Sigma}_i^{1/2} \mathbf{V}_i^\top) \\ \mathbf{Z}_o^\top(\mathbf{p}) &= \text{row}((L_i(\mathbf{p})\mathbf{Z}_{oi}^\top \mathbf{T}_{vi}) = \text{row}(L_i(\mathbf{p})\mathbf{U}_i \mathbf{\Sigma}_i^{1/2}) = \end{aligned} \quad (29)$$

and,

$$\mathbf{Z}_o^\top \mathbf{Z}_c(\mathbf{p}) = \sum_{i=1}^{N_i} L_i(\mathbf{p})\mathbf{U}_i \mathbf{\Sigma}_i \mathbf{V}_i^\top = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^\top \quad (30)$$

with the resulting the projection matrices being

$$\mathbf{T}_v = \text{col} \left(\mathbf{\Sigma}_i^{1/2} \mathbf{V}_i^\top \mathbf{V} \mathbf{\Sigma}^{-1/2} \right) \quad \text{and} \quad \mathbf{T}_w = \text{row} \left(L_i(\mathbf{p})\mathbf{U}_i \mathbf{\Sigma}_i^{1/2} \mathbf{\Sigma}^{-1/2} \mathbf{U}^\top \right) \quad (31)$$

and when applied to the realization in Eq. (28), the new system has the same size as the original one.

3. Affine matrix representation

An more straight forward reduction of the dynamical system Σ can be achieved if the full system matrices can be put in an affine representation, that is, as a (decoupled) sum of functions of the parametric space multiplied by a set of basis:

$$\mathbf{A}(\mathbf{p}) = \mathbf{A}_0 + \sum_{i=1}^{N_i} f_i(\mathbf{p}) \mathbf{A}_i \quad (32)$$

The success of the approach relies on finding a good approximation in a relatively low number of terms in the sum, N_i . Similar expressions are derived for the other matrices in the dynamical system. A global basis approach has been proposed [22] to reduce the full-state by concatenation of the local basis at the parameter samples, $[\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_{N_i}]$, such that $\mathbf{T}_v = [\mathbf{T}_{v1}, \mathbf{T}_{v2}, \dots, \mathbf{T}_{vN_i}]$ and $\mathbf{T}_w = [\mathbf{T}_{w1}, \mathbf{T}_{w2}, \dots, \mathbf{T}_{wN_i}]$. It is likely that the local matrices have common components so an SVD is carried out on these expressions, followed by a truncation to obtain the final projection matrices, $\hat{\mathbf{T}}_v$ and $\hat{\mathbf{T}}_w$, with the resulting projection as,

$$\mathbf{T}_w^\top \mathbf{A}(\mathbf{p}) \mathbf{T}_v = \mathbf{T}_w^\top \mathbf{A}_0 \mathbf{T}_v + \sum_{i=1}^{N_i} f_i(\mathbf{p}) \mathbf{T}_w^\top \mathbf{A}_i \mathbf{T}_v \quad (33)$$

the reduced matrix $\hat{\mathbf{A}}$ is,

$$\hat{\mathbf{A}}(\mathbf{p}) = \hat{\mathbf{A}}_0 + \sum_{i=1}^{N_i} f_i(\mathbf{p}) \hat{\mathbf{A}}_i \quad (34)$$

There are two important points to consider with this approach: on the one hand the UVLM matrices are very large which can hinder the efforts to find a good parametric dependence. On the other, the high sparsity of these matrices indicates it could be a promising approach. Furthermore, the resulting reduced model will be extremely fast to evaluate. Interpolation techniques as presented above are the starting point, thus considering $f_i(\mathbf{p}) = L_i(\mathbf{p})$. This functions can then be trained on a new set of points as to improve the approximation.

IV. Results

A very flexible strut-braced wing as the one shown in Fig. 5 will be used to asses the effectiveness of the optimal LHC design of experiments and the performance of the presented PMORs techniques, and to optimize the design under flutter constraints. Table 1 shows the properties of the model. $\sigma(y)$ is a function of the wing span initially set to one and use in sec. IV.B to modify the stiffness and mass properties of the wing. The model has been built drawing inspiration from HALE type of aircraft [23] that undergo extremely large deflections, and on which UHARW aircraft may take advantage from, considering the variety of examples and tools developed in that area of aeronautics. The model however does not represent any real or close to real design and serves to define a challenging mathematical optimization problem, study the parametric ROMs and test the strut element which modifies the wing properties and the aerodynamic of the wake.

Fig. 5 Optimised strut-braced wing

Table 1 Reference strut-braced wing model

Wing root chord, $chord1$		1.0 m
Initial Wing tip chord $chord2_0$		1.0 m
Initial wing span, b_{y0}		13 m
Wing sweep		0°
Angle between wing and strut		30°
Strut attachment location		1/3 wing span
Axial stiffness	EA	$\sigma(y)^2 \times 1 \times 10^6 \text{ Pa m}^2$
Out-of-plane shear stiffness	GA_y	$\sigma(y)^2 \times 1 \times 10^6 \text{ Pa m}^2$
In-plane shear stiffness	GA_z	$\sigma(y)^2 \times 1 \times 10^6 \text{ Pa m}^2$
Torsional stiffness	GJ	$\sigma(y)^2 \times 4.24 \times 10^5 \text{ N m}^2$
Out-of-plane bending stiffness	EI_{yy}	$\sigma(y)^2 \times 3.84 \times 10^5 \text{ Pa m}^4$
In-plane bending stiffness	EI_{yy}	$\sigma(y)^2 \times 2.46 \times 10^7 \text{ Pa m}^4$
Elastic axis	$x_{ea}/chord1$	0.25
Centre of gravity line	$x_{cg}/chord1$	0.25
Mass per unit length	ρ	$\sigma(y) \times 0.75 \text{ kg/m}$
Inertia per unit length	$[I_{xx}, I_{yy}, I_{zz}]$	$\sigma \times [0.1, 0.05, 0.05] \text{ kg} \cdot \text{m}$

A. Reduced order models of unsteady aerodynamics

In this section we show the accuracy of limited-frequency balancing, the effectiveness of optimal LHC designs, the precision of the interpolated parametric ROMs and the improvements from the optimization in the kernel functions. The aerodynamic design will change with two variables: the tip-chord, $chord2 = \hat{c} \times 1 \text{ m}$ (with a linear variation from the root to the tip along the span), and the wing span, $b_y = \hat{b} \times 13 \text{ m}$.

1. Frequency-limited balancing

Frequency limited balancing is computationally efficient and can attain large reductions of the UVLM system. Here we consider 15 designs of the strut-braced wing with varying tip-chord and varying span. The points are the black circles shown in Fig. 7, scaled to have a factor of the span $\hat{b} \in [1.1, 1.43]$ and of the tip-chord $\hat{c} \in [0.4, 0.8]$. We reduce the aerodynamic system with 532 states to 25 states and the error across frequencies and design points remains negligible as shown in Fig. 6. The error metric chosen, $\epsilon(\mathbf{G}_f, \mathbf{G}_r) = \|\mathbf{G}_f - \mathbf{G}_r\| / \|\mathbf{G}_f\|$, is the relative error in the Frobenius norm between transfer function matrices of the full and reduced systems, $\mathbf{G}_f$ and $\mathbf{G}_r$ respectively, Eq. (26).

Fig. 6 Frequency-limited balancing error across LHC design points

2. Optimal DoE parametric ROMs

The selection of the points where the bases of the PMOR library is built has an important influence in the accuracy of the ROMs. Figure 7 shows the comparison between optimal and regular LHC designs and the fitness evolution in the genetic algorithm producing the OLHC with 65 points is shown in Fig. 8. The blue star points will be used to build the interpolation with 25 state-space aerodynamic reduced models. Unlike the previous example where the comparison was made between between calculated full and reduced systems, here the comparison is made using the interpolation of the reduced state-space models. Thus the aerodynamic ROM is interpolated at the testing points (circle black points in the figure) for the three radial bases functions described above, and using 20, 30 and 65 points in the interpolatory model. The transfer function interpolation in Eq. (31) is the one implemented in this exercise. The error metric used before is employed, averaging across the testing points. The OLHC outperforms the regular LHC in all instances, the multiquadric functions achieve the best performance and adding more points to the model is generally good but does not

Fig. 7 Optimal versus regular Latin Hypercube DoE: blue star points represent the model building points and black circles (same in all plots) are testing points

Fig. 8 Generation of OLHC in 2D with 65 points

guarantee improvement, also as this depends on the current selection of testing points. Errors below 10^{-2} are deemed a good approximation of the system dynamics, since even though only two parameters are being considered, very large variations across variables have been set (100% and 33% for the tip-chord and span variables respectively).

Fig. 9 Error in interpolated system with number of points in the model, radial bases functions and type of LHC

3. Approximation of optimal kernels for parametric ROMs

The functions that encapsulated the parametric ROMs dependence on the previous exercise had a set of constant parameters across all points. We have aimed to train those parameters: 20 model points are used for the functions kernels, three parameters per function yield a total 60 optimization variables; 65 training points are chosen for the optimization (using Scipy.optimize 'Nelder-Mead' algorithm), and the final model is tested on the same 15 points utilized in the previous exercise. Fig. 10 shows these points. There is an improvement in the error metric for the three kernel functions when compared to pure interpolation but this is not very large: 2.7% using the multiquadratic functions, 3.1% with the Gaussian and 4.4% with the inverse functions. Moreover the maximum error across points ($\max(\epsilon(G_f, G_r))$) is more significantly reduced by 6.13% and 7.8% in the inverse and Gaussian functions respectively, and negligible in the multiquadric approximation. The gains in accuracy are not very large, however, there are important points to consider: a) the method is general and allows for the addition of other type of kernel functions in combination

with the ones already used, such as polynomial functions; b) evaluation of the optimization function is fairly fast and global optimization algorithms may provide better solutions directly; c) the number of training points could increase by many and not pose a problem to the optimization –the number of model points, which defines the number of optimization variables, is what cannot increase dramatically.

Fig. 10 Model, training, and testing points in the optimization of the kernel functions building the parametric ROMs

The optimized parameters at each model point are depicted in Fig. 11, in parentheses are the initial values for the optimization and that were used for building the parametric model via interpolation.

Fig. 11 Optimization of kernel function parameters

B. Optimization of a strut-braced UHARW under flutter constraints

In order to apply the methods presented above to an aeroelastic design optimization problem, SHARPy has been couple with CPACs [24], an xml-based language for aircraft models definition, and GEMSEO [9, 25], a multidisciplinary optimization software. The approach is based on the definition of aircraft components through variables (fixed or varying in the optimization) in the form of *component_name*type-variable_name*, then an assembler builds the SHARPy model following an input connectivity. The process workflow is shown in Fig. 12, where the dash arrow marks the start of the optimization from the model definition in CPACs, then a component architecture builds the "SHARPy-ready" model, the solid arrows indicate the optimization loop where GEMSEO acts as the engine and SHARPy solves for the physics, interchanging design variables and objectives and constraints respectively, the final optimize model is then build back to

a CPACs model as represented by the dotted line.

Fig. 12 Nonlinear aeroelastic design optimization workflow

With this tool validated and running, we set out to define an aeroelastic optimization problem using the strut-braced wing Fig. 5 and the properties in Table 1. Note that this problem is posed as a mathematical challenge for the solvers and the coupling of tools rather than an actual design exercise. The model is defined to undergo very large deformations, design variables are allowed to change in a range of more than 100% and so do the objectives and constraints vary during the optimization as a result. Robustness of GEMSEO to orchestrate the optimization problem and the strength of SHARPy in calculating both nonlinear static and dynamic aeroelastic problems are demonstrated. Thus clearing the path for design exercises on more conservative, yet more realist too, configurations.

The operating conditions are a flying speed of $u_\infty = 40$ m/s, air density $\rho_\infty = 0.2$ Kg/m³ and the optimization is set as,

$$\begin{aligned}
 \min f(chord2, \mathbf{p}) &= \text{total_wing_mass} * chord2^{3.5} \\
 \text{s.t.} \quad u_{flutter} &> 50 \text{ m/s} \\
 b_y &= \frac{2 \times chord1}{chord1 + chord2} b_{y0}
 \end{aligned} \tag{35}$$

The design variables are the chord at the tip, $chord2$, which also controls the wing span, b_y , increasing the latter when decreasing the former as to keep the initial reference area constant; and a vector $\mathbf{p}$ of dimension three for the monitoring stations at root, strut-wing contact and tip of the wing, it represents a polynomial characterization along the span of the wing given by function σ in Table 1 that changes the structural and inertia properties, that is, σ will a quadratic polynomial equal to $\mathbf{p}$ at those points. While the objective is a function of the total mass of the wing, the introduction of

the *chord2* variable in the objective of minimizing the is an artifact to reward wings with higher aspect ratio and create a conflict with the flutter constraint as deformations grow large. An SLSQP algorithm is used for the optimization with 10^{-5} tolerances. Table 2 presents the results of the optimization for various angles of attack, with bounds in the design variables in brackets, set after a range of convergence exercises. The nonlinear static equilibrium of the optimized

Table 2 Optimization results of a highly flexible strut-braced wing

AoA	Design variables				Objective value	Flutter constraint (m/s)
	<i>chord2</i>	p1	p2	p3		
0°	0.496 [0.3, 1.]	0.4 [0.4, 1.3]	0.4 [0.4, 1.3]	0.265 [0.2, 1.]	0.559	50
1°	0.487 [0.4, 1.]	0.93 [0.6, 1.3]	0.605 [0.6, 1.3]	0.351 [0.35, 1.]	0.81	52.24
2°	0.2 [0.2, 1.]	0.687 [0.6, 1.3]	0.6 [0.6, 1.3]	0.4 [0.4, 1.]	0.044	70.15
3.5°	0.35 [0.35, 1.]	0.403 [0.4, 1.3]	0.729 [0.4, 1.3]	0.305 [0.3, 1.]	0.305	61.01

designs is shown in Fig. 13. The high flexibility of the more is clearly seen: at 0° AoA the wing deforms downwards under its own wing; at 2° AoA the wing slightly bends upwards and it is the point where the wing gets larger in span an optimal for the objective function. At 4° AoA very large deflections take place in the equilibrium, but the optimizer does not meet the flutter constraint as seen next.

Fig. 13 Optimized strut-braced wing equilibrium condition at different AoA

Fig. 14-Fig. 16 shows both the objective function and design variables evolution in the optimization. The flutter constraint is presented in a color map where feasible solutions are given as green, violation of the flutter constraint as red and white is the neutral point, something clearly visible at the 0° AoA simulation in Fig. 14, where the flutter velocity calculated is very close to 50 m/s.

(a) Objective evolution and flutter constraint

Fig. 14 Strut-braced wing optimization 0° AoA

At 1° AoA in Fig. 15 the convergence is not definitive but the the best feasible solution is picked.

(a) Objective evolution and flutter constraint

(b) Design variables evolution

Fig. 15 Strut-braced wing optimization 1^o AoA

The best solution across AoA is shown in Fig. 16. A converged solution is attained after a succession of spikes in objective function evolution.

(a) Objective evolution and flutter constraint

(b) Design variables evolution

Fig. 16 Strut-braced wing optimization 2^o AoA

Finally in Fig. 17 the evolution of the objective function shows that flutter is violated despite starting from a feasible solution, which raises the task for the optimizer to deal better with the constraints when highly nonlinear deformations do occur in the design search.

Fig. 17 Strut-braced wing optimization 4^o AoA, objective evolution and failure to meet the flutter constraint

V. Conclusions

Design of new aircraft requires a more multidisciplinary approach which also brings cross-coupling design variables among disciplines. We have proposed an aeroelastic solution that incorporates dynamic solutions around a nonlinear equilibrium that enables the calculation of flutter stability boundaries and turbulence loads.

We also aim at reducing computational times by building a parametric ROM of the aerodynamics in a framework that allows fast computations while accounting for arbitrary geometrical deformations in the trim configuration. Balancing methods allow an efficient construction of the reduced model and the use of a novel approach to build parametric reduced order models of large state-space aerodynamic systems. Furthermore, the role of optimization in this work has been three-fold: to build optimal LHC design of experiments; to improve the kernels used in the approximation of the parametric state space systems; to explore the design space in a very flexible strut-braced wing. The results have shown that the coupled framework between the proposed nonlinear solution and a multidisciplinary software works well in a very challenging problem that undergoes large deflections –even under its own weight. The next steps of this work will include testing the PMOR techniques presented on higher dimensional spaces; moving discrete masses for flutter suppression in the optimization and including dynamic loads as the constraints of the optimization; and using a representative model of a strut-braced wing aircraft to build an MDO problem with other tools that incorporate, among others, higher fidelity aerodynamics corrections to the formulation utilized herein based on potential flow.

Acknowledgements

This project has received funding from the Clean Sky 2 Joint Undertaking (JU) under grant agreement No 883670. The JU receives support from the European Union's Horizon 2020 research and innovation programme and the Clean Sky 2 JU members other than the Union

References

- [1] Cesnik, C. E. S., Palacios, R., and Reichenbach, E. Y., "Re-examined structural design procedures for very flexible aircraft," *Journal of Aircraft*, Vol. 51, No. 5, 2014, pp. 1581–1591. doi:10.2514/1.C032464.
- [2] Kier, T. M., "An Integrated Model for Lateral Gust Loads Analysis and Dutch Roll Flight Dynamics Using a 3d Panel Method," *International Forum on Aeroelasticity and Structural Dynamics*, 2017. doi:10.7150/jca.24838.
- [3] Ripepi, M., Verveld, M. J., Karcher, N. W., Franz, T., Abu-Zurayk, M., Görtz, S., and Kier, T. M., "Reduced-order models for aerodynamic applications, loads and MDO," *CEAS Aeronautical Journal*, Vol. 9, No. 1, 2018, pp. 171–193. doi:10.1007/s13272-018-0283-6.
- [4] 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. doi:10.2514/1.J058153.
- [5] Murua, J., Palacios, R., and Graham, J. M. R., "Assessment of wake-tail interference effects on the dynamics of flexible aircraft," *AIAA Journal*, Vol. 50, 2012, pp. 1575–1585. doi:10.2514/1.J051543.
- [6] Afonso, F., Vale, J., Oliveira, É., 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. doi:10.1016/j.paerosci.2016.12.004.
- [7] Cea, A., and Palacios, R., "Nonlinear Modal Aeroelastic Analysis from Large Industrial-Scale Models," *AIAA Scitech 2019 Forum*, American Institute of Aeronautics and Astronautics, 2019. doi:10.2514/6.2019-0208.
- [8] Medeiros, R. R., Cesnik, C. E. S., and Coetzee, E. B., "Computational Aeroelasticity Using Modal-Based Structural Nonlinear Analysis," *AIAA Journal*, 2019, pp. 1–10. doi:10.2514/1.j058593.
- [9] Gallard, F., Barjhoux, P.-J., Olivanti, R., and Gazaix, A., "GEMS, a Generic Engine for MDO Scenarios : Key Features in Application," *AIAA Aviation 2019 Forum*, American Institute of Aeronautics and Astronautics, 2019. doi:10.2514/6.2019-2991.
- [10] Géraudin, M., and Cardona, A., *Flexible multibody dynamics: A finite element approach*, John Wiley, Baffins Lane, Chichester, West Sussex, England, 2001.
- [11] Hesse, H., Palacios, R., and Murua, J., "Consistent Structural Linearization in Flexible Aircraft Dynamics with Large Rigid-Body Motion," *AIAA Journal*, Vol. 52, No. 3, 2014, pp. 528–538. doi:10.2514/1.J052316.
- [12] 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. doi:10.21105/joss.01885.

- [13] Palacios, R., Murua, J., and Cook, R., “Structural and aerodynamic models in nonlinear flight dynamics of very flexible aircraft,” *AIAA Journal*, Vol. 48, No. 11, 2010, pp. 2648–2659. doi:10.2514/1.J050513.
- [14] Goizueta, N., Wynn, A., and Palacios, R., “Parametric Krylov-based order reduction of aircraft aeroelastic models,” *AIAA Scitech Forum*, 2021. doi:10.2514/6.2021-1798.
- [15] Jonsson, E., Riso, C., Lupp, C. A., Cesnik, C. E. S., Martins, J. R. R. A., and Epureanu, B. I., “Flutter and post-flutter constraints in aircraft design optimization,” *Progress in Aerospace Sciences*, Vol. 109, 2019, p. 100537. doi:10.1016/j.paerosci.2019.04.001, URL <https://doi.org/10.1016/j.paerosci.2019.04.001>.
- [16] Goizueta, N., Drachinsky, A., Wynn, A., Raveh, D. E., and Palacios, R., “Flutter predictions for very flexible wing wind tunnel test,” *AIAA Scitech 2021 Forum*, American Institute of Aeronautics and Astronautics, 2021. doi:10.2514/6.2021-1711.
- [17] Engelsen, F., and Livne, E., “Mode Acceleration Based Random Gust Stresses in Aeroservoelastic Optimization,” *Journal of Aircraft*, Vol. 41, No. 2, 2004, pp. 335–347. doi:10.2514/1.9329.
- [18] Bates, S., Sienz, J., and Toropov, V., “Formulation of the Optimal Latin Hypercube Design of Experiments Using a Permutation Genetic Algorithm,” *45th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics Materials Conference*, 2004. doi:10.2514/6.2004-2011.
- [19] Swischuk, R., Mainini, L., Peherstorfer, B., and Willcox, K., “Projection-based model reduction: Formulations for physics-based machine learning,” *Computers and Fluids*, Vol. 179, 2019, pp. 704–717. doi:10.1016/j.compfluid.2018.07.021.
- [20] Fornefett, M., Rohr, K., and Stiehl, H. S., “Radial basis functions with compact support for elastic registration of medical images,” *Image and Vision Computing*, Vol. 19, No. 1-2, 2001, pp. 87–96. doi:10.1016/S0262-8856(00)00057-3.
- [21] 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. doi:10.2514/1.J058894.
- [22] Benner, P., Gugercin, S., and Willcox, K., “A survey of projection-based model reduction methods for parametric dynamical systems,” *SIAM Review*, Vol. 57, No. 4, 2015, pp. 483–531. doi:10.1137/130932715.
- [23] Murua, J., Palacios, R., Michael, J., and Graham, R., “Applications of the unsteady vortex-lattice method in aircraft aeroelasticity and flight dynamics,” *Progress in Aerospace Sciences*, Vol. 55, 2012, pp. 46–72. doi:10.1016/j.paerosci.2012.06.001.
- [24] Alder, M., Moerland, E., Jepsen, J., and Nagel, B., “Recent Advances in Establishing a Common Language for Aircraft Design with CPACS,” , No. 1, 2020, p. 14.
- [25] Gallard, F., Vanaret, C., Guenot, D., Gachelin, V., Lafage, R., Pauwels, B., Barjhoux, P.-J., and Gazaix, A., “GEMS: A Python Library for Automation of Multidisciplinary Design Optimization Process Generation,” *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, American Institute of Aeronautics and Astronautics, 2018. doi:10.2514/6.2018-0657.