Aeroservoelastic Optimisation of an Aerofoil with Active Compliant Flap via Reparametrisation and Variable Selection

Jacob J. Broughton-Venner*

Andrew Wynn[†] and Rafael Palacios[‡]

Imperial College, London, SW7 2AZ, United Kingdom

To aid in the investigation of new simultaneous optimisation strategies for flexible vehicles and their control systems, a two-dimensional aerofoil optimisation which demands minimal computational effort is studied. The aeroservoelastic system consists of a two-dimensional, potential flow over a deforming aerofoil; an actively controlled, but saturated compliant trailing edge; a dynamic observer that uses a series of pressure sensors on the aerofoil; and a heave/pitch linear spring model. Although computationally simple, the design allows for optimisation over multiple disciplines: the structure can be designed by varying the stiffness of the springs; the control architecture through weightings in a LQR controller; the observer by means of the placement of pressure sensors; and the aerodynamics via the shaping of the compliant trailing edge. Optimising the weight and a metric of performance over all these fields simultaneously is compared to a sequential methodology of optimising the open-loop characteristics first and subsequently adding a closed-loop controller. Parametrisation of the design vector and variable selection often require user input and are fixed during optimisation. Our research aims to automate this process. Furthermore, we investigate whether varying the parametrisation and number of design variables during the optimisation can lead to improvements in the final design. To accomplish this, a new basis for the design vector is created via Proper Orthogonal Decomposition (POD) using the trajectories of initial optimisation paths as a “training set”. This parametrisation is shown to make the optimisation more robust with respect to the initial design, and facilitate an automated variable selection methodology. This variable selection allows for the dimension of the problem to be reduced temporarily and it is shown that this makes the optimisation more robust.

Nomenclature

Roman

a_i	Coefficients in the shape of trailing edge
c	Non-linear constraints
f	Cost function
u	Control input
w	Gust profile
x	State vector
y	Output of the dynamics
A	Dynamic matrix
B_u	Control input matrix
B_w	Gust input matrix
K	Control matrix
$Q_{y,\alpha}$	LQR state penalties

*Ph.D Student, Department of Aeronautics, jacob.broughton-venner13@imperial.ac.uk, Student Member AIAA.

[†]Senior Lecturer, Department of Aeronautics.

[‡]Reader, Department of Aeronautics, AIAA Member.

V	Free-stream velocity
X	Design vector
Y	Heave degree of freedom
<i>Greek</i>	
α	Pitch degree of freedom
β	Trailing edge degree of freedom
ξ	Non-dimensional chordwise length
ξ_{ea}	Distance to elastic axis
ξ_h	Position of hinge point
ρ	Aerofoil density
ω	Natural frequency

I. Introduction

Aeroservoelastic (ASE) optimisation is concerned with the design of systems in which significant coupling is present between the aerodynamic and structural forces along with an active control system. Historically, control systems have been added to aeroelastic systems once the aeroelastic design has been fixed, which is typically referred to as a sequential design methodology. This is likely to be sub-optimal compared to the simultaneous optimisation of the entire design-space.¹ Nevertheless, the sequential approach may be sufficient when the coupling between the different components of the system is weak. However, such couplings are in general increasing with some proposed airframe concepts in comparison to traditional airframes, that were built with relatively stiff wings. For example, High-Altitude Long-Endurance concepts typically lead to flexible wings due to weight-power constraints. The increase in coupling complicates design since the first natural frequency of a traditional air frame compared to its flight dynamics will often be at least an order of magnitude greater. This allows the structural and flight dynamic design processes to be separated. Conversely, flexible wings which possess natural frequencies comparable to those of the flight dynamics lead to strong coupling characteristics,² in which a simultaneous design could be beneficial or even necessary for flight integrity. This paper will demonstrate the advantages and limitations of simultaneous design of coupled aeroservoelastic optimisation. Moreover, optimisation strategies will be introduced to investigate how said limitations can be attenuated.

A key difficulty is that even though a sequential design is expected to be sub-optimal, the increased design-space complexity associated with a simultaneous optimisation approach may itself make the optimisation problem intractable. For this reason, early studies in ASE optimisation often used simple models and convenient output-feedback control systems, which allowed simultaneous optimisation over plant and controller. Suzuki³ optimised the design of a cantilever wing by varying spar thickness and control gain, while constraining the system's open and closed loop stability, the spar stress and the control surface deflection angle; whereas Nam et al⁴ used a wide range of design variables (ply orientation of the composite layer, actuator placement and sizing) to minimise the root mean square of gust responses for various airspeeds. An early investigation in aeroelastic and control system optimisation for rotor flight⁵ compared sequential and simultaneous optimisation techniques, confirming that rotor design and control design are strongly coupled and benefit from simultaneous optimisation. Although in these cases a simultaneous plant/controller optimisation is employed, only relatively simple control feedback is synthesised, in which controllability and observability conditions are not considered. Furthermore, observer dynamics were not included in the optimisation, instead observer gains and sensor placement were chosen prior to the design. It was not investigated if adding complexity in these models would still allow a simultaneous approach to be advantageous.

An attempt to increase controller complexity included the modelling of parameter uncertainty, as considered by Moulin et al.⁶ This incorporates modern methods of robust MIMO control synthesis into a multidisciplinary design optimisation problem. However, this comes at the sacrifice of having to use a simpler sequential optimiser. A recent attempt to include more advanced control design⁷ used the internal weighting matrices of an LQR problem as design variables for a linear, flexible aeroplane model. This study sought to optimise a metric of endurance (a function of the lift-drag ratio and fuel fraction) over symmetric manoeuvres and gusts. Optimal actuation within aircraft manoeuvres was also studied by Maraniello et al⁸ using a nonlinear geometrically exact beam model. Here a pre-programmed open-loop actuation is optimised via parametrising the deflection of control surfaces with a set of B-splines. In these cases, more complex

control architecture is incorporated into the optimisation, however the studies lack observer design, actuator/sensor sizing, and actuator/sensor placement considerations, problems well studied outside of the ASE literature. For example, the optimal placement of multiple actuators under $\mathcal{H}_\infty$ -control has been studied by Kasinathan et al.,⁹ conditions for the convergence, to an exact optimal performance and corresponding actuator location, of a sequence of approximations used to an exact problem. Similar existence and convergence proofs are not found within the ASE literature, despite the fact that it was shown in Demetriou¹⁰ that placement of sensors can significantly change overall system performance.

It has yet to be decided what class of optimiser is best suited for MDO of ASE problems and as such, many have been employed. Classically, most optimisation problems were solved with descent methods that used the gradients of the objective function (or approximations to the gradient) to find solutions that satisfied the Karush-Kuhn-Tucker (KKT) conditions. These are necessary conditions for a stationary point of a constrained non-linear objective function, but in general do not guarantee globally optimal solutions. Recently, non-gradient based optimisations have been used for aeroservoelastic design. These methods usually rely on using numerous candidate solutions that are spread throughout the design space. For example, a Particle Swarm Optimisation (PSO) method was used by Haghighat et al.⁷ on a linear, flexible aeroplane model; PSO iteratively attempts to improve the best known solution by using a population of particles (candidate solutions) that update their position and velocity through the design space according to both their local information and information on the population's (swarm's) best known positions. The advantage of non-gradient based methods is that they are able to sample large portions of design space, gathering global information. However, these methods can quickly become infeasible if function evaluations are computationally expensive. The strategies introduced in this paper are gradient based but take advantage of globally collected data.

A major hurdle in the automation of design optimisation is automating the selection or priority of variables. Traditionally, design variables are chosen from a large set of parameters from engineering experience. Since this could limit the potential in new concept designs, automating this choice with a more meticulous process may lead to improvements, as studied in Tu et al.¹¹ As an approximate model is being constructed, the variable space is sampled and variables that are likely to be dominant are kept and less influential ones are dropped. A variable screening process could become imperative as the number of design variables increases, especially if the complexity of the design-space is prohibiting the potential of a simultaneous design. More recently, a variable reduction technique has been investigated by Ghisu et al.¹² Their work shows how a re-parametrisation based on a Proper Orthogonal Decomposition (POD) of initial optimisations can accelerate a design optimisation. Their paper is a clear proof of concept that a general parametrisation can be used to not only to facilitate the optimisation but also to aid variable selection. However it does not investigate when this technique can lead to improvements, why it can accelerate optimisation and how best to choose a sample of data for the POD. Furthermore, the method is not yet fully automated as the number of variables removed is dependent on an arbitrarily chosen user input. This paper proposes to address these problems, namely the question of what is an efficient and effective strategy in situations where a design would benefit from a simultaneous optimisation, i.e. systems that have strongly coupled interdisciplinary forces, but where the complexity of simultaneous design-space restricts the potential benefits. A highly coupled design will be investigated with the aim to answer how and when a reparametrisation and variable reduction strategy could improve an aeroservoelastic optimisation. This will be accomplished via a model with strong interdisciplinary couplings but with minimal computational demand. Low computational costs will allow a careful and thorough investigation. Section II introduces the design model being used and defines the optimisation problem. Section III describes the optimisation methodology, detailing how a new basis is formed and its potential advantages. Results are presented in Section IV. First, it is demonstrated that the design model can benefit from a simultaneous design compared to the traditional sequential plant/controller approach. Results after reparametrisation and dimension reduction are discussed.

II. Model

A computationally inexpensive model will be developed first in order to investigate automated Aeroservoelastic (ASE) optimisation. When creating the model it was made a priority to ensure variables from each of the “aero”, “servo” and “elastic” fields were chosen so that a full ASE optimisation was possible. The optimisation will be then formulated as a traditional non-linear optimisation with non-linear inequality constraints.

A. Design Model

The theoretical basis for the model to be described is based on Gaunaa¹³ which extends Theodorsen's model¹⁴ to include aerofoil deformation. The aerofoil used is shown in Figure 1 and consists of the following: a two-dimensional, potential flow over a deforming aerofoil; an actively controlled, but saturated compliant trailing edge; a dynamic observer that uses a series of pressure sensors across the aerofoil; and a simple heave/pitch linear spring model. The aerofoil has three degrees of freedom: a vertical translation, a rigid-body rotation, and lastly one for the state of the trailing edge. An approximation is made for the unsteady aerodynamics, which allows for a state-space formulation of the full model. Although this model is simple it allows design variables to be picked from the various ASE fields, for example, the natural frequency of the springs for the structure; the shape of the compliant flap for the aerodynamics; the location of pressure sensors for physical actuation; and controller gains for the control law.

Figure 1. Aeroservoelastic model detailing the trailing edge hinge ξ_h , the distance to the elastic axis ξ_{ea} and the free-stream velocity V .

1. State-space representation

This model can be reduced to a linear state-space form following the same finite state approximation in Gaunaa.¹³ The states are

$$x = \left[Y, \alpha, \beta, \dot{Y}, \dot{\alpha}, \dot{\beta}, z_1, \dots, z_N \right]^\top, \quad (1)$$

where Y and α are the heave and pitch degree of freedoms, β is the trailing edge degree of freedom, which will be discussed in the next section, and $\{z_i\}_{i=1}^N$ are the linear aerodynamic states used to approximate the unsteady 2D potential flow over the deforming aerofoil. This gives a state vector of dimension $n = 6 + N$. The equations of motion can then be written in a linear state-space form,

$$\dot{x}(t) = Ax(t) + B_u u(t) + B_w w(t), \quad (2)$$

where $A \in \mathbb{R}^{n \times n}$ is the state matrix, $B_u \in \mathbb{R}^{n \times 1}$ is the control input matrix, $B_w \in \mathbb{R}^{n \times 1}$ is the gust input matrix, $u \in \mathbb{R}$ is the control input, $w \in \mathbb{R}$ is an external vertical gust input and $x(t) \in \mathbb{R}^{n \times 1}$ is the state vector.

2. Trailing edge state

The shape of the aerofoil is defined by

$$S(\xi) = \begin{cases} 0 & \text{for } -1 \leq \xi \leq \xi_h, \\ S_\beta(\xi) & \text{for } \xi_h \leq \xi \leq 1, \end{cases} \quad (3)$$

where ξ is non-dimensionalised by the half-chord length, so that $-1 \leq \xi \leq 1$ from the leading edge to the trailing edge and ξ_h is the hinge point of the actuator. The shape function, $S_\beta(\xi)$, is defined by an arbitrary number of parameters a_i within a polynomial expansion about ξ_h ,

$$S_\beta(\xi) = \frac{(1 - \xi_h)}{\sum_{i=2}^n a_i} \sum_{i=2}^n a_i \frac{(\xi - \xi_h)^i}{(1 - \xi_h)^i}. \quad (4)$$

Note that the series starts at $i = 2$ to ensure that the trailing edge attaches smoothly to the camberline. This formalisation allows for an aerodynamic variable to be optimised via the variables a_i . The shape of the aerofoil at time t is given by,

$$s(\xi, t) = \beta(t) S(\xi). \quad (5)$$

Since the system is linear the scaling of this function is arbitrary. It has been chosen so that $S_\beta(1) = 1 - \xi_h$, meaning that for small angles the state $\beta(t)$ is approximately the angle of the flap. This is clarified in Figure 2: given an input $\beta(t)$, the angle β' made by joining the trailing edge of the aerofoil to the hinge point can be approximated by,

$$\beta' \approx \sin \beta' = \frac{(1 - \xi_h)}{l} \beta \approx \beta. \quad (6)$$

Figure 2. Definition of the trailing edge shape and its physical meaning.

The dynamics of the trailing edge are assumed to elastically oscillate around u , which is described by

$$m_\beta \ddot{\beta}(t) + m_\beta \omega_\beta^2 (\beta(t) - u(t)) = F(t), \quad (7)$$

where ω_β is the natural frequency of the elastic oscillations, F represents the aerodynamic and inertial generalised forces acting on the trailing edge,¹³ and m_β is given by

$$m_\beta = b^3 \int_{-1}^1 \rho(\xi) S_\beta^2 d\xi. \quad (8)$$

where b is the semi-chord and $\rho(\xi)$ is the density of the aerofoil, described in the next section.

3. Mass distribution

Since the aerofoil model is one dimensional the mass distribution can be assigned arbitrarily. We will enforce that stiffening the aerofoil will require adding more mass. Hence the density of the aerofoil is modelled as,

$$\rho(\xi) = \begin{cases} \rho_r (\omega_y^2 + \omega_\alpha^2) & \text{for } -1 \leq \xi < \xi_h, \\ \rho_r \omega_\beta^2 & \text{for } \xi_h \leq \xi \leq 1, \end{cases} \quad (9)$$

where ω_y and ω_α are the heave and pitch natural frequencies of the linear springs seen in Figure 1 and $\rho_r = 1 \text{ kg s}^2 \text{ m}^{-1}$ is a constant of proportionality between the frequencies and density. Integrating this density distribution over the aerofoil gives a total mass of

$$M_{tot} = b \rho_r ((1 + \xi_h) (\omega_y^2 + \omega_\alpha^2) + (1 - \xi_h) \omega_\beta^2). \quad (10)$$

It can be observed that a stiffer wing, with higher natural frequency, will be heavier.

4. Controller and Observer

An LQG controller and observer will be created for the aerofoil to form the control input $u(t)$ from the pressure sensors on the chord. The controller and observer can be designed separately. The controller is defined by a linear quadratic minimisation

$$\min_K \int_0^T \left(Q_y Y^2(t) + Q_\alpha \alpha^2(t) + R(Kx)^2(t) \right) dt, \quad (11)$$

where Q_y and Q_α are control weightings to be used as design variables, whereas R penalises the control input and is given by,

$$R = (1 - \xi_h) \omega_\beta^2. \quad (12)$$

Therefore, it is proportional to the mass of the flap, i.e. a heavy flap would require more control energy. Note that Q_y , Q_α and R must be positive for a solution to Eq. 11 to exist.

The observer dynamics are formed from a series of pressure sensors and the control input,

$$y(t) = Cx(t) + Du(t), \quad (13)$$

for matrices $C \in \mathbb{R}^{N_p \times n}$, where N_p is the number of pressure sensors. The estimated state vector $\hat{x}$ satisfies,

$$\dot{\hat{x}} = A\hat{x} + B_u u + L(y - C\hat{x}). \quad (14)$$

The observer gain $L \in \mathbb{R}^{n \times N_p}$ is calculated by the Kalman filter problem,¹⁵

$$\min_L \lim_{t \rightarrow \infty} E \left[(x(t) - \hat{x}(t)) (x(t) - \hat{x}(t))^T \right], \quad (15)$$

where $E[\cdot]$ is the expectation in response to a stochastic input with known variance. Both K and L are found using the lqr and kalman packages in MATLAB that solve Eq. 11 and 15 via Riccati equations.

To restrict the authority of the control, saturation has been introduced in Eq. 16. This is the only non-linear component of the model. The addition of saturation introduces an equilibrium in the aggression of the controller. Low control activity is not optimal for performance but a controller that acts too aggressively leads to a system with too much saturation, which in this model leads to unstable dynamics.

$$u(t) = \begin{cases} u_+ & \text{for } u > u_+, \\ K\hat{x}(t) & \text{for } |u| \leq u_+, \\ -u_+ & \text{for } u < -u_+. \end{cases} \quad (16)$$

For all results presented in this work, the saturation limit of $u_+ = 5^\circ$ is imposed.

5. Model Parameters

Not all the parameters within the model have been discussed, however, Table 1 lists those remaining parameters needed to reproduce the current model. The density of air affects the matrices A , B_u and B_w in Eq. 2 as described by Gaunaa.¹³

Parameter description	Value
Density of air	1 kgm^{-3}
Length of the chord	2 m
Position of elastic axis	$\xi_{ea} = -0.4$
Flap hinge	$\xi_h = 0.8$

Table 1. Remaining model parameters.

B. Optimisation

1. Design variables

With the goal to optimise a full ASE system the design variables chosen have been taken from the each of the three fields. First, structural parameters have been chosen to be ω_y , ω_α and ω_β ; the natural frequencies of the heave, pitch and trailing edge. These are not only proxies for the flexibility of aerofoil but also define the total mass and the position of the centre of mass, which is vital to the stability of the system. The control law will also be optimised by means of the variables Q_y and Q_α , introduced in Eq. 11. These variables will weight the control law on whether to preference either the heave or pitch motions when stabilising the system. The aerodynamic shaping, i.e. the trailing edge, will be achieved by design of the polynomial coefficients in

Eq. 4. In this investigation only two of these coefficients will be optimised over, a_2 and a_3 , the rest will be set to zero. Finally, the placement of two pressure sensors on the camberline, p_1 and p_2 , will be designed as to optimise the observer design. In summary there are nine design variables in the design vector X ,

$$X = \left(\omega_y \quad \omega_\alpha \quad \omega_\beta \quad Q_y \quad Q_\alpha \quad a_2 \quad a_3 \quad p_1 \quad p_2 \right)^T. \quad (17)$$

2. Cost function

The cost function to be minimised during the aerofoil design is made up of both a static measure of performance and a dynamic performance measure. The static measure $f_1(X)$ is simply the total mass given in Eq. 10, that is,

$$f_1(X) = M_{tot} = b\rho_r \left((1 + \xi_h) (\omega_y^2 + \omega_\alpha^2) + (1 - \xi_h) \omega_\beta^2 \right). \quad (18)$$

The dynamic measure $f_2(X)$ of the aerofoil is defined by the total stress in the elastic axis,

$$f_2(X) = \int_0^T [k_y Y^2(X; t) + k_\alpha \alpha^2(X; t)] dt, \quad (19)$$

where $k_y = M_{tot}\omega_y^2$ is the heave spring stiffness and $k_\alpha = I_{tot}\omega_\alpha^2$ is the pitch spring stiffness, with I_{tot} being the total moment of inertia of the aerofoil. This motion is in response to a vertical gust input given by,

$$w(t) = \begin{cases} w_0 (1 - \cos(\frac{\pi V t}{l})) & \text{for } 0 \leq t \leq \frac{2l}{V}, \\ 0 & \text{else} \end{cases}, \quad (20)$$

where V is the test free-stream velocity of the aerofoil, $w_0 = 0.5 \text{ ms}^{-1}$ and $l = 1 \text{ m}$ is the length of the gust. A value of $T = 50$ was chosen as it is large enough to allow for any transient motion to decay, while not causing excessive computational effort. The external disturbance, w , feeds into the dynamics via Eq. 2. This objective function minimises the stress over a gust response explicitly. This would be key if the design's longevity and reliability were important as minimising stress will lower long term fatigue. The total cost function $f(X)$ will simply be given by

$$f(X) = f_1(X) + f_2(X). \quad (21)$$

3. Constraints

The optimisation will be constrained by two non-linear constraints and by bounds on the design vector. *Design vector bounds:* These bounds ensure that the solution is physical, for example, that the mass is non-negative and that the pressure sensors are on the aerofoil. The design vector will be rescaled by these upper and lower limits so that each component is between 0 and 1, i.e $0 \leq X_i \leq 1$ for all i .

Flutter speed: The closed-loop linear flutter speed defined as the lowest positive free-stream velocity in which the closed-loop dynamics are linearly unstable, i.e. the lowest velocity such that there exists an eigenvalue of $A - B_u K$ with a non-negative real part. Note that A , B_u and K are all functions of V . A constraint on this linear flutter speed will be enforced. To mimic a flight safety margin, the aerofoil will be constrained to cruise at 80% of its linear flutter speed, i.e.

$$V \leq 0.8V_F, \quad (22)$$

where V_F is the linear flutter speed. This constraint will be written as,

$$c_1(X) = V - 0.8V_F \leq 0. \quad (23)$$

Trailing edge shape: It is expected that a trailing edge with larger arc length will be advantageous to the controller, since in this case it will be able to generate a bigger force with the larger surface. Hence, if no constraint is imposed, optimal designs will often have non-monotonic trailing edges, which are physically unrealistic. As such it is necessary to implement a constraint on the trailing edge to ensure it remains monotonic. This can be achieved by,

$$\frac{d}{d\xi} S_\beta(\xi) \leq 0, \quad \xi_h \leq \xi \leq 1, \quad (24)$$

where S_β is defined in Eq. 4. This constraint will be written as,

$$c_2(X) = \max \left\{ \frac{dS_\beta}{d\xi}(\varepsilon) : \xi_h \leq \varepsilon \leq 1 \right\} \leq 0. \quad (25)$$

Now, concatenating the non-linear inequalities as $c = [c_1, c_2]^\top$ the optimisation problem can be written as,

$$\begin{aligned} & \text{minimise} && f(X) \\ & \text{subject to} && c(X) \leq 0, \\ & && 0 \leq X \leq 1. \end{aligned} \quad (26)$$

III. Optimisation Implementation

The question of how to choose a appropriate design vector parametrisation for an optimisation is often challenging. Traditionally, design variables were picked from experience and knowledge, which may not be practical when considering the design of novel platforms. When choosing a parametrisation, maximising how much of the design-space is accessible and connected is key but so is minimising the design-space complexity. These are often conflicting goals. Starting with the set of all variables it would be helpful to find a parametrisation that minimises the number of dimensions needed to describe the design well enough so that the design-space is still able to be adequately explored.

One method of achieving this is to use Proper Orthogonal Decomposition (POD). POD is a powerful data analysis tool that is often used to achieve a low dimensional representation of high dimensional data. This method is sometimes called Principal Component Analysis or Empirical Orthogonal Functions and is used in a plethora of subjects including turbulent flow analysis¹⁶ and image processing.¹⁷ POD uses a sample of points to create an orthogonal basis in which the first basis vector maximises the projection of the sample data, i.e. a basis $\Phi = [\phi_1, \phi_2, \dots, \phi_n]$ with $\Phi^\top \Phi = I$ is found such that its first basis vector ϕ_1 satisfies,

$$\max_{\phi_1} (\phi_1, X)^2 \quad \text{s.t.} \quad \phi_1^\top \phi_1 = 1, \quad (27)$$

where X is the sample data and $(\cdot, \cdot)$ is the scalar inner product operator. The second basis vector maximises the projection of sample data again but is constrained to be orthogonal to the first vector. This process continues until an orthogonal basis is formed with the last vector pointing in the direction that describes the least amount of variance.

This methodology closely follows that in Ghisu et al.¹² Firstly, to use a POD technique a sample of points is needed. With the goal of choosing a parametrisation that is beneficial to the optimiser, the paths of preliminary optimisations will be used as the sample to form the new parametrisation. This set of points will be referred to as the training set. The design vector X can now be written as

$$X = \sum_{j=1}^9 \gamma_j \phi_j, \quad (28)$$

where now X is parametrised by $\gamma_j \in \mathbb{R}$ and $j = 1, \dots, 9$ since $X \in \mathbb{R}^9$. In general, the components will no longer correspond to the physical variables described in Eq. 17. However, the aim of such a transformation is to rotation the basis such that the first basis vector points in the dominant direction of the observed optimisation paths which in turn may help with the numerical conditioning of the problem.

With this new basis, optimisation can now be performed with the intention that the rotation will yield a faster and more robust optimisation routine. Furthermore, this method offers a way to make informed decisions on the removal of dimensions, namely by removing the least variant directions. Not only does POD give these least variant directions but also for each basis vector there is an associated eigenvalue λ_i which is proportional to the variance in that direction. Hence one can calculate the percentage of variance described from removing one or more of these POD dimensions by,

$$\Lambda = \frac{\sum_{i=1}^{d \leq D} \lambda_i}{\sum_{i=1}^D \lambda_i} = \text{Fraction of variance described}, \quad (29)$$

where D is the dimension of the design vector and d a positive integer less than D . This leads to a natural way of deciding how many dimensions to remove. Prior to optimisation, a total variance described by a reduced problem can be set and then subsequently the maximum number of dimensions can be removed while this percentage is still described. The optimiser can then converge in this smaller space, the removed dimensions can then be considered again and the full optimisation restarted at this point.

Throughout the paper optimisations are performed using a gradient based interior point algorithm within the MATLAB `fmincon` function. Interior point algorithms are iterative and minimise the cost function with the constraints using approximations to the indicator function, i.e.

$$\min_X f(X) + I(c(X)), \quad (30)$$

where f is the cost function, c is a constraint and $I(X) = 0$, if the constraint is not violated, or $I(X) = \infty$, otherwise.

IV. Results

A. Model Verification

Verification of the implementation of the ASE model was accomplished by setting the shape of the trailing edge to a flat plate. This is equivalent to the classic Theodorsen's problem,¹⁴ i.e.

$$S_\beta(\xi) = \begin{cases} 0 & -1 \leq \xi \leq \xi_h \\ (\xi_h - \xi) & \xi_h \leq \xi \leq 1 \end{cases}. \quad (31)$$

A plot of non dimensional flutter speed against the non dimensionalised hinge spring frequency for the three degree of freedom (heave, pitch and flap rotation) linear system, using the parameters detailed in,¹⁸ has been compared to those by Zeiler¹⁸ with Figure 3 indicating that there is excellent agreement between both implementations.

Figure 3. Non-dimensional flutter speed, $V_f / (b\omega_\alpha)$ against a nondimensionalised hinge spring frequency, $\omega_\beta / \omega_\alpha$ for a flat plate, flat flap camberline model. The current code is compared to results from.¹⁸

B. A Comparison of Simultaneous and Sequential Optimisation Strategies

Greater benefit is expected to be achieved from using a simultaneous optimisation, compared to a sequential optimisation, on highly coupled aeroservoelastic systems such as the actuated aerofoil described here. However, even though a sequential approach should be sub-optimal it is possible that a simultaneous optimisation can fail to reach its potential due to complexity in the design-space. It is possible to demonstrate these points by comparing optimal designs of the model, detailed in Section II, at various free-stream velocities V . An optimisation will be defined as sequential when, an initial condition's open-loop dynamics are optimised first, i.e. the structural and aerodynamic variables $\omega_y, \omega_\alpha, \omega_\beta, a_2$ and a_3 . These are then fixed and the remaining variables optimised, i.e. Q_y, Q_α, p_1 and p_2 . In comparison, a simultaneous optimisation will refer to the case in which all nine variables are varied simultaneously. For brevity, denote one step of a gradient-based optimizer of type q from the current decision variables $X \in \mathbb{R}^d$ to its subsequent value X^+ by

$$X^+ = \phi^q(X). \quad (32)$$

In the following q represents either sequential ($q = \text{Seq}$) or simultaneous ($q = \text{Sim}$) optimisation. Algorithm 1 produces a trajectory of decision variables $\phi_\varepsilon^q(X^0) \in \mathbb{R}^{d \times n}$ corresponding to convergence of such an algorithm to a tolerance ε , from initial condition X^0 , where n is the total number of algorithm iterations required to reach the chosen tolerance.

Algorithm 1 Convergence for optimisation of type q .

- 1: **inputs** $X^0 \in \mathbb{R}^d$ satisfying $0 \leq X^0 \leq 1$, $\varepsilon > 0$, $q \in \{\text{Sim}, \text{Seq}\}$, $n = 0$.
 - 2: **repeat**
 - 3: $X^{n+1} \leftarrow \phi^q(X^n)$
 - 4: $n \leftarrow n + 1$
 - 5: **until** $|X^n - X^{n-1}| < \varepsilon$
 - 6: $\phi_\varepsilon^q(X^0) \leftarrow [X^0, X^1, \dots, X^{n-1}]$
 - 7: $\phi_\varepsilon^{q,*}(X^0) \leftarrow X^{n-1}$
 - 8: **return** $\phi_\varepsilon^q(X^0) \in \mathbb{R}^{d \times n}$ and $\phi_\varepsilon^{q,*}(X^0) \in \mathbb{R}^d$.
-

Following Algorithm 1, simultaneous and sequential optimisations are compared in Table 2 at various free-stream velocities and over a set of random initial condition $\{X_i^0\}_{i=1}^N$ with $N = 50$ and $\varepsilon = 10^{-8}$. The percentage difference from the sequential optimisation to the simultaneous optimisation will be used to compare various quantities. For a quantity θ the percentage difference PD_θ is defined as

$$PD_\theta = 100 \times \frac{\theta^{\text{Sim}} - \theta^{\text{Seq}}}{\theta^{\text{Seq}}}. \quad (33)$$

The quantities to be discussed are the following:

- The optimal function objective over the initial conditions, $\bar{f}(V) = \min_i f(\phi_\varepsilon^{q,*}(X_i^0))$.
- The mean position of the centre of mass over the initial conditions, cg .
- The mean number of function evaluations, $func$.
- The variance of the converged design vectors $X_i^* = \phi_\varepsilon^{q,*}(X_i^0)$, i.e. $var(V) = \frac{1}{N} \sum_{i=1}^N (X_i^* - \bar{X}^*)^2$, where $\bar{X}^* = \frac{1}{N} \sum_i X_i^*$.

It can be seen in Table 2 that, surprisingly, for lower free-stream velocities a sequential optimisation outperforms the simultaneous optimisation, obtaining a design with a 2.7% reduction in cost function, contradicting expectations. This is likely caused by the extra complexity of a simultaneous optimisation, for example, the simultaneous design-space may contain extra superfluous local minimum that the optimiser cannot avoid. However, as the velocity increases the coupling of the control design to the aerofoil becomes stronger for two reasons: the lift force and moment the trailing edge is able to generate increases with free-stream velocity, hence, at higher velocities the control surface can exert a more significant effect on the dynamics, meaning that integration of its design into the simultaneous optimisation is potentially advantageous. Second, as the free-stream velocity increases, the volume of the feasible set of design space decreases due to the flutter constraint, again implying an increasingly important role of the control design. In any case, for $V \geq 3.5 \text{ ms}^{-1}$ it

$V \text{ ms}^{-1}$	$PD_{\bar{f}}$	PD_{cg}	PD_{func}	PD_{var}
2.5	2.7	0.1	160	-3.3
3	0.1	-1.5	200	2.4
3.5	-5.6	1.5	220	14.5
4	-8.1	4.7	520	18.6
4.5	-8.7	6.0	700	36.7
5	-8.4	6.2	540	20.6

Table 2. Percentage differences from sequential to simultaneous optimisations of various quantities.

is observed that simultaneous optimisation outperforms sequential optimisation by at least 5%. An example of how a simultaneous optimisation can outperform a sequential optimisation at higher velocities can be seen when looking at the paths through design space taken by each method. A set of initial conditions is created such that

$$X_0^{ij} = \{[\omega_y^i, \omega_\alpha^j, \omega_\beta^0, Q_y^0, Q_\alpha^0, a_2^0, a_3^0, p_1^0, p_2^0]^\top : \omega_y^0 \leq \omega_y^i \leq \omega_y^0 + \delta \text{ and } \omega_\alpha^0 \leq \omega_\alpha^j \leq \omega_\alpha^0 + \delta\}.$$

This set forms a small two-dimensional box in ω_y and ω_α space in which $\omega_y^i = \omega_y^0 + (i - 1)\delta/10$ and $\omega_\alpha^j = \omega_\alpha^0 + (j - 1)\delta/10$. From these points both a simultaneous and sequential optimisation is run and their path through $(\omega_y, \omega_\alpha, \omega_\beta)$ space is plotted in Figure 4 for $\delta = 0.01$. When comparing simultaneous

Figure 4. Iteration paths from numerous close initial conditions from a sequential method, blue, and a simultaneous method, green. The final design points of the simultaneous method are marked with dark green the circles. A convex hull has been plotted around these points along with its projections in the $\omega_y - \omega_\alpha$, $\omega_\alpha - \omega_\beta$ and $\omega_y - \omega_\beta$ planes.

optimisation to sequential optimisation it can be seen in Table 2 that at higher velocities the average position of the centre of gravity is further aft, by as much as 6.2%. In general moving the centre of gravity towards the tail leads to an aerofoil that is more prone to flutter instability. The placement of the centre of gravity in this model is calculated using the density in Eq. 9, i.e. the heavier the flap is made relative to the aerofoil the more aft the centre of gravity. Since this is a more unstable location these designs are often open-loop unstable. However, the open-loop instability allows for faster actuation, and a corresponding decrease in the dynamic component of the cost function $f_2(X)$. Although the simultaneous method is able to find better

optimal designs it suffers from greater sensitivity to initial conditions and requires more function evaluations. This sensitivity can be seen both in Figure 4 and Table 2. The percentage change in the variance of the optimal designs is as high as 36.7% and the number of function evaluations increases by a factor of seven. We now discuss methods that reduce these unwanted consequences of simultaneous optimisation.

C. Parametrisation

Clearly every engineering design has to first be parametrised so a design vector can be passed to an optimiser, furthermore, the choice of this parametrisation can affect the optimisation and final design. Hence, it may maximise the performance of an automated Multidisciplinary Design Optimisation (MDO) technique if the design vector parametrisation is automatically selected within the optimiser. Indeed, even a rotation of a parametrisation can change the final design. Suppose an alternative basis of $\mathbb{R}^d$ is represented by a unitary matrix $U \in \mathbb{R}^{d \times d}$ whose columns $\{u_i\}_{i=1}^d \subset \mathbb{R}^d$ are the elements of the basis. Now, for any vector $X \in \mathbb{R}^d$ it is possible to express

$$X = \sum_{i=1}^d \alpha_i u_i, \quad (34)$$

where $\alpha_i = (U^T X)_i \in \mathbb{R}$. Consequently, we may define a gradient-based optimisation algorithm with decision variables $(\alpha_i)_{i=1}^d$ in which gradients are computed with respect to the rotated basis U .

$$\phi_\varepsilon^q(X^0, U) \in \mathbb{R}^{d \times n}, \quad \phi_\varepsilon^{q,*}(X^0, U) \in \mathbb{R}^d, \quad (35)$$

to be the analogous outputs of Algorithm 1 corresponding to optimisation performed in the rotated basis U and with tolerance $\varepsilon > 0$. A natural choice for the rotated basis is to use Proper Orthogonal Decomposition (POD), and we denote the POD basis calculated from a collection of vectors $\mathbf{X} = [X_1, \dots, X_n] \in \mathbb{R}^{d \times n}$ by $\{u_i, \lambda_i\}_{i=1}^d = \text{POD}(\mathbf{X})$. Here, $\lambda_i \in \mathbb{R}$ are the singular values corresponding to each $u_i \in \mathbb{R}^d$. The ensemble of vectors to be passed to the POD analysis are a collection of iteration paths from preliminary optimisations starting from $\{X_i^0\}_{i=1}^N$, i.e. $\mathbf{X} = [\phi_\varepsilon^q(X_1^0, I), \dots, \phi_\varepsilon^q(X_n^0, I)]$, where $I \in \mathbb{R}^{d \times d}$ is the identity matrix. This $\mathbf{X}$ is referred to as *the training set* and its effect on a rotated base optimisation is studied by varying the stopping condition ε used to collect it.

As expected, the training set converges to increasingly better designs as the tolerance condition is reduced

Figure 5. The lowest cost function found from each method, normalised by the lowest found by the training set with $\varepsilon = 1 \times 10^{-8}$ and shown for a range of values for the training set stopping condition ε .

and made more stringent. What can be seen in Figure 5 and may not be expected is that a simple rotation of the basis vectors can lead to the optimisation outperforming the best training set local minimum, found at $\varepsilon = 1 \times 10^{-8}$, by roughly 5% even for the larger values of ε . The POD optimisation plot is approximately constant over eight orders of magnitude of ε and always lies beneath the best local minimum found in the original basis. This result is slightly counter-intuitive, for one, computational effort is “wasted” in the formation of the training set and then the optimiser is able to find a better solution regardless of the fact that both methods are solving the same problem defined in Eq. 26.

D. Removing Dimensions

It has been shown that the aerofoil model studied here can benefit from a simultaneous optimisation strategy. However, the optimiser can struggle with the complexity of the design-space. For computational efficiency it may be of benefit to temporarily optimize over only a subset of the available decision variables. Thus, results are presented on whether removing complexity temporarily can help the optimiser navigate to improved local minima. Since the goal is to study automated MDO the process of removing complexity needs to be algorithmic rather than user defined. This can be achieved by using a POD parametrisation as explained in Section C.

A POD basis is ordered such that the first basis vector points in the best direction to describe the variance seen in the sample, while the last basis vectors describes the least amount of variance in comparison. This is used to temporarily reduce the dimension of the optimisation. As shown in Eq. 29, the total variance described by a reduced POD basis can easily be calculated and, hence, be used to algorithmically remove a number of dimensions. This process is carried out as follows, given an initial condition $\alpha^0 = (\alpha_1^0, \dots, \alpha_d^0)^\top$ with respect to a basis $\{u_i\}_{i=1}^d$ suppose that only the first $1 \leq r < d$ basis vectors are optimized over, to a tolerance ε , while the remaining decision variables $\alpha_{r+1}^0, \dots, \alpha_d^0$ are fixed to be their initial values throughout the optimisation. We write the resulting, partially optimized, vector

$$(\alpha_1^*, \dots, \alpha_r^*, \alpha_{r+1}^0, \dots, \alpha_d^0)^\top =: \phi_{\varepsilon}^{q,*}(\alpha^0, \{u_i\}_{i=1}^r), \quad (36)$$

and note that $\phi_{\varepsilon}^{q,*}(\alpha^0, U) = \phi_{\varepsilon}^{q,*}(\alpha^0, \{u_i\}_{i=1}^N)$.

Algorithm 2 describes a three-step algorithm which makes use of basis rotation and reduction. Given a set of initial conditions $\{X_i^0\}_{i=1}^N$, a preliminary ensemble of optimisation trajectories *in the original basis* are computed using Algorithm 1 to a tolerance $\varepsilon_1 > 0$, referred to as the *training set tolerance*. Next, POD is applied to this ensemble of trajectories to determine the most dominant optimisation directions and a reduction proportion $0 < \Lambda \leq 1$ is selected. A new basis U is formed of the POD vectors and an initial optimisation, to tolerance ε_1 , is performed in only the $100 \times \Lambda\%$ most dominant POD directions. Using the outputs of this initial, coarse, optimisation as initial conditions a final set of optimisations are then performed in the full rotated basis to a *convergence tolerance* $0 < \varepsilon_2 < \varepsilon_1$.

Algorithm 2 Simultaneous optimisation in a rotated/reduced basis.

- 1: **inputs** $\{X_i^0\}_{i=1}^N \subset \mathbb{R}^d$ satisfying $0 \leq X_i^0 \leq 1$, $0 < \varepsilon_2 < \varepsilon_1$, $0 < \Lambda \leq 1$ and $q = \text{Sim}$.
 - 2: **for** $i = 1$ to N **do**
 - 3: $\mathbf{X}_i \leftarrow \phi_{\varepsilon_1}^q(X_i^0)$
 - 4: **end for**
 - 5: $\{u_i, \lambda_i\} \leftarrow \text{POD}([\mathbf{X}_1, \dots, \mathbf{X}_N])$
 - 6: $U \leftarrow [u_1, \dots, u_N]$
 - 7:
 - $$r \leftarrow \underset{1 \leq s \leq N}{\text{argmax}} \left\{ \frac{\sum_{i=1}^s \lambda_i}{\sum_{i=1}^N \lambda_i} \leq \Lambda \right\}$$
 - 8: **for** $i = 1$ to N **do**
 - 9: $Y_i \leftarrow \phi_{\varepsilon_1}^{q,*}(U^\top X_i^0, \{u_j\}_{j=1}^r)$
 - 10: $Y_i \leftarrow \phi_{\varepsilon_2}^{q,*}(Y_i, U)$
 - 11: $X_i^* \leftarrow Y_i$
 - 12: **end for**
 - 13: **return** $\{X_i^*\}_{i=1}^N \subset \mathbb{R}^d$.
-

Note that applying Algorithm 2 with $\Lambda = 1$ results in an optimisation in the rotated basis with no dimension reduction.

With the introduction of the reduction strategy, using $\Lambda = 0.95$, further improvements can be seen in Figure 5. As with the rotation strategy plot, the reduction plot is approximately constant and visibly outperforms the original simultaneous optimisations. Further comparisons between the original basis, POD rotation method and reduction method are found in Table 3.

Method	$PD_{\bar{f}}$	PD_{func}	PD_{var}
Rotation	-2	-28.4	-22.3
Reduction	-4	-37.5	-28.3

Table 3. Percentage differences from the original basis optimisations to the POD rotated optimisations and original basis optimisations to the reduction method.

Both the POD rotation and Reduction methods have successfully attenuated some of the undesired consequences that were demonstrated in Figure 4 and discussed in Section B. While performing slightly better than the standard simultaneous optimisation the investigated strategies have also reduced the number of function evaluations needed and lowered the sensitivity to initial conditions.

V. Conclusion

An aerofoil with an active compliant trailing edge has been modelled to facilitate an investigation into aeroservoelastic optimisation. The model was designed to be minimal in computational complexity while still allowing an optimisation over multiple aeroservoelastic disciplines. The advantages and disadvantages of using a simultaneous optimisation strategy as opposed to a sequential strategy have been presented by the canonical model considered. Namely, even though a sequential approach is sub-optimal it can outperform a simultaneous optimisation due to the design-space complexity preventing the optimiser locating the best solutions. However, with the goal to perform automated MDO of an actively controlled and highly coupled system a reparametrisation and variable reduction technique have been proposed. The reparametrisation of the design vector was accomplished by a linear transformation defined via a Proper Orthogonal Decomposition of preliminary optimisations. The two strategies investigated showed improvements of up to 5% in the cost function of the final design; the mean number of function evaluations needed decreasing by as much as 37%; and an increase in the robustness of the optimisation with respect to the initial designs.

Acknowledgments

This work has been sponsored by the AFOSR European Office of Aerospace Research and Development, grant number GA9550-14-1-0055.

References

- ¹Eli Livne. Integrated aeroservoelastic optimization: status and direction. *Journal of Aircraft*, 36(1):122–145, 1999.
- ²Carlos E S Cesnik, Rafael Palacios, and Eric Y Reichenbach. Reexamined Structural Design Procedures for Very Flexible Aircraft. *Journal of Aircraft*, 51(5):1580–1591, 2014.
- ³Shinji Suziki and Satoshi Yonezawa. Simultaneous structure/control design optimization of a wing structure with a gust load alleviation system. *Journal of Aircraft*, 30(2):268–274, March 1993.
- ⁴Changho Nam, Aditi Chattopadhyay, and Youdan Kim. Optimal wing planform design for aeroelastic control. *AIAA journal*, 38(8):1465–1470, 2000.
- ⁵Vineet Sahasrabudhe, Roberto Celi, André, and L Tits. Integrated Rotor-Flight Control System Optimization with Aeroelastic and Handling Qualities Constraints. *Journal of Guidance, Control, and Dynamics*, 20(2):217–224, 1997.
- ⁶Boris Moulin, Moshe Idan, and Mordechai Karpel. Aeroservoelastic Structural and Control Optimization Using Robust Design Schemes. *Journal of Guidance, Control, and Dynamics*, 25(1):152–159, 2002.
- ⁷Sohrab Haghighat, Joaquim R RA Martins, and Hugh H T. Liu. Aeroservoelastic design optimization of a flexible wing. *Journal of Aircraft*, 49(2):432–443, 2012.
- ⁸Salvatore Maraniello, Robert J S Simpson, and Rafael Palacios. Optimal manoeuvres with very flexible wings. In *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, page 1223, 2016.
- ⁹Dhanaraja Kasinathan and Kirsten Morris. Convergence $\mathcal{H}_\infty$ -optimal actuator locations. *IEEE Conference on Decision and Control and European Control Conference*, pages 627–632, Orlando, Florida, 2011.

- ¹⁰Michael A Demetriou. Robust sensor location optimization in distributed parameter systems using functional observers. In *44th IEEE Conference on Decision and Control and European Control Conference*, pages 7187–7192. IEEE, Seville, Spain, 2005.
- ¹¹Jian Tu and Donald Jones. Variable Screening in Metamodel Design by Cross-Validated Moving Least Squares Method. In *44th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. American Institute of Aeronautics and Astronautics, Norfolk, Virginia, 2003.
- ¹²Tiziano Ghisu, Geoffrey T Parks, Jerome P Jarrett, and P John Clarkson. Adaptive polynomial chaos for gas turbine compression systems performance analysis. *AIAA Journal*, 48(6):1156–1170, 2010.
- ¹³Mac Gaunaa. *Unsteady 2D potential-flow forces on a thin variable geometry airfoil undergoing arbitrary motion*. Risø-R Report, 2006.
- ¹⁴Theodore Theodorsen and W H Mutchler. General theory of aerodynamic instability and the mechanism of flutter. *NACA Technical Report*, 1935.
- ¹⁵Robert F Stengel. *Optimal control and estimation*. Courier Corporation, 2012.
- ¹⁶Nicholas Kevlahan, Robert Hunt, and John Vassilicos. A comparison of different analytical techniques for identifying structures in turbulence. *Applied scientific research*, 53(3-4):339–355, 1994.
- ¹⁷Lawrence Sirovich and Michael Kirby. Low-dimensional procedure for the characterization of human faces. *Josa a*, 4(3):519–524, 1987.
- ¹⁸Thomas A Zeiler. Results of Theodorsen and Garrick Revisited. *Journal of Aircraft*, 37(5):918–920, September 2000.