This document is classified as ISAE SUPAERO Confidential.

Master of Aerospace Engineering Research Project

HALE AEROECODESIGN

S2 Progress report

Author: Víctor Manuel Guadaño Martín

Due date of report: 08/04/2020
Actual submission date: 07/04/2020

Starting date of project: 28/01/2020 Duration: 14 Months

Tutors: J. Morlier, E. Duriez

This document is the property of the ISAE SUPAERO and shall not be distributed nor reproduced without the formal approval of the tutors.

Contents

1 Goal of the project

The main purpose of the project is to refine a modified version of OpenAeroStruct presented in [1] in order to obtain better results. The CO_2 footprint optimization of a solar-powered High Altitude Long Endurance (HALE) drone is studied. It's as important to achieve better results as it is to improve the performance of the program and obtain faster optimizations.

2 Project issues

In this first stage of the project it is difficult to know the main issues. One of the major problems to face is the convergence of the optimizations, since the model could be very close to reality but the starting points must be chosen carefully to find a minimum. For this reason, it is important to find a compromise solution between the convergence of the optimization and the complexity of the model. Finding this equilibrium is the central issue of the project.

3 Main bibliography and State of the Art

High-Altitude Long Endurance (HALE) is the description of an air-borne vehicle which functions optimally at high-altitude (around 60000 ft) and is capable of flights which last for considerable periods of time without recourse to landing. They are also called *atmospheric satellites* or *atmosats* because their development is focused towards providing services conventionally provided by artificial satellites orbiting in space. One of the main advantages of HALE drones, apart from being powered by solar energy, is that they are environment-friendly and much cheaper than satellites.

As no fuel is burned during the operation of HALE drones, their CO_2 emissions come from the manufacturing and the materials. Therefore, continuing the work presented in [1], this is what will be optimized.

Multi-disciplinary design optimization (MDO) consists in using optimization methods to solve design problems incorporating a number of disciplines, as well as finding an optimum for the interaction of disciplines and not for each of them separately. In this case, a modified version of OpenAeroStruct is used, which is a tool based on OpenMDAO. OpenAeroStruct is a global low-fidelity tool that performs aeroestructual optimization, while OpenMDAO is a multidisciplinary design optimization framework developed by NASA.

References

[1] E. Duriez and J. Morlier, "HALE multidisciplinary design optimization with a focus on Eco-Material selection," *ISAE Supaero*, 2020.

[2] G. Gerard and H. Becker, "Handbook of Structural Stability. Part 3. Buckling of Curved Plates and Shells.," tech. rep., NATIONAL AERONAUTICS AND SPACE ADMINISTRATION WASHINGTON DC, 1957.

[3] J. P. Jasa, J. T. Hwang, and J. R. Martins, "Open-source coupled aerostructural optimization using Python," *Structural and Multidisciplinary Optimization*, vol. 57, no. 4, pp. 1815–1827, 2018.

[4] D. Colas, N. H. Roberts, and V. S. Suryakumar, "HALE multidisciplinary design optimization Part I: Solar-powered single and multiple-boom aircraft," in *2018 Aviation Technology, Integration, and Operations Conference*, p. 3028, 2018.

[5] Y. Yang, Y. Chao, and W. Zhigang, "Aeroelastic dynamic response of elastic aircraft with consideration of two-dimensional discrete gust excitation," *Chinese Journal of Aeronautics*, 2019.

[6] S. S. Chauhan and J. R. Martins, "Low-fidelity aerostructural optimization of aircraft wings with a simplified wingbox model using OpenAeroStruct," in *International Conference on Engineering Optimization*, pp. 418–431, Springer, 2018.

[7] B. Morrisey and R. McDonald, "Multidisciplinary design optimization of an extreme aspect ratio hale uav," in *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO) and Aircraft Noise and Emissions Reduction Symposium (ANERS)*, p. 6949, 2009.

[8] J. P. Jasa, S. S. Chauhan, J. S. Gray, and J. Martins, "How certain physical considerations impact aerostructural wing optimization," in *AIAA Aviation 2019 Forum*, p. 3242, 2019.

4 Milestones of the project

At this point of the project, it is really complicated to estimate the time of execution of each task. Therefore, seven innovative tasks are presented hereunder. I will take the liberty of adding or removing tasks in the future in order to meet as well as possible the main objective of this research project within the deadlines proposed.

4.1 Task 1: Add a constraint on the wing surface

This easy task is an attempt to reduce the *snowball* effect. According to [1]: "A small increase in the drone's weight, leads to an increase of battery and solar panel weight in order to thrust the heavier drone, and an increase of structural weight to lift heavier drone. These weight increases contribute to a further increase in the drone's weight. This is a *snowball* effect".

By limiting the wing surface, the weight is also limited, preventing the optimization from diverging. However, this task has already been implemented with no positive effect. Consequently, it should be checked to identify and solve the problem.

4.2 Task 2: Fix some design variables

Following the first task (*Task 1: Add a constraint on the wing surface*), another way to help convergence could be fixing some of the design variables which vary less during the optimization process, such as the taper ratio or the root chord. With less design variables, the model should converge much more easily, making the problem more computationally efficient.

4.3 Task 3: Turn material function into OpenMDAO component

OpenMDAO has a modular implementation. The model is decomposed into *components*. These components are the most efficient way to perform gradient-based optimization. Each component computes the analytic partial derivatives of its outputs with respect to its inputs.

However, the material formulation presented in [1] is performed with a function and not with a component. Thus, this function allows us to use the density as the unique material design variable (input) and extract the material data (outputs) such as the Young's modulus, the shear

modulus, the yield strength and the CO_2 emissions.

Therefore, it would be very interesting to have a component for this purpose instead of a function since, currently, all the components that need the material properties have the density as an input and must call the function to access the other properties. This way, the partial derivatives of the outputs with respect to the density can not be computed analytically. They are computed numerically through finite difference methods. The new component would allow for faster optimizations by using the other material properties (outputs of the new component) as inputs for the other components with analytic partial derivatives.

Another goal of this task is to get to know the OpenMDAO methodology and implementation by creating my first component.

4.4 Task 4: Set different materials for different parts of the wing

Linked to the third task (*Task 3: Turn material function into OpenMDAO component*), it would be useful to allow the optimization to use different materials for the various parts of the wing (spars and skins) in order to obtain better results.

This approach makes possible to reach a better solution in terms of CO_2 footprint and weight. The findings should also be more accurate. On the other hand, the more design variables the model have, the longer the optimization process takes to complete. So, allowing the optimization to choose two different materials, one for the spars and one for the skins, results in an increase in the design variables by one (a new density for the second material).

4.5 Task 5: Introduce a more complex buckling model

Presently, a very simple buckling model is used in [1], taking into account the skins as rectangular flat plates only under axial compression. Considering curved plates under combined axial compression and shear a slightly more complex model is achieved, as the one explained in [2]. It is also said in [2] that: "[...] a long curved plate under shear buckles at a stress in close agreement with the theoretical value derived from linear theory. However, [...] the action of axially compressed long curved plates departs appreciably from the predictions of linear theory. Consequently, a linear analysis of buckling under the combination of these loads would be unconservative". For this reason, they recommend the use of empirical data.

The idea is to calculate the stresses of axial compression and shear separately (simple loading), and then, impose a parabolic interaction between them. This parabolic interaction is given by the expression 1 proposed in [2]. This expression fits very well with the experimental results as shown in Figure 1.

$$R_S^2 + R_C = 1 \tag{1}$$

where R_S and R_C are the stress ratios for shear and axial compression, respectively. The stress ratios are defined as the ratio of the stress at buckling under combined loading to the buckling stress under simple loading.

It is important to note that, as the finite element method (FEM) approach that OpenAeroStruct implements (explained in [3]) uses spatial beams elements, it is not viable to consider a FEM model for buckling. This is why analytical methods are the best option since they are sufficiently accurate and computationally more efficient, which is necessary to obtain faster optimizations.

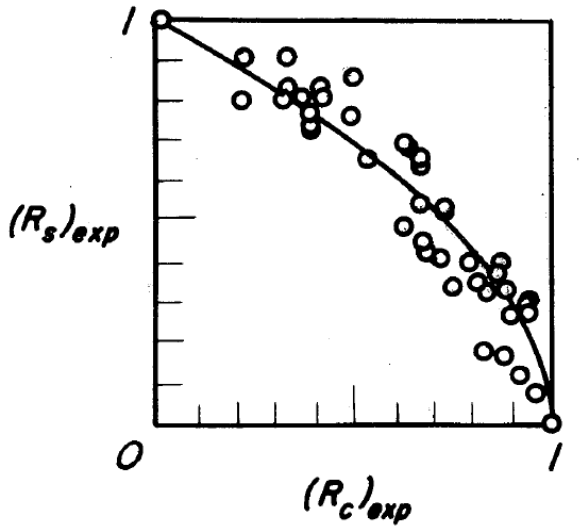

Figure 1: Comparison of test data with parabolic interaction curves for simply supported, curved plates under combined shear and axial compression. [2]

4.6 Task 6: Add engines as point masses

This task is also easy-to-implement since this function is already implemented in OpanAeroStruct. However, it can lead into a more realistic model. The principle is to add loads from point masses on to the wing structure. The loads caused by the point masses are transferred to the structural nodes based on the nodes' proximity to the point masses.

In [1], a comparison is made with the Facebook's single-boom HALE presented in [4], that has two symmetrical engines. Thus, two point masses will be added, considering the distance from the HALE plane of symmetry as a new design variable to optimize. The aim is to reduce the bending moment on the wing due to lift. The mass of each engine will be considered equal to the one in the FB HALE [4].

4.7 Task 7: Model a two dimensional discrete gust

As it is mentioned before, in [1], the HALE is compared with the Facebook's single-boom HALE, displayed in [4], in order to validate the framework. One of the main differences in the results presented in [1] is the aspect ratio, that is much larger than the one in [4]. As said in [1]: "These differences are mainly due to our not taking into account 1-cosine gusts in our framework [...] which would lead to more stress than a gust wall for a large-span wing".

In an attempt to solve this issue and reach more realistic results, a two dimensional gust will be taken into account. A simplification of the two-dimensional gust profile declared in [5] could be used with a shape given by:

$$w_g(y) = U_{de} \cos\left(\frac{2\pi y}{aL_{span}}\right) \tag{2}$$

where w_g is the gust velocity, U_{de} is the derived equivalent gust velocity, L_{span} is the aircraft span, a is a factor that is used to represent the spanwise distribution of the gust, and y is the spanwise coordinate.

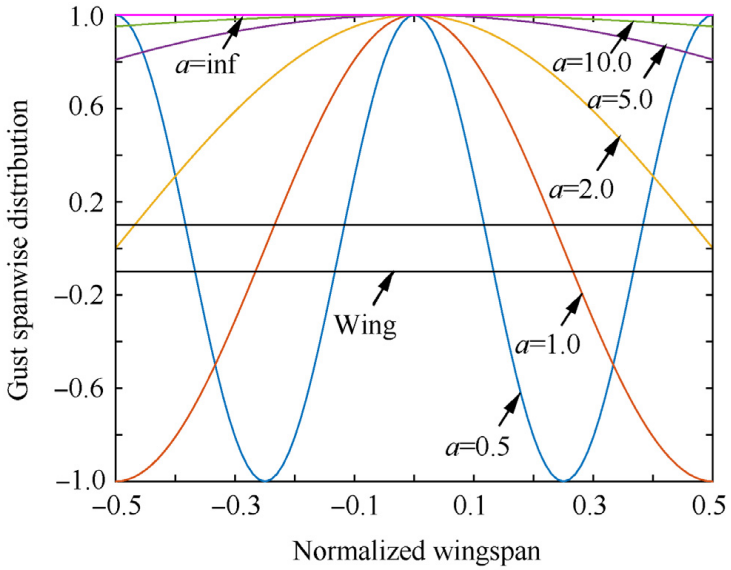

Figure 2: Spanwise distribution of cosine gusts (Normalized wingspan: y/L_{span}, Gust spanwise distribution: $\cos\left(\frac{2\pi y}{aL_{span}}\right)$). [5]

Figure 2 represents the spanwise gust distribution with different values of a. The two-dimensional gust degenerates into the one-dimensional shear gust, currently used in [1], when a tends to infinity.

With the intention of increasing the bending moment and penalizing large aspect ratios, another option could be a 1-cosine gust with the highest vertical velocity near the wing tips, depicted in Figure 3 and with a shape given by:

$$w_g(y) = \frac{U_{de}}{2}\left[1 - \cos\left(\frac{2\pi y}{aL_{span}}\right)\right] \tag{3}$$

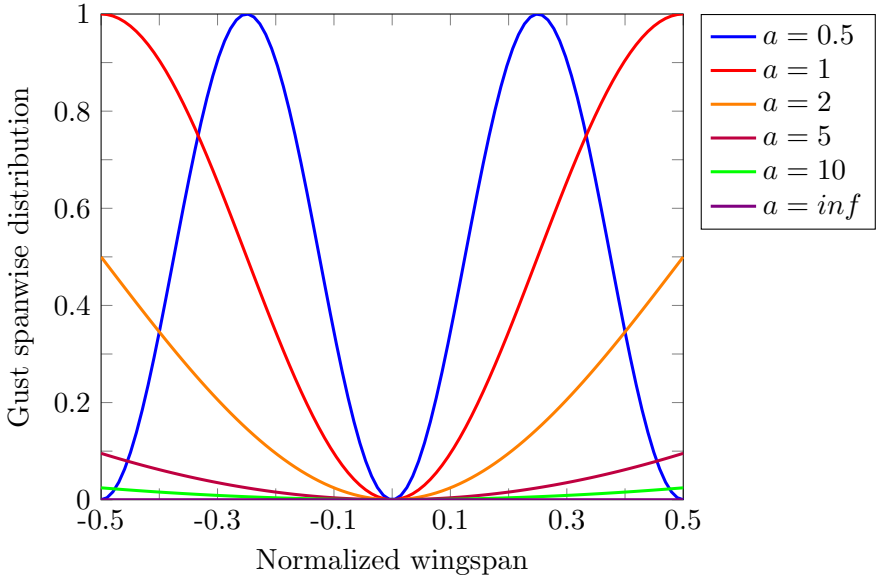

Figure 3: Spanwise distribution of 1-cosine gusts (Normalized wingspan: y/L_{span}, Gust spanwise distribution: $\frac{1}{2}\left[1 - \cos\left(\frac{2\pi y}{aL_{span}}\right)\right]$).

In order to have model conditions as realistic as possible, a gust field distribution orientated normally to the path of the aircraft, as presented in [5], could be added in the future. A two-dimensional gust field has only vertical velocity with time and spanwise amplitude variation. It could be given by the expression 4 shown in [5]:

$$\begin{cases} w_g(y,t) = U_{SA}\left[1 - cos\left(2\pi\left(\frac{Vt}{L_g}\right)\right)\right]\ \cos\left(\frac{2\pi y}{aL_{span}}\right) & 0 \leq t \leq L_g/V \\ U_{SA} = \frac{U_{de}}{2}(L_{span}/2L_g)^{\frac{1}{3}} \end{cases} \tag{4}$$

where L_g is the gust length, V is the velocity of the aircraft and t is the flight time.