Aerodynamic Design of a MAME UAV Wing Using High-Fidelity Numerical Tools

Rúben da Silva Gameiro
rubengameiro@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Portugal

November 2023

Abstract

The UAV market is currently very competitive, with the frequent release of new products and a wide range of solutions already available, forcing manufacturers to explore the design space faster and more efficiently than ever. A cost effective approach is to develop growth versions, improving an existing product with new technologies and design tools. Some of these tools include RANS based high fidelity computational fluid dynamics methods and discrete adjoint gradient-based optimization, which are used in this work on a numerical design framework to explore the aerodynamic shape optimization of a wing, as part of the development of a growth version of a UAV of a leading Portuguese manufacturer. A comprehensive aerodynamic analysis its current wing, including fuselage interference, is performed, followed by an optimization procedure to minimize drag subject to a prescribed lift coefficient constraint. To that end, three different starting geometries are considered and parameterized with common design variables, including twist and chord distributions, sweep, dihedral and airfoil shape. The use of two simple wings as starting geometries allowed the framework and set-up verification, with all optimizations considering different sets of design variables approaching an elliptical lift distribution, although not exactly considering the trade-offs needed between viscous and pressure drag. Challenges associated with the use of a complex wing geometry as a starting point are then addressed and optimizations considering the current UAV wing as a starting point are performed. Notably, a drag reduction of 4.5% is achieved considering all design variables.
Keywords: Aerodynamic optimization, gradient-based optimization, discrete adjoint method, wing design, computational fluid dynamics, free-form deformation

1. Introduction

As speeds reached by manned aircraft increased significantly during the 60's and the 70's, more focus started to be given to aerodynamic design. At this time, CFD was used primarily to obtain a better mission performance by increasing range and speed [1]. However, with the oil crisis, this approach shifted into a customer driven one, meaning that aircraft design started to be a balance between available technology and customer benefits [2]. In that sense, CFD brought several advantages to the design process by making it much more efficient. However, the design of a new vehicle can still take years, with several iterations between the different disciplines, such as structures, aerodynamics and propulsion, often on an intuition and trial and error basis to assess different possible vehicle configurations and shapes [1].

Nowadays, the UAV market in particular is experiencing fast growth and dynamic developments, which naturally attracts investors attention. Notably, the industry received a total investment of US6.96 billion last year [3], which also lead to an influx of new market players, forcing companies seeking external funding to differentiate themselves by offering innovative and unique products.

In this context, this work explores automatic aerodynamic shape optimization using high fidelity methods, in particular the MACH-Aero framework [4], developed by the MDOLab at the University of Michigan. The focus is the development of a growth version of the Tekever AR5, a Medium Altitude Medium Endurace UAV. Firstly, the aerodynamic analysis of the current generation wing is performed. After that, an optimization problem is set-up considering common design variables associated to different geometric parameterizations, with the goal of reducing drag for a prescribed lift coefficient. The resulting optimized geometries are analyzed and commented.

2. Aerodynamic design framework

The MACH-Framework is composed by the modules: ADFlow, a structured multi-block 3D CFD

solver [5] which also solves the adjoint method to compute the derivatives [6]; pyGeo a geometry manipulation tool that was specifically built for multidisciplinary optimization applications and allows the manipulation, parameterization and constraint handling of the geometric shape [7]; and pyOptSparse, which handles the constrained nonlinear optimization problem [8]. Besides the MACH-Aero framework, there is also the need for pre- and post-processing steps.

2.1. Meshing

The first step is to obtain a surface mesh from the CAD file. For that, it is possible to use a variety of different software. However, the chosen grid generator should be able to generate a multi-block structured 2D mesh, as it is the grid type accepted by the flow solver, and store it in CGNS (CFD General Notation System) format.

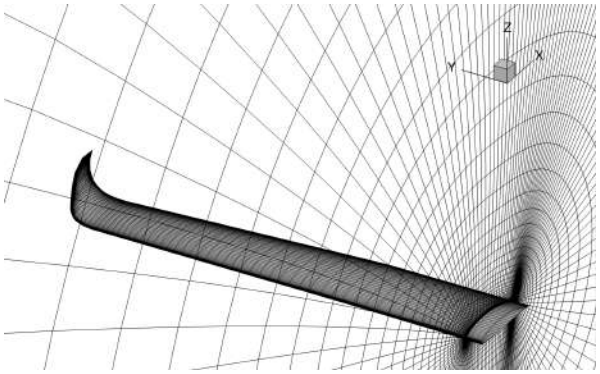

Fig. 1: Volume and surface meshes for the Tekever AR5 wing.

Having a surface mesh in the appropriate format, a volume mesh is then extruded using pyHyp, an hyperbolic mesh generator that also automatically applies the boundary conditions for a wing [9]. Figure 1 shows a volume mesh obtained with pyHyp, as well as the respective surface mesh. Only half domain is modeled due to the flow symmetry. To generate a volume mesh from a surface mesh, pyHyp requires the first layer height, the total height of the volume mesh and the number of layers to extrude.

The overset meshing approach is also used in this work to obtain a volume mesh including wing and fuselage to study the effects of the latter on the optimized wing, shown in Fig.2. Its creation involves three main steps: volume mesh generation for individual components; creation of a collar mesh at the intersection region; and generation of a background mesh and assembly of individual meshes into a single file. The Implicit Hole Cutting (IHC) algorithm is then applied automatically by ADFlow to assign overset connectivities and define the role of each cell (which can be compute, blanked, interpolate or flooded cells).

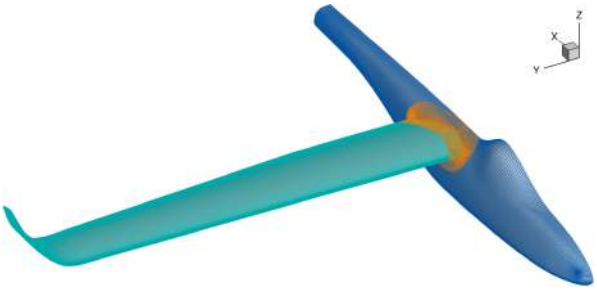

Fig. 2: Overset mesh for the Tekever AR5 wing and fuselage (considering symmetry plane).

2.2. Flow model

To compute the flow, ADFlow includes both inviscid models (Euler) and viscous models, which solve the RANS equations with different turbulence models available, including Spalart-Allmaras, Wilcox k-ω, k-τ, Menter SST k-ω and v2-f In this work, it is intended to optimize the wing for viscous flow, so RANS models will be used.

Regarding the turbulence model, comparisons for external flows around an Onera M6 wing concluded that the Spalart-Allmaras and SST k-ω turbulence models gave the closest results to the experimental data [10]. The flow solution using ADFlow with Spalart-Allmaras turbulence model was also compared with two other solvers and to experimental data for the CRM wing, and successful code validation was achieved [11]. Considering this and the fact that only the Spalart-Allmaras turbulence model is differentiated in the ADFlow code, makes it the most suited for the optimization of a wing. Spalart-Allmaras is a linear eddy viscosity model, thus uses Boussinesq assumption for the constitutive relation [12]. Furthermore, it is considered a low Reynolds number model, so an $y+$ value of approximately one is enough to obtain numerical uncertainties of friction resistance coefficients smaller than 1% [13].

The flow conditions are defined in ADFlow by instantiating the AeroProblem class from baseclasses. Multiple formulations could be used but, for this work, the flow condition is defined using the angle of attack α, flight altitude h, and Mach number Ma of the undisturbed flow. From these parameters the complete flow state information is obtained from the 1976 U.S. Standard Atmosphere [14]. Furthermore, wing reference area and chord should be passed as inputs to the AeroProblem for the computation of the aerodynamic coefficients.

2.3. Geometric parameterization

Using the MACH-Aero framework, there are two main options for geometry parametrization: CAD-

based and free-form deformation (FFD) based, which uses a box that completely embeds the surface mesh and that will be referred in this work simply as the FFD box. The surface mesh nodes are then mapped to the FFD box with an $\mathcal{R}^3 \rightarrow \mathcal{R}^3$ mapping, determined by performing a Newton search. After that, surface mesh nodes can be deformed by performing deformations on the FFD box nodes, given their mapping. The surface mesh deformations are then used to perturb the volume mesh using an hybrid algebraic-linear-elasticity mesh perturbation scheme [7] due to its resulting high grid quality and low computational effort. The FFD approach is used given that it is generally easier to set-up and use with an already existing geometry, has freedom to parameterize multiple design variables easily and allows a more efficient sensitivity computation.

The nodes of the FFD box may be displaced individually in any spatial direction as local design variables in that case, an example of which is given in Fig.3 where vertical displacements of FFD nodes are used to modify the airfoil shape.

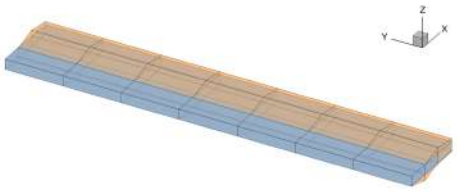

Fig. 3: FFD box deformation for airfoil shape.

However, pyGeo also allows to create relations between node displacements in such a way that they are all affected by a smaller set of design variables, allowing the geometry parameterization with meaningful global design variables such as taper or twist.

To create a global design variable using pyGeo, in addition to the FFD box, a reference axis is necessary. It can be created by specifying its direction and relative position in the FFD box using the fraction of two specified directions. For a typical wing optimization problem, the user should create an axis at the wing quarter-chord in the direction that follows the wingspan. It is important to note that the wing should be centered within the FFD box. The reference axis is used to project all the FFD nodes into it. The points then become rigidly linked to the refence axis, so any deformation on the FFD axis has an effect on several FFD points at once. A deformation on the FFD axis may be of three elementary types, in global x, y or z: displacement, which allows to move reference axis points to a new location and may be used to create dihedral or sweep; rotations, that can be used to create a twist distribution, but are also useful to keep the airfoil sections perpendicular to the reference axis when displaced, thus effectively keeping the same airfoil; and scaling, mainly used for chord distribution as shown in Fig.4 where, besides the obvious scaling in the streamwise direction, a scaling in the vertical direction is also applied to maintain the airfoil thickness to chord ratio constant.

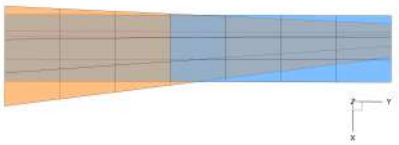

Fig. 4: FFD box deformation for chord distribution.

Global design variables can have as many degrees of freedom as the number of points in the reference axis but it is also possible to parameterize variable distributions with smaller sets of variables: for example, a linear twist distribution can be parameterized with only the slope as the design variable, provided the twist at the wing root is fixed.

When the starting geometry is not planar and, particularly a winglet is present, parameterizing it with meaningful design variables becomes more challenging, as the local reference axis of each FFD section are not aligned with the global ones. For local design variables, pyGeo has a function to displace them using the local sectional reference frame. However, as global design variables are user-built with elementary operations, attention must be paid to their definition.

For twist, the rotation needs to applied around the reference axis and so, the twist angle is projected to the direction vector at any section i, obtained with reference axis points $i+1$ and $i-1$.

For dihedral and sweep, in addition to the displacements, there is the need to rotate the FFD sections to keep them perpendicular to the reference axis. Figure 5 schematizes the process to compute the additional rotation needed, $\Delta\gamma$: the orientation of the section before deflection, γ_{bd} is computed using the coordinates of points $i-1$ and $i+1$ and the required orientation after deflection, γ_{ad}, is computed considering the coordinates of the points $i-1$ and $i+1$ and the vertical displacements, Δz, applied to each of them through the design variables, with the rotation angle to be applied then being the difference between both, given by Eq.(1), where design variables are represented with red.

$$\Delta\gamma_i = \underbrace{\text{arctg}\,\frac{z_{bd_{i+1}} + \Delta z_{i+1} - \left(z_{bd_{i-1}} + \Delta z_{i-1}\right)}{y_{i+1} - y_{i-1}}}_{\gamma_{ad}} \tag{1}$$

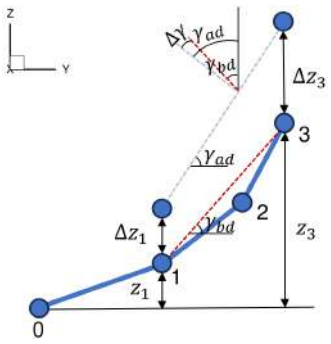

Fig. 5: Scheme depicting section rotation angles.

$$\underbrace{-\operatorname{arctg}\frac{\left(z_{bd_{i+1}}-z_{bd_{i-1}}\right)}{y_{i+1}-y_{i-1}}}_{\gamma_{bd}} \tag{1}$$

2.4. Constraints

pyGeo can also be used to set geometric constraints. It is possible to add different types of constraints, being the most common: minimum thickness constraints to ensure that there is enough room for structural components; minimum volume constraints to guarantee enough internal volume to carry a specified amount of fuel; curvature constraints, typically used to ensure manufacturability; and LeTe constraints to avoid shearing twist at the leading and trailing edge when local design variables are present.

Minimum thickness and volume constraints may be specified with either absolute or relative values in relation to the baseline design. Figure 6 shows an example of the minimum thickness (in black) and LeTe (in red) constraints used in this work. They are only enforced at certain user-specified locations and are independent of the FFD nodes. When the geometry is deformed, the change of the constraint location is handled automatically by pyGeo, which also takes care of the needed operations to use them as constraints in the optimization problem.

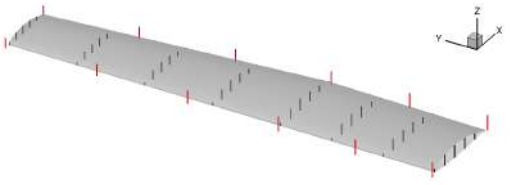

Fig. 6: Minimum thickness and LeTe constraints.

2.5. Optimization problem

With both geometric constraints and design variables defined in pyGeo, they need to be passed to pyOptSparse for the optimization process. Variables from the aero problem, defined using the baseClasses can also be passed as optimization variables and an important example of that is the angle of attack. Non-geometric constraints can also be created by user defined functions. The most important non-geometric constraint, when optimizing for cruise flight, is the lift constraint, which should match the weight coefficient to sustain flight. In a similar way, it is possible to define the function or combination of functions to be used as the optimization objective. In this work, the optimization objective will be the drag coefficient minimization.

With the design variables, usually defined by pyGeo parameterizations, constraints from both the geometry and the AeroProblem and the objective function passed to the optimization problem, the optimization set-up is effectively complete and ready to run. Given that pyOptSparse sets up the optimization in a problem oriented manner, the problem statement in the standard form, presented in Eq.(2) for this particular case, is easily translated to the optimization framework.

$$\begin{array}{ll} \text{minimize} & C_D \\ \text{w.r.t.} & \alpha \\ & \gamma \text{ (twist)} \\ & c \text{ (chord)} \\ & \Lambda \text{ (sweep)} \\ & \Gamma \text{ (dihedral)} \\ & \text{airfoil shape} \\ \text{subject to} & C_{L_{cruise}} = C_{L_{prescribed}} \\ & S_{\text{proj}} = S_{\text{prescribed}} \\ & \text{t (thickness constraint)} \end{array} \tag{2}$$

When running, the CFD solver computes the solution for the baseline geometry and the adjoints, which is then fed back to the optimizer. The optimizer uses this information to update the value of every design variable and fed them back to pyGeo, which deforms the volume mesh. The new volume mesh is solved by ADFlow again and the process is repeated until optimality convergence is achieved.

pyOptSparse can use different optimizers, including both gradient based and gradient-free ones. Both have been used in the past for aerodynamic shape optimization. Comparative studies of both methods were performed in [15], where it was concluded that both can converge to a solution, but gradient-free methods are considerably more computationally expensive, with the difference increasing dramatically as the number of design variables increases. In [16], an example of gradient-based was performed where a multi-component aerodynamic optimization for a wing propeller coupling was solved using gradient-based methods within the OpenMDAO/MPhys framework.

Due to the large set of design variables used in this work, the gradient-based methods are the best option. Furthermore, the functions used are smooth and C_1-continuous, as the previously presented papers have shown.

The SLSQP (Sequential Least Squares Quadratic Programming) is used due to its open source nature and robustness. It solves constrained nonlinear optimization problems, which corresponds to the nature of the problem studied in this work, using the Han-Powell quasi-Newton method with BFGS update of the B-matrix and an L1-test function in the step length algorithm [17].

To compute the gradients needed for the optimization algorithm, given that in aerodynamic shape optimization problems the outputs are often reduced to the drag coefficient but multiple design variables are used as inputs, the adjoint method is the most suitable option for aerodynamic shape optimization and is the one used in the MACH-Aero framework.

3. Baseline wing aerodynamic analysis

3.1. Operating conditions

The performance of the current Tekever AR5 wing is first assessed. Its cruise conditions are summarized in Tab.1.

Tab. 1: Performance parameters of Tekever AR5 [18].

Cruise speed	U_∞	$100km/h$
Cruise altitude	h	$1000ft$
Max. Take-off Weight	$MTOW$	$180kg$
Endurance	E	$20h$

Tekever provided the wing lift coefficient $C_{L_{wing}} = 0.8932$, considering the projected area of the AR5 wing, $S_{wing} = 2.1691m^2$, and the cruise conditions presented in Tab.1. The trim angle of attack is found through optimization with a single design variable and a prescribed lift coefficient as constraint.

For cruise, a Reynolds number $Re = 1.1e6$ is obtained, which is moderate, indicating that turbulent flow will be dominant. The effects of viscous forces are smaller than those caused by pressure forces, but they should not be ignored. Mach number is $Ma = 0.08164$, which is very low, so no transonic effects are to be expected.

For the simulation, the stopping criteria is a relative convergence of the residuals L2-norm to $1e{-}6$.

3.2. Grid convergence study

A grid convergence study is performed by simulating the Tekever AR5 wing at an angle of attack of 1.5°. Starting from a grid with 7.5 million elements, a coarsening ratio of 1.15^3 between grids is successively applied. Sufficient grid refinement is important not only for analysis accuracy but also for optimization, namely chord distribution optimization, during which the mesh in the streamwise direction will get compressed or enlarged together with the geometry, effectively changing the elements local spacing. All grids have a greater element clustering near the leading and trailing edge in the streamwise direction, as well as near the tip in the spanwise direction. Since no wall functions are used, the Spalart-Allmaras turbulence model requires an $y+ \approx 1$, which is satisfied.

Figure 7 shows the evolution of the total drag coefficient and the viscous and pressure components, as well as the lift coefficient. It is possible to observe that the drag coefficient is converging as the grids are refined, with both viscous and pressure components following a similar evolution. For the lift coefficient, the error is minimal throughout the tested range, with the relative error being smaller than 0.4% between the finer and coarser grids. However, for the drag coefficient errors are higher, at 8.4% between the finer and coarser grid. To achieve an error below 1% in relation to the finer grid, a mesh with 4.96 million elements would be required.

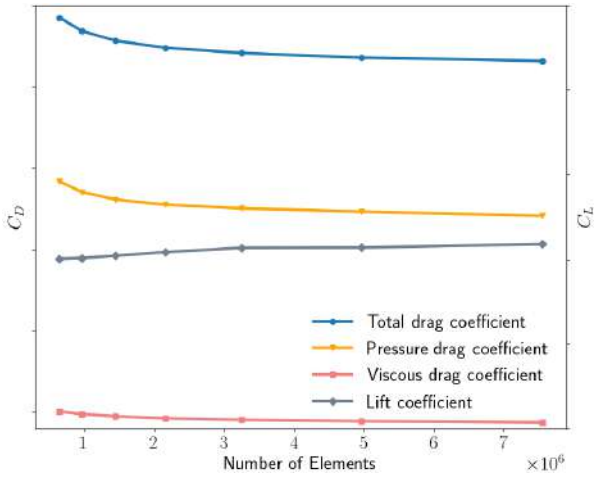

Fig. 7: Lift and drag coefficients variation with grid size.

However, the increase in CPU time needed for the finer grids is very noticeable, with the more refined one taking more than 12 hours (CPU time) to converge running on four cores of a CPU with a clock speed of $4.5GHz$ and the one with 4.96 million elements taking almost six hours (CPU time), which is deemed far too expensive for optimization purposes. For this reason, the grid with 1.45 million elements is selected taking into account that, although the drag error is 3.9% when compared to the finer grid, it converged in 20% of the time required for the grid with 4.96 million elements. The absolute drag values resulting from the optimiza-

tion process are not expected to be accurate, but as long as the grid is fine enough to capture relevant physical phenomena, it is expected that it leads the optimization to the correct trend considering relative gains from the starting point. To ensure this, the final geometry is verified using a finer grid. A linear increase in the required RAM is also verified at around $1.7GB$ per $100,000$ elements.

3.3. Cruise performance

The pressure distribution over the surface can be observed in Fig.8 for an angle of attack of 1.9°, which produces the prescribed wing lift coefficient for cruise conditions. It is possible to observe that, on the upper surface, pressure is higher closer to the leading edge and in the winglet region and, in this condition, the flow remains attached to the wing.

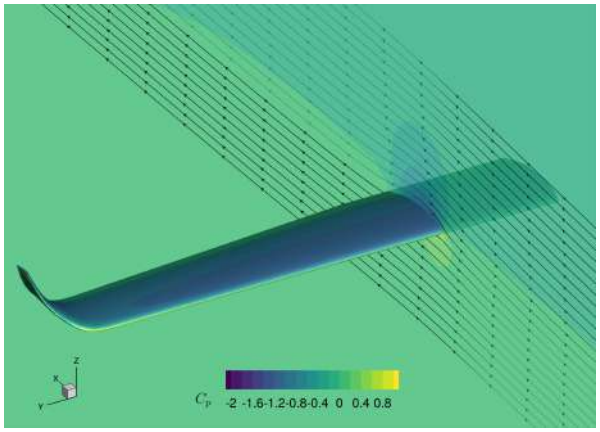

Fig. 8: Pressure distribution and velocity streamlines for cruise condition ($\alpha = 1.9°$).

To observe the behavior of the wing with angle of attack, a study is performed and the results presented in Fig.9. It is possible to see that C_L increases continuously with angle of attack until $\alpha \approx 14°$ and the variation is linear until $\alpha \approx 8°$, after which boundary layer separation starts to occur. In the linear region, $C_{L_\alpha} = 0.075/°$.

3.4. Influence of the fuselage

The wing with fuselage is also analyzed. The trim angle of attack for the isolated wing ($\alpha = 1.9°$) is also used here given that the manufacturer prescribed $C_{L_{wing}}$ was defined for an isolated wing.

Figure 10 shows a comparison of the pressure and lift distribution on a frontal plane between the isolated wing (represented in blue) and the wing with the fuselage (represented in orange). Major differences are observed on the upper side of the wing near the fuselage, with a total lift loss of 5% for the configuration with fuselage as a consequence. However, at 50% wingspan, the lift is just 1.5% smaller for the case with fuselage.

When these effects are considered, it is not expected that the optimized solution found with the isolated wing is the same as the one considering the full Tekever AR5 geometry near the wing-fuselage intersection. However, for the outer portion of the wing, where fuselage disturbances are minimal, similar results should be obtained.

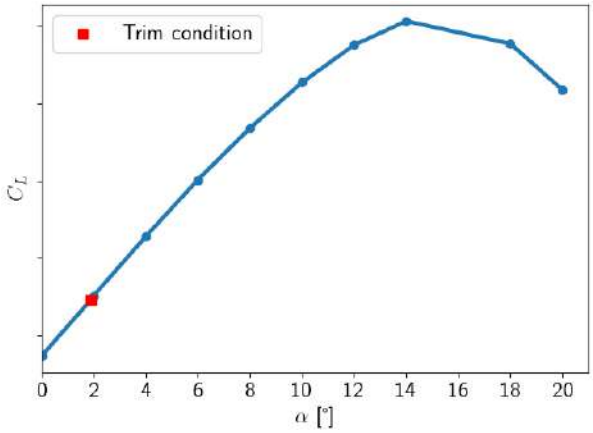

Fig. 9: Lift coefficient curve.

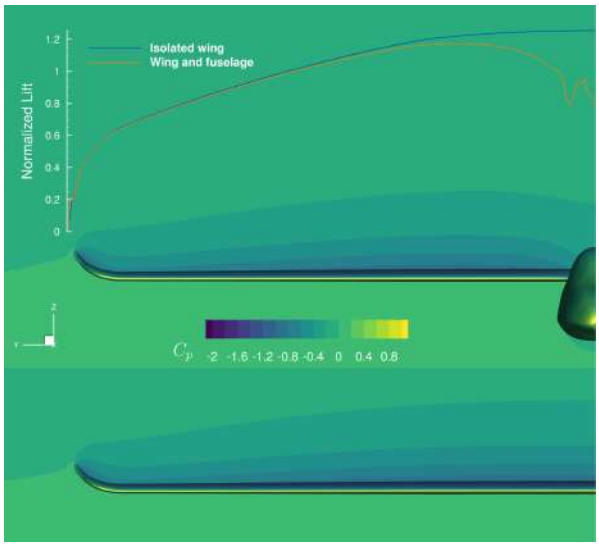

Fig. 10: Pressure coefficient on a frontal plane and lift distribution.

4. Wing aerodynamic optimization

In addition to the current Tekever AR5 wing, two simpler wing geometries are created, with the same projected wingspan and area as the original one. The first is a simple rectangular wing with a symmetric NACA 4-series airfoil with the same thickness-to-chord ratio as the original wing. The other is a planification of the Tekever AR5 wing, without twist and the winglet and a change in the trailing edge sweep of the outer section in order to keep the same projected area as the original wing.

The rectangular wing uses a simple parallelepiped-shaped FFD box, while both

the simplified and the original Tekever AR5 wings employ more complex FFD boxes that conforms to the surface geometry to allow an accurate definition of meaningful design variables, which would be difficult with a straight box. The reference axis is positioned at 25% and 50% in the streamwise and vertical directions, respectively, in all cases. Figure 11 shows the FFD box for the Tekever AR5 wing, with the control points represented with blue spheres and the reference axis represented in red. The links between control points and the reference axis are represented with blue rays.

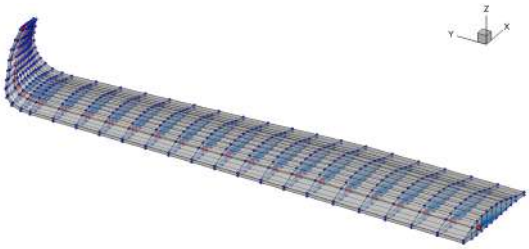

Fig. 11: FFD box used for the Tekever AR5 wing.

4.1. Rectangular wing

The results departing from a rectangular wing are presented in Tab.2.

Tab. 2: Optimization results for the rectangular wing as starting geometry.

Case	C_D
Starting geometry	Reference
Twist (7 FFD sections)	-2.60%
Chord (8 FFD sections)	-5.21%
Twist + Chord + Sweep	-5.18%

Considerable drag reductions are obtained for all cases, mostly due to a lift distribution approach to the elliptical one, as Fig.12 shows. For the twist optimization, however, the elliptical lift approach is not matched. A reason for this is that separation started being observed near the trailing edge for such high incidence angles, an effect that could not have been predicted with optimization considering inviscid flow.

4.2. Simplified Tekever AR5 wing

A summary of the optimization results for the simplified Tekever AR5 wing are found in Tab.3.

Although more modest than for the rectangular wing, drag reductions are also obtained in this case. Three different planform optimizations are shown. For chord optimization, the wing is divided into two sections: the inner one, where taper cannot be

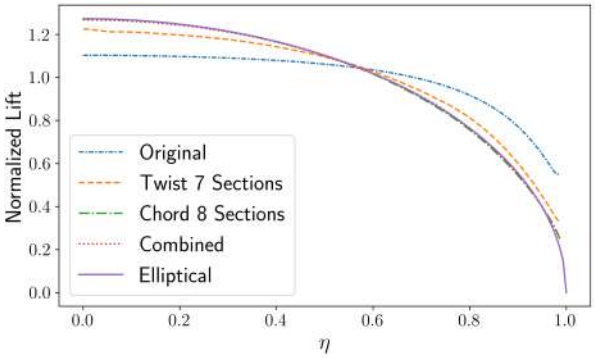

Fig. 12: Lift distribution obtained starting from the rectangular wing.

Tab. 3: Optimization results for the simplified Tekever AR5 wing as starting geometry.

Case	C_D
Starting geometry	Reference
Twist (7 FFD sections)	-0.80%
Chord (Linear 2 sections)	-0.11%
Twist + Chord	-1.16%
Airfoil ($8 \times 12 \times 2$ Control points)	-2.54%

changed and the outer one, where a linear chord variation is applied, resulting in two DoF, although with a projected area constraint there is effectively only one. As a result, the design space did not provided enough freedom for a significant drag reduction. However, the other two parameterizations provided greater freedom, resulting in more significant drag reductions. Figure 13 shows the lift distribution of the optimized geometry with all three parameterizations.

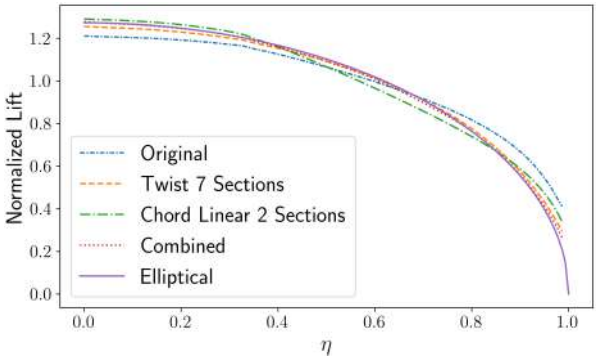

Fig. 13: Lift distribution obtained starting from the simplified Tekever AR5 wing.

For the combined optimization case, the obtained scale factors for chord distribution were 1.13 (at the inner section) and 0.67 (at the tip), and a twist

distribution is also applied, with its lowest value, at the wingtip, being $-4.14°$.

A variation in airfoil shape along the wingspan was also optimized, yielding the greatest reduction. The comparison between the original and optimized profiles reveal that the modifications along most of the wingspan mainly involved a decrease in thickness and a slight reduction in camber. Interestingly, at the wing tip, the optimizer converged to a symmetric airfoil, shown in Fig.14, minimizing the induced drag.

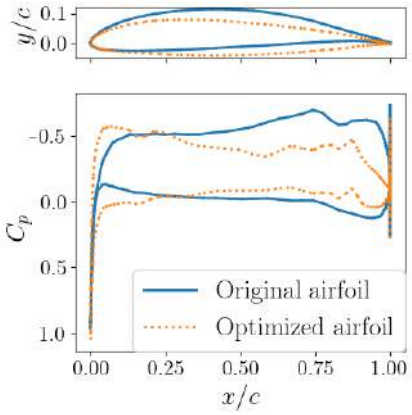

Fig. 14: Original and optimized airfoils at the tip starting from the simplified Tekever AR5 wing.

4.3. Tekever AR5 wing

Finally, the Tekever AR5 wing optimization results are presented in Tab.4. Here, a more refined FFD box is required to correctly capture the winglet shape, resulting in 28 total sections.

Tab. 4: Optimization results for the Tekever AR5 wing as a starting geometry

Case	C_D
Starting geometry	Reference
Twist (28 FFD sections)	-0.42%
Chord (28 FFD sections)	-1.02%
Dihedral (28 FFD sections)	-1.78%
Airfoil (28 x 12 x 2 control points)	-3.90%
All design variables	-4.52%

Optimizations of dihedral, airfoil and all design variables present significant drag reductions. For dihedral, those are achieved by maximizing the winglet height, reaching the design variable bounds.

For airfoil optimization, changes mostly include a reduction in thickness to the minimum allowable, as verified with the simplified Tekever AR5 wing, and a slight shape change near the trailing edge at the suction side, smoothing the pressure recovery. Figure 15 compares the original airfoil (blue) with the result for the optimizations considering only airfoil (orange) and considering all design variables (grey) at a middle section of the wing, and similar changes are applied in both cases. Significant differences are observed at the wingtip region which is deflected inwards, effectively reducing the airfoil camber at this region. A similar effect is observed with the simplified Tekever AR5 as a starting geometry, where a symmetric airfoil was obtained at this region. However, in this case, the airfoil is not aligned with the lift force and so its contribution to the lift production is smaller, allowing more drastic changes.

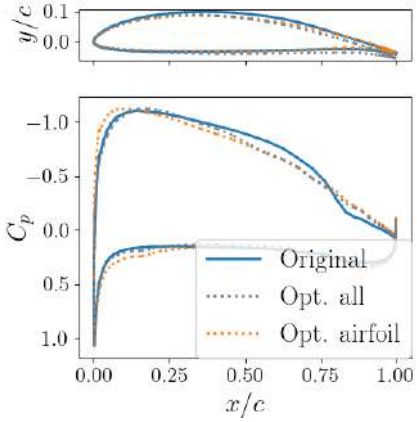

Fig. 15: Original and optimized airfoils for airfoil shape and all design variables cases starting from the Tekever AR5 geometry.

The complete optimization with all variables is performed with twist, dihedral and airfoil parameterization defined in the same way as the individual cases. Chord distribution is kept constant at the inner section and a decreasing monotonic variation constraint is applied at the outer one. The resulting geometry is shown in Fig.18, where visible changes in winglet height and smaller changes in twist can be observed.

Overall, even though changes occurred across the whole wingspan for all optimizations, the more visible deformations are at the winglet region, with the goal of reducing the wingtip vortex intensity. This is demonstrated in Fig.18 with the pressure distribution on a vertical plane closely behind the wing for the original geometry, the one with optimized airfoil and the one optimized with all variables. The reduction in the low pressure region between geometries is clear and correlates with the obtained drag reductions. The main exception to this behavior is chord optimization, which approximated an elliptic wing planform.

As discussed before, the grid used for optimization presented a significant drag error compared to

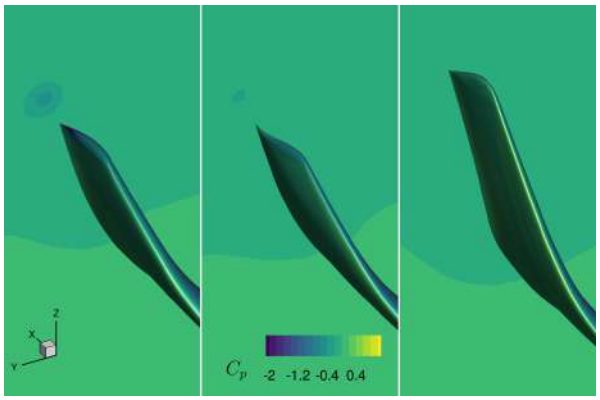

Fig. 16: Winglet pressure distribution of the Tekever AR5 wing (left), the wing with an optimized airfoil (middle) and the wing optimized with all design variables (right).

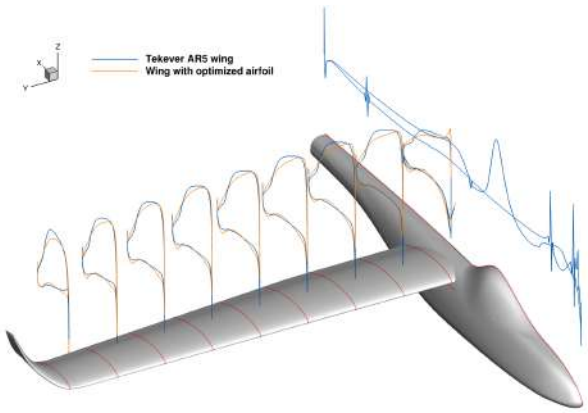

Fig. 17: Pressure distribution of the original geometry and optimized airfoil geometry.

the finer grid tested. Thus, the optimized geometry considering airfoil shape was verified with the finer grid for the found trim angle of attack. Comparing the optimized geometry to the initial one on this grid, a smaller drag reduction of 2.56% was achieved. However, lift for the optimized geometry on the finer grid was also higher than predicted with the coarser grid, resulting in an aerodynamic efficiency increase of 3.25%, much closer to the figure obtained for the coarser grid of 4.06%. Similar trends between grids were also verified with viscous and pressure drag, validating the use of a coarser grid to speed up the optimization process, provided that it is refined enough to capture relevant flow features.

The wing with an optimized airfoil was also mounted in the fuselage and analyzed. A drag reduction of 1.74% when compared to the initial geometry is obtained, being naturally lower than the one for the isolated wing in free-stream, which is to be expected given not only the interference effects but also the portion of drag from the fuselage.

Figure 17 compares the pressure coefficient between the original wing and fuselage assembly (in blue) with the one with the optimized wing (in orange) at several sections across the wingspan. Variations are as discussed before across most of the wingspan. However, near the fuselage intersection, a strong suction peak is verified on the wing with the optimized airfoil, representing an example of something that would probably look different if the optimization was performed with fuselage.

5. Conclusions

This work presents a study on the optimization of three different starting geometries: the rectangular wing with a symmetrical airfoil; the simplified Tekever AR5 wing; and the Tekever AR5. The results demonstrate notable drag reductions of up to 5.18% for the rectangular wing considering only planform optimization, 2.54% for the simplified Tekever AR5 wing with a variable airfoil optimization and 4.52% for the Tekever AR5 wing using all the design variables.

The use of the different starting geometries in this study highlights the potential for optimizing both naive geometries and geometries obtained from extensive design processes, thereby showing capabilities for both refinement of existing designs as well as the creation of new designs from scratch considering a specified flight condition.

Effects of grid refinement and fuselage interference are assessed before and after optimization, verifying the validity of optimizing an isolated wing with a coarser grid as a drag reduction of 1.74% is still obtained for the complete wing-fuselage assembly.

Suggestions for future work include multi-point optimization to assess other flight conditions, addition of the propulsion effects present in front of the wing and naturally, an evolution to an aerostructural optimization problem.

References

[1] J. Vos, A. Rizzi, D. Darracq, and E. Hirschel. Navier-Stokes solvers in european aircraft design. *Progress in Aerospace Sciences*, 38(8):601–697, 2002. doi:10.1016/S0376-0421(02)00050-7.

[2] P. Rubbert. CFD and the changing world of airplane design. In *19th Congress of the International Council of the Aeronautical Sciences*, volume 19, pages LVII–LVII, Anaheim, USA, September 1994. ICAS.

[3] Drone Industry Insights. Global drone market report 2022-2030, 2022. URL `https://droneii.com/product/`

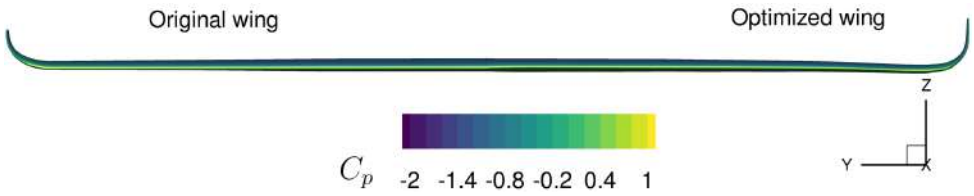

Fig. 18: Pressure comparison of the original and optimized wing with all design variables.

`drone-market-report`. Accessed: 2022-10-28.

[4] J. R. Martins. Aerodynamic design optimization: Challenges and perspectives. *Computers & Fluids*, 239:105391, 2022. doi:10.1016/j.compfluid.2022.105391.

[5] C. A. Mader, G. Kenway, A. Yildirim, and J. R. Martins. ADflow: An open-source computational fluid dynamics solver for aerodynamic and multidisciplinary optimization. *Journal of Aerospace Information Systems*, 17 (9):508–527, 2020. doi:10.2514/1.I010796.

[6] G. Kenway, C. A. Mader, P. He, and J. R. Martins. Effective adjoint approaches for computational fluid dynamics. *Progress in Aerospace Sciences*, 110:100542, 2019. doi:10.1016/j.paerosci.2019.05.002.

[7] G. Kenway, G. Kennedy, and J. R. Martins. A CAD-free approach to high-fidelity aerostructural optimization. In *13th AIAA/ISSMO multidisciplinary analysis optimization conference*, Fort Worth, USA, September 2010. AIAA. doi:10.2514/6.2010-9231.

[8] N. Wu, G. Kenway, C. A. Mader, J. Jasa, and J. R. Martins. pyOptSparse: A Python framework for large-scale constrained nonlinear optimization of sparse systems. *Journal of Open Source Software*, 5(54):2564, 2020. doi:10.21105/joss.02564.

[9] N. Secco, G. Kenway, P. He, C. A. Mader, and J. R. Martins. Efficient mesh generation and deformation for aerodynamic shape optimization. *AIAA Journal*, 59(4), 2021. doi:10.2514/1.J059491.

[10] M. Shah, B. Basuno, and A. Abdullah. Comparative study on several type of turbulence model available in Ansys-Fluent software for Onera M6 wing aerodynamic analysis. *Journal of Advanced Mechanical Engineering Applications*, 1(1):9–19, 2020. doi:10.30880/jamea.2020.01.01.000.

[11] J. G. Coder, T. H. Pulliam, D. Hue, G. Kenway, and A. J. Sclafani. Contributions to the 6th AIAA CFD drag prediction workshop using structured grid methods. In *55th AIAA Aerospace Sciences Meeting*, Grapevine, USA, January 2017. AIAA. doi:10.2514/6.2017-0960.

[12] P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. In *30th Aerospace Sciences Meeting and Exhibit*, Reno, USA, January 1992. AIAA. doi:10.2514/6.1992-439.

[13] L. Eça, F. S. Pereira, and G. Vaz. Viscous flow simulations at high reynolds numbers without wall functions: Is $y^+ \simeq 1$ enough for the near-wall cells? *Computers & Fluids*, 170:157–175, 2018. doi:10.1016/j.compfluid.2018.04.035.

[14] ICAO. U.S. Standard Atmosphere, 1976. Technical report, National Oceanic and Atmospheric Administration, National Aeronautics and Space Administration, United States Air Force, 1976.

[15] Z. Lyu, Z. Xu, and J. R. Martins. Benchmarking optimization algorithms for wing aerodynamic design optimization. In *Proceedings of the 8th International Conference on Computational Fluid Dynamics*, Chengdu, Sichuan, China, July 2014. ICCFD.

[16] H. Koyuncuoglu and P. He. CFD based multicomponent aerodynamic optimization for wing propeller coupling. In *AIAA SciTech 2023 Forum*, page 1844, National Harbor, USA, January 2023. doi:10.2514/6.2023-1844.

[17] D. Kraft. A software package for sequential quadratic programming. Technical Report 88-28, Deutsche Forschungs- und Versuchsanstalt fur Luft- und Raumfahrt, 1988.

[18] Tekever. Tekever>Meet Our UAS Models>AR5, 2022. URL `https://http://www.tekever.com/models/ar5/`. Accessed: 2022-10-30.