TÉCNICO LISBOA

Coupled Aircraft Design and Trajectory Optimization of an Electric UAV

Gonçalo Miguel Guardado Oliveira

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisor: Prof. André Calado Marta

Examination Committee

Chairperson: Prof. Filipe Szolnoky Ramos Pinto Cunha
Supervisor: Prof. André Calado Marta
Member of the Committee: Prof. Aurélio Lima Araújo

June 2021

To my parents.

Acknowledgments

Firstly, I want to express my sincere gratitude to Prof. André Marta for his continuous support, patience and guidance throughout the development of this work.

I want to thank my friends and colleagues for their friendship and for making these last years such an enjoyable experience.

A very special thank you to my parents for always supporting me. It would not have been possible without you.

Resumo

Nesta tese, realiza-se a otimização de trajetória e configuração da aeronove em simultâneo, com o objectivo de obter um UAV e um trajeto que cumpram uma missão com desempenho máximo.

A componente aeroestrutural é tratada pela ferramenta OpenAeroStruct. Esta é uma ferramenta de baixa fidelidade que usa um método de painéis com malha de vórtices e um método de elementos finitos para modelar o escoamento e o comportamento estrutural de superfícies sustentadoras. A ferramenta é modificada para acomodar propulsão elétrica e calcular parâmetros de performance pertinentes, bem como incluir a componente de trajetória, que é implementada atráves de um método de colocação.

A otimização é realizada através de um método de gradientes, para diversos objectivos, tais como minimizar a energia consumida durante a fase de subida, minimizar o tempo de subida até uma determinada altitude e maximizar o alcance, partindo de uma fase de voo cruzeiro. O problema de minimização de energia é também otimizado, individualmente, através da trajetória e da configuração, por forma a melhor quantificar os benefícios da otimização acoplada.

Verifica-se que a otimização acoplada permite minimizar a energia em mais 33% e 10.8%, relativamente às otimizações de trajetória e design isoladas, respetivamente. Observa-se também que o tempo de voo e a energia dispendida estão fortemente correlacionados, com as funções objectivo das duas soluções a variarem menos de 0.2% entre si. A otimização acoplada requer mais esforço computacional, sendo que o custo não aumenta linearmente com o tamanho do problema.

Palavras-chave: Otimização de trajetória, Projeto aeroestrutural, Otimização multidisciplinar, Método de gradientes

Abstract

In this work, coupled aircraft design and trajectory optimization is performed with the objective of producing a tailored UAV configuration and path that fulfil a mission at peak performance.

The aerostructural component is handled by the OpenAeroStruct framework. This is a low-fidelity tool that uses a vortex-lattice method and a 1D finite-element analysis to model lifting surfaces. An upgrade of the framework is developed to accommodate propulsion and its performance metrics, as well as trajectory dependent computations. An electric propulsive system is considered where the propeller is modelled using a relation derived from Blade Element and Momentum Theory. A direct collocation method is used for the trajectory component.

Gradient-based optimization is performed for different objectives, such as minimum energy consumed during climb, minimum time to climb to an arbitratry altitude, and maximum final distance starting from a cruise flight stage. The energy minimization problem is also optimized solely through trajectory or aircraft design, isolated, to better quantify the benefits of the coupled optimization.

We verify that the coupled optimization is able to further minimize energy in 33% and 10.8%, relative to the isolated trajectory and aircraft design optimizations. We also see that flight time and energy are strongly linked, with the objectives of both solutions varying less than 0.2%. We observe that the coupled optimization is computationally more expensive and that the cost does not increase linearly with problem size.

Keywords: Trajectory optimization, Aerostructural design, Multidisciplinary design optimization, Gradient-based optimization

Contents

List of Tables

List of Figures

Nomenclature

Greek symbols

α	Angle of attack
α_{eff}	Effective angle of attack
β	Angle of side-slip
Γ	Circulation
γ	Flight path angle
δ	Variation
δ_T	Throttle parameter
δ_t	Angle of stabilator
$\boldsymbol{\epsilon}$	Elastic strain vector
ζ	Collocation constraints
η	Efficiency
θ	Angle of twist
κ	Factor of induced-power loss
Λ	Angle of sweep
λ	Taper ratio, Lagrange multiplier
μ	Molecular viscosity coefficient
ν	Poisson ration
ρ	Density
$\boldsymbol{\sigma}$	Stress vector
Φ	Boundary term
ϕ	Boundary constraints

ψ Path constraints

$\boldsymbol{\omega}$ Angular velocity vector

ד Angle of dihedral

Mathematical operators

$\sum$ Summation

$\int$ Integral

Δ Variation

∇ Differential operator

∂ Partial derivative

$\cdot$ Dot product

$\times$ Cross product

Roman symbols

$\mathit{\AE}$ Aspect ratio

b Span

$\bar{c}$ Average chord

c Chord

C_D Coefficient of drag

C_L Coefficient of lift

C_M Coefficient of moment

C_{L_0} Coefficient of lift

C_{L_1} Coefficient of induced lift

D Drag

E Young's modulus

e Specific energy

$\boldsymbol{g}$ Gravitational acceleration

$\mathcal{H}$ Hamiltonian

$\boldsymbol{K}$ Stiffness matrix

$\mathcal{L}$ Lagrangian

L	Lift
$\boldsymbol{M}$	Moments vector
M	Mach number
Re	Reynolds number
$\mathbf{s}$	State vector
S	Surface area
$\boldsymbol{T}$	Thrust vector
t	Spar wall thickness
$\mathbf{u}$	Control vector
$\boldsymbol{u}$	Displacements vector
$\boldsymbol{V}$	Velocity vector
$\dot{x}, \dot{y}, \dot{z}$	Velocity Cartesian components
$\boldsymbol{W}$	Weight vector

Subscripts

∞	Free-stream condition
E	Earth frame
i, j, k, l, m, n	Computational indexes
x, y, z	Cartesian components
0	Initial conditions
f	Final conditions
ind	Induced

Superscripts

’	Aircraft frame
*	Optimal
T	Transpose

Glossary

CFD	Computational Fluid Dynamics
CN	Coupled Newton
DNS	Direct Numerical Simulation
DP	Design Problem
DTP	Design and Trajectory Problem
EMF	Electromotive Force
FEA	Finite Element Analysis
FEM	Finite Element Method
LES	Large Eddy Simulation
LGL	Legendre-Gauss-Lobato
LGR	Legendre-Gauss-Radau
LG	Legendre-Gauss
MDA	Multidisciplinary Analysis
MDF	Multidisciplinary Feasible
MDO	Multidisciplinary Design Optimization
NLBGS	Nonlinear Block Gauss-Seidel
NLP	Nonlinear Programming
NS	Navier-Stokes
OAS	OpenAeroStruct
OCP	Optimal Control Problem
RANS	Reynolds Average Navier-Stokes
SQP	Sequential Quadratic Programming
TP	Trajectory Problem
UAV	Unmanned Aerial Vehicle

Chapter 1

Introduction

1.1 Motivation

There has never been a wider range of aircraft. The technological and scientific advancements made in the fields of aeronautics, avionics, materials, among others, allowed the development of many types of aircraft, which have also grown more complex and sophisticated. They differ in size, configuration, type of propulsion and the way they take off and land. Some examples are shown in Fig. 4.2.

The fixed wing airplane is the most common type as it has become the main option for commercial flight. There are many possible configurations just within this type of aircraft. There is the most traditional configuration with main wing and tail, but there is also canard, delta wing and tandem, for example. Then there are also seaplanes, adapted to land on water and the blended wing body aircraft.

The second most common type is helicopters. Their main advantage is that they take off vertically, which is useful in many applications, like transportation within cities, sea rescue and military combat. Configurations vary mostly on the approach to rotors. There is the traditional single rotor, coaxial rotors (seen in Fig. 1.1(a)) and tandem rotor. Then there are also tilt rotor aircraft (Fig. 1.1(b)), which are almost hybrid between helicopters and planes.

Unmanned aerial vehicles (UAV) can be any of the two types above, or some novel configuration, as not having passengers allows for different organization of components. The most advanced UAV have been used specially in military operations, but they are versatile, built in many sizes and with different propulsive solutions.

(a) Kamov Ka-32A11BC [1] (b) Boeing V-22 Osprey [2] (c) General Atomics MQ-9 Italian Air Force [3]

Figure 1.1: Several aircraft for different missions.

Different missions have different requirements that need adaptation of existing aircraft or even new solutions. The search for innovative and more advanced solutions is what drives the development of new aircraft, or once the solution is established, the appearance of novel configurations, in the attempt of improving performance. This is where aircraft design optimization comes in. Manufacturers usually build new products as iterations of proven concepts but with modifications that improve certain characteristics of the previous model. In the case of commercial aircraft, for example, it is usually decrease fuel consumption, as a way to increase profit. In other applications, it might be a faster aircraft, a greater payload, etc. Whatever the objective, the aircraft itself can be improved through any of the different engineering disciplines involved, such as the propulsive system, aerodynamics and structures.

As for a mission objective, it can be further optimized through trajectory optimization. This consists in finding the best route and speed to complete the mission, whether it is simply going from A to B or heading to a certain zone and hovering around a target area. This part of the mission optimization is commonly done post design, so it is limited by the capabilities of the aircraft. Granted, the design is carried to satisfy some extreme flight limit conditions, but combining design and trajectory allows optimizing both configuration and mission to the fullest. The result is not necessarily a flexible solution, but rather very specific aircraft, tailored to fulfill a certain mission at peak performance.

1.2 Aircraft Design and Trajectory Optimization

Aircraft Design

Aircraft design does not have to, but often does, consider multiple disciplines, as it has more practical applicability and is therefore more relevant. This is because designing the aerodynamic shape of a wing without considering the structures that will sustain it might result in a body that is ill prepared to withstand the aerodynamic loads, so both need to be accounted for. Moreover, they are coupled, meaning their behaviours are codependent and changing design parameters usually affects both. Hence, it becomes necessary to perform Multidisciplinary Design Optimization (MDO). This approach incorporates all disciplines and solves them simultaneously, producing an optimal result that is better than if they were solved sequentially.

Trajectory Optimization

Flight is commonly separated in stages and the design is evaluated at the different stage conditions. However, this is not really trajectory optimization, as nothing is being changed in the trajectory itself.

Trajectory optimization is a subset of optimal control problems that consists in finding the sequence of positions and velocities that compose the path which optimizes a parameter of choice. Position and velocity are designated by state variables, which are *controlled* by inputs, also known as control variables. It has multiple applications, such as air traffic management, collision avoidance or simply improving mission performance.

1.3 Objectives and Deliverables

One of the aims of this thesis is to perform trajectory and multidisciplinary design optimization simultaneously, tailored for small fixed-wing electric UAVs, with conventional wing plus tail configuration.

Aerodynamic and structural disciplines are key elements of the optimization. They have been extensively studied, with many models developed to simulate their behaviours, making it important to review them.

Likewise, trajectory optimization is part of a larger field called optimal control. This is a topic that has been actively studied for many years and has yielded many techniques to approach problems. Therefore, it is of interest to study their applicability in trajectory optimization, understand their implementation and compare their performance.

The optimization field itself has undergone continuous development. Here, it is crucial to understand what methods are available to integrate several disciplines in the same problem, as well as what algorithms exist to solve the coupled analysis and the optimization problem.

The integration of disciplines is a key step to attain the objectives of this thesis. To do so, we will resort to the OpenAeroStruct (OAS) framework [4], which will be responsible for handling the aerostructural analyses, which make up the aircraft design component seen in the diagram of Fig. 1.2. Modifications will be made to accommodate the implementation of electric propulsion and the trajectory component. The extension of the framework will allow to perform coupled design and trajectory optimization, as well optimize each of the components individually.

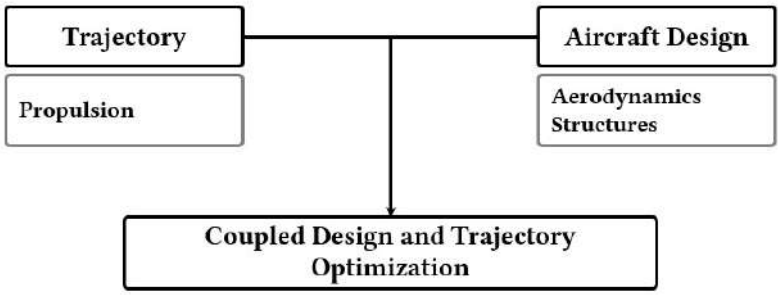


Figure 1.2: The two main topics of the thesis and the underlying disciplines combined in one optimization.

In summary, this thesis aims to:

- Upgrade the OAS framework to include electric propulsion and perform trajectory dependent computations;
- Perform coupled design and trajectory optimization;
- Demonstrate improvements of the combined optimization relative to the isolated optimizations;
- Explore the design space and test the framework by performing optimization for different objectives.

The contributions of this thesis are the upgraded framework and the study of coupled design and trajectory optimization illustrative test cases.

1.4 Thesis Outline

Chapter 2 starts by introducing the parameters that define the wing geometry. Then, the dynamics, aerodynamics, structural and propulsive models are described, culminating in the definition of the design optimization problem.

The theoretical background on optimal control is explored in Chap. 3. It delves into direct transcription methods, where a simple control problem is used to compare shooting and collocation methods, and culminates with the definition of the trajectory optimization problem.

Optimization and multidisciplinary integration are discussed in Chap. 4. It starts with the formulation of the combined design and trajectory optimization problem, followed by the presentation of multidisciplinary analysis and optimization architectures, as well as optimization algorithms background. Finally, the multidisciplinary framework implemented is shown and discussed.

The baseline conditions of optimization are defined in Chap. 5, followed by the discussion of the optimization results for different objectives.

Lastly, this work's achievements are summed up in Chap. 6 where suggestions for future developments are also made.

Chapter 2

Aircraft Design Physics

In this chapter, the models of the disciplines of analyses and their theoretical background are presented. We start by introducing the parameters that define the planform and the structural element. Then, we focus on the assumptions under which the equations of motion are built, followed by the aerodynamic, structural and propulsion models. Finally, the design optimization problem is formulated and the multipoint optimization is discussed.

2.1 Aircraft Geometry

Some geometric parameters of the aircraft are defined in this section. Starting with the planform parameters, let us initially consider a rectangular wing. The two larger sides are called leading and trailing edges, front and back of wing, respectively, and the other two are the wing tips. The distance between tips is called span b, and the distance between leading and trailing edges is the chord c.

In a rectangular wing, the chord is constant, however it is common to have variable chord along the span. In that case, it is designated by $c(y)$, where y is the spanwise distance from the root, such that $c(0) = c_{root}$ and $c(b/2) = c_{tip}$.

From these parameters, others can be obtained, such as the taper ratio,

$$\lambda = \frac{c_{tip}}{c_{root}} \tag{2.1}$$

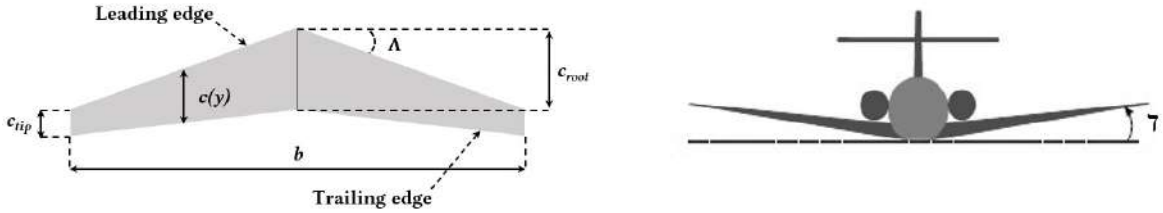
the wing area, where $\bar{c}$ is the average chord,

$$S = b\bar{c} \tag{2.2}$$

and the aspect ratio

$$\mathcal{AR} = b^2/S. \tag{2.3}$$

Another common geometrical feature are swept wings. The sweep angle, considered positive as shown in Fig. 2.1(a), can be determined at any fraction of the chord, but is here shown at the leading edge. Additionally, the wing can make an angle with the horizontal line, which is designated by dihedral angle and considered positive as shown in Fig. 2.1(b).

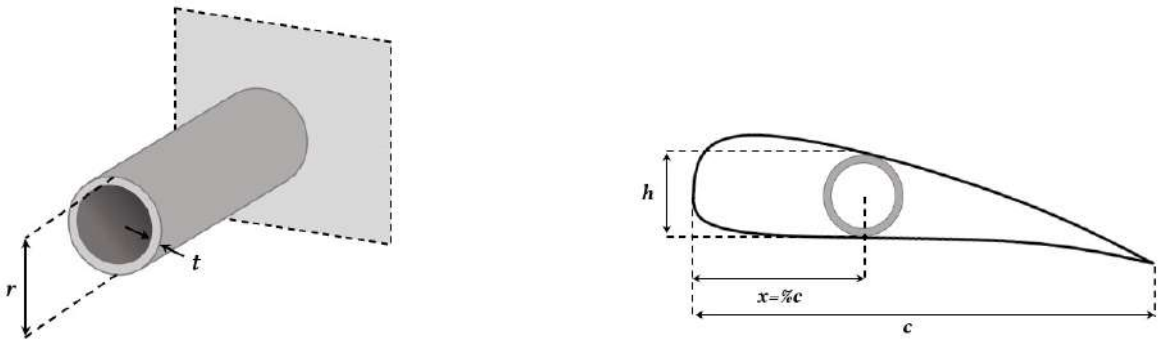


(a) (b)

Figure 2.1: Geometrical parameters of the wing.

In our model, the wing's spar is represented by a tube with a radius r and wall thickness t as shown in Fig. 2.2(a). The tube can be placed at any fraction of chord and this position limits the maximum diameter to the height h of the section. Its length l can be defined for the half wing as

$$l = \frac{b/2}{\cos \Lambda_{\%c} \cos \gamma} . \tag{2.4}$$



(a) (b)

Figure 2.2: Geometrical parameters of the spar.

2.2 Aircraft Dynamics

In this section, the assumptions under which the flight dynamics model is constructed are presented. The forces acting on the aircraft are defined and the equations of motion are derived in two frames.

Three reference frames are used in this work (Fig. 2.3). The aircraft frame $\mathcal{F}_A(x', y', z')$ has its origin at the center of mass of the aircraft and moves with it. The x' axis is the aircraft's longitudinal axis and the z' axis is perpendicular to it, pointing upwards; The flight path frame $\mathcal{F}_P(x, y, z)$ also has its origin in the center of mass of the aircraft and moves with it. It differs from the aircraft frame in that the x axis is aligned with the airspeed vector $\boldsymbol{V}_\infty$; Finally, the inertial Earth frame is designated by $\mathcal{F}_E(x_E, y_E, z_E)$. The Earth is assumed non-rotating and its curvature is neglected, so the origin is on the ground, the x axis is aligned with the local horizon and the z axis is normal to it pointing upwards, so the gravitational acceleration $\boldsymbol{g}$ is negative.

In this work, only the longitudinal component of flight is considered, so it is assumed there is no rotation about the x and z axis and no translational motion along y. Wind speed and sideslip angle (β) are also assumed

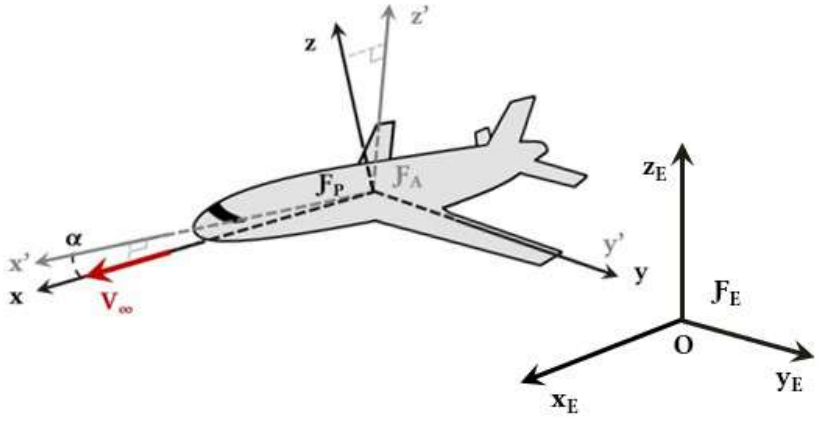
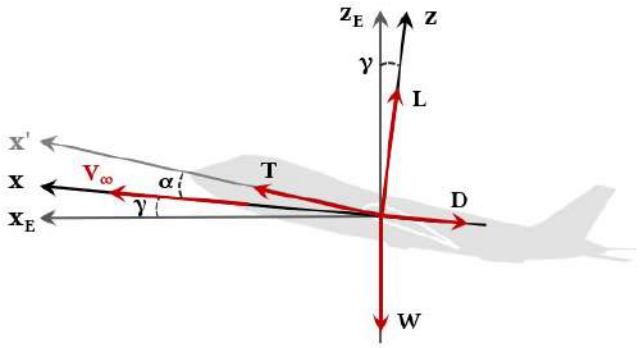


Figure 2.3: Inertial Earth frame $\mathcal{F}_\mathcal{E}$, aircraft frame $\mathcal{F}_\mathcal{A}$ and flight path frame $\mathcal{F}_\mathcal{P}$.

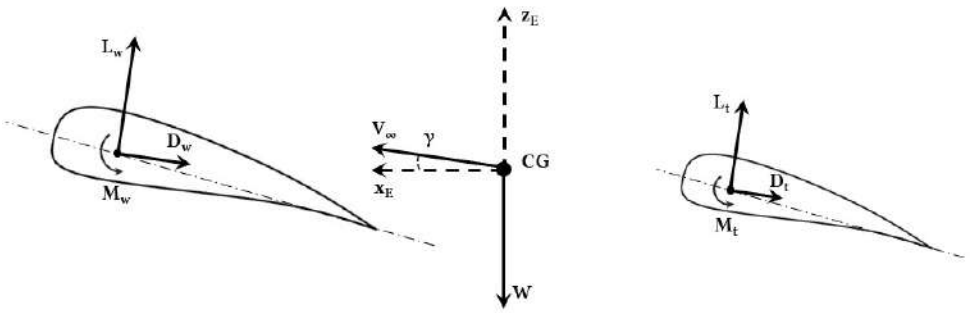
to be zero.

The forces acting on the aircraft are the gravitational force, $\boldsymbol{W} = m\boldsymbol{g}$, where m is the mass, the aerodynamic forces lift ($\boldsymbol{L}$) and drag ($\boldsymbol{D}$), and the propulsive force, thrust $\boldsymbol{T}$.

As it will be seen in detail in Sec. 2.3, the aerodynamic forces are calculated on the panels that compose the lifting surfaces, so $\boldsymbol{L}$ and $\boldsymbol{D}$ are obtained as the sum of the panels' contributions. When writing the equilibrium of forces, the aircraft is characterized by its center of mass, where the all forces are applied. The moments are naturally computed on a panel basis too, but for the sake of simplicity, they are here represented as a resultant force acting on each surface with an arm equal to the distance between the center of mass and the respective aerodynamic center. These simplifications are represented in figures 2.4(a) and 2.4(b), respectively.



(a) Representation of the forces acting on the aircraft's center of mass.



(b) Simplified representation of the moments about the center of mass of the aircraft.

Figure 2.4: Simplified diagram of forces and moments acting on the aircraft.

The derivation of the equations of motion is done in the flight path frame and in the Earth frame. The flight path frame is useful when considering levelled flight, as it moves with the aircraft and yields a simpler system. When flight is not levelled, or if different stages of a trajectory are considered, it is necessary to transpose the equations onto an inertial frame so that quantities can be calculated in the same reference system. The Earth frame is useful in that situation.

The derivation of the equations of motion in the Earth frame starts with Newton's second law relative to the translation of the center of mass,

$$\sum \boldsymbol{F} = m\dot{\boldsymbol{V}}_\infty \tag{2.5}$$

where the resultant of forces applied in the center of mass is

$$\sum \boldsymbol{F} = \boldsymbol{T} + \boldsymbol{L} + \boldsymbol{D} + \boldsymbol{W}. \tag{2.6}$$

Writing the forces in the two components of the Earth frame yields

$$\sum F_{x_E} = T\cos(\alpha + \gamma) - D\cos(\gamma) - L\sin(\gamma) = m\ddot{x} \tag{2.7a}$$

$$\sum F_{z_E} = L\cos(\gamma) + T\sin(\alpha + \gamma) - W - D\sin(\gamma) = m\ddot{z}. \tag{2.7b}$$

Similarly, the derivation of the equations of motion in the flight path frame also has Newton's second law as starting point

$$\sum \boldsymbol{F} = m(\dot{\boldsymbol{V}}_\infty + \boldsymbol{\omega} \times \boldsymbol{V}_\infty) \tag{2.8}$$

and breaking Eq. (2.8) into the two components of the 2D flight path frame leads to

$$\sum F_x = T\cos(\alpha) - D - W\sin(\gamma) = m\dot{V}_\infty \tag{2.9a}$$

$$\sum F_z = L + T\sin(\alpha) - W\cos(\gamma) = m\dot{\gamma}V_\infty. \tag{2.9b}$$

The moments equation is derived from the rotation about the center of mass,

$$\sum \boldsymbol{M} = I(\dot{\boldsymbol{H}} + \boldsymbol{\omega} \times \boldsymbol{H}) \tag{2.10}$$

with the resultant of moments about the center of mass given by

$$\sum \boldsymbol{M} = \boldsymbol{r}_T \times \boldsymbol{T} + \boldsymbol{M}_w + \boldsymbol{r}_w \times \boldsymbol{L}_w + r_w \times \boldsymbol{D}_w + \boldsymbol{M}_t + \boldsymbol{r}_t \times \boldsymbol{L}_t + \boldsymbol{r}_t \times \boldsymbol{D}_t. \tag{2.11}$$

Since flight is considered two dimensional and both the flow and the aircraft are symmetric over the longitudinal axis, there is only pitching moment. It is assumed that thrust is parallel to x' and aligned with the center of mass, so $\boldsymbol{r}_T \times \boldsymbol{T} = 0$. Additionally, the aircraft is considered to always be in a state of equilibrium of moments, as the time and spatial scales considered are large enough for the non stationary terms to be neglected.

$$\boldsymbol{M}_y = \boldsymbol{M}_w + \boldsymbol{r}_w \times \boldsymbol{L}_w + r_w \times \boldsymbol{D}_w + \boldsymbol{M}_t + \boldsymbol{r}_t \times \boldsymbol{L}_t + \boldsymbol{r}_t \times \boldsymbol{D}_t = 0 \tag{2.12}$$

2.3 Aerodynamic Model

There are many Computational Fluid Dynamics (CFD) models and methods that can be used for aerodynamic simulation. They vary in complexity and fidelity and as consequence in the computational cost too, which cannot always be supported by the available computational resources. For that reason, a compromise has to be made.

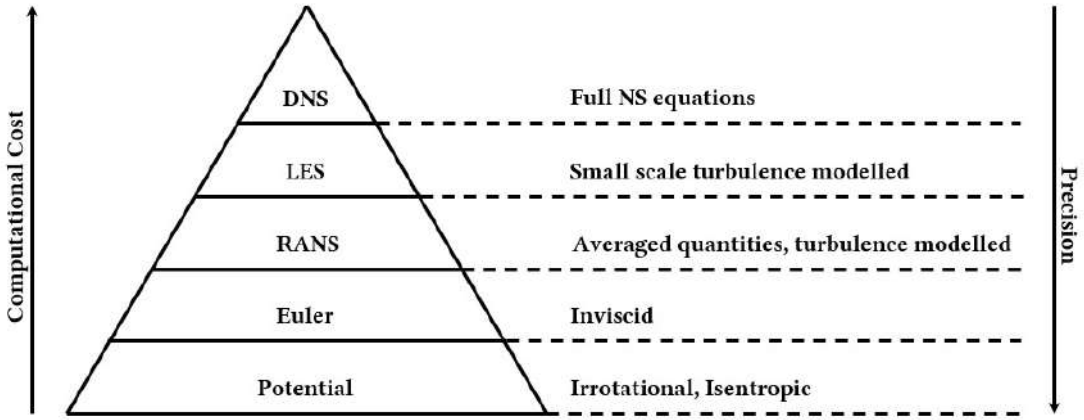


Figure 2.5: Graphical representation of the computational effort - fidelity/precision trade-off for different CFD approaches. Adapted from [5].

The trade-off between computational cost and fidelity is achieved through the simplification of the models used to describe the fluid dynamics, as represented in Fig. 2.5.

Direct Numerical Simulation (DNS) is the most accurate approach for simulating turbulent flows, as it solves the full Navier-Stokes (NS) equations. It uses a very fine mesh that captures all scales of the flow, from the smallest to the largest eddies [6], which makes DNS computationally very expensive and only applicable with high performance computer systems. With the appropriate mesh, time step and a numerical scheme that is designed to minimize dispersion and dissipation errors, an accurate solution for the NS equations is obtained. In this situation, the only errors are those introduced by the residual approximations incorporated in the numerical scheme and in the number-representation technology of the computing machine [7].

The level of detail DNS provides is not always necessary or the cost to obtain it can not be supported, hence approximations can be made. In Reynolds-averaged Navier-Stokes (RANS) approaches, all unsteadiness is regarded as part of turbulence and averaged out [8]. Turbulent variables are averaged through Reynolds decomposition [9] if the flow is assumed incompressible or through Favre decomposition [10] if assumed compressible. Equations are solved for these averaged quantities and the effect of instantaneous turbulent motion is modelled by a turbulence model, such as $k-\varepsilon$ [11], $k-\omega$ [12, 13], Spalart-Allmaras [14], SST [15] or v2f [16].

In Large Eddy Simulation (LES), the large scale motions of turbulent flow are computed directly whereas small scale motions are modelled, resulting in a significant reduction in computational cost compared to DNS. A spatial filter is applied to the velocity so that only large scale motion is captured for direct simulation. That is because large eddies provide more significant information for the flow simulation, as they are generally more energetic than the small scale ones and responsible for most of the momentum transfer and turbulent mixing. LES computes these eddies directly whereas RANS models them and for that reason, LES generally has higher fidelity.

Another simplification is the assumption of inviscid flow, in which case the viscosity effects are neglected and the shear stress terms vanishes, thus reducing the Navier-Stokes equations to the Euler equations. In flows far from solid surfaces, viscosity effects are negligible resulting in small induced errors. Close to surfaces, however, boundary layer effects will be lost, which means coarser grids can be used and less computational resources are needed, yet resulting in lower fidelity. A correction is usually made for the viscous drag contribution, which can be estimated through a skin friction model and form factor model for the added pressure drag. Compressibility effects and shocks are still captured, which is why Euler equations are used for flows at high Mach numbers.

The flow can be further simplified by considering the velocity field irrotational $\nabla \times \boldsymbol{V} = 0$. In that case there is a velocity potential ϕ, such that $\nabla\phi = \boldsymbol{V}$ and the governing equation becomes the Prandtl-Glauert equation, here written for subsonic flow [17]

$$\tilde{\nabla}^2\phi = (1 - M^2)\frac{\partial^2\phi}{\partial x^2} + \frac{\partial^2\phi}{\partial y^2} + \frac{\partial^2\phi}{\partial z^2} = 0. \tag{2.13}$$

If the flow can be considered incompressible, $M \longrightarrow 0$ and the equation becomes the Laplacian. Potential flow leaves out some important flow behavior such as separation, skin-friction drag, and transonic shocks.

Panel methods are numerical schemes that solve the Prandtl-Glauert equation by superimposing surface distributions of singularities over the panels. A Vortex-Lattice Method (VLM) is used in this work, which is a vortex based panel method that extends Prandtl's lifting line theory by superimposing multiple horseshoe vortices in the chord and span directions of the wing surface [18].

The horseshoe vortex, represented in Fig. 2.6, is composed by three filaments: the bound vortex bc and the trailing edge vortices, ab and cd. The latter extend to infinity, all with the same strength Γ, as stated in Helmholtz's vortex theorems [18].

The lifting surface is divided in m trapezoidal panels of length l, as the one delimited by the dashed line in Fig. 2.6, with two sides parallel to the freestream direction. Every panel has a control point placed on its certerline at $\frac{3}{4}l$ from the front of the panel and a bound vortex at $\frac{1}{4}l$.

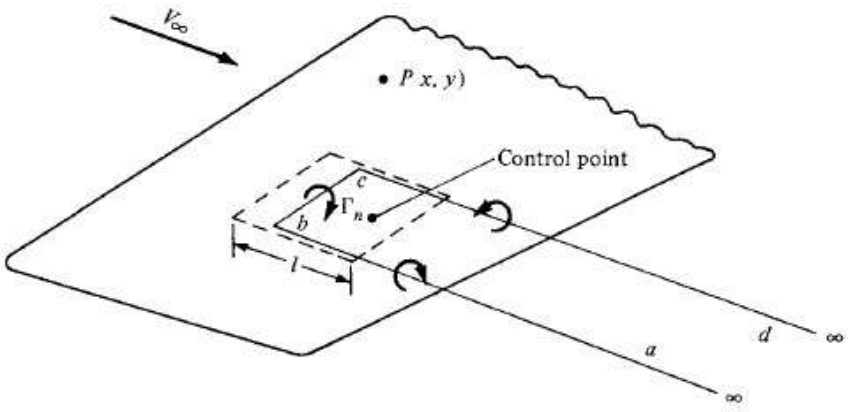


Figure 2.6: Representation of a horseshoe vortex [18].

Vortices induce a flow field in the surroundings. The velocity of this flow field on a point P at a distance $\boldsymbol{r}$,

due to a segment $d\boldsymbol{l}$ of the horseshoe vortex with circulation strength Γ is expressed by the Biot-Savart law [18],

$$d\boldsymbol{V}_{ind} = \frac{\Gamma}{4\pi}\frac{d\boldsymbol{l}\times\boldsymbol{r}}{r^3} = \Gamma\boldsymbol{a}. \tag{2.14}$$

The induced component of velocity on a control point is influenced by each filament of every vortex system, including of its own panel. In other words, the induced velocity on control point k due to panel j is the sum of the filaments' contributions,

$$\boldsymbol{V}_{kj} = \boldsymbol{V}_{kj}^{ab} + \boldsymbol{V}_{kj}^{bc} + \boldsymbol{V}_{kj}^{cd} = \Gamma_j(\boldsymbol{a}_{kj}^{ab} + \boldsymbol{a}_{kj}^{bc} + \boldsymbol{a}_{kj}^{cd}), \tag{2.15}$$

and the total induced velocity on that same control point is obtained by going through every panel,

$$\boldsymbol{V}_{k_{ind}} = \sum_{j=1}^{m}\Gamma_j(\boldsymbol{a}_{kj}^{ab} + \boldsymbol{a}_{kj}^{bc} + \boldsymbol{a}_{kj}^{cd}). \tag{2.16}$$

Adding the freestream velocity to this result gives the total velocity,

$$\boldsymbol{V}_k = \boldsymbol{V}_\infty + \boldsymbol{V}_{k_{ind}}. \tag{2.17}$$

An impermeability condition is imposed on every control point so that the normal velocity is zero,

$$\boldsymbol{V}_k \cdot \boldsymbol{n}_k = 0. \tag{2.18}$$

The linear system arises from combining equations (2.16), (2.17) and (2.18),

$$\boldsymbol{V}_\infty \cdot \boldsymbol{n}_k + \sum_{j=1}^{m}\Gamma_j(\boldsymbol{a}_{kj}^{ab} + \boldsymbol{a}_{kj}^{bc} + \boldsymbol{a}_{kj}^{cd}) \cdot \boldsymbol{n}_k = 0 \tag{2.19}$$

$$\iff A_{kj}\Gamma_j = -\boldsymbol{V}_\infty \cdot \boldsymbol{n}_k \tag{2.20}$$

where A is the aerodynamic influence coefficients matrix ($m \times m$).

Having solved this system for the circulation strengths of the horseshoe vortices, the Kutta-Joukowski theorem [18] is applied to compute the aerodynamic forces acting on each panel,

$$\boldsymbol{F}_k = \rho\Gamma_k(\boldsymbol{V}_\infty + \boldsymbol{v}_k)\times\boldsymbol{l}_k, \tag{2.21}$$

where $\boldsymbol{v}$ is the velocity at the center of the bound vortex and $\boldsymbol{l}$ is the bound vortex vector.

The aerodynamic force is a three dimensional vector computed in the aircraft frame. Because there is symmetry, the net force in the y' axis is zero. As shown in Fig. 2.7, the other two components are transposed onto the flight path frame, yielding the induced lift and drag, respectively

$$L = -F_{x'}\sin(\alpha) + F_{z'}\cos(\alpha) \tag{2.22}$$

$$D = F_{x'}\cos(\alpha) + F_{z'}\sin(\alpha). \tag{2.23}$$

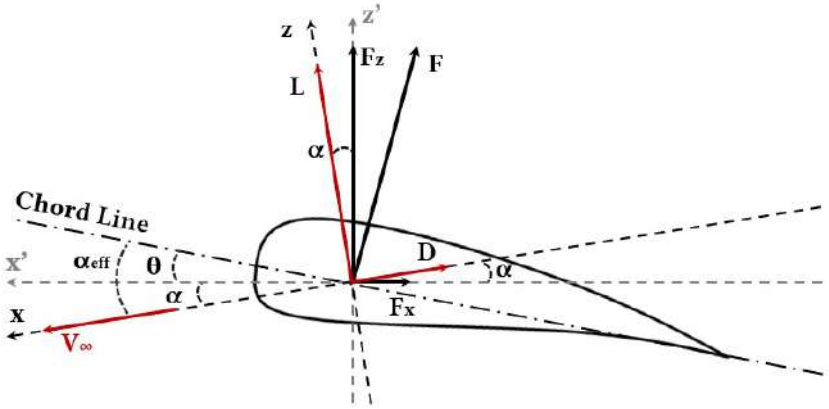


Figure 2.7: Decomposition of the aerodynamic force and wing's angle of attack.

The induced lift coefficient is given by

$$C_{L_1} = \frac{L}{\frac{1}{2}\rho S V_\infty 2} \tag{2.24}$$

and the total lift on the panel is obtained by adding the lift for $\alpha_{eff} = 0$, C_{L_0}

$$C_L = C_{L_1} + C_{L_0}. \tag{2.25}$$

Analogously, the induced drag coefficient is given by

$$C_{D_1} = \frac{D}{\frac{1}{2}\rho S V_\infty^2} \tag{2.26}$$

and the total drag is the result of adding the viscous drag and zero-lift drag contributions,

$$C_D = C_{D_0} + C_{D_1} + C_{D_v}, \tag{2.27}$$

where the viscous drag is computed using flat-plate-based estimates as in [19, Section 12.5.3]. This estimate is adjusted using a form factor, which accounts for pressure drag due to flow separation [4].

The angle α_{eff}, depicted in Fig. 2.7, is the effective angle of attack the airspeed makes with the chord line of the section that contains the panel. For the wing, it can be obtained as the sum of the incident angle of attack α, measured from the longitudinal axis of the aircraft to the airspeed, and the twist angle θ, measured from the longitudinal axis to the chord line. If twist varies along the span, the effective angle of attack can be written for any spanwise position y as

$$\alpha_{eff_w}(y) = \alpha + \theta(y). \tag{2.28}$$

The same principle applies for the tail's angle of attack. The difference is that we chose to define a tail with constant twist along the span, making an angle δ_t with the longitudinal axis, which is controlled during the flight. This way, the tail functions as a stabilator and its effective angle of attack is given by

$$\alpha_{eff_t} = \alpha + \delta_t. \tag{2.29}$$

2.4 Structural Model

Similarly to what is done in aerodynamics, the structures of an aircraft, in particular of the wings, have to be modelled and its behaviour simulated. This is essential to make sure they can withstand the stresses and loads they will be subject to and that the integrity of the aircraft is ensured. This is usually done using numerical techniques in which the continuum is discretized and the governing equations are solved locally for each element.

The discretization is an essential part of the simulation process because these problems are continuous, which makes it impractical to solve them numerically. The domain is discretized through methods such as the Boundary Element Method (BEM), Finite Volume Method (FVM), Finite Element Method (FEM) or Spectral Method, among others. In this work, FEM is used to evaluate the deformation of the lifting surfaces due to the aerodynamic forces.

The constitutive equation relates stress and strain and it can be written as Hooke's law for linearly elastic materials [20],

$$\boldsymbol{\sigma} = \boldsymbol{D}\boldsymbol{\varepsilon}, \tag{2.30}$$

where $\boldsymbol{\sigma} = \begin{bmatrix} \sigma_x & \sigma_y & \sigma_z & \tau_{xy} & \tau_{yz} & \tau_{xz} \end{bmatrix}^T$ and $\boldsymbol{\varepsilon} = \begin{bmatrix} \varepsilon_x & \varepsilon_y & \varepsilon_z & \gamma_{xy} & \gamma_{yz} & \gamma_{xz} \end{bmatrix}^T$ are the stress and strain vectors, respectively, and $\boldsymbol{D}$ is the elasticity matrix in stiffness form. In compliance form, $\boldsymbol{D}^{-1}$ can be written for isotropic materials in terms of the six stress and strain terms as [20],

$$D^{-1} = \frac{1}{E}\begin{bmatrix} 1 & -\nu & -\nu & 0 & 0 & 0 \\ -\nu & 1 & -\nu & 0 & 0 & 0 \\ -\nu & -\nu & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2(1+\nu) & 0 & 0 \\ 0 & 0 & 0 & 0 & 2(1+\nu) & 0 \\ 0 & 0 & 0 & 0 & 0 & 2(1+\nu) \end{bmatrix}, \tag{2.31}$$

where E is Young's modulus and ν is Poisson's ratio of the material.

FEM's underlying principle for computing the deformations is that the total potential energy, for a state of equilibrium, must be stationary for variations of the admissible displacements [20]. To apply this principle, the partial differential equation that governs the physical phenomena must be reformulated in a variational (weak) form, which is done by multiplying the equation by an arbitrary field and integrating over the element. Virtual work is a weak form of equilibrium equations in which the arbitrary function is a virtual displacement [21]. Its application, as in the literature [20, 21], results in a linear system that relates the distributed forces $\boldsymbol{F}$ and the displacements $\boldsymbol{u}$,

$$\boldsymbol{K}\boldsymbol{u} = \boldsymbol{F}, \tag{2.32}$$

where $\boldsymbol{K}$ is the stiffness matrix.

The equivalent spatial beam used in this work is a combination of truss, torsion and beam elements that model the behaviour due to axial, torsional and bending loads. Each spatial beam element has three translational and three rotational DOF for each of its two nodes. It is the result of the superimposition of four elements: one bar with axial displacements, one torsion element with rotation about its longitudinal axis, and two beams with

translational and rotational deformation (bending).

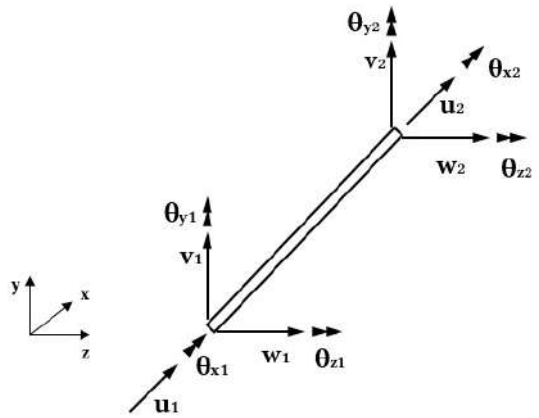


Figure 2.8: Spatial beam with 6 DOF per node. Adapted from [4].

The truss element is represented in Fig. 2.9. It has one translational DOF per node for the axial displacement.

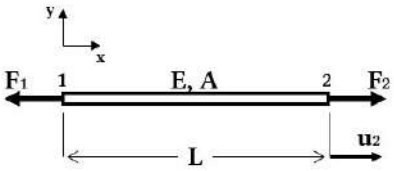


Figure 2.9: Truss element with two DOF for axial displacements.

Its stiffness matrix and displacements vector are given, respectively, by

$$k_a = \frac{EA}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{2.33}$$

and

$$u_a = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^T, \tag{2.34}$$

where A is the element's cross sectional area, L its length and E is Young's modulus.

The torsional element, represented in Fig. 2.10, has also one DOF per node, but in this case it is rotational.

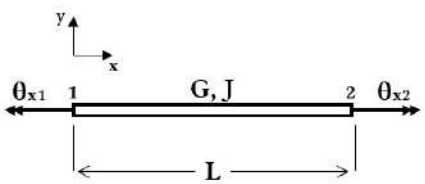


Figure 2.10: Torsional element with two DOF.

The stiffness matrix and the displacements vector for this element are given by

$$k_t = \frac{GJ}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \tag{2.35}$$

and

$$u_t = \begin{bmatrix} \theta_{x1} & \theta_{x2} \end{bmatrix}^T, \tag{2.36}$$

where G is the shear modulus and J is is the polar moment of inertia.

The pure bending element with two translational and two rotational DOF per node is represented in Fig. 2.11 for the bending about the the z axis.

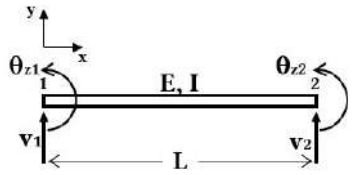


Figure 2.11: Beam element under pure bending with four DOF.

The stiffness matrix is given by

$$k_b = \frac{EI_z}{L^3}\begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix}, \tag{2.37}$$

where I_z is the second moment of inertia about the z axis and the displacements vector is defined as

$$u_b = \begin{bmatrix} v_1 & \theta_{z1} & v_2 & \theta_{z2} \end{bmatrix}^T. \tag{2.38}$$

The complete stiffness matrix of an element is obtained by superimposing the stiffness matrices of the individual elements, yielding

$$k_e = \begin{bmatrix} X & 0 & 0 & 0 & 0 & 0 & -X & 0 & 0 & 0 & 0 & 0 \\ & Y_1 & 0 & 0 & 0 & Y_2 & 0 & -Y_1 & 0 & 0 & 0 & Y_2 \\ & & Z_1 & 0 & -Z_2 & 0 & 0 & 0 & -Z_1 & 0 & -Z_2 & 0 \\ & & & S & 0 & 0 & 0 & 0 & 0 & -S & 0 & 0 \\ & & & & Z_3 & 0 & 0 & 0 & Z_2 & 0 & Z_4 & 0 \\ & & & & & Y_3 & 0 & -Y_2 & 0 & 0 & 0 & Y_4 \\ & & & & & & X & 0 & 0 & 0 & 0 & 0 \\ & & & & & & & Y_1 & 0 & 0 & 0 & Y_2 \\ & & & \text{Symmetric} & & & & & Z_1 & 0 & -Z_2 & 0 \\ & & & & & & & & & S & 0 & 0 \\ & & & & & & & & & & Z_3 & 0 \\ & & & & & & & & & & & Y_3 \end{bmatrix} \tag{2.39}$$

, with the displacements vector as

$$u_e = \begin{bmatrix} u_1 & v_1 & w_1 & \theta_{x1} & \theta_{y1} & \theta_{z1} & u_2 & v_2 & w_2 & \theta_{x2} & \theta_{y2} & \theta_{z2} \end{bmatrix} \tag{2.40}$$

and

$$X = \frac{AE}{L} \qquad Y_1 = \frac{12EI_z}{L^3} \qquad Y_2 = \frac{6EI_z}{L^2} \qquad Y_3 = \frac{4EI_z}{L} \qquad Y_4 = \frac{2EI_z}{L}$$
$$S = \frac{GI_x}{L} \qquad Z_1 = \frac{12EI_y}{L^3} \qquad Z_2 = \frac{6EI_y}{L^2} \qquad Z_3 = \frac{4EI_y}{L} \qquad Z_4 = \frac{2EI_y}{L}.$$

A matrix like (2.39) is assembled for each spatial beam element in its local frame. Subsequently, it needs to be converted to a global frame so that the global stiffness matrix K can be assembled.

Once this is done, the linear system (2.32) can be solved for the displacements, provided that the aerodynamic loads are known. With nodal displacements, the deformation of the structure under loading can be calculated.

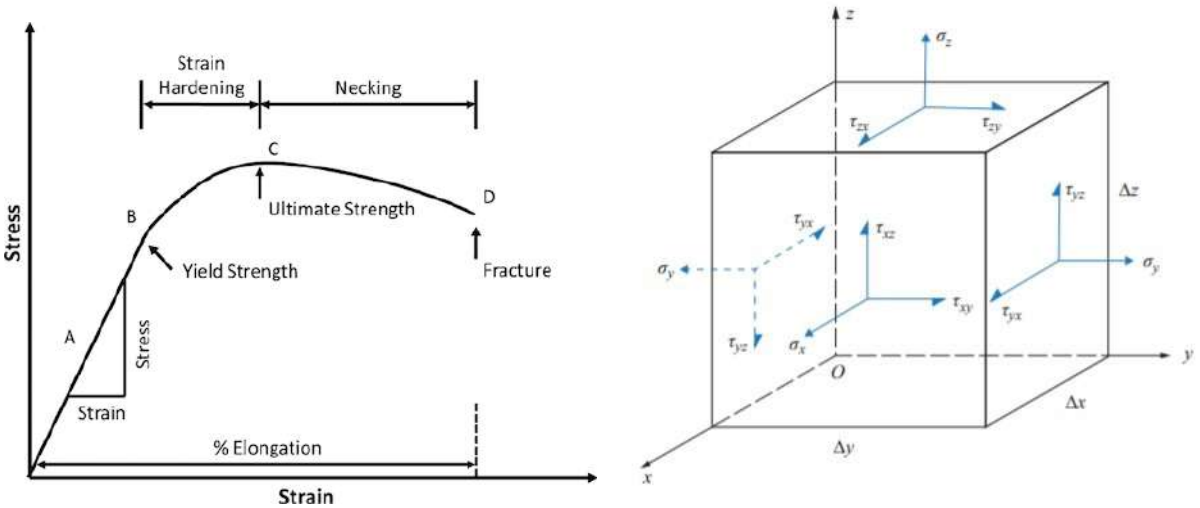
It is also useful to know the stress distribution along the spar. To do so, the stresses on each beam element need to be calculated, which requires transforming the nodal displacements back into the local frame and applying the constitutive equation locally. The spar here considered has a circular cross-section.

$$\text{Axial load} \qquad \sigma_{axial} = \frac{N}{A} = E\varepsilon, \qquad \varepsilon_{axial} = \frac{\Delta u}{L} \qquad \Rightarrow \qquad \sigma_{axial} = E\frac{\Delta u}{L} \tag{2.41}$$

$$\text{Bending load} \qquad \sigma_{bend} = \frac{M_i r}{I_i} = E\varepsilon, \qquad \varepsilon_{bend} = r\frac{\Delta\theta_i}{L} \qquad \Rightarrow \qquad \sigma_{bend} = Er\frac{\Delta\theta_i}{L} \tag{2.42}$$

$$\text{Torsional load} \qquad \tau = \frac{Tr}{J} = E\varepsilon, \qquad \phi = \frac{TL}{GJ} \qquad \Rightarrow \qquad \tau = Gr\frac{\phi}{L} \tag{2.43}$$

where σ and τ represent normal and shear stresses, respectively, the subscript i is the axis y or z and r is the tube's radius, the point of maximum stress.



(a) Typical stress vs. strain behaviour.

(b) Stress components [22].

Figure 2.12: Strees-strain relation and stress components.

The Von Mises yield criterion, which is a result of distortion energy failure theory [23], is used to evaluate failure. The criterion states that a structural component is safe as long as the distortion energy per unit volume

(U_d) in the material does not exceed the distortion energy that causes yield U_Y,

$$U_d < U_Y. \tag{2.44}$$

$$\iff \frac{1+\nu}{3E}\sigma_{VM}^2 < \frac{1+\nu}{3E}\sigma_Y^2, \tag{2.45}$$

where σ_{VM} is the Von Mises stress and σ_Y is the the uniaxial stress yield limit. The Von Mises stress, also referred to as equivalent stress, is defined as the uniaxial tensile stress that would create a distortion energy equal to the one created by the applied stresses represented in Fig. 2.12(b),

$$\sigma_{VM} = \sqrt{\frac{1}{2}\left((\sigma_x-\sigma_y)^2+(\sigma_y-\sigma_z)^2+(\sigma_z-\sigma_x)^2\right)+3(\tau_{xy}^2+\tau_{yz}^2+\tau_{zx}^2)}. \tag{2.46}$$

As seen in Fig. 2.12(a), the yield stress is the limit of the material's elastic behaviour, therefore, to avoid plastic deformation, the equivalent stress must be lower than the yield stress. Simplifying Eq. (2.45),

$$\sigma_{VM}^2 < \sigma_Y^2. \tag{2.47}$$

In this case, normal stresses are being applied on the tube in the axial direction only. Consequently, normal stresses in the y and z directions are zero, $\sigma_y = \sigma_z = 0$, and since there is only torsion about the x axis, $\tau_{yz} = \tau_{zx} = 0$. The Von Mises stress for this tube is thus given by

$$\sigma_{VM} = \sqrt{\sigma_x^2 + 3\tau_{xy}^2}, \tag{2.48}$$

where σ_x is the sum of the axial and bending loads,

$$\sigma_x = \sigma_{axial} + \sigma_{bend} = E\frac{\Delta u}{L} + Er\frac{\Delta\theta}{L}, \tag{2.49}$$

with $\Delta\theta$ being the norm of the rotations about y and z

$$\Delta\theta = \sqrt{\Delta\theta_y^2 + \Delta\theta_z^2} \tag{2.50}$$

The total weight of the spar is calculated by adding the contribution of each element's weight W_e,

$$W_e = g\rho_{material}\int dV \tag{2.51}$$

where $\rho_{material}$ is the material's density and g is the gravitational acceleration.

2.5 Propulsion Model

Fuel mass is usually calculated using the Breguet equation [24], which accounts for the lift to drag ratio (L/D), the proportion of fuel weight to take-off weight and the specific fuel consumption coefficient.

However, the propulsive system in this work is electric, so mass does not vary during flight. The Breguet

equation can be modified for electric propulsion, but it is always written under the assumption that there is equilibrium of forces, which may be true during cruise but not necessarily when trajectory is being optimized. One of the main goals of this work is to perform trajectory optimization. To that end, it becomes necessary to model the propulsive system in a way that thrust is calculated explicitly, as well as the energy spent.

Electric propulsive systems are a common solution for UAV applications because these systems are smaller and easier to integrate into aircraft of this size. It has other advantages, such as lower noise emissions, useful for stealth missions, and a much higher efficiency than that of combustion based systems. However, the low specific energy of state of the art batteries limits electric propulsion's performance and endurance, making it hard to meet the requirements of long duration missions and an economically prohibitive solution for civil aviation.

The type of aircraft and mission being optimized in this work, however, have specifications that allow the use of electric propulsion which is a good option when combined with the advantages mentioned above.

2.5.1 Battery

Batteries are a energy storage solution that has been extensively studied and researched with the objective of reaching metrics that make them a viable alternative to fossil fuels. There are many different types of batteries. Some are already well matured and widely used in our daily lives, like the Lead-acid (Pb-acid), Lithium-ion (Li-ion) and Lithium-polymer (Li-Po) batteries, while others are still in research stage but with promising results, as is the case of Lithium-Air. The typical specific energies of different types of batteries is shown in table 2.1.

Table 2.1: Types of batteries and their specific energy values [25, 26].

Battery	Specific energy [Wh/kg]
Pb-acid	35-40
Li-ion	150-250
Li-Po	100-265
Li-O_2	500-900
Li-Air	1700

In this work, the energy density of the battery is considered 210 Wh/Kg, a value within the range of Li/ion and Li-Po batteries. Considering a Li-Air battery for example, could yield unrealistic results for what is possible today, making it more difficult to evaluate the impact of the aircraft design and trajectory in the performance. Simulating the influence of other batteries in the performance of an aircraft has its interest, but it is not in the scope of this work.

The available energy E_{bat} for the mission is given by

$$E_{bat} = e m_{bat}, \tag{2.52}$$

where m_{bat} is the battery's mass and the its specific energy e is assumed constant.

The typical discharge curve of a Li-Po battery is represented in Fig. 2.13, where it is shown how the voltage varies with the discharge capacity for different constant currents.

As can be seen in the figure, this battery starts at 4.2 V but quickly drops to the nominal voltage (≈ 3.7 V - 3.5 V),

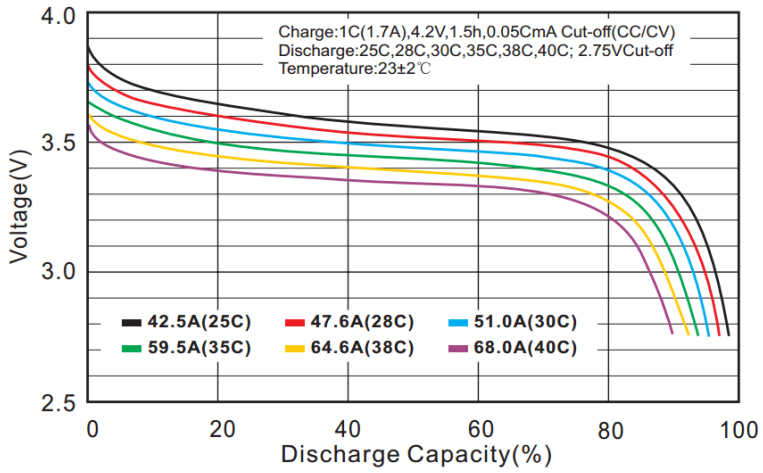


Figure 2.13: Typical discharge curve for Li-Po battery for several C-rates [27].

which it maintains for most of the discharge, until it reaches the performance cliff and drops to the cut-off voltage of 2.75 V, imposed to protect the battery. Li-Po batteries exhibit this nearly flat discharge, from 5% to 80% of the cycle. A linear or even higher model of the discharge could be used, but it would increase the complexity of the overall model due to a component whose behaviour is not the priority of this work. Moreover, the variation of output is very small in this interval (around 5%), so any gain in precision would be minimal. For these reasons, the available electric power is assumed constant and superior to the power drawn by the motor.

2.5.2 Motor

Electric motors convert electric energy into mechanical energy and they can be of alternating current (AC) or direct current (DC). AC motors tend to have higher torque but DC motors have a wider spectrum of optimal power settings and no efficiency losses in the DC-AC conversion.

DC motors can be divided in two main types: brushed and brushless. In brushed motors, the wire coils form an armature that acts as an electromagnet. It is fed electric current, whose direction is reversed through a mechanical commutator. With the change of current's direction, the polarity of the armature's electromagnet also changes and because there is a stationary magnet around it, the armature rotates. In brushless motors, the rotor is the permanent magnet and the coils are static. Commutation is done electronically, which is more precise than with mechanical brushes. Brushless motors have the advantage of being more efficient than brushed motors, generating less noise, having longer lifetime and higher power to weight ratio [28].

DC brushless motors can be further divided in in-runners and out-runners. The main difference between them is that the in-runner has the magnets placed on the shaft and the windings on the outer part of the motor, whereas the out-runner has the magnets turning on the external part of motor, around the stator [28]. The main advantage of the out-runner are that it has high torque which eliminates the need of a gearbox, thus reducing the number of moving parts and weight.

The electric motor's response can be approximated by a first order model, represented by the equivalent circuit in Fig. 2.14.

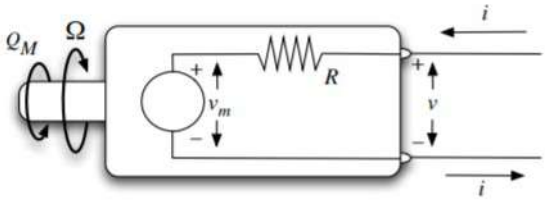


Figure 2.14: Equivalent circuit for a DC electric motor [28].

In this model, the shaft torque is assumed to be proportional to the current i,

$$Q_M = \frac{(i - i_0)}{K_Q} \tag{2.53}$$

where i_0 is the current under no load, also interpreted as friction current, and K_Q is a torque constant, which can be considered equal to the speed constant $K_v \approx K_Q$ [28].

The rotation rate Ω is related to the electromotive force (EMF), v_m, by means of the speed constant

$$\Omega = v_m K_v \tag{2.54}$$

and the EMF can be obtained by solving the circuit's equation, according to Kirchhoff's voltage law

$$v = v_m + iR \iff v_m = v - iR \tag{2.55}$$

where R is the motor's resistance.

Finally, the motor's loss factor LF can be defined as the ratio between the input and output powers

$$LF = \frac{P_m}{P_e} \tag{2.56}$$

where P_m is the power on the shaft, given by

$$P_m = Q_M \Omega \tag{2.57}$$

and P_e is the electric power

$$P_e = vi. \tag{2.58}$$

The motor can be represented by a first order model such as the one just described, but for same reason as in the battery, a simpler model is used. It is assumed that the electric motor has a constant maximum power of 180 W, and the actual power is controlled by a throttle setting δ_T, such that

$$P_m = \delta_T P_{Max}, \quad \delta_T \in [0, 1]. \tag{2.59}$$

The loss factor is given by Eq. (2.56). Although it is usually dependent on the rotation speed and other parameters intrinsic to the motor, as its resistance R, speed constant K_v and no load current I_0, in this work, it will be assumed constant with a value of 50%.

2.5.3 Propeller

The mechanical power that the motor transmits to the shaft is converted by the propeller into thrust force. It is assumed losses in that transmission are negligible. The performance of the propeller is usually described with Blade Element and Momentum Theory, which is the base of the relation used in this work to compute the thrust.

$$P_{disk} = TV_\infty + \frac{\kappa}{2}T\left(-V_\infty + \sqrt{V_\infty^2 + \frac{2T}{\rho A_{disk}}}\right) \tag{2.60}$$

A complete description of the theory and derivation of the equation is made in [29]. P_{disk} is the power supplied to the propeller disk, here assumed equal to P_m, T is the thrust, V_∞ is the freestream velocity, ρ is the air density and A_{disk} is the disk area of the propeller. The correction factor κ accounts for induced-power losses related to non-uniform inflow, tip effects, and other simplifications made in momentum theory [30]. It is assumed $\kappa = 1.2$, where the ideal is 1. The disk area is calculated by $A_{disk} = \pi\frac{d^2}{4}$, where the disk diameter d is 30 cm, corresponding to a typical propeller diameter for the class of UAV being studied in this work.

2.5.4 Propulsive System

The combination of the models of the propulsion components previously described results in the algorithm of the propulsive system.

1. The mechanical power, controlled by the throttle setting δ_T, is calculated through Eq. (2.59).
2. This mechanical power and air speed are inputs to Eq. (2.60), from which thrust is explicitly calculated using the Newton-Raphson method.
3. The electrical power is obtained from the loss factor relation (2.56).
4. It is then used to calculate the electrical energy spent E_e

$$E_e = \int P_e dt. \tag{2.61}$$

which is limited by the energy of the battery

$$E_e \leq E_{bat} = m_{bat}e. \tag{2.62}$$

2.6 Aircraft Design and Control

The optimization of the aircraft's design can be performed with the objective of minimizing the electrical energy spent E_e. The trimming design variables are the angle of attack α and the angle of the horizontal stabilator δ_t. The geometric variables of the wing and tail are the angles of sweep Λ_w, Λ_t and dihedral Γ_w, Γ_t, the span b_w, b_t, and the chord distributions $c(y)_w$, $c(y)_t$. The wing also has a twist distribution design variable $\theta(y)_w$. The structural design variables are the spar thickness distributions $t(y)_w$ and $t(y)_t$. Finally, the propulsion variable is the throttle setting δ_T.

This optimization is performed on a flight segment with prescribed constant parameters (air density, speed, etc). It is assumed that there is equilibrium of forces and moments, therefore three equilibrium constraints are imposed so that equations (2.9a), (2.9b) and (2.12) are equal to zero. One energy constraint ensures that the total energy consumed does not exceed the battery capacity (Eq. (2.62)). One aerodynamic constraint is imposed on each panel so that the lift on every 2D section does not exceed the maximum, thus limiting angles of attack that would lead to stalling conditions and unrealistic results. Two structural constraints are imposed to prevent failure of the wing and tail's spars, which is achieved through the Von Mises yield criterion

$$Failure = \sigma_{VM} - \frac{\sigma_Y}{FoS} < 0, \tag{2.63}$$

where FoS is the factor of safety. Additionally, there are two geometric constraints, one for each spar, to prevent material intersection, which is achieved by enforcing

$$Intersect = t(y) - r(y) < 0, \tag{2.64}$$

where $r(y)$ is the radius of the spar at the spanwise position y.

This optimization problem can be then formulated in standard form as

$$\begin{aligned}
&\text{Minimize} && E_e \\
&\text{w.r.t.} && \delta_T, \alpha, \delta_t, \theta(y)_w, \Lambda_w, \Lambda_t, \text{ד}_w, \text{ד}_t, \\
& && b_w, b_t, c(y)_w, c(y)_t, t(y)_w, t(y)_t \\
&\text{subject to} && \\
&\text{equilibrium constraints} && \sum F_x = 0,\ \sum F_z = 0,\ C_{m_y} = 0 \\
&\text{energy constraint} && E_{spent} \leq E_{bat} \\
&\text{aerodynamic constraints} && Cl_w < Cl_{w_{max}} \\
& && Cl_t < Cl_{t_{max}} \\
&\text{structural constraints} && Failure_w < 0 \\
& && Failure_t < 0 \\
&\text{geometric constraints} && Intersect_w < 0 \\
& && Intersect_t < 0.
\end{aligned} \tag{2.65}$$

2.7 Aircraft Performance and Operating Point

The problem formulated in Sec. 2.6 is written for a single flight stage (one optimization point), but can be generalized for any number of stages (multipoint). This requires identifying and separating variables that change during flight stages (local) from those that are kept constant (global).

For example, if the mission is considered to have a climb phase and cruise, two input values for airspeed and air density are required, and the stabilator angle, a local design variable, would be a design vector of size 2. The spar thickness, on the other hand, is a design variable constant throughout flight stages, so its size would remain the same. The difference between a global and a local design variable is that the former has its optimal value calculated accounting for both stages, whereas the latter has an optimal value for each.

This can be extrapolated for as many flight segments as desired. If the number of stages is sufficiently large, complex trajectories can be considered in the aircraft design while maintaining a simple and linear model. This principle is applied when optimizing the trajectory, which is discretized by several segments. Figure 2.15 depicts a multipoint for a generic trajectory, with a vector of some local variables and another with global ones.

The aircraft performance is measured primarily by the value of the objective of optimization. In the case of energy spent, it is done with the results (2.61) and (2.62) from Sec. 2.5.4. Additionally, the total and propulsive efficiency coefficients can be used. The former is defined as the ratio of the potential and kinetic energy to the electrical energy consumed

$$\eta_{total} = \frac{mgz + 0.5mV_\infty^2}{\int P_e dt} \tag{2.66}$$

and the latter is defined as

$$\eta_{prop} = \frac{TV_\infty}{P_m}. \tag{2.67}$$

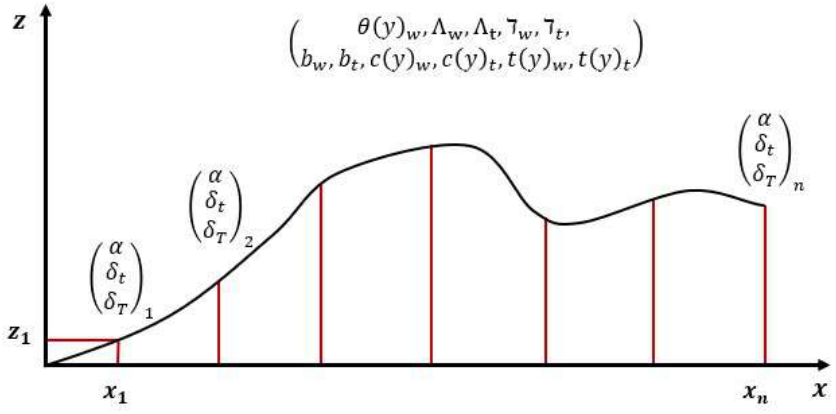


Figure 2.15: Multipoint optimization for generic trajectory with local variables $(\alpha, \delta_t, \delta_T)$ and global variables $(\theta(y)_w, \Lambda_w, \Lambda_t, \lambda_w, \lambda_t, b_w, b_t, c(y)_w, c(y)_t, t(y)_w, t(y)_t)$

Chapter 3

Trajectory Optimization

This chapter starts by presenting the formulation of an optimal control problem, followed by the example of a one dimensional trajectory problem, which is later used to demonstrate and compare the differences of direct shooting and collocation methods of transcription. Lastly, the trajectory optimization problem is formulated.

3.1 Trajectory Optimal Control

In an optimization problem, the aim is to find the feasible inputs to the system that will optimize a chosen parameter. When the inputs are the control variables that will determine the state of the system, it is called an optimal control problem (OCP). A more specific problem is when the state is the position and velocity that compose the trajectory that leads to the optimal solution. In this case, it is a trajectory optimization problem.

Let us take the example of the one dimensional displacement of an unitary point mass, represented in Fig. 3.1. It is initially at rest, positioned at x_0 and the objective is to arrive at an arbitrary final position x_f with zero velocity ($\dot{x} = 0$) in the least amount of time. Hence, the objective variable is the elapsed time t_f and this is a minimization problem.

The *state* of the system at any given time can be described by the position $x(t)$ and velocity $\dot{x}(t)$ of the mass, which are the variables that compose the state vector $\mathbf{s}$. The initial velocity is zero, so the only way the mass will move from x_0 to x_f is if a force F is applied. The sequence of the force values chosen as a function of time is responsible for changing the state, and therefore must be chosen so that the trajectory minimizes the elapsed time. Hence, the force F is the control $\mathbf{u}$.

The restrictions of this problem are the final position and speed, as well as the interval of allowable speed and force values. These would be the state and control constraints that must be obeyed when searching for the optimal control and trajectory.

Figure 3.1: Bang bang control problem.

In a more general way, a system can be defined by the state $\mathbf{s}(t)$, the control $\mathbf{u}(t)$, the time $t \in [t_0, t_f]$ and any eventual time independent parameters $\mathbf{p}$. The parameter of optimization J, called objective variable or functional, is dependent on these variables and can be written as a function of a boundary objective term and/or an integral along the entire trajectory. A problem with both terms is said to be in Bolza form, a problem only with the boundary term is in Mayer form and a problem only with the integral term is in Lagrangian form [31, 32].

A trajectory optimization problem can be posed in Bolza form, to minimize the objective function J,

$$J = \Phi(\mathbf{s}(t_0), t_0, \mathbf{s}(t_f), t_f, \mathbf{p}) + \int_{t_0}^{t_f} \mathcal{L}(\mathbf{s}(t), \mathbf{u}(t), t_0, t_f, \mathbf{p}) dt \tag{3.1a}$$

with respect to the design variables,

$$\mathbf{s}(t), \mathbf{u}(t), t, \mathbf{p} \tag{3.1b}$$

and subject to the dynamics of the system, defined by a set of ordinary differential equations, also referred to as state equations,

$$\dot{\mathbf{s}}(t) = f(\mathbf{s}(t), \mathbf{u}(t), t, \mathbf{p}). \tag{3.1c}$$

The problem is also subject the initial and terminal conditions, the boundary constraints,

$$\boldsymbol{\phi}_{min} \leq \boldsymbol{\phi}(\mathbf{s}(t_0), t_0, \mathbf{s}(t_f), t_f, \mathbf{p}) \leq \boldsymbol{\phi}_{max}. \tag{3.1d}$$

Additionally, the path constraints must be satisfied,

$$\boldsymbol{\psi}_{min} \leq \boldsymbol{\psi}(\mathbf{s}(t), \mathbf{u}(t), t, \mathbf{p}) \leq \boldsymbol{\psi}_{max}, \tag{3.1e}$$

as well as the bounds on state variables,

$$\mathbf{s}_{low} \leq \mathbf{s}(t) \leq \mathbf{s}_{up} \tag{3.1f}$$

and control variables,

$$\mathbf{u}_{low} \leq \mathbf{u}(t) \leq \mathbf{u}_{up}. \tag{3.1g}$$

The dynamics of the system are described by a set of differential equations (3.1c), which can be seen as dynamic constraints. Additionally, there are path constraints that enforce restrictions on the trajectory and boundary constraints that define the values of initial and final states.

The dimension and complexity of the problem would have to be very small for it to be solved analytically, so numerical methods are used. Figure 3.2 presents a diagram adapted from Mir et al. [33], with optimal control problem solving techniques that will be discussed next.

Numerical methods for solving optimal control problems are divided into three major methods: dynamic programming, indirect methods and direct methods.

Dynamic programming relies on the Bellman optimality principle, which states that if a given state-action sequence is optimal, and the first state and action were to be removed, the remaining sequence is also optimal and the second state of the original sequence would now be the initial state [34]. Dynamic programming determines both the necessary and sufficient conditions of optimality through the Hamilton-Jacobi-Bellman partial

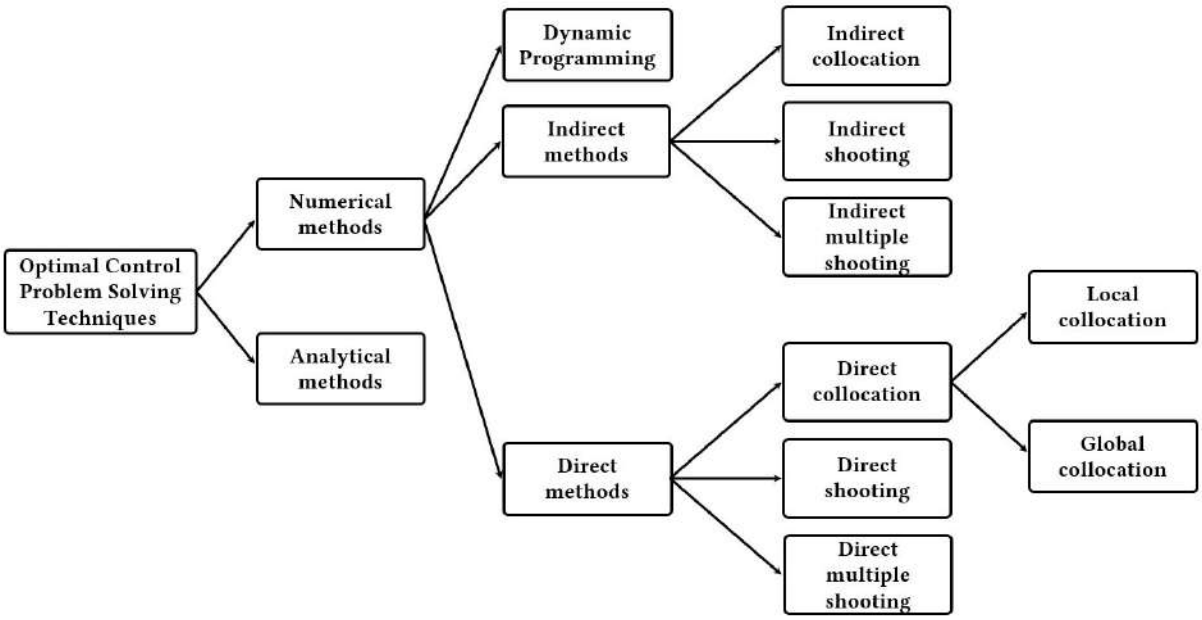


Figure 3.2: Numerical techniques for solving optimal control problems [33].

differential equation. Its major drawback is "the curse of dimensionality", as Bellman [35] called it, which makes achieving an acceptable accuracy only possible for small sized problems.

Table 3.1: Numerical methods and their major optimal control components [36].

Indirect Methods		
Systems of Nonlinear Equations	Differential Equations and Integration of Functions	Nonlinear Optimization
	Direct Methods	

Regarding indirect and direct methods, Rao [36] explains that a well founded optimal control problem has two of three major components at its core (Tab. 3.1). Solving differential equations and integrating functions is required for both methods. The indirect methods make use of these solutions and couple them with the solution of a system of nonlinear equations, whereas direct methods combine it with the solution of the nonlinear optimization.

3.2 Indirect Methods

In indirect methods, the control and state must fulfil a set of conditions to become an extremal, i.e. a candidate for optimal solution. Those conditions are called necessary conditions of optimality and are obtained from the calculus of variations and Pontryagin's minimum principle. A brief review of this class of methods is done, but for more mathematical detail, theorems enunciation and proof refer to Böhme and Frank [37] and Kirk [38].

For a problem that can be stated in a similar way as in Eq. (3.1), with the objective of minimizing a functional $J(\boldsymbol{x})$ by finding the optimal trajectory $\boldsymbol{x}^*$, it can be said that one particular trajectory $\boldsymbol{x}$ minimizes the functional $J(\boldsymbol{x})$ when all the neighbouring trajectories $\boldsymbol{x} + \delta x$ yield a larger functional value. In other words, for $\boldsymbol{x}^*$ to be a

minimum of J, the variation δJ, must be zero,

$$\delta J(x) = \lim_{\tau \longrightarrow 0} \frac{J(x + \tau \delta x) - J(x)}{\tau} = \frac{d}{d\tau} J(x + \tau \delta x)\Big|_{\tau=0}. \tag{3.2}$$

This is the basic principle of the calculus of variations used in the derivation of the necessary conditions of optimality. To exemplify the derivation of the first order necessary conditions, let us consider a problem similar to that stated in Eq. (3.1) with no static parameters. The first step of the derivation is to write the Lagrangian of the objective function. To do so, the differential equation, the path and boundary constraints are appended to the objective functional and multiplied by the costates vector, $\boldsymbol{\lambda}$ and the Lagrange multipliers, $\boldsymbol{\mu}$ and $\boldsymbol{\nu}$, respectively,

$$J = \Phi(\boldsymbol{x}(t_0), \boldsymbol{x}(t_f)) + \int_{t_0}^{t_f} \mathcal{L}(\boldsymbol{x}(t), \boldsymbol{u}(t), t, \boldsymbol{p}) dt + \boldsymbol{\nu}_0^T \boldsymbol{\phi}(\boldsymbol{x}(t_0), t_0) + \boldsymbol{\nu}_f^T \boldsymbol{\phi}(\boldsymbol{x}(t_f), t_f) + \boldsymbol{\mu}^T C(\boldsymbol{x}(t), \boldsymbol{u}(t), t) + \\ + \int_{t_0}^{t_f} \left(\boldsymbol{\lambda}^T(t) f(\boldsymbol{x}(t), \boldsymbol{u}(t)) - \boldsymbol{\lambda}^T(t) \dot{\boldsymbol{x}}(t) \right) dt. \tag{3.3}$$

Then, the term with $\dot{\boldsymbol{x}}$ is integrated by parts, leading to

$$J = \Phi(\boldsymbol{x}(t_0), \boldsymbol{x}(t_f)) + \int_{t_0}^{t_f} \mathcal{L}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) dt + \boldsymbol{\nu}_0^T \boldsymbol{\phi}(\boldsymbol{x}(t_0), t_0) + \boldsymbol{\nu}_f^T \boldsymbol{\phi}(\boldsymbol{x}(t_f), t_f) + \boldsymbol{\mu}^T C(\boldsymbol{x}(t), \boldsymbol{u}(t), t) - \\ - \boldsymbol{\lambda}^T(t_f) \boldsymbol{x}(t_f) + \boldsymbol{\lambda}^T(t0) \boldsymbol{x}(t0) + \int_{t_0}^{t_f} \left(\boldsymbol{\lambda}^T(t) f(\boldsymbol{x}(t), \boldsymbol{u}(t)) - \dot{\boldsymbol{\lambda}}^T(t) \boldsymbol{x}(t) \right) dt. \tag{3.4}$$

The necessary conditions are usually expressed using the Hamiltonian, which is defined as

$$\mathcal{H}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{\lambda}, t) = \mathcal{L}(\boldsymbol{x}(t), \boldsymbol{u}(t), t) + \boldsymbol{\lambda}^T(t) f(\boldsymbol{x}(t), \boldsymbol{u}(t)) - \boldsymbol{\mu}^T C(\boldsymbol{x}(t), \boldsymbol{u}(t), t). \tag{3.5}$$

Having this augmented functional and the Hamiltonian defined, the application of the principle from Eq. (3.2) to Eq. (3.4), with respect to $\delta \boldsymbol{x}$, $\delta \boldsymbol{u}$ and $\delta \boldsymbol{\lambda}$, should result in the vanishing of all terms for a stationary point to be found. This is valid under the variational assumption that the control is unbounded, because when it is bounded by an admissible control set $\mathcal{U}$, the variation w.r.t $\delta \boldsymbol{u}$ is not required to be zero to be a minimum [37]. In this case, the necessary conditions are

$$\dot{\boldsymbol{x}}^*(t) = \frac{\partial \mathcal{H}}{\partial \boldsymbol{\lambda}}(\boldsymbol{x}^*, u^*, \boldsymbol{\lambda}, t), \tag{3.6a}$$

$$\dot{\boldsymbol{\lambda}}(t) = -\frac{\partial \mathcal{H}}{\partial \boldsymbol{x}}(\boldsymbol{x}^*, u^*, \boldsymbol{\lambda}, t), \tag{3.6b}$$

$$\boldsymbol{u}^*(t) = \arg \min_{\boldsymbol{u}(t) \in \mathcal{U}} \mathcal{H}(\boldsymbol{x}^*, u^*, \boldsymbol{\lambda}, t), \tag{3.6c}$$

$$\boldsymbol{\lambda}(t_0) = -\frac{\partial \Phi}{\partial \boldsymbol{x}(t_0)} + \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \boldsymbol{x}(t_0)}, \qquad \boldsymbol{\lambda}(t_f) = \frac{\partial \Phi}{\partial \boldsymbol{x}(t_f)} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial \boldsymbol{x}(t_f)}, \tag{3.6d}$$

and

$$\mathcal{H}(t_0) = \frac{\partial \Phi}{\partial t_0} - \boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_0}, \qquad \mathcal{H}(t_f) = -\frac{\partial \Phi}{\partial t_f} + \boldsymbol{\nu}^T \frac{\partial \phi}{\partial t_f}. \tag{3.6e}$$

Applying these conditions to an optimal control problem results in multi-point boundary value problems. These problems can be solved through indirect shooting or multiple shooting methods and indirect collocation methods.

Indirect methods are highly accurate but their applicability is limited because they require analytical derivation of the first order necessary conditions for each new problem [37].

3.3 Direct Methods

Direct methods are less accurate than indirect methods. However, they can deal with large systems, are more robust, flexible and not as difficult to construct and solve, because they do not require the analytical derivation of the necessary conditions [37].

While indirect methods reduce the optimal control problem to a boundary value problem, direct methods reduce it to a nonlinear programming (NLP) problem. NLP is the name given to a problem when the objective function is nonlinear and/or the feasible region is determined by nonlinear constraints.

The first step for obtaining the NLP is transcription. Direct transcription refers to the transformation of the infinite dimensional continuous part of the OCP into a finite dimensional NLP [37], which is done through the parameterization of the state and/or control. As ilustrated in Fig. 3.3, direct single shooting and direct multiple shooting are the methods used for control parameterization, and direct collocation is the method used for state and control parameterization.

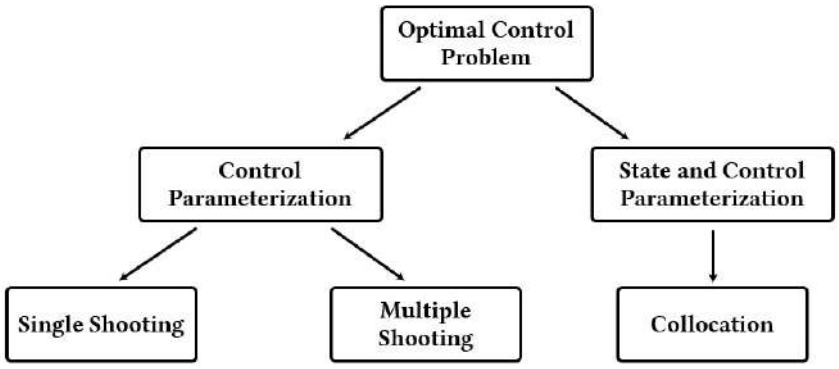


Figure 3.3: Types of direct transcriptions. Adapted from [37].

3.3.1 Direct Shooting

In direct shooting methods, time is first broken into N intervals, such that $h_i = t_{i+1} - t_i, (i = 0, \dots, N-1)$ and control is discretized. Then, the dynamics differential equation is integrated sequentially over each time step to obtain the state $\mathbf{s}_{i+1}$. The integration at time step h_i can be performed using current or previous information about the solution, so shooting methods employ a time marching approach [36]. Having integrated over every time step, the last state constraint is evaluated and if it is violated or the solution is not optimal, the control vector is updated.

Using the example of the one dimensional trajectory of a point mass, the elapsed time was the minimization objective, with respect to the control variable F, subject to a final position $x(t_f) = x_f$, final velocity $\dot{x} = 0$, state

and control bounds. The resultant NLP can be stated as in (3.7), where all units are SI.

$$\begin{aligned}
&\text{Minimize} && t_f \\
&\text{w.r.t} && \boldsymbol{F}, t_0, t_f \\
&\text{subject to} && \\
&\text{boundary constraints} && x(0) = 0, \dot{x}(0) = 0, x(t_f) = 300, \dot{x}(t_f) = 0 \\
&\text{state bounds} && -200 \leq \dot{\boldsymbol{x}} \leq 200, \\
&\text{control bounds} && -2 \leq \boldsymbol{F} \leq 1
\end{aligned} \tag{3.7}$$

In this example, time was broken into $N = 30$ subintervals and the control $F(t)$ was discretized into a vector of size $N = 30$ with initial guess for all entries equal to the upper bound.

Due to the nature of the shooting methods, the NLP has no explicit dynamic constraint, which does not mean the dynamics will not be satisfied. It means that the optimizer will not take that constraint explicitly into account when performing the optimization, even though it will eventually be satisfied through sequential integration. The dynamics of the system can be described in matrix form as

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} F. \tag{3.8}$$

The second equation of the system establishes the relationship between control and acceleration. Because control is the force, this is Newton's second law, $F = ma$ with mass $m = 1Kg$. The integration of this equation over a time interval $h_i = t_{i+1} - t_i$ is given by

$$\int_0^{h_i} \ddot{x} dt = \int_0^{h_i} F dt \tag{3.9}$$

and its solution was obtained using forward Euler scheme,

$$\dot{x}_{i+1} - \dot{x}_i = h_i F_i. \tag{3.10}$$

From Eq. (3.10) results a velocity variation $\Delta\dot{x}$ that is then used to compute the state s_{i+1} as

$$\dot{x}_{i+1} = \dot{x}_i + \Delta\dot{x} \tag{3.11a}$$

$$x_{i+1} = x_i + \Delta\dot{x} h_i \tag{3.11b}$$

Figure 3.4 shows the optimization evolution to the 35^{th} and last iteration, where it can be seen that the force $F(t)$ and final time t_f were chosen and corrected at the end of each iteration until an optimal solution was found.

Shooting methods have the advantage that the problem can be posed in terms of a small number of variables, since the state is not parameterized. The total number of variables is given by the sum of control and state parameterized variables times the number of grid points: $n = (n_s + n_c)N$. In this case, there is only one control and 30 grid points plus one time variable, making it a total of $n = 31$ variables. Additionally, there are 63 constraints, 4 for boundaries, 29 bounds for velocity and 30 for control.

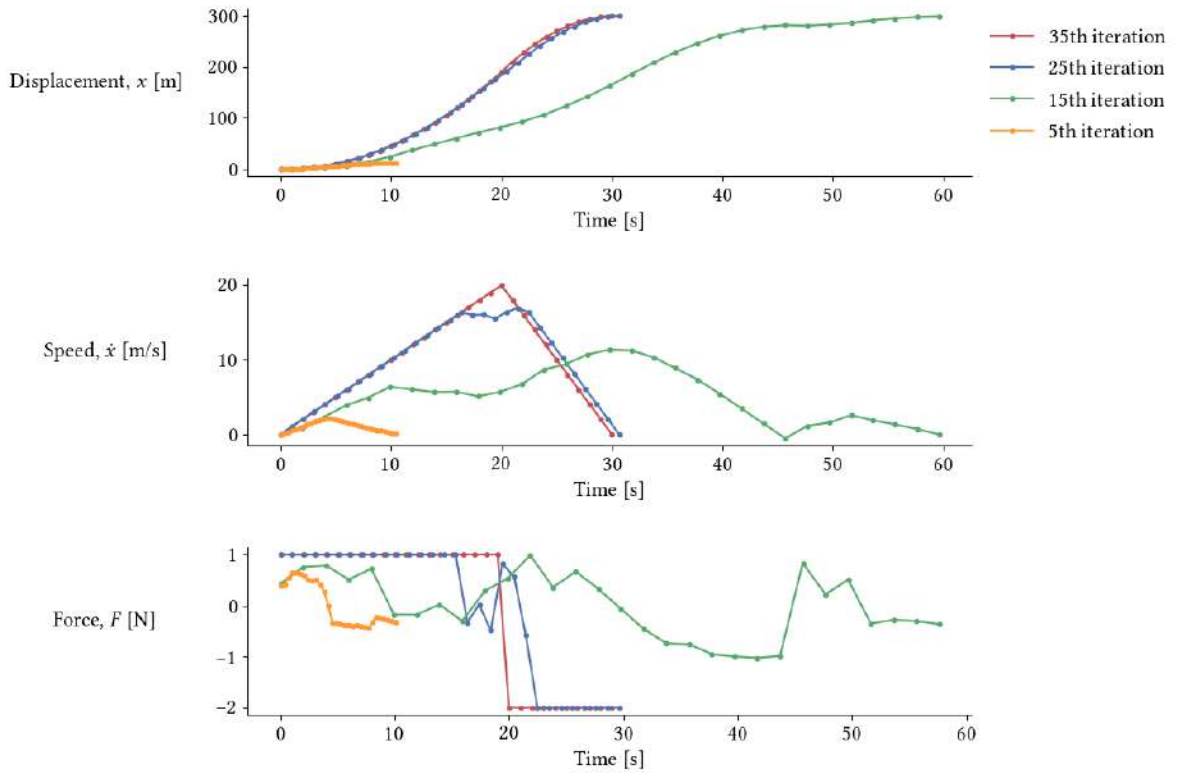


Figure 3.4: Results evolution with iterations of the 1D trajectory optimization problem solved through direct single shooting method.

As Betts [32] pointed out, shooting methods face some difficulties associated to the sensitivity of the variables. Because the state is integrated sequentially, the error of integration and the changes made early in the trajectory will propagate to the end of it. This leads the constraints to behave very nonlinearly with respect to variables and, as a consequence, makes the optimization problem difficult to solve.

3.3.2 Direct Collocation

Direct collocation is a state and control parameterization method and can be divided in local and global collocation. The main difference is that in local, the time domain is divided in segments where the state is approximated by piecewise polynomials, while the latter uses a global polynomial for the entire time interval.

In local collocation, the number of collocation points in a given subinterval is fixed and the number of subintervals is varied [39]. So time is broken into N subintervals,

$$h_i = t_{i+1} - t_i, \quad (i = 0, \dots, N-1) \tag{3.12}$$

and the dynamics of the system, ruled by the differential equations,

$$\dot{\mathbf{s}}(t) = \boldsymbol{f}(\mathbf{s}(t), \mathbf{u}(t)), \tag{3.13}$$

are not integrated sequentially to obtain the state $\mathbf{s}_{i+1}$, like in shooting methods. Instead, it functions as a defect

constraint,

$$\boldsymbol{\zeta} = \dot{\mathbf{s}}(t) - \boldsymbol{f}(\mathbf{s}(t), \mathbf{u}(t)) = 0, \tag{3.14}$$

imposed on each collocation point to ensure consistency between state and control values throughout the trajectory. Equation (3.14) can be discretized using Runge-Kutta or orthogonal methods.

Pseudospectral (or global orthogonal) collocation methods are a class where the parameterization of the state and control is done using global polynomials. The differential equations are also *collocated* at the collocation points, which in this case are the roots of an orthogonal polynomial (or linear combinations of such polynomials and their derivatives), usually Legendre or Chebyshev polynomials. Legendre-Gauss (LG), Legendre-Gauss-Radau (LGR) and Legendre-Gauss-Lobatto (LGL) are examples of commonly used sets of collocation points obtained from the roots of a Legendre polynomial and/or linear combinations of a Legendre polynomial and its derivatives. The difference between them is that LG includes neither of the endpoints, wheres LGR includes one of the endpoints, and LGL includes both of the endpoints [40].

One advantage to pseudospectral methods is that for smooth problems, they typically have faster convergence rates than other methods and exhibit spectral accuracy. However, because they employ global polynomials, they lack the local support that local collocation methods provide, i.e., the discretization points are placed so that they support the local behavior of the dynamics [39].

Compared to shooting methods, collocation has the advantage that the computationally expensive numerical integration of the differential equations can be avoided [41]. Since it does not integrate sequentially, the integration errors are not propagated and it is less sensitive to initial conditions. The resultant NLP is usually larger for collocation methods, but it is sparse and it can be efficiently solved with appropriate solvers.

For the point mass example, time was broken into $N = 30$ intervals, control $F(t)$ and state variables $x(t)$, $\dot{x}(t)$ were discretized into vectors of size N and $N + 1$, respectively. The resultant NLP is written as

$$\begin{aligned}
&\text{Minimize} && t_f \\
&\text{w.r.t} && \boldsymbol{F}, \boldsymbol{x}, \dot{\boldsymbol{x}}, t_f \\
&\text{subject to} && \\
&\text{collocation constraints} && \boldsymbol{\zeta} = 0 \\
&\text{boundary constraints} && x(0) = 0, \dot{x}(0) = 0, x(t_f) = 300, \dot{x}(t_f) = 0 \\
&\text{state bounds} && -200 \leq \dot{\boldsymbol{x}} \leq 200 \\
&\text{control bounds} && -2 \leq \boldsymbol{F} \leq 1.
\end{aligned} \tag{3.15}$$

The vector values of control, position and velocity are all chosen at the beginning of each iteration and updated based on the optimality conditions inherent to the algorithm in use. Regardless of the algorithm, all constraints must be obeyed, including the additional defect constraints (3.16), (3.17), here discretized using the forward Euler numerical scheme,

$$\boldsymbol{\zeta}_1 = \ddot{\boldsymbol{x}} - \boldsymbol{F} = 0 \iff \dot{x}_{i+1} - \dot{x}_i - h_i F_i = 0 \tag{3.16}$$

$$\boldsymbol{\zeta}_2 = x_{i+1} - x_i - h_i \dot{x}_i = 0. \tag{3.17}$$

Figure 3.5 shows the optimal solution and the evolution of the optimization through iterations.

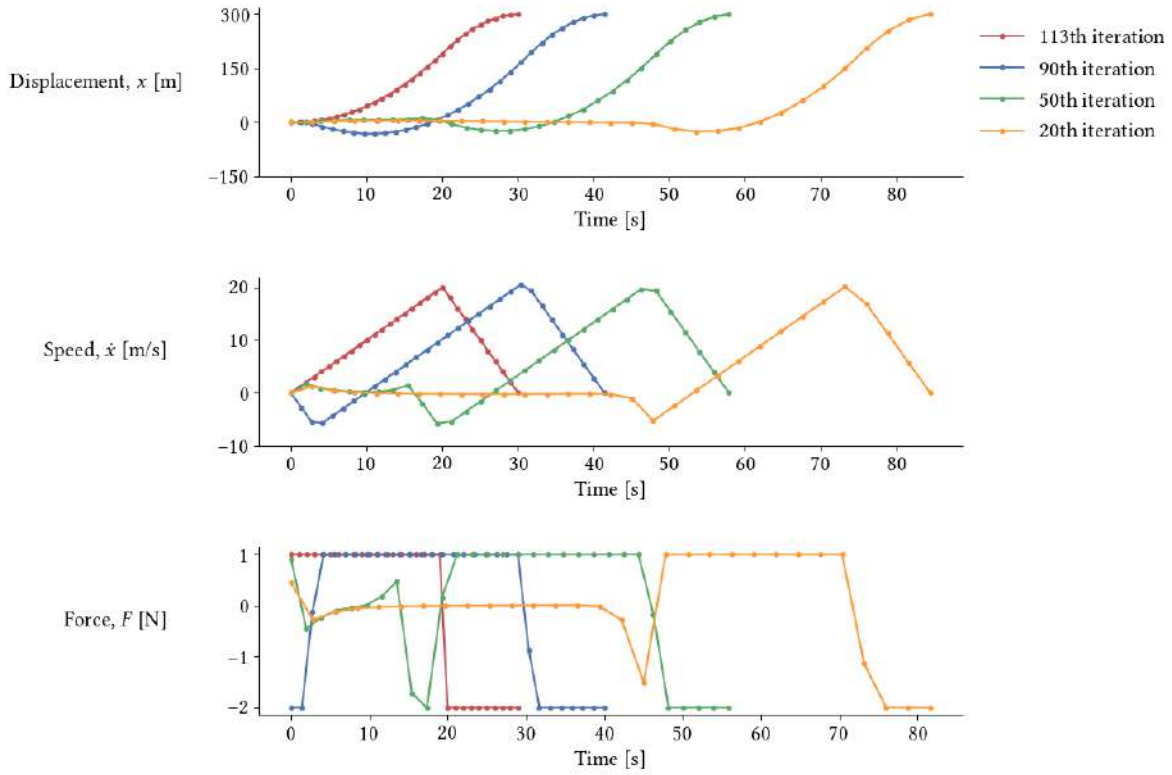


Figure 3.5: Results evolution with iterations of the 1D trajectory optimization problem solved through direct collocation method.

Using this collocation method with the same grid size, the number of variables is the triple of the shooting method and the constraints almost double. The number of variables is 93: 1 for time, 30 for control and 31 for position and velocity each. The total number of constraints is 123: 4 boundaries, 29 and 30 bounds for velocity and control, respectively, and the additional 60 defect constraints.

The most noticeable difference in the results of shooting and collocation methods is that, in the latter, the boundary conditions are always satisfied. This is because in shooting, the final state is dependent on values from the beginning of the trajectory, which might not be correct or optimal, whereas in collocation, the final state is a also design variable and as such, it is accessible to the optimizer, which can immediately choose a value that satisfies the boundary constraints.

Collocation has the advantage of avoiding sequential integration. As a result, errors are not propagated and computational costs are reduced. However, that is done by adding design variables and constraints, thus giving up the simpler NLP and low number of variables of shooting methods. This may lead to more function evaluations, more gradients to be computed and a significant increase in memory needed for storing the Hessian matrix [37].

3.4 Trajectory Design and Control

One of the objectives of this work is to perform trajectory optimization of 2D flight. This is done using a collocation method, in which the dynamics are represented by the equations of motion in the Earth frame, (2.7a) and (2.7b), and are discretized with forward Euler method, yielding the following collocation constraints

$$\zeta_{F_x} = \sum \boldsymbol{F}_{x_E} - m\ddot{\boldsymbol{x}} = 0 \iff \sum F_{x_E,i} - m\frac{\dot{x}_{i+1} - \dot{x}_i}{\Delta t} = 0 \tag{3.18}$$

$$\zeta_{F_z} = \sum \boldsymbol{F}_{z_E} - m\ddot{\boldsymbol{z}} = 0 \iff \sum F_{z_E,i} - m\frac{\dot{z}_{i+1} - \dot{z}_i}{\Delta t} = 0 \tag{3.19}$$

$$\zeta_x = \dot{x}_i - \frac{x_{i+1} - x_i}{\Delta t} = 0 \tag{3.20}$$

$$\zeta_z = \dot{z}_i - \frac{z_{i+1} - z_i}{\Delta t} = 0. \tag{3.21}$$

The collocation method was chosen for the advantages mentioned in Sec. 3.3.2 and because of its easier integration into the OpenAeroStruct framework, as will be explained in Sec. 4.6.

The design variables of this problem are the trajectory variables $x, \dot{x}, z, \dot{z}$, flight time t_f and the throttle control δ_T. The optimization is performed with a fixed aircraft configuration, with exception of the stabilator angle δ_t, which is necessary to trim the aircraft, together with the angle of attack α.

Besides the collocation constraints, are imposed: one equilibrium constraint to force the moments to zero, one energy constraint to guarantee the consumed energy is less or equal to the battery's, one aerodynamic constraint on each panel to limit the lift coefficient on 2D sections and one structural constraint to ensure the integrity of each spar. The resultant NLP is given by

$$\begin{aligned}
&\text{Minimize} && E_e \\
&\text{w.r.t.} && x, \dot{x}, z, \dot{z}, \alpha, \delta_t, \delta_T, t_f \\
&\text{subject to} && \\
&\text{collocation constraints} && \zeta_{F_x} = 0, \zeta_{F_z} = 0,\ \zeta_x = 0,\ \zeta_z = 0 \\
&\text{equilibrium constraint} && C_{m_y} = 0 \\
&\text{energy constraint} && E_{spent} \leq E_{bat} \\
&\text{aerodynamic constraints} && Cl_w < Cl_{w_{max}} \\
&&& Cl_t < Cl_{t_{max}} \\
&\text{structural constraints} && Failure_w < 0 \\
&&& Failure_t < 0.
\end{aligned} \tag{3.22}$$

In this case, the aircraft configuration is fixed, so there are no global design variables. Figure 3.6 depicts a multipoint trajectory with sets of local variables that intend to represent the discretization of the control, trimming and trajectory variables.

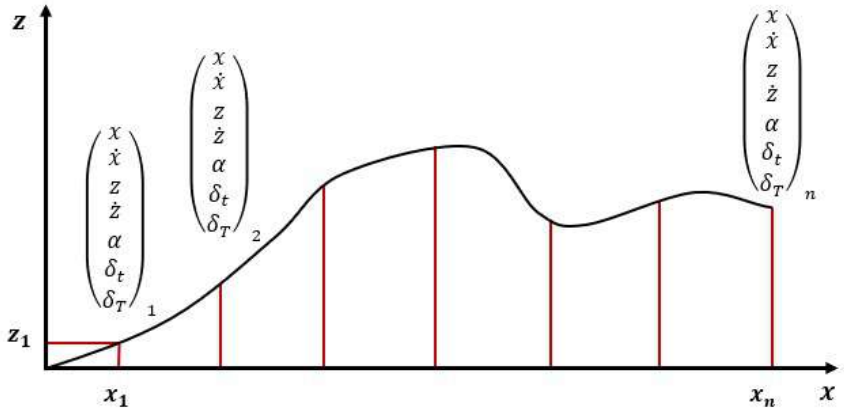


Figure 3.6: Multipoint trajectory with local variables $(x, \dot{x}, z, \dot{z}, \alpha, \delta_t, \delta_T)$

Chapter 4

Coupled Design Framework

This chapter starts with the formulation of the coupled problem. Afterwards, an overview of Multidisciplinary Design Analysis and Optimization is made, where it is described how disciplines are coupled, solved and implemented in an optimization architecture. This is followed by a brief survey of optimization methods and finally, the implemented framework is presented and discussed.

4.1 Problem Statement

The coupled Design and Trajectory problem (DTP) can be defined by the following NLP

$$
\begin{aligned}
\text{Minimize} \quad & E_e \\
\text{w.r.t.} \quad & x, \dot{x}, z, \dot{z}, t_f, \delta_T, \\
& \alpha, \delta_t, \theta(y)_w, \Lambda_w, \Lambda_t, \tau_w, \tau_t, \\
& b_w, b_t, c(y)_w, c(y)_t, t(y)_w, t(y)_t \\
\text{subject to} \quad & \\
\text{collocation constraints} \quad & \zeta_{F_x} = 0, \zeta_{F_z} = 0, \ \zeta_x = 0, \ \zeta_z = 0 \\
\text{equilibrium constraint} \quad & C_{m_y} = 0 \\
\text{energy constraint} \quad & E_{spent} \leq E_{bat} \\
\text{aerodynamic constraints} \quad & Cl_w < Cl_{w_{max}} \\
& Cl_t < Cl_{t_{max}} \\
\text{structural constraints} \quad & Failure_w < 0 \\
& Failure_t < 0 \\
\text{geometric constraints} \quad & Intersect_w < 0 \\
& Intersect_t < 0.
\end{aligned}
\tag{4.1}
$$

The isolated Design and Trajectory problems, (DP) and (TP) respectively, are multidisciplinary problems by themselves. In the former, the aircraft design is done through aerodynamic and structural variables for a

prescribed trajectory. In the latter, the trajectory is designed and the aircraft configuration is fixed. However, aerodynamic and structural constraints have to be satisfied during the entire flight and the aerostructural system is solved to obtain the forces necessary for the equations of motion. So both isolated problems already contain one another, in some way. Merging them means that they are optimized simultaneously. Figure 4.1 depicts a multipoint coupled problem, where local variables are discretized and geometric variables are global and constant throughout the flight.

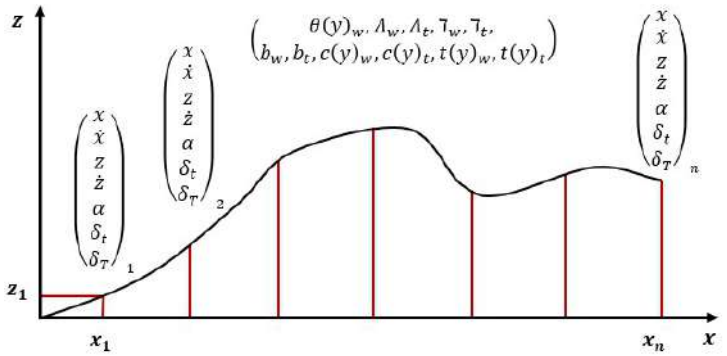


Figure 4.1: Multipoint trajectory with global variables $(\theta(y)_w, \Lambda_w, \Lambda_t, \lambda_w, \lambda_t, b_w, b_t, c(y)_w, c(y)_t, t(y)_w, t(y)_t)$ and local variables $(x, \dot{x}, z, \dot{z}, \alpha, \delta_t, \delta_T)$

4.2 Multidisciplinary Analysis

Solving an optimization problem that involves multiple disciplines requires a careful consideration on how they are related and interact with each other. To understand this and how the system behaves for a set of design variables, Multidisciplinary Analysis (MDA) is performed. Its purpose is to find a solution that satisfies the systems of equations that represent the multidisciplinary system, which comprises the repeated evaluation of the target and response states until they are equal. When that happens, it is said the MDA has converged.

There are several methods to solve the coupled system and they can be grouped in two main categories: fixed-point and Newton approaches. Fixed-point solves one analysis individually while the others remain *fixed*. On the other hand, Newton approaches solve the coupled system by processing all analysis simultaneously.

Jacobi method [42] is one example of an iterative fixed-point approach. It solves each of the n equations of a linear system $Ax = b$ with the outputs from the previous iteration. That is, the new x^k values are obtained with information solely from the previous iteration x^{k-1},

$$\begin{aligned}
a_{11}x_1^k + a_{12}x_2^{k-1} + \ldots + a_{1n}x_n^{k-1} &= b_1 \\
a_{21}x_1^{k-1} + a_{22}x_2^k + \ldots + a_{2n}x_n^{k-1} &= b_2 \\
\vdots \quad \vdots \quad \ddots \quad \vdots \quad \vdots & \\
a_{n1}x_1^{k-1} + a_{n2}x_2^{k-1} + \ldots + a_{nn}x_n^k &= b_n,
\end{aligned} \tag{4.2}$$

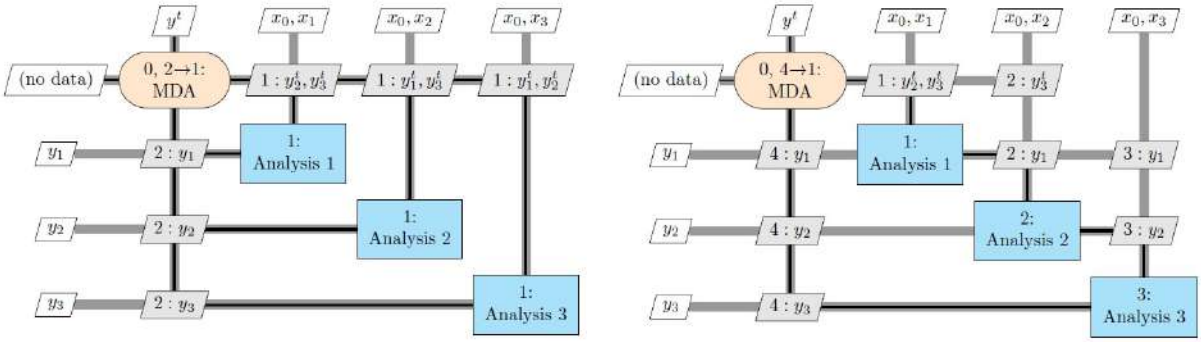
which can be generalized as

$$x_i^{(k)} = \frac{b_i - \sum_{j \neq i} a_{ij}x_j^{k-1}}{a_{ii}} \tag{4.3}$$

or in matrix form

$$x^k = D^{-1}(b - (L + U)x^{k-1}), \tag{4.4}$$

where D, L and U are the diagonal, lower and upper triangular parts of matrix A, respectively. Because only information from previous iteration is used, it is possible to run the analysis in parallel, as shown in Fig. 4.2(a).



(a) Jacobi approach [43].

(b) Gauss-Seidel approach [43].

Figure 4.2: Two MDA fixed-point methods.

Gauss-Seidel [42] is another fixed-point approach. It differs from the Jacobi in that each analysis is run using the most recent outputs from other analysis, as they become available. This is possible because analyses are run in series, as seen in Fig. 4.2(b). As a result, the i^{th} equation of the system $Ax = b$ can be solved for x_i^k using the newly calculated x_m^k, $(m < i)$,

$$\begin{aligned} a_{11}x_1^k + a_{12}x_2^{k-1} + \ldots + a_{1n}x_n^{k-1} &= b_1 \\ a_{21}x_1^k + a_{22}x_2^k + \ldots + a_{2n}x_n^{k-1} &= b_2 \\ \vdots \quad \vdots \quad \ddots \quad \vdots \quad \vdots \\ a_{n1}x_1^k + a_{n2}x_2^k + \ldots + a_{nn}x_n^k &= b_n. \end{aligned} \tag{4.5}$$

$$\iff x_i^{(k)} = \frac{b_i - \sum_{j \le i} a_{ij}x_j^k - \sum_{j > i} a_{ij}x_j^{k-1}}{a_{ii}} \tag{4.6}$$

and in matrix form

$$(L + D)x^k + Ux^{k-1} = b \tag{4.7}$$

$$\iff x^k = (L + D)^{-1}(b - Ux^{k-1}) \tag{4.8}$$

where D, L and U are the diagonal, lower and upper triangular parts of matrix A.

Both Jacobi and Gauss-Seidel methods converge linearly when applied to linear systems, with Gauss-Seidel's convergence rate being twice as fast as Jacobi's [42]. Newton's method [44] converges quadratically when close enough to the solution, but in some situations it might not converge as well or not at all.

Chauhan et al. [45] compare coupled Newton (CN) and nonlinear block Gauss-Seidel (NLBGS) with Aitken's relaxation for a scalable problem with tunable total number of variables, degree of nonlinearity, coupling strength, and sparsity structure. They found that assembly time of the linear systems involved, linear solver efficiency, and strength of coupling play a major role in determining which approach is more efficient, not reaching a final conclusion on which is faster.

It is hard to choose the most efficient approach *a priori*. To address this problem, Chauhan et al. [46] propose a heuristic algorithm that switches between NLBGS and CN approaches using convergence rate information and iteration timings.

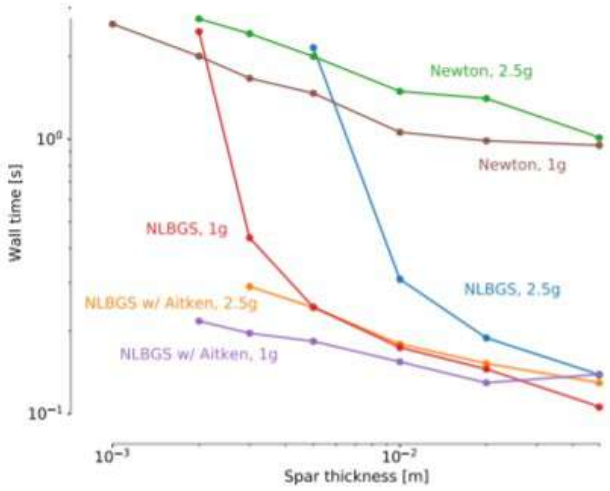


Figure 4.3: Comparison of solvers for the solutions of the coupled aerostructural system for level flight (1 g) and a pull-up maneuver (2.5 g) [4].

Coupling strength is a factor of great importance in MDA. In the aerostructural problem, it increases with load factor and with the decrease of spar thickness. Figure 4.3 shows a comparison of solvers for the aerostructural problem at several spar thickness values and two load factors. It can be seen that the strongly coupled problems take longer to converge for all solvers, with exception of NLBGS without Aitken's relaxation, which does not converge at all. Newton approaches converge always but take longer than NLBGS with Aitken's relaxation. Therefore, NLBGS with Aitken's relaxation is the solver used in this work.

4.3 Multidisciplinary Design Optimization

Multidisciplinary design optimization (MDO) refers to the use of numerical optimization for the design of systems that involve several disciplines or subsystems. In order to find the optimal design, it is necessary to carefully consider how the MDO is implemented. That is, how the discipline-analysis models are organized, algorithms of optimization and the problem formulation. This combination of problem formulation and organizational strategy is referred to as MDO architecture [47].

The two main types of architectures are monolithic and distributed. In a monolithic approach, a single optimization problem is solved, whereas in a distributed approach, the same problem is split into smaller subproblems. Martins and Lambe [47] present an extensive survey on all known MDO architectures in a unified notation, which enables comparisons between architectures and provides a good understanding on how some architectures are derived from others.

In this work, a variation of Multidisciplinary Feasible (MDF) architecture is implemented. Figure 4.4 depicts the extended design structure matrix (XDSM) [48] of the typical MDF architecture. MDF [49] is one of the most traditional and widely used MDO approaches. It is monolithic, so it consists in solving a single optimization problem in which the MDA module (Gauss-Seidel, in this example) takes in the design variables from the optimizer and iterates over the discipline analyses until the coupling variables have converged. Then, design and

coupling variables are used to compute the values of objective and constraint functions. Because the MDA is converged at each design point, MDF ensures that each optimization iteration is *multidisciplinary feasible*. This is desirable, because if the optimization is terminated prematurely, there is a physically realizable design point [50]. This does not mean, however, that design constraints are necessarily satisfied, as this depends on whether the optimization algorithm maintains a feasible design point [47].

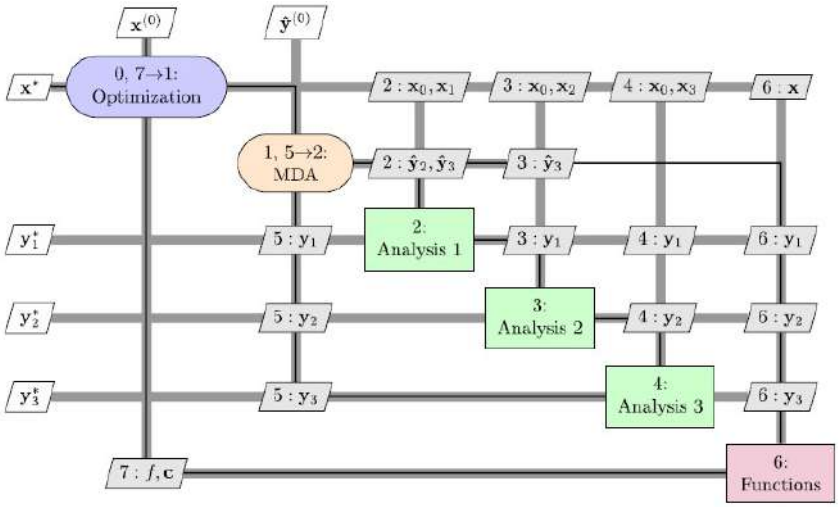


Figure 4.4: XDSM of MDF architecture with a Gauss–Seidel MDA [47].

4.4 Optimization Algorithms

The complexity of the optimization problem depends on the number of variables that compose the system, whether it can be described through continuous or discrete functions, whether it is linear or non-linear, among other factors. Based on the complexity of the problem and its characteristics, some optimization methods might be more suitable than others. For example, in a problem with continuous differentiable functionals, gradient-based methods deliver a solution efficiently, whereas, for discrete functions, gradient-free or heuristic methods are a better alternative. Figure 4.5 shows some optimization methods.

Deterministic and heuristic methods differ in the level of randomness. Deterministic algorithms are built upon mathematical models that make them rigid, meaning that for the same set of conditions they will not only produce the same results, but the path taken towards the optimal solution will be the same every time. On the other hand, heuristic methods will not necessarily follow the same path towards the solution due to the randomness factor they have into account.

Gradient based algorithms are widely used for solving a variety of optimization problems in engineering because of its efficiency in terms of the number of function evaluations required to find the optimum and the ability to solve problems with large number of design variables [51]. This and the fact that trajectory optimization problems are described by continuous functionals is why gradient based methods are commonly used in trajectory optimization. They typically obtain the optimum in a two-step process that comprises finding a suitable search direction d_k and a step size α_k that minimizes the solution vector x_k at iteration k along that direction,

$$x_{k+1} < x_k + \alpha_k d_k, , \tag{4.9}$$

which is done through first and sometimes second order derivatives information.

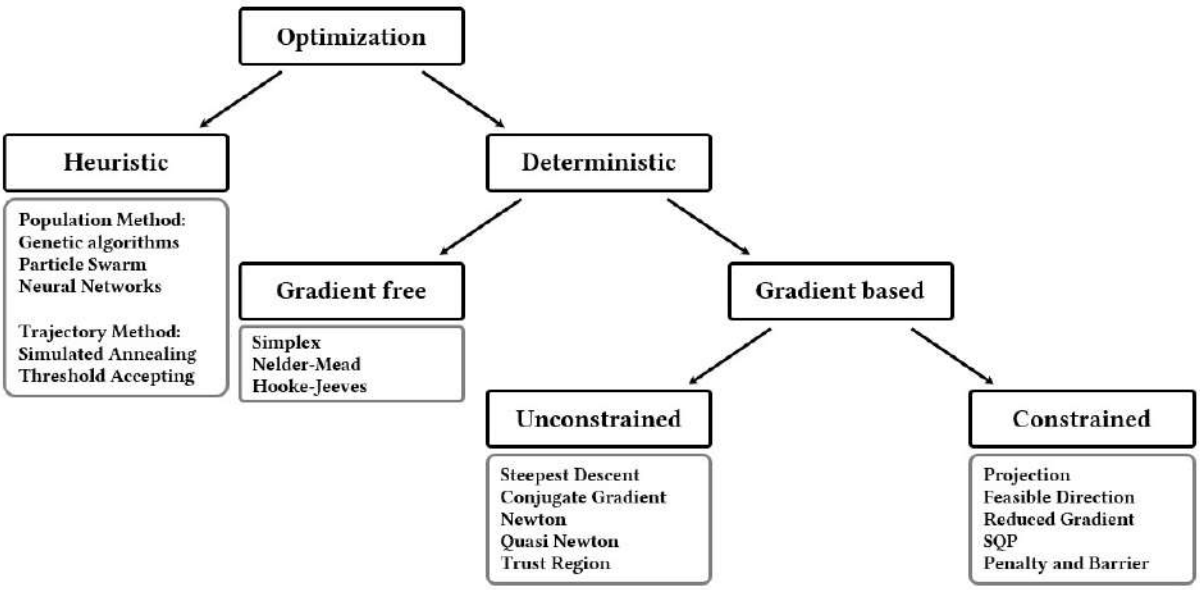


Figure 4.5: Optimization methods. Adapted from [43].

Unconstrained Optimization

In unconstrained gradient based optimization, there are no restrictions on the values design variables can assume, so search direction and the step size can be chosen freely. The general optimality conditions can be derived from the Taylor series expansion of f about the current design point x_k, with $g \equiv \nabla f$ and $H \equiv \nabla^2 f$,

$$f(x_k + q_k) \approx f(x_k) + q_k^T g_k + \frac{1}{2} q_k^T H_k q_k. \tag{4.10}$$

For x_k to be a local minimum, $f(x_k + q_k) \geq f(x_k)$ must be true for any vector q_k, which is equivalent to $f(x_k + q_k) - f(x_k) \geq 0$. For x_k to be a stationary point, every component of the gradient vector $g(x_k)$ must be zero and the sign of the eigenvalues of the Hessian matrix H is what will determine whether x_k is a saddle point or a local minimum. If the eigenvalues are positive or zero, H is positive semi-definite and x_k is a local minimum. If all eigenvalues are strictly positive, then H is positive definite and x_k is a strong local minimum. If the sign of the eigenvalues is mixed, then x_k is a saddle point.

Methods differ in how much information they use to compute the search direction and to evaluate whether a design point is optimal. An example of a first order method is Steepest Descent. It relies on the basic principle that the gradient vector at one point gives the direction of a function's maximum rate of increase at that point, and the rate of increase is given by the norm. First, the gradient, the normalized search direction and the step length are computed, then the convergence conditions are evaluated,

$$|g(x_k)| \leq \epsilon_g \tag{4.11}$$

$$|f(x_{k+1}) - f(x_k)| \leq \epsilon_f \tag{4.12}$$

where ϵ are arbitrary tolerances. If they are satisfied for two successive iterations, the optimization stops.

An example of a second order method is Newton's method. The function is approximated by the second order

Taylor series expansion, as in Eq. (4.10), and then it is differentiated with respect to q_k and set to zero, which yields

$$g_k + H_k q_k = 0. \tag{4.13}$$

Solving Eq. (4.13) automatically gives the search direction $d_k = q_k$ since the step size in the Newton's method is $\alpha = 1$. It may have some convergence problems if f is nonlinear or poorly approximated by a quadratic model.

Constrained Optimization

Engineering problems are almost always limited in some way, be it the maximum power output of an engine or the maximum stress a structure can withstand. Constrained optimization is often nonlinear, either because of variables or the constraints themselves are nonlinear. As seen before, NLP problems can be posed as

$$\begin{aligned} \text{Minimize} \quad & f(\boldsymbol{x}) \\ \text{w.r.t} \quad & \boldsymbol{x} \\ \text{subject to} \quad & g_i(\boldsymbol{x}) \leq 0, \quad (i = 1, \dots, l) \\ & h_j(\boldsymbol{x}) = 0, \quad (j = 1, \dots, m). \end{aligned} \tag{4.14}$$

The necessary optimality conditions for nonlinear constrained problems are given by the Karush-Kuhn-Tucker (KKT) conditions [52]. Considering first just equality constraints, for a point to be stationary, the total differential of $f(x)$ must be zero,

$$df = \nabla f^T d\boldsymbol{x} = 0, \tag{4.15}$$

and for it to be feasible,

$$dh_j = \nabla h_j^T d\boldsymbol{x} = 0, \quad (j = 1, \dots, m). \tag{4.16}$$

If both are zero, each constraint variation can be multiplied by the scalar vector λ_j, called Lagrange multiplier, and added to the objective function variation.

$$\frac{\partial f}{\partial x_k} + \sum_{j=1}^{l} \lambda_j \frac{\partial h_j}{\partial x_k} = 0, \quad (k = 1, \dots, n). \tag{4.17}$$

Defining the Lagrangian function as

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = f(\boldsymbol{x}) + \boldsymbol{\lambda}^T \boldsymbol{h}(\boldsymbol{x}), \tag{4.18}$$

Eq. (4.17) is its derivative with respect to $\boldsymbol{x}$ and the optimality conditions can be expressed in terms of the Lagrangian as

$$\frac{\partial \mathcal{L}}{\partial x_k} = \frac{\partial f}{\partial x_k} + \sum_{j=1}^{l} \lambda_j \frac{\partial h_j}{\partial x_k} = 0, \quad (k = 1, \dots, n), \tag{4.19a}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_j} = h_j(x) = 0, \quad (j = 1, \dots, m). \tag{4.19b}$$

The problem originally with n variables and m constraints is transformed into an unconstrained one with $n + m$ variables and can be solved by looking for the stationary points of the Lagrangian function. Another way of looking at the optimality condition (4.19a) is that at the solution, there is a scalar λ^* that makes $\nabla h(x^*)$ and $\nabla f(x^*)$ parallel

$$\nabla f(x^*) = -\lambda^* \nabla h(x^*). \tag{4.20}$$

If there is a vector $\boldsymbol{d}$ which ensures feasibility $\nabla h_j^T(x_k)\boldsymbol{d} = 0$ and that the direction of optimization is a descent one $\nabla f^T(x_k)\boldsymbol{d} < 0$, a solution is optimal only if neither has room for improvement, which happens when they are parallel.

The same logic can be applied to inequality constraints. In this case, beside the descent direction, there is a feasible one $\nabla g_i^T(x_k)\boldsymbol{d} < 0$ for each active constraint $g_i(x_k) = 0$. If $\boldsymbol{x}^*$ lies within the region delimited by $\boldsymbol{g}(x) \leq 0$, there are no active constraints and the necessary optimality condition is the same as in unconstrained problems $\nabla f(\boldsymbol{x}^*) = 0$. If some constraints are active, then it is necessary that there is no vector $\boldsymbol{d}$ at the point $\boldsymbol{x}^*$ whose direction is both descent and feasible. This yields a result similar to Eq. (4.20), in this case with the Lagrange multiplier μ,

$$-\nabla f = \sum \mu_i \nabla g_i \quad (\mu_i \geq 0). \tag{4.21}$$

A scalar $\mu_i \geq 0$ is also assigned for inactive constraints and the condition $\mu_i g_i = 0$ is imposed.

The necessary optimality conditions for a problem like (4.14) are given by

$$\text{Optimality} \quad \nabla_x \mathcal{L} = 0 \Longrightarrow \frac{\partial \mathcal{L}}{\partial x_k} = \frac{\partial f}{\partial x_k} + \sum_{i=1}^{m} \mu_i \frac{\partial g_i}{\partial x_k} + \sum_{j=1}^{l} \lambda_j \frac{\partial h_j}{\partial x_k} = 0 \qquad (k = 1, \dots, n) \tag{4.22a}$$

$$\text{Non negativity} \quad \mu_i \geq 0 \qquad (i = 1, \dots, l) \tag{4.22b}$$

$$\text{Complementary slackness} \quad \mu_i g_i = 0 \qquad (i = 1, \dots, l) \tag{4.22c}$$

$$\text{Feasibility} \quad g_i \leq 0 \qquad (i = 1, \dots, l) \tag{4.22d}$$

$$\nabla_\lambda \mathcal{L} = 0 \Longrightarrow \frac{\partial \mathcal{L}}{\partial \lambda_j} = h_j(x) = 0 \qquad (j = 1, \dots, m) \tag{4.22e}$$

Conditions (4.22a)-(4.22e) only guarantee that the design point is stationary, thus requiring the analysis of second order information. A point $\boldsymbol{x}^*$ is a minimum if the Hessian matrix of the Lagrangian,

$$\nabla^2 \mathcal{L} = \nabla^2 f(\boldsymbol{x}^*) + \sum_{i=1}^{m} \mu_i \nabla^2 g_i(\boldsymbol{x}^*) + \sum_{j=1}^{l} \lambda_j \nabla^2 h_j(\boldsymbol{x}^*), \tag{4.23}$$

is positive definite in the feasible space.

Sequential Quadratic Programming

Sequential Quadratic Programming (SQP) methods are the state of the art in nonlinear programming methods. Nonlinear constrained problems like Eq. (4.14) are solved by building a sequence of quadratic programming subproblems where the constraints are linearized and the objective function is a quadratic approximation of the Lagrangian function. New x_{k+1}^* are iteratively computed using merit functions that determine step sizes along

the search directions obtained from solving these subproblems.

Boggs and Tolle [53] present a detailed review of SQP and Gould and Toint [54] present a survey of a number of SQP methods and compare them. Lyu et al. [55] did a benchmark study where they tested several optimization algorithms for a multi-dimensional Rosenbrock function, for a RANS-based aerodynamic twist optimization problem and for an aerodynamic shape optimization problem. They concluded that gradient-free methods require two to three orders of magnitude more computational effort than gradient-based methods. In particular, SNOPT and SLSQP algorithms stood out, both SQP algorithms.

SNOPT [56] is a commercial software that uses a smooth augmented Lagrangian merit function, whose Hessian is approximated using a limited-memory quasi-Newton method. It also uses a reduced-Hessian semidefinite QP solver for the QP subproblems. SLSQP [57] is an evolution of Lawson and Hanson [58] nonlinear least-squares solver. It uses a quasi-Newton Hessian approximation and an L1-test function in the line search algorithm.

Given the advantage of gradient based methods and the good performance of the open source SLSQP, this is the algorithm employed in this work.

Sensitivity analysis

Sensitivity analysis is an essential step of gradient based optimization. It refers to the evaluation of how parameters influence the performance metrics, objective function, state and constraints. It is desired that the computation of gradients is as precise and inexpensive as possible, which often involves a compromise between the two. Methods to compute derivatives include finite differences approximations, complex-step approximations and analytical methods [43].

Finite differences are derived from the Taylor series expansion about x,

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(x) + ... \tag{4.24}$$

where truncating for the first order derivative yields

$$f'(x) = \frac{f(x+h) - f(x)}{h} + O(h) \tag{4.25}$$

where h is the step and $O(h)$ is the truncation error. This forward difference estimate converges linearly with decreasing step size. However, there is a limit to how small the step size can be, as decreasing it too much leads to subtractive cancellation errors. These occur when the step size is so small that the number of digits used in computation is no longer sufficient to distinguish $f(x+h)$ from $f(x)$, yielding a numerator equal to zero.

Complex-step approximations estimate the derivatives of real functions with complex calculus. They are also derived from a Taylor series expansion, but with a pure imaginary step ih,

$$f(x+ih) = f(x) + ihf'(x) - \frac{h^2}{2!}f''(x) - \frac{ih^3}{3!}f'''(x) + ... \tag{4.26}$$

Taking only the imaginary parts and dividing it by h, the complex-step derivative approximation is obtained

$$f'(x) = \frac{\mathcal{I}[f(x+h)]}{h} + O(h^2). \tag{4.27}$$

The main advantage of this approach is that it does not have the subtractive cancellation problems encountered in finite differences, as no subtraction is made at all.

Analytical methods are the most accurate and efficient option for sensitivity analysis [43]. The total sensitivity of a function of interest f is computed, which in general is dependent on the design and state variables, x_n and y_i, respectively. The state is related to the design variables through the governing equations,

$$\mathcal{R}(x_n, y_i(x_n)) = 0. \tag{4.28}$$

The total sensitivity of f is given by

$$\frac{df}{dx_n} = \frac{\partial f}{\partial x_n} + \frac{\partial f}{\partial y_i}\frac{dy_i}{dx_n} \tag{4.29}$$

and the total derivative of the governing equation is

$$\frac{d\mathcal{R}}{dx_n} = \frac{\partial \mathcal{R}}{\partial x_n} + \frac{\partial \mathcal{R}}{\partial y_i}\frac{dy_i}{dx_n} = 0. \tag{4.30}$$

The partial derivatives can be easily computed by varying the denominator and re-evaluating the numerator, but the total derivatives require the solution of the multidisciplinary problem. Equation (4.30) can be rewritten as

$$\frac{dy_i}{dx_n} = -\left[\frac{\partial \mathcal{R}}{\partial y_i}\right]^{-1}\frac{\partial \mathcal{R}}{\partial x_n} \tag{4.31}$$

and substituting in Eq. (4.29) yields

$$\frac{df}{dx_n} = \frac{\partial f}{\partial x_n} - \underbrace{\frac{\partial f}{\partial y_i}\overbrace{\left[\frac{\partial \mathcal{R}}{\partial y_i}\right]^{-1}}^{-dy/dx_n}}_{-\Psi_k}\frac{\partial \mathcal{R}}{\partial x_n}. \tag{4.32}$$

Two approaches to compute the analytical sensitivities arise: direct and adjoint methods. The difference between them lies in the order used to calculate the last term of Eq. (4.32). Direct methods solve Eq. (4.31) directly and substitutes the result, whereas adjoint methods solve the adjoint system

$$\frac{\partial \mathcal{R}}{\partial y_i}\Psi_k = -\frac{\partial f}{\partial y_i}, \tag{4.33}$$

where Ψ_k is the adjoint vector.

This procedural difference has a significant impact in the cost of computing the total sensitivity, even though partial derivative terms are the same for both methods. In the direct approach, $\frac{dy_i}{dx_n}$ is valid for any function f, but must be computed for each design variable (n_x times), whereas in the adjoint, Ψ_k does not depend on the design variables and is computed once for each f, n_f times. Therefore, the most efficient approach is the direct method if $n_x < n_f$, or adjoint if otherwise.

4.5 Aerostructural Analysis and Optimization Tool

Multidisciplinary Analysis and Optimization (MDAO) is implemented through the OpenMDAO framework [59]. It is an open source software framework written in Python that achieves high computational efficiency through new hierarchical strategies to exploit problem structure.

OpenMDAO also provides a framework for computing coupled derivatives efficiently. The Modular Analysis and Unified Derivatives (MAUD) architecture [60] formulates the multidisciplinary model as a nonlinear system of equations, leading to a linear equation that unifies all methods for computing derivatives. It works by concatenating all design, state and coupled variables, as well as output functions, into one vector. Then, in a similar way, all residual functions associated with each variable are assembled into one *fundamental system* that represents an unified formulation of the numerical model. Finally, the derivatives are obtained using the *unifying derivative equation*, developed by Martins and Hwang [61]. Its main advantage is that it enables flow-based frameworks that use the MAUD architecture to provide a common interface for the chain rule, adjoint method, coupled adjoint method, and hybrid methods. Furthermore, it automatically uses the appropriate method for the problem at hand [60].

This is a useful feature, specially for gradient-based optimization. Nevertheless, any kind of optimizer is supported through the use of packages such as PyOptSparse [62] and SciPy [63]. OpenMDAO supports several MDA solvers too, as for example linear Krylov, linear and nonlinear block Jacobi and Gauss-Seidel, Newton and Broyden.

OpenMDAO's modular environment allows an easy integration of discipline analyses into a larger multidisciplinary model [59], which is the objective of this work. For the aerostructural disciplines in particular, the OpenAeroStruct (OAS) framework [4] is employed. It is a low-fidelity aerostructural analysis and optimization tool, developed in the OpenMDAO framework, that uses the VLM and FEM methods described in sections 2.3 and 2.4 to model lifting surfaces. OAS is built with fuel burnt or range as the optimization objectives in mind, which are calculated through the Breguet equation. This and all fuel related aspects of OAS are changed to accommodate trajectory optimization and electric propulsion, as discussed in Sec. 2.5.

The fluid-structure interaction (FSI) requires that the aerodynamic loads and structural displacements be transferred between the coupled analysis. The scheme used in OAS satisfies the requirements of being consistent and conservative.

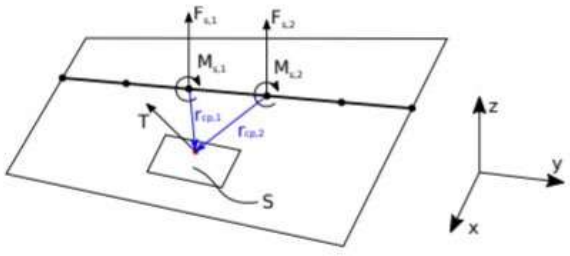


Figure 4.6: Transfer scheme used in OAS. Adapted from [4].

It can be seen in Fig. 4.6 that the edges of the panels are aligned with the structural nodes. This is a simplification made in the transfer scheme used in OAS, that the spanwise discretization of the aerodynamic and structural models are the same. Furthermore, the aerodynamic load is applied on the center of pressure (*cp*), located at the

spanwise centerline, so half of the aerodynamic load is applied on each of the structural nodes. The equivalent nodal force and moment on one of the structural nodes are given by

$$\boldsymbol{F}_{s,i} = \int_{panel} \frac{1}{2}\boldsymbol{T}dS = \frac{1}{2}\boldsymbol{T}S \tag{4.34}$$

$$\boldsymbol{M}_{s,i} = \int_{panel} \boldsymbol{r}_i \times \frac{1}{2}\boldsymbol{T}dS = \boldsymbol{r}_{cp,i} \times \frac{1}{2}\boldsymbol{T}S. \tag{4.35}$$

These equations immediately satisfy the consistency requirement, as they are equivalent results of the aerodynamic load distribution.

The conservative requirement is verified through the virtual works of both meshes. The virtual work on the structural mesh is given by

$$\delta W_s = \sum_{i=1}^{2} \left(\boldsymbol{F} \cdot \delta\boldsymbol{d}_{s,i} + \boldsymbol{M} \cdot \delta\boldsymbol{\theta}_{s,i}\right) \tag{4.36}$$

and substituting force and moment by equations (4.34) and (4.35), yields

$$\delta W_s = \frac{1}{2}\sum_{i=1}^{2} \left(\boldsymbol{T} \cdot \delta\boldsymbol{d}_{s,i} + (\boldsymbol{r}_{cp,i} \times \boldsymbol{T}) \cdot \delta\boldsymbol{\theta}_{s,i}\right). \tag{4.37}$$

The virtual work on the aerodynamic mesh is written as

$$\delta W_a = \int_{panel} \boldsymbol{T}\delta\boldsymbol{u}_a dS \tag{4.38}$$

where the displacements are given by

$$\boldsymbol{u}_a = \frac{1}{2}\sum_{i=1}^{2} \left(\boldsymbol{d}_{s,i} + \boldsymbol{\theta}_{s,i} \times \boldsymbol{r}_{s,i}\right). \tag{4.39}$$

Combining equations (4.38) and (4.39) leads to

$$\delta W_a = \frac{1}{2}\sum_{i=1}^{2} \int \left(\boldsymbol{T} \cdot \delta\boldsymbol{d}_{s,i} + \boldsymbol{T} \cdot (\delta\boldsymbol{\theta}_{s,i} \times \boldsymbol{r}_{s,i})\right) dS \tag{4.40}$$

where $\boldsymbol{T}$, $\boldsymbol{d}_{s,i}$ and $\boldsymbol{\theta}_{s,i}$ are constant over the panel, so it can be rewritten as

$$\delta W_a = \frac{1}{2}\sum_{i=1}^{2} \left(\boldsymbol{T} \cdot \delta\boldsymbol{d}_{s,i} + \boldsymbol{T} \cdot (\delta\boldsymbol{\theta}_{s,i} \times \boldsymbol{r}_{cp,i})\right) S. \tag{4.41}$$

By vector algebra,

$$\boldsymbol{T} \cdot (\delta\boldsymbol{\theta}_{s,i} \times \boldsymbol{r}_{cp,i}) = (\delta\boldsymbol{\theta}_{s,i} \times \boldsymbol{r}_{cp,i}) \cdot \boldsymbol{T} = \delta\boldsymbol{\theta}_{s,i} \cdot (\boldsymbol{r}_{cp,i} \times \boldsymbol{T}) = (\boldsymbol{r}_{cp,i} \times \boldsymbol{T}) \cdot \delta\boldsymbol{\theta}_{s,i} \tag{4.42}$$

which makes Eq. (4.5) equivalent to

$$\delta W_a = \frac{1}{2}\sum_{i=1}^{2} \left(\boldsymbol{T} \cdot \delta\boldsymbol{d}_{s,i} + (r_{cp,i} \times \boldsymbol{T}) \cdot \delta\boldsymbol{\theta}_{s,i})\right) S. \tag{4.43}$$

With this, we arrive at the conclusion that $\delta W_a = \delta W_s$ and the transfer scheme is conservative.

4.6 Framework Implementation

The MDO architecture in this work is a variation of MDF. The usual MDF, as described in Sec. 4.3, handles the task of computing each discipline's states to the MDA, which does so by running a fixed point iterative solver. Once it has converged, the objective function and constraints are computed to be evaluated by the optimizer.

In this case, the coupled analysis are structures and aerodynamics, coupled through the loads and mesh displacements states, and subject, among other, to the flow variables. These flow variables, such as air density and velocity, result from the trajectory (or are prescribed for the isolated design problem) and therefore may vary from one flight point to another. Hence, the fluid-structure interaction needs to be solved for every point of the trajectory discretization.

OpenAeroStruct has a multipoint feature that allows optimizing the aircraft design for several flight conditions. However, is not done in a vectorized way. Instead, a new "mission point" is created for each set of flight conditions and a new MDA is called to solve the FSI.

Figure 4.7 shows the extended design structure matrix of the architecture.

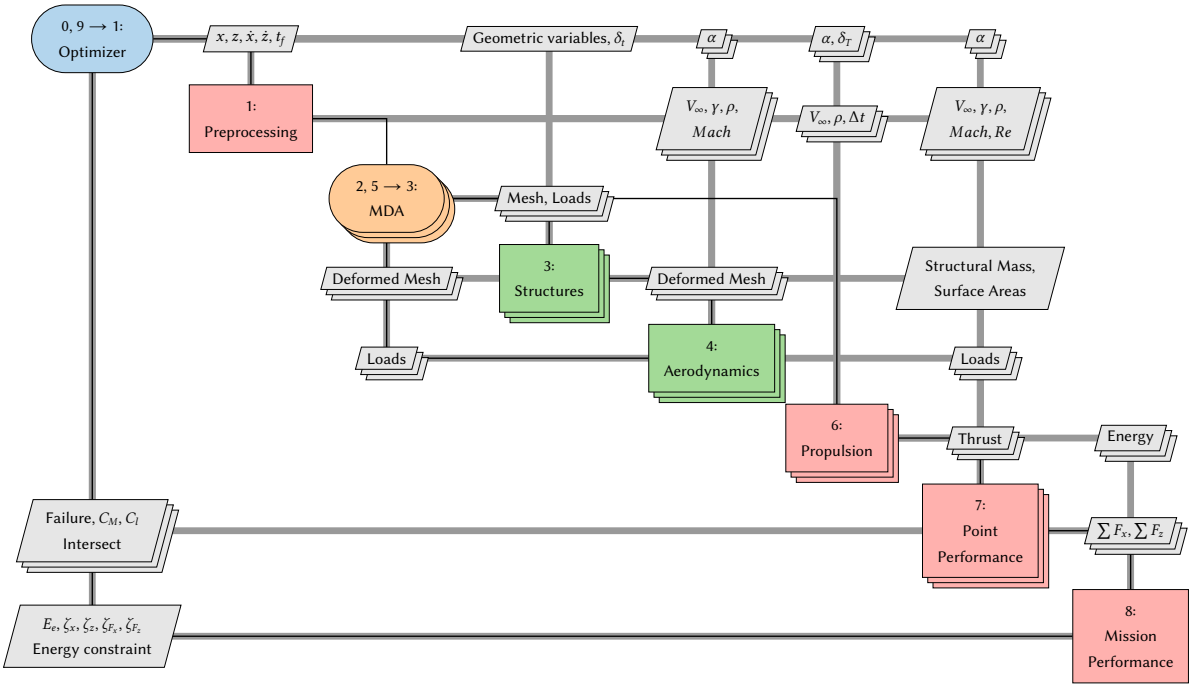


Figure 4.7: Extended design structure matrix of the implemented architecture.

The trajectory variables are inputs to the Preprocessing component, where the flow variables are computed and then forwarded to Aerodynamics and Point Performance components. Each MDA receives a set of design variables. The Structures component receives the geometric variables, which are the same for all MDAs, and the correspondent entry of the stabilator angle vector. The Aerodynamics component receives the angle of attack. The MDA is run until convergence is achieved and once it does, thrust and energy spent during the mission point are calculated in the Propulsion component.

The thrust force, the final load distribution, the structural mass and surface areas are then passed to the Point Performance component, where a series of functions are computed to obtain lift and drag coefficients, total weight, center of mass and moments. All these intermediate computations are necessary to calculate the values

of failure, material intersection and moments coefficient constraints, which are then evaluated by the optimizer. They are also necessary to compute the sum of forces that are then passed to Mission Performance, in order to get the value of the collocation constraints. The energies of all mission points, calculated in Propulsion, are summed in Mission Performance to obtain the total energy spent and the energy constraint value.

MDAs are independent from each other and there is no data flow between them. This is a consequence of having chosen a collocation method. Since states and control are handled directly by the optimizer, variables dependent on trajectory and control can be vectorized and only the correspondent entries are passed to each MDA. Alternatively, a shooting method would require integrating the equations of motion within each mission point and passing the result to next one. Since trajectory states are inputs to the aerostructural analyses, this approach creates a stronger dependency between MDAs, which might not be beneficial for convergence given that errors and results obtained from bad initial guesses are propagated. The collocation approach not only prevents this, but also allows a modular and much easier integration into the OAS framework.

This framework implementation results in two very distinct philosophies for FSI and trajectory. The FSI part of the optimization is carried through a solver-based approach, in which the governing equations are solved, whereas trajectory is handled directly by optimizer, where the states are design variables chosen so that the associated constraints are satisfied.

Figure 4.8 is the N2 diagram [59] of the complete model. The hierarchy tree of the model's components is seen on the left, where systems and subsystems can be subdivided down to variables. The dependencies between components are depicted on the right, where the dark squares on the main diagonal stand for the subsystems and the light squares represent the input/output relationship between subsystems. The outputs of a subsystem are represented by the light squares on the same horizontal line and the inputs, by the light squares vertically aligned. This way, squares above the main diagonal are feedforward and squares below are feedback.

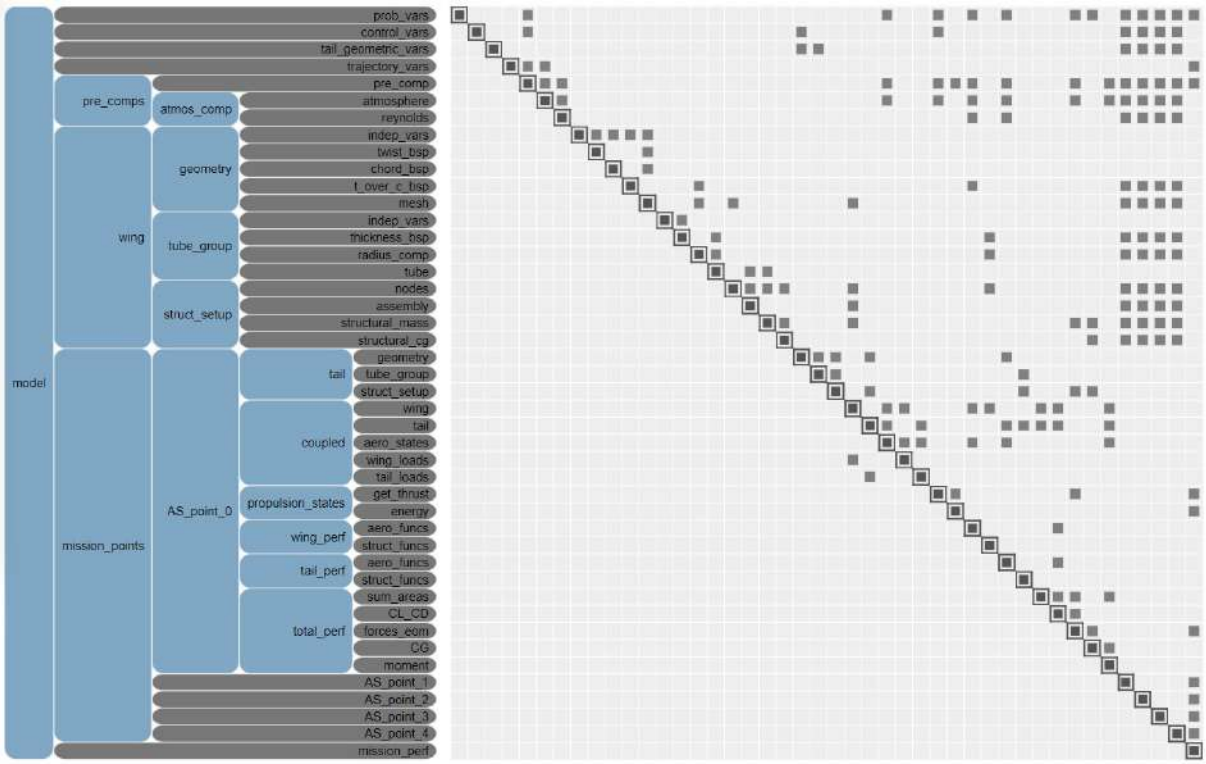


Figure 4.8: Hierarchy tree of the overall model.

The trajectory states are preprocessed in the first subsystem *pre_comps*, where preliminary computations such as calculating the freestream velocity V_∞ and the flight path angle γ are handled. It also computes atmospheric quantities based on altitude, such as air density ρ, Mach number, speed of sound and Reynolds number.

The subsystem *wing* receives the wing's geometric variables as inputs, which can be design variables or user input values, and generates the mesh for the wing. Its geometry is constant throughout the entire mission, as morphing is not being considered. For that reason, this subsystem sits outside of *mission_points*.

The *mission_points* subsystem contains all the flight points (five in this example) where the coupled aerostructural interaction and point performance functions are evaluated. Figure 4.9 is the model zoomed in on a single mission point.

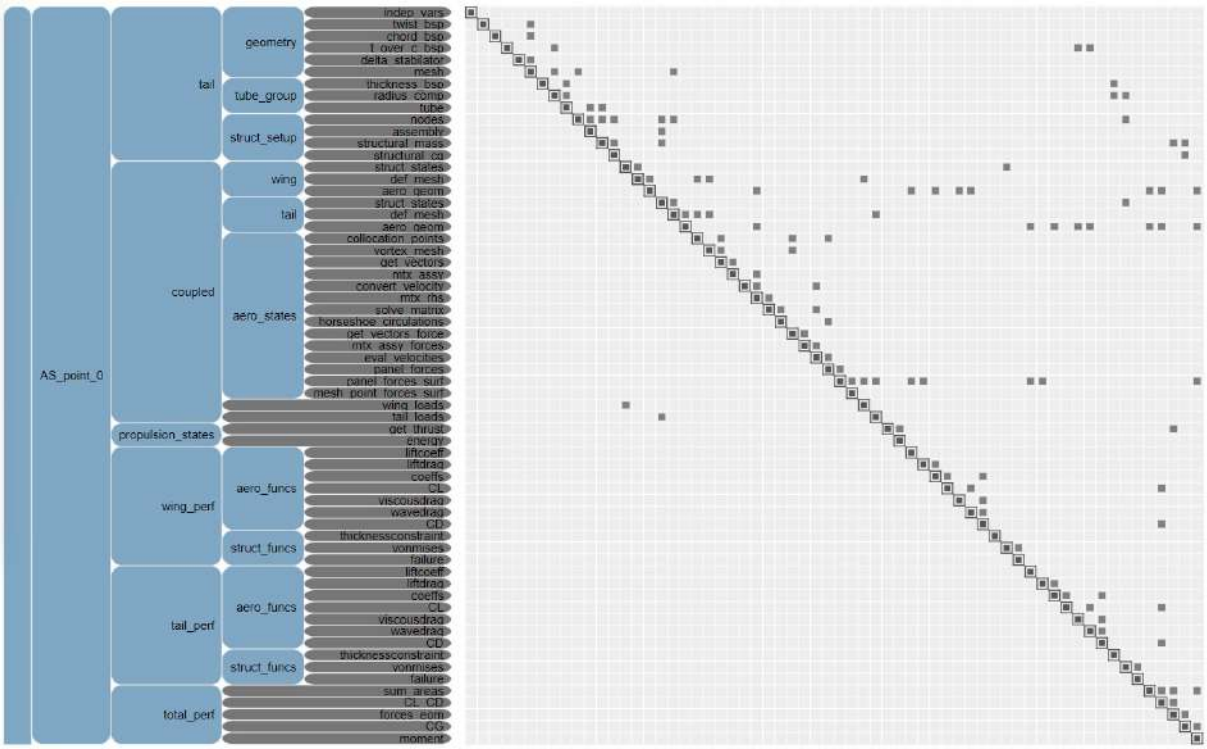


Figure 4.9: Hierarchy tree zoomed on one mission point.

The subsystem *tail* is equivalent to the previously seen *wing*. It is responsible for generating a mesh for the tail, which works as stabilator, as mentioned before. There is no morphing in this surface either, but since its angle is a control variable used to trim the aircraft, its mesh is rotated during flight and therefore varies from point to point, reason why this subsystem lies within each mission point.

The *wing* and *tail* subsystems give the meshes (one global and the other local) necessary to start the coupled analysis. As detailed for the wing in Fig. 4.10, an initial guess is made for each surface's loads, which results in structural displacements that are computed in *struct_states* using the FEM model described in Sec. 2.4. In turn, the displacements are used to compute the deformed mesh (*def_mesh*).

The deformed mesh is then passed on to the aerodynamics analysis (*aero_states*), as shown in Fig. 4.11. In this subsystem, circulations and panel forces are computed with the VLM panel method described in Sec. 2.3. The matrix and the right hand side of the linear system are assembled in *mtx_rhs*, and the system itself is solved in *solve_matrix*. Forces are calculated in *panel_forces* and from these, a new set of loads is computed and passed

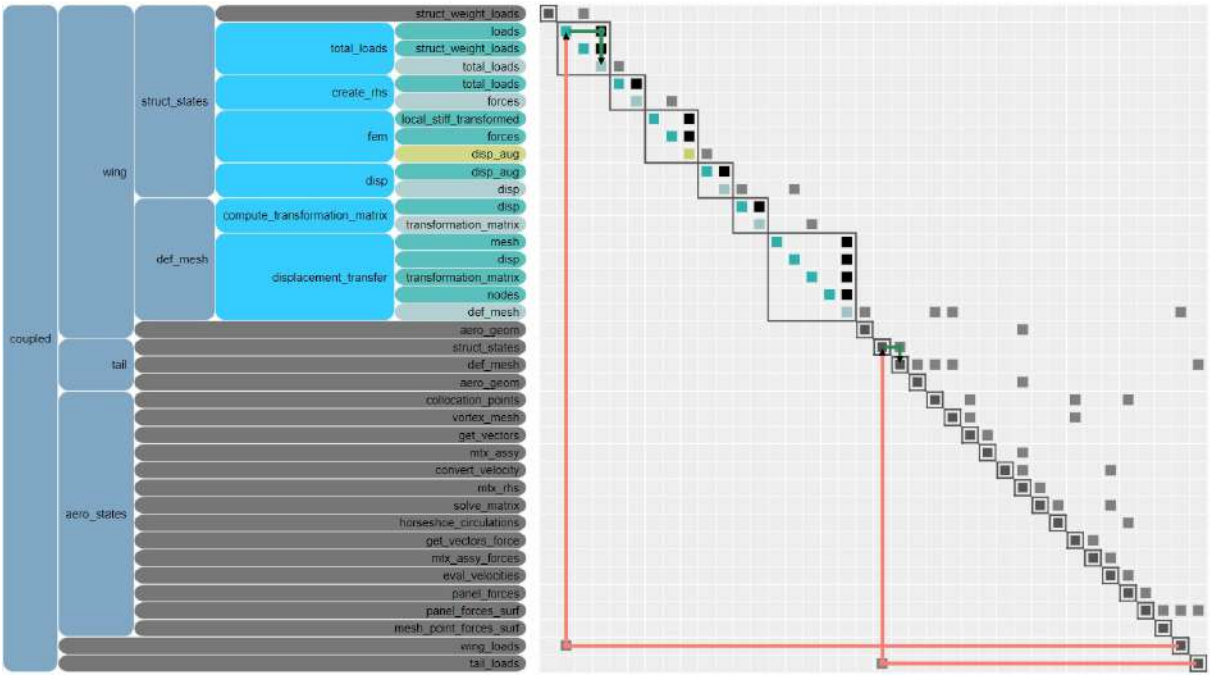


Figure 4.10: Hierarchy tree zoomed on the structural analysis of the coupled system.

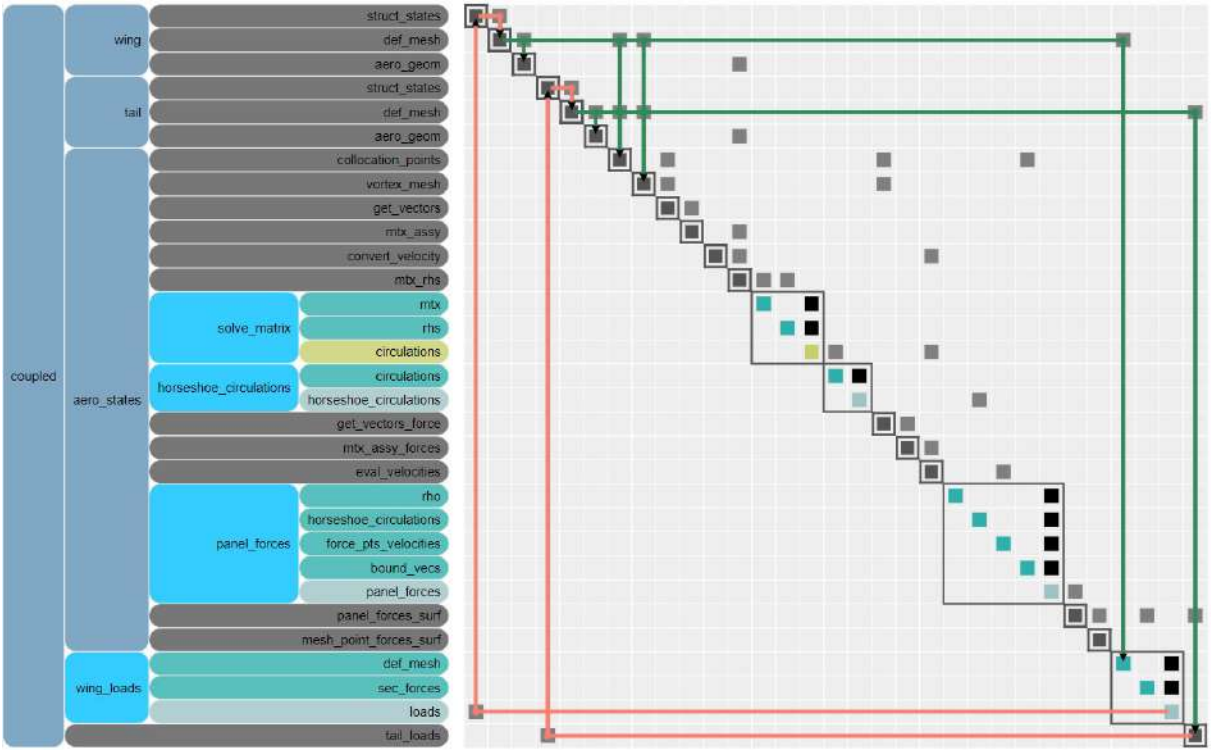


Figure 4.11: Hierarchy tree zoomed on the aerodynamic analysis of the coupled system.

again to the structural analysis, thus closing the cycle.

Once the MDA has converged, the resultant set of aerodynamic forces and structural stresses are forwarded to *wing_perf* and *tail_perf*, seen in Fig. 4.9, to obtain the lift and drag coefficients, the intersection constraint value (Eq. (2.64)), the Von Mises stress and the failure constraint value (Eq. (2.63)).

The subsystem *propulsion_states* receives the throttle control, angle of attack, freestream velocity and air

density, and outputs the energy consumed during that mission point and the thrust force. The latter is an input to *total_perf*, along with the total lift and drag of each surface. These are used in *forces_eom* to compute the sum of forces for the equations of motion (2.7a) and (2.7b). Additionally, the center of gravity is calculated and the panel forces passed on to *moment* in order to compute the moments.

This whole process is done for as many points as the trajectory is discretized into. Subsequently, the sum of forces and the energy consumed at every point is forwarded to *mission_perf*. Here, the energies are summed to obtain the total energy spent and the value of the energy constraint. The forces are assembled into vectors to calculate the value of collocation constraints.

Chapter 5

Sample Applications

In this chapter, optimization results are presented and discussed. The optimization is first performed for the three problems previously defined (DP, TP and DTP) with the objective of minimizing energy. Then, it is performed for minimum flight time and maximum range. Before that, some considerations are made regarding the problem setup and the desgin framework parameters.

5.1 Baseline Problem Definition

In this section, the baseline parameters common to all optimizations are presented. The initial trajectory was the same for all problems in climb optimization, with energy and time as objectives, but different for maximum distance, which are shown in the following sections.

5.1.1 Solver and Optimizer Parameters

As said before, the MDA employed in this work is NLBGS with Aitken's relaxation. As for the optimizer, it was concluded that gradient based methods performed better and that the best open source option was SLSQP. For these reasons, it is the chosen algorithm.

The solver and optimizer tolerances are presented in Tab. 5.1.

Table 5.1: Gauss-Seidel and SLSQP optimizer tolerances.

Parameter	Value
Solver absolute error tolerance	10^{-7}
Solver relative error tolerance	10^{-30}
Optimizer tolerance	10^{-3}

5.1.2 Mesh Convergence Study

Generally, a very refined mesh tends to yield better and more accurate results, but at the expense of computational time and effort. Therefore, a compromise must be made, which is why a mesh convergence study was performed.

The mesh of each lifting surface is defined by the number of nodes in the chord and span directions, *num_x* and *num_y*, respectively. One MDA was converged for several (*num_x*, *num_y*) pairs and the computation times and resultant drag coefficients C_D of an arbitrary wing shape and flow were compared. The number of nodes was varied from 2 to 13 in the chord direction and from 5 to 41 in the span direction. The tail's mesh was kept fixed. Results are shown in Fig. 5.1.

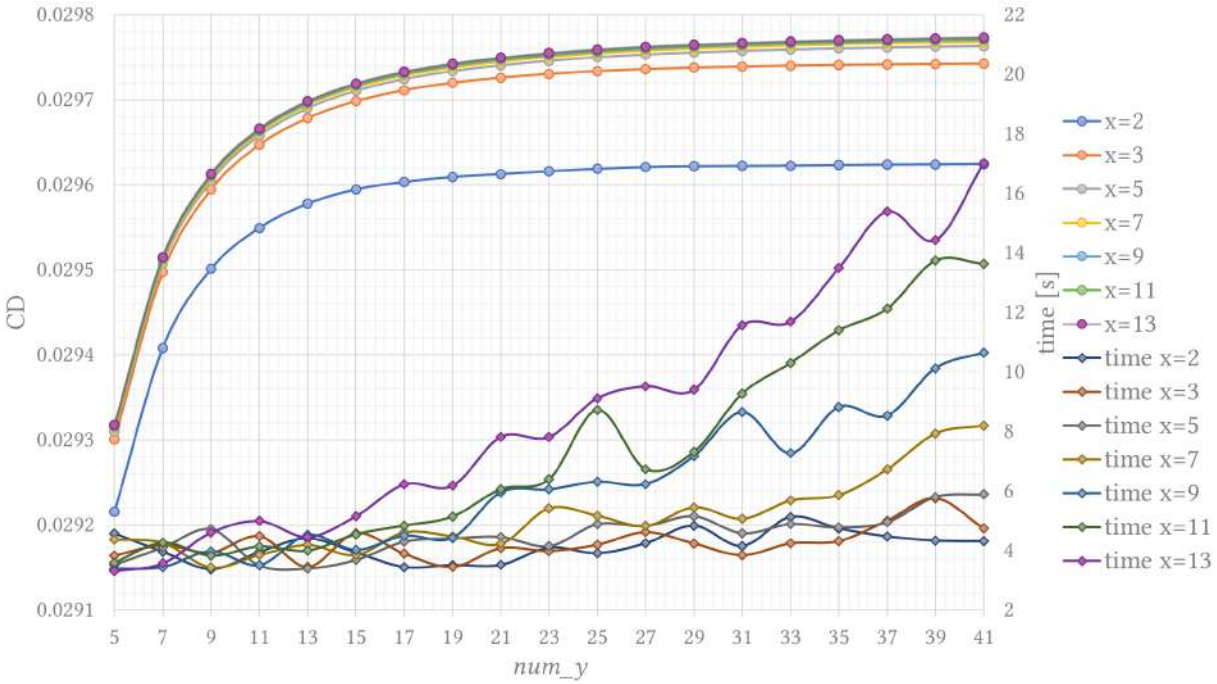


Figure 5.1: Results of the mesh convergence study.

It can be seen that increasing the number of panels yields larger C_D up to a certain point, where it starts to stabilize. Ideally, the chosen number of panels should lie in that region. However, despite the oscillations, the trend is to require more computational time. There is a considerable offset between the ($x = 2$) curve and the others, which means that this option has a lower accuracy. As for the others, they are very close, which makes it hard to weigh the accuracy gain/loss.

To better understand this, the variation in C_D relative to the previous result due to increasing *num_x* for a fixed *num_y* is represented in Tab. 5.2. From this table, it can be concluded that the variation is marginal (below 0.01%) after 7 longitudinal nodes, reason why that is the chosen value.

In a similar manner, Tab. 5.3 shows the variation of C_D relative to the previous result for increasing *num_y* and *num_x* = 7. Although variations start to be negligible after 29 nodes, that is not the only criterion to determine the number of nodes in this case. The ratio of panels should match the aspect ratio of the wing, so that panels are squared and not too skewed. It not possible to predict the final aspect ratio of wing, but it can be expected to be as high as possible due to the aerodynamic advantages it brings. For the 6 chord panels chosen, an aspect ratio of 10 would mean 60 panels in the spanwise direction. As stated before, this is not possible due to computational limitations, so a compromise in this regard must be made as well. Given this mesh convergence data, 31 nodes not only produce an adequate panel ratio, increasing it yields negligible result changes.

Therefore, the number of panels for the wing is 6×30, whereas for the tail, a panel ratio of 3 is applied, given

that the expected aspect ratio is also lower: 4 × 12.

Table 5.2: C_D convergence with *num_x* for fixed *num_y*.

num_y	*num_x*						
	2	3	5	7	9	11	13
5	-	0.291%	0.031%	0.013%	0.007%	0.005%	0.003%
7	-	0.304%	0.032%	0.013%	0.007%	0.005%	0.003%
9	-	0.317%	0.034%	0.013%	0.007%	0.005%	0.003%
11	-	0.331%	0.036%	0.013%	0.007%	0.005%	0.003%
13	-	0.341%	0.039%	0.013%	0.007%	0.005%	0.003%
17	-	0.365%	0.044%	0.014%	0.007%	0.005%	0.003%
19	-	0.374%	0.047%	0.014%	0.007%	0.005%	0.003%
21	-	0.383%	0.050%	0.014%	0.007%	0.005%	0.003%
23	-	0.387%	0.052%	0.015%	0.007%	0.005%	0.003%
25	-	0.388%	0.055%	0.015%	0.008%	0.005%	0.003%
27	-	0.389%	0.057%	0.015%	0.008%	0.005%	0.003%
29	-	0.392%	0.060%	0.016%	0.008%	0.005%	0.003%
31	-	0.395%	0.062%	0.016%	0.008%	0.005%	0.003%
33	-	0.397%	0.064%	0.016%	0.008%	0.005%	0.003%
35	-	0.398%	0.065%	0.017%	0.008%	0.005%	0.003%
37	-	0.398%	0.067%	0.017%	0.008%	0.005%	0.003%
39	-	0.399%	0.068%	0.018%	0.008%	0.005%	0.003%
41	-	0.399%	0.069%	0.018%	0.008%	0.005%	0.003%

Table 5.3: C_D convergence with *num_y* for *num_x* = 7.

num_y	ΔC_D(%)
5	-
7	0.673%
9	0.331%
11	0.180%
13	0.109%
15	0.070%
17	0.046%
19	0.032%
21	0.023%
23	0.018%
25	0.014%
27	0.011%
29	0.008%
31	0.007%
33	0.006%
35	0.005%
37	0.004%
39	0.003%
41	0.003%

5.1.3 Aircraft Configuration

The initial aircraft configuration is based on the mini-UAV Tekever AR4 [64]. It is a fixed wing UAV used primarily in surveillance and target acquisition missions. Due to limited available technical data on this aircraft, many parameters were deduced from the available ones. The information on the AR4 serves only as a starting point and assuming some of the values does not impact negatively the simulation, as the goal of this work is not to improve a specific aircraft.

Figure 5.2: Tekever AR4 UAV [64].

The wing span is 2.1 m and the length of the UAV is 1.3 m. From these measurements, the remaining quantities were deduced. The wing chord is considered to be 0.21 m at the root and 0.18 m at the tip, yielding an aspect ratio a little over 10. The dihedral and sweep angles were assumed to be 4° and 1.5°. Parameters that have a distribution along the span, such as chord, twist and spar thickness, are parameterized with B-splines, to avoid

discontinuities and decrease the number of design variables. While twist and thickness have three control points (cp), the chord has two, making its control points values correspond to the chord at the tip and root, thus keeping the lifting surfaces trapezoidal. The complete list of design parameters and their bounds is presented in Tab. 5.4.

Table 5.4: UAV design parameters and bounds.

Parameter	Wing		Tail		
	Initial Value	Bounds	Initial Value	Bounds	Units
Span, b	2.1	1.2, 2.6	0.42	0.25, 0.6	m
Chord cp, c	0.18, 0.21	0.1, 0.3	0.14, 0.17	0.08, 0.21	m
Twist cp, θ	1.5, 1.5, 1.5	-20, 20	0	0	degrees
Dihedral, Γ	4	-20, 20	0	-20, 20	degrees
Sweep, Λ	1.5	-20, 20	0.5	-20, 20	degrees
Thickness cp, t	0.003, 0.003, 0.003	0.0015, 0.05	0.0025, 0.0025, 0.0025	0.0015, 0.04	m

Additionally, the control and trimming variables are discretized into vectors of size 10, the same number of operating points. All entries are initialized with the same value, as listed in Tab. 5.5.

Table 5.5: UAV trimming and control parameters.

Parameter	Initial Value	Bounds	Units
Throttle, δ_T	0.5	0, 1	
Angle of attack, α	4	-15, 15	degrees
Stabilator angle, δ_t	-0.2	-15, 15	degrees

It is assumed that the tail lies on the same xy plane as the wing, with a distance between trailing edges of 1.05 m. The center of mass of the aircraft without lifting surfaces, designated empty CG, is assumed to be placed 0.2 m before the wing's trailing edge. Based on the information that the maximum take-off weight of the AR4 is 4 kg, the empty mass is considered to be 1.2 kg and the battery mass, 1.5 kg. The specific energy of the battery is 210 Wh/Kg and the loss factor is 50%.

NACA 2410 and NACA 0010 airfoils were considered for the wing and tail, respectively. Lift coefficient at $\alpha = 0$, C_{L_0} and zero-lift drag coefficient, C_{D_0} were obtained from tables produced using Xfoil [65], for a Reynolds number of 200000, which was estimated for sea level conditions, an air speed of 15 m/s and an average chord of 0.2 m

$$Re = \frac{\rho V_\infty c}{\mu} = \frac{1.225 \cdot 15 \cdot 0.2}{1.81206 \times 10^{-5}} = 202807.85 \approx 200000. \tag{5.1}$$

These and the propulsive parameters discussed in section 2.5 are shown in table 5.6.

The material composition of the AR4 is unknown. It is likely that some composite materials are used but since there is no information, it is simplified and considered that the spar tube is made of aluminum 6061, being the mechanical properties listed in Tab. 5.7. A safety factor of 2 is applied to the yield strength. The spar is placed at 30% of the chord in both surfaces, which is the point where thickness over chord ratio (h/c) of the airfoil is greatest.

This configuration serves as the fixed design for TP and as starting point for DP and DTP.

Table 5.6: UAV constant parameters

Parameter	Wing	Tail	Units
C_{L_0}	0.2434	0	-
C_{D_0}	0.00860	0.00852	-
max (h/c)	10	10	%
max (h/c) position	30	30	%
	Global		
Empty CG	-0.2, 0, 0		m
Empty mass, m_0	1.2		kg
Battery mass, m_{bat}	1.5		kg
Specific energy, e	210		Wh/kg
Loss factor, LF	50		%
Max motor power, P_m	180		W
Propeller radius, r	0.15		m
Induced-power loss factor, κ	1.2		-

Table 5.7: Mechanical properties of aluminum 6061 [66].

Parameter	Value	Units
Young's modulus, E	69	GPa
Yield strength, σ_{yield}	276 / 2	MPa
Density, ρ	2700	kg/m^3
Poisson's ratio, ν	0.33	-

5.2 Optimal Design for Minimum Energy Climb

In this application, aircraft design and trajectory are optimized individually (DP and TP) to better quantify the benefits of the combined optimization (DTP). All three have the objective of minimizing the necessary energy to climb to 1000 m of altitude and initiate cruise with an air speed of 16 m/s.

The initial trajectory is presented in the next section, which is fixed for the design optimization (DP) and serves as starting point for the other two problems.

5.2.1 Baseline Trajectory

The trajectory was discretized into $N = 10$ intervals. The initial position is $(x, z) = (0, 0)$, assuming that the UAV is launched at ground level, and the initial speed is 14 m/s. From this point until the final altitude is reached, the speed is kept at 15 m/s and the total flight time is 570 seconds. The flight path angle varies as shown in figure 5.3, where a steady climb is made at a $\gamma = 8°$ angle after the third interval and thus defining every (x, z) coordinates and velocity components. Horizontal and vertical displacement variables are both bounded by $[0, 100000]$ m, so there is no backwards progression nor crashing into the ground. Velocity's horizontal component is bounded by $[0, 18]$ m/s and the vertical by $[-18, 18]$ m/s. These conditions result in a trajectory whose full discretization is presented in Tab. 5.8.

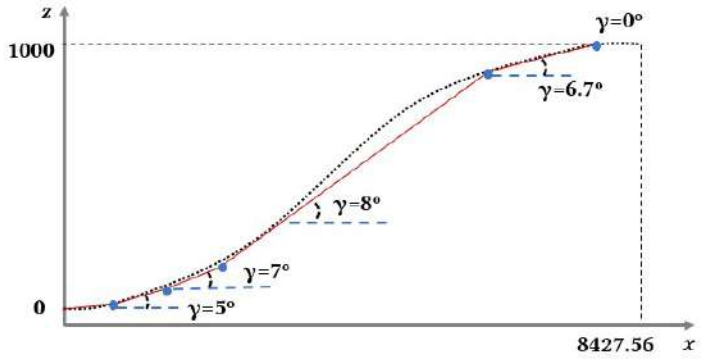


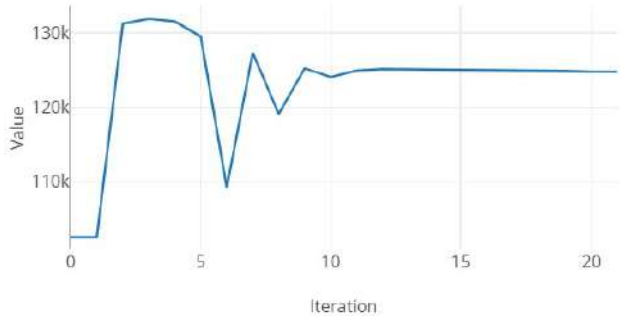
Figure 5.3: Profile of the initial climb trajectory.

Table 5.8: Discretization of trajectory variables.

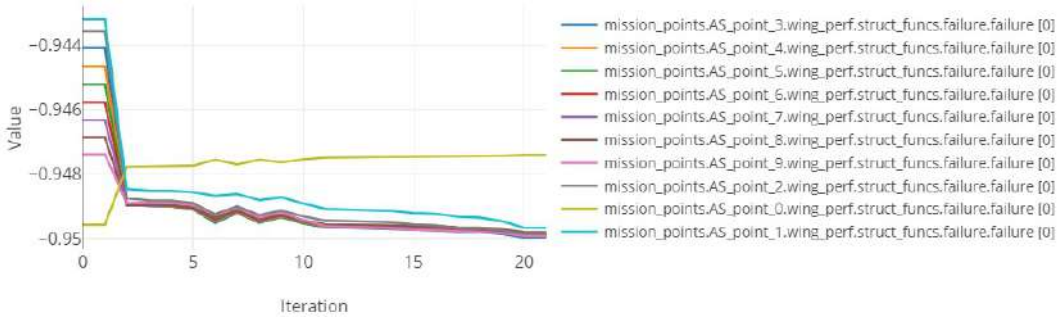
Parameter	Collocation points										
	1	2	3	4	5	6	7	8	9	10	11
γ [deg]	0.6	5	7	8	8	8	8	8	8	6.7	0
x [m]	0	797.97	1649.72	2498.34	3345.02	4191.70	5038.38	5885.06	6731.74	7578.42	8427.56
z [m]	0	8.96	83.48	187.68	306.67	425.67	544.66	663.65	782.65	901.64	1000
$\dot{x}$ [m/s]	13.99	14.94	14.89	14.85	14.85	14.85	14.85	14.85	14.85	14.90	16
$\dot{z}$ [m/s]	0.15	1.31	1.83	2.09	2.09	2.09	2.09	2.09	2.09	1.75	0
V_∞ [m/s]	14	15	15	15	15	15	15	15	15	15	16

5.2.2 Results

All constraints were satisfied within the tolerances established for the solver and optimizer, so it can be concluded that local minima were found and all problems were successfully optimized. Figures 5.4, 5.5 and 5.6 shows convergence plots of the objective function and of the failure constraint, for each of the three problems.

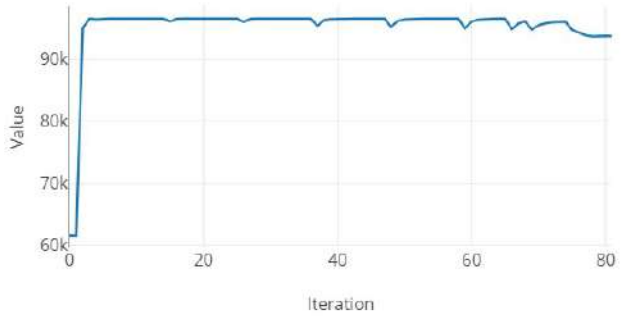


(a) TP - E_e

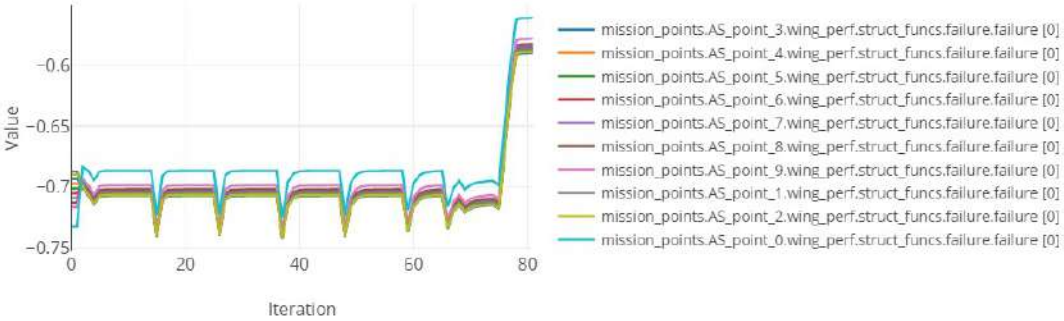


(b) TP - failure wing

Figure 5.4: Convergence plots of the objective function and failure constraint for TP.

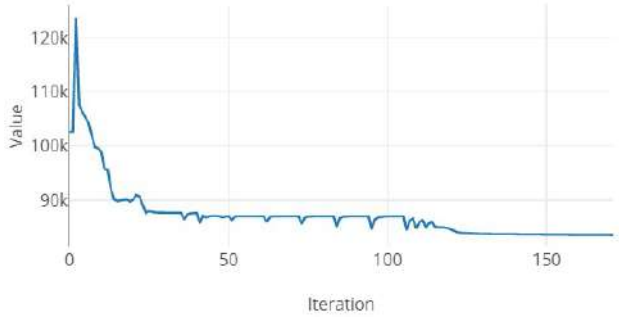


(a) DP - E_e

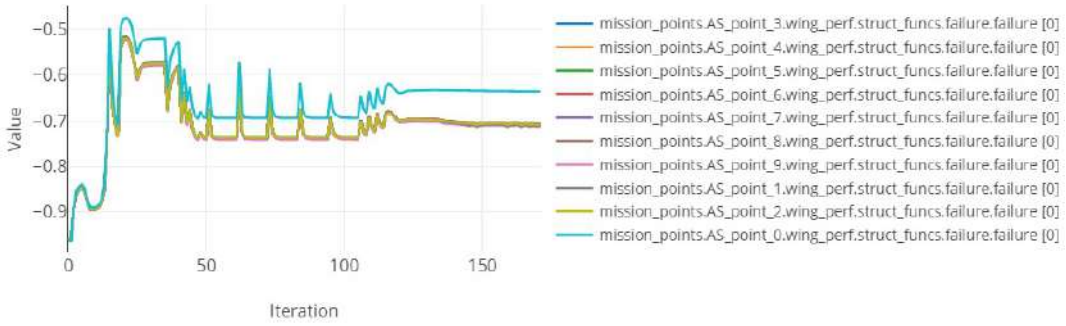


(b) DP - failure wing

Figure 5.5: Convergence plots of the objective function and failure constraint for DP.



(a) DTP - E_e



(b) DTP - failure wing

Figure 5.6: Convergence plots of the objective function and failure constraint for DTP.

The computational costs of the three problems are compared in Tab. 5.9 and, to discuss and partially explain their variation, problem sizes are presented in Tables 5.10 and 5.11.

Table 5.9: Optimization performance with an Intel® Core™ i7-5500 @ 2.4 GHz Processor.

Parameter	Trajectory	Design	Design + Trajectory
Iterations	17	20	74
Function evaluations	22	81	171
Computation time	2h23	2h41	11h28

Table 5.10: Number of design variables.

	Variables	Note	Quantity	Total
Trajectory	$x, \dot{x}, z, \dot{z}$	vectors of size N+1	44	45
	t_f		1	
Geometry	θ_w, t_w, t_t	3 control points	9	19
	c_w, c_t	2 control points	4	
	$b_w, b_t, \Lambda_w, \Lambda_t, \lambda_w, \lambda_t$		6	
Control	$\delta_T, \alpha, \delta_t$	vectors of size N	30	30
		Trajectory Problem (TP)		75
		Design Problem (DP)		49
		Design+Trajectory (DTP)		94

Table 5.11: Number of constraints.

	Constraints	Note	Quantity	Total
Collocation and equilibrium	$\zeta_x, \zeta_z, \zeta_{F_x}, \zeta_{F_z} = 0$	vectors of size N	40	50
	$C_{M_y} = 0$	vector of size N	10	
Aerodynamic	$C_{l_w} < Cl_{w_{max}}$	vector of size $\frac{num\_y-1}{2} \times N$	150	210
	$C_{l_t} < Cl_{t_{max}}$	vector of size $\frac{num\_y-1}{2} \times N$	60	
Structural	$Intersect_w < 0$	vector of size $\frac{num\_y-1}{2}$	15	41
	$Intersect_t < 0$	vector of size $\frac{num\_y-1}{2}$	6	
	$Failure_w, Failure_t < 0$	$N\times$ KS function	20	
Energy	$E_e \leq E_{bat}$		1	1
		Trajectory Problem (TP)		281
		Design Problem (DP)		282
		Design+Trajectory (DTP)		302

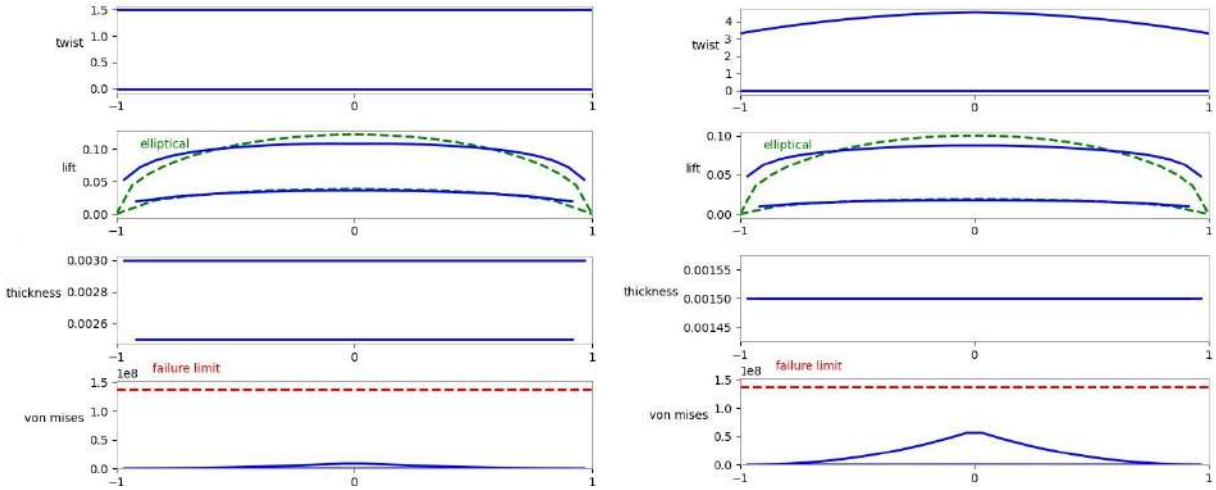
Tables 5.10 and 5.11 list the number of variables and constraints by groups. DTP uses all 94 design variables and 302 constraints. DP has prescribed trajectory, so trajectory variables and collocation constraints pertaining to displacement ζ_x, ζ_z are not used, which gives 49 design variables and 282 constraints. Lastly, TP does not include any geometric variables nor the intersection constraints, which amounts to 75 design variables and 281 constraints. The Hessian sizes are given by the square of the sum of design variables and constraints. They are 126736, 109561 and 156816, for TP, DP and DTP, respectively.

DTP is a considerably larger problem, so it is natural that its optimization took longer. Its Hessian is 43% larger than DP's, and 24% larger than TP's. However, problem size alone does not explain the differences in computational time, as DP is a smaller problem than TP, yet it took more function evaluations and time. One possible explanation might be the handicap that a prescribed trajectory represents, and the difficulty of adapting a design to it. Another possible explanation might lie in fact that DP has less variables for almost the same number of constraints than TP. This results in a more rigid problem, as the optimizer has fewer variables to try to satisfy approximately the same number of constraints, which makes it harder to find a solution. Moreover, DP and DTP deal with optimization of the aerostructural component, which becomes increasingly harder to converge as coupling strength increases. As a consequence, it takes more iterations, which we can see are strongly correlated with computational time.

The final values of geometric design variables and a selection of performance metrics are listed in Tab. 5.12 and the distributions of twist, lift, spar thickness and stresses along the span are depicted in Figs. 5.7 and 5.8. The control and trajectory variables are shown in Figures 5.9 and 5.10, respectively.

Table 5.12: Final values of geometric design variables for TP, DP and DTP problems.

Parameter	Trajectory	Design	Design + Trajectory	Units
Electric energy, E_e	124.76	93.79	83.62	kJ
Flight time, t_f	377.4	570	253.3	s
Total mass, m	3.91	2.99	2.933	kg
		Wing		
Span, b_w	2.1	2.03	1.59	m
Chord cp, c_w	0.18, 0.21	0.1, 0.1	0.1, 0.1	m
Twist cp, θ_w	1.5, 1.5, 1.5	3.31 4.42 4.51	1.99, 4.02, 3.79	degrees
Dihedral, Γ_w	4	4.02	3.48	degrees
Sweep, Λ_w	1.5	1.06	0.83	degrees
Thickness cp, t_w	0.003, 0.003, 0.003	0.0015, 0.0015, 0.0015	0.0015, 0.0015, 0.0015	m
		Tail		
Span, b_t	0.42	0.26	0.25	m
Chord cp, c_t	0.14, 0.17	0.08, 0.08	0.08, 0.08	m
Dihedral, Γ_t	0	0	0	degrees
Sweep, Λ_t	0.5	0	0	degrees
Thickness cp, t_t	0.0025, 0.0025, 0.0025	0.0015, 0.0015, 0.0015	0.0015, 0.0015, 0.0015	m



(a) TP (b) DP

Figure 5.7: Twist, lift, thickness and stresses distribution along span for TP and DP.

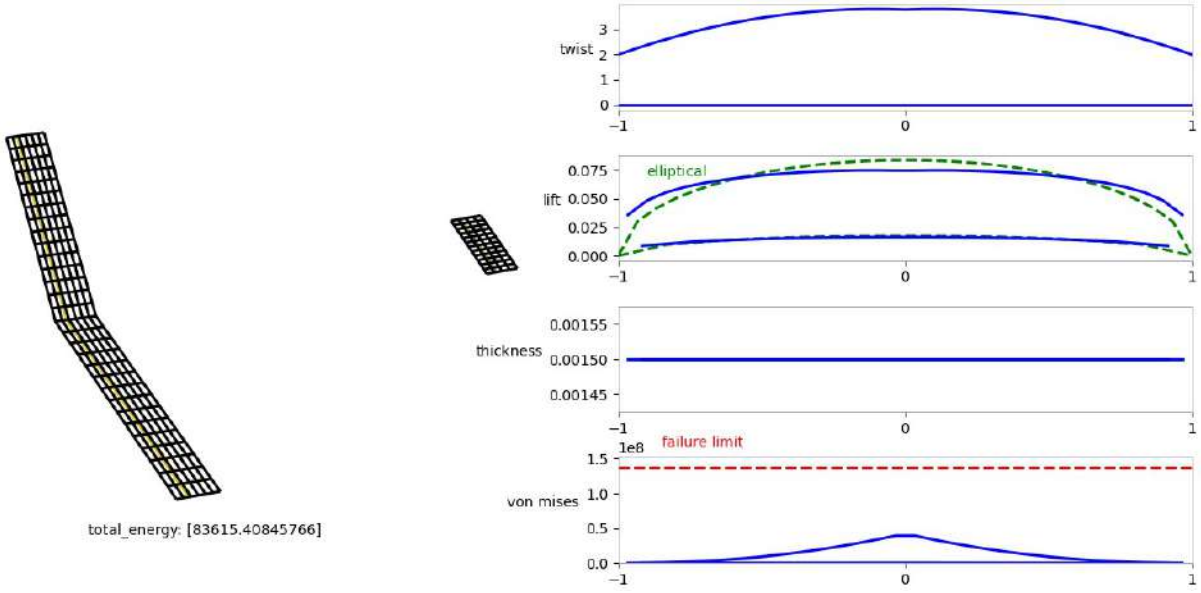


Figure 5.8: Twist, lift, thickness and stresses distribution along span for DTP, with representation of the final configuration.

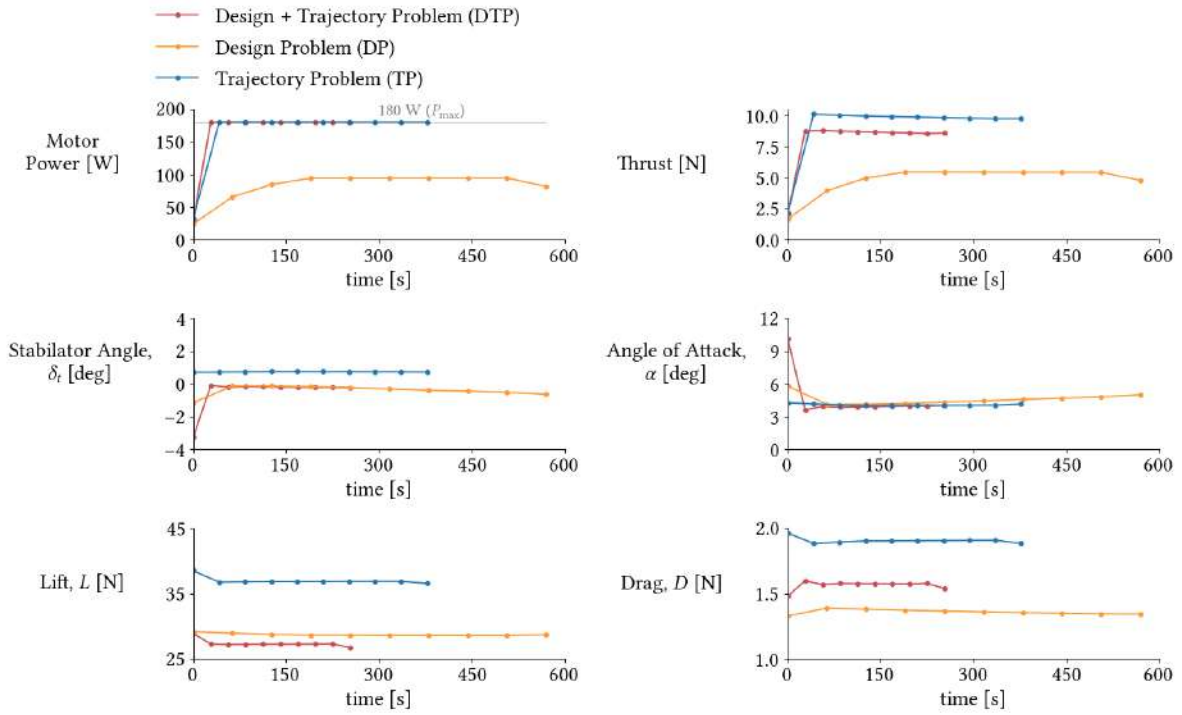


Figure 5.9: Results of control and trimming variables for climb energy optimization.

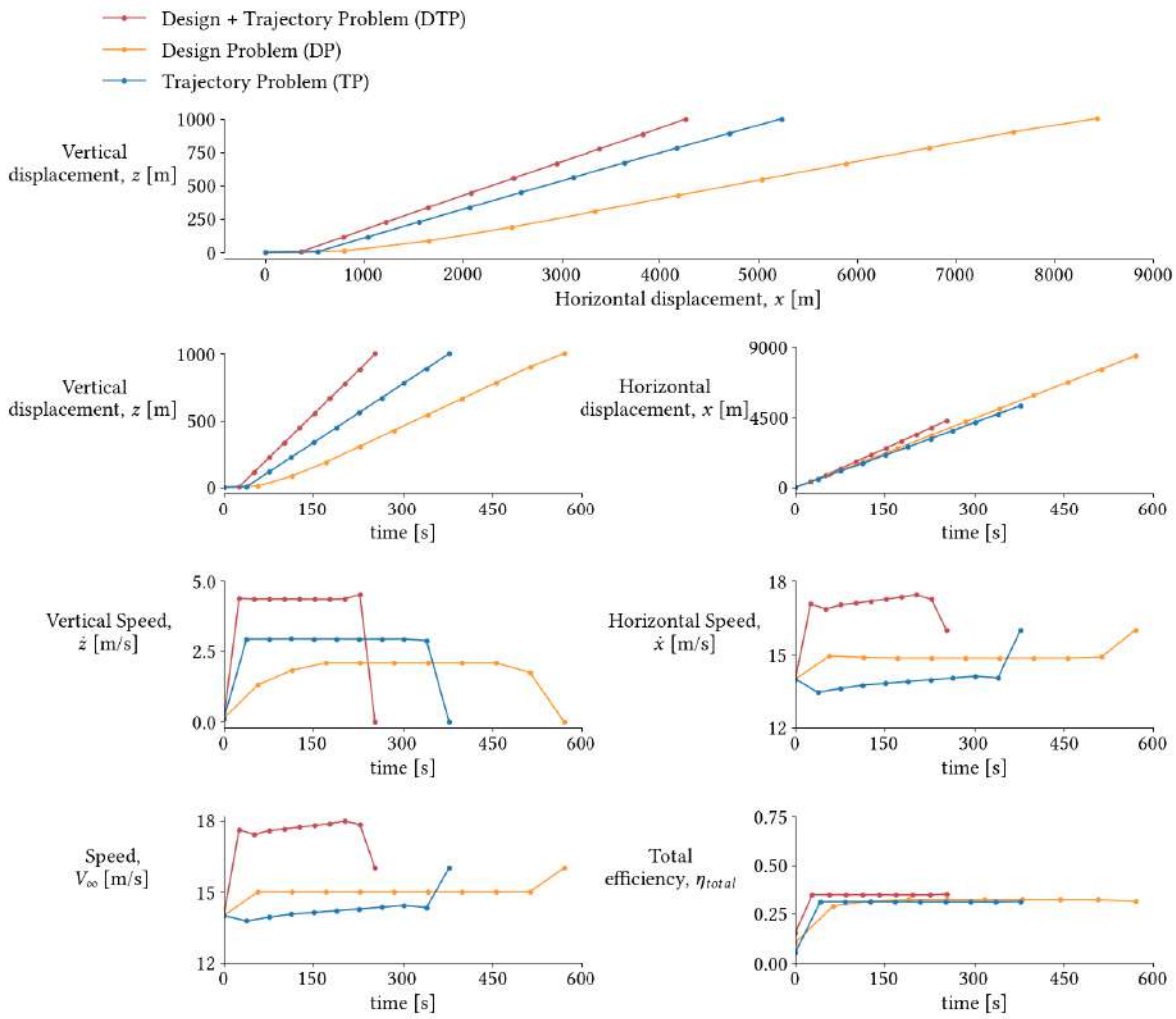


Figure 5.10: Results of trajectory variables for climb energy optimization.

As seen in Tab. 5.12, DTP had the least amount of energy spent E_e, with a reduction of 33% relative to TP, and 10.8% relative to DP. TP and DP are limited by the prescribed trajectory and configuration, respectively, whereas in DTP, the optimizer has access to both design and trajectory variables and therefore, produces the best combined outcome of both. This result is a clear demonstration of how combining trajectory and design in one optimization leads to great improvements.

Furthermore, we observe that less energy was spent in DP than in TP, which indicates that the initial configuration is more of a handicap to the objective of minimization than the prescribed trajectory. Let us now look at the differences in design and trajectory results that led to this variation in energy spent.

We see in Tab. 5.12 that there was a considerable mass reduction in DP and DTP, relative to the fixed configuration (≈ 1 kg, which represents approximately 25 % less mass). This was achieved by decreasing the thickness of the spars and the size of lifting surfaces. The spar wall thickness, in particular, was set to the lower bound on both surfaces, 0.0015 m. This was done in an effort to decrease material as much as possible, provided that stresses had a good margin to the failure limit, and so, meeting the failure constraint was not an issue, as can be verified in Figs. 5.7 and 5.8 in the plot labeled "Von Mises".

The position of the spar is coincidental with the fraction of chord of maximum (h/c), which is 10% for both

surfaces, as per Tab. 5.6. The diameter of the spar is thus also coincidental with the maximum height of the wing's section, which means that in the same way decreasing the span leads to a reduction of the length of the spar, decreasing the chord reduces its radius. Also, obtaining the necessary area by increasing the span instead of the chord leads to a wing with higher aspect ratio, which yields less induced drag for a given value of lift [67].

The span of the tail was set to the lower bound in both problems. Moreover, the stabilator, which has a symmetric airfoil, had its angles set to values close to zero, as shown in the stabilator angle plot in Fig. 5.9. These results indicate that the center of mass was brought closer to the aerodynamic center of the wing, reducing its moment, and the distance of the stabilator to the center of gravity (the arm of the tail moment) might be oversized and is the principal responsible for creating the balancing moment necessary to trim the aircraft.

The reduction of span in DTP was a 20 % greater than in DP. This can be explained by looking at the thrust and lift plots shown in Fig. 5.9. Because trajectory is prescribed in DP, the throttle control was chosen just to equilibrate the forces, rather than to propel the aircraft forward or increase the angle of climb. Because horizontal velocity is kept almost constant, the horizontal component of forces has to be null, and since the flight path angle is not too high, a lower thrust force was required. This means that on the vertical component, weight had to be balanced out mostly through lift. The higher lift (when compared to DTP) was achieved through a slightly higher span and twist angle, which demonstrates how a fixed trajectory penalizes the aircraft design.

The twist angles were also chosen so that lift distribution along the span would match, as closely as possible, that which corresponds to an elliptical distribution of circulation, as it minimizes C_D for a given C_L [67]. This is seen in Figs. 5.7 and 5.8, where it can also be observed that TP's lift distribution is similar to the other two problems, despite having a fixed constant twist distribution. This is a result of having tapered wings, which is another way of achieving elliptical distribution [67].

Dihedral and sweep angles were changed only slightly. Dihedral's major contribution is to lateral stability, which is not accounted in this model. Likewise, sweep's main contribution is delaying divergence mach number, which is not an issue since the operational mach number is below 0.1 in all problems. The only other factor that these angles have influence in is the length of the spar. However, the angles are minimal and so this effect is of little relevance as well.

Energy spent depends directly on electric power and time, as per Eq. (2.62). Therefore, in order to minimize it, power or time have to be reduced. As we can see in Fig. 5.10, the trend was to decrease flight time as much as possible, which was achieved by shortening horizontal distance and flying faster.

To shorten the horizontal distance, it is necessary to fly with higher flight path angles, provided that vertical distance is fixed. Bearing in mind that lift's vertical component gets smaller for higher flight path angles, the burden of counteracting weight falls on thrust. To that end, throttle control was pushed to the maximum in DTP and TP. It seems counter intuitive to have power set to its limit, but the energy saved by making flights shorter surpasses the added expense of having higher power, thus yielding a positive balance and minimizing energy consumed.

The average air speed was decreased from the 15 m/s of the initial trajectory to 14.4 m/s, in TP. This can be seen as way to compensate for the large wingspan and not generate more lift than the necessary to trim the UAV. Since DTP also optimized the configuration, the wingspan was reduced, thus allowing the speed to be increased to 17.5 m/s, on average. Although flying at higher speeds does decrease the time to cover the same distance, it

also lowers thrust forces, as they are approximately inversely proportional. This limits the angle of climb and as a consequence extends horizontal distance. Therefore, the chosen speeds are an optimal balance between travelling fast and travelling less.

The trajectory in DTP had major improvements relative to that of TP, as the latter was heavily penalized by the fixed configuration. TP's larger wingspan generated a much higher lift and consequently, a higher drag and pitching moment. To comply with the zero moments constraint, the stabilator control angle was set to positive values, since the larger structural mass pulled the center of mass more in the direction of the trailing edge.

The flight path angle could not be increased as much in TP, because the aircraft design was fixed and therefore, so was weight. This greater weight was balanced mostly through lift, which was naturally much higher than in the other problems, given the larger wingspan. As described before, for greater climb angles, the vertical component of lift decreases, but, in this case, it also becomes quite significant that the horizontal component grows. Inversely, the vertical component of thrust increases and the horizontal decreases, resulting in a two way burden for thrust. The maximum power was the same for all problems and in TP it was clearly insufficient to trump weight and the horizontal component of lift at high γ. Conversely, design was also optimized in DTP, so a better trajectory was achieved through the configuration modifications previously discussed, which ultimately led to less energy spent.

This shows how the fixed design negatively impacted the trajectory optimization potential, or the other way around, how having both design and trajectory be part of the optimization led to better results.

The efficiency results are shown in Fig. 5.10. Efficiency is capped at 50% because that is the loss factor considered between battery and motor. The remaining losses can be attributed to drag resistance and to the propulsive system, which performed with an efficiency of between 80% and 85%. Using Eq. (2.67) for DTP, for example,

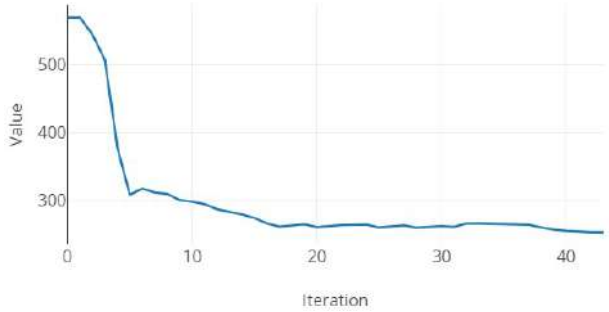
$$\eta_{prop} = \frac{TV_\infty}{P_m} \approx \frac{8.7 \times 17.5}{180} \approx 85.0\%. \tag{5.2}$$

Efficiency plots basically show how effectively electrical energy was converted into gravitational potential energy, given that variations in velocity were much smaller than those of vertical displacement. DTP had higher efficiency than the other problems, which is another evidence of how advantageous it is to have all disciplines being optimized simultaneously.

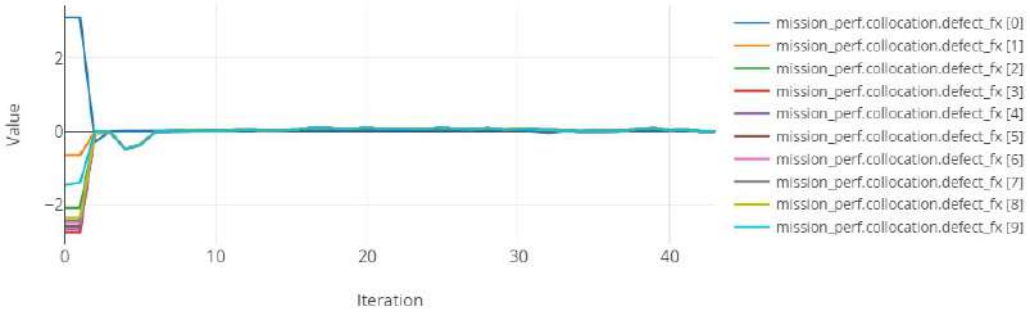
Results from Fig. 5.10 highlight some limitations of the discretization method employed. It is a forward Euler scheme, so the ith state and control values are used in the interval delimited by points i and $i + 1$. Initial and final states are pre defined, so the first interval uses the boundary condition value. This is not a problem in itself, but since the trajectory is parameterized by 10 intervals only, each segment lasts 10 % of the entire flight time, which is still a considerable amount of time to fly at non optimal conditions. This could be mitigated through a more refined discretization, however, not only would it increase the number of variables, it would also incur in more mission points and fluid structure calculations, which drives up the computational cost. A more efficient alternative would be using B-Splines to parameterize state and control. This would result in continuous and smooth function at the cost of few control points and therefore, fewer mission points as well.

5.3 Optimal Design for Minimum Time Climb

To further explore the design space, the DTP climb optimization was performed with the objective of minimizing the flight time. The baseline configuration and trajectory are the same as in Sec. 5.2. Convergence plots of the objective and one collocation constraint are shown in Fig. 5.11.



(a) Flight time, t_f



(b) Collocation constraint ζ_{F_x}

Figure 5.11: Convergence plots of the objective and collocation constraint for time minimization.

The final values of geometric variables and performance metrics are presented in Tab. 5.13 with results previously obtained for energy optimization as reference. Figures 5.12 and 5.13 show results of control and trajectory related variables, respectively, also with minimum energy results for comparison.

The optimal solution found in time minimization is very close to the one obtained for energy minimization. From Tab. 5.13, we see that the difference in flight times was 0.26 s and the energy difference was 160 J, which constitutes variations of approximately 0.1% and 0.2%. The minimal variations might have been a result of the not so tight tolerance, and the fact that the solution was so similar shows how strongly linked energy and flight time are.

The final aircraft design was almost the same, with chord and spar wall thickness values at minimums for both surfaces, once again. Results varied more in the trajectory discipline, with a different balance between distance and speed being achieved in this problem. As seen in Fig. 5.13, freestream velocity was considerably higher in time minimization, mainly due to an increase in the horizontal component. With the same vertical velocity component but a higher horizontal one, the flight path angle decreased, relative to the energy minimization, thus extending horizontal distance in approximately 120 m.

As already described, higher speed yields lower thrust for the same power. That is what happened in the time minimization problem, where in spite of having power at the maximum, thrust was slightly lower compared to the energy minimization result (Fig. 5.12).

Table 5.13: Comparison of final geometric design variable values for DTP with minimum time and minimum energy as optimization objectives.

Parameter	Min time	Min energy	Units
Electric energy, E_e	83.78	83.62	kJ
Flight time, t_f	253.56	253.3	s
Total mass, m	2.93	2.933	kg
	Wing		
Span, b_w	1.56	1.59	m
Chord cp, c_w	0.1, 0.1	0.1, 0.1	m
Twist cp, θ_w	2.61, 3.23, 3.27	1.99, 4.02, 3.79	degrees
Dihedral, Γ_w	3.77	3.48	degrees
Sweep, Λ_w	0.69	0.83	degrees
Thickness cp, t_w	0.0015, 0.0015, 0.0015	0.0015, 0.0015, 0.0015	m
	Tail		
Span, b_t	0.26	0.25	m
Chord cp, c_t	0.08, 0.08	0.08, 0.08	m
Dihedral, Γ_t	0	0	degrees
Sweep, Λ_t	0	0	degrees
Thickness cp, t_t	0.0015, 0.0015, 0.0015	0.0015, 0.0015, 0.0015	m

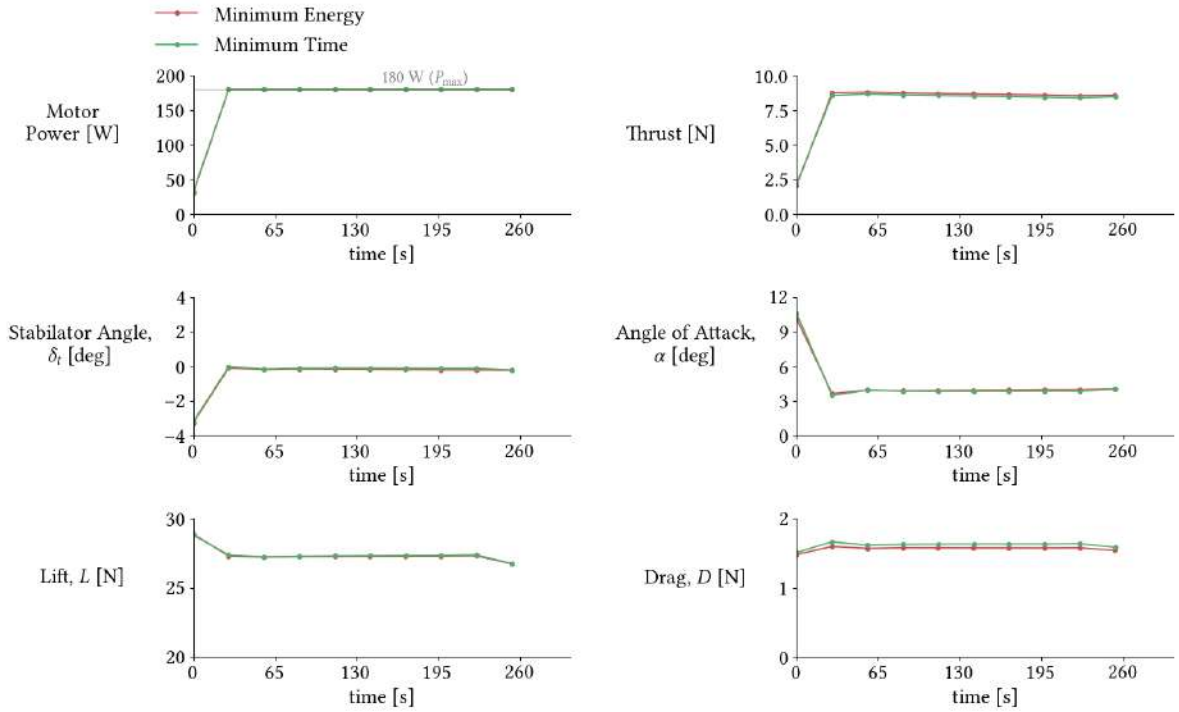


Figure 5.12: Results of control variables for minimum time to climb to 1000m.

Lift absolute value was almost the same in both problems, given the similarity in weight, but it was achieved differently. Since lift increases with speed and angle of attack up to stall condition, twist tended to lower values in the time minimization problem, yielding a lower effective angle of attack as a way to compensate for the

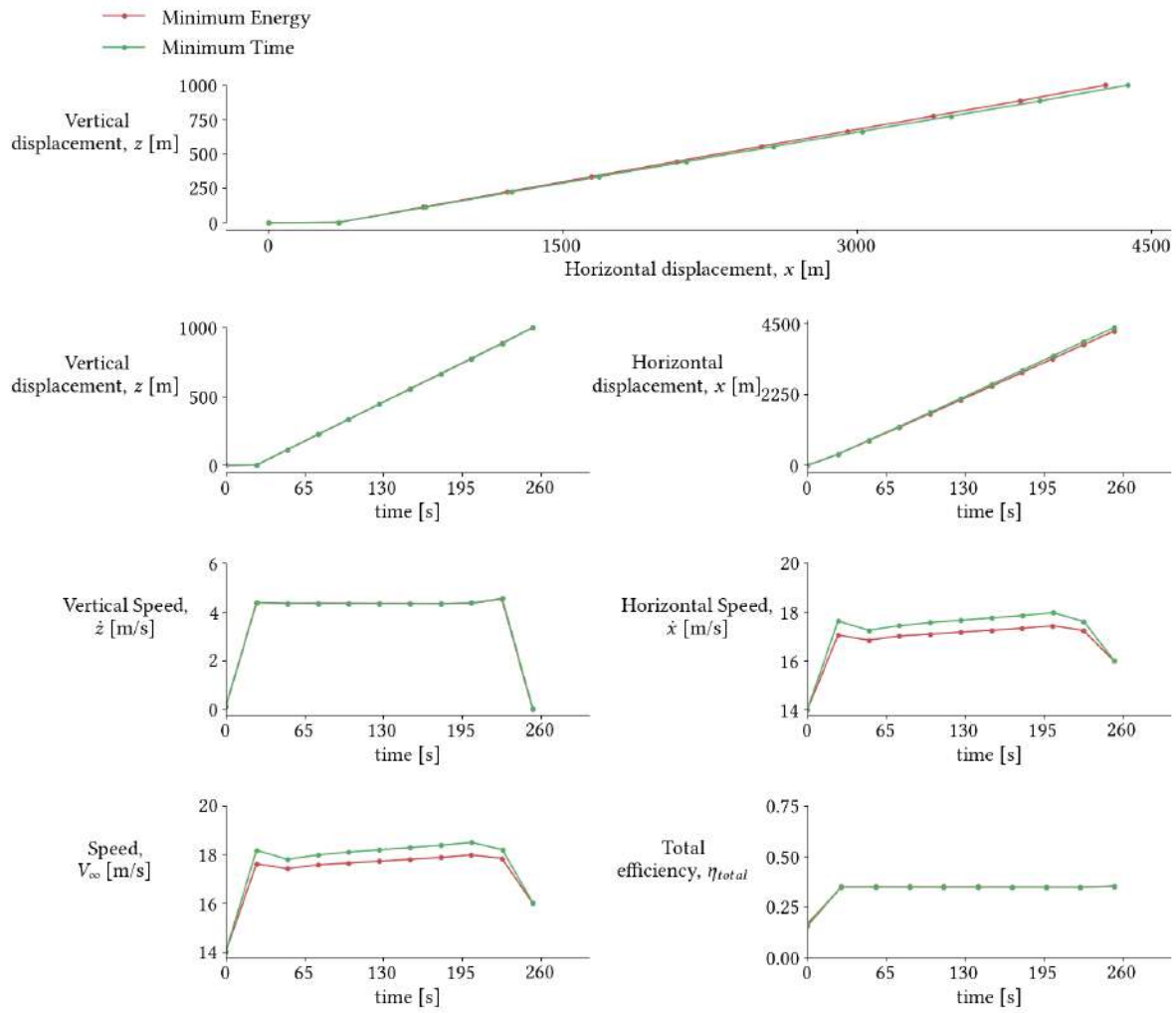


Figure 5.13: Results of trajectory variables for minimum time to climb to 1000m.

higher speed. Span was also reduced in 3 cm, relatively to the energy minimization problem. This change led to a marginal mass reduction, which indicates that the span decrease was most likely another form of compensating for the higher speed. Since lift was almost the same, the stabilator angles necessary to trim the aircraft were also similar, as seen in Fig. 5.12.

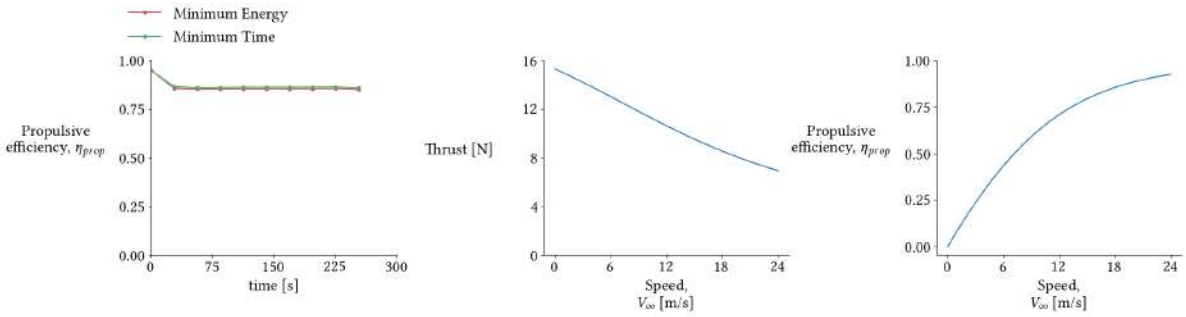


Figure 5.14: Comparison of propulsive efficiencies, thrust and propulsive efficiency relationship with speed.

It is worth noting that the efficiencies are practically the same, but the propulsive efficiency, defined in Eq.

(2.67) and shown in Fig. 5.14, was slightly higher for time minimization. Because the approximate slope of the thrust-speed curve is between -1 and 0, increasing speed in 1 m/s reduces thrust in less than 1 N. As a result, efficiency increases with speed, which explains why it was higher in the time minimization problem.

5.4 Optimal Design for Maximum Range

With the same intent of Sec. 5.3, the optimization was performed for the maximization of range, that is, horizontal distance. The starting point was a cruise flight stage. As such, the trajectory was initialized with a linearly spaced vector for horizontal displacement, starting at 0 and ending at 200 km, and with a constant altitude of 850 m. The horizontal bounds were 700 km, and the vertical's were 0 and 1500 m. The horizontal speeds, bounded by [0, 18] m/s, were initiated at 12 m/s, where the initial and final entries were fixed to the same value. The initial vertical speeds were set to zero and bounded by [-18, 18] m/s. The baseline configuration was the same as the one described in Sec. 5.1.3.

The optimal solution found is presented in Tab. 5.14 and Figures 5.15, 5.16 and 5.17.

Table 5.14: Final geometric design variable values for DTP with maximum distance as optimization objective.

Parameter		Units
Horizontal distance, x	504.5	km
Electric energy, E_e	1134.0	kJ
Flight time, t_f	11.95	h
Total mass, m	3.08	kg
	Wing	
Span, b_w	2.6	m
Chord cp, c_w	0.1, 0.1	m
Twist cp, θ_w	2.11, 10.46, 7.59	degrees
Dihedral, Γ_w	2.53	degrees
Sweep, Λ_w	1.45	degrees
Thickness cp, t_w	0.0015, 0.0015, 0.0015	m
	Tail	
Span, b_t	0.25	m
Chord cp, c_t	0.08, 0.08	m
Dihedral, Γ_t	0	degrees
Sweep, Λ_t	0	degrees
Thickness cp, t_t	0.0015, 0.0015, 0.0015	m

We can see in Tab. 5.14 that the final distance was 504.5 km. A different philosophy in design and trajectory was adopted to achieve this result, compared to the previous problems. It is natural considering that this objective of optimization is diametrically opposite to the previous cases presented, in that the distance and flight duration are being extended to the maximum, whereas before they were being reduced as much as possible. Therefore, different solutions were obtained to satisfy the mission's objective.

To extend horizontal distance, the energy resource was naturally used to travel horizontally rather than vertically, being more efficient. This meant flying at very small flight path angles, as seen in the trajectory plot

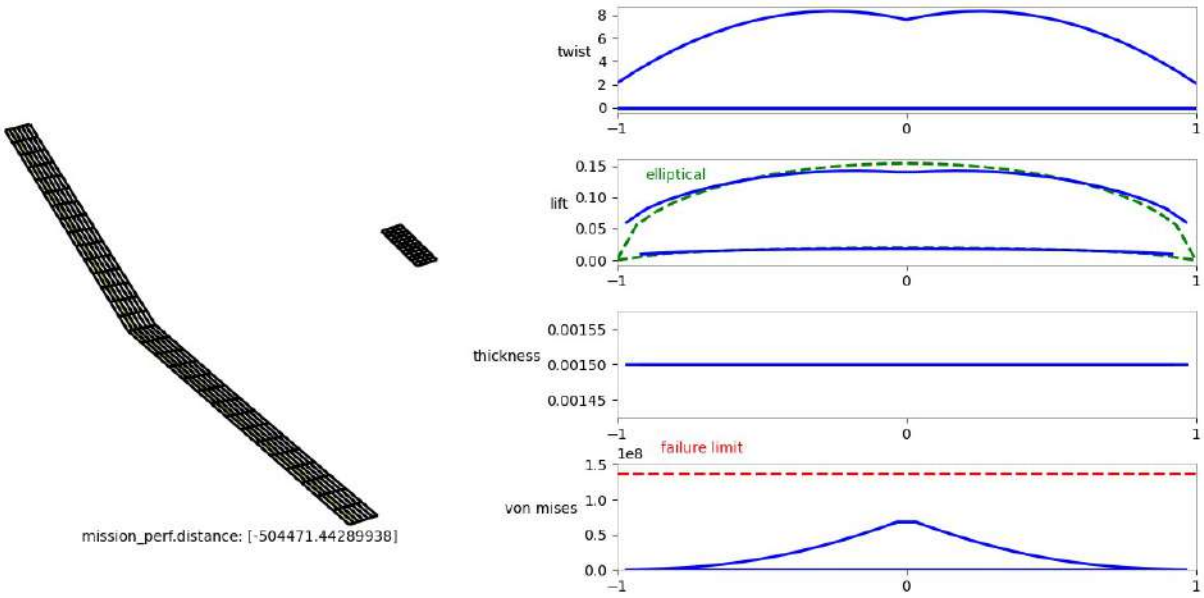


Figure 5.15: Twist, lift, thickness and stresses distribution along span for maximum distance optimization.

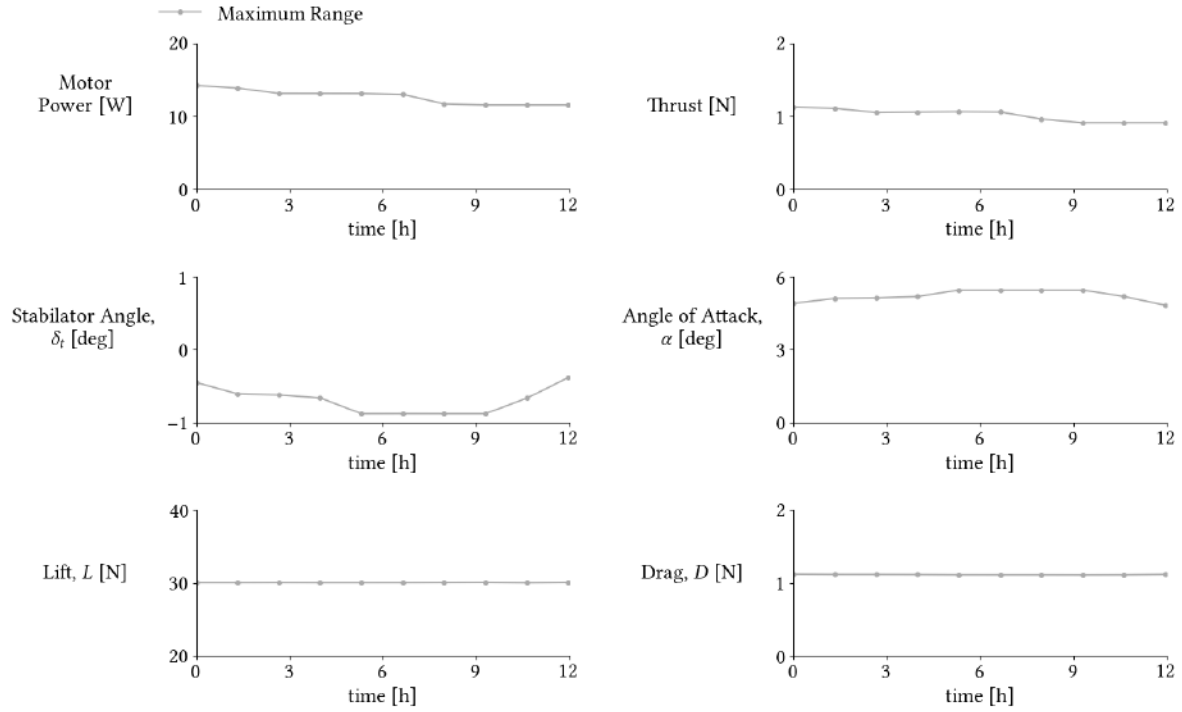


Figure 5.16: Results of control variables for maximum distance optimization.

in Fig. 5.17, being −0.3° the largest angle of descent. As explained before, for low γ, lift is almost vertical and the vertical component of thrust is very small, so conditions were very close to cruise, where $L \approx W$ and $T \approx D$.

Given this flight condition, the final design aimed at generating a surface area capable of producing high lift and a wing with high aspect ratio. We can see that the major change was in wingspan, being set to the upper bound, but chord and spar thickness, on the other hand, were set to the lower bounds, like in the other optimizations. The minimum chord and spar thickness result in minimal structural weight, which requires less

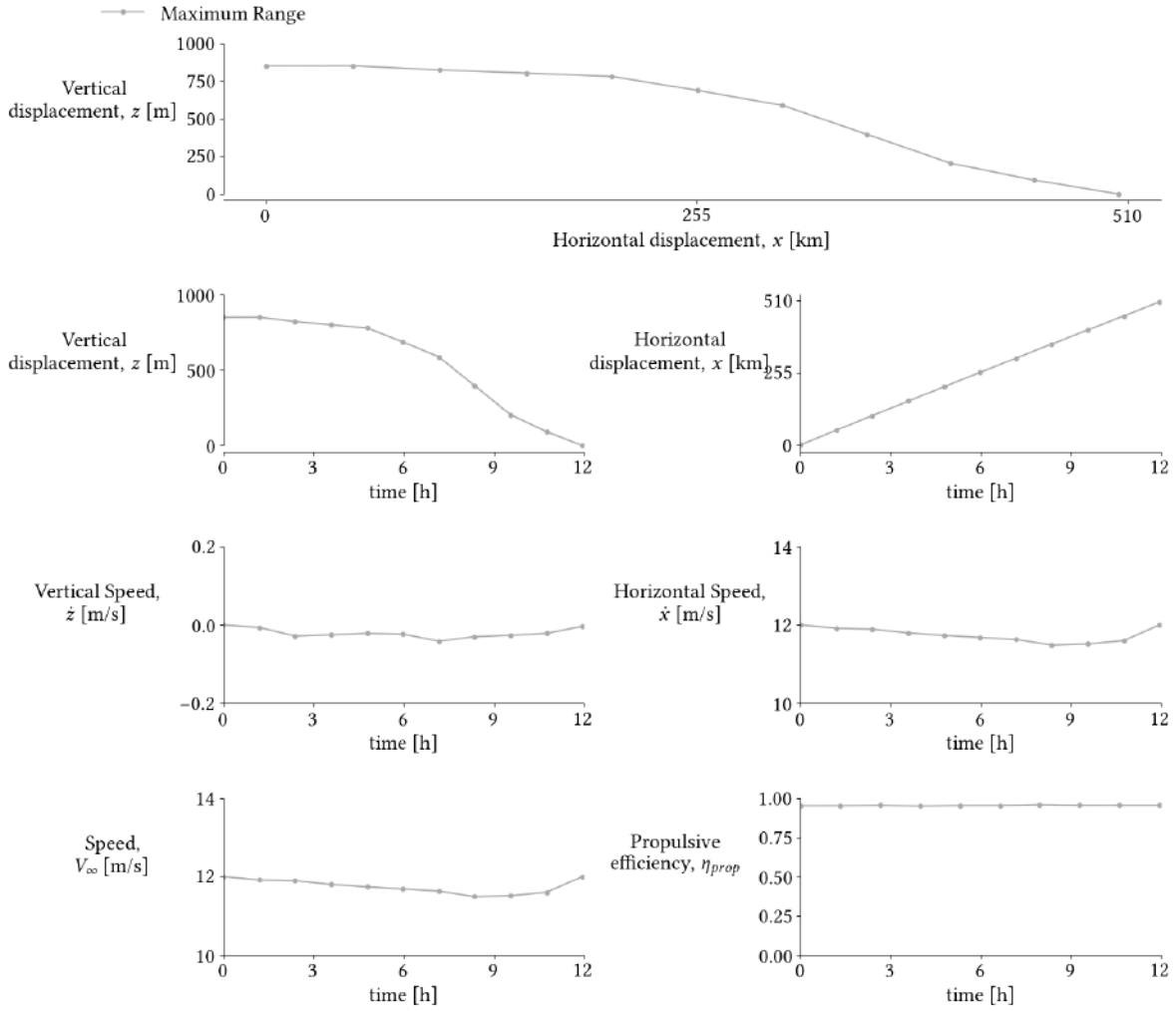


Figure 5.17: Results of trajectory variables for maximum distance optimization.

lift. This span and chord combination result in the largest aspect ratio possible, which yields lower induced drag, thus maximizing lift to drag ratio [67]. In spite of the large aspect ratio, stresses had good margin to the failure limit, as seen in Fig. 5.15. This is evidence that the material chosen far exceeds the demands of this mission.

We also see in Tab. 5.14 that twist control points tended to larger values, yielding a real twist distribution with angles as high as 8°, as observed in Fig. 5.15. Once again, the distribution was made so that lift would match that which corresponds to an elliptical distribution of circulation.

As for the tail, its size was minimized once again, in spite of the larger wing. The stabilator angle had minor changes throughout the flight, varying from −0.45° to −0.87°. The stabilator angle is relatively small, indicating that the distance between tail and center of gravity is doing most of the work in balancing the pitching moments.

As said before, most of the energy was used in horizontal travel rather than vertical. The intent was to extend near horizontal flight as much as possible, but, because energy is finite, inevitably it would have to start descending. We see that the first half of the flight was almost horizontal, with flight path angles lower than −0.1°. During this phase, speed had a slight reduction from 12 m/s to 11.5 m/s, which was compensated with a small increase in the angle of attack from 4.9° to 5.4°, thus maintaining the value of lift. With most of the lift generation

ensured by the large wingspan and angle of attack, the throttle control was chosen so that the output thrust would help achieve equilibrium of forces for as long as possible and to spend energy at a slow rate. We see in Fig. 5.16 that power was almost constant during the first half of the flight and then dropped, from approximately 14.4 W to 11.5 W, which led to the decrease in thrust too. This drop coincides with the beginning of the steeper descent phase, as the output thrust was no longer enough to maintain equilibrium of forces. In this descent phase, the optimizer chose not to descend at zero power, as drag resistance would very rapidly decelerate the aircraft. Instead, some energy was used to help trim the UAV and extend descent time.

This made it possible to spend energy at such a slow rate that flight lasted almost 12 hours. Nevertheless, it can be observed in Tab. 5.14 that this mission required fully depleting the battery energy,

$$E_{bat} = m_{bat} e = 1.5 \times 210 \times 3600 = 1134 \text{ kJ}, \tag{5.3}$$

where it is multiplied by 3600 seconds because $e = 210$ is in Wh/kg units. This is an expected result, as long there is energy to propel the aircraft, the flight can be maintained. Bearing this in mind, battery's capacity is one of the limiting factors of this problem.

Finally, we can see in Fig. 5.17 that propulsive efficiency was close to 95 %. This was a result of the low power and high angle of attack, as for a fixed velocity, lower power increases efficiency, and, likewise, higher angles of attack reduce perpendicular velocity, thus increasing thrust and efficiency.

5.5 Summary

The framework was tested through the coupled design and trajectory optimization of different objectives. We were able to see that the optimizer produced different solutions according to the objective. In energy minimization, the trend was to shorten distance and fly faster, in the attempt of reducing flight time. To do so, the optimizer chose to apply maximum power and fly at high flight path angles. Energy and flight time parameters were strongly correlated, with both minimizations yielding very similar results. In range maximization, the optimizer chose a configuration that maximized aspect ratio, while keeping mass low. Power was distributed throughout the flight to better trim the aircraft for a longer period, thus extending range.

We could verify that optimizing design and trajectory simultaneously constitutes a great improvement in performance relative to the isolated problems. The improvements obtained by the coupled problem came at a higher computational cost, given that all variables and constraints of both components are used by the optimizer.

Chapter 6

Conclusions

6.1 Achievements

In this work, the OpenAeroStruct framework was upgraded. Modifications were made to accommodate an electric propulsive system and compute performance metrics relevant to such a system. Furthermore, the framework was expanded to include trajectory optimization, which was performed through the implementation of a collocation method.

Aircraft design, trajectory and coupled design and trajectory optimizations were performed for the same mission with the objective of minimizing energy. It was concluded that optimizing design and trajectory simultaneously had a great impact in achieving better results, as the isolated problems were greatly limited by the initial guesses for configuration and trajectory. The coupled optimization was able to further minimize energy in 33% and 10.8%, relative to the isolated trajectory and aircraft design optimizations. The coupled optimization was computationally more expensive, and it was also verified that the cost did not scale linearly with problem size.

The coupled design and trajectory were also optimized for other objectives, namely flight time and final distance. It was observed that time minimization yielded very similar results to those of energy minimization, proof of the tight dependence between both parameters. The maximization of final distance showed different results, where the battery was fully depleted and the wing was design to maximize aspect ratio.

6.2 Future Work

In this work, only equilibrium considerations were taken into account when designing the aircraft. An interesting development would be also accounting with stability.

Only 2D flight is considered in this thesis. A possible expansion of the framework would be developing a model that contemplates 3D flight and wind profiles. It could be interesting to test this expansion in missions like travelling to a target point and hovering around it or use a topographic map and plan trajectory to avoid collision.

We saw that the low order discretization method used had an impact in results. Another way the framework

could be improved is to use higher order discretization methods that guarantee continuity and smoothness of trajectory and control. The higher order discretization would also tackle the computational cost problem, as less aerostructural interactions would need to be computed.

Bibliography

[1] Kamov Ka-32A11BC Helicopter. URL https://www.rhc.aero/en/catalog/ka-32a11bc. [Accessed on 08/04/2021].

[2] Boeing V-22 Osprey. URL https://www.boeing.com/defense/v-22-osprey/. [Accessed on 08/04/2021].

[3] General Atomics MQ-9. URL https://www.ga-asi.com/remotely-piloted-aircraft/mq-9a. [Accessed on 08/04/2021].

[4] J. P. Jasa, J. T. Hwang, and J. R. R. A. Martins. Open-source coupled aerostructural optimization using Python. *Structural and Multidisciplinary Optimization*, 57(4):1815–1827, April 2018. doi: 10.1007/s00158-018-1912-8.

[5] A. de Sousa Cardeira. Aeroelastic analysis of aircraft wings. Master's thesis, Instituto Superior Técnico, Dec. 2014.

[6] Y. Zhiyin. Large-eddy simulation: Past, present and the future. *Chinese Journal of Aeronautics*, 28(1):11–24, 2015. doi: 10.1016/j.cja.2014.12.007.

[7] G. Alfonsi. On direct numerical simulation of turbulent flows. *Applied Mechanics Reviews*, 64(2):1–33, 2011.

[8] J. H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springer, 3rd edition, 2002.

[9] O. Reynolds. On the dynamical theory of incompressible viscous fluids and the determination of the criterion. *Philosophical Transactions of the Royal Society of London. (A.)*, 186:123–164, 1895. doi: 10.1098/rsta.1895.0004.

[10] A. Favre. Equations des gaz turbulents compressibles. *Journal de Mecanique*, 4(3):361–390, 1965.

[11] W. Jones and B. E. Launder. The prediction of laminarization with a two-equation model of turbulence. *International journal of heat and mass transfer*, 15(2):301–314, 1972.

[12] D. C. Wilcox. *Turbulence modeling for CFD*, volume 2. DCW industries La Canada, CA, 1998.

[13] D. C. Wilcox. Formulation of the kw turbulence model revisited. *AIAA journal*, 46(11):2823–2838, 2008.

[14] P. Spalart and S. Allmaras. A one-equation turbulence model for aerodynamic flows. *AIAA*, 439, 01 1992. doi: 10.2514/6.1992-439.

[15] F. R. Menter. Two-equation eddy-viscosity turbulence models for engineering applications. *AIAA Journal*, 32(8):1598–1605, 1994. doi: 10.2514/3.12149.

[16] P. A. Durbin. Near-wall turbulence closure modeling without "damping functions". *Theoretical and Computational Fluid Dynamics*, 3(1):1–13, 1991. doi: 10.1007/BF00271513.

[17] L. L. Erickson. Panel methods: An introduction. *NASA*, 1990.

[18] J. D. Anderson Jr. *Fundamentals of aerodynamics*. McGraw-Hill, 5th edition, 2010.

[19] D. Raymer. *Aircraft Design: A Conceptual Approach*. AIAA, 5th edition, 2012.

[20] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu. *The finite element method: its basis and fundamentals*. Elsevier, 6th edition, 2005.

[21] O. C. Zienkiewicz and R. L. Taylor. *The finite element method for solid and structural mechanics*. Elsevier, 2005.

[22] A. Bhaduri. *Mechanical properties and working of metals and alloys*, volume 264. Springer, 2018.

[23] F. Beer, E. Johnston, J. DeWolf, and D. Mazurek. *Mechanics of Materials. 7th Editio*. McGraw-Hill, New York, 2014.

[24] N. Cumpsty and A. Heyes. *Jet Propulsion: A Simple Guide to the Aerodynamics and Thermodynamic Design and Performance of Jet Engines*. Cambridge University Press, 3rd edition, 2015.

[25] P. G. Bruce, S. A. Freunberger, L. J. Hardwick, and J.-M. Tarascon. Li-O2 and Li-S batteries with high energy storage. *Nature materials*, 11(1):19, 2012.

[26] G. Girishkumar, B. McCloskey, A. C. Luntz, S. Swanson, and W. Wilcke. Lithium-Air battery: promise and challenges. *The Journal of Physical Chemistry Letters*, 1(14):2193–2203, 2010.

[27] Discharge curve of li-po battery. URL `http://learningrc.com/lipo-battery/`. [Accessed on 04/10/2020].

[28] M. Bronz, J.-M. Moschetta, and G. Hattenberger. Multi-Point Optimisation of a Propulsion Set as Applied to a Multi-Tasking MAV. In *IMAV 2012, International Micro Aerial Vehicle Conference and Competition*, Braunschweig, Germany, July 2012.

[29] B. W. McCormick. *Aerodynamics, Aeronautics and Flight Mechanics*. John Wiley & Sons, 2nd edition, 1995.

[30] S. S. Chauhan and J. R. R. A. Martins. Tilt-wing eVTOL takeoff trajectory optimization. *Journal of Aircraft*, 57(1):93–112, Jan. 2020. doi: 10.2514/1.C035476.

[31] M. Kelly. An introduction to trajectory optimization: How to do your own direct collocation. *SIAM Review*, 59(4):849–904, Nov. 2017. doi: 10.1137/16M1062569.

[32] J. T. Betts. Survey of numerical methods for trajectory optimization. *Journal of guidance, control, and dynamics*, 21(2):193–207, Mar. 1998. doi: 10.2514/2.4231.

[33] I. Mir, S. A. Eisa, and A. Maqsood. Review of dynamic soaring: technical aspects, nonlinear modeling perspectives and future directions. *Nonlinear Dynamics*, 94(4):3117–3144, Sept. 2018. doi: 10.1007/s11071-018-4540-3.

[34] E. Todorov. Optimal control theory. In K. Doya, editor, *Bayesian brain: Probabilistic approaches to neural coding*, pages 269–298. MIT Press, Cambridge, MA, 2006.

[35] R. Bellman. *Dynamic programming*. Princeton University Press, Princeton, New Jersey, 1957.

[36] A. V. Rao. A survey of numerical methods for optimal control. *Advances in the Astronautical Sciences*, 135 (1):497–528, 2009.

[37] T. J. Böhme and B. Frank. *Hybrid Systems, Optimal control and Hybrid Vehicles*. Springer International Publishing, Cham, Switzerland, 2017.

[38] D. E. Kirk. *Optimal Control Theory: an Introduction*. Dover Publications, Mineola, New York, 2004.

[39] G. T. Huntington and A. V. Rao. Comparison of global and local collocation methods for optimal control. *Journal of Guidance, Control, and Dynamics*, 31(2):432–436, 2008. doi: 10.2514/1.30915.

[40] D. Garg, M. Patterson, W. Hager, A. Rao, D. R. Benson, and G. T. Huntington. An overview of three pseudospectral methods for the numerical solution of optimal control problems. Preprint, Oct. 2017.

[41] O. Von Stryk and R. Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of operations research*, 37:357–373, 1992.

[42] M. Cervera, R. Codina, and M. Galindo. On the computational efficiency and implementation of block-iterative algorithms for nonlinear coupled problems. *Engineering computations*, 13(6):4–30, 1996. doi: 10.1108/02644409610128382.

[43] A. C. Marta. Aircraft optimal design. MSc Course Notes, 2019, Insituto Superior Técnico, Lisbon, Portugal.

[44] J. R. R. A. Martins and A. Ning. *Engineering Design Optimization*. Cambridge University Press, 2020. (to be published).

[45] S. S. Chauhan, J. T. Hwang, and J. R. R. A. Martins. Benchmarking approaches for the multidisciplinary analysis of complex systems using a Taylor series-based scalable problem. In *12th World Congress on Structural and Multidisciplinary Optimization*, Braunschweig, Germany, June 2017. doi: 10.1007/978-3-319-67988-4_7.

[46] S. S. Chauhan, J. T. Hwang, and J. R. R. A. Martins. An automated selection algorithm for nonlinear solvers in MDO. *Structural and Multidisciplinary Optimization*, 58(2):349–377, June 2018. doi: 10.1007/s00158-018-2004-5.

[47] J. R. R. A. Martins and A. B. Lambe. Multidisciplinary design optimization: A survey of architectures. *AIAA Journal*, 51(9):2049–2075, Sept. 2013. doi: 10.2514/1.J051895.

[48] A. B. Lambe and J. R. R. A. Martins. Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes. *Structural and Multidisciplinary Optimization*, 46: 273–284, 2012. doi: 10.1007/s00158-012-0763-y.

[49] E. J. Cramer, J. E. Dennis, Jr, P. D. Frank, R. M. Lewis, and G. R. Shubin. Problem formulation for multidisciplinary optimization. *SIAM Journal on Optimization*, 4(4):754–776, 1994.

[50] N. P. Tedford and J. R. R. A. Martins. On the common structure of MDO problems: A comparison of architectures. In *Proceedings of the 11th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Portsmouth, VA, Sept. 2006. AIAA 2006-7080.

[51] G. Venter. Review of optimization techniques. In R. Blockley and W. Shyy, editors, *Encyclopedia of Aerospace Engineering*. Wiley and Sons, 2010. doi: 10.1002/9780470686652.eae495.

[52] H. W. Kuhn and A. W. Tucker. Nonlinear programming. In J. Neyman, editor, *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 481–493. University of California Press, Berkeley, CA, 1951.

[53] P. T. Boggs and J. W. Tolle. Sequential quadratic programming. *Acta numerica*, 4:1–51, 1995.

[54] N. Gould and P. Toint. SQP methods for large-scale nonlinear programming. In M. J. D. Powell and S. Scholtes, editors, *System Modelling and Optimization, Methods, Theory and Applications*, pages 149–178. Kluwer Academic Publishers, 2000.

[55] Z. Lyu, Z. Xu, and J. R. R. A. 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. ICCFD8-2014-0203.

[56] P. E. Gill, W. Murray, and M. A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM review*, 47(1):99–131, 2005.

[57] D. Kraft. A software package for sequential quadratic programming. *DLR German Aerospace Center—Institute for Flight Mechanics*, 1988. Technical report DFVLR-FB 88-28.

[58] C. L. Lawson and R. J. Hanson. *Solving least squares problems*. SIAM, 1995.

[59] J. S. Gray, J. T. Hwang, J. R. R. A. Martins, K. T. Moore, and B. A. Naylor. OpenMDAO: An open-source framework for multidisciplinary design, analysis, and optimization. *Structural and Multidisciplinary Optimization*, 59(4):1075–1104, April 2019. doi: 10.1007/s00158-019-02211-z.

[60] J. T. Hwang and J. R. Martins. A computational architecture for coupling heterogeneous numerical models and computing coupled derivatives. *Association for Computing Machinery*, 44(4), 2018. doi: 10.1145/3182393.

[61] J. R. R. A. Martins and J. T. Hwang. Review and unification of methods for computing derivatives of multidisciplinary computational models. *AIAA Journal*, 51(11):2582–2599, 2013. doi: 10.2514/1.J052184.

[62] N. Wu, G. Kenway, C. A. Mader, J. Jasa, and J. R. R. A. 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.

[63] P. Virtanen et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

[64] Tekever AR4 UAV. URL http://uas.tekever.com/ar4-evo/. [Accessed on 12/09/2020].

[65] M. Drela. Xfoil: An analysis and design system for low reynolds number airfoils. In T. J. Mueller, editor, *Low Reynolds Number Aerodynamics*, pages 1–12, Berlin, Heidelberg, 1989. Springer Berlin Heidelberg. ISBN 978-3-642-84010-4.

[66] Y. Liao, S. He, J. R. R. A. Martins, and Y. L. Young. Hydrostructural optimization of generic composite hydrofoils. In *AIAA SciTech Forum*, Orlando, FL, Jan. 2020. AIAA. doi: 10.2514/6.2020-0164.

[67] V. d. Brederode. *Aerodinâmica Incompressível: Fundamentos*. IST Press, Lisboa, 2014.