UNIVERSIDADE DA BEIRA INTERIOR
Engenharia

Mission Planner for Solar Powered Unmanned Aerial Vehicles
(Versão corrigida após defesa)

Luís Miguel Marques Coelho

Dissertação para obtenção do Grau de Mestre em
Engenharia Aeronáutica
(Ciclo de Estudos Integrado)

Orientador: Professor Doutor Pedro Viera Gamboa

Covilhã, abril de 2019

Acknowledgments

After living these wonderful student years I have all the pride to say I have the best mother in the world, and I thank her for having worked so hard these years, so I can feel this immense happiness and pride in myself.

I also thank my advisor, Professor Pedro Vieira Gamboa, for all the patience and help during this dissertation.

And finally, I thank my girlfriend and all my friends who gave me motivation and focus to finish and deliver an honest work to be proud of.

To my beloved father.

Resumo

Os veículos aéreos não tripulados (UAV's), inicialmente utilizados para aplicações militares, tornaram-se cada vez mais atraentes para fins civis. A utilização deste tipo de aeronave tem crescido exponencialmente nos últimos anos, tanto para fins profissionais como recreativos, devido às inúmeras vantagens que apresentam. Este aumento da procura levou a um crescente investimento no setor, nomeadamente nos UAVs movidos a energia solar, que hoje em dia já ocupam uma pequena fatia do mercado. No entanto, com o aparecimento deste tipo de UAV's, os softwares de planeamento de missões precisam de ser atualizados de forma a terem em conta a energia fornecida pelo sol. Desta forma, o presente trabalho descreve o desenvolvimento e validação de um planeador de missões para UAV's movidos a energia solar, capaz de planear e otimizar uma missão, considerando uma estimativa inicial dos parâmetros de cada waypoint (latitude, longitude, altitude e velocidade), e ainda considerando dados reais de previsão meteorologica e elevação de terreno. Para isso, o planeador de missões considera vários modelos matemáticos, necessários para o cálculo do desempenho da missão, e um algoritmo quadrático sequencial de forma a otimizar a missão inicial. Depois de descrever os modelos teóricos, uma aplicação prática do planeador de missão é feita com o objetivo de verificar o seu desempenho. Em relação à validação, vários resultados divididos por tópicos de interesse são apresentados e discutidos, concluindo: é eficiente em relação ao planeamento de missões, ainda assim, tendo alguns aspetos a serem melhorados.

Palavras-chave

Planeamento de missões, veículos aérios não tripulados, algoritmo de otimização, modelo solar

Abstract

Unmanned aerial vehicles (UAV's), initially used for military applications, have become increasingly attractive for civilian purposes. The use of this type of aircraft has grown exponentially in recent years, both for professional and recreational purposes, due to the numerous advantages they present. The increasingly demand of UAV led to an increase in investment, namely in the development of solar powered UAVs. Nowadays, with the arising of this type of UAV's, the mission planners have to start to be updated with new features considering UAV's with photovoltaic solar panels. This way, the present work describes the development and validation of a mission planner for solar powered UAV's, capable of planning and optimizing a mission given a initial guess of waypoints parameters (latitude, longitude, altitude and airspeed), considering real weather forecast and terrain elevation data. For this, the mission planner considers several mathematical models, required for the calculation of the mission performance, and a sequential quadratic programming algorithm to optimize the initial mission. After it describes the theoretical models, a practical application of the mission planner is done in order to verify its performance. Regarding its validation, several results divided by topics of interest are presented and discussed, concluding that the mission planner works efficiently, regarding the mission planning, even though, it has some aspects to be improved.

Keywords

Mission planning, unmanned aerial vehicles, optimization algorithm, solar model

Contents

List of Figures

List of Tables

Acronyms List

AGI	Analytical Graphics inc
API	Application Program Interface
ASL	Autonomous System Lab
ENU	East North Up
ERAST	Environmental Research Aircraft Sensor Technology
ESC	Electronical Speed Controller
FFSQP	FORTRAN Feasible Sequential Quadratic Programming
GBAD	Ground Based Air Defence
LEEUAV	Long Endurance Electric Unmanned Aerial Vehicle
MPT	Mission Planning Tools
NASA	National Aeronautics and Space Administration
PV	Photovoltaic
RRT	Rapidly-exploring Random Tree
SQP	Sequential Quadratic Programming
UAV	Unmanned Aerial Vehicle
UBI	Universidade da Beira Anterior

Nomenclature

a	Acceleration $[m/s^2]$
A_Z	Sun's azimuth $[deg.]$
A_{Z_S}	Photovoltaic panel's azimuth $[deg.]$
$\overline{A}, \overline{B}, \overline{C}, \overline{D}, \overline{E}$	Coefficient vectors
B_{ref}	Energy left constraint reference value
C_c	Cloud cover $[Oktas]$
C_D	Drag coefficient
C_L	Lift coefficient
$C_{L_{max}}$	Maximum Lift coefficient
$C_{L_{TO}}$	Take-off Lift coefficient
C_p	Power coefficient
C_{p_0}	Power coefficient at null advance ratio
D	Drag Force $[N]$
d	Propeller diameter $[m]$
d_g	Ground distance $[m]$
d_h	Height variation $[m]$
d_n	Day of the year
d_v	Design variables
E	Mission's consumed Energy $[J]$
E_{left}	Energy left in the battery
$E_{solar\|pred}$	Predicted Solar energy collected with horizontal photovoltaic panels $[J]$
$E_{solar\|Total}$	Total Solar energy collected with tilted photovoltaic panels $[J]$
F	Rolling friction force $[N]$
g	Gravity acceleration $[m/s^2]$
H	Hour angle $[deg.]$
h	Altitude of flight $[m]$
h_p	Actual flight height $[m]$
h_{ref}	Minimum height constraint reference value $[m]$
i	Current value at each waypoint
I	Electric current $[A]$
I_{eff}	Effective current $[A]$
I_{max}	Maximum electric current $[A]$
I_0	No load current $[A]$
j	Current value at each segment
J	Solar irradiation $[W/m^2]$
J_{cloud}	Solar power collected considering the cloud cover $[W/m^2]$
J_{0n}	Intensity of the extraterrestrial normal solar irradiation $[W/m^2]$
J_{max}	Propeller maximum advance ratio
J_{SC}	extraterrestrial normal solar irradiation constant $[W/m^2]$
J_{TS}	Solar power in a tilted photovoltaic panel $[W/m^2]$
K_c	Clear sky index

K_t	Motor torque constant
K_v	Motor speed Constant
L	Lift force $[N]$
M	Mass $[kg]$
N	Propeller speed $[rpm]$
n	Load factor
p	Propeller pitch $[m]$
P_{bat}	Battery power $[W]$
P_{eff}	Effective power $[W]$
P_{ele}	Electric power $[W]$
P_{motor}	Motor power $[W]$
P_{req}	Required power $[W]$
P_{shaft}	Motor power at the shaft $[W]$
P_{solar}	Solar power $[W]$
P_{sys}	Systems power $[W]$
P_T	Total electric power $[W]$
Q	Coordinates of the elevation map's diagonal $[deg.]$
$Q_{elevation}$	Elevation data $[m]$
Q_{IF}	Inertial Force $[N]$
Q_m	Available torque at the shaft $[Nm]$
$Q_{weather}$	Weather data
R	Electric resistance $[\Omega]$
R_{ESC}	Motor speed controller resistance $[\Omega]$
R_U	Battery resistance $[\Omega]$
RC	Rate of climb $[m/s]$
r_{es}	Sun-Earth distance $[km]$
$r_{es,0}$	Mean Sun-Earth distance $[km]$
S	Wing area $[m^2]$
S_{PV}	Photovoltaic panels area $[m^2]$
T	Thrust $[N]$
t	Time $[s]$
U	Input Voltage $[V]$
U_{eff}	Effective voltage $[V]$
v	True anomaly $[deg.]$
V	Aircraft airspeed $[m/s]$
V_a	Average airspeed $[m/s]$
V_{ave}	Average squared airspeed $[m^2/s^2]$
V_g	Ground speed $[m/s]$
V_{g_a}	Average ground speed $[m/s]$
V_{safety}	Safety speed factor
V_{stall}	Stall speed $[m/s]$
V_w	Wind speed $[m/s]$
V_{w_a}	Average wind speed $[m/s]$
w	Waypoint number
W	Aircraft weight $[N]$

Greek Letters

α	Angle of the photovoltaic panel with the horizontal $[deg.]$
β	Angle of the photovoltaic panel with the normal to the centre of the Earth $[deg.]$
γ_a	Air path angle $[deg.]$
γ_g	Angle of trajectory relative to ground $[deg.]$
δ_s	Solar declination angle $[deg.]$
δ_{set}	Motor power setting
δ_{ref}	Maximum motor setting constraint reference value
ε	Eccentricity ratio of earth
ζ	Zenith angle $[deg.]$
$\eta_{gearbox}$	Gear box efficiency
η_{max}	Maximum propeller efficiency
η_{motor}	Motor efficiency
η_p	Propeller efficiency
η_{PV}	Photovoltaic panels efficiency
θ_g	Angle of trajectory projection on xy-plane relative to x-axis $[deg.]$
θ_i	Angle of incidence of the Sun $[deg.]$
θ_g	Angle of incidence of the wind $[deg.]$
λ	Longitude $[deg.]$
μ	Coefficient of static friction
ρ	Air's relative density $[kg/m^3]$
τ	Transmittance factor
ϕ	Latitude $[deg.]$
Ω	Motor speed $[rpm]$

Chapter 1

Introduction

This chapter, first introduces the mission planning by a general aproach, followed by the motivation and the objectives of this work. The goal of this chapter is to help the reader become familiar with the mission planning concept and understand why and for what purposes this work was done.

1.1 Mission Planning Approach

With all the evolution in aerial vehicles, an entire range of practical problems appeared to be solved. One of those problems was the mission planning problem. Since aircraft routes and flight frequencies are essential for making airline timetables, it is important to plan them effectively in order to achieve a profitable timetable [1].

The mission planning, or path planning problem, has long been seen as one of the fundamental problems in aviation. Originally arising from the need of pre-defined path for each aircraft to follow and so prevent saturated air traffic and mid-air collisions. It is the process of producing a flight plan to describe a proposed aircraft flight between a starting and an ending point. It involves several aspects: fuel calculation, to ensure that the aircraft is able to reach the destination, the aircraft control limitations to secure that the vehicle can do the maneuvers to avoid obstacles, and terrain data to know if there is any possible obstacle to avoid [2].

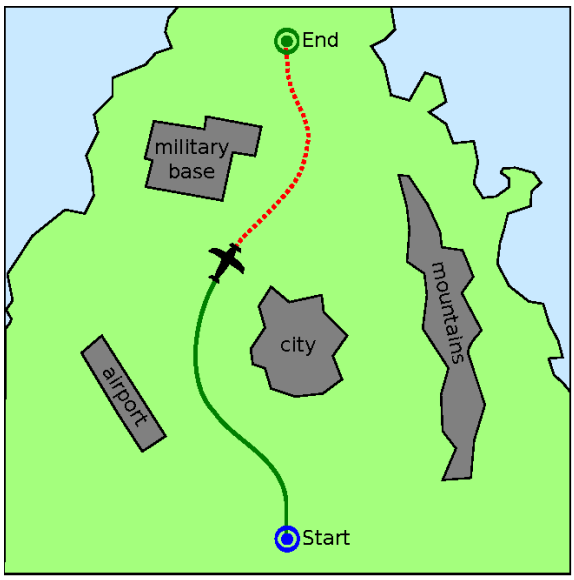


Figure 1.1: Example of a mission planning problem solution. [3]

1.2 Motivation

The first applications of unmanned aerial vehicles (UAV's) were military-type. In spite of that, their reducing costs and their increasing capacity made them attractive for civilian applications. Nowadays, the mission planning has become even more important with the increased use of UAV's. In surveillance missions, for example, UAV's need a predefined plan of an optimal route with obstacle avoidance and target reaching.

New assigned tasks to UAV's are demanding better management of energy, in order to reach specific objectives without compromising safety. Therefore, a mission planner that optimizes mission parameters like energy or time can be very useful while setting up the vehicle for optimal performance. In addition, aircraft performance optimization and flight tests may prove themselves very difficult to do without a previous mission plan. In autonomous flights, if a prototype is tested without a mission plan in unknown environments, there is the risk of a crash and, in that case, all the work is lost. In order to avoid that, many mission planning software were created. Yet for non-profit institutions like universities, those software may prove too expensive for the institution to afford it.

In the Department of Aerospace Sciences of the University of Beira Interior there is the Long Endurance Electric UAV (LEEUAV). As it's name suggests, this UAV was built to do long endurance missions, also, it is a solar powered UAV. So, the main reason for the development of the present thesis is to plan viable missions assuring the safety and the success of its flight.

1.3 Objectives

The main objective of this thesis is to develop and validate a mission planner for solar powered UAV's. To achieve this objective, a detailed description of the various tasks is shown below:

- Identification of the parameters that feed the mathematical models;
- Development and implementation of a mission performance model required to estimate all the necessary parameters to define a mission;
- Selection of databases that provide real weather forecast and terrain elevation data;
- Development and implementation of models that request and return, from the selected databases, the required data for the mission performance model;
- Implementation of a propulsion performance model;
- Implementation of a solar model required to estimate the total energy harvested from the photovoltaic cells;
- Development of the flight energy management algorithm required to estimate the total/partial energy, as well the rates of change;
- Selection and implementation of an algorithm that can solve the mission planning problem;
- Verification of the mission planner.

Chapter 2

State of Art

This chapter first gives an overview of common approaches taken to solve the mission planning problem, then a brief introduction to the solar-powered aircraft is given as well as an historic contextualization of the field of study. Finally, the chapter ends with the presentation of practical applications of different planning tools.

2.1 Methods and Algorithms

To solve the path planning problem it is necessary to research methods and modern computer technology. There are currently five types of methods that can be used to solve this problem [4] [5]

2.1.1 Roadmap-based Method

A roadmap-based method consists in constructing a map that represents the environment space constraints, and then using a search algorithm to find the shortest path [6]. One example of this type of methods is the Voronoi Map method, example in Figure 2.1 . As its name would suggest, it is based on Voronoi diagrams which are used to extract the network representation of the environment. After that, as mentioned above, a search algorithm is used to go through the map analysing the possibilities and, by these means finding the best path.

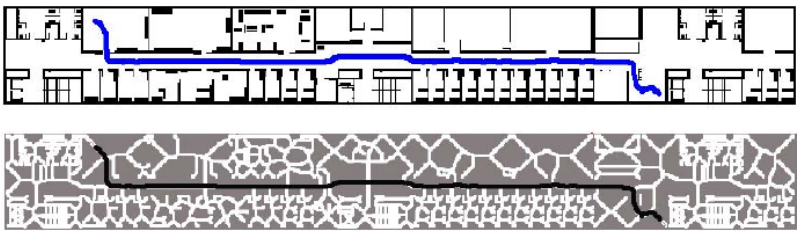


Figure 2.1: Trajectory calculated with a Voronoi method. [6]

2.1.2 Heuristic Search Algorithm

This type of methods, e.g. Dijkstra's algorithm, involves knowing some special information about the domain of the problem, so that it is possible to evaluate the heuristic cost of each solution by an heuristic function. It constructs nonoverlapping regions that cover free space and encode cell connectivity in a graph [7]. Dijkstra's algorithm is used to find the shortest path between two points, it picks the unvisited node with the lowest distance, calculates the distance through it to each unvisited neighbor, and updates the neighbor's distance if it is smaller. In Figure 2.2 an example of a cell connectivity and the cost of each path is represented.

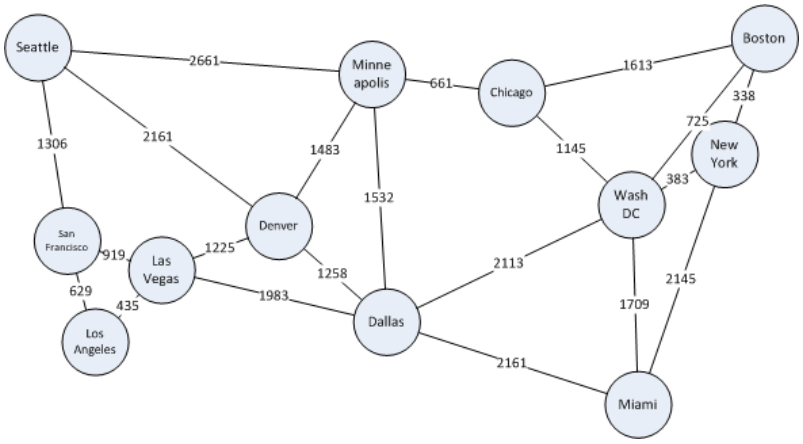


Figure 2.2: Example of a graph with the nodes that can be visited and each cost on the segments. [8]

2.1.3 Stochastic Programming Methods

Many Stochastic Programming approaches are based on random sampling, which has a component of probability, e.g. rapidly-exploring random tree (RRT). This algorithm provides a way to search high-dimensional spaces efficiently. It consists of a tree data-structure of samples in the space, created by an algorithm in a way that provides good coverage. The tree-construction algorithm consists in a loop of the following operations: firstly, RRT picks a random sample in the search space, secondly, finds the nearest neighbor of that sample, thirdly, selects an action from the neighbor that heads towards the random sample, then, creates a new sample based on the outcome of the action applied to the neighbor and lastly, adds the new sample to the tree, connecting it to the neighbour. In addition, RRT is biased to grow towards large unsearched areas of the problem [9].

2.1.4 Potential Field-based Methods

Potential field methods are based on the concept of electrical charges. If we see an UAV as an electrically-charged particle Figure 2.3, then obstacles should have the same type of electrical charge in a way to repulse the UAV. Following this group of methods the stream function is constantly used to path finding and obstacle avoidance.

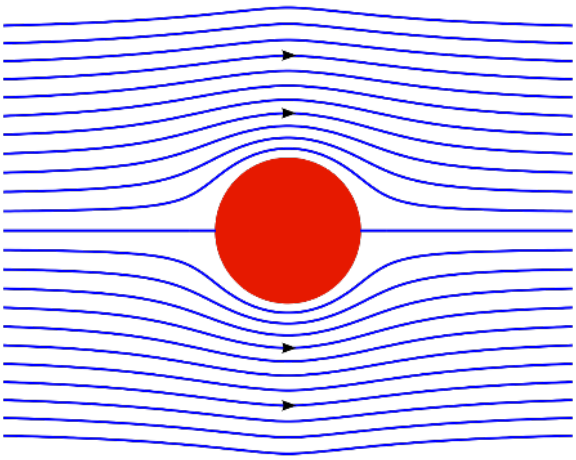


Figure 2.3: Example of a potential field based algorithm paths. [8]

2.1.5 Optimization algorithm Methods

Optimization algorithms help to minimize or maximize an objective function ,$f(x)$, which is a mathematical function dependent on the Model's internal parameters. They are used in computing the target design variables ,x, taking into account the constraints functions, $g(x)$, defined in the model. An optimization problem can be represented in the following way:

- Given a function $f : A \to \mathbb{R}$
- Sought an element $x_0 \in A$ such that $f(x_0) \leq f(x)$ for all $x \in A$ (minimization) or such that $f(x_0) \geq f(x)$ for all $x \in A$ (maximization)

Many real-world and theoretical problems may be modelled in this general framework. Typically, A is some subset of the Euclidean space $\mathbb{R}^n$, often specified by a set of constraints, equalities or inequalities that the members of A have to satisfy. The domain A of f is called the search space or the choice set, while the elements of A are called candidate solutions or feasible solutions. A local minimum is at least as good as any nearby elements, and a global minimum is at least as good as every feasible element [10]. Generally, unless the objective function is convex in a minimization problem, there may be several local minima. In a convex problem, if there is a local minimum that is interior (not on the edge of the set of feasible elements), it is also the global minimum, but a non-convex problem may have more than one local minimum not all of which need to be global minima [10].

2.1.6 Methods Comparison

These different methods are normally combined to use their best capabilities in order to better solve the path planning problem. The roadmap-based method is usually used to extract a network representation of the environment. This helps to stablish the space boundaries of the path, yet it needs a search algorithm to evaluate and optimize each path. On the other hand, heuristic search algorithms need a previously defined space with the heuristic cost of each segment of the map pre-defined. Therefore, sampling-based can also serve as a search algorithm, being faster than the heuristic search algorithms as it does not need to visit each path to find a solution, yet that solution may not be the global solution as it works by randomly exploring the space. The potential field based methods have a great behaviour in respect to obstacle avoidance, even though the path is limited to the defined stream functions. Finally, optimization algorithm can be used to solve any problem, if it is expressed as a mathematical function, as well as, optimize more than one function, yet, it may not find the global solution and find only a local solution like the sampling based methods.

2.2 Brief History of Electrical and Solar Powered Flight

Since 1884, when a couple of French army officers named Renard and Krebs used a hydrogen-filled dirigible powered by batteries and won a 10 km race around Villacoulbay and Medon, the electrical aerial vehicles became a possibility for the researchers [11]. In spite of the this success for the electrical motors, after the arrival of the piston engines this clean energy motors were abandoned almost for a century [11][13]. Nevertheless, a first step towards solar powered flight had been taken.

In the 1960s Fred Militky of Kerkheim Tech begun to work with lightweight free flight models powered by toy motors and one shot saline batteries. Although these models have flown, they were considered unsuccessful. Later in 1970 Roland and Robert Boucher started their experiments with electrical flight [12]. At this point, the combination of electrical flight and solar cells was made for the first time. Robert Boucher built a couple of pilotless solar-powered aircraft under contracts with the Defence Advanced Research Projects Agency [11].

In Germany, there was another solar model airplane project in work, which was under the responsibility of Helmut Bruss. Yet this model could not achieve level flight due to the over heating of its solar cells [12]. One year later, his friend Fred Militky, achieve the first flight totally powered by solar energy, with Solaris. This flight was made on the 16th of August 1976, when it reach 50m of altitude by making three flights by 150 seconds each[12].

Nowadays, aircraft internal combustion engines have a much higher endurance than electrical motors. The problem is not the high energy required by the electrical flight, is because it results in an increase of weight and space. The combination of electric flight and solar cells has become very useful since the batteries size and weight can be reduced, thus allowing long endurance flights.

This reintegration of electrical powered flight, led to an investment in many researches and experiments in electrical aircraft, trying to achieve better results, namely in the efficiency and endurance field [13].

2.2.1 Evolution of Solar Powered Aircraft

The evolution in the electrical flight field made possible to the pioneers to start experiments for the solar powered flight. The first solar powered aircraft, a radio controlled model plane called Sunrise I, was develop by Roland Boucher and made its first flight in 4th of November of 1974 in California [14]. Four years later, on 19 December 1978, Solar One, a manned solar-powered aircraft developed by Britons David Williams and Fred To, achieved its first flight at Lasham Airfield, Hampshire. This conventional shoulder wing monoplane was build in the first place to be human powered. However, it was proved too heavy to be so. Thus, that is how the Solar One became a solar-powered aircraft. Fred To was convinced that, if the wings upper surface was covered with high-efficiency solar cells like the ones used on Sunrise, probably it would be able to fly without the need of batteries. Yet these cells were considered too expensive to afford [11]. Four months later, at Flabob airport, California, the Solar Riser flew for the first time piloted by Larry Mauro. The battery has to be charged for three hours in order to power the motor for ten minutes covering a distance of about 800 m varying between 1.5 m and 5 m of altitude [12].

Aeronautical competitions are always a great method to promote research in the field. The Berblinger flight competition, which is a aeronautical competition hosted by Ulm in Germany, is a proof of that [15]. In the event of 1996, the motorglider Icaré 2 won the contest being the only one ready to fly in the final competition [12]. In the same event, there were also two other interesting competitors. The Sole Mio from the Italian team of Dr. Antonio Bubbico and Solair II of the team of Prof. Günter Rochelt. They did not fulfil the competition airworthiness directives, besides the fact they were in an advance stage of development. Finally in 1998,

Solair II made its first flight [12].

2.2.2 High Altitude Long Endurance Platforms and Eternal Flight

During the last two decades, many projects were developed with purpose to achieve a long endurance and eternal flight. One of this projects was the Pathfinder, project under a US government classified program that was abandoned and reactivated by the Ballistic Missile Defense Organization Organization having achieved its first flight at NASA Dryden in 1993 [16]. After this program ended, NASA's Environmental Research Aircraft Sensor Technology (ERAST) took responsibility of it in 1994. In 1995, it set a new altitude record for solar-powered aircraft by reaching 15392 m and only two years later it set the record to 21802 m [16]. Five years later, a successor of the Pathfinder was developed, the Centurion. It was a modified version of Pathfinder in order to be more efficient. [17].

Finishing this series of prototypes, Helios was the culmination of the group's solar-powered aircraft that, in August, 2001, reached an official world record altitude for a non-rocket powered aircraft, of 29523 m during a maximum-altitude flight. NASA had the objective of flying for 24 hours without stopping. It never achieved this objective as it was destroyed in a crash in the Pacific Ocean in 2003 due to structural failures. [12]

The objective of Helios was reached when Solong flew during 24 hours and 11 minutes without stopping on the 22^{nd} of April 2005, by Alan Cocconi. It used only the available solar energy powered by its solar panels and the currents of warm air rising from the desert floor. Two months later, Solong confirmed its capabilities by flying during 48 hours and 16 minutes in California's Colorado Desert [12].

A British company designated as QinetiQ, was working in a Zephyr aircraft. In 10^{th} of September 2007, One Zephyr, during trials at the US Military's White Sands Missile Range in New Mexico, set official world record time for the longest duration unmanned flight with a 54 hour flight in New Mexico [18] [12].

A project named Solar Impulse started in 2003, with the objective to become the first manned aircraft to accomplish the circumnavigation of the Earth powered enterely by solar energy [13]. On 8 July 2010, Solar Impulse I, achieved the world's first manned 26-hour solar-powered flight. At the time, the flight was the longest and highest ever flown by a manned solar-powered aircraft[19]. Even before the Solar Impulse I had successfully completed its first intercontinental flight in 2012, the Solar Impulse II had started being built in 2011, but a structural failure of the aircraft's main spar occurred during static tests in July 2012, delaying the flight tests. The repair work to the aircraft's main spar delayed Solar Impulse 2's circumnavigation of the Earth from 2012 to 2015 [20]. In the summer of 2016 the Solar Impulse 2 achieved its circumnavigation mission when it landed in Abu Dhabi after 16 and half months reaching 42000 kilometres of flight bowered only by solar energy [21].

In the Autonomous Systems Lab (ASL) of Zurich an unmanned solar aircraft project, the AtlantikSolar was created with the main objective of being the first fully solar powered aircraft to

cross the Atlantic Ocean, a journey of 5000km and 7 days of continuous flight. The AtlantikSolar has already proved its value by successfully completing an 81-hour continuous flight [22].

2.3 Mission Planning Tools

2.3.1 Mission Planner - Ardupilot

One of the most utilized mission planning tools is the Mission Planner, which is a full-featured ground station application for the ArduPilot, an open source autopilot project. This ground control station can be used as a configuration utility or as a dynamic control complement for an UAV [23]. Some features of Mission Planner are the capability of loading the software into the autopilot board that controls a specific UAV; setup and configure the UAV for an optimized performance; plan, save and load autonomous missions into the autopilot with simple point-and-click way-point entry on Google Maps or others [23]. A Mission Planner ground station interface example is shown in Figure 2.4.

The UAV can also be programmed to take off and land autonomously, and loiter over any way-point for a specified number of turns for a given duration, while acquiring aerial photographs or video. The user can also program other flight parameters such as ground/air speed and altitude of the drone over each waypoint. A pre-programmed mission can be uploaded to the UAV before launch. But even when the UAV is already in the air, a new mission can still be programmed and uploaded via data telemetry to give new instructions to the UAV [23].

Figure 2.4: Mission Planner ground station interface example [23].

2.3.2 QBase Mission Planner Software - Quantum-Systems

The company Quantum-Systems GmbH was founded in January 2015 and is specialized in the development and production of automatic transition UAV's for civilian use. To help its clients it also developed a mission planning software that plans with a few input parameters, its UAV's missions [24].

Figure 2.5: QBase Mission Planner Interface.

QBase automatically generates efficient flight paths after the flight area and the mission parameters have been defined. These parameters are the mapping zone boundaries, shown in Figure 2.6, the selected UAV for the mission, the wind speed and direction, which has to be checked online by the user, the speed of flight, the altitude of flight and the used payload, (cameras, sensors), which define the type of the mission. The user also has to check and assure that there is no obstacle in the flight path. Then, the software generates a mission/trajectory, e.g. Figure 2.6. It also provides a monitoring of some relevant parameters of the mission like, battery status, flight time, altitude and number of recorded images of the payload [24].

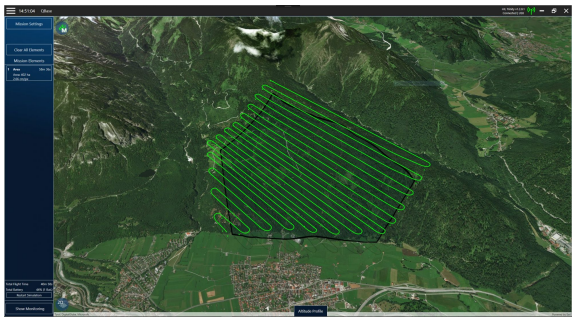


Figure 2.6: QBase Mission Planner Mission [24].

Chapter 3

Theoretical algorithm and Methodology

The main focus of the current chapter is to describe the chosen method to plan the mission for a solar powered UAV. For better understanding and organization during the development of this thesis, the method was divided in individual models. Each module will be described by Subsections along this chapter. In Figure 3.1 it can be seen the four models linked to one central box, these four models are independent subroutines that return the needed data to the main program. The main program is separated in two main phases, mission analysis and mission optimization. Starting with an analysis of a non-optimized initial mission (design variables), followed by an optimization phase, it calculates new design variables to optimize an objective function, taking in account some pre-defined constraints. The final optimized solution is found by an iterative process composed of a sequence of analyses and optimizations that is explained through this chapter.

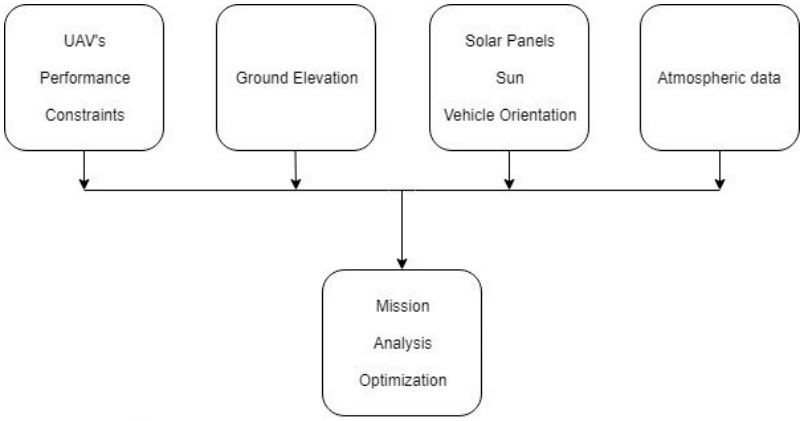


Figure 3.1: Individual models used for the mission planning

3.1 Mission Analysis

The analysis phase purpose is to calculate the objective function and its constraints, from the design variables. The design variables are each latitude, longitude, altitude and airspeed of a set of waypoints that define the mission the user wants to plan. This is represented by Equation 3.1.

$$dv^{4n} = (\varphi_1, \varphi_2, \ldots, \varphi_n, \lambda_1, \lambda_2, \ldots, \lambda_n, h_1, h_2, \ldots, h_{n-1}, h_n, V_1, V_2, \ldots, V_n)$$
$$\varphi_i, \lambda_i, h_i, v_i \in \mathbb{N} for 1 \leq i \leq n \quad (3.1)$$

where dv is the list of design variables, φ and λ represent the coordinates (latitude and lon-

gitude, respectively) of each waypoint, h corresponds the altitude of flight at each waypoint measured from the take-off ground elevation and V is the airspeed at each waypoint. The number of parameters is four times the number of waypoints, n. There are two types of design variables: the first one is expressed in Eq. 3.1 in which the coordinates system is represented by the geographic coordinates represented in decimal degrees; the second one is represented by the Eq.3.2:

$$dv^{4n} = (x_1, x_2, \ldots, x_n, y_1, y_2, \ldots, y_n, h_1, h_2, \ldots, h_{n-1}, h_n, V_1, V_2, \ldots, V_n)$$
$$x_i, y_i, h_i, v_i \in \mathbb{N} for 1 \leq i \leq n \quad (3.2)$$

in which x and y are represented in the East North Up (ENU) coordinates. If the design variables are defined in the geographic coordinates format, the algorithm of analysis has to convert the coordinates to the ENU system, so the final inputs are in SI units. If the design variables are already defined in the second format the mission analysis will proceed. Figure 3.2 shows the Initial trajectory representation, where each waypoint represents each x_i, y_i and h_i, where i

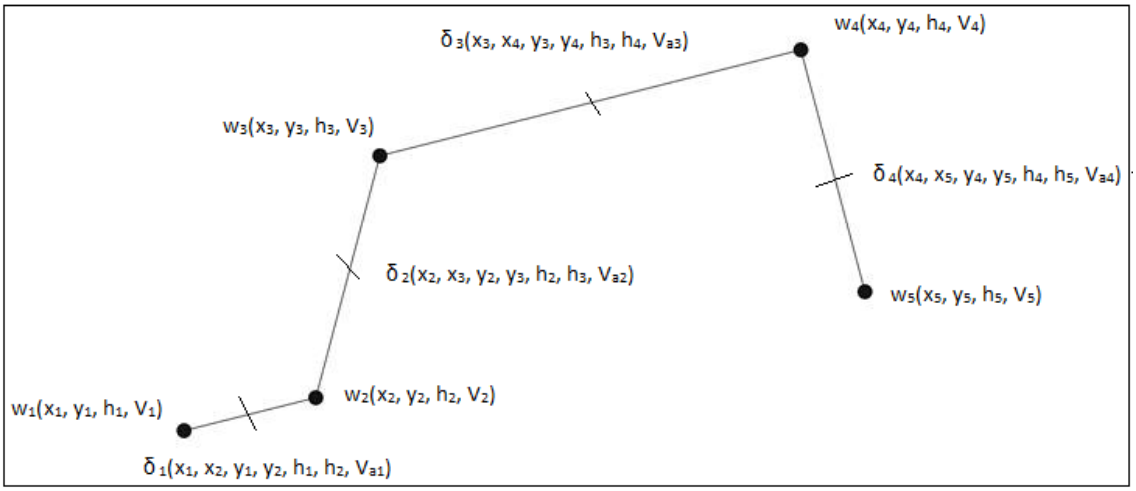


Figure 3.2: Initial 5 waypoints trajectory example.

is the waypoint index, the correspondent V_i is saved in the program in each waypoint vector. The analysis is done by segment performance, which represents the segment performance δ_j of flight between a waypoint w_i and a waypoint w_{i+1}. It is important to note that, in the following equations, if the parameter is referent to a waypoint i is used as the parameter index. On the other hand, if the parameter is referent to a segment, it has j as the correspondent index. The next step of the analysis is to calculate each segment performance parameter, starting with the segment total distance, that is given by

$$ds_j = \sqrt{dg_j^2 + dh_j^2} \quad (3.3)$$

where dg_j is the segment ground distance, calculated by

$$dg_j = \sqrt{(x_{j+1} - x_j)^2 + (y_{j+1} - y_j)^2} \quad (3.4)$$

and, dh_j is the segment height variation that can be written as

$$dh_j = h_{j+1} - h_j \quad (3.5)$$

The next step of the mission analysis algorithm is to calculate the average segment airspeed along the path, that is represented by,

$$V_{aj} = \frac{V_{j+1} + V_j}{2} \quad (3.6)$$

followed by the calculation of average segment wind speed components Vw_{ax_j}, Vw_{ay_j},Vw_{az_j}. It should be mentioned that this step is dependent on the atmospheric data model. The model described in Section 3.6, will return to this step the wind speed Vw at each waypoint and the corresponding angle of orientation θ_{w_i} which assumes the north-clockwise convention, $\theta_{w_i} \in [0,\ 360[$ where 0 represents the North $\rightarrow$ South way.

So the average segment wind speed is given by:

$$Vw_{aj}(Vw_{ax_j}, Vw_{ay_j}, Vw_{az_j}) \quad (3.7)$$

where,

$$Vw_{ax_j} = \frac{Vw_{x_j} + Vw_{x_{j+1}}}{2} \quad (3.8a)$$
$$Vw_{ay_j} = \frac{Vw_{y_j} + Vw_{y_{j+1}}}{2} \quad (3.8b)$$
$$Vw_{az_j} = \frac{Vw_{z_j} + Vw_{z_{j+1}}}{2} \quad (3.8c)$$

and

$$Vw_{x_i} = Vw_i \cos(90 - \theta_{w_i}) \quad (3.9a)$$
$$Vw_{y_i} = Vw_i \sin(90 - \theta_{w_i}) \quad (3.9b)$$
$$Vw_{z_i} = 0 \quad (3.9c)$$

The vertical wind speed component is considered zero due to weather forecast databases limitations, yet the model considered 3D components to be used if a more complete weather database is implemented in a next version of the software. Hence,

$$Vw_{aj} = \sqrt{{Vw_{ax_j}}^2 + {Vw_{ay_j}}^2 + {Vw_{az_j}}^2} \quad (3.10)$$

Due to lack of the database information the angle of orientation of the wind speed, θ_{w_i}, is two dimensional (x, y), so $Vw_{a_{z_j}}$ is considered $zero$.

Once the average wind speed is calculated it is necessary to know the flight path direction. In Figure 3.3, between two waypoints d_s, d_g and d_h are already known from Equations 3.3, 3.4 and 3.5 respectively, θ_{g_j} and γ_{g_j} are described bellow in equations 3.11 and 3.12.

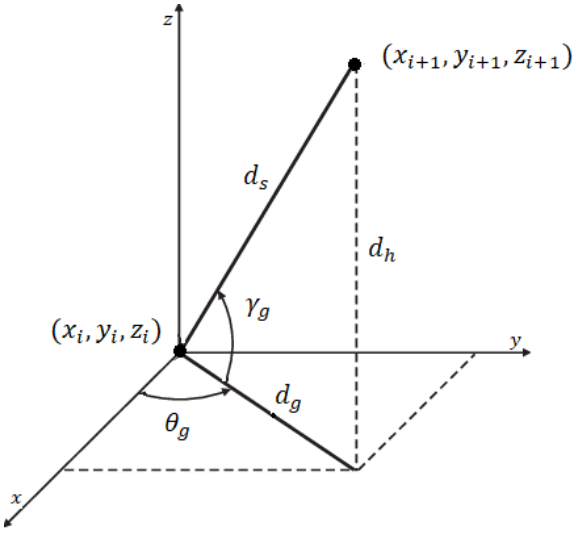


Figure 3.3: Flight Path Direction Scheme

$$\begin{aligned}\theta_{g_j} &= \arctan\left(\frac{y_{j+1} - y_j}{x_{j+1} - x_j}\right) \\ &= \arcsin(\frac{d_{gy_j}}{d_{gx_j}}) \end{aligned} \quad (3.11)$$

$$\gamma_{g_j} = \arctan\frac{dh_j}{ds_j} \quad (3.12)$$

With this, the next step is to calculate the average ground speed Vg_{a_j},

$$Vg_{a_j}(Vg_{a_{x_j}}, Vg_{a_{y_j}}, Vg_{a_{z_j}}) \quad (3.13)$$

where the average ground speed components along the path ds_i are represented by,

$$Vg_{a_{x_j}} = Vg_{a_j}\cos(\gamma_{g_j})\cos(\theta_{g_j}) \quad (3.14a)$$
$$Vg_{a_{y_j}} = Vg_{a_j}\cos(\gamma_{g_j})\sin(\theta_{g_j}) \quad (3.14b)$$
$$Vg_{a_{z_j}} = Vg_{a_j}\sin(\gamma_{g_j}) \quad (3.14c)$$

and $\vec{V}g_{a_j}$ is given by:

$$\vec{V}g_{a_j} = \vec{V}_{a_j} + \vec{V}w_{a_j} \leftrightarrow \begin{cases} Vg_{ax_j} & = V_{ax_j} + Vw_{ax_j} \\ Vg_{ay_j} & = V_{ay_j} + Vw_{ay_j} \\ Vg_{az_j} & = V_{az_j} + Vw_{az_j} \end{cases} \quad (3.15)$$

where V_{ax_j},V_{ay_j} and V_{az_j} are defined in Equations 3.16 - 3.18.

$$V_{ax_j} = Vg_{ax_j} - Vw_{ax_j} = Vg_{a_j}\cos(\gamma_{g_j})\cos(\theta_{g_j}) - Vw_{ax_j} \quad (3.16)$$

$$V_{ay_j} = Vg_{ay_j} - Vw_{ay_j} = Vg_{a_j}\cos(\gamma_{g_j})\sin(\theta_{g_j}) - Vw_{ay_j} \quad (3.17)$$

$$V_{az_j} = Vg_{az_j} - Vw_{az_j} = Vg_{a_j}\sin(\gamma_{g_j}) - Vw_{az_j} \quad (3.18)$$

Adding the square of Equations 3.16, 3.17 and 3.18 it is obtained

$$[Vg_{a_j}\cos(\gamma_{g_j})\cos(\theta_{g_j}) - Vw_{ax_j}]^2 + [Vg_{a_j}\cos(\gamma_{g_j})\sin(\theta_{g_j}) - Vw_{ay_j}]^2 + [Vg_{a_j}\sin(\gamma_{g_j}) - Vw_{az_j}]^2 = V_{a\,j}^2$$

so

$$Vg_{a_j}^2 - 2[\cos(\gamma_{g_j})\left(\cos(\theta_{g_j})Vw_{ax_j} + \sin(\theta_{g_j})Vw_{ay_j}\right) + \sin(\gamma_{g_j})Vw_{az_j}]Vg_{a_j} + [Vw_{ax_j}^2 + Vw_{ay_j}^2 + Vw_{az_j}^2 - V_{a\,j}^2] = 0$$

and can be solved by

$$aVg_{a_j}^2 + bVg_{a_j} + c = 0$$

$$Vg_{a_j} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \quad (3.19)$$

where a, b and c are given by

$$a = 1$$
$$b = Vw_{ax_j}^2 + Vw_{ay_j}^2 + Vw_{az_j}^2 - V_{a\,j}^2$$
$$c = Vw_{ax_j}^2 + Vw_{ay_j}^2 + Vw_{az_j}^2 - V_{a\,j}^2$$

At this phase the time in each segment can be calculated by,

$$dt_j = \frac{ds_j}{Vg_{a_j}} \quad (3.20)$$

and the total time can be expressed as:

$$t_{total} = \sum_{j=1}^{n-1} dt_j \quad (3.21)$$

Since the time of the mission is known, the analysis continues to calculate the missions energy. First, it has to be calculated the air path angle, γ_{a_j},

$$\gamma_{a_j} = \arcsin(\frac{V_{a_{z_j}}}{V_{a_j}}) \quad (3.22)$$

where V_{az_j} also represents the rate of climb in each segment RC_j.

For the calculation of aerodynamic forces, since in Equations 3.25 and 3.26 the speed variable is squared, the squared average velocity is represented in Equation 3.23.

$$Vave_j = \sqrt{\frac{V_j^2 + V_{j+1}^2}{2}} \quad (3.23)$$

Therefore the aerodynamic forces and their coefficients (lift, L_j, drag, D_j, lift coefficient, C_{L_j} and drag coefficient, C_{D_j}) are:

$$L_j = \frac{W_i \cos(\gamma_{a_j})}{cos(\phi_{b_j})} \quad (3.24)$$

$$C_{L_j} = \frac{L_j}{\frac{1}{2}\rho Vave_j^2} \quad (3.25)$$

$$C_{D_j} = f(C_{L_j}) \quad (3.26)$$

$$D_j = \frac{1}{2}\rho Vave_j^2 S C_{D_j} \quad (3.27)$$

where W_j is the aircraft weight, ϕ_{b_j} is the bank angle, S is the wing area and ρ is the air density. More over the drag coefficient is calculated as a function of the lift coefficient multiplied by the values of the drag polar vector presented in Table 4.2. The next step of the analysis is to calculate the average acceleration, a_j, the average inertial force, Q_{IF_j} and the average rolling friction force, F_j. Those can be calculated by

$$a_j = \frac{V g_{a_{j+1}}}{dt_j} \quad (3.28)$$

$$Q_{IF_j} = \frac{W_j}{g} a_j \quad (3.29)$$

$$F_j = \mu_j (W_j - L_j) \quad (3.30)$$

where μ_j is the coefficient of ground rolling friction. Then the average required thrust is given by

$$T_j = D_j + W_j \sin(\gamma_{a_j}) + Q_{IF_j} + F_j \quad (3.31)$$

and finally the average required power P_{req_j}, the average electric power, P_{e_j} and the consumed energy at each segment are calculated by

$$P_{req_j} = T_j V_{aj} \quad (3.32)$$
$$P_{ele_j} = U_j I_j \quad (3.33)$$
$$dE_j = P_{e_j} dt_i \quad (3.34)$$

where U_j and I_j are the input voltage and the motor current, respectively, and can be calculated by an iteration process described in Section 3.4. Note that the energy here is the required energy for each segment. The energy system distribution between the photovoltaic panels (PV panels) and the battery is explained in Section 3.7.

3.2 Motor Performance Model

Electric motors, transform electrical power into mechanical torque and rotational speed. For this work, it is very important to understand how an electric motor works, since the motor is one of the major limitations of the aircraft.

The motor operating properties are defined by some independent parameters, usually specified by the manufacturer, namely: the electric current, I; the no load current, I_0, that represents the electric current when there is no load attached to the motor; the electric resistance, R; And the induced or effective voltage, U_{eff} [25].

Therefore, based on the model presented in [26], the induced voltage can be calculated as

$$U_{eff} = U - RI \quad (3.35)$$

where U is the actual input voltage. Multiplying the induced voltage by the motor velocity constant, results in the motor speed,

$$\Omega = Kv\, U_{eff} \quad (3.36)$$

thus, the induced voltage can also be represented by

$$U_{eff} = \frac{\Omega}{Kv}. \quad (3.37)$$

U is the overall voltage without considering losses due to resistance, and can be represented as a function of motor speed, Ω, and input current, I, according to

$$U(\Omega, I) = U_{eff}\Omega + I\, R = \frac{\Omega}{Kv} + I\, R \quad (3.38)$$

Likewise that there are some voltage losses, the same happens with the total input current. Hence, the effective current, I_{eff} is given by

$$I_{eff} = I - I_0 \quad (3.39)$$

where the no load current, I_0, does not contribute to useful torque. So the input current, I, can be represented as

$$I = \frac{1}{R}\left(U - \frac{\Omega}{Kv}\right) \quad (3.40)$$

With the input current and voltage values known, it is possible to calculate the total electric power consumed, P_{ele}, as

$$P_{ele} = U\,I \quad (3.41)$$

Following the same logic and considering I_{eff} and U_{eff}, the effective power remaining after the power losses, P_{eff}, is given by

$$P_{eff} = U_{eff}\,I_{eff} \quad (3.42)$$

The available torque at the shaft, Q_m, can be represented by Equation 3.43 based on the power and torque relation.

$$Q_m = \frac{P_{eff}}{\Omega} \quad (3.43)$$

Lastly, the motor efficiency, η_{motor}, is obtained by the ratio of the effective power and the overall power converted.

$$\eta_{motor} = \frac{P_{eff}}{P_{ele}} = \frac{I_{eff}U_{eff}}{I\,U} \quad (3.44)$$

3.3 Propeller Performance Model

For the correct simulation of the propulsion system, it is necessary to consider a propeller performance model, as complement to the electric motor performance model. For this purpose, the propeller performance model presented in reference [26] was implemented.

To complement the electric motor performance model, it is necessary to calculate the propeller efficiency, η_{prop}, and the power coefficient, C_p. These depend on the propeller advance ratio, J_{prop}, which is a non-dimensional parameter and is defined by

$$J_{prop} = \frac{60.V}{Nd} \quad (3.45)$$

where V is the linear velocity of the propeller, with respect to the flow field, N is the propeller speed in revolutions per minute, and d is the propeller diameter. Notice that $N/60$ represents the propeller speed in revolutions per second rps.

By knowing how the power coefficient C_p varies with the advance ratio J_{prop}, it is possible to calculate the propeller shaft power P_{shaft}, as:

$$P_{shaft} = C_p\rho\left(\frac{N}{60}\right)^3 d^5 \quad (3.46)$$

For the calculation of the power coefficient, C_p, and the propulsive efficiency, η_p, as functions of the propeller advance ratio, J_{prop}, for a single propeller with a fixed diameter, d, and pitch, p, a polynomial approximation was used presented in reference [26]. This polynomial approximation is defined by

$$Cp = C_{p_0} \times \left[A_0 + \sum_{i=1}^{4} \left(A_i \left(\frac{J_{prop}}{J_{max}} \right)^i \right) \right] \quad (3.47)$$

$$\eta_p = \eta_{p_{max}} \times \left[B_0 + \sum_{i=1}^{6} \left(B_i \left(\frac{J_{prop}}{J_{max}} \right)^i \right) \right] \quad (3.48)$$

where C_{p_0} is the power coefficient at a null advance ratio, $\eta_{p_{max}}$ is the maximum propeller efficiency, J_{max} is the maximum advance ratio, $\overline{A}$ and $\overline{B}$ are the coefficient vectors obtained through the polynomial approximation. The coefficients used in this work are:

$$\overline{A} = \begin{bmatrix} 0.9999747473830 \\ 0.0026886303943 \\ -0.0542821394531 \\ -0.8141198610786 \\ 0.2382888347204 \\ -0.1060271581734 \\ 0.0222789611099 \end{bmatrix} \quad (3.49)$$

$$\overline{B} = \begin{bmatrix} 0.0000000000000 \\ 2.8358158896651 \\ -4.6740787983266 \\ 17.2094772778345 \\ -45.734194221401 \\ 55.789219497612 \\ -25.395785093511 \end{bmatrix} \quad (3.50)$$

J_{max}, $\eta_{p_{max}}$ and C_{p_0} are obtained by

$$J_{max} = C_1 d + C_2 p + C_3 d^2 + C_4 dp + C_5 p^2 + C_6 d^3 + C_7 d^2 p + C_8 dp^2 + C_9 p^3 \quad (3.51)$$

$$C_{p_0} = D_1 d + D_2 p + D_3 D^2 + D_4 dp + D_5 p^2 + D_6 d^3 + D_7 d^2 p + D_8 dp^2 + D_9 p^3 \quad (3.52)$$

$$\eta_{p_{max}} = E_1 d + E_2 p + E_3 d^2 + E_4 dp + E_5 p^2 + E_6 d^3 + E_7 d^2 p + E_8 dp^2 + E_9 p^3 \quad (3.53)$$

where the coefficient vectors $\overline{C}$, $\overline{D}$ and $\overline{E}$ are

$$\overline{C} = \begin{bmatrix} 0.706462000000 \\ -0.046405100000 \\ 0.074350100000 \\ 0.001069860000 \\ -0.001664110000 \\ -0.000007715000 \\ -0.000006521000 \\ 0.000008688670 \\ 0.000002563530 \\ -0.000000703183 \end{bmatrix} \quad (3.54)$$

$$\overline{D} = \begin{bmatrix} 0.050916200000 \\ -0.005511640000 \\ 0.007489280000 \\ 0.000144156000 \\ -0.000239091000 \\ 0.000065509200 \\ -0.000002407300 \\ 0.000005544700 \\ -0.000003824100 \\ 0.000000875200 \end{bmatrix} \quad (3.55)$$

$$\overline{E} = \begin{bmatrix} 0.375474000000 \\ 0.013321100000 \\ 0.014884800000 \\ -0.000358479000 \\ 0.000020627100 \\ -0.000189967000 \\ 0.000003644830 \\ -0.000004047110 \\ 0.000004028760 \\ -0.000000467311 \end{bmatrix} \quad (3.56)$$

Note that the coefficients were obtained from the model presented in [26], and refers to the propeller of the UAV used to verify the mission planner software.

3.4 Mission's Power and Energy Model

Based on Sections 3.2 and 3.3, this Section describes the model used to calculate the parameters needed for the analysis and optimization of the mission. Those parameters are the motor power setting, δ_{set}, the motor speed, Ω, and the input voltage, U.

For the propeller's and motor's proper matching, the power absorbed by the propeller must

equal the motor shaft power. Given an assumed δ, this subroutine allows the calculation of the input voltage, the motor speed and the motor shaft power, P_{shaft}, by adjusting the motor current, I, through an iterative process, in order to match the condition of the propeller-motor matching. For a minimum electric power required, the propeller must assume a value that equals the required power calculated by Equation 3.32. By these means, the motor setting is corrected and adjusted through an iterative process, to match the condition of a minimum electric power required, where the propeller must assume a value as minimum as possible, consequently equalling the required power for levelled flight.

This subroutine is called after the calculation of the segment required power, P_{req_j}, Equation 3.32. When this subroutine finishes, the mission analysis process can proceed to the calculation of P_{ele_j} and E_j. Equations 3.33 and 3.34, based on this subroutine it can be written as

$$P_{ele_j} = \left(UI + (R_{ESC}I^2 + R_U I^2)\right)_j n_{motor} \quad (3.57)$$

(3.58)

where R_U, is the battery resistance and R_{ESC}, is the motor speed controller resistance. These must be considered due to the energy losses in the consumed electric power.

Note that the energy used by the aircraft's electrical systems must be added to Equation 3.57 to calculate the total electric power in each segment.

$$P_{T_j} = P_{ele_j} + P_{sys_j} \quad (3.59)$$

$$dE_j = P_{T_j} dt \quad (3.60)$$

$$E = \sum_{1}^{n} (dE_j) \quad (3.61)$$

Where n is the number of segments.

3.5 Ground Elevation Model

One of the most important features of a mission planner is to be able to detect an obstacle and be able to make the necessary corrections to avoid the obstacle. Thus, the ground elevation model is responsible for providing the necessary ground elevation data to the analysis routine. With this purpose, a free and open-source Application Programming Interface (API) database was choosen to map the ground elevation. All public API instructions documentation is available in [27].

The latitude and longitude of a point on earth (input data) is sent to the API as a request and consequently, it will return the elevation at this point (output). The algorithm that achieves this is described bellow. It starts by defining the boundaries, the user needs to define two points that represent the initial and ending point of the diagonal of the desired map.

$$Q_{11}(Q_{lat_1}, Q_{lon_1}) \quad (3.62)$$

$$Q_{mm}(Q_{lat_m}, Q_{lon_m}) \quad (3.63)$$

where Q_{11} and Q_m are the initial and the ending point of one of the map's diagonals. Notice that,

$$\{\varphi_1, \varphi_2, \ldots, \varphi_n\} \subset]\, Q_{lat_1}, Q_{lat_m}\,[\quad (3.64)$$

$$\{\lambda_1, \lambda_2, \ldots, \lambda_n\} \subset]\, Q_{lon_1}, Q_{lon_m}\,[\quad (3.65)$$

where $\varphi_1, \varphi_2, \ldots, \varphi_m$ and $\lambda_1, \lambda_2, \ldots, \lambda_m$ are the latitudes and longitudes of the mission's waypoints, respectively, n is the number of waypoints and assuming the map as a matrix $m \times m$, so m is the number of points requested by column of the map, and so representing the resolution of the map. Once n is defined it is created a 2D matrix, where Q_{lat_1} and Q_{lat_n} define the $North \rightarrow south$ boundaries as well as Q_{lon_1} and Q_{lon_n} are the $West \rightarrow East$ boundaries of the map. Then n^2 number of points will be sent to the database as a request for each elevation, resulting in a grid with n^2 three-dimensional points $Q_{Elevation}(Q_{lat_i}, Q_{lon_i}, Q_{ele_i})$. The data is saved as a list of the points in three columns, each line representing a three dimensional point $Q_{Elevation}$. Finally, the file is saved as earth elevation in a .txt format. Now that the elevation data is known and saved, the main program has access to this through a subroutine, where the data is read from the earth elevation file and saved in a three dimensional vector. With all the necessary data from elevation API saved in the program, the process that achieves the elevation of a pair of coordinates is simple, firstly the vector is searched to find the four nearest points to the input coordinates $P = (x, y)$. This is represented graphically in Figure 3.4.

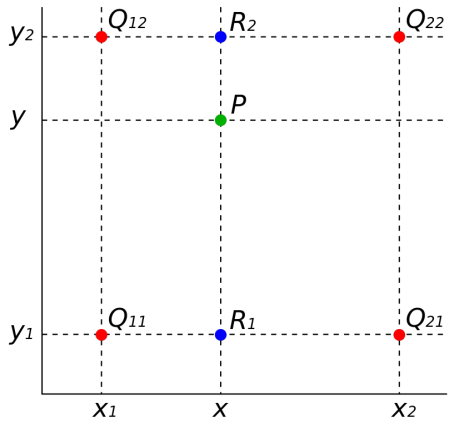


Figure 3.4: The four red dots show the data points and the green dot is the point at which we want to interpolate.

Notice in Figure 3.4 and in the following description of the interpolation's algorithm x and y represent the latitude and longitude of the ground points, respectively. Secondly having four points, $Q_{11} = (x_1, y_1)$, $Q_{12} = (x_2, y_1)$, $Q_{21} = (x_1, y_1)$ and $Q_{22} = (x_1, y_1)$, and knowing the elevation value correspond to each of the four points Q_{ii}, $f(P(x, y))$ can be interpolated by first doing a linear interpolation in the x-direction. This is

$$f(x, y_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \quad (3.66)$$

$$f(x, y_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \quad (3.67)$$

and then proceeding with the interpolation in the y-direction to obtain the desired estimate:

$$\begin{aligned} f(x, y) &\approx \frac{y_2 - y}{y_2 - y_1} f(x, y_1) + \frac{y - y_1}{y_2 - y_1} f(x, y_2) \\ &= \frac{y_2 - y}{y_2 - y_1} \left(\frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \right) + \frac{y - y_1}{y_2 - y_1} \left(\frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22} \right) \\ &= \frac{1}{(x_2 - x_1)(y_2 - y_1)} (f(Q_{11})(x_2 - x) + f(Q_{21})(x - x_1)(y_2 - y) + f(Q_{12})(x_2 - x)(y - y_1) \\ &+ f(Q_{22})(x - x_1)(y - y_1)) \\ &= \frac{1}{(x_2 - x_1)(y_2 - y_1)} \begin{bmatrix} x_2 - x & x - x_1 \end{bmatrix} \begin{bmatrix} f(Q_{11}) & f(Q_{12}) \\ f(Q_{21}) & f(Q_{22}) \end{bmatrix} \begin{bmatrix} y_2 - y \\ y - y_1 \end{bmatrix}. \end{aligned} \quad (3.68)$$

Note that if the interpolation had started through the y-direction and then through the x-direction we would have reached the same result. Once the interpolation is done the subroutine returns the interpolated elevation to the main program.

3.6 Atmospheric Data Model

To make the mission planner viable the atmospheric data model is used to provide the mission's atmospheric data (wind speed, wind direction, temperature deviation and the cloud cover). To achieve that, we choose the OpenWeatherMap API [28], which returns a five days forecast by geographic coordinates. This model is very similar to the ground elevation model, the method to request the data from the database is the same with a particularity: the database instead of returning three-dimensional points $Q_{Elevation}(Q_{lat_i}, Q_{lon_i}, Q_{ele_i})$, it returns six-dimensional points $Q_{Weather}(Q_{lat_i}, Q_{lon_i}, Q_{windspeed_i}, Q_{winddirection_i}, Q_{Temp_i}, Q_{clouds_i})$. At this point, the data is read through a subroutine responsible for the weather data and each variable of atmospheric data is interpolated for the mission's analysis by the same method of 2D interpolation described on the ground elevation model Section. It is important to note that the database does not have data of vertical wind or the variation of the weather parameters through the altitude. So, this work considers the weather parameters constant in altitude, except for temperature, that is estimated calculating the temperature deviation referent to the International Standard Atmosphere (ISA) model.

3.7 Solar Model

The solar energy that the PV panel can harvest depends on its area, S_{PV}, on the solar irradiation J (or solar power per unit area) reaching it, and on its efficiency, η_{PV}. The atmosphere is a strong influence on it, as well as the percentage of clouds. The Irradiation varies with the location on earth (mostly with latitude), with the orientation of solar panels and with the solar zenith angle (defined by the angle between the direction of the Sun's centre and the local zenith). The zenith angle vary with the day of the year and hour of the day.

3.7.1 Estimated Solar Irradiance

Based on Ref. [29], the Solar Power per unit area can be estimated by

$$J = J_{0n}\tau \sin(\zeta), \quad (3.69)$$

where τ is the transmittance factor, which is the ratio of the total radiant or luminous flux transmitted by a transparent object to the incident flux, ζ is the zenith angle and J_{0n} is the intensity of the extraterrestrial normal solar radiation which is given by

$$J_{0n} = J_{SC}\left(\frac{r_{ES,0}}{r_{ES}}\right)^2, \quad (3.70)$$

where J_{SC} is the extraterrestrial normal solar radiation constant, $r_{ES,0}$ and r_{ES} are the mean and real distances between the Earth and the Sun respectively. r_{ES} is a function of $r_{ES,0}$, the eccentricity of Earth's orbit, ε, and the true anomaly, v, which represents an angular parameter that defines the position of a body moving along a Keplerian orbit. So, the real distances between the Earth and the Sun is expressed as

$$r_{ES} = r_{ES,0}\left(\frac{1-\varepsilon^2}{1+\varepsilon\cos(v)}\right), \quad (3.71)$$

and the true anomaly, v, can be calculated by

$$v = 2\pi\frac{d_n - 4}{365} \quad (3.72)$$

where d_n is the day of the year, counting from the first of January (day 1). With J_{0n} Known it is necessary to calculate the zenith angle, which is given by

$$\zeta = \frac{\pi}{2} - \arccos(\sin(\varphi)\sin(\delta_s) + \cos(\varphi)\cos(\delta_s)\cos(\mu_H)), \quad (3.73)$$

where φ is the latitude of the location and δ_s and μ_H are the solar declination angle and the hour angle respectively, which are calculated by

$$\delta_s = \frac{23.45\pi}{180}\sin\left(360\frac{284+d_n}{365}\right), \quad (3.74)$$

and

$$\mu(H) = \pi - \pi\frac{H}{12} \quad (3.75)$$

where H is the hour of the day. The constants used in this model are listed in Table 3.1.

Table 3.1: The constant values used in Eqs. above.

Symbol	Value/unit
τ	0.85
J_{SC}	1367 W/m^2
$r_{ES,0}$	149,597,896 km
ε	0.0167

It is important to note that before sunrise and after sunset some output values for solar irradiance may be negative, so in this case they will be set to zero.

3.7.2 Cloud Cover Effect

As written in the previous Subsection, the energy that a PV panel can harvest is influenced by the clouds. Clouds are one of the largest attenuating factors of solar irradiance. Cloud cover is a useful predictor of solar resource. If the sky is cloudless, irradiance can be predicted from the solar geometry, surface albedo, and optical properties of aerosols, ozone and water vapour using a radiative transfer calculation. Alternatively, several clear-sky models exist in the literature which are empirical relationships between one or more atmospheric variables [30]. The statistics of clear-sky index can be used to determine solar irradiance when the theoretical clear sky irradiance and the cloud cover are known. Based on clear-sky index model on Ref. [30], there is an empirical relationship between the clear-sky index K_c and cloud cover, C_c in (oktas), thus the clear-sky index - cloud cover relationship is,

$$K_C(C_c) = 1 - 0.75(C_c/8)^{3.4} \quad (3.76)$$

The solar irradiance harvested, considering the cloud effect, is given by

$$J_{cloud} = J\,K_C \quad (3.77)$$

3.7.3 Solar Power with Tilt Angles

With the solar irradiance and the cloud cover effect known, aircraft attitude must be considered, because the energy that the PV panels can harvest varies with angle of incidence of the sun, on the panels. Based on models from reference [31], we can find the relation between sun incidence and the PV panels. Knowing the solar declination angle δ_s, the hour angle H and the zenith angle given by the model in Section 3.7.1, it is necessary to calculate the Azimuth angle

Az, that is given by Equation 3.78 from reference [32].

$$Az = \frac{\sin(\omega)\cos(\delta_s)}{\sin(\theta_Z)} \quad (3.78)$$

The scheme used to calculate the sun's Azimuth is illustrated in Figure 3.5.

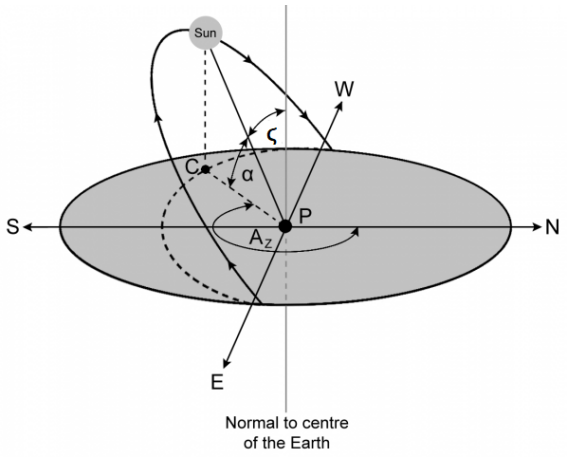


Figure 3.5: Illustration of the sun's position.

Then with the solar azimuth angle known, the angle of incidence θ_i of the Sun on a surface tilted at an angle from the normal to the centre of the Earth, β and with any surface azimuth angle A_{ZS} (Figure 3.6) can be calculated from (when A_{ZS} is measured clockwise from north):

$$\cos(\theta_i) = \cos(90 - \zeta)cos(Az - Azs)\sin(\beta) - \cos(\zeta)\cos(\beta)$$

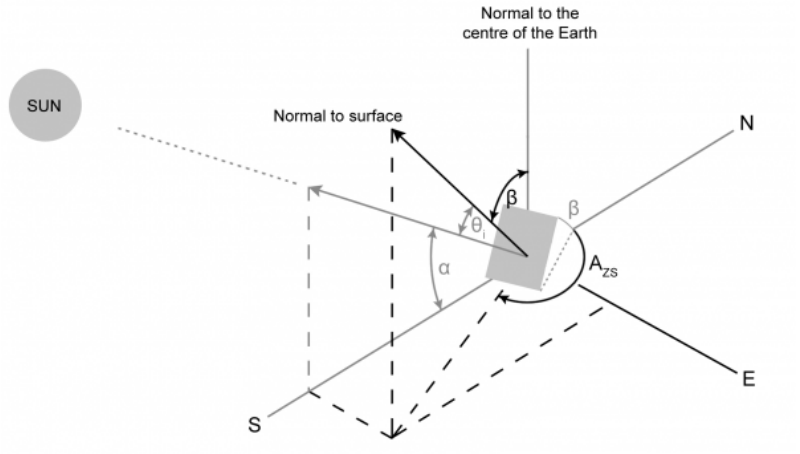


Figure 3.6: Illustration of the problem.

Then the solar irradiance on a tilted surface is given by

$$J_{TS} = J\cos(\theta_i) \quad (3.79)$$

Note that if θ_i > 90° at any point the Sun is behind the surface and the surface will be shading

itself. Finally, the total energy that the PV panels can harvest is given by

$$E_{solar|total} = \int_{t}^{t_f} J\, K_C\, S_{PV}\, \eta_{PV}\, \cos(\theta_i)\, dt. \quad (3.80)$$

3.7.4 Management of Electric Power Flow

The UAV used in this thesis has a parallel systems installation. The systems power distribution is described in Figure 3.7.

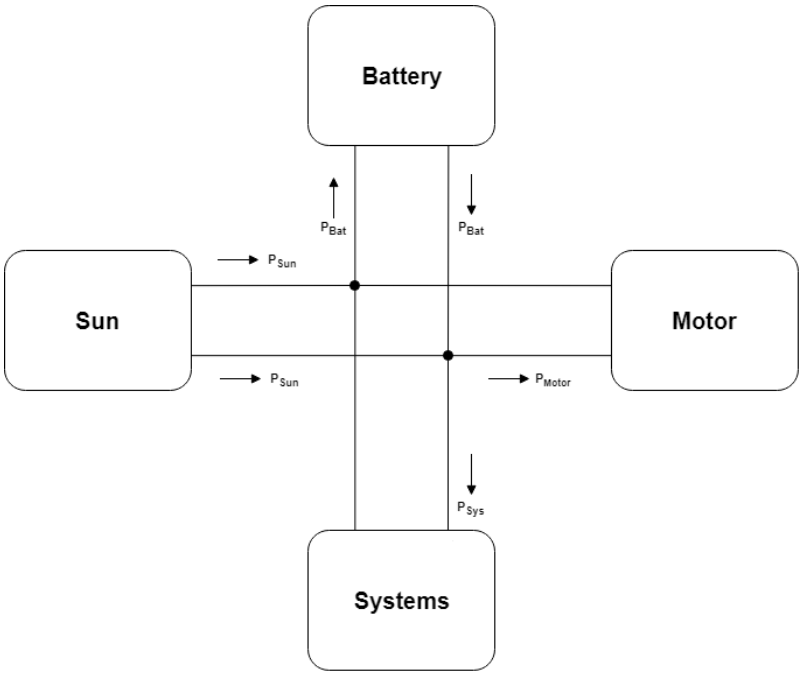


Figure 3.7: Systems Power distribution.

In this type of installation, the battery power is only used to compensate the lack of energy harvested by the PV panels. In the case of the total required power (motor required power plus the systems required power) being greater than the solar power, the used battery power is equal to

$$P_{bat} = P_{motor} + P_{system} - P_{solar} \quad (3.81)$$

on the other hand

$$P_{solar} = P_{motor} + P_{system} - P_{bat} \quad (3.82)$$

This energy management is assured by a subroutine called SystemsElectricManagement(). The subroutines algorithm is described in the following pseudocode.

Alghorithm for Management of Electric Systems

```
1: PV_panels_area              = 0.6875d0
2: PV_panels_efficiency        = 0.237d0
3: Charger_efficiency          = 0.99d0
4: Solar_power                 = Solar_irradiation * PV_panels_area * PV_panels_efficiency *
Charger_efficiency
5: if Solar_power is greater than or equal to electric_loads_power_consumption then
6:       Consumed_solar_power      = electric_loads_power_consumption
7:       if Solar_power is greater than Consumed_solar_power then
8:          battery_consumption    = maximum value between -battery_U_max * battery_C  or
                                                  Solar_power_consumed - Solar_power
9:       end if
10: else if Solar_power is lower than electric_loads_power_consumption then
11:       Consumed_solar_power    = Solar_power
12:       battery_consumption     = electric_loads_power_consumption - Consumed_solar_power
13: end if
```

Figure 3.8: Pseudocode for the management of the electrical Systems.

Notice that, when the solar power is greater than the required power, the battery consumption gets a negative value, this means that the battery is being charged.

3.8 Mission Optimization

The calculation of a optimal mission for aerial vehicles is the main objective of this dissertation. The mission planner was created based on FORTRAN Feasible Sequential Quadratic Programming (FFSQP) optimization algorithm. With the aircraft data and the design parameters stated in Equation 3.1 known, the program will be able to calculate the objective function and then proceed with its optimization. The three available options are the mission time, the mission energy and the mission distance. So, minimizing one of the three, is the objective function, which directly depends of the design variables and the mission constraints set at the beginning. Once the design parameters and the mission constraints are defined, the program proceeds with the analysis of a pre-defined mission, described in Section 3.1 to calculate the value of the objective function. After that, it is all set to the application of the optimization software. Figure 3.9 describes this process.

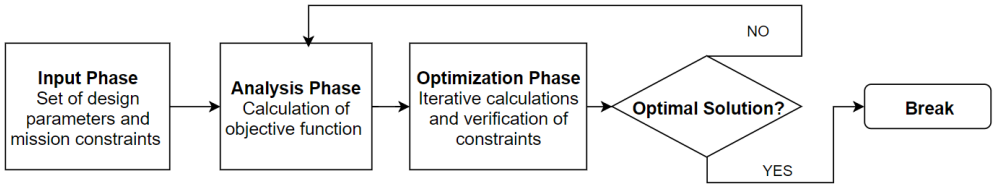


Figure 3.9: Iterative process flowchart.

3.8.1 FFSQP Subroutines

For the optimization stage, a set of FORTRAN subroutines are used for the minimization of the objective functions, subject to general smooth constraints (if there is no objective function, the goal is to simply find a point satisfying the constraints) [33]. These set of constraints may be nonlinear or linear equality and inequality constraints, where the design variables are limited by these boundaries. FFSQP applies the Sequential Quadratic Programming methodology for

nonlinear optimization problems, which solves optimization problems in the form [33]

$$min/max\{f_i(x_n)\}, x \in X \, and \, i \in I \quad (3.83)$$

where $f_i(x_n)$ represents the objective function at each iteration i, X is the whole feasible space, where a set of design parameters, x, are constrained by a lower boundary and upper boundary according to a set of inequality constraints defined as

$$g_j(x_n) \leq g_{ref}(x_n), \; j = 1, 2, ..., m \quad (3.84)$$

$$g_j(x_n) - g_{ref}(x_n) \leq 0 \quad (3.85)$$

where $g_{ref}(x_n)$ is the reference constraint value which is set by the user and m is the total number of constraint functions [34]-[33].

In this optimization, the SQP uses the forward finite differences for estimating the gradients of the constraint function. For better understanding this is presented by a simplified scheme in Figure 3.10, Δx is the algorithm increment set by the user.

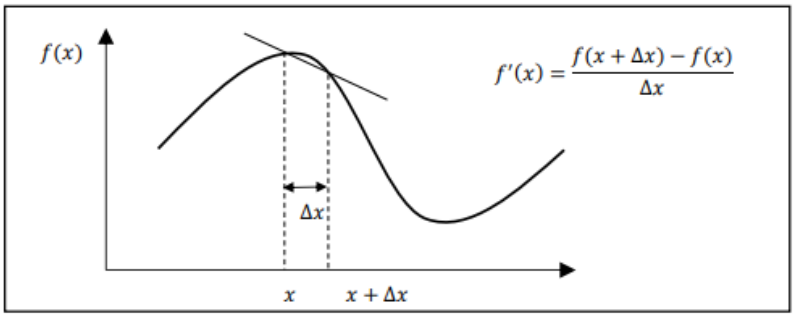


Figure 3.10: Simplified scheme of the forward finite differences method used to estimate the gradient at point x [34].

The algorithm, at each iteration, calculates the gradients of the functions with the objective to find the step direction, p_i. Then, in order to achieve the right direction inner product between p_i and the functions gradients, $\nabla f(x_i)$, has to be negative to minimize the objective function. This is due to the negative of a non-zero gradient is always the descent direction [34] [33]. This can be represented as

$$[p_i, \nabla f(x_i)] < 0 \quad (3.86)$$

Knowing what the correct step direction to take, the algorithms calculates the step length (α_i). The validation of this step is given by

$$f(x_i + p_i \alpha_i) < f(x_i) \quad (3.87)$$

The algorithm increment, Δ_x, is a fixed step used to calculate the gradients, rather than the step length, α_i, that is a variable step used to advance to x_{i+1} where which $f(x_{i+1})$ is minimized.

If the condition in Equation 3.87 is satisfied, then x_{i+1} value is updated to

$$x_{i+1} = x_i + p_i \alpha_i, \quad (3.88)$$

The next iteration of the optimization process begins, the algorithm calculates the forward finite differences of the design variable, obtaining its gradients in order to update it to a new design variable. When the Hessian matrix of the objective function, $\nabla f(x_i)$, converges to zero or to a tolerance value, defined as ε, the final solutions is obtained [34]. Figure 3.11 illustrates the overall SQP procedure.

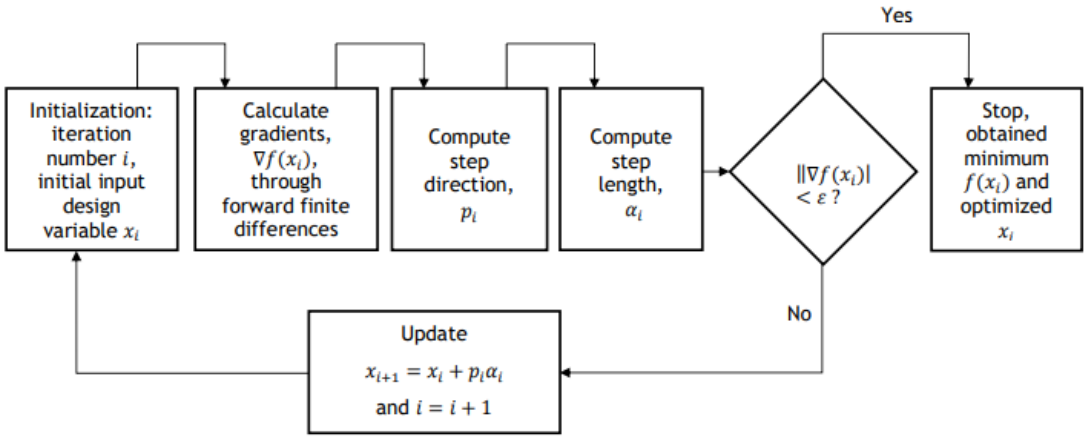


Figure 3.11: Sequential Quadratic Programming (SQP) optimization procedure [34].

Note that in this dissertation, assuming a mission with a specified number of segments (for example a takeoff segment, a climbing segment, a cruise segment, a descending segment and a landing segment). The method to calculate this objective function given the design parameters is described through Equations3.3 to 3.21. The constraints boundaries are defined by the user at the beginning of the analysis. The values and the conditions of the constraints considered in in this work will be explain in the next Subsection.

3.8.2 Objective Functions

The objective of a this mission planning problem is to maximize or to minimize some numerical value. This numerical value is called objective function and it attempts to maximize profits or minimize losses based on a set of constraints and the relationship between one or more design variables. Regarding the present work, there are three options for the objective function. They include, the mission time, mission energy and mission distance. The user is responsible for choosing the most appropriate objective function for its mission planning.

3.8.3 Constraint Functions

The set of parameters that satisfy the equality and inequality constraints is called the feasible set of the nonlinear optimization problem. Its important to note that, for the mission planner program, each iteration solution is the same as a mission's simulation, so the constraint functions to considered in each iteration were chosen so that the simulation are as closest to the real as possible. In this work, there were not considered any equality constraints, yet as inequality constraints it has the design setting condition, which limits the motor setting to its maximum

value or to a value defined by the user, this is given by

$$\delta_{set_j} \leq \delta_{ref},\ \delta_{ref} \in]0,1] \quad (3.89)$$

where δ_{ref} reference value of the motor setting constraint function, the maximum motor current condition that is defined as

$$I_j \leq I_{max} \quad (3.90)$$

where I_{max} is the maximum current of the motor, the stall speed condition, by the means to prevent the stall effect, which is represented as

$$C_{L_j} \left(\frac{V_{a_j}}{V_{stall}} \right)^2 \leq C_{L_{max}} \quad (3.91)$$

where V_{safety} is the safety speed factor, and V_{stall} is de stall speed; the height condition, which limits the minimum altitude to a reference value above the ground, this specific constraint is verified in each waypoint and also in a extra number of points generated automatically by the program included in each segment, this is given by

$$h_p \geq h_{ref}, h_{ref} > 0 \quad (3.92)$$

where p index refers to the waypoints in flight, not including the take off or landing segment and h_{ref} is the the reference value of the minimum height constraint function; the minimum required power constraint, that due to the impossibility of an airplane to break, the mission planner software has to consider only positive values of required power. It is given by

$$P_{req_j} < 0 \quad (3.93)$$

and finally the battery energy left condition, with the purpose of preventing the battery energy to be lower than a reference value, B_{ref}, defined by the user,

$$E_{left} \geq B_{ref} \quad (3.94)$$

where E_{left} is the energy remaining in the battery.

The constraints reference values used to test the program are defined in Chapter 4.

Chapter 4

Practical Application and Results

The development of this work was an evolutionary process with respect to the addition of the various models necessary for the analysis. This chapter: first, describes the input data of the LEEUAV (Long endurance electric unmanned aerial vehicle) which is used to test the program; second, it presents the constraints applied to the tests and then, describes the evolution of the mission planner presenting and analysing several tests that have been done throughout the development of the program. These tests are divided by cases of study. The first case is discussed in Section 4.2. At this phase, the program was not complete and two tests were performed, in order to verify the well functioning of the mission planer. The next cases of study are presented in Section 4.3. The first case described in Section 4.3 has the objective of verifying if the mission planner is able to avoid trajectories crossing terrain. The second one, has the purpose of analysing the solar model behaviour, by presenting and comparing tests with uniform cloud cover through the flight. Then, another case, is presented with the goal of understanding how the optimization algorithm behaves, regarding the position of the waypoints, when there is a variation of the cloud cover through the map, followed by a case of study that presents tests with real weather forecast data in order to obtain results closer to those of a realistic mission. For last, a test is presented to show the full capacity of the mission planner. To better understand this chapter, a code that serves to name each test is described in Table 4.1, also waypoint 1 will be identified in each trajectory figure. Note that the take-off segment in every figure is much shorter than the rest of the segments, so, waypoint 1 and 2 may look coincident.

Table 4.1: Code to name the tests.

Code	Section Name	Number of tests
pre	Preliminary tests	2
TA	Terrain Avoidance tests	3
SM	Solar Model Behaviour	4
NU	Non Uniform Cloud Cover	2
RW	Real Weather Forecast Data	4
FC	Mission Across Portugal	1

4.1 LEEUAV Input Data

The LEEUAV, as its name suggests, is a long endurance electric UAV and is solar powered by a PV panel installed in its wings.the LEEUAV, motor, battery and propeller specifications for the Mission Planner are presented in the Tables 4.2, 4.3 and 4.4.

Table 4.2: LEEUAV Input Specifications

LEEUAV					
Mass	**Total Mass [kg]**	**Systems Mass [kg]**	**PV Systems Mass [kg]**	**Structure Mass [kg]**	**Payload Mass [kg]**
	5.506	0.459	0.801	2.624	0.200
Areas	**Wing Area [m^2]**	**Fuselage Cross Section Area [m^2]**	**Fuselage Wetted Area [m^2]**		
	1.485	0.0144	0.2		
Lift Coeficients	$C_{L_{max}}$	$C_{L_{TO}}$			
	1.5	0.8			
Drag Coeficient	C_{D_1}	C_{D_2}	C_{D_3}	C_{D_4}	C_{D_5}
	0.0575979	-0.133823 C_L	0.2420812C_L^2	-0.151927C_L^3	0.4183674C_L^4

Table 4.3: LEEUAV Motor Specifications

LEEUAV Motor - Scorpion SII-4025-520KV						
M [g]	Kv [rpm/V]	I_{max} [A]	R [Ω]	I_0[A]	U_{max} [V]	R_{ESC} [Ω]
353	520	100	0.09	1.40	25.20	0.05

Table 4.4: LEEUAV Battery Specifications

LEEUAV battery - SLS APL 3S1P 10000mAh					
U_{cell} [V]	Number of cells	Battery Pack U_{max}[V]	Battery Pack R_U[Ω]	Battery Pack I_{max}[A]	Battery Pack M[g]
4.20	3	12.60	0.003	150	750

Table 4.5: LEEUAV Propeller Specifications

LEEUAV Propeller			
M [g]	d [in]	p [in]	Number of blades
72	19.09	15.43	2

The data in Tables 4.2 to 4.5 are considered in the following Sections where the results of the tests that were conducted through the development of the mission planner are discussed.

4.2 Preliminary tests

In a preliminary phase of the mission planner code, when the ground elevation model was programmed, tests were conducted to analyse the well functioning of the algorithm itself. These tests were done with common design variables for different flight conditions. The design variables used for the tests presented in this Section are described in Table 4.6.

Table 4.6: Preliminary Tests design variables

Mission Design Variables Input				
Waypoint number	Latitude [deg]	Longitude [deg]	Altitude [m]	Airspeed [m/s]
1	40.29563	-7.43693	0	0
2	40.29546	-7.43750	0	8
3	40.20458	-7.44433	1000	10
4	39.96362	-7.49178	1000	10
5	39.85038	-7.44293	0	8

Note that neither the atmospheric data model nor the solar model were programmed yet, so in each of the following tests the atmospheric data is defined and fixed for each waypoint and the harvested solar energy is considered zero.

4.2.1 Preliminary Test 1

The design variables were chosen in order to have approximately a North-South flight direction with a curve of approximately 5 km, which had the objective of setting more easily the wind direction variable.

Table 4.7: Fixed data for Mission planning - Test pre 1.

Mission Data Input					
Waypoint number	Ground/ Air	Loiter	Windspeed [m/s]	Wind direction[1] [deg]	Temp.Dev.[2] [K]
1	G	N	0.00	0.00	3.337
2	G	N	5.00	0.00	3.348
3	A	N	5.00	0.00	3.366
4	A	N	5.00	0.00	4.581
5	G	N	5.00	0.00	3.009

S - add a loiter segment to that position;
N - do not add a loiter segment to that position;
G - waypoint located in ground; A - waypoint located in air;
1 - Wind direction North-East orientation; 2 - ISA Temperature deviation

For each test, the user has to choose the objective function option, define the design parameters as well as, choose which waypoints the aircraft is at ground, where these waypoints coordinates are set fixed, and the altitude of each waypoint is set equal to the terrain elevation value automatically. Then, if there are any waypoints where it is required to add a loiter step, it has to be defined in the input data as well. For this case, the mission time was chosen as objective function, with the wind in the same direction as the flight. So, as the pre-defined trajectory has approximately a North-South direction, the wind direction and the wind speed were set to 0 deg and to 5 m/s respectively, except for waypoint 1, where the wind speed was set to 0. The temperature deviation was set as a normal value based on real weather forecast data in the day of the test.

The following Figures, 4.1 to 4.3, show the initial trajectory in red and in black is the optimized trajectory.

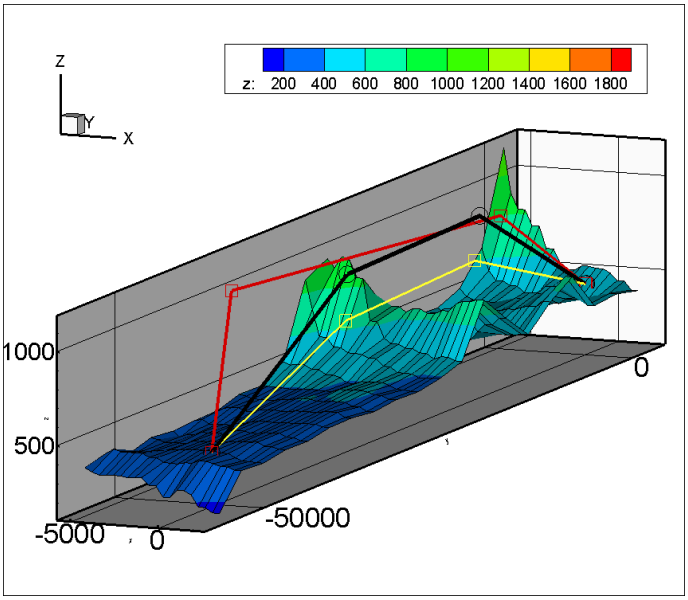


Figure 4.1: 3D View of the output mission -the black trajectory refers to test pre 1 and the yellow trajectory refers to test pre 2.

The flight direction as well as the wind direction are represented from the right to the left (-Y), in Figure 4.2. We can see that, compared to the initial trajectory, the optimized one has a longer climb and descent phases, resulting in a shorter cruise, maintaining approximately the same altitude. This occurred resulting in a trajectory with a shorter distance, in order to obtain a shorter duration of flight, as the wind has the same direction of the flight.

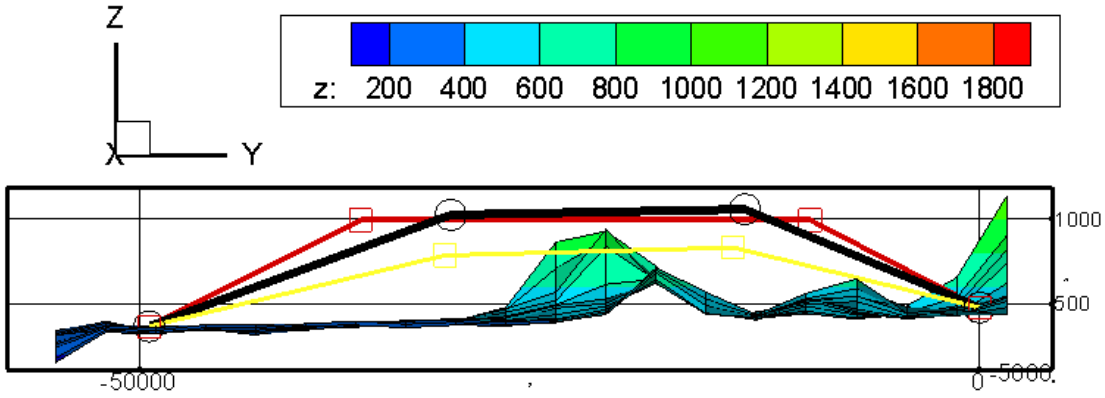


Figure 4.2: YZ View of the output mission - the black trajectory refers to test pre 1 and the yellow trajectory refers to test pre 2.

In Figure 4.3, it is noticed that the red ”curved” trajectory is optimized to a rectilinear black trajectory to the ending point, that confirms the idea of the previous paragraph, and also due to the optimized trajectory having the same direction as the wind it minimizes the lateral forces in the aircraft, as the resulting force applied by the wind in the aircraft has approximately the same direction of the flight.

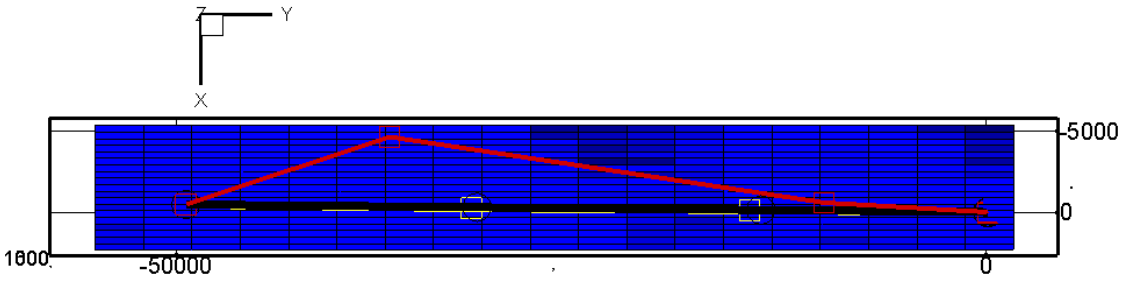


Figure 4.3: XY View of the output mission - the black trajectory refers to test pre 1 and the yellow trajectory refers to test pre 2.

The output design variables are described in Table 4.8. The coordinates values of the waypoints 1, 2 and 5 were fixed because at these points the aircraft is on the ground. The airspeed is only fixed at waypoint 1 as the aircraft initial airspeed is 0 m/s.

Table 4.8: Output Design Variables - test pre 1

Output Design Variables				
Latitude 03 [deg]	**Latitude 04 [deg]**	**Longitude 03 [deg]**	**Longitude 04 [deg]**	**Altitude 03 [m]**
40.1696	40.0119	-7.4390	-7.4409	1063.7
Altitude 04 [m]	**Airspeed 02 [m/s]**	**Airspeed 03 [m/s]**	**Airspeed 04 [m/s]**	**Airspeed 05 [m/s]**
1030.5	13.9	14.0	14.2	14.1

With the results it can be seen that the airspeed was increased in each waypoint. As shown in Figure 4.3, the distance in the optimized trajectory decreased from the initial trajectory, which led to an energy saving that was used to increase the airspeed through the waypoints.

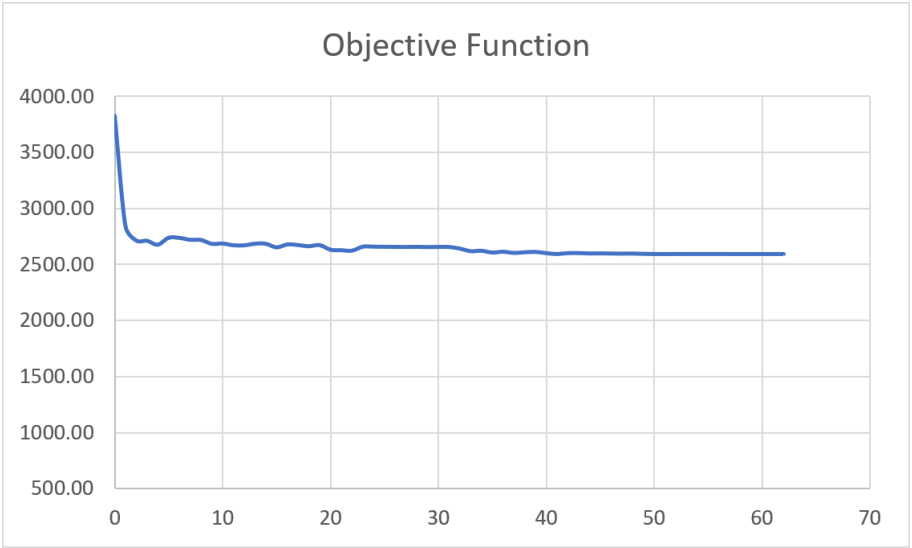


Figure 4.4: Objective Function Convergence - Test pre 1

A significant decrease of the mission time on first iteration can be noticed in Figure 4.4, approximately 1000 seconds (16.7 minutes). In the next 60 iterations the decrease of the mission time was much less significant, converging to a value of 2600 seconds (43 minutes). The whole

optimization process had an overall decrease of 1300 seconds (21.6 minutes).

4.2.2 Preliminary Test 2

The objective of this second test is to analyse the same conditions of the first test changing only the wind direction of all waypoints to 90 degrees.

Table 4.9: Fixed data for Mission planning - test pre 2.

Mission Data Input					
Waypoint number	**Ground/ Air**	**Loiter**	**Windspeed [m/s]**	**Wind direction[1] [deg]**	**Temp.Dev.[2] [K]**
1	G	N	0.00	90.00	3.337
2	G	N	5.00	90.00	3.348
3	A	N	5.00	90.00	3.366
4	A	N	5.00	90.00	4.581
5	G	N	5.00	90.00	3.009

S - add a loiter segment to that position;
N - do not add a loiter segment to that position;
G - waypoint located in ground; A - waypoint located in air;
1 - Wind direction North-East orientation; 2 - ISA Temperature deviation

In figure bellow, Figure 4.1, looking to the results, where the Yellow trajectory is the optimized one, the first thing that becomes clear comparing with test pre 1 is the low altitude of flight. Looking to Figure 4.2, the first idea of a low altitude of flight is confirmed, that occurs due to that, in this test, there is no wind in the same direction of the flight, so the ground speed, in this case, is lower than the first test, and, because of that, the UAV needs more energy to climb further or accelerate. On other hand, Figure 4.3, shows the Yellow trajectory almost coincident with the black one in the horizontal plane, which represent the same distance shortening as discussed in test pre 1. It represents that the design variables, latitude and longitude, were fully optimized in both tests. The airspeed results, shown in Table 4.10, show an almost constant airspeed during flight. Although test pre 2 has the same energy available than test pre 1 (battery energy), it needs more power to accelerate, as the wind speed component of the y axis is approximately zero, which does not increase the ground speed as it happens in test pre 1. So the UAV can not accelerate in order to have enough energy to reach the ending point.

Table 4.10: Output Design Variables - test pre 2

Output Design Variables				
Latitude 03 [deg]	**Latitude 04 [deg]**	**Longitude 03 [deg]**	**Longitude 04 [deg]**	**Altitude 03 [m]**
40.1634	40.008	-7.4388	-74403	836.1
Altitude 04 [m]	**Airspeed 02 [m/s]**	**Airspeed 03 [m/s]**	**Airspeed 04 [m/s]**	**Airspeed 05 [m/s]**
789.8	11.0	11.1	11.0	10.9

To better understand the algorithm behaviour, the objective function gradients calculated through each iteration are shown in Figure 4.5. The first idea that comes clear is that the airspeed has a larger influence on the objective function than the rest of the design variables. This seems correct as, in the same path, a higher speed reduces the time of flight, and as the airspeed gradients are negative, it means that an increase of the airspeed results in a minimization of

the objective function. By observing the graph it is noticed that the position (latitude, longitude and altitude) design parameters are fully optimized, as their gradients converged to 0. The airspeed gradients did not converge to 0 due to the active constraint functions, such as minimum remaining energy in the battery of 20% of full capacity, maximum motor setting and maximum current at design conditions. Also, it comes clear that 40 iterations are necessary to find a good solution. Yet it takes another 60 iterations to fulfil the convergence criteria.

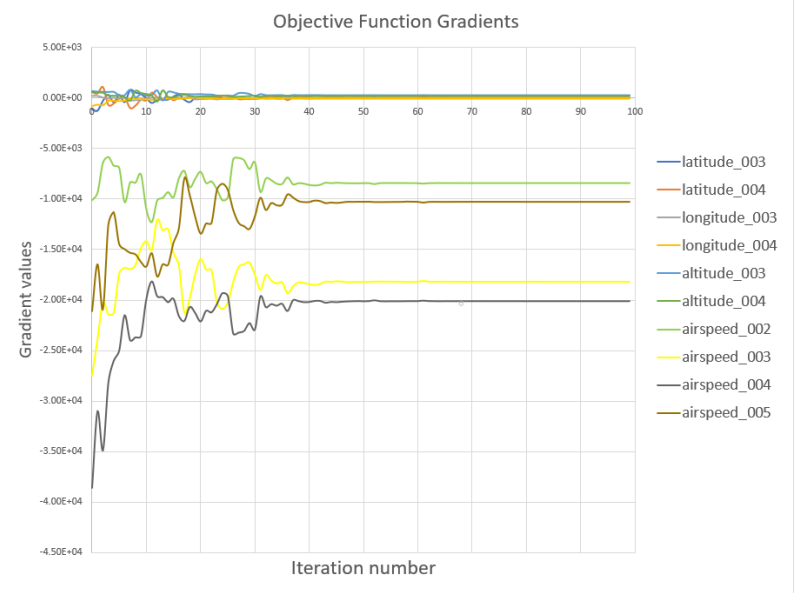


Figure 4.5: Objective Function Gradients - Test pre 2

4.3 Final Program Tests

In this Section, all the models were implemented and the program was ready to start the final tests. Firstly, tests were conducted to analyse the capability of terrain avoidance, as well as, the impact of the solar model in the mission planner behaviour. Secondly, more tests were performed to verify if the mission planer was capable of finding more energetic trajectories (low cloud cover zones). Followed by tests with real forecast data and a final mission across Portugal as an example of the full capability of the mission planner.

4.3.1 Terrain Avoidance

A mission, starting at 10:30AM from Covilhã to Aveiro, where the terrain is very irregular and in some zones it reaches almost two thousand meters of elevation, was chosen in to better test algorithms. Once the solar model was already implemented and the cloud cover was set to zero, the harvested energy through the path increased in a significant way. So, it compensates the UAV to climb to an higher level, where the thinner air at high altitudes reduces significantly the drag such that for the same amount of thrust applied, the UAV will fly faster. In order to avoid high altitude trajectories and better understand the capability of terrain avoidance, the maximum altitude was set to 1800 meters for the following tests.

After planning several missions, it was concluded that the model responsible for the terrain avoidance is effective, by always flying around the mountains and never passing through them. Test TA 1 and 2 presented in Figure 4.6, are two of the tests that confirm the capability of avoiding to pass through the terrain. Despite the terrain avoidance is verified, it can be seen that the test's TA 1 optimized solution is a local minimum of the objective function. This is due to when the problem is a non-convex function with multiple maximums or minimums, the FFSQP algorithm finds the maximum or minimum closest to the first guess (Input trajectory). Then, the trajectory in blue, whose initial design parameters (latitude, longitude and altitude) were set to the left of the mountain, the optimization process found a lower minimum compared to test TA 1.

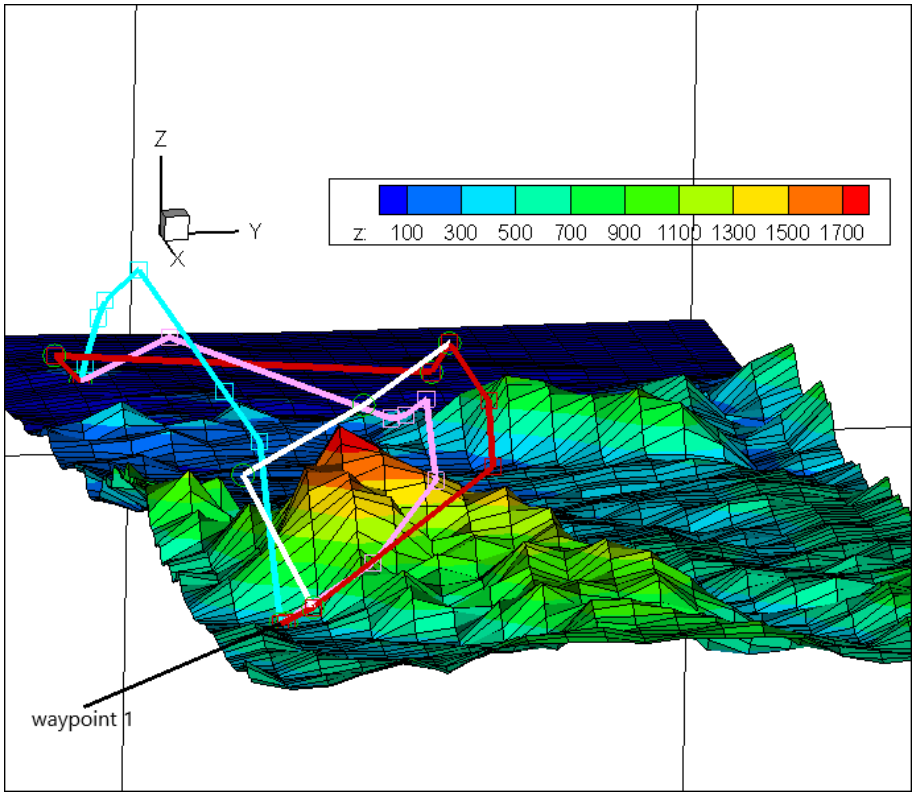
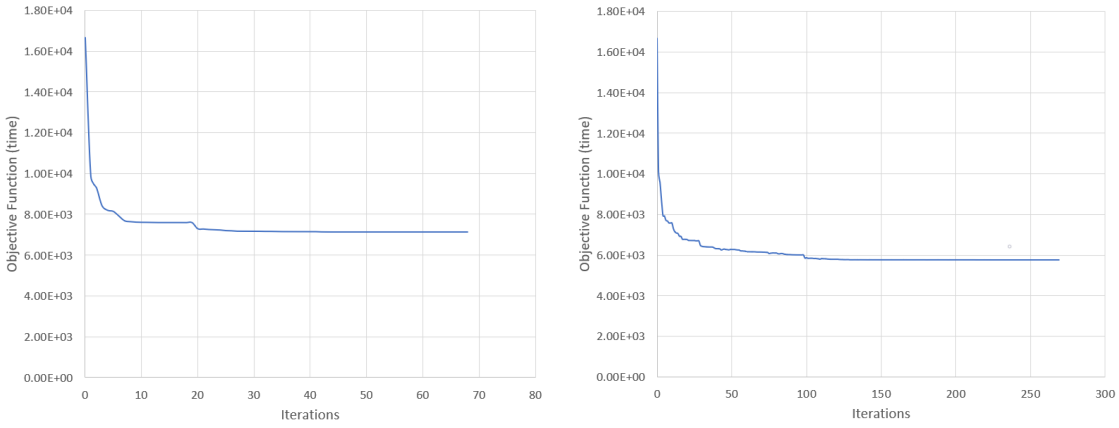


Figure 4.6: Comparison between test TA 1 and test TA 2. Red and white trajectories correspond to the first guesses of test TA 1 and TA 2, respectively. Pink and blue trajectories correspond to the solutions of test TA 1 and TA 2, respectively

Also, it is important to note that the pink solution shown in Figure 4.6 stopped in a local minimum as a consequence of the altitude limitations. In other words, the algorithm reached a constraint boundary (altitude limitation) during the optimization and stopped when the following iterations corresponded to feasible missions (design parameters) with the same objective function value. This is verified by test TA 2, as the algorithm of optimization was able to find a solution (blue one) that the initial trajectory is defined on the left side of the mountain. Furthermore, Figures 4.7a and 4.7b are a validation of this idea. As expected the test´s TA 2 objective function is lower than test's TA 1, and it can be noticed that test TA 1 stopped in a early iteration compared to test TA 2, also due to the altitude limitation.



(a) Objective function convergence of Test TA 1.

(b) Objective function convergence of Test TA 2.

Figure 4.7: Test's TA 1 and TA 2 Objective functions convergence.

4.3.2 Solar Model Behaviour

This Subsection has the purpose of understanding how the solar model with uniform cloud cover distribution throughout the map, affects the mission planning, where the mission time was chosen as objective function. The uniform cloud cover of the first test was set to 90%, the second one to 50% and the last two it was set to 0%, where the fourth has the maximum altitude limited to 1000 meters. The design variables used for the tests are defined in Table 4.11, also the starting time of these tests is 12AM.

Table 4.11: Mission Design Variables Input - Solar Model Behaviour

Mission Design Variables Input				
Waypoint number	Latitude [deg]	Longitude [deg]	Altitude [m]	Airspeed [m/s]
1	40.29563	-7.43693	0.00	0.00
2	40.29546	-7.43750	0.00	14.00
3	40.20458	-7.44433	1000.000	14.00
4	39.96362	-7.49178	1000.000	14.00
5	39.85038	-7.44293	0.00	14.00

Table 4.12: Mission Fixed Data Input - Solar Model Behaviour

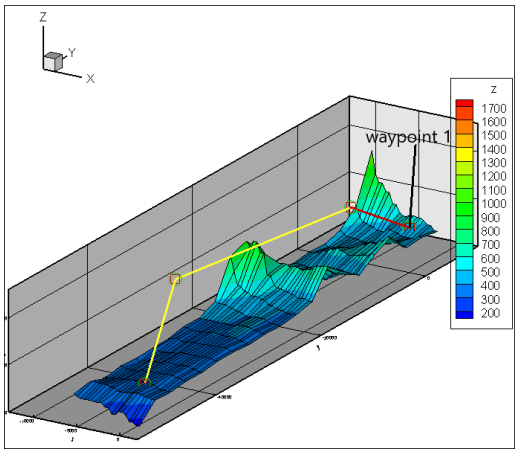
Mission Fixed Data Input					
Waypoint number	Ground/ Air	Loiter	Wind speed [m/s]	Wind direction [deg]	Temp. Dev. [K]
1	G	N	0.00	180	3.337
2	G	N	10.00	180	3.348
3	A	N	10.00	180	3.366
4	A	N	10.00	180	4.581
5	G	N	10.00	180	3.009
S - add a loiter segment to that position;					
N - do not add a loiter segment to that position;					
G - waypoint located in ground; A - waypoint located in air;					
1 - Wind direction North-East orientation; 2 - ISA Temperature deviation					

Table 4.13: Mission Planner Results - Solar Model Behaviour

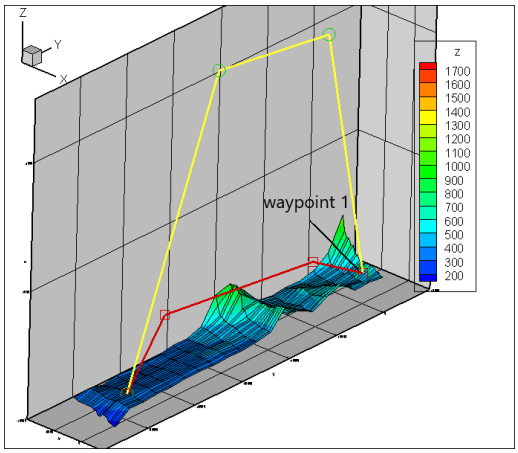
Mission Planner Results				
	Mission Time [s]	Av. Seg Solar Power [W]	Mission energy left [%]	Max. Altitude [m]
Test SM 1	unfeasible	224.47	-127[1]	-
Test SM 2	7452.14	948.18	20	4471.58
Test SM 3	6835.77	1063.43	20	4769.35
Test SM 4	8075.50	1047.82	20	1000.00
1 - -127% at waypoint 4 and 0.28% at waypoint 3.				

Table 4.14: Mission Planner Airspeed Results - Solar Model Behaviour

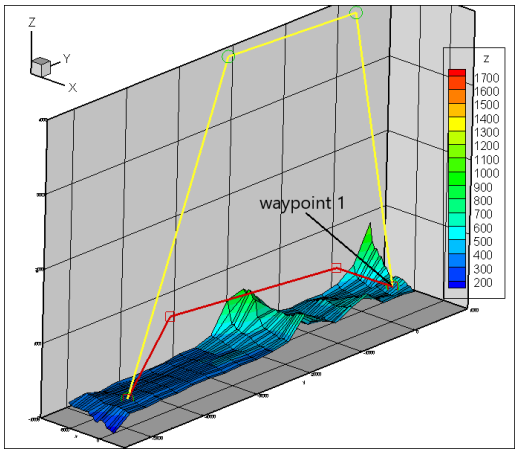
Mission Planner Airspeed Results			
Waypoint number	**Test SM 2 Airspeed [m/s]**	**Test SM 3 Airspeed [m/s]**	**Test SM 4 Airspeed [m/s]**
2	14.12	14.40	15.69
3	15.39	15.89	16.03
4	17.55	18.27	16.15
5	20.24	21.22	18.01



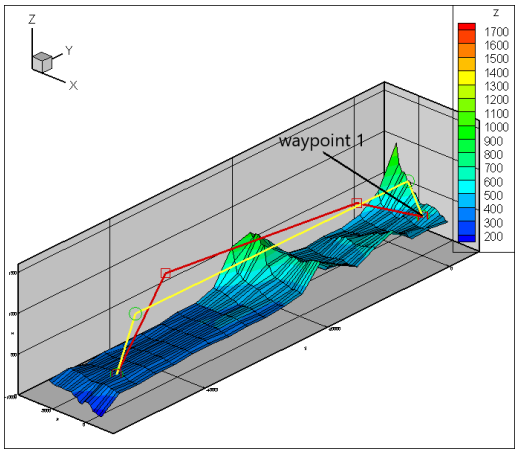
(a) 3D view of test SM 1



(b) 3D view of test SM 2



(c) 3D view of test SM 3



(d) 3D view of test SM 4

Figure 4.8: Comparison between the four tests. The red trajectory is the initial one , the yellow trajectory is the optimized one and the contour represents the terrain elevation.

After the algorithm was applied in each test, the mission planner returned a better solution compared to the first guesses, except for the Test SM 1 that there are no feasible solutions found. Test SM 1 did not have feasible solutions due to the energy available, not being sufficient for the UAV perform the mission. As described in Table 4.13, in the first guess analysis, the energy available was not enough to reach waypoint 4, likewise the energy left is 0.28% at waypoint 3. As the test has no solution, the output values are coincident with the first guess as shown in Figure 4.8a.

Test SM 2, with a decrease of cloud cover, the average solar power harvested through the mission increased, thus making possible to increase airspeed along the flight. The maximum altitude of flight is 4471 meters, where the drag is minimized, being this a reason of the increased airspeed too.

Test SM 3, has similar results compared to test SM 2, with lower value of cloud cover, the average solar power also increased, and consequently a lower optimized mission time.

To validate that the higher altitude trajectories of test SM 2 and SM 3, have better results than a lower altitude trajectory, test SM 4 was conducted with the same parameters of test SM 3, but with the maximum altitude limited to 1000 meters. So, looking at Figure 4.8d, it can be seen that, comparing to test SM 3 in Figure 4.8c, the trajectory's distance is lower than the test SM 3 trajectory's distance, to be precise, 1333.59 meters lower, and yet it has a mission time higher than test SM 3. This validates the idea referred in the beginning of this paragraph.

4.3.3 Non Uniform Cloud Cover

This Subsection has the purpose of understanding how the optimization algorithm behaves, regarding the position of the waypoints, when there is a variation of the cloud cover through the map.

Table 4.15: Mission Design Variables Input - Non Uniform Cloud Cover

Mission Design Variables Input				
Waypoint number	**Latitude [deg.]**	**Longitude [deg.]**	**Altitude [m]**	**Airspeed [m/s]**
1	40.26700	-7.47760	0.00	0.00
2	40.26225	-7.47760	0.00	8.00
3	40.26225	-7.30000	1000.00	10.00
4	40.27687	-7.52311	1700.00	10.00
5	40.35347	-8.01060	1700.00	10.00
6	40.45000	-8.31060	1000.00	10.00
7	40.60000	-8.60000	0.00	10.00

The following two tests, represent a mission from Covilhã to Aveiro, (-x direction), starting at 10 AM. These tests were conducted with interpolated data from weather forecast data file, that were modified in order to permit to achieve the purpose of this Subsection. The modifications done were: the wind speed of the weather files was set to 0 m/s, to not affect the results, and the weather data is considered constant in time. Both tests have the same design parameters defined in Table 4.15, and the same objective function, mission time. Despite test NU 2 has the same design parameters as test NU 1, it considers an extra waypoint in each segment. The purpose of this is explained below.

Table 4.16: Mission Planner General Results - Non Uniform Cloud Cover

Mission Planner Results - Test NU 1			
	Mission Time [s]	Mission energy left [%]	Max. Altitude [m]
Test NU 1	6220.69	20.0	3925.50
Test NU 2	7216.08	20.0	5824.36

Table 4.17: Output airspeed by segment, cloud cover and solar power by segment - Test NU 1

Mission Planner Results - Test NU 1				
Segment number	Airspeed [m/s]	Altitude [m]	Cloud Cover [%]	Solar Power [W]
1	7.65	449.19	86	340.77
2	14.95	2119.67	50	939.02
3	15.37	3776.04	17	1064.75
4	16.30	3842.06	25	1056.80
5	16.87	3643.32	23	1065.53
6	18.63	1705.74	38	1028.58
Average	14.01	2589.3	39.83	915.91

Table 4.18: Output airspeed by segment, cloud cover and solar power by segment - Test NU 2

Mission Planner Results - Test NU 2				
Segment number	Airspeed [m/s]	Altitude [m]	Cloud Cover [%]	Solar Power [W]
1	2.78	450.8	86	340.77
2	8.34	447.3	86	341.22
3	11.52	1112.2	59	858.53
4	12.32	2445.0	26	1047.41
5	13.19	3515.8	38	1022.58
6	14.11	4324.7	60	863.38
7	14.58	5002.9	45	991.10
8	14.59	5550.5	42	1012.98
9	15.69	5729.5	58	887.50
10	17.87	5539.9	38	1028.68
11	19.95	4096.3	40	1012.34
12	21.91	1399.0	62	824.83
Average	13.90	3301.1	53.3	852.70

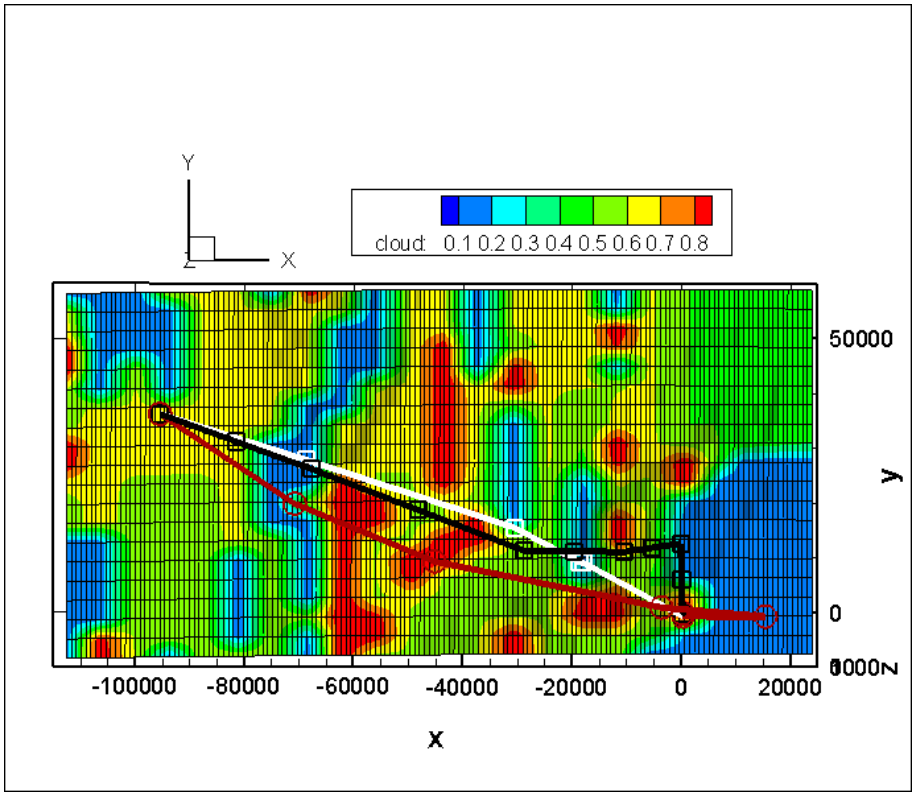


Figure 4.9: Comparison between test NU 1 and test NU 2. The red trajectory is the initial one , the white and the black trajectory are the test's NU 1 and 2 solutions and the contour represents the cloud cover variation.

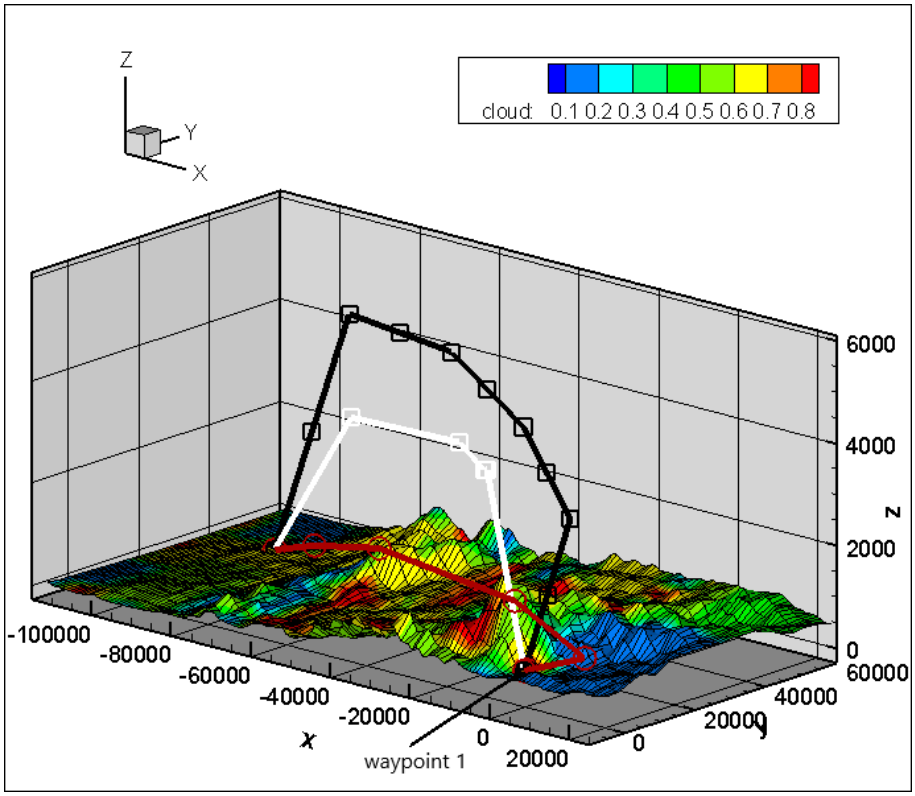


Figure 4.10: Comparison between test NU 1 and test NU 2. The red trajectory is the initial one , the white and the black trajectory are the test's NU 1 and 2 solutions and the contour represents the cloud cover variation.

Test NU 1 results, presented in Tables 4.16 and 4.17 and in Figures 4.9 and 4.10, show that the algorithm is effective in finding zones of low cloud cover. By looking at the optimized white trajectory, in Figure 4.9, it is noticed that each waypoint was ”moved” to a position with lower cloud cover than the initial, which validates the previous idea. Yet, it is important to note that, as the analysis is done by segment, the cloud cover results are average values between the initial and ending waypoint of each segment. So looking to the cloud variation between the waypoints, these values are very different from the real. To solve this, the second test was conducted with the same design variables but, automatically setting one waypoint (not a design variable) between each segment of the mission. This allows the mission planner to have a better estimation of the local weather data, but consequently resulting in a longer computational time.

Regarding a realistic mission planning, test NU 2 has better results, presented in Tables 4.16 and 4.18 and in Figures 4.9 and 4.10, than test NU 1. Test NU 2 also validates the capability of the algorithm to ”find” zones of low cloud cover, but with more realistic estimations of values. Comparing with test NU 1, there is a higher average cloud cover, and consequently a lower average solar power, during the flight. Even though test’s NU 2 weather estimations are more realistic than test’s NU 1, they are not 100% real. It could be added more extra waypoints in each segment, to achieve even better results, yet it would lead to an even longer time of computation.

Comparing both in terms of processing time, test NU 1 took approximately 5 hours and 30 minutes planning the mission. On other hand, test NU 2 took approximately 9 hours. Seeing that adding another waypoint in each segment would take to much time to plan a mission.

4.3.4 Real Weather Forecast Data

This Subsection has the purpose of presenting two tests of a mission planning, considering real unmodified weather forecast data, and a loiter phase. A mission from Covilhã to Viseu was chosen to be tested, where at this point the UAV realizes a loiter phase at 65 meters from the ground, for 1 hour, following then to Aveiro.

The weather data is a forecast from February 12^{th} of 2019, where the missions starting time is set to 8:00 AM. Tests RW 1 and 2 were conducted selecting different objective functions, mission time and mission consumed energy respectively.

Table 4.19: Mission Design Variables Input - Real Weather Forecast Data

Mission Design Variables Input						
Waypoint number	Ground/ air	Loiter	Latitude [deg.]	Longitude [deg.]	Altitude [m]	Airspeed [m/s]
1	G	N	40.26700	-7.47760	0.00	0.00
2	G	N	40.26225	-7.47760	0.00	13.00
3	A	N	40.35033	-7.35322	2000.00	13.00
4	A	N	40.62418	-7.57809	2000.00	13.00

Table 4.19: Mission Design Variables Input - Real Weather Forecast Data

Mission Design Variables Input						
5	A	S	40.65854	-7.91170	60.00	13.00
6	A	N	40.67851	-8.19813	2000.00	13.00
7	G	N	40.60000	-8.60000	0.00	13.00
Loiter Waipoint	**Time [s]**	3600	**Airspeed [m/s]**	10.00	**Radius [m]**	50.00

Table 4.20: Mission Planner General Results - Winter

Mission Planner Results				
	Mission Time [s]	**Mission energy [J]**	**Mission energy left [%]**	**Max. Altitude [m]**
Test RW 1	12177.85	323635.75	20	2163.10
Test RW 2	18667.20	-513641.61	100	1285.79

Table 4.21: Output airspeed by segment - RW 1 and 2

Mission Planner Results		
Segment number	**Test RW 1 Airspeed [m/s]**	**Test RW 2 Airspeed [m/s]**
1	6.64	4.48
2	13.16	8.54
3	13.31	8.74
4	13.15	8.90
5	10.00	10.00
6	12.21	8.93
7	13.08	9.93
Average	13.77	8.50

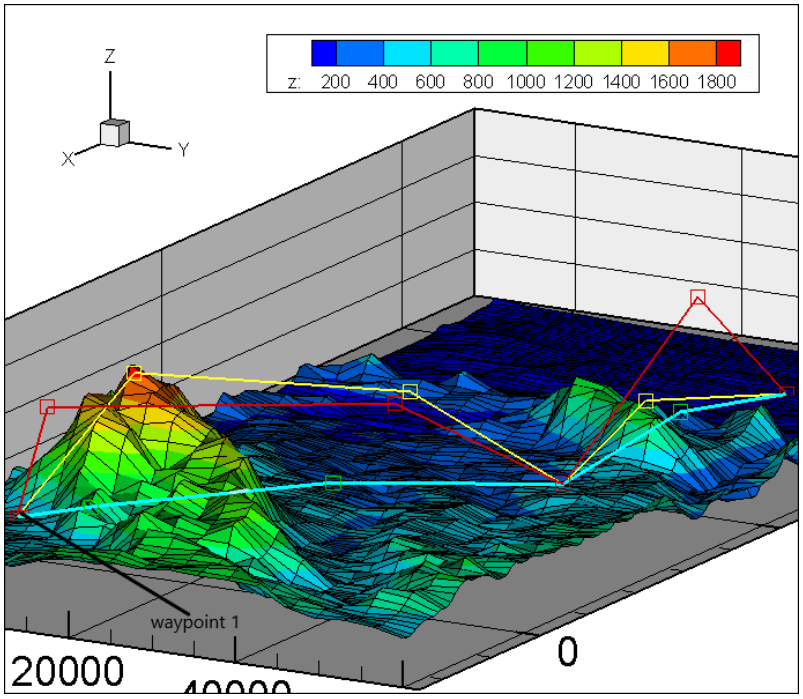


Figure 4.11: 3D view Real Weather Forecast Data Test RW 1.

Figure 4.12: Comparison between test RW 1 and test RW 2. The red trajectory is the initial one , the yellow and blue trajectories are the test RW 1 and 2 and the contour represents the terrain elevation.

These tests result, presented in Tables 4.20, 4.21, and in Figure 4.12, show the well functioning of the mission planner. Test RW 1 optimized the mission time to approximately 12177 seconds (3.4 hours), which required a 323635 J of consumed energy. On the other hand, test RW 2 optimized the mission consumed energy to a value of -513641 J (the negative sign means that 513641 J available were not used), in 18667 seconds (5.2 hours). Both of the tests took approximately one hour and thirty minutes long to plan the mission, which is a good time of processing, compared to the tests presented in the previous Section. This is due to the tests RW 1 and 2 having the loiter waypoint fixed in one position (fewer design parameters), and mainly because the initial trajectory is relatively close to its solution, as opposed to NU tests.

It was thought that it would be interesting to test the same mission planned in the Summer's solstice, in order to compare the results of a mission during Winter with the results of another during the Summer. In spite that this Section is about tests with real weather forecast data, due to that being impossible, at the time these tests were done, to predict weather data from Summer, tests were conducted, considering the same weather data of the previous tests, except for the temperature that was increased considering normal temperatures at summer.

Table 4.22: Mission Planner General Results - Summer

Mission Planner Results				
	Mission Time [s]	Mission energy [J]	Mission energy left [%]	Max. Altitude [m]
Test RW 3	10377.38	341271.12	20	2614.54
Test RW 4	15730.76	-999993.73	100	2767.72

Table 4.23: Output airspeed by segment - RW 3 and 4

Mission Planner Results		
Segment number	**Test RW 3 Airspeed [m/s]**	**Test RW 4 Airspeed [m/s]**
1	7.95	5.80
2	15.89	11.73
3	16.37	11.59
4	16.80	11.78
5	10.00	10.00
6	16.84	11.97
7	17.31	11.95
Average	13.77	8.50

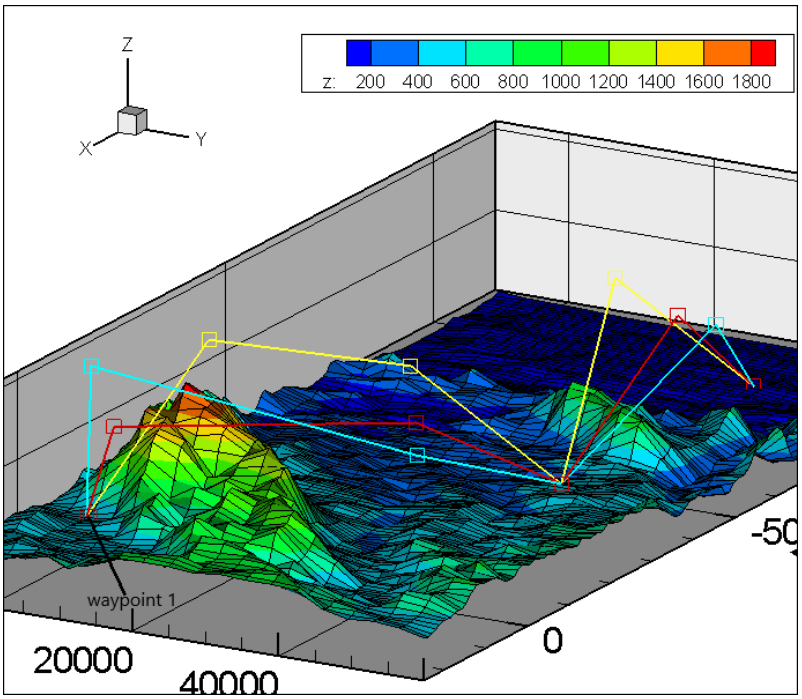


Figure 4.13: Comparison between test RW 1 and test RW 2. The red trajectory is the initial one , the yellow and blue trajectories are the test RW 3 (optimization of mission time) and 4 (optimization of mission energy) and the contour represents the terrain elevation.

These tests results, comparing with tests RW 1 and RW 2 of this Section, represent what was expected: a general increase in solar energy available which led to better performance of the UAV during mission. Also, it can be noticed, that test RW 4 (optimization of mission energy) compared to test RW 2, climbs the mountain instead of going around, what is due to the increase of solar available energy. An interesting subject that must be mentioned is that the 4 only used 20 minutes of planning process, unlike test RW 3 that used 2 hours and 23 minutes. This indicates that the time of processing depends in many factors, like which objective function was chosen, or if the input design parameters are close from the solution. Other factors noticed during this work, to affect the processing time are: if the terrain or the weather variation are to irregular; or the number of segments to be analysed.

4.3.5 Mission Across Portugal

For a final example of the mission planner it is presented a test of a mission starting at 6 AM, from Chaves to Faro minimizing the mission time.

Table 4.24: Mission Design Variables Input - Mission Across Portugal

Mission Design Variables Input						
Waypoint number	Ground/ air	Loiter	Latitude [deg.]	Longitude [deg.]	Altitude [m]	Airspeed [m/s]
1	G	N	41.72519	-7.46067	0.00	0.00
2	G	N	41.71904	-7.46067	0.00	10.00
3	A	N	41.72083	-7.68063	2000.00	10.00
4	A	N	41.14165	-8.06910	2000.00	10.00
5	A	N	40.47084	-7.78000	1700.00	10.00
6	A	N	40.50403	-8.01680	1700.00	10.00
7	A	N	39.84264	-8.04679	1700.00	10.00
8	A	N	38.87701	-8.04783	1700.00	10.00
9	A	N	37.47900	-8.04627	1700.00	10.00
10	A	N	37.29412	-8.04121	1700.00	10.00
11	G	N	37.01338	-7.95798	0.00	10.00

Table 4.25: Mission Planner General Results - Mission Across Portugal

Mission Planner Results - Test FC 1		
Mission Time [s]	Mission energy left [%]	Max. Altitude [m]
29829.8	20.0	3925.50

Table 4.26: Output airspeed by segment.

Mission Planner Results				
Segment number	Airspeed [m/s]	Altitude [m]	Cloud Cover [%]	Solar Power [%]
1	5.11	376.0	0.0	297.92
2	10.39	1813.0	0.0	304.14
3	11.32	3476.6	0.0	514.62
4	12.77	4355.0	0.0	696.66
5	13.80	6062.3	0.0	874.96
6	16.43	7742.0	0.0	994.27
7	19.20	9174.6	0.0	1048.67
8	21.87	9401.8	0.0	1062.46
9	22.28	8494.2	0.0	1023.95
10	22.67	4089.0	0.0	1007.13
Average	15.58	4558.3	0.00	782.5

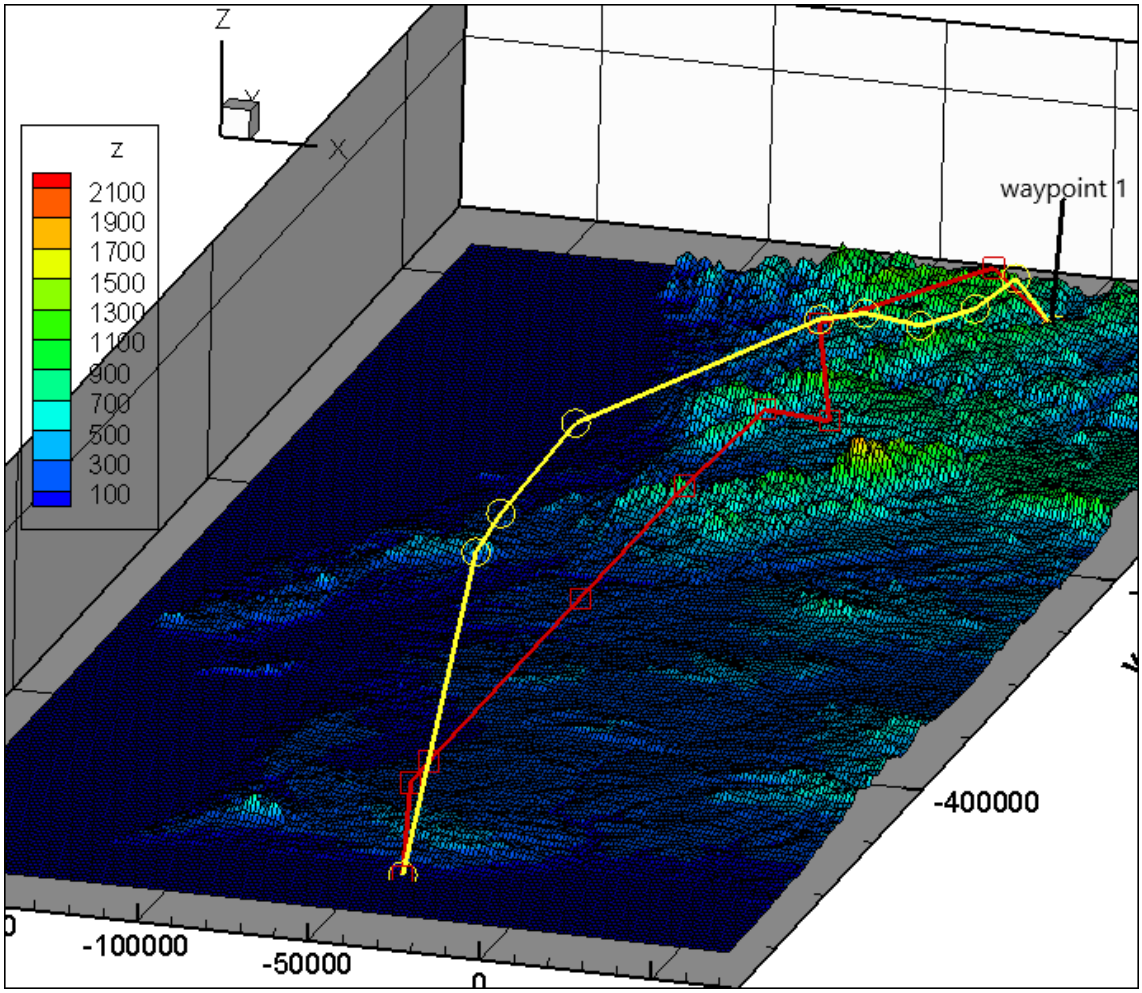


Figure 4.14: 3D view of the Mission Across Portugal. The red and yellow trajectories are the initial and the optimized respectively.

The results presented in Tables 4.25, 4.26 and in Figure 4.14, show that at the first half of the mission, when the sun was low, the solar power was much lower than the rest of the mission, and because of that, the altitude and airspeed of flight are much lower than the rest of the mission. From waypoint 5, the solar power available, due to the sun being higher, increased which permits the UAV to flight at higher airspeed and altitude. It should be mentioned that this test took 20 hours of computational time, which is mainly due to the design parameters not being close to the solution, as we can see in Table 4.24 and 4.26, the design altitude and airspeed of each waypoint are very different from the solution.

Chapter 5

Conclusions

A program that allows to plan a mission in any part of the Earth and for any solar powered UAV, by defining an initial mission and setting parameters regarding the UAV specifications and mission type, was developed and verified through several mission planning tests. To achieve the main goal of this work other tasks were also performed such as the implementation of various mathematical models that allow the program to consider: real weather forecast, the solar power harvested by the PV panels; the real terrain elevation; the correct energy management by the systems and the propulsion performance.

In general the mission planner presents good results in terms of mission planning and optimization. Regarding the terrain avoidance, it is efficient, respecting in any segment the defined minimum height constraint. Yet the results are estimations different from the real. This difference can be minimized by setting extra waypoints in each segment, which leads to a longer planning process, nevertheless, this time of processing depends on how far the first guess is from the solution. Another way to have more realistic results is to request more points from the databases, which is easier to do if the user has a paid subscription account in the databases.

Although the main objective has been achieved, some aspects need to be carefully analysed. The long time to plan missions in a more realistic way is one of them. Yet, this is a first version of the mission planner and some tasks to do in order to improve it are described in the next Section.

5.1 Future Work

As discussed throughout the current thesis, some improvements should be performed. These improvements focus mainly on improving the computational time and more realistic values. Below, these aspects are described in detail:

- Implementation and verification of the Genetic Algorithm, which is also an optimization algorithm, which is based on probability. It differs from the classical optimization algorithm as it generates a population of points at each iteration, where the best point in the population approaches an optimal solution and selects the next population by computation which uses random number generators.
- Make the mission planner capable of optimizing two objective functions at the same time, to be able to have a balance of energy and time in a mission plan;
- Selection of more complete weather database that considers the vertical wind and weather variation with the altitude;
- Development of an intuitive interface.

It is also interesting to develop a mission planner algorithm that plans a mission in real time to correct the mission, during flight, if any parameter is different from the original pre-planned mission.

Bibliography

[1] D. Teodorovic, *Airline Operations Research*. Routledge, apr 2017. [Online]. Available: https://www.taylorfrancis.com/books/9781315211848 1

[2] D. L. BASHIOUM, E. CARR, and L. SIMPSON, "Computer flight planning in the north atlantic," *Journal of Aircraft*, vol. 2, no. 4, pp. 337–346, jul 1965. 1

[3] M. d. S. Arantes, J. d. S. Arantes, C. F. M. Toledo, and B. C. Williams, "A Hybrid Multi-Population Genetic Algorithm for UAV Path Planning," in *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference - GECCO '16*, 2016, pp. 853–860. [Online]. Available: http://dx.doi.org/10.1145/2908812.2908919http://dl.acm.org/citation.cfm?doid=2908812.2908919 1

[4] S. M. Lavalle, *PLANNING ALGORITHMS*. Cambridge University Press, 2006. [Online]. Available: http://planning.cs.uiuc.edu/ 3

[5] P. Yao, H. Wang, and Z. Su, "Hybrid UAV path planning based on interfered fluid dynamical system and improved RRT," in *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*. IEEE, nov 2015, pp. 829–834. [Online]. Available: http://ieeexplore.ieee.org/document/7392202/ 3

[6] S. Garrido, M. Abderrahim, and L. Moreno, "PATH PLANNING AND NAVIGATION USING VORONOI DIAGRAM AND FAST MARCHING," *IFAC Proceedings Volumes*, vol. 39, 2006. 3

[7] C. Cormen, Thomas H.; Leiserson, Charles E.; Rivest, Ronald L.; Stein, ""Section 24.3: Dijkstra's algorithm"." in *Introduction to Algorithms (Second ed.)*. 3

[8] H. O. Norheim, "Graphs and Graph Algorithms in T-SQL." 4

[9] L. Vinet and A. Zhedanov, "Rapidly-Exploring Random Trees: A New Tool for Path Planning," *Journal of Physics A: Mathematical and Theoretical*, vol. 44, no. 8, p. 085201, 2011. 4

[10] J. G. Ibrahim, M. H. Chen, and S. R. Lipsitz, "Monte carlo EM for missing covariates in parametric regression models," *Biometrics*, vol. 55, no. 2, pp. 591–596, sep 1999. 5

[11] R. J. Boucher, "History of Solar Flight By 20th Joint Propulsion Conference," pp. 1–22, 1984. [Online]. Available: https://web.archive.org/web/20110707180738/http://www.astroflight.com/pdfs/SolarHistory.pdf 5, 6

[12] G. Pearson, D. Chapin, and A. Noth, "History of Solar Flight," Tech. Rep., 2008. [Online]. Available: http://www.sky-sailor.ethz.ch1 6, 7

[13] A. S. Rodrigues, "Airframe Assembly , Systems Integration and Flight Testing of a Long Endurance Electric UAV," Ph.D. dissertation, Universidade da Beira Interior, 2017. 5, 6, 7

[14] R. Boucher, "Project Sunrise - Flight of the Worlds First Solar Powered Aircraft," p. 1, 1974. [Online]. Available: http://www.projectsunrise.info/First{_}Solar{_}Powered{_}Aircraft.html 6

[15] "Berblinger Wettbewerb 2013 Ulm." [Online]. Available: http://www.berblinger.ulm.de/html/berblinger{_}prize 6

[16] Y. Gibbs, “NASA Dryden Fact Sheet - Pathfinder Solar-Powered Aircraft,” *NASA Past Projects*, 2015. 7

[17] ——, “Centurion Solar-Powered Aircraft,” *NASA Past Projects*, 2017. 7

[18] L. Blain, “Solar-powered Zephyr smashes record for the longest unmanned flight,” 2007. [Online]. Available: https://newatlas.com/solar-powered-zephyr-smashes-record-for-the-longest-unmanned-flight/7990/ 7

[19] A. Cowell, “Solar-Powered Plane Stays Aloft for 26 Hours - The New York Times,” 2010. [Online]. Available: https://www.nytimes.com/2010/07/09/world/europe/09plane.html 7

[20] “Solar Impulse Clean Technologies to Fly Around the World.” [Online]. Available: https://aroundtheworld.solarimpulse.com/ 7

[21] M. Wei-Haas, “Inside the First Solar-Powered Flight Around the World,” 2018. [Online]. Available: https://www.smithsonianmag.com/innovation/inside-first-solar-powered-flight-around-world-180968000/ 7

[22] P. Oettershagen, A. Melzer, T. Mantel, K. Rudin, T. Stastny, B. Wawrzacz, T. Hinzmann, K. Alexis, and R. Siegwart, “Perpetual flight with a small solar-powered UAV: Flight results, performance analysis and model validation,” in *2016 IEEE Aerospace Conference*. IEEE, mar 2016, pp. 1–8. 8

[23] “Mission Planner Overview — Mission Planner documentation.” [Online]. Available: http://ardupilot.org/planner/docs/mission-planner-overview.html 8

[24] “QBase the all in one mission control system by Quantum-Systems.” [Online]. Available: https://www.quantum-systems.com/qbase/ 8, 9

[25] R. Dasilveira, “Optimum Propeller Designs For Electric UAVs.” *Mycological Research*, vol. 106, no. 11, pp. 1323–1330, 2002. [Online]. Available: https://etd.auburn.edu/bitstream/handle/10415/3158/David{_}Wall{_}Thesis.pdf?sequence=2 17

[26] P. F. Godinho Lopes Fernandes de Albuquerque, “Mission-Based Multidisciplinary Design Optimization Methodologies for Unmanned Aerial Vehicles with Morphing Technologies,” Ph.D. dissertation, Universidade da Beira Interior, 2017. 17, 18, 19, 20

[27] “Elevation API.” [Online]. Available: https://elevation-api.io/ 21

[28] “Current weather and forecast - OpenWeatherMap.” [Online]. Available: https://openweathermap.org/ 23

[29] X.-Z. Gao, Z.-X. Hou, Z. Guo, J.-X. Liu, and X.-Q. Chen, “Energy management strategy for solar-powered high-altitude long-endurance aircraft,” *Energy Conversion and Management*, vol. 70, pp. 20–30, jun 2013. 24

[30] C. J. Smith, J. M. Bright, and R. Crook, “Cloud cover effect of clear-sky index distributions and differences between human and automatic cloud observations,” *Solar Energy*, vol. 144, pp. 10–21, 2017. 25

[31] J. S. Rajshekar, “Design and Development of an Automated Multi Axis Solar Tracker Using PLC,” vol. 2, no. 3, pp. 204–211, 2013. 25

[32] ITACA, “Calculating Solar Angles.” [Online]. Available: https://www.itacanet.org/the-sun-as-a-source-of-energy/part-3-calculating-solar-angles/{#}3.5.-Angle-Of-Incidence 26

[33] J. L. Zhou, A. L. Tits, and C. T. Lawrence, *User’s guide for FFSQP version 3.7: A FORTRAN code for solving constrained nonlinear (minimax) optimization problems, generating iterates satisfying all inequality and linear constraints*, 1998, no. Dmc. [Online]. Available: http://aemdesign.com/download-ffsqp/ffsqp-manual.pdf 28, 29

[34] N. S. Ribau, “Automatic engine and propeller selection for mission and performance optimization,” Master’s thesis, Universidade da Beira Interior, Portugal, 2018. 29, 30