UNIVERSIDAD POLITÉCNICA DE VALÉNCIA

DOCTORAL THESIS

Deterministic model of the radio channel applied to the optimization of the UAV trajectory for optimum air-to-ground communication in the environment of future urban scenarios

Author:
Adrián EXPÓSITO GARCÍA

Supervisor:
Héctor ESTEBAN GONZÁLEZ
Dominic SCHUPKE

*A thesis submitted in fulfillment of the requirements
for the degree of Doctor of Philosophy*

in the

Microwave Applications Group (GAM)
iTEAM

February 22, 2023

Declaration of Authorship

I, Adrián EXPÓSITO GARCÍA, declare that this thesis titled, "Deterministic model of the radio channel applied to the optimization of the UAV trajectory for optimum air-to-ground communication in the environment of future urban scenarios" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.

- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.

- Where I have consulted the published work of others, this is always clearly attributed.

- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.

- I have acknowledged all main sources of help.

- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

UNIVERSIDAD POLITÉCNICA DE VALÉNCIA

Abstract

iTEAM

Doctor of Philosophy

Deterministic model of the radio channel applied to the optimization of the UAV trajectory for optimum air-to-ground communication in the environment of future urban scenarios

by Adrián Expósito García

Modern cities are at the limit of their capacity in the horizontal plane. Many of them have a traffic problem that is highly complex to alleviate or solve. Urban air mobility promises to be the revolution that can solve traffic saturation in future urban scenarios. The growth of the urban air mobility market is expected to show a constant positive tendency, but the associated technology needs to raise its readiness levels. Managing aerial vehicle fleets, dependent on rising technologies such as artificial intelligence and automated ground control stations, will require a solid and uninterrupted connection to complete their trajectories. The requirement for an uninterrupted connection is naturally connected to a complete understanding of phenomena affecting the air-to-ground channel. The first contribution to the field is to propose a channel model that can capture the consequences of said phenomena. Typically, such a model can output the channel state at a given point, predicting the channel state throughout the aircraft's trajectory. A highly detailed model demands tools and data to deliver the necessary information. The description and enumeration of each piece of information required for a successful channel simulation compose the second contribution to the field. Once the channel state is known, the travelled points by the aircraft can be optimised to cover those with better channel performance. The third and last contribution to the field is proposing a set of optimisation algorithms to find the most suitable route. The optimisation algorithm forms the path planner, expected to efficiently explore the search space and propose a trajectory compliant with predefined objectives: maximum air-to-ground quality, availability, and flight time. Each proposed method is tested in various scenarios. These scenarios include various situations that can stress the methods and favour the choice of one. Included situations are different terrain conditions and no-fly zones.

UNIVERSIDAD POLITÉCNICA DE VALÉNCIA

Resumen

iTEAM

Doctor of Philosophy

Deterministic model of the radio channel applied to the optimization of the UAV trajectory for optimum air-to-ground communication in the environment of future urban scenarios

by Adrián Expósito García

Las ciudades modernas están al límite de su capacidad en el plano horizontal. Muchas de ellas tienen un problema de tráfico muy complejo de paliar o resolver. La movilidad aérea urbana promete ser la revolución que puede resolver la saturación del tráfico en los futuros escenarios urbanos. Se espera que el crecimiento del mercado de la movilidad aérea urbana muestre una tendencia positiva constante, pero la tecnología asociada necesita aumentar su madurez. La gestión de múltiples vehículos aéreos, que dependen de tecnologías en auge como la inteligencia artificial y las estaciones de control en tierra automatizadas, requerirá una conexión tierra-aire-tierra sólida e ininterrumpida para completar sus trayectorias. La exigencia de una conexión ininterrumpida está naturalmente relacionada con una comprensión completa de los fenómenos que afectan al canal aire-tierra. La primera contribución es proponer un modelo de canal que pueda capturar las consecuencias de dichos fenómenos. Normalmente, un modelo de este tipo puede emitir el estado del canal en un punto determinado, prediciendo el estado del canal a lo largo de la trayectoria de la aeronave. Un modelo muy detallado exige herramientas y datos que proporcionen la información necesaria. La descripción y enumeración de cada pieza de información necesaria para una simulación de canal satisfactoria componen la segunda contribución. Una vez conocido el estado del canal, se pueden optimizar los puntos recorridos por la aeronave para cubrir aquellos con mejor rendimiento del canal. La tercera y última contribución es la propuesta de un conjunto de algoritmos de optimización para encontrar la ruta más adecuada. El algoritmo de optimización constituye el planificador de trayectorias, del que se espera que explore eficazmente el espacio de búsqueda y proponga una trayectoria que cumpla con los objetivos predefinidos: máxima calidad aire-tierra, disponibilidad y tiempo de vuelo. Cada método propuesto se pone a prueba en varios escenarios. Estos escenarios incluyen diversas situaciones que pueden estresar a los métodos y favorecer la elección de uno de ellos. Las situaciones incluidas son diferentes condiciones del terreno y zonas de exclusión aérea.

UNIVERSIDAD POLITÉCNICA DE VALÉNCIA

Resum

iTEAM

Doctor of Philosophy

Deterministic model of the radio channel applied to the optimization of the UAV trajectory for optimum air-to-ground communication in the environment of future urban scenarios

by Adrián Expósito García

Les ciutats modernes estan al límit de la seua capacitat al pla horitzontal. Moltes tenen un problema de trànsit molt complex de pal·liar o resoldre. La mobilitat aèria urbana promet ser la revolució que pot resoldre la saturació del trànsit als futurs escenaris urbans. S'espera que el creixement del mercat de la mobilitat aèria urbana mostre una tendència positiva constant, però la tecnologia associada necessita augmentar-ne la maduresa. La gestió de múltiples vehicles aeris, que depenen de tecnologies en auge com la intel·ligència artificial i les estacions de control a terra automatitzades, requerirà una connexió terra-aire-terra sòlida i ininterrompuda per completar les seues trajectòries. L'exigència d'una connexió ininterrompuda està relacionada naturalment amb una comprensió completa dels fenòmens que afecten el canal aire-terra. La primera contribució és proposar un model de canal que puga capturar les conseqüències dels fenòmens esmentats. Normalment, un model d'aquest tipus pot emetre l'estat del canal en un punt determinat, predient l'estat del canal al llarg de la trajectòria de l'aeronau. Un model molt detallat exigeix eines i dades que proporcionen la informació necessària. La descripció i l'enumeració de cada peça d'informació necessària per a una simulació de canal satisfactòria componen la segona contribució. Una vegada conegut l'estat del canal, es poden optimitzar els punts recorreguts per l'aeronau per tal de cobrir aquells amb el millor rendiment del canal. La tercera i última contribució és la proposta d'un conjunt d'algorismes d'optimització per trobar la ruta més adequada. L'algorisme d'optimització constitueix el planificador de trajectòries, del qual s'espera que explore eficaçment l'espai de cerca i propose una trajectòria que complisca els objectius predefinits: màxima qualitat aire-terra, disponibilitat i temps de vol. Cada mètode proposat es posa a prova a diversos escenaris. Aquests escenaris inclouen diverses situacions que poden estressar els mètodes i afavorir-ne l'elecció. Les situacions incloses són diferents condicions del terreny i les zones d'exclusió aèria.

Acknowledgements

Contents

List of Figures

List of Tables

Abreviations

A2A Air to air

A2G Air to ground

ACO Ant colony optimization

AER Azimuth elevation range

AESTER Advanced Spaceborne Thermal Emission and Reflection Radiometer

AGI-STK Analytical Graphics Inc. Systems Tool Kit

AGL Above ground level

API Application program interface

AS Aerial system

ATC Air traffic control

Az Azimuth

BS Base station

C2 Command and control

COTS Commercial of-the-shelf

DEM Digital elevation model

EA Evolutionary algorithm

Elev Elevation

EM Electro magnetic

FEF Fat earth frame

GN Ground network

GPS Generalized pattern search

GNSS Global navigation satellite system

LOS Line of sight

MILP Mixed integer linear programming

MIMO Multiple-input multiple-output

MPC Model predictive control

NFZ No fly zone

OSM OpenStreeMap

PSO Particle swarm optimisation

R Range

RDN Ray density normalisation

SAR Synthetic aperture radar

SINR Ssignal interference plus noise ratio

STRM Shuttle Radar Topography Mission

SWAP Size, weight and power

TanDEM-X TerraSAR-X add-on for digital elevation measurement

UAS Unmanned aerial system

UTD Uniform theory of diffraction

List of Symbols

θ	rotation angle	deg
$d\mu$	latitude variation	deg
μ	latitude	deg
μ_0	initial latitude	deg
dl	longitude variation	deg
l	longitude	deg
l_0	initial longitude	deg
Cr_N	radius of curvature in the prime vertical	m
Cr_M	radius of curvature in the meridian	m
$\mathcal{E}_R$	planet radius	m
$\mathcal{E}_F$	planet flattening	
x	x-axis position	m
y	y-axis position	m
z	z-axis position	m
d	distance	m
d_{LOS}	Line-of-sight distance	m
h	height	m
Φ	Roll	deg
Θ	Pitch	deg
Ψ	Yaw	deg
v	Speed	$\mathrm{m\,s^{-1}}$
M	Number of points per route	
N	Number of waypoints per route	
L	Aircrafts' state at a given time	
$\mathcal{L}^L$	Route represented as state vectors	
$\mathcal{L}^{WP}$	Route represented as waypoints	
$\mathcal{V}$	Triangle vertex	
O	Origin	
(u,v)	Baricentric coordinates	
R	Ray	
$\mathcal{M}$	Transformation matrix	
$\mathcal{M}R$	Rotation matrix	
$\mathcal{M}T$	Translation matrix	
Δx	x-axis displacement	
Δy	y-axis displacement	
Δz	z-axis displacement	
S	Power density	$\mathrm{W\,m^{-1}}$
P_T	Transmitted power	W
P_R	Received power	W
A	Area	$\mathrm{m^2}$
λ	Wavelength	m
f	Frequency	Hz

G_T	Gain of transmitter	
G_R	Gain of receiver	
θ	Angle	deg
ϵ	Dielectric permittivity	$\mathrm{F\,m^{-1}}$
η	Magnetic permeability	$\mathrm{H\,m^{-1}}$
n	Refractive index of the medium	
ρ	Reflection coefficient	
σ	Roughness of the surface	
$\mathcal{A}$	Ray amplitude	
k	Wave number	$\mathrm{m^{-1}}$
a_k	Unit vector along wave propagation	
a_R	Unit vector along ray propagation	
α	Elevation	deg
β	Azimuth	deg
ϕ	Phase shift	
γ	Aircraft direction	deg
E	Electric field	$\mathrm{V\,m^{-1}}$
F	Cost function	
a	Objective	
r	Restriction	
g_p	Penalty cost function	
g_a	Adaptive barrier cost function	
d_{ul}	Upper limit distance	m
d_{ideal}	Ideal distance	m
Δh_{ul}	Upper limit height variation	m
Δh_{ideal}	Ideal height variation	m
H	Hessian	
W	Inertia	
ΔT	Time variation	min
$N_{g[a,r]<1}$	Iterations to comply with restrictions	
$N_{\underline{g[a,r]}}$	Total number of iterations	
$\Delta \overline{P}_R$	Average received power increase	%

To my beloved wife Lina, and children.
No matter what, when, or where, I love you, always.

Chapter 1

Introduction

Modern cities are at the limit of their capacity in the horizontal plane. Many of them have a traffic problem that is highly complex to alleviate or solve. Urban air mobility is named the "third aviation revolution" [1]. Other alternatives such as collaborative driving or self driving vehicles have been proposed; nevertheless, the third dimension (the air) is still not used to its full potential. Thus, many hopes are placed in the urban air mobility to solve the mobility problem.

As per the aviation side, only in 2017, 3.5 billion passengers were using conventional aeroplanes. This high volume of passengers hints that there is a predisposition by the public to use such transport. The USA's department of defence estimates that by 2035, manned and unmanned vehicles will grow to 27000. These numbers include 8000 traditional aircraft, 14000 unmanned aerial systems (UASs) of all sizes, and 5000 vehicles with pilot augmentation technologies [2].

Even though this type of vehicle is studied for passenger transportation, these vehicles have various uses. Historically, UASs have found their primary use in the military context. However, recent advances in batteries, software, and the constant reduction of technology needs in size, weight, and power have made these vehicles available to a broader public. Nowadays, these vehicles are used in civil applications due to their ease of deployment, low maintenance cost, high mobility and ability to hover. The main applications of these vehicles can be condensed into four categories: search and rescue, coverage, delivery or transportation and construction, as stated in [3]. When connectivity is considered, there are only two categories; UAS acting as aerial base stations and UAS acting as cellular-connected users. Base stations in the air benefit countries with low coverage or damaged base stations after a natural disaster. This thesis focuses on those UASs acting as cellular-connected users since they have a higher number of applications.

The introduction of new vehicles into the airspace seems to solve many issues and brings new business opportunities. However, these vehicles require central management, especially if there is coexistence with traditional aerial system (AS) in the air. It is a well-known problem that air traffic management is a complex system that relies on human intervention. Current estimations show that the growth of human traffic is already beyond the capacity of a human-centred system [4]. With the addition of the new UAS in the ecosystem, it is paramount to introduce next-generation air traffic control (ATC). The new ATC system should include all types of AS independently of their control type. Moreover, separation of airspace by control type might lead to problems when the UAS operators are not fully aware of regulations and accidentally enter manned airspace [5].

A new kind of ATC that coordinates thousands of UAS and the already increasing number of traditional AS requires reliable communication links, particularly when used for aerial coordination. For example, an interruption of the link between the ATC and the UAS means the message will not be delivered [6]. Collision control
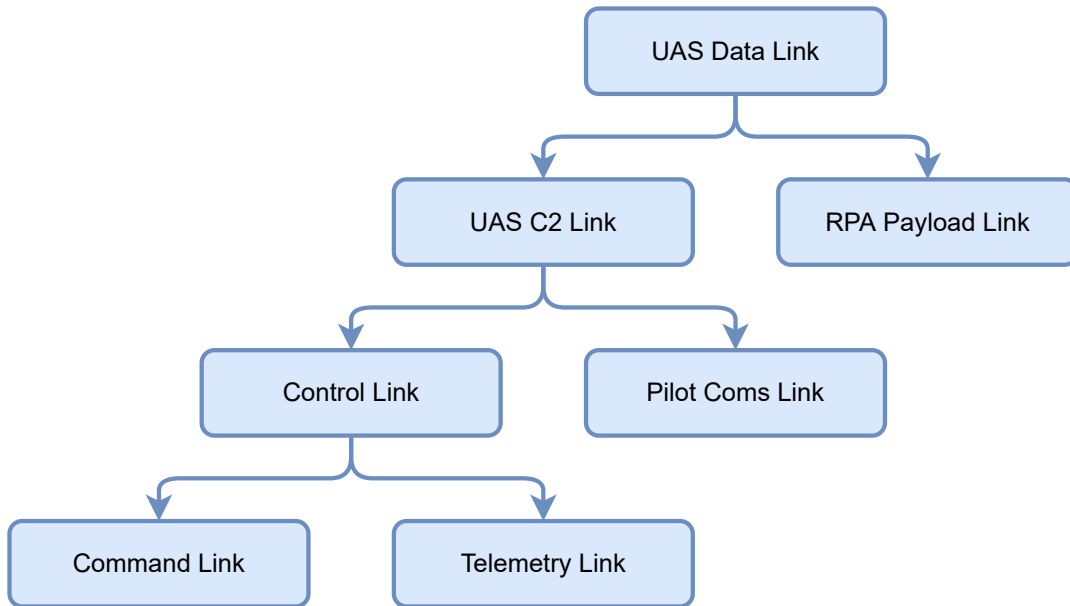
FIGURE 1.1: C2 link terminology

managed by ATC requires that messages are not lost or missed to ensure safe operations. Control links are a fundamental part of future ATC systems, and as such, their main characteristics must be: low latency, high reliability and security [7].

Control links are not the only ones conforming to the data link or air to ground (A2G) link. Payload data, which is not essential to maintain or control the flight, is sent in parallel. This data needs higher bandwidth than the bandwidth required required by a control link since it aggregates all the information gathered by the payload. This payload is usually composed of cameras to record or monitor events and sensors to sense the environment. On the one hand, the main advantage of payload communication is that if lost, it can always be recorded, stored in the UAS and analysed after the flight. On the other hand, as stated before, the command and control (C2) has higher demands for confidentiality, integrity and availability.

The C2 has two main components: pilot communications link and control link. The latter can be divided further into telecommand link and telemetry link. The first subdivision of the control link provides the flight information to the UAS, i.e., where to fly, how to fly and when to fly; it provides the function to modify the behaviour of the UAS. These orders are paramount to avoid conflicts or resolve those that can arise in a scenario with multiple players. Telemetry provides information to the ground management system and the state of the UAS at any given time; it provides the function for the UAS to indicate its state. Since the behaviour of the UAS can be predicted through different algorithms such as the Kalman filter, the loss of samples is not an issue. The main focus is then on the most stringent part of the C2, the telecommand link since it supports safety critical functions. Safety-critical functions require adequate performance to ensure a reliable connection between the controller and the UAS.

From a communication perspective, the C2 link operates based on communication transactions. The transaction begins with sending C2 information and completes when it is verified that the message has been received and interpreted and the required action is completed. The performance of the C2 link is described by the communication transaction time, continuity, availability and integrity, as explained

in [8]. To ensure safe flight operations is paramount to maximise the channel performance as it influences all C2 link parameters.

As of today, the regulation establishes that UAS must be controlled within line of sight (LOS) range. To overcome this limitation and to improve the performance of the C2 link over long distances, or beyond visual line of sight, is to use the existing mobile radio infrastructures to exchange C2 commands [9, 10] or to use satellite connections. Satellite connections are problematic due to the high latency when the connection is made to a GEO satellite or the high number of handovers when the connection is made to a LEO satellite. In addition, UAS have the advantage of having low size, weight and power (SWAP) characteristics; a satellite connection requires bigger and heavier equipment, making it not suitable for low SWAP platforms.

As mentioned earlier, mobile radio infrastructures can be exploited for UAS A2G communications. One of the general issues with mobile radio infrastructures is that these networks, independently of the provider, have been designed for ground usage. Implications are that cells, antennas included, are designed to serve a user that moves at ground level and has a specific mobility pattern. Since the UAS connects to a network not designed for its needs, these effects shall be considered. For example, unless the antenna is directed upwards, the UAS will be out of the main antenna beam at large elevation angles [11].

One way to characterise the radio frequency environment where the UAS flies is to perform measurements to accurately characterise the medium where the operations take place and develop specific models for each scenario where the UAS operates. For example, the channel will not perform identically when the UAS flies over an industrial area or airports when it flies over a small village with low buildings and when the UAS flies over a city with tall buildings. The situation will also change when the UAS flies together with many other UAS since the spectrum is limited. Spectrum management is out of the scope of this thesis; nevertheless, it is an aspect that needs to be considered since users on the ground and users on the sky will communicate in similar frequency bands.

In order to have an accurate prediction of the channel behaviour over different media, the adopted solution is to develop a deterministic channel model that analyses the type of terrain, profile of the terrain and even then presence of buildings [12]. A critical aspect introduced along with the channel model is the modelling of obstructions coming from the structure of the UAS itself. As hinted in Reference [13] and measured in Reference [14], these obstructions can cause a decrease of 30dB in some cases. The usual solution to these effects is to use antenna diversity, widely used in commercial AS, but an option not possible for UAS with stringent low SWAP requirements.

Many challenges exists to maintain a stable C2 A2G link between the UAS and the ground control. Aspects such as the environment, ground network characteristics and even the topology of the UAS must be taken into account to design procedures and routes to maintain a stable connection. Furthermore, in an UAS connected to a ground network, the trajectory of the UAS itself must be optimised considering key performance metrics such as throughput, energy and spectral efficiency and delay [10]. Howerver, with current air navigation systems, heavily based in predefined routes defined by established waypoints, the optimisation of the route does not have enough degrees of freedom to improve such connection metrics. The focus is then in future air navigation systems that move beyond the conventional waypoint-defined routes and enable an airspace where each UAS and AS can define its flying route.
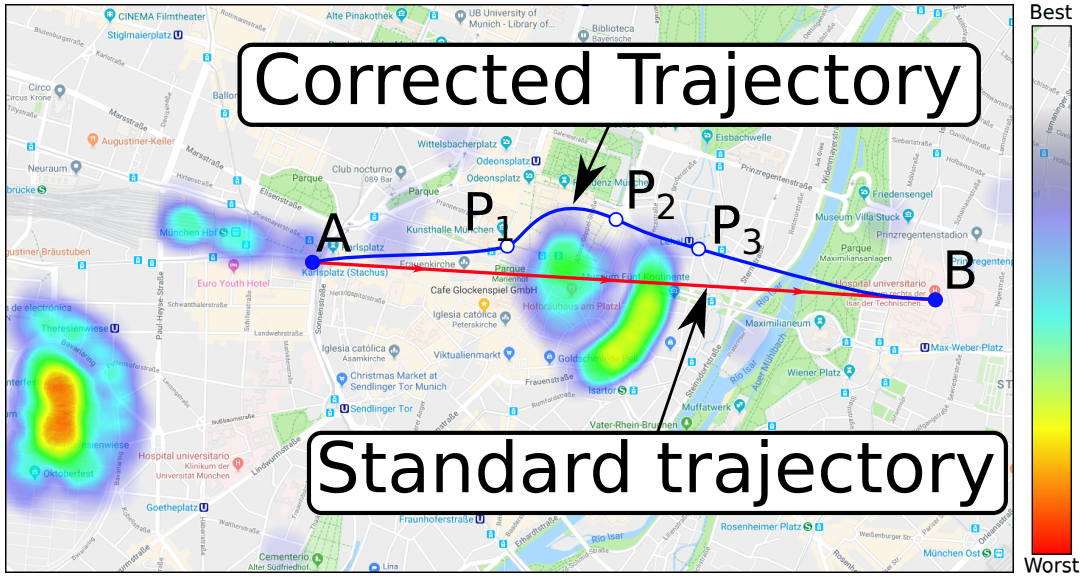
FIGURE 1.2: Corrected trajectory through optimisation vs initial trajectory. The optimisation will propose new waypoints for the unmanned aerial system to follow a trajectory with a better channel performance

With each UAS being free to fly wherever it needs to, situations as in Figure 1.2 will be more frequent. Figure 1.2 depicts the scenario where an UAS is provided with a route plan from *A* to *B*. The initial trajectory crosses several areas of low connectivity that could endanger the completion of the mission. Considering the channel's performance, a new route is designed to circumnavigate the areas of low connectivity. It is also important that the new proposed route does not increase the flight time. Therefore, the new routes must be designed to enhance connectivity and keep flight times close to optimal.

1.1 Contribution

As outlined in the previous section, the improvement of the channel performance for an UAS requires accurate knowledge of various elements. The required knowledge is: terrain elevation, terrain composition, UAS dynamic characteristics, presence of urban environments, UAS physical characteristics, characteristics of the elements in the C2 chain and last but not least, information published by the ATC regulating the airspace.

The enumerated information needs to be combined and processed so that the simulation can properly ingest it. The first contribution is then the description of a method to combine the required information. This data fusion is problematic as a topographic map is referenced in geographical coordinates, the model of the UAS is described by a set of triangles in a local frame of reference, and information from the terrain composition and urban areas will be extracted from areal or satellite images. Data fusion is then paramount, hence a topic present throughout this document.

Once the data is acquired and processed, the state of the A2G channel is required. For that, a deterministic channel model is proposed. The advantage of such model is that it can ingest the characteristics of its surroundings, hence applying to all kind of environments. A deterministic model eliminates the need for costly and time consuming measurement campaigns that will output characteristics that are dependent
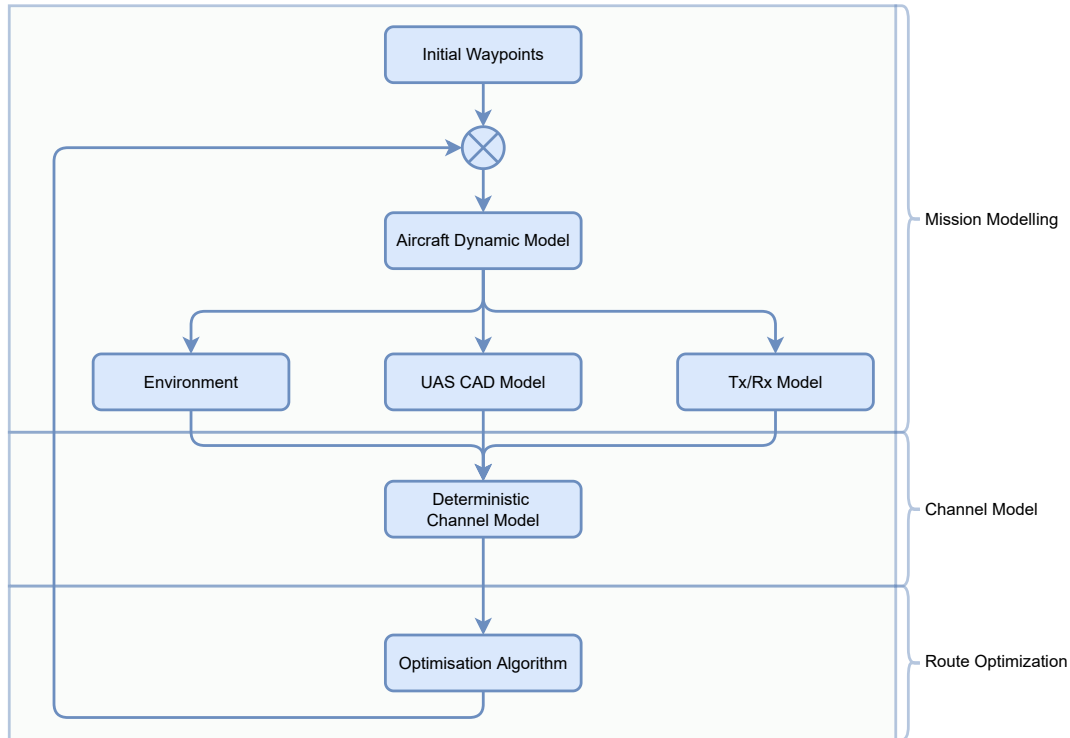
FIGURE 1.3: Outline

on a particular environment. With such a model, the optimisation of the channel performance gains flexibility and can be applied in more situations. The second contribution is then the proposal of a channel model able to ingest the characteristics of the surroundings and the characteristics of the UAS itself.

Once the A2G channel is fully characterised, a route can be designed to avoid those areas with poor C2 link performance. Which optimisation algorithm performs the best for this optimisation case is an open question and the main challenge of this work. There are many optimisation strategies and parameters that can be tweaked. The third contribution is to study and apply the best optimisation algorithms to find a better trajectory that respects all the constraints with the minimum number of iterations.

1.2 Objectives

The objectives to be covered by this thesis are three: the definition of the necessary tools for a successful optimisation, the design of a deterministic channel model to simulate the channel response and a route optimisation able to use the output of the deterministic channel.

1.3 Thesis Outline

This document is organised per chapters and each chapter is matched with an objective and a contribution. Chapter 1 is dedicated to the introduction, providing a historical framework and the main motivations that has lead to this thesis. After the framework is defined, the main expected contributions to the field are outlined. Finally, the chapter ends with the main objectives of the thesis.

Next three chapters and its relation to the overall optimisation process can be seen in Figure 1.3. The second chapter, named as "Mission Modelling", tackles the objective of defining the necessary tools for a successful optimisation. This can be seen in Figure 1.3 in the upper part of the optimisation process. A brief introduction is given to provide the necessary context to the chapter. After this introduction. A study of the state of the art is provided to outline what has been done in the field. This literature study allows identifying gaps in the literature, and differentiating this work. The main body of the chapter describes the necessary building blocks: environment information, aircraft model and communication systems.

Chapter 2 gives the necessary information used in the chapter 3, or "Channel Model". As in the previous chapter, chapter 3 starts with a brief introduction and a state of the art study. Since the number of references is high for this topic, they need to be filtered depending on the channel type. The chapter continues with radio wave fundamentals and channel modelling fundamentals. The chapter finalises with a description of the proposed channel model. This chapter further elaborates the second objective of the thesis. In the overall optimisation process (Figure 1.3), this chapter is the second box and directly provides the inputs to the optimisation algorithm.

The last chapter before presenting results is chapter 4, or "Route Optimisation". The chapter starts with an introduction to route optimization, followed by a description of the current state of the art. Next, it follows a description of the optimisation strategies used: direct search optimisation, nature inspired-optimisation and hybrid optimisation. Finally, after laying down the fundamentals, the optimisation strategy used is described together with the candidate cost functions. The algorithm constitutes the last operation performed in the route optimisation flowchart (Figure 1.3, and will decide whether new waypoints are proposed or the process is finished.

As said, chapter 5, or "Most suitable optimisation strategy", analyses the obtained results. This chapter contains 4 scenarios where each optimisation strategy will be applied and its results analysed. These 4 scenarios are specifically chosen to highlight various connectivity cases. The defined scenarios cover the following propagation environments: hilly, large water bodies and mixture of urban and rural areas. Each scenario poses a different challenge to the optimisation algorithm, not only limited to the propagation characteristics. For example, some scenarios have no fly zones (NFZs) while other scenarios force the optimiser to find a collision-free path with ground level.

Chapter 6, or "Conclusions", gives an overview of this work. The main objective of this section is to provide conclusions and the level of fulfilment of the defined objectives. The same chapter provides an acknowledgement of found limitations and the proposed recommendations for future work. Before ending the document, a statement of the originality of this work is provided together with an end note. Said end note gives the reader a final contextualisation of this work with respect to the body of knowledge.

References

[1] William B Cotton and David J Wing. "Airborne trajectory management for urban air mobility". In: *2018 Aviation Technology, Integration, and Operations Conference*. 2018.

[2] U.S. Department of Transportation. "Unmanned Aircraft System (UAS) Service Demand 2015-2035: Literature Review and Projections of Future Usage, Version 0.1". In: *Unmanned Aircraft System (UAS) Service Demand 2015 - 2035 - Literature Review and Projections of Future Usage* (2013).

[3] Samira Hayat, Evsen Yanmaz, and Raheeb Muzaffar. "Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint". In: *IEEE Communications Surveys and Tutorials* 18.4 (2016), pp. 2624–2661.

[4] Balakrishnan, K. and Polastre, J. and Mooberry, J. and Golding, R. and Sachs, P. "Blueprint for the Sky: The Roadmap for the Safe Integration of Autonomous Aircraft". In: Airbus. 2018.

[5] Dominique Meyer et al. "An air traffic control simulator for test and development of airspace management schemes". In: *IEEE Aerospace Conference Proceedings*. 2018.

[6] Daojing He, Sammy Chan, and Mohsen Guizani. "Drone-Assisted Public Safety Networks: The Security Aspect". In: *IEEE Communications Magazine*. Vol. 55. 8. IEEE, 2017, pp. 218–224.

[7] Hazim Shakhatreh et al. "Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges". In: *IEEE Access* 7 (2019), pp. 48572–48634.

[8] Sherif F Ali and Lee Nguyen. "UAS C2 data link performance for safe automatic flight guidance and control operation". In: *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*. 2016.

[9] Luis Afonso et al. "Feature article: Cellular for the skies: Exploiting mobile network infrastructure for low altitude air-to-ground communications". In: *IEEE Aerospace and Electronic Systems Magazine*. Vol. 31. 8. 2016, pp. 4–11.

[10] Mohammad Mozaffari et al. "A Tutorial on UAVs for Wireless Networks: Applications, Challenges, and Open Problems". In: *IEEE Communications Surveys and Tutorials* 21.3 (2019), pp. 2334–2360.

[11] Ruoyu Sun and David W Matolak. "Over-Harbor Channel Modeling with Directional Ground Station Antennas for the Air-Ground Channel". In: *2014 IEEE Military Communications Conference*. IEEE, 2014, pp. 382–387.

[12] Adrian Exposito, Dominic Schupke, and Hector Esteban. "Route Optimisation for Maximum Air to Ground Channel Quality". In: *IEEE Access* 8 (2020), pp. 203619–203630.

[13] David W Matolak. "Unmanned aerial vehicles: Communications challenges and future aerial networking". In: *2015 International Conference on Computing, Networking and Communications, ICNC 2015*. 2015, pp. 567–572.

[14] David W Matolak and Ruoyu Sun. "Air-ground channels for UAS: Summary of measurements and models for L- and C-bands". In: *2016 Integrated Communications Navigation and Surveillance (ICNS)*. IEEE, 2016, 8B2—-1—-8B2—11.

Chapter 2

Mission Modelling

The term "Mission Modelling" covers everything necessary to perform our defined mission; propose a route from point *A* to point *B* with a maximum performance of the air to ground (A2G) channel. The generation of this route requires accurate modelling of the environment where the unmanned aerial system (UAS) performs its flight. It is equally important to know how the UAS executes the flight.

Before defining the cost function or the optimisation algorithm, many elements and physical effects must be modelled and considered. One of the necessary elements to be modelled is the environment. The environment includes the elevation profile with artificial or natural features and terrain type. The next element is the accurate simulation of the dynamics of the UAS. These dynamic characteristics are used to generate an accurate route and are closely related to the characteristics of such UAS. Factors like propulsion type or shape of the UAS define how the aircraft can handle turns or ascensions.

The antenna radiation and reflection characteristics are necessary as well. This information does not define how the UAS flies a particular route, but it enables the simulation of the A2G channel. The optimisation algorithm requires an accurate state of the A2G channel at each trajectory point. Antennas deployed by the ground network and those installed in the UAS define the antenna radiation characteristic. Reflection characteristics are determined by the terrain type included in the characteristics of the terrain. The A2G channel model will use these elements to determine the link power.

One of the research objectives of this work is to model the communication channel. Apart from the propagation artefacts that may occur, the type of antenna and its radiation pattern is also part of this research. Every object included in the scenario has a position or coordinates, geographical coordinates define a map, an aircraft has a different axes convention, all this variety of coordinate systems need to be accounted for and explained.

The remaining chapter has the following structure: Section 2.1 describes the approach used to simulate a realistic environment, from terrain elevation to the type of terrain. Section 2.2 is devoted to the simulation of the UAS dynamics, along with the algorithm to detect structure shadowing. The chapter ends with Section 2.3 and describes the methodology used to obtain a realistic radiation pattern, close to what the UAS could encounter in an actual flight. Each section includes a revision of state of art. The most relevant literature and state of the art are treated separately for each topic. This separation is due to the diverse nature of the topics covered.
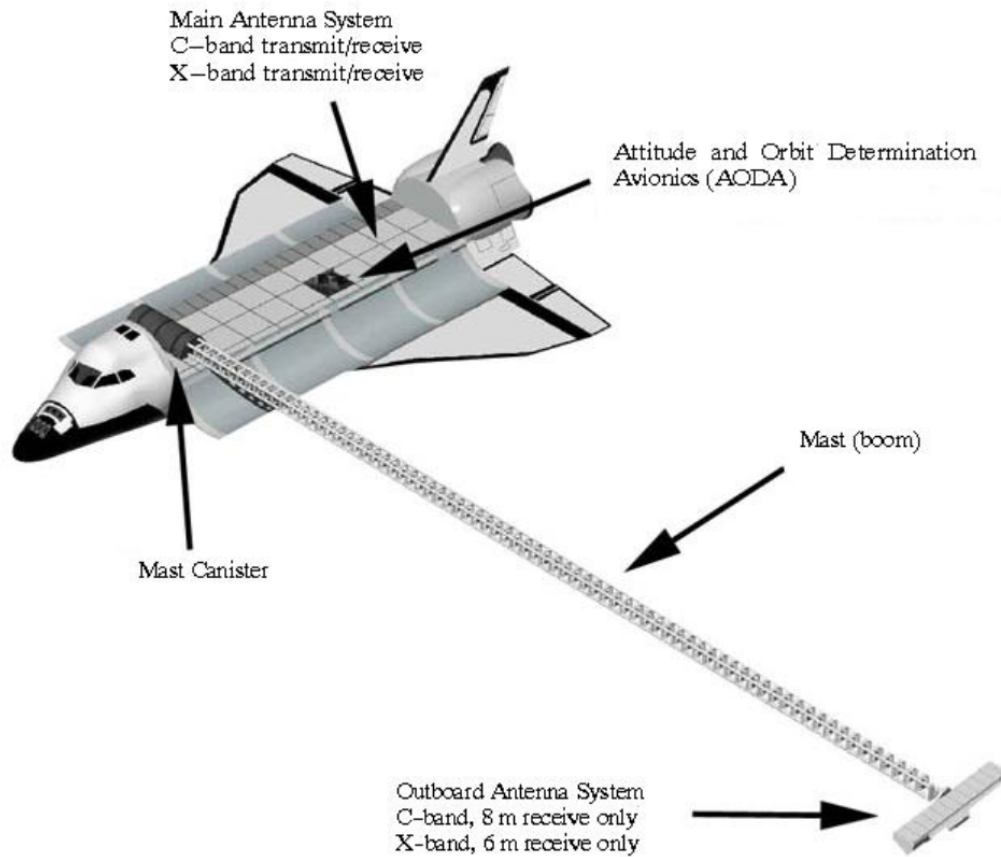
FIGURE 2.1: Schematic overview of the SRTM/X-SAR payload with deployed boom. (Image credit: NASA)

2.1   Environment

The environment is an element of the mission modelling that affects all levels. The environment constitutes all possible static objects that can influence the channel's response or the optimisation. On the one hand, the terrain elevation, terrain type and urban environment influence the channel. On the other hand, the optimisation is affected by the terrain elevation and urban environment; the aircraft must fly above these elements at all times.

The first element to define the environment is the elevation of such an environment. Digital elevation models (DEMs) are best suited for this task as their definition is: a 3D computer graphics representation of elevation data to represent terrain. These models are available on the earth, Moon, Mars, and other celestial bodies in the solar system.

Such a model of an entire planet and its accuracy is possible thanks to the synthetic aperture radar (SAR). The main characteristic of this form of radar is that it uses the motion of the antenna over the target to provide a higher resolution than conventional radars. There is a proportional relation between the size of the antenna and the resolution achieved.

SAR is the primary solution to achieve greater resolutions in space where space is a constraint. Figure 2.1 and Figure 2.2 are two different implementations of SAR. With two antennas, or two antennas mounted in different spacecraft, one antenna emits a microwave signal, and the other antenna receives the backscattered energy from the terrain [1].

FIGURE 2.2: TanDEM-X and TerraSAR-X in formation flight. (Image credit: DLR)

This technology has evolved from theory to a real solution with many applications since its first use, no more than two decades ago. It has research applications such as developing geopotential global models, evaluating glacier volume change, climate modelling, earthquake motion assessment, vegetation mapping, or flood inundation modelling. Reference [2, 3] cite numerous applications of this technology, such as research applications in geomorphology, climatology, oceanography and biodiversity.

Once the DEM accurately describes the shape of the terrain, the type of terrain is included as an additional dimension. Building height information can be superposed to the terrain information to calculate occlusions coming from such elements. There are mainly two players in collecting geographic information: Google maps and the OpenStreeMap (OSM) project. On the one hand, OSM relies on volunteers that add information to the base map; hence all the information produced is open and available to those who request it. On the other hand, Google maps is a private company that provides satellite imagery, aerial photography, street maps, real-time traffic conditions, and route planning. Even though there is a wide range of services, providing information or the generation of custom maps is not their business case.

One additional layer that can be included and studied in this document's scope is the generation of urban environments and the buildings that belong to the landscape. Figure 2.3 shows a virtually generated urban environment. Such urban elements enrich the channel simulation, especially when the simulation requires more than two rays. The design and implementation of a ray tracer require the consideration of reflections and transmissions ([4]) and the objects causing these reflections. The propagation effects of urban objects do not only consider the shape of the buildings. It also considers all structures' materials and surface behaviour ([5]).
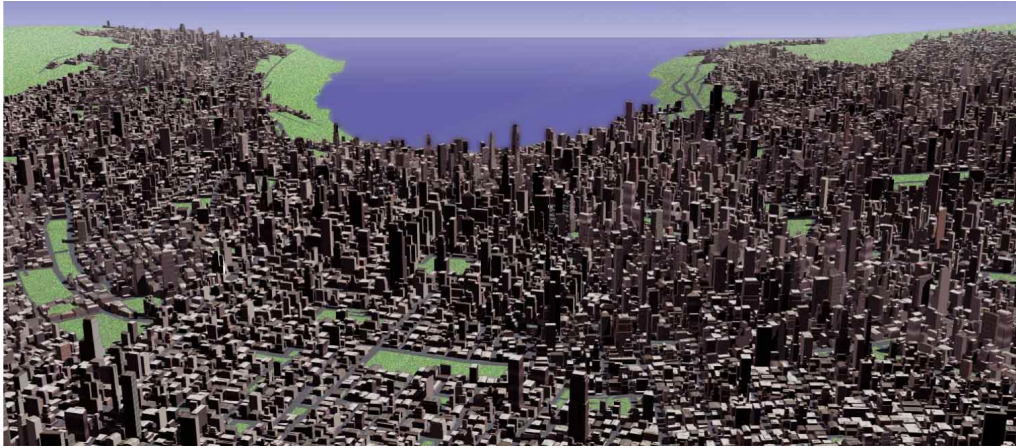
FIGURE 2.3: Virtual city. (Image credit: [6])

2.1.1   State of the art

In the field of SAR for DEM, there were mainly two datasets available: shuttle Radar Topography Mission (STRM) and advanced Spaceborne Thermal Emission and Reflection Radiometer (AESTER). The STRM was the first one to apply SAR in space. This mission mounted a C-band system extended by a passive antenna mounted on a 60-meter long boom, as shown in Figure 2.1. The mission spanned 11 days and covered the terrestrial surface between 60°N and 56°S. When the DEM was released, the accuracy was 1" for the United States of America and 3" for the rest of the world. It means the accuracy of 30 meters for the United States of America and 120 meters accuracy for the rest of the world. In 2014, this situation changed, as the U.S. government announced global 1" data [7, 8, 9, 3]. The dataset is hence available for download for educational use.

The AESTER sensor is currently mounted in the NASA spacecraft Terra, which will last until 2023. The sensor was built in Japan and is one of many sensors mounted in the spacecraft. The mission of AESTER is to provide high-resolution imagery with a resolution between 15 and 90 meters. The generated DEM is publicly available with 83°N and 83°S coverage [10].

These two datasets provide a global map with high resolution, but authors have found several issues with these two datasets. In Reference [3], the authors found that the STRM suffered from systematic positive biasses, especially in forested areas. In these particular areas, errors can be as high as 10 meters. Forests areas can be an issue, but resolution can be coarser than the promised 30 meters, as shown in [2]. On the same reference, one of the main takeaways is that errors exist due to mismatches in the photogrammetric process. Other problematic areas in the STRM and AESTER are high-relief terrain, sandy deserts and terrain with snow or ice [11].

terraSAR-X add-on for digital elevation measurement (TanDEM-X) is a twin satellite flying in close formation with the TerraSAR-X satellite launched in 2007, as shown in Figure 2.2. The main difference with the original TerrSAR-X is that the TanDEM-X incorporates additional cold gas propulsion and an S-band receiver. These two satellites form a unique single-pass SAR interferometer [12]. Since 2010, the TanDEM-X mission has generated consistent global DEM with unprecedented accuracy compliant with HRTI-3 specifications. From the launch of TerraSAR-X, the mission's high resolution, multi-polarisation, and multi-incidence characteristics have opened new applications for mapping and monitoring urban areas [13].
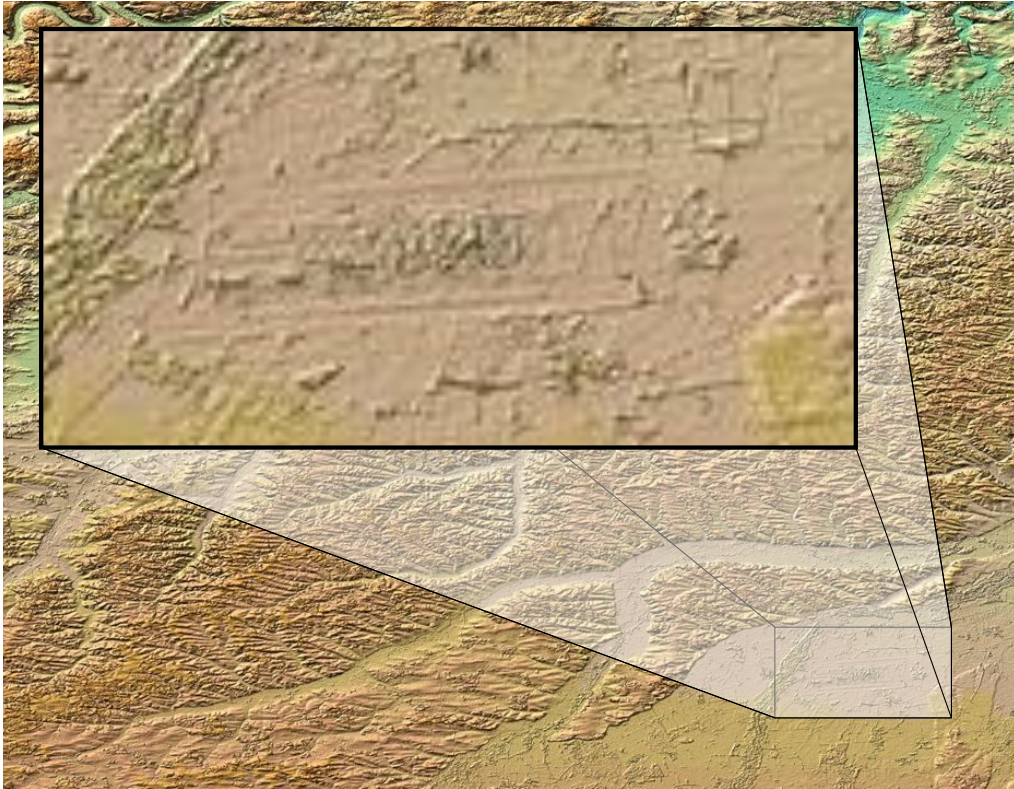
FIGURE 2.4: Level of detail of the DEM captured by the TanDEM-X
mission. The area corresponds to Munich, Bavaria, Germany where
Munich airport is clearly visible. (Image credit: DLR)

As with previous datasets, Grohmnann in [2] analyses available DEMs and compares their resolution. In this case, and as expected, the newest dataset originated from TanDEM-X shows the greatest detail and consistency level. The detail is above the one achieved by STRM and AESTER. Figure 2.4 shows the high detail obtained by using the TanDEM-X mission data.

Since Google Maps is a closed system with no interaction and influence from the scientific community, no scientific papers have been found addressing the quality of their data. Opposite to that, since OSM is a public system that relies on its volunteers to provide information, specific literature questions whether the information provided by OSM is accurate and sufficient.

Among the authors addressing the quality of volunteered geographic information services, Sehra, in Reference [14], states some of the main problems with these types of services based on volunteer input. In the mentioned paper, the first issue is data availability and coverage. No guarantee exists that OSM covers a specific area of interest. The following problem interlinks with availability; there is no knowledge about the contributors. Contributions to the map are performed through a web interface, sometimes without attributions. The skills and motivation of each contributor are unknown. Anonymous contribution can lead to vandalism; deleting correct landmarks and substituting by fraudulent information is a possibility and can be unnoticed for an extended time. Last but not least, Reference [14] highlights the lack of a unified standard, leading to misclassification and reduction in data quality. Mooney also shares this last concern in Reference [15].

Haklay provides some figures to the quality and coverage applied to London, in Reference [16]. As OSM started in London in 2004, it is the best candidate to measure

the expansion and coverage of the data. In four years, coverage reached 29% of the area of England, but approximately 24% are digitised lines without a complete set of attributes. Concerning accuracy, markers appear on average within 6 meters of the actual position. An interesting final thought is shared; participants are encouraged to contribute to a greater good, but also to an ecosystem of companies that benefit from this data.

There are mainly two methods for generating a virtual urban environment in the studied literature. The first method is to model a typical city based on statistical data such as land usage, typical building height and other clutter height [17]. Reference [6] proposes a system called CityEngine. CityEngine creates urban environments based only on statistical and geographical input data and is highly controllable. The urban environment shown in Figure 2.3 is one of the possible outputs of CityEngine.

The second approach appears described in Reference [18]. The urban environment is not obtained through statistics but by using the free geo-data from OSM. Data hosted in the OSM and particularly buildings are modelled as closed rings. Each vertex of these rings is mapped to a specific coordinate. Some, but not all, buildings have an associated height property. Statistics is a suitable tool to determine the heights of those buildings with none present in the system.

Studying the environment modelling literature gives an idea of the authors' approaches. The approach that follows aims at maximising the use of available sources. Using various sources means that the environment does not rely on a single source of information but combines multiple of them. The first layer comprises the elevation profile provided by the TanDEM-X missions' downloadable data. Terrain information composes the next layer. As described later, information is available via the Google application program interface (API) [19]. This API allows the download of images representing various ground compositions, later used to know the composition of the terrain. Last but not least, OSM is used to obtain information about building position and, in some cases, building height. For those cases where height information is not possible, the building height is modelled with a uniformly distributed height of 3 to 7 stories.

2.1.2   Choice of scenarios

The first step to defining a map is the choice of location. The desired features to be incorporated in the simulations are water bodies, mountains and large cities. Simulations use no fly zones (NFZs), whose data does not come from an existing database. This lack of data is the main reason why NFZ are discussed at a later stage in this document. Three regions accumulate the desired physical characteristics:

- The alpine region in Bavaria. This region is known for having a high variation of heights. The highest mountain in Germany (Zugspitze) is located in this area. The height variation is interesting, and it also offers many opportunities to test signal obstruction due to the terrain. As per terrain type variation, small urban areas should be surrounded by green areas and ground areas towards higher heights.

- Munich and its outskirts are the areas that can be populated with a rich ground network in terms of connectivity. The challenge for the UAS will be to find the best possible route out of a fair network situation. Expected terrain types are roads, city constructions, vegetation and water. The area can also include NFZ as there are no challenges due to height variation.

- Alicante and the nearby island of Tabarca: good network coverage can be found only on the mainland in this area, but the UAS will have to fly towards an island at a relative distance. It is interesting to test the optimisation in an environment where getting closer to the target means decreasing the received signal.

To obtain a clear a map with the desired color scheme, "Maps Static API" is the best candidate. This API returns an image (GIS, PNG or JPEG) in response to an HTTP request via URL. The HTTP allows the user to define the location of the map, size of the image, zoom level and map type. The combination of center location of the map plus zoom level sets the boundaries of the map. The zoom level for each map is the minimum value that allows to cover a wide enough area and the one that returns the most detail.

The Google API offers the user the opportunity to customise the style of different geographical features. The maps used by the channel model rely solely upon this feature. Any Maps Static API URL must have the basic format of Listing 2.1, which can then be tailored to display various map styles (Listing 2.1). Figure 2.6 depicts the different styles required for the simulation. The desired effect in this style is to obtain an image where only vegetation-covered areas are visible. The absence of black pixels means that no vegetation is present. Reference [19] provides the reader with more information about the API.

LISTING 2.1: Maps static API URL

```
https://maps.googleAPIs.com/maps/API/staticmap?parameters
```

LISTING 2.2: Style to highlight areas with vegetation

```
&style=element:labels%7Cvisibility:off&style=feature:administrative%7Celement:
    geometry%7Cvisibility:off&style=feature:administrative.land_parcel%7
    Cvisibility:off&style=feature:administrative.neighborhood%7Cvisibility:off&
    style=feature:road.local%7Celement:geometry%7Ccolor:0xFFFFFF&style=
    feature:road.arterial%7Celement:geometry%7Ccolor:0xFFFFFF&style=feature:
    road.highway%7Celement:geometry%7Ccolor:0xFFFFFF&style=feature:transit.
    line%7Celement:geometry%7Cvisibility:off%7Ccolor:0xFFFFFF&style=feature:
    transit.line%7Celement:geometry%7Cvisibility:off%7Ccolor:0xFFFFFF&style=
    feature:transit.station%7Celement:geometry%7Ccolor:0xFFFFFF&style=feature
    :landscape.man_made%7Celement:geometry%7Ccolor:0xFFFFFF&style=
    feature:landscape.natural%7Celement:geometry%7Ccolor:0x000000&style=
    feature:landscape.natural.landcover%7Celement:geometry%7Ccolor:0x000000
    &style=feature:landscape.natural.terrain%7Celement:geometry%7Ccolor:0
    xFFFFFF&style=feature:poi.attraction%7Celement:geometry%7Ccolor:0
    x000000&style=feature:poi.business%7Celement:geometry%7Ccolor:0xFFFFFF
    &style=feature:poi.government%7Celement:geometry%7Ccolor:0xFFFFFF&
    style=feature:poi.medical%7Celement:geometry%7Ccolor:0xFFFFFF&style=
    feature:poi.park%7Celement:geometry%7Ccolor:0x000000&style=feature:poi.
    place_of_worship%7Celement:geometry%7Ccolor:0xFFFFFF&style=feature:poi
    .school%7Celement:geometry%7Ccolor:0xFFFFFF&style=feature:poi.
    sports_complex%7Celement:geometry%7Ccolor:0xFFFFFF&style=feature:
    water%7Celement:geometry%7Ccolor:0xFFFFFF
```

Google does not provide the boundaries of the generated image. The algorithm defined by Zoharby in [20] is partially used to obtain the JPEG image and for the

---

Geographical information

- Lon
- Lat
- Z

Ground information

	R	G	B	Lat	Lon	Coeff
Road	1280x1280	1280x1280	1280x1280	1280x1280	1280x1280	1,6350
Railway	1280x1280	1280x1280	1280x1280	1280x1280	1280x1280	2,5000
Water	1280x1280	1280x1280	1280x1280	1280x1280	1280x1280	1,3335
Vegetation	1280x1280	1280x1280	1280x1280	1280x1280	1280x1280	1,2207
Ground	1280x1280	1280x1280	1280x1280	1280x1280	1280x1280	1,5460
City	1280x1280	1280x1280	1280x1280	1280x1280	1280x1280	2,0000

FIGURE 2.5: Data present inside each map file

geo-referencing of it. After testing other libraries, the one mentioned had the highest performance and code usability.

The data obtained from the TanDEM-X can be downloaded and accessed one unit of degree in latitude and longitude at a time. A map with these dimensions will not be used. A portion of the TanDEM-X map needs to be extracted to meet the map's boundaries. This process can be seen in Figure 2.7a, where the big map corresponds to one tile obtained from the TanDEM-X server, and the small rectangle is the boundary of the region where the simulation takes place. Linear interpolation is applied to adjust the grid to the newer boundaries of the map. Once the map's boundaries are defined, and height values interpolated, those values are saved into a file for later use.

After creating the map, all generated data is saved into a Matlab file to be reused multiple times. The file will contain the data structure displayed in Figure 2.5.

2.1.3 Urban environment

Currently, there is no worldwide database describing each building position and height. Some countries offer such information, some countries require special permissions or need to be requested, but there is no ordinary place to find it. Google maps can generate a realistic view of some cities based on aerial pictures. The main issue with Google is that no available API allows a standard user to access building height information.

The closest system or data base with similar information is OSM. It is a free, editable map of the whole world built by volunteers from scratch and released with an open-content license. Individuals, governments and commercial companies have already begun putting the data to use. Spite this effort and the growing use, there are many places that have not been mapped as the map relies on its volunteers. Some parts in the world have been mapped and are populated with data as can be seen in Figure 2.8. In those places where there is data available, it can be filtered and the coordinates that form building contours can be obtained. Still, the height information for each building is scarce and assumptions must be made.

The proposed approach represents buildings as polygons with a height associated with them. The height, when not available, is modelled using a uniform distribution of three to seven stories. Simulations occur in classical European cities,

(A) Satellite

(B) Structure

(C) Asphalt

(D) Vegetation

(E) Water

(F) Ground
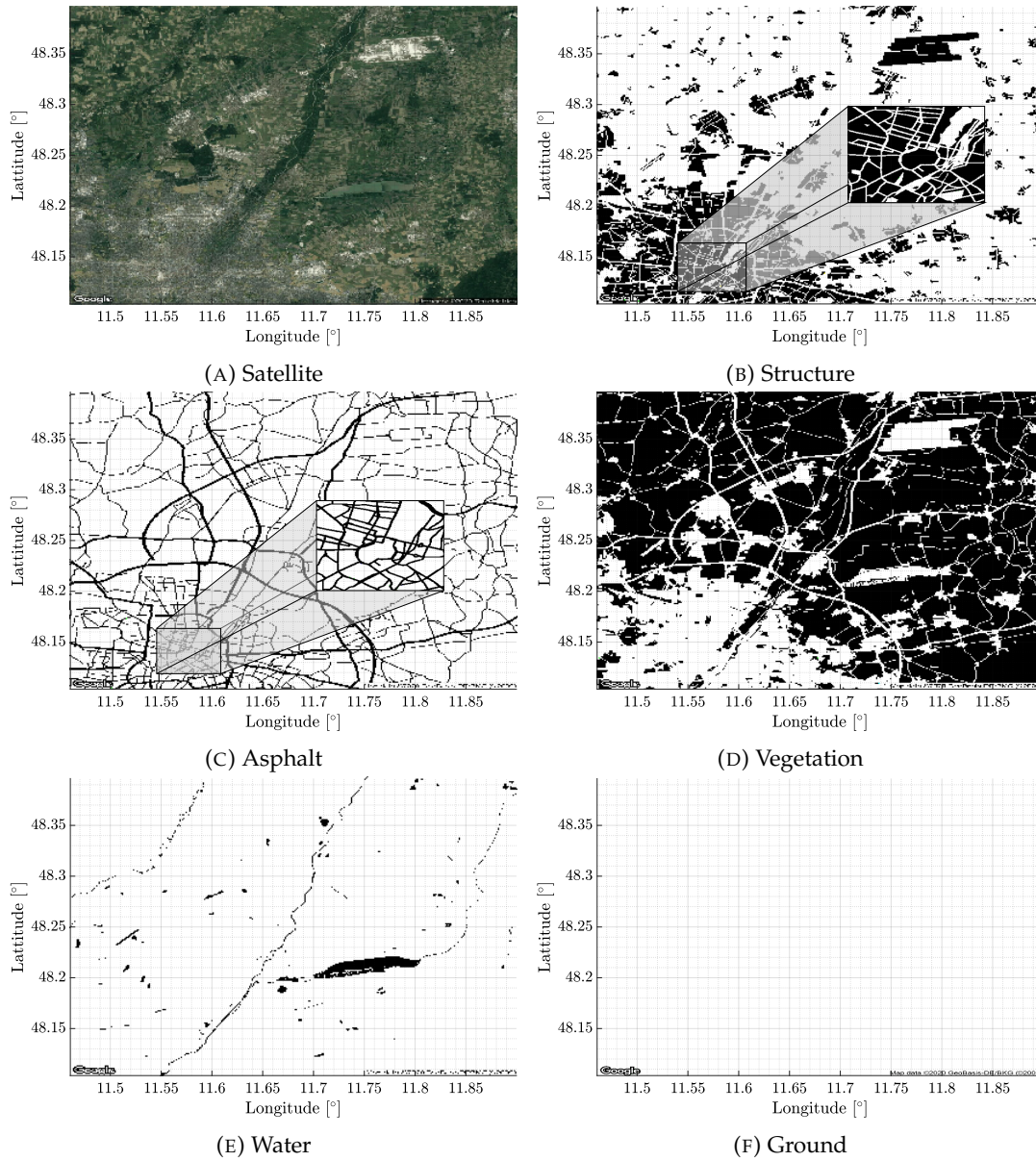
FIGURE 2.6: Variations of the same map used by the channel model
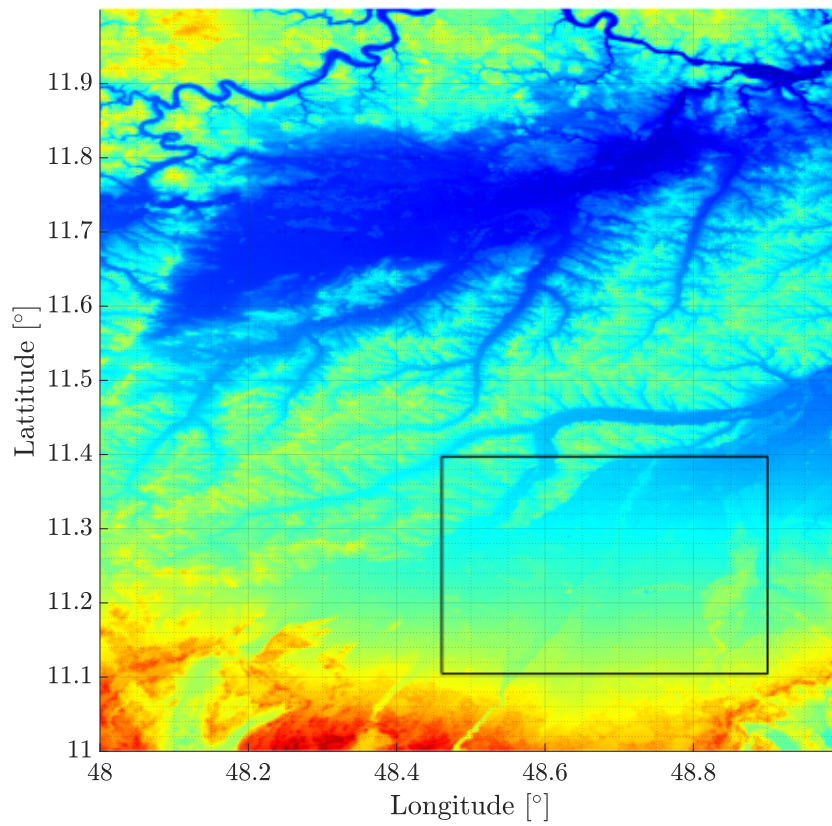
(A) Final grid vs terrain grid from above
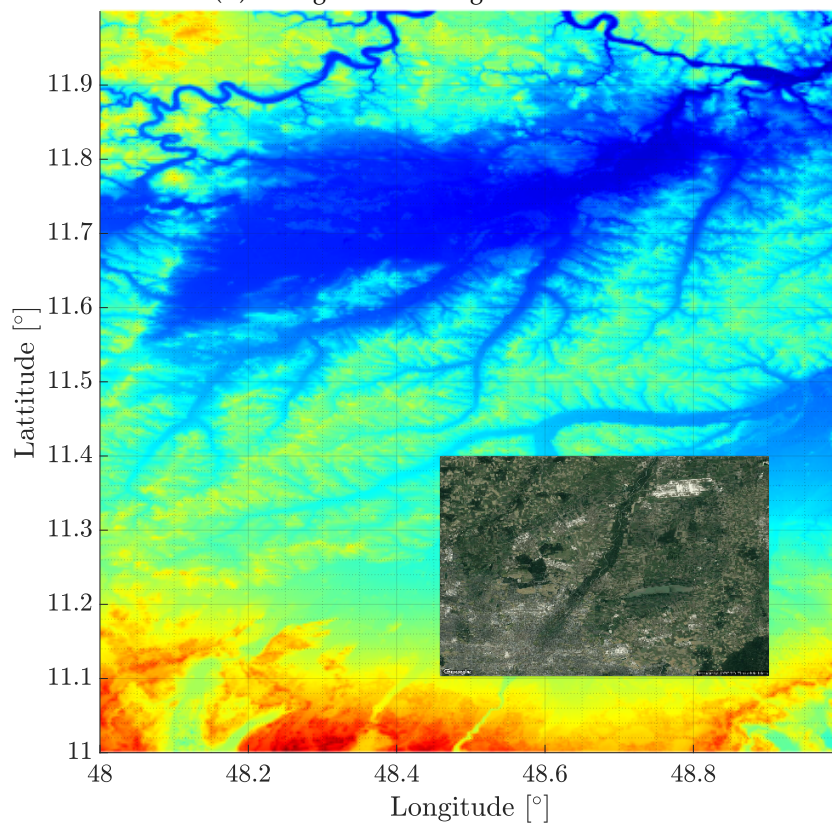
(B) Terrain plus image obtained from Maps Static API

FIGURE 2.7: Process of creation of a map ready for simulation
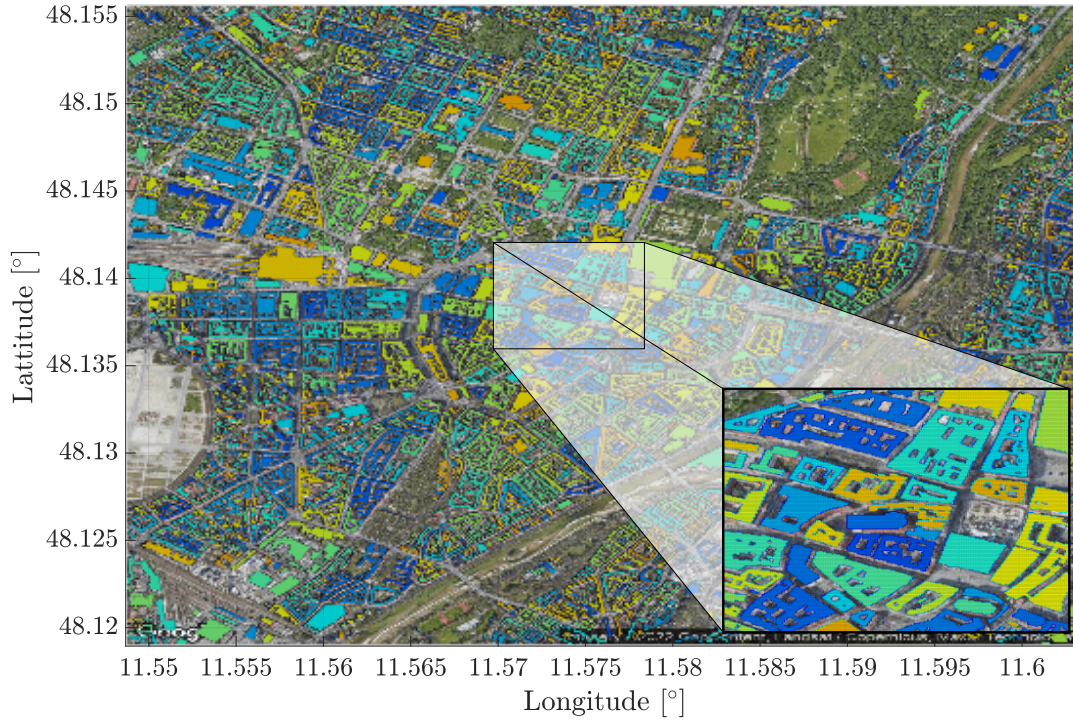
FIGURE 2.8: Building interpolated to grid positions

where the height of the building is regulated not to exceed seven stories. This approach adds a considerable amount of data to be tested. This amount of data will increase calculation times if used as it is. The approach is first to obtain the precise positions of the buildings and, secondly, to interpolate using the grid defined by the ground information (1280x1280 points). With this approach, the amount of points to be tested is reduced. Each building is treated as a bulge in the ground. Figure 2.8 shows how the building contours obtained from OSM overlay with the ground information.

2.1.4 Environment Coordinate system

Before the map can be used in the simulation, some operations need to be performed. The first is to transform from Lat/Lon coordinates to Cartesian coordinates. The map centre with sea level height is chosen as the $[0, 0, 0]$. This point is also used to reference all elements present in the scenario. Finding the reflections in the terrain, obstructions and angles of arrival is transformed into geometrical operations without further transformations in the channel model. One of the first and foremost assumptions to be taken is the consideration of the UAS as a rigid aircraft of constant mass, flying in a stationary atmosphere over flat, non-rotating earth as in [21]. This coordinate system is known as the fat earth frame (FEF).

The transformation from geodetic latitude, longitude and altitude to flat Earth position uses the Matlab function "lla2flat". This function has been available since release 2011A and is one of the most efficient transformation algorithms. The algorithm starts calculating the delta from the defined centre location (centre of the map):

$$d\mu = \mu - \mu_0 \qquad dl = l - l_0 \qquad (2.1)$$
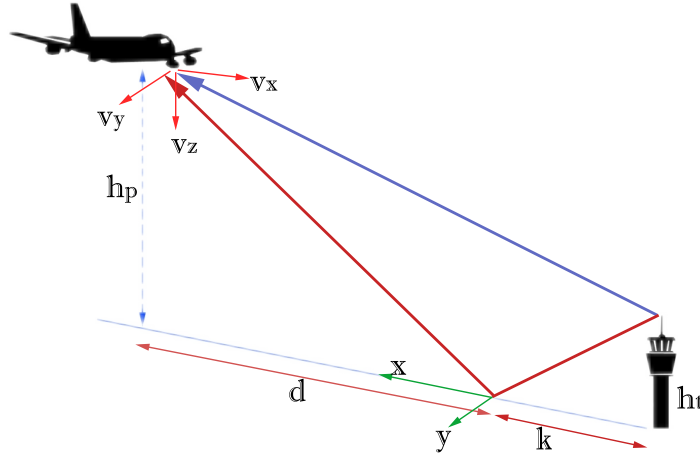
FIGURE 2.9: Calculation of the reflection point in flat terrain [22]

where μ is latitude, μ_0 is the center latitude, l is the longitude and l_0 is the center longitude. Two more intermediate operations are required before obtaining x and y values from the centre of the map. These values are the radius of curvature in the prime vertical (r_N) and the radius of curvature in the meridian (r_M):

$$Cr_N = \frac{\mathcal{E}_R}{\sqrt{1 - \left(2\mathcal{E}_F - \mathcal{E}_F^2\right) \sin^2 \left[\mu_0\right]}} \tag{2.2}$$

$$Cr_M = Cr_N \frac{1 - \left(2\mathcal{E}_F - \mathcal{E}_F^2\right)}{1 - \left(2\mathcal{E}_F - \mathcal{E}_F^2\right) \sin^2 \left[\mu_0\right]} \tag{2.3}$$

where $\mathcal{E}_R$ is the equatorial radius and $\mathcal{E}_F$ is the flattening of the planet. Both parameters can be found in the WGS84 ellipsoid planet model. Finally x, y and z values can be obtained as follows:

$$x = \frac{dl}{\arctan \left[\dfrac{1}{Cr_N \cos \left[\mu_0\right]}\right]} \tag{2.4}$$

$$y = \frac{d\mu}{\arctan \left[\dfrac{1}{Cr_M}\right]} \tag{2.5}$$

$$z = h + h_{ref} \tag{2.6}$$

2.1.5   Ground reflection calculation

Computing the reflection point in a flat terrain can be considered a straightforward process. Reference [22] gives the exact formulation to compute the distance from the emitter to the reflection point and the receiver to the reflection point. Although simple and elegant, this solution is no longer applicable once the terrain becomes irregular.
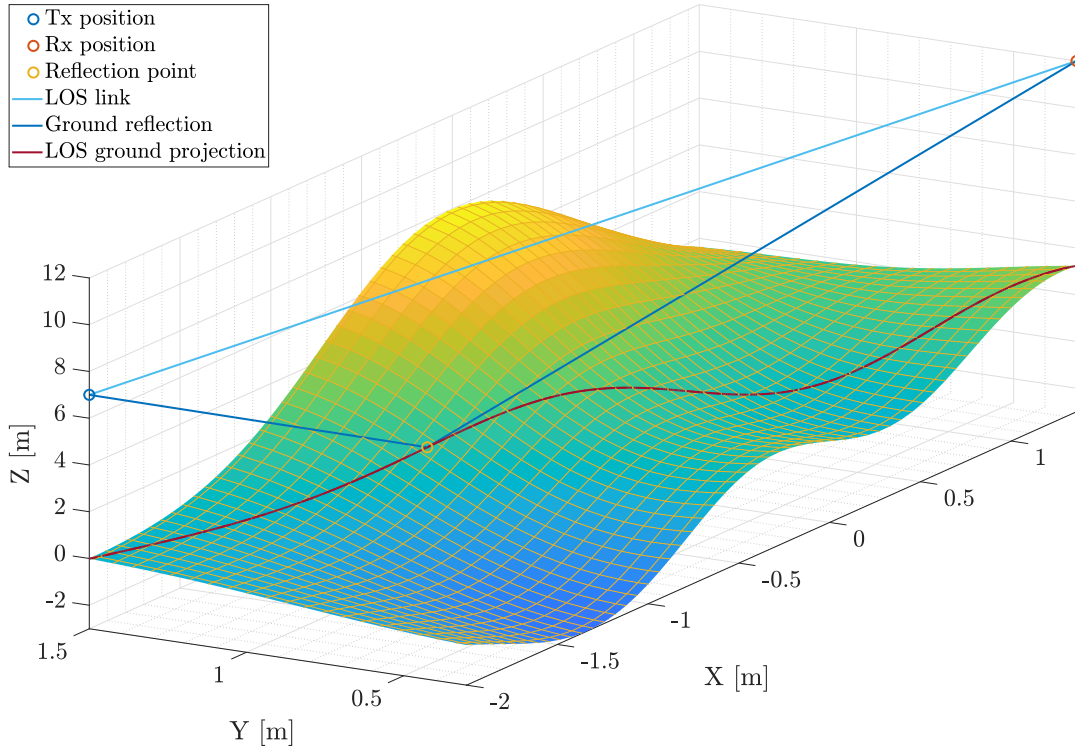
FIGURE 2.10: Three-dimensional view of a hilly terrain. A ground reflection is calculated based on the position of the sender and receiver.

$$d = \frac{h_p \cdot d_{LOS}}{h_p + h_t} \quad k = \frac{h_t \cdot d_{LOS}}{h_p + h_t} \tag{2.7}$$

Figure 2.10 illustrates the issues that Equation 2.7 could face. First of all, once the terrain is not constant, it cannot be assumed that the distance to the reflection point is only influenced by the height of both emitter and receiver. The second issue is to find the intersection point most efficiently. This operation is applied whenever the A2G channel is to be assessed. In the initial stages of the project, the terrain mesh was reduced to the minimum. The next step was to find the intersection point between the terrain and the line connecting the sender and mirrored emitter (negative height). Several complex steps are taken, which leads to higher computational times associated with this operation.

The most efficient algorithm for performing this operation involves several steps but relies on geometrical operations. The proposed algorithm does not rely on loops and sequential iterations, known for their inefficiency. From a graphical perspective, the calculation of the reflection point is similar to going from Figure 2.10 to 2.11. One dimension is eliminated in the two figures, and the problem becomes bi-dimensional. The first step is finding a function to compute and interpolate heights from *x* and *y* points efficiently. The most efficient interpolation process is found through trial and error and the use of code inspecting tools. The best method is to offline compute a three-dimensional interpolant with the whole grid of sample points. Once this is done, the sender-receiver link's ground projection (purple line in Figure 2.11) is calculated. Because the A2G link is known, the slope m is also known. This slope is used to rotate sender, and receiver positions, A2G link and A2G ground projection.
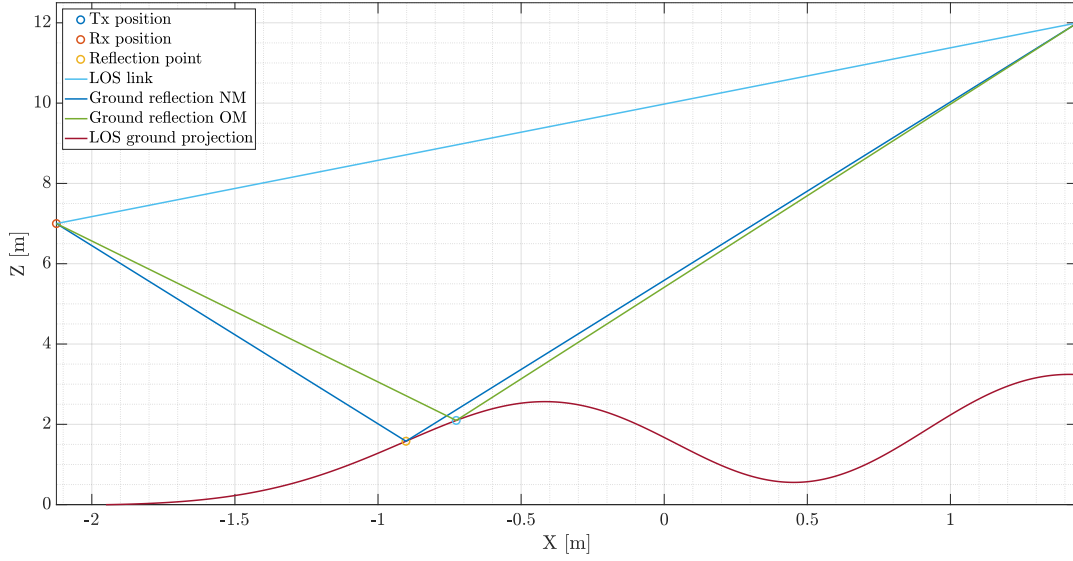
FIGURE 2.11: Two-dimensional view of a hilly terrain. The purple line represents the ground projection underneath the line of sight (LOS) link. Green and blue matches the two available methods to compute the reflection position.

Bi-dimensional problems are commonly simpler to solve. It is paramount to determine whether the terrain is in the middle of the A2G link before finding the reflection point. Since the ground projection is divided into segments, the fastest method is to determine if any of these segments interrupts the link geometrically. Equations 2.8 and 2.9 are equivalent representations of the same concepts. Both equations will provide the intersection point while being computationally efficient. The intersection between the two segments can also be checked with Equation 2.11. An intersection will be found when $0 \leq t \leq 1$ and $0 \leq u \leq 1$. This preliminary confirmation of collision is also used to avoid the application of Equation 2.9 if there is no collision.

$$P_x = \frac{\begin{vmatrix} \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{vmatrix} & \begin{vmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ x_4 & 1 \end{vmatrix} \end{vmatrix}}{\begin{vmatrix} \begin{vmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ x_4 & 1 \end{vmatrix} & \begin{vmatrix} y_1 & 1 \\ y_2 & 1 \\ y_3 & 1 \\ y_4 & 1 \end{vmatrix} \end{vmatrix}} \qquad P_y = \frac{\begin{vmatrix} \begin{vmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{vmatrix} & \begin{vmatrix} y_1 & 1 \\ y_2 & 1 \\ y_3 & 1 \\ y_4 & 1 \end{vmatrix} \end{vmatrix}}{\begin{vmatrix} \begin{vmatrix} x_1 & 1 \\ x_2 & 1 \\ x_3 & 1 \\ x_4 & 1 \end{vmatrix} & \begin{vmatrix} y_1 & 1 \\ y_2 & 1 \\ y_3 & 1 \\ y_4 & 1 \end{vmatrix} \end{vmatrix}} \qquad (2.8)$$

$$\begin{aligned} P_x &= \frac{(x_1 y_2 - y_1 x_2)(x_3 - x_4) - (x_1 - x_2)(x_3 y_4 - y_3 x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)} \\ P_y &= \frac{(x_1 y_2 - y_1 x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 y_4 - y_3 x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)} \end{aligned} \qquad (2.9)$$

$$L_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + t \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \end{bmatrix} \qquad L_2 = \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} + u \begin{bmatrix} x_4 - x_3 \\ y_4 - y_3 \end{bmatrix} \qquad (2.10)$$

$$
\begin{aligned}
t &= \frac{(x_1 - x_3)(y_3 - y_4) - (y_1 - y_3)(x_3 - x_4)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)} \\
u &= \frac{(x_1 - x_3)(y_1 - y_2) - (y_1 - y_3)(x_1 - x_2)}{(x_1 - x_2)(y_3 - y_4) - (y_1 - y_2)(x_3 - x_4)}
\end{aligned}
\tag{2.11}
$$

The last equations serve the purpose of determining terrain occlusion in 2D. The reflection point is the point in the ground projection from where the emitter and receiver are seen under the same elevation. This operation is relatively simple as it only involves trigonometric operations. The described approach can be used under the assumption that at least one point in the ground projection should meet the elevation condition. Each point of the ground projection is tested, the elevation to sender and receiver is calculated, and a vector of elevations is stored. The last step to obtaining the reflection point is to find the position with the same elevation in both vectors.

The described method applied to Figure 2.11 and 2.10 can find the point of equal elevation. Such point presents an elevation of 77.303 and 77.310 from sender and receiver, respectively. The delta between elevation values is only 0.006. In contrast, the traditional method with a mirrored position outputs an elevation of 74.097 and 77.635. The delta obtained with the old method is around 3.5 degrees.

2.2 Aircraft Model

Nowadays, simulation is a vast industry with applications from training to analytical requirements, according to [23]. Nevertheless, this industry has a long history ranging from the early days of flight. Back then, simulators introduced pilots into flight disorientation with the Antoniette simulator [24]. It was not until 1960 and the introduction of the first digital computers that this industry had significant developments. Even at the early stages of computation, those machines could compute the non-linear differential equations of flight. The computer can obtain a replica of the aircraft motion (linear and angular) throughout the flight envelope with sufficient accuracy through these equations.

Due to the increase of accuracy when obtaining flight envelopes, the objective of flight testing changed. Within a few decades, flight tests were used to verify simulation results and not verify aircraft systems' performance. It is no longer practical to train in the actual aircraft for safety and training effectiveness. However, despite the advances in the field, several open topics remain, such as data, parts and standards [23]. The industry's credibility has its roots in the efforts placed by a reduced number of airlines focused on establishing standards for flight simulation. Those companies could appreciate the many benefits of simulation: reduce design times, reduce design faults, cost-saving, and improve decision-making.

Simulating the dynamics of an aircraft independently of its configuration can be a PhD research on its own. The complexity of the models depends on the physical complexity of each phenomenon, the number of physical phenomena and the level of coupling of different types of physics [25]. A fixed-wing aircraft behaves differently than a rotor type aircraft; the aerodynamics and propulsion are not the same, leading to different behaviours. In order to simulate the dynamics, a commercial simulator that provides sufficient accuracy is used.
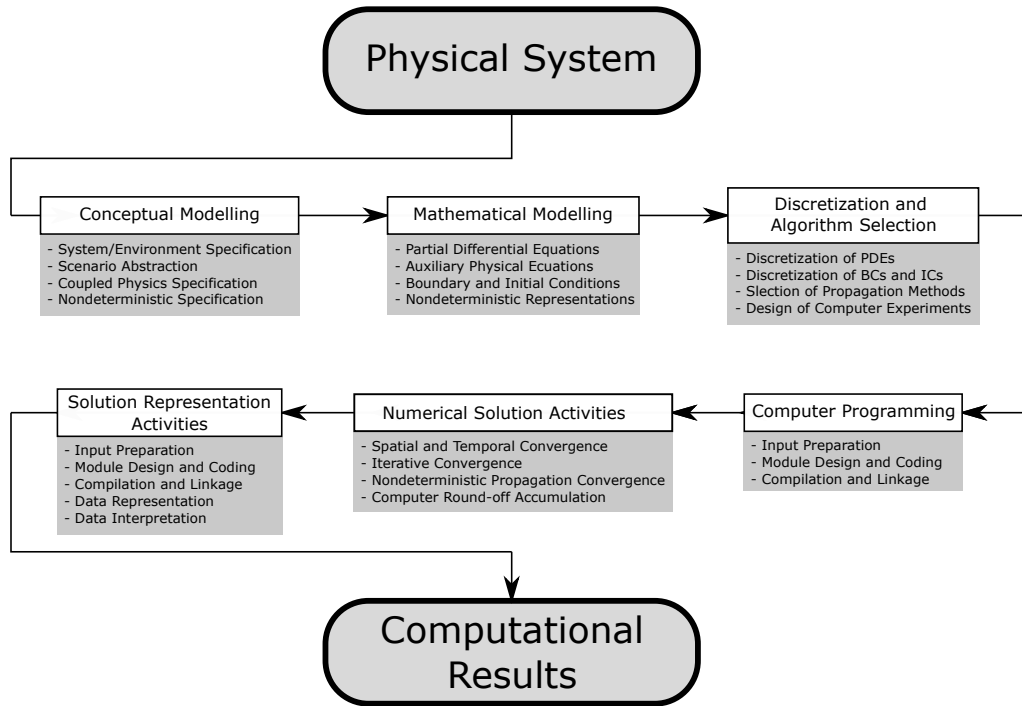
FIGURE 2.12: Necessary simulation steps as defined by [25]

The aircraft model is as important as the dynamics of it. It allows estimating the influence of the aircraft structure depending on the position of the communication system. Aircraft models and dynamics will tightly influence communication performance. The aircraft's attitude on its own is not enough to determine signal occlusions; the 3D model complements the necessary information to determine occlusions.

2.2.1 State of the art

There is no bibliography available on the pure development of a simulator. Nevertheless, literature applies various simplified flight dynamics simulation strategies or commercial of-the-shelf (COTS) simulation software. Authors in [25] propose a new framework for modelling and simulation composed of three phases: conceptual modelling of the physical system, mathematical modelling of the conceptual model, discretisation and algorithm selection for the mathematical model, computer programming, numerical solution and representation of the numerical solution as seen in Figure 2.12. This framework considers non-deterministic effects, uncertainties in the system, environment or human interaction with the system. This approach covers all steps, from the need to simulate the dynamics of an object to the representation of the solution.

The authors' approach in [26] is to model helicopters as a rigid body where the main rotor dominates vertical, pitch and roll dynamic while the tail rotor dominates the yaw dynamic. The various helicopter movements are modelled through this approach, but only one aircraft type is available for simulation. A different set of equations and aircraft data is required to model other aircraft types. The high number of equations, 33, highlights the complexity of each aircraft dynamic simulation.

Reference [27] provides a more direct coupling with this document, as its main focus is the simulation of quad-rotors. This type of UAS has been used in many

commercial applications, and there has been significant progress in control software and hardware. Nevertheless, prototyping and testing still pose a risk due to potential failures. According to the published paper in [27], the main tools used are Matlab/Simulink to simulate and calculate the dynamics, but other simulators exist, such as Flightgear. These approaches do not offer the possibility to test sensor-based high-level control. Because of this, the authors chose to implement the dynamics and control laws using Gazebo. This simulator offers the ability to simulate populations of robots in complex indoor and outdoor environments.

The aim in Reference [28] is to model a collision-avoidance system, not develop a simulation methodology, nor to develop and test the dynamics of an aircraft. The authors selected a COTS simulator to perform the simulation of the dynamics of the chosen UAS. The selected simulator is Analytical Graphics Inc. Systems Tool Kit (AGI-STK) [29], a platform used to analyse and visualise complex systems. The characteristics of AGI-STK will be developed further in this document as the same tool is used for simulating the UAS dynamics. One of the conclusions in [28] is that AGI-STK provides excellent results even when extensive simulations are carried out.

The system called TACAIR-SOAR [30] goes a step beyond the simulation of the dynamics of a UAS. The system, developed by the Air Force Research Laboratory in the U.S.A., can generate human-like behaviour for large scale, distributed military simulations. In this system, Lockheed-Martin Information Systems and BMH Associates generated dynamic models. Those models are defined by a standard generic model for each component. The model's parameters are adjusted to match each particular aircraft.

2.2.2 Aircraft dynamics

The various forces affecting the dynamics of an aircraft are the thurst, drag, lift and weight. Thrust is the forward force produced by the powerplant/propeller or rotor. This force opposes or overcomes the force of drag. Drag is the force caused by airflow disruption by the wing, rotor, fuselage, and other objects. Lift is the force produced by the dynamic effect of the air acting on the airfoil. Last but not least, weight is the combined load of the aircraft, crew, fuel and cargo or baggage.

Draft and lift calculations are the ones involving higher complexity. Atmospheric turbulence makes it impossible to follow a pre-computed plan. In this case, simplifications in terms of linear equations are required. More considerations that make the analysis non-trivial are extensively developed in Reference [31]. In a flight simulator, the numerous non-linear equations of motion are efficiently programmed and integrated through numerical algorithms. The resulting problem space ranges from 5 to 12 dimensions. The software model's data may be obtained from aerodynamic measurements or any other description of external forces. The main reasons to use a COTS simulator are:

- Finding an inertial reference frame to measure accelerations will be dealt with by determining an almost inertial reference frame.

- The mass of the UAS may become redistributed and changed. When a linearisation of the UAS model occurs, the effects of mass distribution change are often neglected.

- External forces and moments are not taken into consideration in linearised models. Forces such as propulsive forces or aerodynamic forces are difficult to predict.
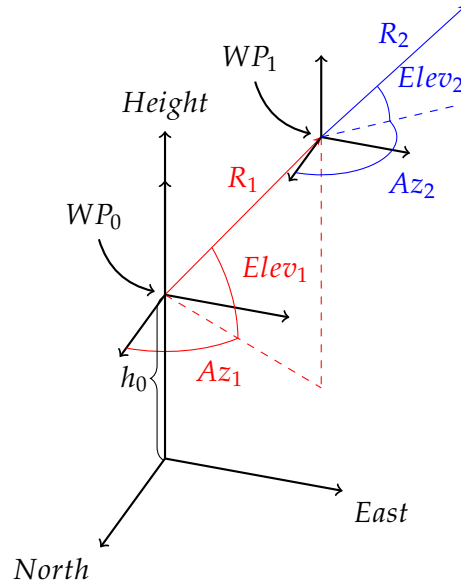
FIGURE 2.13: Azimuth (Az), Elevation (Elev) and Range (R) relation
with waypoints followed by the UAS. Initial waypoint is defined at a
certain height above the take off position.

- The solution of the dynamics equations for positions and velocities is complex, even for linearised systems. Multiple relationships exist among the involved variables. Often analytical solutions are not available, and approximations are necessary to obtain the position and velocity of the UAS.

AGI-STK [29] is the selected simulator to obtain the UAS´s position and behaviour through the proposed route. As in our previous work in[32], to ensure a compromise between granularity and calculation times, the route is sampled 12 times per minute. Due to this sampling rate, the number of points per route (M) will always be greater than the number of waypoints (N). Position (x, y, z), attitude (Φ, Θ, Ψ), and speed vector (Vx, Vy, Vz) are computed and collected in L_i and $\mathcal{L}^L$. L_i represents the state vector of the UAS on a given time. As in Equation 2.13, $\mathcal{L}^L$ represents the UAS's location, attitude, and speed during the flight. The subscript L in $\mathcal{L}^L$ alludes to the representation of the path $\mathcal{L}$ by the state vector values. Equation 2.14 is a different representation of $\mathcal{L}$, this time based on the waypoints flown. The step from waypoints to state vector is deterministic and both can be considered similar representations of the path.

$$L_i = \left[x_i, y_i, z_i, \Phi_i, \Theta_i, \Psi_i, v_{x_i}, v_{y_i}, v_{z_i} \right], \quad i \in [1, M] \tag{2.12}$$

$$\mathcal{L}^L = \{ L_1, L_2, \ldots, L_M \} \tag{2.13}$$

2.2.3 Route generation

One of the already defined goals is to establish a method to optimise a route based on the channels' quality. The state vector $\mathcal{L}^L$ is only generated after the UAS has been given a route to follow, a route that is mainly composed of certain key points named waypoints. A waypoint is a point that the UAS must pass through or in certain cases, circumnavigate. The most common way to define those waypoints in the aeronautics industry is by their latitude, longitude and height.

Defining the route by latitude, longitude and height will lead to an infinite search space. This definition increases optimisation time without an appreciable improvement of results. The proposed alternative is to parametrise the route by a succession of azimuth (Az), elevation (Elev) and range (R) values. This approach reduces the search space, but it causes that changes on a waypoint alters the complete route. Figure 2.13 displays how the waypoints are calculated and how each one is related to the previous one.

The proposed method and relationship between waypoints is already defining the type of UAS used in simulations. Vertical takeoff landing aircrafts have the availability to perform flights without a runway; helicopters belong to this aircraft class. The ability to take off and land vertically is reflected in Figure 2.13 in the initial height of WP_0. These types of UASs will move vertically to a certain height (150 m) before starting the trajectory.

The set of waypoints is expressed in Equation 2.14 they consist of one starting waypoint, a final waypoint and a set of N intermediate waypoints. It has been concluded that 5 intermediate waypoints are sufficient for a successful optimisation. These 5 waypoints turn out to be 15 values to be optimised in order to find a feasible route. The values of Az, Elev and R are then the design parameters and are limited with upper and lower boundaries. The values for Az are ± 30 degrees and Elev are ± 5 degrees.

The boundaries for R need the definition of $\mathcal{L}_{min}^{WP}$ as in Equation 2.15. This route only uses the starting and ending waypoint, the UAS will fly from WP_{start} to WP_{end}. The minimum value for R then is 1 km. The maximum value for R is one third of the distance that the UAS will cover on the straight line route. Path $\mathcal{L}_{max}^{WP}$ is the least efficient path the UAS will fly as it uses maximum values of Az, Elev and R for each waypoint. The representation of $\mathcal{L}_{max}^{WP}$ can be found in Equation 2.16.

$$
\mathcal{L}^{WP} = \left\{ \begin{array}{c} WP_{start} \\ WP_1(Az_1, Elev_1, R_1) \\ WP_2(Az_2, Elev_2, R_2) \\ \vdots \\ WP_N(Az_N, Elev_N, R_N) \\ WP_{end} \end{array} \right\}
\tag{2.14}
$$

$$
\mathcal{L}_{min}^{WP} = \left\{ \begin{array}{c} WP_{start} \\ WP_{end} \end{array} \right\}
\tag{2.15}
$$

$$
\mathcal{L}_{max}^{WP} = \left\{ \begin{array}{c} WP_{start} \\ WP_1(30, 5, |\mathcal{L}_{min}^{WP}|/3) \\ WP_2(30, 5, |\mathcal{L}_{min}^{WP}|/3) \\ \vdots \\ WP_N(30, 5, |\mathcal{L}_{min}^{WP}|/3) \\ WP_{end} \end{array} \right\}
\tag{2.16}
$$

2.2.4 3D aircraft model

The model used for simulations has evolved through time. The model started with a set of polygons, small enough to compute shadowing from the UAS frame as in Figure 2.14. This model was thought for the first simulations and definition of the
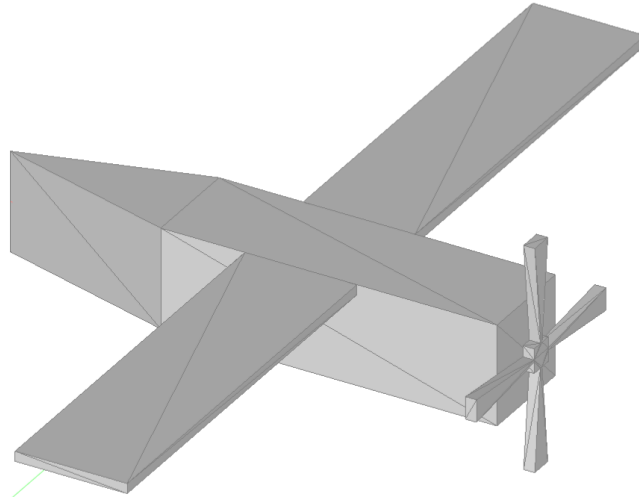
FIGURE 2.14: The first model used for simulations. This model is
designed to be as simple as possible with the shape of an aerial vehicle
propelled by a turboprop engine.

shadowing algorithm. Reference [33] links to the model used for simulations. This
model is composed of 1960572 vertices and 653524 faces. This detail implies a high
computational effort, which is not reflected in better results.

A wide variety of software can be used to reduce the geometric complexity of
the 3D model. MeshLab [34] offers different ways to decimate triangulated surfaces
while preserving geometrical detail. This free software offers different subdivision
schemes remeshing and resampling filters. Meshlab allows the user to select the tar-
get objective or select percentage reduction of triangles. Some level of supervision is
required as an oversimplification will lead to a model that is not suitable for calcu-
lations. Figure 2.15b shows the result after simplification of the mesh. In this case,
the mesh has been reduced to 4107 vertices and 1369 faces, only a 0.2% of the initial
mesh size. It is shown that using the right tools and processes, the size of the mesh
can be reduced without damaging the overall shape. Although the route optimisa-
tion benefits from this reduction, other applications will require a model as close to
reality. These fields include computational fluid dynamics, aerodynamics and heat
transfer simulations.

2.2.5   Obstruction calculation

The calculation of any obstruction caused by the UAS structure and its manoeuvres
uses the basic principles of Ray Tracing. This technique for modelling light propaga-
tion has seen a significant impulse thanks to recent developments in dedicated ray-
tracing acceleration hardware [35]. NVIDIa, in its new architecture for ray tracing,
has designed a set of processors assigned to each consuming task. One processor is
dedicated to finding what element in the scenery the ray hits, freeing the streaming
multiprocessor.

As mentioned in [35], there are three fundamental questions that any ray tracing
algorithm must answer..

- Ray direction: The direction will be known, as the channel is modelled using
  the LOS and reflected rays. Only two directions are tried on the modelled
  channel.
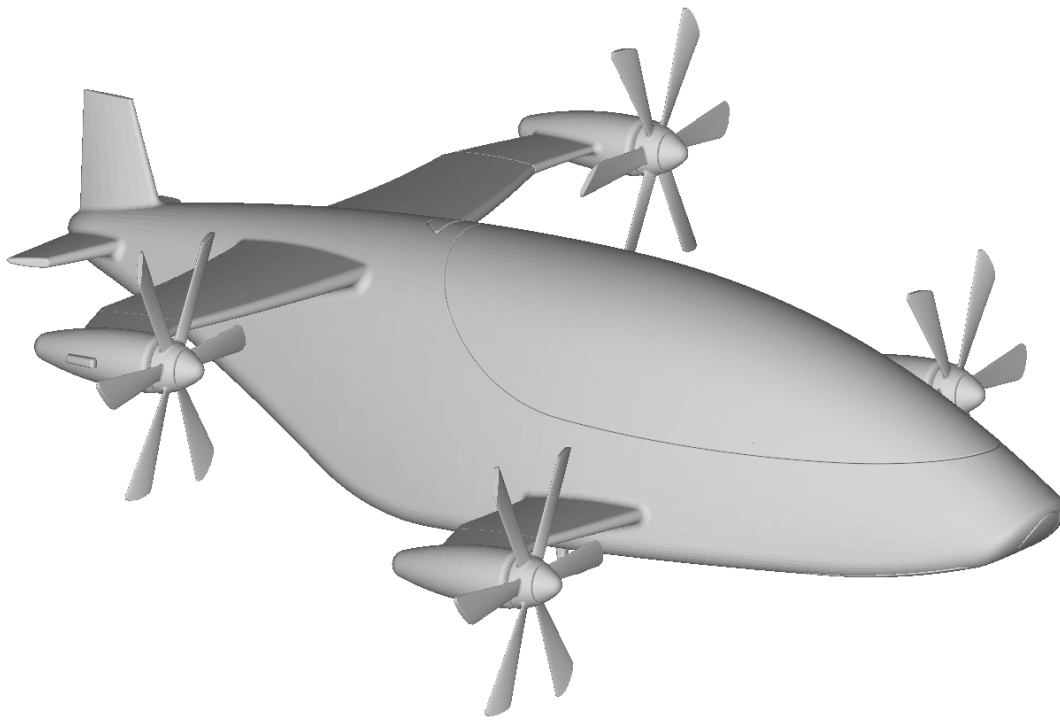
(A) Full model with 1960572 vertices
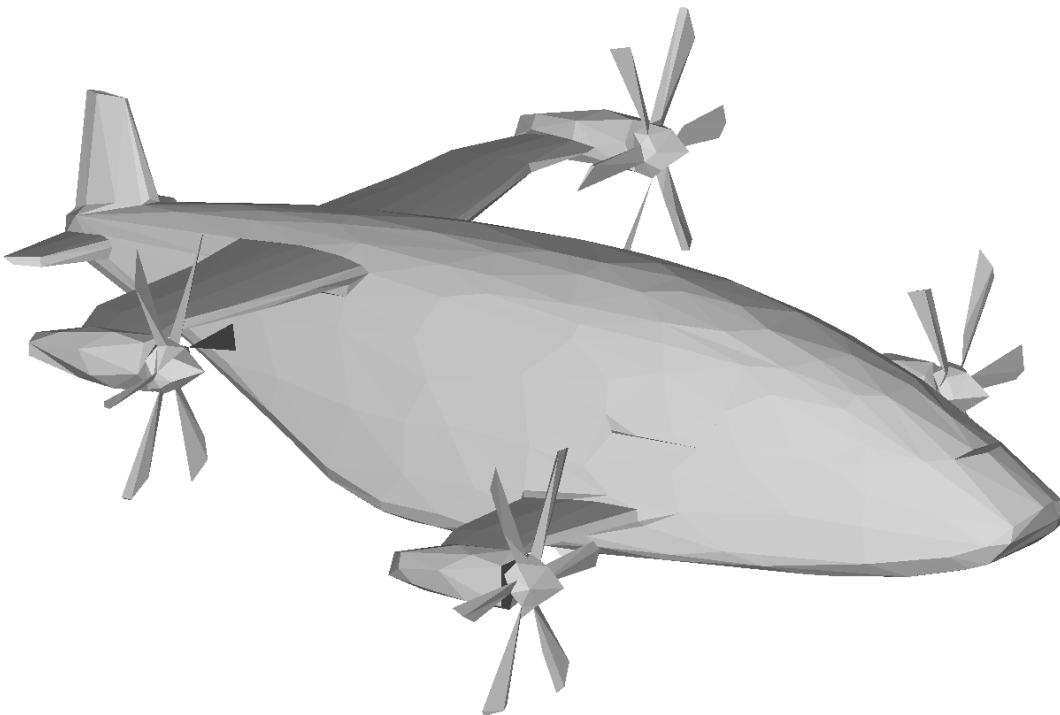
(B) Reduced model with 4107 vertices

FIGURE 2.15: UAS frame model comparison
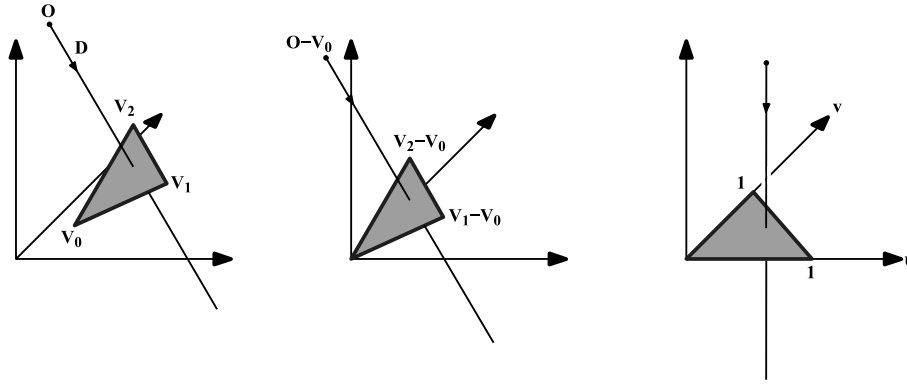
FIGURE 2.16: Translation and change of base of the ray origin [36]

- Hit test: This is answered by checking if the ray hit either the UAS frame or some element in the ground.

- Light scattering at hit point: Scattering is not modelled in the UAS frame. Attenuation is modelled when the ray hits an element of the environment as discussed in Subsection 3.4.

For calculating the obstruction caused by the UAS, only the second question is addressed. The direction to shoot will be provided by the reflection calculation and the LOS ray. As the UAS model is composed of triangles, a fast calculation algorithm is required. For this task, the most commonly used is the one proposed by Thomas Moller in [36]. This algorithm is remarkably efficient for triangles without pre-computed plane equations.

The proposed algorithm only requires the vertices of the triangle to be stored. To further simplify the process, the model is simplified to reduce the number of triangles to be computed. Authors promise 25% to 50% memory savings depending on vertex sharing.

Intersection algorithm

The first step is to define a point T on a triangle based on its barycentric coordinates u and v:

$$T\left[u, v\right] = (1 - u - v)\,V_0 + uV_1 + uV_2 \qquad (2.17)$$

and three main conditions to be fulfilled:

$$u \geq 0 \qquad v \geq 0 \qquad u + v \leq 1 \qquad (2.18)$$

The intersection between a ray $\mathcal{R}\left[d\right]$ and the triangle $T\left[u, v\right]$ is computed as follows:

$$O + dD = (1 - u - v)\,\mathcal{V}_0 + u\mathcal{V}_1 + u\mathcal{V}_2 \qquad (2.19)$$

$$\begin{bmatrix} -D_1 & \mathcal{V}_1 - \mathcal{V}_0 & \mathcal{V}_2 - \mathcal{V}_0 \end{bmatrix} \begin{bmatrix} d \\ u \\ v \end{bmatrix} = O - \mathcal{V}_0 \qquad (2.20)$$

geometrically this can be interpreted as the translation of the triangle to the origin and its transformation to a unit triangle. With this translation and transformation, the ray is aligned with the x. The process is well illustrated by [36] in Figure 2.16.

The terms in Equation 2.17 can be arranged in matrix form:

$$E_1 = \mathcal{V}_1 - \mathcal{V}_0 \qquad E_2 = \mathcal{V}_2 - \mathcal{V}_0 \qquad T = O - \mathcal{V}_0 \tag{2.21}$$

$$\begin{bmatrix} d \\ u \\ v \end{bmatrix} = \frac{1}{\begin{vmatrix} -D_1 & E_1 & E_2 \end{vmatrix}} \begin{bmatrix} \begin{vmatrix} T & E_1 & E_2 \end{vmatrix} \\ \begin{vmatrix} -D_1 & T & E_2 \end{vmatrix} \\ \begin{vmatrix} -D_1 & E_1 & T \end{vmatrix} \end{bmatrix} \tag{2.22}$$

Equation 2.22 can be further simplified with the introduction of new components:

$$P = D \times E_2 \qquad Q = T \times E_1 \tag{2.23}$$

$$\begin{bmatrix} d \\ u \\ v \end{bmatrix} = \frac{1}{P \cdot E_1} \begin{bmatrix} Q \cdot E_2 \\ P \cdot T \\ Q \cdot D \end{bmatrix} \tag{2.24}$$

An actual software implementation requires a simple test to ensure numerical stability. The product $P \cdot E_1$ is compared with a sufficiently and adequately small ϵ value at each calculation. If the $P \cdot E_1$ is less than ϵ, the intersection will not occur as the ray is parallel to the triangle's plane. The obtained u and v must follow the conditions defined in Equation 2.18.

UAS body masking

The UAS model itself is not changing over time. This lack of change means that the triangles defining the shape of the UAS will not change its shape, only the relative position to the centre of the map. The simulation starts with the computation of a body mask. With this intermediate step, the simulation will not use the intersection algorithm at each simulation step. The result is a matrix representing the possible angles of arrival to the antenna in the UAS.

The first step is to reference the vertex forming the UAS to the antenna's position. This step translates the centre of coordinates from the geometrical centre of the UAS to the centre of the antenna in the aircraft. The rotation matrix is then applied to each vertex of the aircraft.

After the coordinates origin has been translated to the antenna position, a further coordination transformation is applied. This time, spherical coordinates give information about the minimum and maximum elevation and azimuth angles where a vertex of the UAS is seen. This information is paramount as it significantly reduces the number of points in the matrix. Intervals of elevation and azimuth with UAS vertices are sampled with more points (18 azimuth and 36 elevation points). Simulation only uses 4 points to cover intervals with no UAS points.

Spherical coordinates have three dimensions: azimuth, elevation and range. So far, only the first two have been discussed, and range determines if an azimuth and elevation value corresponds to UAS or not. If the range value is equal to a predefined value, this direction has no UAS triangle in the middle. If the range value is smaller than a predefined value, a point in the UAS structure has been hit. This direction is then recorded and is considered as airframe shadowing.

As discussed, directions with no obstructions return a constant distance. Obstructed directions will return the distance from the antenna position to the frame's close obstruction point. The resulting body mask can be seen in Figure 2.17, with the left (up left), front (upright), down (down left) and side (down right) perspectives.
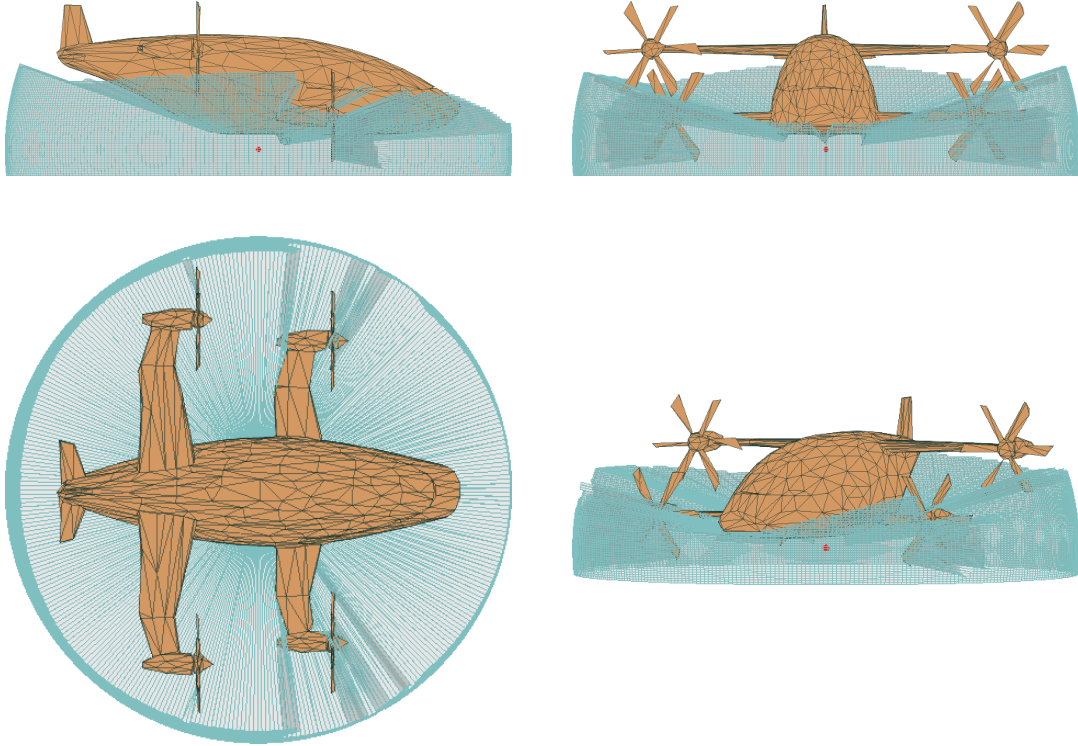
FIGURE 2.17: UAS body mask

2.2.6 Coordinate systems

The points composing the UAS model are defined in a north-east-down reference frame. The UAS can also vary its attitude through the flight. For asymmetrical UAS, the x-axis is directed toward the nose of the aircraft, the y-axis is directed towards the right-wing, and the z-axis is directed toward the bottom of the aircraft. The attitude of the UAS is defined in terms of heading, pitch and bank angles Ψ, Θ and Φ, respectively.

The rotations required to transform the UAS axes to the FEF are shown in Equation 1. Through this geometrical operation, every triangle point defining the frame of the UAS can be transformed in FEF for later calculations. The transformation of a point in the FEF to the UAS local axes can be obtained by inverting $\mathcal{M}^{UAS}_{FEF}$. This operation becomes particularly useful to determine airframe shadowing. When the position of a point is known and referenced to the local UAS coordinates, those cartesian coordinates can be transformed to spherical coordinates.

$$\mathcal{M}R_Z\left[\theta\right] = \begin{bmatrix} \cos\left[\theta\right] & -\sin\left[\theta\right] & 0 & 0 \\ \sin\left[\theta\right] & \cos\left[\theta\right] & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.25}$$

$$\mathcal{M}R_X\left[\theta\right] = \begin{bmatrix} 1 & 0 & & 0 \\ 0 & \cos\left[\theta\right] & -\sin\left[\theta\right] & 0 \\ 0 & \sin\left[\theta\right] & \cos\left[\theta\right] & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.26}$$

$$\mathcal{MR}_Y\left[\theta\right] \;=\; \begin{bmatrix} \cos\left[\theta\right] & 0 & \sin\left[\theta\right] & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\left[\theta\right] & 0 & \cos\left[\theta\right] & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.27}$$

$$\mathcal{MT}\left[\Delta x, \Delta y, \Delta z\right] \;=\; \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.28}$$

$$\mathcal{M}_{\text{FEF}}^{UAS} \;=\; \mathcal{MT}\left[P_{UAS}^{\text{FEF}}\right] \cdot \mathcal{MR}_X\left[\pi\right] \cdot \mathcal{MR}_Z\left[\Psi_{UAS}^{\text{FEF}}\right] \tag{2.29}$$

$$\cdot\,\mathcal{MR}_Y\left[\Phi_{UAS}^{\text{FEF}}\right] \cdot \mathcal{MR}_X\left[\Theta_{UAS}^{\text{FEF}}\right]$$

$$\mathcal{M}_{\text{UAS}}^{\text{FEF}} \;=\; \left[\mathcal{M}_{\text{FEF}}^{UAS}\right]^{-1} \tag{2.30}$$

2.3 Communication Systems

After defining the scenario where the action will take place and the dynamics and shape of the object, it is necessary to define the communication frame. It involves the different communication systems used, such as the ground and air segments. The data significantly influences the link budget, a crucial variable that will change the reach of the communication already influenced by the propagation channel characteristics.

The main job of an antenna is to transfer an electro magnetic (EM) wave coming from a transmission line into free space [37]. An antenna can also be defined as the interface between systems inside the UAS frame and the outside world. To properly characterise this interface, it is essential to determine the spatial distribution of the power in all directions (i.e. radiation pattern in 3D space).

There is a difference between civilian aircraft and military aircraft. Both share the same systems required for take-off, navigation and landing. The systems involved in aircraft might not have the same direction of arrival; systems such as the instrument landing system are located on the ground, while global navigation satellite system (GNSS) and SatCom require satellite infrastructure. Other systems such as radar do not require any support other than a reflecting surface to return the signal. The list of systems connected to an antenna in a typical UAS is wide and grows as more and more communication systems are included in the UAS.

Where the communication system connects affects the positioning of the antenna. Finding the adequate location of an antenna in an aircraft is known as siting process. Figure 2.18 depicts a simplified version of the process required to fix the position of each antenna. An incorrect position of the antenna could affect all the other antennas. In addition, the structure of the aircraft must be taken into account. Many elements can cause obstructions, such as the tail fin, wings and other antennas.

The antenna sitting process aims to correctly place an antenna, ensuring radio frequency systems performance and interoperability. According to the authors in [38], an aircraft program's erroneous antenna sitting results in time and cost implications. It is paramount to incorporate adequate and efficient antenna integration methodologies at an early design stage. Computational electromagnetic allows the replacement of traditional techniques by simulations using powerful EM engineering tools. The main benefits of simulations over scale mock-up measurements are:
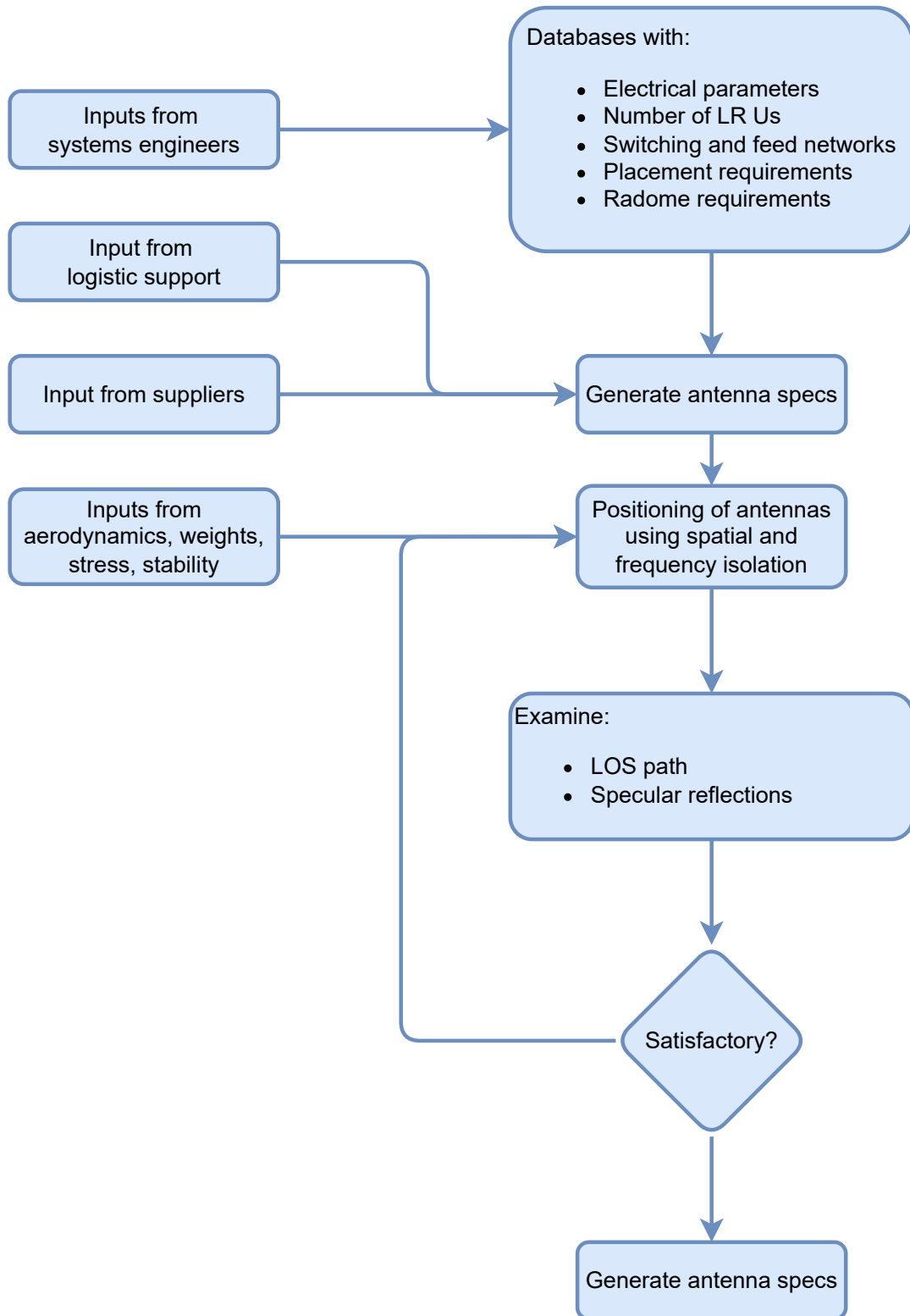
FIGURE 2.18: Flow chart of the initial design phase [37]

(A) 120 Hz (B) 240 Hz (C) 1093 Hz
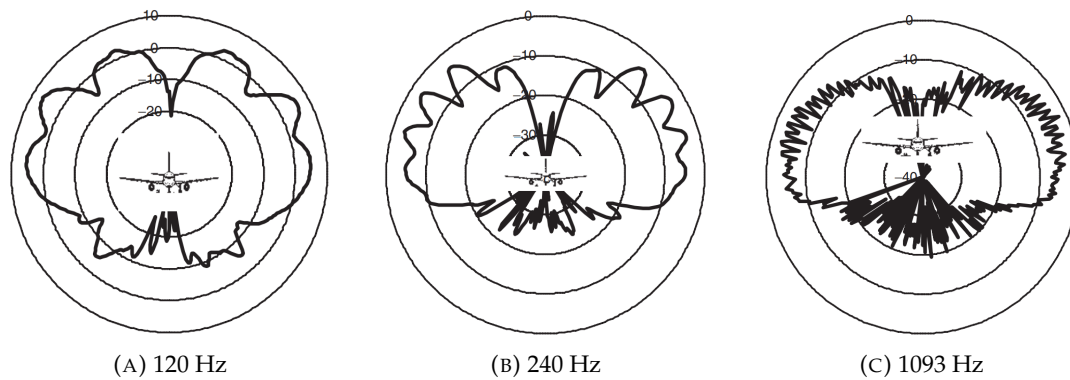
FIGURE 2.19: Roll plane radiation pattern of an antenna on the upper fuselage, between wings and engines measured at different frequencies [37]

- Flexibility: little additional work is required to find an optimal location once aircraft and antennas are modelled.

- Time and cost reduction: according to [38], a virtual test is only a 10% of the cost of a scaled mock-up test.

- Risk reduction: the process of antenna sitting can be integrated into the design process of the aircraft. No need to wait until a final model of the aircraft is released.

- Modelled antenna accuracy and variety: no limitation exists to the types of antennas that can be modelled.

- Material and environmental effects can be taken into account. Not only materials but weather conditions can be included in the process.

- Additional information and applications: EM simulation tools produce more information than scale model measurements.

Once the process of antenna sitting is done, the antenna will not be moved from its position. Antenna performance measurements are typically carried out in ideal conditions and measurement chambers. This ideal distribution of the radiated power will change when installed on a structure. Commonly, two principal planes are used for measurement; azimuth or horizontal and elevation or vertical plane. Figures 2.19 and 2.20 summarise the output of such measurements. The figures also highlight the variation of such influence when different frequencies are used.

Figures 2.19 and 2.20 also show areas with backlobes and areas with low power. Communication system designers and researchers often avoid these positions and neglect them. Flying into areas of low power can occur due to the route performed by the UAS or its manoeuvres. Even with a ground station divided into sectors, the UAS has a probability of flying into those areas with low power.

Authors in [39] did a series of flights to test the influence of certain attitudes on the performance of control and non-payload communication. The findings after such flights are that path loss can be up to 15 dB in C-band and 20 dB in L-band larger than free-space path loss. Compared to typical channel models, the signal amplitude fading was worse than that in a Rayleigh model representing many scatters.

One method to reduce the effect of sidelobes or even back lobes is to influence the configuration of the antennas used and their configuration. In particular, the

(A) 120 Hz   (B) 240 Hz   (C) 1093 Hz
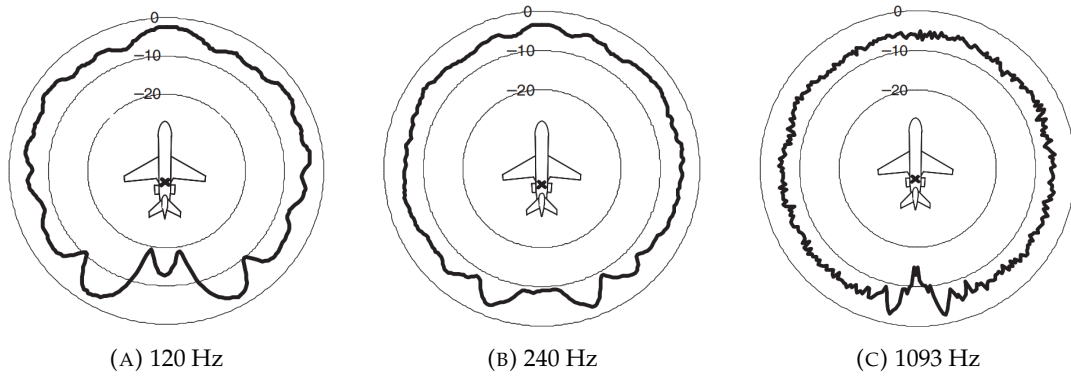
FIGURE 2.20: Azimuth radiation pattern of an antenna on the upper fuselage, between wings and engines measured at different frequencies [37]

parameter with the most influence is the tilt angle, according to [40]. This angle can improve network performance and reduce the costs and complexity of the system. This recommendation cannot be considered in this research as we assume the UAS cannot influence the vendor to tilt the antenna and benefit aerial over terrestrial users.

In order to increase the fidelity of the research, each transmitting or receiving element will have a realistic radiation pattern based on publicly available information. Each antenna is then attached to either the UAS or an antenna in the ground network. The antenna is also moved and rotated to simulate and study the effect of such movements. From work published in [39], it is already proven that the movement and attitude changes play a significant role in the A2G link performance.

2.3.1   Ground network radiation pattern

The communication system requires the design of a ground network. The UAS will not connect to a single antenna covering the complete map. In the simulations, a mobile network in an urban environment is considered. As in every urban environment, each base station is equipped with three antennas covering one sector [40]. This design corresponds to a typical mobile network serving users on the ground. The UAS does not influence the network, and the network operator does not prioritise aerial over ground users. One of the assumptions is that the UAS connects to the network as it is.

Antennas composing sectors can be tilted both electronically and mechanically. It is common to find antennas composed of an array of dipoles. The tilting angle of such antennas can be modified easily by changing the currents feeding each element. This characteristic is mainly used to improve coverage in hotspots of users. It can also be used to modify the coverage based on our activities. Tilting the antenna electronically removes the need for site visits and can be done remotely.

There are many antenna arrays, and their use varies from personal, commercial, and military applications. The most commonly used array is composed of elements along a line. Various antennas can be used in arrays, such as dipoles, loops, apertures, microstrips, horns, reflectors. Figure 2.21 represents an array used in base-station antennas.

AGI-STK is used to simulate the ground network antenna's radiation pattern. It is possible to import the radiation pattern of the designed antenna from AGI-STK to Matlab through plugins provided by the simulator. Once the ground network

FIGURE 2.21: Cell tower carrying antennas of four cellular networks

radiation pattern has been imported, it can be used by the optimisation algorithm. How to configure the antenna array can be found in [41].

2.3.2 UAS radiation pattern

According to [37], dipoles, helixes, horns, loops, monopoles, notches, patches, spirals, and reflectors are the most used antennas. Table 2.1 shows the possible combinations of frequencies and types of antennas used for aircraft systems. Telemetry frequencies fit with the scenario where a UAS exchanges information through the A2G channel. The type of antennas commonly used in telemetry systems is monopoles or dipoles. Hence, the selected antenna installed in the UAS is a dipole operating at a frequency of 1.9 GHz.

AGI-STK is once more used to calculate the radiation pattern of the antenna. The most commonly used ratio of length to wavelength is 0.5. With this ratio and the frequency of operation, the radiation pattern does show the typical doughnut shape of a dipole. The radiation pattern of the UAS can be seen in Figure 2.23.

2.3.3 Coordinate systems

Ground network antennas do not move through the simulation and are not attached to secondary objects. Their rotation matrices are more straightforward than those used for the UAS radiation pattern. It is only necessary to rotate ($\mathcal{M}R_z \cdot \mathcal{M}R_y \cdot \mathcal{M}R_x$ chain) plus a translation to have the antenna located adequately in the global coordinate system, the FEF already defined.

$$\mathcal{M}_{Rx}^{\text{FEF}} = \mathcal{M}T\left[P_{Rx}^{\text{FEF}}\right] \cdot \mathcal{M}R_Z\left[\Psi_{Rx}^{\text{FEF}}\right] \cdot \mathcal{M}R_Y\left[\Phi_{Rx}^{\text{FEF}}\right] \cdot \mathcal{M}R_X\left[\Theta_{Rx}^{\text{FEF}}\right] \qquad (2.31)$$

the position of the transmitter antenna Rx requires the translation chain from Equation 2.30. The position of the antenna is first obtained in the reference system of the

(A) Radiation pattern in city

(B) 3D view of the radiation pattern plus horizontal and vertical cuts
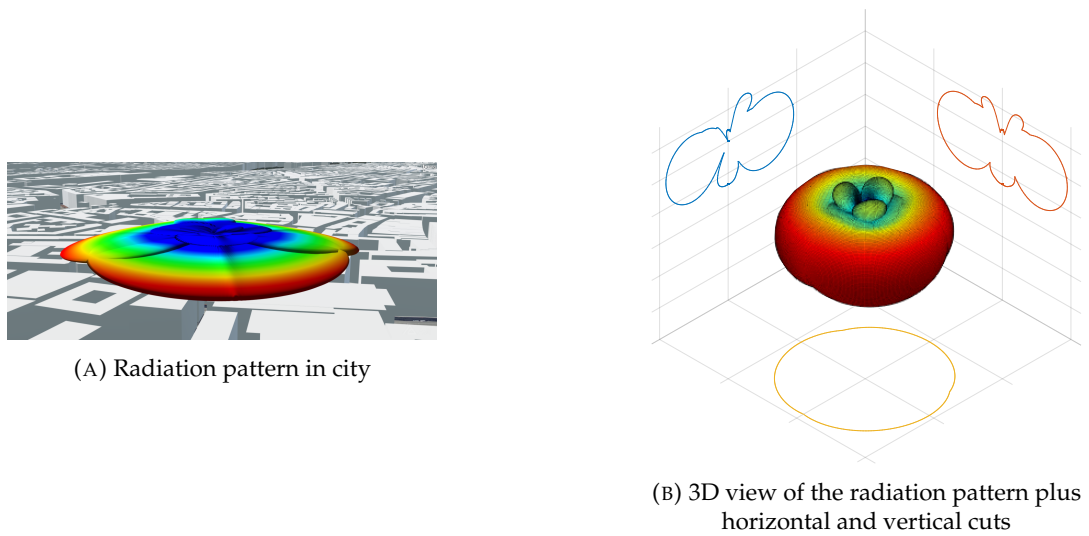
FIGURE 2.22: Ground network radiation pattern

TABLE 2.1: Types of antennas used for aircraft systems as defined by [37]

System	Frequency range [MHz]	Polarization	Type of antenna
ADF	0.19-1.8	Vert	Loop
Coms-Military	30-88	Vert	Monopole
DF	30-407	Vert	Loop
Distress-Maritime	2.182	Vert	Monopole
Distress-Civil	121.5	Vert	Monopole
Distress-Satellite	406.028	Vert	Monopole
DME	960-1215	Vert	Monopole
EL	121-243	Vert	Monopole
ESM	500-40000	All	Monopole, Spirals
GNSS L1	1565-1586	RHCP	Patches and patch arrays
GNSS L2	1217-1238	RHCP	Patches and patch arrays
GLONASS	1602-1615.5	RHCP	Patches and patch arrays
GLONASS	1240-1256	RHCP	Patches and patch arrays
HF	2-30	Horiz and Vert	Monopoles, loops
ILS	74.75-335	Horiz and Vert	Dipoles, loops, monopoles
MLS	5031-5091	Vert	Monopole and yagui
RadAlt	4200-4400	Horiz	Rect. waveguide horns, patches
Radar S-band	2000-4000	Vert	Slotted waveguides
Radar X-band	8200-12000	All	Reflectors and slotted waveguides
Radio Broadcast FM	88-108	Vert	Monopole
SatCom Military UHF	240-310	Vert and circular	Monopole and crossed dipoles
SatCom Military SHF	7200-8400	Circular	Reflector
Telemetry	1500-6400	Vert	Monopoles, dipoles
TACAN	960-1215	Vert	Monopole
TCAS	1030-1090	Vert	Monopole
UHF	225-400	Vert	Monopole
UMTS	1800-1900	Vert	Monopole
VHF Civil	108-137	Vert	Monopole
VHF Maritime	150-174	Vert	Monopole
VHF Military	108-152	Vert	Monopole
VOR	108-118	Horiz	Monopoles, dipoles
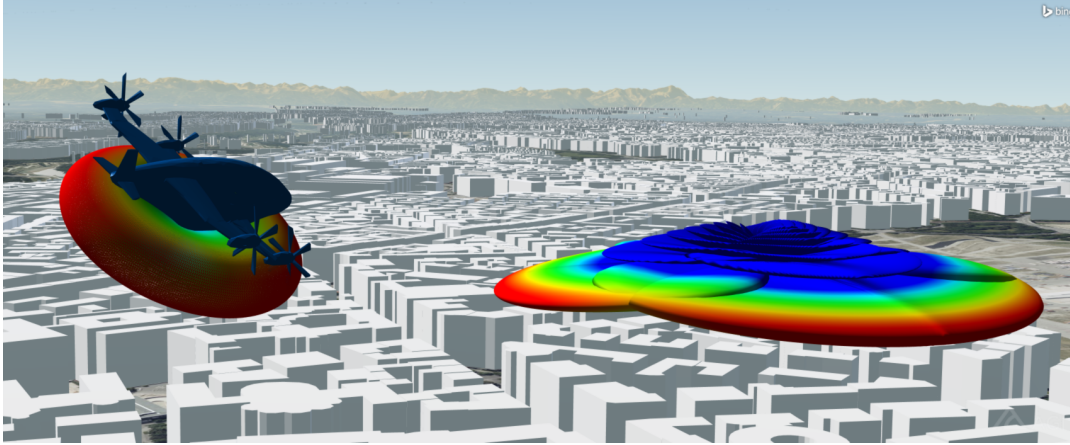
FIGURE 2.23: UAS CAD Model, UAS radiation pattern and ground station radiation pattern used for simulations

UAS and then translated and rotated to the global coordinate system:

$$\mathcal{M}_{Tx}^{\text{FEF}} = \mathcal{M}_{UAS}^{\text{FEF}} \cdot \mathcal{M}T \left[P_{Tx}^{UAS} \right] \cdot \mathcal{M}R_Z \left[\Psi_{Tx}^{UAS} \right] \cdot \mathcal{M}R_Y \left[\Phi_{Tx}^{UAS} \right] \cdot \mathcal{M}R_X \left[\Theta_{Tx}^{UAS} \right] \quad (2.32)$$

References

[1] Roland Bürgmann, Paul A Rosen, and Eric J Fielding. "Synthetic Aperture Radar Interferometry to Measure Earth's Surface Topography and Its Deformation". In: *Annual Review of Earth and Planetary Sciences* 28.1 (2000), pp. 169–209.

[2] Carlos H Grohmann. "Evaluation of TanDEM-X DEMs on selected Brazilian sites: Comparison with SRTM, ASTER GDEM and ALOS AW3D30". In: *Remote Sensing of Environment* 212 (2018), pp. 121–133.

[3] Dai Yamazaki et al. "A high-accuracy map of global terrain elevations". In: *Geophysical Research Letters* 44.11 (2017), pp. 5844–5853.

[4] Bram De Greve. "Reflections and Refractions in Ray Tracing". In: *October* (2004).

[5] Akrama Al-Hourani, Sithamparanathan Kandeepan, and Simon Lardner. "Optimal LAP Altitude for Maximum Coverage". In: *IEEE Wireless Communications Letters* 3.6 (2014), pp. 569–572.

[6] Yoav I H Parish and Pascal Müller. "Procedural modeling of cities". In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques - SIGGRAPH '01*. August. New York, New York, USA: ACM Press, 2001, pp. 301–308.

[7] K. G. Nikolakopoulos, E. K. Kamaratakis, and N. Chrysoulakis. "SRTM vs ASTER elevation products. Comparison for two regions in Crete, Greece". In: *International Journal of Remote Sensing* 27.21 (2006), pp. 4819–4838.

[8] Katherine Arrell et al. "Spectral filtering as a method of visualising and removing striped artefacts in digital elevation data". In: *Earth Surface Processes and Landforms: The Journal of the British Geomorphological Research Group* 33.6 (2008), pp. 943–961.

[9]   Hannes I. Reuter et al. "A first assessment of Aster GDEM tiles for absolute accuracy, relative accuracy and terrain parameters". In: *2009 IEEE International Geoscience and Remote Sensing Symposium*. Vol. 5. 2009, pp. V-240-V–243.

[10]   Isabel Pipaud, David Loibl, and Frank Lehmkuhl. "Evaluation of TanDEM-X elevation data for geomorphological mapping and interpretation in high mountain environments — A case study from SE Tibet, China". In: *Geomorphology* 246 (2015), pp. 232–254.

[11]   Paola Rizzoli et al. "Generation and performance assessment of the global TanDEM-X digital elevation model". In: *ISPRS Journal of Photogrammetry and Remote Sensing*. Vol. 132. 2017, pp. 119–139.

[12]   Gerhard Krieger et al. "TanDEM-X: A Satellite Formation for High-Resolution SAR Interferometry". In: *IEEE Transactions on Geoscience and Remote Sensing* 45.11 (2007), pp. 3317–3341.

[13]   Nikolaus Faller, Marco Weber, and Infoterra GmbH. "TerraSAR-X and TanDEM-X: Revolution in spaceborne radar". In: *2007 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2007, pp. 4924–4928.

[14]   Sukhjit Sehra, Jaiteg Singh, and Hardeep Rai. "Assessing OpenStreetMap Data Using Intrinsic Quality Indicators: An Extension to the QGIS Processing Toolbox". In: *Future Internet* 9.2 (2017), p. 15.

[15]   Peter Mooney, Padraig Corcoran, and Adam C Winstanley. "Towards quality metrics for OpenStreetMap". In: *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '10*. New York, New York, USA: ACM Press, 2010, p. 514.

[16]   Mordechai Haklay. "How Good is Volunteered Geographical Information? A Comparative Study of OpenStreetMap and Ordnance Survey Datasets". In: *Environment and Planning B: Planning and Design* 37.4 (2010), pp. 682–703.

[17]   Y Corre and Y Lostanlen. "3D urban propagation model for large ray-tracing computation". In: *2007 International Conference on Electromagnetics in Advanced Applications, ICEAA'07*. Vol. 33. 0. 2007, pp. 399–402.

[18]   M Over et al. "Generating web-based 3D City Models from OpenStreetMap: The current situation in Germany". In: *Computers, Environment and Urban Systems* 34.6 (2010), pp. 496–507.

[19]   *Maps Static API*. `https://cutt.ly/QIaPMv2`. Accessed: 2022-01-13.

[20]   *Plot Google map code by Zohar Bar-Yehuda (2022)*. `https://cutt.ly/GIaSaYj`. Accessed: 2022-01-13.

[21]   Eugene L Duke, Robert F Antoniewicz, and Keith D Krambeer. *Derivation and definition of a linear aircraft model*. Vol. 1207. National Aeronautics, Space Administration, Scientific, and Technical . . ., 1988.

[22]   Mostafa Ibrahim and Huseyin Arslan. "Air-Ground Doppler-delay spread spectrum for dense scattering environments". In: *MILCOM 2015 - 2015 IEEE Military Communications Conference*. Vol. 2015-Decem. IEEE, 2015, pp. 1661–1666.

[23]   Rl Page. "Brief history of flight simulation". In: *SimTecT 2000 Proceedings* (2000).

[24]   D J Allerton. "The impact of flight simulation in aerospace". In: *The Aeronautical Journal* 114.1162 (2010), pp. 747–756.

[25]   William L Oberkampf et al. "Error and uncertainty in modeling and simulation". In: *Reliability Engineering and System Safety* 75.3 (2002), pp. 333–357.

[26] Tobias Paul, Thomas R Krogstad, and Jan Tommy Gravdahl. "Modelling of UAV formation flight using 3D potential field". In: *Simulation Modelling Practice and Theory* 16.9 (2008), pp. 1453–1462.

[27] Johannes Meyer et al. "Comprehensive Simulation of Quadrotor UAVs Using ROS and Gazebo". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 2012, pp. 400–411.

[28] Edgardo V Oliveros and a. Jennifer Murray. "Modeling and Simulation of an UAS Collision Avoidance Systems". In: *Report* (2010).

[29] *Analytical Graphics, Inc.* `https://www.agi.com/products/stk`.

[30] Randolph M Jones et al. "Automated intelligent pilots for combat flight simulation". In: *AI Magazine* (1999).

[31] Wayne Durham. *Aircraft flight dynamics and control*. John Wiley & Sons, 2013.

[32] Adrian Exposito, Dominic Schupke, and Hector Esteban. "Route Optimisation for Maximum Air to Ground Channel Quality". In: *IEEE Access* 8 (2020), pp. 203619–203630.

[33] *3D model used for simulations*. `https://cutt.ly/uIYSJC0`. Accessed: 2022-01-13.

[34] Paolo Cignoni et al. "MeshLab: an Open-Source Mesh Processing Tool". In: *Eurographics Italian Chapter Conference*. Ed. by Vittorio Scarano, Rosario De Chiara, and Ugo Erra. The Eurographics Association, 2008.

[35] John Burgess. "RTX on—The NVIDIA Turing GPU". In: *IEEE Micro* 40.2 (2020), pp. 36–44.

[36] Tomas Moller and Ben Trumbore. "Fast, Minimum Storage Ray/Triangle Intersection". In: *ACM SIGGRAPH 2005 Courses*. 1. 2005, p. 7.

[37] Thereza M Macnamara. *Introduction to Antenna Placement and Installation*. Chichester, UK: John Wiley and Sons, Ltd, 2010.

[38] Francisco Javier Jimenez et al. "Modern electromagnetic simulation tools applied to On-aircraft Antenna Integration". In: *2012 6th European Conference on Antennas and Propagation (EUCAP)*. IEEE, 2012, pp. 912–916.

[39] Ruoyu Sun and David W Matolak. "Over-Harbor Channel Modeling with Directional Ground Station Antennas for the Air-Ground Channel". In: *2014 IEEE Military Communications Conference*. IEEE, 2014, pp. 382–387.

[40] Nikolay Dandanov et al. "Dynamic Self-Optimization of the Antenna Tilt for Best Trade-off Between Coverage and Capacity in Mobile Networks". In: *Wireless Personal Communications* 92.1 (2017), pp. 251–278.

[41] *AGI STK Phased Array Antenna*. `https://cutt.ly/fInzsol`. Accessed: 2022-01-13.

Chapter 3

Channel Model

3.1 Introduction

Connectivity in networks for unmanned aerial system (UAS) is one of the qualitative demands to be explored. Other qualitative demands exist, such as traffic demands, infrastructure, adaptability or scalability. Those should be studied and adequately addressed; however, this chapter focus solely on connectivity for UAS. A UAS in the air transmits information to at least three groups of users. The first group is conformed by decision making entities that track the position and status of the UAS at all times. Tracking of the UAS is paramount for safety and security reasons. The second group is ground personnel, relevant for applications such as disaster management. This group counts on the UAS to transmit any sensor data or relevant position gathered through the mission. Last but not least, connectivity between UASs is relevant for those applications relying on a network of UASs to provide coverage.

In order to meet the qualitative demands in terms of connectivity, a good understanding of the air to ground (A2G) channel is required. A good understanding of the channel is equivalent to depicting the different mechanisms that hinder connectivity. Ideally, a channel model characterising such mechanisms can prepare for what the UAS might encounter in real scenarios. Such a channel can also serve as the enabling foundation for flexible and practical design and testing for various systems. As highlighted in Reference [1], significant research attention has been attracted to vehicular channel measurements and understanding the underlying physical phenomena.

Through time, such channels have gained complexity. A2G communications started with favourable conditions such as high transmitted power levels, narrow signal bandwidths, high ground site antennas. For these types of communications, simple models for channel attenuation sufficed. Future UAS systems will not benefit from such ideal conditions. As mentioned in previous chapters, the UAS base station (BS) might not be located in a clear area in high ground, ground operators will not benefit aerial users over ground users, and wider bandwidths are required. Let us not forget about low size, weight and power requirements that might reduce the transmission and reception capabilities of UASs. More comprehensive A2G channel characteristics will be required to ensure robust signal designs, as mentioned by [2].

Modelling signal propagation requires accurate and fast tools, two opposite features. On the one hand, one can have an extremely accurate modelling tool such as ray tracing, which will take elevated amounts of computing power. On the other hand, some tools rely on statistics to model the channel. This chapter covers both tools, providing a full view of the options available. After providing sufficient arguments to support the modelling choice, follows the description of the mathematical model of the channel.

This chapter will cover the different channel models and simulation approaches available. The first sections provide the basics of wave propagation and how this can be translated into a channel model. This unconventional order allows the reader to gain basic knowledge and get familiar with the terminology used later in the state-of-the-art review. The state-of-the-art section gathers the most recent publications related to channel modelling and modelling of the various effects that can disturb the A2G channel. After presenting the current research directions and advances, the custom channel model is introduced. This chapter ends with early results and the channel's performance under different conditions.

3.2   Radio Wave Fundamentals and Transmission

The successful implementation of radio systems requires knowledge of how radio waves are transmitted in various media. More important is to determine the available power to the receiver. The very basics of transmission were first laid down by James Clerk Maxwell (1831-1379). This physicist and mathematician proved that electricity, magnetism and light are different manifestations of the same phenomenon. Maxwell proved that propagation is dependent on the medium where it travels. The transmission media can be divided into linear versus nonlinear, bounded versus unbounded, homogeneous versus non-homogeneous and isotropic versus non-isotropic. If the principle of superposition can be applied at some point, the media can be considered linear. A bounded media is one finite in extent or unbounded otherwise. Homogeneous physical properties at different points are related to homogeneous media. Anisotropic medium has the same physical properties in different directions.

The difference between the transmitted and received power is known as transmission loss. This effect receives other names, such as attenuation and path loss. Attenuation is not only limited to radio links; other areas are affected by it: attenuation of sound and light in seawater and losses over an optical link or electric power transmission line.

3.2.1   Attenuation mechanisms

The simplest case of attenuation is free space propagation. The attenuation occurs when transmit and receive antenna try to establish a link through free space. The received power can be calculated, and the formulas are readily available. The following explanation follows the reasoning of Reference [3]. The first step in obtaining the received signal power starts by establishing the first assumptions:

1. $d > d_f$ where d_f is the Friis distance.

2. $d_f > (2D^2/\lambda)$ and $d_f >> \lambda$

3. The transmit antenna is an ideal isotropic antenna

under these assumptions, the radiated energy is spread over the surface of a sphere, where the power density S in units of watts per m^2 (W/m^2) at distance d is equal to:

$$S = \frac{P_T}{4\pi d^2} \tag{3.1}$$

Where $4\pi \cdot d^2$ is the surface area of the sphere of radius d, and P_T is the transmitted power. The available power P_R in watts (w) at the receive antenna with effective

area A is given by:

$$P_R = \frac{P_T}{4\pi d^2} A = \frac{\lambda^2}{(4\pi d)^2} P_T \frac{4\pi A}{\lambda^2} = \frac{\lambda^2}{(4\pi d)^2} P_T G_R \tag{3.2}$$

the last equation can be rewritten to express path loss, which is the ratio of the transmitted power to the received power. For an isotropic receive antenna with unity gain, this gives:

$$\frac{P_T}{P_R} = \left[\frac{4\pi d}{\lambda}\right]^2 = \left[\frac{4\pi d f}{c}\right]^2 \tag{3.3}$$

the equations change when the transmit and receive antenna have a gain greater than unity. Equation 3.3 becomes Equation 3.4 when the gains are taken into account:

$$\frac{P_T}{P_R} = \frac{1}{G_T \cdot G_R} \left[\frac{4\pi d}{\lambda}\right]^2 \tag{3.4}$$

the received signal power in decibels is then expressed as:

$$P_R = P_T + G_T + G_R + 20\log\left[\frac{\lambda}{4\pi d}\right] \tag{3.5}$$

Generally, whenever a wave travels through a true-to-life environment, it will encounter several media. The properties of such media define which propagation mechanisms occurs. Different propagation mechanisms include reflection, absorption, refraction, diffraction and scattering.

Reflection by an isotropic material

This phenomenon occurs when an electromagnetic wave collides with a medium considerably bigger than its wavelength. The reflection can be either diffuse or specular depending on the medium where reflection occurs. Another characteristic of this phenomenon is its mirror-like properties: the angle of incidence and the angle of reflection are equal.

The reflection can occur at a normal incidence and an oblique incidence. Both can be divided into subcases if the reflection media is included. The cases to be studied are then further divided into four possibilities. Two variants exist of waves with normal incidence:

- Normal incidence with a perfect conductor: this situation makes the electric and magnetic fields in the perfect conductor equal to zero. The presence of a perfect conductor causes the incident wave to be reflected. A secondary effect is that the reflected wave has the same magnitude as the incident wave but in the opposite direction.

- Normal incidence with perfect dielectric: the two media have different properties, the above assumptions no longer apply. The presence of a perfect dielectric causes the reflected wave to have a different magnitude.

the scenario under oblique incidence produces a different output. First of all, two waves result upon arrival of the incident wave. One is reflected in the same medium, and a refracted or transmitted wave travels through the secondary medium. The two possibilities are:

- Oblique incidence with a perfect conductor: the wave is reflected in this case without losses.

- Oblique incidence with perfect dielectric: The wave is not entirely reflected; part of it travels through the second medium.

The case that will happen repeatedly is a wave travelling through certain media and impacting obliquely into another. Additionally, the most frequent type of media is a dielectric one. In this group, we can find elements such as mica, glass, plastics, and the oxides of various metals.

From the geometry displayed in Figure 3.1, the following relationships apply:

$$\cos{[\theta_1]} = \frac{d_1}{r_1} \quad \cos{[\theta_2]} = \frac{d_1}{r_3} \quad \frac{\sin{[\theta_1]}}{\sin{[\theta_3]}} = \frac{r_3}{r_1} \tag{3.6}$$

$$\sin{[\theta_1]} = \frac{d_2}{r_3} \quad \sin{[\theta_2]} = \frac{d_2}{r_2} \quad \frac{\sin{[\theta_2]}}{\sin{[\theta_1]}} = \frac{r_3}{r_2} \tag{3.7}$$

For a wave travelling in the two media with frequency f the rightmost relationships can be translated into:

$$\frac{\lambda_2}{\lambda_1} = \frac{v_2}{v_1} = \sqrt{\frac{\epsilon_1}{\epsilon_2}} \tag{3.8}$$

plugging Equation 3.8 into 3.7:

$$\frac{\sin{[\theta_2]}}{\sin{[\theta_1]}} = \frac{v_2}{v_1} = \frac{\lambda_2}{\lambda_1} = \sqrt{\frac{\epsilon_1}{\epsilon_2}} \tag{3.9}$$

Equation 3.9 can be further modified by introducing $n = \sqrt{\epsilon_r \mu_r}$, which for nonmagnetic material is equal to $\sqrt{\epsilon_r}$. The coefficient n is generally described as a complex quantity dependent on the frequency. It is known as the refractive index of the medium. Including the coefficient n into Equation 3.9 gives the characteristic relationship known as Snell's law:

$$\frac{\sin{[\theta_2]}}{\sin{[\theta_1]}} = \frac{n_2}{n_1} \tag{3.10}$$

Assuming a lossless medium where conservation of energy applies, the incident power can be expressed as the sum of the transmitted and reflected power:

$$\frac{1}{\eta_1} E_i^2 \cos{[\theta_1]} = \frac{1}{\eta_1} E_r^2 \cos{[\theta_1]} + \frac{1}{\eta_2} E_t^2 \cos{[\theta_2]} \tag{3.11}$$

Two cases should be regarded per the oblique incidence with a perfect conductor. The first case to consider is a perpendicularly polarised wave. For this polarisation, the electric field at the boundary gives Equation 3.12:

$$E_i = E_r + E_t \tag{3.12}$$

Said equation can be introduced into Equation 3.11, resulting in the reflection coefficient $\rho_\perp$, also known as the Fresnel reflection coefficient.

$$\rho_\perp = \frac{E_r}{E_i} = \frac{\sqrt{\epsilon_1}\cos{[\theta_1]} - \sqrt{\epsilon_2}\cos{[\theta_2]}}{\sqrt{\epsilon_1}\cos{[\theta_1]} + \sqrt{\epsilon_2}\cos{[\theta_2]}} \tag{3.13}$$
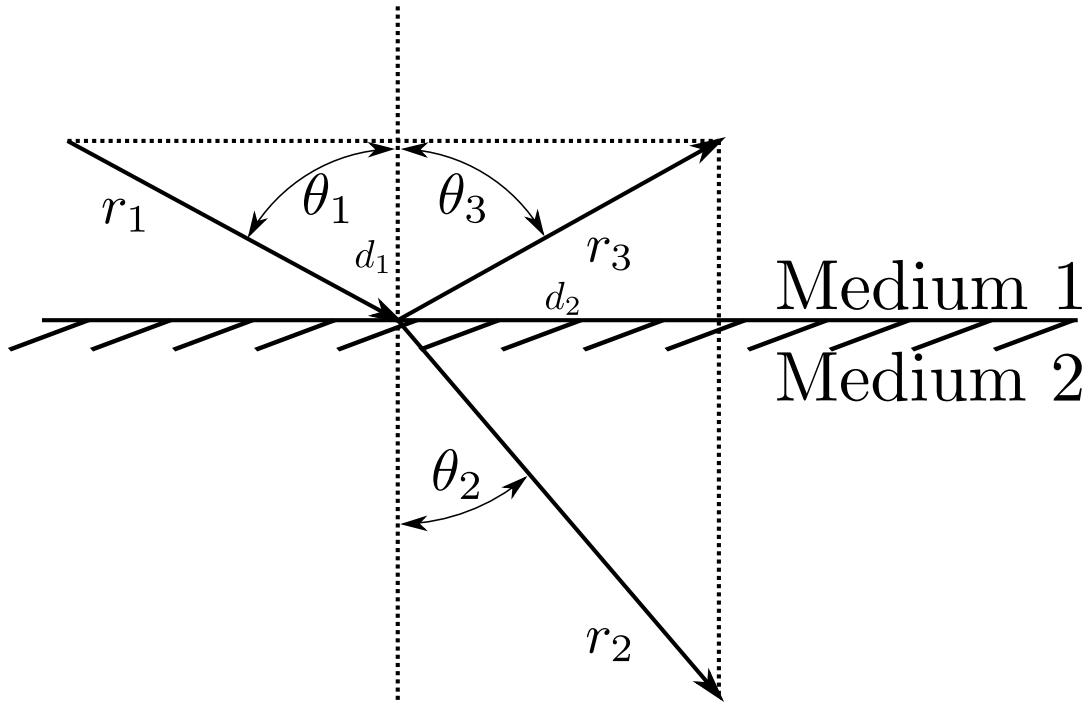
FIGURE 3.1: Oblique incidence at a dielectric material

Parallel polarisation starts with a modified version of Equation 3.11:

$$(E_i - E_r) \cos [\theta_1] = E_t \cos [\theta_2] \tag{3.14}$$

although the starting point is different, Equation 3.11, representing conservation of energy, is applied. By substitution, the Fresnel reflection coefficient is obtained:

$$\rho_\parallel = \frac{E_r}{E_i} = \frac{\sqrt{\epsilon_2} \cos [\theta_1] - \sqrt{\epsilon_1} \cos [\theta_2]}{\sqrt{\epsilon_2} \cos [\theta_1] + \sqrt{\epsilon_1} \cos [\theta_2]} \tag{3.15}$$

Reflection/Refraction by an anisotropic material

Anisotropic materials cause propagation to be inconsistent in all directions. This inconsistency results in two indices of refraction known as double refracting or bire-fringent. This characteristic is the optical property of a material having a refractive index that depends on the polarisation and propagation direction of the emission.

Diffuse reflection/scattering

This phenomenon occurs when the wave impinges on a rough surface with dimensions comparable to the wavelength. After encountering such a surface, a fraction of the scattered energy reaches the receiver. To consider the energy loss caused by the scattering, [3] proposes introducing a scattering loss factor. This factor is used to correct the reflection coefficient in Equations 3.15 and 3.13.

$$\rho_s = \exp \left[-8 \left[\frac{\pi \sigma \sin [\theta_1]}{\lambda} \right]^2 \right] \tag{3.16}$$
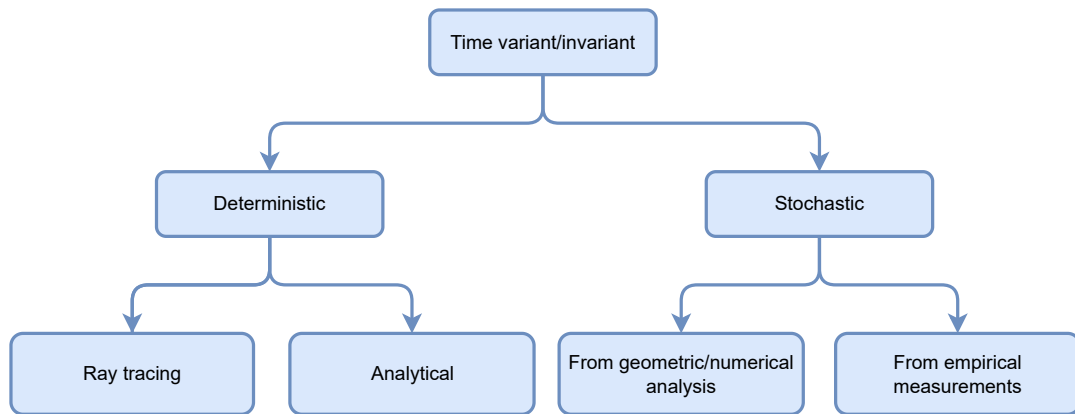
FIGURE 3.2: A2G channel characterisation

Diffraction

Generally speaking, interacting objects have a finite extent. Until now, all effects consider an infinite interacting object. A finite-sized object causes diffraction due to the wave nature of electromagnetic radiation. Every diffraction problem can be divided into diffraction of a homogeneous plane wave by a knife-edge and a wedge.

The Huygens-Fresnel principle characterises this phenomenon. The principle states that every point on a wavefront acts as a point source for a secondary radial wave. Those newly generated waves interfere with each other by adding constructively or destructively depending on the amplitude and phases. Interactions between the waves give rise to maxima and minima along the propagation path.

3.3 Channel Models

The study of the various available channel models is structured in Figure 3.2. It starts with the most relevant references covering ray tracing. Ray tracing models are carried out through simulations and can generate accurate models but require high computational resources. The following kind of model is geometric/numerical analysis. These models often assume the scatters to be distributed on regular shapes, typically ellipsoids, cylinders, or spheres. Frequently, geometrical models end with a close form solution. The third group is composed of models generated from the analysis of empirical measurements. It will be developed further, but such A2G channel measurement often proposes a 2-ray model. A careful reader should have noticed that this group appearing in Figure 3.2 was not mentioned before. Most of the models generated from empirical models already incorporate an analytical 2-ray model.

3.3.1 Ray tracing

As mentioned at the beginning of the chapter, there are fast tools and accurate tools. In the domain of accurate tools for propagation modelling, ray tracing is one of the most used ones. This technique is based on the uniform theory of diffraction (UTD) [4]. A full ray tracing implementation finds all the contributions between a transmitter and a receiver.

A full ray-tracing simulation starts with the identification of all possible propagation paths. Rays intersecting with objects causing shadowing will not be used in
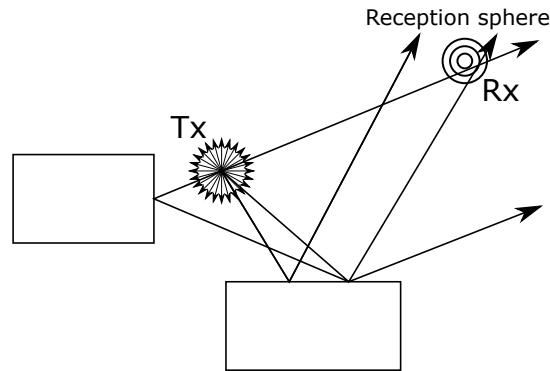
FIGURE 3.3: Illustration of ray launching process from [6]

final computations. Although not finally used, those tested rays consume computational resources, leading to higher computation times. After all possible directions are known, the UTD is applied to each ray arriving at the receiving antenna. Some rays may be terminated if they exceed a maximum attenuation. The application of UTD provides information about the amplitude, phase, delay and polarisation of each ray. The final step is to perform the coherent combination of all rays that have reached the receiver taking into account the antenna pattern. The concept of ray-launching can be seen in Figure 3.3.

One of the advantages of ray launching is the applicability even in curved geometries. Determining if a ray reached its destination is one of the most complex issues in ray launching. Reference [5] proposes two alternatives: discrete ray tubes and reception spheres. The first option is soon discarded: delimitation of adjacent ray tubes become ambiguous after reflection. Reception spheres can be used along with ray density normalisation (RDN). Ray density is defined as the number of rays per unit area. The number of theoretical rays hitting the same sphere can be obtained by simply multiplying ray density with the area of the sphere.

Reference [7] proposes to modify reflection rays as a method to improve the accuracy of calculations. Commonly, surfaces are assumed to be perfectly smooth, resulting in specular reflections. With a higher surface roughness, specular reflections are not applicable. Each ray is modified according to the properties of the surface, aiming at a better calculation. The authors in [7] introduce the term standard deviation of the height function (σ_h) to modify the reflected angle.

Computational effort and time have been mentioned already, and aspects that make a complete ray tracing implementation less suitable to be included in the optimisation routine. Nevertheless, the ray tracing process can be optimised. Favour speed will directly affect the precision of the calculations. Valid propagation paths will be neglected to reduce calculation times. Ageless in Reference [6] explores various ray-tracing optimisation techniques such as visibility graphs.

A visibility algorithm, similar to what has been explained in Subsection 2.2.5, is carried out to identify all possible ray paths between transmitter and receiver. A visibility algorithm is first applied in 2D to simplify the process. 3D will only be applied to those possible 2D directions (2D/3D hybrid method). Another simplification applied is the following: only specular reflections and transmission through walls for planes and diffractions for edges.

Polar sweep algorithms and bounding boxes are techniques used to generate visibility graphs. Both techniques drastically reduce computation time, especially in

urban models with many obstacles. In a nutshell, the polar sweep algorithm generates the next level on a visibility graph. Bounding boxes become useful in 3D environments with a high number of vertices. The central concept of bounding boxes is dividing the study area into rectangular flat-top boxes. The height of each of these boxes is equal to the highest wall within it.

Reference [6] successfully applies all these optimisation techniques. To prove its performance, they compare results and computational times against measurements. The average delay, delay spread, and delay window show good agreement between simulations and measurements. Authors also point out that information about material present in the environment could reduce discrepancies between simulations and measurements.

Monte Carlo techniques can be further applied to reduce computation times, as shown in Reference [4]. These techniques rely on repeated random sampling to obtain numerical results. Since not all theoretically possible paths are explored, computation times are reduced.

Improvements in graphics hardware implementations have made it possible to map general algorithms to fit in the graphics computation pipeline. Graphics hardware can be programmed as an extension of the C programming language. The flexibility of graphics hardware and the increase in computational power makes it possible to apply parallelisation to ray tracing algorithms. Reference [8] explores this possibility, including the simulation delay spread, reflections and refractions. Their method achieves a simulation with an accuracy of 5 meters in no more than 2 seconds. Other algorithms take at least 36 seconds with less accuracy and fewer effects.

After defining the various improvements applied to ray tracing, the different applications and enhancements to urban propagation models are also worth mentioning. Urban propagation models profit from ray tracing due to the characteristics of the environment. It is characterised by a high variation of heights and a heterogeneous distribution of street width and length. These urban models can be applied to inter-vehicle communications as in References [9, 10]. Both references successfully validate the respective 3D ray tracing models against measurements. The comparison shows a good agreement between the predicted and the measured path loss levels over time. Authors in [11] apply ray tracing to assess wireless network performance with multiple-input multiple-output (MIMO) systems. This technology makes the requirements on channel modelling still more challenging. Nevertheless, ray tracing proves to be a suitable prediction tool.

Ray tracing can also be used for network planning. References [12, 13] use techniques already commented, such as separating into two planes: vertical and horizontal. Authors also separate by near- and far-reception regions. This separation supports multiresolution and heterogeneous geographical map data. The model has been validated against measurements, showing a good agreement in a hilly city.

As shown in References [9] and [10]; ray tracing can assess inter-vehicle communications in a 2D plane. Authors in [14] apply ray-tracing techniques to low altitude platforms. One of the takeovers is to divide the model into two parts: above and below the base station. Due to the base station being tilted down to favour ground users, attenuation values are well below free space propagation. Once the UAS flies above the ground station, attenuation is rapidly increasing. In this area, most well-known channel models are not applicable.

Reference [15] goes a step beyond and presents a simulation capable of estimating receiving power, time of arrival and delay spread. Authors conclude that the channel is influenced by direct nearby buildings of the UAS and terrestrial terminal.

Although this is the Reference found with the most complex simulation, the authors still do not consider the UAS a moving point.

Not only can buildings influence the results of the propagation estimation, but the body of the UAS can also influence the results. Identifying aircraft by the radar signature or improving low observability capabilities are two applications that use ray tracing. The first optimisation approach is to use the already described RDN. RDN, together with UTD, can efficiently obtain the scattering properties of a large body. The proposed method in [16] achieves moderate memory requirements while remaining universally applicable to large objects' electromagnetic field simulations.

The same author proposes to remove triangular facets in [17]. Non-uniform rational B-splines substitute those triangular facets as a surface representation. B-splines are commonly used in CAD software, and although they can increase shape accuracy, there is no analytical method to calculate ray intersection. This issue is solved by enclosing each B-spline in a bounding box. With this method, accuracy increases while the number of objects in the scenario is significantly reduced.

3.3.2 Statistical models

According to Reference [18], the best and most rigorous approach to produce accurate A2G channels is first to perform a comprehensive measurement campaign. This measurement campaign should be followed by the development of a model and the validation of it. The literature study will then start with reviewing the most relevant measurement campaigns. Whilst measurements are classified based on terrain type, it is impossible to make a clear cut between the different terrain groups so overlaps can occur between the references. Measurement campaigns often end with a proposal for a statistical model; they will be commented among other statistical models not generated from measurements.

The complexity of such models depends on the author. Some models are simple extensions of the two-ray model that introduce a random variable into the surface reflection amplitude or phase. Other models get a high complexity by introducing many time-varying multipath components.

Measurement campaigns

References [19, 20, 21] share both authors and measurement setup, which can conduct measurements in the C-band (5060 MHz) and L-band (968 MHz). An allowance to use any frequency is sometimes complex as it involves coordination with local telecommunication operators and regulatory bodies. In addition to these permits, authors have access to an S-3B Viking aircraft for measurements. Having a dedicated aircraft for measurements is unique in the literature; other authors can only get access to custom UAS with commercial of-the-shelf equipment. The commented uniqueness of these channel measurements is why the following references will have similar authors (David W. Matolak and Ruoyu Sun).

The following mountainous areas have been flown in references [19, 20, 21]: Telluride (US), Latrobe (US) and Palmdale (US). Aircraft height in all scenarios ranges from approximately 350 to 3000 meters. The distance between sender and emitter varies from 1.1 km to 47.6 km. These numbers are similar to what a UAS will encounter in future scenarios. Among the three measurement sites, there are standard agreements within the results. Among the three scenarios, a strong ground reflection was found. The third tap delay (third ray) was present in 10% of the samples, not

a reflection that constantly happens through the route. The presence of the ground reflection is affected by the ground roughness and vegetation coverage.

References [22, 23, 24, 25] have performed flights in the vicinity of Latrobe (US) and Cleveland (US). These two areas qualify as suburban areas because they include build-up areas and diverse structures. The terrain is assumed to be flat without high altitude variations. The measurements have captured occasional clusters of multipath components when flying over these areas. These clusters are likely reflections from nearby buildings.

Previous references were also considered in [26] and new measurements campaigns. A total of 5 measurements campaigns were performed in three different places: Cleveland, Latrobe and Palmdale, all three in the US. As in [21], a two-ray behaviour dominates the channel. Proximity to the cities propitiates the appearance of a third ray more frequently than in mountainous areas. Reference [26] provides the necessary data to construct channel impulse responses for the suburban and near-urban A2G channels.

Flights over water have been performed by references [27, 28, 29]. Authors in [27] flew over the Singapore Strait and sensed the C-band (5.7 GHz), like early commented references. After measuring campaigns and evaluating results, they conclude that 2-ray models might overestimate the propagation loss. Authors attribute a significant propagation loss to ducting effects caused mainly by evaporation duct near the surface. As the platform ascends, the elevated duct will significantly decrease its contribution to path loss. Likewise, tests have been performed in References [28, 29] without clear evidence of elevated dust effects. Authors conclude that a 2-ray model with intermittent third rays is enough to characterise the over water channel. Figure 3.4 shows the received multipath components corresponding to flights over water.

Although primary A2G measurement campaigns have been commented on different terrains, other effects can influence A2G communications. The first effect interesting for this study is how the airframe and its shadowing influence the channel. Authors in Reference [30] study the effect in a fixed-wing aircraft. Authors found that this type of shadowing occurs independently of the local environment and link distance. Airframe shadowing is dependent on the roll angle with an average loss of 15.5 dB and at some points 47 dB. This effect is not instantaneous and can last for 35.2 seconds on average. Authors advise performing studies on aircraft with different configurations.

Literature covering airframe shadowing for fixed-wing aircraft is scarce, but it has higher coverage in rotatory-wing aircraft. Reference [31] studies rotor shadowing for satellite to air communications. Two aircraft were used to perform measurements, a smaller and a bigger one. Results showed that rotor blades have the potential to hinder the reception of the direct signal. During measurements, they were fading frequency-matched well with the angular velocity of rotor blades for both aircraft.

Measurements studying the effect of the environment and airframe in the A2G link have been covered. The next step is to cover measurements studying the effect of the antennas' radiation pattern in real-life flights. It is impossible to assume that ground network operators will favour aerial over ground users. Consequently, it is fair to assume that the main beam will not cover the A2G. Authors in [32, 33] encountered a higher path loss variation in areas outside the main beam. Fading in those areas covered by secondary beams was fast and worse than Rayleigh. Another effect was that the number of multipath components and delay was more significant within the back lobes.
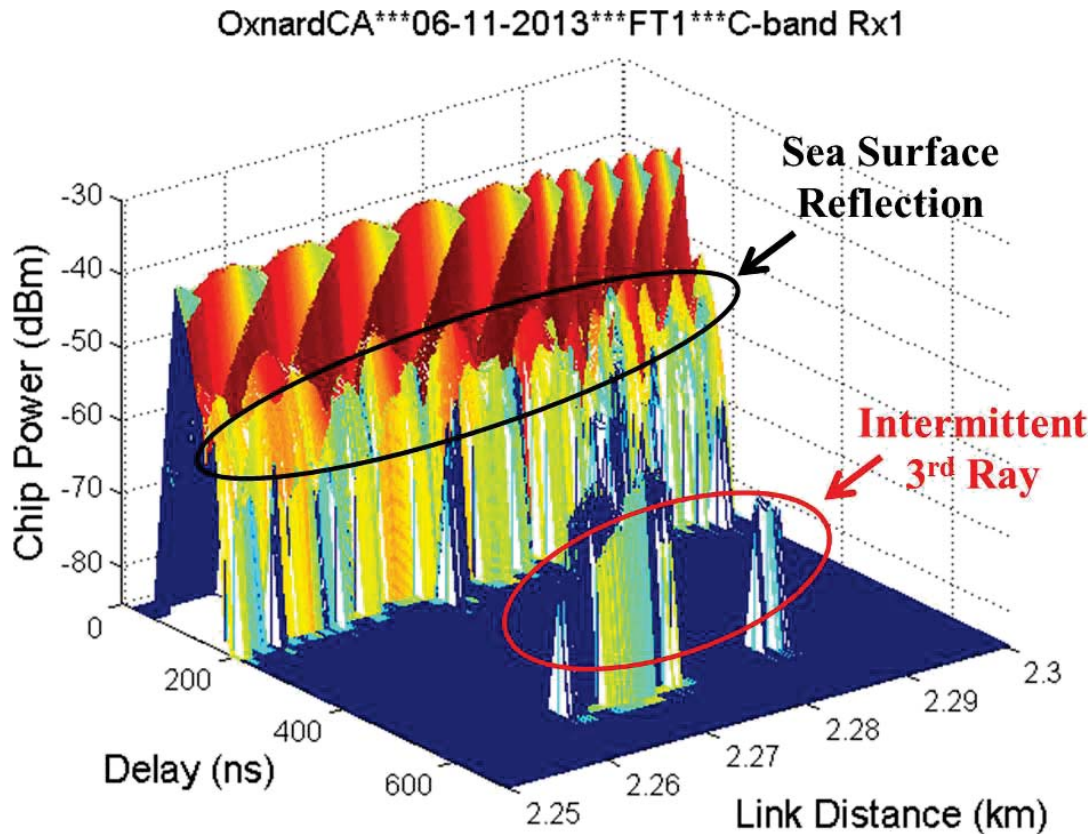
FIGURE 3.4: Multipath components found over measurement [29]

Proposed models

The selected proposed models can be divided between small-area and large-area models. Small-area models are used to validate and compare transmission standards. These models are only valid within a few dozen wavelengths and are characterised by the scattering function. The opposite happens with large-area models: these models take shadowing and other propagation path losses into account. They can be helpful in link budget calculation or in simulating outage probabilities. They are not required for the design and validation of physical-layer transmission techniques, according to Reference [34].

References [35, 36, 34] extend a channel model for aerial use on the previous one. The authors intend to develop a model suitable for validation and comparison purposes. For this purpose, a small-area model is the most suitable for this type of application. The authors propose a Doppler power spectrum for non-isotropic 2-D scattering. They propose four different spectrums to be applied en-route, arrival, taxi and parking, respectively. Depending on the fading and delays power spectra associated with each flight phase, the power spectrum varies. The conclusion section highlights the relevance in distinguishing between different scenarios due to the different effects in the signal.

Reference [37] couples a Monte Carlo simulation with the sum of sinusoids concept. This simulation results in an efficient scheme that has low computational complexity. The authors highlight the importance of such a model to evaluate handover and macro-diversity algorithms. This spcific application of the model steers the choice of a metric towards the outage probability of signal-to-noise ratio. By implementing three models (uniform sampling method, non-uniform sampling method

and Monte Carlo sampling method) and its use, it is found that the non-uniform sampling method provides the best results.

Last but not least, the extensive model is proposed in Reference [29]. In this model, the authors use a combination of measurements and analysis. The analytical part comprises a line of sight (LOS) and ground reflected ray. Both rays and amplitude modification are calculated using terrain information and antenna properties. The first step is determining the terrain type to use the correct samples. Collected and processed samples from measurements are used in the model as power delay profiles. The proposed statistical model can be considered a four-ray model, including the two primary and more reflected rays collected in measurements. The uniqueness of this model is that it uses the extensive measurements campaigns performed by the authors. Such measurement campaigns have a one to one match with the channel application, which simulates an A2G channel.

3.3.3   Geometric models

Geometric models contrast with high-specific/terrain-specific/building/specific empirical measurements or ray-shooting and ray-tracing computer simulations. These models consist of analytical equations which predict mobile radio delay spread power. Contrary to ray-tracing techniques, geometric models do not require such an amount of data to estimate the channel response. Geometric models thus attempt to integrate measurable fading metrics into the propagation channel's idealised geometry. The end goal is to generate a model where only a few geometric parameters would affect the various fading metrics.

The state of the art coverage is grouped by the geometry used to describe the model. The geometry of the model is defined by where the model locates scatters. Three main groups are shared through the literature: elliptical, two-ring and street models. Elliptical models place the scatters in elliptical regions or regions of equal delay. Two ring models assume that scatters are located within a circular perimeter of sender and receiver. Last but not least, street models tailor the geometry and assume scattering is happening in areas parallel to the vehicle's travel direction.

Elliptical geometrical model

The majority of the geometrical channel models use elliptical shapes, as shown in Figure 3.5. Each scatterer can be associated with a determined path length or propagation path. Scatters with the same propagation path can be located in an ellipse with transmitter and receiver at is foci. Authors in [38] propose an elliptical shape to characterise the average received power for one particular delay path. The results of this model are compared against measurements, showing a good agreement under some conditions. The primary condition is that the distance between transmitter and receiver is greater than the delay path.

Reference [39] proposes a three-dimensional ellipsoid to define a scattering region. This model considers the elevation of the UAS. Scatters are then uniformly distributed along the region. The number of scatters, in this case, can be tuned with measurement data to increase the fidelity of the model. Poisson distribution can be used to define the number of scatters to maintain a location-agnostic model. Authors claim this model provides temporal and spatial signal characteristics, particularly useful for simulating antenna array systems.

Authors in [40] apply the principles already developed by [39] and [38] to A2G channels. Using the proposed model allows the authors to characterise the Doppler

(A) Geometrical model proposed by [38]

(B) Geometrical model proposed by [39]

(C) Geometrical model proposed by [40]

(D) Geometrical model proposed by [41]
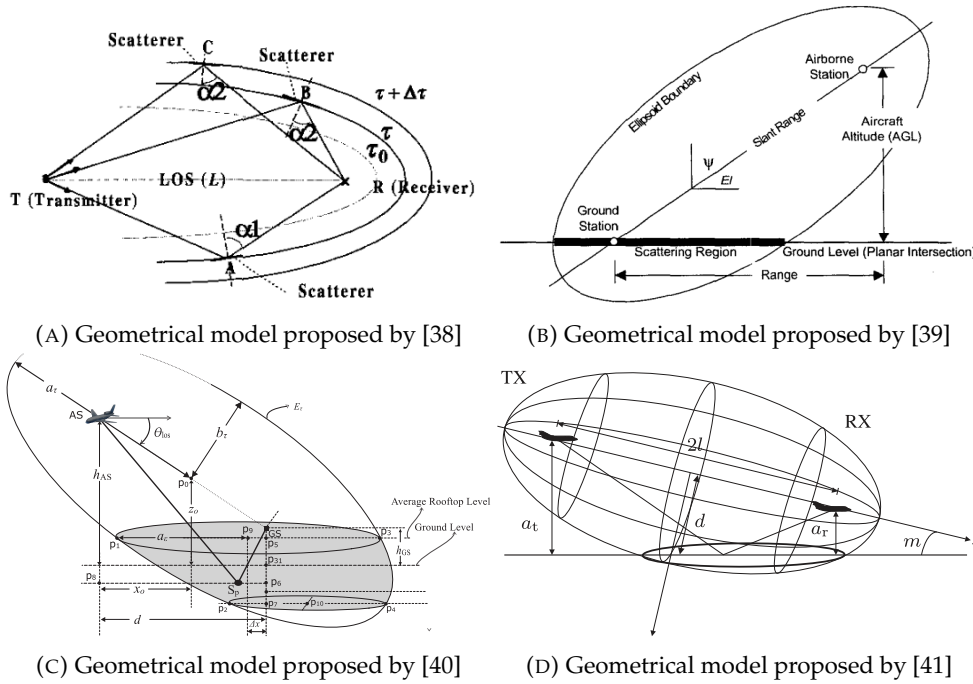
FIGURE 3.5: Geometric models using ellipses to distribute scatters on the ground.

spectrum and spatial spread of the second-order fading statistics. These results are validated against measurements, showing a good match with the model's outputs. The model is not far from the models already discussed. Nevertheless, every equation and concept is extensively described, making this Reference the one with the most detail.

An air to air (A2A) geometric channel model can be found in Reference [41]. Once more, scattering is assumed to come from inside the ellipsoid shape. Since the focus is on the A2A model, both aircraft are located at the focal points. The authors introduce a new concept: to vary the ellipsoid's shape based on the aircraft's respective position. The shape variation introduces changes to the geometrical calculation and directly affects the Doppler frequency probability distribution function. It is worth noticing that it is considered specular when reflection occurs. Although the authors provide an extensive description of the concepts, Reference [41] does not compare the results against real measurements. The lack of measurements spans from something already commented on: it is costly and complex to perform measurements with flying objects, not to mention two flying objects for even one scenario. This view is also shared by Reference [42], where it is also highlighted that the number of trajectories and system configurations covered by flight trials is limited.

A similar concept can be found in Reference [43]. The authors first specify that the aircraft is moving at low altitudes. This assumption is equivalent to assuming a rich scattering environment. The assumption narrows the flight phases to take-off, hold and landing. Since the scenario develops close to the airport, it can be assumed that the ground is even and flat. Distance calculations and position of reflecting point is then simplified as seen in equations 2.7, 3.17 and 3.18. Graphical representation of the variables used in the mentioned equations can be found in Figure 3.6. Once delay formulas are defined, scatters are placed in elliptical contours on the x, y plane.
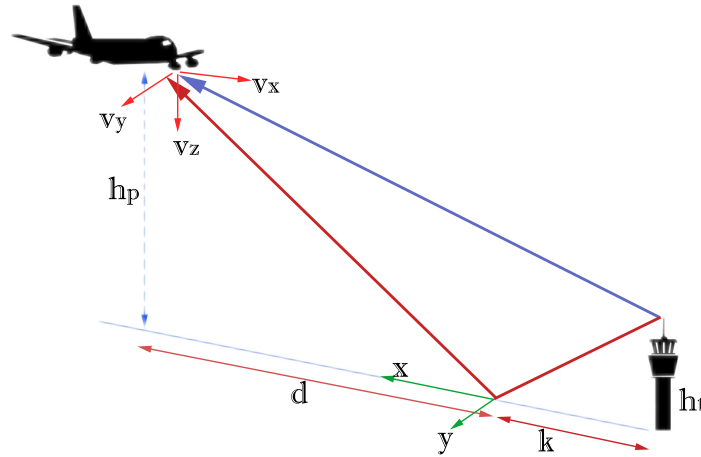
FIGURE 3.6: Geometric model from Reference [43]. This reference provides high details and the mathematical expressions that define the position of each scatter.

Equation 3.19 shows the expression used to locate the scatters in elliptical zones.

$$\tau_{LOS} = \frac{1}{c}\sqrt{d_{LOS}^2 + (h_p - h_t)^2} \tag{3.17}$$

$$\tau_{SPE}(x,y) = \frac{1}{c}\left[\sqrt{h_t^2 + (k+x)^2 + y^2} + \sqrt{h_p^2 + (d-x)^2 + y^2}\right] \tag{3.18}$$

$$y(x,\tau) = \pm\sqrt{ax^2 + bx + c} \tag{3.19}$$

Authors in [44] still consider scattering within an elliptical area. The main difference is that the authors apply the geometrical model for satellite to air communications. Another unique point is the inclusion of the scattering to the LOS and reflected path. The reflected path also incorporates the Fresnel equations with vertical and horizontal polarisation plus the relative permittivity of the reflection surface. The newly introduced channel model is similar to the one presented in this document.

Two-ring geometrical model

The second type of geometrical model mentioned is the "two-ring" scattering model. These geometrical models are particularly interesting for MIMO mobile-to-mobile fading channels. Those scenarios are characterised by being surrounded by many local scatters, as in Figure 3.7a. This figure belongs to Reference [45]. The proposed model is generated assuming an infinite number of local scatters. The infinite number of scatterers implies that the receiving antenna will receive an infinite number of homogeneous plane waves. The gain change caused by each scatter does not consider the reflecting surface. It is also assumed that each scatters introduces an infinitesimal constant gain. Reference [46] is an extension of [45]. In the extension, the authors use the model to prove that the MIMO channel simulation model is ergodic to the mean capacity. The authors do not provide a comparison of the results against other methods or measurements.

Authors in [47] go beyond a 2D representation of the scatters. They introduce a 3D spatial relationship among the mobile station, the scatters, and the BS. The

(A) Geometrical model proposed by [45]

(B) Geometrical model proposed by [47]

(C) Geometrical model proposed by [48]

(D) Geometrical model proposed by [49]
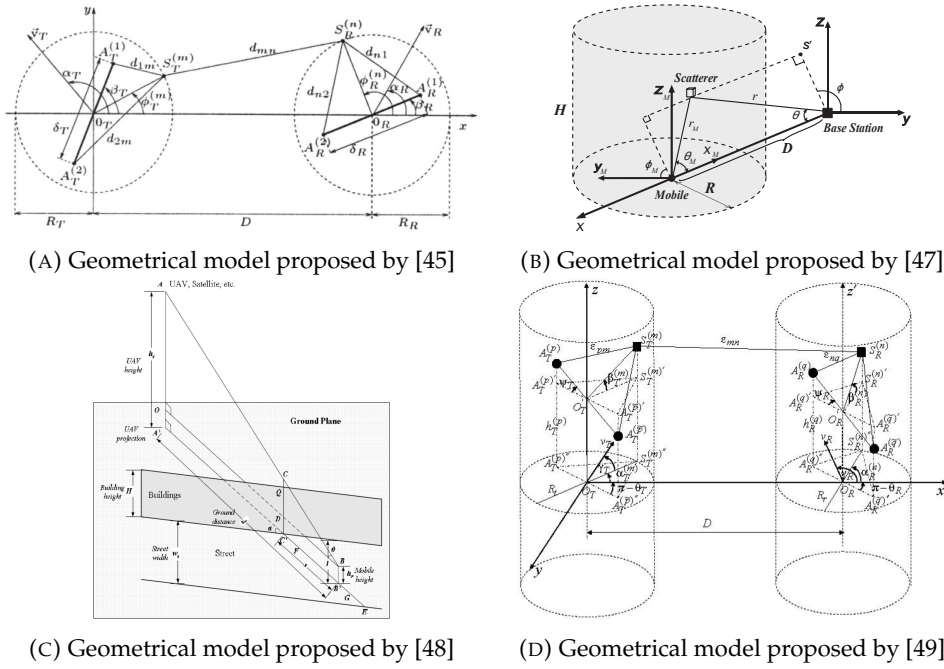
FIGURE 3.7: Geometric models that use other geometrical shapes rather than ellipsoids.

aim is to reproduce better a bad urban propagation channel with the novel cylindrical distribution. Reference [49] expands the concept by merging the cylindrical distribution with the concept that each scatters introduces an infinitesimal constant gain change from [45]. The authors propose a simulation model with a finite number of scatters to circumnavigate the introduction of infinite scatters. Reference [49] ends by proving that the reference model matches the statistical model, assuming an infinite number of scatters.

Street geometrical model

The last variation of geometric models to comment on are those who directly model a street. According to Reference [50], the two-ring model is not suitable for describing propagation in streets. It can be extracted from measurements in [51] that double-bounce rays are predominant for those vehicles driving in the middle lanes. As with other sources, the authors assume that the LOS is blocked.

One of the main conclusions from [50] is that scatters cannot be neglected. The authors found that reducing the relative speed among vehicles causes the channel coherence time to increase. This finding makes the channel model suitable for a highway environment under congestion.

3.4 Radio Link Simulation

The different concepts to understand channel modelling have been explained throughout this chapter. The fundamental equations composing the channel model used in simulations have already been explained. Some equations might be repeated to derive the channel model equations for completeness.

This paragraph briefly describes the main takeaways after studying the current state of the art in channel modelling. As proven by measurements in various areas,

references such as [2] show that a model comprising the LOS ray plus a reflection is enough to characterise the response found in the A2G channel. Indirect observations have shown that the airframe of the UAS can result in considerable shadowing effects. From References [9, 10, 14], it is clear that ray tracing is a suitable tool for inter-vehicle communications. Last but not least, numerous flight tests and simulations highlight that the radiation pattern of both sending and receiving antennas cannot be neglected and should be included as realistically as possible in simulations.

In order to start deriving the model, the first step is to define the slow variations. Variations of the channel are generally classified into very slow, slow and fast. Range-dependent, very slow variations can be modelled using the following equation previously defined (Equation 3.5). With this expression, the model considers the effects caused by the distance between emitter and receiver.

$$P_R = P_T + G_T + G_R + 20 \log \left[\frac{\lambda}{4 \pi d} \right] \tag{3.20}$$

in the previous equation, P_T introduces the emitted power prior to the antenna, G_T is the gain of the transmitting antenna, and G_R is the corresponding gain of the receiver antenna. Last but not least, $20 \log \left[\frac{\lambda}{4 \pi d} \right]$ is the term introducing the distance between transmitter and receiver. The antenna's gain is not constant through the flight and will vary depending on the link's azimuth (α) and elevation (β). To correctly introduce this effect, Equation 3.5 can be modified:

$$P_R = P_T + G_T [\alpha, \beta] + G_R [\alpha, \beta] + 20 \log \left[\frac{\lambda}{4 \pi d} \right] \tag{3.21}$$

Effects caused by the path loss can be seen in Figure 3.8. The path loss tendency significantly reduces the received signal as link distance increases. This effect is common to all communication systems and must be assessed. As shown through the literature review, slow and fast variations are subject to many effects. These effects are shown as variations over the path loss tendency and vary through time. Those effects are typically modelled using distributions such as Rice or Rayleigh if not simulated through ray tracing.

Reference [52] provides an excellent description of modelling the wireless propagation channel. The authors provide a suite of Matlab scripts, functions and data to test the various concepts developed throughout the book. The data shown in Figure 3.9 corresponds to a series of measurements provided by the same Reference where both shadowing and multipath effects are present. The slow variation has been obtained by applying a sliding rectangular window through the series.

Slow and fast variations are expected to be found in the channel. The primary source for slow variations in the channel is the obstruction caused by the structure of the UAS itself. The leading source of fast variations is expected to be the reflection on the ground. This reflection is still to be modelled, but the main equations have already been presented. The combination of Equation 3.13 and Equation 3.16 provides the expression defining the amplitude of the reflected ray:

$$\mathcal{A} = \rho_\| \cdot \rho_s \tag{3.22}$$

$$\mathcal{A} = \frac{\sqrt{\epsilon_2} \cos [\theta_1] - \sqrt{\epsilon_1} \cos [\theta_2]}{\sqrt{\epsilon_2} \cos [\theta_1] + \sqrt{\epsilon_1} \cos [\theta_2]} \exp \left[-8 \left[\frac{\pi \sigma \sin [\theta_1]}{\lambda} \right]^2 \right] \tag{3.23}$$
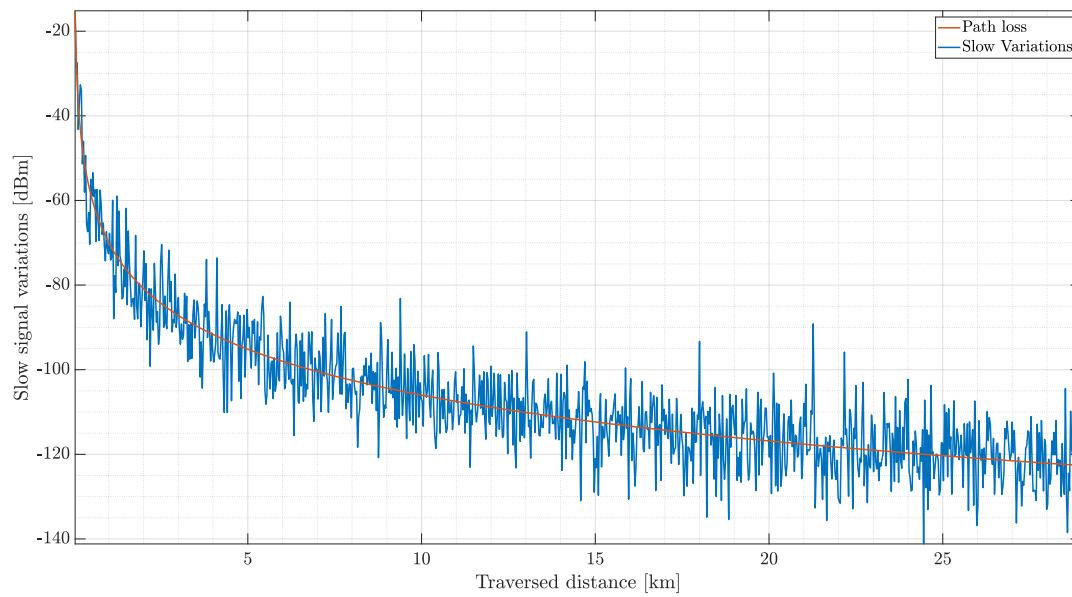
FIGURE 3.8: Simulated path loss together with slow variations due to shadowing
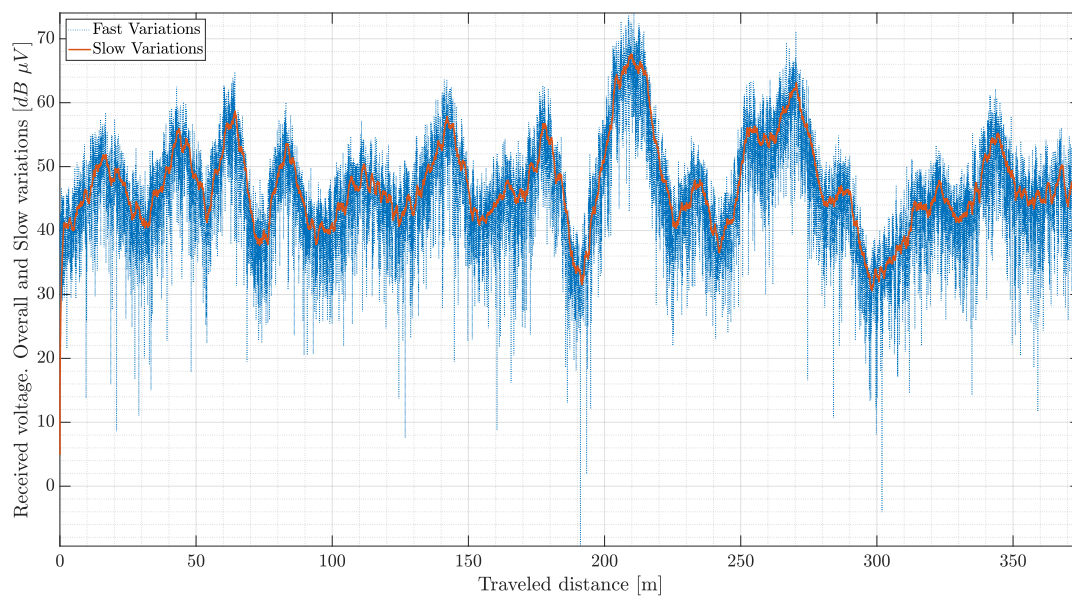
FIGURE 3.9: Slow and fast channel variations

where the first fraction models the reflection that depends on the terrain characteristics, ϵ_2 being the relative permittivity of the propagation media, and ϵ_2 being the relative permittivity of the terrain. The values used for these parameters are displayed in Figure 2.5. The second part of the amplitude models the roughness of the terrain, with σ_h representing the standard deviation of the roughness and θ_i the angle of incidence of the ray on the terrain's surface. λ represents the wavelength at the central frequency.

Permittivity coefficients in Equation 3.23 require terrain information. These coefficients constitute the connection point between the data obtained with Google Maps application program interface (API) and the channel model. The expression shows how the terrain can affect the A2G channel and will do in the form of fast variations. The effect is similar to the fast variations shown in Figure 3.9.

Knowing the amplitude of the incoming reflected ray is the first but not the last step. Before the signal arrives at the receiver, other effects must be taken into account. These effects will provoke changes in the phase of the arriving signal. Reference [3] provides a comprehensive explanation of how the scattering model can be obtained. The first step to derive the scattering model is to consider a horizontally polarised wave arriving at a point in space. The signal will arrive with an azimuth angle α_n an elevation angle β_n. The received electric field is then:

$$E_n\left[t\right] = \mathcal{A} \exp\left[-jka_k \cdot Ra_R\right] \exp\left[j\omega_c t\right] \tag{3.24}$$

where a_R represents the unit vector along with R, k is the wavenumber, a_k is the unit vector along the direction of wave propagation. Vector along xyz gives the following relationships:

$$
\begin{aligned}
ka_k &= k\left[\cos\left[\alpha_n\right]\cos\left[\beta_n\right]a_x + \sin\left[\alpha_n\right]\cos\left[\beta_n\right]a_y + \sin\left[\beta_n\right]a_z\right] & (3.25)\\
Ra_R &= x_0 a_x + y_0 a_y + z_0 a_z & (3.26)
\end{aligned}
$$

The combination of Equation 3.24, Equation 3.25 and Equation 3.26 provides the expression for the electric field:

$$
\begin{aligned}
E_n\left[t\right] = \mathcal{A}_n \exp\Bigg[&j\omega_c t + \phi_n \\
&-j\frac{2\pi}{\lambda}\left[x_0\cos\left[\alpha_n\right]\cos\left[\beta_n\right] + y_0\sin\left[\alpha_n\right]\cos\left[\beta_n\right] + z_0\sin\left[\beta_n\right]\right]\Bigg]
\end{aligned}
\tag{3.27}
$$

where ϕ is a phase shift with respect to an arbitrary reference.

Due to the moving nature of UAS, there is still one effect to be introduced in Equation 3.27: the phase shift caused by the Doppler effect. The UAS has a speed of v in the xy plane. This movement will have a direction that can be described with the angle γ. This angle is calculated to the x plane. In the interval Δt, the UAS will translate to the new xyz coordinates:

$$
\begin{aligned}
x &= x_0 + v\Delta t\cos\left[\gamma\right] \\
y &= y_0 + v\Delta t\sin\left[\gamma\right] \\
z &= z_0
\end{aligned}
\tag{3.28}
$$

plugging this effect into Equation 3.26:

$$
\begin{aligned}
E_n\left[t\right] = \mathcal{A}_n \exp\Bigg[&j\omega_c t + \phi_n \\
&- j\frac{2\pi}{\lambda}\Big[x_0 \cos\left[\alpha_n\right]\cos\left[\beta_n\right] + y_0 \sin\left[\alpha_n\right]\cos\left[\beta_n\right] + z_0 \sin\left[\beta_n\right] \\
&+ v\Delta t \cos\left[\alpha_n - \gamma\right]\cos\left[\beta_n\right] \Big]\Bigg]
\end{aligned}
\tag{3.29}
$$

Once the electric field has been completely defined, the amplitude of the n-reflected ray can be included in Equation 3.29:

$$
\begin{aligned}
E_n\left[t\right] = \frac{\sqrt{\epsilon_2}\cos\left[\theta_1\right] - \sqrt{\epsilon_1}\cos\left[\theta_2\right]}{\sqrt{\epsilon_2}\cos\left[\theta_1\right] + \sqrt{\epsilon_1}\cos\left[\theta_2\right]} \exp\left[-8\left[\frac{\pi\sigma\sin\left[\theta_1\right]}{\lambda}\right]^2\right]\exp\Bigg[&j\omega_c t + \phi_n \\
&- j\frac{2\pi}{\lambda}\Big[x_0 \cos\left[\alpha_n\right]\cos\left[\beta_n\right] + y_0 \sin\left[\alpha_n\right]\cos\left[\beta_n\right] + z_0 \sin\left[\beta_n\right] \\
&+ v\Delta t \cos\left[\alpha_n - \gamma\right]\cos\left[\beta_n\right] \Big]\Bigg]
\end{aligned}
\tag{3.30}
$$

thus, the electric field is completely defined. The relation between the electric field and received power found in Reference [3] can be used to incorporate Equation 3.30 into Equation 3.21:

$$
P_E = \frac{E^2}{120\pi}
\tag{3.31}
$$

$$
P_R = \underbrace{P_T + G_T\left[\alpha, \beta\right] + G_R\left[\alpha, \beta\right] + 20\log\left[\frac{\lambda}{4\pi d}\right]}_{\text{Slow variations}} + \underbrace{10\log\left[\sum_{n=1}^{N}\frac{E_n^2\left[t\right]}{120\pi}\right]}_{\text{Fast variations}}
\tag{3.32}
$$

3.4.1 Channel estimation

A first channel simulation has been designed. This small demonstration intends to glimpse the channel conditions that the UAS will encounter. The area used for this demonstration is located in Munich and covers both the city and the outskirts. Regarding connectivity, several BSs have been located in Munich and nearby towns. This distribution intends to provide connectivity to those UAS travelling from and to the airport located in the north of Munich. The UAS will try to connect to every BS and choose the one with the highest received power, P_R.

The methodology applied to obtain Figure 3.10 is the following. Up to three heights have been selected. Those heights are displayed in each Subfigure (Figure 3.10a) related to mean sea level. Distances are not shown in Latitude and Longitude values to simplify the interpretation of figures.

The various parameters in Equation 3.32 are shown in the different figures. The first parameter to influence the channel response is the terrain profile. The effect of the terrain can be seen in Subfigures 3.10a, 3.10b, 3.10c and 3.10d. Terrain mainly causes obstructions on the LOS, as shown in Subfigure 3.10a. The point at $[-5, 10]$ is the perfect example: there is simply no connectivity because the terrain height is higher than the measured position. It is not possible to gain visibility underground. The statement might seem trivial, but it proves that the algorithm can calculate channel obstructions. Another effect can be found in terrain shadowing, as in Subfigure

3.10d. The highlighted portion of the map shows an elevated area preventing the UAS from connecting to the nearest base station. This effect is reflected in a reduced P_R in the vicinity of such an elevated area.

Next in the list of visible effects is the role of the "slow variations" part of Equation 3.32. This effect is most visible in Subfigures 3.10a, 3.10c, and 3.10e as they only take into account slow variations. The study of the figures shows that the radiation pattern area of influence is close to the BS. In this area, the shape of the radiation pattern of BS antennas shown in Figure 2.22 is visible. As the distance from the BS increases, the signal starts to fade, and the radiation pattern's influence is less and less evident. Because only slow fadings are considered, remote positions show an almost constant P_R.

Last is the effect of the "fast variations" part of Equation 3.32. The effect is present in Subfigures 3.10b, 3.10d, and 3.10f as they consider the entire equation. The first visible effect is that the shape of the radiation pattern in the vicinity of the antenna is no longer delimited. Instead, what can be seen is an unstable channel with a succession of fast variations as distance increases. The unstable variations stabilise, and a series of waves are visible. It is clear that by considering the fast variations, the channel becomes unstable and changes rapidly. The analysis of these figures gives an idea of the optimiser's challenges when finding the route with the best connectivity.

This illustrative simulation proves the different capabilities of the proposed channel model:

1. Compute terrain influence: terrain plays a significant role in the simulation, as described in Section 2.1. The shadowing affects the reception of the signal at certain positions, but the terrain type also influences the reflected ray.

2. Realistic radiation patterns: the effect is visible in the slow variations subfigures. Accurate radiation patterns translate into a better prediction of the channel. Even though not visible in the images, Tilting is an effect that is also reflected in channel estimations.

3. Prediction of slow variations is a fundamental part of every link budget calculation. Including this in the simulation helps the future optimisation algorithm design a trajectory. Thanks to the accurate prediction, the proposed route has a higher chance of success once in operation.

4. Prediction of fast variations: Figure 3.10 shows the signal fluctuations introduced due to reflections. As with the slow variations, making this information available to the route optimiser raises the probability of finding a successful route.

(A) Slow variations at 520 m

(B) Slow and fast variations at 520 m

(C) Slow variations at 596 m

(D) Slow and fast variations at 596 m

(E) Slow variations at 670 m

(F) Slow and fast variations at 670 m
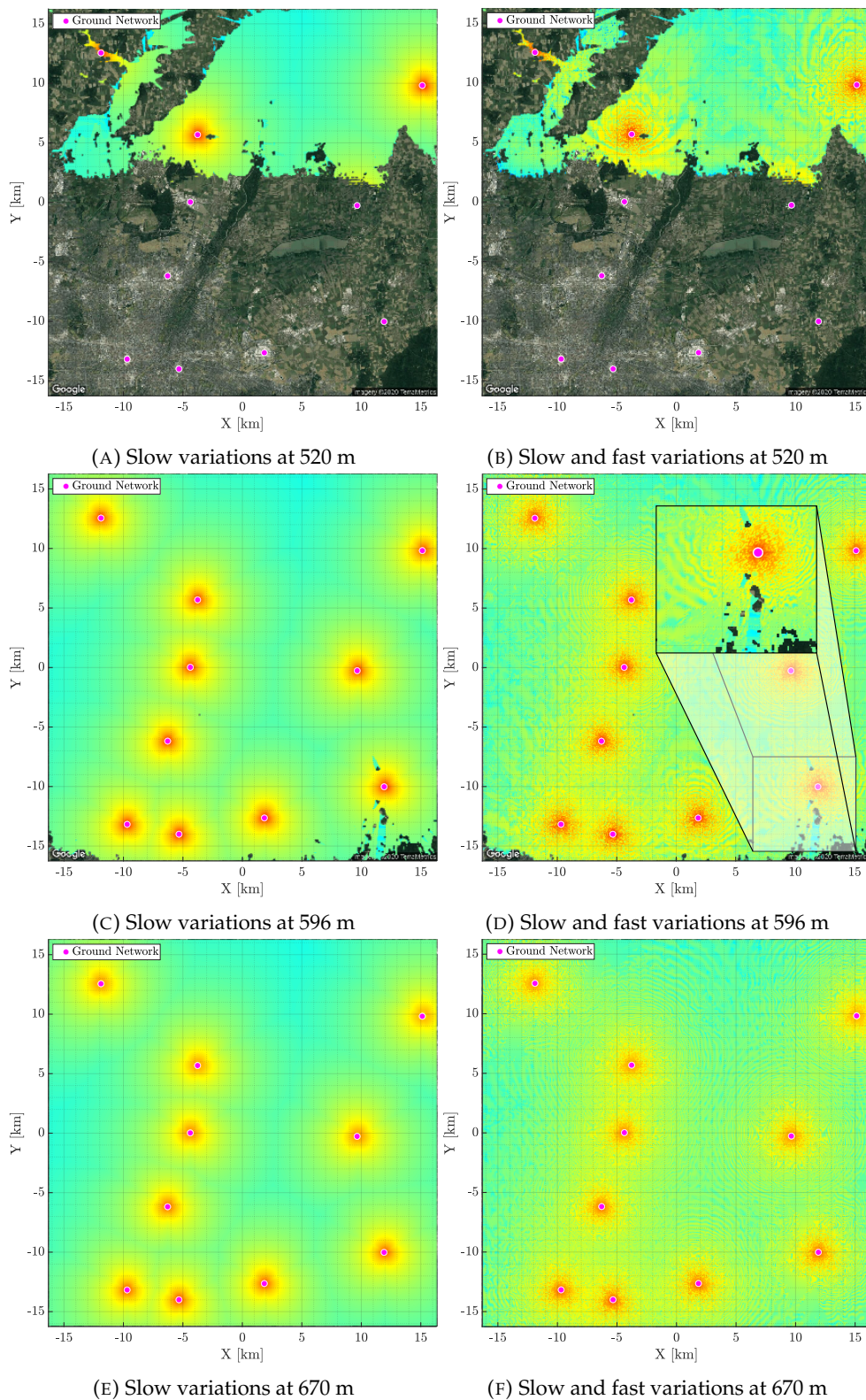
FIGURE 3.10: Channel model simulation at different heights in Munich and its outskirts. Left and right images are depicted to highlight the differences between using only the path loss model and the application of a two-ray channel model.

References

[1] Cheng-xiang Wang, Xiang Cheng, and D Laurenson. "Vehicle-to-vehicle channel modeling and measurements: recent advances and future challenges". In: *IEEE Communications Magazine*. Vol. 47. 11. 2009, pp. 96–103.

[2] David W Matolak. "Air-ground channels and models: Comprehensive review and considerations for unmanned aircraft systems". In: *2012 IEEE Aerospace Conference*. IEEE, 2012, pp. 1–17.

[3] Sana Salous. *Radio Propagation Measurement and Channel Modelling*. 1st ed. Wiley Publishing, 2013.

[4] L Aveneau and P Combeau. "Path finding based on Monte Carlo techniques compared with a full ray-tracing approach in narrow and wide bands". In: *2004 IEEE 59th Vehicular Technology Conference. VTC 2004-Spring (IEEE Cat. No.04CH37514)*. Vol. 1. 3. IEEE, 2004, pp. 271–274.

[5] Dirk Didascalou et al. "Ray-density normalization for ray-optical wave propagation modeling in arbitrarily shaped tunnels". In: *IEEE Transactions on Antennas and Propagation* 48.9 (2000), pp. 1316–1325.

[6] F Aguado Agelet et al. "Efficient ray-tracing acceleration techniques for radio propagation modeling". In: *IEEE transactions on Vehicular Technology* 49.6 (2000), pp. 2089–2104.

[7] Frank Weinmann. "Stochastic scattering model for the application of SBR to rough surfaces". In: *Proceedings of the 5th European Conference on Antennas and Propagation (EUCAP)*. 2011, pp. 6–9.

[8] a. Schmitz et al. "Beam tracing for multipath propagation in urban environments". In: *2009 3rd European Conference on Antennas and Propagation*. 2009, pp. 2631–2635.

[9] Jürgen Maurer et al. "A new inter-vehicle communications (IVC) channel model". In: *IEEE 60th Vehicular Technology Conference, 2004. VTC2004-Fall. 2004*. Vol. 1. IEEE, 2004, pp. 9–13.

[10] Thomas Fugen et al. "Verification of 3D ray tracing with measurements in urban macrocellular environments". In: *2006 First European Conference on Antennas and Propagation*. IEEE, 2006, pp. 1–6.

[11] Gregory Gougeon et al. "3D ray-based propagation channel modeling for multi-layer wireless network performance simulation: Focus on the MIMO channel rank". In: *8th European Conference on Antennas and Propagation, EuCAP 2014*. EuCAP. 2014, pp. 1582–1586.

[12] Y Corre and Y Lostanlen. "3D urban propagation model for large ray-tracing computation". In: *2007 International Conference on Electromagnetics in Advanced Applications, ICEAA'07*. Vol. 33. 0. 2007, pp. 399–402.

[13] Yoann Corre and Yves Lostanlen. "Three-Dimensional Urban EM Wave Propagation Model for Radio Network Planning and Optimization Over Large Areas". In: *IEEE Transactions on Vehicular Technology* 58.7 (2009), pp. 3112–3123.

[14] Kai Daniel et al. "Three dimensional channel characterization for low altitude aerial vehicles". In: *2010 7th International Symposium on Wireless Communication Systems*. IEEE, 2010, pp. 756–760.

[15] Eran Greenberg and Pascal Levy. "Channel characteristics of UAV to ground links over multipath urban environments". In: *2017 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems, COMCAS 2017*. 2017.

[16] F Weinmann. "Ray Tracing With PO/PTD for RCS Modeling of Large Complex Objects". In: *IEEE Transactions on Antennas and Propagation* 54.6 (2006), pp. 1797–1806.

[17] Frank Weinmann. "SBR ray tracing on NURBS for electromagnetic scattering simulations". In: *2012 6th European Conference on Antennas and Propagation (EUCAP)*. 2. IEEE, 2012, pp. 2948–2951.

[18] David W Matolak. "Channel characterization for unmanned aircraft systems". In: *2015 9th European Conference on Antennas and Propagation (EuCAP)*. IEEE. 2015, pp. 1–5.

[19] David W Matolak and Ruoyu Sun. "Air-ground channels for UAS: Summary of measurements and models for L- and C-bands". In: *2016 Integrated Communications Navigation and Surveillance (ICNS)*. IEEE, 2016, 8B2—-1—-8B2—-11.

[20] Ruoyu Sun and David W Matolak. "Air-ground channel characterization for unmanned aircraft systems: The mountainous environment". In: *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*. IEEE, 2015, pp. 5C2—-1—-5C2—-9.

[21] Ruoyu Sun and David W Matolak. "Air–Ground Channel Characterization for Unmanned Aircraft Systems Part II: Hilly and Mountainous Settings". In: *IEEE Transactions on Vehicular Technology* 66.3 (2017), pp. 1913–1925.

[22] David W Matolak and Ruoyu Sun. "Air-Ground Channel Characterization for Unmanned Aircraft Systems: The Hilly Suburban Environment". In: *2014 IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*. IEEE, 2014, pp. 1–5.

[23] David W Matolak and Ruoyu Sun. "Unmanned Aircraft Systems: Air-Ground Channel Characterization for Future Applications". In: *IEEE Vehicular Technology Magazine*. Vol. 10. 2. 2015, pp. 79–85.

[24] Wahab Khawaja, Ismail Guvenc, and David Matolak. "UWB Channel Sounding and Modeling for UAV Air-to-Ground Propagation Channels". In: *2016 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2016, pp. 1–7.

[25] David W Matolak and Ruoyu Sun. "Air-ground channel characterization for unmanned aircraft systems: The near-urban environment". In: *MILCOM 2015 - 2015 IEEE Military Communications Conference*. Vol. 2015-Decem. IEEE, 2015, pp. 1656–1660.

[26] David W Matolak and Ruoyu Sun. "Air–Ground Channel Characterization for Unmanned Aircraft Systems—Part III: The Suburban and Near-Urban Environments". In: *IEEE Transactions on Vehicular Technology* 66.8 (2017), pp. 6607–6618.

[27] Yu Song Meng and Yee Hui Lee. "Measurements and Characterizations of Air-to-Ground Channel Over Sea Surface at C-Band With Low Airborne Altitudes". In: *IEEE Transactions on Vehicular Technology* 60.4 (2011), pp. 1943–1948.

[28] David W Matolak and Ruoyu Sun. "Initial results for air-ground channel measurements and modeling for unmanned aircraft systems: Over-sea". In: *2014 IEEE Aerospace Conference*. July. IEEE, 2014, pp. 1–15.

[29] David W Matolak and Ruoyu Sun. "Air–Ground Channel Characterization for Unmanned Aircraft Systems—Part I: Methods, Measurements, and Models for Over-Water Settings". In: *IEEE Transactions on Vehicular Technology*. Vol. 66. 1. 2017, pp. 26–44.

[30] Ruoyu Sun, David W Matolak, and William Rayess. "Air-Ground Channel Characterization for Unmanned Aircraft Systems—Part IV: Airframe Shadowing". In: *IEEE Transactions on Vehicular Technology* 66.9 (2017), pp. 7643–7652.

[31] Tanja Pelzmann et al. "Airborne measurements enhancing the satellite-to-aircraft channel model in L-band". In: *2016 10th European Conference on Antennas and Propagation, EuCAP 2016*. 2016.

[32] Ruoyu Sun and David W Matolak. "Over-Harbor Channel Modeling with Directional Ground Station Antennas for the Air-Ground Channel". In: *2014 IEEE Military Communications Conference*. IEEE, 2014, pp. 382–387.

[33] Nicolas Schneckenburger et al. "Measurement of the l-band air-to-ground channel for positioning applications". In: *IEEE Transactions on Aerospace and Electronic Systems* 52.5 (2016), pp. 2281–2297.

[34] Erik Haas. "Aeronautical channel modeling". In: *IEEE Transactions on Vehicular Technology*. Vol. 51. 2. 2002, pp. 254–264.

[35] Peter Hoeher. "A Statistical Discrete-Time Model for the WSSUS Multipath Channel". In: *IEEE Transactions on Vehicular Technology*. Vol. 41. 4. 1992, pp. 461–468.

[36] Peter Hoeher and Erik Haas. "Aeronautical Channel Modeling at VHF Band". In: *Vehicular Technology Conference, 1999. VTC 1999 - Fall. IEEE VTS 50th*. Vol. 4. 1999, pp. 1961–1966.

[37] Xiaodong Cai and Georgios B Giannakis. "A Two-Dimensional Channel Simulation Model for Shadowing Processes". In: *IEEE Transactions on Vehicular Technology*. Vol. 52. 6. 2003, pp. 1558–1567.

[38] S Yun-di Lien and M Cherniakov. "Analytical approach for multipath delay spread power distribution". In: *IEEE GLOBECOM 1998 (Cat. NO. 98CH36250)*. Vol. 6. IEEE, 1998, pp. 3680–3685.

[39] W G Newhall and J H Reed. "A geometric air-to-ground radio channel model". In: *MILCOM 2002. Proceedings*. 2. IEEE, pp. 632–636.

[40] Sardar Gulfam et al. "A Novel 3D Analytical Scattering Model for Air-to-Ground Fading Channels". In: *Applied Sciences*. Vol. 6. 8. 2016, p. 207.

[41] Michael Walter et al. "Analysis of non-stationary 3D air-to-air channels using the theory of algebraic curves". In: *IEEE Transactions on Wireless Communications* 18.8 (2019), pp. 3767–3780.

[42] Nicolas Schneckenburger et al. "Modeling the air-ground multipath channel". In: *2017 11th European Conference on Antennas and Propagation (EUCAP)*. IEEE, 2017, pp. 1434–1438.

[43] Mostafa Ibrahim and Huseyin Arslan. "Air-Ground Doppler-delay spread spectrum for dense scattering environments". In: *MILCOM 2015 - 2015 IEEE Military Communications Conference*. Vol. 2015-Decem. IEEE, 2015, pp. 1661–1666.

[44] Zaixue Wei et al. "Analytical non-stationary satellite to aircraft channel modeling over open area based on regular shaped geometry-based stochastic model". In: *Applied Sciences* 10.15 (2020), p. 5041.

[45] M Patzold et al. "A MIMO Mobile-To-Mobile Channel Model: Part I - The Reference Model". In: *2005 IEEE 16th International Symposium on Personal, Indoor and Mobile Radio Communications*. Vol. 1. IEEE, 2005, pp. 573–578.

[46] B O Hogstad et al. "A Mimo Mobile-To-Mobile Channel Model: Part II - The Simulation Model". In: *IEEE 16th Int. Symp. Personal, Indoor and Mobile Radio Commun.* Vol. 1. 4876. 2005, pp. 562–567.

[47] A Y Olenko Qasmi, K T Wong, and Qasmi S. A. "SPC04-4: Distribution of Bad Urban Uplink Multipaths' Arrival Delay and Azimuth-Elevation Arrival Angles, with Scatterers Distributed Cylindrically Above the Mobile". In: *IEEE Globecom 2006*. IEEE, 2006, pp. 1–6.

[48] Qixing Feng et al. "WLCp2-06: Modelling the Likelihood of Line-of-Sight for Air-to-Ground Radio Propagation in Urban Environments". In: *IEEE Globecom 2006*. Vol. 2006. IEEE, 2006, pp. 1–5.

[49] Alenka G Zajic and Gordon L Stuber. "3-D MIMO Mobile-to-Mobile Channel Simulation". In: *2007 16th IST Mobile and Wireless Communications Summit*. July. IEEE, 2007, pp. 1–5.

[50] Ali Chelli and Matthias Patzold. "The Impact of Fixed and Moving Scatterers on the Statistics of MIMO Vehicle-to-Vehicle Channels". In: *VTC Spring 2009 - IEEE 69th Vehicular Technology Conference*. IEEE, 2009, pp. 1–6.

[51] Alenka G Zajic et al. "Statistical modeling and experimental verification of wideband MIMO mobile-to-mobile channels in highway environments". In: *2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications*. IEEE. 2008, pp. 1–5.

[52] F Pérez Fontán and P Mariño Espiñeira. *Modeling the Wireless Propagation Channel*. Chichester, UK: John Wiley and Sons, Ltd, 2008, p. 272.

Chapter 4

Route Optimisation

4.1 Introduction

Route optimisation is a fundamental aspect of the guidance of an autonomous vehicle. This optimisation has been studied primarily in robotics and dynamics and control. Although they both share typical treats, robotics has its focus on robot control and computational issues, while the latter will traditionally focus on dynamics and trajectory performance. Whilst a unmanned aerial system (UAS) might seem like a robot-like system, it has more differences than similarities. Any study of a UAS has to deal with non-trivial dynamics, operational disturbances, a changing three-dimensional environment, and high levels of uncertainty. If someone wants to simplify the flight environment, it will also face several non-trivial decisions involving where to sacrifice accuracy for simplicity.

Aside from the challenging ever-varying environment, UAS are supposed to navigate and the non-trivial dynamics, the guidance of these vehicles can apply robotic motion planning techniques. Mission characteristics are shared between robotic motion planning and UAS path planning: partial understanding of the environment, reconnaissance, surveillance goal interception missions. One essential shared characteristic is the need for continuous interaction and knowledge about the environment. Past chapters have been devoted to the fundamentals and description of the physical and radiofrequency environment the UAS will face.

The main differences between classic robotics and UAS motion planning have been the subject of interest. Among the various topics this chapter covers, some topics stand out like differential constraint handling, limited knowledge about the environment, and complex objective functions with no available analytical formulation.

Optimisation theory and methods is a mathematical theory with a field of application wider than path planning. For example, Reference [1] applies optimisation theory to improve the response of a radio frequency filter. According to Reference [2], science, engineering, business, military and space technology can also benefit from the research and advances in the optimisation theory field. As a result of this broad applicability, the search for a state of the art optimisation methods is not bounded to UAS but can also extend to other optimisation applications.

This chapter covers optimisation approaches and concepts applicable to UAS path planning. As with previous chapters, the first sections provide a general overview of the used concepts and algorithms. Once the fundamentals are presented to the reader, state the art is presented. This section gathers the most recent publications related to optimisation and path planning. This study is performed from a general standpoint without guaranteeing that the developed methods will be the final selected ones. After presenting the current research directions and advances, the methods to incorporate constraints and the candidate cost functions are introduced. The
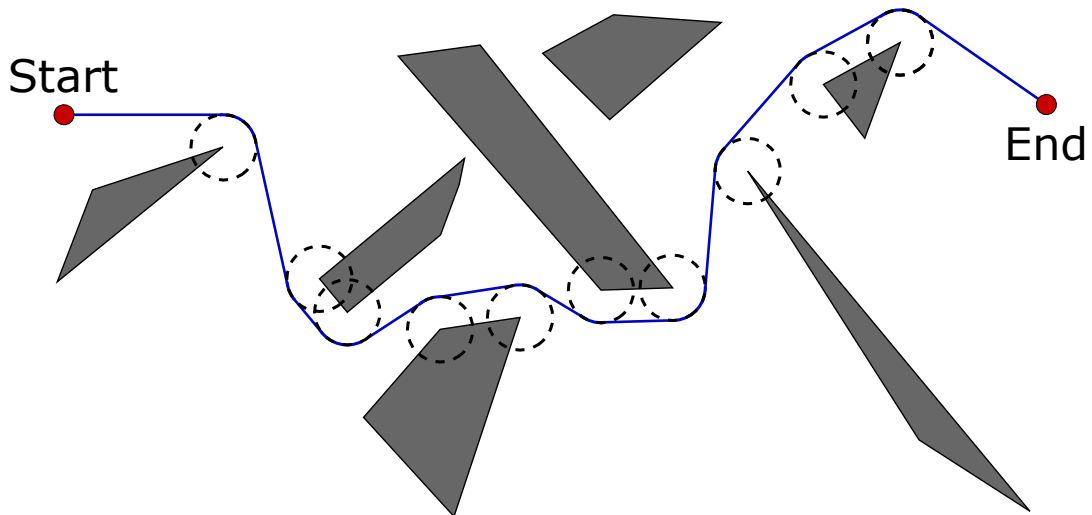
FIGURE 4.1: Trajectory followed by a NonHolonomic vehicle

chapter ends with an enumeration and detailed description of the selected optimisation algorithms. Those algorithms are the predominant ones, and their performance is discussed in the following chapters.

4.1.1 Optimisation problems and methods

Before tackling the optimisation of a UAS, researchers were focused on ground vehicle optimisation. Reference [3] tries to optimise the route of a nonholonomic vehicle among obstacles. The reference raises interest because of those nonholonomic vehicles. In short, these types of vehicles are characterised by having non-integrable equations that describe constraints. For example, a nonholonomic constraint is the turning radius of a car.

The first procedure to obtain a successful trajectory does not need further computation if the scenario is sufficiently simple. It starts by drawing a series of circles in each obstacle vertices. Each circle has a radius equivalent to the turning radius required by the vehicle. By connecting the successive circles, the vehicle is ensured to travel the minimum distance and keeps a safe distance from each obstacle. The result of such a procedure can be seen in Figure 4.1. The vehicle safely travels from the starting point to the endpoint. The vehicle navigates through the scenario without colliding with any obstacle with this procedure.

This first descriptive example represents the case where the vehicle is described as a point. The problem that the optimisation needs to solve depends on the representation of the vehicle and its characteristics. In this sense, the first approach is to consider the vehicle as a point. The vehicle is then modelled as a point inside a bounding ball. This modelling requires regions around each obstacle vertice. Optimally speaking, this problem is defined as the distance between initial and endpoints.

The complexity of the point vehicle can arise by adding differential constraints. Each state of the UAS has to satisfy the equations of motion of the vehicle. Typically these equations are associated with Newton's second law. Limits on velocity and acceleration are defined to match the constraints of the UAS and its flight capabilities.

The case of a point vehicle with differential constraints is best optimised in static environments. A path can be obtained before the actual flight takes place. For a dynamic environment with unpredictable changes, it is best to move to a Jogger's

problem. This problem considers that the vehicle has no knowledge of the search space and has a limited field of view attempting to reach its destination.

A higher level of complexity in optimisation problems is achieved considering a vehicle with differential constraints. Nonholonomic constraints, already mentioned, use kinematics constraints. The other problem is the constraints imposed by the vehicle's dynamics. The consideration of differential constraints makes it insufficient to model the vehicle with only a point in the world space. Differential constraints impose the need for six variables to model the vehicle's position correctly.

Numerous methods can be applied to solve the optimisation problems already described. The list of methods that can be applied is broader and can be found in [4]. The described methods are those used by the selected related works mentioned later. The best way to cover them is to divide the methods by the type of constraints they can tackle.

The first group of methods is suitable for problems that do not involve differential constraints. Roadmap methods are a subclass involving methods that reduce the problem to a graph search. The well-known Dijkstra's algorithm belongs to this subcategory, and it is used to find the shortest paths between nodes in a graph. Those nodes generally represent a road network. One similar approach is called a visibility graph which tries to find the shortest route while avoiding obstacles. The shortest path might come close to obstacles, leading to hazardous situations for the vehicle's integrity. This issue can be solved by expanding the obstacle space and allowing a safe margin for the vehicle. Voronoi roadmap attempts to solve the issue of the vehicle moving too close to the obstacle. The approach builds a skeleton that guarantees a maximum distance from obstacles. Later, the optimum path for minimum distance is computed following said skeleton.

Approximate cell decomposition methods are gaining popularity because of their efficiency. Tree decomposition, quadtrees and octrees belong to cell decomposition methods. The search space is recursively divided into smaller areas to represent the presence or absence of obstacles. This division reduces the search space by grouping areas with the same content. Roadmap methods can be combined with cell decomposition to reduce computation times while keeping a resolution-complete search space.

The potential field method is the last suitable family of methods suitable for problems without differential constraints. This family of methods assign a potential function to the free space. The vehicle resembles a particle reacting to the forces due to the potential field. This point is modelled as the point with the lowest potential for the vehicle to reach its destination. Points surrounding obstacles have a higher potential, repelling the vehicle. The search space must be discretised into a grid of points, with the consequent rise of computational time as the number of points increases. Methods that belong to this family are potential fields with gradient descent method, the potential field with guided search method or harmonic potential method.

As opposed to ground vehicles, the behaviour of UAS cannot be well described by their kinematics. Any planner tackling the trajectory of a UAS should consider the equations of motion of the aerial vehicle. This requirement equals using differential constraints. In other words, time and the state-space are tied in a way that cannot be directly solved. Even the most simple use case involving the connection of two states without obstacles has no immediate solution. Mainly two families are described due to their relevance in the literature: mathematical programming and receding horizon control. More algorithms and their description can be found in Reference [4].

Mathematical programming considers the planning process as a numerical optimisation problem. One of the characteristics of these methods is that the cost functions have an unknown number of local minima. Finding the global solution depends strongly on the initial guess. The optimisation converges in polynomial time when the correct initial guess is selected. Known methods that belong to this family are mixed integer linear programming (MILP), nonlinear programming, and other constrained optimisation problems.

Model predictive control (MPC), also known as receding horizon control, solves the numerical optimisation problem over a reduced time horizon. In this method, information gathered by the sensors in the UAS can be used to search for the new location. Since only local information is used, the computational time can be reduced. However, care must be taken to design the cost function that guarantees completeness and near-optimality. The optimisation will not converge to a globally optimal solution without such a carefully designed cost function.

Intelligent optimisation algorithms are the last family of methods targeting problems with differential constraints. The family name was given by Reference [5] and [6]. This family of methods has been gaining particular interest over the last years, especially in 3D path planning. The methods include a set of nature-inspired computational methodologies and approaches. Nature-inspired algorithms are a great tool combined with cost functions with many local minima. Despite the popularity of these methods in a stand-alone version, their performance can be significantly improved by combining these algorithms with direct search algorithms. evolutionary algorithm (EA), particle swarm optimisation (PSO) and ant colony optimization (ACO) are part of the intelligent optimisation algorithm family.

4.1.2   Path smoothing

Regardless of the associated constraints, the selected optimisation method proposes a series of points for the UAS to fly. The followed trajectory is then analysed, and a metric associated with it is based on a cost function. This trajectory and how it is followed is known as the navigation task. According to Reference [7], the navigation tasks can be broken down into three main tasks: location, path planning and plan execution.

Each path planner has a common core: the plan representation. Such a plan consists of a sequence of path segments. The path segment has only three states for the simplest vehicles: straight line, maximal turn to the right or maximal turn to the left. This plan representation is sufficient for the simplest of vehicles to navigate through the environment safely. The next step to increase the complexity and flexibility of a path is to use angle bearings, curvature radius and length. Through this, the vehicle can adapt to changes in obstacles.

Naturally, a representation sufficient for simple vehicles is not for such a UAS. The authors in Reference [8] propose B-spline curves to represent the path of the UAS. As defined by Schoenberg in the 1940s, a B-spline curve is composed of a piecewise polynomial curve comprising several polynomial segments. These curves are continuous, making these curves most desirable for aircraft trajectories; no corrections nor adjustments are required to have a smooth trajectory.

B-spline or Bezier curves are used in Reference [9]. The application benefiting from these geometrical figures is airspace management. One of the advantages of using a Bezier curve is that airspace boundary constraints are satisfied by construction. The second advantage is the implementation's numerical efficiency, although
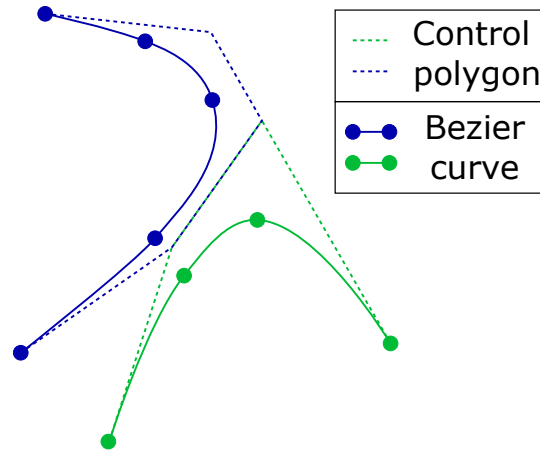
FIGURE 4.2: Method for path smoothing using Bezier curves to avoid conflicts between moving UASs as proposed by Reference [9].

it is recursive. Figure 4.2 displays the use case of Reference [9], two UAS in a collision situation avoided by using the Bezier curve. The authors acknowledge that the generated trajectories are ideal, and the dynamics of the UAS can modify them.

Reference [6] comments on imposed modifications by obstacles in the environment. The proposal is to perform a two-step approach. In the first step, a straight path that avoids obstacles is generated. Once the UAS has an obstacle-free trajectory, such trajectory is smoothed using Bezier curves. The two-step approach can increase the computational cost of the trajectory search.

The proposed approach in this work has already been introduced in Subsection 2.2.3. The selected optimisation approaches will propose a succession of Waypoints as defined by Equation 2.14. As it has already been mentioned, this approach reduces the search space as opposed to Waypoints purely defined by Latitude, Longitude and Altitude.

The proposed definition of Waypoints, combined with our simulation software Analytical Graphics Inc. Systems Tool Kit (AGI-STK) [10], ensures a more realistic flight path. The resulting flight path is smooth and already incorporates first and second-order dynamics. The sometimes used requirement that the UAS passes through each waypoint is relaxed. In the proposed method, the UAS can circumnavigate a Waypoint. Exceptions are first and last Waypoints. There is no need for Bezier curves in the proposed method; the UAS navigating through the Waypoints generates a smooth trajectory.

4.1.3 Constraint handling

The most desirable outcome of a route planning algorithm is a route compliant with all the objectives. Nevertheless, some particularities of the scenario cannot be defined as an objective. No fly zones (NFZs), the terrain itself, life-threatening areas or even adversary weather conditions constitute regions to be avoided. A route might be compliant with all the objectives by being the fastest one, for example, but might navigate through undesirable areas. Constraints can force the optimisation algorithm to avoid those undesirable areas.

As in our approach, Reference [11] uses a grid to describe the terrain and elevation. In the authors' proposed approach, the algorithm assigns a cost to each map cell. This initial cost is based purely on the objectives and the consequences of the

UAS passing through this particular cell. In a second step, the constraints are considered by generating another cell-based map with values. The problem with this approach is that the generation of the constraints map is heavily based on experience and expert knowledge. Each map and its threat configuration requires expert knowledge and custom assignment of constraint values.

Authors in [12] propose a two-level approach prioritising constraints rather than objectives. In their approach, the constraints are satisfied first. These constraints relate to the physical restrictions of the UAS and terrain. In this sense, constraints constitute objectives that must be satisfied for a trajectory to be successful. The second level, objectives, are tackled once constraints are satisfied. In this sense, objectives are tailored for each mission. Because of the application, Reference [12] defines objectives tied to the survival of the UAS. The authors have selected the EA as an optimisation algorithm with promising results.

Numerous references cover constrained optimisations, such as Reference [13] or [14]. Reference [15] is unique in the sense that they provide a view of the approaches that are currently used. Penalty functions are the first group to be addressed. These functions increase the overall cost function value when constraints are violated. These functions might seem simple; nevertheless, they require a fine-tuning of their penalty factors. One of the most common penalty factors is the "death penalty". This scheme assigns the worst possible value when a constraint is violated.

The objective function and constraint can be divided into two functions. In this approach, a feasible solution's cost function value is only based on the cost function value. Solutions violating a constraint have a cumulative value based on the accumulated constraint violation. One of the highlighted drawbacks in Reference [15] of this approach is the loss of diversity. This loss of diversity can be overcome with appropriate diversity maintenance mechanisms.

Another approach with similar issues is the use of feasibility rules. Reference [16] provides three simple rules to deal with constraints. The first rule is to select the solution with the highest fitness value. The second rule applies when there is a tight between two solutions. In this case, the feasible solution should be selected if the other is not feasible. If both solutions are unfeasible, the one violating fewer constraints is preferred. This set of rules, albeit simple, is highly effective and easy to apply to any optimisation method. As with the division of the objective function, this method is prone to premature convergence.

Looking into the future of constraint handling, according to Reference [15], four approaches are the most promising. Many-objective, different to multi-objective, considers many objectives (more than four). This optimisation is strongly related to constraint-handling techniques. The difference is that such constraints are considered an objective. The second promising approach is the approximation of constraints. The approximation promises to be very valuable for real-world applications. The third approach on the list is the use of dynamic constraints. As the name indicates, such constraints will change over time. For example, an NFZ could change its position or size over time. Reference [17] is one of the latest examples using dynamic constraints. The last future trend is to use hyper-heuristics. These methods are popular in combinatorial optimisation but have never been applied to constrained problems. They are a possible future trend because of their similarities with constraint-handling techniques.

4.2 State of the Art

Some highlights can be obtained after studying possible optimisation algorithms, techniques for a smooth trajectory and handling of optimisation restrictions. The first highlight revolves around the numerous alternatives. From non-intelligent algorithms to nature-inspired algorithms, and the combination (hybridisation) of multiple algorithms, the number of possible algorithms to tackle optimisation problems is high. The numerous optimisation algorithms demand a classification. The following references are classified between non-intelligent algorithms, EA, PSO and those references proposing a hybrid approach. EA and PSO have their category due to the importance those two algorithms have later on in this document.

The second highlight is that the time to complete a trajectory can be optimised. The use of multiple objectives can influence the trajectory of the UAS towards other goals different than the time of flight or energy consumption. This division will not be used, but the reference topics will be limited. The topics of interest are: data gathering, UAS acting as a relay, UAS providing coverage for ground users and traditional flight objectives. As the reader might notice, the topics are either related to the optimisation of connectivity capabilities of the UAS or traditional flight objectives.

The third and final highlight is that optimisation is a field with vast applications, not only in path planning. Optimisation can be used in economics, civil engineering, control, geophysics and molecular modelling. Those fields of applications have not been studied. Even though they might propose alternatives, the border-line has been cut in those references in electrical engineering and mechanics. Electrical engineering because of the communication and its characteristics to determine optimisation objectives. Mechanical engineering because of the objectives related to flight time, obstacle avoidance or flight performance that the UAS route must be compliant.

4.2.1 Non-intelligent algorithms

These types of optimisation algorithms have opposing characteristics concerning intelligent algorithms. Non-intelligent algorithms are those not inspired by nature's processes. Multiple algorithms can be found within this category: Dijkstra, MILP, or MPC. They do not produce good results when coupled with cost functions with many local minima. The advantage of this group is that time for convergence is lower than the time required by intelligent algorithms.

The Dijkstra algorithm was first proposed in 1956, and it has become the most used to find the shortest path between nodes. Although the most used, this does not imply that it has not suffered any modifications. Authors in Reference [18] propose a weighted cost function, naming the algorithm D*. The fundamental idea is to mix Dijkstra with breadth-first search (favouring vertices close to the goal) in the cost function. Equation 4.1 displays the proposed cost function: g is the Dijkstra component, and h is the heuristic (breadth-first search) component. This approach is beneficial in environments where a low-resolution map is available, and the UAS is required to avoid unexpected obstacles. The low resolution of the map makes the path not smooth enough for the UAS. Because of this, the authors propose B-splines to smooth the trajectory.

$$F[X, G] = w_1 \cdot G[S, X] + w_2 \cdot H[X] \qquad (4.1)$$

Theta* is an adaptation of the original Dijkstra for three-dimensional path planning. Reference [19] proposes such adaptation. One of the main drawbacks is the computation time resulting from the addition of the third dimension. Nodes below terrain or inside obstacles are masked out from any calculation to reduce computation time. Reference [18] requires a smoothing strategy because kinematics are not part of the path generation. It is worth noticing that out of 52 simulations, 11 showed no convergence. Although these algorithms are capable of obstacle avoidance, the computational needs of this optimisation algorithm discard it as a good candidate.

The optimisation algorithm known as MILP is the one chosen by authors in [13], [20], [21] and [22]. Limitations of the algorithm itself push the authors to consider the movement of the UAS as lacking acceleration and without height variation. Naturally, the authors justify this design by assuming that UAS will behave and organise the same as conventional aircraft. The non-linearities are removed from the equation with these assumptions, and MILP can be applied.

The authors in [20] provide an interesting approach. It consists of a two-step optimisation where MILP proposes a route, and the quality of the connectivity is addressed in a different simulator called SPLAT. For non-compliant routes, the algorithm reduces the constraint on the minimum distance between UAS. After the reduction of this constraint, MILP optimisation is invoked once more. The process is repeated until a satisfactory path is found. Reference [20] is the only applying MILP that considers the third dimension.

Moving on to the following algorithm, we can find MPC. Authors traditionally apply this algorithm to applications focused on gathering as much data as possible from a scenario. Reference [23] applies MPC to a swarm of UAS to search and localise a stationary target. The novelty of this work is to tackle the myopic issue within MPC algorithms. The proposal is to start with a reduced horizon and increase it if no searchable areas are found. The topic of searchable areas is also introduced to help and direct the UASs towards the goal.

Reference [24] provides an application bridging search and rescue and coverage enhancement. The chosen scenario is a group of UASs adapting their positions to provide the best coverage for mobile vessels. Authors acknowledge that real-time optimisation is not feasible and opt for a snapshot. Each UAS finds the best route at each interval using the MPC algorithm. Authors do also include collision avoidance restrictions. The coverage is only modelled by communication range.

MPC is the chosen algorithm in references [25] and [26]. The UAS is expected to form a communication network to multiple targets with given radio communication capabilities in both works. The chosen metrics by the authors differ; nevertheless, the final application remains the same. As with references [23] and [24], it is assumed that the UAS knows where the best conditions might reside and applies this knowledge in the searching process. Authors have applied ad-hoc versions of the optimisation algorithm without claims that it has been modified.

Authors in Reference [27] provide an exciting optimisation use case. Instead of focusing on data gathering, data relay, or link quality, authors optimise the distance between two entities to avoid eavesdropping from a third one. The distance is optimised to make communication possible between the legitimate actors, but the eavesdropping entity perceives a degraded signal. The authors chose block coordinate descent and successive convex optimisation to modify the trajectory accordingly, algorithms that fall in the same family as MILP. One of the assumptions is that the position of the evil entity is known. This assumption cannot be fulfilled in a real scenario as the evil entity will hide its position as long as possible.

4.2.2   Genetic algorithm

A specialised section is devoted to new advances in EA for UAS route optimisation. Other optimisations will receive this special treatment based on relevance in the following pages. A simple yet representative example can be found in Reference [28]. This reference proposes offline and online path planning. The offline path planning is used for known terrains, whereas the online path planning is applied to unknown terrains. The online planner is simplified by using fewer constraints.

One of the most notorious contributions to this thesis from Reference [28] is Equation 4.2, where w_i and a_i are weights and objectives respectively. This equation allows to introduce multiple objectives without modifying the EA algorithm implementation. Authors introduce objectives such as terrain avoidance, safety distance to obstacles and compliance with nonholonomic constraints of the vehicle. The absence of an objective to reduce flight time deserves mention. This missing objective is common to all previous references, a paramount one to reduce the energy consumption of the proposed route. To obtain a smooth path, the authors apply those methods mentioned in previous Subsection 4.1.2.

$$F = \frac{1}{\displaystyle\sum_{i=1}^{4} w_i \cdot a_i} \tag{4.2}$$

Reference [14] is the continuation of Reference [12]. In the earlier work, the authors first introduce the concept of constraints and optimisation indexes. On the one hand, constraints are conditions that each path must fulfil. A violation of one of the constraints makes a path non-viable, and the individual is removed from the pool. Restrictions considered are terrain avoidance, turning radius (nonholonomic constraint), map limits, maximum climbing and diving slope, NFZs and fuel. On the other hand, optimisation indexes determine the optimality of the path. Those indexes are minimum length path, minimum probability of kill, minimum flight altitude and minimum radar detection probability.

The later work in Reference [14] introduces essential modifications to their proposal. The first and most important is to move from Cartesian Waypoints to relative polar coordinates. Relative polar coordinates have already been developed in Subsection 2.2.3. Some implications are drawn: the search space is reduced with this approach, but a modification of initial waypoint parameters leads to the modification of the entire path. Another modification is leaning towards a distributed optimisation rather than a centralised one. Each UAS optimises its route whilst avoiding a possible collision with other UASs in the vicinity. Authors also reduce computation times by using knowledge acquired during previous optimisations. Parameters are tuned after a statistical performance analysis. Whilst Reference [12] only provides a listing of restrictions and optimisation indexes, Reference [14] provides the complete expressions for each parameter to be tuned.

Authors in Reference [29] apply some assumptions to simplify the search space. One of those assumptions, also shared with Reference [20], is the constant height and speed. The optimisation is used to find a path that allows a UAS swarm to collect data from sensors. Naturally, the location of the sensors is known beforehand. For simplicity, the vehicle is not required to fly exactly over the sensor location.

The canonical EA workflow is not maintained in Reference [29]. Not all phases of EA are executed in sequential order: selection, crossover and mutation are performed on separated cores to obtain a path for each UAS. After mutation, local
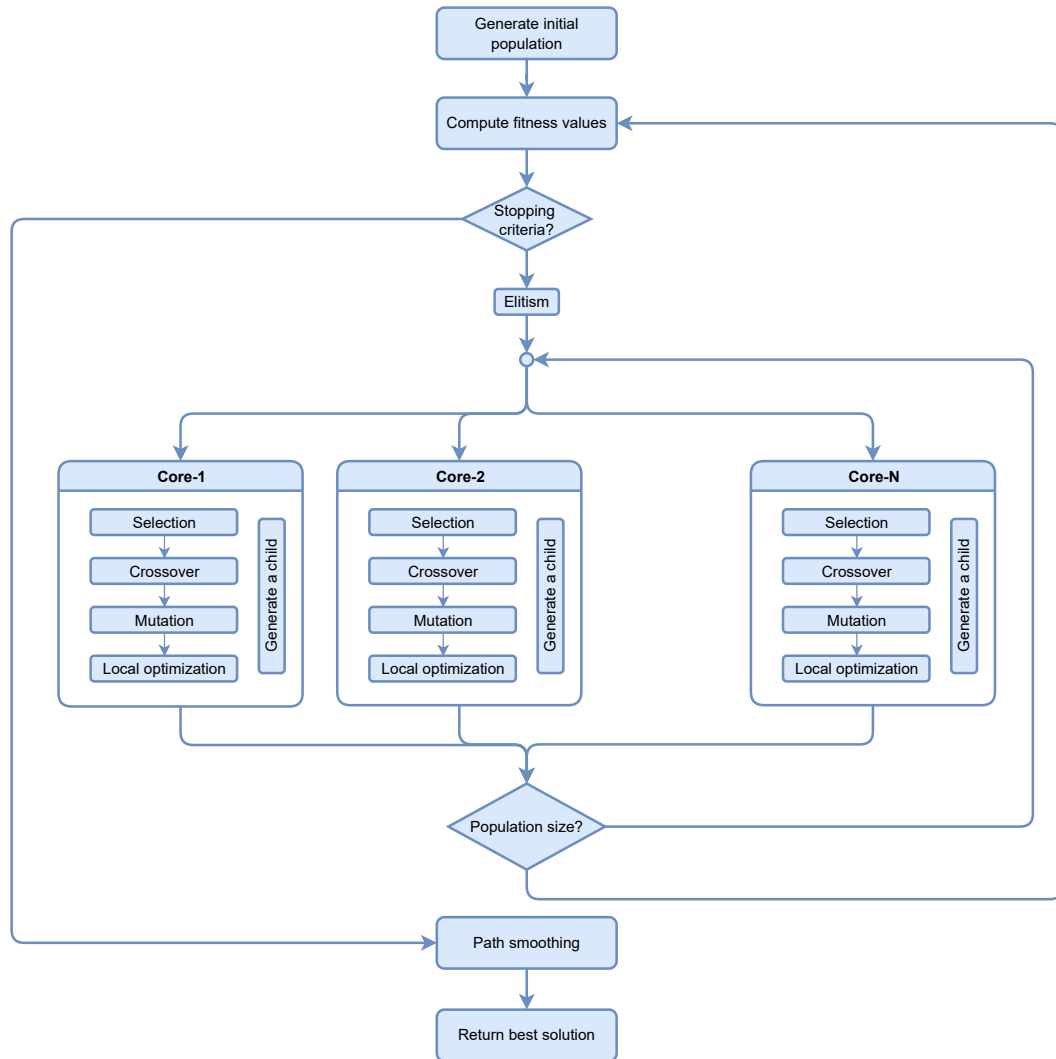
FIGURE 4.3: Parallel structure of the genetic algorithm as proposed
by Reference [29]

optimisation is used to improve the proposed solution for each core. It is not guaranteed that the optimiser would suggest routes that pass through all sensors. If it is not compliant, the process is restarted. The proposed modification concerning the original EA can be seen in Figure 4.3. Another novelty found in this reference is to propose upper and lower limits to the number of core processor optimisation.

Reference [30] introduces the idea that path optimisation with EA has one intrinsic limitation. This limitation considers the complete path as an individual, each waypoint a gene in it. According to the authors, optimisations might not find a solution when presented with a scenario with multiple obstacles. The authors propose that objectives and constraints are evaluated at the waypoint level. Hence, every single waypoint can be evaluated and evolved. The EA suggests a new path, and each waypoint that composes it is evolved. After each waypoint is considered, the resulting path is evaluated against its parent. Naturally, the computational effort is increased due to the recursive optimisation of each waypoint. Results and comparisons with other EA show that the proposed new EA performs better with a high number of obstacles. A secondary conclusion can be drawn from the algorithm comparisons performed by the authors. Polar coordinates produce better results than
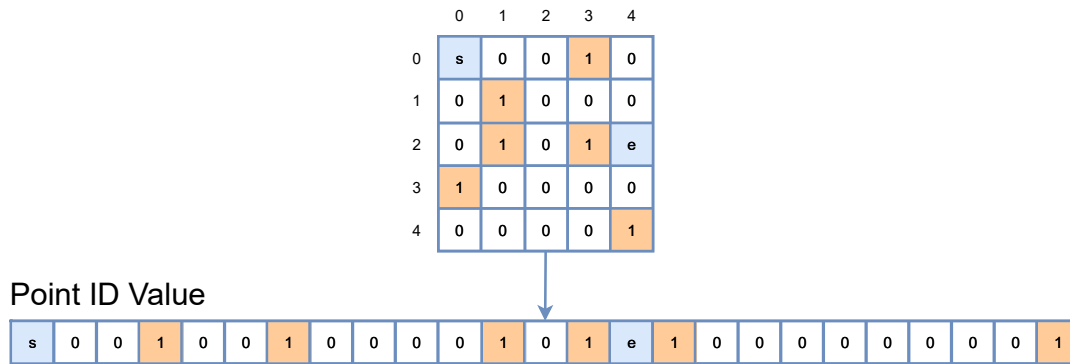
FIGURE 4.4: Binary chromosome encoding proposed by [31]. Feasible points are represented with value 0, non-feasible points with 1. S and e stand for starting and ending point, respectively.

those algorithms using cartesian coordinates.

A new type of waypoint encoding for EA is proposed in Reference [31]. The authors state that the traditional encoding results in a loss of diversity in the solutions. A binary chromosome encoding is proposed to reduce diversity losses. In this encoding, values are predefined concerning the availability of such a waypoint. Non-available waypoints are represented with a one, as in Figure 4.4, whereas available ones are encoded with a 0. Start and end positions need to be encoded too. The final solution will consist of a succession of points used as control points in a Bezier curve. Authors name this algorithm as Modified Genetic Algorithm or MGA.

4.2.3 Particle swarm optimisation (PSO)

The following algorithm worthy of its subsection is the PSO. Reference [32] is the earliest reference applying PSO to find the optimal route for a UAS. The novelty of this work is the combination of MPC together with PSO. The latter is applied to obtain a viable path through the scenario before the flight. MPC is used throughout the flight and avoids any appearing threat. Although the fundamentals of MPC and PSO have not been modified, authors claim that the combination results in increased flexibility and reduced complexity.

Reference [33] has been selected because of the inclusion of the air to ground (A2G) channel as an objective. The channel model is not as sophisticated as the one presented in Chapter 3. Nevertheless, the authors include a simple simulation of a directional antenna and line of sight probability. With these parameters, the authors apply the PSO algorithm. This algorithm will provide the best positions of a UAS swarm that maximises coverage probability. The same application has been studied in Reference [24] using MPC.

A similar application can be found in Reference [34]. In this case, it is not the signal power subject of optimisation but the ssignal interference plus noise ratio (SINR). This parameter is appropriate as the swarm of UAS is required to minimise interference to the users on the ground. The interference minimisation can do this by modifying the position of each UAS together with the radiation power. Simulations are repeated for many users, and the SINR is provided to such users. An initial step is introduced before PSO to raise the chances of acceptable output. A K-means clustering-based initialisation algorithm analyses the network and proposes the first PSO positions. Naturally, this requires prior knowledge of the users' locations. As

in previous PSO references, the advantage of PSO over EA is never cited by the authors.

In contrast with previously mentioned references, the authors in [35] compare previous EA implementations. The previous implementations considered include the modified EA proposed in Reference [31]. According to the authors, PSO algorithms maintain simplicity whilst accessing a global optimum for all particles. The combination of simplicity and effectiveness makes PSO an excellent candidate for UAS path planning optimisation. Authors modify the initial step by a chaos-based initialisation. This type of initialisation is a powerful tool to diversify the swarm and improve the performance of PSO. Results show the goodness of PSO over modified GA and the promised performance increase of the chaos-based initialisation.

Previous work in Reference [35] is extended in [36]. Authors start from the already modified PSO and introduce simulations with constraints such as performance, minimal flying length, mountain terrain and radar threat cost. The chosen method to manage constraints is to add a penalty to each solution that violates a constraint. Since the kinematic model of the UAS is considered, each particle represents a set of control variables, and the main PSO requires an update.

Authors in [37] correlate a poor global search with insufficient population diversity at the initial stage. To provide a higher population diversity, the authors borrow the sparrow search algorithm for the structure direction of the population. This search algorithm improves the learning factor formula, which enhances the ability of global search and local development. Upon testing in a mountainous environment with NFZs, the modified PSO achieves higher accuracy and better final values. For simplification, splines are used to interpolate the position of the UAS between waypoints.

Last but not least, Reference [38] is the most recent one when writing these lines. Although the authors combine multiple optimisation strategies such as PSO, Dijkstra and divide-and-conquer, they are implemented in the same loop, so the proposed method cannot be considered part of hybrid optimisation. Through several simulations, the authors show that Dijkstra can quickly find a solution from PSO particles. Divide-and-conquer strategies attempt to group several waypoints into one. Each group of waypoints need a subcost function to assess the quality of the subpath. The solution used for iteration is composed of the best group of all the groups. Combining the three algorithms results in a method that promises to increase performance speed, robustness, and premature avoidance.

4.2.4 Hybridisation

Up to now, several combinations of optimisation algorithms have been presented. Although Reference [38] serves as a good example, it is not included in this subsection. The main reason is that the secondary optimisation algorithm (Dijkstra) is executed inside the workflow of PSO.

This work considers a hybridisation when the main algorithm is executed up to a conservative threshold. Usually, this first optimisation algorithm is one of the deemed intelligent algorithms: EA or PSO. Once the intelligent algorithm has finished, a non-intelligent one is applied using the intelligent solution as a starting point.

Authors in Reference [39] propose a hybrid metaheuristic method to plan 4D trajectories. The proposal is to use simulated annealing together with hill-climbing local search. Simulated annealing is considered the algorithm with lower probabilities of falling into local minima. On the other side, Hill-climbing will focus on the

proposed solution by simulated annealing. In theory, the less exploratory algorithm should obtain a better solution when pointed in the right direction.

The problem of travelling through specific locations has already been studied in Reference [29]; Reference [40] proposes a new optimisation approach for a similar problem. In this further reference, a swarm of UAS is given the task of navigating a complex environment with many occlusions that difficult the sensor reading. Because of the occlusions, MILP is first used to assign sensing locations to each UAS. In the second stage, EA is used to find the optimal path for each UAS. The result of such hybridisation achieves paths with lower energy requirements.

A combination of PSO and simulated annealing is proposed in Reference [41]. The optimisation consists of a UAS whose mission is to fly from A to B in the shortest time whilst avoiding NFZ. With this approach, the number of iterations is reduced from sixty to thirty combined iterations. Unfortunately, the authors do not provide a forecast of the future uses of their research.

Reference [42] proposes to solve a similar optimisation problem to the one used in Reference [34]. In this case, the problem has a higher complexity because the UAS swarm is heterogeneous. The transmitting capabilities of each UAS are not equal and should be taken into consideration. The metrics to be optimised are the number of ground users covered and the mean channel quality. Authors propose to combine PSO and EA, a novel combination for hybridisation, as it combines two intelligent algorithms. Within the conclusions, it is acknowledged that finding a scenario with heterogeneous user demands is not straightforward.

4.3 Constrained optimisation

The general form of an optimisation problem is:

$$\min F[x] \tag{4.3}$$

Where $x \in R^n$ is a design variable, $f[x]$ an objective function, $X \subset R^n$ a constraint set of the feasible region. Depending on X, the optimisation can be considered constrained or unconstrained. If the constraint set $X = R^n$, the problem is unconstrained:

$$\min_{x \in R^n} F[x] \tag{4.4}$$

The unconstrained optimisation searches for an optimum vector x that produces the minimum value of the function $F[x]$. Once such a vector is found, it can be expressed x^*. According to Reference [43], the values of $F[x]$ may be obtained through different methods. The first method, and more straightforward, is when an analytical formula is available. In this case, the x values are placed into the formula, and $F[x]$ is instantly known. The following case has higher complexity: the outcome involves several calculation steps and cannot be expressed as a simple formula. Last but not least, the outcome of using x can be obtained from measurements.

Judging by the complexity of $F[x]$ in Figure 4.5, the problem to be optimised belongs to neither option one nor three. The best approach is to consider the cost value due to several calculation steps. As seen in Figure 4.5, the design variables x are the azimuth (Az), elevation (Elev) and Range (R) values. Design variables are translated into waypoints to be flown by the UAS. Together with the UAS dynamic model, the array $\mathcal{L}^{WP}$ is transformed into $\mathcal{L}^L$. This step is not as simple as described; the flight simulator plans all necessary actions for the UAS to fly through each waypoint. Upon calculation, $\mathcal{L}^L$ is obtained with the desired sampling rate. $\mathcal{L}^L$ is then
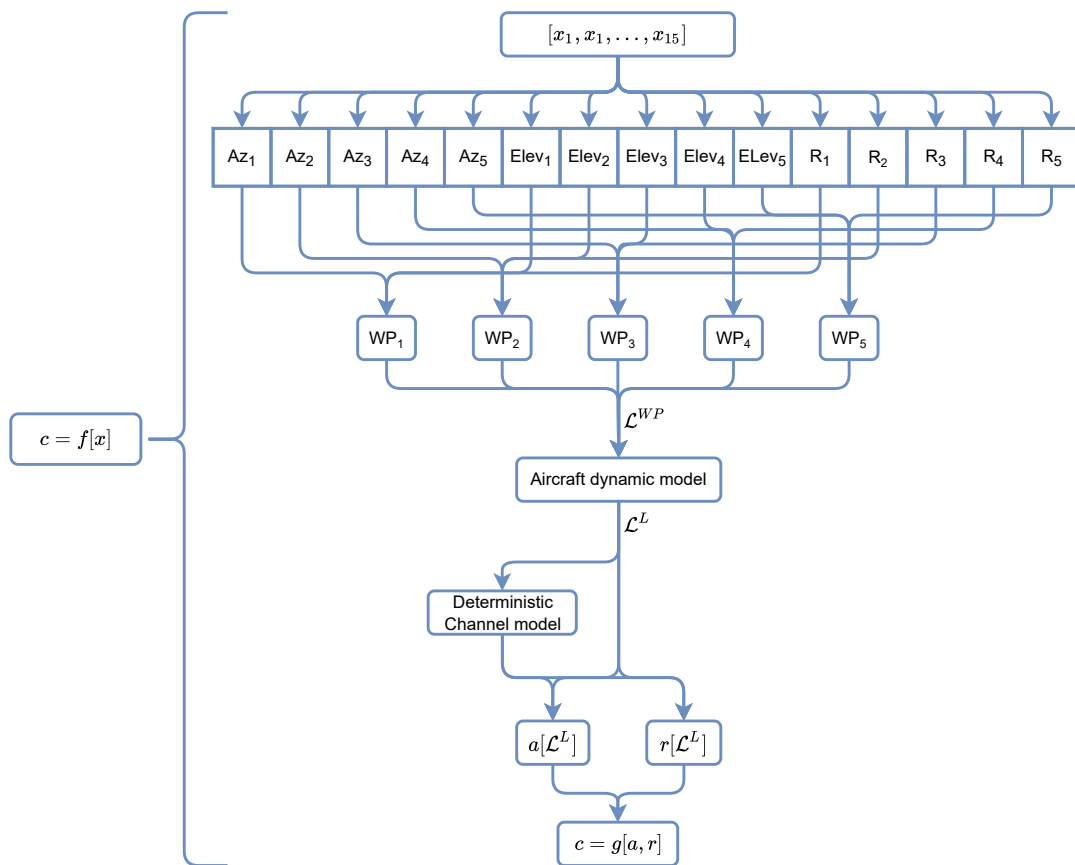
FIGURE 4.5: Steps taken by the optimiser to assign a cost value to the
proposed vector x.

used by the deterministic channel model to determine the received power level and later on by restrictions r and objectives a. The result is the cost value, used by the optimiser to determine the fitness of x and either propose new or improve the design values.

The objectives and restrictions to be used are independent of the selected optimisation method. Objectives define those parameters or attributes of the simulation that need to be improved somehow. Those objectives, ideally, should be contained between zero (best case) and 1 (worst case). If the attribute to be optimised is designed with this mindset, the optimisation algorithm focus on achieving the minimum value, zero.

The first objective to be minimised is the trajectory distance. It is not physically possible to achieve a null distance, as the UAS is required to travel from WP_{start} to WP_{end}. Since the objective is not to bend reality and achieve null distance travel, the alternative is to approximate the travelling distance to the distance between WP_{start} and WP_{end}. Equation 4.5 models this objective. The delta between the actual travelled distance ($\sum_{i=1}^{M-1} |L_{i+1} - L_i|$) and the ideal distance (d_{ideal}, the distance of a straight line that connects WP_{start} end WP_{end}) is positioned in the numerator. This delta can easily increase, and it is cumbersome to establish an upper limit. The choice is to model the maximum distance as the distance of $\mathcal{L}_{max}^{WP}$ as defined in Equation 2.16. The resulting travelled distance is then used as d_{ul}. Since the values are also passed to the optimiser, it is guaranteed that no path will exceed d_{ul}. The first objective, a_1, is bounded between zero and one with this approach.

$$a_1 = \frac{\sum_{i=1}^{M-1} |L_{i+1} - L_i| - d_{ideal}}{d_{ul} - d_{ideal}} \tag{4.5}$$

The second objective that is passed to the optimiser is the vertical variation. Ideally, the UAS flies in a trajectory constant in height. A null height variation contributes to higher energy efficiency. It also contributes to the comfort of the passengers, if any. Although the optimiser will propose trajectories with height variation to avoid some obstacles, trajectories with flatter profiles are preferred. Equation 4.6 mathematically models this objective. The delta between the current height variation ($\sum_{i=1}^{M-1} |z_{i+1} - z_i|$) and the ideal height variation (Δh_{ideal}) is located at the numerator. The process to obtain Δh_{ideal} is similar to the process to obtain D_{ideal}. The denominator of Equation 4.6 is used to normalise a_2. Delta Δh_{ul} is the height variation performed by a UAS following $\mathcal{L}_{max}^{WP}$. As height variation is optimised, the value of a_2 is minimised.

$$a_2 = \frac{\sum_{i=1}^{M-1} |z_{i+1} - z_i| - \Delta h_{ideal}}{\Delta h_{ul} - \Delta h_{ideal}} \tag{4.6}$$

Previous objectives have been focused on a fast and smooth trajectory. Whilst this is relevant in terms of energy consumption, they do not target the main focus of this work: optimum A2G communications. A dedicated objective is designed to maintain the average received power close to the defined maximum power. E_u^{PWR} and E_l^{PWR} in Equation 4.7 are the respective upper and lower received powers. These values are obtained from the already described shortest path connecting WP_{start} with WP_{end}. Matching the behaviour of a_1 and a_2, a_3 is bounded between 0 and 1. Higher mean received power values are rewarded with a lower objective value.

$$a_3 = \frac{E_u^{PWR} - \bar{E}}{E_u^{PWR} - E_l^{PWR}} \tag{4.7}$$
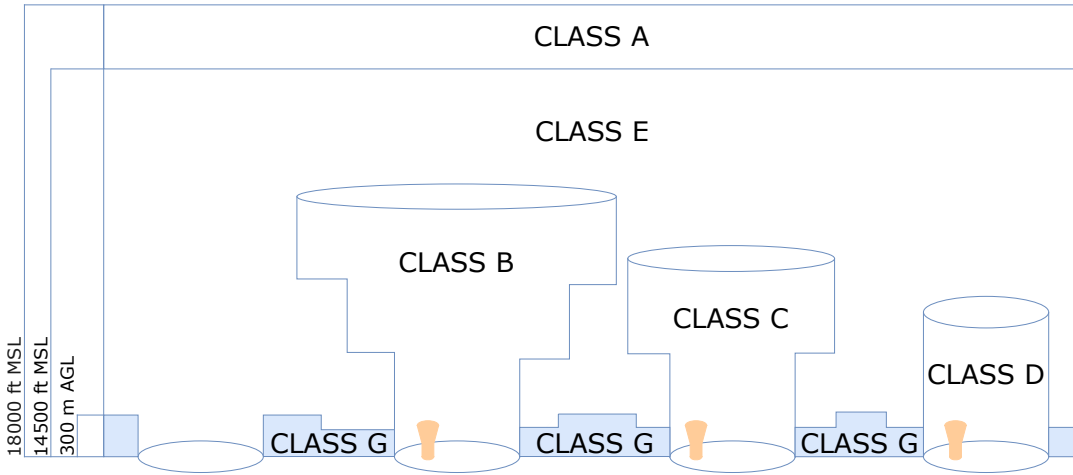
FIGURE 4.6: Airspace classification as defined by ICAO. Class G is
the only airspace without air traffic control.

Objectives are functions or mathematical expressions that guide the optimiser towards global minima. They can adopt any value and are generally designed to be bounded between zero and one. Restrictions model those cases where a given solution, although mathematically feasible, it renders such solution unusable. Reference [44] gives the example of the process to maximise a box; without constraints on the dimension, the box volume has no limit.

The modelling of restrictions differs from the modelling of objectives. The focus of the restrictions lies in avoiding any unfeasible solutions. For a flying UAS, an unfeasible solution is one leading to any L_i with negative z value. A higher limit is also established under the assumption that the UAS will fly through controlled airspace. Figure 4.6 has been included to contextualise where the UAS is allowed to fly and why an upper height limit is necessary. Class G is the only class that is not controlled and the one free from commercial flights. Classes F to A are under the supervision of air traffic control. Considering the maximum height of class G is 300 meters above ground level (AGL) around airports, it is safe to establish an upper limit of height in the 200 meter AGL. The mathematical expression of r_1 can be found in Equation 4.8. The nominator contains the number of points in the path that violates either minimum or maximum height limit. This number of points is normalised with the total number of points M in $\mathcal{L}^L$.

$$r_1 = \frac{\text{Number of points } L_i \text{ outside height limit}}{M} \tag{4.8}$$

Keeping $\mathcal{L}$ within horizontally appropriate limits is carried by r_2. This restriction ensures that each point of $\mathcal{L}$ is inside map limits and outside NFZs. Such flight prohibited areas are defined ad-hoc for each scenario. NFZs are areas that cannot be flown over and thus must be avoided. Examples of protected areas with an NFZ are governmental buildings, areas used for military exercises, or even events such as the 2012 London Olympic Games. The number of points violating map limits or an NFZ is counted and normalised with the number of points M in $\mathcal{L}^L$.

$$r_2 = \frac{\text{Number of points } L_i \text{ inside NFZ and outside MAP limits}}{M} \tag{4.9}$$

Like the third objective, the third restriction deals with the received power. Whereas the corresponding objective rises the average received power, r_3 reduces the number

of points under the system sensitivity. System sensitivity is the value limiting the capacity of the system to separate the received signal from the background noise. Whenever the received power drops below this value, it is not guaranteed that the A2G link will be maintained. Equation 4.10 normalises the number of points with a received power below system sensitivity. The third restriction is then bounded between zero and one.

$$r_3 = \frac{\text{Number of points } L_i \text{ with received power below system sensitivity}}{M} \tag{4.10}$$

Transforming values of objectives and restrictions to cost or developing methods to deal with restrictions is the subject of study, as shown in Section 4.1.3. Previous references proposed several approaches, such as providing a penalty to each map cell ([11]), solving restrictions and later objectives ([12]), penalty functions ([15]), feasibility rules ([16]), or the consideration of dynamic constrains ([17]). Penalty and barrier methods are selected because of their simplicity and effectiveness. As proposed in Reference [45], the general idea is to translate hard constraints into barriers, so the problem subject to optimisation remains unconstrained.

Penalties are introduced to steer the optimisation process towards the feasible region. Over time, the optimisation evolves towards avoiding such constraint or the convergence to it. Exact penalty methods are not widely used because of the issues when dealing with non-differentiable functions. Adding a convex penalty can regularise a problem and reduce noise effects and local minima. Another use is to steer the solutions into productive solutions. Regardless of the formulation, a fundamental assumption is that the non-feasibility of a candidate solution can be measured and quantified.

4.3.1 Penalty method

Penalty methods modify the cost function directly. Because the problem to be optimised is highly complex, the focus lies now on the Function $g_p[a, r]$ displayed in Figure 4.5. This function aims to gather the resulting values of objectives and restrictions and produce a single value, the final costs associated with the candidate solution. The main characteristic of the penalty method is that the penalty should be null when all restrictions are satisfied.

Conceptually, a penalty function is straightforward. Because of its simplicity, a wide range of applications is guaranteed. Such functions are suited for disjoint feasible regions or when the constraint boundary contains the optimum. Despite its simplicity, effort should be invested in balancing the exploratory phase through the infeasible region and not being trapped in such a region. Reference [46] elaborates on the necessary balance and its coupling with the penalty factor. This factor should be sufficient so that infeasible regions do not dominate over feasible solutions. At the same time, it should be small enough not to discard the exploration of the unfeasible region.

The mathematical expression of the proposed penalty function can be seen in Equation 4.11. Values from zero to one have been given to objectives and restrictions to achieve Figure 4.7. Equation 4.11 meets the necessary conditions to satisfy a penalty method; if restrictions are violated, a penalty is applied to $g_p[a, r]$. The optimiser will perceive the proposed solution as a non-suitable candidate and propose a different one.
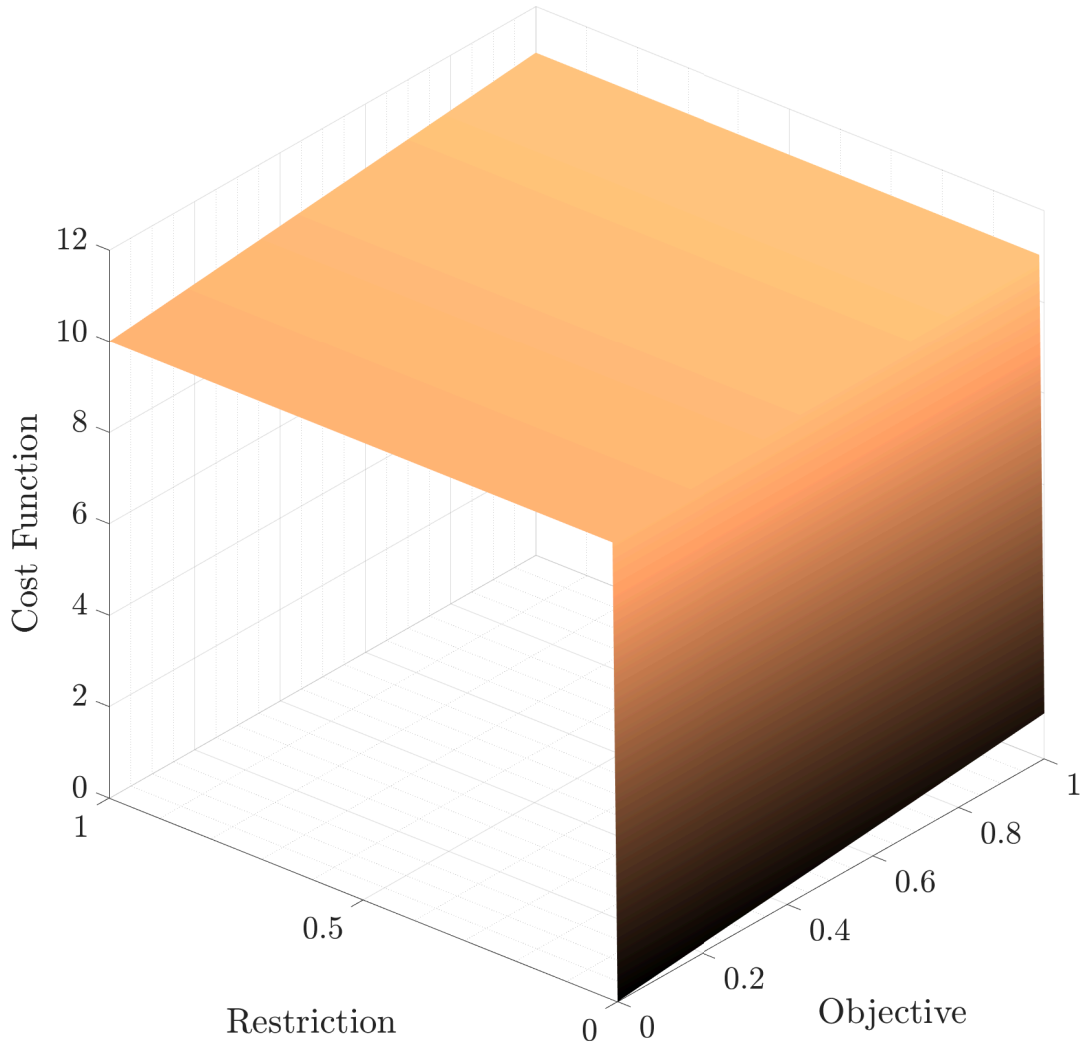
FIGURE 4.7: Values obtained by applying the penalty cost function.
Only one objective and restriction have been used for this figure.

$$c = g_p\,[a,r] = \begin{cases} \dfrac{1}{3}\sum\limits_{i=1}^{3} a_i, & \text{if } \sum\limits_{j=1}^{3} r_j = 0 \\[2ex] \dfrac{1}{3}\sum\limits_{i=1}^{3} a_i + \dfrac{1}{3}\sum\limits_{j=1}^{3} r_j + 10, & \text{if } \sum\limits_{j=1}^{3} r_j > 0 \end{cases} \qquad (4.11)$$

Studying Figure 4.7, an aspect of the cost function stands out. A plateau of un-feasible solutions with a reduced slope towards the feasible region exists. The risk of an optimiser being trapped in the unfeasible solution exists. Although stalling is possible, a solution should exist that avoids the restrictions. Once this is found, the optimiser will prioritise these solutions and move around the edge where the feasible solutions lie.

It might seem contradictory that this method is included presenting such a stable plateau of infeasible solutions. Until now, only viable and accepted methods have been presented for conciseness. Discarded maps, inefficient methods to compute ground reflections, or flight simulation approaches have not been included. In this case, an exception is made because optimisation is the main focus of this work and because the penalty method was a significant part of the Reference [47].
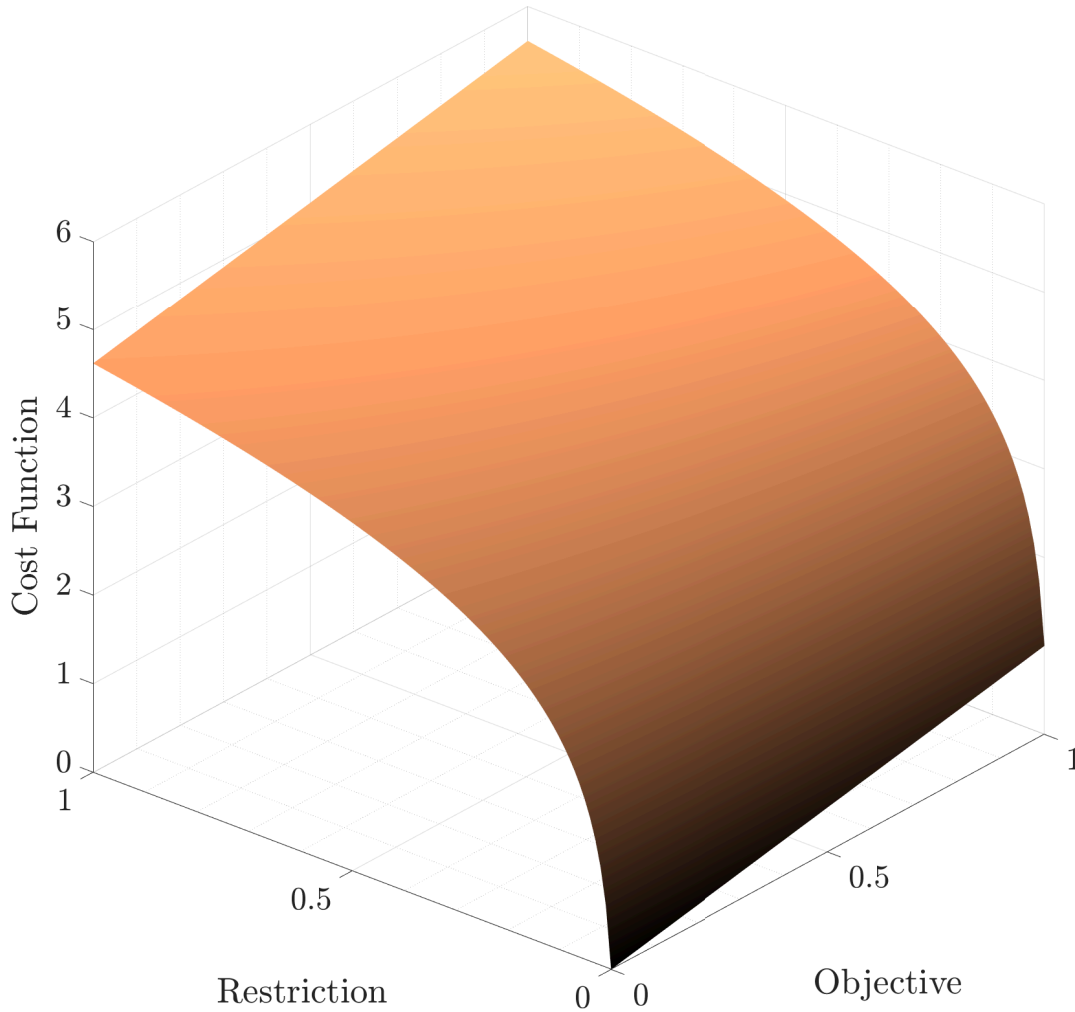
FIGURE 4.8: Values obtained by applying the adaptive barrier cost function. Only one objective and restriction have been used for this figure.

4.3.2 Adaptive barrier method

The adaptive barrier method, proposed by authors in [45], can be seen as an alternative to the previous penalty function. The main similarity with the penalty method is converting constrained optimisation to an unconstrained one. Regarding expected convergence and performance, this method is expected to have a higher performance and convergence rate.

$$c = g_a\left[a, r\right] = \frac{1}{3}\sum_{i=1}^{3} a_i + \log\left[\frac{100}{3}\sum_{j=1}^{3} |r_j| + 1\right] \quad (4.12)$$

Equation 4.12 gathers the mathematical expression required for the customised adaptive barrier method. As opposite to Equation 4.11, there is no change in the function based on the value of the restrictions. Objectives and restrictions are integrated into the same function, achieving a continuous function with no abrupt changes. The result of plotting Equation 4.11 can be seen in Figure 4.8. There is no reason for the optimisation algorithm to stall and not advance towards a better solution with this function. Furthermore, the closer the proposed solution reduces the
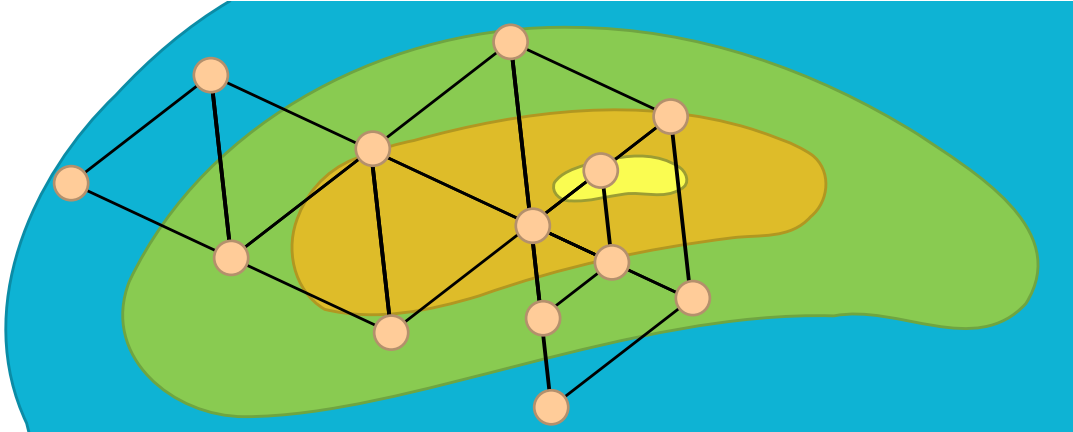
FIGURE 4.9: Steps taken by the simplex algorithm to minimise a given
cost function.

value of restrictions, the higher the slope.

4.4   Candidate optimisation algorithms

The following enumeration of optimisation approaches covers those that produced
results worth analysis. The most suitable combination of algorithms and cost func-
tion has been found through trial, error, and intensive result study. The most suitable
optimisation strategy is going to be presented in Chapter 5. Before dealing with the
results, the optimisation algorithms are explained.

The same procedure is followed with the state of the art study. Algorithms are
divided between non-intelligent and intelligent algorithms. It is worth mentioning
that there are mainly two main characteristics shared among all used algorithms.
The first characteristic or requirement is that the optimisation algorithm is available
in Matlab either in [48] or the global optimisation toolbox [49]. The second aspect
is that algorithms are not going to be modified. One of the objectives is to find the
most suitable optimisation strategy and not modify existing algorithms. It assumed
that the performance and reliability of the Matlab implementation are good enough
to provide answers to the research question.

4.4.1   Non-intelligent algorithms

Simplex

The simplex is based on the algorithm developed by Nelder and Mead [50]. The
simplex method does not share characteristics with the line search strategy or the
trust region method. The popularity of the simplex method lies in its derivative-
free formulation. This optimisation approach is one of the most used for nonlinear
optimisation. In Matlab, it can be used by applying the function "fminsearch".

How the simplex searches for a minimum can be seen in Figure 4.9. The simplex
is an N-dimensional figure with $N + 1$ vertices, which means that it can be applied
to any given search space independently of the number of dimensions. The simplex
takes the form of a triangle for a $2D$ search space, as in Figure 4.9. In a $3D$ search
space, the simplex takes the form of a tetrahedron. The first step is to define the
initial search point, and the first simplex is generated. The cost function is applied
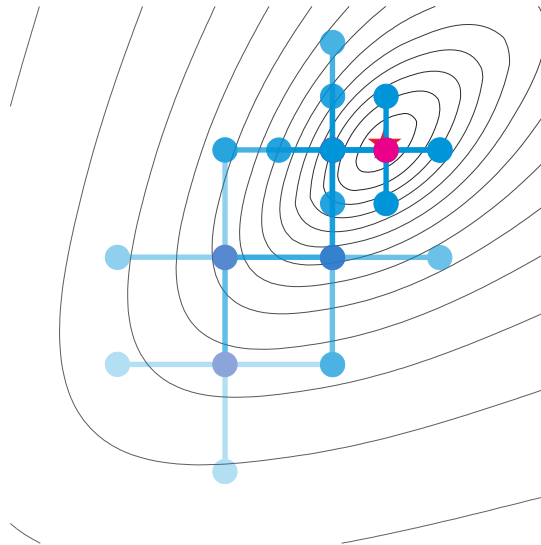
FIGURE 4.10: Steps taken by the generalised pattern search algorithm
to find the minimum of a given cost function

to assign a value to each vertex of the simplex. The worst vertex is either removed
or updated in each iteration.

There are several movements that the simplex figure can make: reflection, expansion, and different forms of contractions. All the points except for the worst one are used to find an intermediate point when a reflection movement is performed. This point will be used to generate the reflection axis. If the worst point is in an area where error increases, the opposite direction might be an error decreasing. As the name indicates, the reflected point is moved twice the distance. When this modification is not successful, contraction is the following modification. Contraction can be internal or external. External when the point is moved between reflection and expansion. Internal when the point is moved between the intermediate and the original point. If nothing works, multidimensional contraction is the last option. In this step, the simplex is reduced by moving the rest of the points closer to the best vertex.

The simplex method is not an ambitious algorithm like other non-intelligent algorithms. Nonetheless, this makes the simplex a robust algorithm. Advances are not made based on the best vertex, and somewhat the worst vertex is continuously improved. Thanks to this behaviour, local minima can be avoided. The expansion of the simplex took place when reflection was successful.

Generalised Pattern search

Generalized pattern search (GPS) shares the main attribute with the simplex method described above. This attribute is independent of the gradient. As a result, this optimisation algorithm can be applied to functions that are not continuous or differentiable.

Opposite to the simplex method, this method does not use a geometrical figure. Instead, a mesh of points is defined around the starting point by adding unitary values in each axis direction [51]. This mesh can be seen in Figure 4.10 around the initial point in the left south portion. As the algorithm evolves, the centre point is translated to the most successful points of the mesh. The mesh shrinks around the centre point if no improvement is found. The contraction of the mesh is also visible

(A) Generalised pattern search positive basis 2N

(B) Generalised pattern search positive basis N plus 1
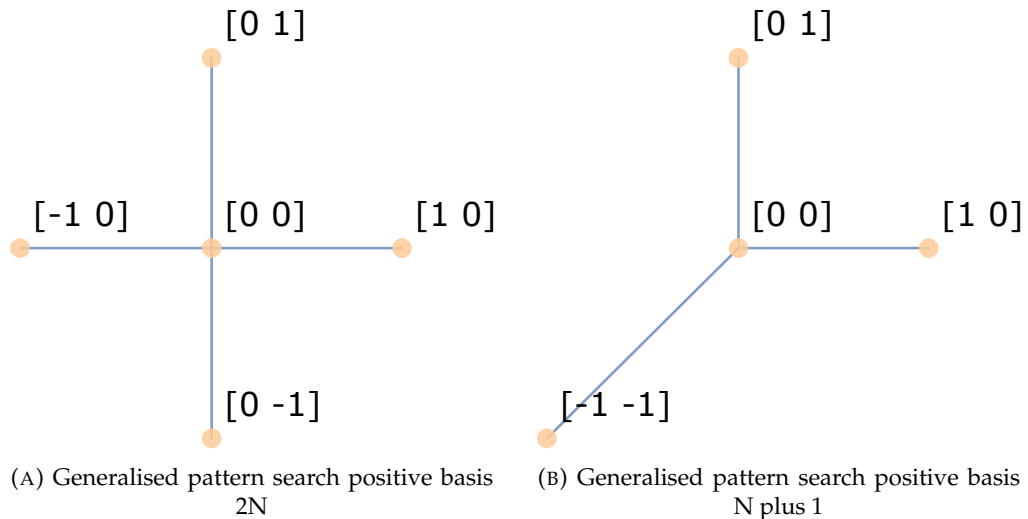
FIGURE 4.11: Generalised pattern search polling options

in Figure 4.10 towards the minimum of the cost function. The algorithm will stop when the mesh size is smaller than a defined tolerance.

The various polling methods available in Matlab are represented in Figure 4.11. Subfigure 4.11a details the case where the initial point is expanded by twice the number of independent variables. As the figure depicts a two-dimensional space, the mesh comprises four points. Another option is to have the number of independent variables plus one. Subfigure 4.11b shows the example of a two-dimensional problem being expanded with a mesh of three points. In a function with many local minima, using as many points as possible is advisable as in the 2*N* approach.

Although not implemented by Matlab, a more sophisticated approach is direct search with coordinate rotation. In this approach, the unitary vectors form an *N*-dimensional base. The initial set of coordinate directions eventually forms a maximal positive basis. The progress of its iterative points depends on obtaining an improving point at some vertex of the above mentioned n-dimensional structure. In this flavour of direct search, a new movement is added to the pool; rotation of the base.

Annealing

This process and its characteristics mimic the same process in the metallurgy and materials environment. This term is used for the heat treatment that alters physical and sometimes chemical properties in such context. Usually, this process is applied to increase ductility and reduce hardness. The final aim of the annealing process is to migrate atoms to the crystal lattice and decrease the number of dislocations. Crystallisation occurs as the material reduces its temperature. In this case, it is essential to carefully apply the heating and cooling rates.

The principle of cooling a structure is applied to optimisation problems. In Reference [49], simulated annealing is composed of five steps. First, a random trial point is chosen whose distance from the current point is determined by a probability function. Such probability function has a scale depending on the current temperature. The user has three options: fast, Boltz and custom. The step equals the current temperature in a random direction in the fast setting. Boltz uses a step equal to the square root of the current temperature, and the direction is still random. The user could also design a custom function to choose the distance and direction from the

current point. As previously highlighted, the objective is to find a suitable algorithm or combination of algorithms taking already implemented options. The default in [49] is the fast option, and this is the one chosen when annealing is applied.

Once the trial point is defined, it is decided whether the trial point is better than the actual one in the second step. Even if the proposed point provides no improvement, it can still be chosen under certain conditions. In [49], Equation 4.13 determines the probability of acceptance of the worst point. In Equation 4.13, Δ is the difference between the cost of both points, T_0 is the initial temperature, and T is the current temperature. Smaller temperatures lead to a more negligible acceptance probability. The same phenomena occur with a more significant Δ; the bigger the difference, the smaller the probability. As with the distance of the trial point, the user can define a custom function. In this case, the approach uses the default function proposed by [49].

$$P = \frac{1}{1 + \exp\left[\dfrac{\Delta}{\max[T]}\right]} \tag{4.13}$$

Similar to the metallurgy process, the temperature is lowered systematically. Such temperature behaviour can be customised. Equation 4.14 displays the three possibilities to model this behaviour. As with previous steps, the chosen one is the default. The annealing parameter k is equal to the iteration number until reannealing occurs.

$$\begin{aligned} \text{Exponential (default)} \quad & T = T_0 \cdot 0.95^k \\ \text{Fast} \quad & T = T_0/k \\ \text{Boltz} \quad & T = T_0/\log[k] \end{aligned} \tag{4.14}$$

The fourth step, known as reannealing, does not occur at each iteration. As proposed in Reference [52], the annealing parameter is redefined every hundred iterations following Equation 4.15. The equation is composed of the present temperature T_0, the initial temperature T_i and the gradient of the objective function s_i. This parameter lowers the value of the iteration number. Ultimately, this parameter raises the temperature in each direction.

$$k_i = \log\left[\frac{T_0}{T_i}\frac{\max[s_j]}{s_i}\right] \tag{4.15}$$

As with previous algorithms, the final step is to check compliance with any existing stopping criterion. The various stopping criteria are function tolerance, maximum number of iterations, maximum function evaluations, maximum time and objective limit. Usually, function evaluations are set to a multiple of the number of variables. Function evaluation results in a more restrictive criterion than the number of iterations.

Quasi-Newton

Quasi-newton methods are used to either find zeroes or local maxima of functions. These methods are applied if the Hessian or Jacobian is unavailable or is too expensive to compute at every iteration. The Jacobian or Hessian should be available to have the complete version of the newton method. The Jacobian or Hessian means that the cost function can be described as an analytical formula. Section 4.3 gives

enough arguments to consider either the Jacobian or the Hessian as non-available. Because of this condition, quasi-newton methods could be a suitable candidate to optimise a proposed route.

As in Newton's method, a second-order approximation can be applied to find a solution of a Function $f[x]$. Such second-second order approximation can be found in Equation 4.16, $H[x]$ represents the approximated Hessian evaluated in the x point. Once the derivatives are computed, the optimal direction is computed using the extra information provided by the curvature.

$$f[x + \Delta x] \approx f[x] + \nabla f[x]^T \Delta x + \frac{1}{2}\Delta x^T H[x]\Delta x \qquad (4.16)$$

The minimum in a one-dimensional function can be found when the slope is zero and the derivatives positive. With the increase in dimensions, the gradient provides information to find the minimum. The minimum can be found where the gradient is null in such a case. The optimal solution is found when the partial derivatives concerning x are zero, and the Hessian is positive:

$$\nabla f[x + \Delta x] = \nabla f[x] + H[x]\Delta x = 0 \qquad (4.17)$$

$$x^T H[x + \Delta x]x > 0, \forall x \qquad (4.18)$$

if the gradient is null, the next Newton step can be formulated as:

$$\Delta x = -H[x]^{-1}\nabla f[x] \qquad (4.19)$$

so that the new point in the process can be obtained following Equation 4.20. In said equation, α is known as the learning rate, which controls the size of the step at each iteration. If this value is oversized, the model will diverge. If the step size is undersized, the search becomes inefficient. In a linear search process, the optimal value for α is also subject of optimisation.

$$x_{k+1} = x_k + \alpha \left(-H[x_k]^{-1}\nabla f[x_k]\right) \qquad (4.20)$$

In the Newton methods, H is known as it can be computed from an analytical formula. In Quasi-Newton methods, H is approximated with the evolution of $f[x]$ and the gradient. The identity matrix approximates H in the first iteration. After the first iteration and subsequent iterations, H is estimated recursively based on past knowledge.

One of the most popular techniques to approximate the Hessian is the Broyden-Fletcher-Goldfarb-Fhanno (BFGS) [53]. The approximation to the inverse of the Hessian used in BFGS can be found in Equation 4.23. Computing the inverse of the Hessian can also be the subject of study. The work of Davidon-Fletcher-Powell (FDP), summarised in Equation 4.24, can be used to perform the inverse of the Hessian. BFGS might perform better than DFP in some situations. Although both methods might look like they cannot be combined, Equation 4.25 shows how to apply both methods.

$$q_k = \nabla f[x_{k+1}] - \nabla f[x_k] \qquad (4.21)$$

$$s_k = x_{k+1} - x_k \qquad (4.22)$$

$$H_{k+1} = H_K + \frac{q_k q_k^T}{q_k^T s_k} - \frac{H_k s_k s_k^T H_k}{s_k^T H_k s_k} \qquad (4.23)$$

$$H_{k+1}^{-1} = H_k^{-1} + \frac{s_k s_k^T}{s_k^T q_k} - \frac{{H_k^{-1}}^T q_k^T q_k H_k^{-1}}{q_k^T H_k^{-1} q_k} \tag{4.24}$$

$$H_\phi^{-1} = (1 - \phi) H_{DFP}^{-1} + \phi H_{BFGS}^{-1} \tag{4.25}$$

Surrogate

A surrogate model is a resource applied when the outcome of the process of interest cannot be easily measured or computed. For some processes, obtaining a value out of the cost function can take minutes, hours, or even days to complete. The optimisation of such a process might take a long time, especially for iterative optimisation algorithms that require a high number of samples.

Instead of trying to model the behaviour of the cost function, some costs are obtained. These costs are used to generate a model that approximates the behaviour of the cost function. This simplified model is known as a surrogate model. The aim is not to construct an exact cost function model but to have a computationally reasonable approximation. The algorithm balances the exploration to search for the global minimum with speed to obtain a good solution.

Surrogate optimisation alternates between two phases: construct a surrogate and search for a minimum. The first step creates random points within the bounds. Each point is evaluated to construct a surrogate of the objective function through the tested points. The second step defines several points within the bounds, and the best point is chosen. Such a point is called an adaptive point. This point is then used to update the surrogate. Reference [54] provides a detailed explanation of the steps taken by the surrogate optimisation.

4.4.2 Evolutionary Algorithm

Recent research in evolutionary computation has proved that the inspiration taken from natural evolution can yield robust algorithms [55]. The EA will find a local unconstrained minimum for the cost function based on natural selection, the process that drives biological evolution. This algorithm will modify a population or propose new individuals or waypoints for the UAS to follow. The algorithm finds the optimal position by producing descendants over the best ascendants and introducing random mutations to these descendants. The solution will be found when the population has evolved over successive generations towards a solution.

The algorithm evolves the population following six steps. The first, shared by all the optimisation algorithms, is to propose a random initial point. For EA, the first step is not a random point but a random initial population. The size of the population can significantly influence the behaviour of optimisation. On the one side, an oversized population introduces inertia, evolution is not agile, and the search space is not efficiently explored. On the other side, an undersized population is more agile and can be influenced by small changes. Nevertheless, not all solutions are adequately explored. The cost function is then applied to each individual, giving a value to each one.

Once each initial individual has a score or costs function value, they are scaled. The scaled values are known as expectation values. An individual with a cost function value with a big difference from the rest is undesirable. This individual will influence the rest of the population, risking the algorithm falling into a local minimum. In order to minimise the risk, there are four different options to scale the

values. The first is to rank the values; individuals are ordered based on the cost function value. The second output values are scaled between zero and one in the proportional method. The third option is to select the best n individuals from the pool. Last but not least, the average value is used as a reference in the linear shift method.

After sorting the individuals based on their performance, it is time to generate the next generation. Such a generation will be composed of an elite, cross and mutated individuals. Elite individuals are the best ones from the previous generation, chosen and used as they are without modifications. When individuals are crossed, some of the elite are chosen and combined. The combination is performed by using half-half genes from the donor parents. The last option is to perform mutations; this operation consists of taking random individuals and modifying random parts of their genetic material. Once the next generation has been decided, it is evaluated with the cost function, and the process starts again.

EA can also make use of linear constraints. The initial population and mutations are chosen based on such constraints in this particular case. Usually, constraints are imposed on the design values or genetic material. For the particular case of this research, linear constraints can be found in Section 2.2.3.

4.4.3 Particle Swarm Algorithm

Particle swarm will find a local unconstrained minimum for the cost function based on bird flocking, fish schooling, and swarming theory. Swarming theory describes many animals' behaviours, such as birds collaborating and finding resources more efficiently. This algorithm starts with a specific swarm size that explores the search space, proposing new positions or waypoints for the UAS to follow.

The main driver for change in the particle swarm is the speed of each particle. Equation 4.26 represents how the speed is changed for each particle. As reflected in the equation, the speed depends upon several parameters. First in appearance is the inertia (W); this parameter will amplify or reduce the previous velocity. The main algorithm can modify the inertia itself whenever a better point is found, also depending on the number of previous iterations without change. The parameters y_1 and y_2 are known as the cognition and social adjustment weights. As developed by Reference [56], y_1 helps expand the search space, increasing the ability of the algorithm to explore new areas. The social adjustment weight y_2 pulls the swarm towards known locations. Without the cognition adjustment weight, the search space statistically shrinks. The difference between the current position and the best position seen by the particle is $p - x$. The parameter $g - x$ represents the difference between the best position in the current neighbourhood.

$$v = W \cdot v + y_1 \cdot u_1 \cdot (p - x) + y_2 \cdot u_2 \cdot (g - x) \qquad (4.26)$$

Once the speed of the particles is correctly defined, Equation 4.27 uses it to modify the current set of particles. The new set of particles goes through the cost function, and new performance values are obtained. Based on the results, vectors p and g are modified. The modification is based on comparing the best position in the neighbourhood and the best position seen by each particle. Iterations continue until a stopping criterion has been reached.

$$x = x + v \qquad (4.27)$$

In our work, the number of particles that better perform is 50, ten times the number of waypoints. It is worth noting that the number of individuals in the EA method is the same, 50 individuals. Other relevant parameters and their implementation can be found in [49] for both EA and particle swarms.

4.4.4 Hybrid optimisation

Hybrid optimisation algorithms combine several optimisation strategies. The central concept here is to use the best qualities out of two optimisation algorithms. Common knowledge in hybridisation is to start with an exploratory algorithm. Good candidates for a prime algorithm are the so-called intelligent algorithms. They excel in exploring a vast search space but require many iterations to converge. The secondary algorithm is usually one of the non-intelligent optimisation algorithms. These algorithms will converge faster to the nearest minimum, given an appropriate starting point.

Care must be taken when selecting the switching condition. In the case of this study, the threshold is defined through the study of previous results. Chapter 5 compares different results obtained with intelligent algorithms and a decision based on the results. Such a chapter also covers the threshold to switch from prime to secondary.

4.5 Way forward

Various optimisation techniques have been described throughout this chapter. The main goal of it was not to make a decision on which one is most suitable for the research question. The main goal was to provide an overview to the reader on the beginnings of route optimisation and the latest research directions in the field.

Dealing with constraints is part of the cost function. Because of that, various techniques have been proposed together with two different cost functions. Said cost functions have the mission to transform constraints into a final value. A final value gives the optimisation algorithm a notion of how suitable the proposed solution is.

The following chapter, Chapter 5, is devoted to discussing the results. Some of the questions the reader might have will be answered there. The chapter is intended to answer questions like which algorithm is best, which algorithm makes better use of cost function one or two, and which hybridisation technique performs best. The narrative of Chapter 5 has a chronological order; the reasons why one algorithm was chosen for the first journal [47] and how it evolved towards the described algorithm in the journal [57].

References

[1] J V Morro et al. "Automated design of waveguide filters using Aggressive Space Mapping with a segmentation strategy and hybrid optimization techniques". In: *IEEE MTT-S International Microwave Symposium Digest, 2003*. Vol. 2. IEEE, 2003, pp. 1215–1218.

[2] Wenyu Sun and Ya-Xiang Yuan. *Optimization theory and methods: nonlinear programming*. Vol. 1. Springer Science & Business Media, 2006.

[3]   Antonio Bicchi, Giuseppe Casalino, and Corrado Santilli. "Planning shortest bounded-curvature paths for a class of nonholonomic vehicles among obstacles". In: *Proceedings of 1995 IEEE International Conference on Robotics and Automation*. Vol. 2. 4. IEEE, 1996, pp. 1349–1354.

[4]   C Goerzen, Z Kong, and B Mettler. "A survey of motion planning algorithms from the perspective of autonomous UAV guidance". In: *Journal of Intelligent and Robotic Systems: Theory and Applications*. Vol. 57. 1-4. 2010, pp. 65–100.

[5]   Yijing Zhao, Zheng Zheng, and Yang Liu. "Survey on computational-intelligence-based UAV path planning". In: *Knowledge-Based Systems* 158 (2018), pp. 54–64.

[6]   Baoye Song, Gaoru Qi, and Lin Xu. "A survey of three-dimensional flight path planning for unmanned aerial vehicle". In: *2019 Chinese Control And Decision Conference (CCDC)*. IEEE. 2019, pp. 5010–5015.

[7]   Jacky Baltes and Nicholas Hildreth. "Adaptive Path Planner for Highly Dynamic Environments". In: *RoboCup-2000: Robot Soccer World Cup IV*. 2001, pp. 76–85.

[8]   Chi Tsun Cheng et al. "Cooperative path planner for UAVs using ACO algorithm with gaussian distribution functions". In: *Proceedings - IEEE International Symposium on Circuits and Systems*. IEEE, 2009, pp. 173–176.

[9]   MI Lizarraga and GH Elkaim. "Spatially deconflicted path generation for multiple UAVs in a bounded airspace". In: *Proceedings of IEEE/ION PLANS 2008*. 2008, pp. 1213–1218.

[10] *Analytical Graphics, Inc.* https://www.agi.com/products/stk.

[11] R J Szczerba. "Threat netting for real-Time, intelligent route planners". In: *IDC 1999 - 1999 Information, Decision and Control, Data and Information Fusion Symposium, Signal Processing and Communications Symposium and Decision and Control Symposium: Proceedings*. 1999.

[12] Jesus Manuel de la Cruz et al. "Evolutionary path planner for UAVs in realistic environments". In: *Proceedings of the 10th annual conference on Genetic and evolutionary computation - GECCO '08*. New York, New York, USA: ACM Press, 2008, p. 1477.

[13] Arthur Richards and J P How. "Aircraft trajectory planning with collision avoidance using mixed integer linear programming". In: *Proceedings of the 2002 American Control Conference (IEEE Cat. No.CH37301)*. IEEE, 2002, 1936–1941 vol.3.

[14] Eva Besada-Portas et al. "Evolutionary Trajectory Planner for Multiple UAVs in Realistic Scenarios". In: *IEEE Transactions on Robotics* 26.4 (2010), pp. 619–634.

[15] Efrén Mezura-Montes and Carlos A Coello Coello. "Constraint-handling in nature-inspired numerical optimization: Past, present and future". In: *Swarm and Evolutionary Computation*. Vol. 1. 4. 2011, pp. 173–194.

[16] Efrén Mezura-Montes, Carlos A Coello Coello, and Edy I Tun-Morales. "Simple feasibility rules and differential evolution for constrained optimization". In: *Mexican International Conference on Artificial Intelligence*. Springer. 2004, pp. 707–716.

[17] Yong Wang et al. "Evolutionary dynamic constrained optimization: Test suite construction and algorithm comparisons". In: *Swarm and Evolutionary Computation* 50 (2019), p. 100559.

[18] Yingchun Chen, Ye Zhao, and Huakui Wang. "Real time path planning for UAV based on Focused D". In: *The Fourth International Workshop on Advanced Computational Intelligence*. 1. IEEE, 2011, pp. 80–85.

[19] Luca De Filippis, Giorgio Guglieri, and Fulvia Quagliotti. "Path Planning Strategies for UAVS in 3D Environments". In: *Journal of Intelligent and Robotic Systems* 65.1-4 (2012), pp. 247–264.

[20] Esten Ingar Grøtli and Tor Arne Johansen. "Path Planning for UAVs Under Communication Constraints Using SPLAT! and MILP". In: *Journal of Intelligent and Robotic Systems* 65.1-4 (2012), pp. 265–282.

[21] Shaimaa Ahmed et al. "Energy efficient path planning techniques for UAV-based systems with space discretization". In: *2016 IEEE Wireless Communications and Networking Conference*. Vol. 2016-Septe. Wcnc. IEEE, 2016, pp. 1–6.

[22] Shuowen Zhang, Yong Zeng, and Rui Zhang. "Cellular-Enabled UAV Communication: A Connectivity-Constrained Trajectory Optimization Perspective". In: *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[23] J Tisdale, Zuwhan Kim Zuwhan Kim, and J Hedrick. "Autonomous UAV path planning and estimation". In: *IEEE Robotics and Automation Magazine*. Vol. 16. 2. 2009, pp. 35–42.

[24] Seungkeun Kim et al. "Coordinated trajectory planning for efficient communication relay using multiple UAVs". In: *Control Engineering Practice*. Vol. 29. Elsevier, 2014, pp. 42–49.

[25] Alexandra Grancharova et al. "UAVs Trajectory Planning by Distributed MPC under Radio Communication Path Loss Constraints". In: *Journal of Intelligent and Robotic Systems* 79.1 (2015), pp. 115–134.

[26] Pawel Ladosz, Hyondong Oh, and Wen-Hua Chen. "Prediction of air-to-ground communication strength for relay UAV trajectory planner in urban environments". In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vol. 2017-Septe. IEEE, 2017, pp. 6831–6836.

[27] Guangchi Zhang et al. "Securing UAV Communications via Trajectory Optimization". In: *2017 IEEE Global Communications Conference, GLOBECOM 2017 - Proceedings*. Vol. 2018-Janua. 2018, pp. 1–6.

[28] I K Nikolos et al. "Evolutionary algorithm based offline/online path planner for uav navigation". In: *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)* 33.6 (2003), pp. 898–912.

[29] Ozgur Koray Sahingoz. "Generation of Bezier Curve-Based Flyable Trajectories for Multi-UAV Systems with Parallel Genetic Algorithm". In: *Journal of Intelligent and Robotic Systems* 74.1-2 (2014), pp. 499–511.

[30] Peng Yang et al. "Path planning for single unmanned aerial vehicle by separately evolving waypoints". In: *IEEE Transactions on Robotics* 31.5 (2015), pp. 1130–1146.

[31] Mohamed Elhoseny, Alaa Tharwat, and Aboul Ella Hassanien. "Bezier curve based path planning in a dynamic field using modified genetic algorithm". In: *Journal of Computational Science* 25 (2018), pp. 339–350.

[32] Zhihong Peng et al. "Online route planning for UAV based on model predictive control and particle swarm optimization algorithm". In: *Proceedings of the 10th world congress on intelligent control and automation*. IEEE. 2012, pp. 397–401.

[33]   Zhang Yuheng, Zhang Liyan, and Liu Chunpeng. "3-D deployment optimiza-
       tion of UAVs based on particle swarm algorithm". In: *2019 IEEE 19th Interna-
       tional Conference on Communication Technology (ICCT)*. IEEE. 2019, pp. 954–957.

[34]   Wentao Liu et al. "Particle swarm optimization for interference-limited un-
       manned aerial vehicle-assisted networks". In: *IEEE Access* 8 (2020), pp. 174342–
       174352.

[35]   Shikai Shao et al. "Efficient path planning for UAV formation via comprehen-
       sively improved particle swarm optimization". In: *ISA transactions* 97 (2020),
       pp. 415–430.

[36]   Shikai Shao et al. "A New Method of Solving UAV Trajectory Planning Under
       Obstacles and Multi-Constraint". In: *IEEE Access* 9 (2021), pp. 161161–161180.

[37]   Fang Wangsheng, Wang Chong, and Zhao Ruhua. "Application of simulated
       annealing particle swarm optimization in complex three-dimensional path plan-
       ning". In: *Journal of Physics: Conference Series*. Vol. 1873. 1. IOP Publishing. 2021,
       p. 012077.

[38]   Chen Huang. "A novel three-dimensional path planning method for fixed-
       wing UAV using improved particle swarm optimization algorithm". In: *Inter-
       national Journal of Aerospace Engineering* 2021 (2021).

[39]   Supatcha Chaimatanan, Daniel Delahaye, and Marcel Mongeau. "A hybrid
       metaheuristic optimization algorithm for strategic planning of 4D aircraft tra-
       jectories at the continental scale". In: *IEEE Computational Intelligence Magazine*
       (2014).

[40]   Rui Dai et al. "Quality-aware UAV coverage and path planning in geometri-
       cally complex environments". In: *Ad Hoc Networks* 73 (2018), pp. 95–105.

[41]   Zhi Wang, Gang Liu, and Ao Li. "Three-dimensional path planning of UVAs
       based on simulated annealing and particle swarm optimization hybrid algo-
       rithm". In: *2021 IEEE 3rd International Conference on Civil Aviation Safety and
       Information Technology (ICCASIT)*. IEEE. 2021, pp. 522–525.

[42]   Bo Zhang et al. "Genetic Algorithm enabled Particle Swarm Optimization for
       Aerial Base Station Deployment". In: *2021 IEEE 94th Vehicular Technology Con-
       ference (VTC2021-Fall)*. IEEE. 2021, pp. 1–7.

[43]   Jan A Snyman, Daniel N Wilke, et al. *Practical mathematical optimization*. Springer,
       2005.

[44]   Mark French. "Fundamentals of Optimization". In: *Springer International Pub-
       lishing, DOI* 10 (2018), pp. 978–3.

[45]   Kenneth Lange. *Optimization*. Vol. 95. Springer Science & Business Media, 2013.

[46]   Agoston E Eiben, James E Smith, et al. *Introduction to evolutionary computing*.
       Vol. 53. Springer, 2003.

[47]   Adrian Exposito, Dominic Schupke, and Hector Esteban. "Route Optimisa-
       tion for Maximum Air to Ground Channel Quality". In: *IEEE Access* 8 (2020),
       pp. 203619–203630.

[48]   Inc. The MathWorks. *Optimization Toolbox*. `https : / / www . mathworks . com /
       help/optim/`. Natick, Massachusetts, United State, 2022.

[49]   Inc. The MathWorks. *Global Optimization Toolbox*. `https : / / www . mathworks .
       com/help/gads/`. Natick, Massachusetts, United State, 2022.

[50] Jeffrey C Lagarias et al. "Convergence properties of the Nelder-Mead simplex method in low dimensions". In: *SIAM Journal on Optimization* (1998).

[51] Tamara G Kolda, Robert Michael Lewis, and Virginia Torczon. "Optimization by direct search: New perspectives on some classical and modern methods". In: *SIAM review* 45.3 (2003), pp. 385–482.

[52] Lester Ingber. "Adaptive simulated annealing (ASA): Lessons learned". In: *Control and Cybernetics* (1996).

[53] Larry Nazareth. "A relationship between the BFGS and conjugate gradient algorithms and its implications for new algorithms". In: *SIAM Journal on Numerical Analysis* 16.5 (1979), pp. 794–800.

[54] H-M Gutmann. "A radial basis function method for global optimization". In: *Journal of global optimization* 19.3 (2001), pp. 201–227.

[55] Thomas Bäck and Hans-Paul Schwefel. "An Overview of Evolutionary Algorithms for Parameter Optimization". In: *Evolutionary Computation* 1.1 (1993), pp. 1–23.

[56] Yuhui Shi and Russell Eberhart. "A modified particle swarm optimizer". In: *1998 IEEE international conference on evolutionary computation proceedings. IEEE world congress on computational intelligence (Cat. No. 98TH8360)*. IEEE. 1998, pp. 69–73.

[57] Adrián Expósito García, Héctor Esteban González, and Dominic Schupke. "Hybrid Route Optimisation for Maximum Air to Ground Channel Quality". In: *Journal of Intelligent & Robotic Systems* 105.2 (2022), pp. 1–16.

Chapter 5

Most suitable optimisation strategy

Since this document started nearly a hundred pages ago, many concepts have appeared. Figure 1.3 is brought from the beginning of this document as Figure 5.1. It is convenient to revisit the previous concepts and define the aim of this chapter. Starting from Chapter 2, said chapter covered the first third of Figure 5.1. In a nutshell, this part aims to lay the foundations and prepare the required data by the channel model and the optimisation algorithm. The second part of Figure 1 can be found in Chapter 3. This chapter describes the formulation of the channel model and related concepts. The third and last portion of Figure 5.1 is devoted to the optimisation algorithm, covered in Chapter 4. The chapter mentioned provided a set of suitable candidates to find the optimal. This search will use one of the two candidate cost functions (Equation 4.11 and Equation 4.12).

Detaching this chapter from the previous is advantageous. The current chapter combines the knowledge acquired throughout the previous ones. Justifying the choice for a specific optimisation strategy may involve revisiting concepts presented in chapters 2 to 4. The detachment of Chapter 5 from 4 avoids the growth of the optimisation chapter and contributes to its clarity and isolation.

The central purpose of this chapter is to present the acquired results. Each optimisation strategy is tested within four scenarios designed to find their limits. Boundary conditions applied to each scenario are also included in the section. Having similar conditions ensures fair analysis and comparison of the obtained data. The scheme of this chapter guides the reader from the firstly tested optimisation strategies to the latest and more complex ones. Five approaches are presented, along with the main metrics that define their performance.

5.1 Designed test scenarios and methodology

Testing the strength and weaknesses of each optimisation strategy is a time-consuming activity. Various scenarios have been defined to assess the performance boundaries. Each scenario will have a set of unique conditions. Those conditions might be dependent on terrain, air space or coverage conditions. One thing is sure; those conditions are explicitly designed to challenge the capabilities of each optimisation strategy.

5.1.1 Scenario 1

The first scenario used as a benchmark includes a challenging air-space situation, as described in Figure 5.2a. As seen in the figure mentioned, the flight is supposed to connect the airport of Munich with the city centre of Munich. This flight path is chosen due to its applicability in future scenarios. The public will choose an aerial bridge between the two strategic locations if the cost is acceptable. From a communications perspective, it is also fascinating. The area in between covers many areas:
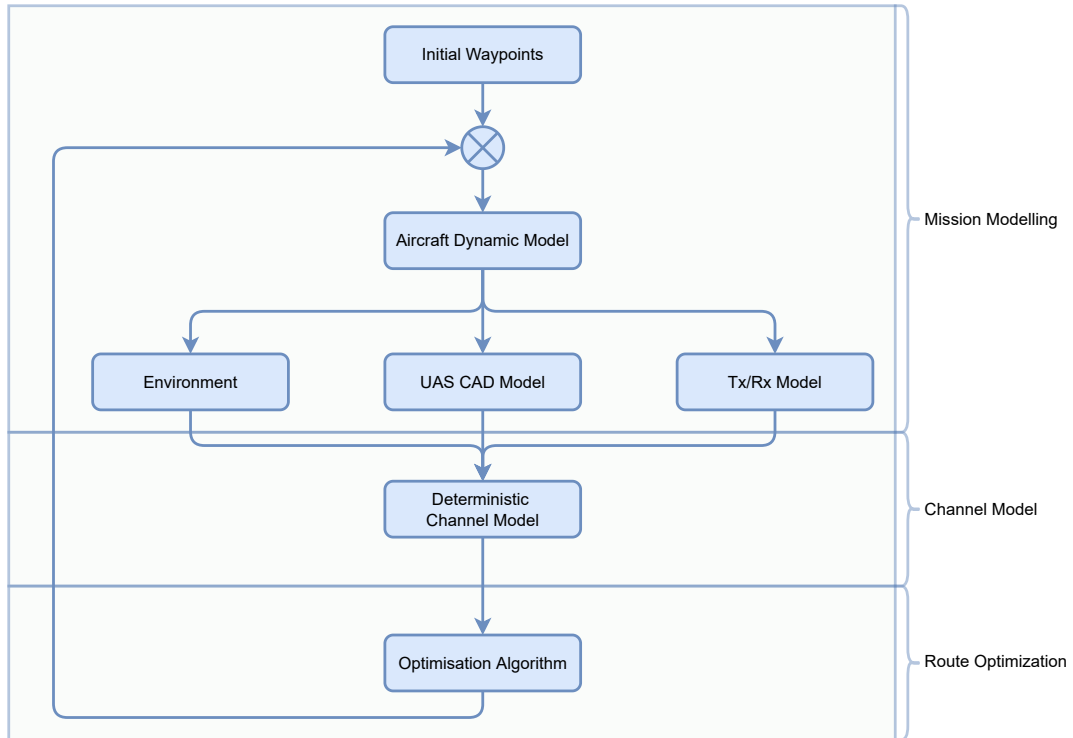
FIGURE 5.1: Optimisation workflow

urban, near-urban, and countryside. The heterogeneous surroundings are an excellent choice to analyse how the optimiser can adapt to these changes in propagation characteristics.

In this case, two no fly zones (NFZs) positioned along the fastest route defined by $\mathcal{L}_{min}^{WP}$. Although the unmanned aerial system (UAS) is constantly pulled towards $\mathcal{L}_{min}^{WP}$, the NFZs invalidates the route. From a pure air-space perspective and neglecting the state of the network, which will be described soon, the UAS has only one option: circumnavigate as close as possible to the NFZ. Any other route distant from $\mathcal{L}_{min}^{WP}$ will have a penalisation.

Given that this scenario focuses on avoiding any NFZ, the network should not pose a threat. A favourable network condition helps to evaluate NFZ avoidance without influence from other parameters. Many base stations are near the route, as seen in Figure 5.2b. Both east and west are well served with ground network (GN) stations from a fictitious provider. The terrain contributes to ease the network availability; scenario 1 has no mountains that can block line of sight (LOS) in the air to ground (A2G) channel.

The expectations for a successful optimisation are twofold. The first and most important requirement is to avoid crossing any NFZ. The cost function, either penalty or adaptive, is already designed to take NFZ into account. The second expectation is to raise the minimum received power and hopefully achieve a better average received power. When results are analysed, the difference between a restriction and an objective is also considered.

5.1.2 Scenario 2

From an air-space situation perspective, nothing has changed between Figure 5.2a and 5.3a. Both NFZs remain applicable in this scenario. The UAS is still supposed to

(A) Terrain view Scenario 1.

(B) Coverage view Scenario 1.
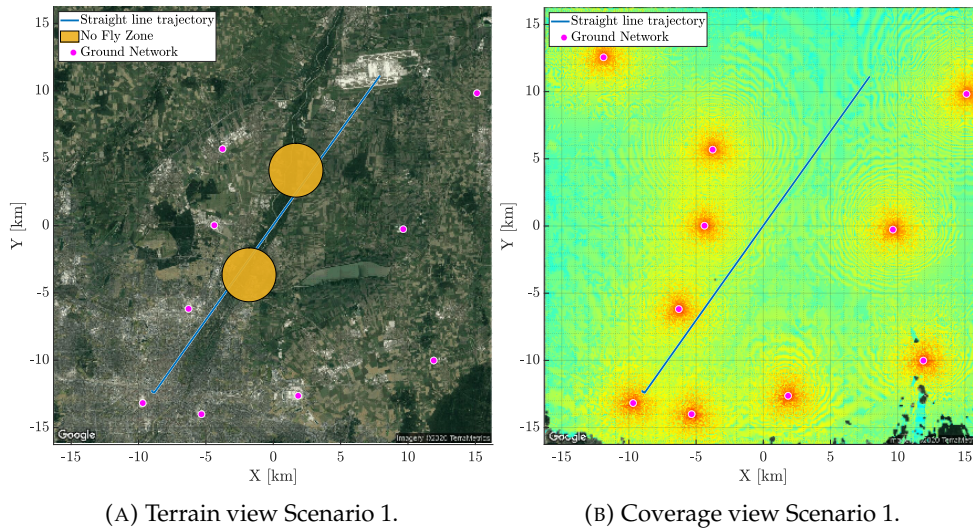
FIGURE 5.2: Scenario 1: Munich Airport to Munich central station [1]

(A) Terrain view Scenario 2.

(B) Coverage view Scenario 2.
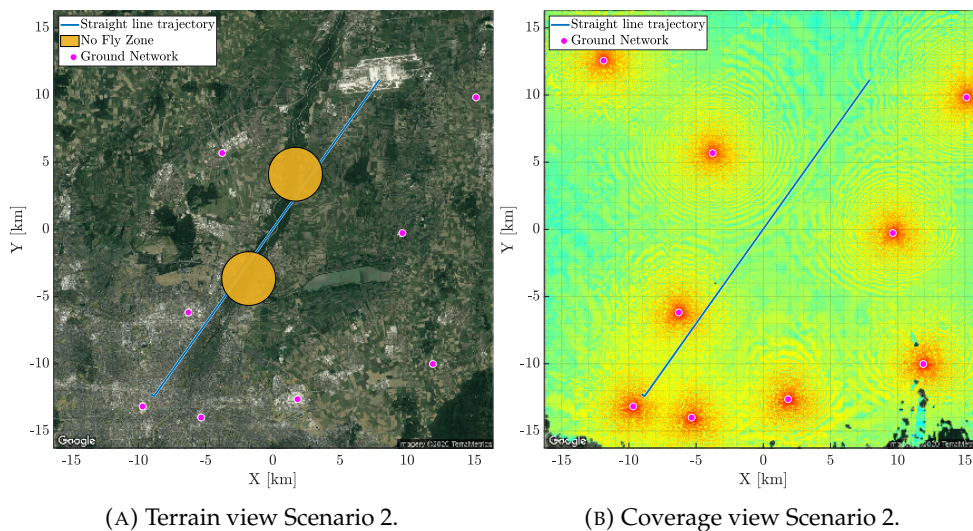
FIGURE 5.3: Scenario 2: Munich Airport to Munich central station
with an antenna failure [1].

go from the Munich airport to the city centre. The situation remains the same from a propagation perspective, provided that the terrain has not changed.

The most noticeable change that justifies this scenario is the malfunction of one of the GN serving the UAS. Since one base station (BS) is missing, the UAS is supposed to satisfy all constraints. The optimiser is expected to find a route that avoids NFZ, keeps a minimum flight time, and improves the minimum received power.

By observation of the coverage heat map, a first estimation of the resulting flight path can be made. The malfunctioning BS is located in the middle of the map and serves a significant portion of the original route. As a result, the UAS has to invest flight time in getting closer to the other BS and compensate for the drop in average received power.

5.1.3 Scenario 3

The before-mentioned scenarios covered the presence of NFZ and a network malfunction. The scenario that is about to be presented has a different situation. In

(A) Terrain view Scenario 3   (B) Coverage view Scenario 3.
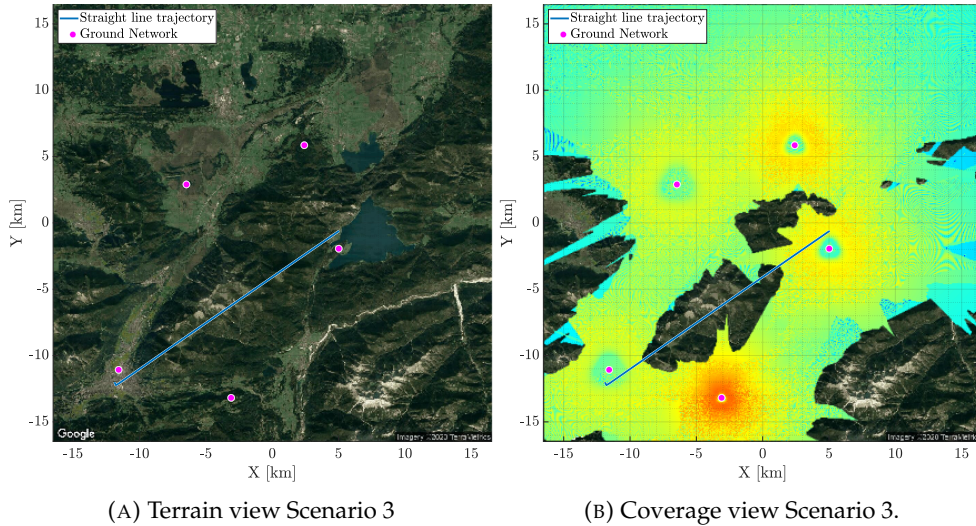
FIGURE 5.4: Scenario 3: Flight on the Bavarian alps [2].

this scenario, the UAS is requested to fly in an area known for its high mountains and deep valleys, the Bavarian alps. The UAS flies from Kochelsee to Garmisch-Partenkirchen as described in Figure 5.4a. Cities in this area are not so well connected, and any journey around has to circumvent mountains and use secondary roads. The situation suits a UAS used for medical transport where time is critical.

In this case, the $\mathcal{L}_{min}^{WP}$ route travels through the mountain; the terrain profile forbids the UAS from using the shortest route. Thanks to restriction r_1 (Equation 4.8) and objective a_1 (Equation 4.5), the optimisation algorithm is adequately configured to obtain a route that avoids going underground. The terrain profile significantly reduces the space with reasonable solutions.

Full connectivity in a mountain area does not fit reality; the main reason the area is only covered with five BS. This configuration, plus the height profile, reduces LOS to only one BS at a time. Figure 5.4b highlights the lack of connectivity in the area.

An acceptable solution must at least avoid the terrain. The optimiser is expected to find a solution with full coverage through the route, a better-received power value, and no collision with the ground. An even better solution goes around the mountains with minimum height variation, achieving a fuel-saving route.

5.1.4   Scenario 4

Applying scenarios one to three will challenge the capabilities of each optimisation strategy tested. Scenario four has been designed to have favourable visibility conditions, a favourable network and no restrictions in the air space. The main aim is to test the optimiser's performance under nearly perfect conditions and discard any unexpected behaviour. As an example, a sign of unexpected behaviour is an unbalanced prioritisation between flight time and channel performance.

This scenario is located in Alicante, on the Mediterranean sea. The terrain is primarily flat with hilly terrain towards the map's north, as seen in Figure 5.5a. Water, the main physical feature, does affect channel propagation. As already explained in Chapter 3, water is considered a reflective, uniform surface. From a connectivity perspective, the GN operator has deployed five BS. One BSs is located on the same island where the UAS is expected to land.

(A) Terrain view Scenario 4   (B) Coverage view Scenario 4.
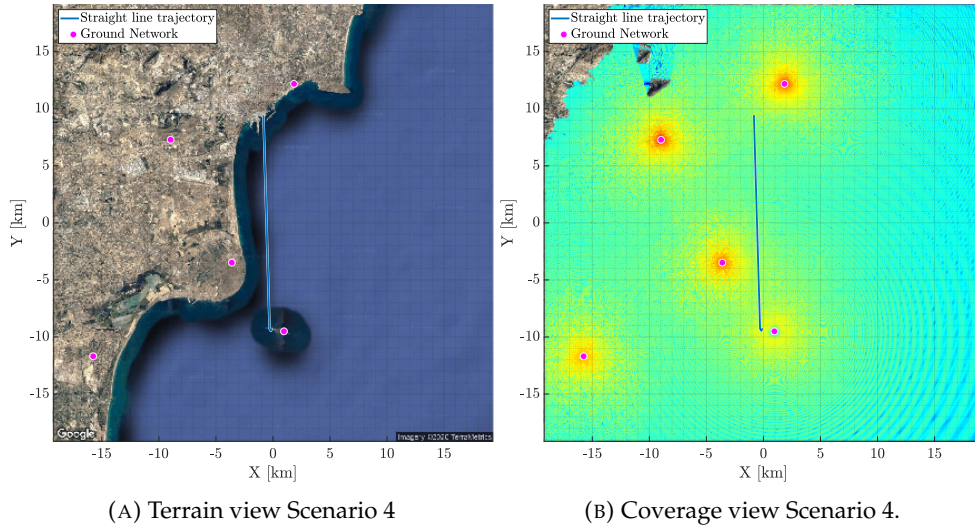
FIGURE 5.5: Scenario 4: Flight coast to island [3].

The expected result is to display the versatility of our approach by optimising the route through two very different environments. Other capabilities such as terrain or NFZ avoidance have been proven in previous scenarios. Results should show that the optimiser can improve channel performance by increasing the levels of the minimum and average power received.

5.1.5 Test and analysis methodology

Each cost function, scenario and optimisation algorithm produces different sets of data. Only simulations performed with the same inputs will be compared among each other. Evolutionary algorithm (EA) and particle swarm optimisation (PSO) rely on a random number generator to propose new waypoints, so its seed is changed at every simulation. The seed is changed to ensure that each algorithm is non-deterministic. All algorithms and simulations use $\mathcal{L}_{min}^{WP}$ as starting point for the optimisation. Except for Section 5.2, whose algorithms were tested only once, the rest of the algorithms have been tested ten times. Ten tests are done to ensure that mentioned algorithm's randomness is also considered.

The analysis and comparison of data require an explicit methodology. A methodology that will help to make a fair comparison among the results obtained by each optimisation strategy. Tables 5.1-5.12 provide helpful insight into the performance of each optimisation algorithm. Since two different cost functions have been applied, the cost function value is not a reliable source of comparison. Although not used for comparison among different algorithms, the metric does indicate if restrictions have been satisfied or not. In the case of the penalty cost function, a value below ten indicates restrictions satisfied. A value below one indicates the same for the adaptive cost function.

Another set of metrics are $N_{g_p[a,r]<1}$ and $N_{g_p[a,r]}$. The first value represents the number of iterations required to find a solution that satisfies all restrictions. This number is relevant as it indicates how fast is the optimisation strategy at complying with restrictions. In the case of hybridisation, a lower $N_{g_p[a,r]<1}$ means a lower time until handing over to the secondary algorithm. The final number of iterations, combined with the final cost function value, indicates if the algorithm can greatly improve early solutions.

TABLE 5.1: Multiple optimisation algorithms applied to scenario 1

Algorithm	$N_{g_p[a,r]<1}$	First Value < 1	$N_{g_p[a,r]}$	Final Value	$\Delta \overline{P}_R$ [%]
Particle Swarm Penalty	1193	0.2049	3404	0.1688	4.3445
Evolutionary Penalty	238	0.1856	3808	0.1609	4.0947
Surrogate	538	0.2089	679	0.1989	2.1155
Annealing	11729	0.2939	19713	0.2920	1.0746
GPS	108	0.2690	17715	0.2217	-0.0534

TABLE 5.2: Multiple optimisation algorithms applied to scenario 2

Algorithm	$N_{g_p[a,r]<1}$	First Value < 1	$N_{g_p[a,r]}$	Final Value	$\Delta \overline{P}_R$ [%]
Evolutionary Penalty	202	0.2079	3688	0.1870	3.4623
Particle Swarm Penalty	502	0.2276	4883	0.1903	3.1286
Annealing	653	0.3184	8579	0.3184	2.0629
Surrogate	538	0.2517	699	0.2464	1.4408
GPS	80	0.2341	20643	0.2085	0.5634

5.2 Search for a first optimisation algorithm

As the name of this section indicates, an algorithm must be appointed as the starting point. The first option is to look at the algorithms defined in the previous chapter to find such an algorithm. The primary candidates are EA, PSO, annealing, surrogate and generalized pattern search (GPS). Algorithms have been listed according to their capabilities to search for as much solution space as possible. EA optimisation is considered the most exploratory, whereas GPS is the less exploratory one. For this comparison, all algorithms use the penalty cost function.

The numbers in Table 5.1 show the expected behaviour and some other conclusions. First of all, EA and PSO are the ones with better solutions. Their achieved increment in average received power ($\Delta \overline{P}_R$) is almost identical, at least compared to the third algorithm in the list: surrogate. One benefit of surrogate is that it achieves acceptable results in less time. An acceptable solution is the one achieving compliance with restrictions. GPS and annealing take 17715 and 19713 iterations to achieve a solution. For the first scenario, EA and particle are the best candidates for a more extensive study.

The same scenario with a faulty BS gives a similar output, as seen in Table 5.2. Using an optimiser based on the EA or PSO yields similar results. Number of iterations, $N_{g_p[a,r]<1}$, final cost function values and the $\Delta \overline{P}_R$ are similar. Annealing performs similarly, but the average received power drops by 1%. Surrogate optimisation seems to be the fastest but with lower improvements in the received power. Although GPS needs the highest number of cost function evaluations, it still manages to find a suitable solution to the optimisation problem.

Moving on to a different location, the alps and its scenario, Table 5.3 gathers the obtained results. The reader might first notice the lack of data for the annealing. No data exists because the optimiser could not find a solution compliant with restrictions. Because of this non-compliance, the final value is above one. An also above ten because the penalty function has been applied. As restrictions are not fully satisfied, the 29% of channel improvement cannot be considered. On a more positive side, PSO and EA are still at the first positions and with similar results. EA finds its first compliant solution within 77 iterations, contrary to other optimisation approaches. Surrogate is still the fastest but not the one that genuinely maximises

TABLE 5.3: Multiple optimisation algorithms applied to scenario 3

Algorithm	$N_{g_p[a,r]<1}$	First Value < 1	$N_{g_p[a,r]}$	Final Value	$\Delta \overline{P}_R$ [%]
Particle Swarm Penalty	340	0.3426	7663	0.0774	31.5417
Evolutionary Penalty	77	0.1786	3127	0.1206	29.9205
GPS	718	0.1481	2965	0.1350	29.8538
Annealing	-	-	10369	10.1463	29.6322
Surrogate	194	0.1197	741	0.1104	27.6701

TABLE 5.4: Multiple optimisation algorithms applied to scenario 4

Algorithm	$N_{g_p[a,r]<1}$	First Value < 1	$N_{g_p[a,r]}$	Final Value	$\Delta \overline{P}_R$ [%]
Evolutionary Penalty	2	0.1822	4106	0.1266	4.0269
Particle Swarm Penalty	25	0.2288	4410	0.1305	3.9696
Annealing	2	0.1674	11329	0.1522	2.7530
GPS	2	0.1674	22812	0.1457	2.5088
Surrogate	68	0.2173	713	0.2126	1.6642

channel performance.

Scenario 4 is the last one to be analysed in this set of simulations, whose results are gathered in Table 5.4. A scenario with virtually no restrictions outputs minimum $N_{g_p[a,r]<1}$ for most optimisation approaches. Any cost function value greater than one corresponds to a trajectory non-compliant with map or height limits. Once again, EA and PSO have a better and almost equal performance. Although annealing and GPS achieve good results, the number of iterations to terminate is above ten thousand tries. Surrogate continues to be the fastest with lower maximisation of channel performance.

Summarising initial results on each candidate optimisation strategy, EA and PSO are the most promising techniques to continue with. The next phase focuses more deeply on the obtained results no matter the number of iterations. In this sense, both algorithms have shown good performances, expected to continue through different runs.

Annealing results in Table 5.3 show that it is possible for a simulation to end with no compliance with restrictions. This another reason to do more than one simulation with each optimisation technique. The perfect candidate for the next phase must accomplish a 100% compliance with restrictions. If the number is not achieved, such optimisation technique won't be studied any further.

5.3 Performance of evolutionary and particle swarm

This section focuses on comparing the performance of EA and PSO. As the previous section shows, both algorithms are excellent candidates for finding adequate trajectories for each scenario. The fundamentals of each algorithm have been explained in Section 4.4. The results of the EA using the penalty function can also be found in Reference [4]. Reference [4] has a broader scope but also covers the results obtained using PSO with the penalty function.

Table 5.5 gathers the first results, only applicable for the first scenario. Results obtained by the EA are not satisfactory as not all simulation results comply with the restrictions. The EA using the penalty cost function could not find a solution in one of the ten simulations. The PSO algorithm meets restrictions on each simulation. The metrics are also better compared to EA. PSO manages higher average received

TABLE 5.5: Comparison on the performance obtained by using evolutionary and particle swarm in scenario 1

	Evolutionary Penalty				Particle Swarm Penalty			
	$N_{g_p[a,r]<1}$	$N_{g_p[a,r]}$	ΔT [min]	$\Delta \overline{P}_R$ [%]	$N_{g_p[a,r]<1}$	$N_{g_a[a,r]}$	ΔT [min]	$\Delta \overline{P}_R$ [%]
	173	1381	4.16	4.47	421	3503	4.12	5.12
	634	2032	4.20	4.39	976	3358	3.96	4.65
	585	1604	4.02	4.37	1111	2598	4.11	4.54
	359	1459	4.37	4.36	577	4184	3.86	4.31
	502	1648	3.70	4.06	1135	4156	3.95	4.19
	682	1787	3.78	3.80	716	1910	3.92	3.94
	352	1656	3.54	3.60	1749	2998	4.03	3.73
	769	1735	3.52	2.40	933	2540	3.80	3.44
	255	1378	3.97	−0.73	911	2627	3.61	3.18
	-	-	-	-	1654	2883	4.62	2.53
Average	-	-	-	-	1019	3076	4.00	3.96

TABLE 5.6: Comparison on the performance obtained by using evolutionary and particle swarm in scenario 2

	Evolutionary Penalty				Particle Swarm Penalty			
	$N_{g_p[a,r]<1}$	$N_{g_p[a,r]}$	ΔT [min]	$\Delta \overline{P}_R$ [%]	$N_{g_p[a,r]<1}$	$N_{g_a[a,r]}$	ΔT [min]	$\Delta \overline{P}_R$ [%]
	206	1337	4.69	3.74	655	3033	4.03	3.27
	252	1207	3.75	2.72	355	3674	4.22	3.22
	528	1499	3.45	2.43	357	1584	4.15	2.98
	302	1288	3.51	2.20	498	3890	4.10	2.78
	252	1709	3.37	1.95	632	3641	3.78	2.67
	203	1228	4.33	1.77	489	2170	4.85	2.37
	38	998	4.53	0.46	742	2189	3.74	2.20
	156	1373	3.85	0.31	830	2472	3.53	2.11
	53	1331	4.54	0.29	653	2191	4.37	1.74
	215	1185	4.54	−0.34	342	2178	3.95	0.17
Average	221	1316	4.06	1.55	556	2703	4.07	2.35

power. As the EA could not satisfy all simulations' restrictions, other metrics in the table will not be commented on.

Although the EA does not reach the required performance for scenario 1, it does in scenario 2 with a reduced solution space. Numerical values can be found in Table 5.6. The results of Scenario 2 can be used for a comparison between EA and PSO results. The first value is the time taken to find a route that satisfies restrictions. EA requires 221 iterations on average, whereas PSO requires 556 iterations. Similar behaviour is found in $N_{g_a[a,r]}$, EA tends to be the fastest compared with PSO. Albeit faster, the EA does not achieve higher $\Delta \overline{P}_R$ than PSO. The difference in the number of iterations is not higher enough to justify the difference in $\Delta \overline{P}_R$.

Similar results to Table 5.5 can also be found in Table 5.7. Scenario 3 is a real challenge for the EA. Out of the ten simulations, only two show compliance with restrictions. As in previous scenarios, the PSO can handle the particularities of scenario 3. It should be noted that $\Delta \overline{P}_R$ achieves up to a 31.50 % increase because of the nature of $\mathcal{L}_{min}^{WP}$. This minimal distance and time route goes through the central mountains, which causes parts of the route with no connection to the GN. Aside from this scenario particularity, it is also worth noting that the proposed route takes a different average of four minutes compared with $\mathcal{L}_{min}^{WP}$.

Scenario 4 results, available in Table 5.8, show that both algorithms can find routes compliant with restrictions. Parameter $N_{g_p[a,r]<1}$ has the same behaviour as

TABLE 5.7: Comparison on the performance obtained by using evolutionary and particle swarm in scenario 3

	Evolutionary Penalty				Particle Swarm Penalty			
	$N_{g_p[a,r]<1}$	$N_{g_p[a,r]}$	ΔT [min]	$\Delta \overline{P}_R$ [%]	$N_{g_p[a,r]<1}$	$N_{g_a[a,r]}$	ΔT [min]	$\Delta \overline{P}_R$ [%]
	154	1637	4.16	30.25	623	2879	4.40	31.50
	221	1620	3.86	29.76	101	3934	4.15	31.43
	-	-	-	-	131	4264	4.23	31.37
	-	-	-	-	181	3259	4.29	31.32
	-	-	-	-	157	2637	4.23	31.28
	-	-	-	-	278	3744	4.38	31.27
	-	-	-	-	190	3479	4.22	31.11
	-	-	-	-	304	3631	4.47	30.87
	-	-	-	-	75	3066	4.61	30.85
	-	-	-	-	85	3406	3.24	27.68
Average	-	-	-	-	213	3430	4.22	30.87

TABLE 5.8: Comparison on the performance obtained by using evolutionary and particle swarm in scenario 4

	Evolutionary Penalty				Particle Swarm Penalty			
	$N_{g_p[a,r]<1}$	$N_{g_p[a,r]}$	ΔT [min]	$\Delta \overline{P}_R$ [%]	$N_{g_p[a,r]<1}$	$N_{g_a[a,r]}$	ΔT [min]	$\Delta \overline{P}_R$ [%]
	2	1736	3.08	4.27	256	3816	3.32	4.09
	16	1703	3.82	3.74	217	3098	3.44	3.91
	47	1170	3.15	3.15	194	3830	3.46	3.89
	149	1471	3.25	3.04	49	2870	3.42	3.88
	2	1276	3.07	2.40	235	3915	3.38	3.73
	32	1202	2.16	0.99	140	2796	3.54	3.60
	88	1118	2.02	0.82	139	3843	3.27	3.27
	2	962	1.75	0.67	50	2019	3.67	2.94
	2	984	1.82	0.31	80	1535	3.66	2.07
	39	1077	1.75	0.12	140	1243	2.49	0.72
Average	38	1270	2.59	1.95	150	2897	3.37	3.21

in scenario 2. The EA optimisation approach requires a significantly lower amount of iterations to find a route compliant with restrictions. The amount of iterations is three times higher for PSO optimisation. The number of iterations to finish a simulation required by PSO doubles the number of EA. Although PSO takes longer, it can produce routes with higher $\Delta \overline{P}_R$.

The four scenarios and the performance obtained by each optimisation approach leave a clear winner; PSO. Tables 5.5 and 5.7 show that the evolution-based optimiser cannot find a solution in all the simulations. Irrespective of the compliance to objectives and the number of iterations, restrictions must be met. If restrictions are not met, the route might height limits, enter an NFZ, violate map limits or not find a route that improves the minimum received power. Tables 5.6 and 5.8 show that PSO finds routes with better $\Delta \overline{P}_R$. Both reasons are the main drivers for PSO to continue in the competition for the selected optimisation approach.

5.4 Improving particle swarm

The previous section determined that an optimiser based on PSO is better than one based on the EA approach. Still, the performance showed that PSO requires more

TABLE 5.9: Comparison on the performance obtained by changing
the cost function for particle swarm in scenario 1

Particle Swarm Penalty				Particle Swarm Adaptive			
$N_{g_p[a,r]<1}$	$N_{g_p[a,r]}$	ΔT [min]	$\Delta \overline{P}_R$ [%]	$N_{g_a[a,r]<1}$	$N_{g_a[a,r]}$	ΔT [min]	$\Delta \overline{P}_R$ [%]
421	3503	4.12	5.12	625	2384	3.93	4.19
976	3358	3.96	4.65	232	4036	4.78	3.38
1111	2598	4.11	4.54	665	4208	4.94	3.30
577	4184	3.86	4.31	477	3042	4.45	2.23
1135	4156	3.95	4.19	130	2695	4.53	1.19
716	1910	3.92	3.94	160	2515	4.27	0.61
1749	2998	4.03	3.73	705	3276	4.47	0.42
933	2540	3.80	3.44	467	3658	4.32	0.17
911	2627	3.61	3.18	568	3360	4.17	−0.95
1654	2883	4.62	2.53	350	3779	5.26	−1.13
Average							
1019	3076	4.00	3.96	438	3296	4.51	1.34

iterations to find a solution. Two promising mechanisms could reduce the before-mentioned metric. Both mechanisms have been described in sections 4.3.2 and 4.4.4, respectively.

5.4.1   New cost function

First resource is to change the cost function. By doing so, the optimiser is better guided towards the solution. The description of the new cost function can be found in Section 4.3.2, but one of the most predominant advantages is continuous behaviour. A continuous behaviour gives the optimiser better information to determine the next search direction.

As in the previous comparison, each scenario has been evaluated ten times. After the simulations' finalisation, data was gathered for each simulation and condensed into a set of tables. These tables are intended to give the reader a comprehensive view of the optimiser performance. Tables 5.9 to 5.12 compare the results obtained using the penalty cost function and results obtained with the adaptive one. The test will show if changing the cost function brings an advantage or if a new optimisation algorithm is required.

Table 5.9 provides an overview of the obtained results in scenario 1. Penalty cost function results have been taken from Table 5.5. Scenario 1 shows that the adaptive cost function brings an advantage over the penalty cost function. The optimiser can find a route compliant with restrictions faster. The change does not influence the final number of iterations in the cost function, altitude variation or route time. Parameter $\Delta \overline{P}_R$ is reduced to half; in some cases, there is even a decrease in the average received power. It is worth noting that even if the average decreases, every trajectory point is guaranteed to be above the noise floor.

Scenario 2 shows similar behaviour in Table 5.10. The new cost function helps PSO find the first solution within fewer iterations. Contrary to the previous table, the difference between the final number of iterations and average $\Delta \overline{P}_R$ has been reduced. The difference in the number of iterations is negligible. $\Delta \overline{P}_R$ shows improvements on all simulations and an average similar to the penalty function.

A flight in the alps also benefits from the change from the penalty to the adaptive cost function. $N_{g_p[a,r]<1}$ is significantly reduced with the new cost function, going from an average of 213 to 76 iterations. If the simulation stops right after finding the first compliant solution, the adaptive cost function considerably reduces calculation

TABLE 5.10: Comparison on the performance obtained by changing the cost function for particle swarm in scenario 2

	Particle Swarm Penalty				Particle Swarm Adaptive		
$N_{g_p[a,r]<1}$	$N_{g_p[a,r]}$	ΔT [min]	$\Delta \overline{P}_R$ [%]	$N_{g_a[a,r]<1}$	$N_{g_a[a,r]}$	ΔT [min]	$\Delta \overline{P}_R$ [%]
655	3033	4.03	3.27	416	2127	4.53	3.12
355	3674	4.22	3.22	99	2791	4.03	2.97
357	1584	4.15	2.98	207	2510	3.97	2.67
498	3890	4.10	2.78	146	3747	4.30	2.47
632	3641	3.78	2.67	215	2533	4.61	2.38
489	2170	4.85	2.37	482	2407	3.87	1.96
742	2189	3.74	2.20	84	2232	3.93	1.84
830	2472	3.53	2.11	191	1626	4.30	1.76
653	2191	4.37	1.74	599	4218	4.03	1.09
342	2178	3.95	0.17	373	3777	5.20	0.64
Average	556	2703	4.07	2.35	282	2797	4.28	2.09

TABLE 5.11: Comparison on the performance obtained by changing the cost function for particle swarm in scenario 3

	Particle Swarm Penalty				Particle Swarm Adaptive		
$N_{g_p[a,r]<1}$	$N_{g_p[a,r]}$	ΔT [min]	$\Delta \overline{P}_R$ [%]	$N_{g_a[a,r]<1}$	$N_{g_a[a,r]}$	ΔT [min]	$\Delta \overline{P}_R$ [%]
623	2879	4.40	31.50	31	4557	4.31	31.69
101	3934	4.15	31.43	48	4138	4.31	31.24
131	4264	4.23	31.37	15	4501	4.49	31.13
181	3259	4.29	31.32	207	3357	4.14	31.00
157	2637	4.23	31.28	20	4102	4.06	30.93
278	3744	4.38	31.27	141	3681	4.31	30.92
190	3479	4.22	31.11	97	4380	4.56	30.88
304	3631	4.47	30.87	84	3254	8.69	29.40
75	3066	4.61	30.85	22	1819	3.81	29.27
85	3406	3.24	27.68	94	1480	7.69	27.85
Average	213	3430	4.22	30.87	76	3527	5.04	30.43

times. The difference on $\Delta \overline{P}_R$ is lower between both cost functions, to the point of being almost equal. Following the proposed final route, the time was increased by around five minutes for both cost functions.

In scenario 4; $N_{g_p[a,r]<1}$ is reduced with the adaptive cost function. Compared with the penalty cost function, the number of iterations is reduced to half. In this scenario, there is no penalty on $\Delta \overline{P}_R$, and even the average value shows minimal improvement. The increase in time is maintained at around three minutes extra for covering the proposed route.

5.4.2 Hybridisation

The application of this technique is driven by the result obtained with the adaptive cost function. In view that $N_{g_p[a,r]<1}$ is greatly improved, the primary algorithm (PSO) could be stopped at an early stage. After the disruption of the main algorithm, a secondary algorithm can continue with the optimisation. The secondary algorithm does not need to explore the solution space thoroughly. This algorithm, which is more eager to find the minimum in the cost function, and less exploratory, find the closest optimum as quick as possible. The selected algorithms for the second search are Nelder-Mead simplex and GPS. Both algorithms have been described in sections 4.4.1 and 4.4.1, respectively.

TABLE 5.12: Comparison on the performance obtained by changing
the cost function for particle swarm in scenario 4

	Particle Swarm Penalty			Particle Swarm Adaptive			
$N_{g_p[a,r]<1}$	$N_{g_p[a,r]}$	ΔT [min]	$\Delta \overline{P}_R$ [%]	$N_{g_a[a,r]<1}$	$N_{g_a[a,r]}$	ΔT [min]	$\Delta \overline{P}_R$ [%]
256	3816	3.32	4.09	27	3529	3.82	3.99
217	3098	3.44	3.91	12	2903	3.49	3.89
194	3830	3.46	3.89	113	2492	3.57	3.89
49	2870	3.42	3.88	6	2614	3.17	3.72
235	3915	3.38	3.73	21	3461	3.33	3.62
140	2796	3.54	3.60	76	2813	3.42	3.59
139	3843	3.27	3.27	188	1468	3.20	3.44
50	2019	3.67	2.94	36	3731	4.01	3.33
80	1535	3.66	2.07	83	2647	3.65	3.17
140	1243	2.49	0.72	153	2495	3.17	2.56
Average 150	2897	3.37	3.21	72	2816	3.48	3.52

The methodology is as follows, results obtained in the previous section are used as the baseline for the hybridisation. Reusing the data clarifies what could have happened using PSO or hybridisation. The secondary algorithm starts when a certain threshold in the cost function value is obtained. This value is always below one to ensure that the secondary algorithm starts with a feasible trajectory. The non-exploratory algorithm is given 500 cost function evaluations to find a better route.

It has been decided to use box plots instead of tables to display the results obtained. The box plot shows the median, outliers, and bottom and top edges. The bottom and top edges of the box indicate 25th and 75th percentiles, respectively. It is a condensed method to display summary statistics. In the figures, the respective names have been simplified for simplicity. EA with a penalty cost function, PSO with a penalty cost function, PSO with an adaptive cost function, hybrid PSO-simplex with an adaptive cost function, hybrid PSO-GPS with adaptive cost function have been translated to Ev Pc, PS PC, PS AC, PS-S AC, PS-GPS AC, respectively.

The objective is to find a hybrid algorithm that can outperform $\Delta \overline{P}_R$ with a significantly lower number of iterations. Figure 5.6 summarises the statistics related to the number of iterations. PS-S (PSO plus simplex) and PS-GPS (PSO plus GPS) have no rival in the number of iterations achieved. These two combinations of algorithms and cost functions have box edges below the previous bottom edges. It can be considered that the objective of finding a fast optimisation approach is achieved using either PS-S or PS-GPS.

The improvement in the average received power ($\Delta \overline{P}_R$) is condensed in Figure 5.7. The study in Figure 5.7a shows that both hybridisation techniques perform worst than PSO with the adaptive or penalty cost function. Figure 5.7b shows similar performance for all the optimisation approaches. The study of figures 5.7c and 5.7d gives an advantage to the PS-GPS with an adaptive cost function approach. $\Delta \overline{P}_R$ achieved by the later hybridisation approach gives virtually no reason to choose PSO.

5.5   Surrogate optimisation

Although arguments for selecting an optimisation approach might be clear, a new candidate has been identified. Tables 5.1-5.4 indicate that the surrogate approach might be suitable. This approach was not selected for references [4] and [5] as the focus was on improving EA and PSO. These two optimisation approaches were the
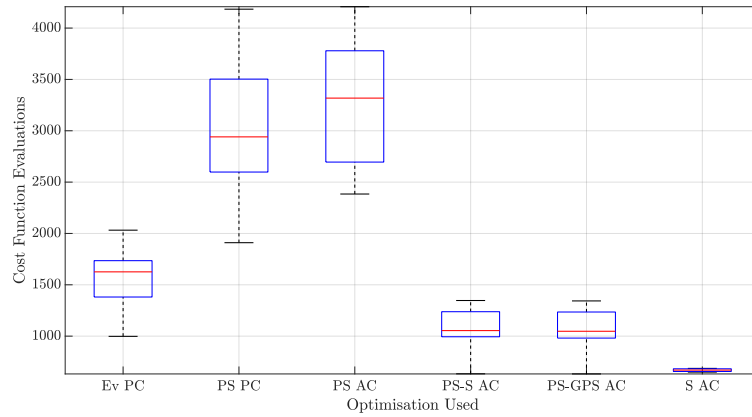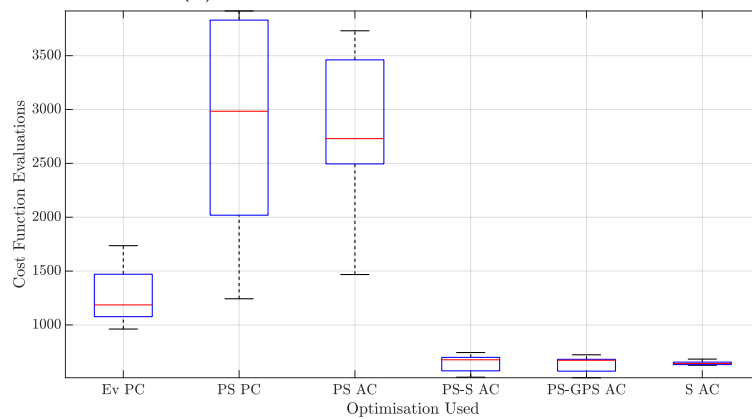
(A) Number of iterations for Scenario 1
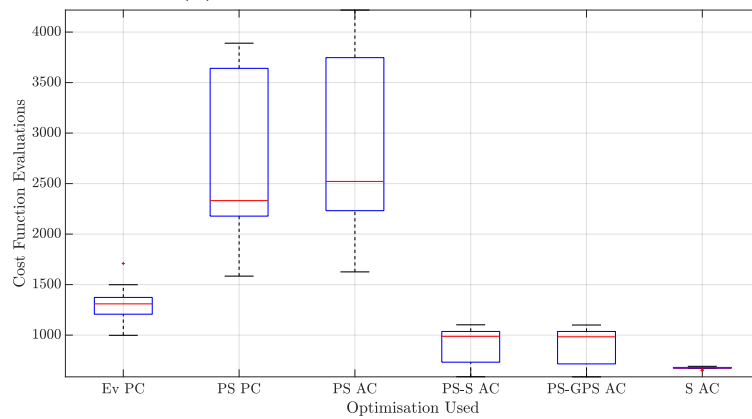
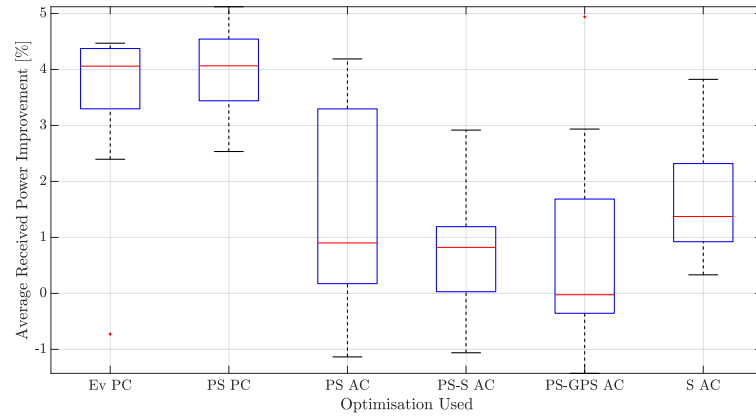(B) Number of iterations for Scenario 2
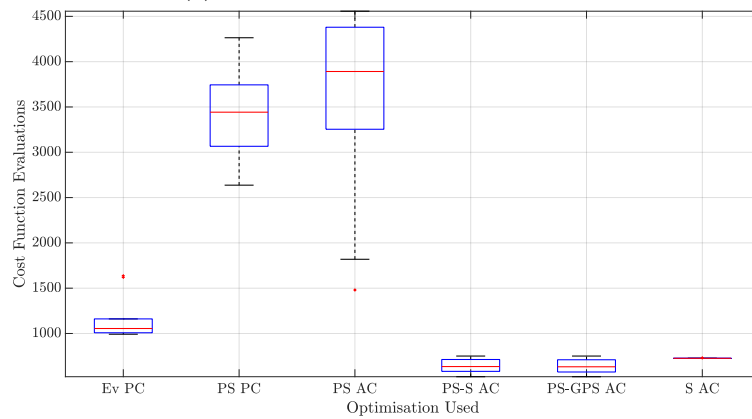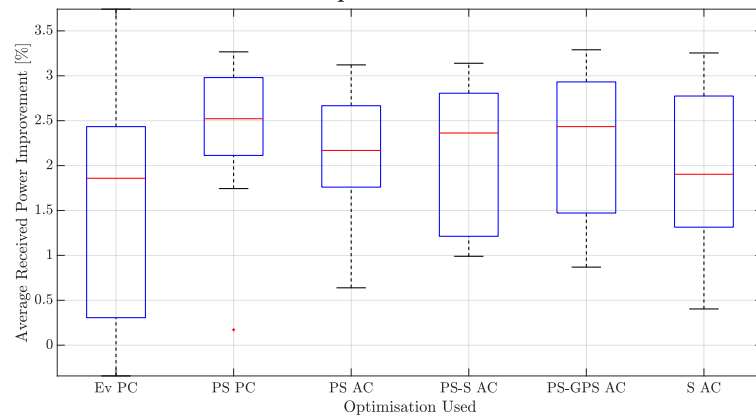

(C) Number of iterations for Scenario 3

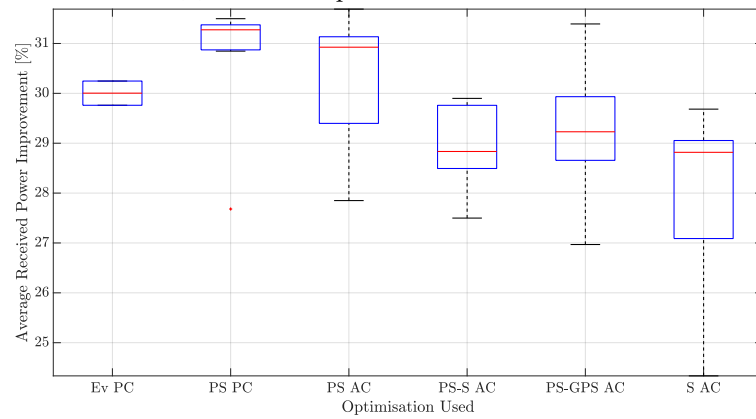(D) Number of iterations for Scenario 4

FIGURE 5.6: Number of iterations required by each optimisation strategy on all four scenarios.
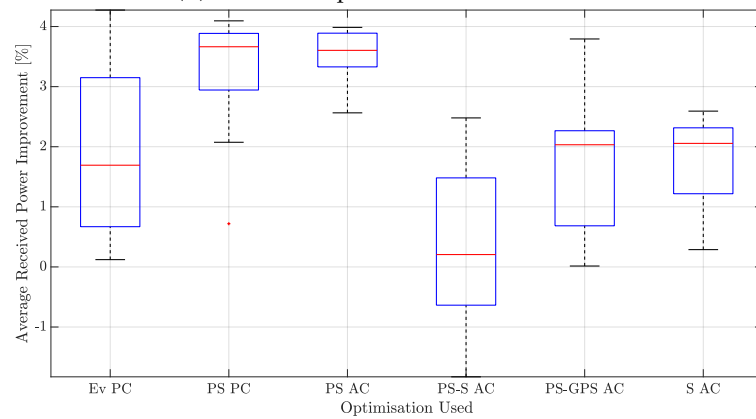
(A) Channel improvement for Scenario 1

(B) Channel improvement for Scenario 2

(C) Channel improvement for Scenario 3

(D) Channel improvement for Scenario 4

FIGURE 5.7: Channel improvement achieved by each optimisation strategy on all four scenarios.

best candidates according to initial results. Further study of the surrogate algorithm in Section 4.4.1 highlights one key characteristic. This characteristic is the generation of a pseudo-search space based on initial trial points. Each trial updates the pseudo-search space, and a minimum is searched based on it.

The number of iterations required to achieve a solution is notably low for scenario 1. The number of iterations required is low enough not to be seen by the scale of Figure 5.6a. Similar behaviour is obtained in 5.6b, where this number is below the performance of PS-S and PS-GPS. Scenarios 3 (Figure 5.6c) and 4 (Figure 5.6d) exhibit no advantage by choosing surrogate optimisation over the hybrid approach.

Improvement of the average received power achieved by the surrogate optimisation show different tendencies for each scenario, as seen in Figure 5.7. In scenario 1, the surrogate performs better than the proposed hybridisation techniques. The results obtained in scenario 2 are almost equal for the applied optimisation techniques. Scenarios three and four give the advantage to PS-GPS AC.

5.6 Selected optimisation approach

The first set of conclusions can be drawn from the analysed results. The EA optimiser cannot be considered a reliable algorithm as it does not find suitable routes for all scenarios and simulations. Although, in some scenarios, this optimiser could find a suitable route, one of the requisites for selecting an optimisation approach is to find a reliable one.

The low reliability of the EA motivated the switch to PSO. Using the penalty cost function, this algorithm can find a compliant route with restrictions and objectives for all simulations and scenarios. However, the reliability increases with this new algorithm, and the number of iterations also increases. Since the new optimiser gives reliability, a better algorithm aims to reduce the required number of cost function evaluations. The cost function's study proved its disadvantageous design; the optimiser can be trapped in the plateau. The new cost function should give more information to the optimiser about where to best direct the efforts toward finding a solution.

The adaptive cost function is the answer to such a problem. The logarithmic part of Equation 4.12 gives a continuous and ever-decreasing value towards compliance with restrictions. PSO using the adaptive cost function shows similar performance in the total number of cost function evaluations. A closer look at the data displayed a significant reduction on $N_{g_p[a,r]<1}$. Having routes compliant with restrictions early indicates that hybridisation can be a good option.

A different improvement path is to try a different optimisation technique. Surrogate has shown promising results in early tests, providing restrictions-compliant routes in lower times. Four scenarios and simulations prove that surrogate can keep up with the time expectation. $\Delta \overline{P}_R$ achieved by surrogate is not as good as the other candidates. It might not be as advantageous as it seemed in an early stage.

Nelder-mead simplex and GPS are the chosen algorithms for hybridisation. Out of the two, GPS is the optimisation technique that achieves better results whilst keeping cost function evaluations under reasonable values. Using a hybrid approach can reduce the cost function evaluations by 10. Not only the number of evaluations to convergence is critical, but also the improvement on $\Delta \overline{P}_R$. PSO combined with GPS always provides a flight route with the maximum connectivity in any scenarios investigated.

References

[1] *Google Maps, Aerial view of the Munich area*. `https://cutt.ly/1jnLnyu`. Accessed: 2021-02-23.

[2] *Google Maps, Aerial view of the Alps area*. `https://cutt.ly/KjnLWuw`. Accessed: 2021-02-23.

[3] *Google Maps, Aerial view of the Alicante area*. `https://cutt.ly/DjnLvqA`. Accessed: 2021-02-23.

[4] Adrian Exposito, Dominic Schupke, and Hector Esteban. "Route Optimisation for Maximum Air to Ground Channel Quality". In: *IEEE Access* 8 (2020), pp. 203619–203630.

[5] Adrián Expósito García, Héctor Esteban González, and Dominic Schupke. "Hybrid Route Optimisation for Maximum Air to Ground Channel Quality". In: *Journal of Intelligent & Robotic Systems* 105.2 (2022), pp. 1–16.

Chapter 6

Conclusions

The first chapter of this work defined a series of objectives that have steered the research involved and the narrative of it. It is now the moment to go back individually to address those objectives and justify the degree of compliance to each one of them. The justification of objectives is not the only topic to be discussed in this chapter. A summary of the research performed follows where the connection between each chapter is highlighted and the developed building blocks are enumerated one last time.

As in every research work, it must criticise itself and acknowledge the flaws on it. For that, there is a section dedicated to the limitations found whilst performing the necessary research. Furthermore, recommendations on how to circumvent such limitations are provided. With both elements, the reader is provided with the necessary tools to build on the research already made in this work.

The novelties on this research are also highlighted and contained within its own section. The main objective for this section is to provide the reader with an holistic view of the custom building blocks. Although the majority of them have been previously described, having them under the same umbrella helps bringing closure to this work.

Having addressed the objectives, limitations and innovations, this work ends with an end note. The purpose of this note is many-fold. First, this note positions this research within the existing body of knowledge. Second objective is to address existing gaps within the various fields of research that have been addressed. Commenting on the novelties of this research forms the third and last objective.

6.1 Completion of defined objectives

Back in the introduction, this work defined the main three objectives to be completed. Prior to enumerate said objectives, the meaning of completion should be addressed. An objective is considered complete when not only a solution is proposed but also the implementation of necessary interfaces and the solution itself. Presenting, implementing, testing and analysing the roots of concepts is the path that has lead to the completion of this work.

6.1.1 Tools for a successful optimisation

The first objective was to define the necessary tools for a successful optimisation. This objective corresponds to the definition of the first part of Figure 6.1; mission modelling. Chapter 2 is solely devoted to the completion of this first objective. Any simulation needs an environment, an scenario where to position all possible static objects influencing the channel's response, or the optimisation. This scenario is composed by three layers: ground profile, terrain type and man-made structures. Data
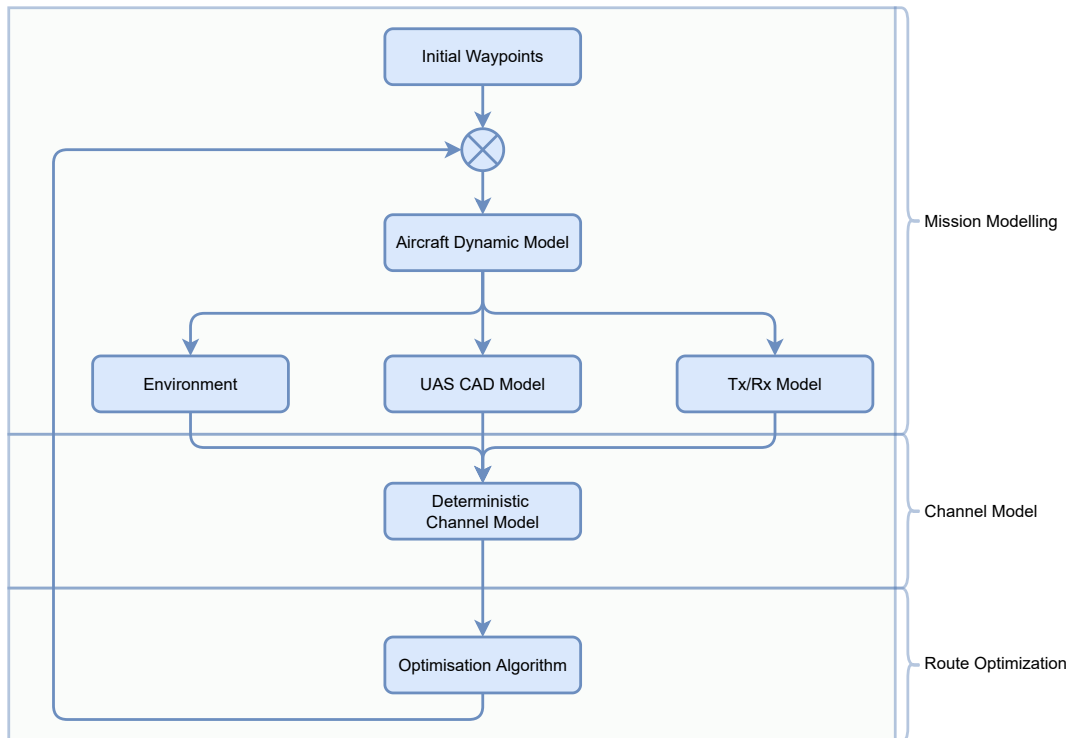
FIGURE 6.1: Optimisation building blocks

sources and calculation methods have been provided for the use of each one of the three layers.

Next block in first third of Figure 6.1 is the unmanned aerial system (UAS) CAD model, or aircraft model. Being able to calculate frame obstructions requires knowledge from the shape of the UAS. A comparison is provided between the first (Figure 2.14) and the final (Figure 2.15) model. This is to display the incremental approach that has been followed in this work. How to determine frame obstructions is provided along with a more advance method: body masks. The later provides more agility to the optimiser, a key aspect to reduce optimisation times.

Next point related to the UAS, decoupled from its CAD model but not from its shape, its the dynamics of it. There are many methods to simulate the dynamics: implementation of the full equations of motion, simplified equations of motion, or make use of a flight simulator. The choice made is to make use of a flight simulator. The implementation or the equations of motion, no matter if simplified, have been and are a complex subject. In the interest of simplification and accuracy, a flight simulator has been chosen for the task. Analytical Graphics Inc. Systems Tool Kit (AGI-STK) is the selected simulator to obtain the UAS position and behaviour. Through this simulator it is possible to know the position, attitude and speed of the UAS at any given time.

Naturally, the UAS needs a set of waypoints that define the flight path. The azimuth elevation range (AER) method is chosen to determine how each waypoint relates to the next one. This approach reduces the search space, but any change on a waypoint alters the complete route. As waypoints are the main design parameters used by the optimiser, low and upper boundaries are required and provided.

There is a missing block in Figure 6.1 that has not been addressed yet. This block "Tx/Rx Model" is described in the communication systems subsection. The ground network (GN) and UAS radiation pattern are provided. The GN pattern is composed

of various radiating elements to simulate a base station favouring ground over aerial users. The design matches with the assumption that network providers will not give priority to aerial over ground users. UAS' radiation pattern is a dipole operating at 1.9 GHz.

As in the objective description, there is no single tool but a set of tools required. Such tools have been carefully chosen to provide the necessary data to the channel model. This work can be then considered compliant with the first objective. The channel model can now make use of the environment information, UAS model, communications system and dynamics of the UAS to estimate the received power.

6.1.2 Deterministic channel model

The second objective was to make use of all the generated information and translated into received power, or link quality. This activity corresponds to the second third of Figure 6.1. In Chapter 3, the fundamentals of radio wave transmission are given. It follows the study of existing literature in channel simulation, generation and validation.

By the study of existing literature and current research trends, it has been decided that a deterministic channel model is the best candidate to suit our needs. This channel and its formulation can make use of data like the position of the UAS, environment and terrain profile. In order to elaborate further in the set of equations describing the channel model, each component has been placed to account for specific propagation mechanisms. Equation 3.32 is the global expression used in the channel including slow variations and fast variations. The two parts are paramount to capture the variations on the channel, as displayed in Figure 3.10. Figure in which the effect of using only slow variations is compared with the use of the full equation. With only slow variations, the channel has an homogeneous evolution through the terrain. When the full equation is enabled, ground coverage presents a more heterogeneous and unpredictable behaviour.

Influence originating from type of terrain and height profile is modelled in Equation 3.30. The first part of the equation models influence from the reflection point. Terrain permittivity and roughness of it are used to model this effect. The second part of the equation focuses on the position and speed of the UAS. Equation 3.32 incorporates the variation of the relative angles between radiating partners in the slow variations section. This can be seen in parameters G_R and G_T, dependent on α and β.

The compliance to this objective is proven with the use of Figure 3.10. The figure compares coverage obtained using only path loss and the proposed channel. The many ripples that can be found in the figure matches with the terrain variation. Thus far, it is possible to translate a set of waypoints to received power throughout a route.

6.1.3 Route optimisation

Without underestimating the importance of previous objectives, the choice for an optimiser is the objective that uses more space in this work. Once an algorithm is chosen, it will be used in the block "route optimisation", part of the last third of Figure 6.1. What is required is an algorithm, or a combination of them able to propose new routes based on the performance of previous ones. There is no time limit imposed for a successful optimiser, nevertheless, as several algorithms are compared, the fastest with best results will be the chosen one.

Chapter 4 is entirely devoted to the description of all the necessary means for the optimisation. The description naturally includes which optimisation approaches are candidates, but also cost function, restrictions and objectives. The objectives and restrictions to be used are independent of the selected optimisation method. Objectives are functions or mathematical expressions that guide the optimiser towards global minima. They can adopt any value and are generally designed to be bounded between zero and one. Restrictions model those cases where a given solution, although mathematically feasible, it renders such solution unusable. This function ingest design parameters (waypoints) and outputs a score assigned to the route described by the waypoints.

Candidate cost functions are the so-called penalty method and the adaptive barrier method. Both are supposed to help or guide the optimiser algorithm towards a global minima. Penalty methods modify the cost function directly. The main characteristic of the penalty method is that the penalty should be null when all restrictions are satisfied. Effort is required in balancing the exploratory phase through the infeasible region and not being trapped in such a region. A newer, more refined cost function is presented; the adaptive barrier function. The main similarity with the penalty method is converting constrained optimisation to an unconstrained one. Objectives and restrictions are integrated into the same function, achieving a continuous function with no abrupt changes like in the penalthy method.

The mentioned cost functions can be applied to any proposed optimisation algorithms. Out of the described methods in Chapter 4, only evolutionary algorithm (EA), particle swarm optimisation (PSO), annealing, surrogate, generalized pattern search (GPS), simplex have been tested. The results of these tests can be found in Chapter 5. Each algorithm is subjected to several scenarios to verify their behaviour under different situations. Through the different scenarios, the ability of the optimisation algorithms to avoid terrain, no fly zones (NFZs) and overcome different terrain compositions is certified.

As a result of analysis of results, the best is to apply a hybrid approach. The algorithm should be composed of an exploratory phased using PSO, followed by GPS, both using the adaptive penalty cost function. This will guarantee that the solution space is efficiently explored by the first algorithm. Once a suitable solution is found, the information is passed to GPS, an algorithm that focuses on finding a solution with fewer cost function calls.

The third an last objective is considered closed with the proposal of an optimisation strategy. This strategy has been found to be the fastest, in terms of cost function calls. Not only that but it provides routes compliant with all restrictions and best optimisation of objectives. In this particular sense, the hybrid approach is the one that provides routes that better maximise air to ground (A2G) channel performance.

To provide a summary to this section; the three objectives proposed at the beginning of this work have been greatly satisfied. A set of methods and tools have been proposed to include all necessary information from the environment, UAS and GN. This information is then used by the proposed channel model to determine the A2G channel performance through the route. All the information is then used by the proposed optimiser, hence, making it able to search and efficiently find a route improving channel performance under minimum iterations.

6.2   Limitations and recommendations

Even though objectives have been greatly satisfied, this does not mean that there is no room for improvement and that this work is flawless. Through the statement of the limitations of this work, a set of measures can be established. This set of measures is traditionally known as future work. Said measures can be considered as recommendations for future work, a point from where this work can be picked up and further improved.

Limitations are firstly presented for Chapter 2 so the order of previous section is followed. Terrain profile makes use of state of the art data obtained from the TanDEM-X mission. The density of points can be increased, the accuracy of them too. Nevertheless, this improvement affects calculation time and the available sources are not available without costs. One aspect subject of improvement is the source of terrain composition. OpenStreeMap (OSM) provides highly detailed maps, type of ground and their limits. Although this approach is the first candidate to improve current map details, there are some considerations to be made. First of it, OSM relies on its users to provide information to populate their maps. Second, the number of tags present for each type of map requires filtering and understanding of their data structures. Third and last point connects OSM with the channel model, new expressions are required to simulate influence of objects like treas and even leaves from treas in case of forested areas.

A single limitation has been identified within the UAS model. This limitation relates to how frame obstruction of line of sight (LOS) is calculated. In this work, it is considered that if the direct ray is obstructed by the frame, there is no link between UAS and GN. One way of improvement is to condition performance degradation to the Fresnel zones and the respective distance to them. A deeper literature research and even field measurements should be considered to better characterise performance loss due to frame blocking the LOS.

Chapter 3, dedicated to the channel model, provides some ideas on possible recommendations to overcome the limitations of it. As seen in the literature review provided in said chapter, many authors have tried to get as much data as possible and then develop models fitting the data. A set of selected routes should be flown and the data compared with the simulated one. Although literature suggest that the two-ray model is sufficient to characterise the environment, the assumption should be validated with real-world data. Previous recommendations like improvement of terrain data and obstructions from the UAS frame should also be reflected in the formulation of the channel model.

Optimisation methods described in Chapter 4 have one main limitation. This limitation is the use of algorithms provided by Matlab. To achieve better results, new implementations of the described algorithms should be tested. Custom implementations of the genetic algorithm or PSO have shown good results in the studied literature. Another path of improvement could be to thoroughly study the algorithm characteristics and identify code areas than can be improved, or non-identified parameters playing an unnoticed role.

The verification of the overall results is a path of improvement. This path can also be complemented with the design of new scenarios that could further test the capabilities of the optimisation approach. If the above changes are finally implemented into the channel model and the optimisation algorithms are further improved, new results on the scenarios should be obtained. If the new algorithms show no problem by tackling current scenarios, new challenges can be added. The main challenge will be to tackle varying temporal conditions such as air traffic. Once the density of

UAS increases, this is a variable the needs careful consideration to avoid incidents between two flying objects.

6.3   Originality of this work

The originality of this work has been well accredited by both academia and the industry. Proof of that are the multiple publications using the described concepts and the filled patent. The originality of this work is described based on said publications, from the ones using less parts of this work to the publications benefiting the most from this work.

The work performed in [1] makes use of the concepts developed in Chapter 2. More precisely, sections 2.2 is directly applied in the mesh network optimisations. This work does not consist exclusively on a set of unconnected boxes doing separate functions, a complete framework for simulations have been developed. This framework was used to obtain results presented in [1]. At the time the work was developed, the deterministic channel model was not in an advance stage, so the conventional path-loss equation was preferred.

Reference [2] documents the earliest use of the deterministic channel described in Chapter 3. The main concept behind Reference [2] is to prove that the channel variations can be exploited to obtain key material and secure communications between two aerial system (AS). Although exploiting sources of randomness to secure communications is not a new concept, applying it to air to air (A2A) communications is. Apart from the channel model, the whole simulation framework is used. The ideas presented in Chapter 2 are applied to support channel estimation. As the end application is physical layer security, optimisation is not part of this publication.

Both references [3] and [4] cover similar aspects. They share a common objective which is to find the most suitable optimisation algorithm, equal to one of the already discussed objectives. The first Reference [3] covers with details chapters 2 and 3 but it only showcases performance for one optimisation approach. This approach is composed by the penalty cost function and EA optimisation. The intention to improve the selected optimisation approach lead to [4]. Contrary to the first reference, this one does not focus on previous chapters and develops further possible alternative optimisation approaches.

Perhaps the best evidence of originality of this work is Reference [5]. It refers to the patent originated through this work. The patent protects the developed intellectual property. Moreover, it ensures that the proposed ideas have not been protected before neither by another patent nor by scientific publication. The main idea subject of protection comprises an optimization of the trajectory based on a simulation model of the UAS. The main originality is not the channel model, nor the optimisation algorithms, nor the dynamic simulation of the UAS. The enumerated methods have been developed before as shown in the literature and it won't be right to claim ownership over them. To conclude, the original idea of this work is the method to combine all of them, develop the necessary tools, interfaces, and achieve a workflow that outputs an optimised route.

6.4   End note

This note provides the reader with a final contextualisation of this work with respect to the body of knowledge. With respect to that, every reader that got to this point have realised that the uses of this work are multiple. It can be used as it is to continue

the research on route optimisation for maximum channel performance. Another option for the reader is to take the chapters as separate entities and study the necessary building blocks for a flight simulation in a realistic environment. The reader can use the terrain sources, the maps, methods on how to use the frame in simulations and the proposed radiation patterns. The second chapter provides a view on the necessary expressions to formulate a deterministic channel model. Last but not least, many optimisation methods have been discussed with the associated results.

Briefly speaking, if the individual pieces are studied, the reader might think that novelty is not enough. It is with the big picture and the depiction of all necessary pieces where the novelty arises. Moreover, and taking the channel model as an example, various sources comment on using sort of deterministic channel models but fail in delivering sufficient details. The task in this particular case was to develop and explain with as much detail as possible how the proposed channel model came to be.

A second research gap covered is the lack of data fusion between maps, terrain type, and flight models. Google provides nice-looking maps even with some visualisation of the terrain, but at the same time terrain profile cannot be retrieved. AGI-STK can provide terrain heights for points, but not as a generalised mesh. The methods described close the gap by providing sources of information that can be merged into a single map with all the required information. Software exists to evaluate the perturbations an airframe can introduce in signal propagation, but under static conditions and at high computational cost. The proposed method can be used to evaluate in-flight airframe obstructions with interchangeable models. To the best of the author's knowledge, this work provides an unprecedented level of detail and flexibility.

Last but not least, the proposal of an optimisation approach closes the already identified gap. This gap consisted on a lack of a comprehensive review of optimisation algorithms that could be applied to this kind of optimisation problem. An optimisation problem where dynamics of the UAS, channel performance and challenging environment conditions is mixed.

References

[1]   Sandra Hofmann et al. "Connectivity in the air: Throughput analysis of air-to-ground systems". In: *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE. 2019, pp. 1–6.

[2]   Adrián Expósito García et al. "Evaluation of a wireless physical security method for flying objects based on the frequency selectivity of the propagation channel". In: *2019 15th International Conference on the Design of Reliable Communication Networks (DRCN)*. IEEE. 2019, pp. 59–66.

[3]   Adrian Exposito, Dominic Schupke, and Hector Esteban. "Route Optimisation for Maximum Air to Ground Channel Quality". In: *IEEE Access* 8 (2020), pp. 203619–203630.

[4]   Adrián Expósito García, Héctor Esteban González, and Dominic Schupke. "Hybrid Route Optimisation for Maximum Air to Ground Channel Quality". In: *Journal of Intelligent & Robotic Systems* 105.2 (2022), pp. 1–16.

[5]   Adrian Exposito Garcia and Dominic Schupke. *Method For Operating An Unmanned Aerial Vehicle As Well As An Unmanned Aerial Vehicle*. US Patent App. 16/709,179. 2020.