DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA
ESCUELA SUPERIOR DE INGENIERÍA
UNIVERSIDAD DE SEVILLA

Threat Management for Missions with Multiple Unmanned Aerial Systems operating in the U-space

por

Carlos Capitán Fernández

Ingeniero Industrial

PROPUESTA DE TESIS DOCTORAL
PARA LA OBTENCIÓN DEL TÍTULO DE

DOCTOR POR LA UNIVERSIDAD DE SEVILLA

SEVILLA, 2022

Directores

Dr.-Ing. Ángel Rodríguez Castaño, Profesor Contratado Doctor

Dr.-Ing. Jesús Capitán Fernández, Profesor Titular

UNIVERSIDAD DE SEVILLA

Memoria para optar al grado de Doctor por la Universidad de Sevilla

Autor:	**Carlos Capitán Fernández**
Título:	**Threat Management for Missions with Multiple Unmanned Aerial Systems operating in the U-space**
Departamento:	**Departamento de Ingeniería de Sistemas y Automática**

V° B° Director:

Ángel Rodríguez Castaño

V° B° Director:

Jesús Capitán Fernández

El autor:

Carlos Capitán Fernández

What we do in life...echoes in eternity.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my coadvisor Prof. Jesús Capitán for his continuous support during my Ph.D., and particularly for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me greatly throughout the research and writing phases of this thesis.

Additionally, I would like to thank my coadvisor Prof. Ángel R. Castaño, who was the first to trust me to join the GAUSS project team, which motivates the work in this thesis, and who guided my work throughout the project. Also, I would like to thank Prof. Aníbal Ollero for giving me the opportunity to work in the Group of Robotics, Vision, and Control of the University of Seville.

I thank my fellow labmates in the Group of Robotics, Vision, and Control: Alejandro Braza, Fran Real, Héctor Pérez, Alfonso Alcántara, Ángel Montes, and Manuel Fernández for their support and the time they have devoted to work with me, and for all the fun we have had in the last four years.

Last but not least, I would like to thank my family for supporting me throughout my life.

The work presented in this thesis has received funding from the European Union's Horizon 2020 Research and Innovation Programme under grant agreement No 776293 (GAUSS project) and No 955269 (OMICRON project).

Abstract

The number of Unmanned Aircraft Systems (UAS) operating in civil airspace is growing quite fast, mainly due to their recent technological advances and the wide spectrum of current applications for this type of vehicles. However, this increasing number of UAS flying together in a common urban airspace is also a challenge, as they need to be integrated with manned aircraft in a safe and automated manner. There exist initiatives for UAS Traffic Management (UTM) in the Very Low Level (VLL) airspace, i.e., below 150 meter of altitude. In particular, this thesis focuses on the U-space, which is a European ecosystem consisting of a set of new services relying on a high level of digitalisation and automation of functions and specific procedures designed to support safe, efficient and secure access to airspace for large numbers of UAS.

This thesis addresses the problem of threat management in missions with multiple UAS operating in the U-space. This means the development of procedures to react to dangerous situations caused by unexpected events or conflicts during operation. We analyze the most common types of threats in the U-space and how to select mitigation actions to minimize risks and resolve conflicting flight plans. For that, we take three steps. First, we analyze the European regulatory framework and the U-space definitions, in order to understand better the requirements of methodologies for operation risk assessment and threat management in multi-UAS scenarios. Second, we present a UTM system architecture for U-space services, focusing on the implementation of those related with real-time threat management during operation. Nonetheless, the architecture is modular and flexible enough to be extended with more U-space services and functionalities in future implementations. Third, built upon this software architecture, we develop a methodology for autonomous decision-making to handle unexpected events in U-space operations with multiple UAS. The method is

capable of handling a list of usual threats in UTM systems, and provide optimal mitigation actions in terms of cost and risk level. Finally, it is important to remark that all the results in the thesis have been validated through a realistic *Hardware-In-The-Loop* simulation environment and by means of field experiments with actual UAS. Moreover, all the software produced has been published as open source for the UAS community.

Acronyms

ADS-B - Automatic Dependent Surveillance Broadcast
AEC - Airspace Encounter Category
ARC - Air Risk Class
ATLAS - Air Traffic Laboratory for Advanced Systems
ATC - Air Traffic Control
ATM - Air Traffic Management
BVLOS - Beyond Visual Line Of Sight
ConOps - Concept of Operations
CV - Contingency Volume
DAA - Detect And Avoid
DB - Data based
DDS - Data Distribution Service
EASA - European Union Aviation Safety Agency
EM - Emergency Management
EU - European Union
FG - Flight Geometry
FOCA - Federal Office of Civil Aviation
FTA - Fault Tree Analysis
GAUSS - Galileo-EGNOS as an Asset for UTM Safety and Security
GCS - Ground Control Station
GNSS - Global Navigation Satellite System
GRC - Ground Risk Category
GUI - Graphical User Interface

GUTMA - Global UTM Association
HITL - Hardware In The Loop
ICAO - International Civil Aviation Organization
JARUS - Joint Authorities for Rulemaking on Unmanned Systems
JSON - JavaScript Object Notation
LiDAR - Light Detection And Ranging
M2M - Machine To Machine
MQTT - Message Queuing Telemetry transport
MULTICOP - Autonomous multi-aerial systems for cooperative maneuvers with physical interaction
MULTIDRONE - MULTIple DRONE platform for media production
NAA - National Aviation Authority
NASA - National Aeronautics and Space Administration
NTP - Network Time Protocol
OV - Operational Volume
OSO - Operational Safety Objectives
QoS - Quality of Service
ROS - Robotic Operating System
RPAS - Remotely Piloted Aircraft System
RPS - Remote Pilot Station
RTPS - Real Time Publish-Suscribe
SAA - See And Avoid
SAIL - Specific Assurance and Integrity Level
SESAR - Single European Sky ATM Research
SITL - Software In The Loop
SORA - Specific Operational Risk Assessment
STS - Standard Scenario
TCAS - Traffic Collision Avoidance System
TD - Tactical Deconfliction
TMPR - Tactical Mitigation Performance Requirement
UAS - Unmanned Aircraft Systems

USM - U-space Service Manager
USP - U-space Service Provider
UTM - UAS Traffic Management
VLL - Very Low Level
VLOS - Visual Line Of Sight
VTOL - Vertical Take-Off and Landing
WP - Waypoint

Contents

Acknowledgments **vii**

Abstract **ix**

Acronyms **xi**

1 Introduction **1**

1.1 Motivation 1

1.2 Thesis objectives 3

1.3 Thesis contributions 4

1.4 Thesis framework 7

2 Background **9**

2.1 Regulatory framework 9

2.2 U-space 11

2.2.1 U-space functional architecture 15

2.2.2 Operational volume in U-space 16

3 Risk Assessment based on SORA Methodology for a UAS Media Production Application **19**

3.1 Introduction 20

3.2 Related work 22

3.3 UAS operation for media production 23

3.4 Overview of the SORA methodology 25

3.5 Risk assessment for media production 26
3.5.1 Pre-application evaluation 26
3.5.2 Ground risk process 28
3.5.3 Air risk process 30
3.5.4 Final SAIL and Operational Safety Objectives assignment 33
3.6 Discussion 35
3.6.1 Updates with SORA version 2.0 37
3.7 Conclusions 38

4 Unmanned Aerial Traffic Management System Architecture for U-space In-flight Services 41
4.1 Introduction 42
4.2 Related work 43
4.3 Design framework 47
4.3.1 Guidelines for system design 47
4.3.2 Robot Operating System 48
4.4 UTM system architecture 50
4.4.1 U-space Service Manager 51
4.4.2 Data Base 52
4.4.3 Tracking 53
4.4.4 Monitoring 55
4.4.5 Emergency Management 57
4.4.6 Tactical Deconfliction 58
4.4.7 Discussion 60
4.5 Experiments 61
4.5.1 Use cases definition 62
4.5.2 Experimental setup 64
4.5.3 Results 66
4.6 Conclusions 69

5 Threat Management Methodology for UAS operating in the U-space 71
5.1 Related work 72

5.2 Methodology for threat management 75
  5.2.1 Overview 75
  5.2.2 Threat types 77
  5.2.3 Decision-making procedure 78
5.3 Experimental results 85
  5.3.1 Definition of the use cases 85
  5.3.2 Experimental setup 89
  5.3.3 Results 91
5.4 Conclusions 97

6 Conclusions and Future Work **99**
6.1 Conclusions 99
6.2 Future work 101

References **103**

List of Figures

1.1 The team with fixed- and rotary-wing UAS used to validate the approach presented in this thesis (a) and the ATLAS test facilities located in Spain (b). 2

1.2 Autonomous aircraft performing outdoor media production within the context of MULTIDRONE project. Left, a cycling race; right, a boat race. . 8

2.1 Implementation roadmap for the U-space initiative (SESAR, 2017), consisting of 4 deployment phases. 12

2.2 Functional overview of actors in U-space operations. The USP gathers information from UAS in the airspace and provides U-space services to UAS operators and authorities. A human pilot is placed for safety in the RPS. 15

2.3 Graphical representation of the Operational Volume. The inner volume is the Flight Geometry, whereas the outer is the Contingency Volume. 17

3.1 Example UAS operation for media production. UAS should fly over non-densely populated areas to film a rowing race. The UAS are not expected to fly over crowded areas. 23

3.2 Scheme of the process to apply the SORA methodology. There are several steps to evaluate ground and air risks and estimate a SAIL (JARUS, 2018). . 27

3.3 SORA process to determine the ARC based on the AEC. The blue thick arrows indicate the path for the evaluated operation. 30

4.1 Overview of the U-space ecosystem proposed by EASA (European Aviation Safety Agency (EASA), 2020). The UTM system offers U-space services to the different actors and runs on a remote server called U-space Service Provider Platform. 42
4.2 Overview of the proposed UTM system architecture. This system would be running in a remote server named U-space Service Provider Platform. The red arrows indicate remote communication links with other machines in the ecosystem. 50
4.3 Scheme with the internal components of the Tracking module. The *data association* component matches the measurements from the UAS with their tracks, to update the corresponding Kalman filters. The future UAS trajectories are predicted using the tracks and the flight plans. 54
4.4 Graphical representation of the OV of a UAS operation (the orange arrow represents its radius). Given a flight plan, the green cylindrical volume around represents its Flight Geometry (the green arrow indicates its radius), and the OV also includes an outer volume that is the Contingency Volume. 56
4.5 A 4D grid representation of the airspace. The dark grey cells would be the neighboring cells of the black cell (Acevedo et al., 2020). 57
4.6 Iterative procedure to solve a conflict in the case of a loss of separation (from left to right). The flight plans of the two lower UAS are in conflict and need to be separated. Then, the middle UAS enters in conflict with the upper UAS, so these two get separated again. As the plan of the middle UAS gets modified, the lowest UAS is also adapted to achieve a final solution without loss of separation (Acevedo et al., 2020). 59
4.7 Short caption 62
4.8 Top views including the initial flight plans of the use case 1 (left) and the use case 2 (right). All the operations were planned in an area of the Loring aerodrome in Madrid (Spain). 63
4.9 Setup for the experiments. The computers running the RPS for the two UAS and the UTM system were placed at remote locations and communicated through the Internet via the MQTT protocol. 65

4.10 Screenshots of the graphical user interface developed by SATWAYS running on the RPS Client Application. 65

4.11 Timeline of the experiment of the use case 1, where a loss of separation event is resolved. Single arrows indicate isolated interactions between modules, whereas double arrows indicate periodic communication. 66

4.12 A top (left) and a perspective view (right) of the initial flight plans in the use case 1. The Operational Volumes are shown for both UAS. There are no conflicts given the UAS 4D trajectories. 67

4.13 A perspective view of the conflict resolution in the use case 1. A new flight plan for UAS_1 (with a final *go down* maneuver) was selected to keep the safety distance with UAS_2. The other alternative maneuvers generated by the TD module (*go left* and *go right*) are also shown. 68

4.14 Timeline of the use case 2, where a jamming attack and a geofence conflict are resolved. Single arrows indicate isolated interactions between modules, whereas double arrows indicate periodic communication. 69

4.15 Left, top view with the initial flight plans in the use case 2. The Operational Volumes without conflicts are also shown. Right, situation after the detection of the jamming attack. A geofence (in red) is created around the attacked UAS, which generates a conflict with the flight plan of the other UAS. 70

4.16 A top (left) and a perspective (right) view of the optimal solution in the use case 2. An alternative flight plan for UAS_2 is generated to avoid the geofence. 70

5.1 Left, the team of fixed- and rotary-wing UAS used to demonstrate our approach for threat management. Right, the ATLAS test facilities (Spain) where the experiments were carried out. 72

5.2 Overview of our threat management methodology. 76

5.3 Top (left) and perspective (right) views of the initial flight plans for use case 1. All operations were planned in an area of the ATLAS aerodrome in Villacarrillo (Spain). 86

5.4 Perspective view of the initial flight plan for use case 2. The operation was planned in an area of the ATLAS aerodrome in Villacarrillo (Spain). 87
5.5 Top (left) and perspective (right) views of the initial flight plans for use case 3. All operations were planned in an area of the ATLAS aerodrome in Villacarrillo (Spain). 88
5.6 The Atlantic I (left) and DJI M600 Pro (right) UAS used in the field experiments. 89
5.7 Setup for the experiments. On top, an view of the interfaces between the components run on each computer. The computers running the RPS for the two UAS and the UTM system were communicated through the Internet via the MQTT protocol. At the bottom, pictures of the UAS Ground Control Station (left) and the UTM computer (right). 90
5.8 Screenshots of the Graphical User Interface developed by SATWAYS running on the RPS Client Application. 90
5.9 Timeline of the experiment for use case 1, where a loss of separation is resolved. The events involving the threat management methodology are shown in green. 92
5.10 Resulting flight plan for UAS_1 in use case 1. On the left, the initial flight plan and the alternative solutions to avoid the other UAS (going back home and to the landing spots are not shown for an easier visualization). On the right, the selected solution, its Operational Volume, and the actual trajectory followed by UAS_1. 93
5.11 Timeline of the experiment for use case 2, where an alert warning (wildfire notification) and a geofence intrusion are resolved. The events involving the threat management methodology are shown in green. 94
5.12 Resulting flight plan for UAS_3 in use case 2. On the left, the initial flight plan and the alternative solutions to land on different spots. On the right, the selected solution to the closest landing spot, its Operational Volume, and the actual trajectory followed by UAS_3. In red, the geofence which the UAS is getting out of. 95

5.13 Timeline of the experiment for use case 3, where a jamming attack and a geofence conflict are resolved. The events involving the threat management methodology are shown in green. 96
5.14 Resulting flight plan for UAS_3 in use case 3. On the left, the initial flight plan and the alternative solution to avoid the geofence (red cylinder). On the right, the selected solution turning around the geofence, its Operational Volume, and the actual trajectory followed by UAS_3. 97

Chapter 1

Introduction

This chapter introduces the motivation, main objectives, and scope of this thesis. Then the main contributions are outlined, as well as the research framework in which it has been developed.

1.1 Motivation

In the last few years, there has been a clear trend to use Unmanned Aircraft Systems (UAS), or drones, for many commercial and civil applications. There are reports (SESAR, 2020) that estimate that up to 400,000 drones will be providing services in the airspace by 2050, with a total market value of 10 billion euros per year by 2035. Last-mile delivery (Aurambout et al., 2019), surveillance (Capitan et al., 2016), infrastructure inspection (Sanchez-Cuevas et al., 2020), traffic monitoring (Garcia-Aunon et al., 2019), media production (Alcantara et al., 2020) or health emergency situations management (Kramar, 2020) are just a few examples of the wide spectrum of drone applications. Thus, the integration of UAS in the Very Low Level (VLL) airspace [1] is probably one of the most revolutionary events for Air Traffic Management (ATM) since the beginning of its implementation. Although ATM has been traditionally based on voice communication through an Air Traffic Control (ATC) entity, its bounded workload and communication capacities turn this centralized resource into a bottleneck for system scalability (Janzen, 2019). Therefore, the rise of UAS

---

[1] The VLL airspace is the space below 150 meters and above ground level.

operations brings the need for a new paradigm for airspace management, where digital communication will play a key role and the responsibilities will be shared among different stakeholders, instead of a single central entity.

There are already some initiatives to integrate UAS into the VLL airspace and fulfill their operational requirements. The National Aeronautics and Space Administration (NASA) has created the concept UAS Traffic Management (UTM) (Kopardekar, 2015) to enable safe and large-scale operations with UAS in the low-altitude airspace (Kopardekar et al., 2016); whereas Europe has extended this UTM concept by proposing the U-space ecosystem (SESAR, 2017).

Besides, with this increasing number of UAS operating in the airspace, the likelihood of possible mid-air conflicts and threatening events increases dramatically. Therefore, the current regulation and the methodologies for UAS management should be prepared to prevent and mitigate these dangerous situations. In particular, the Joint Authorities for Rulemaking on Unmanned Systems (JARUS) [2] is an entity taking care of initiatives in this sense. For instance, JARUS has recently published a methodology named Specific Operational Risk Assessment (SORA) (JARUS, 2019), which allows the evaluation of potential risks for UAS missions before operation as well as the proposal of possible mitigation actions. Nonetheless, unexpected events or threats might still occur once the operation has started, leading to hazardous situations in the airspace.

(a) (b)

Figure 1.1: The team with fixed- and rotary-wing UAS used to validate the approach presented in this thesis (a) and the ATLAS test facilities located in Spain (b).

---

[2] http://jarus-rpas.org.

The purpose of this thesis is twofold: (i) to propose novel methodologies for automated threat management and conflict resolution in multi-UAS operations; and (ii) to implement them through a common framework integrated into the U-space initiative. This work has been developed within the context of the European Union (EU) project GAUSS [3], whose main objective was leveraging high-performance positioning functionalities provided by the Galileo ecosystem for U-space operations, including a validation phase performed in a segregated airspace [4]. Figure 1.1 shows the UAS used to validate the methods developed in this thesis at the ATLAS facilities located in Spain.

1.2 Thesis objectives

The main objective of this thesis is to establish mechanisms for the management of threatening situations in missions with multiple UAS operating in the U-space framework. For this purpose, we defined the following goals:

- **Analyzing the current EU regulatory and U-space framework to integrate our methods for managing multi-UAS operations.** The final objective is to develop approaches that can be potentially integrated in the actual European airspace. For that, we will study the requirements of EU regulation in terms of risk assessment and explore the possibilities provided in the U-space for threat management.
- **Building a framework for Unmanned Aerial Traffic Management in the U-space**. This framework should be able to manage multi-UAS operations in a common airspace by means of services provided by U-space stakeholders. The system is aimed to be flexible, modular, and digital, in order to work in an automated fashion. Moreover, it should be able to deal with fixed- and rotary-wing UAS, as well as 4D (x,y,z, and time) conflict resolution. Such a framework will be the base to implement and integrate specific methodologies for efficient and safe traffic management in multi-UAS missions in the U-space.

---

[3] https://cordis.europa.eu/project/id/776293

[4] http://atlascenter.aero/

- **Building a methodology for automated threat management during U-space operations**. This implies an approach integrated in the U-space that evaluates potential threats due to unexpected events occurring during the operation of one or multiple UAS, as well as real-time decision-making procedures to propose mitigation actions. This method should be able to identify and classify generic threats and handle them taking into account the types of UAS involved, the priority of each operation, the geofences established externally, and so on. Given the dynamic nature of the civil airspace, we pursue methods that can make the best decisions in real time, adapting to varying conditions during flight (e.g., operation delays, UAS failures, or other unexpected events).
- **Validating the developed framework and methodology in realistic conditions**. We aim to develop simulation tools that allow us to test UTM approaches in the U-space ecosystem. This will be essential for the integration and implementation of our methodologies for threat management. Furthermore, new methods can work well in simulation but failing when applied to real systems and scenarios. Thus, a key objective in this thesis is to evaluate our developments, not only in simulation but in field experiments, where the UAS traffic management needs to cope with imperfect communication, onboard computational resources, less controlled dynamic actors, etc. We also aimed to develop open-source resources that are available for the community, as an attempt to foster research on UTM in the U-space.

1.3 Thesis contributions

This thesis makes several contributions in the field of threat management for UAS operating in the U-space. First, in **Chapter 2**, we provide some background on relevant U-space concepts and the current EU regulatory framework. Then we analyze the SORA methodology for risk assessment in UAS operations in the VLL airspace (**Chapter 3**). This methodology has been proposed by EU regulatory bodies for UAS operation approval in certain categories. We describe SORA and explain how to use it for applications with multiple UAS. For that, we carry out a risk assessment of an aerial cinematography operation to be conducted with a small team of UAS. The assessment allows an evaluation of the risk level of the operation, discussing possible mitigation actions to reduce risks when deploying multi-UAS systems. This study paves the way for integrating future UAS platforms for autonomous cinematography into the VLL airspace. Moreover, the discussion will help understand better the complexities of risk assessment for multi-UAS operations. The results of this chapter have been published in:

- Capitan, C., Capitan, J., Castaño, A. R., and Ollero, A. (2019). Risk Assessment based on SORA Methodology for a UAS Media Production Application. *International Conference on Unmanned Aircraft Systems (ICUAS).*

Chapter 4 presents a UTM system architecture implementing the U-space concept. We present a general framework for U-space services, which is modular, flexible, and technology-agnostic; and we describe our specific implementation for a set of core in-flight services dealing with unexpected UAS conflicts during their flight phase. This software framework integrates automated decision-making procedures. Additionally, we show an actual realization of our UTM architecture that is available as open-source software for the community and we demonstrate its capabilities. In order to showcase the correct integration of all our components and services, we define use cases for UAS operations involving all developed functionalities, and we assess our results in terms of performance by running the whole system in a realistic simulation setup for multi-UAS operations. The results of this chapter have produced the following publication:

- Capitan, C., Perez-Leon, H., Capitan, J., Castaño, A., and Ollero, A. (2021). Unmanned Aerial Traffic Management System Architecture for U-Space In-Flight Services. *Applied Sciences*, 11(9).

Chapter 5 presents a new methodology for threat management in multi-UAS operations in the U-space. We identify a generic set of threats that could occur while UAS are flying in the U-space, and then the methodology is based on proposing a set of mitigation actions that are evaluated in terms of cost and risk level, in order to take optimal decisions in an autonomous fashion. Moreover, our approach is flexible enough to accommodate additional threats or mitigation actions in the future. The chapter also includes a demonstration of the functionalities of our method for threat management in real flight tests, integrated within the U-space framework presented in Chapter 4. The results of this chapter have produced the following publications:

- Capitán, C., Castaño, A. R., Capitán, J., and Ollero, A. (2019). A framework to handle threats for UAS operating in the U-space. In *International Workshop on Research, Education and Development on Unmanned Aerial Systems (RED-UAS)*, Cranfield, United Kingdom.
- Capitan, C., Capitan, J., Castaño, A. R., and Ollero, A. (2022). Threat Management Methodology for Unmanned Aerial Systems operating in the U-space. *IEEE Access*.

Lastly, **Chapter 6** summarizes the conclusions of this thesis, and it also discusses directions for future research.

Finally, apart from the aforementioned contributions, throughout his thesis, the author has carried out some collaborations that led to the publications below. Even though these works are not part of the thesis core, they are worth mentioning as additional contributions in: (i) open-source simulation tools for UTM, and (ii) conflict detection and resolution in the U-space context.

- Millan-Romera, J. A., Acevedo, J. J., Perez-Leon, H., Capitan, C., Castaño, A. R., and Ollero, A. (2019). A UTM simulator based on ROS and GAZEBO. In *International Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS)*.
- Acevedo, J. J., Capitán, C., Capitán, J., Castaño, A. R., and Ollero, A. (2020). A Geometrical Approach based on 4D Grids for Conflict Management of Multiple UAVs operating in U-space. In *International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 263–270, Athens, Greece.

1.4 Thesis framework

This thesis has been developed within the framework of several research projects. The core part of the thesis has been carried out within the framework of the GAUSS project [5]. GAUSS (Galileo-EGNOS as an Asset for UTM Safety and Security) is a research project funded by the European Commission (H2020-EU.2.1.6-776293), and its main objective was the development of a solution to integrate and exploit Galileo-EGNOS features for precise and secure UAS positioning in a real UTM system. The GAUSS systems were validated in heterogeneous field trials, including multi-UAS missions with fixed- and rotary-wing aircraft. In particular, most of the work in this thesis was framed within Work package 5 of GAUSS. This work package was devoted to the assessment and development of technologies for extracting value from Galileo-EGNOS in an automated UTM system with UAS coordination capabilities. The specific objectives of Work package 5 were the following:

1. Design of an architecture system for a UTM infrastructure.
2. Design and development of a functional automatic surveillance broadcast system for UAS.
3. Design and development of a UTM functional solution and safe operation methods and tools for emergency management.
4. Design of a prototype of security system for UTM communication.

More specifically, the work in Chapters 4 and 5 was carried out to comply with the objectives 1 and 3 of GAUSS Work package 5.

Additionally, part of the work in this thesis, in particular Chapter 3, has been applied to multi-UAS missions for media production. This application for aerial filming was motivated by the EU-funded (H2020-EU.2.1.1-731667) MULTIDRONE project [6]. MULTIDRONE (Multiple drone platform for media production) was a project devoted to implementing cooperative teams of UAS for filming outdoor sport events (see Figure 1.2), and one of the

---

[5] `https://cordis.europa.eu/project/id/776293`.

[6] `https://multidrone.eu`

Figure 1.2: Autonomous aircraft performing outdoor media production within the context of MULTIDRONE project. Left, a cycling race; right, a boat race.

activities framed within the project was the evaluation of risks for operational approval. This thesis has also been partially supported by the projects MULTICOP (Autonomous multi-aerial systems for cooperative maneuvers with physical interaction), funded by the regional Andalusian government in Spain (FEDER-US-1265072) and OMICRON (Intelligent Road Asset Management Platform) [7] funded by the EU H2020-EU.3.4.

Finally, the author of this thesis did a 3-month stay at the University of South-Eastern Norway in Kongsberg (Norway). This stay was planned within the framework of his thesis to collaborate in the USEPE project [8] (U-space Separation in Europe), which is a project funded by the European Commission (H2020-SESAR-890378) to develop a concept of operations that ensures the safe separation of unmanned aerial vehicles (from other drones and manned aircraft) in a crowded urban airspace. During the stay, algorithms for UAS separation management in highly demanding environments such as cities were studied.

---

[7] `https://omicronproject.eu/`

[8] `https://usepe.eu`

Chapter 2

Background

This chapter introduces the regulatory framework for UAS operating in the European VLL airspace. Then the main concepts of the U-space initiative are presented, together with a roadmap of its implementation and its current state of development.

2.1 Regulatory framework

Integrating UAS into the VLL airspace is challenging as it entails risks from a safety perspective. When UAS are operating in an outdoor scenario, they face relevant risks of air collision (e.g., with other nearby UAS) and ground collision (e.g., with buildings, trees, or existing infrastructure such as electrical lines). Also, UAS may fly over people, becoming a threat in case of battery or communication losses. Hence, it is necessary to identify and analyze hypothetical risks before getting authorization for a UAS operation. The European member states and other stakeholders requested the European Union Aviation Safety Agency (EASA) [1] to develop a common regulatory framework to integrate UAS operations in the European airspace, and EASA developed a proposal for an operation risk-based regulatory framework for all UAS (European Aviation Safety Agency (EASA), 2015, 2017). EASA proposes three main risk-based categories for UAS operations:

- *Open*: The Open category is defined as a category of UAS operation that, considering the risks involved, does not require a prior authorization by the competent authority

---

[1] https://www.easa.europa.eu/

nor a declaration by the UAS operator before the operation takes place. The Open category is the main reference for the majority of leisure drone activities and low-risk commercial activities. This category is in turn subdivided in three sub-categories: A1, when flying over people but not over assemblies of people; A2, when flying close to people; and A3, when flying far from people. Each subcategory comes with its own set of requirements. Therefore, in the Open category, it is important to identify the subcategory of operation to determine which rules apply to the operator, and the type of training the remote pilot needs to undertake. In order to operate in the Open category, the weight of the UAS has to be less than 25 kg.

- *Specific*: The Specific category caters for riskier operations not covered under the Open category. To operate in this category, the UAS operator needs an operational authorization from the National Aviation Authority (NAA) where the operator is registered, unless the operation is covered by a Standard Scenario. A Standard Scenario (STS) means a type of UAS operation in the Specific category, as defined in Appendix I of EU regulation 2019/947, for which a precise list of mitigation measures has been identified in such a way that approval could be granted if the operator declares that she/he will apply those measures during operation. To date two STS have been published, STS-01 and STS-02. STS-01 focuses on “Visual-Line-Of-Sight (VLOS) over a controlled ground area in a populated environment” operation, whereas STS-02 focuses on “Beyond-Visual-Line-Of-Sight (BVLOS) with Airspace Observers over a controlled ground area in a sparsely populated environment”. If your operation is not covered by a STS and does not fall in the Open category, then an operational authorization is needed before starting the operation. The UAS operator would be required to conduct a risk assessment of the intended operation by using the SORA methodology (or an equivalent methodology accepted by the NAA) and submit the risk assessment and all identified means to mitigate risks and comply with the operational safety objectives to the NAA. If the NAA is satisfied with the information the operator provided, the NAA will issue an operation authorization. SORA (JARUS, 2018) is a methodology for risk assessment in UAS operations within the Specific category promoted by JARUS [2], which is an entity that pursues a consensus from various NAAs and stakeholders on a common procedure to identify and qualitatively assess safety risks for UAS operations. Basically, SORA is a step-by-step procedure to evaluate risks that outputs a Specific Assurance and Integrity Level (SAIL) that determines the necessary mitigation actions in order to achieve an acceptable level of risk. More details about SORA will be given in Chapter 3.

- *Certified*: The Certified category caters for the operations with the highest level of risk. Future UAS flights with passengers on board such as the air taxi, for example, will fall into this category. The approach used to ensure the safety of these flights will be very similar to the one used for manned aviation. For this reason, these aircraft will always need to be certified (i.e., have a type certificate and a certificate of airworthiness), the UAS operator will need an air operator approval issued by the competent authority, and the remote pilot is required to hold a pilot license. In the long term, it is expected that the level of automation of UAS will gradually increase up to having fully autonomous UAS without the need for the intervention of a remote pilot. In order to allow operations in the certified category, almost all the aviation regulations will need to be modified, which is a major task.

In this context, three additional EU regulations (European Commission, 2021a,b; European Comission, 2021), which will be applicable from 2023, establish the framework for safe UAS and manned aircraft operations in the U-space. These regulations establish requirements for manned aviation operating in the U-space, requirements for providers of air traffic management/air navigation services, and other air traffic management network functions in the U-space for controlled airspace.

2.2 U-space

U-space can be defined as a set of services and procedures to support safe, efficient, and secure access to the VLL airspace for UAS. Particularly, U-space is a collaborative effort among industry and regulators to enable situational awareness, data exchange,

---

[2] http://jarus-rpas.org.

and digital communication with UAS as well as manned aircraft, ATM service providers, and authorities. There exists a roadmap to deploy U-space services divided into the four phases depicted in Figure 2.1. Each phase proposes a new set of services with increasing complexity and integration level between UAS and manned aircraft, as well as an upgraded version of the existing services in previous phases.

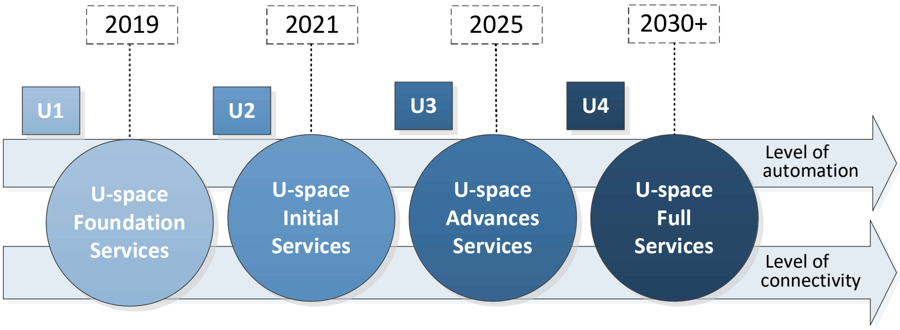

Figure 2.1: Implementation roadmap for the U-space initiative (SESAR, 2017), consisting of 4 deployment phases.

Although the detailed U-space system is being developed, there is already a list of services defined for each deployment phase (Barrado et al., 2020), and a report with the current progress of their implementation and deployment (Eurocontrol, 2020). Table 2.1 depicts these services and their approximate level of implementation in Europe. In short, the defined U-space services are the following:

- *E-registration*: It should manage the registration of the UAS, its owner, its operator, and its pilot. Different types of users may access these data depending on their defined permissions.
- *E-identification*: It should provide information about the UAS and other relevant information to be verified without physical access to the UAS.
- *Pre-tactical geofencing*: It should provide to the UAS operator geo-information about predefined restricted areas during flight preparation.
- *Tactical geofencing*: It should provide geofence and other flight restriction information to UAS pilots and operators for their consultation up to the moment of take-off.

Phase	**Service**	**Overall implementation level**
U1 Foundation Services	E-registration	19 %
	E-identification	17 %
	Pre-tactical geofencing	23 %
U2 Initial Services	Tactical geofencing	13 %
	Flight planning management	6 %
	Weather information	3 %
	Tracking	4 %
	Monitoring	5 %
	Drone aeronautical information management	18 %
	Procedural interface with ATC	20 %
	Emergency management	9 %
	Strategic deconfliction	6 %
U3 Advanced Services	Dynamic geofencing	5 %
	Collaborative interface with ATC	8 %
	Tactical deconfliction	0 %
	Dynamic capacity management	4 %
U4 Full Services	To be defined	0 %

Table 2.1: U-space services for each development phase, together with their approximate implementation level in Europe (Eurocontrol, 2020).

- *Flight planning management*: It should provide assistance to the operator with the building of an operation plan. This service functions as the interface between the UAS operator and the operation plan processing.

- *Weather information*: It should collect and present relevant weather information for the UAS operation, including hyperlocal weather information when available/required.

- *Tracking*: It should receive UAS location reports, fuse data from multiple sources, and provide tracking information about current UAS in operation.

- *Monitoring*: It should provide real-time warnings of non-conformance with the granted flight authorization and inform UAS operators when deviating from their flight plan.

- *Drone aeronautical information management*: It is the equivalent for UAS of the Aeronautical Information Management service. This service should maintain a map of the airspace, including permanent and temporary changes (e.g., a weekend festival would change an area from sparsely to densely populated). It provides information to the geofencing services as well as to the operational planning preparation service.
- *Procedural interface with ATC*: This is a mechanism invoked by the flight planning management service for coordinating the entry of a flight into controlled airspace before flying. Through this, the ATC can either accept or refuse the flight and can describe the requirements and process to be followed by that flight.
- *Emergency management*: It should provide assistance to UAS pilots experiencing an emergency with their UAS, and it communicates recommendation actions to the interested parties.
- *Strategic deconfliction*: It should check for possible conflicts in a specific operation plan, and it proposes solutions during the processing of the operational plan.
- *Dynamic geofencing*: It is an enhancement of geo-awareness that allows geofence changes to be sent to UAS immediately. UAS should have the ability to request, receive, and use geofencing data.
- *Collaborative interface with ATC*: It should offer verbal or textual communication between the remote pilot and the ATC when a UAS is in a controlled area. This service replaces previous ad-hoc solutions and enables flights to receive instructions and clearances in a standard and efficient manner.
- *Tactical deconfliction*: It should check for possible conflicts in real time and issue instructions to aircraft to change their course or speed as needed.
- *Dynamic capacity management*: It is responsible for balancing traffic demand and capacity constraints during operational plan processing.

U-space services can be classified in two complementary architectures: pre-flight and in-flight services. In this thesis, we will focus on in-flight services.

- The pre-flight functional architecture contains all the functionalities needed to prepare and schedule a UAS flight, namely, e-registration, pre-tactical geofencing, flight planning management, and strategic deconfliction.
- The in-flight functional architecture includes all the functionalities needed once the UAS operation has started. That means, e-identification, tactical geofencing, tracking, monitoring, traffic information, emergency management, and tactical deconfliction.

2.2.1 U-space functional architecture

The U-space functional system architecture is still under development by some projects funded by SESAR Joint Undertaking [3]. Figure 2.2 shows a functional overview of the main actors in U-space operations. The U-space Service Provider (USP) is the core component, which is a server running on the cloud that provides all U-space services.

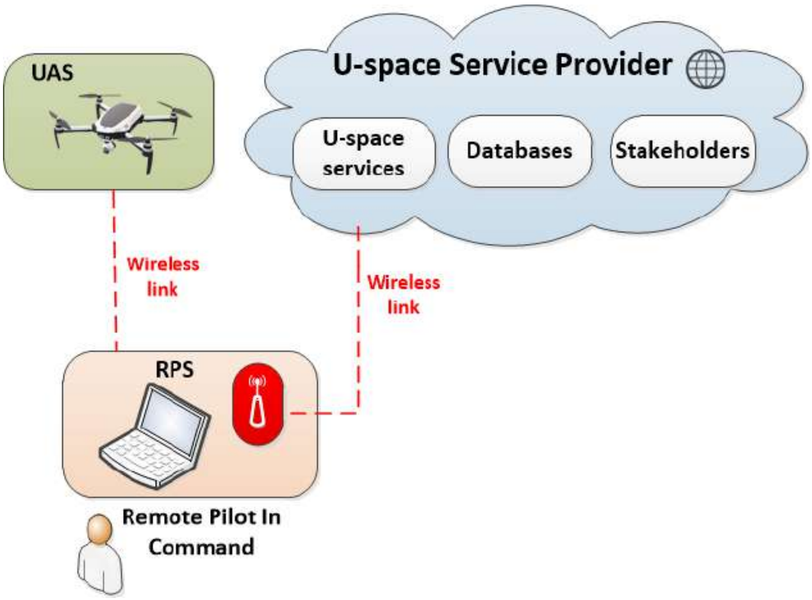

Figure 2.2: Functional overview of actors in U-space operations. The USP gathers information from UAS in the airspace and provides U-space services to UAS operators and authorities. A human pilot is placed for safety in the RPS.

---

[3] https://www.sesarju.eu/

These services need access to information about the UAS operating in the airspace, which is gathered by the USP and then stored in databases. This means, e.g., identification for registered UAS and operators, scheduled flight plans, current UAS tracked positions, fixed and temporal geo-fences, etc. Eventually, the objective of U-space is to have UAS communicating directly with the USP. However, due to safety reasons, current regulations also include a Remote Pilot Station (RPS) that runs on the ground, so that all information is checked by a human pilot in charge of the UAS. The functional architecture within the USP, which is still under design, describes the relationships and interfaces among the different components that implement the services. More details about the complete U-space functional architecture will be provided in Chapter 4.

2.2.2 Operational volume in U-space

A key operational component for airspace management in the U-space is the *Operational Volume* (OV) of a UAS flight. The OV is a 4-dimensional volume that consists of a single or multi-segmented 3-dimensional polygonal volume with a temporal component that represents the time and duration that the volume(s) is expected to be occupied. The OV represents the intent of an operator to perform an operation and maintain the aircraft within the bounds of that volume(s) at all times. After the OV for a proposed UAS operation is established, the USP performs checks on the airspace to ensure that there is no overlapping with other operations or airspace constraints (e.g., airports or restricted areas). Given the temporal aspect of an OV, geographic overlapping is allowed between operations as long as they are separated in time.

A graphical representation of the OV is depicted in Figure 2.3. More specifically, the OV is composed by the *Flight Geometry* and the *Contingency Volume*. The Flight Geometry (grey area in Figure 2.3) is what the operator submits to the USP; and it defines the volume of airspace where the UAS is intended to remain during the operation, as well as the period of time in which this volume will be active. The Contingency Volume (orange area in Figure 2.3) is larger and represents an additional buffer to account for environmental or performance uncertainties. As long as a UAS remains within its OV, the operation is considered *under control*, otherwise it would be tagged as *out of control*. Different

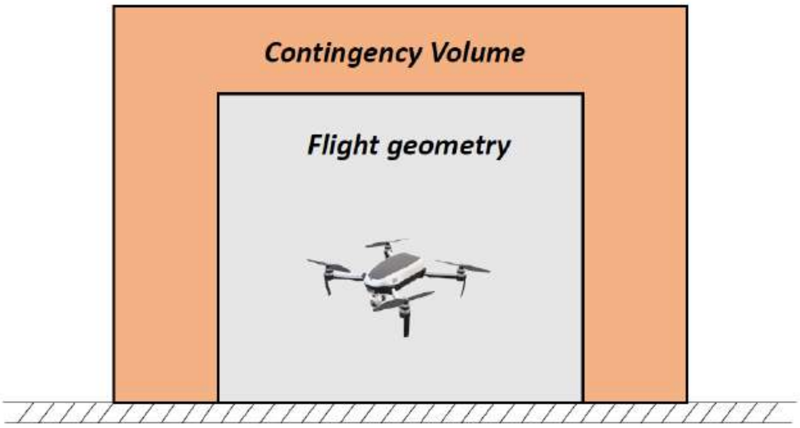

Figure 2.3: Graphical representation of the Operational Volume. The inner volume is the Flight Geometry, whereas the outer is the Contingency Volume.

procedures or mitigation actions may be carried out depending on this UAS state, as we will discuss in next chapters.

Chapter 3

Risk Assessment based on SORA Methodology for a UAS Media Production Application

The objective of this thesis is to develop methodologies for threat management in multi-UAS applications framed in the U-space. For that, it is key to understand the requirements and procedures established by the current European regulation to approve UAS operations. According to the classification of UAS operations described in Chapter 2, we focus on operations within the *Specific* category. For those activities, a previous operational authorization is required from the NAA. The UAS operator needs to submit a risk assessment of the intended operation to identify and mitigate the risks associated with the operation. This chapter describes the application of the SORA (Specific Operational Risk Assessment) methodology, which is the one proposed by JARUS (JARUS, 2018), to perform a risk assessment of an operation for aerial cinematography with multiple UAS. This example application, where a multi-UAS team needs to film sport outdoor events in rural scenarios, will help understand better the procedure to conduct a risk assessment for multi-UAS operations in the VLL airspace. The chapter goes through all steps in SORA, evaluating operational risks and discussing mitigation actions in the system. This study paves the way for integrating future multi-UAS systems in the civil airspace for autonomous cinematography and other applications with similar requirements.

3.1 Introduction

The use of UAS in civil airspace for commercial purposes is spreading fast, due to the decrease of their cost and the rapid increase of their level of autonomy. Many companies are turning to UAS to perform tasks traditionally done by humans, like packet delivery (Dorling et al., 2017), surveillance (Capitan et al., 2016), or infrastructure inspection (Ollero et al., 2018), among others. On the one hand, some of these applications benefit from the use of multiple cooperative aircraft, which makes operation approval more complex. On the other hand, many of the current running prototypes are only deployed in rural areas instead of highly populated hubs. This fact allows companies to circumvent some of the strong requirements of current regulation. Another appealing application in this context is media production for outdoor events (Alcántara et al., 2021). Their maneuverability and capacity to transport high-quality cameras make UAS interesting for amateur and professional cinematographers. Moreover, they can achieve unique viewpoints of a scene, produce visually pleasant shots, film places of difficult access, etc. However, operating UAS in civil airspace for autonomous media production is still challenging, as it entails relevant risks from a safety perspective. These can be classified into *ground* and *air* risks: ground risks are basically those involving third parties in the ground, whereas air risks are those involving third parties in the air.

The work of this Chapter is framed within the context of MULTIDRONE (MULTIple DRONE platform for media production) project [1], which is a EU-funded project that aimed to develop an autonomous system for aerial cinematography with a small team of UAS. The project focuses on outdoor applications for filming sport events (Mademlis et al., 2019), where each UAS has to coexist with its collaborative teammates and fly over areas non-densely populated. The system was demonstrated for filming cycling/rowing races in rural scenarios, where the UAS need to perform autonomous shots avoiding obstacles and no-fly zones (e.g., areas with gathered public).

UAS operations in MULTIDRONE lie within the Specific category, and hence, the final system deployment would require a safety risk assessment. The first contribution of this

---

[1] https://multidrone.eu

chapter is to apply SORA [2] to carry out a risk assessment of an aerial cinematography operation to be conducted with an autonomous small team of UAS. Multi-UAS operations for multimedia production are novel and related risk analyses are lacking. To the best of our knowledge, the work in this chapter is the first attempt to apply SORA for multi-UAS media production. The assessment will allow an evaluation of the risk level of the operation, discussing possible mitigation actions to reduce risks when deploying MULTIDRONE system.

As a second contribution, this study is of interest for UAS media production in general, as it will pave the way for integrating future UAS platforms for autonomous cinematography into civil airspace. Indeed, the considered operation fits with the first *STandard Scenario* (STS) defined by EASA. As mentioned in Section 2.1, EASA is compiling a set of STS (European Aviation Safety Agency (EASA), 2018), whose idea is to group several UAS operations for the Specific category within a same set of common specifications. These STS use risk operational aspects (e.g., over sparsely/congested areas, UAS characteristics, airspace use, etc.) to classify operations depending on standard specifications. Thus, the chapter contributes to the development of the SORA methodology for one of the STS being defined by EASA, focusing on multi-UAS multimedia operations, which is a novel application. Moreover, the discussion will help understanding better the flaws of SORA for multi-UAS operations.

The remainder of this chapter is organized as follows: Section 3.2 discusses some related work; Section 3.3 describes the main requirements of the UAS operations for media production considered in this chapter; Section 3.4 provides an overview of the SORA methodology; Section 3.5 details the application of SORA to the mentioned media production operation; Section 3.6 discusses the results obtained; and Section 3.7 includes conclusions and future work.

---

[2] This chapter is based on SORA 1.0, which is more extended, although SORA 2.0 was already published by JARUS in March 2019.

3.2 Related work

An essential issue for UAS integration into civil airspace has turned out to be risk evaluation. For that, there are different approaches that come from the world of traditional manned aviation. In Cour-Harbo (2018), many of these approaches are reviewed. They also compare the accuracy of SORA against an alternative approach based on a high-fidelity risk model. A risk analysis of UAS integration into non-segregated airspace is presented in Ferreira et al. (2018). The analysis is conducted from both a qualitative and quantitative perspective, and probabilities for risky events are estimated through simulation. The qualitative analysis uses the *Safety Risk* model proposed by the International Civil Aviation Organization (ICAO); whereas the quantitative analysis is done by means of a Fault Tree Analysis (FTA). The authors propose as future work to apply the method to fully autonomous aircraft, but the paper focuses on Remotely Piloted Aircraft Systems (RPAS). In Ancel et al. (2017), it is presented a framework to develop a series of tools capable of providing real-time risk assessment for UAS operations. The framework proposes the use of aircraft-generated health monitoring data along with augmented population density and other dynamic environmental inputs, to evaluate casualty risk and inform the operator of imminent failures. In McFadyen and Martin (2016), data-driven modeling techniques are used to evaluate existing air traffic before UAS operations. Thus, no-fly zones are discovered. The lack of a common accepted framework for risk management is addressed in Clothier et al. (2015), where some guidelines are also provided to apply the existing models. Another generic safety case is described in Denney and Pai (2016), based on experience with NASA UAS missions. Recently, a Bayesian framework has been proposed (Martin et al., 2018) to link the performance of detect-and-avoid functions with the probability that they will be needed. The system is compared with the qualitative methods from SORA.

The Federal Office of Civil Aviation (FOCA) in Switzerland has a national regulation with a risk-based approach for RPAS integration (FOCA, 2015). They propose a holistic approach called GALLO (FOCA, 2017) to guide authorization procedures for RPAS operations in low-level airspace. The SORA methodology is aligned with this national approach, since it is also a holistic safety risk-based assessment model used to evaluate operations of UAS of any class and size. SORA is being promoted by JARUS in the current UAS regulatory framework, and this is why we selected it as risk assessment methodology. Moreover, it is particularly suited for the type of UAS usually employed in media production operations.

3.3 UAS operation for media production

This section describes more specifically the UAS operation for media production that is assessed in this chapter. The objective is to analyze its main requirements from a safety point of view. The application at hand is autonomous cinematography with a small team of UAS. The system will be deployed to film sport events in outdoor settings. In particular, it will be tested to cover rowing and cycling races taking place in *rural* areas with some public (see an example in Figure 3.1).

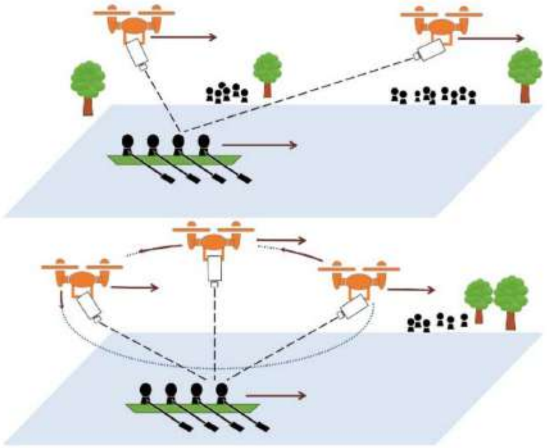

Figure 3.1: Example UAS operation for media production. UAS should fly over non-densely populated areas to film a rowing race. The UAS are not expected to fly over crowded areas.

Although the aerial cinematographers may also be remotely operated, this chapter focuses on the fully autonomous mode, where the UAS operate taking aerial shots without human intervention. This is a BVLOS operation, as the system is thought to operate at large-scale scenarios. Moreover, despite the existence of public in the sport events, these will take place in the countryside (e.g., around rivers or mountain roads); and the UAS are not supposed to fly over the public, who will be in no-fly zones. Therefore, the operation can be considered to happen in a sparsely populated area. The UAS considered for this application (from the MULTIDRONE project) are multicopters of medium size with the characteristics in Table 3.1. This kind of platforms are used due to their maneuverability. There are smaller products available, but a platform with a bigger payload was selected to transport heavier high-resolution, multimedia cameras and to increase the flight time (more batteries can be carried).

Part	**Brand & Model**
Frame	DJI S1000+
FCU	Pixhawk 2.1
GPS RTK	Here+
On-board communication module	Thales LTE/Wi-Fi communication module
Back-up radio	Futaba T14SG
On board computer	Nvidia TX2 + Intel NUC
LiDAR	Leddartech M16 (0,2 kg, 50m)
Parachute	Galaxy GRS 10/350
Camera + gimbal	BMMC + iFlight G40
Size (prop to prop)	1,45 m
Weight	11 kg

Table 3.1: Summary of characteristics for the MULTIDRONE UAS.

Table 3.2 summarizes the operational safety-related requirements. They comply with the definition of the *Generic* STS (European Aviation Safety Agency (EASA), 2018), which is the first standard scenario described by EASA to ease risk management for UAS operations.

UAS operation	**Media production**
Level of human intervention	Autonomous
Overflown areas	Sparsely populated area
Operational scenario	BVLOS over sparsely populated environment
Altitude limit	<$150m$ / $500ft$
Maximum UAS dimensions	$1.5m$ / approx. $50ft$
Typical kinetic energy expected	<$34kJ$ (approx. $25,000ft\ lb$)

Table 3.2: Summary of the main requirements of the UAS operation for media production.

In conclusion, it is required to apply a risk assessment for the MULTIDRONE system since the operation itself is considered Specific by the regulation. Moreover, the system is supposed to fly in autonomous mode, which may entail relevant risks due to collisions with other UAS, the ground infrastructure, or people.

3.4 Overview of the SORA methodology

SORA is a method based on the principle of a holistic/total system safety risk-based assessment model used to evaluate the risks involved in the operation of a UAS. Thus, it is based on a *Holistic Risk Model* that provides a generic framework to identify possible hazards and threats, as well as relevant harm and threat barriers applicable to a UAS operation. Given a specific operation, each risk can be defined as the combination of its frequency (probability) of occurrence and its associated level of severity. There are multiple risks to consider in a UAS operation but they all can be classified into *ground* and *air* risks in terms of safety. Ground risks are basically those involving third parties in the ground, whereas air risks are those involving third parties in the air.

In the end, SORA determines how confident one is, in a qualitative manner, about the fact that the UAS operation will remain safely in the Operational Volume (European Aviation Safety Agency (EASA), 2018), as it was defined in Chapter 2. As the UAS is inside the Flight Geometry, it is considered to be in normal operation and under operational procedures. However, if the UAS enters the Contingency Volume, it gets into an abnormal situation, being necessary the application of contingency procedures (e.g., returning home, manual control, landing on a predetermined site, etc). Last, if the UAS gets out of the Contingency Volume (i.e., out of the Operational Volume), emergency procedures must be executed, as the operation would be out of control.

The SORA procedure begins with a description of the so-called *Concept of Operations* (ConOps) , which specifies details of the operation assessed, such as the airspace requirements, the population density of the area, etc. It also describes the level of involvement of the crew and autonomous systems during each phase of the flight. After that, SORA proposes a step-by-step evaluation of ground and air risks. Last, a SAIL (Specific Assurance and Integrity Level) is determined for the operation. With this evaluation in mind, there is a table called Operational Safety Objectives (OSO), which defines the objectives to be met by the operation depending on the estimated SAIL. In summary, SORA provides a logical process to establish an adequate level of confidence to conduct the UAS operation with acceptable level of risk. Essentially, the SORA method is based on a number of steps, which are depicted in Figure 3.2.

3.5 Risk assessment for media production

This section elaborates the risk assessment for the proposed UAS operation in media production. Although the system consists of a team with more than one UAS operating together, the risk evaluation is centered on a single one. Thus, the operation is assessed from the perspective of one of the UAS of the system. The methodology used is SORA, which provides a logical process to analyze the operation step by step (JARUS, 2018). The following sections will go through the different steps of the SORA procedure to determine a level of confidence to conduct the media production operation in Section 3.3 within acceptable risk level.

3.5.1 Pre-application evaluation

Step #0 - Initial evaluation

The media production operation in this work has higher operational risks than one for the Open category. Indeed, it belongs to the Specific category, so running this risk assessment is justified and expected.

Step #1 - ConOps description

The operation consists of a team of UAS to film a sport event (rowing/cycling race) around a rural area. The system and its operational procedures are described in Alcantara et al. (2020). There is a ground station with three different modules: a *Director Dashboard*; a *Supervision Station*; and a module for mission planning and execution. The Dashboard is a Graphical User Interface (GUI) (Messina et al., 2018) wherein the Director and her/his media production team can specify artistic shots to film the event. The Supervision Station is another GUI so that a human *Supervisor* can check the safety of the missions. There is

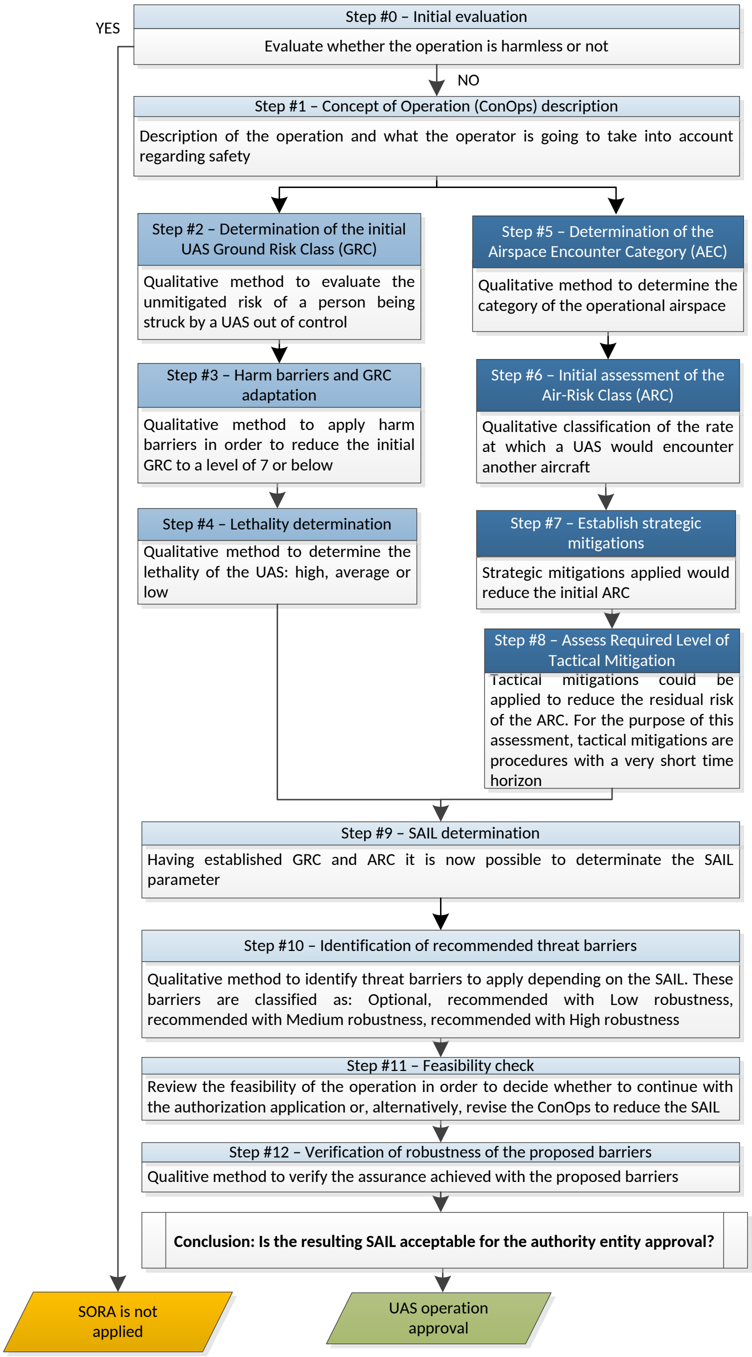

Figure 3.2: Scheme of the process to apply the SORA methodology. There are several steps to evaluate ground and air risks and estimate a SAIL (JARUS, 2018).

.

also a Long-Term Evolution (LTE) communication station to connect the ground station with the UAS.

The operation would be as follows. In the day of the race, the Director and her/his crew specify the desired shots with their duration and their starting time, position, etc. These parametric shots are sent to the module for mission planning, which is in charge of computing autonomously flight plans for each UAS in the team. Once the plan is computed, it is sent to the Supervision Station, so that the Supervisor checks it for safety. After approval, the flight plan is sent to each UAS from the ground station. Last, the Director will trigger the mission start at some point, and the UAS will take off and execute their plans autonomously. During mission execution, the Supervisor can monitor the system and cancel the mission due to safety risks at any moment. UAS would then go to landing stations.

In current media production with drones, in addition to the Director, two additional operators per drone are usually used: one to control the vehicle and another to control the camera. MULTIDRONE system tries to replace both operators with the autonomous modules in the ground station plus a safety Supervisor. This Supervisor should be a trained person with an essential role in terms of safety.

3.5.2 Ground risk process

Step #2 - Determination of the initial UAS Ground Risk Class

The initial UAS ground risk determines the unmitigated risk of having a person struck by the UAS (in case of a UAS loss of control) and it can be represented by the Ground Risk Class (GRC). The specifications of the media production operation in Table 3.2 can be input into the table of Figure 8 (page 25) in JARUS (2018), to establish an initial GRC of value 3.

Initial GRC = 3

Step #3 - Harm barriers and GRC adaptation

Once the initial GRC has been determined, it can be studied whether the system applies additional barriers that would reduce this GRC level. In particular, SORA indicates that optional mitigation measures can be adopted in order to achieve a lower final GRC. Each of these mitigation measures has associated a level of robustness depending on its integrity and assurance. The integrity indicates how useful the mitigation is to reduce the risk (e.g., a parachute over an area plenty of people would have a low level of integrity). The assurance tries to analyze if the mitigation is proven or not (e.g., a mitigation system tested in the field would have higher assurance than one tested only in simulation).

Harm barriers for GRC adaptation	**Robustness**		
	Low/None	Medium	High
An Emergency Response Plan is in place, validated by operator and effective	1	0	-1
Effects of ground impact are reduced (e.g., emergency parachute, shelter, etc.)	0	-1	-2
Technical containment in place and effective (e.g., tether)	0	-2	-4

Table 3.3: Harm barriers for GRC adaptation. Depending on the level of robustness of each barrier, the GRC is decreased with a certain value. The barrier applied in MULTIDRONE is highlighted (JARUS, 2018).

The mitigation measures considered by SORA at this step and their influence in the GRC are depicted in Table 3.3. MULTIDRONE system does not implement a detailed Emergency Response Plan validated by operator, but the UAS carries an emergency parachute. The UAS flies over a non-densely populated area, so the parachute integrity is high. Considering also its level of assurance, we can estimate the robustness of the mitigation as medium. Thus, the GRC is reduced by 1 unit to achieve the following final GRC:

$$\textit{Final GRC} = 2$$

The resultant GRC is a good number. However, in SORA both the GRC and the correction factors are defined on ordinal scales where it is only meaningful to compare relative risks. Therefore, SORA should improve the way of calculating these numbers as it is explained in Denney et al. (2018).

Step #4 - Lethality determination

This step of the process is supposed to evaluate the UAS lethality. Different UAS might have different lethality characteristics. SORA defines lethality with three qualitative descriptors: high, average, or low. However, the current version of SORA does not still provide information on how to establish these levels of lethality for a UAS.

3.5.3 Air risk process

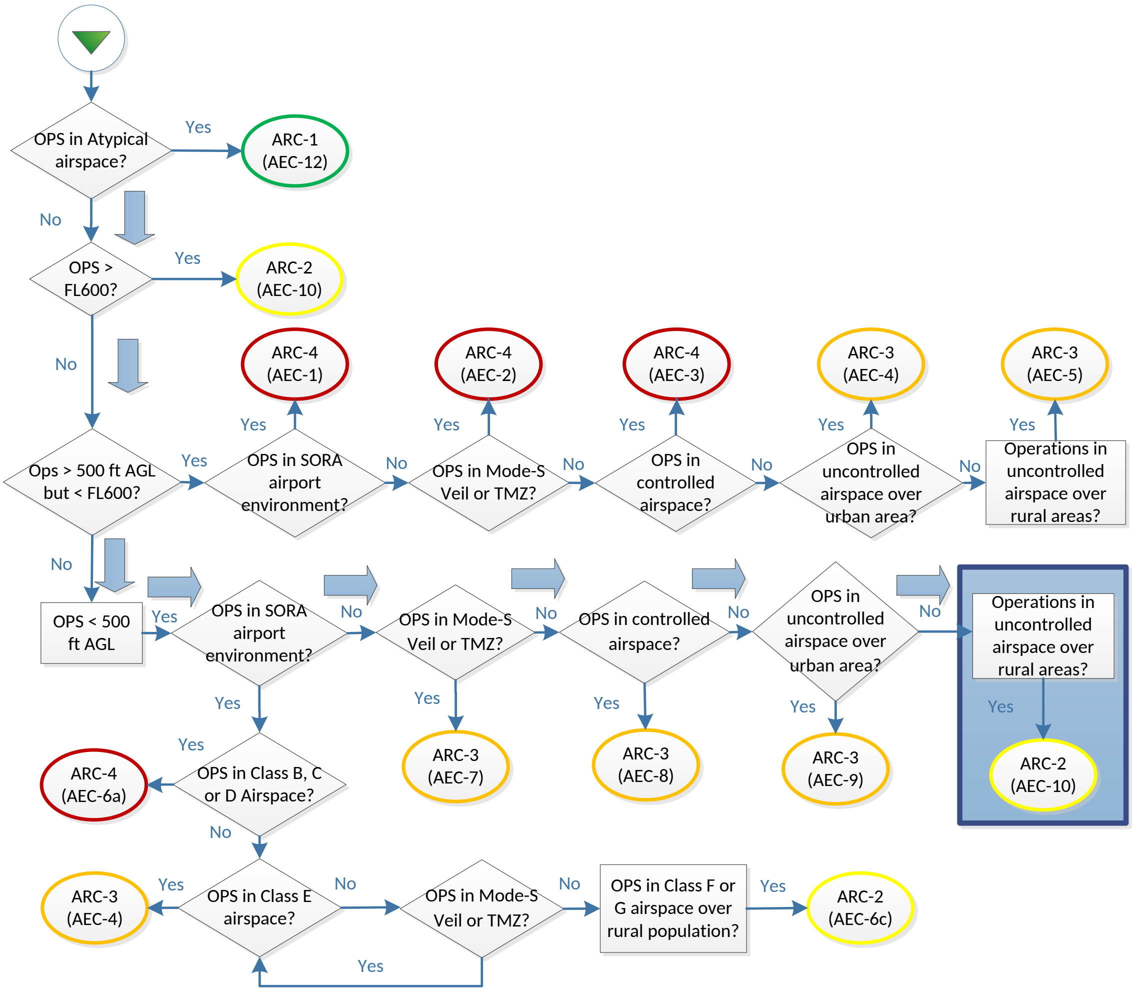

Figure 3.3: SORA process to determine the ARC based on the AEC. The blue thick arrows indicate the path for the evaluated operation.

Step #5 - Determination of the Airspace Encounter Category (AEC)

The AEC is a qualitative measurement to classify the airspace depending on the expected level of air collision risk. The AEC is grouped into 12 categorizations depending on the operational scenario. According to Figure 9 (page 28) in JARUS (2018), the airspace of the operation defined in Section 3.3 would have an AEC of level 10. Recall that the operation is below 120 meters on an uncontrolled airspace over rural areas. Level 10 of AEC is defined in SORA for *"Operations within Class G airspace below 120 m Above Ground Level (AGL) over rural population"*.

Step #6 - Initial assessment of the Air Risk Class (ARC)

The ARC is a qualitative classification of the rate at which a UAS would encounter another aircraft in typical scenarios for civil airspace. There are four types of ARC. First, the ARC evaluates the initial generic collision risk, before mitigation actions are applied. According to SORA, the ARC is computed related to the AEC through the flowchart shown in Figure 3.3. Therefore, the media production operation in this work would have an initial ARC of value 2.

Initial ARC = 2

SORA includes explicitly the following definition: *"ARC 2 is generally defined as airspace where the risk of collision between a UAS and manned aircraft is very low. This collision risk class requires some sort of collision mitigation but the amount of mitigation, and performance level of that mitigation will be low"*.

Once the initial ARC is determined, optionally, strategic mitigation can be used to reduce the ARC.

Step #7 - Application of strategic mitigation to determine final ARC (optional)

Strategic mitigation actions are those that are established before flying to reduce collision risks. They can be of different types: mitigation by boundary, restricting the geographical volume of operations; mitigation by chronology, restricting operations to certain times of day; mitigation by behavior, informing others about operations; and mitigation by exposure, limiting the time of exposure to risks. The MULTIDRONE system implements a mitigation by boundary, to restrict the operational volume. In particular, there are some no-fly zones (where there will not be other drones) in a semantic map that are established in pre-flight to avoid the areas where the public of the event may be placed. Thus, the operation of the UAS is restricted to a volume excluding those no-fly zones. SORA (JARUS, 2018) (page 32) indicates that the application of this mitigation, which is proven with low level of robustness, can reduce the ARC 1 unit.

Final ARC = 1

Step #8 - Assessment of required level of tactical mitigation

For this kind of assessment, tactical mitigation means procedures with a very short time horizon (seconds to a few minutes) which change the UAS encounter geometry to mitigate collision risk. This means reactive actions in flight. Some examples of on-board systems implementing tactical mitigation would be:

- Traffic Collision Avoidance System (TCAS).
- Air Traffic Control (ATC).
- Detect And Avoid (DAA).
- See and Avoid.

Air Risk Class	Tactical Mitigation Performance Requirement
ARC 4	High performance
ARC 3	Medium performance
ARC 2	Low performance
ARC 1	Optional - the operator may still need to show some form of mitigation as deemed necessary by the local authority

Table 3.4: Tactical Mitigation Performance Requirement (TMPR). Depending on the ARC, a different level of tactical mitigation is required. The level expected for MULTIDRONE system (highlighted) is optional, as the ARC is low (JARUS, 2018).

Table 3.4 is included in SORA (JARUS, 2018) (page 34) to determine the Tactical Mitigation Performance Requirement (TMPR). Depending on the ARC, a different level of tactical mitigation will be required for the system. The TMPR is the total performance required by all tactical mitigation combined. When combining multiple tactical mitigation actions, they will interact with each other and the system robustness will not be additive or multiplicative. For instance, the operator might decide to equip the UAS with DAA capabilities as one way to meet the required TMPR. In this case, the operator should know or assess the performance level of the DAA system on the UAS performing the operation. DAA or similar systems performance levels are generally described in the form of risk ratios. Annex D of SORA provides information on how to determine the performance levels of tactical mitigation and how to satisfy the TMPR based on the available tactical mitigation.

Since the operation has low ARC, the tactical mitigation would be optional for the MULTIDRONE system. Despite that, the system includes two different tactical mitigation actions with low level of performance. First, each UAS runs an algorithm for crowd detection based on image processing. Areas with crowds are included in a semantic map as no-fly zones so that UAS do not fly over them. Second, each UAS is equipped with a LiDAR (Light Detection And Ranging) sensor to detect and avoid obstacles during flight. Moreover, a communication channel with Quality of Service (QoS) between the different UAS in the team is available to share their positions. This information is also used for collision avoidance.

3.5.4 Final SAIL and Operational Safety Objectives assignment

Step #9 - SAIL determination

Having established the final ARC (ARC after strategic mitigation), it is now possible to derive the SAIL (Specific Assurance and Integrity Level) associated with the operation. The SAIL is the level of confidence that a specific operation will stay under control (it states whether the operation is safe). There are 6 possible values to reflect 6 increasing levels of confidence. The lowest, SAIL I, is adequate for operations with low intrinsic risk; the highest, SAIL VI, is adequate for operations with high intrinsic risks. Table 3.5

indicates how to compute the SAIL. The higher the SAIL, the higher the number of safety objectives to be met by the applicant with a higher level of robustness (in order to get operation approval). The UAS media production operation of this chapter would have a SAIL I according to Table 3.5.

	Final ARC			
Final GRC	1	2	3	4
1	I	II	IV	IV
2	I	II	IV	IV
3	II	II	IV	IV
4	III	III	IV	IV
5	IV	IV	IV	IV
6	V	V	V	IV
7	VI	VI	VI	IV

Table 3.5: SAIL values are computed from the final GRC and ARC. The media production operation assessed achieves SAIL I (highlighted) (JARUS, 2018).

Step #10 - Identification of recommended threat barriers

After computing the SAIL, the Operational Safety Objectives (OSOs) need to be identified. Depending on the SAIL, the system will have to comply with certain OSOs. Table 3.6 (page 36 of JARUS (2018)) depicts the OSOs and specifies their level of robustness recommended depending on the SAIL achieved by the operation. "O" indicates that the OSO is optional, "L" that low robustness is recommended, "M" medium robustness, and "H" high robustness. The OSOs are grouped based on the threat they help mitigate. The list gathers OSOs derived from the experience of experts and represents a solid starting point, but competent authorities may include additional OSOs. The OSOs and their level of robustness recommended for a SAIL I have been highlighted in Table 3.6. It can be seen that the proposed media production operation has only OSOs recommended as optional or with low level of robustness.

Step #11 - Feasibility check

In this step the feasibility of the ConOps proposed in Step #1 is checked to decide whether to submit the application to the competent authorities or, alternatively, revise the operation to reduce the risks. After having accomplished the whole processed established by SORA, the final SAIL obtained is low. Therefore, it can be stated that the UAS operation can be performed with a high level of confidence and there is not need to modify the ConOps before applying to the authority entity for operation approval.

Step #12 - Verification of robustness of the proposed barriers

This step concludes the SORA procedure. A successful evaluation of the robustness and effectiveness of all proposed barriers for the system will result in the approval of the operation. This robustness defines both the level of integrity (i.e., safety gained by each mitigation) and the level of assurance (i.e., proof that the safety gain is achieved). In Section 3.6, the different mitigation measures adopted by MULTIDRONE will be further discussed and related to the recommended OSOs.

3.6 Discussion

This chapter has gone through the whole process described in SORA to assess the risks of a UAS operation for media production. In summary, the evaluation has been positive and the risk level is low. The following statements can be made:

- A low level of GRC (2) is determined, since the operation will take place in areas non-densely populated. The integration of a parachute into the UAS as a strategic mitigation has a relevant impact in the system, reducing risks. Note that parachutes would be less helpful in urban areas, where UAS could fall on top of people anyway.
- A low level of ARC (1) is determined, as the UAS operates in uncontrolled and restricted airspace (no-fly zones established as strategic mitigation), and at a low altitude.

- The SAIL (I) determined is also low, so the operation could get approval from the competent authorities with no further adaptation.

Additionally, SORA recommends some OSOs to be met. With a SAIL I, all of them are only recommended as optional or with a low expected level of robustness. Next, the accomplishment of these OSOs in the MULTIDRONE system is detailed. Measures taken in MULTIDRONE are related with the corresponding OSOs from Table 3.6.

1. Ensuring competent operators (OSOs #01 and #03): MULTIDRONE system established the figure of the safety Supervisor, who must be a trained person in the operational procedures and UAS in order to monitor safety during operation. Also, the system capabilities, as well as its maintenance and insurance procedures were well documented.
2. System safety (OSO #06): Each UAS was designed considering system safety. In particular, a parachute and a LiDAR sensor for collision avoidance were included.
3. UAS inspection (OSO #07): Pre- and post-flight inspection procedures were documented.
4. Remote crew training (OSOs #09, #12, #13, #19, and #20): The Supervisor was a trained person to react to abnormal situations and failures, and to identify critical environmental conditions (visually and measuring). The Supervisor is in charge of coordinating operations with the media production crew.
5. Emergency management (OSOs #10 and #22): An on-board module to manage emergencies autonomously was implemented as tactical mitigation. Three types of emergencies can be detected by the UAS during flight: a low battery level, a loss of GPS signal, a loss of UAS localization by any means. If an emergency is detected, the UAS computes a safe route toward the closest landing station and navigates there to land. If that is not possible, the UAS tries to land on site safely.
6. Detecting and avoiding crowds (OSOs #15 and #16): An on-board system to detect crowds by image processing during operation was implemented as tactical mitigation.

The UAS avoids flying over those areas even if sent by the media Director by mistake. With the on-board LiDAR, the UAS can also react to unforeseen collisions. Moreover, no-fly zones were also identified before operation as strategic mitigation, to avoid private areas or with population. These areas are avoided during operation even if the operator sends the UAS there.

7. Adequate operational procedures (OSOs #08, #11, #18, and #22): Operational procedures were defined to deal with adverse conditions. As procedures, the system includes pre-flight planning, pre- and post-flight inspection, procedures to evaluate environmental conditions, emergency management, and other tactical mitigation.

8. Appropriate Human-Machine Interfaces (OSO #17): Two GUIs were developed to interact with the system. The Dashboard allows the Director to define artistic shots and missions, whereas the Supervision Station allows the Supervisor to check the safety of the operation. Both GUIs are ergonomic for human operators and subjective studies were performed to evaluate their usability.

A sufficient level of assurance was achieved by all mitigation implemented, as the system was tested as a prototype in field experiments.

3.6.1 Updates with SORA version 2.0

As mentioned before, our SORA study was performed based on version 1.0. However, after our study, SORA version 2.0 was released (JARUS, 2019). Therefore, we devote this section to describe the main updates in SORA 2.0 and how they affect our risk assessment.

1. SORA 2.0 is a document easier to follow and more practical. Step #4, Step #5, Step #11, and Step #12 of SORA 1.0 are eliminated in SORA 2.0, while two new steps are included: Step #9 "Adjacent area/airspace considerations" and Step #10 "Comprehensive safety portfolio".

2. The ConOps considers a formation flight as several flights that have a special relationship. This special relationship means that the UTM will not attempt to separate

the flights from each other and will never consider them to have lost separation between each other.

3. The GRC for the MULTIDRONE system in SORA 2.0 would be the same as in SORA 1.0:

   Final GRC = 3

4. The final ARC for the MULTIDRONE system in SORA 2.0 would be more restrictive than in SORA 1.0:

   Final ARC = 2

5. The SAIL for the MULTIDRONE system in SORA 2.0 would also be more restrictive than in SORA 1.0:

   SAIL = II

   With a SAIL of II, SORA 2.0 recommends some OSOs to be met, all of them as optional or with a low/medium expected level of robustness. The measures taken in MULTIDRONE that have been already discussed for SORA 1.0 are also enough to fulfill with the recommendations in SORA 2.0.

6. Specifically, Step #9 - “Adjacent area/airspace consideration” in SORA 2.0 addresses the risk posed by a loss of control of the operation, resulting in an infringement of the adjacent areas on the ground and/or adjacent airspace. As we implemented in MULTIDRONE a geofence system in order to consider no-fly zones, the risk of leaving the Operational Volume to enter adjacent areas is low.

3.7 Conclusions

In this chapter, a risk assessment has been done for a UAS operation in media production. The system evaluated was developed within the context of the EU-funded MULTIDRONE project and it consists of a team of UAS to film autonomously sport events in outdoor settings. According to the current UAS regulatory framework, a safety risk assessment is needed for approval operation in the Specific category, which is the one addressed in this chapter. Moreover, the SORA methodology is being developed and promoted by JARUS for this purpose. Thus, the chapter has applied SORA to the media production operation at hand, in order to assess whether the MULTIDRONE system could get operational approval from authorities. We carried out our study based on the first version of SORA, and discussed the updates when using the new version published in 2019.

As main conclusion, the technology in MULTIDRONE can be considered ready in terms of safety risk for the context of this chapter. The SAIL obtained throughout the assessment is low and the system would likely get an approval for operation from authorities. This is interesting for technology transfer, because companies could use the MULTIDRONE system, which has already been tested in field trials (Alcantara et al., 2020), to develop future commercial products.

We applied SORA 1.0 to evaluate operational risks from the perspective of a single UAS, and we conclude that SORA 1.0 lacks for a more specific treatment of risks associated with multi-UAS operations. In multi-UAS systems as the one in MULTIDRONE, there are inherent risks due to the nature of the operation. However, SORA 2.0 considers a formation flight as several flights that have a special relationship. This special relationship means that UTM will not attempt to separate the flights from each other and will never consider them to have lost separation between each other. Moreover, after comparing SORA 2.0 with SORA 1.0, we think that SORA 2.0 is a simpler and more practical version. Some steps that were less relevant have been removed and a couple of new ones included. The GRC and ARC are calculated in the same way but SORA 2.0 is a bit more restrictive than SORA 1.0. SORA 2.0 also takes into account risks associated with invading the adjacent airspace, something that SORA 1.0 did not consider.

OSO Number		**SAIL**					
		I	II	III	IV	V	VI
	Technical issue with the UAS						
OSO #01	Ensure the operator is competent and/or proven	O	L	M	H	H	H
OSO #02	UAS manufactured by competent and/or proven entity	O	O	L	M	H	H
OSO #03	UAS maintained by competent and/or proven entity	L	L	M	M	H	H
OSO #04	UAS developed to authority recognized design standards	O	O	O	L	M	H
OSO #05	C3 link performance is appropriate for the operation	O	L	L	M	H	H
OSO #06	UAS is designed considering system safety and reliability	O	O	L	M	H	H
OSO #07	Inspection of the UAS (product inspection) to ensure consistency to the ConOps	L	L	M	M	H	H
OSO #08	Operational procedures are defined, validated and adhered to	L	M	H	H	H	H
OSO #09	Remote crew trained and current and able to control the abnormal situation	L	L	M	M	H	H
OSO #10	Safe recovery from technical issue	L	L	M	M	H	H
	Human error						
OSO #11	Operational procedures are defined, validated and adhered to	L	M	H	H	H	H
OSO #12	Remote crew trained and current and able to control the abnormal situation	L	L	M	M	H	H
OSO #13	Multi crew coordination	L	L	M	H	H	H
OSO #14	Adequate resting times are defined and followed	L	L	M	M	H	H
OSO #15	Automatic protection of critical flight functions	O	O	L	M	H	H
OSO #16	Safe recovery from Human Error	O	O	L	M	M	H
OSO #17	A Human Factors evaluation has been performed and the HMI found appropriate for the mission	O	L	L	M	M	H
	Adverse operating conditions						
OSO #18	Operational procedures are defined, validated and adhered to	L	M	H	H	H	H
OSO #19	The remote crew is trained to identify critical environmental conditions and to avoid them	O	L	M	M	M	H
OSO #20	Environmental conditions for safe operations defined, measurable and adhered to	L	L	M	M	H	H
OSO #21	UAS designed and qualified for adverse environmental conditions	O	O	M	H	H	H
	Deterioration of external systems supporting UAS operation						
OSO #22	Procedures are in-place to handle the deterioration of external systems supporting UAS operation	L	M	H	H	H	H
OSO #23	The UAS is designed to manage the deterioration of external systems supporting UAS operation	L	L	M	H	H	H
OSO #24	External services supporting UAS operations are adequate to the operation	L	L	M	H	H	H

Table 3.6: Operational Safety Objectives (OSOs). Depending on the SAIL, each objective is recommended as optional (O), or with low (L), medium (M) or high (H) robustness (JARUS, 2018).

Chapter 4

Unmanned Aerial Traffic Management System Architecture for U-space In-flight Services

This chapter presents a software architecture for UTM implementing the U-space concept. In particular, we propose a system that provides the required in-flight services for automated decision-making during real-time threat management and conflict resolution. Our software architecture is implemented as open-source and it is modular and flexible enough to accommodate additional U-space services in future developments. In its current implementation, our UTM solution is capable of tracking the aerial operations and monitoring the airspace in real time, in order to perform in-flight emergency management and tactical deconfliction. We show experimental results in order to demonstrate the UTM system working in a realistic simulation setup. For that, we performed our tests with the UTM system and the operators of the aerial aircraft located at remote locations, with the consequent communication issues, and we showcased that the system was capable of managing in real time the conflicting events in different multi-UAS use cases.

4.1 Introduction

There are already some initiatives to integrate UAS into civil airspace and fulfill their operational requirements (Peinecke and Kuenz, 2017). NASA created the UTM concept (Kopardekar, 2015), whereas Europe has extended this UTM framework by proposing the U-space ecosystem (SESAR, 2017). An overview of the U-space ecosystem recently proposed by EASA (European Aviation Safety Agency (EASA), 2020) is depicted in Figure 4.1. The idea is to have a U-space Service Provider Platform, which is a server running on the cloud, as the core component. There, the UTM system consists of a software architecture that provides U-space services to the different actors in the U-space ecosystem using as bridge the U-space Service Manager (USM), which is a specific module of this UTM system.

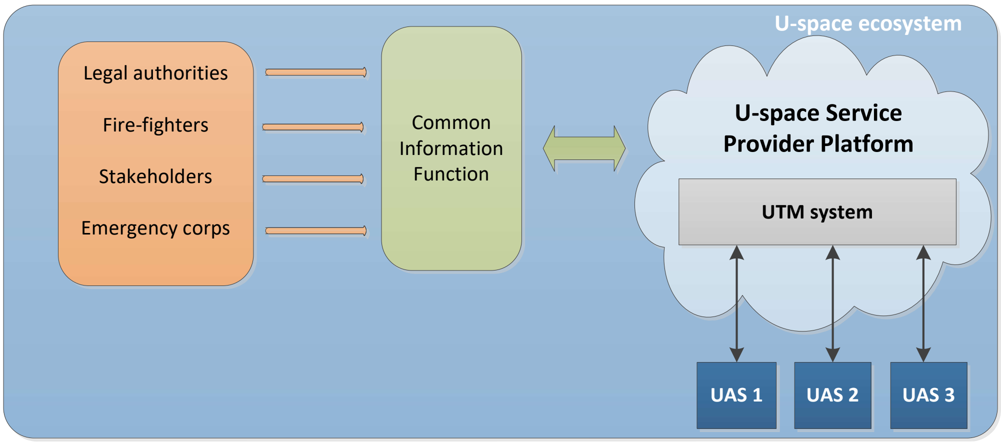

Figure 4.1: Overview of the U-space ecosystem proposed by EASA (European Aviation Safety Agency (EASA), 2020). The UTM system offers U-space services to the different actors and runs on a remote server called U-space Service Provider Platform.

Currently, the community is in the process of developing further these U-space services. In this chapter, we take a first step and propose a novel software architecture that aims to serve as a common framework for implementing and integrating U-space services. Our solution has been developed within the context of the European project GAUSS [1], whose main objective is leveraging high-performance positioning functionalities provided by the Galileo ecosystem for U-space operations, including a validation phase with actual

---

[1] https://cordis.europa.eu/project/id/776293

fixed-wing and rotary-wing UAS. We present an architecture that is service-oriented and safety-centered, and that allows the airspace actors to abstract from specific UAS technologies. Besides, we implement a set of U-space services to manage complete UAS operations, but focusing on *in-flight* services (i.e., those required to handle the operations during the flight phase). Nonetheless, the architecture is modular and flexible enough to be extended with additional functionalities as new services become functional.

In this chapter, we first review other relevant works about UTM (Section 4.2). Second, we analyze the design properties for our UTM architecture (Section 4.3). Given a series of desired architectural guidelines (Section 4.3.1), we propose the open-source Robot Operating System (ROS) [2] as underlying middleware for our UTM system (Section 4.3.2). Third, we contribute a new UTM system architecture implementing the U-space concept (Section 4.4). Our proposal represents a general framework for U-space services, which is modular, flexible, and technology-agnostic; but we describe our specific implementation for a set of core in-flight services dealing with unexpected UAS conflicts during their flight phase. Our software framework integrates automated decision-making procedures, which is one of the main gaps for current UTM solutions. Additionally, we show an actual realization of our UTM architecture that is available as open-source software for the community and we demonstrate its capabilities (Section 4.5). In order to showcase the correct integration of all our components and services, we have defined use cases for UAS operations involving all the developed functionalities (Section 4.5.1); and we have assessed our results in terms of performance by running the whole system in a realistic Hardware-In-The-Loop (HITL) simulation setup for multi-UAS operations (Sections 4.5.2 and 4.5.3). Finally, we draw the main conclusions of this work and point at future lines for further development (Section 4.6).

4.2 Related work

The U-space framework proposes a UTM system as the software architecture that provides services to the different U-space actors. A possible classification for the services is depending on whether they are activated in the UAS pre-flight phase or during the flight:

---

[2] http://www.ros.org

- **Pre-flight** services involve the functionalities required to prepare and schedule a UAS operation. The aircraft and the operator need to register (*E-registration*), and the initial flight plan has to be handled before being accepted (*Flight planning management*). Then the pilot may get assistance through information about predefined restricted areas (*Pre-tactical geofencing*) and the resolution of possible conflicts before flying (*Strategic deconfliction*).
- **In-flight** services involve the functionalities required to handle the operation after the UAS has taken off. This includes updates for the operator (*Tactical geofencing*) or the UAS itself (*Dynamic geofencing*) regarding geofences during the flight. Also, tracking information about the current position and predicted trajectory for each UAS (*Tracking*). This information is then used to create a situation of the airspace (*Monitoring*) and to generate warnings and contingency actions under possible threats (*Emergency management*). In order to keep a safety distance between aircraft and geofences, alternative plans could also be suggested during the flight (*Tactical deconfliction*).
- Last, there are some services that can be activated both in the pre-flight or in-flight phase. These services provide identification (*E-identification*), weather forecasts (*Weather Information*) or more generic information (*Drone Aeronautical Information Management*), create an interface with the ATC (*Procedural Interface with ATC* and *Collaborative interface with ATC*), or control and manage the UAS density in the airspace (*Dynamic Capacity Management*).

In Table 2.1, we summarized the current level of implementation in Europe of U-space services (Eurocontrol, 2020). According to Table 2.1 and to our study of the state of the art, in-flight services have been less addressed by UTM systems in general, with a notorious integration gap still existing. In this chapter, we focus on in-flight functionalities to develop a UTM system, although our architecture is general enough to cover all kinds of services. In particular, we integrate those services related to the management of unexpected events while the UAS are flying, namely Tracking, Monitoring, Emergency management, and Tactical deconfliction. These services belong to the U2 and U3 implementation phases, which are scheduled to be developed between 2021 and 2029. The detailed functional system architecture is still under development, but there is already a list of services defined for each deployment phase (Barrado et al., 2020).

The development of completely operational UTM systems is still at an early stage, even though it has recently become a growing field. The authors in Jiang et al. (2016) define what a UTM system should be, and they give an overview of both a physical UTM architecture and a UTM software manager based on automated services. Big companies are one of the major parties interested in boosting the deployment of UTM. For instance, Google has proposed an ecosystem (Google, 2015) where all UAS should be equipped with communication and *sense & avoid* technologies in order to perform cooperative flights when encountering other UAS or manned aircraft. In their proposal, the separation and planning services would be provided by an *Airspace Service Provider*. Furthermore, Amazon has put forward a one-operator-to-many-vehicle model (Amazon Prime Air, 2015), where the decision-making authority gets significantly distributed among the operators.

Additionally, there exist several commercial UTM system applications in the market. They implement most pre-flight services but just partially a few in-flight services. For instance, Airmap (AirMap, 2018) has its focus on UAS registration, geographic information systems, flight communication, traffic monitoring, and user interfaces. The Unifly platform (Unifly, 2020) connects authorities with pilots to safely integrate UAS into the airspace. On the one hand, authorities can visualize and approve UAS flights, as well as manage *No Flight Zones* in real time. On the other hand, pilots can manage their UAS (e.g., with the E-registration, E-identification and Flight plan management services) and they can plan and receive flight approvals aligned with international and local regulations. Another framework is the Thales ECOsystem UTM (Thales, 2017), which integrates UAS and pilot registration. ECOsystem provides a flight planning functionality, using airspace rules and situational awareness as guidelines. It also includes tools to manage map overlays and 3D terrain views.

The aforementioned UTM applications offer pre-flight UTM services and some in-flight capabilities such as UAS tracking. Even though they are capable of publishing real-time information about the UAS, they do not manage operations autonomously during the flight phase. Moreover, it is important to highlight that all those applications are commercial products that are not available for the community as open software.

The scientific community has also been putting effort into functional UTM frameworks; a recent review of related works can be seen in Rumba and Nikitenko (2020). A prototype UTM for flight surveillance has recently been proposed in Taiwan (Lin et al., 2019). One of its core properties is the capability to monitor vehicles, being the ADS-B (Automatic Dependent Surveillance Broadcast) technology used for surveillance. There is a pre-flight procedure to schedule and approve flights and then, the UTM system can send surveillance alerts during the operation, though all decisions for conflict resolution are up to the pilot. Another UTM system has been presented in Sweden (Lundberg et al., 2018). It incorporates a complete toolkit to manage traffic, geofences, flight altitude segregation as in the general aviation, and complex visualization. This research also identifies problems that dense traffic in the low-level airspace will bring to city users, by simulating the future urban airspace. In general, the functionalities of the aforementioned systems have only been demonstrated through simplistic simulations, and quite a few works have been devoted to field flight campaigns for preliminary tests (Aweiss et al., 2019; Alarcon et al., 2020). We also proposed in a previous work (Millan-Romera et al., 2019) a more realistic simulator for UAS operations, based on the ROS middleware and the 3D simulation suite Gazebo [3]. In that work, we introduced a preliminary definition of our in-flight services and a tool for mission validation. In this chapter, we go beyond by implementing a complete UTM architecture. We integrate in-flight services to handle unexpected conflicts that may occur while UAS are flying, and we showcase the performance of our system through heterogeneous use cases.

Finally, regarding the implementation of particular in-flight services, there are different approaches for conflict resolution and emergency management. Many works (Tan et al., 2019; Ho et al., 2019; Sacharny et al., 2020) focus on flight planning and scheduling at a strategic level, i.e., in the pre-flight phase; though in-flight automated decision-making has not been properly covered in UTM systems. In general, given the massive search space to find optimal solutions for conflicts in VLL airspace scenarios, approximate solutions based on heuristic solvers (Tan et al., 2019) or lane maneuvers (Sacharny et al., 2020) predominate over optimal deconfliction approaches. In Rubio-Hervas et al. (2018), a probabilistic framework is proposed to formulate the risk involved in UAS operations. That

---

[3] http://gazebosim.org

methodology could be integrated for automated, real-time data analysis in an emergency management solution. We take methodological ideas from these previous works, in order to implement conflict resolution and emergency management in our system considering the specifics of UAS operations in a civil airspace. However, the focus of this chapter is on the architecture design and integration, rather than on the particular algorithms for conflict resolution.

4.3 Design framework

This section settles the framework for our UTM architecture. First, we analyze the desired properties and requirements for a UTM architecture from a design perspective. Then, we introduce ROS, which is the open-source middleware that we use to implement our architecture. We justify this selection by discussing the main features in ROS and how they fit our UTM system requirements.

4.3.1 Guidelines for system design

The Global UTM Association (GUTMA) is a non-profit consortium of worldwide UTM stakeholders, and it has promoted a discussion about which key properties should be present in future UTM systems (Global UTM Association, 2020). After reviewing their technical report, we came up with a summary of these key features for UTM systems. We believe that the following aspects should be taken into account during the design phase of any UTM architecture:

- **Digital**. The process of system digitization consists of making the communication between the different actors and components digital, and introducing automated decision-making procedures. This is a key aspect in UTM to reduce the operators' workload in an efficient and secure manner. Moreover, it enables the real-time exchange of data between relevant parties for situation awareness and an easier integration of the UTM services.

- **Flexible and modular**. A UTM architecture should be flexible and adaptable to incorporate new actors (e.g., stakeholders) and functionalities (e.g., services) as they appear. Besides, the system should be modular, i.e., made of composable and reusable modules, in order to ease the process of creating more complex functionalities.

- **Scalable**. A scalable architecture is needed to grow with new actors and services. In order to achieve that, not only is the aforementioned modularity desirable but also a paradigm with distributed responsibilities, rather than the obsolete scheme with a centralized ATC.
- **Safe and secure**. These two features are top priorities in any UTM ecosystem. In this sense, the system should know who is flying each unmanned aircraft, where they are flying (or intend to fly) to, and whether they are conforming (or not) to mandatory operating requirements.
- **Automated**. A UTM system providing automated services to assist the UAS operators will be more efficient and secure. Therefore, the system should provide support through automated functionalities for flight planning, monitoring, and real-time deconfliction, in order to ensure safe operations for both manned and unmanned aircraft.
- **Open-source**. The use of open-source technologies is preferable, as they offer a global approach towards creating and evolving the necessary services and protocols for scalable operations. Moreover, open-source components can speed up the development and the deployment of UTM services.

4.3.2 Robot Operating System

ROS (Robot Operating System) is an open-source framework for robot software development. It consists of a collection of libraries, tools, and conventions to ease the creation of complex applications in robot systems; including hardware abstraction, low-level device control, implementation of commonly-used functionalities, message-passing between processes, and package management. ROS is also well known among the UAS community, as it provides drivers to communicate with a wide spectrum of both open-source and commercial autopilots and onboard sensors. The use of ROS for multi-UAS systems is extending fast, as it paves the way for integration of heterogeneous hardware and software systems. ROS is a framework based on multiple processes (so-called *nodes*) that run in a distributed fashion. These processes can be grouped into *packages*, and communicate with each other by passing *messages*, which are typed data structures. On the one hand, ROS implements asynchronous communication through a publish/subscribe paradigm where nodes can stream messages over different *topics*. On the other hand, synchronous communication is implemented through *services* for request/response interactions.

We decided to use ROS as middleware for our UTM architecture because it offers multiple features that fit our design guidelines. First, ROS is designed to create modular and reusable components, and its preferred development model is to write ROS-agnostic libraries with clean functional interfaces. Therefore, ROS yields flexible and scalable systems that can be adapted easily to incorporate new functionalities. Second, ROS is open-source and strongly supported by a large community. Its federated system of code repositories enables collaboration and fast development for UAS complex systems. Communication solutions and drivers for most popular autopilots (e.g., PX4, ArduPilot, DJI, etc.) are already available in ROS. Moreover, ROS provides remarkable tools for system integration and testing, and there exist multiple options for multi-UAS simulation, including Software-In-The-Loop (SITL) solutions for common autopilots (Real et al., 2020).

ROS also presents some issues for multi-UAS systems. Mainly, its centralized nature due to the existence of a single *master* node that handles all procedures for node registration, and its lack of proper QoS policies. However, there exist efficient solutions for these issues. Multi-master architectures have already been used for applications with multiple UAS (Alcantara et al., 2020); and the adoption of ROS 2 is growing fast among the community, with a smooth transition from primary ROS. ROS 2 proposes a fully distributed scheme, where each node has the capacity to discover other nodes without the need for a central master. Since it is built on top of the industrial standards DDS (Data Distribution Service) and RTPS (Real-Time Publish-Subscribe), ROS 2 is capable of offering multiple QoS options for improved communication.

Even though we have chosen ROS to implement our UTM architecture, mainly due to its advantages for system integration and realistic SITL simulation, it is important to remark that the proposed UTM architecture is a more general concept, and it could be adapted to alternative middleware solutions.

4.4 UTM system architecture

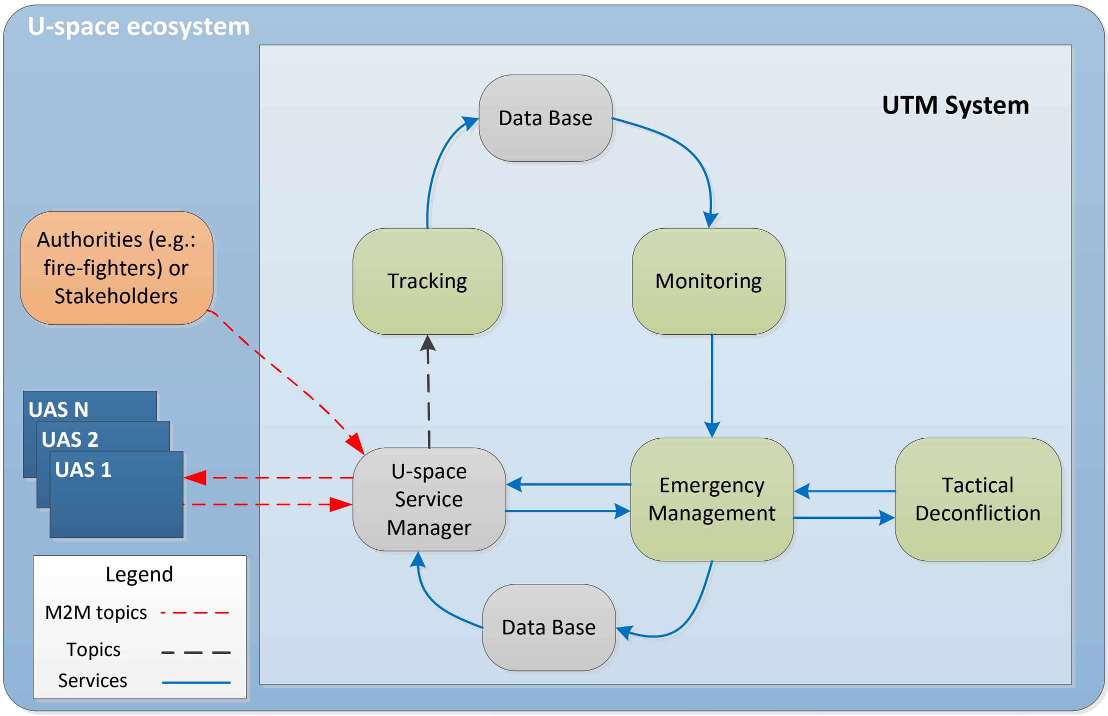

Figure 4.2: Overview of the proposed UTM system architecture. This system would be running in a remote server named U-space Service Provider Platform. The red arrows indicate remote communication links with other machines in the ecosystem.

This section describes our proposed UTM system architecture. Figure 4.2 depicts an overview of all the software modules involved, as well as their interactions. The modules in green implement specific U-space services. As it was explained in Section 4.2, we focus on those services that are required to address unexpected events during the flight operation of a UAS. In particular, we cover four services with their corresponding modules: Tracking, Monitoring, Emergency Management (EM) and Tactical Deconfliction (TD). Besides, our system includes additional software modules that provide support to the UTM architecture. First, there is a Data Base (DB) component that is in charge of handling all the relevant information about the state of the airspace, for instance, the current flight plans and tracks for all UAS operations (which are updated by the Tracking module) and the active geofences (which can be activated externally by auxiliary stakeholders like fire brigades or internally by the Emergency Management module). Second, the U-space Service Manager (USM) is a key module that acts as an interface between the UTM system and the rest of the U-space ecosystem. Basically, it receives state information and alerts from both the UAS and the external auxiliary stakeholders, and it communicates back recommended actions to deal with threatening events. These recommendations are generated by means of the interaction between the Tracking, Monitoring, Emergency Management and Tactical Deconfliction modules.

Our system is built upon ROS (Section 4.3.2) and hence, each module consists of a software process implemented as a ROS node. The communication between modules takes place through ROS topics and services. In particular, the system is designed to use services in a preferable manner, as they provide the possibility of acknowledging message reception, which is crucial to manage reliably many of the UTM interactions. In those cases, one of the modules acts as a server while others act as clients, which results in an asynchronous communication between modules. Upon a client request, the server module will carry out the requested activity and then it will reply indicating whether the result was successful or not. Nevertheless, there are also a few cases where ROS topics are needed. Topics provide a synchronous communication and they are used by modules that need to publish information at a constant rate.

In the following sections, we will provide a more detailed description of the different modules in our UTM system. For each module, we describe its functionality and interactions with other modules, as well as the methodology that we have used to implement them.

4.4.1 U-space Service Manager

The U-space Service Manager is a key module in the UTM system, as it provides an interface with the rest of the actors in the U-space ecosystem, i.e., the UAS operators and auxiliary stakeholders like the airspace authorities, the fire-fighters, or the police.

First, the USM receives positioning measurements from the control station of each UAS, which is transmitted by their onboard telemetry and ADS-B transceivers (if available). This information is forwarded to the Tracking module in order to keep updated a list of tracks for all operational UAS. Second, the USM receives warning information that may be relevant for the UTM system, coming from external stakeholders (e.g., a declaration of a wildfire by fire-fighters) or from the UAS (e.g., the detection of a jamming attack [4] or a technical failure due to a lack of power). These events are treated as possible threats by the system and are forwarded to the EM, which is in charge of processing them. Last, the USM communicates back to the UAS operators any action determined by the EM (e.g., an immediate landing or an alternative flight plan). Due to regulatory restrictions, the actions involving the variation of a UAS flight plan are just recommendations that must be confirmed or rejected by the corresponding pilot. In case of acceptance, the USM would notify the DB to update the state of that operation and its flight plan.

4.4.2 Data Base

The function of the Data Base module is to handle a digital data base with the required information to represent the situation of current UAS operations. Basically, this information is made up of active geofences and UAS operations. The DB works as a server for the rest of the UTM system and hence, other modules can read the database in order to carry out their tasks (e.g., the Monitoring module uses the UAS predicted trajectories to detect lack of separation events); or they can write the database to update the airspace situation (e.g., the USM can notify new accepted flight plans and the EM new geofences).

Attribute	Data type	Description
Identifier	Integer	Unique number for geofence identification
Type	Enum	Cylindrical or polygonal
Geometry	List of 2D waypoints	Definition of the horizontal shape, defined by a circle or a polygon
Min/max altitude	Float	Altitude range where the geofence is active
Start/end time	Float	Time period in which the geofence is active

Table 4.1: Attributes of a geofence object.

The DB manages two types of objects internally: geofences and UAS operations. Table 4.1 and Table 4.2 depict the data structures for each of these objects. A geofence is

---

[4] A jamming attack consists of an attempt to jeopardize the GNSS (Global Navigation Satellite System) signal of a UAS.

Attribute	Data type	Description
Identifier	Integer	Unique identification of the aircraft
Priority	Enum	Priority of the operation in the airspace
Flight plan	List of waypoints (x, y, z, t)	Reserved 4D trajectory for the operation
Next waypoint	Integer	Waypoint index that the UAS is currently targeting
Predicted trajectory	Float	Prediction of the future UAS trajectory
ConOps	String	Description of the concept of the operation
Flight Geometry	Float	Radius of the cylindrical volume where the UAS is intended to remain during its operation
Operational Volume	Float	Radius of the outer cylindrical volume to account for environmental or performance uncertainties

Table 4.2: Attributes of a UAS operation object.

a 4D portion of the airspace (a 3D geometrical space with an activation period of time) which has special restrictions for UAS, like flight prohibition. In the UTM context, the term *dynamic* geofence is used for those created during the UAS operation, while *static* geofences are set in a pre-flight phase. The DB stores for each geofence the following information: a unique identifier, its type (cylindrical or polygonal), its geometry definition, its minimum and maximum altitude, and its starting and finishing time instants. Besides, the DB stores each UAS operation, which consists of the following data: a unique identifier for the UAS given by its ICAO (International Civil Aviation Organization) address [5], the priority level of the operation, its associated flight plan, the next waypoint assigned to the UAS, the predicted trajectory of the UAS, a brief description of the UAS operation, and the sizes of the Flight Geometry and the Operational Volume.

4.4.3 Tracking

The Tracking module implements the U-space service with the same name. According to the U-space definition (Section 4.2), the main functionality of this service is to track the operational UAS in the airspace. These tracks contain information updated in real

---

[5] The ICAO addresses are 24-bit numbers to identify aircraft uniquely worldwide.

time about the UAS current position and its predicted trajectory within a certain time horizon. The module computes the tracks by fusing information from different sources that it receives through the USM. In particular, measurements from the UAS telemetry and ADS-B transceivers (when available) are integrated to achieve a more accurate estimation of the UAS positions. Moreover, the future trajectory of each UAS is predicted given its current position and velocity, as well as its flight plan. The Tracking component keeps updated the UAS tracks in the DB module, so that this information is available for the rest of the system.

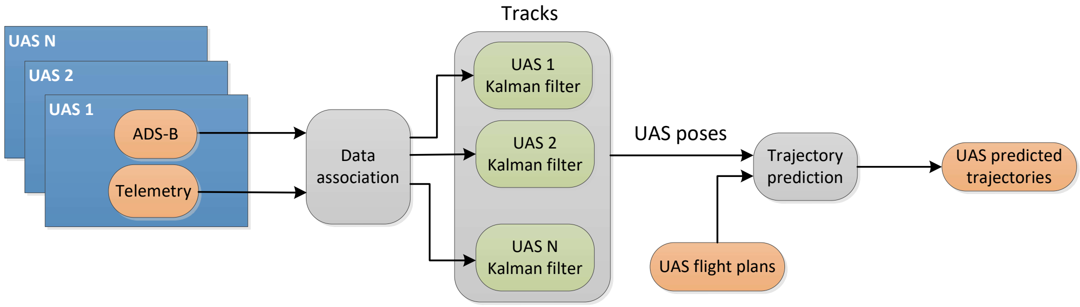

Figure 4.3: Scheme with the internal components of the Tracking module. The *data association* component matches the measurements from the UAS with their tracks, to update the corresponding Kalman filters. The future UAS trajectories are predicted using the tracks and the flight plans.

Mathematically, the Tracking module implements a stochastic filter that maintains a list of objects to estimate the state of each UAS, as depicted in Figure 4.3. This filter allows the system to cope with noisy and delayed measurements, as well as irregular sensor rates. The state of each UAS consists of its 3D position and velocity (expressed in Universal Transverse Mercator coordinates), and its current waypoint, i.e., the next waypoint of the flight plan that the UAS is targeting. The continuous variables are estimated through a Kalman Filter that integrates the measurements coming from the UAS telemetry and the onboard ADS-B transceivers. These data are previously transformed from geographic to Universal Transverse Mercator coordinates.

The procedures is as follows. At a constant rate, the list of operations is read from the DB, in order to identify the active UAS. The state of all those UAS is predicted and then updated with the received observations. Each observation can be easily associated with its corresponding track, since they all come with a unique UAS identifier. Observations with *unknown* identifier are ignored by the filter, as they are considered non-cooperative aircraft. Moreover, the current waypoint for each UAS is computed by searching for the waypoint in its flight plan that is closest to its current position. The future trajectory within a given time horizon is also predicted for each track. If the current position of the UAS is close enough to its current waypoint (according to a given distance threshold), the prediction of the future trajectory sticks to the flight plan. Otherwise, the Kalman Filter is used to predict a trajectory given the current UAS position and velocity. Finally, after each step, the Tracking module updates all the information about the tracks in the DB module.

4.4.4 Monitoring

The functionality of the Monitoring module is to monitor the state of the airspace and to detect potential conflicts or threats that need to be managed by the UTM system. In particular, the module deals with conflicts related with UAS trajectories. Thus, it detects: (i) whether a UAS gets out of its reserved flight volume; (ii) whether it is in conflict with a geofence; or (iii) whether two UAS lose a minimum required separation. For that, the Monitoring module reads periodically information from the DB about the UAS tracks and the geofences, and it analyzes that information to determine when a threatening situation should be reported to the EM. When the Monitoring notifies the EM, it indicates the type of the detected threat, a prediction of the time instant when the event will occur and a snapshot with the current predicted trajectories of the involved UAS. This last piece of information is sent so that the modules resolving the conflicts use exactly the same data to evaluate the situation and hence, time glitches and incoherent solutions are avoided.

The first type of issue that is evaluated by the Monitoring module is related to the Operational Volume (OV) that is reserved by each UAS operation (see Figure 4.4). Recall from Chapter 2 that the OV is a 4D space that consists of a 3D volume around the flight plan with a temporal component representing the time that the volume, as part of an operation, will be reserved in the U-space ecosystem. The OV is composed by: the Flight Geometry, which defines the volume of airspace where the UAS is intended to remain during its operation; and the Contingency Volume, which is an outer surrounding volume

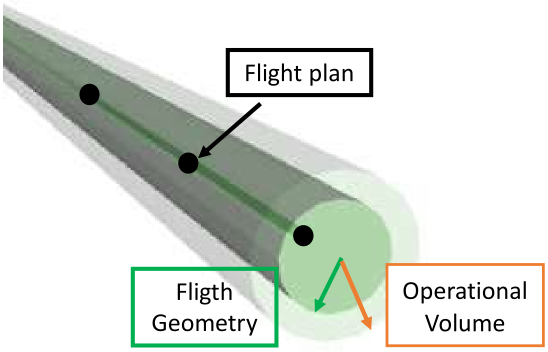

Figure 4.4: Graphical representation of the OV of a UAS operation (the orange arrow represents its radius). Given a flight plan, the green cylindrical volume around represents its Flight Geometry (the green arrow indicates its radius), and the OV also includes an outer volume that is the Contingency Volume.

to account for environmental or performance uncertainties. The closest distance between the current UAS position and its flight plan is computed to determine whether the UAS is out of its OV.

In addition, this module monitors possible intrusions in geofences. For that, every waypoint belonging to the predicted trajectory of each UAS is compared against the active geofences, to determine whether the UAS is already intruding a geofence or it is estimated to enter one in a short future time. This check is carried out in 4D, i.e., the 3D volume of the geofence and its activation time are taken into account. More specifically, apart from checking the waypoint altitude with the minimum and maximum altitudes of the geofence, an evaluation on the horizontal plane is done depending on the shape of the geofence. If it is cylindrical, the distance of the given waypoint to the cylinder center is computed and compared with the geofence radius. If the geofence is defined by a polygonal shape, the *signed angle* method is applied. More details can be seen in Capitan et al. (2021).

Finally, the Monitoring module checks whether there is any loss of separation between UAS that needs to be notified. This check is done with a geometrical approach whose details can be seen in Acevedo et al. (2020).

Basically, the idea is to discretize the airspace to model it as a 4D grid (see Figure 4.5), where each cell represents a 4D volume in space and time (dX, dY, dZ, dT) and stores a list of all the UAS whose trajectory is estimated to be inside. Thus, each waypoint of a

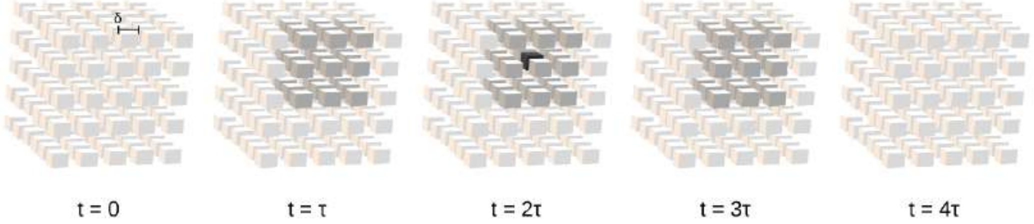

Figure 4.5: A 4D grid representation of the airspace. The dark grey cells would be the neighboring cells of the black cell (Acevedo et al., 2020).

UAS trajectory only needs to be compared with other waypoints within the neighboring cells (space and time neighborhood). For each waypoint in the 4D grid, the distances to the waypoints in the lists of its neighboring cells are calculated. If any of these distances is shorter than a safety distance, a threatening event of loss of separation will be reported.

4.4.5 Emergency Management

The Emergency Management module is the component of the UTM system that handles the unexpected situations in the U-space ecosystem. The module centralizes all the information related to the events that may become a threat, either due to conflicting UAS operations or to external warnings (e.g., a jamming attack or a bad weather situation). After analyzing the events, the EM determines which are the recommended actions to resolve the detected conflicts, and it sends them to the corresponding UAS operators.

The EM is a central module in the UTM architecture and, as such, it interacts with the Monitoring, the USM, the DB, and the TD. The possible threats or conflicts are notified to the EM by the Monitoring or the USM modules. The former reports about conflicts related with the UAS flight plans, as it was explained in Section 4.4.4. The latter reports about external warnings coming from UAS technical issues, UAS operators, or auxiliaries stakeholders in the U-space. For instance, this is the case of a jamming attack, a bad weather forecast, the declaration of a wildfire by the fire brigades, or any other threatening event notified by emergency corps.

Depending on the severity of each threat, the EM executes a decision-making procedure to determine the best possible actions to solve the conflict (Capitán et al., 2019; Capitan et al., 2022). In this procedure, the EM takes into account the current flight plans for the involved UAS, the priority of their operations, and other restrictions in the airspace like the geofences. As output, the EM can decide to take three different types of actions: (i) to send a specific command to a particular UAS to terminate the flight, to go back to the flight plan, etc.; (ii) to create a geofence to isolate the detected threat; and (iii) to propose an alternative flight plan to one or several UAS to resolve the conflict. For the computation of these alternative plans, the EM receives the support of the TD module, which computes alternative routes by applying a set of predefined maneuvers for each UAS. More details about the decision-making procedure implemented by the EM can be seen in Chapter 5.

Finally, it is important to remark that all actions sent by the EM to the UAS are just recommendations. According to the current regulation of the U-space ecosystem, the UTM can only suggest automatically possible correction actions, but those must be accepted or rejected by each UAS operator eventually. Nonetheless, our approach would be able to accommodate a UTM system where the whole process were executed autonomously without the need for human intervention, which is the final objective in the U-space framework.

4.4.6 Tactical Deconfliction

The Tactical Deconfliction module provides support to compute alternative flight plans for UAS that need to resolve a potentially threatening or conflicting situation. The TD receives requests from the EM indicating the necessary information related to the event to solve, i.e., the type of threatening situation and the data of the affected operations and the active geofences. Depending on the situation, the TD will attempt different types of maneuvers to generate a list of alternative flight plans for the involved UAS. For each possible solution, the TD will compute an associated cost and riskiness level, which will be reported back to the EM, together with the generated alternative flight plans. Then the EM is the module that makes a final decision about the best solution.

The specific algorithm for tactical deconfliction is out of the scope of this thesis, and different alternatives could be integrated within the presented U-space architecture. In particular, we used an implementation for the TD module developed within the framework of the GAUSS project. This TD uses two different approaches to compute the alternative routes, depending on whether the threat is a conflict between different UAS or a situation with a single UAS involved. The first case occurs when the flight plans of several UAS are in conflict, e.g., due to a loss of separation. In that case, a geometric approach based on repulsive forces is used to modify the original flight plans. The details of the implemented algorithm can be seen in Acevedo et al. (2020), but it basically models the UAS trajectories as cords with electrical charges that repel each other, in order to increase their separation. By applying vertical or horizontal separation maneuvers between the involved UAS trajectories in an iterative procedure (see Figure 4.6), the TD can generate several alternative solutions. The priorities of the conflicting flight plans are also considered. The algorithm tends not to modify the flight plans of those UAS whose operations present a higher priority in the U-space. Even though these types of conflicts are solved in an iterative manner, by applying the tactical deconfliction procedure for each pair of UAS sequentially, the final solution could still produce additional conflicts with geofences. In this case, the Monitoring module would report those new pending conflicts in subsequent iterations.

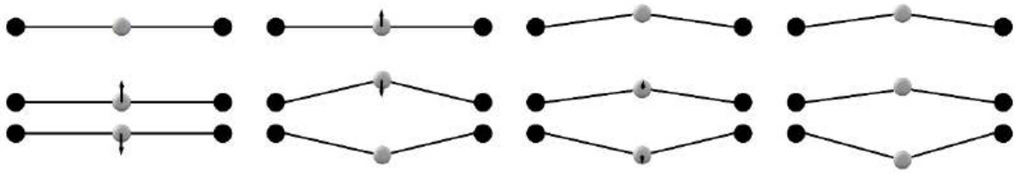

Figure 4.6: Iterative procedure to solve a conflict in the case of a loss of separation (from left to right). The flight plans of the two lower UAS are in conflict and need to be separated. Then, the middle UAS enters in conflict with the upper UAS, so these two get separated again. As the plan of the middle UAS gets modified, the lowest UAS is also adapted to achieve a final solution without loss of separation (Acevedo et al., 2020).

A second approach is used to solve situations with a single UAS involved. This is the case of a UAS that presents a technical problem, that is out of its Operational Volume or that has a conflict with a geofence. In all those cases, a heuristic path planner based on the well-known A* algorithm is used. First, if the UAS flight plan goes through a geofence, the path planner generates an alternative route avoiding that geofence. Second, if the UAS is already within a geofence, it gets out of the geofence through an *escape* point and then, it avoids the geofence to resume with its flight plan afterwards. Third, if a UAS is out of its Operational Volume, two alternative routes are computed: one from the current

UAS position to the closest point of its flight plan; and another from the current UAS position to its next waypoint in the flight plan, regardless of how long the UAS remains out of its Operational Volume. In the three cases, an alternative route to return back to the home station is also computed. The EM could select this option if all the other solutions to continue with the operation are too risky.

4.4.7 Discussion

In this section, we discuss the functionalities implemented by the U-space services of our UTM architecture, when compared to those expected in the current definition of the U-space ecosystem. For that, we have summarized in Table 4.3 the expected functionalities to be covered by each of the U-space services included in our system, according to the bibliography studied in Section 4.2. In the following, we discuss which capabilities are already covered by our system and the missing points for future implementations.

U-space service	Functionalities	Covered
Tracking	Cooperative UAS tracking	✓
	Non-cooperative UAS tracking	✗
	Capability to exchange data with other services	✓
	Real-time tracking with data fusion from multiple sources	✓
	Tracking data recording	✓
Monitoring	Air situation monitoring	✓
	Non-cooperative UAS identification	✗
	Flight non-conformance detection	✓
	Restricted area infringement detection	✓
	Provision of traffic information for UAS operators	✗
	Conflict alerts	✓
Emergency Management	Emergency alerts	✓
	Provision of assistance information for UAS operators	✓
Tactical Deconfliction	Transmission of deconfliction information from the USM to the UAS	✗
	Transmission of deconfliction information in real time	✓

Table 4.3: Summary of the functionalities to be covered by the U-space services included in our UTM system.

- *Tracking.* This service is supposed to consider cooperative and non-cooperative UAS, but our current implementation only manages cooperative UAS. This is because we have focused on enabling automated decision-making for the operating UAS, which makes no sense for non-cooperative vehicles. Those should be treated as uncontrollable intruders (i.e., threats) in the airspace. However, our Tracking module does have the capability to update and record data in real time from different sources. Other services can also access these data through the DB module if needed.

- *Monitoring.* As in the previous case, our current implementation does not consider non-cooperative UAS. We did not establish a specific communication link to provide traffic information to the UAS operators either, though this could be easily done through the USM. However, our Monitoring module does accomplish all the other expected functionalities, i.e., it detects and alerts in real time about conflicts related to flight non-conformances, geofences, and inter-UAS separation.

- *Emergency Management.* This service is expected to provide the UAS operators with notifications about alerts and any other emergency assistance. Besides, our EM module includes automated decision-making capabilities, in order to manage threats in real time by proposing safe and optimal actions to the UAS.

- *Tactical Deconfliction.* Although this service is supposed to provide deconfliction information to the UAS operators through the USM, in our scheme this role is played by the EM module. This is because the automated decision-making capability is implemented in the EM module, which uses the TD module to get support generating possible alternative plans. The EM is then the one in charge of deciding the best option for real-time deconfliction.

4.5 Experiments

This section contains experimental results to showcase the capabilities of the proposed UTM system. The objectives of these experiments are twofold: (i) we show the integration of the complete architecture, with all its functional modules interacting together to accomplish the specified UAS operations; and (ii) we demonstrate our system operating in real time in a realistic setup, to test its capabilities to solve different types of conflicts in an automated manner. For that, we have defined two use cases (Section 4.5.1) involving heterogeneous UAS and several types of conflicts, in order to validate all the modules in our UTM system. The tested use cases are realistic both in terms of the UAS operational parameters and the experimental setup (Section 4.5.2). Our experiments were carried out by means of HITL simulations where the UAS operators and the UTM framework ran at different physical locations, with a real long-distance communication link in between. All the results of the tests are described in Section 4.5.3.

4.5.1 Use cases definition

We defined two use cases using simulated versions of the heterogeneous UAS that are depicted in Figure 4.7: the multirotor DJI M600 and the fixed-wing Atlantic I. These UAS are used in the GAUSS project to run tests integrating aircraft with different maneuverability and different proprietary autopilots. Both use cases involve a pair of UAS performing operations with different or equal priorities, and both require the interaction of all the modules of the proposed UTM system.

Figure 4.7: The Atlantic I (left) and DJI M600 (right) UAS are used to validate the UTM functionalities developed in the GAUSS project.

Figure 4.8 depicts a top view of each use case, with the corresponding initial flight plans. Table 4.4 summarizes the operational parameters for the use case 1. UAS_1 is a multirotor performing an operation for precision agriculture, while UAS_2 is a fixed-wing aircraft that

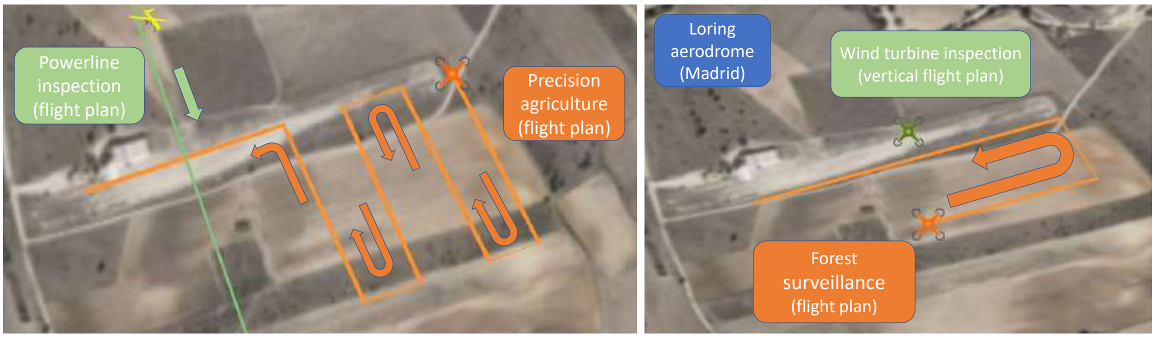

Figure 4.8: Top views including the initial flight plans of the use case 1 (left) and the use case 2 (right). All the operations were planned in an area of the Loring aerodrome in Madrid (Spain).

has to inspect an electrical powerline. Given its easier maneuverability, the priority of the UAS$_1$ operation is set lower. The initial flight plans (see Figure 4.8, left) are such that the UAS do not coincide in space and time throughout their operations. However, we simulated an unexpected delay in the start of the UAS$_1$ operation, which resulted in a later violation of the minimum safety distance between both UAS. Thus, this use case is used to test how the UTM is able to detect a loss of separation between the UAS and to perform real-time tactical deconfliction for an inter-vehicle conflict, deciding new flight plans for both UAS.

	Operation 1.1	**Operation 1.2**
ConOps	Precision agriculture	Powerline inspection
UAS type	M600 (UAS$_1$)	Atlantic I (UAS$_2$)
Cruising speed	3.3 m/s	30 m/s
Altitude (Above Ground Level)	40 m	100 m
Operation priority	Low	High
Events involved	Loss of separation	Loss of separation

Table 4.4: Operational parameters for the use case 1.

Table 4.5 summarizes the operational parameters for the use case 2. In this case, both UAS$_1$ and UAS$_2$ are multirotors, performing two operations with equal priority. In their initial flight plans (see Figure 4.8, right), UAS$_1$ moves on a vertical line to accomplish the inspection of a wind turbine, while UAS$_2$ has to fly on a horizontal plane to surveil a nearby

	Operation 2.1	**Operation 2.2**
ConOps	Wind turbine inspection	Forest surveillance
UAS type	M600 (UAS_1)	M600 (UAS_2)
Cruising speed	1 m/s	1 m/s
Altitude (Above Ground Level)	30-90 m	70 m
Operation priority	High	High
Events involved	Jamming attack	Geofence conflict

Table 4.5: Operational parameters for the use case 2.

forest. During the operation, a jamming attack is simulated over UAS_1. The objective of this use case is to test how the UTM is able to react in an automated manner to an emergency generated by an external source, creating a new geofence and adapting the conflicting flight plans.

4.5.2 Experimental setup

We have developed our UTM system architecture in ROS (Kinetic version) and the software is available online [6]. First, we used an airspace SITL simulation based on Gazebo Millan-Romera et al. (2019) for system integration and preliminary tests. Then we setup a realistic environment to run experiments in real time with HITL simulations. These experiments were carried out within the framework of the GAUSS project, with the configuration depicted in Figure 4.9.

The company EVERIS [7] ran on its headquarters in Madrid (Spain) a Remote Pilot Station (RPS) for each type of UAS. Each RPS has an integrated HITL simulation producing real-time telemetry data for the operating UAS, a graphical user interface to show this telemetry and the operational information to the safety pilot (*RPS Client Application*), and an *RPS MQTT Broker* to communicate data over the Internet. The RPS Client Application was developed by the company SATWAYS [8] and it can be seen in Figure 4.10. Simultaneously, we ran our UTM system on a server located in Seville (Spain), connected to the Internet via a *ROS MQTT bridge*. The UAS RPS communicated with the remote UTM system

---

[6] https://github.com/grvcTeam/gauss

[7] https://www.everis.com/global/en

[8] https://www.satways.net

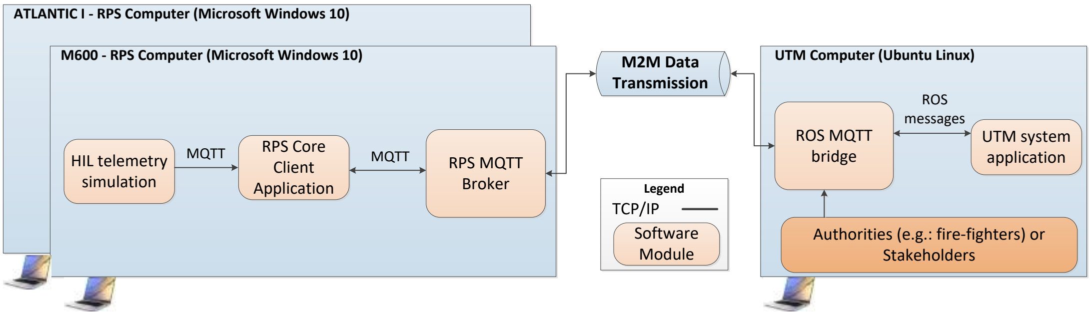

Figure 4.9: Setup for the experiments. The computers running the RPS for the two UAS and the UTM system were placed at remote locations and communicated through the Internet via the MQTT protocol.

exchanging JSON (JavaScript Object Notation) messages sent over the MQTT (Message Queuing Telemetry Transport) transport protocol [9]. Moreover, time synchronization for the exchanged data between the remote computers was achieved thanks to an NTP (Network Time Protocol) server. It is important to highlight that this experimental setup is close to the real U-space ecosystem, where the UTM system would be running on a server located at a remote distance of the UAS operators.

Figure 4.10: Screenshots of the graphical user interface developed by SATWAYS running on the RPS Client Application.

---

[9] We used the open-source Apache Active MQ broker.

4.5.3 Results

In this section, we present results of the experimental tests for the two proposed use cases [10]. It is important to highlight that the experiments were carried out in real time, with the UTM system monitoring the operations and managing the unexpected events properly. Moreover, the proposed solutions to solve the conflicts were executed in an automated manner by the simulated UAS, and supervised by human safety controller.

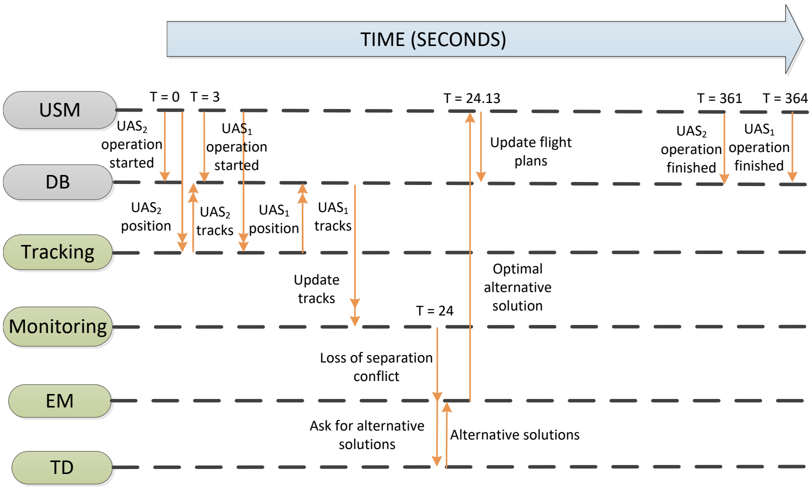

Figure 4.11: Timeline of the experiment of the use case 1, where a loss of separation event is resolved. Single arrows indicate isolated interactions between modules, whereas double arrows indicate periodic communication.

Figure 4.11 shows a timeline for the experiment of the use case 1. Both UAS were supposed to start their operations simultaneously ($t = 0\ s$) according to their pre-flight generated plans, without conflicts. However, we simulated a delay of 3 seconds in the start of the UAS$_1$ operation. The Tracking module received periodically positioning information from both UAS and it updated the DB accordingly. The Monitoring module checked for conflicts periodically using the updated tracks from the DB and, at $t = 24\ s$, it detected a future loss of separation conflict between the UAS. This was communicated to the EM, which ran an automated decision-making process (supported by the TD) to propose the optimal conflict resolution. In this case, an alternative flight plan was sent to UAS$_1$ through

---

[10] An illustrative video with the use cases can be seen at `https://www.youtube.com/watch?v=2hEpPcUP4qs`

the USM module. In real operations, there would be a delay between the autonomous decision-making and the communication of the new flight plans, as the UAS operator needs to confirm them. As this validation experiment is simulated, we did not considered that response time.

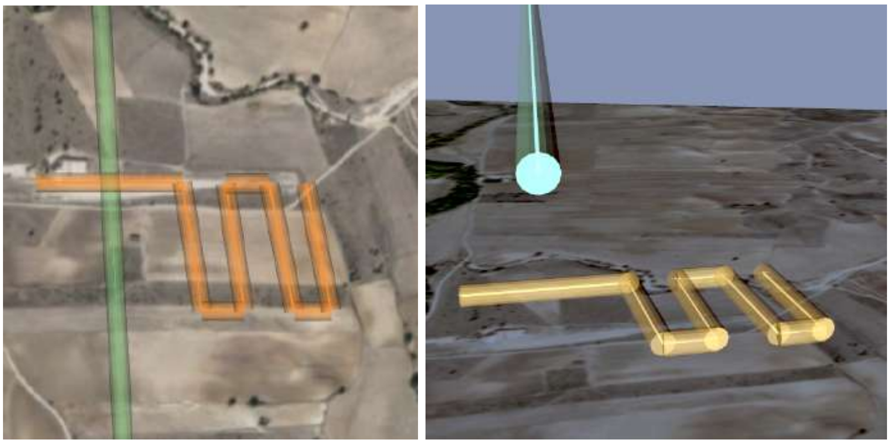

Figure 4.12: A top (left) and a perspective view (right) of the initial flight plans in the use case 1. The Operational Volumes are shown for both UAS. There are no conflicts given the UAS 4D trajectories.

Figure 4.12 shows the initial flight plans for the UAS and their reserved Operational Volume. Despite not having conflicts initially, the delay in the UAS$_1$ operation provoked an eventual loss of separation in the last part of its operation, which was resolved with an alternative flight plan.

Figure 4.13 depicts the three options generated by the TD module and the optimal solution (in terms of risk and traveled distance) selected by the EM. In the experiment, the conflict was detected by the UTM system well in advance and the total time between the detection and the communication of a solution to the USM took 0.13 seconds.

Figure 4.14 shows a timeline for the experiment of the use case 2. Both UAS started their operations simultaneously ($t = 0\ s$) following pre-flight plans without conflicts. The Tracking module received periodically positioning information from both UAS and it updated the DB accordingly. The Monitoring module checked for conflicts periodically using the updated tracks from the DB. We simulated a jamming attack over UAS$_1$ ($t = 12\ s$) that was notified by the USM to the EM, which ran an automated decision-making process.

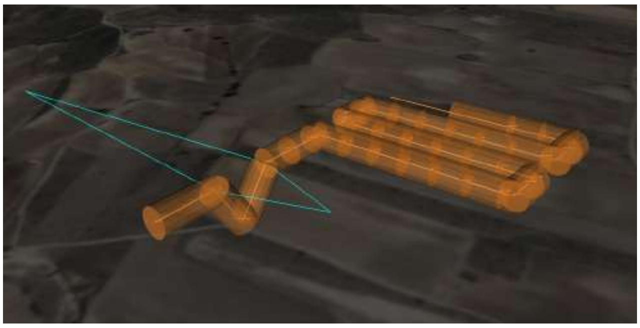

Figure 4.13: A perspective view of the conflict resolution in the use case 1. A new flight plan for UAS$_1$ (with a final *go down* maneuver) was selected to keep the safety distance with UAS$_2$. The other alternative maneuvers generated by the TD module (*go left* and *go right*) are also shown.

In this type of threat, due to the involved risks, the EM decided to suspend the UAS$_1$ operation (notifying the USM) and to create a geofence around (updating the DB). Then the Monitoring module detected ($t = 15\ s$) a future geofence conflict with the UAS$_2$ flight plan, which was resolved by the EM (with the support of the TD) with an alternative plan avoiding the geofence. Again, the time between the detection of the conflict and the communication of the optimal solution to the USM was less than 1 second. Again, as this validation experiment is simulated, we did not considered the delay caused by the confirmation of the new flight plans by the UAS operator. Figure 4.15 shows the initial flight plans for the UAS and their reserved Operational Volumes, and the situation right after the jamming attack. Despite not having conflicts initially, the creation of a new geofence provoked an eventual conflict, which was resolved with an alternative flight plan for UAS$_2$ (see Figure 4.16).

Finally, it is important to recall that the experiments were carried out with a setup where the UTM system ran at a remote distance of the UAS stations. Despite that, the communication delays and response times by the UTM system were adequate for a real-time resolution of the unexpected conflicts. In particular, we measured a reception of the UAS telemetry data at the USM at an average rate of $1\ Hz$ with a maximum delay of $40\ ms$.

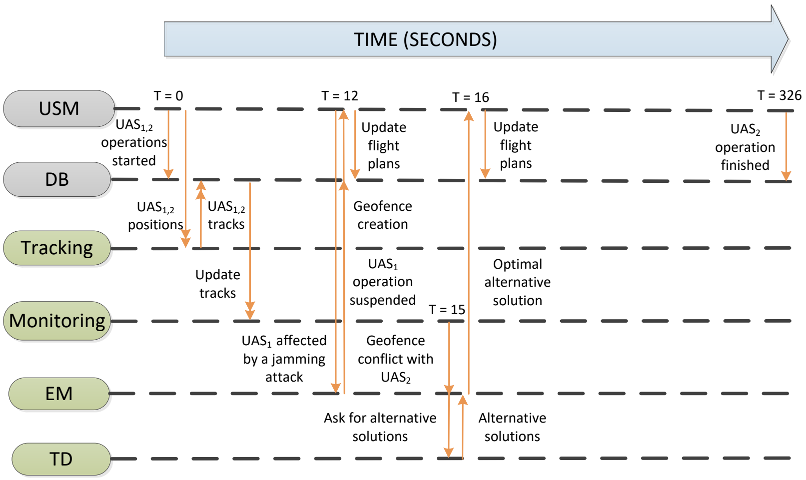

Figure 4.14: Timeline of the use case 2, where a jamming attack and a geofence conflict are resolved. Single arrows indicate isolated interactions between modules, whereas double arrows indicate periodic communication.

4.6 Conclusions

In this chapter, we have presented a UTM system architecture framed within the U-space ecosystem. Our software architecture is flexible and general, and it is built as an open-source solution that could be easily extended with additional U-space functionalities. Nonetheless, we have focused on in-flight services for automated threat management and conflict resolution, which is a major gap in the current state of the art. In our realistic experimental setup, with the involved systems running HITL simulations communicated through a remote link with the UTM system, we have demonstrated that the proposed UTM solution is capable of managing unexpected events in real time, proposing solutions in an automated manner. In our experiments, the system was able to detect and resolve different types of conflicts, reasoning about 4D UAS trajectories and Operational Volumes. Besides, we have tested the feasibility of the system for the future U-space, integrating heterogeneous types of UAS (fixed and rotary wing), heterogeneous positioning technologies (ADS-B and telemetry from different autopilots), and a database to keep track in real time of the different UAS operations and geofences.

Figure 4.15: Left, top view with the initial flight plans in the use case 2. The Operational Volumes without conflicts are also shown. Right, situation after the detection of the jamming attack. A geofence (in red) is created around the attacked UAS, which generates a conflict with the flight plan of the other UAS.

Figure 4.16: A top (left) and a perspective (right) view of the optimal solution in the use case 2. An alternative flight plan for UAS_2 is generated to avoid the geofence.

Chapter 5

Threat Management Methodology for Unmanned Aerial Systems operating in the U-space

This chapter presents a methodology for threat management in multi-UAS operations in the U-space. This is carried out by implementing the U-space service Emergency Management, together with the Tactical Deconfliction service, within the UTM architecture described in Chapter 4. First, we review other relevant works about emergency management and multi-UAS conflict resolution (Section 5.1). Second, we describe the methodology for threat management (Section 5.2). After providing a general overview (Section 5.2.1), we identify a generic set of threats that may occur during UAS operations in the U-space (Section 5.2.2). Then our methodology is based on proposing a set of mitigation actions that are evaluated in terms of cost and risk level, in order to take optimal decisions in an autonomous fashion (Section 5.2.3). The method is able to make decisions in real time, and it is flexible enough to accommodate additional threats or mitigation actions in the future. Finally, the chapter also includes a demonstration of the functionalities of our method for threat management in real flight tests (Section 5.3) using actual fixed- and rotary-wing UAS (see Figure 5.1), through an experimental campaign in the ATLAS Test Centre [1].

---

[1] http://atlascenter.aero

Figure 5.1: Left, the team of fixed- and rotary-wing UAS used to demonstrate our approach for threat management. Right, the ATLAS test facilities (Spain) where the experiments were carried out.

5.1 Related work

In this section, we review the related work on emergency management and conflict resolution methodologies. Even though we focus on UAS, the threats commonly handled by manned aircraft and RPAS can be taken as a good starting point to study unexpected events in the airspace and how to manage them. In European Aviation Safety Agency (EASA) (2016), the characteristics of airborne conflict occurrences are detailed. Manned aviation controller guidelines to handle emergency situations are also presented in Considine (2003). Moreover, Finke (2016) describe a set of common RPAS specific emergency situations and derive corresponding contingency measures whenever feasible. They study the usability of existing procedures and standards coming from manned aviation, and then they extend some cases to unmanned aviation or RPAS (e.g., electrical failure or navigational failure). Regarding UAS, the EU-funded project CORUS [2] defined a new approach for the threat concept framed in the U-space. A threat is considered an unexpected event that may happen and cause harm when UAS are operating in the U-space. This project has published an exhaustive list of possible threats and events that may happen during a UAS operation in the U-space (CORUS, 2019).

In the literature, there are multiple works that address threat management in the airspace for UAS, but mainly focusing on particular types of threats. They can be split

---

[2] https://www.sesarju.eu/projects/corus

into two main categories: (i) approaches to cope with emergency events that cause malfunctioning UAS; and (ii) those dealing with conflict resolution, with other vehicles or no-fly zones. Within the first category, East Gippsland Shire Council (2017) propose some procedures to address certain events of malfunctioning UAS (e.g., a motor failure, a GPS failure, or a loss of orientation). The specific failure of loss of command and control communication link is considered in Fern et al. (2014). In Pastor et al. (2012), a structured approach to classify contingency sources and select contingency reactions depending on the severity is developed. Also, many works manage these malfunctioning situations by means of emergency landing operations (Masri, 2017; Mejias and Eng, 2013; Guo et al., 2014; Mejias L. and J., 2014; Warren et al., 2015; Ten Harmsel et al., 2017; Atkins et al., 2015). For example, works in Masri (2017); Mejias and Eng (2013) center on landing operations in the case of unpowered UAS. The detection of safe landing zones can be done using machine learning techniques (Guo et al., 2014) or vision-based approaches (Warren et al., 2015). An overview of automated emergency landing systems can be found in Mejias L. and J. (2014). In Warren et al. (2015), it is presented a guidance, navigation, and control method for an automated emergency landing system with a fixed-wing UAS. Ten Harmsel et al. (2017) propose a meta-level emergency landing planner to calculate safe paths for small UAS when low-energy reserves are detected unexpectedly while flying over populated urban environments. Moreover, an emergency management architecture has also been presented for piloted or autonomous aircraft in Atkins et al. (2015). They design and implement an adaptive flight planner that dynamically computes feasible flight plans in response to events that degrade aircraft performance.

Regarding the second category, conflict resolution can be approached in a pre-flight (strategic deconfliction) or in-flight phase (tactical deconfliction). Pre-flight solutions usually formulate the problem as multi-agent path finding. Ho et al. (2022) propose a priority-based method and a negotiation method to solve this problem in a distributed manner, assuming the existence of multiple U-space service providers. In Doole et al. (2022), a heavily constrained urban airspace with a high density of UAS traffic is tackled. They apply the one-way street concept plus heading/altitude rules to segment the airspace, and delay- and speed-based actions to resolve conflicts. An approach for the *Dynamic capacity management* service defined in U-space is implemented in Tang et al. (2022).

Pre-flight UAS planning is enhanced with a dynamic reconfiguration algorithm, to balance airspace allocation by rescheduling alternative trajectory options to route away from possible congested areas.

In terms of in-flight conflict resolution, see-and-avoid (Mcfadyen and Mejias, 2016) and velocity-obstacle methods (Alonso-Mora et al., 2015) have been traditionally used for UAS collision avoidance. More recently, the use of 4D bubbles for conflict management has been applied in the U-space (Dubot and Joulia, 2021). Moreover, a method to predict conflicts and adapt the velocity vectors to avoid them has been proposed (Jover et al., 2021). This method has also been extended (Jover and Casado, 2022) to consider the operation priorities established in the U-space policy. Other works present high-level architectures for U-space. In Campusano et al. (2021), a software modular design is introduced, but not focusing on decision-making procedures. Lappas et al. (2022) do implement a U-space architecture for conflict resolution, focusing on functionalities which involve the use of vehicle-to-vehicle and vehicle-to-infrastructure technologies for communication between UAS and operators.

Most of these previous works address threat management focusing on specific types of threats, either emergency situations or inter-vehicle conflicts. The main contribution of our work is proposing a holistic methodology that considers multiple kinds of threats and mitigation actions in an integrated decision-making procedure. We selected a set of common threats and mitigation actions in UAS airspace operations, but the framework is general enough to accommodate additional ones in the future. A second contribution is that our framework is integrated within the U-space initiative. On the one hand, the considered threats, mitigation actions, and airspace constraints, are in line with those defined in the U-space. On the other hand, our implementation is based on the actual U-space services and has been tested within a software architecture replicating them (see Chapter 4). Last but not least, most existing works, except for the one in Lappas et al. (2022), only provide results in simulation. There have also been recent field trials to test conflict resolution procedures in civil airspace (Alarcon et al., 2020). However, this work reproduces predefined maneuvers instead of performing real-time decision-making. Instead, we demonstrate our methodology working on illustrative use cases with actual UAS and an actual U-space software architecture. In summary, we take ideas from previous works to define sets of relevant threats and mitigation actions in U-space operations, and then we develop a generic framework for autonomous real-time threat management. Our methodology accepts threats of multiple types and, according to a certain categorization, feasible mitigation actions are evaluated to make optimal decisions.

5.2 Methodology for threat management

In this section, we describe our methodology for threat management in the U-space. First, we provide an overview of the problem and our solution. Then we identify and provide a description of the threats that may occur during multi-UAS operations in the U-space. Finally, we describe our decision-making procedure to apply the corresponding mitigation actions.

5.2.1 Overview

We consider a set of UAS operating in a common airspace, where each operation has a predefined *priority* established by the operator. These priorities are later used to decide which UAS will modify its flight plan in case of conflict (if two UAS are in conflict, the least critical varies its plan).

Then we consider different threats that may happen during the in-flight phase of the operation, from a predefined set of types. Each type of threat has associated a *severity* level, which will determine the kind of mitigation actions to apply. Given the operations in course and the detected threats, our problem is to decide the best mitigation actions to apply to each UAS, in order to manage all threats and solve the existing conflicts at the same time that the U-space constraints are held.

Figure 5.2 depicts an overview of the elements considered by our methodology: in orange, the different types of threats; in red, the U-space constraints to be taken into account; in blue, the possible mitigation actions; and in green, the U-space services involved. Once we have a list of detected threats and their types (see details in Section 5.2.2), this information, together with the active U-space constraints, is input to a decision-making procedure that selects the optimal mitigation actions for the required UAS. A

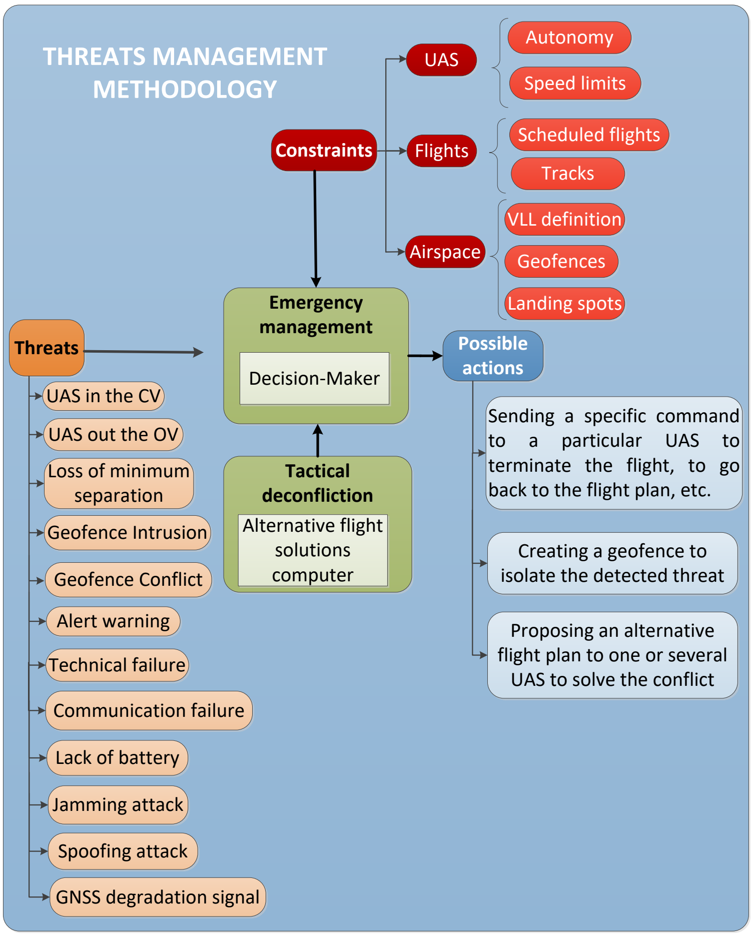

Figure 5.2: Overview of our threat management methodology.

multi-objective optimization is carried out in order to select actions maximizing efficiency and safety. The different mitigation actions and the selection procedure will be described in Section 5.2.3. Our threat management methodology is applied in real time, using in-flight U-space services. In particular, the decision-making procedure is implemented through the Emergency Management service, which uses the Tactical Deconfliction service to compute alternative flight plans when needed.

5.2.2 Threat types

Reviewing the literature on threat management (see Section 5.1), we have identified a list of relevant threats that cover most of the unexpected events that may occur during multi-UAS operations in the U-space. In the following, we describe the different types of threats that we consider:

- *UAS within its Contingency Volume.* The UAS is out of its Flight Geometry but still within its Contingency Volume. In this situation, the UAS is considered under control, because it is still within the Operational Volume, but minor mitigation actions could still be applied so that it returns to its Flight Geometry.
- *UAS out of its Operational Volume.* In this situation, the UAS is considered out of control, as it is flying out of its Operational Volume. Therefore, a mitigation action will be required to solve this occurrence.
- *Loss of minimum separation.* In the U-space, there is a minimum safety distance between each pair of UAS, which is determined by the sum of the radii of both Operational Volumes. If two UAS are closer than this safety distance, a mitigation action will be necessary to avoid a potential collision.
- *Geofence intrusion.* This happens when a UAS enters a geofence, i.e., a forbidden 4-dimensional volume (e.g., a static no-fly zone specified by the authorities before operation or a restricted area dynamically created by the U-space during operation). In this situation, a mitigation action to leave the volume will be mandatory.
- *Geofence conflict.* This happens when a UAS detects along its flight plan a geofence that was not planned to be there. In this case, the UAS should avoid entering that geofence and then resume its operation.
- *Alert warning.* Authorities (e.g., fire-fighters, emergency corps, etc.) or stakeholders could notify a wildfire, a bad weather forecast, or any other threatening event. Those occurrences should be managed with the corresponding actions.
- *Technical failure.* A technical failure is an unwanted error of technology-based systems. In the case of UAS, this can involve hardware or software components.

- *Communication failure.* This entails a loss of communication between the UAS and the U-space Service Provider, which is an event of difficult solution. Emergency actions will be taken in order to mitigate potential damages.
- *Lack of battery.* This event implies the impossibility of ending the UAS operation ordinarily.
- *Jamming attack.* A jamming attack consists of an attempt to jeopardize the GNSS signal of a UAS.
- *Spoofing attack.* A spoofing attack is a situation in which a malicious person or software fakes the UAS information, e.g., so that it seems to be located somewhere else, instead of at its right location. This kind of threat is rather difficult to detect.
- *GNSS degradation signal.* In an era of increasing wireless radio frequency congestion, GNSS systems are becoming more at risk of signal degradation due to interference. GNSS signal deterioration typically occurs by signal masking caused by natural (e.g., foliage) and man-made (e.g., buildings) obstructions, ionospheric scintillation, Doppler shift, and antenna effects. This degradation could result in partial or total loss of the UAS tracking.

It is worth noticing that, although we focus on the previous list of threats (some of them are tested in Section 5.3 with field trial use cases), we believe those categories are general enough to accommodate all possible U-space events. Additional events could fit in some of the mentioned types, as they would produce similar effects on UAS flight plans.

5.2.3 Decision-making procedure

This section describes our decision-making procedure to select the best mitigation actions for each UAS. In terms of U-space architecture (Capitan et al., 2021), this decision-making procedure is implemented within the Emergency Management (EM) service. The EM component is in charge of centralizing all information related to the events that may become a threat, and applying the corresponding mitigation actions. Besides, if an alternative flight plan needs to be computed for the UAS in conflict, the EM relies on the support of the Tactical Deconfliction (TD) service, which is a U-space component providing non-conflicting flight plans.

Field type	Field name	Description
integer	threat_type	Threat type, as described in Section 5.2.2
integer	threat_id	Each new threat has a unique identifier
integer[]	uav_ids	List with identifiers of the threatened UAS
integer[]	geofence_ids	List with identifiers of the geofences involved
integer[]	priority_ops	Priorities of the operations involved in the threat
4D waypoint	location	Waypoint (x,y,z,t) where the threat was detected

Table 5.1: Specified data for each threat.

Source	Name	Description
UAS	Autonomy	Maximum distance the UAS can still flight
	Speed limit	Maximum UAS speed
Flights	Scheduled flights	Flight plans of all UAS
	Tracks	UAS positions in real time
Airspace	VLL definition	Volume of air below $150\ m$ above ground level
	Geofences	4D volumes (x,y,z,t) for each restricted flight zone
	Landing spots	Waypoints in (x,y) where UAS could land

Table 5.2: U-space constraints.

In the decision-making procedure (see Algorithm 1), the EM takes as input the information of each detected threat, as specified in Table 5.1, together with the U-space constraints described in Table 5.2. As output, the EM can decide to take three different types of mitigation actions defined in Table 5.3: **type A**, to send a specific command or notification to a particular UAS operator, e.g., flight termination, going back to the flight plan, alert warning, etc.; **type B**, to create a geofence to isolate the detected threat; and **type C**, to propose an alternative flight plan to one or several UAS operators for solving a conflict. In case several threats are simultaneously detected, they are solved in order of decreasing severity (lines 1-2 of Algorithm 1). The severity is defined as the level of damage that a threat can cause in the airspace (e.g., in principle, the damage that a spoofing attack can cause is bigger than that of a UAS which leaves its Flight Geometry (FG) volume). The severity level of each type of threat is manually determined by U-space operators.

Algorithm 1: Decision-making procedure

```
   Input: $\mathcal{T} \leftarrow$ list<threat>,
          $\mathcal{C} \leftarrow$ uspace_constraints
1  $\mathcal{S} \leftarrow$ obtain_severities($\mathcal{T}$)
2  $\mathcal{T} \leftarrow$ sort($\mathcal{T}$,$\mathcal{S}$)
3  foreach $th$ in $\mathcal{T}$ do
4      $Type \leftarrow$ action_type(th.threat_type)
5      if $Type == A$ then
6          $a \leftarrow$ newAction(th,$\mathcal{C}$,TYPE_A)
7          sendAction($a$)
8      else if $Type == B$ then
9          $a \leftarrow$ newAction(th,$\mathcal{C}$,TYPE_A)
10         $b \leftarrow$ newAction(th,$\mathcal{C}$,TYPE_B)
11         sendAction($a$,$b$)
12     else if $Type == C$ then
13         $\mathcal{M} \leftarrow$ TD($th$,$\mathcal{C}$)
14         $\xi \leftarrow$ bestManeuvers($\mathcal{M}$)
15         foreach $i$ in $N$ do
16             $c \leftarrow$ newAction(th,$\mathcal{C}$,TYPE_C,$\xi[i]$)
17             sendAction($c$)
18         end
19     end
20 end
```

Regarding the implementation of the mitigation actions, in the actions of type A, the EM acts just sending a command or notification to the corresponding UAS operator through the U-space communication layer (line 7 of Algorithm 1). In the actions of type B, the EM creates a new geofence that will be stored in a database of the U-space architecture. This database is in charge of storing all updated operational information related to both UAS and geofences. Besides creating the geofence, a command or warning is also sent to the UAS (line 11 of Algorithm 1). In the actions of type C, the EM sends alternative flight plans to the UAS involved in a conflicting situation. The EM asks the TD module for alternative flight plans, providing information related to the threat to solve (i.e., the type of threatening situation, the data of the affected UAS operations, and the active geofences). Then the TD attempts different types of maneuvers, selected from those in Table 5.4, to generate a list

Mitigation action type	Description
A	Sending a specific command or notification to a particular UAS operator, e.g., flight termination, going back to the flight plan, alert warning, etc
B	Creating a geofence to isolate the detected threat
C	Proposing an alternative flight plan to one or several UAS operators for solving a conflict

Table 5.3: Definition of the possible mitigation actions proposed by the methodology.

of alternative flight solutions for the involved UAS (line 13 of Algorithm 1). Last, the EM chooses the optimal solution among the possible alternatives according to a multi-objective optimization problem (line 14 of Algorithm 1), and each of the UAS involved is notified its new flight plans (line 17 of Algorithm 1). It is important to remark that, although our methodology has the ability to operate autonomously, the current regulatory restrictions do not allow to operate UAS in a totally autonomous manner. Human supervision for accepting or rejecting the alternative flight plans is still mandatory. Nonetheless, we expect more flexibility in the near future in terms of regulation, as authorities are pushing for an Unmanned Aircraft System Traffic Management as much automated as possible.

Identification	Description
1	Route to the destination avoiding a geofence
2	Route back home
3	Route to land in a landing spot
4	Route to the destination getting out of a geofence as soon as possible
5	Route avoiding an aerial vehicle which is too close
6	Route to go back as soon as possible to the FG and resume the flight plan

Table 5.4: Different maneuvers considered to propose alternative flight plans for a UAS.

A threat management methodology should be able to evaluate the operation and propose alternative solutions that are safe, minimizing risks. However, a UAS operation not only needs to be safe but also efficient. This is why we propose a multi-objective optimization to select the best mitigation actions, trading off efficiency and safety. For that, two metrics are defined for each of the possible maneuvers: *cost* and *riskiness*. The former evaluates how costly the maneuver is with respect to the original plan in terms of additional distance covered (efficiency); the latter indicates the level of risk that the maneuver implies, i.e., how close it comes to other existing flight plans or geofences (safety). Depending on the type of maneuver, there are two ways of computing its **cost**:

- If the maneuver avoids the threat with an alternative route and then resumes the initial plan, the cost measures the distance (in meters) to be traveled along that additional path.
- On the contrary, if the UAS operation is aborted and the initial flight plan is replaced by a totally new one, e.g, toward a landing spot or back home, the cost measures the distance (in meters) to be traveled along that new flight plan. In order to favor operation completion, we penalize these maneuvers by adding to the cost the length of the uncovered part of the original flight due to the new plan. This means that the earlier the initial flight plan is interrupted, the higher the penalty.

Additionally, the **riskiness** of the maneuver can be computed by measuring two metrics:

- *Risk I*: This measures the risk due to conflicting situations generated by the maneuver. In particular, we measure the length (in meters) of the new flight plan that is still in conflict. For instance, in a maneuver to get out of a geofence or to go back to a FG, the initial part of the flight plan will still go through the conflicting volume.
- *Risk II*: This measures the risk of getting close to conflicts. In particular, we measure the minimum distance (in meters) of the new flight plan with respect to existing conflicts. For instance, the closer it gets to an existing geofence, the riskier the maneuver.

Table 5.5 summarizes the types of mitigation actions and maneuvers that are applicable for each threat. In case of *UAS in the CV*, the UAS is just warned (type A action). In case of *Alert warning*, the UAS is warned (type A action), but a geofence is also created around the dangerous situation (type B action). In case of *Technical failure* or *Jamming/spoofing attack*, the UAS is commanded a flight termination (type A action) and a geofence is created around (type B action). In case of *Communication failure* only the geofence is created (type

Threat type	**Mitigation actions applied**	**Possible maneuvers**
UAS in the CV	A	Not applicable
UAS out OV	C	2
	C	3
	C	6
Loss of separation	C	2
	C	3
	C	5
Geofence intrusion	C	2
	C	3
	C	4
Geofence conflict	C	1
	C	2
	C	3
Alert warning	A & B	Not applicable
Technical failure	A & B	Not applicable
Communication failure	B	Not applicable
Lack of battery	C	2
	C	3
Jamming attack	A & B	Not applicable
Spoofing attack	A & B	Not applicable
GNSS degradation signal	C	2
	C	3

Table 5.5: Mitigation actions applied for each type of threat, as well as the possible maneuvers.

B action), since the UAS could not be notified. For the remaining cases, different types of avoidance maneuvers are applied (type C action).

Given the threat information (Table 5.1) and the U-space constraints (Table 5.2), we determine the applicable actions and maneuvers, as indicated in Table 5.5. The TD generates possible solutions for the applicable maneuvers, with their associated cost and riskiness, so that the threat is avoided and the constraints met. Constraints regarding UAS autonomy/speed and VLL definition are considered to discard some alternative plans which may be unfeasible. Constraints related to scheduled flights, current UAS tracks, and geofences are included as no-fly zones. The known landing spots are used in maneuvers of type 3, so that the TD computes the flight plans to each of them. In case of a maneuver of type 5, i.e., several conflicting UAS avoiding each other, the TD would compute alternative flight plans for the involved UAS, attempting different avoiding directions to generate multiple solutions. Moreover, priorities are considered to only modify flight plans for those UAS with less priority operations. Once all maneuvers for the UAS involved in a given threat have been computed, the EM selects the best option for each UAS by minimizing the following value function:

$$\sum_{i=1}^{N}\sum_{j=1}^{M_i} \alpha \cdot c_{ij} + \beta_1 \cdot r_{ij}^{I} - \beta_2 \cdot r_{ij}^{II} \ , \tag{5.1}$$

where N and M_i represent the number of conflicting UAS for the given threat and the number of available maneuvers for each UAS, respectively; c_{ij} is the cost incurred if UAS i executes maneuver j; r_{ij}^{I} and r_{ij}^{II} are the riskiness I and II associated with maneuver j executed by UAS i; and $\alpha, \beta_1, \beta_2 \in [0, 1]$ are optimization weights. The values of those weights need to be tuned by a human operator. In general, the system should favor safety over efficiency, so a lower penalization for α is expected. Recall that Equation (5.1) is only used to select maneuvers in actions of type C ($\mathcal{M}$ in line 13 of Algorithm 1 is an $M \times N$ matrix containing cost and riskiness information for the maneuvers of all involved UAS, whereas ξ is a vector with the best maneuver for each UAS). Actions of type A or B are just selected for certain threats (see Table 5.5).

Tactical deconfliction

Our methodology for threat management is general enough to work with different implementations of the TD module. Any algorithm able to provide alternative plans for the conflicting UAS using the defined maneuvers could be used. In this work, we used a particular implementation integrated within the U-space architecture presented in Chapter 4. For situations where the flight plans of several UAS in conflict need to be computed (i.e., due to a loss of separation), a geometric approach based on repulsive forces is used to modify the original flight plans (Acevedo et al., 2020). Basically, the algorithm models the UAS trajectories as cords with electrical charges that repel each other, in order to increase their separation. By applying vertical or horizontal separation maneuvers between the involved

UAS trajectories in an iterative procedure, several alternative solutions can be generated. The priorities of the conflicting flight plans are also considered, as the algorithm tends not to modify the flight plans of those UAS whose operations present a higher priority in the U-space.

For other situations where a single UAS needs to compute its flight plan avoiding possible threats, e.g, to avoid a geofence, return to its OV, or go to a landing spot, a heuristic path planner based on the well-known A* algorithm is used. Geofences and running flight plans of other UAS are considered no-fly zones by this path planner.

5.3 Experimental results

This section contains experimental results to showcase the capabilities of the proposed methodology for threat management. The objectives of these experiments are twofold: (i) we show the integration of the methodology in the complete U-space architecture from Chapter 4, with all its functional modules interacting together to accomplish the specified UAS operations; and (ii) we demonstrate our method operating in real time in field experiments, testing its capabilities to solve different types of conflicts in an automated manner. For that purpose, we have defined three use cases (Section 5.3.1) involving heterogeneous UAS and several types of conflict. The experimental tests were conducted in the ATLAS Test Centre located in Villacarrillo (Jaén, Spain), which offers to the international aerospace community an aerodrome equipped with excellent technological scientific facilities and a segregated airspace, ideal for experimental flights with UAS. The use cases tested are realistic both in terms of the UAS operational parameters and the experimental setup (Section 5.3.2). All results of the tests are described in Section 5.3.3.

5.3.1 Definition of the use cases

We define three use cases using heterogeneous UAS to test different maneuverability, namely, multirotor and fixed-wing aircraft. Two of the use cases involve a pair of UAS performing operations with different priorities and the other one just involves a single UAS. In every use case, different unexpected events or threats show up while the UAS are flying and need to be managed by the UTM system. These use cases are inspired by those in Chapter 4, which were defined for simulation tests. In this chapter, we adapt them for real tests, focusing on the demonstration of the Emergency Management functionalities.

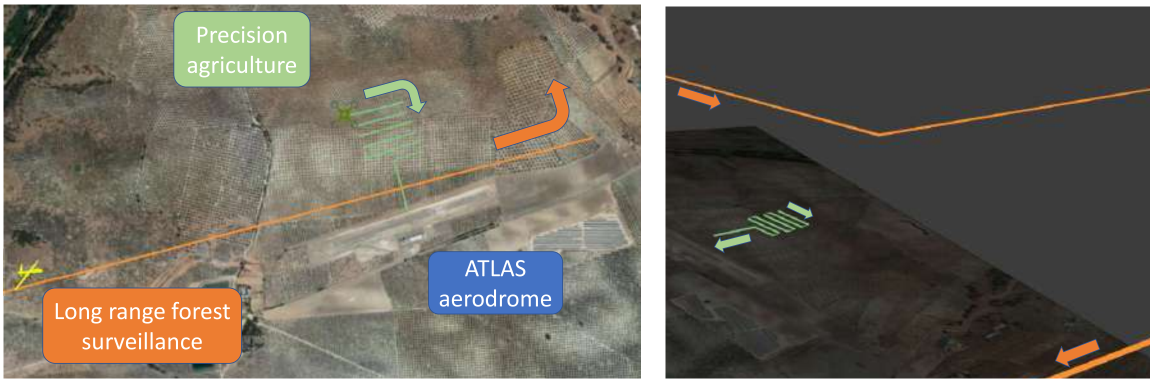

Figure 5.3: Top (left) and perspective (right) views of the initial flight plans for use case 1. All operations were planned in an area of the ATLAS aerodrome in Villacarrillo (Spain).

	Operation 1.1	**Operation 1.2**
ConOps	Precision agriculture	Long-range forest surveillance
UAS type	Multirotor (UAS_1)	Fixed-wing (UAS_2)
Cruising speed	3.3 m/s	30 m/s
Altitude (AGL)	70 m	600 m
Operation priority	Low	High
Threats involved	Loss of separation	Loss of separation

Table 5.6: Operational parameters for use case 1.

Figure 5.3 depicts the initial flight plans for use case 1. Table 5.6 summarizes the operational parameters. UAS_1 is a multirotor performing an operation for precision agriculture, while UAS_2 is a fixed-wing aircraft that performs a long-range forest surveillance operation. Note that UAS_2 flew above $150\ m$ (VLL airspace). This was done for safety reasons when operating the particular fixed-wing UAS used in the trials. Given the easier maneuverability of UAS_1, the priority of its operation is set lower. The initial flight plans are such that the UAS do not coincide in space and time throughout their operations. However, we forced a delay in the start of the UAS$_1$ operation, which resulted in a later violation of the minimum safety distance between both UAS. Thus, this use case is used to test how our threat management methodology is able to detect a loss of separation event between the UAS and to perform real-time tactical deconfliction for an inter-vehicle conflict, deciding new flight plans for both UAS. Among the available options, the Emergency Management service chooses the optimal solution to solve the conflict.

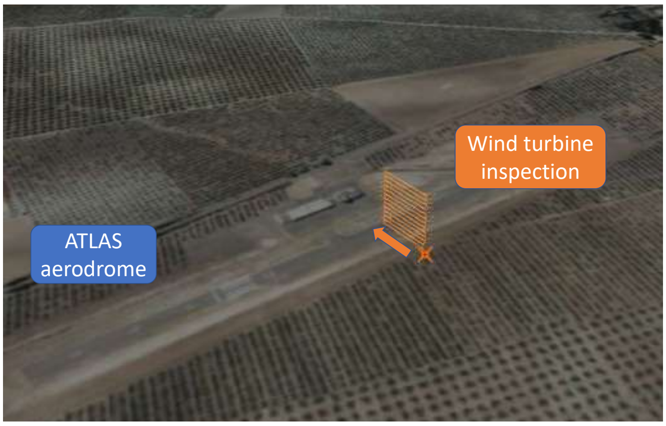

Figure 5.4: Perspective view of the initial flight plan for use case 2. The operation was planned in an area of the ATLAS aerodrome in Villacarrillo (Spain).

	Operation 2.1
ConOps	Wind turbine inspection
UAS type	Multirotor (UAS$_3$)
Cruising speed	3.3 m/s
Altitude (AGL)	30-90 m
Operation priority	Low
Threats involved	Alert warning & Geofence intrusion

Table 5.7: Operational parameters for use case 2.

Figure 5.4 depicts the initial flight plan for use case 2. Table 5.7 summarizes the operational parameters. In this case, a multirotor (UAS$_3$) is used. In its initial flight plan, UAS$_3$ moves on a vertical sweep to accomplish the inspection of a wind turbine. During the operation, a wildfire notification is simulated close to UAS$_3$. The objective of this use case is to test how our threat management methodology is able to react in an automated manner to an emergency notified by an external source (e.g.; a wildfire notified by firemen), creating a new geofence (no-fly zone) and then leaving the dangerous area.

Figure 5.5: Top (left) and perspective (right) views of the initial flight plans for use case 3. All operations were planned in an area of the ATLAS aerodrome in Villacarrillo (Spain).

	Operation 3.1	**Operation 3.2**
ConOps	Long-range powerline inspection	Event surveillance
UAS type	Fixed-wing (UAS$_2$)	Multirotor (UAS$_3$)
Cruising speed	30 m/s	3.3 m/s
Altitude (AGL)	400 m	70-100m
Operation priority	High	Low
Threats involved	Geofence conflict	Jamming attack

Table 5.8: Operational parameters for use case 3.

Figure 5.5 depicts the initial flight plans for use case 3 and Table 5.8 summarizes the operational parameters. UAS$_3$ is a multirotor performing a surveillance operation, while UAS$_2$ is a fixed-wing aircraft that has to inspect an electrical powerline. Again, note that UAS$_2$ flew above $150\ m$ (VLL airspace), due to safety reasons when operating the particular fixed-wing UAS used in the trials. Given the UAS$_3$ easier maneuverability, the priority of its operation is set lower. The initial flight plans are such that the UAS are not affected by any threat. However, during the operation, we simulated a jamming attack over UAS$_3$. The objective of this use case is to test how our threat management methodology is able to react in an automated manner to this emergency (jamming attack), creating a new geofence around the UAS attacked and then avoiding to fly inside that geofence.

5.3.2 Experimental setup

The experimental campaign shown in this chapter was carried out within the framework of the GAUSS project. The two UAS depicted in Figure 5.6 were used, in order to test their heterogeneous maneuverability and different autopilots. Table 5.9 summarizes the main features of those UAS.

Figure 5.6: The Atlantic I (left) and DJI M600 Pro (right) UAS used in the field experiments.

Model	**Type**	**Dimensions**	**MTOM**	**Range**	**Payload**
M-600 Pro (DJI)	Multirotor	*Diameter: 1.5m* *Height: 0.5m*	15kg	500m	3kg
Atlantic I	Fixed-wing	*Wingspan: 3.8m* *Length: 2.8m*	50kg	100km	3.5kg

Table 5.9: Main features of the UAS.

Our threat management methodology is implemented in Python, using the ROS middleware. This methodology is integrated within the complete U-space architecture that was developed in GAUSS (see Chapter 4), which can be found as open-source code [3]. For software integration and preliminary testing, we used a simulation based on ROS (Millan-Romera et al., 2019).

The whole experimental setup for the field experiments is similar to the one that was used for the HITL simulations in Chapter 4, and it is depicted in Figure 5.7. A Ground Control Station (GCS) was established for each UAS, with proprietary software of the

---

[3] `https://github.com/grvcTeam/gauss`

Figure 5.7: Setup for the experiments. On top, an view of the interfaces between the components run on each computer. The computers running the RPS for the two UAS and the UTM system were communicated through the Internet via the MQTT protocol. At the bottom, pictures of the UAS Ground Control Station (left) and the UTM computer (right).


Username
Password
Deployment: cloud

Figure 5.8: Screenshots of the Graphical User Interface developed by SATWAYS running on the RPS Client Application.

company EVERIS [4], who provided the aircraft. This was connected to a Remote Pilot

---

[4] https://www.everis.com/global/en

Station (RPS) with a Graphical User Interface (*RPS Client Application*) developed by the company SATWAYS [5]. The RPS Client Application, depicted in Figure 5.8, was in charge of showing telemetry and other operational data to the safety pilot. The computers on board the UAS (Intel NUC) where in charge of producing real-time telemetry data for the operation. A *RPS MQTT Broker* on the RPS was used to communicate data over the Internet to the UTM system, which ran on a different computer on the ground, implementing the U-space services involved in threat management: Emergency Management and Tactical Deconfliction. The UAS RPS communicated with the UTM system exchanging JSON messages sent over the MQTT protocol [6]. This hardware setup is realistic in terms of the U-space ecosystem, where the UTM system control station is supposed to be at a different physical location than the UAS operators, communicating via internet. Note that, in case of situations with a large number of UAS sharing the airspace, the methodology would still be scalable, as the decision-making process would just need to take into account local conflicts with nearby UAS. Besides, a cloud-based distributed architecture for the Emergency Management and Tactical Deconfliction modules could be thought.

5.3.3 Results

This section presents the results of experimental tests for the three proposed use cases [7]. The main objective is to demonstrate the actual implementation of our methodology in field tests and to assess its feasibility to handle different types of threats in real time and autonomously, only supervised by human safety pilots.

Figure 5.9 shows a timeline for an experiment implementing use case 1. According to their initial flight plans, both UAS were supposed to start their operations simultaneously at $t = 0\ s$. However, in order to test the system, we simulated a delay of $11\ s$ in the start of the UAS_1 operation, which produced a conflict between the two flight plans. Once UAS_1 and UAS_2 were flying, this conflict, which was a *loss of separation* between both UAS in the last part of their operation, was detected and notified to the EM module ($t = 12.2\ s$).

---

[5] https://www.satways.net

[6] We used the open-source Apache Active MQ broker.

[7] An illustrative video with the use cases can be seen at `https://www.youtube.com/watch?v=XerzS_IL7qQ`.

By means of our threat management methodology, the EM module evaluated the type of threat and the priorities of each UAS operation, and it decided to apply a mitigation action of type C. For that, the TD module was asked for support ($t = 12.4\ s$) to attempt different maneuvers, and it computed the flight plans whose metrics are depicted in Table 5.10.

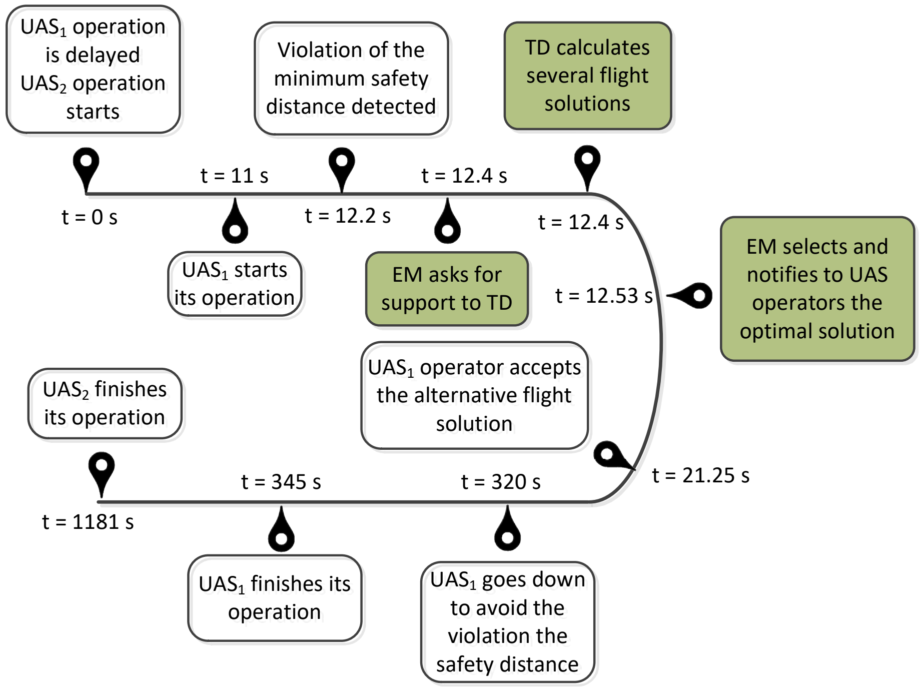

Figure 5.9: Timeline of the experiment for use case 1, where a loss of separation is resolved. The events involving the threat management methodology are shown in green.

Maneuver	Description	Cost (m)	Riskiness I (m)	Riskiness II (m)	Total value (m)
2	Go back home	1523.32	0.00	829.65	295.15
3	Land in landing spot 1	1985.30	0.00	1047.25	480.00
3	Land in landing spot 2	2213.50	0.00	1047.25	571.23
5	Turn right	550.43	0.00	829.65	-28.72
5	Turn left	494.88	0.00	727.87	-20.41
5	**Go down**	**128.52**	**0.00**	**829.67**	**-197.50**

Table 5.10: Different maneuvers computed by our method for use case 1. The selected solution (in bold) is that with the minimum weighted sum of cost and riskiness.

The option to go back home is checked by default, as well as landing on two predefined spots. Recall that these options are penalized adding to the cost the length of the initial

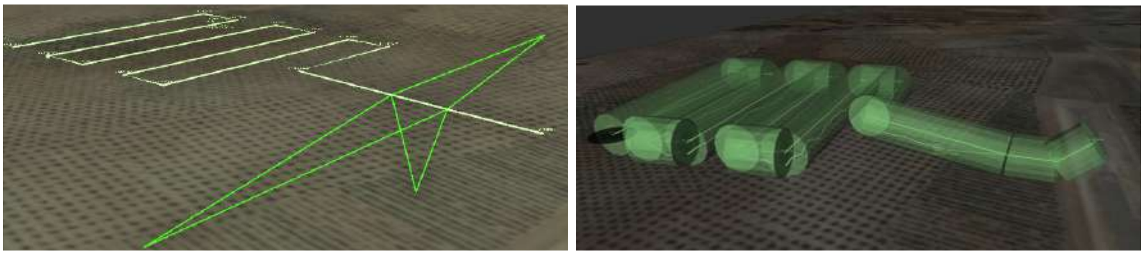

Figure 5.10: Resulting flight plan for UAS_1 in use case 1. On the left, the initial flight plan and the alternative solutions to avoid the other UAS (going back home and to the landing spots are not shown for an easier visualization). On the right, the selected solution, its Operational Volume, and the actual trajectory followed by UAS_1.

flight plan that is not covered. Besides, three different options so that one UAS (the one with the least priority) avoids the other are evaluated. All resulting flight plans are compared in terms of cost and riskiness. The weights were set by design to $\alpha = 0.4$, $\beta_1 = 0.3$ and $\beta_2 = 0.3$, in order to prioritize safety over efficiency. In this use case, the optimal solution was that the multirotor, which had more maneuverability, went down some meters to avoid the conflict and finish its operation, while the fixed-wing UAS kept its flight plan. This solution was notified by the EM to the UAS_1 operator ($t = 12.53\ s$). Figure 5.10 shows the resulting flight plan executed in the field trials. It is important to highlight that, although our UTM system is able to handle threats in an autonomous fashion, all mitigation actions were sent to the UAS operators for confirmation. This was done for operational safety reasons. Moreover, this is in line with the current U-space regulation, which states that U-space services can only suggest automatically possible correction actions, but those must be accepted or rejected by each UAS operator eventually. Nonetheless, our approach would be able to accommodate threat management based on EM and TD U-space services where the whole process were executed autonomously without the need for human intervention, which is the final objective in the U-space.

Figure 5.11 shows a timeline for an experiment implementing use case 2. UAS_3 started its operation at $t = 0\ s$, following its initial flight plan. While UAS_3 was flying, a wildfire was notified by the firemen in a nearby location ($t = 2.43\ s$), resulting in a threat of type *alert warning*. Upon that threat, the EM module decided to create a geofence around the fire ($t = 2.63\ s$), to protect aircraft around. Since UAS_3 was within the geofence, a threat

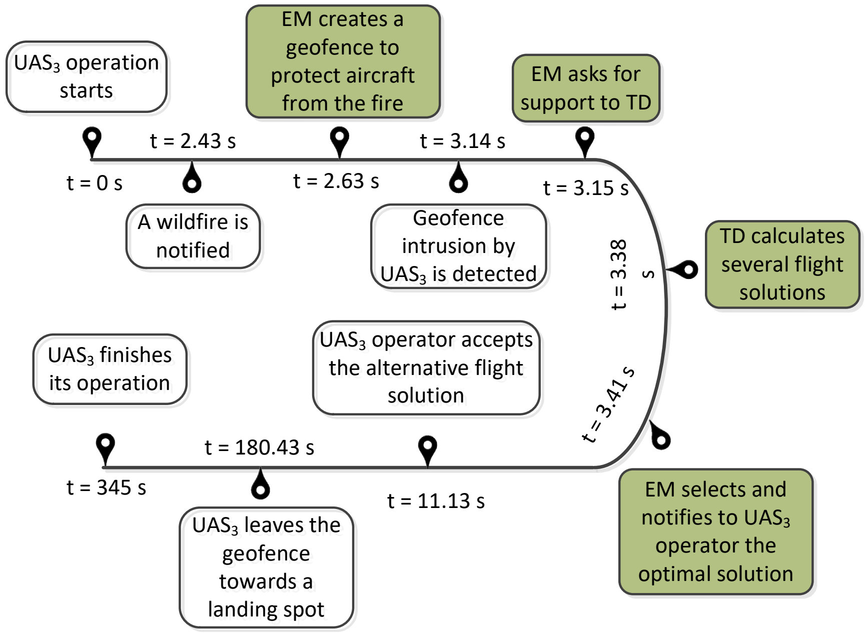

Figure 5.11: Timeline of the experiment for use case 2, where an alert warning (wildfire notification) and a geofence intrusion are resolved. The events involving the threat management methodology are shown in green.

of type *geofence intrusion* was detected and notified to the EM module ($t = 3.14\ s$). At that moment, our threat management methodology decided to apply a mitigation action of type C, given the type of threat. For that, the TD module was asked for support ($t = 3.15\ s$) to attempt different maneuvers, and it computed the flight plans whose metrics are depicted in Table 5.11.

Maneuver	Description	Cost (m)	Riskiness I (m)	Riskiness II (m)	Total value (m)
3	**Land on landing spot 1**	**197.97**	**94.00**	**0.00**	**107.39**
3	Land on landing spot 2	382.55	122.00	0.00	189.62

Table 5.11: Different maneuvers computed by our method for use case 2. The selected solution (in bold) is that with the minimum weighted sum of cost and riskiness.

Options to go back home and to get out of the geofence as soon as possible and resume the flight plan were discarded, as they did not fulfill the U-space constraints. This happened because the whole initial flight plan of UAS_3 was within the created geofence. Alternative options to land on the known landing spots were checked instead. All resulting flight plans were compared in terms of cost and riskiness. The weights were also set by design to $\alpha = 0.4$, $\beta_1 = 0.3$ and $\beta_2 = 0.3$. In this use case, the optimal solution was that the multirotor landed on the closest landing spot outside the geofence. This solution was notified by the EM to the UAS$_3$ operator ($t = 3.41\ s$). Figure 5.12 shows the resulting flight plan executed in the field trials.

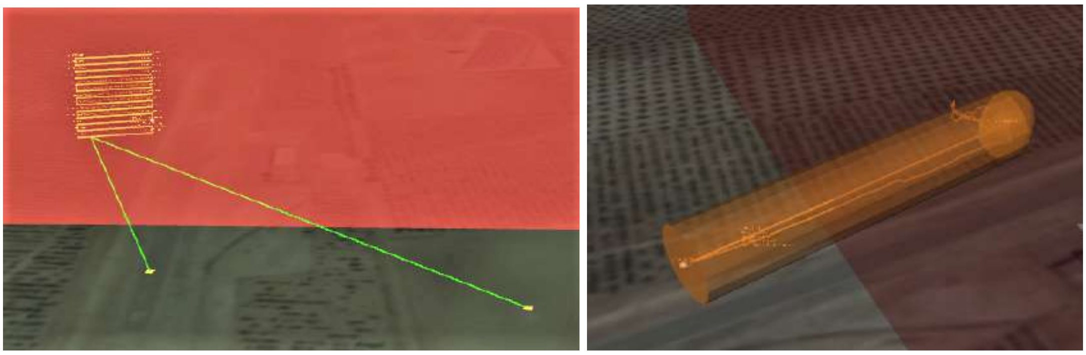

Figure 5.12: Resulting flight plan for UAS$_3$ in use case 2. On the left, the initial flight plan and the alternative solutions to land on different spots. On the right, the selected solution to the closest landing spot, its Operational Volume, and the actual trajectory followed by UAS$_3$. In red, the geofence which the UAS is getting out of.

Figure 5.13 shows a timeline for an experiment implementing use case 3. Both UAS started their operations simultaneously ($t = 0\ s$), following their initial flight plans. During their operation, we simulated a jamming attack over UAS$_3$ ($t = 12\ s$). For that type of threat, our method for threat management created a geofence around the attacked UAS (action of type B) and asked the UAS$_3$ operator (action of type A) to land now ($t = 12.1\ s$). While UAS$_3$ was landing, a geofence conflict between UAS$_2$ and the new geofence was detected and notified ($t = 12.38\ s$), i.e., the flight plan of UAS$_2$ was going through that new geofence. Our threat management methodology decided to apply a mitigation action of type C, given the type of threat. For that, the TD module was asked for support ($t = 13.07\ s$) to attempt different maneuvers, and it computed the flight plans whose metrics are depicted in Table 5.12.

Options to land on the known landing spots and to go back home are checked by default. Besides, an additional option so that UAS$_2$ avoids the geofence and resumes its flight plan. All resulting flight plans are compared in terms of cost and riskiness. The

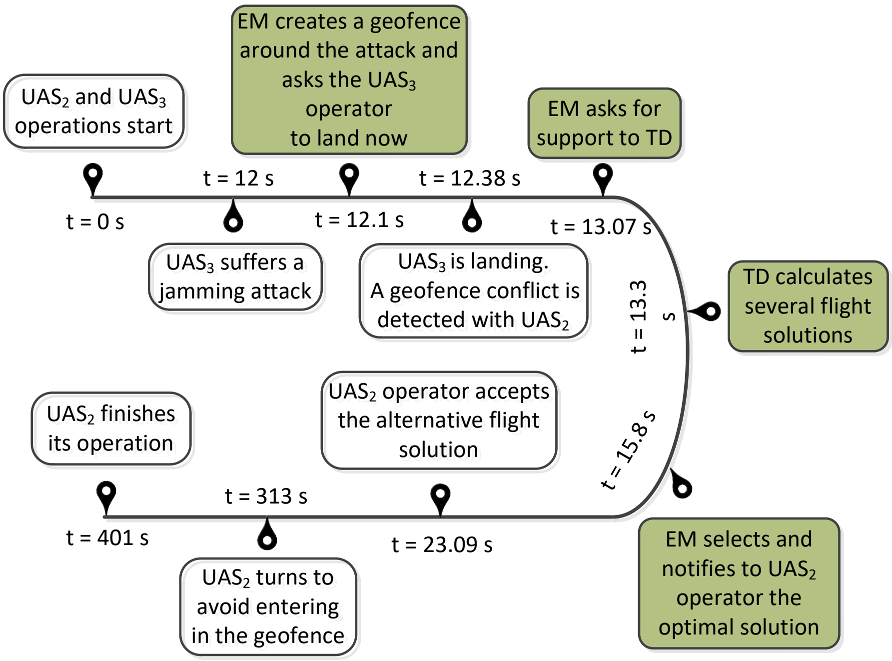

Figure 5.13: Timeline of the experiment for use case 3, where a jamming attack and a geofence conflict are resolved. The events involving the threat management methodology are shown in green.

weights were also set by design to $\alpha = 0.4$, $\beta_1 = 0.3$ and $\beta_2 = 0.3$. In this use case, the optimal solution was that the fixed-wing UAS circumvented the geofence and then resumed with its original plan. This solution was notified by the EM to the UAS$_2$ operator ($t = 15.8\ s$). Figure 5.14 shows the resulting flight plan executed in the field trials.

Maneuver	Description	Cost (m)	Riskiness I (m)	Riskiness II (m)	Total value (m)
1	**Route avoiding the geofence**	**1400.29**	**0.00**	**214.84**	**495.92**
2	Go back home	12393.85	0.00	9002.26	2256.94
3	Land on landing spot 1	20943.95	0.00	469.57	8237.00
3	Land on landing spot 2	20855.74	0.00	622.65	8155.50

Table 5.12: Different maneuvers computed by our method for use case 3. The selected solution (in bold) is that with the minimum weighted sum of cost and riskiness.

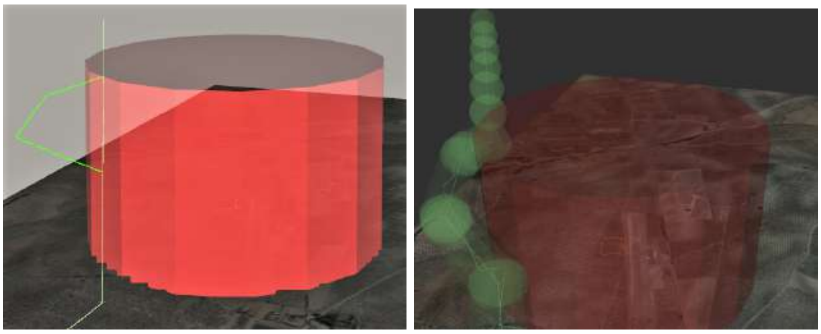

Figure 5.14: Resulting flight plan for UAS_3 in use case 3. On the left, the initial flight plan and the alternative solution to avoid the geofence (red cylinder). On the right, the selected solution turning around the geofence, its Operational Volume, and the actual trajectory followed by UAS_3.

5.4 Conclusions

In this chapter, we have presented a threat management methodology for UAS operating within the U-space ecosystem. Our method is capable of handling all usual threats in UTM systems, and it performs real-time and autonomous decision-making to provide optimal mitigation actions in terms of cost and risk level. The methodology is integrated with a U-space architecture, implementing in-flight services for emergency management and tactical deconfliction.

We have demonstrated that our methodology is capable of autonomously handling heterogeneous threats in real time, through a set of use cases implemented on real rotary- and fixed-wing UAS. In our experiments, the system was able to resolve different types of conflicts, reasoning about 4D UAS trajectories, geofences, and Operational Volume. Moreover, the experimental setup was realistic with respect to the actual U-space ecosystem, as the onboard and on-ground systems were running at different places and communicated over the Internet.

However, our system has still some limitations. It relies on an accurate positioning of UAS, dismissing possible uncertainties. For instance, during the experiments performed, we noticed that the telemetry of the UAS, especially the fixed-wing aircraft, were unstable at some periods, which could result in the detection of "fake" conflicts. These uncertainties could be increased due to communication delays or blackouts. As future work, we plan to introduce security margins in our method to consider uncertainties in the detection and resolution of threats. Besides, this chapter could be the base for the design of a digital and automated methodology for risk assessment, working in real time as different UAS are flying and unexpected events show up. The method could be integrated in a real UTM system through the specific U-space service *Risk analysis assistance*.

Chapter 6

Conclusions and Future Work

This chapter summarizes the main contributions and results of this thesis. Also, the limitations of the approach proposed are discussed, together with a proposal for future work.

6.1 Conclusions

In this thesis, we have presented a UTM system architecture based on the U-space definition. This software architecture is service-based and focuses on *in-flight* U-space services for autonomous threat (or unexpected events) management and conflict resolution in real time for multi-UAS operations. Although we have focused on the relevant services for threat management, the architecture is modular and flexible enough to be extended with more U-space services and functionalities in future implementations. We have also proposed a methodology for autonomous decision-making to handle these unexpected events while multiple UAS are operating in a common U-space. Our method is capable of handling a list of usual threats in UTM systems, and it performs real-time and autonomous decision-making to provide optimal mitigation actions in terms of cost and risk level. The methodology is fully integrated within the aforementioned UTM system architecture, implementing the *in-flight* U-space services for emergency management and tactical deconfliction. Furthermore, we have described the SORA risk assessment methodology, defined by EASA for UAS operation approval following the current U-space regulatory framework; and we have applied it to a particular multi-UAS civil operation as an example, to understand better the procedure.

The thesis includes field experiments with heterogeneous UAS (fixed and rotary wing) that showcase that our approach is capable of autonomously handling different types of conflicting situations in real time with several UAS operating in the U-space. We propose a realistic experimental setup with respect to the actual U-space ecosystem, with the onboard and on-ground systems running at different locations and communicated over the Internet. Moreover, all the software produced has been published as open source for the UAS community.

During our experiments, we realized that having a reliable communication network where the signal Quality of Service is ensured is a key aspect to deploy a U-space system. A permanent and secure communication link between the UTM Service Provider and the UAS is needed to handle multi-UAS operations safely and efficiently. Another issue to highlight is the fact there is no current regulatory framework for threat management in the U-space. In this regard, all developments in this topic belong to the category of R&D activities. Therefore, before moving to more commercial solutions, it would be necessary to define a common regulatory framework specifying concepts like safety distance, operational volume, threat severities, operation priorities, acceptable level of safety, etc.

Finally, we would also like to remark that our system has still some design limitations. First, it relies on a centralized UTM server that requires continuous communication with other airspace actors. This bottleneck could be addressed by splitting the UTM system into a set of distributed and interconnected servers. Second, our approach does not consider non-cooperative vehicles in the VLL airspace, such as ultralight planes, nor pre-flight services. However, the architecture is flexible enough to integrate additional services, e.g., for flight operation pre-planning or strategic deconfliction. Moreover, non-cooperative vehicles could be tackled by working with see&avoid systems on board the UAS.

6.2 Future work

Guided by the main conclusions of this thesis, there are several research lines that have been identified of interest for future developments in U-space. In the following, we sum up these ideas for future work:

- **Reliable communication infrastructure.** As it has been discussed in the conclusions, a key enhancement in the UTM architecture developed in Chapter 4 would be the integration of a reliable communication infrastructure underneath. Due to safety reasons, it is crucial relying on a robust and secure communication network for U-space interactions. In this line, 5G technology looks promising in order to guarantee communication links with low latency and high rate of data transmission to share UAS telemetry, mitigation actions, and other relevant information.

- **Distributed architecture.** Another interesting concept for our multi-UAS architecture would be having a more distributed design. Different U-space services could be implemented in multiple remote servers in a decentralized fashion, in order to alleviate the traffic congestion that may be caused by our centralized communication design. For that purpose, the use of cloud-based infrastructures like Amazon Web Services is a promising idea.

- **Autonomous risk assessment.** Regarding the emergency management methodology described in Chapter 5, risk assessment could be addressed in a more proper and systematic manner. Thus, this thesis could be the base for the design of a digital and automated methodology for risk assessment in U-space operations, working in real time as different UAS are flying and unexpected events show up. The method could be integrated in a real UTM system through the specific U-space service *Risk analysis assistance*.

- **Heterogeneity.** Additionally, our decision-making procedures for emergency management could be extended to deal more explicitly with a wider range of the heterogeneous aircraft that will be found in future urban air mobility applications, such as e-VTOLs (Vertical Take-Off and Landing), airtaxis, and convertible drones.

- **Scalability tests.** Finally, our system has been tested in some use cases involving several UAS as proof of concept. Nonetheless, more exhaustive tests to assess the scalability of the framework is still missing. These tests would be helpful to evaluate the feasibility of our U-space emergency management solutions in a real urban environment with many vehicles operating in parallel.

Bibliography

Acevedo, J. J., Capitán, C., Capitán, J., Castaño, A. R., and Ollero, A. (2020). A Geometrical Approach based on 4D Grids for Conflict Management of Multiple UAVs operating in U-space. In *International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 263–270, Athens, Greece.

AirMap (2018). Five Critical Enablers or Safe, Efficient, and Viable UAS Traffic Management (UTM). Technical report, AirMap.

Alarcon, V., Garcia, M., Alarcon, F., Viguria, A., Martinez, A., Janisch, D., Acevedo, J. J., Maza, I., and Ollero, A. (2020). Procedures for the integration of drones into the airspace based on U-space services. *Aerospace*, 7(9).

Alcantara, A., Capitan, J., Torres-Gonzalez, A., Cunha, R., and Ollero, A. (2020). Autonomous Execution of Cinematographic Shots with Multiple Drones. *IEEE Access*.

Alcántara, A., Capitán, J., Cunha, R., and Ollero, A. (2021). Optimal trajectory planning for cinematography with multiple unmanned aerial vehicles. *Robotics and Autonomous Systems*, 140:103778.

Alonso-Mora, J., Naegeli, T., Siegwart, R., and Beardsley, P. (2015). Collision avoidance for aerial vehicles in multi-agent scenarios. *Autonomous Robots*, 39(1):101–121.

Amazon Prime Air (2015). Revising the Airspace Model for the Safe Integration of Small Unmanned Aircraft Systems. `https://www.documentcloud.org/documents/2182311-amazon-revising-the-airspace-model-for-the-safe.html`.

Ancel, E., Capristan, F. M., Foster, J. V., and Condotta, R. C. (2017). Real-time Risk Assessment Framework for Unmanned Aircraft System (UAS) Traffic Management (UTM). *17th AIAA Aviation Technology, Integration, and Operations Conference*, pages 1–17.

Atkins, E. M., Portillo, I. A., and Strube, M. J. (2015). Emergency Flight Planning Applied to Total Loss of Thrust. *Journal of Aircraft*, 43.

Aurambout, J. P., Gkoumas, K., and Ciuffo, B. (2019). Last mile delivery by drones: an estimation of viable market potential and access to citizens across European cities. *European Transport Research Review*, 11(1).

Aweiss, A., Homola, J., Rios, J., Jung, J., Johnson, M., Mercer, J., Modi, H., Torres, E., and Ishihara, A. (2019). Flight Demonstration of Unmanned Aircraft System (UAS) Traffic Management (UTM) at Technical Capability Level 3. In *IEEE/AIAA Digital Avionics Systems Conference (DASC)*, pages 1–7.

Barrado, C., Boyero, M., Brucculeri, L., Ferrara, G., Hately, A., Hullah, P., Martin-marrero, D., Pastor, E., Rushton, A. P., and Volkert, A. (2020). U-Space Concept of Operations : A Key Enabler for Opening Airspace to Emerging Low-Altitude Operations. *Aerospace*, pages 1–18.

Campusano, M., Jensen, K., and Schultz, U. P. (2021). Towards a Service-Oriented U-Space Architecture for Autonomous Drone Operations. In *IEEE/ACM International Workshop on Robotics Software Engineering (RoSE)*, pages 63–66.

Capitan, C., Capitan, J., Castaño, A. R., and Ollero, A. (2022). Threat Management Methodology for Unmanned Aerial Systems operating in the U-space. *IEEE Access*.

Capitan, C., Capitan, J., Castaño, A. R., and Ollero, A. (2019). Risk Assessment based on SORA Methodology for a UAS Media Production Application. *International Conference on Unmanned Aircraft Systems (ICUAS)*.

Capitán, C., Castaño, A. R., Capitán, J., and Ollero, A. (2019). A framework to handle threats for UAS operating in the U-space. In *International Workshop on Research, Education and Development on Unmanned Aerial Systems (RED-UAS)*, Cranfield, United Kingdom.

Capitan, C., Perez-Leon, H., Capitan, J., Castaño, A., and Ollero, A. (2021). Unmanned Aerial Traffic Management System Architecture for U-Space In-Flight Services. *Applied Sciences*, 11(9).

Capitan, J., Merino, L., and Ollero, A. (2016). Cooperative decision-making under uncertainties for multi-target surveillance with multiples UAVs. *Journal of Intelligent & Robotic Systems*, 84(1):371–386.

Clothier, R. A., Williams, B. P., and Fulton, N. L. (2015). Structuring the safety case for unmanned aircraft system operations in non-segregated airspace. *Safety Science*, 79:213 – 228.

Considine, B. (2003). Guidelines for controller training in the handling of unusual / emergency situations. `https://skybrary.aero/sites/default/files/bookshelf/15.pdf`.

CORUS (2019). Intermediate ConOps Annex E - list of Threats and Events. `https://www.sesarju.eu/sites/default/files/documents/u-space/CORUS_Intermediate_ConOps.pdf`.

Cour-Harbo, A. L. (2018). The Value of Step-by-Step Risk Assessment for Unmanned Aircraft. *International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 149–157.

Denney, E. and Pai, G. (2016). Architecting a safety case for UAS flight operations. In *34th International System Safety Conference*.

Denney, E., Pai, G., and Johnson, M. (2018). Towards a Rigorous Basis for Specific Operations Risk Assessment of UAS. In *IEEE/AIAA Digital Avionics Systems Conference (DASC)*, pages 1–10.

Doole, M., Ellerbroek, J., and Hoekstra, J. M. (2022). Investigation of merge assist policies to improve safety of drone traffic in a constrained urban airspace. *Aerospace*, 9(3).

Dorling, K., Heinrichs, J., Messier, G. G., and Magierowski, S. (2017). Vehicle routing problems for drone delivery. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 47(1):70–85.

Dubot, T. and Joulia, A. (2021). Towards U-space conflict management services based on 4D protection bubbles. In *AIAA AVIATION 2021 FORUM*.

East Gippsland Shire Council (2017). Emergency procedures - Drone (Remote Piloted Aircraft). `https://bit.ly/3qkMzNg`.

Eurocontrol (2020). U-space services implementation monitoring report. Technical report, Eurocontrol.

European Aviation Safety Agency (EASA) (2015). Advance Notice of Proposed Amendment 2015-10: Introduction of a regulatory framework for the operation of drones. Technical report, EASA document library.

European Aviation Safety Agency (EASA) (2016). UAS Safety Risk Portfolio and Analysis. `https://www.easa.europa.eu/document-library/general-publications/uas-safety-risk-portfolio-and-analysis`.

European Aviation Safety Agency (EASA) (2017). NPA 2017-05 (B) Introduction of a regulatory framework for the operation of drones Unmanned aircraft system operations in the open and specific category. Technical report, EASA document library.

European Aviation Safety Agency (EASA) (2018). UAS workshop on Standard Scenarios. `https://www.easa.europa.eu/newsroom-and-events/news/uas-workshop-2018-standard-scenarios`.

European Aviation Safety Agency (EASA) (2020). High-level regulatory framework for the U-space. Technical Report 18, EASA document library.

European Comission (2021). Requirements for manned aviation operating in U-space airspace. `https://ec.europa.eu/transport/sites/transport/files/c20212673-u-space.pdf`.

European Commission (2021a). Regulatory framework for the U-space. `https://ec.europa.eu/transport/sites/transport/files/c20212671-u-space.pdf`.

European Commission (2021b). Requirements for providers of air traffic management/air navigation services and other air traffic management network functions in the U-space airspace designated in controlled airspace. `https://ec.europa.eu/transport/sites/transport/files/c20212672-u-space.pdf`.

Fern, L., Rorie, R. C., and Shively, R. (2014). UAS Contingency Management: The Effect of Different Procedures on ATC in Civil Airspace Operations. *American Institute of Aeronautics and Astronautics*.

Ferreira, R. B., Baum, D. M., Neto, E. C. P., Martins, M. R., Almeida, J. R., Cugnasca, P. S., and Camargo, J. B. (2018). A Risk Analysis of Unmanned Aircraft Systems (UAS) Integration into non-Segregate Airspace. *International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 42–51.

Finke, M. (2016). Defining RPAS emergency procedures for controllers , remote pilots and automatic on-board systems. *Computer Science*, pages 1–10.

FOCA (2015). RPAS in Switzerland Rules and Integration. Technical report, FOCA.

FOCA (2017). Authorization of drone flights in Switzerland. Technical report, Federal Office of Civil Aviation (FOCA).

Garcia-Aunon, P., Roldán, J. J., and Barrientos, A. (2019). Monitoring traffic in future cities with aerial swarms: Developing and optimizing a behavior-based surveillance algorithm. *Cognitive Systems Research*, 54:273–286.

Global UTM Association (2020). Designing UTM for global success. `https://gutma.org/designing-utm-for-global-success/`.

Google (2015). Google UAS Airspace System Overview. Technical report, NASA.

Guo, X., Denman, S., Fookes, C., Mejias, L., and Sridharan, S. (2014). Automatic UAV forced landing site detection using machine learning. In *International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–7.

Ho, F., Geraldes, R., Gonçalves, A., Rigault, B., Oosedo, A., Cavazza, M., and Prendinger, H. (2019). Pre-Flight Conflict Detection and Resolution for UAV Integration in Shared Airspace: Sendai 2030 Model Case. *IEEE Access*, 7:170226–170237.

Ho, F., Geraldes, R., Gonçalves, A., Rigault, B., Sportich, B., Kubo, D., Cavazza, M., and Prendinger, H. (2022). Decentralized Multi-Agent Path Finding for UAV Traffic Management. *IEEE Transactions on Intelligent Transportation Systems*, 23(2):997–1008.

Janzen, J. (2019). European ATC Remains Inefficient, Expensive and Unreliable for Millions of Passengers. *Airlines for Europe*.

JARUS (2018). JARUS guidelines on Specific Operations Risk Assessment (SORA). *JARUS publications*.

JARUS (2019). Joint Authorities for Rulemaking of Unmanned Systems JARUS guidelines on SORA 2.0. `http://jarus-rpas.org/sites/jarus-rpas.org/files/jar_doc_06_jarus_sora_v2.0.pdf`.

Jiang, T., Geller, J., Ni, D., and Collura, J. (2016). Unmanned aircraft system traffic management: Concept of operation and system architecture. *International Journal of Transportation Science and Technology*, 5(3):123–135. Unmanned Aerial Vehicles and Remote Sensing.

Jover, Jesús; Bermúdez, A. and Casado, R. (2022). Priority-aware conflict resolution for u-space. *Electronics*, 11(8).

Jover, J., Bermúdez, A., and Casado, R. (2021). A Tactical Conflict Resolution Proposal for U-Space Zu Airspace Volumes. *Sensors*, 21(16).

Kopardekar, P. (2015). Unmanned Aerial System (UAS) Traffic Management (UTM): Enabling Civilian Low-Altitude Airspace and Unmanned Aerial System Operations. `https://n9.cl/6yxi`.

Kopardekar, P., Rios, J., Prevot, T., M, J., J, J., and J, R. (2016). Unmanned Aircraft System Traffic Management (UTM) Concept of Operations. *Aviation Technology, Integration, and Operation Conference AIAA Aviation*.

Kramar, V. (2020). UAS (drone) in Response to Coronavirus. In *Conference of Open Innovations Association (FRUCT)*, pages 90–100.

Lappas, V., Zoumponos, G., Kostopoulos, V., Lee, H. I., Shin, H.-S., Tsourdos, A., Tantardini, M., Shomko, D., Munoz, J., Amoratis, E., Maragkakis, A., Machairas, T., and Trifas, A. (2022). EuroDRONE, a European Unmanned Traffic Management Testbed for U-Space. *Drones*, 6(2).

Lin, C. E., Chen, T., Shao, P., Lai, Y., Chen, T., and Yeh, Y. (2019). Prototype Hierarchical UAS Traffic Management System in Taiwan. In *Integrated Communications, Navigation and Surveillance Conference (ICNS)*, pages 1–13.

Lundberg, J., Palmerius, K. L., and Josefsson, B. (2018). Urban Air Traffic Management (UTM) Implementation In Cities - Sampled Side-Effects. In *IEEE/AIAA Digital Avionics Systems Conference (DASC)*, pages 1–7.

Mademlis, I., Mygdalis, V., Nikolaidis, N., Montagnuolo, M., Negro, F., Messina, A., and Pitas, I. (2019). High-Level Multiple-UAV Cinematography Tools for Covering Outdoor Events. *IEEE Transactions on Broadcasting*, pages 1–9.

Martin, T., Huang, Z. F., and McFadyen, A. (2018). Airspace Risk Management for UAVs A Framework for Optimising Detector Performance Standards and Airspace Traffic using JARUS SORA. In *IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*.

Masri, M. A. (2017). Autolanding a Power-off UAV Using On-line Optimization and Slip Maneuvers. *Journal of Intelligent & Robotic Systems*, 86:255–276.

McFadyen, A. and Martin, T. (2016). Terminal airspace modelling for unmanned aircraft systems integration. In *International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 789–794.

Mcfadyen, A. and Mejias, L. (2016). A survey of autonomous vision-based see and avoid for unmanned aircraft systems. *Progress in Aerospace Sciences*, 80:1 – 17.

Mejias, L. and Eng, P. (2013). Controlled Emergency Landing of an Unpowered Unmanned Aerial System. *Journal of Intelligent & Robotic Systems*, 70:421–435.

Mejias L., C. P. and J., R. (2014). Field and service robotics. *Springer Tracts in Advanced Robotics.*

Messina, A., Metta, S., Montagnuolo, M., Negro, F., Mygdalis, V., Pitas, I., Capitán, J., Torres, A., Boyle, S., Bull, D., and Zhang, F. (2018). The future of media production through multi-drones' eyes. In *Proceedings of the International Broadcasting Convention (IBC)*, Amsterdam.

Millan-Romera, J. A., Acevedo, J. J., Perez-Leon, H., Capitan, C., Castaño, A. R., and Ollero, A. (2019). A UTM simulator based on ROS and GAZEBO. In *International Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS).*

Ollero, A., Heredia, G., Franchi, A., Antonelli, G., Kondak, K., Sanfeliu, A., Viguria, A., Martinez-de Dios, J. R., Pierri, F., Cortes, J., Santamaria-Navarro, A., Trujillo Soto, M. A., Balachandran, R., Andrade-Cetto, J., and Rodriguez, A. (2018). The AEROARMS Project: Aerial Robots with Advanced Manipulation Capabilities for Inspection and Maintenance. *IEEE Robotics Automation Magazine*, 25(4):12–23.

Pastor, E., Royo, P., Santamaria, E., Prats, X., and Barrado, C. (2012). In-flight contingency management for unmanned aerial vehicles. *Journal of Aerospace Computing, Information, and Communication*, 9(4):144–160.

Peinecke, N. and Kuenz, A. (2017). Deconflicting the urban drone airspace. In *IEEE/AIAA Digital Avionics Systems Conference (DASC)*, pages 1–6.

Real, F., Torres-Gonzalez, A., Ramon-Soria, P., Capitan, J., and Ollero, A. (2020). Unmanned aerial vehicle abstraction layer: An abstraction layer to operate unmanned aerial vehicles. *International Journal of Advanced Robotic Systems*, 17(4):1–13.

Rubio-Hervas, J., Gupta, A., and Ong, Y.-S. (2018). Data-driven risk assessment and multi-criteria optimization of UAV operations. *Aerospace Science and Technology*, 77:510–523.

Rumba, R. and Nikitenko, A. (2020). The wild west of drones: a review on autonomous-UAV traffic-management. In *International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1317–1322.

Sacharny, D., Henderson, T. C., and Cline, M. (2020). Large-Scale UAS Traffic Management (UTM) Structure. In *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 7–12.

Sanchez-Cuevas, P. J., Gonzalez-Morgado, A., Cortes, N., Gayango, D. B., Jimenez-Cano, A. E., Ollero, A., and Heredia, G. (2020). Fully-actuated aerial manipulator for infrastructure contact inspection: Design, modeling, localization, and control. *Sensors*, 20(17):1–26.

SESAR (2017). U-space Blueprint. Technical report, SESAR Joint Undertaking publications.

SESAR (2020). Supporting Safe and Secure drone operations in Europe A report of the consolidated SESAR U-space research and innovation results. `https://www.sesarju.eu/sites/default/files/documents/reports/U-space%20research%20innovation%20results.pdf`.

Tan, Q., Wang, Z., Ong, Y., and Low, K. H. (2019). Evolutionary Optimization-based Mission Planning for UAS Traffic Management (UTM). In *International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 952–958.

Tang, Y., Xu, Y., and Inalhan, G. (2022). An Integrated Approach for On-Demand Dynamic Capacity Management Service in U-space. *IEEE Transactions on Aerospace and Electronic Systems*, pages 1–1.

Ten Harmsel, A. J., Olson, I. J., and Atkins, E. M. (2017). Emergency Flight Planning for an Energy-Constrained Multicopter. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 85(1):145–165.

Thales (2017). Thales launches ECOsystem UTM and joins forces with Unifly to facilitate drone use. `https://www.thalesgroup.com/sites/default/files/database/d7/asset/document/170307_thales_launches_ecosystem_utm_and_joins_forces_with_unifly_to_facilitate_drone_en.pdf`.

Unifly (2020). Unifly UTM platform. `https://www.unifly.aero/solutions/unmanned-traffic-management`.

Warren, M., Mejias, L., Kok, J., Yang, X., Gonzalez, F., and Upcroft, B. (2015). An Automated Emergency Landing System for Fixed-Wing Aircraft: Planning and Control. *Journal of Field Robotics*, pages 1114–1140.