UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
TELECOMUNICACIÓN

Contributions to Drone Fleet Mission Planning and Monitoring

Tesis Doctoral

Iván Campaña Ramos
Graduado en Ingeniería de Tecnologías y Servicios de
Telecomunicación
Máster Universitario en Ingeniería de Telecomunicación

2020

Departamento de Señales, Sistemas y Radiocomunicaciones

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE
TELECOMUNICACIÓN

UNIVERSIDAD POLITÉCNICA DE MADRID

Contributions to Drone Fleet Mission Planning and Monitoring

Tesis Doctoral

Autor:

IVÁN CAMPAÑA RAMOS

Graduado en Ingeniería de Tecnologías y Servicios de Telecomunicación

Máster Universitario en Ingeniería de Telecomunicación

Director:

JUAN ALBERTO BESADA PORTAS

Doctor Ingeniero de Telecomunicación

2020

DEPARTMENT:

Departamento de Señales, Sistemas y Radiocomunicaciones

Escuela Técnica Superior de Ingenieros de Telecomunicación

Universidad Politécnica de Madrid

PH.D. THESIS:

Contributions to Drone Fleet Mission Planning and Monitoring

Author:

Iván Campaña Ramos

Advisor:

Juan Alberto Besada Portas

Committee named by the Rector of Universidad Politécnica de Madrid, on the ____ of _____ 2020.

President:

José Ramón Casar Corredera

Universidad Politécnica de Madrid

Member:

Jesús García Herrero

Universidad Carlos III de Madrid

Member:

José Manuel Molina López

Universidad Carlos III de Madrid

Member:

José Miguel Canino Rodríguez

Universidad de Las Palmas de Gran Canaria

Secretary:

Ana M^a Bernardos Barbolla

Universidad Politécnica de Madrid

Reserve:

José Antonio López Orozco

Universidad Complutense de Madrid

Reserve:

Juan Jiménez Castellanos

Universidad Complutense de Madrid

After the defence of the Ph.D. Thesis on the ____ of _____ 2020, at the Escuela Técnica Superior de Ingenieros de Telecomunicación, the committee agrees to grant the following qualification:

President

Secretary

Members

Acknowledgements

The road to the completion of this thesis has not been easy. There have been many ups and downs, but I have always had the help I need to keep going. Therefore, I would like to start this thesis by thanking all the people who have helped me and supported me during all this time.

First of all, I would like to thank Prof. Juan Besada for being my tutor since 2015, when I joined the group to do my final master's thesis, after which I became a Ph.D. student. During all this time, Juan has been helping me in everything I have needed, as well as giving me good advice. Thank you very much for all the time and help you have given me, especially during this 2020, my last year. It has been a very hard year because of everything that has happened, but you have taken time to help me finish this thesis in difficult times. I would also like to thank the other professors of the group - José Ramón, Gonzalo, Ana and Javier - for all their advice, as well as the help provided when I asked for it.

I would like to thank my colleagues, of whom there have been many, who have offered me help and advice throughout this time, especially when I was new to the GPDS. In particular, I would like to thank Jaime and Ángel for the good reception, all the help offered and the good times they have given to me in this group during all the time we have spent together. I would also like to thank Luca and David because they have contributed many good pieces of advice, as well as help, to the realization of this thesis, and also for making me more entertaining this year since we are the ones who remain in the group. And finally to Diego for being my Ph.D. partner, the one who has understood me during all this process.

Thank you very much to my group of friends, who every weekend have stayed with me so that I could get away and have fun and thus recharge my batteries to start the week in full swing. Also, their group therapies have been very beneficial during difficult times. Besides, I am grateful to my friends and colleagues who have continued with me in this adventure that is the doctorate. We have all supported each other during our crises and they more than anyone else know what these four years have been like.

And finally, the greatest thank goes to my family, especially to my parents - Valentina and Jose - and to my brother Jorge because without their teachings,

support, advice and patience, I would not have been able to get this far. Thank you for helping me and for putting up with me during all this time. And also to thank my aunt Isa for the good advice she has given me throughout my life.

Resumen

Los servicios basados en los vehículos aéreos no tripulados están todavía en sus primeras fases. Pero los escenarios futuros prevén espacios aéreos urbanos y no urbanos compartidos por múltiples vehículos aéreos no tripulados con misiones muy diferentes como pueden ser la prestación de servicios logísticos, la respuesta a emergencias o el transporte, entre otros muchos. En estos escenarios aparece la figura del proveedor de servicio, que puede ser una autoridad pública o una empresa, quien tendrá en propiedad y operará una flota de vehículos aéreos teledirigidos y que en ocasiones podrá pagar a operadores de drones externos para que realicen ciertos vuelos específicos. La aparición de una herramienta para definir con facilidad la misión, calcular la trayectoria óptima y vigilar la ejecución de la misión en tiempo real beneficiará, en todas las situaciones posibles, al usuario que necesite un servicio gestionando una flota de drones.

Esta tesis propone un sistema que puede contribuir a la gestión y control de flotas de drones constituido por una arquitectura basada en microservicios que permite la planificación de misiones y la vigilancia de los drones durante su ejecución. La arquitectura está formada por una serie de microservicios que contienen la principal lógica del sistema: el Servicio de Planificación de Misión, el Servicio de Planificación de Vuelo, el Servicio de Predicción de Trayectoria y el Servicio de Seguimiento y Monitorización.

Primero, esta tesis describe los requisitos necesarios que debe cumplir una herramienta de planificación de misiones para la realización de operaciones urbanas y no urbanas. Esta herramienta es capaz de diseñar planes de vuelo teniendo en consideración las restricciones de vuelo y las limitaciones aeroespaciales. El Servicio de Planificación de Misión permite la definición de diferentes tipos de misiones que pueden ir desde las inspecciones hasta las entregas. Estos servicios de planificación y predicción están diseñados para trabajar coordinadamente con otros servicios externos a cargo de la distribución de recursos aeroespaciales de muy bajo nivel como pueden ser las geocercas y las aerovías que presenta una determinada zona. Con esta información, el Servicio de Planificación de Vuelo es capaz de proporcionar un plan de vuelo que cumpla con todas las restricciones vigentes.

Por otro lado, el Servicio de Predicción de Trayectoria utiliza un lenguaje formal para describir las instrucciones de una aeronave de tipo quadrotor (QR-AIDL) cuyo objetivo es predecir de forma precisa las instrucciones de guía que modelan las trayectorias de este tipo de aeronaves de forma unívoca permitiendo la obtención eficiente de una trayectoria común. Gracias a este servicio se obtiene la trayectoria que va a seguir un dron para ejecutar una misión. Al tratarse de una herramienta para gestionar una flota de drones, estos servicios de planificación deben proporcionar planes de vuelo que cualquier dron pueda interpretar. Por esta motivo, la herramienta presenta un traductor de planes de vuelo a lenguaje MAVLink para que puedan realizarse las misiones planificadas por cualquier tipo de dron presente en la flota.

La función de vigilancia se encarga de estimar la posición y cinemática de los drones a partir de las medidas de los sensores disponibles y el análisis de su comportamiento. Para ello, lo primero que debe hacer este servicio es el filtrado de las medidas que llegan al sistema mediante una fusión de datos provenientes de diferentes tipos de sensores (cooperativos y no cooperativos), así como de la telemetría que envían los drones. Se ha propuesto que el Servicio de Seguimiento y Monitorización esté formado por un sistema que utiliza dos filtros en paralelo (basado en mapa, robusto) con una lógica de selección para obtener la estimación de la posición de los blancos con una mayor precisión. Por otro lado, la función de monitorización se encarga de detectar los posibles conflictos que pueda haber en el espacio aéreo que ocupa la misión ejecutada por el dron captado. Esta funcionalidad emplea las posiciones filtradas de los blancos para comprobar el comportamiento del dron frente a una serie de restricciones y limitaciones espacio-temporales.

Esta tesis presenta contribuciones en todos y cada uno de los servicios de los que se compone el sistema que puede contribuir a la gestión y control de flotas de drones.

Por último, se presentan resultados que demuestran el correcto funcionamiento de cada uno de los servicios de los que se compone la herramienta, así como su actuación en conjunto. Finalmente se presentan las principales conclusiones de esta tesis, así como las posibles futuras líneas de investigación.

Abstract

UAV-based services are still in their early stages. While future scenarios foresee urban and non-urban air spaces shared by multiple UAVs with very different missions such as logistics, emergency response or transport, among many others. The figure of the service provider, which may be a public authority or a company, appears in these scenarios. The service provider will own and operate a fleet of drones and may sometimes pay external drone operators to carry out certain specific flights. The appearance of a tool to easily define the mission, calculate the optimal trajectory and monitor the execution of the mission in real time will benefit the user who needs a service managing a fleet of drones.

This thesis proposes a system, that can contribute to the management and control of drone fleets, constituted by an architecture based on microservices that allows the planning of missions and the monitoring of drones during their execution. The architecture presented is composed by a series of microservices that contain the main logic of the system: the Mission Planning Service, the Flight Planning Service, the Trajectory Prediction Service and the Tracking and Monitoring Service.

First, this thesis describes the necessary requirements that a mission planning tool must meet in order to carry out urban and non-urban operations. This tool is capable of designing flight plans taking into consideration flight restrictions and aerospace limitations. The Mission Planning Service allows the definition of different types of missions that can range from inspections to deliveries. These planning and prediction services are designed to work in coordination with other external services in charge of the distribution of very low level aerospace resources such as geofencing and airways in a given area. With this information, the Flight Planning Service is able to provide a flight plan that meets all current restrictions.

On the other hand, the Trajectory Prediction Service uses a formal language to describe the instructions of a quadrotor type aircraft (QR-AIDL) whose objective is to accurately predict the guidance instructions that model the trajectories of this type of aircraft in a univocal way allowing the efficient obtaining of a common trajectory. Thanks to this service, the trajectory that a drone will follow to execute a mission is obtained. As a tool for managing a drone fleet, these planning services must provide flight plans that any drone can interpret. For this reason, the tool

features a flight plan translator into MAVLink language so that planned missions can be performed by any type of drone present in the fleet.

The monitoring function is responsible for estimating the precise position and kinematics of the drones from the measurements of the available sensors and the analysis of its behaviour. To do this, the first thing this service must do is filter the measurements that arrive at the system by merging data from different types of sensors (cooperative and non-cooperative), as well as the telemetry sent by the drones. It has been proposed that the Tracking and Monitoring Service should consist of a system that uses two filters in parallel (map-based, robust) with a selection logic to obtain the estimate of the position of the targets with greater precision. On the other hand, the monitoring function is in charge of detecting possible conflicts in the airspace occupied by the mission executed by the tracked drone. This functionality uses the filtered positions of the targets to check the behaviour of the drone against a number of space and time constraints.

This thesis presents contributions in each one of the services that make up the system that can contribute to the management and control of drone fleets.

Finally, results which demonstrate the correct performance of each of the services of which the tool is composed are presented, as well as their performance as a whole. Finally, the main conclusions of this thesis are presented, as well as possible future lines of research.

Contents

Acknowledgements	III
Resumen	V
Abstract	VII
List of Figures	XIII
List of Tables	XVI
1 Introduction	1
1.1 Drones	2
1.1.1 Definition, management and fields of application	2
1.1.2 Legislation	4
1.2 Operators and drone fleet management systems	5
1.2.1 Drone operator	6
1.2.2 Drone services	6
1.2.3 UTM	7
1.3 Thesis structure	10
2 Drone fleet management	11
2.1 Management and control of a drone	12
2.1.1 Ground Control System	13
2.2 Drone mission planning	15
2.2.1 Architectures of drone mission planning systems	16
2.2.2 Use cases	17
2.2.3 Type of missions	19
2.3 Flight planning	19
2.3.1 Tools for flight planning systems	20
2.3.2 Optimal planning	22
2.3.3 Trajectory prediction	24

2.4	Tracking and monitoring	26
2.4.1	Detection, localization and tracking techniques	26
2.5	Architecture of drone fleet management systems	38
2.5.1	Scheme of the architecture of a drone fleet management system	38
2.5.2	Real cases	38
2.5.3	Integration with ATM and UTM systems	41
2.6	Conclusions	45
3	Architecture proposal	47
3.1	Terminology	48
3.2	Architecture schema	49
3.3	Use cases	52
3.3.1	Pre-Flight Use Cases	54
3.3.2	In-Flight Use Cases	56
3.4	Brief explanation	56
3.4.1	Pre-flight architecture	57
3.4.2	In-flight architecture	63
3.5	Conclusions	64
4	Flight planning service	65
4.1	High level requirements and service interactions	65
4.2	Architecture schema and context	68
4.3	Flight plan calculation	68
4.3.1	Input and output requirements	69
4.3.2	Height restrictions calculation	70
4.3.3	Terrain avoidance calculation	70
4.3.4	No drone zones avoidance calculation	73
4.3.5	Integration of airspace restrictions	77
4.3.6	Time restrictions integrator	80
4.4	Results of the flight planning service	80
4.4.1	Flight Plan design example	81
4.4.2	Performance evaluation	86
4.5	Conclusions	87
5	Mission planning service	89
5.1	High level requirements and service interactions	89
5.2	Architecture schema and context	92
5.3	Input data models	93
5.4	Mission planning functionality	99
5.4.1	Waypoint calculation for simple missions	99
5.4.2	Waypoint calculation for complex inspection missions	100

5.4.3	Flight Plan translation	105
5.4.4	Flight Plan execution	107
5.5	Results of mission planning service	107
5.5.1	Waypoint calculation validation	107
5.5.2	Results of trajectory	108
5.5.3	Mission execution time	112
5.6	Conclusions	113
6	Tracking and Monitoring service	115
6.1	High level requirements and service interactions	115
6.2	Architecture schema and context	119
6.3	Functions	121
6.4	Data Models	122
6.5	Tracking and monitoring calculations	125
6.5.1	Tracker	125
6.5.2	Surveillance systems integrator	135
6.5.3	Break-in detector	136
6.5.4	Deviation detector	138
6.5.5	Obstacles detector	140
6.6	Results of tracking and monitoring service	141
6.6.1	Tracker validation	141
6.6.2	Break-in detector validation	150
6.6.3	Deviation detector validation	154
6.6.4	Obstacles detector validation	161
6.7	Conclusions	163
7	Conclusions	165
7.1	Overview	166
7.2	Contributions	170
7.3	Further work	172
	References	174

List of Figures

1.1	Unmanned Traffic Management architecture	8
2.1	Geometry of the multi-static FSR	28
2.2	ART Midrange 3D from ART	28
2.3	ARMS antidrone from Indra	29
2.4	RF-300 from Dedrone	30
2.5	Quantum’s vector series QA-100 from Quantum Technology Sciences	31
2.6	DroneShield’s omnidirectional sensor and long-range sensor	32
2.7	Grayscale image of input, detector, blob extractor and tracker output	33
2.8	Image of drone detection	33
2.9	SpynelVision from HGH Infrared Systems	34
2.10	Dedrone DroneTracker	35
2.11	RFD900 Long-Range Telemetry from jDrones	36
2.12	ADS-B system architecture and protocol hierarchy	37
2.13	Air!Squitter solution from jetvision	37
2.14	Architecture 1 of an ATM-UTM system	44
2.15	Architecture 2 of an ATM-UTM system	45
3.1	Flight plan definition schema	49
3.2	General architecture schema	50
3.3	Pre-flight services architecture	50
3.4	In-flight service architecture	50
3.5	Proposed microservice-oriented architecture schema	53
3.6	Mission Planning Service Architecture	58
3.7	Flight Planning Service Architecture	59
3.8	Trajectory Prediction Service Architecture	60
3.9	Tracking and Monitoring Service Architecture	64
4.1	Life Cycle of a flight plan	67
4.2	General schema of the architecture of the Flight Planning Service	68
4.3	Flight Plan Definition Support Architecture	69

4.4	Universal Transversal Mercator grid of Madrid	72
4.5	GeoTiff image from an urban zone of Madrid	73
4.6	Example of geofences in the VLL airspace from Spain	74
4.7	No Drone Zones avoidance calculation flowchart	75
4.8	Original input flight plan	82
4.9	No Drone Zone information	83
4.10	Other No Drone Zone information	84
4.11	Resulting flight plan	84
4.12	Height profile of resulting flight plan	85
4.13	Time profile of resulting flight plan	85
5.1	Life cycle of a mission in Mission Planning Service	91
5.2	General schema of the architecture of the Mission Planning Service	92
5.3	Specific schema of Mission Planning Service Architecture	92
5.4	Cylinder parameters and sampling points	101
5.5	Input data to get the waypoints orthogonal	103
5.6	Input data to get the waypoints constant tilt and yaw	104
5.7	MAVLink example	106
5.8	Calculated waypoints views: (left) lateral (right) zenith	108
5.9	A more complex mission with multiple inspections	109
5.10	Detailed of the mission to be performed	110
5.11	X of predicted and real flights	111
5.12	Y of predicted and real flights	111
5.13	Altitude of predicted and real flights	112
6.1	Life cycle of a track in Tracking and Monitoring Service	118
6.2	General schema of the architecture of the Tracking and Monitoring Service	119
6.3	Specific schema of Tracking and Monitoring Service Architecture	120
6.4	Global tracker structure	126
6.5	Global Map-based tracker filter structure	127
6.6	Longitudinal estimator diagram	129
6.7	Transversal estimator diagram	130
6.8	Surveillance Systems Integrator	135
6.9	Example of vertical deviation calculation	140
6.10	Detail in 3D of the mission performed	142
6.11	Detail in 2D of the mission performed and the trajectories obtained for the different trackers	143
6.12	x-y positions comparison	144
6.13	Distance error for the different trackers	145
6.14	Groundspeed comparison	146

6.15	Ground speed error for the different trackers	147
6.16	Heading comparison	148
6.17	Heading error for the different trackers	149
6.18	Height error for the different tracker	150
6.19	Mission to validate the Break-In Detector	151
6.20	Detail of non alarm in a not active geofence	151
6.21	Detail of non alarm in a geofence with heights limits that not include the flight height	152
6.22	Detail of an alarm in an active geofence	152
6.23	Break-In Detector validation	153
6.24	Second Break-In Detector validation	154
6.25	Mission to validate the Deviation Detector	155
6.26	Lateral alarms	156
6.27	Vertical alarms	156
6.28	Time alarms	157
6.29	Other alarms in Deviation detector	157
6.30	Lateral alarm validation from Deviation Detector	158
6.31	Vertical alarm validation from Deviation Detector	160
6.32	Time alarm validation from Deviation Detector	161
6.33	Missions to validate the Obstacles Detector	162
6.34	Obstacle alarm	162
6.35	Obstacles Detector validation	163
7.1	Proposed planning and surveillance architecture	171

List of Tables

4.1	Example parameters	82
4.2	Example parameters	86
5.1	Inspection types and their data	98
5.2	Example parameters	113
6.1	Absolute residual of the position	145
6.2	Absolute residual of the ground speed	147
6.3	Heading error	149
6.4	Absolute residual of the height	150

Chapter 1

Introduction

Human flight has captivated our imagination for thousands of years. The key step was taken when the Wright Brothers first flew in 1903. The evolution of aviation has been unbelievable in just one century. In 2017, more than 3.5 billion people traveled by air (Bank, 2018).

Recent discoveries in battery capacity, autonomy and on-board technology have made possible the emergence of new types of aircraft, such as unmanned aircraft (UAVs), to be also called drones in this PhD thesis. These new aircraft have new shapes, capabilities and operations that cannot be handled by the existing Air Traffic Management (ATM) system to date. This expansion of new vehicles has led to the following questions: How can drones be safely introduced into existing airspace? How can new unmanned aircraft coexist with existing unmanned aircraft? Will new uses emerge for these aircraft in the future? And these questions are giving answers that require airspace to be redesigned to allow innovation while prioritising high safety. So the first proposals to modernise airspace using digital systems are rising.

For instance, NASA's UAS Traffic Management (NASA UTM) (NASA, 2020) creates a framework for safely managing use of low-altitude airspace. Meanwhile, in Europe, the SESAR Joint Undertaking is developing U-space (SESAR, 2020b), which is endorsed by the European Commissioner for Transport and opens the continental market for low altitude drone services and aircraft. Both plans show a scenario of a coordinated and decentralized network of services that safely open the airspace to new uses. But NASA's UTM and SESAR U-space also leave implementation details open.

Infrastructure inspection, sensing, maintenance, security and precision agriculture are just some of these aerial platforms endless applications. In this scenario, to fulfil the imposed operation and safety requirements, new techniques and tools are needing, enabling to easily define and execute a flight.

The workflow to define a drone mission generally involves a client/user, who may

benefit from an aerial view and a pilot, who flies the drone following the guidelines and requirements imposed by the client. The definition of the mission is often done through a meeting where the client explains to the pilot the objective of the flight often using a map, and they agree on the flight plan taking into account drone capabilities such as flight endurance.

1.1 Drones

Nowadays the primary users of airspace are commercial jets and general aviation. As technological advances take hold, it is necessary to open the skies to many more possibilities. Specifically, in the last decade, there has been a continuous market-growth and diffusion of unmanned aerial vehicles (UAV, drones), due to the important cost reduction of the involved technologies. Therefore, it is expected that in the near future, drones will become one of the most prevalent users in the sky.

1.1.1 Definition, management and fields of application

First of all, it is important to distinguish between UAV and drone terms. A drone is an aircraft without human pilot on board that can be divided into an unmanned aerial vehicle (UAV) and remotely-piloted aircraft (RPA) (Balakrishnan et al., 2018). So, an Unmanned Aircraft Vehicle (UAV) is a type of drone (Cir, 2011). Inside drones there are many different types (Balakrishnan et al., 2018):

- **Hobby drones.** Users fly for fun and most flights are remotely controlled (DJI, 2020d). There are newer drones that can automatically follow the user or fly pre-programmed patterns. Most of these pilots are not trained to fly, but rely on what they read in pamphlets or on programs such as FAA’s Know Before You Fly (FAA, 2016)
- **Imaging and analytics drones.** These drones can perform inspections and capture imagery faster and more safely than people (DJI, 2020e). This recolected data is often used in construction and agriculture (Huuskonen and Oksanen, 2018). The flights can cover a region at a specific schedule or be ordered on demand covering a long distance.
- **Deliveries drones.** In the near future, everything from retail parcels to urgent medical deliveries may be moved by drones (Amazon, 2016), (Álvarez, 2020). In 2017, four billion parcels were ordered online for home delivery in Europe (PostEurop, 2020). Even if a small percentage of these deliveries is made in the near future with drones, this will result in a large number of drones in the airspace, that will have to operate safely.

- **Transport drones.** Nowadays, there are light planes and helicopters that provide air taxi services with platforms like Blade and Airbus Voom. But in the future, this transport will be made with drones. Urban Air Mobility (UAM) aircraft will take off and land vertically from airports all over towns and cities for passengers and emergency transport (Airbus, 2020b), (Aurora, 2020).
- **Government and military drones.** Drones are currently used by national and regional governments for law enforcement, emergency management and military training and operations (Military.com, 2020). They receive priority access to airspace when they need it and they should be able to enact airspace restrictions.
- **High altitude drones.** Self and remotely piloted drones can operate far above normal commercial altitudes for long periods. These drones will be used for satellite imaging and provision of internet access (Airbus, 2020a).

Nowadays, UAVs are used in multiple applications and operation fields (Hasanalian and Abdelkefi, 2017). Industry infrastructure inspection is one of the most popular uses of drones, such as in oil refineries, telecom (Besada et al., 2018) and electric towers (Martinez et al., 2014), wind turbines, solar plants (Matsuoka et al., 2012), nuclear plants (Torii and Sanada, 2015), etc. The use of drones has benefited in the civil infrastructures building and in inspection procedures because now these processes are faster, safer and cheaper (Greenwood et al., 2019), (Sony et al., 2019). Some other applications include patrolling pipelines, railways or electricity power lines (Santos et al., 2017), logistics (Jones, 2007), mapping and surveying of mining, pavement defects (Branco and Segantine, 2015), agriculture fields, fisheries (Ventura et al., 2016) or forestry (Tang and Shao, 2015). Other use of drones is in emergency situations to prevent and intervene in floods, hurricanes, fires or earthquakes (Nedjati et al., 2016). These emergencies also include medical emergencies such as delivery of medical supplies or transportation of vaccines (Haidari et al., 2016) and organs for transplants (Scalea et al., 2019). Drones have cameras that can be used to transmit real-time images using specific communication networks or 5G to control crowded spaces (Motlagh et al., 2017) or to respond to urban attacks. In urban environments, UAVs might help to the improvement of traffic dimensioning and Intelligent Transportation Systems acting as aerial IoT urban network elements (Menouar et al., 2017) or to make safety operations (Besada et al., 2019). One future purpose is to employ drones for the transport of people.

All these applications need tools to accelerate the creation of missions, the calculation of the optimal trajectories and the automatic execution of parts of the mission with the least human intervention, in order to obtain safe and cost-effective solutions. So it is necessary the creation of systems that enable the effective and quick generation and monitoring of missions.

1.1.2 Legislation

In view of the rapid rise of drones, a regulation for their use was created. Each country has its own regulations, although a number of rules are basic to any regulation. The Spanish regulation (BOE, 2017) is ruled by the Ministry of Public Works, and sometimes by the Ministry of Defense. Since December 30, 2017 the law that applies to the civil use of drones in Spain is the Royal Decree 1036/2017. In other countries there are regulations similar to the Spanish one. The current Spanish legislation is surely going to change in the future, as it is very restrictive. The main regulation rules are:

1. A pilot's licence is required for those who will be working with professional drones.
2. Users of remotely piloted aircraft intended for leisure flights shall comply with the following conditions:
 - Fly at least 8 km from any airport or airfield.
 - Fly is only allowed outside of controlled airspace.
 - Do not exceed 120 meters above the ground, or the highest obstacle located within a radius of 150 meters from the aircraft.
 - Fly during the day and in good weather conditions. If the aircraft weighs less than 2 kilograms, night flights are allowed as long as they do not exceed 50 meters in height.
 - The flights will always be within the visual range of the pilot (VLOS).
 - Aircraft weighing less than 250 grams may fly in the city and over crowds of people and buildings as long as they do not exceed 20 meters in height.
 - It is highly recommended to have liability insurance.
3. Any remotely piloted aircraft must carry a fireproof nameplate showing the identification of the aircraft.
4. Users of aircraft intended for profesional flights must comply with the following conditions for flights in urban areas and over conurbations:
 - The maximum take-off mass of the aircraft may not exceed 10 kg.
 - The operation must be performed within the pilot's visual range (VLOS).
 - The area to be flown must be cordoned off by the competent authority, or otherwise the drone must maintain a minimum horizontal distance of 50 metres from buildings or other structures and maintain a distance of 50 metres from persons not involved in the operation.

- The aircraft must have an impact energy limitation system installed (parachute, airbag, ...).
 - It must be authorized by the EASA through an aeronautical safety study.
5. Professional night flights will be allowed but will require the express authorization of EASA through the submission of a specific safety study. In addition, the aircraft must be fitted with devices to ensure visibility (lights, paintwork, ...).
 6. Professional flights in controlled airspace areas shall be permitted for aircraft equipped with a Mode S transponder (except aircraft below 25 kg on VLOS flights). EASA authorisation must also be obtained through a safety study.
 7. Professional flights beyond visual line of sight (BVLOS) with aircraft with $MTOW > 2$ kg will be allowed as long as the aircraft has systems to detect and avoid other airspace users and are approved by EASA.
 8. Professional flights within the extended visual line of sight (EVLOS) are permitted through the figure of the intermediate observers provided that a direct view of the aircraft is guaranteed at all times. In addition, intermediate observers must be in continuous communication with the pilot by radio and must be able to demonstrate at least the theoretical knowledge of a remote pilot.

Nowadays, European Union Aviation Safety Agency has published an opinion with the objective of creating and harmonising the necessary conditions for manned and unmanned aircraft to operate safely in the airspace (EASA, 2020). This opinion is a draft regulation and is submitted to the European Commission, which will use it as a technical basis in order to prepare an EU regulation.

1.2 Operators and drone fleet management systems

Drones that perform professional flights are handled by expert pilots, but generally there are other people or entities that are above these pilots. These people on whom the pilots depend are the ones in charge of carrying out the missions. They define a mission and the pilots carry it out. What is their objective? How do they work? All these questions will be answered in this section.

1.2.1 Drone operator

The person or entity above the pilots is the operator. The definition of an operator (Balakrishnan et al., 2018) is the person or organization that sets a mission for a drone, provides oversight of the drone in flight, and takes responsibility for the effect of the drone’s flight.

As the definition states, the operator is in charge of defining the mission, planning it and controlling its execution, in addition to being the person responsible for the drone’s flight. So there are three different phases in which the operator works. In the pre-flight phase, the operator is in charge of defining the mission, meeting with the pilot and explaining to him/her what he/she wants to do and then, he/she plans the flight. In the in-flight phase, the operator must monitor the flight of the drone to check that nothing unexpected happens during the execution, so that everything works properly. And in the post-flight phase, the operator takes responsibility for what has happened during the flight, analyses the possible problems that have occurred or simply analyses the data that has been collected during the mission.

Quite often, in order to carry out missions, the operator has a fleet of drones. Operators must maintain a description of each of their drones, with the corresponding liability insurance for each one. In addition, the documentation relating to the characterisation of the drones must have the definition of their configuration, characteristics and performance, as well as the procedures for their piloting, if the drone does not have an Remotely-Piloted Aircraft (RPA) certificate of airworthiness, or a special certificate for experimental flights. In addition, the operators are responsible for the maintenance of the drones. They are responsible for ensuring the proper performance of all of them at any time.

1.2.2 Drone services

A drone service is a set of technologies and disciplines that might interact in the functioning of a drone, such as drone engineering, service computing, mobile computing and cloud computing to carry out the intended mission. So, Drone as a service (DaaS) is the emerging market for services built around flying robots that can be remote-controlled or flown autonomously using software-controlled flight plans in their embedded systems. Commercial drone services are developing UAV services to help industries, such as agriculture, construction, search and rescue, package delivery, industrial inspection, insurance and videography, with tasks like collecting imagery and measurements and managing or broadcasting events (Rouse, 2020).

Accordingly to (Alwateer et al., 2019), each drone service may have some essential characteristics:

- At least, there should be two stakeholders, one who provides the service and another one who receives the service or the client.

- The service must meet a need, such as need of businesses, government or individuals.
- To offer the service, it should use data and information (e.g. take photos, make measures ...) or physical capabilities (e.g. carry something from one place to another using a drone).
- The services should be broad enough to cover most applications of the technology and uses of the drones.

As there is a wide range of drone services in design, there are a lot of architectures, systems and functionalities that define each application. More versatile platforms, enhanced sensors, autonomous flight, consolidated UTM systems to enable coordination with traditional Air Traffic Management, authorization and control support-tools, indoor operations ... are some of the topics in research. Among the multiple challenges, there is a special need for tools enabling better coordination and management of the available drones, in particular for drone operators governing fleets of drones. As fleets are forecasted to increase, the possibility of sharing resources to e.g. provide support to multiple urban services by a centralized fleet (thus optimizing acquisition, renewal and maintenance) or to put resources from different public owners together to optimize their use could be possible. This is one of the possible approaches that emerge to obtain a drone as a service architecture.

1.2.3 UTM

Nowadays, aircraft traffic is mostly managed by air traffic controllers. Each of them is responsible for a sector, keeping aircraft safe by talking with pilots using radio communications. But, with the growth of unmanned aerial vehicle market, the air traffic will increase by several orders of magnitude (SJU, 2016). To handle this situation, the traditional air traffic control and management must shift to a model that includes a digital system which can monitor and manage this activity. That system is known as Unmanned Traffic Management or UTM and its architecture can be seen in Figure 1.1.

(Balakrishnan et al., 2018) defines UTM as *A networked collection of services that work together to safely direct self-piloted air traffic based on common rules.*

As can be extracted from the definition, UTM is not a single, central system that mandates one way of operating for everything, it is a framework. UTM is built to enable future applications. The system that must be designed should remain relevant as technology progresses and market needs mature without knowing what's coming next. UTM frameworks around the world use the principle of distributed authority (Administration, 2020). This enables to have more service providers to the system. This decentralization makes the privatization of the cost of serving

Figure 1.1: Unmanned Traffic Management architecture

and adapting to the market needs, but the government regulators may ensure that safety, access and equity are maintained.

Aircraft could communicate freely with their service providers, and are in charge of coordinating with the rest of the network to make efficient decisions based on specific flight objectives. The former air traffic controllers will become airspace managers.

UTM allows the same provider to serve different needs in different geographies at different times. Regulators can adapt requirements to match their local needs while the operators can select the UTM service providers that need to complete their missions. The creation, updating and deployment of services can be done by the UTM service providers. A operator can choose to build, certify and supply its own service and another one can find these services in a market-place. Providers will be responsible for coordinating with each other (NASA, 2018).

Several countries and transnational bodies have already adopted this general approach as the basis for their own UTM applications, but they differ on how authority should be distributed:

- **Europe - U-Space (SESAR).** The main functions are provided by U-space Service Providers (USP). These require to exchange information and coordinate through a System Wide Information Management (SWIM). The providers must also communicate with a U-space system manager, which acts as a centralized coordinator. Other providers are responsible for non safety critical services as well as data on weather or terrain. U-space is scheduled in four stages, called U1-U4 (SESAR, 2020a). This architecture, like the other architectures present in this list, is still under discussion.
- **United States - UTM (NASA).** In NASA's model (NASA, 2018), private Unmanned Aircraft Systems Service Suppliers (USS) are certified by FAA (Federal Aviation Administration). Each USS has the responsibility for exchanging data and coordinating with the others USS. The data needed to coordinate the model with the ATM system goes through the SWIM, which is run by the FAA and stores information such as flight plans. The Flight Information Management System (FIMS) is also running at the same time by the FAA to coordinate USS providers with traditional air traffic management and the national airspace system. The first USS have already been approved to provide Low Altitude Authorization and Notification Capabilities (LAANC) to operators who want to fly drones in controlled airspace near airports.
- **China - (UOMS).** The China CAAC has specified the use of UAS Operation Management System (UOMS) (CAAC, 2018). This system has several Unmanned Aircraft Cloud System (UACS) providers which are responsible for the final link to the operator. These UACS provides alerts, geocaging, registrations and vehicle location services. Nowadays, there are seven UACS providers approved by the CAAC.
- **Japan - UTM (JUTM).** The system in Japan is being built by the Japan UTM Consortium (JUTM) (Association, 2017) and a national UTM project founded by NEW Energy and Industrial Technology Development (NEDO). The system is composed of one FIMS, several UAS Service Providers (UASSP), a layer of Supplemental Data Source Providers (SDSP) and operators. The FIMS is responsible of managing all flight plans, handling emergency alerts and providing avoidance instructions. In 2017 the JUTM demonstrations began, while individual systems were demonstrated in 2018. The first full system demonstration was in 2019 and the implementation will be in 2020s.

1.3 Thesis structure

The contents of this thesis are organized in six chapters, outlined next. Chapter 2 contains literature reviews covering several areas. A review and evaluation of the different services that are part of a system like this is included here. Also in Chapter 2 there is a small study of the evolution of ATM systems to UTM systems and the differences between both systems, as well as their architectures. Finally, in this chapter there is a summary of techniques used for the distribution of tasks between drones.

The architecture of the proposal of control and management system of fleets of drones is included in Chapter 3, where the schemas and a comparative of architectures is presented.

Chapter 4 and the next two chapters deal with each of the services that make up the overall system architecture. Although a small division has been made to present these chapters. On the one hand, the Flight Planning Service (Chapter 4) and Mission Planning Service (Chapter 5) are presented under the name of pre-flight services because, as in the architecture presented, they are services that perform their tasks during the phase prior to the flight of the drone, the flight preparation phase. On the other hand, the Tracking and Monitoring Service (Chapter 6) is presented under the name of in-flight services, as in architecture, since its performance occurs during the flight phase of the drone. This division allows to easily differentiate the moment of action of the services and to put stricter computation requirements to the services that act during the flight phase than to the services that work in the pre-flight phase. This is because services in the flight phase work in real time. In each chapter the service is explained from its objectives to its architecture. Also, the service design is described as well as the results obtained for that system.

Finally, the conclusion of this thesis, providing future lines of work and a list of contributions, are included in Chapter 7.

Chapter 2

Drone fleet management

The drone market will explote in the next thirty years in agriculture, energy, public safety/security, e-commerce/delivery and mobility and transport according to the European Drones Outlook Study (SJU, 2016), to achieve 7 million consumer leisure drones to be operating across Europe. A fleet of 400.000 UAVs is expected to be used for commercial and government missions in 2050. The challenges are multidisciplinary and they include more versatile platforms, improved sensors, autonomous flight planning tools, consolidated UTM (Unmanned Traffic Management, explained in Section 1.2.3) systems to be able to coordinate with traditional Air Traffic Management (ATM), authorization and control support-tools, swarm behaviour or improved indoor navigation. Therefore, a huge number of research, technical, social and political barriers still need to be overcome in order to incorporate drone-based services in our daily life.

On the one hand, in the public sector, institutions and state services are making grow their own fleets, in an independent and non-coordinated manner. The public service authorities, such as police, fire fighters, park maintenance or building inspection services, are detecting needs that may be better implemented through drones, so they are deciding to acquire them, organize personnel training and, at the end, become drone operators with the purpose of flying their own drones. Each public service will have its own drones, so they will be independently managed. The coordination among them must be handled at a higher organization level, thus being less immediate and cost-efficient, and undoubtedly more complicated.

On the other hand, there is an increasing number of drone operators which are fighting for having their own niche market. Specialized equipment may be costly, non-supervised flights are not allowed yet due to safety and legal constraints, thus providers do need to leverage the existing resources and pilots. Nowadays, drone services are peer-to-peer negotiated through complex procedures, which usually include the preparation of requests for quotation and accurate scheduling, making it

complex and costly to integrate drones in procedures.

2.1 Management and control of a drone

A drone is a pilotless aircraft operated by remote control. The state of the art related to drone design and development has typically taken two distinct approaches (Giones and Brem, 2017), (Wolf, 2017). The first approach is to convert an existing aircraft into a drone. Conversions from traditional to unmanned aircraft have required major modifications to existing aircraft. Many times it is necessary to develop additional and unique hardware and software systems, both in the aircraft and in a ground station. In addition, many of these conversions also require the development of new or modified aircraft control laws. These control laws, or algorithms, are part of the additional hardware and software systems on board the drone and are used to govern the drone's operations. Due mainly to the above-mentioned modifications, this first approach is often costly and technically complex. For this reason, the conversion of each aircraft into a drone has typically been a one-time development effort, with no generic kits being used to convert multiple aircraft types. Similarly, once an aircraft has been converted to a drone using conventional methods, it is impractical and prohibitively expensive to convert it back. In other words, there are no known aircraft that can be dismantled and converted into drones, which would simplify the conversion process.

The second approach used for the design and development of the drones is to design and manufacture the drone from scratch. Therefore, these vehicles are designed not to have a pilot on board. These drones are traditionally known as Remote Pilot Vehicles (RPV) and Unmanned Aerial Vehicles (UAV). Both are vehicles that are originally designed without a pilot on board.

From these approaches to design and develop a drone, they have evolved over time and more and more technologies are incorporated. This has made that different types of drones according to the control method are found, since there are models that are equipped with different technologies such as GPS systems, infrared sensors, high resolution cameras ... Taking into account this characteristic, drones can be classified according to the control method in:

- **Autonomous drone.** Those which, as the name suggests, do not need a remote pilot to control the flight. They are equipped with systems that allow them to plan their route and have the flight completely programmed. In addition, they usually also have GPS location systems to keep them located in real time.
- **Remote-controlled drone.** A remotely controlled drone is directed by a radio control manipulated by the pilot. It is the most common and used

control method.

- **Supervised or monitored drone.** These are the ones between the autonomous drones and those directed by radio control. That is, they can do certain things autonomously, but always under the supervision of a pilot.

The typical drone system consists of an air vehicle, or air system, and a ground station. The ground station is adapted to provide positive control of the drone throughout its flight. In addition, any associated electrical or mechanical devices are also controlled by the ground station. These electrical or mechanical devices are used to operate things like the landing gear, camera, spoilers, slats, speed brakes, front wheel steering and many other things.

The most common method of controlling a remotely piloted aircraft usually employs a central command computer system. These are physically located at the ground station, which transmits control signals to the remote piloted aircraft. The drones can be adapted to process these signals or, alternatively, they can execute a pre-programmed flight plan.

2.1.1 Ground Control System

An increasing interest has been driven towards the definition of adequate human-computer interfaces both for Ground Control Systems (GCS) and others, as accidents involving drones due to operators' errors are somehow frequent. For example, (Bevacqua et al., 2015) reports a study meant to design GCS interfaces for multiple RPAS control, with the objective of enhancing operators' performance. (Eiband and Kern, 1993) designs a generic drone control system essentially comprising a converted aircraft or other air vehicle and a ground station from which the drone is remotely controlled. It also shows a large amount of means for transferring information and data between the ground station and the drone. (Rath et al., 2007) defines a ground control station to control an unmanned air vehicle during a manual mode of operation. This ground control station is composed of a processing unit, a telemetry/telecommand module, a user control module, a graphical user interface and a wireless datalink subsystem.

As it can be seen, there are many designs of different types of ground control systems in the literature. So a person with the need knowledge can build one from these designs, but drone companies already provide their own ground control systems so that people can operate their drones. In addition, many of these companies allow you to build your own version of a ground control system from their base software, using Software Development Kits (SDKs) consisting of a package of tools and data that enables programmers to develop programs in a specific language or for a specific platform or application. In this way, GCS can be developed with the specific needs of each user.

A ground control system is typically a software application running on a ground-based computer that communicates with a drone via wireless telemetry. It displays real-time data on the UAVs performance and position and can show many of the instruments that have a real plane. A GCS is also used to control a drone in flight uploading new mission commands and setting parameters. Finally, it is also used to monitor the live video streams from the cameras of the drone.

There are dozens of different ground control systems. Below is a small selection of some of the most important:

- **On desktop.**

- **SDKs**

- * **DJI Windows SDK.** The goal of (DJI, 2020b) is to give any developer with UWP application experience the knowledge and understanding required to create "ad hoc" applications using DJI's technology.
 - * **Parrot SDK.** (Parrot, 2020b) will help the pilot to connect, pilot, receive stream, save and download medias, send and play autopilot flight plans and update the drone. This SDK is mainly written in C and provides libraries for Unix system.

- **Applications**

- * **Mission Planner.** (Ardupilot, 2020b) is a ground control station for Plane, Copter and Rover. It is compatible with Windows. Mission Planner can be used as a configuration utility or as a dynamic control supplement for the autonomous vehicle.
 - * **APM Planner 2.** (Ardupilot, 2020a) is an open-source ground station application for MAVlink based autopilots including APM and PX4/Pixhawk that can be run on Windows, Mac OSX, and Linux.
 - * **QGroundControl.** (Dronecode, 2020) provides full flight control and mission planning for any MAVLink enabled drone. Its primary goal is ease of use for professional users and developers. All the code is open-source source.

- **For smartphone or tablet.**

- **SDKs**

- * **DJI mobile SDK.** The goal of (DJI, 2020c) is to give any developer with iOS or Android experience the knowledge and understanding required to create world changing applications using DJI's technology.

- * **Parrot Ground SDK.** (Parrot, 2020a) allows any developer to create its own application. It is available as a ready-to-compile source code, an iOS CocoaPods and an Android AAR.
- **Applications**
 - * **DJI GS PRO.** (DJI, 2020a) allows to run the drone program efficiently, performing automatic flight missions, managing flight data in the cloud and collaborating on various projects from an iPad.
 - * **MAV Pilot 1.4.** (Drones, 2020) is a GCS in the pocket that supports predominantly ArduPilot autopilot on iPhone/iPad. Supports for Plane, Copter and Rover vehicle types.
 - * **Andropilot.** (Hester, 2020) is an Android based Ground Control Station.
- **Others.**
 - **Desert Rotor.** (Technology, 2020) is a leading manufacturer and supplier of professional, portable ground control stations (GCS) for unmanned aerial vehicles (UAVs). Their complete GCS solutions are compatible with almost any fixed-wing or multirotor UAV and many drone autopilots.

To sum up, the commercial offer of tools for Ground Control Systems includes tailored proposal from drone hardware manufacturers (DJI, 2020b), (DJI, 2020c), (Parrot, 2020a) ... or components (ArduPilot, 2020a) and also specific software platforms (ArduPilot, 2020b), (Dronecode, 2020) ... These tools delivered by drone manufacturers allow the user to configure the path to be followed by the UAV through a set of waypoints on a map or 2D view and to add the flight plan to a GCS to automate the flight. More specialized platforms (Technology, 2020) provide automate mapping, photo flights, 3D model creation ... Existing tools require the planner to specify the maneuvers to be performed, they are not addressed by any resource assignment responsibility or multi-drone management.

2.2 Drone mission planning

Most of the tools used in the Ground Control Systems are designed to enable a fast and easy way for the users to interact with them, to the detriment of more complex scenarios and systematic mission definition. Additionally, there are several problems with such tools, such as the tools are intended to be used with the drones of the manufacturer, they are not open to be used with any arbitrary drone or the interface is oriented to simple flight plans defined through a time-ordered waypoints sequence,

and to manual definition of other data such as altitudes or heading. Some tools, as DJI, only allow to create most advanced automatic flight plans to the professional pilots, it is not allowed to all users.

For these reasons, studies are emerging which develop and design systems to design missions for drones, these are the mission planning systems. Currently, in the literature it is possible to find global or partial mission planning systems or solutions for their components with different approaches.

The mission type obviously determines the type of flight to be completed (González-Jorge et al., 2017), with clear differences for local (Visual Line of Sight VLOS) and remote (Beyond Visual Line of Sight, BVLOS) operations. All these systems need tools to accelerate and partially automate the creation of missions, the optimal calculation of the best trajectories and the automatic execution of most parts of the mission with the least human intervention for obtaining cost effectiveness. But some parts of the flight plan, the most critical ones, quite often have to be piloted by humans because of the higher adaptability of human pilots to mission requirements and changing environment.

2.2.1 Architectures of drone mission planning systems

As it has been said, there are many drone mission planning systems architectures in the literature. In (Besada et al., 2018), authors describe a tool to define and handle real-time inspection missions. These missions are easily configured through the definition of simple geometric volumes around the target to inspect. The system enables dynamic mission tailoring and adjustment in case of a modification is needed. The objective in (Zorbas et al., 2016) is to determine the optimal observation altitudes minimizing the cost of the flight while ensuring the surveillance of all targets. In (Areias et al., 2018) a platform that offers an abstraction layer between the end-user and the drone is described. This platform implements an event-driven control platform that achieves automated control of multiple drones, while performing high-level control operations for the drones and delivering real-time information (telemetry). (Tseng et al., 2017) studies flight mission planning together with autonomous recharging optimization and proposes different algorithms to solve the battery problem of the drones. A mission planning model as an orienteering problem is analysed in the study of (Evers et al., 2014), specialized in the uncertainty in the fuel usage between targets because of the weather variations. Not only horizontal (2D) mission planners are studied in the literature, but also vertical mission planners are explored, as is the case proposed by (Cavalcante et al., 2017). In this paper, the objective of the mission planning system is to solve the intralogistics of a UAV equipped with a robotic gripper in an industrial environment. The system generates a complete mission with all the stages of the process, which goes from the delivery of the inputs collected in the warehouse to the delivery to the customer. In

this paper a technique with task programming is being compared with basic heuristics. (Mangharam, 2019) details a tool that allows an operator to plan missions for multirotor UAV fleets with complex time constraints where missions are translated into Temporal Signal Logic.

(Thomason et al., 2017) discusses how to facilitate the task of the teleoperator. In the proposed system, an interface provides real-time views adapted to the environment that are automatically configured to improve safety and operation, because nearby objects can generate risks of collision and the occlusions that can result make precise handling difficult. The system uses simultaneous location and map-based reconstruction (known as SLAM) and combines the position and orientation of airborne robots and 3D point cloud information to modify the user’s viewpoint to maximize visibility. A description of a hierarchical dynamic mission planning framework can be found in (MahmoudZadeh et al., 2018). This framework is optimized for autonomous underwater vehicles, but the concept of task priority assignment and the regeneration of optimal trajectories on unexpected changes may be applied to UAV.

It can be concluded that there are many systems in the literature that deal with mission planning as it is a subject that involves great complexity. These systems must be able to operate in the presence of different types of missions for different environments.

2.2.2 Use cases

These architectures can be used in many environments with different mission objectives. The main use cases are described below:

- **Agriculture inspection.** The use of drones are achieving a better agricultural productivity and food management because a precision agriculture monitoring at large scales is getting. An approach which works with image statistics of the selected region using freely available satellite data along with drone data is proposed in (Murugan et al., 2017). Missions for this use case are typically 2D patterns covering specific areas.
- **E-commerce and delivery.** Despite the heavy impact of the weight on the battery duration, the delivery of small objects is already a reality such as the service of transportation of small medicines and blood in Africa using a fixed wing UAV described in (Zipline, 2017). For the e-commerce there are several proposals of the big technology enterprises, such as the Prime Air service of Amazon (Amazon, 2016). A typical mission for this use case will directly link delivery destinations and return to the origin, which will usually be a distribution centre.

- **Public safety and security.** Some drones can be used counter terrorism through different ways. If the drone incorporates X-ray cameras, IR camera and metal detectors such as the one presented in (Hamza et al., 2017), it is possible to detect metal, tracking its location through GPS and making video of its target. Characteristic missions of this use case will either automatically cover areas to perform a search, and manual missions for precise follow-up or monitoring.
- **Energy inspections.** The procedure of using a UAV in the detection of different failures of photovoltaic modules is much faster and effective than when utilising traditional methods as can be checked in (Matsuoka et al., 2012). Conversely, long periods of observation and large number of sensors are required to monitor environmental gases for risk assessment indoors and outdoors. But with the use of drones, this process is easier as demonstrated in (Rossi et al., 2014), (Kersnovski et al., 2017). Another application of UAV can be observed in soil pollution monitoring with an example of a multisensor approach for copper detection (Capolupo et al., 2015) and an example of a UAV that inspect contaminated areas such as the fission reactors for leakage detection, storage areas of nuclear sources or hazardous scenarios of nuclear disasters (Vale et al., 2017), (Tang and Shao, 2015). This use case is characterized by 2D and 3D missions or patterns.
- **Transport inspections.** Drones may be useful to inspect roads to detect early signs of erosion and pavement distress such as the studies (Rau et al., 2011), (Niethammer et al., 2011) focused on landslide detection and monitoring or (Carvajal et al., 2011) on road ditches. Another use is the railway infrastructure inspection such as the proposal of (Máthé and Buşoniu, 2015) where a camera-based sensing and control methods are oriented to railway track following for the sake of recording the infrastructure. Linear flights following infrastructure are the typical missions performed in this use case.
- **Construction management.** A drone can be used to provide visual assets to verify the safety checklists in two different projects as the particular application in safety inspection on construction sites proposed by (De Melo et al., 2017). Another common application is building inspection where (Richman et al., 2017) defines a method to perform an inspection of a rooftop by using a UAV and to extract information from a damaged area. 3D missions, either manual or automatic, are the most common missions in this use case.

2.2.3 Type of missions

These systems have to take into account the type of mission to be carried out, since depending on it the planner must act in one way or another. The types of existing drone missions are:

1. **Visual Line of Sight (VLOS).** Commonly known as flight within the pilot's visual range. Operations within the pilot's visual range are those in which the pilot maintains direct visual contact with the drone, without the aid of optical or electronic devices. These drone flights are the most common and do not require any special authorization from the regulator (except for those required for professional, commercial and experimental operations).
2. **Extended Visual Line of Sight (EVLOS).** Drone flights within the increased visual range are operations in which direct contact with the aircraft is made with the help of observers who are in constant radio contact with the pilot. Observers must have a professional pilot's licence, although a practical drone rating is not mandatory. This means that the support people or observers will have the theoretical knowledge required by the regulator for professional operations (advanced drone pilot course).
3. **Beyond Visual Line of Sight (BVLOS).** In these operations, the pilot does not have direct visual contact with the drone nor he/she is assisted by an observer. The pilot performs the control thanks to the broadcast of the remote control station, which allows the link with the on board command for an effective control. For this type of flight with drones, it is necessary that the drone has systems that allow its pilot to detect and avoid other users of the airspace. These systems must be approved and permitted by the regulator.

Regardless of the type of mission being carried out, current commercial systems for UTM (i.e. Airmap (Airmap, 2020) or Unifly (Unifly, 2020)) allow for declaration of basically two types of operation:

1. **Waypoint based flight plans:** specially for automated BVLOS operations.
2. **Flight Areas/Volumes:** specially for VLOS/EVLOS and manual operations.

2.3 Flight planning

In order to carry out this mission, it is necessary to transform it into a flight plan that can be executed by a drone.

2.3.1 Tools for flight planning systems

The academic world has also focused on the objective of generalizing and abstracting the definition of missions, as can be seen in (Ruscio et al., 2016), in which the authors developed a Domain Specific Language that allows for the establishment of mission specifications and the prediction of trajectories. This contribution is similar to efforts in commercial aircraft trajectory predictions for Air Traffic Management (Frontera et al., 2014). These efforts have been extended to unmanned vehicles (Frontera Sánchez, 2016), where a description of a complete language hierarchy and associated trajectory prediction/computation engines is included. (Frontera et al., 2019) summarizes the multi-rotor oriented trajectory prediction basis.

A lot of research has been performed in the field of trajectory predictors, specially on the prediction of commercial aircraft trajectories and the definition of means to describe missions, flight plans and trajectories, such as (López-Leonés et al., 2007) or (Lopez Leones, 2008). These trajectory predictors are used to optimize the mission or as a support tool for air traffic controllers to enable safe and efficient operations. The idea behind these predictors is to calculate flight paths and trajectories minimizing the costs of a flight, either by following predefined routes, respecting no-fly zones, taking advantage of the wind and selecting ideal flight levels. This idea has been approached through different approaches such as dynamic programming (Waller et al., 1990), genetic algorithms (Hu et al., 2004), heuristic search algorithms (Grabbe et al., 2009), (Frontera et al., 2016), or a combination of both genetic algorithms and heuristic search (Qu et al., 2005).

But the problem of dynamics (fixed wing vs. multicopter), and the limitations of accuracy are very different from those of the Air Traffic Management area. There are also examples of research in the prediction of the trajectory of unmanned aircraft depending on the type of UAV to be controlled and the application to which it is being applied (Santana et al., 2014), (Pastor et al., 2010). Some examples in literature use an open source platform for mission planning of autonomous quadrotor called FLYAQ (Bozhinoski et al., 2015). This tool allows different types of movement and sensor control, translating from a high level specification of the mission into a Quadrotor Behaviour Language (QBL), similar to the ones described for ATM applications. FLYAQ does not allow the definition of automatic missions for different geometric shapes, as well as the visualization or planning of 3D paths, although the extension made in (Ruscio et al., 2016) includes the possibility of defining points, lines (with intermediate points), polygons and volume primitives (polyhedrics).

The specification of missions by non-expert users (e.g. firefigther, rescue workers, police, etc.) is being provided by other support tools (Bozhinoski et al., 2015). There, the detailed flight plan that each of the drones must carry out to fulfill the specified mission is automatically generated avoiding collisions between the drones and the obstacles, as well as ensuring the preservation of the no-fly zones. Another

example of this approach is explained in (Besada et al., 2018), where a system for partially automated design of infrastructure inspection is described, which, from 3D primitives derives flyable flight plans expressed in terms of the language hierarchy described in (Frontera Sánchez, 2016) and even translates them to a MAVLINK specification to be uploaded in a compatible drone for its execution.

Neptus, a specific system for defining missions, is described in (Pereira et al., 2006), where mission planning is focused on heterogeneous teams of vehicles such as autonomous, remotely operated underwater, surface, ground and air vehicles. The system has a pre-defined set of manoeuvres and automatic routes to inspect static targets are not allowed. On the other hand, a system for defining specific missions that mixes manned and unmanned aircraft with limitations on use with commercial drones can be seen in (Valenti et al., 2004). Another route planner that focuses on surveillance of moving targets is described in (Gentilini et al., 2016). In this, the mission is calculated by minimizing a cost function adapted to follow moving targets. Other types of mission planners are presented in (Sun et al., 2015), where the mission is oriented to obtaining a path to a non-static target, so it is recalculated in real time.

Finally, an analysis of the requirements for a planning system can be seen in (Nelson, 1995), while (Tulum et al., 2009) outlines ways to increase the probability of mission success. To increase this probability, the dynamic constraints of the drones have to be taken into account since each drone has a minimum and a maximum altitude and the power consumption of the drone is related to these altitudes. The distribution of resources is also important for mission optimization. For example, in (Zorbas et al., 2016) the problem of finding the best drone locations to minimize the cost of energy consumption is addressed.

Most of the applications mentioned in this section are based on the definition of "ad hoc" procedures for the design of missions that assume a safe environment. As it is known, with urban environments this cannot be assumed as they are made up of a large number of obstacles, flight-restricted areas, etc. In addition, there is the need to coordinate operations to avoid collisions and impose physical restrictions on flights in the form of corridors, airways and explicit air rules if air traffic begins to grow (Balakrishnan et al., 2018). In order to ensure flight safety, in the last few years, a collection of concepts and tools have been developed, such as those related to the FAA-NASA UTM (Unmanned Traffic Management (FAA-NASA, 2020)) program, or to the European U-Space concept from SESAR program (SESAR, 2020b). Also, there are so many companies developing UTM solutions (most of the relevant actors are included in GUTMA association (Association, 2020)), and the flight regulators (i.e. ICAO, AESA, etc.) are working in the problem of opening the sky for these new actors. The JARUS (JARUS, 2020) group is working with respect to regulators, which is defining a methodology for risk assessment (called SORA (JARUS, 2019)) with the final aim to be able to automate flight authorization, as part of the UTM

process.

2.3.2 Optimal planning

Flight planning is a complex problem because it is necessary to identify the goals of a mission and generate a set of tasks to be performed for each UAV. This process was carried out by human operators until the techniques weren't advanced enough (Nelson, 1995). These techniques have led to full automation these flight planning processes, based on their formulation as a mathematical optimization problem. To formulate it, different paradigms and representations have been used.

A tool for developing general planners was developed by NASA's Ames Research Center. This tool, called EUROPA (Extensible Universal Remote Operations Planning Architecture (Bedrax-Weiss et al., 2005), (Barreiro et al., 2012)) consists on a constraint-based temporal planning paradigm. In this tool, the mission can be represented as a set of assets, targets, paths, relevant mission elements, actions and constraints. The actions can modify the state of assets, targets, paths and relevant mission elements, while the constraints represent the relationships among these items. EUROPA includes inference and search algorithms used in getting a flight plan from the mission specification. Following the same paradigm is Temporal Action Logic (TAL) which is used to calculate actions and their effects. A restricted form of TAL is used as input in Threaded Forward-chaining Partial Order Planner (TFPOP (Doherty et al., 2013)) which uses a backtracking algorithm to find a compatible plan.

The problem of obtaining a flight plan is similar to the Travelling Salesman Problem (TSP) (Laporte and Martello, 1990), the orienteering problem (Evers et al., 2014), the Vehicle Routing Problem (VRP) (Pohl and Lamont, 2008), (Doherty et al., 2013) or derivatives problems of these. The Travelling Salesman Problem determines the shortest path visiting each target once and returning to the origin. If it is not necessary to visit all targets, the orienteering problem is useful because it returns the path with the highest score given a limited amount of resources such as fuel or time. The problem with these methods is that only a drone is computed at a time. If there are several drones, the VRP must be used but the optimal solution can not be obtained in reasonable time (Laporte and Martello, 1990). But from the VRP Mixed Integer Linear Programming (MILP (Karaman and Frazzoli, 2011), (Kim et al., 2007), (Leary et al., 2011)) solutions may be used. This problem gives near-optimal solutions in a reasonable time (Kim et al., 2007).

Nowadays, the most widespread solutions for obtaining optimal planning are genetic algorithms (Pohl and Lamont, 2008), (Sahingoz, 2013).

UAVs fly within a range of low altitudes, which in an urban environment is full of obstacles that the drone has to avoid. So the flight planner must find the optimal path to avoid these obstacles. In the literature there are many examples of

algorithms to derive a collision-free path in two and three dimensions, as outlined in (Frontera et al., 2017).

The most typical obstacles in a two-dimensional environment are polygons. The 2D Euclidean Shortest Path (2ESP) between two points is a polygonal path whose inner vertices are vertices of the obstacles (Van Kreveld et al., 2000). The easiest way to solve a 2ESP is building a data structure with n vertices that belongs to the polygonal obstacles and m edges connecting every pair of vertices, which is called visibility graph. The time to build it that some algorithms have is $O(n^2)$ (Asano et al., 1986), (Ghosh and Mount, 1991) and other has $O(m + n \log n)$ (Welzl, 1985). After obtaining the visibility graph, a Dijkstra's algorithm can solve the 2ESP in $O(m + n \log n)$. Instead of building the visibility graph, there is the chance to compute the shortest path map, which is a division of the plane in regions that share the same vertex sequence in their shortest path to a particular point. The time to find a shortest path is $O(n \log n)$ in (Hershberger and Suri, 1999).

There are other algorithm solutions such as the 2ESP problem inside a single polygon with a solution in $O(n)$ time (Hershberger and Suri, 1999), (Guibas et al., 1987). The Base Line-Oriented Visibility Line (BLOVL (Omar and Gu, 2009)) algorithm reduces the complexity of the problem by limiting the number of obstacles taken into account. And others give the near-optimal path solutions (Clarkson, 1987), (Chew, 1989) with a lower cost of computing.

A more complex problem is found in a three-dimensional environment because obstacles are polyhedra and the 3D Euclidean Shortest Path (3ESP) is a polygonal path whose inner vertices lie on vertices or edges of the obstacles. At the beginning, the problem was dealt as NP-hard (Canny and Reif, 1987), but later a single-exponential algorithm was found (Reif and Storer, 1994). Depending on the number of obstacles and their shapes, the time to obtain the optimal path can be $O(n^2 \log n)$ when only a convex polyhedron is the present obstacle (Mount, 1985), or if there are two the time is $O(n^3 \log n)$ (Sharir and Baltsan, 1986). A relationship between the number of vertical polyhedrals with k different heights and the time to obtain the optimal path has been calculated in (Gewali et al., 1990) and is $O(n^{6k-1})$.

There are approximations that give near-optimal paths such as (Choi et al., 1997) and (Choi et al., 1995) where a visibility graph is built using these vertices in polynomial time and the approximate shortest path is computed in that graph. Finally, there are algorithms that take into account UAV movement constraints. These algorithms search the shortest path using search methods (Qi et al., 2010), (De Filippis et al., 2012), transforming the 3ESP problem into a 2ESP problem (Omar and Gu, 2010), (De Lellis et al., 2013), (Xiao-dong et al., 2013) or using generic algorithms (Roberge et al., 2012). (Frontera et al., 2017) provides another approximation based on these ideas.

2.3.3 Trajectory prediction

UAVs have to communicate their intended trajectories but there is not a standard format to express it, so own ad-hoc solutions have been used by manufacturers and organizations. MAVLink (Micro Air Vehicle Link (QgroundControl, 2019)) is a popular protocol which allows defining a flight plan and also provides mechanisms for communication between GCS and UAVs. Instead, in the Paparazzi UAV project (Brisset et al., 2006) more complete trajectory descriptions have been used to define a sequence of blocks, where a block is the horizontal profile of the flight with navigation instructions and optionally vertical profiles. While other authors have provided more flexibility to trajectory description by allowing more complex specifications such as iterations and forks in the flight plan (Santamaria et al., 2012).

One of the main objectives is to obtain a standardized method to express aircraft intent information (Vilaplana et al., 2005). First approximation was made by Boeing Research & Technology Europe (BR&TE) developing the Aircraft Intent Description Language (AIDL (López-Leonés et al., 2007), (Lopez Leones, 2008)) which is a formal language that comes from the generic mathematical formulation of the trajectory computation process. This AIDL was designed to described trajectories of fixed-wing aircraft so it is not useful to describe multirotors trajectories. Whereas other authors have designed systems that somehow rely on the existence of the quadrotor AIDL to describe trajectories of quadrator aircraft (Tarhan et al., 2014), (Maza et al., 2014).

The trajectory prediction research in (Besada et al., 2013) is based on the definition of a hierarchy of formal languages where languages extend other languages. Each language is made up of lexemes and a grammar that enables the rules to be defined to combine these lexemes in a way that is compatible with the needs and constraints of trajectory prediction. The trajectory prediction is modelled as the process of calculating intended trajectory descriptions compatible with higher-level specifications, where operational context, aircraft performance and weather are taken into account. The lowest level of trajectory specification is understood to be the sampled trajectory.

Aircraft Intention Description Language (AIDL) is a formal language designed to describe the aircraft intention information rigorously but flexibly, through a list of instructions covering all aircraft movement parameters (all degrees of freedom (DoFs) must be closed for any instant of time on the trajectory). AIDL instructions (lexemes of this language) represent basic commands, orientation modes and control inputs over a certain interval, which are delimited by triggering conditions (triggers). The instructions are combined following the AIDL grammar that allows certain combinations of the instructions because this ensures that the trajectory is computed uniquely using the appropriate contour information such as weather models and aircraft performance models. In addition, AIDL allows iterative calculations to

close a specific type of triggers (auto-triggers) with their conditions, because they can be used to define the final dynamic state conditions for the instructions.

Intent Composite Description Language (ICDL) is a formal language used to represent complex and/or parametric aircraft intents as a result of combining AIDL instructions. ICDL extends AIDL by introducing the concept of composites, which are combinations of AIDL instructions that allow some parts of the intention to be unspecified. Therefore, some (or even all) DoFs can be left open and the parameters of AIDL instructions may not be fully defined. A composite can be defined as a combination of AIDL instructions and with a syntax similar to AIDL, but with some rules removed or modified. Several operations on composites, which provide new semantics not possible in AIDL, are defined. Composites are used to represent a sequence of aircraft maneuvers or parametric aircraft intention with open parameters and/or unspecified operations available for a later optimization process. The ICDL has the advantage of being more human readable than AIDL, which makes the generation process easier for a human operator. Therefore, both AIDL and ICDL can be used to represent aircraft intent information with different levels of detail, completeness and properties.

Finally, the Flight Intent Description Language (FIDL) is used to express flight intention information, such as flight plans, restrictions or targets. As it extends ICDL and FIDL, it has the advantage of being able to provide descriptions of different parts of the flight at different levels of detail. Thus, the ascent and descent phases of the flight could be completely defined (closing all DoFs), while the cruise phase is simply defined as a list of route points.

The trajectory prediction architecture is based on decomposing the whole trajectory prediction process in a set of modules that perform an intent based trajectory generation by means of manipulating instances of the trajectory prediction formal languages. These modules are the Trajectory Language Processing Engines (TLPEs) and are responsible for transforming a phrase expressed in one language into another phrase expressed in the same or a lower level language. There are different types of generic TLPEs responsible for enrichment, translation, transcription and editing. Of all these types of TLPEs, the most important is the so-called Trajectory Computation Engine (TCE) because it is the basis of the entire architecture. The TCE is a translation TLPE that transforms an AIDL instance into a sampled trajectory, using an aircraft performance model and a meteorological model. Using these boundary data, this translation step is unambiguous, since for each valid AIDL instance there is only one computed trajectory. The computed trajectory can be obtained by integrating the differential equations of motion using the equations expressed by AIDL in the defined execution intervals.

There is a variant of the AIDL that can be used to described trajectories of quadrotor aircraft called quadrotor AIDL (QR-AIDL). The QR-AIDL comprises a number of instructions, corresponding to guidance modes, and the rules to combine

these instructions in a way such that the trajectory for quadrotor description is complete and unambiguous. A complete development of the QR-AIDL can be seen in (Frontera et al., 2019) and in (Frontera Sánchez, 2016). The QR-AIDL has four threads of execution to define the movement of a quadrotor, one thread is used for the yaw angle, two threads define the horizontal profile and the last thread defines the altitude of the quadrotor. As AIDL, the computed trajectory can be obtained by integrating the differential equations of motion using the threads expressed by QR-AIDL in the defined execution intervals. It may also be integrated in the ICDL/FIDL languages hierarchy.

2.4 Tracking and monitoring

Once the mission is created, it will be carried out at some time. In order to know if the real time execution is being done properly, there must be some means to detect the drone is accurately performing the intended mission and analyze its behavior.

2.4.1 Detection, localization and tracking techniques

There are many different technologies enabling drone detection, localization and tracking, including:

- **Non cooperative sensors**
 - Detection using radar techniques.
 - Detection through UAS radiofrequency signals.
 - Detection by acoustic signals
 - Detection through images
 - Detection by merging these techniques (data fusion)
- **Cooperative sensors**
 - Telemetry.
 - ADS-B.
 - Remote identification.

An explanation of these techniques is given below.

2.4.1.1 Radar techniques

The use of RF radar techniques is a possible approach for detecting/tracking drones during day and night. A big challenge with radar based detection of UAVs is that they have very small radar cross section (RCS) and they fly at lower altitudes and lower speeds compared to larger aircrafts (Fioranelli et al., 2015). In the literature (Güvenç et al., 2017) there are several types of radar used for detection/tracking/classification of drones, such as mmWave Radar, UWB Radar or NLOS Radar.

(Krátký and Fuxa, 2015) verifies the ability of radars to detect small, slow and low-flying targets. This verification of the capability has been done through the use of simulations, laboratory experiments and open-field experiments. This paper explains the existence of two important factors to be considered for the use of radars to detect airborne threats. These two factors are the target to be detected and the radar itself. When a radar is used to detect small and slow targets, the limiting factor is the radar section of the target (RCS - Radar Cross-Section). In this work, "mini UAVs" have been treated as a Medium of Airborne Attack (MoAA) and it has been concluded that radars working in the X-band are the ones that best detect "mini UAVs" due to their dimensions and radar sections. These radars offer optimal accuracy for measuring the coordinates of the targets being detected, and sufficiently small antenna dimensions. But, on the other hand, as antidrone radars must be mobile or offer a very easy deployment platform to be relocated and therefore short wavelengths must be used.

Another way of detecting UAVs using radars is by means of the Multi-static Forward Scattering Radar (Blyakhman et al., 2014). The most important principle of a forward scattering radar (FSR) for target detection is the use of the shadow field. When the diffraction angle, which is the angle between the direction of the transmitter-white and the direction of the target-receiver, is approximately zero, the shadow field can be observed at the receiving point. This shadow field is taken into account in narrow regions where the diffraction angle is approximately zero, causing the forward scatter radar section to increase considerably compared to monostatic radar sections, which occurs only when the size of the target is larger than the wavelength used. In the multi-static configuration of a radar forward scattering, a certain number of transmitting and receiving positions in the air and receiving positions on the ground must be used. The altitude of the targets must always be equal to or less than the altitude of the airborne transponder positions, which could be placed on board UAVs or any other type of aircraft. An example of a multi-static FSR is shown in Figure 2.1. It has been assumed that a multi-beam antenna is used at each receiving position, and these antennas in turn automatically angle-track each mobile transmitting position through electronic beamforming. On the other hand, the required beam direction is calculated by finding the direction of the signal from the direct transmitter. In addition, airborne transmitters move

around the managed region, so solid dynamic coverage is composed of the crossing of the FS mobile regions of each transmitter and receiver pair.

Figure 2.1: Geometry of the multi-static FSR

A radar system so small that it can be mounted aboard a small UAV, and capable of detecting and identifying other small aircraft using the Doppler footprint, is described in (Moses et al., 2011). This radar system is a lightweight X-band (10.5 GHz) radar. The operation of this radar is divided into two phases. In the first phase, the radar scans in continuous wave (CW) mode. Once the presence of the target has been detected, the second phase of operation is initiated, and the radar's mode of operation is changed from CW to FMCW. The objective of this mode of operation is to provide distance data to the target.

Figure 2.2: ART Midrange 3D from ART

There are many commercial antidrones solutions. The ART Midrange 3D, which can be observed in Figure 2.2, is a high-resolution C-UAS surveillance radar. This

high performance sensor is specifically designed to detect small unmanned aerial vehicles (C-UAS) and unmanned aircraft traffic management (UTM). The radar is composed of a 3D multibeam antenna system and an additional high power amplification stage, and is capable of detecting, tracking and classifying micro quadcopters and micro fixed-wing UAS with extended elevation coverage (ART, 2020). Another commercial solution is provided by Indra (Drones, 2019). The ARMS, which can be seen in Figure 2.3, is an anti-drone system designed to detect and neutralize any possible threat generated by this equipment. From micro drones to large UAS, the ARMS system improves the detection capacity of traditional systems.

Figure 2.3: ARMS antidrone from Indra

2.4.1.2 UAS radiofrequency signals detection

Another approach to detect/track UAVs is using signal that are broadcasted from the UAVs such as control signals between the UAV and the GCS (typically at 2.4 GHz) or payload signals (video communications) between the UAV and the GCS. Three possible approaches are considered for drone detection in (Nguyen et al., 2016). These approaches are the analysis of the reflected signal from propellers, sniffing the communication between the drone and its controller and the analysis of the reflection patterns from vibration of the drone's body. Other approach is the one explained in (Shin et al., 2015) where the frequency hopping spread spectrum signals from a UAV are extracted. It is possible to generalize it for training a classifier for identifying unique RF transmission patterns from UAVs. Finally, in (Haluza and Čechák, 2016) the RF techniques are used to detect UAVs.

A commercial solution is provided by Dedrone enterprise (Dedrone, 2020b). The RF-300, showed in Figure 2.4 is a passive, network-attached radio sensor using in the detection, classification, and geolocation of drones and their remote controls. The

Figure 2.4: RF-300 from Dedrone

RF-300 has an internal digital signal processing of radio signals used to identify the direction and in combination with other RF-300 determines the position of signals from drones and pilots.

2.4.1.3 Acoustics sensors

Another type of UAV detection is through the use of acoustic signals. An array of acoustic sensors can be used to detect and estimate the direction of arrival of sounds from sources such as UAVs. An example of implementation of this type of UAV detection is explained in (Benyamin and Goldman, 2014). This article shows how the location of a target is estimated and followed by triangulation with two or more microphone arrays, and how the UAV model can be estimated from the sound spectrum of the target. A small tetrahedral array of microphones has been used in this report.

Another example of UAV detection using acoustic signals is shown in (Pham and Srour, 2004). In this work the data collection equipment consists of two individual microphone arrays in 16-X and 4-L configurations where the microphones are placed on the ground and mounted on metal studs, while the elevated sensors are placed on tripods. These microphones are covered by six-inch thick foam screens to protect them and limit the effects of wind. Once the signals have been captured by the arrays, they are processed and analyzed. The data processing developed, as well as the analysis of the acoustic sensor arrays have been proven to be used to detect and track UAVs at low altitude and at close distances. This process operates best under benign daytime conditions and is approximately five times better at detecting

medium sized UAVs with petrol engines than small UAVs with electric motors.

A final example of acoustic signals being used for the detection of UAVs is presented in (Case et al., 2008). This article describes a low-cost microphone array designed to locate and track small UAVs. The array is composed of 24 microphones that are initially located in an unknown location and employs a delay-summing beam-forming algorithm. This array, as mentioned above, is low cost and mobile, so it has the ability to detect and track small UAVs. To do this, the microphones must be configured and calibrated to determine their exact positions. Once they have been calibrated, data collection begins and is stored as audio signals. These audio signals must be preconditioned for interference and ambient noise suppression. After all this, the delay and summing algorithm for beamforming begins. The results of the beamforming and extracted drone plots are consistent with the flight paths that have been made.

Quantum offers a solution for the detection, classification and tracking of unmanned aerial systems thanks to its device called Quantum’s vector series QA-100 (QTSI, 2020). The QA-100 uses seismic-acoustic technology to create a security perimeter around the site to be protected. When a drone is flying, whether it is an electric or gas engine drone, its engine emits energy into the air that generates sound waves in all directions. These acoustic waves are detected by the QA-100 sensor, which, in addition to detecting them, processes them and obtains the characteristic fingerprints to certify the presence of a drone in real time. The Quantum’s vector series QA-100 device can be seen in Figure 2.5.

Figure 2.5: Quantum’s vector series QA-100 from Quantum Technology Sciences

Another company that markets a solution for UAV detection is DroneShield (DroneShield, 2020). DroneShield uses real-time alerts and a collection of digital evidence to assist in security by identifying drones. The solution they propose consists of an acoustic detection technology that is composed of sensors that recognize

the sound properties of the most common UAVs. Once the sensors pick up the sound, it is compared in a database and if there is a match, the system alerts that a drone has been detected. The product they sell is shown in Figure 2.6.

Figure 2.6: DroneShield's omnidirectional sensor and long-range sensor

2.4.1.4 Computer vision

In (Dudani et al., 1977) an online recognition system for the identification of three-dimensional objects has been developed. This system uses a black and white television camera to provide a two-dimensional image on a digital computer. After obtaining the image on the computer, the next step is to remove the clutter from the image by means of a pre-process that provides a clean silhouette as well as its boundaries. Using these at the time of the calculations, certain characteristics are obtained that are used to identify the objects in three dimensions, the position they occupy and their orientation in space by means of a recognition algorithm.

Another way in which aircraft detection is carried out is the one developed in (Besada et al., 2001) and (Besada et al., 2005). This system consists of a set of tracking architectures composed of cameras that calculate the target trajectories of the aircraft projected on the camera. The system starts by taking the first image, which is used for initialization of the background estimation. Then a loop is initiated where the trajectories are predicted in the capture time for each new image taken by the cameras. All those pixels that differ from the background that has been previously estimated are detected and form one or more blobs related to the current targets. These blobs have to be extracted using trajectory predictions, edge detectors and motion detectors. The remaining pixels that are not detected are stored to enable detection of the next frame. With blobs and an association process, one or more blobs are associated to each target and, in addition, the blobs within the association are used to initialize the paths. Finally, each path is updated with its corresponding blobs and the not updated paths are deleted. Figure 2.7 shows the process and the results obtained for aircraft, but these processes have been extended to drones as can be seen in Figure 2.8 (Campañá Ramos, 2016).

Figure 2.7: Grayscale image of input, detector, blob extractor and tracker output

Figure 2.8: Image of drone detection

A commercial solution is proposed by HGH (HGH-Infrared, 2020). The Spynel V-LRF shown in Figure 2.9 is composed of a full HD (1920x1080) camera with

a continuous x30 optical zoom that allows zooming in on the threat. The drone is detected and tracked by the Spynel sensor thanks to the use of the panoramic detection system operated by the Cyclope software.

Figure 2.9: SpynelVision from HGH Infrared Systems

2.4.1.5 Sensor fusion

Detection using a set of these techniques outlined above is the ultimate way to detect UAVs. A low-cost, low-power methodology consisting of a fusion of technologies linking several sensors is presented in (Shi et al., 2011). This technology includes a simple radar, an acoustic array of microphones and optical cameras that are used to detect, track and discriminate potential airborne targets. The multimode sensor fusion algorithms employ the Kalman filter for target tracking and an acoustic and visual recognition algorithm is implemented to classify targets. The first component in the multimode sensor network is radar, which is responsible for detecting targets that are approaching the area of interest. The second mode of sensors is the acoustic microphone array whose main objectives are to provide target arrival direction, provide target identification and classification, and mitigate false alarms. The last sensor mode is the optical system composed of infrared detectors to improve the resolution of the target in the area of interest. Results show that this sensor fusion is suitable for detecting, tracking and discriminating small UAVs.

Data fusion should be done with great caution. The key problems to be solved can be referred to as data association, positional estimation and temporal synchronization. Data association is a general method of combining data from different sensors by correlating one sensor observation with the other observations. This process should be done with caution to ensure that all measurements that refer to the

same drone are associated so that no association between measurements from different drones occurs. One way to make this association is by spatial synchronization, i.e. seeing that a pair of measurements from different sensors have very similar position values. Sometimes, to see this behaviour it is necessary to make coordinates changes because each sensor can give the position captured in different coordinates. This process of changing coordinates must also be done very carefully because it can be a great source of error if it is done incorrectly. But before making any kind of association it is necessary to make a time synchronization so that all the measures are of the same instant of time. Another problem is the bias estimation and correction as the different sensor measures come with systematic errors. And the last problem faced by data fusion systems is filtering and prediction for which they usually use common techniques such as Kalman filtering and Bayesian methods.

A commercial solution is proposed by the company Dedrone (Dedrone, 2020a). Its UAV detector product is called DroneTracker and uses an array of different sensors that detect the drones in real time. These sensors are of two types, acoustic sensors and optical sensors. The acoustic sensors are microphones that pick up the typical acoustic characteristics that UAVs possess, while the optical sensors, in turn, are divided into two categories. The first type is used with sunlight and is a camera that is useful for analyzing live video. On the other hand, the second type of optical sensors is used at night because it consists of an infrared camera, and like the camera used during the day, it is used to detect drones using image analysis methods. The solution proposed by DroneTracker is shown in Figure 2.10.

Figure 2.10: Dedrone DroneTracker

2.4.1.6 Telemetry caption

Another source for drone positioning is telemetry. UAVs must have an on-board telemetry module capable of interconnecting the ground control station with the autopilot and thus enable the status and navigation signals of the aircraft to be monitored and controlled. Telemetry is typically sent at 435 MHz or 915 MHz operating frequencies. A specialized protocol called MAVLink is usually used for data

transfer. A typical telemetry module has a range of coverage in the order of kilometres (Marsh, 2009). Commercially there are telemetry radios used to provide a wireless connection between an unmanned aircraft and a ground control station. An example is the RFD900 Long-Range Telemetry, which can be seen in Figure 2.11, which provides reliable connectivity at greater than 5 km ranges with normal antennas (jDrones, 2020).

Figure 2.11: RFD900 Long-Range Telemetry from jDrones

Newer solutions have been explored to provide telemetry info through cellular networks (In-Situ, 2020). These systems provides a quickly connection to remote monitoring stations by using cellular network technology. This connection is to obtain real-time access to data, event notifications and system status updates.

2.4.1.7 ADS-B

ADS-B is a way to monitor the airspace in the air traffic control systems (McCallie et al., 2011). In ADS-B, the use of GNSS data allows the aircraft to determine its own position. Every aircraft broadcast this position in a short periodic position message. These position messages are detected and recorded by ground sensors and other nearby aircraft. Not only the position is sent, these messages also includes velocity, identification, aircraft intent, emergency alerts ... The ADS-B infrastructure consists on simple radio stations that are cheaper to install and maintain than the mechanical infrastructure associated with traditional ground-based radar stations. (Schäfer et al., 2014) shows the system architecture of ADS-B and it can be seen in Figure 2.12.

A commercial solution is the one proposed by jetvision (jetvision, 2020), its Air!Squitter is a low cost ADS-B MLAT FLARM receiver and it can be observed in Figure 2.13.

Figure 2.12: ADS-B system architecture and protocol hierarchy

Figure 2.13: Air!Squitter solution from jetvision

2.4.1.8 Remote identification

(Federal Aviation Administration, 2019) describes the Remote identification process. Remote identification or Remote ID is the ability of a UAV in flight to provide certain identification and location information that people on the ground and other airspace users can receive. It is based on a network of Remote ID UAS service Suppliers (Remote ID USS) that collects the real-time identification and location of UAS in flight. In order to use this technique, the drone must have a remote identification equipment on-board, or in the GCS acting as positioning relay, that be able to connect to the internet and transmit through that internet connection to a Remote ID USS.

2.5 Architecture of drone fleet management systems

Since there will be an increase in the fleets of both public and private stakeholders, in the near future it will be possible to share resources to, for example, support various urban services through a centralized fleet or to bring the drones of different public owners to optimize their use. With this objective in mind, it is getting closer to the implementation of the Drones-as-a-Service (DaaS) concept, a concept that allows an end user to define a mission to be accomplished and choose the optimal combination of resources to complete it.

2.5.1 Scheme of the architecture of a drone fleet management system

The main objective of an architecture of drone fleet management system is to efficiently manage fleets of drones from different owners and pilots and with different sensing and flying capabilities to fulfill different types of missions. Usually, the architectures of drone fleet management facilitate the management of the whole process of a mission, from the creation to the evaluation, meaning before, during and post flight. The mission definition is made in terms of location and time, also specifying the sensors needed, the flight mode to carry out the mission, which can be automatic or manual, the pilots and the platforms. This mission definition also looks for materials available at the time of executing the mission, i.e. drones, payloads and sensors that are unused at the time the mission is to be carried out and finally looks for the human resources needed to carry out the mission. Once all these processes are done, the best option to carry out the mission is identified and the flight plan is generated for the pilot and to incorporate it to the selected Ground Control System. These types of architectures are also designed to allow the reception of real time video in streaming, as well as the telemetry of the drones that are flying. These video and telemetry receptions are usually analyzed in order to present them in interfaces in real time, to monitor if the mission is being performed correctly or if problems are occurring. Finally, these systems can also have historical logs about the flights performed by the drone fleet, as well as the recordings of those flights so that the mission that has been performed can be analyzed.

2.5.2 Real cases

In literature there are many examples of architectures of drone fleet management systems. Mission planning for drone fleets, where autonomous lightweight UAVs coordinate themselves in real time has been studied in (Bevacqua et al., 2015). In

(Lacroix et al., 2016) it is analyzed the use of UAV fleets for atmospheric science, as well as the way that it is necessary to manage the trade-offs between payload and endurance. (Sanchez-Lopez et al., 2016) proposes a framework for autonomous multi-drone operations that are defined by waypoints. The operations are validated using simulations. The planner, when it is possible, tries to find collision-free paths and if it is not possible, stops the drone until the risk disappears. (Causa et al., 2018) describes a multi-UAV path planning strategy where the GNSS coverage is variable. It includes an algorithm for cooperative navigation at planning level that maximizes the efficiency in task assignment by distributing the targets among the UAVs and altering drone trajectories to facilitate other trajectories in challenging areas. This algorithm has been validated on a virtual 3D scenario, includes edge definition and cost evaluation, as well as target assignment through a customized insertion based technique and timing and polynomial path definition.

Detailed flight plans for multicopters where preventing collisions with obstacles and ensuring the preservation of no drone zones are the main functionalities are generated by the system proposed in (Bozhinoski et al., 2015). The avoidance of no drone zones is a basic functionality in Unmanned Traffic Management Systems (UTM) which are currently being implemented. The authors of (Furini et al., 2016) propose an optimal route planning which allow the coexistence of UAV and aircrafts in civil airspace thanks to model the cost of drone route taking into account the air traffic and the avoidance maneuvers to prevent conflicts.

In a fleet, as the drones have to consistently exchange flight information among them to maintain the formation of the fleet and follow the commands given from the GCS through a robust network, (Yoo et al., 2015) has proposed a multi-drone platform for applications that require cooperation of multiple drones called Net-Drone. This multi-drone platform is equipped with strong network functionalities that allow to have a control system for a fleet of drones. This platform is able to provide a network service over a wide area, which is not possible with a single drone. It can be used in many fields as delivery, agriculture, inspection ... An example of a fleet managing model for drones in the delivery field is proposed by (Cordova and Olivares, 2016). This model performs the delivery or pickup of supplies and materials in a production plant with alternation in drone operations to obtain an efficient use of batteries. This model considers the load capacity, the cycle time of the transport operation and the demand of units to be manufactured. Another example of use of a fleet of drones in the delivery field is proposed by (Liu et al., 2019). In this paper, a novel drone fleet deployment and planning problem with uncertain delivery demand is presented. In this problem, the delivery routes are fixed and couriers work in collaboration with drones to deliver surplus parcels with a relatively higher labor cost and the objective is to minimize the sum of the drone leasing and operating costs and the expected labor cost.

Some patents that have been made of systems or methods for the control or

management of drone fleets are (Maor, 2019) in the technical field of agriculture equipment, specifically harvesting and (Simon and Mays, 2019) in the technical field of delivery operations.

In order to support those different services, a stable and reliable standard-based communication scheme is needed in realizing efficient control and management of UAV/drone-based operations. One of these standards which present more benefits is the oneM2M global standard, which provides machine-to-machine (M2M) and the Internet of Things (IoT) service layer for globally interoperable M2M/IoT solutions. For instance, a oneM2M global standard platform based UAV/drone management system based on the Interworking Proxy Entity (IPE) for drones is presented in (Choi et al., 2017). This paper also shows the implementation and demonstration for the oneM2M system.

Defence in Depth (DiD) is an established strategy for designing systems with improved reliability, safety and security performance and is suitable for complex systems executing complex processes. (Papakonstantinou et al., 2019) proposes a methodology for early assessment of basic DiD characteristics of a fleet of autonomous and manned drones on a Naval Mine Counter Measures (NMCM) mission based on its dependency model and other DiD-related attributes. The DiD evaluated principles in that paper are Redundancy, Diversity, Functional Isolation and Physical Separation. A possible strategy of drone deployment for large scale area surveying is discussed in (Liang and Delahaye, 2019). This paper proposed a strategy where three key parts have been analyzed (coverage decomposition, shape of mission area and cooperative path planning and weather effect), including a fleet of drones, cooperative coverage path planning, communication and data processing. In addition, in this paper, the key operation constraints for large scale agriculture and forestry surveying have been analyzed.

Reliability analysis of a system is a complex problem but it is also important. A matrix of drone fleet reliability assessment attributes has been analyzed in (Fesenko et al., 2019). That matrix covers standby redundancy techniques, types of drones/drone fleets, functions, state level and heterogeneousness. From that matrix, a reliability model for the multi-fleet is developed and analyzed. In that paper, it is shown that reliability of the multi-fleet can be improved both by adding drones to the reserve drone fleet and using more reliable drones for the multi-fleet. In (Rusnak et al., 2019) the reliability analysis of a drone fleet is made through the structure function of the system because it allows that the system has homogeneous and heterogeneous components. This structure function is also useful in the use of another tools of reliability analysis such as importance measures or critical cut and path sets. This paper presents results that can be used to decide on number of drones in main fleet, to find the time when the maintenance will be scheduled or to find drones with best compromise between price and reliability

A multicriteria optimization framework to minimize operational risk for multiple

UAVs in partially unknown scenarios is proposed in (Rubio-Hervas et al., 2018). This article is focused in addressing the trade-off between the path-integral risk measure and classical path-efficiency. Other studies, such as (Chornenky, 2018), are focused on demonstrating the economic potential of multi-sided platforms for remote operations of drones over the Internet. In other words, it studies how a drone sharing platform is able to provide technological solutions such as remote operation over the Internet and advanced autonomy. In addition, this work also assesses the restrictions on remote operation and the regulatory requirements for creating spaces for innovation and competitive advantage.

2.5.3 Integration with ATM and UTM systems

Air Traffic Management (ATM) is defined by the International Civil Aviation Organization (ICAO) as *the dynamic, integrated management of air traffic and airspace-safety, economically and efficiently through the provision of facilities and seamless services in collaboration with all parties* (Organization, 2005). An ATM system is made up of three parts. The first one is Air Traffic Flow Management (ATFM) which takes place before the flight begins as it is responsible for optimising the flow of traffic in accordance with the capacity of the third part of the ATM system (ATC) and maintaining the safety and efficiency of flights. The second part is the Aeronautical Information Service (AIS) which is in charge on transmitting all the necessary information to airspace users. This information includes safety, navigation, technical and even legal information. The third and final part is Air Traffic Control (ATC) which is the service responsible for maintaining separation of aircraft during flight and at airports when landing and taking off.

This last part (ATC) is the most relevant of an ATM system because is a process which is responsible for preventing collisions and expediting and maintaining an orderly flow of air traffic (ICAO, 2007). Some functions performed by the ATC service are to provide information to aircraft such as start time to take off, the aerodrome information, the runway to be used ... Other function is to separate the aircraft thereby having no problems with wake turbulences and jet blast hazards.

ATFM and ATC are the parts of the ATM system that need to cooperate with the UTM system.

In 1920, Croydon Airport, London, was the first airport in the world to introduce air traffic control. In the United States, air traffic control developed three divisions. The first of the Air Mail Radio Stations (AMRS) was created in 1922 after World War I when the U.S. Post Office began using techniques developed by the military to direct and track the movements of reconnaissance aircraft. Over time, AMRS was transformed into flight service stations. Today's flight service stations do not send control instructions, but provide pilots with many other flight-related information services. They are responsible for transmitting ATC control instructions in areas

where flight service is the only facility with radio or telephone coverage. In October 1929, the position reporting service was established for aircraft flying on so-called federal routes.

The airport's first traffic control tower, responsible for regulating arrivals, departures and surface movement of aircraft at a specific airport, was inaugurated in Cleveland in 1930. The approach/departure control facilities were created after the adoption of radar in the 1950s to monitor and control airport airspace. In addition, the first air route traffic control center, which directed the movement of aircraft between origin and destination, opened in Newark, NJ in 1935, followed in 1936 by Chicago and Cleveland. This centre made possible to provide pilots with information on the proximity of other aircraft in the environment of the airfield during weather conditions that required it. In April 1970, as part of the ATM, the "Central Flow Control Facility" unit was created to restrict the number of aircraft in the airspace. Its aim was to achieve a reduction in congested air traffic areas.

Nowadays, ATM systems are very complex and consist of a large number of different functions. These functions are provided one to many using a central entity as the control centre, while the services are deployed en masse as a monolithic system. Among the functions that ATM systems perform today are the acceptance and approval or rejection of flight plans, aircraft tracking, guidance and separation ratio of aircraft to pilots and the handling of emergency situations, among many others. This approach works well with the needs of aircraft, which are well defined, but with the new needs that arise from the emergence of drones, ATM systems are a little outdated. For this reason, the first systems that combine ATM systems with UTM systems are emerging.

These new air traffic management systems will perform a series of functions very similar to those performed so far by ATM systems, the difference lies in the way these functions must be delivered due to the large increase in air traffic density and the different types of aircraft, which each have a different performance model and technology. For example, commercial flights are scheduled by ATM systems a long time in advance and follow fixed schedules. In contrast, today an air taxi mission can be ordered at short notice before take-off, so planning must be done quickly to provide service. And the problem is that most of these missions are required in urban spaces with high densities of air traffic and with many obstacles (EUROCONTROL, 2004). Therefore, a new air traffic control and management system is needed that combines today's problems with the problems already solved before with the ATM systems.

To solve these new problems that have arisen with the appearance of drones, Unmanned Traffic Management systems have begun to be created, which consist of microservice-oriented architectures where services are built and provided by multiple users. A microservice is a piece of software created to perform a single function. Microservices have well-defined interfaces and requirements that allow them to be

added, removed or updated quickly, leaving consumers with the choice of those that best meet their requirements. Just as users can choose the microservices they need most, they can also choose the provider, as multiple providers give the same microservice. But this does not mean that all the functionalities of such a system are provided by multiple users. Governments must have a directory of services to ensure that only those with certain certifications can operate.

The microservices approach is extremely powerful because of its flexibility and there are different architectures that can be used in a UTM system (Krishnan, 2013):

- **Centralized architecture.** A single entity provides for all vehicles in an airspace.
- **Hub and spoke architecture.** Multiple entities exist in an airspace, but each vehicle gets services from single entity.
- **Federated architecture.** Multiple providers exist for most services, each aircraft can choose between entities.
- **Peer to peer or closest peer architecture.** No entity provides services, aircraft communicate locally with nearest neighbors.
- **Distributed architecture.** No entity provides services, vehicle communicate globally and directly, relative to flight plan.

A complete air traffic management system requires real-time two-way communications on board all aircraft, and these must be effective and secure. Therefore, high bandwidth and low latency are required to respond quickly and safely to threats. All these requirements are forcing operators to equip their aircraft with the best available technology and providing incentives for innovation to improve. In addition to the traffic management services that maintain the separation of drones, the detect and avoid function (DAA) is a backup. Various simulations have been carried out showing that DAA works well in low-density regions, while strategic and tactical management works better at higher densities (Altiscope, 2020). Therefore, the best solution is a hybrid between management and DAA.

So, once the two air traffic management systems (Aircraft Traffic Management and Unmanned Traffic Management) have been created, voices are emerging that want to bring the two systems together to have a single ATM-UTM system. Some system administrators are trying to provide a single, authoritative system to coordinate traffic services. This system is implemented and operated under the direction of government regulatory agencies.

Digital traffic management services are responsible for managing aircraft flights over a wider airspace and are coordinated with each other to ensure safety at all

stages, to ensure that flight plans are completed, aircraft manoeuvres are coordinated and emergency response is carried out quickly. These services are complemented by aircraft control services, which provide compensation to unmanned aircraft taking off, landing or crossing specific airspace corridors. All other services support traffic management and aircraft control services, such as weather information, surveillance, registration . . . Each of these services must inform the others of its decisions to help each other make the best possible decision.

Different architectures of an ATM-UTM system are shown in Figure 2.14 and in Figure 2.15.

Figure 2.14: Architecture 1 of an ATM-UTM system

A typical use case of an ATM-UTM system is coordination during an emergency response. The rapid creation of new flight plans in emergencies is becoming more complex due to the increase in the number of actors in the airspace. Even today when an emergency aircraft receives an urgent call, it communicates directly with a control facility, and the ATC gives it priority clearance to fly without regard to the other actors flying through the airspace. In the near future, thanks to ATM-UTM systems, the aircraft's request will have to be sent through the entire network so that it can safely cross existing flight corridors or routes without conflict. The system will also be robust enough to cope with a communications failure, while remaining secure and allowing emergency operations to continue.

Figure 2.15: Architecture 2 of an ATM-UTM system

2.6 Conclusions

An overview of a drone fleet management was made in this chapter. It has started with a review of the control and handling of a drone, as well as of the different existing solutions. The chapter has continued with a comprehensive review of literature. Of these systems, architectures used today have been detailed as cases of uses and functionalities they perform, as well as real examples. Finally, an extensive revision of the fleet management systems has been carried out.

The future UTM will rely heavily on the sharing of information among different actors, and the exploitation of such information by automatic processes. In this sharing of information, the first actors to appear have been drone fleet operators. And operators, in turn, face the problem of controlling and managing the drone fleet. But many operational scenarios would be much better addressed if automated means for coordinated resource allocation and simultaneous control of drones were available. This PhD Thesis explores the use of different methods and systems for better control and management of a drone fleet, and the applications

that these systems could provide to UTM. In addition, this PhD Thesis describes a microservice-oriented architecture that provides organizational and optimization functionalities to orchestrate the operation of drones.

Most of the applications discussed in this state of the art are based on the definition of ad hoc procedures for the design of missions. These procedures tend to disregard flight safety limitations as they assume a secure environment. This PhD Thesis describes the requirements of a flight planning system for safe urban operations, which can be used to design operations taking into account flight restrictions and limitations.

The PhD Thesis will describe the life cycle of a mission using different systems or services during all phases of a flight, from the creation of the mission in the pre-flight phase to the study of the recolected data in the post-flight phase. Contributions in the different steps of the life cycle will be performing resulting in enhanced efficiency and safety in a UTM-enabled scenario.

Chapter 3

Architecture proposal

Properly managing a fleet of drones is a problem that many companies have nowadays. Even managing and controlling a drone is complicated because many aspects need to be considered (weather, terrain, flight zone, etc.). Things become even more complicated when drones traffic must be considered. This aspect is important to ensure the safety of the actors involved.

A system for control and management of a fleet of drones can have different type of requirements. In some cases, the time to prepare a mission is a limiting factor, so the system must allow the quick design and analysis of a flight plan. In other cases, it is the guarantee of the use of those closest assets that are adapted to the requirements of the mission. In others, a central system could operate on a global fleet scale and be composed of small systems responsible for different geographical areas and each of them would have to communicate with the central system to ensure control and management of their zone without inferring in other areas. As it can be seen, there are many ways to approach the realization of a drone fleet control and management system, but in all of them the main stakeholders must be taken into account.

The main roles of the actors involved in any control and management drone fleet system are the following:

1. Mission definer. It is the entity in charge of defining a mission. It could be a user willing to provide a given service as packet delivery or surveillance by using his/her own drone fleet or available resources from different drone operators. Also, it could be a drone operator managing a fleet of drones.
2. System Manager. Person who overviews the system behavior: drone fleet, assignments, ongoing missions ... It could be a different person to the one who creates the mission because this person oversees all the performed mission at the same time.

3. Operator. Physical entity that provides drone services in the system because it owns a fleet of drones.
4. Pilots. These are in charge of setting up and flying drones, both in manual or automated modes. Each pilot may have a set of drones owned or be available to fly any drone authorized in his/her license.
5. Payload Manager. It is the person in charge of setting up and controlling the payload. The drones and payloads needed to fulfill each mission differ, thus it is necessary to manage these resources to check their availability.
6. Logistic personnel. People needed to transport drones and payloads to specific locations to make service feasible.

Several roles can be implemented by a single person in certain cases.

3.1 Terminology

Some concepts must be defined to understand the whole architecture and it can be seen in Figure 3.1 to a better understanding:

- Mission. Set of geometric elements, take-off point and landing point defined by the user to carry out a task such as inspecting a building or going from one point to another in the city.
- Flight plan. Specification of the waypoints that the mission becomes with its height and time restrictions and fulfilling the rest of the restrictions so that the mission is accepted.
- Instruction. Each of the drone actions (take off, land, go to a waypoint, change heading, change yaw, hover ...) in which the flight plan can be decomposed.
- Trajectory. A set of time-ordered aircraft states. Each aircraft state contains the values of the state vector of the aircraft, sampled at a particular time. Among the state variables, the geometric (position and attitude) and kinematic (speeds) information is obtained from modelling mathematically the different instructions of the flight plan.

Figure 3.1: Flight plan definition schema

3.2 Architecture schema

On the basis of the above, an architecture designed with the purpose of making a step forward towards the implementation of the Drones-as-a-Service (DaaS) concept is presented. In the DaaS concept, a final user defines a mission to fulfill and chooses the optimal combination to complete it. The objective of the architecture is to efficiently manage fleets of drone from different pilots and/or operators, with different sensing and flying capabilities. The proposed architecture is thought to ease the management of the workflow composed by the pre-flight and the in-flight phases.

The architecture presented in Figure 3.2 is a microservices-oriented architecture that enables the management of drone missions from different operators and pilots. That means that the architecture may be used to control a fleet of drones from a single system and enables full-service delivery, from mission definition, resource allocation, to flight plan generation and real-time monitoring. The microservices of which the architecture is composed can be divided into pre-flight services and in-flight services that interact with each other. The inputs are the mission specification (take-off and landing areas, some high level details on the mission targets and some constraints), telemetry from drones and other surveillance information. The outputs are flight plans, trajectories, flight instructions, alarms and tracks.

As mentioned above, the system is composed of two set of microservices that operate in different stages of the complete flow of service operation, although they interact with each other: one of them needs information from the other to carry out its operation. The first set of microservices is devoted to the Pre-flight as can be observed in Figure 3.3. It is composed of three microservices, the Mission

Figure 3.2: General architecture schema

Planning Service, the Flight Planning Service and the Trajectory Prediction Service. The input to these services is the mission specification and the outputs are the flight plan, flight instructions and the trajectory. The second set is devoted to In-flight services, that can be observed in Figure 3.4, and it is composed of the Tracking and Monitoring Service, whose inputs are telemetry from drones and other surveillance information and whose outputs are alarms and tracks. This service needs information from the Pre-flight microservices, as explained in Chapter 6 of this thesis.

Figure 3.3: Pre-flight services architecture

Figure 3.4: In-flight service architecture

The functional requirements of the complete drone fleet flight planning and monitoring system will be described next:

- The interface must allow to define the mission specifications in the easiest way possible and it must be user-friendly. This way a user will be able to create missions quickly.

- It is possible that in some cases resources should be shared among different pilots/operators. That is one of the main features of Drone-as-a-service systems.
- The arrival of priority missions should lead to the cancellation of non-priority in-flight missions and the denial of non-priority missions affected by this new priority mission and the search for alternatives to carry out these missions.
- The use of airspace must be orchestrated to maintain safety, both for airspace users and for other possible involved actors as pedestrians or drivers.
- Potential changes in the flight context or unpredicted conflicting/emergency situations must be monitored to allow the system to perform appropriately.
- During the flights, the system must be able to receive, process and store the information from all sensors.
- This information may be shown in real-time to different users and on different devices. These must be associated to the user permissions.
- All gathered data must be stored and deleted according to the local privacy law.

Apart from the functional requirements of the systems, the non-functional requirements accomplished by the system are:

- Performance. The system could be used for emergencies, so the system must be responsive and fast and the least amount of resources possible must be used.
- Supportability. Modify, upgrade or add some of the services or components must be easy.
- Replication. The system must be reusable and re-instantiated for different end-clients, if needed.
- Usability. The system must be able to provide a user-friendly interface to facilitate the creation of missions to users, as well as provide real-time information to know the status of missions.
- Reliability. All databases involved must be stored in periodical backups made by the system. The system must ensure stability to operate at all times, as well as provide reliability against service disruptions.
- Availability. A very high rate of availability must be guaranteed by the system.

From the functional and non-functional requirements and the general architecture, the proposed microservice-oriented architecture is shown in Figure 3.5. The system has four main microservices which communicate with the other components to provide all the procedures and features. These four main microservices are the Mission Planning Service described in Chapter 5, the Flight Planning Service detailed in Chapter 4, the Trajectory Prediction Service and the Tracking and Monitoring Service explained in Chapter 6. All the architectures of these microservices are discussed individually in section 3.4.

The Mission Planning Service creates a trajectory from the user’s mission specification using the weather forecast, the aircraft performance model and flight patterns using the Trajectory Prediction Service. The Flight Planning Service generates a flight plan respecting all the conditions combined with the ground shape, the airspace and the no drone zone models. Ground and Weather services are context services: they gather the information from external services and perform some initial calculations on the data. The Trajectory Prediction Service is in charge of transforming the flight plan into a predicted trajectory through a flight plan to instruction conversion where the time restrictions to incorporate to the flight plan in the Flight Planning Service are calculated. Finally, the Tracking and Monitoring Service maintains the state and the updated position of all the drones that the system must handle and ensures real-time airspace security.

3.3 Use cases

Nowadays in civilian drone operations, most of the systems are specialized on performing VLOS operations. But the future applications of drone technologies to urban areas will demand implementation of safe BVLOS flights. Additionally, the 5G networks that are going to be deployed in the near future, will allow for safe remote ubiquitous command and control and information retrieval from drones in urban environments. Consequently, a extremely low latency, high reliability and high bandwidth communications environment is assumed for the description of the following use cases. In addition, the drone’s autonomy manifested in the presence of on-board intelligence will become less significant thanks to the 5G networks. In any case, it is assumed that coordinated on-board efforts and ground systems, which include human pilots for contingency control, allow two types of basic operations for pilots.

1. **Automated flight following 3D waypoints.** This type of flight is designed in simple deliveries or basic inspections in which there are no dangerous elements. In this type of operations, the flight must be precisely planned following an open-loop control approach based on the flight plan conversion and trajectory prediction tools using the drone’s dynamic models.

Figure 3.5: Proposed microservice-oriented architecture schema

2. **Manual/autonomous flight.** This other type of operation depends on the information provided by the drone sensors in real time that cause the human

pilot or autopilot takes decisions in real time on the flight plan implemented. The operations of this type depend on the information provided by the drone that causes changes in the flight, for example in emergency missions, in inspections of complicated infrastructures, ... In addition, this type of operation can be used to model the drone landing in an unknown environment in which the skills of a human pilot will achieve a better and more careful landing. This type of flight follows a closed-loop control approach, and the flight plan is not as precise because only imposes restrictions as it defines the areas or volumes in which the pilot performs the target operation.

Based on these types of operations, the flight planning use cases will be defined, describing the interactions between the different functions and the potential outcomes. A use case is a list of actions or event steps typically defining the interactions between a role and a system to achieve a goal. The use cases will be divided according to the stage in which they come into action, so there will be pre-flight use cases and in-flight use cases.

3.3.1 Pre-Flight Use Cases

The pre-flight use cases are detailed below:

1. **"Public flight plan" waypoint based flight planning.** The end user has to design a mission that goes from the takeoff area to the landing area, with different actions during the journey. In addition, the end user has no privacy or security problems with publishing the details of his/her flight plan in the service providers. This does not mean that the details of the flight plan will be public for the rest of the operators. In this case, the end user will make use of the Flight Planning Service, which will provide him/her through a HMI based on a map, all the information related to the context of the mission:
 - No Drone Zones.
 - Airspace areas with flight limitations or manual authorization request.
 - Airspace airways structure. The end user can select the waypoints following those airways or let the service to calculate these waypoints.
 - Weather prediction as wind, storms or other hazardous events that difficult drone operations and are necessary to calculate the time restrictions.
 - GNSS availability and communication coverage predictions necessary to the proper performance of the flight.

In this case, the mission is designed iteratively using the HMI of the Mission Planning Service since the waypoints or areas are selected on a map, or a previously saved mission is selected. In both cases, the Flight Planning Service completes the flight plan based on drone dynamics and flight context information such as airways, terrain models, weather forecasts, etc. In addition, the end user must define the minimum and maximum initial time for take-off. Also the system has to check that the flight plan is going to be completed by the drone in a time less than the maximum duration allowed for that flight with that drone.

2. **Manual/autonomous area operation flight planning.** In this use case, the mission is assumed to require a manual/autonomous flight operation, where the end user will not be able to define waypoints before the flight. Takeoff and landing are within the manual/autonomous flight zone. Instead, the flight plan is defined by a 3D volume during a certain time interval. This volume becomes a closed zone from which the drone cannot exit, which is known as geocage. The design of this flight plan is done easily since the service does not have to use the dynamics of the drone. As in the previous case, the end user has to have access to the context information about the airspace airways structures, no drone zones, weather information, ..., to be able to design the mission. To carry out the definition of this type of mission, the end user directly employs the HMI of the Mission Planning Service and sets the volume and time constraints.
3. **Hybrid operations flight planning.** It may be the case that the end user wants to make different individual inspections on a single flight or different deliveries at different landing points through a single flight. In both cases, the operations to be done are hybrid as they link manual operations with operations following a list of waypoints. For example, the takeoff to the inspection area is done following the waypoints, once in the area the pilot takes control and performs the inspection, then the drone follows the waypoints again and goes to the next inspection, where the pilot retakes control of the drone and finally takes it to the landing point. This use case is a mixture of the two previous use cases. The end user defines each of the zones in the Mission Planning Service HMI and to calculate the flight plan, context information and time restrictions calculation from Flight Planning Service are necessary.
4. **"Blind flight plan" waypoint based/hybrid flight planning.** In this case, the end user needs to design a mission from a take-off area to a landing area in which there may be a hybrid mix of operations, but in any case, the end user needs to have privacy and security with the publication of the flight plan

details to service providers. To carry out this process, two different phases have to be performed:

- Description of a hybrid operation or waypoints operation as detailed above that will be used internally by the end user.
- Definition of an associated "Bounding-Box" volume or collection of volumes and their respective time intervals that completely cover the previous operation. The resulting flight plan will be sent to the UTM service provider.

This use case has obvious advantages for the end user, but has a major disadvantage. This disadvantage is that having to make a rougher specification of the mission requires additional airspace resources reducing airspace capacity for subsequent flights.

3.3.2 In-Flight Use Cases

During the in-flight phase, the flight plans made during the pre-flight phase are accessed by the Tracking and Monitoring Service. These flight plans are used in conjunction with real time positions and speeds, provided by drone telemetry or by cooperative or non-cooperative sensors that detect the drone, to detect potentially hazardous events such as:

- Lack of conformance with the current flight plan being followed by the drone. This lack of conformance can occur in the spatial dimension, both vertical and horizontal, in the event that the drone leaves the 3D volume or moves away from the segments that join the waypoints. Also the lack of conformance can occur in the temporal dimension if the drone is not respecting the temporal constraints imposed in the flight plan. The drone may have conformance problems in one, two or three dimensions at a time.
- Tactical conflicts or security separation losses between drones in the near future. These tactical conflicts are evaluated using maneuvers that are expected to be implemented following the flight plan.
- Interactions with dynamic geofences, with hazardous changing weather, etc.

3.4 Brief explanation

The proposed architecture shown in Figure 3.5 enables full-service delivery, from mission definition to flight plan generation and trajectory prediction and submission to operation, or tracking and real-time monitoring.

3.4.1 Pre-flight architecture

In the Pre-flight, the mission definer (this could be the end user or the drone operator) describes the mission. This description must be done in terms of location and schedule, sensing needs, fly modes (manual or automated), pilots and payloads. Once the flight plan is planned, the pilot must enter it into a Ground Control System before flying. These actions are made by the Pre-flight services composed by Mission Planning Service, Flight Planning Service and Trajectory Prediction Service. These three microservices make a collaborative work, although the results obtained are different in one service and another, so a more detailed analysis of their respective architectures is necessary.

During the design stage, a human-machine interface (HMI) will be used. This HMI must be agile and easy to understand but it is not covered by this thesis. A mission specification is composed of a take-off geo-reference, an operation specification, different for different applications, and a landing area. Once the mission specification is created by a client, it is sent from the HMI to the Mission Planning Service in charge of parsing the mission specification and creating a mission object that the whole system understands. The mission specification can arrive to the Mission Design functionality from the HMI or from a repository of created missions as it can be seen in Figure 3.6. Different Mission Design services are needed for different applications. All the outputs of these different translation services are converted into the waypoints that form the mission in the so-called Waypoints Calculator functionality. Then, the Flight Planning Service is called, to complete the flight plan from the original flight plan specification, i.e. a waypoint list. Then a flight plan from the Flight Planning Service arrives to the functionality that distributes the flight plan to every function that needs it. So, this functionality sends the complete flight plan with the time constraints to the flight plan translator functionality where the flight plan, written in an internal format, is translated into the flight script format that the drone that is going to be used can understand. This translation or flight instructions are sent to the pilot's Ground Control System, so he/she can performance the mission. On the other hand, this translated flight plan arrives at the HMI where the user that defines the mission can observe the result flight plan from his/her mission specification and can accept it, modify it or reject it. And it is also stored in the mission repository, as it is going to be used in the Tracking and Monitoring Service.

A waypoint list (Initial Flight Plan) created in the Mission Planning Service arrives at the Flight Planning Service as it is showed in Figure 3.7. This service is composed by a set of functionalities that converts that waypoint list to a flight plan taking into account different conditions to plan the UAVs flight from the take-off point to the end of the mission at the landing point. This list of waypoints is sent to the terrain avoidance service to calculate the heights of these waypoints to ensure that the drone is not going to crash with the ground or some obstacles. The

Figure 3.6: Mission Planning Service Architecture

terrain elevation is obtained thanks to a call made to the Ground Service, which receives a list of waypoints and calculates the terrain elevation in the paths between the waypoints, avoiding obstacles. The next step is to calculate if there is a conflict between the list of waypoints with their heights and the no drone zone model database, which specifies the zones where the UAVs flight is forbidden. This functionality makes the appropriate modifications to avoid these zones. Next, the airspace restrictions integrator functionality receives from the no drone zone avoidance the list of waypoints with their heights and it calculates the flight plan that fulfill the airspace restrictions, modifying it when it is necessary using the airspace model database. Finally, the flight plan without time restrictions reaches the functionality in charge of integrating them. To do so, this functionality communicates with the Trajectory Prediction Service to obtain the time restrictions of each of the points that make up the flight plan. Once these restrictions are obtained, the Time Constraint Integrator functionality adds them to the flight plan. From this functionality, a complete flight plan is obtained and it is sent back to the Mission Planning Service. This complete flight plan incorporates the waypoints of the flight with their heights and time restrictions and complies with airspace restrictions and no drone zones.

Figure 3.7: Flight Planning Service Architecture

The Time Constraints Integrator functionality of the Flight Planning Service is communicated with the Trajectory Prediction Service, as can be seen in Figure 3.8. This service receives a flight plan with the waypoints that form it, with their respective heights and complying with the aerospace restrictions and no drone zones, but without time restrictions. These are calculated in the Flight Plan to Instruction functionality, in charge to convert the flight plan into each of the drone instructions. Then, those instructions are converted to a trajectory using the weather model from the Weather Service and the aircraft performance model of the UAV that is going to be flown. Once the predicted trajectory is obtained, it is sent to the HMI and the time constraints are sent to the Flight Planning Service. These two functionalities are explained with more detailed in Sections 3.4.1.1 and 3.4.1.2.

3.4.1.1 Flight Plan to Instruction calculation

When the Flight Planning Service performs its calculations and generates the flight plan without time restrictions, it sends the incomplete flight plan to the Trajectory Prediction Service, specifically to the Flight Plan to Instruction functionality. This section will explain the transformation from flight plan to instructions with the consequent calculation of time restrictions to include in the flight plan and thus complete it.

The first step in this functionality is to convert the waypoints that make up the flight plan to a list of straight segments.

Figure 3.8: Trajectory Prediction Service Architecture

Then, the flight plan must be decomposed into simpler elements capable of simulating the flight along segments joining consecutive waypoints. These simpler elements are called instructions in the original idea of the QR-AIDL (Frontera Sánchez, 2016). Thus, the instructions are the elements of the AIDL which represent the algebraic constraints that encompass the mathematical DoFs of the system of equations which represents the movement of a drone. Then, the instructions are defined through physical laws allowing to model the drone movement until an end condition, such as arriving to a given position or reaching a defined speed. These physical laws are converted into mathematical expressions constraining the trajectory kinematic parameters. There are many different types of parameters that can be used to model trajectory constraints, but the ones, expressed with respect to the earth-fixed frame, used in a system like this are:

- Horizontal groundspeed $(v_x, v_y) = (\frac{dx_e}{dt}, \frac{dy_e}{dt})$
- Vertical speed $v_z = \frac{dz_e}{dt}$
- Yaw ψ

Then, the instructions are a set of expressions for the parameters (v_x, v_y, v_z, ψ) that allow any kind of maneuver. Therefore, a maneuver is unambiguously defined by performing a sequence of instructions, from which the 3D position (x_e, y_e, z_e) of

the drone is obtained at any time and the rotation angles (Euler angles) (ϕ, θ, ψ) around the Cartesian axes.

For example, if the movement of the drone to be simulated has the instructions a constant groundspeed, a constant vertical speed and fixed yaw, the mathematical expressions to be used are showed in Equation 3.1, where k_i represent real constants constraining the speed and yaw along this part of the flight. To end these instructions, a possible end condition could be $x_e = 0$ which means when the drone reaches the 0 in X position, the instructions will finish.

$$v_x(t) = k_1; v_y(t) = k_2; v_z(t) = k_3; \psi(t) = k_4 \quad (3.1)$$

If the beginning of a horizontal acceleration maneuver is required to be modelled, the mathematical expressions to be used are seen in Equation 3.2 and are defined as a constant horizontal acceleration (k_a and k_b , from instant t_0), a zero vertical speed and a fixed yaw k_4 . Two possible end conditions for this instruction could be $v_x(t) = v_{max}$ or $v_y(t) = v_{max}$ which cause the restrictions to take effect until one of the two speeds takes the maximum value.

$$v_x(t) = k_a(t - t_0); v_y(t) = k_b(t - t_0); v_z(t) = 0; \psi(t) = k_4 \quad (3.2)$$

These examples demonstrate that by properly concatenating consecutive instructions, the different guidance patterns used by the drones to implement the segments can be modeled. For example, a straight horizontal segment with stopped start and end states, and with a target yaw, could be implemented through a series of accelerate - keep speed - decelerate - turn instructions, where the acceleration instruction must define the heading of the movement that will be kept for the whole segment. Meanwhile, if the drone performs a vertical ascent maneuver without changing the yaw, the segment that performs this movement is implemented with the consecutive instructions of vertical accelerate - keep speed - vertical decelerate. All instruction sets that implement any segment must fulfill three requirements:

1. The initial and end conditions must be respected. They have to start and finish into the waypoint positions defined in the Flight Plan.
2. All the drone physic parameters have to be continuous along the trajectory.
3. All dynamic constraints should be taken into account. Specifically, the maximum rotation speed of the motors must be respected. This requirement could impose changes on some of the instruction parameters.

When these instructions representing the movements of each of the segments (accelerate, keep speed, turn, decelerate ...) are obtained, the time it takes to perform the movement is calculated. In this way, the service knows how long it

takes the specific drone to accelerate to a certain speed, or how long it takes to decelerate to stop, or how long it takes to make a 180° turn on itself.

3.4.1.2 Trajectory calculation

The sequence of instructions calculated in Section 3.4.1.1 is sent to Trajectory Integrator functionality in order to obtain the sampled trajectory. Two additional information must be considered to compute an accurate trajectory: (1) the drone physical features such as its weight, aerodynamical information, motor thrust ... which are part of the Aircraft Performance Model and (2) the effect of the wind which is obtained from the weather model. A system of differential equations that models the drone motion dynamics is solved in the Trajectory Integrator functionality and the relevant parameters that model the trajectory are obtained too from these data.

After the instructions are obtained, these are transformed into a predicted set of positions in a high fidelity trajectory prediction engine in Trajectory Integrator functionality. To do this, a QR-AIDL underlying model is used (Frontera Sánchez, 2016). In this model the motion of a quadrotor is that of a point mass where the sum of all propellers forces and its orientation (Euler angles) are the control inputs. This carries to a simplified model with four mechanical degrees of freedom (closed by the four constraints in the instruction for each time). So, the model can be written as:

$$\frac{d^2 x_e}{dt^2} = \frac{(\cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi)U_1 - k_s(v_x - w_x)}{m} \quad (3.3)$$

$$\frac{d^2 y_e}{dt^2} = \frac{(\cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi)U_1 - k_s(v_y - w_y)}{m} \quad (3.4)$$

$$\frac{d^2 z_e}{dt^2} = \frac{(\cos \phi \cos \psi)U_1 - k_u(v_z - w_z)}{m} - g \quad (3.5)$$

$$U_1 = b \sum_{i=1}^4 \Omega_i^2 \quad (3.6)$$

where (v_x, v_y, v_z) is the quadrotor's speed with respect to the same earth-fixed frame, (ϕ, θ, ψ) are the rotation angles (Euler angles) around the Cartesian axes, (w_x, w_y, w_z) are the Cartesian local wind speed respect to the same earth-fixed frame, m is its mass, g is the acceleration due to gravity, (k_s, k_u) are the horizontal and vertical friction coefficients, U_1 is the sum of the forces of all propellers, Ω_i is the rotation speed of i-th propeller and b is the thrust coefficient.

Thanks to these equations 3.3, 3.4, 3.5 and 3.6, each of the physical laws explained in Section 3.4.1.1 are integrated obtaining a trajectory prediction for the

Flight Plan. In (Frontera Sánchez, 2016) a full study of the effect of the constraints is made. The most common QR-AIDL instructions used to constraint the problem of defining flight, and the ones used in the service are:

- **Horizontal speed law.** In these instructions, horizontal accelerations, horizontal decelerations and constant horizontal velocity are implemented.
- **Vertical speed law.** In these instructions, vertical accelerations, vertical decelerations and constant vertical speed are implemented.
- **3D speed law.** Oblique accelerations, oblique decelerations and constant oblique speed are implemented in these instructions.
- **Yaw law.** In these instructions the yaw turns are implemented.

Then, the times of each of the instructions in which a segment is divided are added up to obtain the total time it takes the drone to perform that segment, that is, how long it takes the drone to go from one waypoint to the next. Once all the times to perform each of the segments are obtained for different maximum and minimum velocities, they are sent to the Flight Planning Service in the form of time constraints.

3.4.2 In-flight architecture

The Tracking and Monitoring Service, that can be observed in Figure 3.9, tracks drones using their telemetry and other surveillance sensors information.

Different sensors localization information is received by the tracker in charge of filtering and predicting the position of the drones. The tracks are used to complete the monitoring service functionality.

The predicted tracks are sent to a deviation detector functionality, where the compliance of the flight plan by the drone is verified. If the drone is deviated too much from the established flight plan, an alarm will be generated. On the other hand, if the drone does not follow a flight plan, this deviation detector functionality is not executed.

The next functionality to be executed is the break-in detector functionality. In this functionality, all drones are checked if they are going to a no drone zone or are inside these zones. If the drone is inside a no drone zone, an alarm is generated. In the case of the drone is approaching to a no drone zone, the distance to that no drone zone is calculated and an approaching alert may be raised.

Finally, the predicted tracks goes through an obstacles detector that indicates if the drone is in danger of colliding with the ground or some other drone. To calculate if the drone is going to collide with the ground, a call to the Ground Service is made.

Figure 3.9: Tracking and Monitoring Service Architecture

And to calculate the possible collisions with other drones, the tracks of those drones are used. If a collision may be possible, an alarm is also generated.

The predicted and monitored tracks and the alarms are sent to the HMI and the GCS.

3.5 Conclusions

In this chapter the architecture of control and management system for drone fleet has been defined. Starting from the requirements that a system like this must have, the actors that participate in such a system and the objectives that have to be achieved, system architectures have been analyzed, studied and developed. In order to explain the architecture that has been developed, different use cases, both pre-flight and in-flight, have been explained.

The proposed architecture of control and management system for drone fleet is based on microservices to be used during the life of a mission, from its creation by a user to its implementation and monitoring to ensure its proper execution.

The following chapters will delve into exemplary implementations of each of the services that make up the system.

Chapter 4

Flight planning service

In this chapter an example of the Flight Planning Service will be described in detail. The preparation phase of a flight with a drone begins when a user fulfills a mission. The flight that the drone will make can be automatic or be manually flown by a pilot. The objective of the service is to convert the generated mission, explained in section 3.1, into a list of waypoints that the drone must follow, called flight plan, complying with the requirements of height, time, avoidance of No Drone Zones and airspace model.

This chapter is organized as follows: Section 4.1 presents the high-level requirements and the interactions of the service. A schema of its architecture and context is presented in section 4.2. Section 4.3 is in charge of presenting and analyzing the calculations that allow the transformation of the mission into a flight plan, and its results are presented in section 4.4. Finally, in section 4.5 the conclusions of the service are presented.

4.1 High level requirements and service interactions

The high level functional requirements of the Flight Planning Service are:

- This service must give support to the flight planning of missions defined as a combination of sequences of 3D waypoints and areas (3D volumes) of manual flight, potentially with user-defined time and altitude constraints.
- This service must support different modes of flight and drones: fixed wing, multirotors, VTOL, etc.
- The service must support georeferenced height definition, enabled by the use of digital terrain elevation data.

- The service must ensure compliance with height and time restrictions imposed by the user.
- The service must guarantee compliance with the height and time restrictions calculated for the fulfillment of the mission established by the user.
- The service must guarantee to avoid the terrain so that the drone in its flight does not hit the ground or any obstacle.
- The service must ensure that No Drone Zones are avoided so that the drone does not enter into restricted or forbidden airspace zones.
- The service must ensure compliance with airspace restrictions in order to comply with current unmanned aircraft flight regulations.

In addition to the functional requirements of the service, also the non-functional requirements to be accomplished by the Flight Planning Service have to be defined, and these are:

- **Performance.** The service must be fast and must use the least amount of resources possible: the service can work with a collection of missions involving different users and has to be able to respond to everyone in the shortest possible time, especially in case any of the missions is an emergency.
- **Supportability.** Modify, upgrade or add some of the services or components must be easy.
- **Replication.** The system must be reusable and re-instantiated for different end-clients as different end users can work on different missions at the same time.
- **Usability.** The system must be able to provide a user-friendly interface to facilitate the creation of missions to users and provide a flight plan understandable by any end user.
- **Reliability.** All databases involved must be stored in periodical backups made by the system. The system must ensure stability to proportionate flight plans if there is solution, as well as provide reliability against service disruptions.
- **Availability.** A very high rate of availability must be guaranteed by the system.

After presenting the functional and non-functional requirements of the Flight Planning Service, then the life cycle of a flight plan must be presented. The different use cases relevant to the Flight Planning Service, that were explained in Section

3.3, describe summarising the interaction with the Mission Planning Service. In Figure 4.1, the life cycle of a flight plan is depicted. The last part of the life cycle of a flight plan is performed in the Trajectory Prediction Service. The calculations to transform the flight plan into instructions and calculate time constraints were explained in Chapter 3. Finally, the different phases of the life cycle are explained:

1. **Restrict the heights to the waypoints list.** From the list of waypoints, the ground service is called to obtain the minimum and maximum heights at which the drone can fly in the area in which the mission occurs. When the heights are recovered, compatible height constraints are added to the waypoints.
2. **Restrict No Drone Zones.** Taking as a starting element the list of waypoints with the height restrictions, the nearby No Drone Zones are to be analyzed. It is checked if the list of waypoints crosses any of these zones and in that case an alternative path to avoid it is sought. Once the alternative is found, the necessary waypoints are added to the list with the height restrictions to avoid this No Drone Zone.
3. **Airspace restriction.** The areas of restricted airspace that are close to where the mission is to be flown are obtained from the airspace model. The service checks if the list of waypoints interferes with any of these restricted airspace zones. In the affirmative case, the way in which it is possible to fly in that

Figure 4.1: Life Cycle of a flight plan

zone of airspace is found and added to the list of waypoints where it proceeds with the height restrictions for new waypoints.

4. **Time restrictions.** From the Flight Planning Service a flight plan without time restrictions is got and it is sent to the Trajectory Prediction Service. The time restrictions are calculated here since conversions from waypoints to instructions are necessary to obtain these times and they are sent back to the Flight Planning Service, where they are added to the flight plan.

4.2 Architecture schema and context

From the life cycle of a flight plan, the inputs and outputs of the Flight Planning Service can be obtained. In this way, the general architecture of the service, with its inputs and outputs, can be seen in Figure 4.2.

Figure 4.2: General schema of the architecture of the Flight Planning Service

At a very high level, the Flight Planning Service can be described as an architecture conformed by a collection of layers as is depicted in Figure 4.3. The lower layer is related to avoid the terrain and physical objects described in a ground model and add height restrictions imposed by users. Over it, there is a layer related to make flight plan compatible with No Drone Zones. Then, a function takes care of making flight plans compatible with the aeronautical constraints in the airspace, the aerospace restrictions. The last upper layer is related to incorporate time restrictions to the flight plan. Finally, all those functions are managed through a HMI according to the use cases that were described in Section 3.3.

4.3 Flight plan calculation

Next, each one of the service functionalities will be explained in detail. In the first subsection, Subsection 4.3.1, the input and output requirements of the service will be presented. The following subsections explain different functionalities: the

Figure 4.3: Flight Plan Definition Support Architecture

calculation of the height constraints in Subsection 4.3.2, Subsection 4.3.3 presents the terrain avoidance calculation, Subsection 4.3.4 explains the avoidance of the no drone zones, Subsection 4.3.5 continues with the integration of the airspace restrictions and the section ends with the integration of the time restrictions in Subsection 4.3.6.

4.3.1 Input and output requirements

The system offers the possibility of having different types of missions, as explained in Chapter 5. But, regardless of the type of mission, the Mission Planning Service turns these missions into a list of waypoints along with the height restrictions imposed by the user on some or all of the waypoints.

From the list of waypoints that enters the service, the service is responsible for converting it into a flight plan composed of the following fields:

- **Operation type.** To identify the type of operation which is going to be executed. This could be VLOS or BVLOS.
- **Endurance.** To know the maximum time allowed to undertake the mission.
- **Identifiers.** The operation, pilot and drone identifiers are include to know who is carrying out the mission and with what drone.

- **Take off area.** The area from which the drone takes off to perform the mission. It has the height restrictions and the minimum and maximum time allowed to take off. As in other missions, this area can be the location of the drone from which it has to reach the takeoff zone imposed by the end user.
- **Landing area.** The area to which the drone lands to end the mission. It contains the height restrictions.
- **Waypoint list.** It contains all the waypoints through which the drone will have to pass, as well as the height restrictions that each of the waypoints presents. It also contains the time restrictions that each waypoint has. If the mission specification was made by the service's format, each waypoints could include attitude restrictions and waiting periods.

4.3.2 Height restrictions calculation

As mentioned above, the mission specification, which will be detailed in Chapter 5, is composed mainly of take-off and landing points and operation specification, whether simple or complex. But in the Mission Planning Service HMI, the end user can specify some mission features that restrict flight behavior. For example, the user can specify a minimum, maximum or both height restrictions for a certain operation specification. So, these constraints will in general take the form of height intervals. Then, it is clear that is necessary to calculate height constraints for the rest of the flight if the flight planning monitoring is implemented. These height constraints must be compatible with potential drone dynamics.

Once all the height restrictions that the end user has added are known, they must be incorporated to each of the waypoints and to those points without restriction, a height restriction is necessary to be established. The goal is to keep the flight as low as possible but always respecting a minimum height above ground level in each of the positions of the waypoints, thus avoiding the waste of time and energy. Therefore, a height window is established that is compatible with typical technical errors in piloting and flying. The heights added are heights above ground level (AGL) as the ground service has not yet been called and the height of the ground is not known.

4.3.3 Terrain avoidance calculation

After having added the height restrictions, it is possible to avoid the terrain in the waypoints. But this avoidance of the terrain in the waypoints does not guarantee that the terrain will be avoided during the trajectory that joins two consecutive waypoints. These segments can be very long and different obstacles can appear. This method of avoiding the terrain is explained here.

The waypoints are collected, converted to UTM coordinates and passed on to Ground Service to obtain the DSM (Digital Surface Model) heights and DTM (Digital Terrain Surface) heights of the drone path. This Ground Service will be described in detail in Section 4.3.3.1. Once the Ground Service answered with the different heights every few meters (typical underlying map resolution), these must be parsed and stored in the case where subsequently necessary. The heights that the Ground Service provides are heights above the sea level (WGS84 heights).

Once the heights of the terrain are known, the way to avoid them along the route is to incorporate intermediate 3D waypoints to ensure collision avoidance. The horizontal position of these waypoints needs to be aligned with the expected trajectory, so the new intermediate waypoints will be added to the straight lines joining two consecutive waypoints. In this system, the condition that has been established to avoid the terrain is that every time there are abrupt ups or downs, a new waypoint must be added. An abrupt ascent or descent is the one greater than the safety margin established between the minimum and maximum restriction heights. This process is repeated until the landing waypoint is reached. The way to add the new intermediate waypoints to the list of waypoints is by inserting them in their original positions, that is to say, the new ones are added between the old ones, while the old ones are still maintained.

This calculation of intermediate points must also comply with the regulations. And there is one that determines that the airspace for drones is between 20 and 150 meters, the so called Very Low Level airspace. For this reason, all heights must be checked, once the intermediate waypoints have been added, meet the condition of being below 150 meters AGL. If this is not the case, the maximum height of this waypoint should be restricted to 150 meters. When the requirements for this functionality have been met for the heights above sea level (WGS84 heights), the heights above ground level (AGL heights) are calculated from the WGS84 heights and then stored. These two heights are added to the waypoints.

4.3.3.1 Ground Service

This service has been created to provide a highly accurate ground height profile so that the ground avoidance functionality can fulfill its purpose. Therefore, the Ground Service is responsible for accessing the underlying terrain models and must provide terrain height information to the rest of the system.

The Ground Service incorporates two different models from which to obtain the height. On the one hand, the service presents the Digital Terrain Model (DTM) which is responsible for returning the heights of the terrain. The Digital Terrain Model can be described as a 3D representation of the terrain surface in which each position pair (latitude, longitude) has stored above Mean Sea Level altitude. On the other hand, the service has the Digital Surface Model (DSM) which provides the

heights of the surface, counting the obstacles and physical barriers. The heights of trees, buildings and other elevated obstacles above the earth's surface are represented by LiDAR or photogrammetry reconstruction in the Digital Surface Model (Früh and Zakhor, 2003).

Figure 4.4: Universal Transversal Mercator grid of Madrid

In the system, the Ground Service has different sources from which to get the height. Mainly the system is centered in Spain so the DSM height is filled with LiDAR data published by the Spanish IGN (IGN, 2019) after going through various data processing processes to eliminate sources of error, fill in white spaces. The resolution that the LiDAR data provides is 5 meters and after processing, GeoTiff images of $1km \times 1km$ size are generated, indexed using an Universal Transversal Mercator grid and stored in a quick access database of images. An example of the mesh images the system has stored can be seen in Figure 4.4, while a specific GeoTiff image can be seen in Figure 4.5. As it is necessary that the Ground Service provides heights all over the world and not just in Spain, for the rest of the world there is another source of information, which is TanDEM-X 90m Digital Elevation Model (Rizzoli et al., 2017). Instead of doing like the previous process, the service performs external queries to obtain heights from the rest of the world with an accuracy of 90 or 250 meters depending on the area. These received heights are treated and processed by the service to finally return meaningful DSM heights.

The heights used are orthometric thanks to a geoid model with sufficient precision to convert ellipsoidal altitude increments into orthometric altitude increments, since the original altimetry is ellipsoidal.

The requests that are attended by the Ground Service and that come from the rest of the system are:

- Requests for the height of a specific point or a set of points if they are expressed as (longitude, latitude)

Figure 4.5: GeoTiff image from an urban zone of Madrid

- Requests of the profile of heights in a segment between two consecutive points. An example is the one that performs the terrain avoidance functionality.

4.3.4 No drone zones avoidance calculation

Before explaining the calculation process, it is necessary to explain the No Drone Zones model

4.3.4.1 No Drone Zones Model

A No Drone Zone is an area where a drone can not fly because it is unsafe or illegal. Some examples of these zones can be airports, monuments, prisons, ... In this system, as in UTM systems, these No Drone Zones are modelled using geofences. A geofence is defined as *a virtual geographic boundary that enables software to trigger a response when a device enters or leaves a particular area* (Stevenson, 2015). In UTM systems when a drone invaded a geofence, an alarm is generated and alerts the pilot and the UTM authorities but often also the Ground Control Systems and the drone autopilots preclude that invasion.

The way these geofences are incorporated into the system is through a Polygon geometry with specific height and time constraints. These height restrictions, consisting of a minimum height and a maximum height, determine the height range at

which a drone cannot fly, although it can go below or above if allowed by regulations. The time restrictions composed of a geofence start time and a geofence end time indicate the period of time that the geofence is active, times at which no drone can fly in the area delimited by the volume originated with the polygon positions and the height margin. If a geofence has to occupy the entire airspace, the height restrictions range from 0 to 150 (maximum height at which a drone can be flown by regulation). In addition, if the geofence is to be infinite in time, the time from 1970 to the maximum date is set. A collection of geofences from Spain obtained in Spanish AIP (ENAI, 2019) can be seen in Figure 4.6.

Figure 4.6: Example of geofences in the VLL airspace from Spain

Other parameters that a geofence has are the list of drones, operators and pilots with permission to fly inside the geofence. As mentioned above, geofences need to be visible in the HMI to the end user when creating a mission so that no geofence is invaded by design.

4.3.4.2 Calculation

The process to avoid these No Drone Zones is depicted in Figure 4.7. The first step is to retrieve the No Drone Zones in the area where the mission is going to be performed. To do this, the No Drone Zones model is accessed with the waypoints that make up the flight plan, and this model will return those NDZ that are close to that mission. If the model does not return any zone, the functionality finishes its process and passes to the next one. On the contrary, if it returns any zone, it is necessary to check if effectively the flight plan breaks into that zone. From the list of waypoints, the segments that join them are checked for potential intersection

Figure 4.7: No Drone Zones avoidance calculation flowchart

with the NDZ. An intersection occurs if the segment intersects with the NDZ border or it is inside the NDZ. If no segment intersects with a No Drone Zone, the mission does not break into any of these zones and the list of waypoints passes to the next functionality. On the other hand, if any of the segments intersects with any NDZ, there is a violation of the No Drone Zone and the functionality is in charge of modifying the flight plan following any of the following strategies, all of them designed to modify the flight plan as little as possible.

These strategies are carried out in a certain order and when one of them fulfills the purpose of avoiding the No Drone Zone, the rest are not calculated because the problem has already been solved. The strategies that the No Drone Zones avoid functionality are:

- **Height Avoidance.** The No Drone Zones don't need to occupy the entire height of a volume of airspace, but may have a restricted height margin. In such cases, this strategy is useful since the drone can fly below or above the No Drone Zone within regulatory limits and without breaking into the NDZ. In the case where this strategy can be followed, the functionality changes the height of the waypoints involved. The first attempt is to lower the height since

a rule of the system is that the drone flies as low as possible. If it cannot be lowered, the height of the involved waypoints is increased.

- **Time Avoidance.** In case the previous strategy could not be carried out, the next strategy is to change the start date of the mission. This strategy of date change makes sense since some No Drone Zone are only temporary and have a start date and an end date as they can be created to restrict airspace by various events such as demonstrations, sporting events, cultural events . . . The process is to see the activation period of the No Drone Zone, if it is infinite or if it is less than the start date minus one year or if it is greater than the start date plus one year, then this strategy is not done. A year has been set as the time limit for change because it has been considered to be the maximum time a user can wait to perform a mission. If the activation period of the NDZ is shorter than the established limits, the algorithm tries to move the start of the mission to the earliest possible time. In this way, the strategy changes the mission start time to a time greater than the current time but keeping a space of time equal to the mission duration. This ensures that the mission is completed before the NDZ activation period begins. If it is not possible to move the start of the mission before the NDZ starts, the strategy places the start of the mission as soon as possible after the end of the NDZ. This avoids the NDZ.
- **Route Avoidance.** The last strategy is to change the route. This strategy is done in the worst case, when the other two strategies have not worked. This strategy is the last one because if a user has created a specific mission he/she will want to fly that mission and not a similar one, but quite often it is better to fly a similar mission than not to fly directly. This strategy changes the part of the flight plan that breaks into the No Drone Zone to surround it and continue with the mission. To do this, it is analyzed which segments of the flight plan are those that intersect with the NDZ and are analyzed one by one. If it is an input segment, the initial waypoint of the segment is taken and the shortest way to the first waypoint outside the No Drone Zone, which has already been analyzed, is made. All waypoints in the flight plan list between these two waypoints are deleted and the necessary waypoints are added to avoid the No Drone Zone. If the segment is a figure (polygon or inspection), the part remaining inside the No Drone Zone is removed and the external part remains. If, by chance, it is the takeoff or landing that is inside the NDZ, the mission has to start or end at the first or last waypoint that is not inside the NDZ. If the entire flight plan occurs within a No Drone Zone no modification is possible and this mission cannot be flown. This is the last strategy that the functionality has to avoid the No Drone Zone. If it doesn't work, the NDZ

can't be avoided and the mission can't be accomplished.

If the user restricts some part of the flight because he/she needs it to be that specific way, and that area is the one that causes conflict, the strategies shown here cannot be carried out. Consequently, this functionality does not always get a solution to avoid No Drone Zones and not all missions can be performed. Therefore, in order to avoid this happening, the best first prevention is to avoid making flight plans that interact with No Drone Zones in the HMI.

4.3.5 Integration of airspace restrictions

The objective of this functionality is that the flight plan meets all the requirements that the airspace imposes to ensure the proper performance of the mission. To do this, the functionality must check that all restrictions are met by the mission created and for those that are not met, try to make them met by modifying the flight plan as little as possible. In the case that any of the restrictions is not met even after modification, the mission is rejected because it does not meet the airspace requirements. The functionality created presents two different processes:

- **Flight rules integration.** It may be the case that the mission crosses some airspace volume that has some flight restriction and this must be complied with. Some flight rules concern the allowed flight height, the weight a drone can carry in that volume, the speed at which it can be flown, the hours at which the flight is allowed ... The regulation depends on the country, but what is referred to here is the Spanish regulation (BOE, 2017). All these restrictions/rules must be checked one by one to see if the flight plan being formed complies with them. What it does is to go one by one and see if it has to modify something in the case that it is necessary, these modifications could be the height of a waypoint, or the speed or the takeoff time ...
- **Airspace route integration.** In the airspace there may be airspace volumes in which different airways have been defined so that the flight occurs in an orderly and structured manner. It is in this case that the functionality must access the airspace model that will be detailed in Section 4.3.5.1 in order to know how to proceed when one of these volumes is invaded. What the functionality does in these cases is to go segment by segment of the flight plan and see if it crosses an airspace area, which has already been requested the airspace volumes in the area. In the affirmative case that this intersection occurs, the volume itself provides the possible paths out of the section. Then, the functionality calculates which is the closest entry to the volume and which is the closest exit and once known it obtains the path that the flight plan must follow within the airspace zone. The remaining waypoints within the airspace

zone are removed and new ones are added with their pre-established heights according to airways definitions.

Once the airspace restrictions have been integrated, the flight plan has been completed, with the exception of time restrictions. For this reason, this flight plan is sent to the next functionality to complete it with the time restrictions.

4.3.5.1 Airspace model

As it has just seen, airspace integration uses the airspace model that it was created. This airspace model contains:

- **Airspace volumes.** It is defined as a polygon so the coordinates of its vertices are specified.
- **Airspace volume properties.** Each airspace volume has a field called properties, where the specific rules and constraints are defined. These constraints/rules could be, among others:
 - Maximum/Minimum heights above ground level.
 - Maximum weight allowed.
 - Maximum/Minimum groundspeeds.
 - Minimum/Maximum times allowed.
- **Airspace volume route.** If the airspace volume has a network of 3D airways, these are added in this field, as follows:
 - Entry points to the airspace volume. The volume has a few entry points available to access it, the drone cannot enter from anywhere.
 - Exit points of the airspace volume. The volume has a few exit points available to leave it, the drone cannot leave the volume anywhere.
 - The ordered collection of 3D nodes that connect each entry point with each exit point. These nodes are defined by their latitude, longitude and geometric height above sea level. The calculation for the arrangement of these nodes is explained in Section 4.3.5.2.

4.3.5.2 Optimal path calculation

The areas where there is an airspace model is usually because the presence of obstacles is quite high inside and it is better to create a series of airways through which the drones can fly calmly without having the danger of hitting any of these obstacles

or to reduce traffic interactions complexity. In the model of the airspace that has been designed, a series of entry points and exit points are established, just like the airways inside the area. The airspace model calculates the optimal path for each of the entries with each of the exits.

As the height is established at each of the nodes, the problem of solving each of the paths is summarized in solving a problem of the 2D shortest path (2SP). Moreover, it is easy to prove that the 2SP between two points (entry and exit) is a path whose inner vertices are vertices of the nodes (De Berg et al., 1997). A 2SP can be found by building a graph. In graph theory, the problem of the shortest path is the problem of finding a path between two vertices (or nodes) in such a way that the sum of the weights of the edges that constitute it is minimal. This graph is a data structure comprising n vertices (corresponding to the nodes) and m edges connecting every pair of vertices that it may be possible. There are different algorithms to resolve this problem:

- **Dijkstra algorithm.** It solves the problem of the shortest paths from a single origin vertex to all other vertices of the graph. It is a non-directed graph with a temporal complexity of $O(m + n \log n)$ (Barbehenn, 1998).
- **Bellman-Ford algorithm.** It solves the problem of shortest paths from one source if the weighting of the edges is negative. It is a non negative weighted directed graph with a temporal complexity of $O(nm)$ (Goldberg and Radzik, 1993).
- **A* Search algorithm.** It solves the problem of the shortest paths between a couple of vertices using heuristics to try to speed up the search. It has a temporal complexity of $O(n)$ (Zeng and Church, 2009).
- **Floyd-Warshall algorithm.** It solves the problem of the shortest paths among all vertices. It is a non-directed graph with a temporal complexity of $O(n^3)$ (Hougardy, 2010).
- **Viterbi algorithm.** It solves the problem of the shortest stochastic path with an additional probabilistic weight at each vertex. It has a temporal complexity of $O(nm)$ (Forney, 1973).

From all these possible algorithms, the algorithm selected to pre-calculate the shortest path is the Floyd-Warshall algorithm. The reason for this choice is that this algorithm provides a solution for every entry point - exit point pair that the airspace area has. As these airspace areas have many entry and exit points each of them, it is necessary that it is done quickly and the calculation has the minimum computational cost, so this choice is the most adequate.

4.3.6 Time restrictions integrator

A list of waypoints with their height restrictions and complying with aerospace restrictions arrives to this functionality. Now it is time to add the time restrictions to complete the flight plan. To carry out this function, it is necessary to send this list of waypoints to the Trajectory Prediction Service so that it returns the temporary restrictions to be added.

The minimum and maximum takeoff times are taken from the flight plan and the times restrictions are added up to complete the time restrictions of the flight plan. To obtain a time window in which to perform the flight plan, the service makes, on the one hand, a quick flight plan, so, it calculates the minimum time that would take the drone to perform the mission going at the maximum speed allowed by the drone and meeting all constraints. On the other hand, the service makes a slow flight plan, that is, it calculates the maximum time that a drone would take to perform the mission if it were at the slowest speed allowed by the drone. Both flight plans would comply with the restrictions imposed by the user, that is, if a segment has to be made at a groundspeed determined by the user, that segment in both flight plans is made at that speed. In other words, the height or path restrictions imposed are not altered in this time constraints calculation.

Once all the times are calculated, the time constraints are sent back to the Flight Planning Service, in particular to Time Constraints Integrator, where these time restrictions are added to the flight plan, which is finally completed. This flight plan is sent to the Mission Planning Service.

4.4 Results of the flight planning service

In this section a full example of application of the previously described techniques and some performance results to identify the computational bottleneck are provided. But first of all, the practical implementation of the service is described.

The flight planning tool is built as a microservice architecture making use of REST APIs for service-oriented communications and connecting contextual data recovered from outer and internal Web services. The front-end is built in HTML5 / CSS / JavaScript, and the backend logic are built either in JavaScript (Node.js) or C++ for computationally complex parts of the system. The contextual data is obtained from the following services to which the Flight Planning Service connects to:

- From Google Maps, the mapping information is retrieved and is used as the basis for the map-based interface.

- The Digital Terrain Model and Digital Surface Model are retrieved from the Spanish National Geographic Institute (IGN, 2019).
- Geofencing is automatically created on the basis of a digital version of Spanish AIP and other public databases as hospitals, national parks, nuclear facilities, ...
- Airspace Model is created automatically on the basis of a digital version of Spanish AIP and the ones that a user creates.

A delivery mission will be done in which the results of all the functionalities will be shown in Section 4.4.1.

4.4.1 Flight Plan design example

In this section an example of a flight plan design exercise using the Flight Planning Service is going to be showed. The first step is to define the type of mission that is going to be executed, in this case, a delivery mission, so the mission specification will have a GeoJSON format. Next, the path must be defined by the user. The mission 2D specification can be seen in Figure 4.8. As it can be seen, the mission specification is composed of a take-off area, that is marked with a 0 in the map and a landing area marked with a 10 in the map. The specification also has three phases:

- A linear operation that includes the take off position and waypoints among 0 to 6.
- A volume between waypoints 6 and 7.
- Another linear operation from 7 to 9 waypoints, and the landing point (10).

In Figure 4.8 it can be seen that the mission specification goes through four different special areas. The three areas represented in red, are no drone zones, whose details can be seen in Figure 4.9 and Figure 4.10. The NDZ areas will be active during the execution of the mission. Each one has different characteristics, so the Flight Planning Service will propose different alternatives for each one of them. The NDZ in Figure 4.9 has height and time restrictions that will cause the drone will not be able to break into that area. The NDZ in Figure 4.10a has a height restriction that allow to the drone fly over it and the NDZ in Figure 4.10b is an emergency and it has an active period of 2 days, so the drone will fly the proposed mission when the period of activity of the NDZ ends. The last is an airspace volume, represented with a blue contour, where a dense route network has been defined. In this area, the network follows the same structure as the underlying streets, so the drone will fly over the streets.

Figure 4.8: Original input flight plan

On the other hand, the parameters of the drone selected to perform the mission are summarized in Table 4.1.

Parameter	Value
Drone	DJI Phantom 4
Take Off min time	06/07/2020 11:00
Take Off max time	06/07/2020 13:00
Min height above ground level (m) in volume	20
Max height above ground level (m) in volume	40
Min flight time (s) in area	100
Max flight time (s) in area	300

Table 4.1: Example parameters

Before the mission specification and parameters arrive at the Flight Planning Service and this resolves and calculates the flight plan, a series of aspects can be concluded just by looking at what has been exposed so far:

1. The first No Drone Zone will delay the take off time of the mission.
2. The second No Drone Zone must be avoided because it is active, therefore the drone can not enter.
3. The third No Drone Zone will be over fly it.

Figure 4.9: No Drone Zone information

4. In the airspace model, the route structure must follow the restrictions imposed.
5. The terrain must be respected with special care because it is an urban zone and the abrupt changes of heights are constant.

So finally the mission specification comes to the service and goes through all the functionalities. The result obtained at the output can be seen in Figure 4.11. As it can be seen, the Service introduces new waypoints at elevated altitude to pass over the third No Drone Zone, changes the trajectory adding new waypoints to avoid the

(a) Detail of the No Drone Zone with height restriction

(b) Detail of the No Drone Zone with time restriction

Figure 4.10: Other No Drone Zone information

Figure 4.11: Resulting flight plan

second No Drone Zone, delays the take off time to avoid the first No Drone Zone, introduces additional waypoints to avoid terrain and another additional waypoints

following the constraints imposed by the airspace model. The associated height profile can be seen at Figure 4.12.

Figure 4.12: Height profile of resulting flight plan

On the other hand, the time restrictions are calculated in the Trajectory Prediction Service. The user, when generating the mission, sets a time window for takeoff, that is, a time window for the beginning of the mission. This window must be maintained, unless it influences some geofence that is invaded during the flight, so the window will move along the time but always keeping the same duration. Then, from the minimum time of the time window, the fastest path is calculated, that is, the one that would take the drone the least amount of time to complete. The slowest path is calculated using the maximum time of the time window. In this way, the minimum and maximum time that the drone can take to make each part of the mission at all times are obtained. These times are taken to a graph and represented so that the end user can see the duration of the flight, as can be seen in Figure 4.13 which is the result of the exercise with which this section has been started.

Figure 4.13: Time profile of resulting flight plan

As it can be seen in the figure, the take off time window has been maintained, but delaying the start time to avoid the first NDZ. There is a minimum take-off time and a maximum take-off time. There is also a minimum landing time and a maximum landing time. Also different paces for each of the trajectories are observed. The blue

one has less pace than the red one, so it is faster. Finally, the grey part represents the time that the drone can be in the manual zones.

With this example it has been demonstrated that this service is able of planning missions taking into account the requirements imposed by the user as well as the limitations, requirements and circumstances of the scenario. The example performed has a certain level of complexity which leads to the conclusion that the system is capable of planning and orchestrating operations of different types of drones and complexities in any airspace, whether urban or rural. In addition, it is verified that the system operates and provides solutions according to the current regulation. Finally, it is verified that the system prioritizes the requirements imposed by the user, since its creation is the most convenient for him/her, and in case of not being able to give solution to his/her requirements, the system provides a solution that less disrupts the desires of the user.

To sum up, the Flight Planning Service is able to provide adequate solutions following the architecture described in this chapter.

4.4.2 Performance evaluation

The performance of the Flight Planning Service is calculated in this section. For that, the time spent in each of the functionalities/layers for flight plan calculation is measured using a dual CPU Intel ®Xeon ®E5-2620 2.00 GHz server with SSD storage. To test this performance evaluation, three flight plans of different lengths (250 m, 2000 m and 6500 m) passing throught a No Drone Zone and an airspace area have been used. The time results obtained with these tests can be seen in Table 4.2.

Functionality	Short FP	Average FP	Long FP
Terrain avoidance	3.038 s	17.63 s	27.46 s
No Drone Zones avoidance	0.655 s	1.13 s	1.69 s
Incorporation of Airspace constraints	0.65 s	0.68 s	0.7 s
Time constraints calculation	0.653 s	1.13 s	1.69 s
Total	4.996 s	20.57 s	31.54 s

Table 4.2: Example parameters

From this temporal evaluation of the performance of the Flight Planning Service it can be concluded that the bottleneck is in the terrain avoidance functionality. Within this functionality is the call to the Ground Service, which being a REST request is the most time consuming part of this functionality, so the bottleneck is actually in the Ground Service. This would lead to the service only being used for simple missions because if they are complex, the execution time can be very high to

be able to function correctly in real time. The solution to be able to use the service in real time would be the parallel implementation of the Ground Service that would lead to an important reduction of this time.

4.5 Conclusions

In this chapter the Flight Planning Service has been defined. Starting from the requirements that a service like this must have and the system interactions, the architecture has been analyzed, studied and developed.

The Flight Planning Service generates the flight plan following all the specifications that have been marked to the service at the beginning of the approach. It does so correctly and with a not too high execution load, that can be improved in future versions.

Chapter 5

Mission planning service

In this chapter the Mission Planning Service will be presented. The high-level requirements and the interactions of the service are presented in Section 5.1. Section 5.2 exposes a schema of the architecture and its context. The input data models are explained in section 5.3. Section 5.4 is in charge of presenting and analyzing the calculations that allow the transformation of the mission specification into a list of waypoints to be used for the flight plan translation. Once these calculations have been analysed, the results are presented in section 5.5. Finally, in section 5.6 some conclusions are presented.

5.1 High level requirements and service interactions

As it was explained in Section 3.1, a mission is a set of geometric elements, take-off and landing points defined by the user to carry out a task such as inspecting a building or going from one point to another in the city. As the tasks are different, there are different types of missions to carry out different actions. Instead, a flight plan is a specification of the waypoints that the mission becomes with its height and time restrictions and fulfilling the rest of the restrictions so that the mission is accepted. The flight plan must be unique, there must be only one type of flight plan for the different types of missions. Therefore, it is necessary that there are different converters that transform each type of mission into a single flight plan. These different conversions are performed in different services, but all result in a common flight plan. In this chapter, two mission examples will be presented, but there are many others.

The functional requirements of the service are:

- This service must give support to the mission planning, allowing the definition of missions as a combination of sequences of 3D waypoints and areas (3D volumes) of manual flight, potentially with user-defined time and altitude constraints.
- This service must support different modes of flight and drones: fixed wing, multicopters.
- The service must ensure compliance with height and time restrictions imposed by the user.
- The service must guarantee compliance with the height and time restrictions calculated for the fulfillment of the mission established by the user.
- This service must use weather data to predict trajectories and assess the viability of the flight considering battery and time constraints.
- The service must translate the completed flight plan to a language that the drone, that is going to fly, understands what it has to do.
- The interface that allows to define mission specifications must be user-friendly to create missions quickly.
- The interface must show the results obtained in the Flight Planning Service, the Mission Planning Service and the Trajectory Prediction Service, i.e. the flight plan and the trajectory.
- All gathered data must be stored and deleted according to the local privacy law.

In addition to the functional requirements of the service, also the non-functional requirements to be accomplished by the Mission Planning Service have to be defined, and these are:

- Performance. The service must be fast and must use the least amount of resources possible because the service can work with a collection of missions from different users. Some use cases (such as emergencies) have very tight time constraints.
- Supportability. Modify, upgrade or add some of the services or components must be easy.
- Replication. The system must be reusable and re-instantiated for different users which can concurrently work on different missions.

- Usability. The system must be able to provide a user-friendly interface to facilitate the creation of missions to users and provide a flight plan and a understandable trajectory.
- Reliability. All databases involved must be stored in periodical backups made by the system. The system must ensure stability to operate at all times, as well as provide reliability against service disruptions.
- Availability. A very high availability must be guaranteed by the system.

Figure 5.1: Life cycle of a mission in Mission Planning Service

Following the life cycle of a mission must be presented as the service requirements, both functional and non-functional, are known. In Figure 5.1, the life cycle of a mission in Mission Planning Service is depicted. Next, the different phases of the life cycle are explained:

1. **Obtaining a mission specification.** In the web interface, the end user can create the mission he/she wants to fly, in the shape he/she wants, for the purpose he/she wants, with the restrictions he/she demands. . . Once the end user has created the mission, it is converted to a list of waypoints understandable by all services, which they can work with and convert into a flight plan.
2. **Obtaining a flight plan.** Once the list of waypoint is obtained, it is sent to the Flight Planning Service where it is converted into a flight plan. The Flight Planning Service sends the flight plan and the predicted trajectory, which are stored in the database.
3. **Obtaining flight instructions.** From the flight plan of the Flight Planning Service, the flight instructions are obtained. This process is carried out by translating the flight plan into a common language that most drones understand (MAVLink) and in this way the drone will execute the created mission following the flight instructions. This flight instructions are displayed on the interface.

5.2 Architecture schema and context

After explaining the service requirements, as well as the life cycle of a mission, the architecture of the Mission Planning Service will be detailed. From the life cycle of a mission and from Figure 5.1, it is possible to obtain the inputs and outputs of the service, which are reflected in Figure 5.2.

Figure 5.2: General schema of the architecture of the Mission Planning Service

Going into more detail on the architecture of the Mission Planning Service, the service architecture follows the architecture that can be seen in Figure 5.3.

Figure 5.3: Specific schema of Mission Planning Service Architecture

The Mission Planning Service is composed of several cooperating functionalities.

Everything starts at the user interface. In it, the user creates the mission according to his/her specific needs, detailing the type of mission he/she will want to perform, the necessary drone and payload, the shape of the mission and the restrictions he/she considers appropriate. The interface presents a map to facilitate the user to explore and know the area where the mission will be carried out. In this way, the user will be able to see the restrictions that the chosen zone presents. It is also possible that the user wants to work with a mission created previously and stored in a database.

Once the mission has been created or selected, it must be transformed into a mission specification so that the Flight Planning Service can work. For this transformation, the service presents the Mission Design functionality that converts it into a mission object that can be understood by the Waypoints calculator functionality. Different Mission Design instances may be available, tailored for different applications. In this thesis, it will describe two examples: simple linear/manual flights and complex inspection flights. The Mission Design instances are in charge of converting the different mission specifications from the HMI into missions. When the mission object is obtained, it is sent to the waypoint calculator functionality. This functionality converts the mission object into a series of waypoints. Then this list of waypoint, which is a flight plan without any constraint, is sent to the Flight Planning Service to obtain the flight plan with the restrictions associated to the generated mission as explained in Chapter 4.

After receiving the complete flight plan from the Flight Planning Service, it is distributed to the rest of the system by the Flight Plan Dispatcher. This complete flight plan is also sent to the functionality in charge of translating it so that the drone understands what it has to do to carry out the mission, the Flight Plan translator. Depending on the drone, this functionality either leaves the flight plan as it is or changes the language so that the drone is able to act. Once translated (or not) it is sent to the GCS (Ground Control System) for the drone to perform the flight plan, and it is also sent to the interface for the end user to see the resulting flight plan from his/her mission.

5.3 Input data models

The input of the Mission Planning Service is a mission specification. The example of the system implemented in this thesis offers the possibility of having two different types of missions. On the one hand, there are missions that consist of a linear flight and a manual inspection or a flight to a delivery. These missions consist of simple elements that can be lines or polygons. On the other hand, there are the missions that consist of automatic inspections, which are constituted by complex 3D elements.

This implies that the system must offer the user the possibility of defining the

mission specification in different ways depending on the type of flight he/she wants to make. Simple missions are specified following the GeoJSON format (Butler et al., 2016), (Butler et al., 2014). This format allows to define the mission as a series of basic forms (Point, LineString or Polygon) that the drone has to follow during the flight to perform the mission. In addition, this format offers the possibility to include properties to each of these basic forms to the user's liking, so the mission can be specified according to the requirements of the mission. Complex inspection missions are specified through a series of geometric shapes with any 3D orientation.

The different types of missions have the following fields in common:

- **Mission ID.** It is a unique string that indicates the identifier of the mission.
- **Operator ID.** It is a string that identifies the operator in charge of this mission.
- **Pilot ID.** It is a string that indicates the identifier of the pilot that the user wants to perform the mission.
- **Drone ID.** It is a string that identifies the drone that the user wants to fly the mission.
- **State.** It is a string that informs about the state of the mission. This state is constantly changing and the possible values are:
 - Edition, when the user is creating the mission.
 - Planned, when the mission has been created and calculated after the whole process of the Mission Planning Service and Flight Planning Service.
 - Active, when the mission is being flown.
 - Terminated, when the mission has finished.
 - Aborted, when the mission is aborted in mid-flight.
 - Removed, when the mission is deleted without being flown.
- **Operation type.** It is a string that indicates the type of operation that is going to be done with this mission. The possible types are:
 - BVLOS (Beyond Visual Line Of Sight).
 - VLOS (Visual Line Of Sight).
 - EVLOS (Extended Visual Line Of Sight).
- **Endurance.** It is the maximum duration of the flight due to battery/fuel limitations and it is represented in seconds.

- **Take Off area.** It is an object that represents the information related with the take off area. It is composed of:
 - **Center.** It is an object that represents the center of the take off area in the form (longitude, latitude) in degrees.
 - **Radius.** It represents the radius, in meters, of the take off area.
 - **Minimum time.** It indicates the minimum time to take off in seconds.
 - **Maximum time.** It represents the maximum time to take off in seconds.
 - **Altitude constraint.** It is an array with the altitude constraint that the user imposes on the take off if he/she wants. These altitudes are above the sea level (AGL) and above ground (WGS84).
- **Landing area.** It is an object that represents the information related with the landing area. It is composed of:
 - **Center.** It is an object that represents the center of the landing area in the form (longitude, latitude) in degrees.
 - **Radius.** It represents the radius, in meters, of the landing area.
 - **Altitude constraint.** It is an array with the altitude constraint that the user imposes on the landing if he/she wants. These altitudes are above the sea level (AGL) and above ground (WGS84).

The fields specifying simple missions are:

- **Flight phases.** It is an object that represents the phases of the flight and it follows the GeoJSON standard (RFC7946). It is composed of:
 - **Geometry.** It is an object that indicates the geometry of this feature and the type of feature that it is. It is composed of:
 - * **Type.** It is a string that indicates the type of feature that is. The possible GeoJSON types are point, lineString and polygon.
 - * **Coordinates.** It is an array of pair of values that indicates the longitude and latitude of each point that contains the feature.
 - **Properties.** It is an object which contains the properties of the features. Its fields are:
 - * **Altitude constraint.** It is an array with the minimum and maximum altitude constraints that the user imposes on each feature if he/she wants. These altitudes are above the sea level (AGL) and above ground (WGS84).

- * **Automatic flight phase.** It is a boolean that determines if the feature is going to be flown by the autopilot or by a human remote pilot.
- * **Groundspeed.** This field is only for points and lineString features and it indicates the minimum and maximum groundspeeds of the drone to make the feature.
- * **Operation duration.** This field is only for polygons and specifies the minimum and maximum time that the drone can stay inside the polygon

Complex inspection missions consist of the concatenation of basic inspections of elements/infrastructures of interest. These basic inspections could be points, lines, catenaries, polygons, walls, circles, cylinders, cones, truncated cylinder, prisms and pyramids. The fields contained in this type of mission object are:

- **Objective list.** It is an object that contains the different objectives that are going to be inspected in the mission. It is composed of:
 - **Inspection list.** It is an object that contains the different basic inspections that the drone must have. Each basic inspection contains:
 - * **Type of inspection.** It is a number that represent the complex inspection selected.
 - * **Inspection data.** It is an object that represent the data that defines the inspection. Table 5.1 shows the different parameters that define each of the inspections. All the inspections have the following information in common.
 - **Inspection ID.** It is a unique string that identifies the inspection.
 - **Maximum distance constraint.** It is the maximum distance the drone can be placed to carry out the mission.
 - **Minimum distance constraint.** This is the minimum distance the drone can be placed to carry out the mission.
 - **Hot points inspection.** It represents an array of interest 3D points that the drone must inspect.
 - **Objective 3D Model.** It is an object that represents the 3D model of the objective that is going to be inspected, in the case that the 3D model exists. It is composed of:
 - * **Position.** It is an object that represents the 3D position of the objective. It has the form (longitude, latitude, altitude) in radians and meters respectively.

- * **Orientation.** It is an object representing the orientation of the objective. It has the form (pitch, roll, yaw) in degrees.
- * **URL.** It represents the url where the 3D model can be downloaded.

Number	Type of inspection	Inspection data
1	Point	1. Position (latitude, longitude, altitude)
2	Line	1. Segment limits. The 3D position of each of the limits. 2. Measure process list.
3	Catenary	1. Segment limits. The 3D position of each of the limits. 2. Minimum height. 3. Measure process list.
4	Wall	1. Parallelogram positions. The 3D position of each of the vertexes. 2. Interior measure process list.
5	Polygon	1. Polygon positions. The 3D position of each of the vertexes. 2. Interior measure process list.
6	Circle	1. Center. The 3D position of the center. 2. Radius. 3. Minimum height. 4. Interior measure process list. 5. Border measure process list.
7	Prism	1. Base polygon positions. The 3D postions of each of the vertexes of the base. 2. Edge length. 3. Edge unitary vector. It represents the heading and elevation. 4. Faces interior measure process list. 5. Base interior measure process list.
8	Cylinder	1. Base center. The 3D position of the base center. 2. Base radius. 3. Top center. The 3D position of the top center. 4. Top radius. 5. Base border measure process list. 6. Base interior measure process list. 7. Side interior measure process list.

9	Pyramid	<ol style="list-style-type: none"> 1. Base polygon positions. The 3D postions of each of the vertexes of the base. 2. Edge length. 3. Opposite vertex 3D position. 4. Faces interior measure process list. 5. Faces edge measure process list. 6. Base interior measure process list. 7. Base edge measure process list.
10	Cone	<ol style="list-style-type: none"> 1. Base center. The 3D position of the base center. 2. Base radius. 3. Top center. The 3D position of the top center. 4. Top radius equal to 0. 5. Base border measure process list. 6. Base interior measure process list. 7. Side interior measure process list.
11	Truncated cone	<ol style="list-style-type: none"> 1. Base center. The 3D position of the base center. 2. Base radius. 3. Top center. The 3D position of the top center. 4. Top radius. 5. Base border measure process list. 6. Base interior measure process list. 7. Side interior measure process list.

Table 5.1: Inspection types and their data

Each basic inspection may have one or several measurement processes detailing the sensor or sensors that must be used to perform the measurements and the sampling process for the inspection. The information that must be provided for each sensor is:

- **Sensor name.** It is a identifier of the sensor to be used.
- **Measurement duration.** It is the time needed to complete all the measures at each sampling position.
- **Measurement period.** It represents the time between two measures at each sampling position.
- **Linear sampling distance.** It is the linear distance in meters between two sampling positions.

- **Angular sampling distance.** It represents the angular distance in radians between two sampling positions.

5.4 Mission planning functionality

Next the mission planning functionality will be detailed. The first subsection, Subsection 5.4.1, explains the process to obtain the list of waypoint that is sent to the Flight Planning Service for simple missions, while the Subsection 5.4.2 explains it for complex inspection missions. The next subsection, Subsection 5.4.3 exposes the explanation of the translation of the flight plan. Finally, the execution of the flight plan is presented in Subsection 5.4.4.

5.4.1 Waypoint calculation for simple missions

This functionality converts the list of flight phases that arrives in the simple mission object into a series of waypoints through which the drone has to pass to carry out the mission.

Although the calculation process is different in the different types of missions, all coordinates are transformed from the Geographic coordinate system (latitude, longitude) to Universal Transverse Mercator coordinate system (UTM coordinates) (Langley, 1998) and (Manchuk and Deutsch, 2009). Universal Transverse Mercator coordinate system is a coordinate system based on Mercator's transverse cartographic projection, which is built like the normal Mercator projection, but instead of making it tangent to the equator, it is made secant to a meridian. Unlike the system of geographic coordinates, expressed in longitude and latitude, the magnitudes in the UTM system are expressed in meters only at sea level, which is the basis of the projection of the reference ellipsoid. This conversion is necessary because all calculations that are made must be performed in meters as it is the usual and simplest way to work with distances and positions.

In this case where the simple mission consists of a flight and a manual inspection or is a delivery, the mission will be written following the GeoJSON format as it was commented in 5.3. This format allows the definition of Points, LineStrings and Polygons passing as parameter the 2D position of the points, the segments that form the line and the vertices that delimit the polygon respectively. Therefore, in this case, in the object flight phases comes a list of these geometrical shapes, in which the 2D positions are already defined. The only thing that the calculation of waypoints functionality has to do is to go through the list of geometrical shapes and one by one obtain the 2D positions of the waypoints that form them and to pass them to coordinates UTM. But the first point to insert in the list is the takeoff area. Both lists, the list formed by the waypoints in geographic coordinates and the list with

the waypoints in UTM coordinates are stored to return one, after all the necessary modifications and to work with the other.

To complete the waypoints list, it is necessary to calculate the landing area and add it.

After this waypoint calculation process, a list of waypoints is obtained and sent, together with their properties, to the Flight Planning Service.

5.4.2 Waypoint calculation for complex inspection missions

This functionality converts the list of intermediate inspections that arrives in the mission object input into a series of waypoints and obtains a list of waypoints through which the drone has to pass to carry out the mission.

As the mission is an automatic inspection, the mission is built following the format created by this purpose. In this type of mission, the Geographic coordinate system is also transformed to Universal Transverse Mercator coordinate system and the first point to add to the waypoints list is the takeoff area. In this case, complex inspections are allowed and the object objective list of the mission object is made up of each of the allowed inspections which have been detailed in 5.3. For example, a prism is formed by the waypoints that cover the base and the height of the prism. Next it is going to be detailed an example of the calculation of waypoints for a cylinder type inspection.

In the case where the inspection is a straight cylinder, five are the geometric figures to be inspected and each of them has an independent measurement process. The measurement processes are explained in more detail in 5.3. The five geometric figures to be inspected are:

1. **Bottom Base Circumference.** This figure is sampled at constant angular distance.
2. **Bottom Base Circle.** This is sampled using polar coordinates at constant angular and linear sampling distances.
3. **Top Base Circumference.** As well as the bottom one, this is sampled at constant angular distance.
4. **Top Base Circle.** Just like the bottom, this is sampled using polar coordinates at constant angular and linear sampling distances.
5. **Cylinder Wall.** This figure is sampled using constant angular and height-aligned linear sampling distances. This is the sampling process that is going to be detailed next.

To obtain the sampling points that composed the cylinder wall the next input data is required and these cylinder parameters are depicted in Figure 5.4:

1. **Cylinder Radius.** (R)
2. **Bottom Base 3D center.** ($\vec{x}_b = (x_b, y_b, z_b)$)
3. **Top Base 3D center.** ($\vec{x}_t = (x_t, y_t, z_t)$)
4. **Maximum drone separation to the cylinder.** It represents the maximum separation from which the drone has to inspect the cylinder (d_{max}).
5. **Minimum drone separation to the cylinder.** It represents the minimum separation from which the drone has to inspect the cylinder (d_{min}).
6. **Linear Sampling distance.** Vertical distance between two sampling points in meters (d_H).
7. **Angular Sampling distance.** Horizontal distance between two sampling points in radians (d_θ).

Figure 5.4: Cylinder parameters and sampling points

In Figure 5.4, the blue points represent the sampling points that result of the cylinder wall sampling process as described below.

To inspect a cylinder wall, a set of circles at different height is built. To know the number of needed circles ($n_{circles}$), the calculation to make is:

$$n_{circles} = \frac{\sqrt{(x_t - x_b)^2 + (y_t - y_b)^2 + (z_t - z_b)^2}}{d_H} \quad (5.1)$$

The sampling points are calculated in a reference plane centered in the bottom circle cylinder, with the height axis along the axis of the cylinder. First it is needed to calculate the polar coordinates of those points $((R, \theta_i, H_j))$, where i subindex goes from 0 to $2\pi/d_\theta$ and j subindex goes from 0 to $n_{circles}$. These coordinates are calculated as $((R, \theta_i, H_j) = (R, id_\theta, jd_H))$. The next step is to convert the sampling points obtained to Cartesian coordinates in the previous reference plane established as $((x_{i,j}, y_{i,j}, i, j) = (R \cos(\theta_i), R \sin \theta_i, H_j))$. Each of these sampling points calculated need to be rotated and translated to local horizontal XYZ centred in take off position, so the elevation (α) and the heading (β) of the cylinder are needed. These two angles are calculated from the vector difference of the circles centres ($\vec{d} = (d_x, d_y, d_z) = (x_t - x_b, y_t - y_b, z_t - z_b)$) as:

$$\begin{aligned}\alpha &= \frac{\pi}{2} - \arctan \left(\frac{\sqrt{d_x^2 + d_y^2}}{d_z} \right) \\ \beta &= \arctan \left(\frac{d_y}{d_x} \right)\end{aligned}\tag{5.2}$$

To rotate and translate the sampling points to local horizontal XYZ centred in take off position $(x_s(i, j), y_s(i, j), z_s(i, j))$, it is had:

$$\begin{bmatrix} x_s(i, j) \\ y_s(i, j) \\ z_s(i, j) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & -\sin(\alpha) \\ 0 & \sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} \cos(\beta) & -\sin(\beta) & 0 \\ \sin(\beta) & \cos(\beta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{i,j} \\ y_{i,j} \\ z_{i,j} \end{bmatrix} + \begin{bmatrix} x_b \\ y_b \\ z_b \end{bmatrix}\tag{5.3}$$

Once the points are calculated, a strategy to obtain the waypoints must be follow. There are many potential strategies for this generation but, in this thesis implementation, there are three different approaches:

- Normal to surface illumination.
- Defined constant camera tilt and drone yaw angles.
- Defined constant camera tilt, normal to surface drone yaw angle.

Some of these strategies can not be used in some inspections because lead to ambiguous definition of the waypoint positions. In addition, it is assumed in the model that the sensor gimbal does not rotate horizontally, so the line of sight direction is set by drone yaw angle. This assumption simplifies the calculation.

From a sampling point, a waypoint is calculated in the perpendicular direction to the cylinder wall at a distance between d_{min} and d_{max} . To do this, the first

thing is to define the difference vector from the bottom center to the sampling point (i,j) of the inspection $\vec{d}_{i,j} = (d_x(i,j), d_y(i,j), d_z(i,j)) = (x_s(i,j) - x_b(i,j), y_s(i,j) - y_b(i,j), z_s(i,j) - z_b(i,j))$. The normal vector $(\vec{n}_{i,j} = (n_x, n_y, n_z))$ is calculated using cross vector product:

$$\vec{n}_{i,j} = (\vec{d} \times \vec{d}_{i,j}) \times \vec{d} \quad (5.4)$$

The waypoints positions are obtained adding to the previous waypoint position a distance equal to the average between the minimum and maximum distance ($d_{med} = \frac{d_{min} + d_{max}}{2}$) because we are moving along the normal vector calculated before. Following, the resulting equations for calculating the positions are:

$$\begin{aligned} x_W(i,j) &= x_s(i,j) + d_{med} \frac{n_x}{\sqrt{n_x^2 + n_y^2 + n_z^2}} \\ y_W(i,j) &= y_s(i,j) + d_{med} \frac{n_y}{\sqrt{n_x^2 + n_y^2 + n_z^2}} \\ z_W(i,j) &= z_s(i,j) + d_{med} \frac{n_z}{\sqrt{n_x^2 + n_y^2 + n_z^2}} \end{aligned} \quad (5.5)$$

The resulting waypoints can be seen in Figure 5.5.

Figure 5.5: Input data to get the waypoints orthogonal

If, on the other hand, a constant camera tilt (ϕ) and drone yaw (θ) angles approach is preferred, the calculation process differs with the previous one. Instead the normal vector, the unitary vector ($\vec{u}$) from the sampling point to the waypoint is

first calculate the yaw angle using 5.4 to calculate the normal vector to cylinder wall surface and following is to make the calculations for the constant tilt and yaw approach using this calculated yaw.

This has been the explanation of calculation of waypoints for a cylinder and like this, different processes of obtaining waypoints have been developed for each of the inspection shapes.

To complete the waypoints list, it is necessary to calculate the landing area and incorporate it to the list of waypoints. But the waypoints of the inspections have to go through an ordering process beforehand. The two possible ordering strategies are: search for an efficient order following the shortest path or using a Disjkstra algorithm or make use of predefined order strategies. In this functionality the strategy followed is the second since the objective is to facilitate the manual control minimizing the number of control variables to be modified in the legs. Thus, each inspection has different ordering strategies defined so that the inspection is carried out in the least number of legs possible. In addition, the functionality has also minimized the number of legs to join the different inspections.

After this waypoint calculation process, a list of waypoints is obtained and sent, together with their properties, to the Flight Planning Service.

5.4.3 Flight Plan translation

This functionality is in charge of translating the flight plan so that the drone that is going to make the mission understands the flight it has to make, whatever the drone is. So, in order to actually implement the flight in an autonomous way, the Flight Plan translator functionality translates the flight plan into languages understandable by the drone autopilot/FCS and measurement control systems.

As mentioned in Section 1.1.1, nowadays there are many commercial drones on the market, some of which have their own languages.

In the case of an implementation for DJI drone (DJI, 2020d), (DJI, 2020e), the flight plan is implemented using DJI SDK (DJI, 2020c) to control the autopilot from the flight plan data models. This implementation of the DJI SDK takes each of the instructions that fulfil the flight plan and translates it, using its language, into an instruction that the drone understands to carry out the flight plan instruction that it goes next. It consists of a set of instructions (take off, go up, turn, increase speed, decrease speed, land ...) to which some parameters are passed and the SDK is responsible for making the necessary translations for the drone to perform the mission entrusted to it.

On the other hand, there are many drones that can be controllabled through MAVLink, such as those of Parrot (Parrot, 2019). MAVLink or Micro Air Vehicle Link (QgroundControl, 2019) is a protocol for communicating with small unmanned vehicle that is based on a very lightweight, header-only message marshalling library

5.4.4 Flight Plan execution

Once the flight plan has been created and translated, it could already be flown because it is already sent to the GCS (Ground Control Station).

The Ground Control Station System is quite often an application that the pilots have in their PCs, smartphones or tablets that allows the integration of drones in the drone fleet platform.

It allows to visualize a list of previously generated flight plans. these flight plans can be flown both manually and automatically. In the latter case, the pilot could override the automated behaviour to take him/her the control and deviate the aircraft from the flight plan. It may be also possible that the pilot can fly the drone without a flight plan. So, the pilot only has to select in the GCS the flight plan that was generated before and the mission is executed.

During this execution, the drone typically sends telemetry information periodically to the GCS that is re-sent to the system, specifically to the Tracking and Monitoring Service, that is going to be explained in Chapter 6.

5.5 Results of mission planning service

In this section, a full example of application of the previously described techniques and some performance results to identify the computational bottleneck are provided. First of all, an inspection mission will be carried out in which the result of the waypoint calculation will be shown in Section 5.5.1. Then, a exemplary flight plan is created and flown to check the mission trajectory calculation in Section 5.5.2. Finally, in Section 5.5.3 a performance evaluation will be carried out.

5.5.1 Waypoint calculation validation

Next it is showed an example of a flight plan design to inspect a telephone antenna. The first step is to define the desired target location. To do this, it is used the map-based interface and the user colocates the inspection pattern in the map. The interface offers the possibility to modify the inspection pattern graphically using transformations based on placeholders dragging or through text input. Then, other mission information must be defined to complete the definition of the measurement processes for the inspection. To finish the mission specification, the user must introduce the initial and final waypoints (take-off area and landing area) which are defined by clicking on the map. So, in this example, the mission specification can be summarised as a take-off area near to the telephone antenna, a cylinder 20 meters in radius and 10 meters in height and a landing area near to the telephone antenna.

Once the mission has been created, the mission specification arrives to calculate

waypoints functionality. In this example, as the cylinder has 10 meters in height and the measurement process to inspect the antenna is 5 meters in height, 2 sets of waypoints at different heights are calculated. This calculation results can be seen in Figure 5.8. On its left side, the created inspection shape can be seen in the 3D view with green lines. The red dots represent the calculated waypoints. From each waypoint, a yellow line is pointing towards the inspection which indicate the bearing direction of the sensors. On the other hand, the right part of the figure holds the interactive map that shows the 2D representation of the line that joins the waypoints.

Figure 5.8: Calculated waypoints views: **(left)** lateral **(right)** zenit

In the Figure 5.8 it can be seen that the functionality performs its task properly. Now, in Figure 5.9, a more complex mission is performed to check if the functionality also gets the right result. It is composed of a straight cylinder with a truncated cone on top and a wall. Each type of inspection has its own measurement process and the calculation of the waypoints is different depending on the type of inspection. The straight cylinder has 20 meters in radius and 10 meters in height, while the measurement process is 3 meters in height and every 60° around the cylinder. The truncated cone has a 20 meters base radius and 10 meters top radius with a height of 10 meters and the measurement process is 5 meters in height and every 60° around the truncated cone. The wall is 10x10 meters and the measurement process is 5 meters in vertical and horizontal.

These examples show that regardless of the type of mission the user wants, the flight plan obtained is common and in the same format. In this way, different input sources can generate the same output flight plan.

5.5.2 Results of trajectory

In order to show the capability of the service to predict the drone trajectory, a short flight plan is defined to be performed by a DJI Phantom drone. To see how the

Figure 5.9: A more complex mission with multiple inspections

service works and whether the prediction is accurate, the generated flight plan will be translated into language the drone understands and will be executed so that a comparison between flown and predicted trajectories could be made. The scenario to be analyzed is the one shown in Figure 5.10, where the instructions that model the routes between waypoints can be summarized as:

- Take off until the selected height is reached and, once reached, turn to the next waypoint.
- Go to the next waypoint, accelerating to the required constant speed and decelerating before reaching the waypoint to stop at the set position.
- Rotate to the next waypoint and repeat the previous step.
- Once the last waypoint is reached, land.

This mission, converted to a flight plan, has been provided to the drone and flown. On the other hand, the predicted trajectory has been calculated from the

Figure 5.10: Detailed of the mission to be performed

mission and the results of the comparison between both trajectories can be seen in the following figures. Also, the capability of the proposed techniques are evaluated to accurately model the trajectory of multirotors in the presence of wind. Ten iterations of the same trajectory (the previously explained) have been flown at different times and under different wind conditions. These actual flights may have some horizontal and/or vertical deviation due to GPS accuracy. Depending on how the receiver picks up the GPS signal, the positions transmitted by the drone will be more or less adjusted to the predicted trajectory. But it will never have a large deviation since the error for GPS accuracy is very small. In the vertical the DJI Phantom GPS has an accuracy of ± 0.5 meters and in the horizontal it has an accuracy of ± 1.5 meters (DJI, 2019).

In Figure 5.11, Figure 5.12, Figure 5.13 the blue lines represent the actual ten measured flights according to drone telemetry, while the red line is the trajectory prediction. These ten measured flights were performed on different days, in different moments of time and with different weather conditions to represent a greater variety of conditions in possible missions. Thanks to this distribution of the flights in the time, the missions were carried out in the absence of wind, a very favorable condition for the realization of the mission, as well as with different wind speeds and directions that can affect the execution of the flight. The drone used, the DJI Phantom, has a maximum wind resistance of 10 m/s (DJI, 2019), which means that the drone is able to stabilize the flight by compensating the wind speed to follow the trajectory that has been sent. If the wind speed is higher, the drone is not able to compensate and would move a little bit with respect to the sent trajectory.

It can be seen that the system is able to accurately predict the flight position vs time for different wind conditions (in these experiments winds have speeds up to 15 m/s), especially in the lateral and vertical dimensions, where errors were always well

Figure 5.11: X of predicted and real flights

Figure 5.12: Y of predicted and real flights

below 1 m. In the longitudinal dimension there is an initial error (random take off delay), and an additional time drift due to groundspeed mismatch, mainly during maneuvers. In the altitude graph, it can be seen that the predicted trajectory starts to land earlier than the real ones. This behavior is correct since in real flights the landing is done manually so that the pilot makes sure that there is nothing in the

Figure 5.13: Altitude of predicted and real flights

landing zone that could endanger this operation.

As it can be seen, the wind speed does not influence too much the position of the drone, even having winds with speeds higher than the maximum speed resisted by the drone. The vertical and horizontal displacements presented by the ten flights executed can come from the GPS precision as well as from the wind. But with wind speeds lower than 10 m/s the dominant imprecision must be caused by the GPS, while at wind speeds higher than 10 m/s the dominant imprecision must be caused by the wind. Although observing the results, neither of the two sources of uncertainty causes an error that prevents the flights from being carried out.

5.5.3 Mission execution time

The performance of the Mission Planning Service is calculated in this section. For that, the time spent in calculating the trajectory is measured using a dual CPU Intel® Xeon® E5-2620 2.00 GHz server with SSD storage. To test this performance evaluation, the three different flight plans of different lengths used in Section 4.4.2 are measured. The time results obtained with these tests can be seen in Table 5.2.

From this temporal evaluation of the performance of the Mission Planning Service it can be concluded that there is not a huge bottleneck, although the difference of time between functionalities is big. This difference may be due to the transformation of instructions into a trajectory, which consists of a large number of calculations, and the opening and writing of a new file to include the trajectory, which usually takes quite a long time. But the times are within reasonable margins to be able to

Functionality	Short FP	Average FP	Long FP
Trajectory calculation	4.367 s	7.832 s	11.106 s
Translation	0.437 s	0.762 s	1.052 s
Total	4.804 s	8.594 s	12.158 s

Table 5.2: Example parameters

work in real time, so there is no problem in using this service in a system that works in real time.

5.6 Conclusions

In this chapter the Mission Planning Service has been defined. Starting from the requirements that a service like this must have and the service interactions, the architecture has been analyzed, studied and developed. And also, input data models have been detailed to know what a mission consists of.

As it can be seen in the figure of the Mission Planning Service architecture, it is composed of different functionalities that carry out the functions required for the transformation from a mission specification to a trajectory passing through a flight plan.

It has been concluded that the Mission Planning Service generates the list of waypoints that the Flight Planning Service needs, translates the flight plan so that the different drones in the fleet can carry out the different missions and generates the trajectories with which to compare or work through simulation to improve the system. It does so correctly and with a not very high execution time that could be improved in future versions with further optimizations.

Chapter 6

Tracking and Monitoring service

The missions, flight plans and trajectories created in the in-flight services are flown by drones piloted by pilots or autonomously and must be tracked by the in-flight services. In addition to being tracked, they must be monitored at all times to know if the user-generated mission is being flown correctly or if problems are happening and the flight must be aborted or modified to ensure the safety of people, infrastructure and material goods. It is in this aspect where the tracking and monitoring service comes into operation, obtaining the telemetry of all the drones that are flying and checking that the different safety related restrictions imposed on the mission are being fulfilled. Therefore, the tracking and monitoring service is connected to the pre-flight services.

The high-level service requirements, as well as its interactions are presented in Section 6.1. In Section 6.2, a schema of its architecture and its context is described. The functions that the Tracking and Monitoring Service performs are listed in Section 6.3. Data models are presented in Section 6.4. Section 6.5 is in charge of presenting and analyzing the calculations that allow the tracking and monitoring of the drones. Once these calculations have been explained, the results are presented in Section 6.6. Finally, the conclusions of the Tracking and Monitoring Service are presented in Section 6.7.

6.1 High level requirements and service interactions

The functional requirements of the service are:

- The service will provide the most current estimate of the drone/manned aircraft kinematic state.

- Potential changes in the flight context or unpredicted conflicting/emergency situations must be monitored to allow the system to alert of potentially unsafe situations.
- During the flights, the system must be able to receive, process and store the information from all sensors.
- This service will track cooperative drones, using cooperative sensors such as ADS-B, direct GPS telemetry via mobile, GPS telemetry via through GCS forwarding, mobile localization based location . . . , providing a single track by drone.
- The service should support the incorporation of ATM tracks in Asterix62 format (Eurocontrol, 2019).
- This service should also incorporate other UTM systems tracks.
- The service will try to correlate track data with active flight plans. If there is a track with no corresponding flight plan, an alert will be issued.
- The tracking service may use a correlated active flight plan to enhance tracking.
- The tracking service must have means to initiate tracks and maintain track continuity just on the basis of measures.
- The monitoring service must perform an assessment of the lateral and vertical excursion and time drift with respect to the intended trajectory. If the excursion or time drift is too high, an alert should be issued.
- This monitoring service must perform an assessment of the compatibility with air rules, i.e. minimum and maximum altitude, maximum speed, airspace protected areas, and raise potential conformance alerts accordingly.
- The monitoring service must perform an assessment of the incursion of the track into geofences volumes. If some geofence is violated, the system must issue an alert.
- This information may be shown in real-time to different users and on different devices. These must be associated to the user permissions.
- The service must be able to delete tracks when no additional measure has been received for a certain time interval.

- All gathered data must be stored and deleted according to the local privacy law.

In addition to the functional requirements of the service, also the non-functional requirements to be accomplished by the Tracking and Monitoring Service have to be defined, and these are:

- Performance. The service must be fast and use the least amount of resources possible. The service must work in real time.
- Supportability. Modify, upgrade or add some of the components must be easy.
- Replication. The service must be reusable and re-instatiated for different end-clients as different end users can monitor its own tracks at the same time.
- Usability. The service must be able to provide a user-friendly interface to facilitate the flight monitoring. In this way, any final user familiar with air traffic control must be able to answer quickly if he or she perceives a problem in order to ensure safety.
- Reliability. All databases involved must be stored in periodical backups made by the service. The system must ensure stability to operate at all times, as well as provide reliability against service disruptions.
- Availability. A very high rate of availability must be guaranteed by the service.

After presenting the functional and non-functional requirements of the Tracking and Monitoring Service, then the life cycle of a track must be presented. The different functions that the service is performed, that are going to be explained in Section 6.3, will be described summarising the interaction with the Tracking and Monitoring Service. In Figure 6.1, the life cycle of a track in this service is depicted. Finally, the different phases of the life cycle are explained:

1. **Transform telemetry/tracks/other surveillance information into track.**

When the drone is flying, it sends its telemetry to the GCS, so the pilot can know the position, speed, attitude and battery of the drone. This telemetry is sent to the Tracking and Monitoring service via MQTT. It is also posible that the service receive tracks from other services/systems or other surveillance information from other sources. All this information could be a input of the service. Then, the data is parsed and a track is initialized if it is the first telemetry message of that drone or the track is updated. Tracks are maintained during the whole flight.

Figure 6.1: Life cycle of a track in Tracking and Monitoring Service

2. **Correlate the tracks.** Since the system has different input sources that may be measuring the position of the same target, it is necessary to correlate the tracks from the different input sources to obtain a single track. This correlation is based on code and distance. Once the correlation has been made, the track information for a target can come from one or more input sources.
3. **Obtaining flight plan conformance.** Once the track is obtained, it is sent to the Deviation Detector to obtain the flight plan conformance. To do this, the flight plan that the drone is flying is obtained for the database of flight plans. And then, the position sent by the drone is compared with the one that the flight plan said to that instant of time. In this manner, deviations of the drone from the flight plan are obtained and added to the track.
4. **Obtaining geofences intrusions.** Then, the position in the track is sent to the No Drone Zone Model explained in Section 4.3.4.1 and the geofences invaded are obtained. This process is performed in the Break-In Detector. These invaded geofences are added to the track.
5. **Obtaining obstacles.** Next, the position of the track is sent to the Ground Service explained in 4.3.3.1. The task of this service is to provide the height at which the ground is or any obstacles that may appear in the way, as it is

important to avoid collisions with the ground or obstacles. If any obstacle is detected, it is added to the track. This process is made in Obstacles Detector.

6. **Obtaining nearby aerial traffic.** Finally, the nearby traffic is checked and if the distance between aircrafts is small enough to cause some kind of accident, so the pilot must be alert, this aircraft is added to the track as a blind track.
7. **Send complete tracks and alarms.** Once the track is completed with the information of the previous steps, the service is ready to send it via MQTT to the GCS and via REST to the HMI. In this way, the pilot or the persons in charge of control the UAVS traffic could observe the traffic, the alerts and the possible problems that may appear and thus act as soon as possible to solve it.

6.2 Architecture schema and context

After explaining the service requirements, as well as the life cycle of a track, the architecture of the Tracking and Monitoring Service is detailed hereunder. From the life cycle of a track and from Figure 6.1, it is possible to obtain the inputs and outputs of the service, which are reflected in Figure 6.2.

Figure 6.2: General schema of the architecture of the Tracking and Monitoring Service

The service architecture can be seen in Figure 6.3.

The Tracking and Monitoring Service is composed of several cooperating functionalities. Everything begins when a mission is started, so a drone goes to fly driven by a pilot or following an automated flight plan. This drone sends its telemetry information to the pilot's GCS (Ground Control Station), and then, the GCS via MQTT publishes that telemetry information so the Tracking and Monitoring Service can receive these messages. It is also possible that the drone doesn't send the telemetry

Figure 6.3: Specific schema of Tracking and Monitoring Service Architecture

messages because it doesn't have the capacity of generating these messages or there may be a problem in the data link so the messages are not sent or are lost. This is the reason for the existence of other instruments capable of detecting the drones and sending their position and speed in order to monitor them. These are the so-called other surveillance systems, which can be anything from radars capable of detecting drones to surveillance cameras. All these devices detect the position of the drone and generate positioning messages that are also sent to the Tracking and Monitoring service.

Once this information arrives at the tracker, the track must be built or updated. To do this it is necessary to see if the data comes from a drone that is already being monitored or is the first data received. If the latter occurs, the tracker builds a new track with the drone position, speed and covariances, as well as the drone and track identifier to know how to associate the next messages from that drone with this track that has just been generated. If, on the contrary, the messages are related to a drone that already has a track in the system, the tracker only has to update the track information. The association process must be very careful not to

confuse tracks and therefore mix the information of different drones. Also, in the association process, the tracker associates the information that comes from different surveillance systems, since a drone can send its telemetry but also be followed by another surveillance service, so different messages must be associated to a single track and not generate multiple tracks for the same drone. This process will be further explained in Section 6.5.1. Finally, the tracker must delete those tracks that have not been updated for a certain period of time, in order to remove them from the system and stop monitoring them because the flight has already ended. Once the tracks are built or updated, the tracker has to predict the tracks and filter them. Different methods can be used here and will be discussed in Section 6.5.1.

To perform flight monitoring, the track goes through a series of functionalities where different behaviours are analysed. These functionalities are deviation detection, break-in detection and obstacle detection and will be extensively explained in Sections 6.5.3 and following. The deviation detector is in charge of checking if the drone follows a flight plan generated in the Flight Planning Service. The break-in detector functionality indicates if a drone is invading a geofence where its entry is not allowed. Finally, the obstacle detector functionality fulfills two tasks. The first one is to check if the distance of the drone to the ground or to any obstacle is too small. In addition, the functionality is in charge of checking the distance with the rest of the drones that are being monitored to make sure that no flight is endangering the flight of another drone. To do this, the distances to the nearest drones are calculated and added to the track, and if one distance is too small, an alarm is generated.

Finally, once the predicted tracks are completed with all the monitoring information, they are sent to interested users. On one hand, the tracks are sent to the pilots who are flying via MQTT. This way they know all the information about the flight they are doing. And on the other hand, the tracks are sent to the fleet management HMI so that the operator may monitor the correct operation of the different flights.

6.3 Functions

Once the architecture of the Tracking and Monitoring Service has been presented and the functionalities of which it is composed have been briefly explained, it is now necessary to make a list of its functions:

- Processing of all drones' telemetry that it is sent to the tracker functionality of the Tracking and Monitoring Service.
- Processing of other surveillance systems measures. Non-cooperative sensors information as radar, infrared and cameras, ADS-B data from equipped drones,

UTM system tracks and ATC/ATM tracks are included. With these inclusions the system manages to have a greater number of aircraft monitored, and different sensors allow to track aircraft so that in the event of the loss of data from one of them, other tracking sensors are still available to detect it and guarantee the safety of the airspace. The idea is to be able to work as a federation of monitoring services from different vendors, being able to adapt to different UTM architectures.

- Presentation of drones in flight related to users. Each user should only see the drones allowed to its role and ownership, and the ones that conflict with theirs, so that all privacy constraints are respected.
- Processing of all flight plan information and correlation with tracks, based on both telemetry or other sensors positional information.
- Alerts to users of aircraft deviations from the planned route after comparing the telemetry received and the flight plan created and authorized.
- Alerts to users of incursions in restricted areas or areas with a set of rules that must be satisfied.
- Alerts to users of obstacles presence comparing telemetry sent by the drone and the AGL height obtained making use of a high resolution and accuracy ground service.
- Presentation of weather information to assist with navigation in adverse conditions.
- Distributed monitoring, configuration, recording and technical exploitation of the entire service for different users.

6.4 Data Models

Before continuing with the explanation of the calculations that are implemented in the different functionalities, the definitions of the telemetry and track data models are necessary. The telemetry message in the model used in this thesis is the main input model to the Tracking and Monitoring Service and contains the measurement of the position and other characteristics of a drone. The fields contained in the telemetry message are:

- **Position.** It is an object that represents the position where a drone is. It is composed of three numbers that represent, in this order, longitude in degrees, latitude in degrees and WGS84 altitude in meters.

- **Attitude.** It is an object that represents the attitude that a drone has. It is composed of three numbers that represent, in this order, yaw, roll and pitch, all in degrees.
- **Speed.** It is an object that represents the ground speed that a drone has. It is composed of three numbers that represent, in this order, ground speed in m/s , track angle in degrees and vertical speed in m/s .
- **Battery.** It indicates the battery level and it is represented as a percentage.
- **Time.** It is a timestamp that indicates the moment that the drone send this telemetry message.
- **Pilot alerts.** This field represents the possible alerts that a pilot can send during a flight with a drone, so the field could be an array with the following alerts that may or may not be present:
 - LoL. Loss of Link.
 - LoS. Loss of Signal.
 - LoG. Loss of GPS.
 - LoE. Loss of Engine.
 - LoP. Loss of Power.
 - LoO. Loss of Ownership.

The other specially relevant data model is the track. The track is the output data model of the Tracking and Monitoring Service, but it could also be another possible input data model to the service if the service is connected to other monitoring systems in a federated architecture. The main fields contained in the track are:

- **Track number.** It is a number that represents an identifier of a track belonging to a drone. This number must be unique.
- **Time.** It is the reference time of track information (which may be extrapolated).
- **Last time.** It is the time of the last update (last measure incorporated to the track).
- **Last telemetry time.** It is the time of last telemetry data update.
- **Tracking identifiers.** It is an object that contains the identifiers of the drone. Some possible values could be ADSB code, telemetry identifier, ...

- **Drone identifier.** It is the identifier that represents the drone this track belongs to.
- **Sources.** It is an array that contains the type of sensors used in the drone. Some possible values could be telemetry, ADSB, ATC, non-cooperative sensor, ...
- **Height source.** It is an array that contains the type of the source that measures the height. The possible values are GNSS, geo altimeter or baro altimeter.
- **Correlated flight plan.** It is a string with the identifier of the flight plan that the drone is following. It could be possible that the field will be undefined if the track is a non-correlated track.
- **Position.** It is an object that represents the position where a drone is. It is composed of three numbers that represent, in this order, longitude in degrees, latitude in degrees and altitude in meters.
- **Speed.** It is an object that represents the speed that a drone has. It is composed of three numbers that represent, in this order, ground speed in m/s , track angle in degrees and vertical speed in m/s .
- **Horizontal covariance.** It represents the horizontal state covariance. It is composed of 10 numbers that represent the longitude standard deviation in meters, the latitude standard deviation in meters, the speed in x axis standard deviation in m/s , the speed in y axis standard deviation in m/s , the correlation coefficient between longitude and latitude, the correlation coefficient between longitude and speed in x axis, the correlation coefficient between longitude and speed in y axis, the correlation coefficient between latitude and speed in x axis, the correlation coefficient between latitude and speed in y axis and the correlation coefficient between speed in x axis and speed in y axis.
- **Vertical covariance.** It represents the vertical state covariance. It is composed of 3 numbers that represent the altitude standard deviation in meters, the vertical speed standard deviation in m/s and the correlation coefficient between the altitude and vertical speed.
- **Conformance monitoring state.** It is an array that represents the deviation with respect to the central trajectory prediction. It is composed of lateral drift in meters, vertical drift in meters and time drift in seconds.
- **Conformance monitoring alert.** It is an object that appears when a drone is deviated with respect to the central trajectory prediction or does not have

a correlated flight plan. It can be LATERAL, that represents an excessive lateral drift, VERTICAL if the drone is vertical deviated, TIME if the drone is too much behind or ahead of time and NOFP if the drone does not have a correlated flight plan.

- **Geofence violation alert.** It is an object composed of the id of the involved geofences in which the drone is and the distances to that geofences.
- **Obstacles alert.** It is an object composed of the distances to the obstacles detected.
- **Tactical conflict.** It is an object that is composed of an alert field that represents if there is a tactical conflict or not and a nearby traffic object that it is a list of close tracks that contains the track number, the drone identifier, the position of the close track, the speed of the close track and the drone's type.
- **Telemetry alert.** This field represents the possible alerts that a pilot can send during a flight with a drone, so the field could be an array with the following alerts that may or may not be present:
 - LoL. Loss of Link.
 - LoS. Loss of Signal.
 - LoG. Loss of GPS.
 - LoE. Loss of Engine.
 - LoP. Loss of Power.
 - LoO. Loss of Ownership.

6.5 Tracking and monitoring calculations

Each subsection explains a different part of the service, starting with the tracker in Subsection 6.5.1. Then, the integration of other surveillance systems is explained in Subsection 6.5.2. Finally, the three detectors are explained later: the break-in detector in Subsection 6.5.3, the deviation detector in Subsection 6.5.4 and the obstacles detector in Subsection 6.5.5.

6.5.1 Tracker

To be able to detect and track all the drones simultaneously, it has been implemented two different types of filters for each track making use of the flight plan information

to improve tracking for the horizontal part of the measurement, while a separate filter has been made for the vertical part. From the horizontal filter, one of the implemented filters is an adaptation of map-based tracking using the flight plan information, loosely inspired in the approaches in (Kirubarajan et al., 2000)(Blasch et al., 2014). And the other one is a robust filter that consists of an Interacting Multiple Model (IMM) algorithm (Li and Bar-Shalom, 1993) (Cho et al., 2013). The vertical filter is an IMM filter.

The structure implemented to carry out the tracker function can be seen in Figure 6.4.

Figure 6.4: Global tracker structure

Each drone sends telemetry few times per second (with a certain temporary tolerance). Other sensors need to have similar update rates for this application.

The robust horizontal filter is used as a backup for the map-based filter to obtain the estimated horizontal position while the vertical filter is used to obtain the estimated vertical position and joining them, the tracker has the estimated position. This proposed structure is inspired on the tracker explained in (Lopez, 2018), but with the necessary modifications to make it work with UAVs, since the behavior is different. The modifications it presents are that the tracking is now done in 3D, the dynamic models of the filters, which are explained in this section, are different from those used there and the values of the parameters of the filters are different as they are adapted for the proper operation of this system. Therefore, the robust filter is in charge of providing the surveillance output in situations where the map-based tracker cannot obtain the target estimated positions. The situations in which the map-based tracker does not obtain the estimated position are when either the target is not following a flight plan or it is not in a segment defined through a line.

The logic of the horizontal combined system is the following. When the horizontal trackin system receives a measurement, the measure is processed by both filters if there is a flight plan correlated. Otherwise, it is only processed by the robust filter. In case there is a correlated flight plan, once they have been processed, a selection is made between the estimated position of the robust filter and the map-based tracker. If successive measures have not been located on the map or no hypothesis of the map-based tracker is predominant then the robust filter is used. The vertical filter output is always used.

The Map-based tracker receives as input the 2D position measured in Cartesian coordinates after a transformation from latitude and longitude. It is assumed that the measures are perfectly associated beforehand target. The target will provide as output the identification of the segment of the flight plan where the target lies with highest probability and its position and velocity in Cartesian coordinates which will be transformed into latitude and longitude. A key factor of the tracker is the kinematic constraints imposed by the presence of the flight plan. The approach followed in this tracker is that each segment is represented with a straight line, which are represented by an origin and a destination.

The filter that is part of the Map-based tracker decouples the tracking in two components, one related to along-segment movement and another to track movement transversal to segment. In the along-segment tracking process it is assumed the target moves in parallel to the centre line of the segment. The transversal movement is estimated by a second part of the tracker which uses as input the difference between sensor measure and the projected measure. Therefore, the 2D position of the target might be represented by three values:

1. A segment identifier (s_{ID}), unambiguously defining the flight plan segment where the target position is being projected.
2. The distance along the segment, measured from the segment origin node (d_{long}).
3. The signed transversal position (d_{trans}). If it is positive, that means that the offset is to the right of movement direction, while negative values means the offset is to the left.

Figure 6.5: Global Map-based tracker filter structure

The overall recursive procedure for tracking a target following a map segment is as follows. The filter receives sensor measures which are corrected with the estimated transversal offset obtained from the previous measures. As there is residual error in

transversal offset estimator, and measures also suffer from noise, once the measure is offset-corrected, it is projected to the segment flight plan (using a maximum likelihood approach). This 1D projection is used as input measure of the IMM based longitudinal estimator, which tracks the traversed distance and its 1D velocity in this segment. In parallel to the longitudinal tracking, the lateral drift estimation is updated through a 1D Kalman filter whose input is the distance between the input measure and the segment. Finally, the updated lateral drift and the tracked longitudinal position are combined to obtain the final target position and speed. This routine can be summarised in Figure 6.5. The Longitudinal Tracking Filter tracks d_{long} and its associated velocity while the Transversal Filter tracks d_{trans} using the heading obtained by the longitudinal filter. The 2D measurement error is modelled as bias free, white and following a bivariate normal distribution with potentially non constant covariance R_k , where k is the number of the measure.

The filter needs s_{ID} to track targets in a map with segments, but there is a problem in junctions due to noisy measures that causes a difficult decision on which is the value of s_{ID} . So, the filter has a list of these values in the network with an assessment of the associated probability of each s_{ID} . Only, those segments whose probability is above a certain threshold P_{Hmin} are kept in this collection. Those segments would be in the vicinity of the last measures. Therefore using a Map-Based Multiple Hypothesis Filtering (MBMHF) approach, this list is updated along time with new segment hypotheses as the target arrives to junctions. Their relative probabilities are updated as new evidence is obtained based on the compability of measures with expected target movements. Finally, the hypothesis list is pruned as the target leaves the junction. So, a hypothesis $H(j)$ is composed of:

1. A segment identifier $s_{ID}(j)$ specifying the segment where the target is assumed to be.
2. Another segment identifier $s_{prevID}(j)$ specifying the segment where the target was before.
3. A 1D IMM based segment-adapted filter $F(j)$ tracking d_{long} , its velocity and acceleration.
4. The probability of the hypothesis $P(j)$.

In Figure 6.6 the general functional structure for the Longitudinal Tracking Filter is presented. This filter makes use of a data base containing all the information about the segments, used for locating and projecting the measures. As it was said, the filter keeps the list of hypotheses because it is following the MBMHF approach. When a new measure is received, the filter projects it in all the segments in the vicinity of the segments included in the hypotheses in the module Projection 2D to Longitudinal.

Figure 6.6: Longitudinal estimator diagram

Then, the projected measure is processed by the segment-adapted filter that best fits within the module Longitudinal IMM. The Hypotheses Management block manages all the hypotheses associated to the possible trajectories, creating new hypotheses as the target arrives to junctions, calculating the probability of each hypothesis and pruning the collection of hypotheses as the probability of a given hypothesis becomes lower than P_{Hmin} . In the calculation of the probability of each hypothesis, the kinematic state in all IMM filters in the hypotheses list, the map and the measure are used. In the Projection Longitudinal to 2D module, the IMM filters outputs are projected to 2D and then they are combined using its probabilities. This process is explained in detail in (Lopez, 2018) for a different application.

The longitudinal movement along a given segment is modelled using this state vector:

$$x_{long} = (d_{long}, v_{long}, a_{long})^T \quad (6.1)$$

In this vector, d_{long} represents the signed distance measured from the segment origin node, v_{long} is the signed longitudinal velocity at which the target moves away from the origin node and a_{long} represents the signed acceleration of the target in this direction. To track the movement, each segment is estimated by a longitudinal IMM filter (Blom and Bar-Shalom, 1988), which uses two Kalman filters defined by two dynamic motion models. One of these filters is in charge of segments without longitudinal acceleration, so it models a constant velocity movement with the state variables:

$$x_1 = (d_{long}, v_{long})^T \quad (6.2)$$

And the other filter assumes constant longitudinal acceleration, so the state variables is:

$$x_2 = (d_{long}, v_{long}, a_{long})^T \quad (6.3)$$

This filter has a higher process noise in order to be able to follow not fully modeled manoeuvres. So it is used a plant noise model as, where σ_{KA} is the comfort standard deviation and T is the period between measures:

$$Q_{KA} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_{KA}^2 T^2 \end{bmatrix} \quad (6.4)$$

The measurements feeding the IMM filter are those obtained, after lateral drift removal, from the projection processes. The filter tracks three states, but a 2D vector containing distance and velocity is provided as output and finally transformed to kinetic state coordinates. The same occurs with the covariance. An important parameter to be estimated for the MBMHF is the IMM filter likelihood, which is a measure of the compatibility of longitudinal projected measures with the prior IMM state. In this tracker, the likelihood of the constant speed filter is used as the IMM likelihood, which is calculated as:

$$\lambda_{IMM} = \frac{1}{2\pi|S|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} \cdot res^T \cdot S^{-1} \cdot res\right) \quad (6.5)$$

where res is the measurement residual and S is its covariance.

Figure 6.7: Transversal estimator diagram

The lateral drift tracking is simpler and its structure can be seen in Figure 6.7. It is remarkable that this estimation is only performed when there is no ambiguity on the segment being followed. The first process is projecting the 2D measure in the transversal-to-segment direction. This projected measure of d_{trans} is used as the input for a Kalman filter in charge of estimating the drift. Then, the filtered d_{trans} estimate is projected back to 2D.

To obtain the filtered drift estimation $d_{trans}(f)$, it is assumed drift dynamic behaviour can be modelled as a slowly changing signal. In this case, this is done through a one dimensional vector state Kalman filter, with an exponential time-correlation. So, this filter state vector is:

$$x_{trans} = (d_{trans}) \quad (6.6)$$

And this model can be expressed as follows, where $[k]$ is the time index.

$$d_{trans}[k] = \rho \cdot d_{trans}[k-1] + w[k] \quad (6.7)$$

In this model, ρ is a constant related to the decorrelation time of the process and measurement time and $w[k]$ is a stochastic discrete white process with normal distribution of zero mean and variance equals to $\sigma_{drift}^2(1 - \rho^2)$. This model also assumes that d_{trans} , in stationary state, has a normal distribution of zero mean and variance equals to σ_{drift}^2 . This estimation is obtained through a one state Kalman Filter following this prediction model and assuming measurement error $v[k]$ has zero mean and variance equal to σ_{trans}^2 . The measurement model then results:

$$d_{trans}(m)[k] = d_{trans}[k] + v[k] \quad (6.8)$$

So, once the offset is calculated, the sensor measures are corrected with it so the MBMHF tracking performance is not deteriorated.

Finally, the output kinematic state can be obtained through the additive process:

1. Position estimation is obtained through direct addition of 2D position estimates of both trackers.
2. Velocity estimation is Longitudinal tracker filtered estimate transformed to 2D because there is no velocity component in the lateral tracker.
3. The output covariance is the addition of the longitudinal tracker covariance and the lateral tracker covariance.

On the other hand, there is a Robust filter as backup to the MBMHF in a way that it obtains the estimation of the targets that are not following the map. This can happen for a variety of reasons, such as targets don't following a flight plan or targets going out of the segment. It is implemented through an Interacting Multiple Model (IMM) algorithm (Li and Bar-Shalom, 1993)(Cho et al., 2013), both in the horizontal and vertical dimensions.

The horizontal tracker has to be able to process measures from different sensors, so a centralized architecture has been implemented. So, each of the modes of an IMM is associated with a dynamic model and implemented by a Kalman Filter (KF). As the filter has to be robust and works in Cartesian coordinates, the horizontal IMM has two different dynamic models:

1. Constant ground speed with high accelerations.
2. Constant ground speed with low accelerations.

These models are in charge of representing the movement of an UAV at constant speed with different accelerations, the higher ones will be in the actions of accelerating, braking and turning while the lower one will be in the actions of constant movement. The dynamic of this motion model is defined by the following equations,

where x represents the x position in Cartesian coordinates of the measure, y is the y position in Cartesian coordinates of the measure, v_x represents the signed x velocity at which the target moves, v_y is the signed y velocity at which the target moves, a_x represents the signed acceleration of the target in the x direction, a_y is the signed acceleration of the target in y direction and T is the period between measures:

$$\frac{dv_x}{dt} = a_x \quad (6.9)$$

$$\frac{dv_y}{dt} = a_y \quad (6.10)$$

$$\frac{dx}{dt} = v_x \quad (6.11)$$

$$\frac{dy}{dt} = v_y \quad (6.12)$$

The vector state has four components: both Cartesian coordinates of the position and the two components of the ground speed.

$$x_f = [x \quad y \quad v_x \quad v_y]^T \quad (6.13)$$

The transition function is defined as follows:

$$f(\vec{x}_{k+1}) = \begin{pmatrix} x_k + T \cdot v_x + \frac{T^2}{2} \cdot a_x \\ y_k + T \cdot v_y + \frac{T^2}{2} \cdot a_y \\ v_x \\ v_y \end{pmatrix} \quad (6.14)$$

As we have an EKF, its associated Jacobian matrix is needed:

$$F = Jacob(f) = \begin{pmatrix} 1 & 0 & T & 0 \\ 0 & 1 & T & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (6.15)$$

The input measures are in Cartesian coordinates and if they have speed information, the measurement matrix takes the following form (H_1):

$$H_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (6.16)$$

And if they don't have speed information (H_2):

$$H_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad (6.17)$$

This filter has a process noise in order to be able to follow non fully modelled manoeuvres. Therefore, the plant noise model is modelled by the following equation, where σ_{AH} is the plant noise standard deviation:

$$Q = \begin{pmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{pmatrix} \cdot \sigma_{AH}^2 \cdot \begin{pmatrix} \frac{T^2}{2} & 0 & T & 0 \\ 0 & \frac{T^2}{2} & 0 & T \end{pmatrix} \quad (6.18)$$

The transition probabilities are those used to premix the state estimators for each of the motion models after the previous update. This take into account the possibility that the UAV may have changed the motion mode in the time interval between measures. These probabilities evolution is dependent on the time that has passed between measures. Low transition probabilities will make the system to take longer to detect manoeuvres and high transition probabilities deteriorates the non-maneuvering performance because the filters assume manoeuvres between each measure.

In the literature, a solution is the one defined in (Kastella and Biscuso, 1995). There, the transition probabilities are defined for an IMM with only two modes, as the horizontal filter. So, the transition matrix is:

$$T[k] = \begin{pmatrix} 1 - \frac{T}{T_{ba}} & \frac{T}{T_{ba}} \\ \frac{T}{T_{ba}} & 1 - \frac{T}{T_{ba}} \end{pmatrix} \quad (6.19)$$

where T_{ba} represent the time between acceleration.

The robust vertical filter processes height and vertical speed. The vertical tracking filter is based on a centralized IMM filter. The IMM filter has 3 modes: constant vertical speed, vertical acceleration and stop.

The vertical speed mode is in charge of representing the height change at constant rate, so the parameters to be estimated are the height (h) and the vertical speed (v_v). The vector state is:

$$x_f = [h \quad v_v]^T \quad (6.20)$$

The transition matrix is defined following a constant speed movement:

$$F = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix} \quad (6.21)$$

The measure matrix could be depending on whether there is speed information (H_1) or not (H_2):

$$H_1 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad (6.22)$$

$$H_2 = \begin{pmatrix} 1 & 0 \end{pmatrix} \quad (6.23)$$

And the plant noise matrix is:

$$Q = 0_{2 \times 2} \quad (6.24)$$

The vertical acceleration mode is in charge of representing a vertically accelerating/decelerating UAV. The vector state has three components: height, vertical speed and vertical acceleration.

$$x_f = [h \quad v_v \quad a_{vert}]^T \quad (6.25)$$

The transition matrix is defined as follows, where T is the period between measures:

$$F = \begin{pmatrix} 1 & T & 0.5 \cdot T^2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{pmatrix} \quad (6.26)$$

The measure matrix could be depending on whether there is speed information (H_1) or not (H_2):

$$H_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \quad (6.27)$$

$$H_2 = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \quad (6.28)$$

And the plant noise matrix is, where σ_{AH} is the plant noise standard deviation:

$$Q = \begin{pmatrix} 0.5T^2 \\ T \end{pmatrix} \cdot \sigma_{AH}^2 \cdot (0.5T^2 \quad T) \quad (6.29)$$

Finally, the vertical stop mode is responsible for representing an UAV in flight at constant height, so the vector state only have one component, the height.

$$x_f = [h]^T \quad (6.30)$$

$$F = (1) \quad (6.31)$$

As this mode does not estimate vertical speed, the measure matrix is:

$$H = (1) \quad (6.32)$$

And the plant noise matrix is:

$$Q = (0) \quad (6.33)$$

The transition probabilities for the vertical IMM filter are deduced using the same argument as for the horizontal filter, so the transition matrix is:

$$T[k] = \begin{pmatrix} 1 - \frac{T}{T_{bva}} & \frac{T}{T_{bva}} & 0 \\ \frac{T}{T_{va}} & 1 - \frac{T}{T_{va}} - \frac{T}{T_{ivs}} & \frac{T}{T_{ivs}} \\ 0 & \frac{T}{T_{vs}} & 1 - \frac{T}{T_{vs}} \end{pmatrix} \quad (6.34)$$

where T_{bva} represents the time between vertical accelerations, T_{va} is the average time of vertical acceleration, T_{vs} is the average vertical stop time and T_{ivs} is the average time it takes for the target to stop at the current vertical speed. The zeros of the matrix represent the changes that are not allowed.

6.5.2 Surveillance systems integrator

Apart from the telemetry sent by the drones, it is necessary to incorporate other surveillance systems in the service that send the necessary information to track and monitor all the drones.

Figure 6.8: Surveillance Systems Integrator

All these sensor systems will send to the Tracking and Monitoring Service the information that they capture for each drone that they are capable of detecting.

Once in the service, it will be necessary to convert this type of data to the one used internally in order to operate properly and thus track and monitor those drones that do not send telemetry.

But not only are these other surveillance systems integrated in this module here explained. The emergence of drones led to the need for the creation of a control service and management of drone air traffic. So the system is connected to the UTM system, which is in charge of sending the drones of other operators and other fleets that are operating in the area of interest. Also the UTM system would send the ATC tracks of the system of control and management of air traffic for large aircraft (ATM/ATC), because they would be interconnected. The tracking information is in Asterix 62 format (Eurocontrol, 2019), so it must be translated to the data used internally in the Tracking and Monitoring Service. In this way, the service is able to track and monitoring the aircrafts, not only the drones.

Figure 6.8 shows the process that is carried out in the Surveillance Systems Integrator functionality. On the one hand, it is received the telemetry from drones that is entered in the Data fusion module, on the other hand it is received the tracks of the sensors (one for each sensor that sends information) and finally ATC and UTM tracks are received in Asterix 62 format and converted into tracks. Once all these object arrives to the data fusion module, an association process is carried out. It may be that a drone that sends telemetry is also tracked by other sensors, so its information must be merged so that the service remains as accurate as possible. To do this, different processes are done to select the best one. On the other hand, the drones that do not send telemetry can also be captured by different sensors so another association process must be carried out so that in the end only one track is available for each drone.

6.5.3 Break-in detector

The regulation of drones is different for different scenarios, but in general it is clear about the conditions under which a drone can fly. For instance, in the Spanish regulation (BOE, 2017), the basic rules state flights must be performed at less than 120 meters high, never over populated areas or special buildings and always maintaining eye contact. But there are geographical areas where a drone cannot be flown.

Flying drones is not allowed in all airspace. There are areas delimited by which it can not be flown as it could be airports surrounding areas (the Spanish regulation forbids flights around airports in a circle of 8 kilometers radius). These zones are stipulated by the relevant organizations, as the government of a country, the government of a city, a company, an individual ... For example, the Spanish regulation is ruled by the Ministry of Public Works, and sometimes by the Ministry of Defense. Even if a pilot has the approval of the mayor to fly over a town hall or the neighbor

to fly over a house, the final permit depends on the air safety agency, or AESA (AESA, 2019).

The Spanish airspace is within the European airspace and, in turn, it is divided into three smaller regions, called FIR (Flight Information Region): Madrid, Barcelona and the Canary Islands. These regions are also divided into several volumes, which are stacked on top of each other. According to current Spanish regulations, RPAS/drones are prohibited from entering areas of controlled airspace and limited airspace zones.

Under common names internationally, there are several types of controlled airspace. These can be used by aircraft under certain circumstances:

- **CTA or control area.** It serves to protect aircraft from the moment they take off until they enter a route or an airway. CTAs are formed by a volume of controlled airspace in the vicinity of airports.
- **TMA or terminal area of maneuvers.** It is the same as the CTA, but for airports with more traffic.
- **CTR or control area.** One of the best known among drone pilots, it forms a bubble around each airport to protect incoming and outgoing traffic.
- **ATZ or aerodrome transit area.** It covers the movement of aircraft in a volume of about 8 kilometers in radius and less than one kilometer in height around airports.
- **AWY or airway of air route.** They are defined corridors that connect two geographical points to a certain altitude.

In addition to controlled areas, drones cannot fly over limited airspace, as do most aircraft. According to Spanish legislation, four types are recognized:

- **Prohibited.** In the aeronautical charts it appears marked as LEP. To fly over it, express permission from the Ministry of Defense is required. LEP are, for example, nuclear power plants or real estate properties.
- **Dangerous.** These are identified as LED. Some aircraft can fly over these areas as long as they make sure that no dangerous activity is taking place. They are, for example, military practice zones.
- **Restricted.** These areas are indicated in the aeronautical charts as LER and can only be flown over by state aircraft in case of emergency. As examples are the corridor of Algeciras or national parks such as Picos de Europa.

- **Areas with sensitive fauna.** These are protected areas that are identified with the letter F.

It is for all this that it is necessary to have a break-in detector in restricted areas to let the users know that this drone is committing an infraction that must be checked, because it may be the case that this drone has the necessary permissions to make that incursion.

For this purpose, the location information of all these areas is available in first approximation in the No Drone Zone Model explained in Subsection 4.3.4.1. So the process to know if a drone is inside one of these areas is the following one. The position of the drone predicted in the tracker functionality is sent to the No Drone Zone Model. This functionality answers to the Break-in detector functionality if the drone is or not inside any of these areas. In the case the drone is not estimated to be inside the area, this detector does not send any alert, but it indicates the minimum distance to any one of these restricted/protected areas. If, on the contrary, this module detects that the drone is inside one of these restricted or forbidden areas, the detector sends an alert to all involved users saying that it is entering an area where it can not be flown and that the appropriate measures will be taken in the case the drone does not have the necessary permits, so leave that area as soon as possible. This information is written down in the track, specifically in the Geofence violation alert field, where the identifier of the involved area is include and the distances to that geofences.

6.5.4 Deviation detector

A good drone flight plans monitoring system must take into account the original flight plan that is intended to fly. It is not enough to know at all times the position of a drone, but it is necessary to know if the established flight plan is followed. The reason of that is because of the drone is diverting more that it is possible, the drone can have different types of problems and becomes a threat not only for the rest of the drones in the airspace, but also for people.

For these reasons, there must be a comparison between the generated flight plan and the monitored target state according to the tracking module. But not all drones that fly and send telemetry follow a flight plan, some drones can make free flights. Therefore, these tracks are discarded from the deviation detector functionality because they cannot be compared with a flight plan, so conformance is not possible. Only those drones that follow a flight plan that is stored in the database can be compared.

At each time a position of the drone is received, it is checked if the drone follows a flight plan. If the answer is affirmative, then the position of the drone is compared with the position that the drone should be at the planned time interval, always with

a certain margin of error, to take into account both flight operation uncertainty and tracking errors.

As it was said in Section 5.3, the flight plan can be a time-ordered combination of two different specifications, described more in depth in (Besada et al., 2019):

- Linear segments to be followed by the aircraft, with associated height and time constraints.
- Volumes, where the flight must be confined for a given time interval (duration), also with associated height constraints.

The process of calculating deviations starts with the check if the drone is following a flight plan. If it is the case, the aircraft class is obtained so the horizontal deviation limit is set. Then, with the drone position and the flight plan, the distances between that position and each of the segment of the flight plan is calculated and compared to a threshold, although there is a difference between linear segments and volumes. For linear segments, the process consists of calculating the distances between the drone position and the segment boundary waypoints. If any of the two distances is greater than the segment distance, then the track is outside the segment, so the horizontal deviation is calculated as the orthogonal distance of the track with respect to the beginning or end of the segment. Otherwise, as the two distances are less than the segment distance, the track is in that segment and the horizontal deviation is calculated with the point on the line. For volumes, the detector checks the position is inside that volume. If the result is negative, the detector calculates the distances between the track position and each of the sides of the volume. These distances are calculated as the linear segments and are compared among them. The shortest distance between the track and the volume is obtained. Once all the distances between the track position and each of the segments have been obtained, the detector checks any of these distances is less than a threshold. If none of them meet this check, the smallest distance among all distances will be the horizontal deviation distance and horizontal alarm will be sent.

Then, the vertical deviation is calculated. So, as for the horizontal deviation, the calculation is made with the position of the drone and the information of the flight plan. The calculation process can be observed in Figure 6.9, where the height profiles as two linear functions are represented. The first step of the process is to know the segment that the drone is flying through, which is made in the horizontal deviation calculation. Once the segment is known, it is necessary to obtain the distance the drone is within the segment from the segment's starting point. Then, the minimum and maximum height profiles of the flight plan that the drone follows are drawn. The second step is to obtain the point on the height profiles where the drone is located. Once the position of the drone in the height profile is obtained,

the third and last step is to obtain the minimum height and the maximum height that the drone can have in the currently location and which has been established in the flight plan. In the case where the drone is within a volume, the process is the same. The volume in which the drone is located is calculated and the minimum and maximum heights at which the drone can fly are set from the volume height limits, which are uniform throughout the volume. Once the minimum and maximum height limits at which the drone can fly are known, they are compared with the height at which it actually flies. If it is between these limits, there is no problem, the drone is following the flight plan and there is no deviation. If, on the other hand, the height at which the drone flies is less than the minimum height at which it should fly at that point or greater than the maximum height at which it should fly at that point, there is a vertical deviation since it does not comply with the established flight plan. This deviation is calculated as the difference between the current height and the minimum/maximum height. In this case, the height difference is returned as the vertical deviation and a vertical alarm will be sent.

Figure 6.9: Example of vertical deviation calculation

Finally, the temporal deviation is calculated using a similar process with the minimum and maximum expected times of arrival (ETA) to the waypoints and entry/exit times at the volumes. Instead of the height profile, this calculation uses the time profile.

Once all the calculations are done, the lateral, vertical and temporal deviations are provided as track outputs in the conformance monitoring state and alarms related with these deviations are also included in the track.

6.5.5 Obstacles detector

On the one hand, when planning a flight plan, it is done knowing the height of the ground to fly at a distance from it and thus avoid collisions, as it was explained in

Section 4.3.3. In addition, a guard margin is added to take into account both flight technical error and altimeter errors. In any case, this module compares the height of the flight with the terrain calling the ground service, taking into account if the drone is already on the landing segment.

On the other hand, the obstacles detector is completed with the process to calculate the nearby traffic to avoid collisions with other aircrafts. This calculation is made for all the tracks that the module has, so the functionality compares the position of each drone with the rest of positions of aircrafts. Depending on the type of the aircraft is, the limits of the distance without conflict are different. When all drones have been compared, those with conflicting drones and alerts are added to their respective tracks in the tactical conflict field.

6.6 Results of tracking and monitoring service

In this section, a full example of application of the previously described techniques and some performance results are provided. The test will consist of creating a mission and simulating and analyzing the behavior of the different modules that make up the Tracking and Monitoring Service. The results of the tracker will be shown in the Section 6.6.1, while the different detectors are validated in Sections 6.6.2 for the Break-in detector, 6.6.3 for the Deviation detector and 6.6.4 for the Obstacles detector.

6.6.1 Tracker validation

In order to demonstrate the performance of the tracker system with data, a simulated scenario located in Montegancedo Campus, of Universidad Politécnica de Madrid, Madrid is analysed. The filtered data come from a simulation of a UAV performing the mission created in the Mission Planning Service. To these filtered data the same error has been added for the three filters, a Gaussian error with zero mean and standard deviation equal to 5 meters. The sensor available is a telemetry system. The flown mission in 3D and the scenario can be seen in Figure 6.10, while the trajectory in 2D is shown in Figure 6.11. The shortest segment corresponds with a side of a volume, so in this segment the tracker will work with a volume instead of a segment.

The trajectory that has been simulated can be described as:

- A drone that takes off to a height of 20 meters above the ground.
- It turns on itself to the direction of the first segment and accelerates until it reaches the cruising speed of 15 m/s. When it reaches the end point of the segment, the drone begins to decelerate.

- Step 1 is repeated 4 more times because there are 4 segments as the first one (same cruising speed) but different directions and lengths.
- The drone reaches the shortest segment that corresponds to one side of a volume. In a real flight, the drone would go into manual mode and the pilot would fly freely within the limits of the volume. In the simulation, the drone will go from the entry point to the exit point at the speed calculated by the predictor taking into account the distance between the two points and the minimum time the drone can be within the volume, in this case 25 seconds.
- It turns on itself to the direction of the next segment and accelerates until it reaches the cruising speed of 15 m/s.
- This process is repeated in the last segment but when the drone reaches the last point, the drone starts to land.

Figure 6.10: Detail in 3D of the mission performed

To check the performance of the system, a comparison of the results from the different filters is made. In this comparison, the ideal trajectory, calculated in the Mission and Flight Planning Service is used as the measures of the performance. Specifically, the following performance measures are used:

1. To assess position accuracy, the difference between the measures and the predicted position is measured. This difference has two components. One of this components is related to measurement noise, which is equal for all filters used

Figure 6.11: Detail in 2D of the mission performed and the trajectories obtained for the different trackers

in the performance measure. The other component is related to predicted filter error. It can be concluded that smaller difference means better predicted filter error and a better filter with respect to position too.

2. To assess ground speed accuracy, the difference between the filtered ground speed values and the ideal values is also measured. In the trajectory observed in Figure 6.11, the ground speed changes smoothly as the drone moves, but any filter will produce a somehow noisy ground speed and may also have rebound near acceleration manoeuvres. To sum up, a reduced difference between filtered ground speed values means better filter with respect to velocity.
3. To assess heading accuracy, the difference between the filtered heading and the ideal heading of the segment being followed by the target is calculated.
4. To assess height accuracy, the difference between the filtered height and the ideal height of each point is calculated. The smaller residual means better predicted filter error and a better filter with respect to height too.

Before obtaining the distance error, first a comparison of the x-y positions that the drone flies in the real path with the positions that each of the trackers calculate is made, so it can be analyzed the behavior that the distance error will have. In Figure 6.12a the positions of the real trajectory and the ones obtained from the map+robust filter are very similar so the distance error will be small except for some points where there will be small peaks. Practically the same behaviour happens with

(a) Comparison of x-y positions from the real trajectory and Map + Robust filter

(b) Comparison of x-y positions from the real trajectory and Robust filter

(c) Comparison of x-y positions from the real trajectory and Map-based filter

Figure 6.12: x-y positions comparison

the positions obtained from the real trajectory and from the robust filter as can be seen in Figure 6.12b. On the other hand, in Figure 6.12c, where the positions of the real trajectory and the map-based filter are represented, it can be seen big peaks that happen when the drone reaches the end of the segments and has to turn, since in the turns the map-based filter behaves badly. It can be also seen a part around the second 140 where there is not values because it is a volume and the map-based filter does not return values in volumes. With all this, the distance error between these positions will be big.

Figure 6.13 shows the absolute value of the residual of the position along time, where it can be observed that the filter of the system has reduced error peaks than other algorithms have and it has the smallest average value. In Table 6.1 the average value of this magnitude and its 95th percentile along the trajectory can be seen. For the map-based filter, the volume positions does not count to calculate the average

value and its 95th percentile. In this result the quality of the composition of the map + robust filter is better than the robust filter and the map-based filter independently.

Figure 6.13: Distance error for the different trackers

Filter	Average Residual [m]	95th Percentile Residual [m]
Map and Robust	2.11	6.64
Robust	3.09	14.36
Map-based	4.48	16.68

Table 6.1: Absolute residual of the position

Before calculating the groundspeed error, a comparison between the groundspeeds from the real trajectory and the values obtaining from the three trackers can be observed in Figure 6.14. As it can be seen, they all behave similarly, although the robust filter is somewhat better. All of them have errors at the beginning of the segments, in the accelerations, and in the volume area because it is a manually flown area even if a simulation has been made. In the case of the map-based filter, there is not values of the tracker in the volume area. So all three trackers will have similar groundspeed errors. These will have a low average and some error peaks corresponding to the drone accelerations at the beginning of the segments.

Figure 6.15 shows the absolute value of the residual of the ground speed along time. It can be observed that the map + robust filter and the robust filter have similar values while the map-based filter presents bigger error values. Then, Table 6.2 shows the average value of this magnitude and its 95th percentile along the

(a) Comparison of groundspeed from the real trajectory and Map + Robust filter

(b) Comparison of groundspeed from the real trajectory and Robust filter

(c) Comparison of groundspeed from the real trajectory and Map-based filter

Figure 6.14: Groundspeed comparison

trajectory. For the map-based filter, the volume area does not count to calculate the average value and its 95th percentile. In this result the quality of the map + robust filter and the robust filter show to be better than the map-based filter.

Figure 6.15: Ground speed error for the different trackers

Filter	Average GS diff [m/s]	95th Percentile GS diff [m/s]
Map and Robust	0.98	4.87
Robust	1.18	5.06
Map-based	1.41	6.28

Table 6.2: Absolute residual of the ground speed

Before calculating the heading error, a comparison between the headings from the real trajectory and the values obtaining from the three trackers can be observed in Figure 6.16. As it can be seen, they all behave similarly. All of them have errors at the beginning of the segments, in the accelerations, while the map + robust filter and the robust filter have problems in the volume area because it is a manually flown area even if a simulation has been made. In the case of the map-based filter, there is not values of the tracker in the volume area. So all three trackers will have similar heading errors. These will have a low average and some error peaks corresponding to the drone accelerations at the beginning of the segments and bigger errors in the volume area.

The results showing the quality of the heading estimation are depicted and analysed. In Figure 6.17, the heading error is depicted for all competing filters. It can be observed that none of the filters behave well against heading in the change of

(a) Comparison of heading from the real trajectory and Map + Robust filter

(b) Comparison of heading from the real trajectory and Robust filter

(c) Comparison of heading from the real trajectory and Map-based filter

Figure 6.16: Heading comparison

segments when the drone has to turn. The rest of the time the error they present is relatively small, although at the time of the landing a big error is produced in the three trackers. This is clearly seen in Table 6.3 where it is summarize along-trajectory average value of the absolute heading error and its 95th percentile. For the map-based filter, the volume area does not count to calculate the average value and its 95th percentile.

Figure 6.17: Heading error for the different trackers

Filter	Average Abs Error [°]	95th Percentile Abs Error [°]
Map and Robust	16.51	104.73
Robust	14.67	78.94
Map-based	21.27	121.13

Table 6.3: Heading error

After analyzing the performance of the horizontal filter, it is time to analyze the performance of the vertical filter. In Figure 6.18 the value of the residual of the height along time is showed and it can be seen that the map + robust filter, the robust filter and the map-based filter have the same performance because the vertical filter is equal to the three trackers. Table 6.4 shows the average value of this magnitude and its 95th percentile along the trajectory.

It can be concluded that the proposed system has a better performance than the state of the art algorithms. It should be pointed that in curves the system shows much lower error peaks that the rest of algorithms. It also obtains that the difference between the robust filter and the map and robust filter is not big, but adding the

Figure 6.18: Height error for the different tracker

Filter	Average Residual [m]	95th Percentile Residual [m]
Map and Robust	2.73	5.89
Robust	2.73	5.89
Map-based	2.73	5.89

Table 6.4: Absolute residual of the height

map information to the robust filter makes it slightly better because it has that information, so it gets better results. The map-based filter quickly degenerates in the turns and intersections due to lack of projection in areas with these paths. So, when both filters are working in parallel (map-based filter and robust filter), the quality of the system is noticeably better than other solutions, since the junction peaks are eliminated using the robust filter as output in these situations and the map-based filter performs better in the rest of situations.

6.6.2 Break-in detector validation

In this section the operation of the Break-in detector will be analysed. For this purpose, a flight has been simulated in the vicinity of geofences. Depending on the state of the geofence and the drone being flown, as well as the limits of the geofence and the height at which the drone flies, the monitoring service will set alerts or not.

Figure 6.19a shows the flight simulated. It can be observed that the flight is located in Campus de Montegancedo, Universidad Politécnica de Madrid, Madrid and this area presents three different geofences. The circular geofence is expired, so any flight inside it won't raise any alarms. The 5-sided polygon shaped geofence has a height margins where is not allowed to fly between 0 and 40 meters and is active. Finally, the geofence with the shape of a rectangle is active and has height

margins at which flight is not allowed from 20 meters to 150 meters. As it can be observed not all geofences are crossed because when calculating the flight plan, the Flight Planning Service would not allow a flight that crosses a geofence as shown in Chapter 4. For this reason, this flight plan is made first and then in the simulator it is modified to cross geofences. Thus, the mission of Figure 6.19b is obtained.

Figure 6.19: Mission to validate the Break-In Detector

In Figure 6.20 it can be observed that the Monitoring Service does not give an alarm of break-in because as it was said, this geofence is not active. This is to check that the break-in detector works correctly when the geofence has expired.

Figure 6.20: Detail of non alarm in a not active geofence

In Figure 6.21 it can be observed that the Monitoring Service does not give an alarm of break-in because as it was said, this geofence has a heights margins where the flight is forbidden that don't include the height at which the drone is flying.

With this geofence, it has been checked that the break-in detector works correctly when the drone flies at a height that is not included in the height limits of a geofence.

Figure 6.21: Detail of non alarm in a geofence with heights limits that not include the flight height

Finally, the Monitoring Service shows an alarm when the drone breaks into the last geofence because it is flying at an altitude that is not allowed to fly in that area. This alarm can be seen in Figure 6.22. With this behaviour the operation of the Break-in detector is validated in all possible cases of geofences.

Figure 6.22: Detail of an alarm in an active geofence

To validate the Break-in detector, a simple scenario has been set up in which a flight plan crosses an active geofence and generates a conflict. This scenario can be seen in Figure 6.23a. The aim is to calculate the time it takes for the invasion alarm to appear, as well as the time it takes for the same alarm to disappear compared to the actual time it takes for this alarm to appear and disappear on an error-free flight. These values have been calculated and are 22.875s for the appearance and 32.128s for the disappearance. This scenario has been validated with a MonteCarlo simulation of 500 iterations and the result obtained can be seen in Figure 6.23b.

It can be seen that most of the iterations the alarm has appeared and disappeared in similar moments of time, although in some iterations the appearance has been advanced or delayed in a short time. The average time it takes for the alarm to appear in this scenario is 1.309s and it can be observed in Figure 6.23c and the average time it takes for the alarm to disappear in this scenario is 1.692s and it can be observed in Figure 6.23d.

Figure 6.23: Break-In Detector validation

Another validation to the Break-in detector can be observed in Figure 6.24. In this validation the same mission is carried out, but instead of taking out the moments of time in which the alarm appears and disappears, the distances at which the break-in alarm appears and disappears are obtained. First the distances on the flight plan from the take-off point to the geofence entry point and to the geofence exit point has been calculated on the mission, and they can be seen in Figure 6.24a. Then, through a MonetCarlo simulation of 500 iterations, the distances from the real entry and exit points of the geofence to the points where these alarms appear

and disappear have been obtained. The result of the simulation can be seen in Figure 6.24b. It can be seen that all the alarms appear and disappear at more or less the same distance, except for some specific cases. The average distance it takes for the alarm to appear in this scenario is $2.089m$ and the average distance it takes for the alarm to disappear in this scenario is $2.705m$. These values can be observed in Figure 6.24c and Figure 6.24d. There is not a big difference between the real values at which the alarm should appear and disappear with those that really do.

Figure 6.24: Second Break-In Detector validation

6.6.3 Deviation detector validation

In this section the performance of the Deviation detector will be analysed. For this purpose, it has been simulated a flight that can be observed in Figure 6.25a. As

this is a simulated flight, no matter how much error there is in the simulation, as demonstrated in section 6.6.1, the values of the conformance margins will not be reached to trigger the various alarms. So, what it is needed to do is to modify the flight plan to be simulated a little bit so that when it is compared with the stored flight plan, there is a difference and the alarms will go off. The modified flight plan in the simulator can be seen in Figure 6.25b.

(a) Detail of the created mission for the Deviation detector validation

(b) Detail of the simulated mission for the Deviation detector validation

Figure 6.25: Mission to validate the Deviation Detector

The first alarm to be validated is the lateral alarm. This alarm is obtained when the drone occupies a position whose difference with the point it should occupy exceeds a certain value. Figure 6.26a shows that since the drone is quite far from the original flight plan, the horizontal non-conformity alarm is shown. On the other hand, in Figure 6.26b it can be seen that as the drone is outside the polygon, the horizontal non-conformity alarm is shown.

The next alarm to be validated is the vertical alarm. It is obtained when the drone is at a height whose difference with the height it should occupy exceeds a certain value in absolute terms. Figure 6.27a and Figure 6.27b show that when the drone is at a much higher height than it should be, the vertical non-conformity alarm is shown.

The last alarm to be validated is the time alarm, which is triggered when the drone is being flown in an instant of time out of the time it should be flying at that point, or when the drone is longer than it should be within a polygon. In Figure 6.28a and Figure 6.28b it can be seen that when the drone is flying in a time after the time in which the flight plan was valid, the time non-conformity alarm is shown.

(a) Lateral alarm in a segment

(b) Lateral alarm in a polygon

Figure 6.26: Lateral alarms

(a) Vertical alarm in a segment

(b) Vertical alarm in a polygon

Figure 6.27: Vertical alarms

In the case where a drone is not following a flight plan, the Deviation Detector displays a flight plan non-tracking alarm as shown in Figure 6.29a. In all other cases, when the drone is following a flight plan and the flight plan is within the Deviation Detector's compliance range, the Deviation Detector does not release an alarm as shown in Figure 6.29b.

To validate the Deviation detector, a number of scenarios have been analysed. In the case of analyzing the lateral alarm, a simple scenario has been set up from the scenario used to validate the Break-In detector (Figure 6.23a). This was the active flight plan, but in the simulation the waypoint 2 was moved out of the geofence so that the lateral alarm would appear, as the drone is not following the flight plan. The simulated scenario can be observed in Figure 6.30a. The aim is to calculate

Figure 6.28: Time alarms

Figure 6.29: Other alarms in Deviation detector

the time it takes for the lateral alarm to appear, as well as the time it takes for the same alarm to disappear comparing them to the actual moments of time when the drone stops following the flight plan and when it follows the flight plan again. These moments occur at 23.964s and 30.846s respectively. This scenario has been validated with a MonteCarlo simulation of 500 iterations and the result obtained can be seen in Figure 6.30b. It can be seen that most of the iterations the alarm has appeared and disappeared in similar moments of time, although in some iterations the appearance has been advanced or delayed in a short time. The average time it takes for the alarm to appear in this scenario is 1.729s, which is reflected in the Figure 6.30c and the average time it takes for the alarm to disappear in this scenario

is 1.751s that it can be seen in Figure 6.30d. It can also see the appearance of a false alarm that occurs in some of the simulations performed. This false alarm may be due to the noise introduced in the drone positions that may cause the tracker to sometimes pass a sample of the landing point, as it can be seen that the false alarm occurs at the time of the start of the landing and has a duration of one sample. The probability of occurrence of false alarm in this scenario is 9% and the average time of duration of the false alarm is 0.858s.

(a) Conflict scenario for lateral alarm

(b) Lateral alarms in conflict scenario

(c) Lateral alarms detection in conflict scenario

(d) Lateral alarms disappearance in conflict scenario

Figure 6.30: Lateral alarm validation from Deviation Detector

In the case of analyzing the vertical alarm, the simple scenario used to validate the Break-In detector is used. This was the active flight plan, but in the simulation the waypoint 2 was displaced in height, instead of a range of heights between 20m and 40m, it was moved to a range of heights between 60m and 80m. Then the vertical alarm would appear, as the drone is not following the flight plan. The simulated scenario can be observed in Figure 6.31a. The aim is to calculate the time it takes

for the vertical alarm to appear, as well as the time it takes for the same alarm to disappear comparing them to the actual moments of time when the drone stops following the flight plan and when it follows the flight plan again. These moments occur at $24.537s$ and $31.214s$ respectively. This scenario has been validated with a MonteCarlo simulation of 500 iterations and the result obtained can be seen in Figure 6.31b. It can be seen that most of the iterations the alarm has appeared and disappeared in similar moments of time, although in some iterations the appearance has been advanced or delayed in a short time. The average time it takes for the alarm to appear in this scenario is $0.603s$ and it can be observed in Figure 6.31c and the average time it takes for the alarm to disappear in this scenario is $0.446s$ which is reflected in Figure 6.31d. It can also see the appearance of a false alarm that occurs in some of the simulations performed. This false alarm is due to the takeoff, which due to the noise introduced may exceed the maximum takeoff height or may not reach the minimum takeoff height due to the acceleration of the drone to take off. The probability of occurrence of false alarm in this scenario is 16.2% and the average time of duration of the false alarm is $0.663s$.

Figure 6.31: Vertical alarm validation from Deviation Detector

Finally, in the case of analyzing the time alarm, the simple scenario used to validate the Break-In detector is used and it can be observed in Figure 6.23a. This was the simulated flight plan, but in the original a change in the time profile was made. The maximum time of the waypoint 2 was equal to the value of the minimum time which results in a time profile that can be seen in Figure 6.32a. Then the time alarm would appear in the vicinity of waypoint 2, as the drone is not following the flight plan. The aim is to calculate the time it takes for the time alarm to appear, as well as the time it takes for the same alarm to disappear comparing them to the actual moments of time when the drone stops following the flight plan and when it follows the flight plan again. These moments occur at 24.382s and 32.869s respectively. This scenario has been validated with a MonteCarlo simulation of 500 iterations and the result obtained can be seen in Figure 6.32b. It can be seen that most of the iterations the alarm has appeared and disappeared in similar moments

Figure 6.32: Time alarm validation from Deviation Detector

of time, although in some iterations the appearance has been advanced or delayed in a short time. The average time it takes for the alarm to appear in this scenario is $0.832s$ and the average time it takes for the alarm to disappear in this scenario is $0.691s$ and they can be observed in Figure 6.32c and Figure 6.32d.

6.6.4 Obstacles detector validation

To validate the Obstacles detector, two flight plans have been created, from different operators, whose area of operation is very close, as can be seen in Figure 6.33. As these flight plans are so close, when they are within a certain distance, the Obstacles detector detects an obstacle, in this case, another drone, and shows it to its operator/pilot on the screen in another colour so that they are aware of the presence of a danger. This detector does the same for all operators and pilots as

can be seen in Figure 6.34. This is how the Obstacles detector has been validated.

Figure 6.33: Missions to validate the Obstacles Detector

Figure 6.34: Obstacle alarm

To validate the Obstacles detector, a simple scenario with two drones has been set up. The scenario consists of two missions more than 1 km apart at the take-off point, starting at the same time and approaching each other. The two flight plans can be seen in figurename 6.35a, where the distance separating each of the points of the flight plans is detailed. The aim is to calculate the time it takes for the obstacle alarm to appear, as well as the time it takes for the same alarm to disappear comparing them with the real times at the drones are close enough and then they are no longer close. These moments of time are 16.578s and 91.709s respectively. This scenario has been validated with a MonteCarlo simulation of 500 iterations and the result obtained can be seen in Figure 6.35b. It can be seen that most of the iterations the alarm has appeared and disappeared in similar moments of time,

Figure 6.35: Obstacles Detector validation

although in some iterations the appearance has been advanced or delayed in a short time. The average time it takes for the alarm to appear in this scenario is 2.298s and it can be observed in Figure 6.35c and the average time it takes for the alarm to disappear in this scenario is 1.389s and it can be seen in Figure 6.35d.

6.7 Conclusions

In this chapter the Tracking and Monitoring Service has been defined.

It has been concluded that the Tracking and Monitoring Service adequately fulfils its purpose. The tracking part of the system has a better performance than the typical state of the art algorithms. It also concluded that the difference between the robust filter and the map and robust filter is not big, but adding the map

information to the robust filter makes it better because it has that information, so it gets better results. The monitoring functionality has also been validated.

Chapter 7

Conclusions

In this Ph.D. thesis, a system able to perform some drone management functionalities has been designed and analysed. To be more specific, the system focuses on planning a drone mission and tracking and monitoring it.

This thesis has explored some of the applications of UAVs and the systems to control and manage them. Several drone use cases have been analyzed and it has been found that all of them can be specified through a mission to fulfill the required objective. To this end, systems have emerged to plan missions and analyse their suitability. In addition, to check the proper performance of the mission in real time, systems to track and monitor the behavior of the drones during their flight are needed. The architectures of all these systems, both mission planning and tracking and monitoring systems, have been analyzed and a drone fleet management system has been designed and developed to plan missions and track and monitor drones in real time. These systems are modular, allowing the addition of new features or the removal or editing of some existing ones in a simple way. The modularity can provide more flexibility than other systems that perform mission planning and monitoring, and it is not tied to a single UAV model.

Additionally, some insights on the use of that type of systems in drone fleet mission planning and monitoring have been provided. The analysis of the mission planning problem has revealed that an efficient and effective way of computing paths is critical to the problem. For this reason, all the developed algorithms to make the mission planning are developed to minimize the time invested and thus obtain the planification as soon as possible. On the other hand, the analysis of the tracking and monitoring problems has revealed that an effective and robust filter is critical to the problem. So, a good algorithm has been developed to obtain the most precise positions and thus obtain the least errors during monitoring.

The remainder of this chapter contains an overview of the challenges and solutions developed in this thesis, a list of the main contributions and a description of

the lines of work remain open for further development.

7.1 Overview

Public services are starting to use or have interest into the potencial of Unmanned Aircraft Vehicles (UAVs) as support elements to their day-to-day procedures. Overall, each authority is developing its own solution for operation in an independent and redundant manner. It is usually quite complicated to share and coordinate resources among different stakeholders, but it is obvious that an efficient coordination is possible to enable the full exploitation of drone fleets in a coordinated way.

In this context, among all the possible service that could be developed to management of unmanned aircrafts, this thesis focuses on those services that can be used to plannify the drone's missions and track and monitorize them. Much of the effort in this thesis was dedicated to define a series of services that can satisfy this purpose. These services have been divided in functions reflecting the relationships between them.

Many of the tools used to design missions make it quickly and efficiently but do not take into account the performance of more complex scenarios. These missions are created in many environments with very different objectives but they are encompassed under the same concept, to go from one place to another with a drone and performing something during the journey, whether it is an inspection or a delivery. Under these premises arises the need to bring together all the drone mission planning under a single system that is capable of obtaining an effective planning, fast, taking into account possible problems and considering alternatives regardless of the drone used and the scenario created. With this tool it is possible to manage a drone fleet in an efficient way. This system is ready to evolve, as nowadays legislation is not stable and it may require changes in the type of procedures or in the sequence of tasks accomplish.

As the first step towards the building of the Mission Planning Service adequate for drones, it was necessary to define a formal structure to encompass all the different types of mission that exist and work in the system in a unified way regardless of the drone or scenario. Different forms of mission definition have been analysed. These two forms of mission definition are a basic GeoJSON inspired solution and a second one that allows the elaboration of more complex scenarios. These mission definitions are composed of the same elements but with a different level of complexity. These structures are mainly composed of the start and end points of the mission as well as a series of geometric figures with a number of constraints. These geometrical figures must be transformed into a series of points so that both mission definition structures are equivalent and the flight plan and trajectory can be calculated indistinctly from the scenario.

The calculation of the points that compose the geometric figures of the mission can be done in different ways as the literature exposes, but in the system the geometry is used for this calculation. From the definition of the geometric figure and a series of restrictions, the points that compose it and through which the drone will have to pass to carry out the mission are determined. The geometrical figures can be as simple as a line defined by the points, so the calculation is immediate, or as complex as a cylinder, so the calculation is complicated. In addition, this functionality also calculates the points that lead from one geometric figure to another getting a succession of pass points of the drone that defines the mission that it must carry out. This succession of calculated points arrives at the Flight Planning Service that works independently of the mission object entered into the system.

The idea of a unification of types of missions exists in this thesis to allow the separation between the planning of the flight plan and the calculation of the trajectory. In addition to the division in three services, these in turn are broken down into different functionalities which further simplifies the change, removal or addition of functionalities to the system, i.e. this modularity gives the service great flexibility to make changes.

A flight plan is obtained after the list of points passes through a series of functionalities that calculate the suitability and make the necessary modifications to carry out a mission. In order to calculate flight plans, the service makes use of other services that provide it with context information. These external services are the ground service, the no drone zone model and the airspace model.

Despite these calls to external services, the Flight Planning Service is fast and a solution is obtained in a short space of time. One might think that these calls would cause a bottleneck to the service and slow down the achievement of a planning solution, but this is not the case as it has been analyzed and demonstrated. The Ground Service is slow to respond if the mission is too long, but it has been proven that if its implementation is parallelized, the response time drops dramatically. In this way, the ground service returns the heights of the area surrounding the mission, the no drone zone model returns the geofences located in the vicinity of the mission and the airspace model returns the airspace routes that are in the mission area, as well as the airspace rules that the drone must follow.

A summary of the flight plan creation process is explained below. The restrictions imposed by the user are added to the waypoints so the drone is going to avoid the terrain in these waypoints, but this does not guarantee the avoidance of the terrain during the trajectory that joins two consecutive waypoints. In this point, the terrain avoidance functionality comes into play because with the heights of the terrain, the functionality modifies the heights of the waypoints to avoid the terrain or obstacles, and if it is not possible, the functionality creates a waypoint to guarantee the safety during the flight. This method of avoiding the terrain has been analysed and concluded that it performs quickly and properly so that the terrain can be

avoided. Then the functionality of no drone zones avoidance checks if the flight plan passes through any geofence that has returned the no drone zone model. In the case that such an intersection occurs, a series of alternatives have been considered to avoid such geofence. The system has been proposed to that this functionality is not forced to give alternatives if it does not find a way to avoid such geofence. It also acts quickly for the system to plan missions in the shortest time possible as it can give the situation of wanting to do missions urgently. Finally, the flight plan must go through the airspace restrictions functionality to check that it complies with current legislation and if any airspace route is found. In an affirmative case, the flight plan is modified to follow the route. With all this, a flight plan without time restrictions is achieved but ensuring compliance with spatial restrictions and obtaining results in a short time as demonstrated in the relevant chapter. Then, the flight plan arrives to a functionality where the time constraints are added. These time constraints are obtained when calculating the trajectory, that is, the evolution over time that follows the behavior of a drone to perform a mission. This process is made in the Trajectory Prediction Service and it is explained below. Once the time constraints are added to the flight plan, this is completed and it returns to the Mission Planning Service. There, the flight plan can be returned to the user and stored. In turn the flight plan is translated into a format (MAVLink) understandable by most UAVs so they can perform the mission autonomously if that is how it should be done.

This separation of objects (mission, flight plan and trajectory) is made to distinguish the processes that infer in each of them. This separation of functionalities has been done to differentiate well the different processes that are passed through from the moment a mission is defined until a trajectory is obtained.

A trajectory predictor can be formulated as a trajectory language processing engine where the input is a flight plan and the output is a sampled trajectory. This engine was named Trajectory Planning Service. It was also found that many of the components that compose this engine could be also seen as simpler engines.

Upon arrival at the Trajectory Prediction Service of the unfinished flight plan, the process of conversion into instructions begins. Instructions are defined as the orders a drone must follow to go from one point to another in the flight plan characterized by a position and height restrictions. Therefore, these instructions take an execution time and this time is added as time restrictions to each of the points that make up the flight plan. In turn, the instructions can be executed more or less quickly depending on the speed of the drone, which provides a time frame for each point in the flight plan. Once all the instructions to be followed by the drone to complete the mission have been calculated, all the time restrictions are obtained and they are sent back to the Flight Planning Service.

Once the sequence of instructions is obtained, it is time to get the sampled trajectory. To compute an accurate trajectory it is necessary the drone physical features (aircraft performance model) and the effect of the wind which is acquired

from the weather model. A system of differential equations that models the drone motion dynamics is solved in this functionality and the relevant parameters that model the trajectory are obtained from these data.

After obtaining the path, the planning part of the system is completed. But as it was said, the calculated flight plans are executed by the drones, automatically or manually, to carry out the missions created for that purpose. This execution of the flight plans must be tracked and monitored to verify that the mission is carried out as defined and respecting all indications to ensure the safety of the flight, the actors involved in it and third parties.

For the system to be able to track the drones, their position as well as their behavior must be captured in some possible way. There are many possible ways to capture this information in the literature, but the system proposed in this Ph.D. thesis may process the telemetry sent by the drones, as well as information obtained after the drones are detected by non-cooperative sensors, or from cooperative sensors. The fusion of data from different sources provides more accurate information on the behavior of the drones being monitored in real time.

After receiving the position information from the drones, they must be tracked. To perform this function, a tracking system has been designed which processes the horizontal measures in parallel with a robust filter and a map-based filter. In this system, the robust filter follows a classic approach using an IMM filter whereas the map-based filter uses the flight plan information to project the measures and then performs a decoupled tracking of the longitudinal and transverse components. On the other hand, the vertical tracking is also implemented through a robust filter. Once the two components are obtained, the longitudinal component is processed by an IMM filter and the transversal component is processed by a Kalman filter. But, by exploiting the flight plan information in the map-based filter, it is not possible to have this filter to track all kinds of drones in every circumstance. For this reason, a backup robust filter is needed to track cases where the flight plan information is not available.

Once the track is built, its monitoring begins. It is made up of a series of functionalities that monitor certain behaviours.

The airspace contains areas delimited by which it can not be flown. For this reason it is necessary to have a Break-in detector capable of detecting when a drone invades one of this areas. By means of a series of calculations it is checked whether the position of the track is within a geofence. These geofences are provided by the No Drone Zone Model. If the track falls within the delimited area, the detector generates a geofence invasion alarm. Otherwise, the detector does not display anything.

A flight plan deviation detector checks the good performance of a drone at following planned flight plan. For these reasons, the Deviation detector makes a comparison between the flight plan generated in the system and the position and time information of the generated track. If there is a spatial deviation, either horizontal

or vertical, or a temporal deviation from the margins established in the flight plan by the planned time and height restrictions, the Deviation detector displays a conformance alarm. And if the drone is not following a flight plan that is in the system, the detector adds a label alerting of this situation.

Managing a fleet of drones can lead to operate in a certain area with several drones simultaneously, which can suffer deviations as just seen. A tactical conflict detector is responsible for analyzing the potential pairwise interactions between drones and alerting about them.

The system is capable of dealing with errors caused by the different uncertainty sources that are present throughout the process from the creation of a mission to its follow-up and monitoring. The main uncertainty sources are navigation, control and observation and they occur in the different functions that the system undertakes such as mission planning, flight execution and drone tracking and monitoring. Specifically, the tracking system has a filter that serves to reduce the uncertainty related to sensors noises. Meanwhile the monitoring system has adjustable margins in each of the detectors that allow to manage the alerts related to the uncertainty of navigation, as well as control.

Finally, it can be concluded that thanks to the contributions of this thesis, a system capable of managing a fleet of drones can be achieved, since mission planning, tracking and monitoring are available.

7.2 Contributions

The contributions of this thesis can be summarized as the following:

- A planning and monitoring services architecture for a drone fleet management system has been developed and validated in this thesis. This architecture, which is based on microservices, can be seen in Figure 7.1.
- A mission service able to unify different types of missions that operators and companies carry out today creating a flight plan structure has been defined. This service allows to plan urban and non-urban operations and provides flight plans that comply with user or airspace restrictions. This mission service has been implemented and validated.
- A prototype trajectory computation engine which translates a flight plan into a sampled trajectory was developed and validated. This service generates the instructions that a drone must follow to carry out the specified mission. Its input is independent of the drone used and its output is also independent as the service provides a translation to MAVLink language.

- A tracking service with an architecture composed by a robust filter and a map-based filter which run in parallel has been designed and evaluated. This service allows the processing of positioning information from different types of sensors (cooperative and non-cooperative) as well as telemetry sent by the drones. This tracking structure ensures robustness and guarantees high precision.
- Monitoring of the real-time performance of a mission by a drone is possible with high performance thanks to the tracking and monitoring service defined in this thesis. The service checks a number of parameters of the drone's performance against aerospace and flight plan restrictions. A series of experiments have been conducted to validate this service.

Figure 7.1: Proposed planning and surveillance architecture

7.3 Further work

Many of the works in this thesis open new lines of research. This section tries to emphasize those areas where new results would be particularly useful.

An important improvement that would be useful would be connecting this system to airspace management and flight authorisation authorities, by converting the flight plans and missions to format compatible to UAV Traffic Management (UTM) systems.

A major improvement would be the ability to modify flight plans in real time. This incorporation would allow the Flight Planning Service to act when in the middle of the execution of a flight an obstacle appears that does not allow to continue with the execution of the mission without having alarms that can lead to the cancellation of the flight. If this functionality were available, in the case of an unforeseen event of this type, the Flight Planning Service would look for an alternative flight plan from the current point and complying with the requirements originally established, and would send it to the drone so that it could change its action.

Another interesting line of research would be the incorporation of a resource management service to save costs and time when carrying out missions. With this service it would be possible to obtain different alternatives of goods and people to carry out a certain mission depending on the cost of displacement, the time of execution, the availability... In this way the user could select the option that best suits him/her to carry out his/her mission. With this improvement a better drone fleet management system would be obtained.

The search engine for alternatives when planning a flight is simple and does not always provide a solution. An important improvement would be the development of a more powerful alternatives search engine, which takes into account more restrictions, as well as the possibility of resolution by different alternatives together and always provides a solution. With this functionality the system would gain in precision and quality.

The system works for all quadrotor-type aircraft, but there are more types of aircraft on the market. A major improvement would be the incorporation of fixed-wing and VTOL aircraft management into the system, both in planning their missions and in monitoring their execution. With this incorporation, a wider range of missions could be carried out with this system.

Another important contribution would be to add in-flight contingency management. When a contingency occurs during a mission, the system should be able to have an closed feedback control loop, potentially presenting flight management alternatives for the mission and thus be able to solve the contingency in real time. This comparative study of alternatives should incorporate a series of metrics to determine the most suitable contingency resolution.

It is also undeniable the need for continuous study of the legislation for the flight of drones with the consequent changes in mission planning and monitoring of the drone.

References

- Administration, F. A. Laanc concept of operations, 2020. URL https://www.faa.gov/uas/programs_partnerships/uas_data_exchange/industry/media/laanc_concept_of_operations.pdf. [Web; accedido el 29-02-2020].
- AESA. Aesa: Agencia estatal de seguridad aérea, 2019. URL [URL{https://www.seguridadeaerea.gob.es/lang_castellano/home.aspx}](https://www.seguridadeaerea.gob.es/lang_castellano/home.aspx). [Web; accedido el 10-02-2019].
- Airbus. Zephyr. pioneering the stratosphere, 2020a. URL [URL{https://www.airbus.com/defence/uav/zephyr.html}](https://www.airbus.com/defence/uav/zephyr.html). [Web; accedido el 07-04-2020].
- Airbus. Airbus vahana, 2020b. URL [URL{https://www.airbus.com/innovation/urban-air-mobility/vehicle-demonstrators/vahana.html}](https://www.airbus.com/innovation/urban-air-mobility/vehicle-demonstrators/vahana.html). [Web; accedido el 07-04-2020].
- Airmap. Airmap, 2020. URL <https://www.airmap.com/>. [Web; accedido el 09-03-2020].
- Altiscope. Tr-004: Metrics to characterize dense airspace traffic, 2020. URL [URL{http://bit.ly/altiscopetr004}](http://bit.ly/altiscopetr004). [Web; accedido el 10-04-2020].
- Alwateer, M. et al. Civilian drone services: Concept, strategies and experimentation. 2019.
- Amazon. Amazon prime air, 2016, 2016. URL <https://www.amazon.com/Amazon-Prime-Air/b?ie=UTF8&node=8037720011>. [Web; accedido el 27-02-2020].
- Ardupilot. Apm planner 2 home, 2020a. URL <https://ardupilot.org/planner2/>. [Web; accedido el 07-03-2020].
- Ardupilot. Mission planner, 2020b. URL <https://ardupilot.org/planner/docs/mission-planner-overview.html>. [Web; accedido el 07-03-2020].

- Areias, B., Humberto, N., Guardalben, L., Fernandes, J. M., and Sargento, S. Towards an automated flying drones platform. In *VEHITS*, pages 529–536, 2018.
- ART. Art midrange 3d, 2020. URL [URL{https://www.advancedradartechnologies.com/products/art-drone-sentinel-2/}](https://www.advancedradartechnologies.com/products/art-drone-sentinel-2/). [Web; accedido el 21-06-2020].
- Asano, T., Asano, T., Guibas, L., Hershberger, J., and Imai, H. Visibility of disjoint polygons. *Algorithmica*, 1(1-4):49–63, 1986.
- Association, G. U. Utm project in japan, 2017. URL https://gutma.org/montreal-2017/wp-content/uploads/sites/2/2017/07/UTM-Project-in-Japan_METI.pdf. [Web; accedido el 29-02-2020].
- Association, G. U. Utm, 2020. URL <https://gutma.org>. [Web; accedido el 11-03-2020].
- Aurora. Pav – passenger air vehicle. redefining the future of safe on-demand air travel and transportation., 2020. URL [URL{https://www.aurora.aero/pav-evtol-passenger-air-vehicle/}](https://www.aurora.aero/pav-evtol-passenger-air-vehicle/). [Web; accedido el 07-04-2020].
- Balakrishnan, K., Polastre, J., Mooberry, J., Golding, R., and Sachs, P. Blueprint for the sky. *The roadmap for the safe integration of autonomous aircraft. Airbus A*, 3, 2018.
- Bank, W. Air transport, passengers carried. 2018.
- Barbehenn, M. A note on the complexity of dijkstra’s algorithm for graphs with weighted vertices. *IEEE transactions on computers*, 47(2):263, 1998.
- Barreiro, J., Boyce, M., Do, M., Frank, J., Iatauro, M., Kichkaylo, T., Morris, P., Ong, J., Remolina, E., Smith, T., et al. Europa: A platform for ai planning, scheduling, constraint programming, and optimization. *4th International Competition on Knowledge Engineering for Planning and Scheduling (ICKEPS)*, 2012.
- Bedrax-Weiss, T., McGann, C., and Iatauro, M. Europa 2: plan database services for planning and scheduling applications. In *International Conference on Automated Planning and Scheduling*, pages 18–19. Citeseer, 2005.
- Benyamin, M. and Goldman, G. H. Acoustic detection and tracking of a class i uas with a small tetrahedral microphone array. Technical report, ARMY RESEARCH LAB ADELPHI MD, 2014.

REFERENCES

- Besada, J. A., Portillo, J., García, J., Molina, J. M., Varona, Á., and González, G. Image-based automatic surveillance for airport surface. In *4th International Conference on Information Fusion, Fusion*, pages 11–18, 2001.
- Besada, J. A., García, J., Portillo, J., Molina, J. M., Varona, A., and Gonzalez, G. Airport surface surveillance based on video images. *IEEE transactions on aerospace and electronic systems*, 41(3):1075–1082, 2005.
- Besada, J. A., Frontera, G., Crespo, J., Casado, E., and Lopez-Leones, J. Automated aircraft trajectory prediction based on formal intent-related language processing. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1067–1082, 2013.
- Besada, J. A., Bergesio, L., Campaña, I., Vaquero-Melchor, D., López-Araquistain, J., Bernardos, A. M., and Casar, J. R. Drone mission definition and implementation for automated infrastructure inspection using airborne sensors. *Sensors*, 18(4):1170, 2018.
- Besada, J. A., Campaña, I., Bergesio, L., Bernardos, A. M., and de Miguel, G. Drone flight planning for safe urban operations: Utm requirements and tools. In *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 924–930. IEEE, 2019.
- Bevacqua, G., Cacace, J., Finzi, A., and Lippiello, V. Mixed-initiative planning and execution for multiple drones in search and rescue missions. In *Twenty-Fifth International Conference on Automated Planning and Scheduling*, 2015.
- Blasch, E., Wang, Z., Shen, D., Ling, H., and Chen, G. Surveillance of ground vehicles for airport security. In *Geospatial InfoFusion and Video Analytics IV; and Motion Imagery for ISR and Situational Awareness II*, volume 9089, page 90890B. International Society for Optics and Photonics, 2014.
- Blom, H. A. and Bar-Shalom, Y. The interacting multiple model algorithm for systems with markovian switching coefficients. *IEEE transactions on Automatic Control*, 33(8):780–783, 1988.
- Blyakhman, A., Burov, V., Myakinkov, A., and Ryndyk, A. Detection of unmanned aerial vehicles via multi-static forward scattering radar with airborne transmit positions. In *2014 International Radar Conference*, pages 1–5. IEEE, 2014.
- BOE. Disposición 15721 del boe núm. 316 de 2017, 2017. URL [URL{https://www.boe.es/boe/dias/2017/12/29/pdfs/BOE-A-2017-15721.pdf}](https://www.boe.es/boe/dias/2017/12/29/pdfs/BOE-A-2017-15721.pdf). [Web; accedido el 10-02-2019].

- Bozhinoski, D., Di Ruscio, D., Malavolta, I., Pelliccione, P., and Tivoli, M. Flyaq: Enabling non-expert users to specify and generate missions of autonomous multi-copters. In *2015 30th IEEE/ACM International Conference on Automated Software Engineering (ASE)*, pages 801–806. IEEE, 2015.
- Branco, L. H. C. and Segantine, P. C. L. Maniac-uav-a methodology for automatic pavement defects detection using images obtained by unmanned aerial vehicles. In *Journal of Physics: Conference Series*, volume 633, page 012122. IOP Publishing, 2015.
- Brisset, P., Drouin, A., Gorraz, M., Huard, P.-S., and Tyler, J. The paparazzi solution. 2006.
- Butler, H., Daly, M., Doyle, A., Gillies, S., Hagen, S., Schaub, T., and Wilde, E. Geojson. *Electronic*. URL: <http://geojson.org>, 2014.
- Butler, H., Daly, M., Doyle, A., Gillies, S., Hagen, S., Schaub, T., et al. The geojson format. *Internet Engineering Task Force (IETF)*, 2016.
- CAAC. Uoms in china, 2018. URL https://rpas-regulations.com/wp-content/uploads/2018/06/1.2-Day1_0910-1010_CAAC-SRI_Zhang-Jianping_UOMS-EN.pdf. [Web; accedido el 29-02-2020].
- Campaña Ramos, I. Diseño de un prototipo de un sistema de vigilancia basado en visión artificial. Master’s thesis, Telecomunicacion, 2016.
- Canny, J. and Reif, J. New lower bound techniques for robot motion planning problems. In *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*, pages 49–60. IEEE, 1987.
- Capolupo, A., Pindozi, S., Okello, C., Fiorentino, N., and Boccia, L. Photogrammetry for environmental monitoring: The use of drones and hydrological models for detection of soil contaminated by copper. *Science of the Total Environment*, 514:298–306, 2015.
- Carvajal, F., Agüera, F., and Pérez, M. Surveying a landslide in a road embankment using unmanned aerial vehicle photogrammetry. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(1/C22), 2011.
- Case, E. E., Zelnio, A. M., and Rigling, B. D. Low-cost acoustic array for small uav detection and tracking. In *2008 IEEE National Aerospace and Electronics Conference*, pages 110–113. IEEE, 2008.

- Causa, F., Fasano, G., and Grassi, M. Multi-uav path planning for autonomous missions in mixed gnss coverage scenarios. *Sensors*, 18(12):4188, 2018.
- Cavalcante, T. R. F., de Bessa, I. V., and Cordeiro, L. C. Planning and evaluation of uav mission planner for intralogistics problems. In *2017 VII Brazilian Symposium on Computing Systems Engineering (SBESC)*, pages 9–16. IEEE, 2017.
- Chew, L. P. There are planar graphs almost as good as the complete graph. *Journal of Computer and System Sciences*, 39(2):205–219, 1989.
- Cho, T., Lee, C., and Choi, S. Multi-sensor fusion with interacting multiple model filter for improved aircraft position accuracy. *Sensors*, 13(4):4122–4137, 2013.
- Choi, J., Sellen, J., and Yap, C.-K. Precision-sensitive euclidean shortest path in 3-space. In *Proceedings of the eleventh annual symposium on Computational geometry*, pages 350–359, 1995.
- Choi, J., Sellen, J., and Yap, C.-K. Approximate euclidean shortest paths in 3-space. *International Journal of Computational Geometry & Applications*, 7(04):271–295, 1997.
- Choi, S.-C., Sung, N.-M., Park, J.-H., Ahn, I.-Y., and Kim, J. Enabling drone as a service: Onem2m-based uav/drone management system. In *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 18–20. IEEE, 2017.
- Chornenky, D. *A multi-sided platform for remote operation of shared drones: deriving strategic opportunities from regulatory trends*. PhD thesis, Massachusetts Institute of Technology, 2018.
- Cir, I. 328: Unmanned aircraft systems (uas). *International Civil Aviation Organization*, 2011.
- Clarkson, K. Approximation algorithms for shortest path motion planning. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 56–65, 1987.
- Cordova, F. and Olivares, V. Design of drone fleet management model in a production system of customized products. In *2016 6th International Conference on Computers Communications and Control (ICCCC)*, pages 165–172. IEEE, 2016.
- Crespo, G., Glez-de Rivera, G., Garrido, J., and Ponticelli, R. Setup of a communication and control systems of a quadrotor type unmanned aerial vehicle. In *Design of Circuits and Integrated Systems*, pages 1–6. IEEE, 2014.

- De Berg, M., Van Kreveld, M., Overmars, M., and Schwarzkopf, O. Computational geometry. In *Computational geometry*, pages 1–17. Springer, 1997.
- De Filippis, L., Guglieri, G., and Quagliotti, F. Path planning strategies for uavs in 3d environments. *Journal of Intelligent & Robotic Systems*, 65(1-4):247–264, 2012.
- De Lellis, E., Morani, G., Corrado, F., and Di Vito, V. On-line trajectory generation for autonomous unmanned vehicles in the presence of no-fly zones. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 227(2):381–393, 2013.
- De Melo, R. R. S., Costa, D. B., Álvares, J. S., and Irizarry, J. Applicability of unmanned aerial system (uas) for safety inspection on construction sites. *Safety science*, 98:174–185, 2017.
- Dedrone. Dedrone, 2020a. URL <http://www.dedrone.com/en/>. [Web; accedido el 15-03-2020].
- Dedrone. Rf-300, 2020b. URL [URL{https://www.dedrone.com/products/hardware/rf-sensors/rf-300}](https://www.dedrone.com/products/hardware/rf-sensors/rf-300). [Web; accedido el 21-06-2020].
- DJI. Phantom 4 pro especificaciones, 2019. URL [URL{https://www.dji.com/es/phantom-4-pro/info#specs}](https://www.dji.com/es/phantom-4-pro/info#specs). [Web; accedido el 21-02-2019].
- DJI. Dji gs pro, 2020a. URL <https://www.dji.com/es/ground-station-pro>. [Web; accedido el 07-03-2020].
- DJI. Windows sdk, 2020b. URL <https://developer.dji.com/windows-sdk/>. [Web; accedido el 07-03-2020].
- DJI. Mobile sdk, 2020c. URL <https://developer.dji.com/mobile-sdk/>. [Web; accedido el 07-03-2020].
- DJI. Mavic series, 2020d. URL [URL{https://www.dji.com/es/products/mavic?site=brandsite&from=nav}](https://www.dji.com/es/products/mavic?site=brandsite&from=nav). [Web; accedido el 07-04-2020].
- DJI. Phantom series, 2020e. URL [URL{https://www.dji.com/es/products/phantom?site=brandsite&from=nav}](https://www.dji.com/es/products/phantom?site=brandsite&from=nav). [Web; accedido el 07-04-2020].
- Doherty, P., Heintz, F., and Kvarnström, J. High-level mission specification and planning for collaborative unmanned aircraft systems using delegation. *Unmanned Systems*, 1(01):75–119, 2013.

REFERENCES

- Dronecode. Qgroundcontrol, 2020. URL <http://qgroundcontrol.com/>. [Web; accedido el 07-03-2020].
- Drones, D. Mav pilot 1.4 for iphone released, 2020. URL <https://diydrones.com/profiles/blogs/mav-pilot-1-4-for-iphone-released>. [Web; accedido el 07-03-2020].
- Drones, I. A. Arms, 2019. URL [URL{https://www.indracompany.com/es/arms}](https://www.indracompany.com/es/arms). [Web; accedido el 21-06-2020].
- DroneShield. Droneshield, 2020. URL <http://www.droneshield.com/>. [Web; accedido el 15-03-2020].
- Dudani, S. A., Breeding, K. J., and McGhee, R. B. Aircraft identification by moment invariants. *IEEE transactions on computers*, 100(1):39–46, 1977.
- EASA. Opinion 01/2020. high-level regulatory framework for the u-space, 2020. URL [URL{https://www.easa.europa.eu/document-library/opinions/opinion-012020}](https://www.easa.europa.eu/document-library/opinions/opinion-012020). [Web; accedido el 15-04-2020].
- Eiband, D. M. and Kern, L. R. Generic drone control system, August 31 1993. US Patent 5,240,207.
- ENAIRE. Aip españa, servicio de información aeronáutica, 2019. URL [URL{https://ais.enaire.es/aip/}](https://ais.enaire.es/aip/). [Web; accedido el 21-02-2019].
- EUROCONTROL. Acas safety studies, 2004, 2004. URL [URL{http://www.eurocontrol.int/sites/default/files/article/content/documents/nm/safety/ACAS/safety-acas-paper-safety-studies-20040204.pdf}](http://www.eurocontrol.int/sites/default/files/article/content/documents/nm/safety/ACAS/safety-acas-paper-safety-studies-20040204.pdf). [Web; accedido el 10-04-2020].
- Eurocontrol. Eurocontrol standard document for surveillance data exchange part 9: Category 062 transmission of system track data, 2019. URL [URL{https://www.eurocontrol.int/sites/default/files/content/documents/nm/asterix/archives/asterix-cat062-system-track-data-part9-v1.12-092010.pdf}](https://www.eurocontrol.int/sites/default/files/content/documents/nm/asterix/archives/asterix-cat062-system-track-data-part9-v1.12-092010.pdf). [Web; accedido el 13-03-2019].
- Evers, L., Dollevoet, T., Barros, A. I., and Monsuur, H. Robust uav mission planning. *Annals of Operations Research*, 222(1):293–315, 2014.
- FAA, A. Auvsi.(2016). *Know Before You Fly.[online]* Available at: <http://knowbeforeyoufly.org/>[Accessed 13 Oct. 2016], 2016.
- FAA-NASA. Utm, 2020. URL <https://utm.arc.nasa.gov/index.shtml>. [Web; accedido el 11-03-2020].

- Federal Aviation Administration, F. Remote identification of unmanned aircraft systems, 2019. URL [URL{https://www.federalregister.gov/documents/2019/12/31/2019-28100/remote-identification-of-unmanned-aircraft-systems}](https://www.federalregister.gov/documents/2019/12/31/2019-28100/remote-identification-of-unmanned-aircraft-systems). [Web; accedido el 21-06-2020].
- Fesenko, H., Kharchenko, V., and Zaitseva, E. Evaluating reliability of a multi-fleet with a reserve drone fleet:: an approach and basic model. In *2019 International Conference on Information and Digital Technologies (IDT)*, pages 128–132. IEEE, 2019.
- Fioranelli, F., Ritchie, M., Griffiths, H., and Borrión, H. Classification of loaded/unloaded micro-drones using multistatic radar. *Electronics Letters*, 51(22): 1813–1815, 2015.
- Forney, G. D. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.
- Frontera, G., Besada, J. A., Bernardos, A. M., Casado, E., and López-Leonés, J. Formal intent-based trajectory description languages. *IEEE Transactions on Intelligent Transportation Systems*, 15(4):1550–1566, 2014.
- Frontera, G., Besada, J. A., and López-Leonés, J. Generation of aircraft intent based on a microstrategy search tree. *IEEE Transactions on Intelligent Transportation Systems*, 18(6):1405–1421, 2016.
- Frontera, G., Martín, D. J., Besada, J. A., and Gu, D.-W. Approximate 3d euclidean shortest paths for unmanned aircraft in urban environments. *Journal of Intelligent & Robotic Systems*, 85(2):353–368, 2017.
- Frontera, G., Campana, I., Bernardos, A. M., and Besada, J. A. Formal intent-based trajectory description languages for quadrotor aircraft. *IEEE Transactions on Aerospace and Electronic Systems*, 2019.
- Frontera Sánchez, G. *Applications of formal languages to management of manned and unmanned aircraft*. PhD thesis, Telecomunicacion, 2016.
- Früh, C. and Zakhor, A. Constructing 3d city models by merging aerial and ground views. *IEEE Computer Graphics and Applications*, 23(6):52–61, 2003.
- Furini, F., Persiani, C. A., and Toth, P. The time dependent traveling salesman planning problem in controlled airspace. *Transportation Research Part B: Methodological*, 90:38–55, 2016.

- Gentilini, D., Farina, N., Franco, E., Tirri, A. E., Accardo, D., Moriello, R. S. L., and Angrisani, L. Multi agent path planning strategies based on kalman filter for surveillance missions. In *2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*, pages 1–6. IEEE, 2016.
- Gewali, L. P., Ntafos, S., and Tollis, I. G. Path planning in the presence of vertical obstacles. *IEEE Transactions on Robotics and Automation*, 6(3):331–341, 1990.
- Ghosh, S. K. and Mount, D. M. An output-sensitive algorithm for computing visibility graphs. *SIAM Journal on Computing*, 20(5):888–910, 1991.
- Giones, F. and Brem, A. From toys to tools: The co-evolution of technological and entrepreneurial developments in the drone industry. *Business Horizons*, 60(6): 875–884, 2017.
- Goldberg, A. and Radzik, T. A heuristic improvement of the bellman-ford algorithm. Technical report, STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, 1993.
- González-Jorge, H., Martínez-Sánchez, J., Bueno, M., et al. Unmanned aerial systems for civil applications: A review. *Drones*, 1(1):2, 2017.
- Grabbe, S., Sridhar, B., and Mukherjee, A. Sequential traffic flow optimization with tactical flight control heuristics. *Journal of Guidance, Control, and Dynamics*, 32(3):810–820, 2009.
- Greenwood, W. W., Lynch, J. P., and Zekkos, D. Applications of uavs in civil infrastructure. *Journal of Infrastructure Systems*, 25(2):04019002, 2019.
- Guibas, L., Hershberger, J., Leven, D., Sharir, M., and Tarjan, R. E. Linear-time algorithms for visibility and shortest path problems inside triangulated simple polygons. *Algorithmica*, 2(1-4):209–233, 1987.
- Güvenç, İ., Ozdemir, O., Yapici, Y., Mehrpouyan, H., and Matolak, D. Detection, localization, and tracking of unauthorized uas and jammers. In *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, pages 1–10. IEEE, 2017.
- Haidari, L. A., Brown, S. T., Ferguson, M., Bancroft, E., Spiker, M., Wilcox, A., Ambikapathi, R., Sampath, V., Connor, D. L., and Lee, B. Y. The economic and operational value of using drones to transport vaccines. *Vaccine*, 34(34): 4062–4067, 2016.

- Haluza, M. and Čechák, J. Analysis and decoding of radio signals for remote control of drones. In *2016 New Trends in Signal Processing (NTSP)*, pages 1–5. IEEE, 2016.
- Hamza, M., Jehangir, A., Ahmad, T., Sohail, A., and Naeem, M. Design of surveillance drone with x-ray camera, ir camera and metal detector. In *2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, pages 111–114. IEEE, 2017.
- Hassanalain, M. and Abdelkefi, A. Classifications, applications, and design challenges of drones: A review. *Progress in Aerospace Sciences*, 91:99–131, 2017.
- Hershberger, J. and Suri, S. An optimal algorithm for euclidean shortest paths in the plane. *SIAM Journal on Computing*, 28(6):2215–2256, 1999.
- Hester, K. Andropilot users guide, 2020. URL <https://github.com/geeksville/arduleader/wiki/Andropilot-Users-Guide>. [Web; accedido el 07-03-2020].
- HGH-Infrared. Spynel v-lrf, 2020. URL [URL{https://www.hgh-infrared.com/Videos-Resources/Optronics-for-Security/Brochures/Spynel-V-LRF}](https://www.hgh-infrared.com/Videos-Resources/Optronics-for-Security/Brochures/Spynel-V-LRF). [Web; accedido el 21-06-2020].
- Hougardy, S. The floyd-warshall algorithm on graphs with negative cycles. *Information Processing Letters*, 110(8-9):279–281, 2010.
- Hu, X.-B., Wu, S.-F., and Jiang, J. On-line free-flight path optimization based on improved genetic algorithms. *Engineering Applications of Artificial Intelligence*, 17(8):897–907, 2004.
- Huuskonen, J. and Oksanen, T. Soil sampling with drones and augmented reality in precision agriculture. *Computers and electronics in agriculture*, 154:25–35, 2018.
- ICAO, P. Air traffic management. *Doc-4444*, 2007.
- IGN. Instituto geográfico nacional, 2019. URL [URL{http://www.ign.es/web/ign/portal}](http://www.ign.es/web/ign/portal). [Web; accedido el 06-12-2019].
- In-Situ. Cellular network telemetry systems, 2020. URL [URL{https://dicore.cl/PDFs/In-Situ/TUBE%20&%20CUBE%20300R%20TELEMETRY%20SYSTEM.pdf}](https://dicore.cl/PDFs/In-Situ/TUBE%20&%20CUBE%20300R%20TELEMETRY%20SYSTEM.pdf). [Web; accedido el 21-06-2020].
- JARUS. Jarus guidelines on specific operations risk assessment (sora). ed. 2.0 (2019)., 2019. URL <http://jarus-rpas.org/content/jar-doc-06-sora-package>. [Web; accedido el 11-03-2020].

- JARUS. Joint authorities for rulemaking on unmanned systems, 2020. URL <http://jarus-rpas.org>. [Web; accedido el 11-03-2020].
- jDrones. jd-rf900plus longrange, 2020. URL [URL{http://store.jdrones.com/jD_RD900Plus_Telemetry_Bundle_p/rf900set02.htm}](http://store.jdrones.com/jD_RD900Plus_Telemetry_Bundle_p/rf900set02.htm). [Web; accedido el 21-06-2020].
- jetvision. Air!squitter, 2020. URL [URL{https://airsquitter.com/}](https://airsquitter.com/). [Web; accedido el 21-06-2020].
- Jones, E. New approaches for autonomous logistics aircraft and ground systems. In *2007 Institution of Engineering and Technology Conference on Autonomous Systems*, pages 1–7. IET, 2007.
- Karaman, S. and Frazzoli, E. Linear temporal logic vehicle routing with applications to multi-uav mission planning. *International Journal of Robust and Nonlinear Control*, 21(12):1372–1395, 2011.
- Kastella, K. and Biscuso, M. Tracking algorithms for air traffic control applications. *Air Traffic Control Quarterly*, 3(1):19–43, 1995.
- Kersnovski, T., Gonzalez, F., and Morton, K. A uav system for autonomous target detection and gas sensing. In *2017 IEEE aerospace conference*, pages 1–12. IEEE, 2017.
- Kim, Y., Gu, D.-W., and Postlethwaite, I. Real-time optimal mission scheduling and flight path selection. *IEEE Transactions on Automatic control*, 52(6):1119–1123, 2007.
- Kirubarajan, T., Bar-Shalom, Y., Pattipati, K. R., and Kadar, I. Ground target tracking with variable structure imm estimator. *IEEE Transactions on Aerospace and Electronic Systems*, 36(1):26–46, 2000.
- Krátký, M. and Fuxa, L. Mini uavs detection by radar. In *International Conference on Military Technologies (ICMT) 2015*, pages 1–5. IEEE, 2015.
- Krishnan, K. *Data warehousing in the age of big data*. Newnes, 2013.
- Lacroix, S., Roberts, G., Benard, E., Bronz, M., Burnet, F., Bouhoubeiny, E., Condomines, J.-P., Doll, C., Hattenberger, G., Lamraoui, F., et al. Fleets of enduring drones to probe atmospheric phenomena with clouds. In *EGU General Assembly Conference Abstracts*, volume 18, 2016.
- Langley, R. B. The utm grid system. *GPS world*, 9(2):46–50, 1998.

- Laporte, G. and Martello, S. The selective travelling salesman problem. *Discrete applied mathematics*, 26(2-3):193–207, 1990.
- Leary, S., Deittert, M., and Bookless, J. Constrained uav mission planning: A comparison of approaches. In *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, pages 2002–2009. IEEE, 2011.
- Li, X.-R. and Bar-Shalom, Y. Design of an interacting multiple model algorithm for air traffic control tracking. *IEEE Transactions on Control Systems Technology*, 1(3):186–194, 1993.
- Liang, M. and Delahaye, D. Drone fleet deployment strategy for large scale agriculture and forestry surveying. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 4495–4500. IEEE, 2019.
- Liu, M., Liu, X., Zhu, M., and Zheng, F. Stochastic drone fleet deployment and planning problem considering multiple-type delivery service. *Sustainability*, 11(14):3871, 2019.
- Lopez, J. L.-A. Contributions to surveillance and air traffic management on the airport area. 2018.
- Lopez Leones, J. *Definition of an aircraft intent description language for air traffic management applications*. PhD thesis, University of Glasgow, 2008.
- López-Leonés, J., Vilaplana, M. A., Gallo, E., Navarro, F. A., and Querejeta, C. The aircraft intent description language: A key enabler for air-ground synchronization in trajectory-based operations. In *2007 IEEE/AIAA 26th Digital Avionics Systems Conference*, pages 1–D. IEEE, 2007.
- Álvarez, R. Xataka. aquí está la nueva versión del drone mensajero de amazon: ahora con inteligencia artificial para entregar paquetes de forma autónoma, 2020. URL [URL{https://www.xataka.com/drones/}](https://www.xataka.com/drones/). [Web; accedido el 07-04-2020].
- MahmoudZadeh, S., Powers, D. M., Sammut, K., Atyabi, A., and Yazdani, A. A hierarchal planning framework for auv mission management in a spatiotemporal varying ocean. *Computers & Electrical Engineering*, 67:741–760, 2018.
- Manchuk, J. G. and Deutsch, C. Conversion of latitude and longitude to utm coordinates. *Paper 410, CCG Annual Report*, 11, 2009.
- Mangharam, R. Fly-by-logic: A tool for unmanned aircraft system fleet planning using temporal logic. *NASA Formal Methods. Springer*, page 355, 2019.

- Maor, Y. System and method for drone fleet management for harvesting and dilution, July 25 2019. US Patent App. 16/324,981.
- Marsh, L. M. Filtering and detection of telemetry, January 20 2009. US Patent 7,480,207.
- Martinez, C., Sampedro, C., Chauhan, A., and Campoy, P. Towards autonomous detection and tracking of electric towers for aerial power line inspection. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 284–295. IEEE, 2014.
- Máthé, K. and Buşoniu, L. Vision and control for uavs: A survey of general methods and of inexpensive platforms for infrastructure inspection. *Sensors*, 15(7):14887–14916, 2015.
- Matsuoka, R., Nagusa, I., Yasuhara, H., Mori, M., Katayama, T., Yachi, N., Hasui, A., Katakuse, M., and Atagi, T. Measurement of large-scale solar power plant by using images acquired by non-metric digital camera on board uav. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 39:B1, 2012.
- Maza, I., Muñoz-Morera, J., Caballero, F., Casado, E., Perez-Villar, V., and Ollero, A. Architecture and tools for the generation of flight intent from mission intent for a fleet of unmanned aerial systems. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 9–19. IEEE, 2014.
- McCallie, D., Butts, J., and Mills, R. Security analysis of the ads-b implementation in the next generation air transportation system. *International Journal of Critical Infrastructure Protection*, 4(2):78–87, 2011.
- Menouar, H., Guvenc, I., Akkaya, K., Uluagac, A. S., Kadri, A., and Tuncer, A. Uav-enabled intelligent transportation systems for the smart city: Applications and challenges. *IEEE Communications Magazine*, 55(3):22–28, 2017.
- Military.com. Drones, 2020. URL [URL{https://www.military.com/equipment/drones}](https://www.military.com/equipment/drones). [Web; accedido el 07-04-2020].
- Moses, A., Rutherford, M. J., and Valavanis, K. P. Radar-based detection and identification for miniature air vehicles. In *2011 IEEE International Conference on Control Applications (CCA)*, pages 933–940. IEEE, 2011.
- Motlagh, N. H., Bagaa, M., and Taleb, T. Uav-based iot platform: A crowd surveillance use case. *IEEE Communications Magazine*, 55(2):128–134, 2017.

- Mount, D. M. On finding shortest paths on convex polyhedra. Technical report, MARYLAND UNIV COLLEGE PARK CENTER FOR AUTOMATION RESEARCH, 1985.
- Murugan, D., Garg, A., and Singh, D. Development of an adaptive approach for precision agriculture monitoring with drone and satellite data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(12): 5322–5328, 2017.
- NASA. Concept of operations, 2018. URL <https://utm.arc.nasa.gov/docs/2018-UTM-ConOps-v1.0.pdf>. [Web; accedido el 29-02-2020].
- NASA. Uas traffic management (utm), 2020. URL <https://utm.arc.nasa.gov>. [Web; accedido el 26-02-2020].
- Nedjati, A., Vizvari, B., and Izbirak, G. Post-earthquake response by small uav helicopters. *Natural Hazards*, 80(3):1669–1688, 2016.
- Nelson, M. Uav mission planning. Technical report, 1995.
- Nguyen, P., Ravindranatha, M., Nguyen, A., Han, R., and Vu, T. Investigating cost-effective rf-based detection of drones. In *Proceedings of the 2nd Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use*, pages 17–22, 2016.
- Niethammer, U., Rothmund, S., Schwaderer, U., Zeman, J., and Joswig, M. Open source image-processing tools for low-cost uav-based landslide investigations. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 38(1):C22, 2011.
- Omar, R. and Gu, D.-W. Visibility line based methods for uav path planning. In *2009 ICCAS-SICE*, pages 3176–3181. IEEE, 2009.
- Omar, R. and Gu, D. 3d path planning for unmanned aerial vehicles using visibility line based method. In *ICINCO (1)*, pages 80–85, 2010.
- Organization, I. C. A. Global air traffic management operational concept, 2005.
- Papakonstantinou, N., Bashir, A. Z., O’Halloran, B., and Van Bossuyt, D. L. Early assessment of drone fleet defence in depth capabilities for mission success. In *2019 Annual Reliability and Maintainability Symposium (RAMS)*, pages 1–7. IEEE, 2019.
- Parrot. Parrot. standard, parrot., 2019. URL [URL{http://developer.parrot.com}](http://developer.parrot.com). [Web; accedido el 11-12-2019].

- Parrot. Parrot ground sdk, 2020a. URL <https://developer.parrot.com/>. [Web; accedido el 07-03-2020].
- Parrot. Parrot sdk, 2020b. URL <https://developer.parrot.com/docs/SDK3/#general-information>. [Web; accedido el 07-03-2020].
- Pastor, E., Santamaria, E., Royo, P., Lopez, J., and Barrado, C. On the design of a uas flight plan monitoring and edition system. In *2010 IEEE Aerospace Conference*, pages 1–20. IEEE, 2010.
- Pereira, F., Pinto, J., Sousa, J., Gomes, R., Gonçalves, G. M., and Dias, P. Mission planning and specification in the neptus framework. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 3220–3225. IEEE, 2006.
- Pham, T. and Srour, N. Ttcp ag-6: Acoustic detection and tracking of uavs. In *Unattended/Unmanned Ground, Ocean, and Air Sensor Technologies and Applications VI*, volume 5417, pages 24–30. International Society for Optics and Photonics, 2004.
- Pohl, A. J. and Lamont, G. B. Multi-objective uav mission planning using evolutionary computation. In *2008 winter simulation conference*, pages 1268–1279. IEEE, 2008.
- PostEurop. Facts figures, 2020. URL <http://deliver4europe.eu/facts-figures/>. [Web; accedido el 26-02-2020].
- QgroundControl. Mavlink. standard, qgroundcontrol, 2019. URL [URL{http://qgroundcontrol.org/mavlink/start}](http://qgroundcontrol.org/mavlink/start). [Web; accedido el 13-12-2019].
- Qi, Z., Shao, Z., Ping, Y. S., Hiot, L. M., and Leong, Y. K. An improved heuristic algorithm for uav path planning in 3d environment. In *2010 Second International Conference on Intelligent Human-Machine Systems and Cybernetics*, volume 2, pages 258–261. IEEE, 2010.
- QTSI. Detecting, classifying, and tracking unmanned aerial system aircraft with quantum’s vector series qa-100tm intrusion detection solution, 2020. URL <http://www.qtsi.com/>. [Web; accedido el 15-03-2020].
- Qu, Y.-h., Pan, Q., and Yan, J.-g. Flight path planning of uav based on heuristically search and genetic algorithms. In *31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005.*, pages 5–pp. IEEE, 2005.
- Rath, M., Shekarappa, N., and Ramachandra, V. Ground control station for uav, October 18 2007. US Patent App. 11/403,472.

- Rau, J., Jhan, J., Lo, C., and Lin, Y. Landslide mapping using imagery acquired by a fixed-wing uav. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, 38(1/C22): 195–200, 2011.
- Reif, J. H. and Storer, J. A. A single-exponential upper bound for finding shortest paths in three dimensions. *Journal of the ACM (JACM)*, 41(5):1013–1019, 1994.
- Richman, B., Bauer, M. P., Michini, B. J., and Poole, A. J. Unmanned aerial vehicle rooftop inspection system, March 28 2017. US Patent 9,609,288.
- Rizzoli, P., Martone, M., Gonzalez, C., Wecklich, C., Tridon, D. B., Bräutigam, B., Bachmann, M., Schulze, D., Fritz, T., Huber, M., et al. Generation and performance assessment of the global tandem-x digital elevation model. *ISPRS Journal of Photogrammetry and Remote Sensing*, 132:119–139, 2017.
- Roberge, V., Tarbouchi, M., and Labonté, G. Comparison of parallel genetic algorithm and particle swarm optimization for real-time uav path planning. *IEEE Transactions on Industrial Informatics*, 9(1):132–141, 2012.
- Rossi, M., Brunelli, D., Adami, A., Lorenzelli, L., Menna, F., and Remondino, F. Gas-drone: Portable gas sensing system on uavs for gas leakage localization. In *SENSORS, 2014 IEEE*, pages 1431–1434. IEEE, 2014.
- Rouse, M. drone services (uav services), 2020. URL <https://internetofthingsagenda.techtarget.com/definition/drone-services-UAV-services>. [Web; accedido el 29-02-2020].
- Rubio-Hervas, J., Gupta, A., and Ong, Y.-S. Data-driven risk assessment and multicriteria optimization of uav operations. *Aerospace Science and Technology*, 77: 510–523, 2018.
- Ruscio, D. D., Malavolta, I., Pelliccione, P., and Tivoli, M. Automatic generation of detailed flight plans from high-level mission descriptions. In *Proceedings of the ACM/IEEE 19th International Conference on Model Driven Engineering Languages and Systems*, pages 45–55, 2016.
- Rusnak, P., Kvassay, M., Zaitseva, E., Kharchenko, V., and Fesenko, H. Reliability assessment of heterogeneous drone fleet with sliding redundancy. In *2019 10th International Conference on Dependable Systems, Services and Technologies (DESSERT)*, pages 19–24. IEEE, 2019.
- Sahingoz, O. K. Flyable path planning for a multi-uav system with genetic algorithms and bezier curves. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 41–48. IEEE, 2013.

- Sanchez-Lopez, J. L., Pestana, J., De La Puente, P., and Campoy, P. A reliable open-source system architecture for the fast designing and prototyping of autonomous multi-uav systems: Simulation and experimentation. *Journal of Intelligent & Robotic Systems*, 84(1-4):779–797, 2016.
- Santamaria, E., Pastor, E., Barrado, C., Prats, X., Royo, P., and Perez, M. Flight plan specification and management for unmanned aircraft systems. *Journal of Intelligent & Robotic Systems*, 67(2):155–181, 2012.
- Santana, L. V., Brandao, A. S., Sarcinelli-Filho, M., and Carelli, R. A trajectory tracking and 3d positioning controller for the ar. drone quadrotor. In *2014 international conference on unmanned aircraft systems (ICUAS)*, pages 756–767. IEEE, 2014.
- Santos, T., Moreira, M., Almeida, J., Dias, A., Martins, A., Dinis, J., Formiga, J., and Silva, E. Plined: Vision-based power lines detection for unmanned aerial vehicles. In *2017 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 253–259. IEEE, 2017.
- Scalea, J. R., Restaino, S., and Scassero, M. The final frontier? exploring organ transportation by drone. *American Journal of Transplantation*, 2019.
- Schäfer, M., Strohmeier, M., Lenders, V., Martinovic, I., and Wilhelm, M. Bringing up opensky: A large-scale ads-b sensor network for research. In *IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks*, pages 83–94. IEEE, 2014.
- SESAR. U-space, 2020a. URL <https://www.sesarju.eu/U-Space>. [Web; accedido el 29-02-2020].
- SESAR. U-space, 2020b. URL <https://www.sesarju.eu/U-Space>. [Web; accedido el 26-02-2020].
- Sharir, A. and Baltsan, A. On shortest paths amidst convex polyhedra. In *Proceedings of the second annual symposium on Computational geometry*, pages 193–206, 1986.
- Shi, W., Arabadjis, G., Bishop, B., Hill, P., Plasse, R., and Yoder, J. Detecting, tracking, and identifying airborne threats with netted sensor fence. In *Sensor Fusion-Foundation and Applications*. InTech, 2011.
- Shin, H., Choi, K., Park, Y., Choi, J., and Kim, Y. Security analysis of fhss-type drone controller. In *International Workshop on Information Security Applications*, pages 240–253. Springer, 2015.

- Simon, S. P. and Mays, W. M. Vehicle and drone management system, January 10 2019. US Patent App. 15/642,027.
- SJU. European drones outlook study. unlocking the value for europe. *SESAR, Brussels*, 2016.
- Sony, S., Laventure, S., and Sadhu, A. A literature review of next-generation smart sensing technology in structural health monitoring. *Structural Control and Health Monitoring*, 26(3):e2321, 2019.
- Stevenson, A. OUP Oxford, 2015.
- Sun, X., Liu, Y., Yao, W., and Qi, N. Triple-stage path prediction algorithm for real-time mission planning of multi-uav. *Electronics Letters*, 51(19):1490–1492, 2015.
- Tang, L. and Shao, G. Drone remote sensing for forestry research and practices. *Journal of Forestry Research*, 26(4):791–797, 2015.
- Tarhan, A. F., Koyuncu, E., Hasanzade, M., Ozdemir, U., and Inalhan, G. Formal intent based flight management system design for unmanned aerial vehicles. In *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 984–992. IEEE, 2014.
- Technology, U. S. Desert rotor, 2020. URL <https://www.unmannedsystemstechnology.com/company/desert-rotor/>. [Web; accedido el 07-03-2020].
- Thomason, J., Ratsamee, P., Kiyokawa, K., Kriangkamol, P., Orlosky, J., Mashita, T., Uranishi, Y., and Takemura, H. Adaptive view management for drone teleoperation in complex 3d structures. In *Proceedings of the 22nd International Conference on Intelligent User Interfaces*, pages 419–426, 2017.
- Torii, T. and Sanada, Y. Radiation measurement by unmanned aircraft after fukushima daiichi nuclear power plant accident. In *Remotely Piloted Aircraft Systems Symposium ICAO Headquarters, Montréal, Canada*, pages 23–25, 2015.
- Tseng, C.-M., Chau, C.-K., Elbassioni, K., and Khonji, M. Autonomous recharging and flight mission planning for battery-operated autonomous drones. *arXiv preprint arXiv:1703.10049*, 2017.
- Tulum, K., Durak, U., and Yder, S. K. Situation aware uav mission route planning. In *2009 IEEE Aerospace conference*, pages 1–12. IEEE, 2009.

- Unifly. Unifly, 2020. URL <https://www.unifly.aero/>. [Web; accedido el 09-03-2020].
- Vale, A., Ventura, R., and Carvalho, P. Application of unmanned aerial vehicles for radiological inspection. *Fusion Engineering and Design*, 124:492–495, 2017.
- Valenti, M., Schouwenaars, T., Kuwata, Y., Feron, E., How, J., and Paunicka, J. Implementation of a manned vehicle-uav mission system. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, page 5142, 2004.
- Van Kreveland, M., Schwarzkopf, O., de Berg, M., and Overmars, M. *Computational geometry algorithms and applications*. Springer, 2000.
- Ventura, D., Bruno, M., Lasinio, G. J., Belluscio, A., and Ardizzone, G. A low-cost drone based application for identifying and mapping of coastal fish nursery grounds. *Estuarine, Coastal and Shelf Science*, 171:85–98, 2016.
- Vilaplana, M. A., Gallo, E., Navarro, F. A., and Swierstra, S. Towards a formal language for the common description of aircraft intent. In *24th Digital Avionics Systems Conference*, volume 1, pages 3–C. IEEE, 2005.
- Waller, M., Rigopoulos, J., Blackman, D., and Berreen, T. Considerations in the application of dynamic programming to optimal aircraft trajectory generation. In *IEEE Conference on Aerospace and Electronics*, pages 574–579. IEEE, 1990.
- Welzl, E. Constructing the visibility graph for n-line segments in $O(n^2)$ time. *Information Processing Letters*, 20(4):167–171, 1985.
- Wolf, H. G. *Drones: Safety risk management for the next evolution of flight*. Taylor & Francis, 2017.
- Xiao-dong, C., De-yun, Z., and Ruo-nan, Z. New method for uav online path planning. In *2013 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC 2013)*, pages 1–5. IEEE, 2013.
- Yoo, S., Kim, K., Jung, J., Chung, A. Y., Lee, J., Lee, S. K., Lee, H. K., and Kim, H. Poster: A multi-drone platform for empowering drones’ teamwork. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 275–277, 2015.
- Zeng, W. and Church, R. L. Finding shortest paths on real road networks: the case for a. *International journal of geographical information science*, 23(4):531–543, 2009.

REFERENCES

- Zipline. Zipline, 2017, 2017. URL <http://www.flyzipline.com>. [Web; accedido el 27-02-2020].
- Zorbas, D., Pugliese, L. D. P., Razafindralambo, T., and Guerriero, F. Optimal drone placement and cost-efficient target coverage. *Journal of Network and Computer Applications*, 75:16–31, 2016.