



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

Diseño y desarrollo de una herramienta para la resolución
de conflictos estratégicos entre aeronaves no tripuladas

Trabajo Fin de Grado

Grado en Ingeniería Aeroespacial

AUTOR/A: Calatayud García, Víctor

Tutor/a: Balbastre Tejedor, Juan Vicente

Cotutor/a: Claramunt Puchol, Cecilia

CURSO ACADÉMICO: 2022/2023

TRABAJO DE FIN DE GRADO

**Diseño y desarrollo de una herramienta
para la resolución de conflictos
estratégicos entre aeronaves no
tripuladas**



Alumno: Víctor Calatayud Garcia

Grado en ingeniería aeroespacial

Año: 2023

Tutor: Juan Vicente Balbastre Tejedor

Cotutora: Cecilia Claramunt Puchol

Índice de Contenidos

I. MEMORIA	XI
1 Introducción	1
2 Objetivos y alcance	2
2.1 Relación del presente trabajo con los objetivos de desarrollo sostenible	2
3 Revisión bibliográfica	5
3.1 Regulación y estándares	5
3.1.1 CIR 2021/664	5
3.1.1.1 Servicio de identificación de red	5
3.1.1.2 Servicio de geoconsciencia	6
3.1.1.3 Servicio de autorización de vuelo de UAS	6
3.1.1.4 Servicio de información sobre el tráfico	7
3.1.1.5 Servicio de información meteorológica	7
3.1.1.6 Servicio de supervisión de la conformidad	7
3.1.2 AMC&GM a la CIR 2021/664	8
3.1.2.1 GM7 Artículo 3(4) espacio aéreo U-space	8
3.1.2.2 GM8 Artículo 3(4) espacio aéreo U-space	8
3.1.2.3 GM1 Artículo 10 servicio de autorización de vuelo de UAS	9
3.1.2.4 AMC2 Artículo 10(1) servicio de autorización de vuelo de UAS	9
3.1.2.5 GM1 Artículo 10(2) servicio de autorización de vuelo de UAS	10
3.1.2.6 AMC1 Artículo 10(2)(c) servicio de autorización de vuelo de UAS	10
3.1.2.7 AMC1 Artículo 10(5) servicio de autorización de vuelo de UAS	10
3.1.2.8 GM1-3 Artículo 10(5) servicio de autorización de vuelo de UAS	11
3.1.3 ASTM F3548-21	11
3.1.3.1 Estados de un <i>U-plan</i>	12
3.1.3.2 Requisitos de los USSPs	13
3.1.3.3 Requisitos de interoperabilidad y Discovery Synchronization Service (DSS)	15
3.1.4 CIR 2019/947	18
3.1.4.1 Categoría abierta	18
3.1.4.2 Categoría específica	19
3.1.4.3 Categoría certificada	19
4 Proyectos SESAR	20
4.1.1 CORUS	20
4.1.1.1 Servicios U-space	20
4.1.1.2 Volúmenes U-space	23
4.1.2 BUBBLES	25
4.1.2.1 Definición de conflicto según BUBBLES	25
4.1.2.2 Definición de escenarios	26
4.1.2.3 Proceso de gestión de la separación	27
4.1.3 USEPE	29
4.1.3.1 Pasillos de drones	29
4.1.3.2 Gestión del espacio aéreo basado en la densidad	30
4.1.3.3 Geovectores	31
4.1.3.4 Relacionar la estructura del espacio aéreo y la capacidad	32
4.1.3.5 Concepto de pasillos de densidad dinámica	34
4.1.4 SPATIO	35
4.1.4.1 Modelo SPATIO	36

5	Desafíos en el desarrollo de herramientas para la detección de conflictos estratégicos entre UAS	38
5.1	Intersección de trayectorias 4D: complejidad y necesidad de discretización	38
5.1.1	Enfoque estadístico	39
5.1.2	Enfoque seguro	40
5.2	Cantidad de comprobaciones	41
5.2.1	Estructuras de datos	43
5.2.1.1	Grid	44
5.2.1.2	Octree	44
5.2.1.3	R-Tree	45
5.2.1.4	Kd-Tree	46
5.2.1.5	Selección de la estructura de datos óptima para el SCRS.	47
5.3	Tiempo de protección	48
5.3.1	Factor dt-Tiempo de protección	50
5.3.2	Disminución de la capacidad	54
6	Implementación	60
6.1	Diagrama de Clases UML	60
6.2	SCRS	63
6.2.1	Atributos	63
6.2.2	Métodos	63
6.2.2.1	InsertUplan	63
6.2.2.2	deconflict	64
6.2.2.3	activate	65
6.2.2.4	searchNeighbors	66
6.2.2.5	Otros métodos	68
6.3	Clase Uplan	69
6.3.1	Definición del U-plan	69
6.3.2	Atributos	71
6.3.3	Metodos de la clase	71
6.4	Clase WayPoint	72
6.4.1	Atributos de la clase	72
6.4.2	Método	73
6.5	Clase Octree	73
6.5.1	Atributos	74
6.5.2	Métodos	75
6.5.2.1	Constructor	75
6.5.2.2	Insert	75
6.5.2.3	Divide	76
6.5.2.4	Find	77
6.5.2.5	Otros métodos	79
6.5.2.6	Resumen de los métodos implementados en la clase octTree	80
6.6	Análisis del código	81
7	Resultados	84
7.1	Definición de los escenarios y el equipo utilizado	84
7.2	Exposición y análisis de resultados	87

7.2.1	Ratio de conflictos detectados _____	87
7.2.2	Ratio de activaciones en menos de cinco segundos _____	88
8	Conclusiones _____	91
9	Trabajo futuro/siguientes pasos _____	92
10	Referencias _____	93
II.	PRESUPUESTO _____	XII
III.	PLIEGO DE CONDICIONES _____	XV

Índice de Figuras

Figura 2.1 Lista objetivos de desarrollo sostenible	3
Figura 3.1 Volumen de protección UAS	9
Figura 3.2 Propósitos operativos	12
Figura 3.3. Diagrama de estados de un U-plan	13
Figura 3.4 Paradigma de interoperabilidad	16
Figura 3.5 Uso de OVNs.....	17
Figura 4.1 Conflicto según la definición de BUBBLES (Izquierda: vista en planta, derecha: vista alzada).....	26
Figura 4.2 Clasificación de tráfico según BUBBLES	27
Figura 4.3 Proceso de provisión de separación por BUBBLES	27
Figura 4.4 Servicio de gestión de separación a nivel pre-operacional	28
Figura 4.5 Ejemplo de distribución de "Pasillo de drones"	30
Figura 4.6 Ejemplo de volumen de seguridad.....	31
Figura 4.7 Ejemplo de Geovector.....	32
Figura 4.8 Ejemplo de estructura en capas.....	33
Figura 4.9 Ejemplo de estructura en zonas.....	33
Figura 4.10 Ejemplo de estructura de tubos	34
Figura 4.11 Disposición del espacio aéreo para el método D2-C2.....	35
Figura 4.12 Relación entre los servicios propuestos por SPATIO	37
Figura 5.1 LoS indetectable debido a la discretización temporal.....	39
Figura 5.2 Enfoque seguro sobre las pérdidas de separación debido a la discretización temporal.....	41
Figura 5.3 Variación de las operaciones requeridas en función del número de <i>U-plan</i> activos para distintos tiempos.....	43
Figura 5.4 Estructura de datos en "Grid".....	44
Figura 5.5 Diagrama octree.....	45
Figura 5.6 Diagrama esquemático de la estructura de datos R-tree	46
Figura 5.7 Diagrama kd-tree	47
Figura 5.8 comparación del orden de "octree" vs fuerza bruta	48
Figura 5.9 Volumen de protección extendido por el tiempo de protección	49
Figura 5.10 Esquema de la técnica de puntos virtuales	50
Figura 5.11 Aumento de puntos debido al tiempo de protección.....	51
Figura 5.12 Diagrama de la técnica de puntos virtuales aplicada a octree.....	53

Figura 5.13 Diagrama área de intersección de dos circunferencias	56
Figura 5.14 incremento de volumen debido a el tiempo de protección, el factor dt-tiempo de protección y dt	58
Figura 5.15 incremento de volumen con el tiempo de protección	59
Figura 5.16 Incremento de volumen debido al factor y dt.....	59
Figura 6.1 Diagrama de clases servicio propuesto	62
Figura 6.2 Diagrama de flujo del método InsertUplan	64
Figura 6.3 Diagrama de flujo del metodo deconflict	65
Figura 6.4 Diagrama de flujo del método activate.....	66
Figura 6.5 Diagrama de flujo searchNeighbors.....	67
Figura 6.6 División de los nodos tras realizar una inserción	74
Figura 6.7 Diagrama de flujo del método insert.....	76
Figura 6.8 Diagrama de flujo del método Find	78
Figura 6.9 Proceso del método Find	79
Figura 6.10 Resultado del analisis de "profiler" de MATLAB.....	82
Figura 6.11 Resultado del analisis de "profiler" de el método octree.Insert	83
Figura 7.1 Zona de simulación	85
Figura 7.2 Ejemplo estado del octree	86
Figura 7.3 porcentaje de conflictos detectados en diferentes escenarios y dt	88

Índice de Tablas

Tabla 3-1 Requisitos de los <i>USSPs</i>	13
Tabla 4-1. Servicios propuestos por CORUS	21
Tabla 6-1 Resumen métodos de la clase SCRS.....	68
Tabla 6-2 Descripción de los campos de un U-plan	69
Tabla 6-3 Resumen de los métodos de la clase octTree	80
Tabla 7-1 rango de valores para la creación de los U-plans.....	84
Tabla 7-2 definición de los escenarios de simulación.....	85
Tabla 7-3 Separación horizontal y vertical por categoría de operación	86
Tabla 7-4 equipos utilizados para la simulación	87
Tabla 7-5 Porcentaje de activaciones en menos de 5 segundos en diferentes escenarios y dt equipo 1	88
Tabla 7-6 Porcentaje de activaciones en menos de 5 segundos en diferentes escenarios y dt equipo 1	89
Tabla 7-7 porcentaje de activaciones en menos de 5 segundos con dt=2 s en el escenario 4.....	89
Tabla 10-1 Coste de personal.....	XII
Tabla 10-2 Coste de materiales.....	XIII
Tabla 10-3 Costes indirectos.....	XIII
Tabla 10-4 Presupuesto total del proyecto.....	XIV
Tabla 10-5 instrucciones para modificar los parametros de la herammienta.....	XV

Glosario

AMC	Acceptable Means of Compliance
ATM	Air Traffic Management
CIR	Commission Implementing Regulation
CIS	Common Information Service
CMS	Conformance Monitoring Service
CNS	Communication, Navigation and Surveillance
D2-C2	Dynamic Density Corridor Concept
DAA	Detect And Avoid
DCB	Demand and Capacity Balancing
DCMS	Dynamic Capacity Management Service
DSS	Discovery and Synchronization Service
EVLOS	Extended Visual Line Of Sight
FAS	Flight Authorisation Service
FI-FS	First In First Served
FTE	Flight Technical Error
GAS	Geo-Awareness Service
GM	Guidance Material
GS	Ground Speed
LoS	Loss of Separation
NIS	Network Identification Service
NSE	Navigation System Error
OBJ	Objectives
OVN	Opaque Version Number

PDE	Path Definition Error
RoC	Rate of Climb
SC	Success Criteria
SCPS	Strategic Conflict Prediction Service
SCRS	Strategic Conflict Resolution Service
SMS	Separation Management Service
SPR/INTEROP	Safety, Performance, and Interoperability Requirements
TCRS	Tactical Conflict Resolution Service
TIS	Traffic Information Service
TLS	Target Level of Safety
TRL	Technology Readiness Level
TSE	Total System Error
UAS	Unmanned Aircraft Systems
USSP	U-space Service Provider
VFR	Visual Flight Rules
VLL	Very Low Level airspace
VLOS	Visual Line Of Sight
VS	Vertical Speed
WIS	Weather Information Service

Agradecimientos

Antes de empezar este Trabajo de Fin de Grado, me gustaría tomar un momento para expresar mi más profundo agradecimiento a todas las personas que han hecho posible la realización de este proyecto.

En primer lugar, me gustaría agradecer a mis tutores Cecilia Claramunt y Juan Vicente Balbastre, por haberme dado la oportunidad de trabajar en SNA donde he adquirido una experiencia valiosísima a la vez que he estado en un entorno de trabajo ideal. Muchas gracias por todo, habéis sido más que tutores.

También me gustaría agradecer a mis amigos. En primer lugar, a “amoriños”, junto a vosotros es uno de los pocos sitios donde me siento completamente cómodo, pase el tiempo que pase y me lleve la vida donde me lleve sé que con vosotros me sentiré como en casa. A los “@9”, siento que la vida nos debe mucho, por mucho tiempo que pase con vosotros nunca parece ser suficiente, sois mucho más que simples amigos de universidad, sé que habéis llegado para quedaros, ¡que lo que ha unido la carrera no lo separe nada! Por último, una dedicación especial a Elena, por aguantarme en el piso y llevarme recto en los estudios, hemos sido inseparables, te echaré de menos.

Por último, me voy a tomar la libertad de agradecer a los más importantes en mi lengua natal, *Mama, Papa, Diego, no hi ha paraules suficients que puga dir per a agrair-vos tot el que heu fet per mí, no podria demanar una millor familia, el suport que m'heu donat en tots els sentits i en tots els moments és insuperable. Sé que no ho dic tot el que deuria pero vos vull. I no m'oblidge de vosaltres, iaaios, em senc tot un afortunat tenint-vos encara per ací, moltes gràcies per haver cuidat de mi en l'infància i estar ahí encara per a tot, i disculpeu-me per no ser el net mes atent que podieu tindre.*

Cada uno de vosotros ha jugado un papel crucial en mi camino y este logro no solo es mío, sino también vuestro. Os agradezco de todo corazón por ser parte de este importante capítulo de mi vida. Este Trabajo de Fin de Grado es el resultado de muchas horas de esfuerzo y dedicación, pero, sobre todo, es el resultado de la amistad, el apoyo y el amor que he recibido de vosotros. Muchas gracias a todos.

Resum

S'anticipa un increment exponencial en la utilització d'aeronaus no tripulades en entorns urbans en els propers anys. Aquest document, explora l'aplicació de l'estructura de dades octree per a resoldre conflictes estratègics en aquest escenari, presentant una nova tècnica que introdueix un període de protecció per a l'aeronau. A més, s'examina l'impacte d'aquesta tècnica en la capacitat de l'espai aeri. Per avaluar els límits de l'eina proposada, s'han simulat diferents escenaris.

Paraules clau: Aeronaus no tripulades, conflictes estratègics, octree, U space.

Resumen

Se anticipa un incremento exponencial en la utilización de aeronaves no tripuladas en entornos urbanos en los próximos años. Este documento explora la aplicación de la estructura de datos octree para resolver conflictos estratégicos en este escenario, presentando una nueva técnica que introduce un periodo de protección para la aeronave. Además, se examina el impacto de esta técnica en la capacidad del espacio aéreo. Para evaluar los límites de la herramienta propuesta, se han simulado diferentes escenarios.

Palabras clave: Aeronaves no tripuladas, conflictos estratégicos, octree, U-space.

Abstract

An exponential increase in the use of unmanned aircraft in urban environments is anticipated in the coming years. This document explores the application of the octree data structure to resolve strategic conflicts in this scenario, introducing a new technique that establishes a protection period for the aircraft. Additionally, the impact of this technique on airspace capacity is examined. Different scenarios have been simulated to evaluate the limits of the proposed tool.

Keywords: Unmanned aircraft, strategic conflicts, octree, U space.

I. MEMORIA

1 INTRODUCCIÓN

El avance vertiginoso de la tecnología en las últimas décadas ha generado un amplio abanico de oportunidades y retos en diversos campos de estudio y aplicación. Un ámbito particular que ha experimentado una transformación significativa es el de la aviación, con el surgimiento y proliferación de las aeronaves no tripuladas, más conocidas como drones o UAS, por sus siglas en inglés. A medida que estos dispositivos se vuelven más sofisticados y accesibles, la necesidad de una gestión adecuada y eficaz de su uso se ha vuelto cada vez más imperativa.

El presente Trabajo de Fin de Grado, aborda la problemática emergente que supone el creciente tráfico de drones y la necesidad de mecanismos efectivos para resolver los conflictos que pueden surgir entre ellos y otros tráficos.

Las aeronaves no tripuladas han demostrado su utilidad en un sinfín de aplicaciones, desde la agricultura y la inspección de infraestructuras hasta la entrega de paquetes y la seguridad pública. Sin embargo, con este aumento en la utilización de los drones, también se incrementa la probabilidad de conflictos en el espacio aéreo, especialmente en áreas urbanas y suburbanas densamente pobladas donde la seguridad aérea y ciudadana es crucial. Estos conflictos pueden variar desde simples interferencias de trayectorias hasta situaciones más peligrosas que pueden poner en peligro la seguridad de las personas, infraestructuras y propiedades en el suelo.

Por lo tanto, es esencial desarrollar una herramienta que pueda gestionar y resolver estos conflictos de manera efectiva. El diseño y desarrollo de dicha herramienta para la fase estratégica, es decir, previa al vuelo, es el objetivo principal de este trabajo, que pretende contribuir a un uso más seguro y eficiente de las aeronaves no tripuladas en el futuro.

En este TFG, se analizarán a fondo las características y requisitos necesarios para diseñar una herramienta que pueda identificar, prever y resolver conflictos entre aeronaves no tripuladas de manera efectiva en fase estratégica. Además, se desarrollará un prototipo funcional que será probado y evaluado para determinar su rendimiento y eficacia.

Este documento está dividido en varias secciones, en primer lugar, realizaremos una revisión bibliográfica analizando los proyectos existentes y el marco regulatorio y jurídico respecto a la gestión de conflictos y la separación entre drones. A continuación, expondremos los desafíos a los que nos hemos enfrentado en el desarrollo de una herramienta para la detección de conflictos estratégicos, para continuar hablando de la implementación final y los resultados obtenidos. Con todo esto realizaremos una conclusión del trabajo realizado y sugeriremos posibles trabajos futuros en relación al TFG.

A través de este proyecto, aspiramos a profundizar en la comprensión de los desafíos asociados con el uso creciente de aeronaves no tripuladas y a contribuir al desarrollo de soluciones innovadoras y efectivas que permitan su uso seguro y eficiente.

2 OBJETIVOS Y ALCANCE

Este Trabajo de Fin de Grado (TFG) tiene como objetivo principal el diseño y desarrollo de una herramienta para la resolución de conflictos estratégicos entre aeronaves no tripuladas. Este objetivo se alcanzará a través de la consecución de varios objetivos específicos que definen el alcance de este trabajo.

- **Investigación y análisis:** El primer objetivo específico es la comprensión profunda de la problemática actual en la gestión de conflictos entre aeronaves no tripuladas. Este objetivo se logrará a través de una revisión de la literatura, investigaciones existentes, tecnologías actuales y análisis de las normativas relacionadas con las aeronaves no tripuladas.
- **Diseño de la solución y del formato de plan de vuelo:** Una vez comprendido el panorama actual, el objetivo es diseñar una solución teórica para la resolución de conflictos y definir el formato y estructura que debe tener un plan de vuelo.
- **Desarrollo de la herramienta:** Con un diseño de solución en la mano, el siguiente objetivo es transformar este modelo conceptual y diseño de plan de vuelo en una herramienta funcional. El desarrollo de la herramienta se realizará utilizando técnicas y tecnologías apropiadas, con un enfoque limitado en la resolución de conflictos. Esta herramienta se centrará en aceptar o rechazar los planes de vuelo, en caso de que causen conflicto. En este segundo caso, la herramienta notificará la causa del rechazo ofreciéndole información sobre el punto de la trayectoria deseada que conflictúa con otra en espacio y tiempo.
- **Pruebas y evaluación:** Una vez desarrollada la herramienta, será probada con planes de vuelo. Hay que tener en cuenta que la construcción de trayectorias realistas no está dentro del alcance de este TFG y que por tanto las pruebas se realizarán con trayectorias aleatorias. La evaluación de la herramienta se realizará mediante medidas cuantitativas y cualitativas para validar su eficacia y eficiencia.
- **Optimización de la herramienta:** En base a los resultados obtenidos durante las pruebas y la evaluación, se llevarán a cabo las modificaciones necesarias para optimizar el rendimiento de la herramienta.

Es importante destacar que este TFG no incluye la implementación de la herramienta en un entorno real de aeronaves no tripuladas, ni tampoco su integración con otros servicios de gestión de tráfico no tripulado o su comercialización. El alcance se limita a la investigación, diseño, desarrollo, pruebas y optimización de la herramienta. Las conclusiones y recomendaciones proporcionadas estarán basadas en los hallazgos obtenidos durante estas fases.

Además de estos objetivos propio, el proyecto esta alineado con los objetivos de desarrollo sostenible, en el siguiente apartado analizaremos su relación.

2.1 Relación del presente trabajo con los objetivos de desarrollo sostenible

Los Objetivos de Desarrollo Sostenible (ODS) son una colección de 17 objetivos globales

establecidos por la Asamblea General de las Naciones Unidas en 2015, como parte de la Agenda 2030 para el Desarrollo Sostenible. Estos objetivos abarcan una amplia gama de temas sociales, económicos y ambientales. Se espera que todos los países adopten estas metas y trabajen para alcanzarlas para el año 2030.



Figura 2.1 Lista objetivos de desarrollo sostenible

El uso y desarrollo de tecnologías para aeronaves no tripuladas, en concreto en el desarrollo de las herramientas de para la detección y resolución de conflictos entre estas aeronaves tienen el potencial para influir en los siguientes ODS:

- **ODS 9: Industria, Innovación e Infraestructura.** La creación de una herramienta de este tipo se alinea con la promoción de la infraestructura tecnológica y la innovación. Este proyecto contribuye a la modernización de la industria aeronáutica, en concreto, al uso de drones en entornos urbanos y alienta el desarrollo de tecnologías útiles y sostenibles como es el U-space.
- **ODS 11: Ciudades y Comunidades Sostenibles.** El uso de drones en entornos urbanos tiene un impacto positivo en la sostenibilidad ambiental, económica y social. En el primer caso, los drones, en su mayoría eléctricos, reemplazarían parte del transporte terrestre, reduciendo así la contaminación por gases y el nivel de ruido en la calle. Esto mejoraría la calidad del aire y disminuiría la huella de carbono, lo cual impacta también en el ODS 13. En cuanto a sostenibilidad social, los drones agilizarían el transporte de mercancías y suministros médicos, incluso en zonas rurales de difícil acceso, mejorando la calidad de vida de las personas. Además, la implantación del U-space, del cual forma parte la herramienta planteada en este trabajo, haría que el uso de drones sea más seguro y eficiente, viéndose reducido el riesgo de vidas humanas e infraestructuras. Por último, en cuanto a la sostenibilidad económica, el uso de drones crearía nuevos puestos de

trabajo y nuevos nichos de mercado, además de que reduciría costes en caso de tratarse de transporte de mercancías.

- **ODS 13: Acción por el Clima.** Al minimizar las colisiones y maximizar la eficiencia del uso de aeronaves no tripuladas, es posible reducir las emisiones de carbono, contribuyendo así a la lucha contra el cambio climático.

El papel de la tecnología en la aviación está cambiando rápidamente y, por lo tanto, es vital que desarrollemos herramientas que no sólo resuelvan los problemas inmediatos, sino que también tengan en cuenta las implicaciones a largo plazo para la sostenibilidad. Al alinear el diseño y desarrollo de esta herramienta con los ODS, podemos asegurarnos de que estamos contribuyendo a un futuro más sostenible y equitativo.

3 REVISIÓN BIBLIOGRÁFICA

En esta sección, se realiza un recorrido por las principales regulaciones y proyectos que han abordado el uso creciente de *UAS* (Unmanned Aircraft Systems) y su gestión. Esta revisión bibliográfica proporciona un marco teórico y conceptual que sirve de base para el desarrollo de la herramienta propuesta en este trabajo.

3.1 Regulación y estándares

Dada la naturaleza emergente y dinámica del uso de tecnología de drones, la regulación y los estándares pertinentes se encuentran en un estado de constante evolución y pueden resultar insuficientes para abordar los desafíos actuales. A pesar de este panorama incierto y fluctuante, en este apartado nos enfocaremos en analizar y discutir las regulaciones y estándares existentes hasta la fecha.

3.1.1 CIR 2021/664

El Reglamento de Ejecución (UE) 2021/664 de la Comisión, emitido el 22 de abril de 2021, establece el marco regulador para el U-space.

La regulación define el U-space como: “zona geográfica de UAS designada por los Estados miembros, en la que solo se permite que se lleven a cabo operaciones de UAS con el apoyo de servicios de U-Space”[[1]-Capítulo 1 artículo 2.(1)] y a su vez define un servicio U-space como “servicio basado en servicios digitales y automatización de funciones diseñados para facilitar un acceso protegido, eficiente y seguro al espacio aéreo U-space para un gran número de UAS ”[[1]-Capítulo 1 artículo 2.(2)]. Los servicios U-space, que se definen en el capítulo 4 de la regulación[[1]-Capítulo 3 artículos 8-13], se describen a continuación:

3.1.1.1 Servicio de identificación de red

El servicio de identificación de red o *NIS* por sus siglas en inglés es el encargado de:

1. **Permitir** la identificación remota y continua del UAS durante todo el vuelo y proporcionar la identificación del UAS a los usuarios autorizados. Los usuarios autorizados del NIS son:
 - a. El público en general,
 - b. Otros proveedores de servicios U-space para garantizar la seguridad de las operaciones en el espacio aéreo U-space,
 - c. Los proveedores de servicios de tráfico aéreo correspondientes,
 - d. Cuando se designe, el proveedor único de servicios de información común,
 - e. Las autoridades competentes relevantes responsables de la regulación de los UAS y su supervisión.
2. **Permitir** que los usuarios autorizados reciban mensajes con la siguiente información:
 - a. Número de registro del operador del UAS.
 - b. Número de serie único de la aeronave no tripulada o, el número de serie

- único del accesorio.
- c. La posición geográfica del *UAS*.
 - d. la trayectoria medida en el sentido de las agujas del reloj a partir del norte geográfico y la velocidad del *UAS* respecto al suelo.
 - e. La posición geográfica del piloto remoto o, el punto de despegue.
 - f. El estado de emergencia del *UAS*.
 - g. La hora en la que se generaron los mensajes.
3. **Actualizar** la información con una frecuencia determinada por la autoridad competente.

3.1.1.2 Servicio de geoconsciencia

Este servicio, abreviado por sus siglas en inglés como *GAS*, debe proporcionar a los operadores de *UAS*:

1. Información sobre las condiciones y limitaciones del espacio aéreo U-space,
2. las zonas geográficas de *UAS* que son pertinentes para el espacio aéreo U-space,
3. las restricciones temporales aplicables a la utilización del espacio aéreo dentro del espacio aéreo U-space.

Los *USSPs* (U-space service providers) deberán notificar a los operadores con la información de forma rápida incluyendo la hora de actualización y la hora de validez para que los operadores puedan hacer frente a las contingencias y emergencias.

3.1.1.3 Servicio de autorización de vuelo de *UAS*

El *FAS* (por sus siglas en inglés) juega un papel crucial en la gestión del espacio aéreo dedicado al vuelo de aviación no tripulada. El *FAS* es responsable de autorizar o denegar cada vuelo de un *UAS*, basándose en una serie de factores que incluyen la seguridad, las condiciones del espacio aéreo y las capacidades del *UAS*. Es precisamente para este servicio que estamos desarrollando herramientas de detección y resolución de conflictos estratégicos en el presente trabajo.

Ante una solicitud de autorización de vuelo, el *FAS* realiza los siguientes pasos:

1. **Comprobar** si la solicitud de la autorización es correcta y de conformidad con el anexo IV de la CIR 2021/664.
2. **Aceptar** la solicitud de autorización de vuelo de *UAS* si el vuelo no interseca en espacio y tiempo con otro vuelo ya autorizado de acuerdo con las reglas de prioridad.
3. **Notificar** al operador sobre la aceptación o rechazo de la autorización.
4. **Indicar** al operador los umbrales de desviación permitidos para la autorización de vuelo.

Además, el *FAS*:

- **Debe** utilizar la información del *WIS* (Weather information service) cuando proceda.
- **Puede** proponer una alternativa al operador cuando la autorización sea denegada.

- Una vez recibida la solicitud de activación de la autorización de vuelo de *UAS*, **debe** confirmar la activación de la autorización de vuelo de *UAS* sin retraso injustificado¹.
- **Debe** establecer disposiciones adecuadas para resolver incompatibilidades entre solicitudes de autorización de vuelo de *UAS* de distintos *USSPs*.
- **Debe** comprobar que las restricciones del U-space no sean violadas por los vuelos.
- **Debe** tramitar las solicitudes según la política de prioridades establecida en el reglamento de ejecución UE 923/2012 [3], y ante la misma prioridad dará la autorización al que antes la haya solicitado².
- **Debe** actualizar o retirar las autorizaciones en función de las circunstancias de restricciones y limitaciones dinámicas del espacio aéreo y la información de tráfico aéreo tripulado.

3.1.1.4 Servicio de información sobre el tráfico

Abreviado como *TIS*, este servicio debe proporcionar a los operadores de *UAS* información sobre otros tráficos que puedan estar cerca de la posición o ruta prevista del *UAS*, incluyendo tráfico de aeronaves tripuladas y *UAS* de otros *USSPs*. Esta información debe actualizarse con frecuencia e incluir la posición (junto con su hora de notificación), velocidad, rumbo y estado de emergencia (cuando se conozca) de todas las aeronaves próximas.

3.1.1.5 Servicio de información meteorológica

Abreviado como *WIS*, este servicio debe recopilar los datos meteorológicos e informar a los operadores y otros servicios de estos datos, que incluirán **como mínimo**:

- Dirección y velocidad del **viento**, incluyendo ráfagas,
- Altura de la capa más baja de **nubes**,
- **Visibilidad**,
- **Temperatura** y punto de rocío,
- Indicadores de actividad y **precipitación** convectivas,
- **Horas** y **ubicaciones** válidas de la previsión, o, el lugar y hora de las observaciones,
- la altura sobre el nivel del mar (**QNH**), con localización geográfica de su aplicabilidad.

3.1.1.6 Servicio de supervisión de la conformidad

El *CMS*, por sus siglas en inglés, permite a los operadores verificar si cumplen los requisitos y los términos de su autorización de vuelo aceptada previamente por el FAS. Cuando el *CMS* detecte una desviación respecto a la ruta descrita en la autorización de vuelo, debe alertar al operador del *UAS*, así como a otros operadores y otros *USSPs* o *ATS* que se puedan verse afectados por el *UAS* fuera de conformidad.

¹ El concepto de retraso justificado se contempla en el estándar ASTM F3548-21. [2]

² Pese a que la regulación establezca esto, se prevé que sea temporal, y en el futuro se establecerán algoritmos de decisión más justos.

3.1.2 AMC&GM a la CIR 2021/664

En el marco del creciente uso de drones y la necesidad de gestión eficaz del espacio aéreo, el documento " *Acceptable Means of Compliance and Guidance Material to Regulation (EU) 2021/664 on a regulatory framework for the U-space*" [4] de la Agencia de la Unión Europea para la Seguridad Aérea (EASA) ofrece directrices y aclaraciones valiosas para la correcta interpretación y aplicación de la CIR 2021/664. El desarrollo de este documento tiene por objetivo mitigar el riesgo de colisión, permitir un uso eficiente y equitativo del espacio aéreo, y facilitar operaciones de drones seguras y complejas. En este apartado, nos enfocaremos en las partes del documento que afectan al desarrollo de la herramienta de resolución de conflictos estratégicos.

3.1.2.1 GM7 Artículo 3(4) espacio aéreo U-space

El GM7 indica que los Estados Miembros pueden establecer ciertas restricciones para garantizar la eficiencia y la equidad en el acceso al espacio aéreo U-space. Las restricciones pueden ser:

- El tiempo mínimo y máximo (tamaño de la ventana de tiempo) antes de la hora de despegue programada en la que se solicita la activación del vuelo.
- El tiempo máximo de antelación con que puede enviarse una solicitud de autorización de vuelo para garantizar la implementación efectiva del principio de FI-FS (first in, first serve)³ y prevenir la ocupación indebida del espacio aéreo U-space.

Este artículo será relevante porque, como veremos en futuros apartados, el tamaño de la ventana de tiempo es crucial para el servicio de resolución de conflictos estratégicos. El tamaño de la ventana temporal que se considere impactará en los tiempos de cómputo de la herramienta, así como en la capacidad y eficiencia del espacio aéreo.

3.1.2.2 GM8 Artículo 3(4) espacio aéreo U-space

La autorización de vuelo de UAS describe la trayectoria de vuelo como una serie de uno o más volúmenes 4D expresados en límites de altura (base, techo), longitud y lateral, y duración (tiempos de entrada y salida). Cada dimensión incluye la incertidumbre del vuelo, considerando el rendimiento operativo de UAS y las suposiciones sobre la competencia del operador y las condiciones meteorológicas. Se recomienda como mejor práctica que estas incertidumbres estén limitadas en la probabilidad dada del 95%.

Con esta definición de volúmenes, se concluye que por cada punto de la trayectoria existirá un volumen en forma cilíndrica, con una dimensión extra indicando el tiempo en el que es posible que el UAS se encuentre dentro de este cilindro, tal y como se ve en la Figura 3.1, la intersección de estos cilindros conformará la trayectoria de vuelo.

³ Ante una misma prioridad de operación en dos autorizaciones que se solapen en espacio y/o tiempo, la legislación establece que el primer operador en enviar la solicitud será al que se le otorgue la autorización.

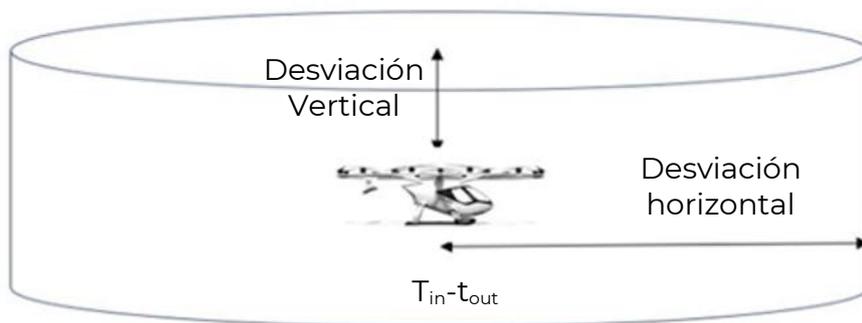


Figura 3.1 Volumen de protección UAS

3.1.2.3 GM1 Artículo 10 servicio de autorización de vuelo de UAS

El FAS proporciona autorizaciones a los operadores de UAS para cada vuelo individual, teniendo en cuenta otras solicitudes de vuelo notificadas que puedan entrar en conflicto en el mismo espacio aéreo. Este servicio se utiliza como herramienta estratégica para evitar conflictos y garantizar la seguridad y requiere que los operadores presenten una solicitud de autorización de vuelo antes de cada operación.

El servicio FAS será obligatorio en el espacio aéreo designado como U-space y se proporcionará a todos los operadores de UAS.

Además, el FAS también realiza una última comprobación de intersección en espacio y tiempo con otras autorizaciones o tráfico cuando los operadores de UAS activan su autorización de vuelo. La finalización del vuelo detiene la provisión de este servicio. En caso de que haya múltiples USSPs en un mismo espacio aéreo, se debe realizar un intercambio de solicitudes de autorización de vuelo y comunicar cualquier cambio en dichas solicitudes.

3.1.2.4 AMC2 Artículo 10(1) servicio de autorización de vuelo de UAS

Tras aceptar una autorización de vuelo, el proveedor de servicios USSP debe establecer unos términos y condiciones para dicha autorización que incluyan:

- Una cláusula de recordatorio sobre las condiciones aplicables y las restricciones del espacio aéreo.
- Los requisitos técnicos, como los requisitos de rendimiento necesarios para el UAS.
- Una lista de permisos necesarios para volar en espacios aéreos restringidos (por ejemplo, zonas geográficas de acceso limitado).
- Instrucciones detalladas sobre cómo manejar las solicitudes de autorización de vuelo y de activación, así como las limitaciones, como el plazo de tiempo para activar o desactivar el vuelo.

3.1.2.5 GM1 Artículo 10(2) servicio de autorización de vuelo de UAS

- El *FAS* es un mecanismo de resolución de conflictos y autoriza vuelos que no se intersecan con otras autorizaciones de vuelo.
- La solicitud de autorización de vuelo de UAS describe la trayectoria de vuelo como una serie de uno o más volúmenes 4D expresados en altura, límites longitudinales y laterales, y duración (tiempos de entrada y salida). Cada dimensión incluye las incertidumbres del vuelo, como posibles retrasos o adelantos en el tiempo de paso.
- La detección de conflictos se realiza teniendo en cuenta las trayectorias 4D planificadas de los vuelos con los umbrales de desviación añadidos.
- El *FAS* garantiza que la trayectoria no se conflictúa con una zona de vuelo no permitido y advierte si el vuelo entra en un área restringida.
- El *FAS* describe una trayectoria 4D típicamente en términos de altura, longitud, anchura y duración, y garantiza que la trayectoria no se conflictúa con una zona de vuelo no permitido.
- El rendimiento requerido está impulsado principalmente por la consideración de la separación y la evitación de colisiones.

3.1.2.6 AMC1 Artículo 10(2)(c) servicio de autorización de vuelo de UAS

Cuando un proveedor de servicios U-space, *USSP*, rechaza una autorización de vuelo, también llamada *U-plan*, debe indicar al operador del *UAS* el motivo del rechazo.

3.1.2.7 AMC1 Artículo 10(5) servicio de autorización de vuelo de UAS

El *USSP* debe realizar una verificación final de la autorización de vuelo de *UAS* y confirmar la activación de la autorización de vuelo de *UAS* sin demora cuando se cumplan las siguientes condiciones:

- El operador de *UAS* ha aceptado los términos y condiciones asociados a la autorización de vuelo.
- La autorización de vuelo de *UAS* se activa dentro del plazo permitido, cuando se especifica.
- El espacio aéreo U-space no está sujeto a reconfiguración dinámica y el vuelo de *UAS* sigue siendo compatible con las restricciones y las limitaciones temporales del espacio aéreo U-space.
- El vuelo de *UAS* planificado es compatible con las condiciones climáticas máximas o mínimas actuales, cuando corresponda.
- La autorización de vuelo de *UAS* no interseca con otra autorización de vuelo con mayor prioridad.
- En la proximidad del vuelo del *UAS*, no hay aeronaves tripuladas en estado de emergencia; drones cooperativos, pero no conformes, o drones no cooperativos; aeronaves tripuladas electrónicamente visibles que intersecan la trayectoria de vuelo del *UAS* planificada.

Cuando no se pueda activar la autorización de vuelo del *UAS*, el *USSP* debe indicar al operador el motivo o motivos y puede proponer opcionalmente una alternativa.

3.1.2.8 GM1-3 Artículo 10(5) servicio de autorización de vuelo de UAS

Petición de autorización de vuelo

- Se espera que el *USSP* realice una verificación final de la autorización de vuelo aceptada para asegurar la seguridad de las operaciones.
- La solicitud de activación de la autorización de vuelo debe realizarse cerca del inicio del vuelo.
- La activación de la autorización de vuelo desencadena la provisión de servicios como el *NIS*, *CMS*, *TIS*, etc. si corresponde.
- El *USSP* verifica la solicitud de autorización de vuelo al recibir la solicitud de activación y responde negativamente si hay conflictos con vuelos de mayor prioridad o situaciones de emergencia.
- Si no se recibe una solicitud de activación para un vuelo, el *USSP* retira la autorización de vuelo después del tiempo indicado en los términos y condiciones de la solicitud.

Activación de la autorización de vuelo

El Estado miembro puede establecer el intervalo de tiempo mínimo y máximo antes del despegue en el cual puede solicitarse la activación de la autorización de vuelo para un espacio aéreo U-space específico. El *USSP* puede imponer restricciones adicionales al intervalo de tiempo mínimo y máximo antes del despegue en el cual se solicita la activación. Estas restricciones deben ser proporcionadas a los operadores en los términos y condiciones de la solicitud de autorización del *U-plan*.

Retraso injustificado

Una posible interpretación de la expresión "sin demora injustificada" se proporciona en [2], que establece cual es el límite de tiempo para que se considere que la activación no ha tenido retraso, en concreto requiere que la activación de las autorizaciones de vuelo se confirme en menos de 5 segundos en el 95 % de los casos.

3.1.3 ASTM F3548-21

El estándar F3548-21 [2] define la trayectoria deseada o planeada de un UAS como representaciones basadas en volúmenes 4D que definen los límites espaciales y temporales en los que se encuentra dicho UAS. Dicha trayectoria se encuentra definida dentro de una autorización de vuelo o *U-plan*, que es el nombre que recibe el plan de vuelo para espacios aéreos U-space, por su equivalencia a los *Flight Plans* de la aviación tripulada. Estos volúmenes se construyen a partir del rendimiento de los *UAS* y generalmente se pueden clasificar como intenciones operativas basadas en área u

operaciones basadas en trayectorias, tal y como se ve en la Figura 3.2

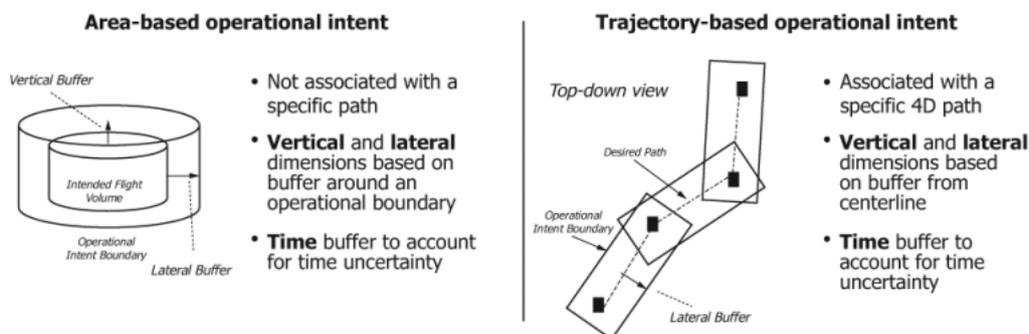


Figura 3.2 Propósitos operativos

Para las intenciones operativas basadas en trayectoria, el volumen se basa en el error total del sistema (TSE), que es la suma del error de definición de la trayectoria (PDE), el error técnico de vuelo (FTE) y el error del sistema de navegación (NSE).

Por otro lado, en las intenciones operativas basadas en área, el volumen se basa en el TSE a lo largo de la frontera del volumen que define el U-plan. Por último, para añadir la cuarta dimensión a estos volúmenes, se añade un “buffer”, que tiene en cuenta los posibles errores temporales, debidos al viento o a una inexactitud en la hora de despegue.

3.1.3.1 Estados de un U-plan

Un *U-plan* puede encontrarse en diferentes estados, y las transiciones entre ellos se detallan en la Figura 3.3:

- **Aceptado:** Un U-plan se encuentra en este estado cuando ha sido creado, desconflictuado estratégicamente, y compartido entre los USSPs.
- **Activo:** un U-plan está activado cuando el vuelo ha comenzado y el UAS se encuentra dentro del volumen operacional definido en el *U-plan* aceptado. Para entrar en este estado, el operador ha de pedir la activación del plan de vuelo al USSP.
- **No conforme** (off-nominal): Este estado corresponde a un UAS que ha salido del volumen operacional definido en el U-plan.
- **Contingente** (off-nominal): Un *U-plan* entra en el estado contingente cuando no puede seguir Activado. Puede entrar en este estado de forma manual o automática. Si un UAS se encuentra en estado no conforme durante un tiempo concreto deberá también marcarse como contingente.
- **Finalizado:** Este estado indica que el UAS ya no está usando o no usará el *U-plan*. Cuando finaliza un *U-plan*, el USSP debe eliminarlo del sistema U-space y los detalles sobre U-plan en este estado no se comunican entre los USSP.

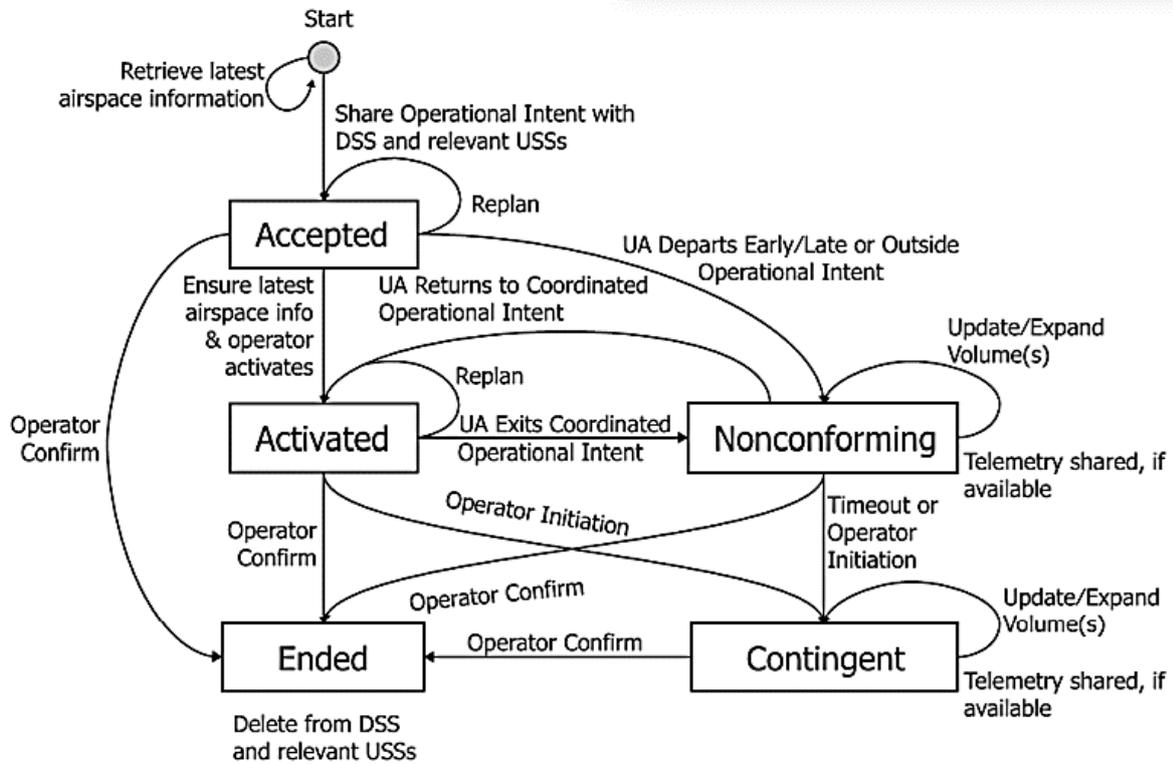


Figura 3.3. Diagrama de estados de un U-plan

3.1.3.2 Requisitos de los USSPs

Este estándar especifica algunos requisitos que deben cumplir los USSPs, la Tabla 3-1 contiene los que se deben tener en cuenta para el FAS. Para ver todos los requisitos del estándar, consultar. [2]

Tabla 3-1 Requisitos de los USSPs

Ref. F3548-21	
SCD0005	Si un U-plan se encuentra aceptado y el USSP responsable del mismo no responde o ha fallado, al U-plan se le aplicará la mínima prioridad.
SCD0010	Si un U-plan se encuentra Activado/No conforme/Contingente y el USSP responsable del mismo no responde o ha fallado, al U-plan se le aplicará la máxima prioridad.
SCD0015	Antes de aceptar un U-plan, el USSP deberá comprobar que no conflictúa con otros U-plans de mayor prioridad.
SCD0020	Antes de modificar un U-plan ya aceptado, el USSP deberá comprobar que no conflictúa con otros U-plans de mayor prioridad.

Ref. F3548-21	
SCD0025	Antes de activar <i>U-plan</i> , el <i>USSP</i> deberá comprobar que no conflictúa con otros <i>U-plans</i> de mayor prioridad.
SCD0030⁴	Antes de modificar un <i>U-plan</i> ya activo, el <i>USSP</i> deberá comprobar que no conflictúa con otros <i>U-plans</i> de mayor prioridad, excepto el caso en el que el conflicto ya existiese antes de ser modificado.
SCD0035	Antes de aceptar un <i>U-plan</i> , el <i>USSP</i> deberá comprobar que no conflictúa con otros <i>U-plan</i> de igual prioridad cuando la legislación no lo permita.
SCD0040	Antes de modificar un <i>U-plan</i> ya aceptado, el <i>USSP</i> deberá comprobar que no conflictúa con otros <i>U-plans</i> de igual prioridad cuando la legislación no lo permita.
SCD0045	Antes de activar un <i>U-plan</i> , el <i>USSP</i> deberá comprobar que no conflictúa con otros <i>U-plans</i> de igual prioridad cuando la legislación no lo permita.
SCD0050⁴	Antes de modificar un <i>U-plan</i> ya activo, el <i>USSP</i> deberá comprobar que no conflictúa con otros <i>U-plans</i> de igual prioridad cuando la legislación no lo permita, excepto el caso en el que el conflicto ya existiese antes de ser modificado.
SCD0055	Cuando un <i>U-plan</i> que conflictúa con otro <i>U-plan</i> ya existente de la misma prioridad es aceptado, porque la legislación lo permita, el <i>USSP</i> a cargo del nuevo <i>U-plan</i> debe enviar una notificación al <i>USSP</i> a cargo del plan ya existente con un tiempo límite de 1 segundo en el 95 % de las veces.
SCD0060	Cuando un <i>U-plan</i> que se encuentra en estado aceptado es modificado y conflictúa con un <i>U-plan</i> ya existente de la misma prioridad, siempre que la legislación lo permita, el <i>USSP</i> a cargo del nuevo <i>U-plan</i> debe enviar una notificación al <i>USSP</i> a cargo del plan ya existente con un tiempo límite de 1 segundo en el 95 % de las veces.

⁴ Esta excepción se incluye porque no todos los conflictos pueden resolverse mediante la replanificación debido a la falta de tiempo u otros factores, como otros objetivos operativos o limitaciones en la zona. Por lo tanto, esta especificación no obliga a la replanificación como único medio para resolver el conflicto. El conflicto podría, por ejemplo, abordarse a través de medios tácticos como el sistema DAA a bordo de la aeronave.

Ref. F3548-21	
SCD0065	Cuando un <i>U-plan</i> que se encuentra en estado aceptado es activado y conflictúa con un <i>U-plan</i> ya existente de la misma prioridad, el <i>USSP</i> a cargo del nuevo <i>U-plan</i> debe enviar una notificación al <i>USSP</i> a cargo del plan ya existente con un tiempo límite de 1 segundo en el 95 % de las veces.
SCD0070	Cuando un <i>U-plan</i> que se encuentra en estado Activo es modificado y conflictúa con un <i>U-plan</i> ya existente de la misma prioridad, el <i>USSP</i> a cargo del nuevo <i>U-plan</i> debe enviar una notificación al <i>USSP</i> a cargo del plan ya existente con un tiempo límite de 1 segundo en el 95 % de las veces.
SCD0075	Si un <i>USSP</i> pide datos de interés de un determinado <i>U-plan</i> , el <i>USSP</i> a cargo de dicho plan debe responder con un tiempo límite de 1 segundo el 95 % de las veces.
SCD0080	Durante el tiempo en el que un <i>U-plan</i> se encuentre en estado activo, no conforme o contingente; el <i>USSP</i> a cargo debe mantenerse alerta de <i>U-plans</i> nuevos o modificados relevantes para el <i>U-plan</i> en cuestión.
SCD0085	Ante la recepción de una notificación de que un <i>U-plan</i> puede ser relevante para una suscripción, el <i>USSP</i> a cargo debe enviar la información del <i>U-plan</i> con un tiempo límite de 5 segundos el 95 % de las veces.
SCD0090	Cuando un <i>USSP</i> crea o modifica un <i>U-plan</i> que entra en conflicto con otro <i>U-plan</i> , el <i>USSP</i> debe enviar una notificación que informe del conflicto al operador del <i>UAS</i> o al sistema de automatización del <i>UAS</i> con un tiempo límite de 5 segundos el 95 % de las veces.
SCD0095	Cuando un <i>USSP</i> detecta un conflicto entre un <i>U-plan</i> cualquiera y un <i>U-plan</i> del que está a cargo, el <i>USSP</i> debe enviar una notificación que informe del conflicto al operador del <i>UAS</i> o al sistema de automatización del <i>UAS</i> con un tiempo límite de 5 segundos el 95 % de las veces.

3.1.3.3 Requisitos de interoperabilidad y Discovery Synchronization Service (DSS)

Esta especificación asume que uno o más *USSPs* proporcionan servicios U-space en una región determinada. Los *USSPs* participantes deben ser interoperables, compartiendo datos según sea necesario para respaldar varios servicios. Por ejemplo, se deben intercambiar intenciones operativas para respaldar la detección de conflictos

estratégicos.

El paradigma de interoperabilidad consta de dos partes:

1. Un mecanismo estandarizado de descubrimiento, denominado *DSS* (Servicio de Descubrimiento y Sincronización, *DSS* por sus siglas en inglés), cuyas funciones principales son identificar los *USSPs* con los que se requiere el intercambio de datos y verificar que un *USSP* tenga en cuenta entidades relevantes gestionadas por otros *USSPs* cuando sea necesario (por ejemplo, al planificar una nueva intención operativa);
2. Protocolos de intercambio de datos específicos del servicio (API) utilizados para intercambiar datos de manera estandarizada una vez que se ha determinado a través del *DSS* con qué *USSP* se requiere el intercambio de datos.

La Figura 3.4 ilustra los intercambios de datos de este paradigma de manera independiente del servicio. El término "entidades" (Entity) refiere a los tipos de datos que deben compartirse entre los *USSPs*. Se utilizan diferentes tipos de entidades para diferentes servicios U-space.

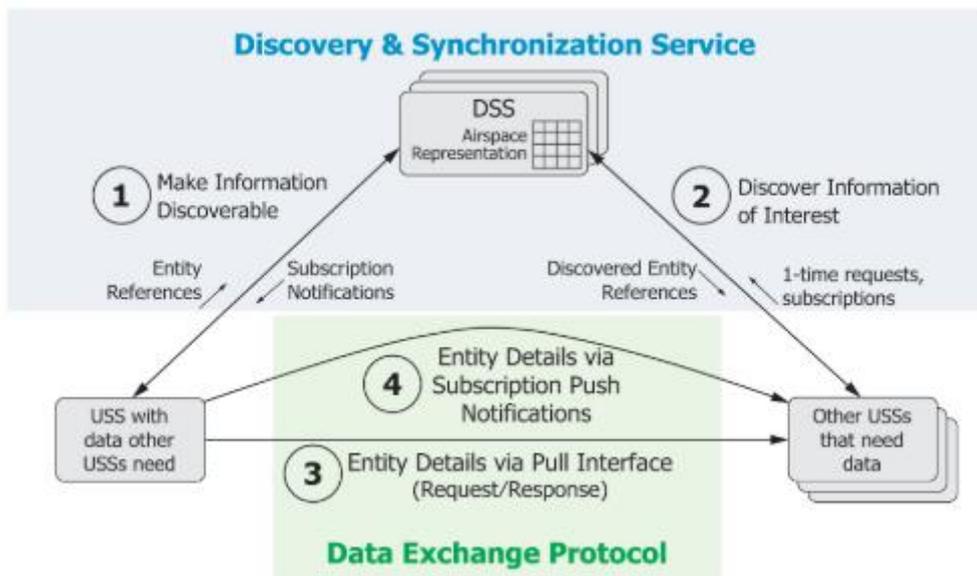


Figura 3.4 Paradigma de interoperabilidad

Una "Referencia de Entidad" se almacena en el *DSS* y permite el descubrimiento. Las referencias de entidad incluyen solo información limitada sobre la entidad correspondiente, como su tipo, identificador único, ubicación aproximada y rango de tiempo, número de versión opaco actual (*OVN*) (descrito a continuación) y cómo contactar al *USSP* gestor para solicitar detalles adicionales. Los detalles completos de las entidades no se almacenan en el *DSS*, sino que son mantenidos y deben obtenerse del *USSP* gestor.

El *DSS* encapsula una representación del espacio aéreo en la que se asignan las referencias de entidad. Los detalles de la implementación de esta representación del espacio aéreo son ocultos. Pero se puede conceptualizar de la siguiente manera: como

una cuadrícula, en la que se le asigna una referencia de entidad a un vuelo que determina que celdas interseca.

Dado este contexto, las interacciones principales (numeradas en la Figura 3.4) son las siguientes:

1. **Hacer Descubrible:** Un *USSP* con una entidad sobre la cual otros *USSP* pueden necesitar información la hace descubrible escribiendo una referencia de entidad en el DSS. El DSS crea el OVN para la entidad y lo proporciona al *USSP* gestor.
2. **Descubrir:** Otros *USSPs* interesados en entidades de algún tipo consultan el *DSS* utilizando un volumen 4D para caracterizar el área y el tiempo de interés. Los *USSPs* también pueden suscribirse para recibir notificaciones futuras de nuevas referencias de entidad, modificadas o eliminadas en el área de interés. El *DSS* asigna la consulta a la representación del espacio aéreo y encuentra las celdas de la cuadrícula que intersecan con referencias de entidad del tipo deseado (si las hay). Luego, el *DSS* devuelve al *USSP* solicitante una lista de las referencias de entidad descubiertas.
3. **Solicitudes y Respuestas de Datos:** El *USSP* solicitante utiliza el protocolo de intercambio de datos aplicable para contactar al *USSP* gestor correspondiente y obtener los detalles de cada referencia de entidad descubierta.
4. **Notificaciones de Suscripción:** Si un *USSP* establece una suscripción en el *DSS* (para un área 4D de interés), cuando otro *USSP* crea, modifica o elimina una referencia de entidad que interseca con la suscripción, el *DSS* informa al *USSP* escritor de la suscripción. Al recibir esta información, el *USSP* escritor debe proporcionar al *USSP* suscriptor los detalles de la entidad nueva, modificada o eliminada.

El estándar F3548-21 ha desarrollado una ingeniosa estrategia para asegurarse de que los *USSPs* no introducen una entidad sin tener en cuenta otras entidades relevantes, veamos la Figura 3.5 para entender como se ha conseguido.

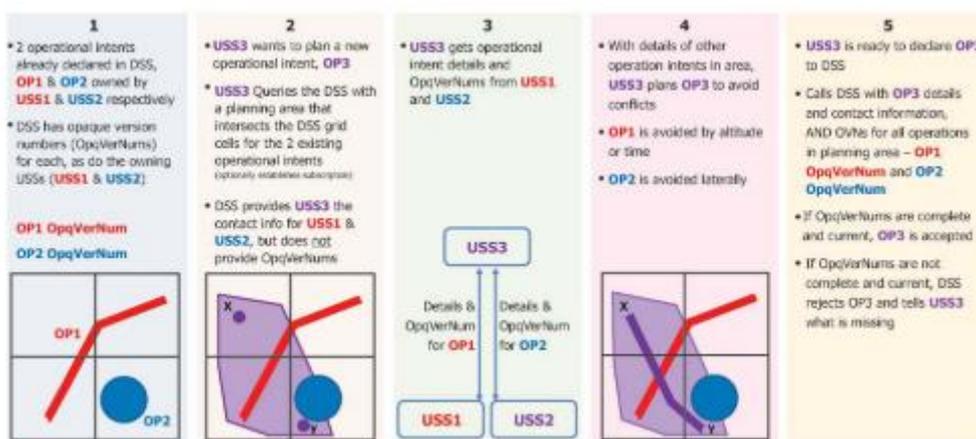


Figura 3.5 Uso de OVN

- El panel 1 muestra que se han escrito previamente dos referencias de entidad de

intenciones operativas en el DSS. OP1 es gestionada por USS1⁵ y OP2 por USS2⁵. El DSS tiene una copia del OVN para cada una.

- En el panel 2, USS3⁵ consulta el DSS para obtener información sobre un área (representada por el polígono morado) en la que tiene la intención de planificar una nueva operación. El DSS asigna el área de planificación a la representación del espacio aéreo y determina que interseca con las cuatro celdas en las que se asignaron OP1 y OP2. El DSS proporciona a USS3 las identificaciones de OP1 y OP2 y los detalles de contacto de USS1 y USS2, pero **NO** proporciona los OVN de las dos operaciones.
- En el panel 3, USS3 se pone en contacto con USS1 y USS2 para obtener los detalles de OP1 y OP2, y para obtener los OVN.
- En el panel 4, USS3 utiliza los detalles de OP1 y OP2 para planificar OP3 sin conflictos.
- En el panel 5, USS3 escribe la nueva referencia de entidad de la intención operativa en el DSS. Incluye los números de versión opacos para todas las operaciones relevantes. El DSS solo permite agregar la referencia de entidad de la nueva intención operativa si el conjunto de números de versión opacos está completo y actualizado. Esto último es lo que asegura de que se hayan tenido en cuenta todas las entidades necesarias.

3.1.4 CIR 2019/947

El Real Decreto 2019/947 de la Comisión Europea del 24 de mayo de 2019 establece las normas y procedimientos para la operación de sistemas de aeronaves no tripuladas (UAS). Esta regulación abarca disposiciones detalladas tanto para los sistemas de aeronaves no tripuladas como para el personal y las organizaciones involucradas en estas operaciones. [3]

Las operaciones de UAS se clasifican en tres categorías:

3.1.4.1 Categoría abierta

Las operaciones en categoría abierta se consideran de menor riesgo. Las operaciones en esta categoría no requieren una autorización específica por parte de la autoridad competente, sin embargo, deben cumplir con reglas y limitaciones específicas. Para pertenecer a la categoría abierta, una operación debe cumplir los siguientes requisitos:

- El UAS pertenece a una de las clases especificadas en la regulación delegada (EU) 2019/945 o si es de construcción privada, cumple las condiciones del artículo 20 de este mismo real decreto.
- El peso máximo en despegue es menor de 25 kg.
- El piloto debe asegurarse de mantener una distancia adecuada con la gente y no sobrevolar zonas con alta densidad de personas.
- El piloto debe mantener la aeronave en “Visual Line Of Sight” (VLOS), excepto cuando vuele en modo “Follow me” o con un observador (Extended Visual Line Of

⁵ En el estándar USS es equivalente a USSP

Sight -EVLOS).

- Durante el vuelo debe mantenerse el UAS a un máximo de 120 metros de altura, desde el punto más cercano de la superficie.
- Durante el vuelo, la aeronave no puede transportar material peligroso ni dejar caer material/objetos.

Además, este real decreto subdivide esta categoría en tres subcategorías: A1, A2 y A3. La descripción de estas subcategorías puede encontrarse en [5], pero queda fuera del alcance de este trabajo puesto que no tiene relevancia en el desarrollo de la herramienta.

3.1.4.2 Categoría específica

La categoría específica abarca las operaciones que presentan un nivel medio de riesgo o en las que no se pueden cumplir los requisitos de la categoría Abierta. Estas operaciones requieren una autorización operativa por parte de la autoridad competente, la cual se otorga en base a una evaluación de riesgos realizada por el operador del dron. La evaluación de riesgos debe identificar posibles peligros y proponer medidas para mitigar los riesgos. Dicha evaluación se realiza siguiendo la metodología SORA. [6]

Esta categoría cubre las operaciones de sistemas de aeronaves pilotadas por control remoto que no se encuentran ni en categoría abierta ni en certificada. Estas aeronaves, cuando deseen operar en la categoría específica, deben solicitar una Autorización Operativa emitida por la autoridad competente, a menos que la operación se realice dentro del alcance de un Escenario Estándar. Para otorgar la autorización, el operador debe realizar una evaluación de riesgos siguiendo un procedimiento estándar y proponer medidas de mitigación para reducir el riesgo a valores aceptables.

3.1.4.3 Categoría certificada

La categoría certificada está destinada a operaciones con un nivel de riesgo mayor, que requiere la certificación de la aeronave de acuerdo con los estándares de aviación establecidos. Estas operaciones requieren que el UAS, el operador y el piloto remoto estén certificados, y las operaciones deben ser aprobadas por la autoridad competente. Sin embargo, en cuanto a lo que a la gestión del tráfico se refiere, las operaciones en categoría certificada y específica son indistinguibles.

Se consideran operaciones en categoría certificada aquellas que conllevan el transporte de personas, lo que se conoce como aerotaxis, de mercancía peligrosa o el sobrevuelo de aglomeraciones de personas.

4 PROYECTOS SESAR

SESAR es el pilar tecnológico de la política del Cielo Único Europeo de la UE y un facilitador clave de la Estrategia de Movilidad Sostenible e Inteligente de la Comisión Europea. SESAR define, desarrolla e implementa tecnologías para transformar la gestión del tráfico aéreo en Europa.[7]

La Asociación Conjunta SESAR 3 (SESAR 3 JU) es una asociación público-privada institucional europea creada para acelerar, a través de la investigación y la innovación, la implementación de esta transformación digital, es decir, el Cielo Único Europeo Digital. Para lograrlo, aprovecha, desarrolla y acelera la adopción de las soluciones tecnológicas más avanzadas para gestionar aeronaves convencionales, drones, taxis aéreos y vehículos que vuelan a altitudes más altas. Con el Cielo Único Europeo Digital se pretende hacer el espacio aéreo europeo más eficiente y respetuoso con el medio ambiente.

En este apartado hablaremos de algunos de los proyectos financiados por SESAR que son relevantes y servirán de base para el desarrollo de la herramienta que se presenta en este trabajo.

4.1.1 CORUS

CORUS (Concept of Operations for European UTM Systems) es el principal proyecto en el marco del U-space. Este proyecto definió el Concepto de Operaciones (ConOps) para el U-space y la arquitectura de los servicios, entre otras muchas cosas, en 2019. Este proyecto perseguía facilitar a integración de drones en el espacio aéreo VLL (Very Low Level). Este ConOps fue extendido por un proyecto posterior llamado CORUS-XUAM [8].

CORUS-XUAM es un proyecto de validación a gran escala de 24 meses que pretende demostrar cómo los servicios y soluciones U-space podrían apoyar las operaciones de vuelo de movilidad aérea urbana, permitiendo que los UAS y otros usuarios del espacio aéreo (aviación tripulada y no tripulada) operen de manera segura, segura, sostenible y eficiente en un espacio aéreo controlado y totalmente integrado, sin un impacto indebido en las operaciones actualmente gestionadas por ATM [9].

Las actividades de CORUS-XUAM comenzaron con la actualización del ConOps de U-space, abordando la integración de las operaciones UAM/UAS en el espacio aéreo, e identificando también nuevos servicios y volúmenes U-space.

4.1.1.1 Servicios U-space

CORUS y CORUS-XUAM sugieren la implementación del U-space en 4 fases, (U1, U2, U3 Y U4) y describen los siguientes servicios y en qué fase deberían implementarse:

Tabla 4-1. Servicios propuestos por CORUS

SERVICIO	SUB SERVICIOS	FASE
Registration	e-registration	U1
	Registration Assistance	
Network identification	e-identification	U1
	Position report submission subservice	U2
Tracking		U2
Surveillance Data Exchange		U2
Drone Aeronautical Information Management		U2
Geo-awareness	Geo-awareness	U1
	Geo-fence provision	U2
Operation Plan Preparation/Optimisation service		U2
Risk Analysis Assistance		U2
Flight Authorization Service		U2
Strategic Conflict Prediction		U2
Strategic Conflict Resolution		U2
Dynamic Capacity Management		U3
Tactical Conflict Prediction		U3
Tactical Conflict Resolution		U3
Legal Recording		U2
Emergency Management		U2
Incident/ Accident Reporting		U2
Traffic Information		U2
Navigation Infrastructure Monitoring		U2
Communication Infrastructure Monitoring		U2

SERVICIO	SUB SERVICIOS	FASE
Digital Logbook		U2
Weather Information		U2
Geographical information Service		U2
Population density map		U2
Electromagnetic Interference Information		U2
Navigation Coverage Information		U2
Communication Coverage Information		U2
Procedural interface with ATC		U2
Collaborative Interface with ATC		U3
Vertical Conversion Service		U3
Vertical Alert and Information Service		U3

Aunque CORUS [10] define un total de 31 servicios U-space, tal y como se muestra en la Tabla 4-1, en esta sección nos centraremos en describir solo aquellos servicios relacionados con el objetivo de este proyecto, la resolución de conflictos en fase estratégica, siendo éstos el *FAS*, el *SCPS* (Strategic Conflict Prediction Service) y el *SCRS* (Strategic Conflict Resolution Service). El resto de servicios U-space quedan fuera del alcance del presente proyecto.

4.1.1.1 Flight Authorization Service

Este servicio, que ya se ha explicado en el apartado **3.1.1 CIR 2021/664**, es ampliado en [10] y se espera que se despliegue en la fase U2, su función consiste en recibir los *U-plans* y los utiliza en una diversa cantidad de actividades basadas en la seguridad. Desde un punto de vista operacional, esto es un único servicio, sin embargo, aún no se ha especificado como será su implementación. Aunque aún no está confirmado, *FAS* debería funcionar de la siguiente manera con la recepción de un U-plan:

1. Comprobación sintáctica.
2. Comprobación semántica.
3. Generar identificador.
4. Comprobar que la trayectoria sea correcta (es decir, que no haya huecos o incongruencias entre puntos de paso).
5. Comprobación de avisos climáticos.
6. Comprobación de fronteras o zonas de vuelo no permitido o restringido (geo-

fencing y geo-awareness).

7. Si interfiere con alguna zona controlada por el ATC, se activa el procedimiento adecuado.
8. Se invoca el SCPS y, si procede, al SCRS.

4.1.1.2 Strategic Conflict Prediction Service

Anteriormente, en el apartado 3.1.2 se ha definido como serán los volúmenes de protección alrededor de los UAS, el SCPS se define como el servicio encargado de detectar intersecciones entre los volúmenes 4D, que implican una probable LoS (Loss of Separation).

Actualmente los detalles de las operaciones se consideran como comercialmente sensibles, es por eso por lo que la interoperabilidad entre USSP ha sido estandarizada [7] y describe como detectar conflictos revelando la mínima información posible mediante el uso del “Discovery and Synchronization Service” (DSS). Este estándar se describe más ampliamente en el apartado 3.1.3.3.

4.1.1.3 Strategic Conflict Resolution Service

El SCRS debe resolver los conflictos detectados por el SCPS. Esto se lleva a cabo modificando uno o varios de los planes en conflicto para que no exista intersección en sus trayectorias tanto en tiempo como en espacio. Los nuevos U-plans serán propuestos a los operadores para que estos los modifiquen y reenvíen, o los cancelen definitivamente. La forma en la que realizar estos cambios aún está en desarrollo por parte de las autoridades competentes, y definir esto no se encuentra entre los objetivos del presente trabajo.

4.1.1.2 Volúmenes U-space

VLL (Very Low Level airspace) es el espacio aéreo por debajo del usado por el tráfico VFR (Visual flight rules). En el anexo 2 de la OACI [[11] (4.6)] se especifica que, excepto cuando sea necesario para despegue y aterrizaje o cuando haya permiso expreso de la autoridad competente un VFR **no** debe volar:

- Por debajo de **300 metros** sobre el obstáculo más alto en un radio de 600 m de la aeronave. sobre ciudades, pueblos o asentamientos congestionados o sobre grupos de gente en campo abierto.
- En cualquier otro lugar a una altura menor a **150 metros** sobre el terreno o el agua.

CORUS sugiere dividir el VLL en tres tipos de volúmenes diferentes según [10]:

- **Cantidad** de vuelos esperados.
- **Riesgo** en **tierra**.
- **Riesgo** en **aire**.
- Factor de **aceptación** público.
- **Servicios U-space** necesarios para una operación segura.

Estos volúmenes difieren entre ellos esencialmente en dos aspectos: los servicios ofrecidos, y por tanto las operaciones posibles, y los requisitos de entrada. Los tres tipos

mencionados se identifican como “X”, “Y” y “Z”, y todo espacio VLL se encasilla en uno de estos tres tipos. La diferencia más importante entre estos tipos es:

- **X**: No se ofrece ningún servicio de resolución de conflictos.
- **Y**: Se ofrece solamente resolución de conflicto estratégico.
- **Z**: Se ofrece resolución estratégica y táctica de conflictos.

4.1.1.2.1 Volúmenes X

En los volúmenes X solo hay unos pocos requisitos básicos para el operador, el piloto y el dron, como también, pocos son los servicios ofrecidos en estos espacios. En el volumen X el piloto es el responsable de mantener la separación.

4.1.1.2.2 Volúmenes Y

El acceso a Y requiere tener un plan operacional aprobado. También puede requerir requisitos técnicos específicos que tendrán que ser demostrados para la aprobación del vuelo.

En estos volúmenes los conflictos entre vuelos son resueltos previos al inicio de la operación. Los volúmenes Y se esperan en áreas donde el riesgo aéreo o terrestre es demasiado grande para volúmenes X (por encima de población, densidad de drones alta, etc.). Estos volúmenes también se espera que se creen para delimitar el acceso a ciertas zonas, como parques nacionales.

4.1.1.2.3 Volúmenes Z

Estos volúmenes admiten mayor densidad que los Y, es por eso por lo que se esperan en zonas con alta demanda, o con riesgo particular. En estos espacios, se requiere un plan operacional aprobado, pero además también una conexión continua del piloto al U-space y un reporte de posición de la aeronave con suficiente calidad para permitir el seguimiento. Según quien provea el servicio de resolución de conflicto táctico se puede distinguir entre volúmenes Za, Zu y Zz.

4.2.1.2.3.1 Za

El servicio de resolución de conflictos tácticos es provisto por el ATS. Cualquier volumen designado como Za se considera espacio controlado, ATS está en control y provee los servicios necesarios para este volumen, haciendo uso, si es necesario, de los servicios de U-space.

4.2.1.2.3.2 Zu

El servicio de resolución de conflictos tácticos es provisto por al menos un *USSP*. Este volumen se considera espacio aéreo controlado, pero sin existir un ATS, por lo que el operador debe seguir las instrucciones ofrecidas por el *USSP* de manera obligatoria.

4.2.1.2.3.3 Zz

El servicio de resolución de conflicto táctico es provisto por al menos un *USSP* Este volumen no se considera espacio aéreo controlado, por lo que las instrucciones ofrecidas

por el USSP se consideran recomendaciones.

4.1.2 BUBBLES

BUBBLES es un proyecto de investigación exploratoria que tuvo como objetivo formular y validar un ConOps sobre un nuevo servicio U-space para la gestión de la separación entre UAS con el fin mejorar la seguridad y eficiencia del espacio aéreo VLL. Para ello, BUBBLES desarrolló algoritmos basados en Inteligencia Artificial para calcular los mínimos de separación entre cada par de UAS, según la clase de tráfico UAS, y actualizar dichas separaciones en base al rendimiento de los servicios de comunicación, navegación y vigilancia (CNS) para mantener un nivel de seguridad aceptable (*TLS*). Además, desarrolló una herramienta para calcular el riesgo de colisión de los *UAS* en un escenario dado y la capacidad de dicho escenario. El proyecto también se centró en formular requisitos para los sistemas de CNS, y en promover la aceptación pública del uso de *UAS* mediante la concienciación sobre los niveles de seguridad alcanzados.[12]

La gestión de conflictos se divide en tres capas: gestión estratégica, provisión de separación y evitación de colisiones. El servicio de gestión de separación se centra en la fase de provisión de separación, que se da en la fase táctica del vuelo y que consta de cuatro pasos: (1) detección de conflictos, (2) formulación de soluciones, (3) implementación de soluciones y (4) monitoreo de las soluciones. El objetivo del servicio es proporcionar la información necesaria a todos los USSP y operadores *UAS* para resolver conflictos tácticos de acuerdo con reglas establecidas y lograr un *TLS* aceptable. BUBBLES propone que el servicio de gestión de separación sea proporcionado por el proveedor de información común (*CIS*), formando parte de él.[13]

BUBBLES sugiere que el servicio de gestión de separación sea:

- **Opcional** en las áreas de U-space sin servicios de resolución táctica de conflictos (volúmenes Y). Aquí se puede implementar una red de seguridad basada en tierra para apoyar a los RPICs (Remote Pilot in Command) en la realización de procesos de deconflicción táctica de manera coordinada.
- **Obligatorio** en las áreas de U-space donde se proporciona resolución táctica de conflictos a través de los servicios de U-space (volúmenes Z). Aquí el servicio de gestión de separación actuará como un único punto de referencia para los diferentes USSPs que operan en el espacio aéreo. Su función será garantizar que todos los procesos de deconflicción táctica se ejecuten de manera armonizada, justa y eficiente.

4.1.2.1 Definición de conflicto según BUBBLES

BUBBLES utiliza un enfoque centrado en la operación y basado en el riesgo para la gestión de separación en U-space. En lugar de tener una separación general para un espacio aéreo determinado, se establecen mínimas de separación específicas para cada par de aeronaves en función de sus características y rendimiento (véase Figura 4.1). Para simplificar el proceso, se utilizan "volúmenes de protección" de forma cilíndrica alrededor de cada aeronave, donde la superposición de estos volúmenes indica un *LoS*. Además, se propone un esquema de clasificación de tráfico basado en

el rendimiento de las aeronaves para asignar las separaciones mínimas de manera eficiente. Este enfoque representa un cambio respecto a los métodos tradicionales, donde se establecía una distancia fija como umbral para la separación. En el desarrollo de las herramientas desarrolladas en este TFG, se parte de esta definición de separación y de esta clasificación del tráfico.

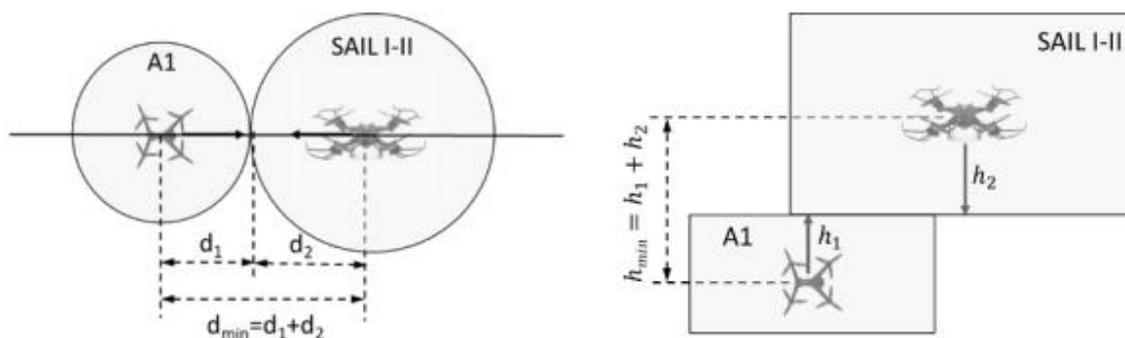


Figura 4.1 Conflicto según la definición de BUBBLES (Izquierda: vista en planta, derecha: vista alzada)

4.1.2.2 Definición de escenarios

La definición del TLS para el U-space en términos de fatalidades por hora de vuelo puede ser malinterpretada si no se gestiona correctamente. En la aviación tripulada, una colisión en el aire suele ser desastrosa, mientras que, en la no tripulada, los efectos son más variables y dependen del tipo de aeronave involucrada. BUBBLES utilizó datos históricos de accidentes aéreos tripulados para definir un TLS para colisiones en el espacio U-space, proponiendo un valor de $2.5e-7$ fatalidades por hora de vuelo. Sin embargo, debido al alto número de horas de vuelo de los UAS, este TLS podría dar lugar a un número inaceptable de fatalidades.

El proceso desarrollado por BUBBLES considera las particularidades del escenario, incluyendo el tipo de tráfico y el efecto de posibles colisiones. Si el TLS resulta en demasiadas fatalidades, se deben revisar las medidas de mitigación o modificar el TLS.

BUBBLES ha definido diez tipos de tráfico según su categoría operacional y su nivel de riesgo dado por el tamaño, tipo de operación típica, rendimiento y maniobrabilidad.

Traffic characteristics		Traffic class		
Unmanned Open	Non-carrying people	A1	I	
		A2	II	
		A3	III	
Unmanned Specific		SAIL I-II	IV	
		SAIL III-IV	V	
		SAIL V-VI	VI	
Unmanned Certified		No passenger	VII	
		Passenger	VIII	
Manned Certified		Carrying people	VFR	IX
			IFR	X

Figura 4.2 Clasificación de tráfico según BUBBLES

4.1.2.3 Proceso de gestión de la separación

El servicio de gestión de la separación se centra en la definición de cómo la provisión de separación a nivel táctico debe ser implementada en un U-space dado, de forma que todos los conflictos que surjan en el mismo se resuelvan aplicando las mismas reglas y se pueda alcanzar un *TLS* determinado. A continuación, describiremos cómo los conflictos tácticos se resuelven aplicando las mismas reglas y se consigue un determinado *TLS*, y cómo se implementa el servicio de gestión de separación tanto a nivel táctico como a nivel estratégico.

4.1.2.3.1 Proceso de provisión de separación.

El proceso de provisión de separación se compone de cuatro pasos, como se ha dicho anteriormente y queda resumido en la siguiente figura:



Figura 4.3 Proceso de provisión de separación por BUBBLES

- La **detección de un conflicto** se basa en la última posición conocida de las aeronaves implicadas y su trayectoria prevista. Se define un conflicto como una convergencia en espacio y tiempo de la trayectoria predicha de las aeronaves que viola los mínimos de separación.
- La **formulación de la solución** consiste en definir de manera unívoca un agente separador y un modo separador para resolver el conflicto. La designación de un separador y la selección de un modo de separación dependen en gran medida de la capacidad de intervención. El modo de separación consiste en un conjunto aprobado de reglas, procedimientos y condiciones de aplicación asociadas con los mínimos y métodos de separación.
- La **implementación de la solución** es un proceso en tres pasos:
 1. Se notifica la solución definida en el paso anterior al agente que controla la aeronave,

2. El agente ejecuta las maniobras necesarias para evitar el peligro, y
 3. La aeronave realiza las maniobras ejecutadas por el agente de control.
- La **monitorización de la ejecución de la solución**. La gestión de la separación es un proceso iterativo. El servicio de gestión de la separación de BUBBLES considera diferentes agentes encargados de supervisar la implementación de la solución y alertar al agente separador en caso de que sea ineficaz.

4.1.2.3.2 Obtención de los mínimos de separación

Puesto que las separaciones mínimas entre UAS dependen en gran parte del escenario, en la fase estratégica o pre-operacional, mientras se designe un espacio aéreo U-space, se calcularán dichos valores para un escenario o volumen U-space dado, siendo las separaciones mínimas de referencia. Dichas separaciones, podrán ser actualizadas en la fase táctica, durante la provisión de la separación, en caso de ser necesario según el rendimiento de los servicios CNS. También en la fase pre-operacional, el servicio de gestión de la separación se encarga de definir qué información se proporcionará a los usuarios del U-space durante la fase táctica. Para ello, el servicio de gestión de la separación ejecuta el proceso esbozado gráficamente en la Figura 4.4.



Figura 4.4 Servicio de gestión de separación a nivel pre-operacional

1. **Caracterizar el escenario** que define el U-space en la forma más representativa posible. Esto incluye la definición de la mezcla de tráfico y las principales características de las clases de tráfico que componen la mezcla, así como la identificación del tamaño del volumen U-space y su estructura.
2. **Establecer los modos de separación** aplicables definiendo los modos de automatización apropiados, los métodos de separación y el horizonte de conflicto.
3. **Proponer objetivos iniciales para SPR/INTEROP** (Requisitos de seguridad, rendimiento e interoperabilidad) que permitan una primera aproximación de los

rendimientos que se requerirán para las aeronaves y los sistemas CNS que operan dentro del U-space.

4. **Calcular los mínimos de separación**, de acuerdo con los niveles de confianza objetivo para las relaciones de riesgo y el modo de separación establecido.
5. **Calcular las F(MAC)** (frecuencia de colisión aérea) mitigadas y LoS, así como las relaciones de riesgo de las barreras de mitigación, a través de simulaciones.
6. **Verificar si se logra el TLS**, considerando la demanda esperada y la capacidad de referencia derivada de la frecuencia mitigada de colisiones en el aire.
7. **Actualizar los objetivos para SPR/INTEROP y/o los mínimos de separación** y repetir el proceso iterativamente hasta que se logre un resultado razonable en términos de capacidad, seguridad y SPR/INTEROP.

Estos mínimos de separación son los que utilizará la herramienta que estamos diseñando para evaluar si existe conflicto entre diferentes UAS.

4.1.3 USEPE

USEPE es un proyecto que tiene como objetivo investigar en los métodos de separación de UAS en situaciones de alta densidad. Para ello en [8] analizan diferentes métodos propuestos por otros proyectos o empresas y proponen uno nuevo. A continuación, se detallarán los procedimientos más importantes que analiza USEPE.

4.1.3.1 Pasillos de drones

La idea principal de este método es generar diferentes pasillos, similares a aerovías, para diferentes capacidades de drones, tal y como se puede ver en la Figura 4.5. Los UAS pueden salir de estas aerovías, pero a una velocidad más limitada. La ubicación de las aerovías la pueden determinar las autoridades en función de algunos requerimientos, como ruido, seguridad, etc. El ancho y alto de los pasillos se debe decidir en función de la precisión de los UAS que los vuelen.

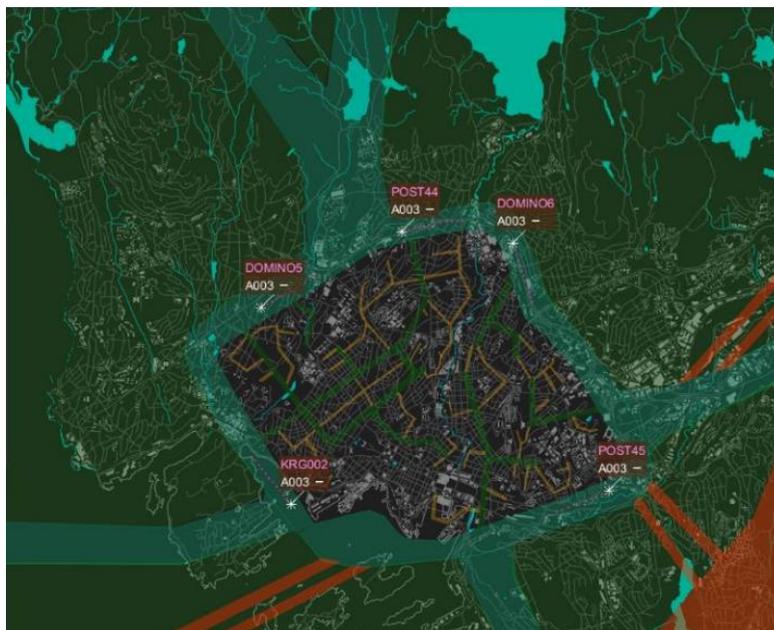


Figura 4.5 Ejemplo de distribución de “Pasillo de drones”

Cada pasillo está definido como un volumen en tres dimensiones y unas reglas de Geo-Vectores (véase 4.1.3.3) de forma que cada pasillo sea como una calle de sentido único en la que todos los drones llevan direcciones y velocidades similares. En este método la entrada a estos pasillos deberá ser supervisada, ya que, aunque se haya realizado un deconflicción estratégica, cambios en velocidades podrían conllevar un evento de pérdida de separación (LoS).

4.1.3.1 Rendimiento esperado del método

El rendimiento del método estaría indicado por la capacidad total del espacio aéreo disponible en el U-space, su capacidad para permitir muchos vuelos simultáneos sin colisiones y un mínimo de incidentes de separación. En [14] se indica que un espacio aéreo ordenado funcionará mejor que un sistema total de vuelo libre cuando el número de vuelos diarios para un área metropolitana supera los 10.000 vuelos por día.

4.1.3.2 Gestión del espacio aéreo basado en la densidad

Este concepto se basa en modelizar el espacio aéreo y sus usuarios, para después determinar la densidad en la que el uso del espacio aéreo es seguro.

- El espacio aéreo es dividido en **celdas** de requisitos similares.
- Cada usuario se modela como un **volumen de seguridad** definido por los parámetros relevantes de rendimiento del UAS (navegación comunicación habilidad de detectar a otros UAS o ser detectado, tanto de forma activa como pasiva).

Teniendo esto en cuenta, cuando se añaden nuevos usuarios, estos llenan las celdas de forma que los usuarios con rendimiento más pobre llenan más rápidamente el espacio que los de más alto rendimiento. Esto provoca que en zonas de baja densidad haya un

alto nivel de libertad, mientras que las de más alta densidad sean más restrictivas.

Las celdas se pueden adaptar dinámicamente dependiendo de condiciones externas (clima, disponibilidad de servicios U-space, etc.) y se puede describir mediante coordenadas, tiempo en el que es válida, características y requisitos específicos de la celda. La distancia de separación para una determinada aeronave debe ser determinada paramétricamente con el azimut y la elevación. En este método la distancia puede ser más grande en la dirección en la que se desplaza el UAS (Figura 4.6) permitiendo límites que no sean elipsoidales centradas en el UAS a diferencia de otros métodos.

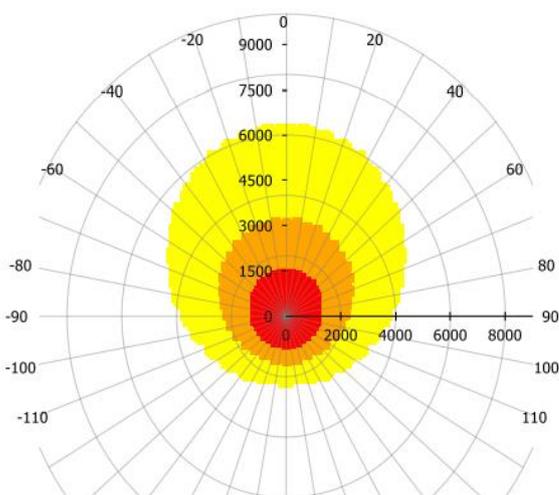


Figura 4.6 Ejemplo de volumen de seguridad

4.1.3.2.1 Rendimiento esperado del método.

El rendimiento de este método estaría indicado por la capacidad de las celdas y depende de la “performance” de los drones. Actualmente los drones tienen grandes diferencias en su habilidad de DAA (Detect And Avoid) por lo que pedir un mínimo de DAA no es posible.

La principal ventaja de este método es el hecho de que los volúmenes más afinados reducen la separación entre usuarios manteniendo la seguridad. Esto a costa de los costos de computación relacionados con el cálculo de los volúmenes. Además, también permite la operación de aeronaves con unos niveles de “performance” más bajo.

4.1.3.3 Geovectores

Este método utiliza geovectores para limitar la movilidad en un área dada. El método consiste en definir un volumen dado, en este caso se mencionan primas definidos por latitud, longitud y altitud mínimas y máximas, y los geovectores que se pueden aplicar a este volumen. Un geovector está definido por una GS (Ground Speed), un VS (Vertical Speed) y un rumbo mínimo y máximo, así como un tiempo durante el que están vigentes los geovectores, un ejemplo de estos geovectores se puede observar en la Figura 4.7. Este método no evita conflictos en sí mismo por lo que un servicio de SCR debería ser

igualmente implementado.

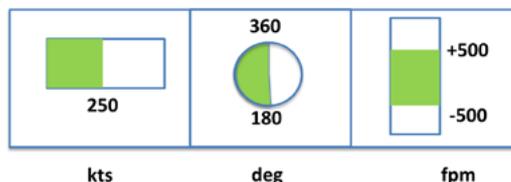


Figura 4.7 Ejemplo de Geovector

4.1.3.3.1 Rendimiento esperado del método

Para garantizar un rendimiento adecuado, los sistemas externos deben ser capaces de localizar correctamente el dron dentro de su área correspondiente con un error mínimo, y es posible que sea necesario más de un sistema trabajando en tándem.

Además, la infraestructura de comunicación debe transmitir los mensajes en un tiempo mínimo. Los operadores y USSPs deben cumplir con un conjunto de estándares para proporcionar servicios en una región con regulaciones de geovectorización. El diseño adecuado de los geovectores debe tener en cuenta las excepciones, como las misiones de observación o vigilancia que no permiten límites de velocidad o dirección.

4.1.3.4 Relacionar la estructura del espacio aéreo y la capacidad

Este método propone diseños de estructuras de espacio aéreo complementarias. En primer lugar, el método distingue entre UAVs y "Personal Aerial Vehicles" (PAVs) y divide el espacio aéreo en dos zonas, de 1100 ft a 1650 ft para UAVs y de 1650 a 6500 ft para PAVs. Adicionalmente este método Asume que cada vehículo es responsable de la detección y resolución de conflictos tácticos.

4.1.3.4.1 Full Mix

Esta estructura, en realidad, se trata de una "no estructura", donde las aeronaves solo están sujetas a restricciones físicas al planificar su ruta. Por lo tanto, las trayectorias seguidas por las aeronaves son directas, lo que resulta en ahorros de combustible.

4.1.3.4.2 Capas

Este concepto divide verticalmente el espacio aéreo en bandas, cada una con un rango de direcciones permitido, agrupando así todo el tráfico que se dirige en la misma dirección disminuyendo la probabilidad de un evento de LoS Figura 4.8.

Second Set PAV Layers	
Safety Layer	6450 ft
Level Layer (315 to 360°)	6150 ft
Level Layer (270 to 315°)	5850 ft
Level Layer (225 to 270°)	5550 ft
Level Layer (180 to 225°)	5250 ft
Level Layer (135 to 180°)	4950 ft
Level Layer (90 to 135°)	4650 ft
Level Layer (45 to 90°)	4350 ft
Level Layer (0 to 45°)	4050 ft
First Set PAV Layers	
Level Layer (315 to 360°)	3750 ft
Level Layer (270 to 315°)	3450 ft
Level Layer (225 to 270°)	3150 ft
Level Layer (180 to 225°)	2850 ft
Level Layer (135 to 180°)	2550 ft
Level Layer (90 to 135°)	2250 ft
Level Layer (45 to 90°)	1950 ft
Level Layer (0 to 45°)	1650 ft

Figura 4.8 Ejemplo de estructura en capas

4.1.3.4.3 Zonas

Este concepto consiste en dos elementos, radiales y anulares como se observa en Figura 4.9. Los anillos se comportan como aerovías, mientras que los radiales permiten el tránsito entre estos.

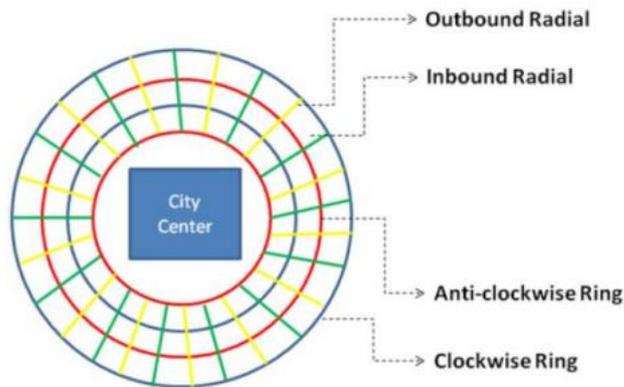


Figura 4.9 Ejemplo de estructura en zonas

4.1.3.4.4 Tubos

Este es el concepto más estructurado de todos. El espacio está definido por tubos de cuatro dimensiones. Los tubos de baja altitud son relativamente densos. Por otro lado, en las altitudes más altas se definen tubos de baja densidad, alta velocidad y larga distancia. Los vuelos cortos se benefician de redes más finas en las altitudes más bajas. Además, las capas están diseñadas de tal manera que los vuelos pueden ascender una capa a la vez.

La separación de los vuelos se introduce prescribiendo la misma velocidad para una altitud particular. Por lo tanto, los posibles conflictos ocurren solo en los nodos. Cuando un avión ocupa un nodo, el conflicto se resuelve retrasando el siguiente vuelo. El

siguiente vuelo sigue la ruta que se define en función de la distancia más corta. No despegamos hasta que la ruta esté completamente libre. Si no hay una ruta disponible, el vuelo se pospone durante 10 segundos y luego se aplica el mismo algoritmo.

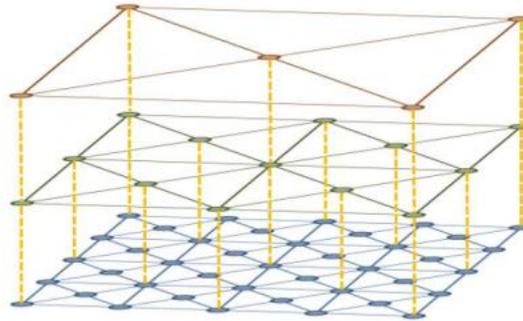


Figura 4.10 Ejemplo de estructura de tubos

4.1.3.5 Concepto de pasillos de densidad dinámica

Tras un análisis de los métodos anteriores, USEPE, ha diseñado un nuevo método de separación llamado Concepto de pasillos de densidad dinámica *D2-C2* (Figura 4.11) por sus siglas en inglés (Dynamic Density Corridor Concept). En *D2-C2* el espacio aéreo se organiza en segmentos dinámicos para permitir una configuración rápida, flexible y eficiente del espacio aéreo, e incluye pasillos de alta velocidad definidos por geovectores. Las transiciones de estos pasillos a los segmentos se manejan de forma que cada nuevo usuario que entra en un segmento de espacio aéreo se le asigna un volumen de seguridad con respecto a los demás habitantes de la celda y se genera una ruta. Si esto no es posible sin crear un conflicto, la celda ha alcanzado su límite de capacidad y la operación solicitada deberá ser reprogramada, re-enrutada o rechazada. Por lo contrario, las transiciones de los segmentos a los pasillos deben ser aceptadas por el USSP, de igual forma que en el caso de los pasillos de drones.

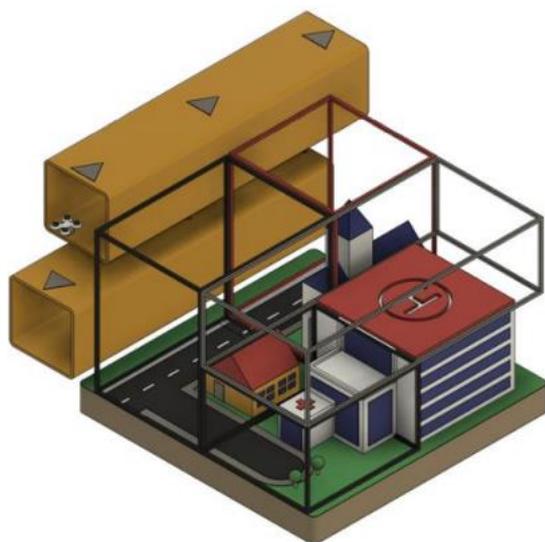


Figura 4.11 Disposición del espacio aéreo para el método D2-C2

4.1.4 SPATIO

SPATIO es un proyecto de investigación industrializada que empezará en septiembre de 2023 y que pretende desarrollar hasta un nivel de madurez elevado (TRL7) varios servicios U-space avanzados centrados en la separación, llegando hasta su industrialización y su validación mediante pruebas de vuelo en entornos reales. Para ello, SPATIO propone cuatro soluciones, agrupadas en dos Paquetes de Trabajo (WP1 y WP2) [15].

En el WP1, SPATIO abordará la gestión de separación mediante la deconflicción estratégica de los *U-plans*. El objetivo es garantizar una separación segura y eficiente, teniendo en cuenta la capacidad del U-space. En la solución 1A, SPATIO desarrollará un *DCMS* (Dynamic Capacity Management Service) que se centrará en la gestión dinámica de la capacidad del espacio aéreo disponible. En la solución 1B, SPATIO proporcionará un *SCRS* que irá más allá de las regulaciones existentes al incorporar mejoras como reglas de equidad (*Fairness rules*) y teniendo en cuenta entornos multi-USSP.

En WP2, SPATIO se centrará en la gestión táctica de separación y su relación con el *SCRS*. Aquí se proponen dos enfoques para la implementación de la detección y resolución táctica de conflictos. La primera opción (Solución 2A) implica la implementación manual de las instrucciones de resolución por parte del piloto remoto utilizando una Estación de Control en Tierra. La segunda opción (Solución 2B) aprovecha la comunicación aire-tierra para transmitir instrucciones de comando directamente al *UAS* para su implementación autónoma. En este caso, el *UAS* será capaz de ejecutar de manera autónoma las instrucciones recibidas.

El grupo SNA es uno de los participantes de este proyecto, con representación en ambos paquetes de trabajo. El presente proyecto y la herramienta desarrollada en el mismo servirán como punto inicial al trabajo de la solución 1B.

4.1.4.1 Modelo SPATIO

Para el diseño de los servicios propuestos, SPATIO sigue un enfoque basado en el riesgo. Este enfoque consiste en considerar el riesgo de fatalidad debido a colisiones como el factor limitante para evaluar la viabilidad de una operación o la viabilidad de todas las operaciones que podrían producirse simultáneamente en un determinado volumen de espacio aéreo. El cálculo del riesgo se basa en una predicción de las trayectorias, y, por tanto, como todas las predicciones, implica probabilidades.

Teniendo esto en cuenta, el servicio de separación y el *DCMS* están fuertemente ligados. El *SCRS* es activado cuando la probabilidad de que se produzca una *LoS* sea demasiado alta. De todas formas, el riesgo nunca va a poder ser cero, pero puede considerarse aceptable y luego manejado por el *TCRS*. El riesgo residual es función de muchos parámetros uno de los cuales es el número de trayectorias activas en el tiempo de ejecución. Por lo que, para cualquier esquema de gestión de separaciones, existe una densidad de vuelos instantánea máxima segura. El objetivo del *DCMS* es evitar que este máximo se supere.

El enfoque basado en riesgo se usará para determinar el esquema aplicable de gestión de separación asociado a un determinado espacio aéreo *U-space*. Cualquier esquema implica una definición de las separaciones mínimas aplicables juntamente con el modo de separación y el separador (reglas y responsabilidades).

Basándose en *BUBBLES*, SPATIO define la separación mínima como la mínima distancia, o tiempo que deben mantener dos aeronaves tal que el riesgo de colisión sea menor el requerido por el *TLS*. La separación mínima también puede ser dinámica cuando el valor dependa de factores como la hora del día, características ambientales, estructura del espacio aéreo, etc. Teniendo todo esto en cuenta, SPATIO identificará aquellas variables con mayor impacto en la separación mínima y creará un set de escenario estándar basado en estas variables.

SPATIO tendrá en cuenta el enfoque basado en el riesgo y los modelos de riesgo de colisión existentes (*BUBBLES* y *DACUS*, entre otros) para definir un nuevo servicio de *U-space* al que denominan "Separation Management service" que proporcionará el esquema de gestión de la separación aplicable. Además, SPATIO considera que ambos, el esquema de gestión de separación y el objetivo de capacidad inicial, se deben determinar previamente a que ningún *U-plan* sea enviado, para que esta información sea usada por el *SCRS*, *TCRS* y *DCMS*. A continuación, se adjunta una imagen de la interrelación de los servicios descritos.

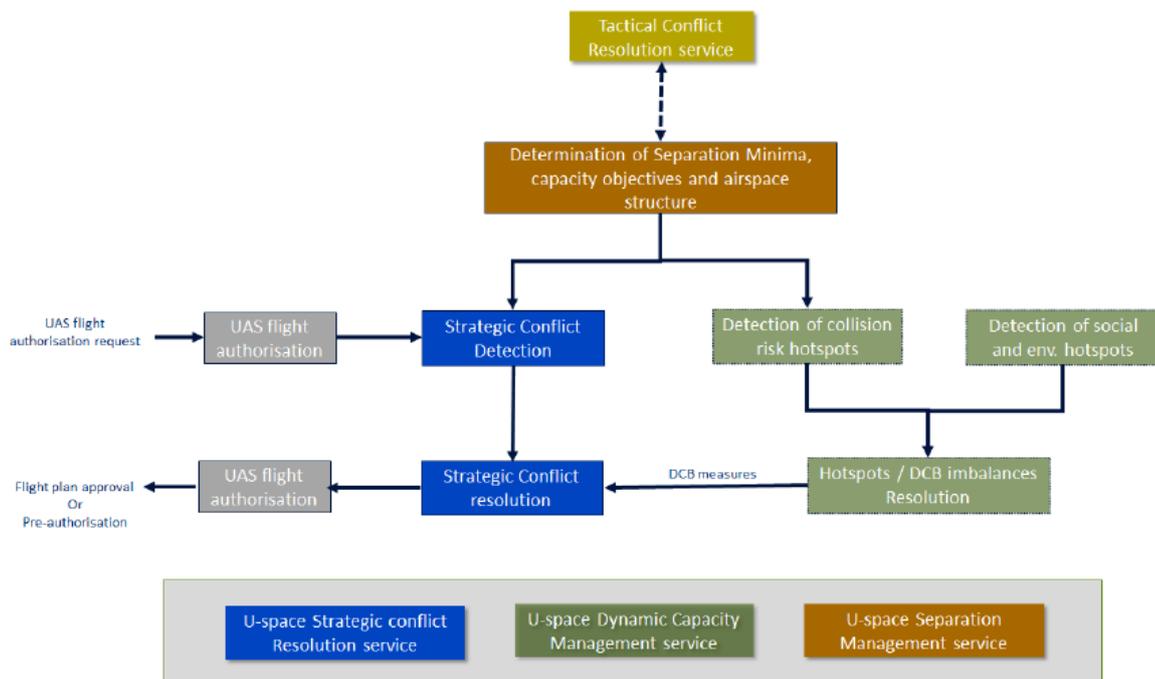


Figura 4.12 Relación entre los servicios propuestos por SPATIO

SPATIO ofrece una mayor explicación sobre cómo deben actuar los diferentes servicios y consideraciones para tener en cuenta en su propuesta.

5 DESAFIOS EN EL DESARROLLO DE HERRAMIENTAS PARA LA DETECCIÓN DE CONFLICTOS ESTRATÉGICOS ENTRE UAS

A primera vista, el desarrollo de herramientas para la detección de conflictos estratégicos entre UAS puede parecer un proceso relativamente sencillo. Sin embargo, a medida que nos sumergimos en la investigación y desarrollo de estas herramientas, nos encontramos con una serie de desafíos que requieren atención y soluciones adecuadas.

En este apartado, se abordarán los desafíos que surgieron durante el desarrollo de la herramienta para la detección de conflictos estratégicos entre UAS. Estos desafíos representan obstáculos significativos que debemos superar para lograr una detección precisa, efectiva y sobre todo suficientemente rápida de los posibles conflictos entre los UAS para que el U-space sea una realidad.

5.1 Intersección de trayectorias 4D: complejidad y necesidad de discretización

Aunque las trayectorias se definen mediante volúmenes 4D, calcular la intersección entre estos volúmenes en general es extremadamente complejo.

Para determinar si los volúmenes 4D se intersecan, primero se requeriría encontrar una manera de parametrizar estos volúmenes y luego aplicar algoritmos complejos para calcular la intersección de estos hipervolumenes. De hecho, según [16]: *“For the intersection, we prove that computing the facets or the vertices of the intersection of two polytopes is NP-hard⁶ if one of them is given by vertices and the other by facets. Also, computing the vertices of the intersection of two polytopes given by vertices is shown to be NP-hard”*.

Como se dice en el párrafo anterior, calcular la intersección entre dos volúmenes generales en 4 dimensiones es computacionalmente muy costoso, por lo que, para tener una herramienta eficaz, es necesario discretizar las trayectorias si deseamos que el problema tenga tiempos de resolución suficientemente cortos para que el mecanismo de detección y resolución de conflictos sea suficientemente rápido para el entorno tan automatizado que se pretende con el U-space. Discretizar una trayectoria consiste en dividir una trayectoria continua en una serie de puntos en intervalos regulares de tiempo. Sin embargo, discretizar tiene algunos inconvenientes muy claros, y es que existe la probabilidad de que si se dan ciertas condiciones se produzca un evento de LoS indetectado, tal y como se ve en la Figura 5.1.

⁶ Un problema con complejidad NP indica que no se conoce ningún algoritmo que lo resuelva en un tiempo de orden polinomial.

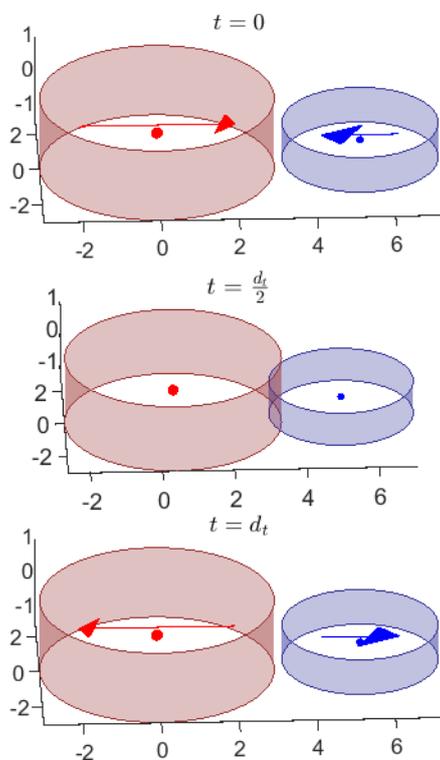


Figura 5.1 LoS indetectable debido a la discretización temporal

En la figura se muestran dos UAS diferentes (rojo y azul) representados por sus volúmenes de protección. La flecha indica el sentido de movimiento que siguen en cada instante de tiempo. En la parte superior e inferior de la imagen se muestra a ambos UAS en un instante $t=0$ y $t=d_t$ en el que se comprobaría si existe una pérdida de separación, LoS. En el instante $t=0$ los UAS se acercan de frente y, de repente, se dan la vuelta y en el instante $t=d_t$ se alejan. Así pues, cuando se compruebe el LoS en este caso, el resultado sería negativo, aunque en un instante intermedio (no comprobado) sí que hayan estado intersecando ambos volúmenes, como se observa en la figura del medio en $t= d_t/2$. La probabilidad de que esto suceda dependerá del intervalo de tiempo escogido, de la velocidad, y de las trayectorias de la aeronave.

El problema de la no detectabilidad de LoS debido a la discretización se puede abordar de 2 formas:

5.1.1 Enfoque estadístico

Uno de los enfoques más efectivos consiste en llevar a cabo un análisis estadístico del número de incidentes de pérdida de separación de este tipo y ajustar el paso de tiempo discreto, al que a partir de ahora llamaremos “delta time o dt ”, para cumplir con los estándares aceptables. Este enfoque resulta especialmente atractivo debido a la existencia prevista de un sistema de resolución de conflictos tácticos en tiempo real, lo que significa que no es estrictamente necesario detectar el 100% de los eventos. Esta

estrategia ofrece una mayor eficiencia desde un punto de vista de capacidad, garantizando un nivel óptimo de seguridad.

5.1.2 Enfoque seguro

Una alternativa adicional consiste en ampliar los niveles de separación incluyendo en la desviación lateral y vertical de la que hablábamos en la sección 3.1.2.2 de acuerdo con la Ecuación 2 y Ecuación 2, representada en la Figura 5.2. A diferencia del método anterior, este enfoque tendría como consecuencia una reducción en la capacidad del espacio aéreo.

$$R = r + GS * \frac{dt}{2} \quad \text{Ecuación 1}$$

$$H = h + RoC * \frac{dt}{2} \quad \text{Ecuación 2}$$

Donde:

R = Valor de la desviación horizontal ampliada.

H = Valor de la desviación vertical ampliada.

r = Valor previo de la desviación horizontal.

h = Valor previo de la desviación horizontal.

GS = Velocidad horizontal.

dt = Delta Time.

RoC = Velocidad vertical.

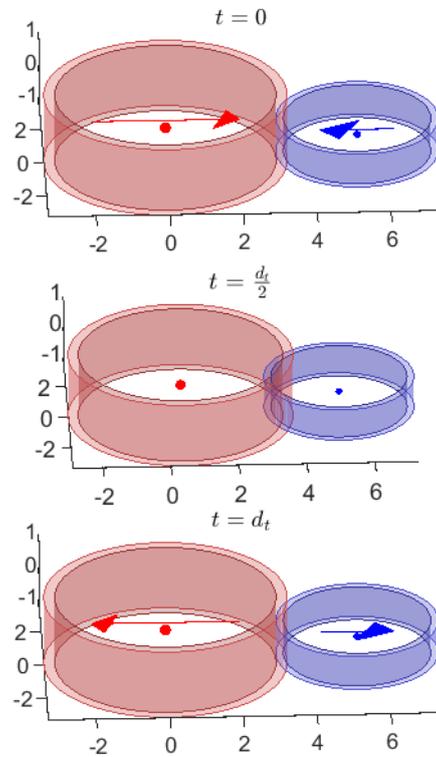


Figura 5.2 Enfoque seguro sobre las pérdidas de separación debido a la discretización temporal

En el apartado 7.2.1 "Ratio de conflictos detectados" se realiza un estudio de la cantidad de conflictos detectados, concluyéndose que para un dt suficientemente pequeño, el enfoque estadístico es lo suficientemente bueno y cumple con el TLS esperado.

5.2 Cantidad de comprobaciones

Tras haber concluido que vamos a discretizar la dimensión temporal, el próximo desafío a abordar es la cantidad de comparaciones que se deben hacer para comprobar si se produce alguna LoS. Para un momento dado, el número de operaciones que habría que hacer, sin ningún método de optimización, en un espacio aéreo con n *U-plan* sería igual al número de combinaciones posibles en las que se pueden agrupar n elementos en grupos de 2 elementos sin importar el orden⁷. Esta fórmula, generalizada para grupos de m elementos, se describe en [17] y se representa como:

$$C(n_U) = \binom{n}{m} = \frac{n!}{m!(n-m)!}$$

Ecuación 3

⁷ El conflicto entre 2 UAS no es direccional, es decir, si el UAS 1 conflictúa con el UAS 2, el UAS 2 también lo hará con el UAS 1.

Donde:

C = Número de ordenaciones posibles

n = Número de *elementos a agrupar*

m = Número de elementos por grupo

Que particularizando para grupos de 2 UAS ($m=2$), obtenemos:

$$C(n_U) = \frac{n_U!}{2!(n_U - 2)!} = \frac{n_U \cdot (n_U - 1)}{2} = \frac{n_U^2 - n_U}{2} \quad \text{Ecuación 4}$$

Donde:

C = Número de comparaciones necesarias

n_U = Número de *U-plan* simultáneamente activos

Además, la fórmula de la Ecuación 4 es solamente válida en un instante de tiempo, si lo extendemos durante un tiempo " t " obtenemos la fórmula de la Ecuación 5:

$$C(n_U, t, d_t) = \frac{t}{d_t} \cdot \frac{n_U^2 - n_U}{2} \quad \text{Ecuación 5}$$

Donde:

C = *Número de comparaciones necesarias*

n_U = *Número de U-plan simultáneamente activos*

t = *Tiempo total*

d_t = *Delta time*

De esta forma, el número de comprobaciones crece muy rápidamente, tal y como se observa en la Figura 5.3. Por poner un ejemplo numérico, en el caso de que haya 10 planes de vuelo simultáneos durante una hora con $dt=1$ segundo realizaríamos 162.000 comprobaciones; ante la misma situación, pero con 50 planes de vuelo, alcanzaríamos las 4.410.000 comprobaciones; y para 250 planes superaríamos las 100.000.000 de comprobaciones.

Esto limitaría enormemente la cantidad de *U-plans* que esta herramienta puede procesar teniendo en cuenta las limitaciones impuestas por los estándares, como el que

se define en [4]: “A possible interpretation of the expression ‘without unjustified delay’ is provided in ASTM F3548-21 ‘Standard Specification for UAS Service Supplier (USS) Interoperability’, which requires that the activation of flight authorisations be confirmed within 5 seconds for 95 % of the time.”

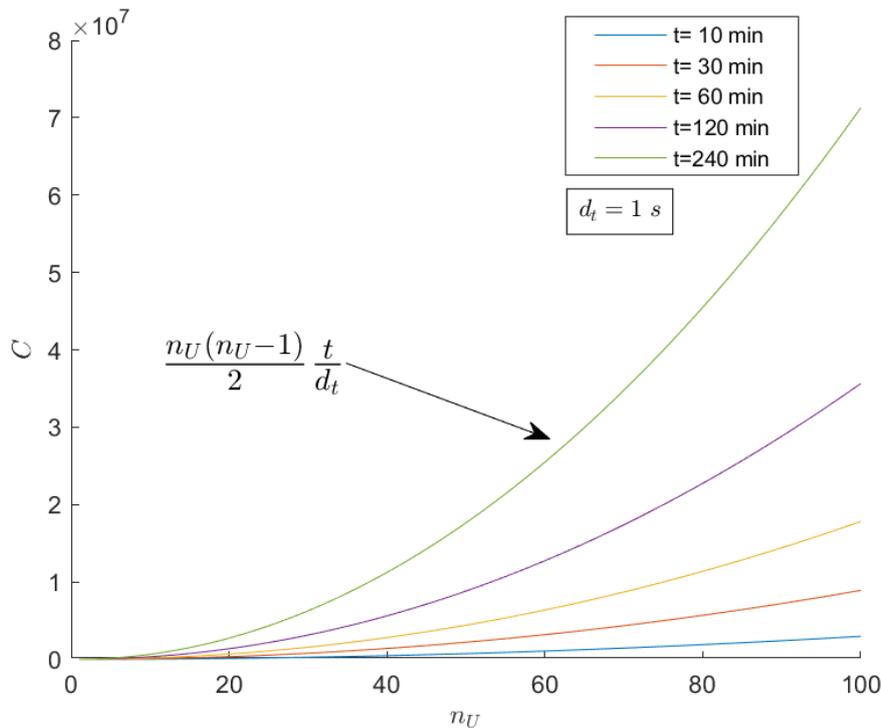


Figura 5.3 Variación de las operaciones requeridas en función del número de U-plan activos para distintos tiempos

En esta figura se representa la relación entre la cantidad de comprobaciones que se necesitan realizar y el número de UAS y tiempo de vuelo. En ella se puede apreciar lo rápido que escala con ambas variables. Nótese que el tiempo de 240 min no significa que tenga que haber 100 UAS volando durante 2 horas, ya que vuelos tan largos no serán comunes en el U-space, sino que hay simultáneamente volando 100 UAS durante ese tiempo.

Frente al problema de la cantidad de comparaciones, la respuesta más lógica es buscar algún método para no tener que comprobar la distancia de todos los UAS respecto a todos los UAS, pues deberíamos de ser capaces de descartar los más lejanos, que no son de interés puesto que ya que sabemos que no son capaces de producir LoS. Para lograr este propósito vamos a estudiar la viabilidad de aplicar distintas estructuras de datos.

5.2.1 Estructuras de datos

Las estructuras de datos son formas organizadas de almacenar y manipular datos en un programa o sistema informático. Son esenciales para gestionar eficientemente grandes

volúmenes de información y permiten realizar operaciones de manera eficiente, Para representar datos espaciales, existen varias estructuras de datos especializadas que permiten organizar y manipular eficientemente información relacionada con la ubicación y geometría de objetos en un espacio.

De todas las estructuras de datos existentes, algunas de las más interesantes para nuestro problema concreto son las estructuras: Grid, octree, R-Tree y Kd-Tree.

5.2.1.1 Grid

La estructura Grid, también conocida como malla o cuadrícula, es una estructura de datos bidimensional que divide el espacio en una matriz regular de celdas. Cada celda representa una región espacial del mismo tamaño y se utiliza para almacenar objetos o datos que se encuentran dentro de ella. Esta estructura es fácil de implementar y ofrece una búsqueda rápida en coordenadas discretas, pero puede tener limitaciones en términos de escalabilidad y eficiencia en espacios tridimensionales o con densidad variable.

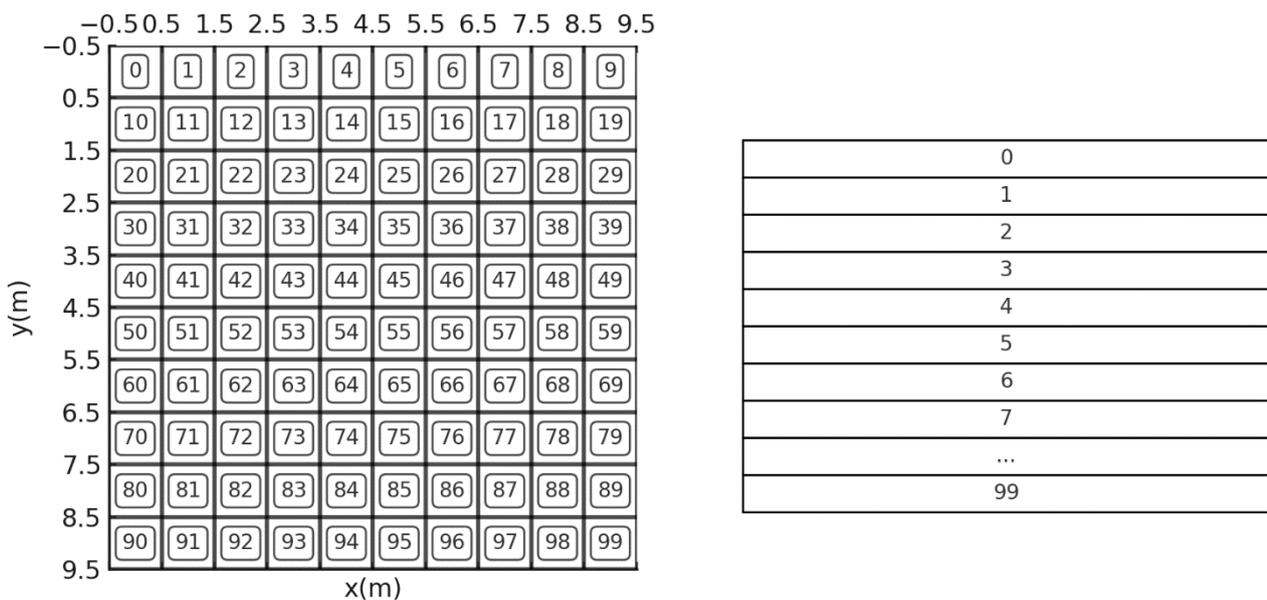


Figura 5.4 Estructura de datos en "Grid"

5.2.1.2 Octree

Un octree, o árbol de octantes, es una estructura de datos jerárquica y se basa en el principio de subdivisión recursiva del espacio en regiones más pequeñas conocidas como octantes. Cada octante representa una porción del espacio tridimensional y se divide en otros ocho octantes más pequeños si es necesario, como puede observarse en la Figura 5.5. La complejidad al realizar una búsqueda en de datos es de tipo logarítmico [18]: $\mathcal{O}(\log(n))$.

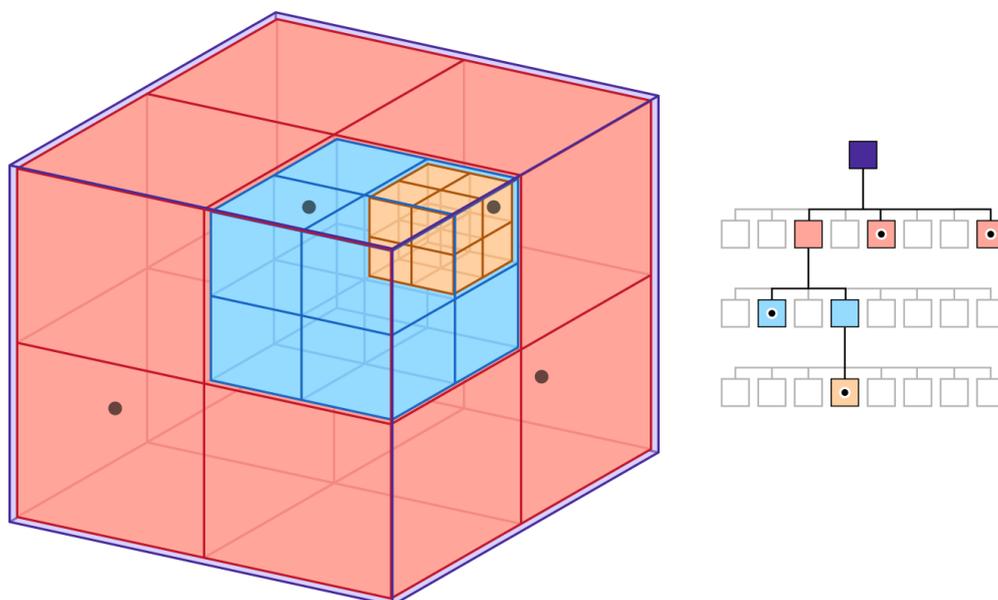


Figura 5.5 Diagrama octree

5.2.1.3 R-Tree

Un R-tree es una estructura de almacenamiento e índice similar a un árbol basado en un rectángulo de límite mínimo (MBR, por sus siglas en inglés). La "R" del R-tree es la abreviatura de "Rectangle" (rectángulo). Como se muestra en la Figura 5.6, se asume que hay 12 objetos espaciales irregulares diferentes rodeados por 12 MBRs (R8-R19), que se toman como los nodos hoja que solo tienen un nodo padre y no tienen nodos hijos. Además, los MBRs con una distancia cercana se clasifican y se rodean con MBRs más grandes (R3-R7), que son los nodos internos que tienen tanto nodos padre como nodos hijos. Este proceso continúa hasta que solo dos MBRs (R1 y R2) forman el nodo raíz que solo tiene nodos hijos y no tiene nodo padre.

Una vez construido, el R-tree se puede utilizar para obtener rápidamente los objetos espaciales dentro del rango del punto P (línea azul gruesa, en adelante denominada cuadro de búsqueda). En cuanto a operaciones necesarias, el algoritmo para búsquedas

en esta estructura de datos también es de tipo logarítmico [19]: ($\mathcal{O}(\log(n))$).

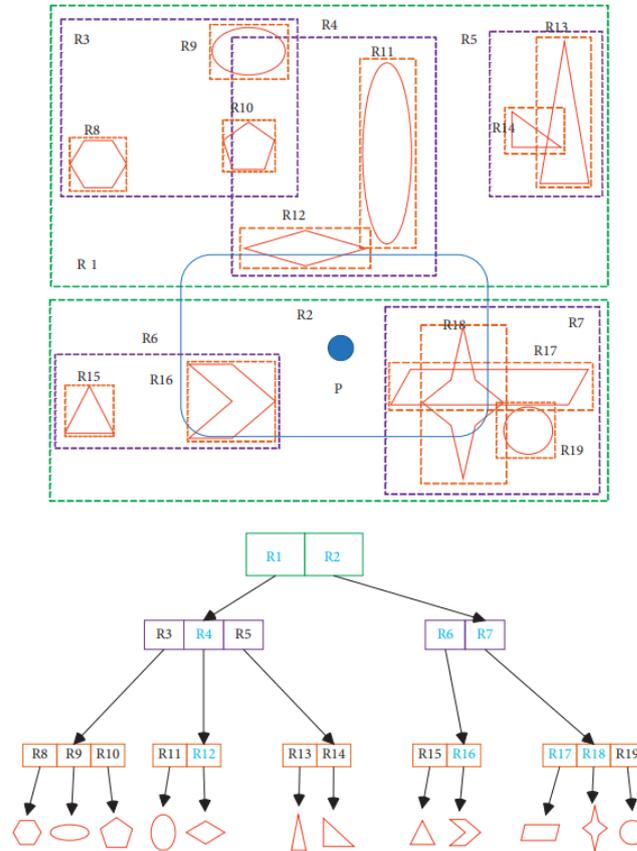


Figura 5.6 Diagrama esquemático de la estructura de datos R-tree

5.2.1.4 Kd-Tree

El “k3-tree” (árbol k-dimensional) es otra estructura de datos utilizada para particionar el espacio en múltiples dimensiones. A diferencia del R-tree y el octree, el kd-tree particiona el espacio de forma binaria en cada nivel, utilizando hiperplanos ortogonales a los ejes. Es eficiente en términos de búsqueda de vecinos cercanos en espacios de alta dimensión, pero puede tener un rendimiento inferior en espacios tridimensionales y con densidad variable.

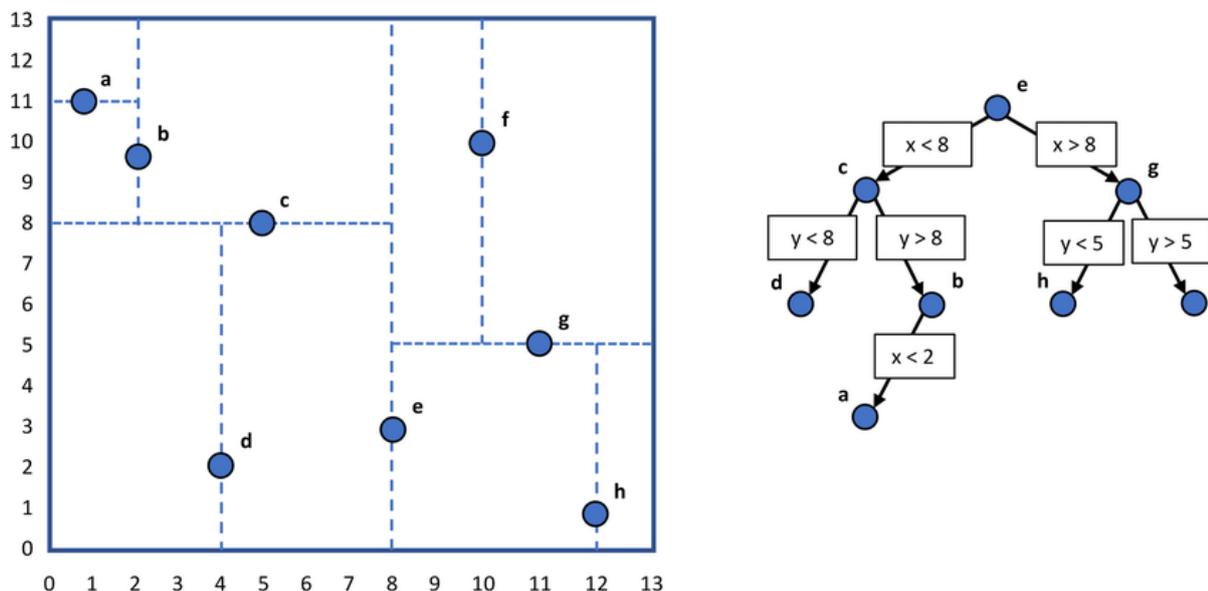


Figura 5.7 Diagrama kd-tree

5.2.1.5 Selección de la estructura de datos óptima para el SCRS.

Todas estas estructuras reducen enormemente el número de operaciones a realizar. Aunque todas tienen sus ventajas y desventajas, finalmente se ha optado por utilizar Octree debido a las siguientes razones:

- Flexibilidad**
 Tanto el kd-tree, como el R-tree y el Octree son estructuras de datos flexibles que se adaptan a los elementos que contienen. Sin embargo, una estructura de datos en forma de grid puede resultar demasiado rígida, lo que puede llevar a una representación ineficiente de los elementos, especialmente cuando existe una gran variación en su distribución. Si las celdas son pequeñas, se generarán numerosas celdas vacías en zonas con baja densidad de elementos, lo que dará lugar a un exceso de operaciones innecesarias. Por otro lado, si las celdas son grandes, se reducirá la capacidad del espacio aéreo, ya que las celdas no se adaptarán correctamente a los volúmenes de protección de los UAS.
- Algoritmos de búsqueda**
 El kd-tree es especialmente efectivo para buscar los vecinos más cercanos, pero no es eficiente cuando se trata de buscar dentro de una región específica, especialmente si la forma de la región no es esférica. Por otro lado, tanto el octree como el R-tree ofrecen algoritmos de búsqueda basados en regiones que son eficientes, siempre y cuando existan algoritmos para verificar las intersecciones entre estas regiones y los prismas rectangulares.
- Inserción de elementos**
 En cuanto a la inserción de elementos, tanto el octree como el R-tree tienen algoritmos rápidos. Sin embargo, hay una peculiaridad con el algoritmo de R-tree: es altamente eficiente cuando todos los puntos se tienen disponibles antes de

iniciar la construcción. En caso contrario, si los puntos se agregan de manera aleatoria, puede resultar en MBRs no óptimos, lo que ralentizaría significativamente el algoritmo de búsqueda.

Así pues, la decisión final ha sido la de utilizar la estructura de datos octree por ser la que más ventajas ofrece frente a las desventajas.

Tras esta decisión, en la Figura 5.8 se muestra una comparación de la cantidad de comprobaciones a realizar por el método de fuerza bruta (el descrito al inicio de la sección 0) y utilizando la estructura de datos de octree. Es importante destacar que se ha graficado el orden de los algoritmos, (ej. $k_1 \cdot n^2 + k_2 \cdot n = O(n^2)$) que puede distar mucho de las comprobaciones reales. Para obtener los datos exactos se debería realizar un test de caja blanca después de su implementación.

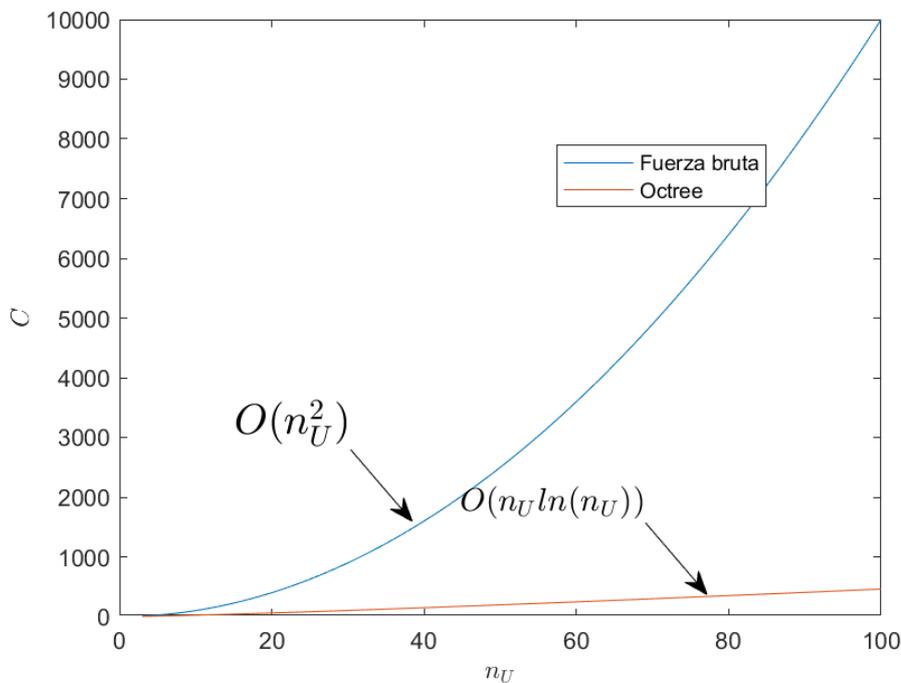


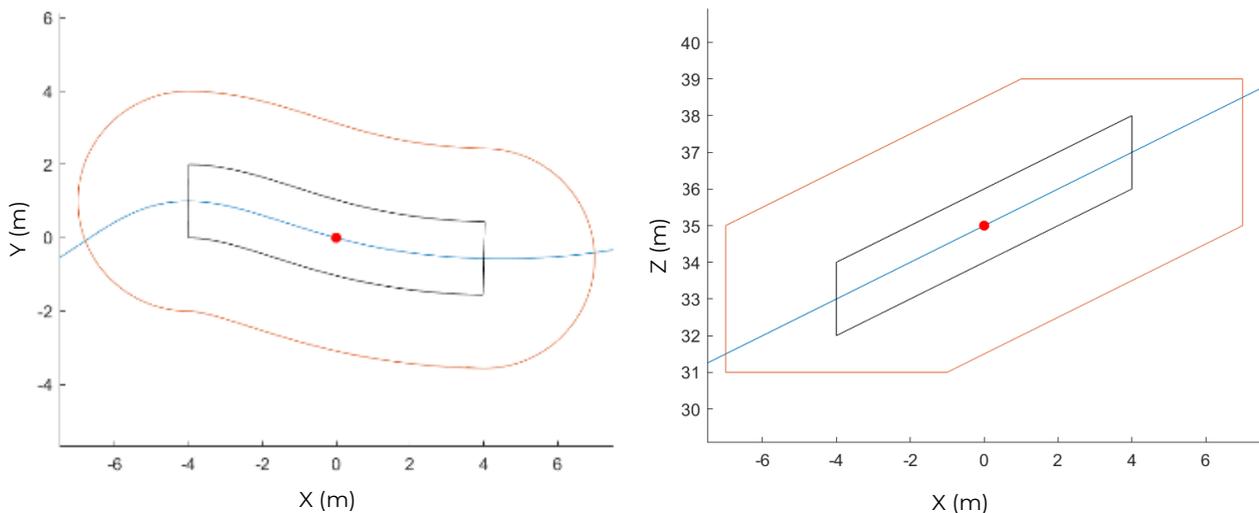
Figura 5.8 comparación del orden de "octree" vs fuerza bruta

5.3 Tiempo de protección

Otro desafío al que nos enfrentamos es encontrar una manera de incorporar un tiempo de protección a la trayectoria para tener en cuenta posibles desviaciones o incertidumbres en el inicio del vuelo, ya sean retrasos o adelantos, así como la degradación de la trayectoria a lo largo del tiempo.

Este tiempo de protección provoca una deformación en el volumen de protección, lo que implica que deja de tener forma de cilindro. A continuación, analizaremos paso a paso este proceso.

En la Figura 5.9 puede observarse una trayectoria (línea en azul) y el tiempo de protección, que se simboliza con la región delimitada por las líneas negras, lo que indica que la trayectoria nominal junto con el tiempo de protección permitiría (sin tener en cuenta el volumen de protección) que el UAS se encontrase en cualquier punto de la trayectoria dentro de esa región. En la misma figura se observa el volumen de protección (delimitado por la línea naranja). Este volumen es el resultado de desplazar el volumen de protección a lo largo de la trayectoria para un momento dado.



a) distorsión del volumen de protección debido al tiempo de protección en el plano horizontal

b) distorsión del volumen de protección debido al tiempo de protección en el plano vertical

Figura 5.9 Volumen de protección extendido por el tiempo de protección

La creación de este volumen y sus intersecciones son complejas por lo tanto en este proyecto se propone una nueva técnica a la que hemos denominado “Inserción de puntos virtuales”. Esta técnica consiste en insertar puntos en un instante temporal determinado, los cuales representan la posición de un UAS en diferentes instantes de tiempo dentro del tiempo de protección. Sin embargo, solo con esto no se cubre todo el volumen.

Para lograrlo, es necesario aumentar tanto el radio como la altura del volumen de protección original, como se muestra en la siguiente ecuación.

$$r' = \sqrt{\left(\frac{d_t * GS}{2}\right)^2 + r^2} \tag{Ecuación 6}$$

$$h' = h + |RoC| \left(d_t + \frac{r - r'}{GS} \right) \tag{Ecuación 7}$$

Donde:

r' = Radio de protección modificado

h' = Semi-altura de protección modificada

r = Radio de protección original

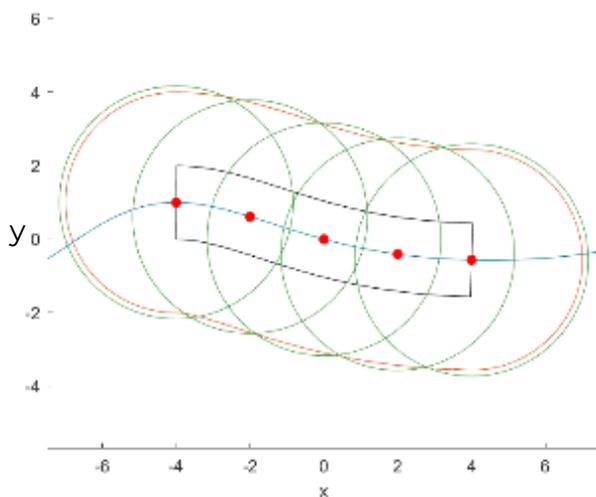
h = Altura de protección original

d_t = Delta Time

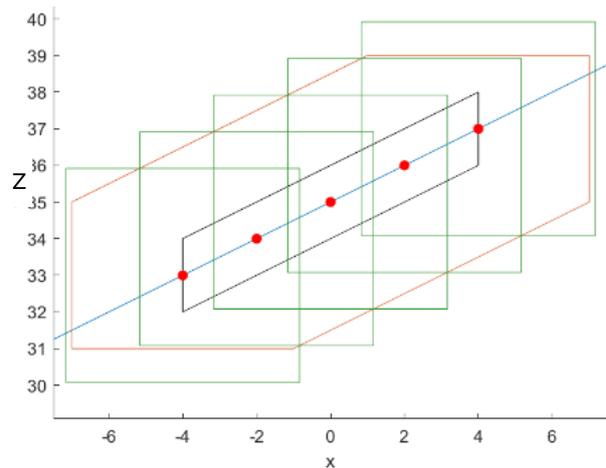
GS = Velocidad horizontal

RoC = Velocidad vertical

El resultado de esta técnica, con los volúmenes de protección de los puntos virtuales en verde, se puede observar en la Figura 5.10.



a) volumen de protección real y volumen de protección conseguido con puntos virtuales en el plano horizontal



b) volumen de protección real y volumen de protección conseguido con puntos virtuales en el plano vertical

Figura 5.10 Esquema de la técnica de puntos virtuales

5.3.1 Factor dt-Tiempo de protección

La técnica de los puntos virtuales tiene una desventaja puesto que implica añadir una cantidad mucho mayor de puntos, de forma que si anteriormente un *U-plan* lo discretizábamos con $\frac{t}{d_t}$ puntos, este método implica añadir $\frac{t}{d_t} \cdot \left\lceil \frac{t_p}{d_t} \right\rceil$ puntos, donde t_p es el tiempo de protección. En la Figura 5.11 se muestra el gran aumento de puntos que esto conlleva.

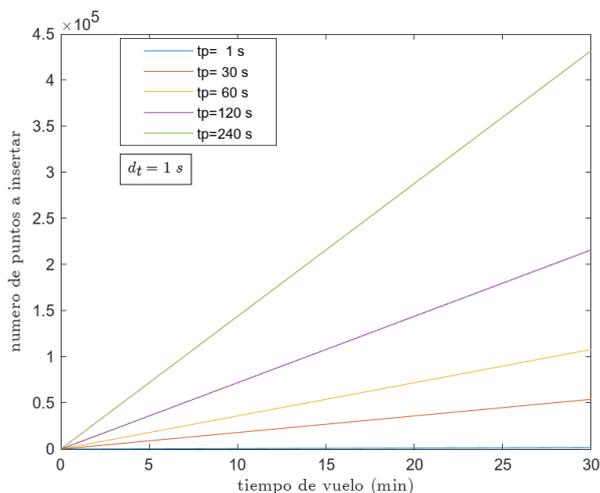


Figura 5.11 Aumento de puntos debido al tiempo de protección

Para solucionar este problema se introduce el factor d_t -tiempo de protección que lo que implica es que no se proyectan todos los puntos de la trayectoria, sino los que el factor dicte, insertando un total de $\frac{t}{d_t} \cdot \left\lceil \frac{t_p}{f \cdot d_t} \right\rceil$ puntos, con ello las fórmulas de Ecuación 6 y Ecuación 7 quedan:

$$r' = \sqrt{\left(\frac{d_t * GS * f}{2}\right)^2 + r^2} \quad \text{Ecuación 8}$$

$$h' = h + |RoC| \left(d_t * f + \frac{r - r'}{GS} \right) \quad \text{Ecuación 9}$$

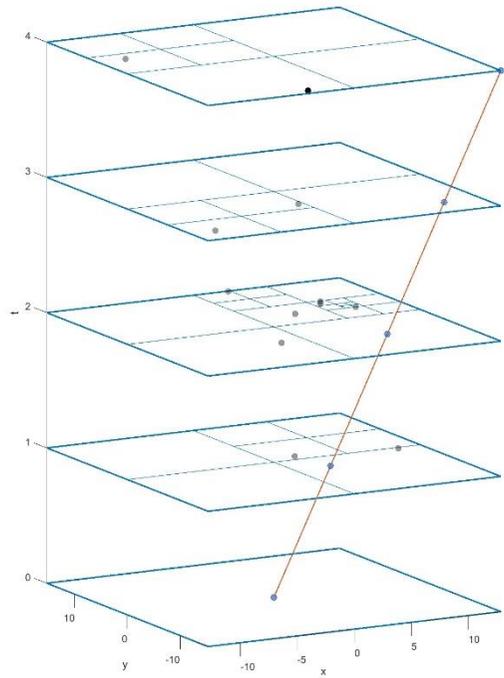
Donde:

- r' = Radio de protección modificado
- h' = Semi-altura de protección modificada
- r = Radio de protección original
- h = Semi-altura de protección original
- d_t = Delta time
- GS = Velocidad horizontal
- RoC = Velocidad vertical
- f = Factor dt-tiempo de protección

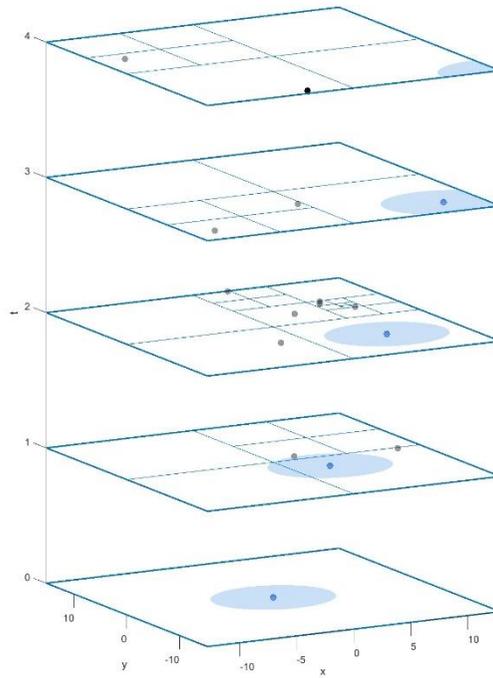
Como se puede observar, el factor afecta al radio, a la altura del cilindro de protección y, por tanto, a la capacidad del espacio aéreo, por lo que es necesario ajustarlo para

conseguir un buen balance entre capacidad y eficiencia.

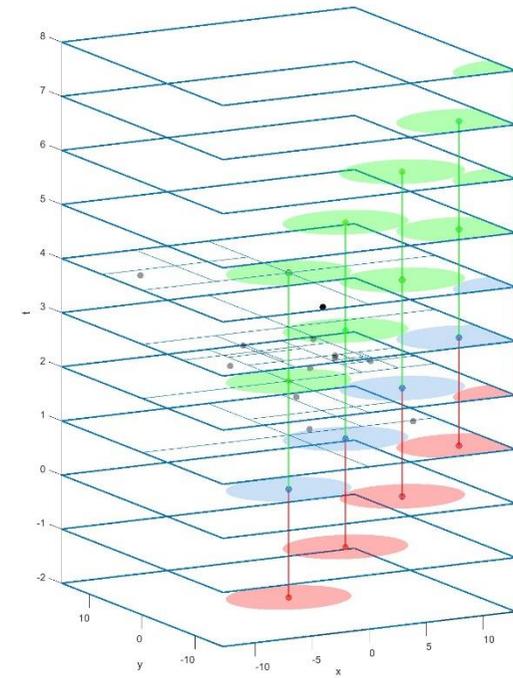
En la Figura 5.12 se puede observar una abstracción de la misma técnica a la vez que el algoritmo octree. En ella se ha reducido el número de dimensiones para poder visualizarla correctamente, en el eje z se representa el tiempo de forma discretizada.



a)



b)



c)

Figura 5.12 Diagrama de la tecina de puntos virtuales aplicada a octree

En la Figura 5.12 a) se observa la trayectoria del UAS y los puntos de la trayectoria discretizada cuando esta interseca en un plano donde $t = k \cdot d_t, k \in N$.

En la Figura 5.12 b) a cada uno de estos puntos se le añade el volumen de protección.

Por último, en la Figura 5.12 c) se proyectan los puntos tal y como se ha explicado anteriormente siguiendo la técnica de los puntos virtuales con un factor de 2, las zonas en verde son aquellas en las que se encontraría el UAS si saliese con retraso, mientras que las zonas son las que marcarían la posición del UAS si saliese antes de lo previsto.

Esta representación visual permite comprender cómo se aplican los puntos virtuales junto con el algoritmo octree para crear una cobertura adecuada del volumen de protección.

5.3.2 Disminución de la capacidad

Como se ha anticipado en el apartado anterior, la técnica de los puntos virtuales implica aumentar tanto el radio como la altura del volumen de protección de los UAS por lo que reducimos la capacidad del espacio aéreo.

En este apartado, vamos a analizar cómo y cuánto se ve afectada la capacidad del espacio aéreo por aplicar los puntos virtuales. El caso general es bastante complejo, por lo que calcularemos el caso particular en el que la trayectoria que siga el UAS es una línea recta con velocidad horizontal y vertical.

En primer lugar, calculemos el volumen de protección incluyendo la deformación debido a el tiempo de protección.

De la Figura 5.10 se observa que este volumen es la suma del cilindro de protección más el cuboideo que resulta de desplazar un rectángulo con lados, diámetro y altura del cilindro de protección a lo largo de la trayectoria:

$$V = 2h \cdot \pi r^2 + t_p GS \cdot 2r \cdot 2h = 2hr \cdot (2t_p GS + \pi r) \quad \text{Ecuación 10}$$

Donde,

V = Volumen de protección verdadero

r = Radio de protección original

h = Altura de protección original

GS = Velocidad horizontal

t_p = Tiempo de protección

A continuación, se calcula el volumen de uno de los cilindros de que envuelven a los puntos virtuales:

$$V_v = 2h' \cdot \pi r'^2$$

Ecuación 11

Donde,

V_v = Volumen cilindro punto virtual

r' = Radio de protección ampliado

h' = Altura de protección ampliado

Tras ello, obtiene el volumen de intersección de estos cilindros. Para ello, se calcula primero el área de intersección de dos círculos y luego, se multiplica por la porción de altura del cilindro que interseque:

$$\theta = 2 * \tan^{-1} \left(\frac{\sqrt{r'^2 - \left(\frac{d}{2}\right)^2}}{\frac{d}{2}} \right)$$

Ecuación 12

$$A_\theta = \frac{\theta}{2\pi} \cdot \pi r'^2 = \frac{\theta}{2} \cdot r'^2$$

Ecuación 13

$$A_\Delta = \frac{d}{2} \cdot \sqrt{r'^2 - \left(\frac{d}{2}\right)^2}$$

Ecuación 14

$$A_i = 2A_\theta - 2A_\Delta = d\sqrt{-0,25d^2 + r'^2} + 2r'^2 \tan^{-1} \left(\frac{\sqrt{-d^2 + 4r'^2}}{d} \right)$$

Ecuación 15

Con:

θ = Angulo del sector circular que forma la intersección

r' = Radio de protección modificado

d = Distancia entre los centros

A_θ = Área sector circular

A_Δ = Área triangular

A_i = Área intersección

Con la ayuda de la Figura 5.13, se explica a continuación el conjunto de ecuaciones anterior. En primer lugar, θ corresponde al ángulo formado entre el centro de la circunferencia y los puntos de intersección, por lo que A_θ es el área del sector circular de cpc'. Si se suman los sectores de ambos círculos, se tendría dos veces el área morada de la figura. Para evitar eso, se calcula A_Δ , que corresponde al área del triángulo formado por cpc'. Con esto finalmente ya se puede calcular el área de

intersección A_i (Ecuación 15).

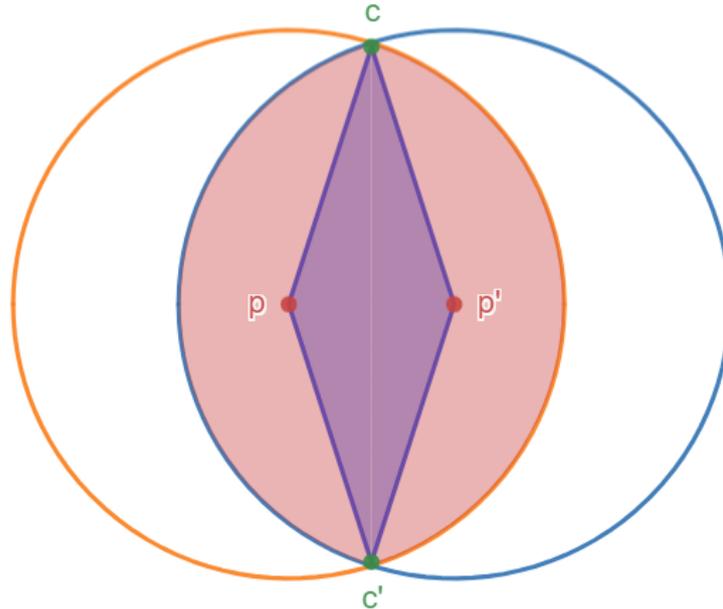


Figura 5.13 Diagrama área de intersección de dos circunferencias

A continuación, se calcula el volumen ocupado por todos los puntos virtuales junto a sus cilindros:

$$V_v = 2h' \cdot \pi(2r')^2 \left\lceil \frac{t_p}{d_{tf}} \right\rceil - \sum_{n=1}^{\left\lceil \frac{t_p}{d_{tf}} \right\rceil - 1} \left(\left\lceil \frac{t_p}{d_{tf}} \right\rceil - n \right) \cdot \max(A_{in} \cdot h_n, 0) \quad \text{Ecuación 16}$$

Con:

$$A_{in} = \left(d_n \sqrt{-0,25d_n^2 + r'^2} + 2r'^2 \tan^{-1} \left(\frac{\sqrt{-d_n^2 + 4r'^2}}{d_n} \right) \right) \quad \text{Ecuación 17}$$

$$d_n = GS * d_t * f * n \quad \text{Ecuación 18}$$

$$h_n = h' - Roc * d_t * f * n \quad \text{Ecuación 19}$$

Donde:

r' = Radio de protección modificado

h' = Semi-altura de protección modificada

d_t = Delta time

GS = Velocidad horizontal

RoC = Velocidad vertical

f = Factor dt-tiempo de protección

Seguidamente, se analiza la Ecuación 16 paso a paso. En primer lugar, hay que entender que $\left\lceil \frac{t_p}{d_{tf}} \right\rceil$, es el número de puntos virtuales que se insertan. El primer término, $2h' \cdot \pi(2r')^2 \left\lceil \frac{t_p}{d_{tf}} \right\rceil$, suma todos los volúmenes de los cilindros, mientras que el sumatorio resta las intersecciones.

Cada término del sumatorio, por tanto, está contando los volúmenes de intersección de orden n , es decir la intersección entre un punto virtual y el eneavo siguiente. Mientras que el término $\left(\left\lceil \frac{t_p}{d_{tf}} \right\rceil - n \right)$ cuenta el número de intersecciones.

Por último, la función \max se utiliza para no contar volúmenes negativos, que resultan del volumen de intersección cuando no intersecan los cilindros.

La siguiente figura muestra el incremento de volumen en relación con diferentes valores de tiempo de protección, delta time y factor dt-tiempo de protección.

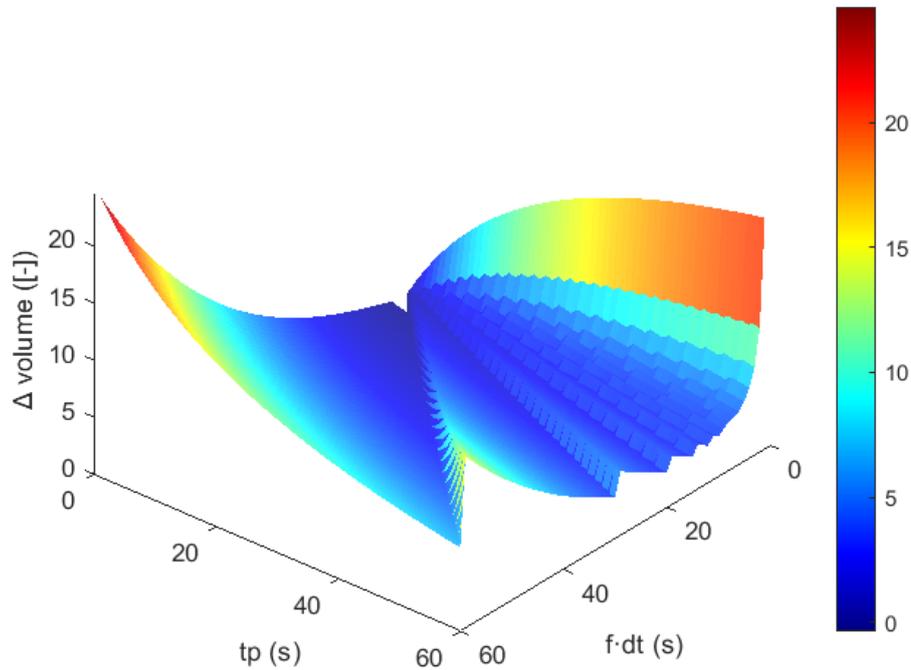


Figura 5.14 incremento de volumen debido a el tiempo de protección, el factor dt-tiempo de protección y dt

El incremento se calcula como $\frac{V_p - V}{V}$. Es muy llamativo los escalones que se ven, que corresponden a los planos $f \cdot dt = k \cdot t_p$ para $k \in \mathbb{N}$, y es causada por la función *ceil* de la Ecuación 16. Aparentemente el mínimo global de la función corresponde con el plano $f \cdot dt = t_p$. Para asegurarnos de ello, vamos a ver secciones de la figura, que pueden observarse en las siguientes imágenes:

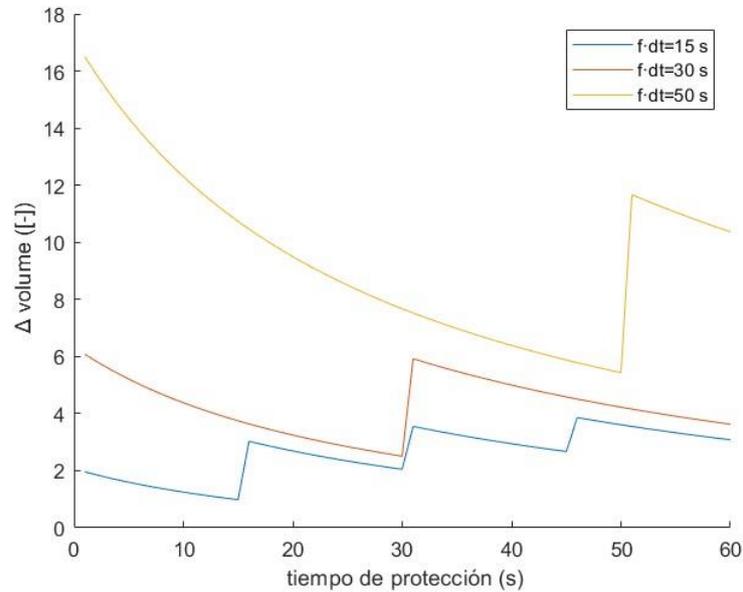


Figura 5.15 incremento de volumen con el tiempo de protección

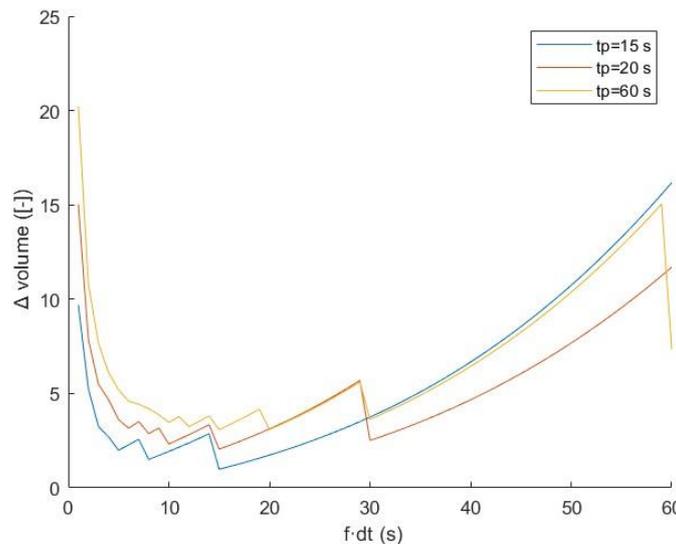


Figura 5.16 Incremento de volumen debido al factor y dt

Como se puede ver en estas dos figuras, la función se minimiza donde $f \cdot dt$ es submúltiplo de t_p , además, como es obvio, el carácter escalonado desaparece cuando $f \cdot dt$ es mayor que t_p , pues no tiene sentido insertar puntos virtuales separados temporalmente más allá del valor del tiempo de protección (en este caso se insertaría solo el punto original).

6 IMPLEMENTACIÓN

En este proyecto, se ha realizado la implementación del octree utilizando programación orientada a objetos (POO). La POO es un paradigma de programación que se basa en la idea de organizar el código en objetos, los cuales son instancias de clases. Una clase puede considerarse como una plantilla o molde que define las propiedades y comportamientos de un objeto.

En el contexto de la programación orientada a objetos, los atributos son las características o datos que describen el estado de un objeto. Estos atributos pueden representar diferentes aspectos de un objeto, como su tamaño, color, posición, etc. Por otro lado, los métodos son funciones o procedimientos asociados a una clase, que definen el comportamiento o las acciones que pueden ser realizadas por los objetos de esa clase.

La principal ventaja de utilizar la POO en la implementación del octree es que proporciona una metodología organizada y modular para estructurar el código. Al utilizar clases, se puede encapsular la lógica y los datos relacionados con el octree en un solo lugar, lo que facilita su comprensión y mantenimiento. Además, la POO permite la fácil extensibilidad y reutilización del código, ya que se pueden crear nuevas clases que hereden las propiedades y métodos de una clase base, adaptándola a nuevos requisitos o escenarios.

A continuación, se detallarán las clases que se han utilizado en la implementación y su relación, brindando una visión completa de su funcionamiento y capacidades.

6.1 Diagrama de Clases UML

UML, o Lenguaje Unificado de Modelado, es un estándar para visualizar la arquitectura de sistemas de software. Dentro de UML, un diagrama de clases muestra las clases, sus atributos y relaciones dentro de un sistema.

En el siguiente apartado, se presenta el diagrama de clases del servicio desarrollado, esencial para entender su estructura y las interacciones entre las diferentes clases, que son: SCRS, Uplan, WayPoint y octTree. Los detalles específicos de cada clase se encuentran en los siguientes apartados, sin embargo, las relaciones entre ellas se explican a continuación.

La clase principal es la clase SCRS. Esta es la clase sobre la que actúa el FAS, siendo todas las demás clases almacenadas y gestionadas por esta. Ante una petición del FAS, este activará un método del SCRS y será el SCRS el que hará las llamadas a métodos propios y a las otras clases creando instancias cuando sea necesario.

Por ejemplo, cuando un usuario envíe un U-plan para ser aprobado, el USSP lo almacenará en la base de datos y le transmitirá la orden al FAS. El FAS en ese momento hará ciertas comprobaciones previas y, si esta todo correcto, invocará al SCRS. Ante esto, el SCRS decodificará la información del U-plan y creará un objeto de la clase Uplan, con la información correspondiente. Tras ello, extraerá la información de este objeto Uplan y preguntará para cada punto de la trayectoria a

diferentes instancias de la clase octTree si los puntos se encuentran libres de conflicto. En caso afirmativo, los insertará definitivamente.

Como se puede observar, es la clase SCRS la que gestiona la petición del FAS y dirige el flujo de trabajo.

La clase Uplan guarda información concreta de un U-plan como la trayectoria, prioridad, estado, etc. Se crea a partir de un archivo en formato Json que enviará el usuario al USSP. La forma en la que se genera el archivo no se encuentra en el alcance de este trabajo. A continuación, el SCRS será el que decodifique el Json y cree un objeto de esta clase.

La clase WayPoint no tiene mucho protagonismo y actualmente podría reducirse a una estructura ya que solo tiene un método y podría conseguirse el mismo resultado a través de un método en la clase Uplan. No obstante, ya que implementarlo como una clase no tiene coste extra ni en memoria ni en velocidad, se ha decidido implementarlo de esta forma para otorgar mayor flexibilidad al servicio, ya que futuros trabajos podrían requerir que fuese una clase propia.

Por último, la clase octTree es la clase que almacena los U-plans con la estructura de datos del mismo nombre. Sobre esta es sobre la que se realiza la búsqueda de conflictos, y será el SCRS el que se encargue de crear los octTrees para cada instante temporal, e insertar en él los puntos de acuerdo con la técnica de los puntos virtuales.

Sobre el mismo ejemplo anterior, cuando el SCRS pregunte al octTree por el conflicto de un U-plan lo que hará será, en primer lugar, recorrer la trayectoria en orden temporal, y para cada instante de tiempo (si existe una instanciación del octree correspondiente a ese instante), hará una consulta al octree mediante el método find. Esto provocará que el octree busque en los nodos puntos que estén dentro del nivel de búsqueda del U-plan en cuestión y más adelante compruebe si producen un LoS. Con esto el octTree devolverá todos estos puntos, y otra vez será la clase SCRS la que compruebe la prioridad de los diferentes planes de vuelo y actúe en consecuencia.

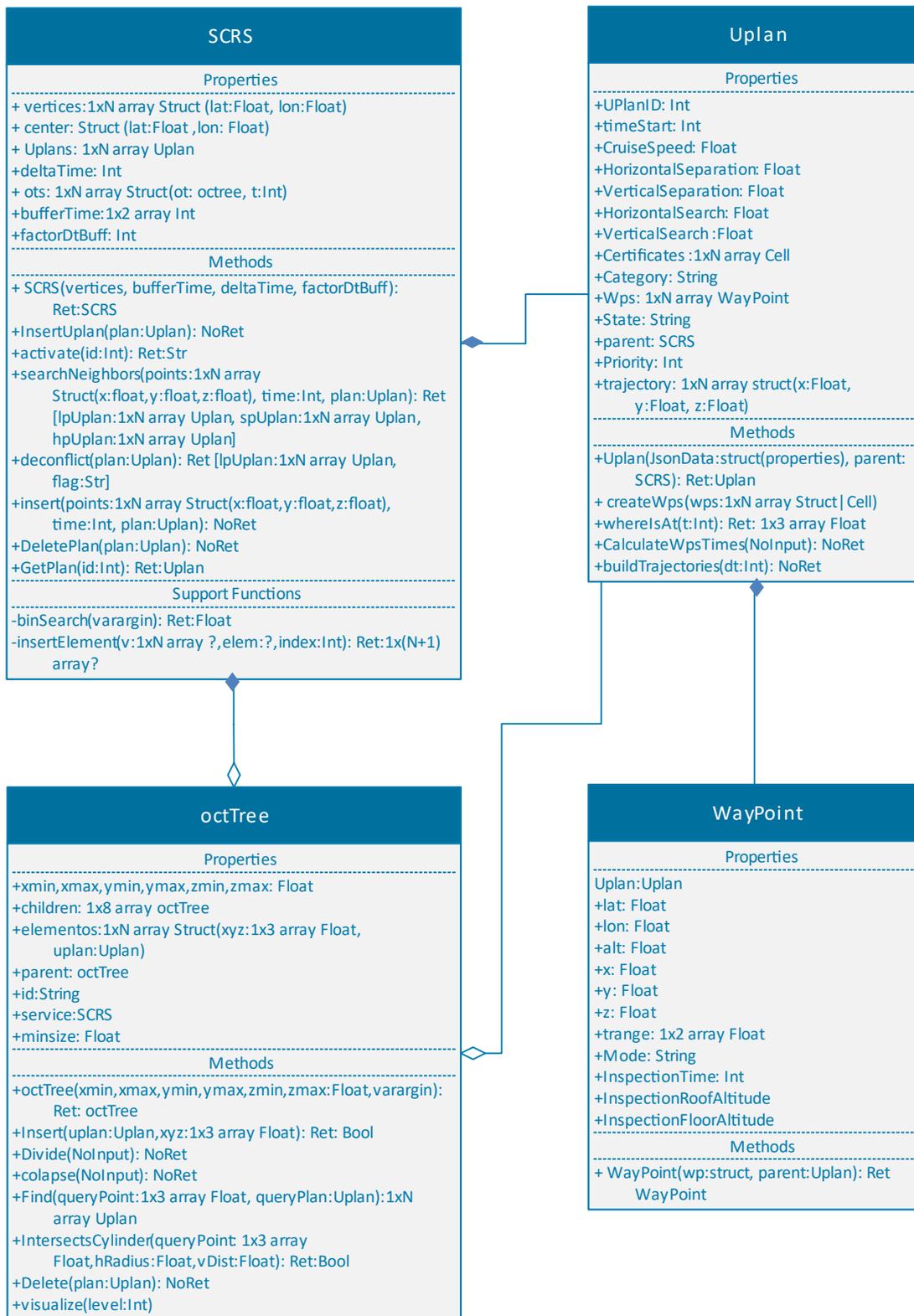


Figura 6.1 Diagrama de clases servicio propuesto

6.2 SCRS

6.2.1 Atributos

- Atributo **vértices**, almacena los vértices que conforman los límites de la zona en la que actuará el servicio.
- Atributo **center** almacena el centro del polígono que conforma la zona de actuación, este centro se calcula como el centroide del mínimo rectángulo envolvente con sus lados paralelos y perpendiculares a el ecuador. Este centro se utilizar para aproximarla geometría esférica en una geometría plana. Esto se puede hacer con muy poco error ya que los espacios aéreos en los que se espera que operen los USSPs son lo suficientemente pequeños.
- Atributo **Uplans**, es un vector que almacena los U-plans que están en un momento dado aceptados, ordenados en el mismo orden en el que se aceptan, lo cual es útil para aplicar la norma de FI-FS.
- Atributo **ots**, es un vector de estructuras con 2 campos:
 - “ot”, objeto de la clase octTree
 - “t”, valor que indica a que instante temporal pertenece el octTree.

Almacena todos los octrees ordenados temporalmente, se usa para establecer una conexión entre el SCRS y los octrees, p.ej. cuando se quiere añadir un punto se busca en el vector el tiempo al que se quiere añadir y se inserta en el octree correspondiente.

- Atributo **bufferTime**, almacena el valor que se le quiere aplicar a todos los vuelos de tiempo de protección como un vector de 2x1 (tiempo previo al despegue nominal y tiempo posterior), se utiliza el mismo tiempo de protección en la activación que en la aceptación de un U-plan, aunque está previsto que esto cambie en futuras versiones.
- Atributo **deltaTime**, almacena el valor de tiempo de paso discreto del servicio.
- Atributo **factorDtBuff**, almacena el valor del factor dt-Tiempo de protección explicado en el apartado 5.3.1.

6.2.2 Métodos

En el proceso de construcción de la clase SCRS, se han diseñado diversos métodos que permiten que realice las funciones necesarias. En las siguientes líneas, se desglosan algunos de los métodos principales implementados en esta clase:

6.2.2.1 InsertUplan

Este método se encarga de guardar el U-plan e insertar todos los puntos correspondientes a los octrees necesarios, sin embargo, no se asegura que el vuelo esté libre de conflictos, es por eso que siempre debe de utilizarse después del método deconflict. la siguiente figura muestra el diagrama de flujo del método:

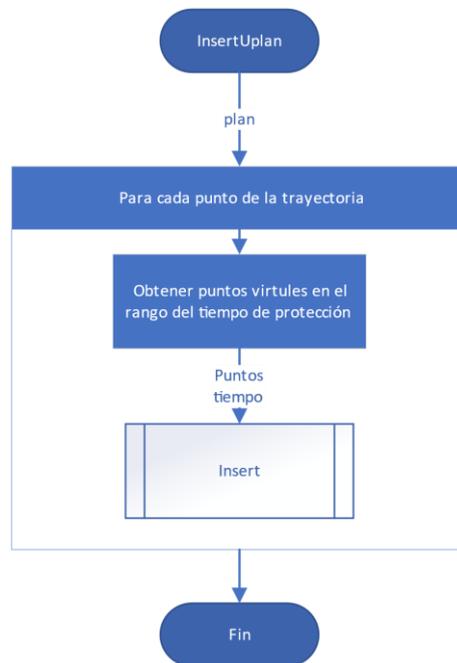


Figura 6.2 Diagrama de flujo del método InsertUplan

6.2.2.2 deconflict

Este método recorre todos los puntos de la trayectoria del U-plan, buscando posibles conflictos y los clasifica según la prioridad del U-plan que causa el conflicto.

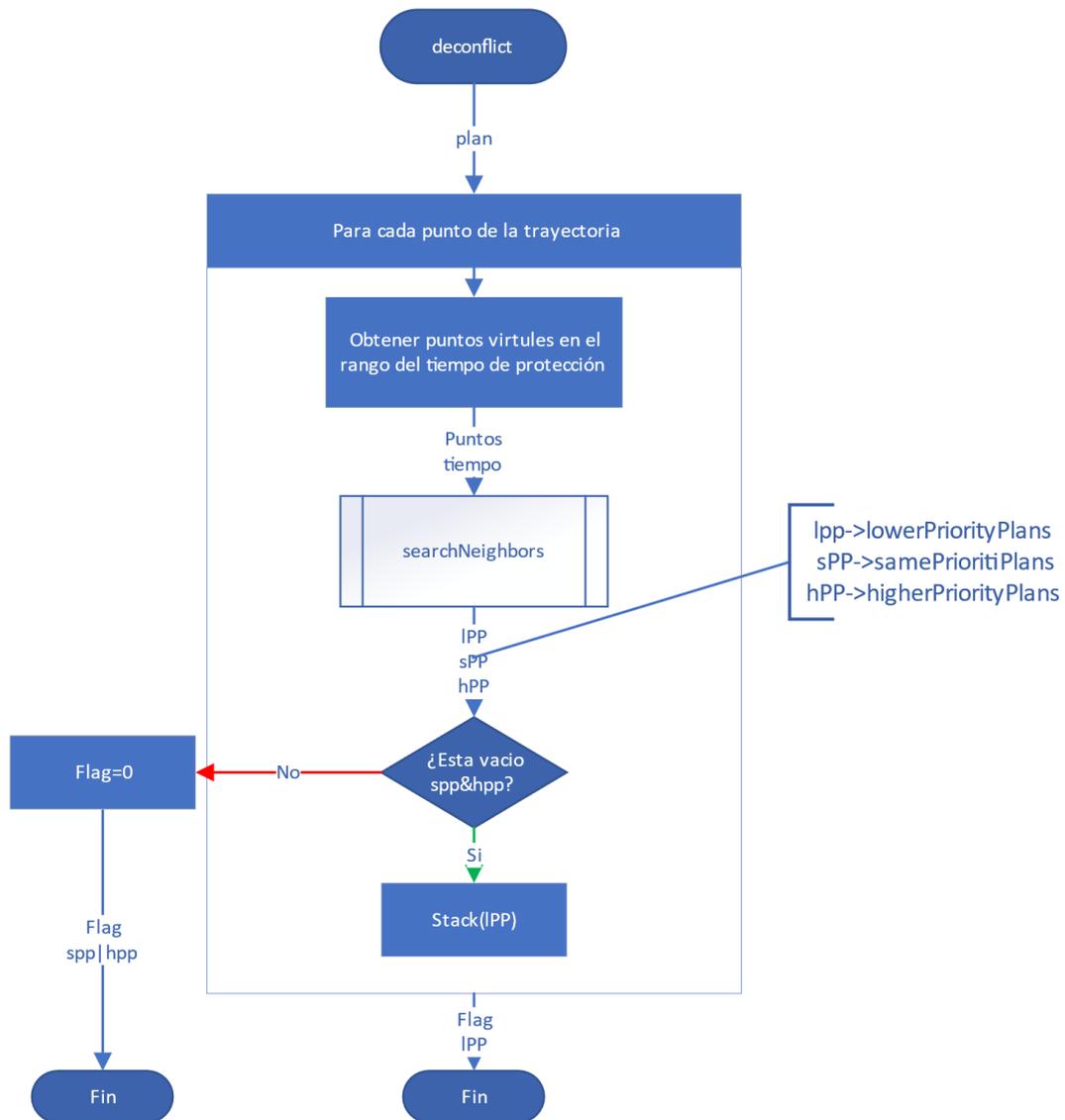


Figura 6.3 Diagrama de flujo del metodo deconflict

6.2.2.3 activate

Este método se encarga de comprobar si un U-plan puede ser activado, para ello realiza un proceso similar al del método deconflict (Figura 6.3), pero en lugar de cancelar directamente el vuelo si encuentra un U-plan de la misma prioridad primero compara cuál de los dos fue aceptado antes aplicando la regla de FI-FS, a continuación, se presenta el diagrama de flujo de método.

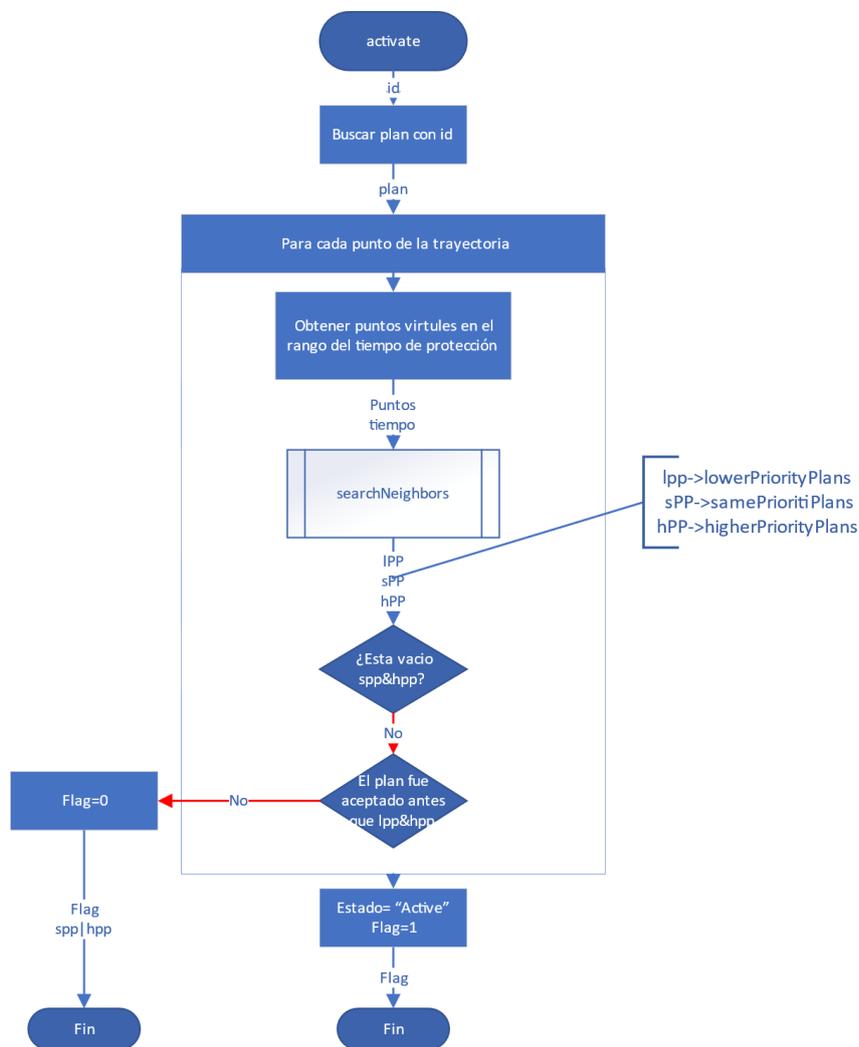


Figura 6.4 Diagrama de flujo del método activate

6.2.2.4 searchNeighbors

Este método se utiliza para buscar todos los planes que tengan conflicto con un plan en un instante temporal y unos puntos dados. Para lograr esto el método recorre el vector de puntos y pregunta al octree correspondiente, si existe (si no existe, implica que no hay ningún plan en este tiempo, y por tanto no puede haber conflictos), por todos los planes que causan conflicto, para posteriormente clasificarlos.

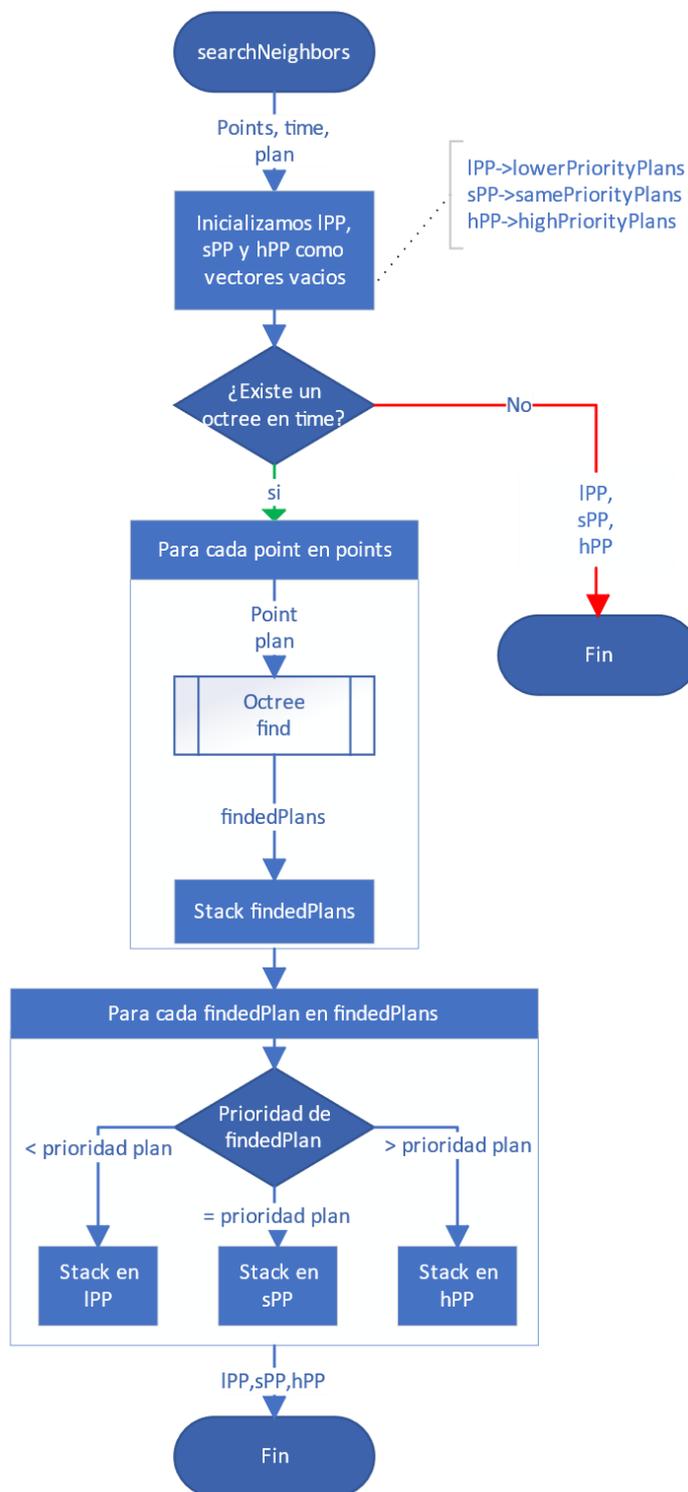


Figura 6.5 Diagrama de flujo searchNeighbors

6.2.2.5 Otros métodos

Para el correcto funcionamiento de la clase también ha sido necesario implementar los siguientes métodos:

- **Insert**, este método inserta una serie de puntos en un octree dado, si el octree no existe todavía lo crea y lo añade a el vector de octrees (ots).
- **getIndexPlan**, el método devuelve el índice de un plan contenido en el vector Uplans.
- **DeletePlan**, este método elimina un plan del servicio, incluyendo todos los puntos que había insertados en los octrees.

Además, también se han implementado las siguientes funciones privadas:

- **binSearch**, realiza una búsqueda binaria para en un vector ordenado.
- **insertElement**, inserta un elemento a un vector en un determinado índice, desplazando el resto del vector.

A continuación, en la Tabla 6-1 se realiza un resumen de todos los métodos y se indica como debería realizarse su llamada:

Tabla 6-1 Resumen métodos de la clase SCRS

Llamada	Input	Output
obj = SCRS (in1, in2, in3, in4)	In1: vértices de la zona de actuación del USSP In2: dt In3: Tiempo de protección In4: Factor dt-Tiempo de protección	obj SCRS
obj. InsertUplan (in)	In: Uplan	<i>No output</i>
[out1, out2] = obj. deconflict (in)	In: Uplan	out1 ⁸ 0: unable to insert 1: able to insert
[out1, out2, out3] = obj. searchNeighbors (In1, In2, In3)	In1 Puntos a los que buscar vecinos In2 Instante de tiempo de búsqueda In3 Uplan	out1 Uplans con menor prioridad out2 Uplans con igual prioridad out3 Uplans con mayor prioridad
obj. insert (in1, in2, in3)	In1 Puntos a los que buscar vecinos In2 Instante de tiempo de búsqueda In3 Uplan	<i>No output</i>
obj. DeletePlan (in)	In Uplan	<i>No output</i>

⁸ Esta salida es variable. Si el plan ha sido deconflictado con éxito (out2=1) esta salida almacena los planes con menor prioridad que deben ser eliminados para insertar el plan deconflictado; si el plan no ha podido ser deconflictado (out2=0) esta salida almacena el plan con el que ha tenido conflicto. Esta variable podría modificarse si se necesitase más información sobre el conflicto para resolverlo.

Llamada	Input	Output
out=getIndexPlan(in)	in Id de un Uplan	out Índice que ocupa el Uplan en el vector Uplans

6.3 Clase Uplan

Con instanciaciones de la clase Uplan es como esta herramienta archiva todos los *U-plans* y por tanto en esta clase se debe poder acceder a toda la información relevante de un *U-plan*, no necesita tener métodos que realizan acciones externas, todos los métodos implementados en Uplan sirven para decodificar un *U-plan* y prepararlo para ser procesado. En esta sección explicaremos estos métodos, pero antes definiremos la información necesaria que han de tener los *U-plans*:

6.3.1 Definición del U-plan

Como se ha dicho con anterioridad, la construcción de las trayectorias no está en el objetivo de este trabajo. Por ejemplo, esta herramienta asume que todos los waypoints se encuentran a la altitud de crucero, y que todos los tramos se realizan a velocidad de crucero. Pese a ser obvias las limitaciones que esto conlleva, la herramienta está diseñada de forma que si se realizan modificaciones en la forma en la que se construye la trayectoria, no afecte al funcionamiento general del sistema.

Tras esta advertencia inicial pasamos a explicar los campos mínimos que ha de tener un *U-plan* para que la herramienta funcione adecuadamente, estos campos no coinciden con los que indica el AMC&GM a la CIR 2021/664 [4] (en el apartado 3.1.2), aquí explicaremos los campos mínimos para que la herramienta funcione, no los campos mínimos que ha de tener según la regulación vigente.

Tabla 6-2 Descripción de los campos de un U-plan

ID	Identificador único de U-plan, no se establece quién es el responsable de establecerlo (<i>USSP</i> , <i>FAS</i> , ETC.), pero debe ser un identificador numérico que no se repita al menos mientras el <i>U-plan</i> este pendiente, aceptado o activo. Una vez sea eliminado (ya sea por haberse finalizado o por haberse cancelado) el mismo ID puede ser usado por otros <i>U-plans</i> .
Fecha de inicio	Fecha y hora de inicio del vuelo, en el siguiente formato: <i>HH:mm:ss dd/mm/aaaa</i> .
Velocidad de crucero	En metros por segundo.

Altitud de crucero	En metrod, la referencia de la altitud no es importante para el sistema mientras sea consistente y no sea altura.	
Prioridad	Prioridad de la operación para resolver conflictos.	
categoría	Categoría de la operación según BUBBLES.	
Listado de waypoints	Listado con los waypoints que conforman la trayectoria, este listado debe tener los siguientes campos:	
	Latitud	Decimal.
	Longitud	Decimal.
	Modo	Un waypoint puede tener varios modos, por el momento se han definido 2 tipos, paso o inspección
	Paso	Indica que por el waypoint se pasará sin detenerse, tan solo se usa para construir la trayectoria deseada.
	Inspección	indica que el <i>UAS</i> estará durante una cierta cantidad de tiempo alrededor del waypoint realizando cualquier actividad (escaneo, hover, descargando un paquete, recargando un depósito, etc.), este modo necesita ciertos argumentos adicionales.
	Tiempo de inspección	En segundos.
	Radio de inspección	En metros.
	Techo de inspección	En metros respecto a la misma altitud de referencia que la altitud de crucero.
Suelo de inspección	En metros respecto a la misma altitud de referencia que la altitud de crucero.	
Estado	Estado en el que se encuentra el <i>U-plan</i> (véase Figura 3.3)	

Estos son los campos necesarios para el buen funcionamiento de la herramienta, sin embargo, el *U-plan* puede tener tantos como quiera, ya que la clase los ignorará y tratará solamente aquellos que sean de su interés. Como requisito, es necesario que el *U-plan* se pase como un archivo Json. Este archivo tiene que tener como

nombre de campo los mismos que los atributos de la clase. En el Anexo A se encuentra un ejemplo con todos los campos necesarios donde puede observarse también la estructura del Uplan.

6.3.2 Atributos

- **UPlanID**⁹: ID U-plan.
- **TimeStart**⁹: Fecha y hora de inicio.
- **CruiseSpeed**⁹: velocidad de crucero.
- **CruiseAltitude**⁹: altitud de crucero.
- **HorizontalSeparation, VerticalSeparation**: radio y semi-altura del volumen de protección (vease 6.4 más adelante).
- **HorizontalSearch, VerticalSearch**: radio y semi-altura del volumen de búsqueda (vease 6.4 más adelante).
- **Category**⁹: categoría de la operación.
- **Wps**⁹: lista de objetos de la clase Waypoint que marcan la trayectoria
- **State**⁹: estado del plan.
- **parent**: objeto de la clase SCRS al que pertenece el U-plan.
- **Priority**: prioridad de la operación.
- **trajectory**: se trata de un listado con las coordenadas (4d) del UAS en cada instante de tiempo, consiste en un vector de estructuras con los siguientes campos:
 - “**x**” coordenada x
 - “**y**” coordenada y
 - “**z**” coordenada z
 - “**time**” instante de tiempo

6.3.3 Metodos de la clase

La clase Uplan cuenta con los siguientes métodos:

- **Constructor**: es el método que se utiliza para instanciar la clase, como parámetro necesita una estructura de datos con los campos del *U-plan*.
- **createWps**: La información sobre los waypoints almacenada en el *U-plan* es derivada desde el constructor a este método, que se encarga de instanciar objetos de la clase waypoint y añadirlos al atributo Wps
- **whereIsAt**: esta función devuelve las coordenadas de la trayectoria de un UAS en un determinado momento
- **calculateWpsTimes**: calcula la hora de llegada y de salida (será la misma si el waypoint es de tipo paso, pero no si lo es de tipo inspección) de cada uno de los waypoints
- **buildTrajectory**: Calcula las coordenadas del UAS para cada instante temporal y los añade al vector trajectory, se espera que en futuras versiones

⁹ Propiedad necesaria para la instanciación de la clase.

calcule también otros parámetros como GS, RoC, Radio de curvatura, etc.

La siguiente tabla ofrece información sobre los argumentos de entrada y de salida de los métodos:

Llamada	Input	Output
obj= Uplan (<i>in1</i> , <i>in2</i>)	iln1 <hr/>	<i>coordenada x min</i>
	iln2	<i>coordenada x max</i>
obj. createWps (<i>in</i>)	in Vector de estructuras con la información de l <i>U-plan</i> necesaria para la creación de los Waypoints	<i>No output</i>
obj. calculate_wps_times ()	<i>No input</i>	<i>No output</i>
obj. whereIsAt (<i>in</i>)	in Tiempo en el que se desea saber la posición del UAS (POSIX)	<i>No output</i>
obj. buildTrajectory (<i>in</i>)	in dt	<i>No output</i>

6.4 Clase WayPoint

La principal función de la clase waypoint es almacenar la información que contiene el U-plan de los waypoints, o puntos de paso. Estos se consultan una sola vez por la clase Uplan durante la creación de la trayectoria. A pesar de su funcionalidad simple, se ha decidido diseñarla de esta manera para añadir flexibilidad al código. De este modo, si las futuras versiones requieren que los waypoints sean más que una estructura básica, el código puede adaptarse fácilmente.

A continuación, explicaremos el funcionamiento del método constructor y describiendo los atributos que la componen.

6.4.1 Atributos de la clase

- **Uplan** almacena el objeto de clase Uplan al que pertenece el waypoint
- **lat** latitud decimal
- **lon** longitud decimal
- **alt** altitud
- **x** coordenada x
- **y** coordenada y
- **z** coordenada z
- **trange** tiempo (POSIX) de llegada y salida al waypoint
- **Mode** modo del waypoint (paso o inspección)
- **InspectionTime** tiempo de duración de la inspección

- **InspectionRoofAltitude** altitud máxima de inspección
- **InspectionFloorAltitude** altitud mínima de inspección

6.4.2 Método

Como se ha mencionado, el único método que existe es el constructor, que se llama de la siguiente forma:

Obj=Waypoint (in1, in2)

Donde in1 es una estructura con la latitud, longitud, altitud modo, y en caso necesario tiempo de inspección, techo de inspección y suelo de inspección. I in2 es una referencia al objeto Uplan que al que pertenece el waypoint.

Los atributos x,y,z, los calcula el método constructor a partir del centro de SCRS, mientras que el atributo trange es inicializado por el método “calculateWpsTimes” de la clase Uplan.

6.5 Clase Octree

La clase octree es una estructura de datos jerárquica y se basa en el principio de subdivisión recursiva del espacio en regiones más pequeñas conocidas como octantes. Cada octante representa una porción del espacio tridimensional y se divide en otros ocho octantes más pequeños si es necesario. Como se puede ver en la Figura 6.6.

Los nodos se pueden clasificar según su nivel en la jerarquía:

- Nodo **raíz**: El octree comienza con un único nodo raíz que abarca todo el espacio tridimensional de interés. Este nodo raíz tiene forma de cubo y se considera el nivel más alto de la jerarquía del octree.
- Nodos **internos**: Son aquellos que tienen subdivisiones adicionales y contienen información de referencia a sus octantes hijos.
- Nodos **hoja**: son aquellos que no están subdivididos y almacenan la información.

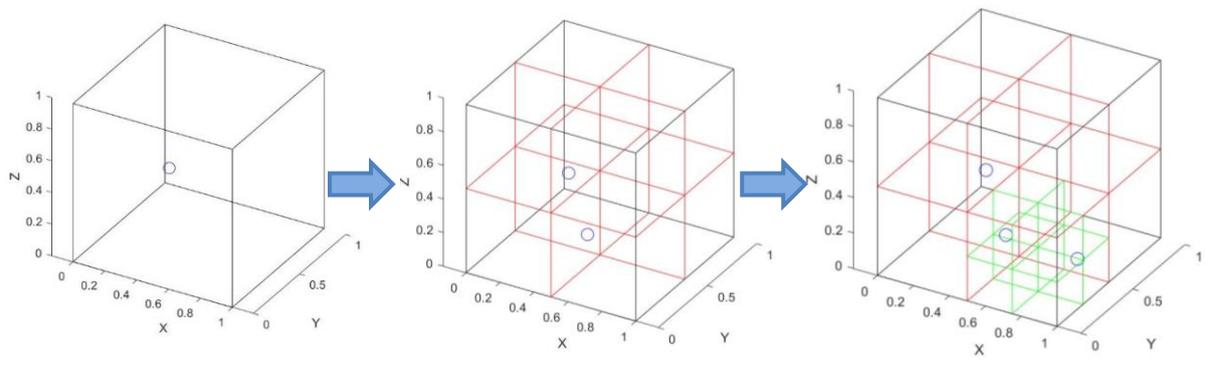


Figura 6.6 División de los nodos tras realizar una inserción

Antes de empezar a explicar la implementación de la clase, es necesario realizar una aclaración previa de dos conceptos muy importantes:

- El **volumen de protección** es un cilindro centrado en el UAS con radio igual a la separación mínima horizontal y semi-altura la separación mínima vertical definidas por el servicio de gestión de la separación.
- La búsqueda de conflictos en el diseño de esta herramienta se ha decidido que sea egocéntrica, lo que significa que cuando se busquen conflictos se hará respecto a un UAS en concreto. Esto implica, que el deconflictuado de este UAS debe ser capaz de encontrar todos UAS con los que tenga conflicto, incluso sin saber a qué clase pertenecen. Por este motivo definimos el **volumen de búsqueda**, que se trata de un volumen con las mismas características que el de protección, pero el radio es la suma de la separación mínima horizontal de la categoría a la que pertenezca el UAS, más la separación mínima horizontal de la categoría más elevada que pueda volar en el U-space dado. Este mismo principio se aplica a la semi-altura del cilindro.

6.5.1 Atributos

- Atributos de coordenadas (***xmin, xmax, ymin, ymax, zmin, zmax***): Estos atributos indican las coordenadas límite del nodo raíz en sus 3 ejes.
- Atributo ***children***: vector que guarda referencia a los hijos del nodo en caso de ser un nodo interno o raíz. Este vector se encontrará vacío, si el nodo es hoja, o con 8 elementos en caso contrario.
- Atributo ***Parent***: guarda referencia al nodo de jerarquía superior del cual es hijo el nodo actual.
- Atributo ***Service***: objeto de la clase SCRS que se utiliza para realizar consultas de atributos de los planes de vuelo, de modo que con solamente almacenar el U-plan se pueden obtener todas las propiedades de los vuelos.
- Atributo ***Minsize***: se utiliza como una condición de fin de recursividad. Si un nodo es más pequeño que este valor se convierte en un nodo hoja. Y no permite posteriores subdivisiones
- Atributo ***elementos***: vector de estructuras que guarda la información de

cada punto que se inserta, contiene los siguientes campos:

- “**x**”, coordenada x
- “**y**”, coordenada y
- “**z**”, coordenada z
- “**uplan**” objeto de clase Uplan.

6.5.2 Métodos

En la implementación del octree, se han desarrollado varios métodos que permiten realizar diversas operaciones en la estructura de datos. A continuación, se detallan algunos de los métodos claves implementados:

6.5.2.1 Constructor

Este método se encarga de inicializar y construir el nodo, como inputs debe recibir los 6 atributos de coordenadas y además puede recibir parejas de atributo-valor para permitir flexibilidad en su llamada.

El método está diseñado para ser llamado solo con los 6 atributos de coordenadas y el atributo *Service* para crear el nodo raíz, y de forma automática al dividirlo en el método *Divide*.

6.5.2.2 Insert

Este método se encarga de insertar elementos a los nodos. Como se ve en la Figura 6.7, se trata de una función recursiva. Cuando es llamada primero comprueba si el elemento a insertar se encuentra dentro de los límites del nodo. A continuación, en caso de que tenga hijos inserta el elemento a los nodos y en caso contrario o bien almacena el elemento, si el nodo debe almacenarlo, o bien se divide y reinicia el proceso.

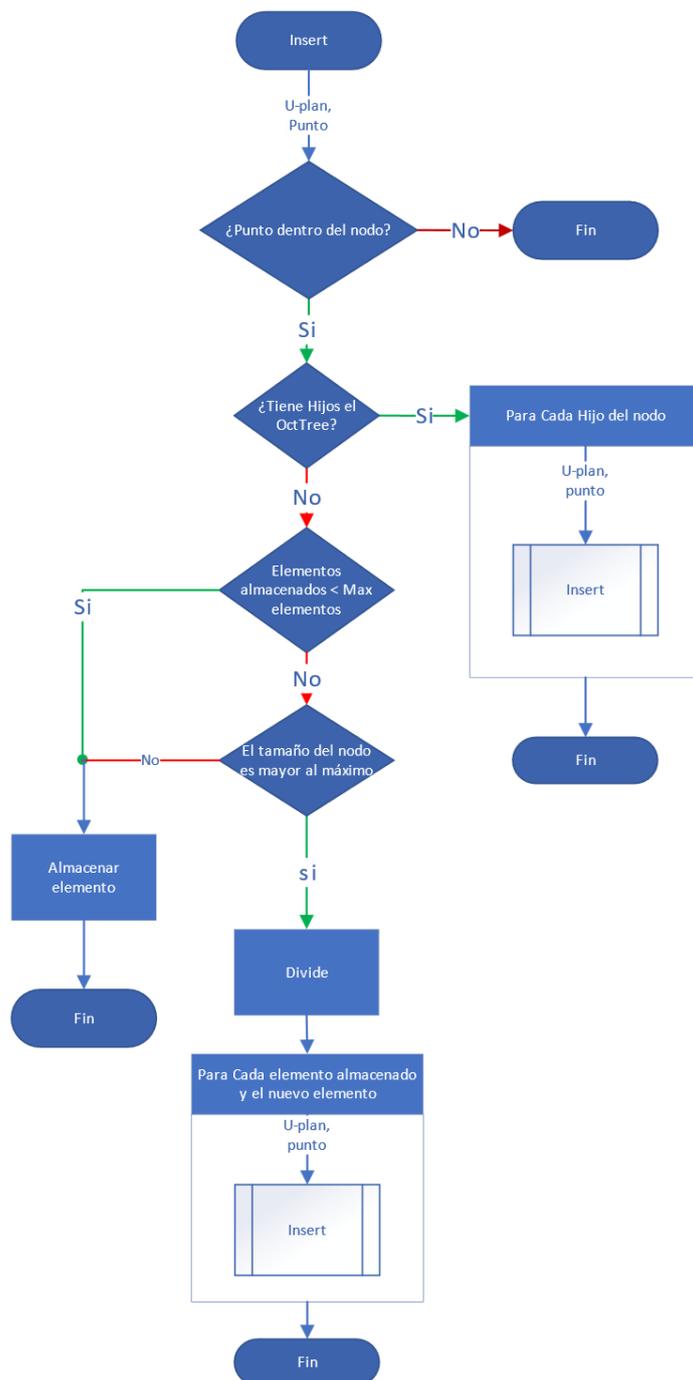


Figura 6.7 Diagrama de flujo del método insert

6.5.2.3 Divide

El método *Divide* divide los límites del nodo actual en 8 regiones y crea instancias

Implementación

de la clase `octTree` con estos límites, añadiéndolos al array `children`.

6.5.2.4 Find

El método *Find* se utiliza para encontrar elementos en una zona concreta. En la Figura 6.8 se puede observar la implementación del método. En primer lugar, se obtienen los parámetros de búsqueda, que consisten en un radio para el plano horizontal y una distancia para el plano vertical. Nótese que estos parámetros no son de protección, sino que es el nivel de protección de la operación sobre la que se está haciendo la búsqueda más el nivel de protección de una hipotética operación con la categoría que necesite mayor nivel de protección.

Con esto se construye un cilindro alrededor del punto de búsqueda y la función *intersectsCylinder* retorna verdadero si este cilindro interseca con el nodo y falso si no, como se ve en amarillo en la Figura 6.9 b)¹⁰.

A continuación, para cada uno de los elementos de estas hojas, se consulta a los planes en posible conflicto los parámetros de protección de las operaciones correspondientes a dicho elemento y los propios. Si la distancia que separa los puntos, tanto en el plano horizontal como en el vertical, es menor a la suma de estos parámetros, se produce un *LoS* y se añade el *U-plan* del elemento a la lista *finded*.

¹⁰ Nótese que en la figura solo se han representado en amarillo los nodos hoja, sin embargo, los padres de dichos nodos también intersecan con el cilindro.

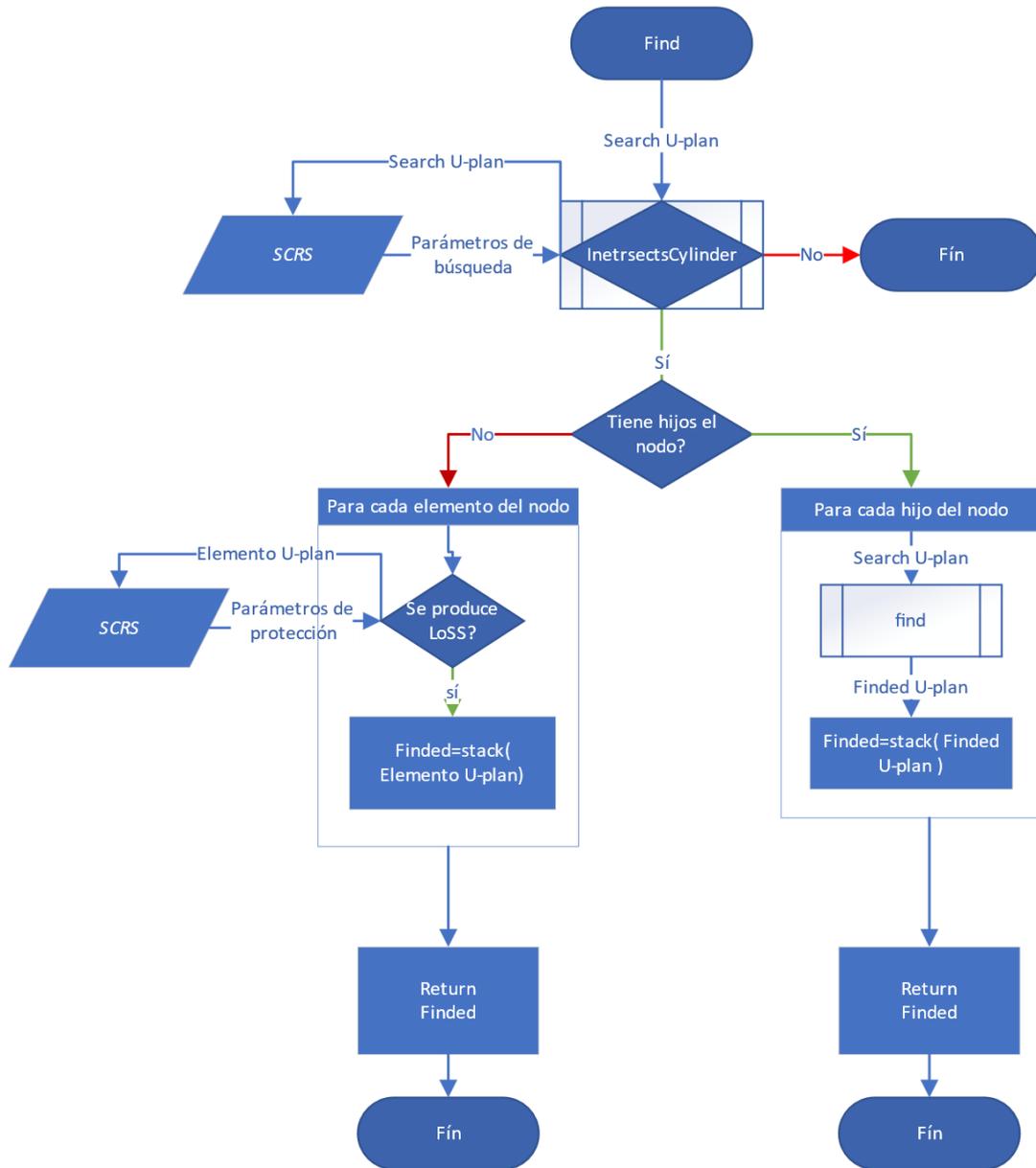
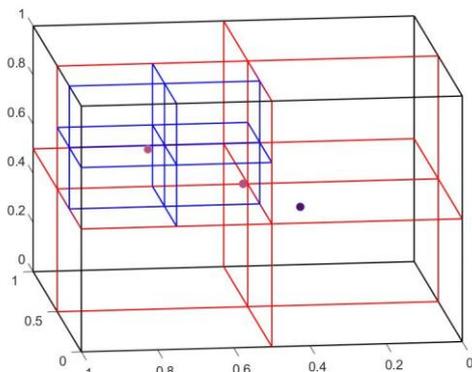
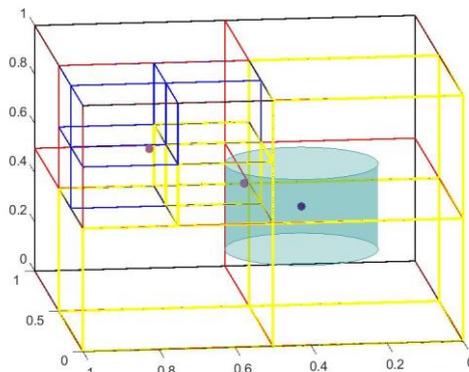


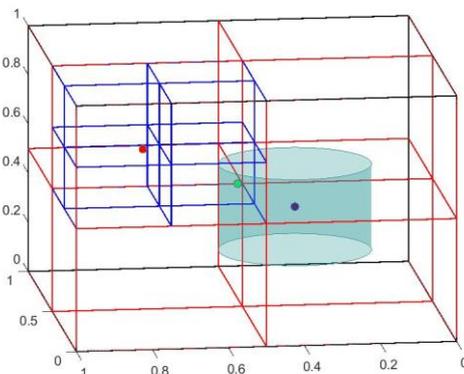
Figura 6.8 Diagrama de flujo del método Find



a) Se establece un punto para hacer la búsqueda (punto azul).



b) Se busca todos los nodos hoja que intersecan con el cilindro de búsqueda (octantes amarillos).



c) Se consulta a dichos nodos que elementos están produciendo un LoS.(punto verde).

Figura 6.9 Proceso del método Find

6.5.2.5 Otros métodos

- **Delete:** Este método se encarga de borrar un elemento, para ello busca primero el nodo que lo contiene, se utiliza para por ejemplo, borrar un *u-plan* que ya se había insertado
- **Colapse:** se llama siempre después de Delete, y comprueba si los 8 hijos de un nodo están vacíos de elementos, en dicho caso elimina los hijos y por tanto el nodo rama pasa a ser nodo hoja. Esto se hace en contraposición del método *divide*, y se utiliza cuando se elimina un *U-plan* (p. ej. Si se ha desechado por causar conflicto con un plan d mayor prioridad) para deshacer los efectos que tenía este U-plan en el octree
- **IntersectsCylinder:** comprueba si un cilindro y un cubo intersectan. Se utiliza en la función *find* para encontrar que nodos intersectan con el cilindro de búsqueda.
- **Visualize:** ofrece una visualización del nodo y sus hijos, así como de los elementos que contienen, se utiliza solamente en caso de necesitar depurar el programa.

6.5.2.6 Resumen de los métodos implementados en la clase octTree

Tabla 6-3 Resumen de los métodos de la clase octTree

Llamada	Input	Output	
obj= octTree (in1, in2, in3, in4, in5, in6, "attribute", value ¹¹) ¹²	in1	coordenada x min	octTree
	in2	coordenada x max	
	in3	coordenada y min	
	in4	coordenada y max	
	in5:	coordenada z min	
	in6:	coordenada z max	
out=obj. insert (in1, in2)	in1: id del U-plan in2 coordenadas del punto a insertar ([x,y,z])	out: 0: Not inserted 1: Inserted	
obj. Divide ()	No input	No output	
out=Ot. Find (in1, in2)	in1 coordenadas del punto a alrededor del cual se hace la consulta ([x,y,z]) in2 id del plan alrededor del cual se hace la consulta	out: array de los de los planes que producen un LoS	
Ot. Delete (in)	in id del U-plan	No output	
Ot. colapse ()	No input	No output	
out=Ot. IntersectsCylinder (in1, in2, in3)	in1 coordenadas del punto a alrededor del cual se hace la consulta ([x,y,z]) in2 Radio del cilindro in3 Semi-altura del cilindro	out 0: No inerseca 1: interseca	
Ot. visualize ()	No input	No output	

¹¹ "attribute" y "value" son parámetros opcionales. Se puede modificar cualquier atributo de la clase y añadir tantos como se desee. Por ejemplo: ("minsize",100,"parent").

¹² Cuando se cree una raíz, solamente será necesario definir el "valor" para el atributo "service". El valor representará el objeto de la clase SCRS al que pertenece el octree.

6.6 Análisis del código

Una parte esencial del desarrollo de software es la optimización del código, asegurando que la herramienta diseñada y desarrollada funcione de manera eficiente. Para esto, en MATLAB contamos con la herramienta 'Profiler', un poderoso instrumento de análisis que rastrea la ejecución del programa, recopilando datos detallados sobre el tiempo de ejecución, el número de llamadas a funciones y el uso de memoria.

El 'Profiler' nos proporciona una visión completa de dónde se consume la mayoría de los recursos, lo cual es útil para identificar partes del código que pueden ser optimizadas, mejorando así el rendimiento general.

En esta sección, se analizará el código desarrollado con 'Profiler'. Este análisis permitirá detectar áreas de mejora para optimizar su rendimiento y asegurar que la herramienta desarrollada no solo cumple con su propósito funcional, sino que también lo hace de manera eficiente.

Para ello, el código se ejecutó con mil planes de vuelo, utilizando un delta time de 2 segundos, un factor de 5, y un tiempo de protección de 30 segundos. Todos los planes se deconflictuaron en una primera instancia. Luego, los que no tenían conflictos se insertaron en el octree. Finalmente, se simuló la activación de los planes de vuelo que habían sido aceptados.

En esta sección, nos centraremos más en el rendimiento en términos de porcentaje en lugar del tiempo de ejecución, ya que este último depende de la máquina en la que se ejecute. Este segundo aspecto se analizará en la sección correspondiente (7. Resultados).

La Figura 6.10 muestra el resultado de este análisis:

Profile Summary (Total time: 119868.604 s)

Generated 10-jul-2023 09:09:53 using performance time.

Function Name	Calls	Total Time (s) ↓	Self Time* (s)	Total Time Plot (dark band = self time)
main	1	119868,572	0,001	
main>fun	1	119868,571	1,215	
SCRS>SCRS.InsertUplan	448	108004,200	11,397	
SCRS>SCRS.insert	288092	107981,301	135,254	
octTree>octTree.Insert	104219167	107262,981	105929,535	
SCRS>SCRS.searchNeighbors	614790	8452,890	50,661	
octTree>octTree.Find	162877608	7197,344	1774,087	
SCRS>SCRS.deconflict	1000	5692,367	7,496	
union	165527378	4566,193	1131,658	
union>unionR2012a	165527378	3434,535	908,931	
SCRS>SCRS.DeletePlan	58	3362,845	1,730	
octTree>octTree.Delete	10833421	3361,115	1143,776	
SCRS>SCRS.activate	390	2797,108	3,479	
unique	166336787	2544,327	2006,720	
octTree>octTree.collapse	1316253	2217,338	2217,338	
SCRS>binSearch	902882	1552,023	1552,023	
octTree>octTree.Divide	891721	1333,447	406,356	
octTree>octTree.IntersectsCylinder	162877608	1102,273	1102,273	
octTree>octTree.octTree	7139564	927,563	303,402	
isprop	21418100	624,548	109,373	
unique>uniqueR2012a	166336787	537,606	537,606	
isprop>hasProp	21418100	515,175	515,175	
Uplan>Uplan.Uplan	1000	9,917	1,124	
SCRS>insertElement	5796	8,584	8,584	
Uplan>Uplan.buildTrajectory	1000	7,140	7,140	

Figura 6.10 Resultado del análisis de "profiler" de MATLAB

Lo primero que podemos observar es que se han creado 1000 planes (1000 llamadas a la función Uplan.Uplan,) los cuales todos fueron deconflituados (1000 llamadas a SCRS.Deconflict), pero solo 448 han sido insertados (448 llamadas a SCRS.InsertUplan), siendo de estos finalmente 58 eliminados por causar conflicto con planes de mayor prioridad (llamadas a SCRS.Delete). Con esto, se activaron 390 con éxito (llamadas a SCRS.activate).

El siguiente paso en el análisis es determinar qué función consumió la mayor parte del tiempo. Claramente, la función octree.insert fue la más tiempo consumía, consumiendo un 88,37% del tiempo total (un "self time" de 105929,535 respecto a un "total time" de 119868,604). Este resultado era de esperar, puesto que el algoritmo de inserción de puntos en un octree es el más lento de los utilizados (construct, insert y query).

Los otros dos métodos importantes a analizar son deconflict y activate. Dado un profundo trabajo de optimización, estos han sido optimizados de manera eficiente,

consumiendo solo un 4,74% y un 2,33% del tiempo total, respectivamente (tomando "total time" en lugar de "self time").

Lines that take the most time

Line Number	Code	Calls	Total Time (s)	% Time	Time Plot
92	end	16428269	69888,658	65,2%	
66	return	10649197	24686,392	23,0%	
85	obj.elementos=[];	891721	10570,587	9,9%	
83	obj.Divide()	891721	1335,934	1,2%	
64	inserted=obj.children(i).Insert(uplan,xyz);	93851754	341,050	0,3%	
All other lines			440,360	0,4%	
Totals			107262,981	100%	

Children (called functions)

Function Name	Function Type	Calls	Total Time (s)	% Time	Time Plot
octTree>octTree.Divide	Class method	891721	1333,447	1,2%	
octTree>octTree.Insert	Class method	100093801	0,000	0,0%	
Self time (built-ins, overhead, etc.)			105929,535	98,8%	
Totals			107262,981	100%	

Code Analyzer results

No Code Analyzer messages.

Coverage results

[Show coverage for parent folder](#)

Total lines in function	40
Non-code lines (comments, blank lines)	11
Code lines (lines that can run)	29
Code lines that did run	29
Code lines that did not run	0
Coverage (did run/can run)	100,00 %

Figura 6.11 Resultado del análisis de "profiler" de el método octree.Insert

Para concluir con el análisis del código, en la Figura 6.11 mostramos los resultados del análisis del método octree.Insert. Aquí observamos que el 98,1% del tiempo se consume en un "end", un "return" y una limpieza de variable. Esta situación hace que la optimización de la función sea prácticamente imposible. La única opción viable es tratar de reducir el número de llamadas. Esto es precisamente lo que logra el factor dt-tiempo de protección. Sin embargo, aumentar este factor demasiado implicaría reducir la capacidad del espacio aéreo.

Con estos resultados, podemos concluir que el código ha sido optimizado satisfactoriamente. Las funciones que se necesita que sean más rápidas, deconflict y activate, parecen estar eficientemente optimizadas, tanto en términos de código como de llamadas. En la sección de trabajos futuros, sugeriremos una técnica que podría reducir significativamente el tiempo total del proceso.

7 RESULTADOS

Para analizar los resultados obtenidos, se han llevado a cabo simulaciones en varios escenarios independientes. Estos escenarios se han diseñado con el objetivo de probar el comportamiento de la herramienta en diferentes contextos. En cada uno de ellos, se ha alimentado la herramienta con los U-plans correspondientes y se ha evaluado su respuesta en términos de tiempo de procesamiento y efectividad.

7.1 Definición de los escenarios y el equipo utilizado

Para la creación de los planes de vuelo, se ha elaborado un script simple que genera Jsons con el formato de U-plan presentado anteriormente con valores aleatorios dentro de un rango, tal y como se muestra en la Tabla 7-1 rango de valores para la creación de los U-plans, este script se presenta en

Tabla 7-1 rango de valores para la creación de los U-plans

	min	max
Para cada U-plan		
Numero de waypoints	2	7
Velocidad de crucero	10	30
Altitud de crucero	0	500
prioridad	1	5
categoría	{A 1, A2, SAIL I-II, SAIL III-IV, SAIL V-VI, Certificada}	
Para cada waypoint		
latitud	39.4167	39.4988
longitud	-0.4257	-0.3180
Modo	{paso, inspección} ¹³	
Tiempo de inspección	300	900
Techo de inspección	350	450
Suelo de inspección	250	350

¹³ Paso un 80% de probabilidad, inspección el 20% restante

La zona de acción de todos estos UAS, comprende aproximadamente el área de la ciudad de Valencia, en la siguiente figura se ofrece una visualización de esta:



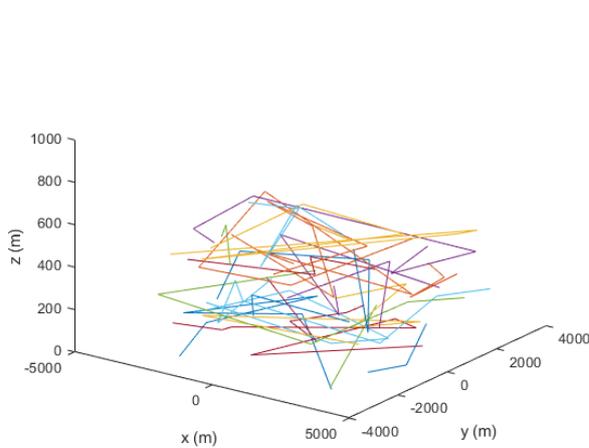
Figura 7.1 Zona de simulación

Con esto se han creado 6 escenarios descritos en la Tabla 7-2, por la forma en la que se han construido los escenarios, implica que los U-plans son “free- roam”, lo que significa que los vuelos no siguen ninguna estructura de espacio aéreo, vuelan entre los waypoints deseados de punto a punto

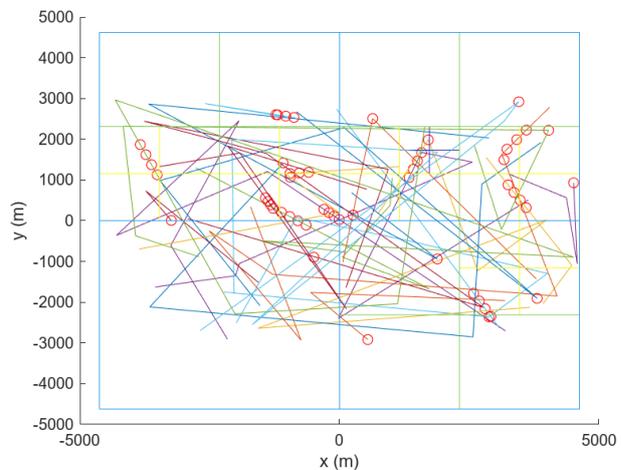
Tabla 7-2 definición de los escenarios de simulación

Escenario	Número de U-plans	Rango de hora de inicio
1	500	1 hora
2	750	1 hora
3	1000	1 hora
4	1000	2 horas
5	1000	4 horas
6	2000	4 horas

La siguiente figura muestra un ejemplo del estado de un octree, para el escenario 2 en un momento dado:



a) vista trayectorias activas.



b) vista cenital trayectorias activas junto a octree.

Figura 7.2 Ejemplo estado del octree

Respecto a las categorías, hemos tenido la suerte de contar como cotutora del proyecto a Cecilia Claramunt Puchol, que nos ha otorgado una tabla con separaciones para la categoría. Hay que recalcar que cuando el servicio esté integrado en el entorno U-space, estas separaciones pueden variar, por lo que la siguiente tabla es tan solo un ejemplo concreto para contar con separaciones del orden de las que se utilizarán en entornos reales:

Tabla 7-3 Separación horizontal y vertical por categoría de operación

categoria	Separachón horizontal	Separación vertical
A1	179.22	36.22
A2	179.22	36.22
A3	337.75	36.22
SAIL I II	401.16	36.22
SAIL II IV	439.14	36.22
SAIL V VI	469.03	36.22
Certificada	767.92	36.22

La ejecución del código se ha realizado con dos equipos con las siguientes especificaciones:

Tabla 7-4 equipos utilizados para la simulación

Equipo 1	Procesador	Intel(R) Core (TM) i5-10400 CPU @ 2.90GHz 2.90 GHz
	RAM	16 GB
	Sistema operativo	Windows 10 Pro versión 22H2
Equipo 2	Procesador	Intel core i7-11700K @3.60 GHz
	RAM	64 GB
	Sistema operativo	Windows 10 Pro versión 22H2

7.2 Exposición y análisis de resultados

7.2.1 Ratio de conflictos detectados

En primer lugar, como se había explicado en el apartado 5.1, al discretizar en el tiempo es posible que nos dejemos ciertos conflictos por detectar, sobre todo con el cálculo de trayectorias que estamos aplicando, ya que las aeronaves pueden cambiar de dirección y sentido de forma instantánea.

Teniendo esto en mente el primer resultado que mostraremos será un estudio realizado en el que estudiamos el porcentaje de conflictos descubiertos. En la Figura 7.3 se puede ver el resultado de este estudio, hay que tener en cuenta que se ha asumido que en con dt de 1 segundo, se detectan todos los casos, pese a no ser cierto nos puedo dar una idea de la tendencia. Esto podría revolverse con un escenario ficticio conocido, pero no contamos con un escenario de ese tipo y está fuera del alcance de este proyecto realizar uno.

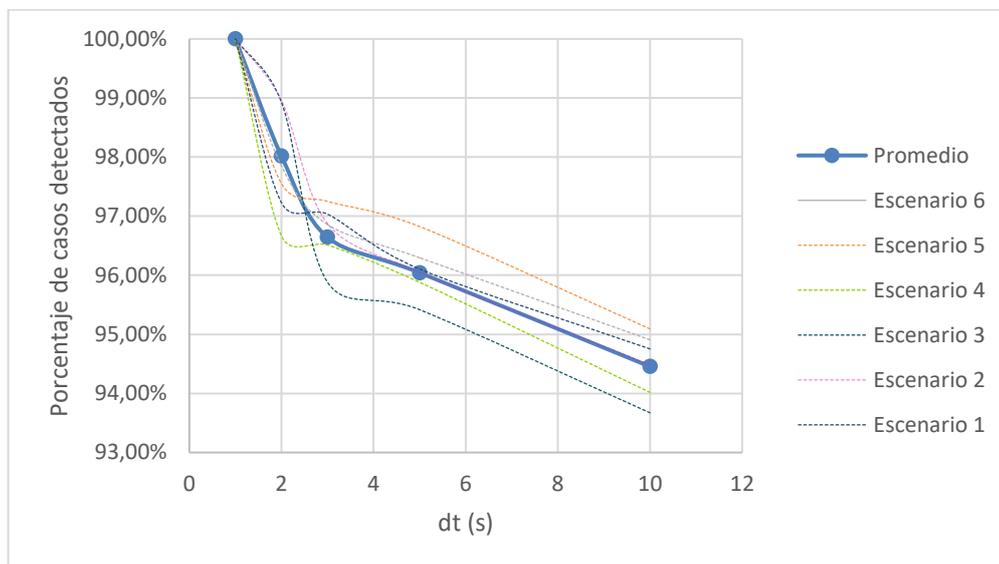


Figura 7.3 porcentaje de conflictos detectados en diferentes escenarios y dt

7.2.2 Ratio de activaciones en menos de cinco segundos

La única restricción que establece alguna legislación o estándar en cuanto a tiempo en el procesamiento por parte del FAS, se encuentra en el estándar F3548-21 [2] en él, establece que el tiempo de activación debe mantenerse por debajo de 5 segundos en el 95% de las veces, por lo que usaremos este parámetro para analizar la calidad de la herramienta.

Se simuló en el equipo 1 estos escenarios con cero segundos de tiempo de protección y se guardaron los tiempos que se tardaron en realizar las diversas acciones, deconflictuar los U-plans, insertarlos y activarlos. Obteniendo los resultados de la siguiente tabla para el equipo 1 y el equipo 2:

Tabla 7-5 Porcentaje de activaciones en menos de 5 segundos en diferentes escenarios y dt equipo 1

dt	0.5 s	1 s	2 s	5 s	10 s
Escenario 1	33,8%	71,2%	98,7%	100%	100%
Escenario 2	28,6%	73,7%	98,7%	100%	100%
Escenario 3	33,5%	73,9%	100%	100%	100%
Escenario 4	25,4%	65,1%	100%	100%	100%
Escenario 5	16,2%	41,0%	92,2%	100%	100%

dt	0.5 s	1 s	2 s	5 s	10 s
Escenario 6	11,7%	38,1%	95,1%	100%	100%

Tabla 7-6 Porcentaje de activaciones en menos de 5 segundos en diferentes escenarios y dt equipo 1

dt	0.5 s	1 s	2 s	5 s	10 s
Escenario 1	39%	80%	100%	100%	100%
Escenario 2	31%	70%	100%	100%	100%
Escenario 3	39%	81%	100%	100%	100%
Escenario 4	27%	74%	100%	100%	100%
Escenario 5	13%	47%	96%	100%	100%
Escenario 6	2%	41%	97%	100%	100%

Las siguientes simulaciones se realizaron con el equipo 2. Este equipo consiguió los resultados que se muestran en la Tabla 7-7, como estas simulaciones son muy costosas computacionalmente, se realizaron solamente para el escenario 4 y con un dt de 2 segundos. Este dt fue escogido para mantener alrededor de un 98% de detectabilidad, como se demostró en la Figura 7.3.

Tabla 7-7 porcentaje de activaciones en menos de 5 segundos con dt=2 s en el escenario 4

f	1	5	15	30	60
tp					
0 s	100,0%	100,0%	100,0%	100,0%	
30 s	42,4%	97,9%	100,0%	99,6%	
60 s		95,0%	97,8%	99,6%	
120 s		66,1%	91,7%	94,9%	96,7%
150 s			49,2%	53,7%	65,8%

Como se puede observar, aumentar el tiempo de protección es muy perjudicial con el tiempo de activación y hay que aumentar mucho el factor dt-tiempo de protección para conseguir lograr el objetivo de 95%.

8 CONCLUSIONES

Este Trabajo Final de Grado ha permitido el diseño y desarrollo de una herramienta útil para la resolución de conflictos estratégicos entre aeronaves no tripuladas. A través de una revisión exhaustiva de la regulación y estándares relevantes, hemos adquirido un entendimiento sólido de los requisitos y restricciones técnicas y legales que afectan a la operación de las aeronaves no tripuladas.

Los proyectos SESAR analizados, han proporcionado importantes antecedentes y consideraciones prácticas en el campo de gestión del tráfico de drones. Este análisis ha demostrado la complejidad y los desafíos de la intersección de las trayectorias 4D.

La implementación con POO de las clases SCRS, Uplan, WayPoint y Octree, han permitido la construcción de una herramienta de software que es capaz de manejar de manera eficaz la detección de conflictos estratégicos. La adopción de un enfoque basado en la discretización temporal y la estructura de datos Octree ha resultado ser una solución efectiva a los problemas planteados.

Podemos concluir que la herramienta que hemos diseñado puede soportar tráfico del orden de 500 peticiones por hora, mientras que el espacio aéreo sobre el que se ha hecho la simulación parece aceptar alrededor de 250 vuelos por hora, con tráfico en "Free-roam", una capacidad del espacio aéreo más que suficiente para la implementación del U-space a corto-medio plazo.

Pese a haber conseguido una solución efectiva, el problema requiere de grandes tiempos de computación, y se debe mejorar aún más para que las herramientas sean útiles. Para ello, en el apartado de Trabajos futuros proponemos como se podría optimizar más.

En conclusión, este TFG ha logrado su objetivo de diseñar y desarrollar una herramienta eficaz para la resolución de conflictos entre aeronaves no tripuladas. Sin embargo, se reconoce que el campo de la gestión del tráfico de drones es un área de investigación en rápida evolución y se necesitarán investigaciones futuras y mejoras continuas para mantener y mejorar la eficacia de la herramienta en el futuro.

9 TRABAJO FUTURO/SIGUIENTES PASOS

A pesar de los avances significativos en el proyecto, el servicio aún se encuentra en fase de desarrollo y mejora. Existen numerosas necesidades que satisfacer y problemas por resolver. En este apartado, expondremos algunos de los siguientes pasos a realizar:

En primer lugar, es de vital importancia añadir realismo a la construcción de la trayectoria. En la actualidad, se asume que los UAS no cambian de velocidad ni de altitud y no se considera la dinámica de vuelo de los UAS para realizar giros ni aceleraciones. Este punto es crucial ya que permitiría realizar pruebas y estudios con mayor precisión.

En segundo lugar, es necesario convertir las herramientas actuales en un servicio automatizado, permitiendo que los procesos se ejecuten de manera autónoma desde el momento en que un usuario realiza una operación.

Un tercer paso sería implementar algoritmos de resolución de conflictos para proporcionar alternativas a los usuarios cuando un U-plan sea cancelado.

Otra de las tareas futuras más importantes es paralelizar la detección de conflictos. Esta medida podría ser una gran solución para disminuir el dt y aumentar el tiempo de protección, manteniendo o incluso mejorando el tiempo de procesamiento, tanto para deconflictuarlos como para activarlos y, sobre todo, insertarlos. Esto es posible dado que, una vez calculada la trayectoria, los octrees de diferente instante temporal no están interrelacionados. Por lo tanto, distribuir estos octrees entre varias CPUs reduciría drásticamente el tiempo computacional.

Finalmente, para llevar este proyecto a un nivel TRL superior, sería necesario integrarlo con otros servicios U-space, adaptarlo a un entorno multi-USSP e incorporar la capacidad de gestionar restricciones y limitaciones del espacio aéreo. Este último aspecto se avanzó en gran medida durante las prácticas en SNA, pero no se ha incluido en este proyecto porque no se probó su correcto funcionamiento ni su impacto.

10 REFERENCIAS

- [1] Commission Implementing Regulation (EU) 2021/664 of 22 April 2021 on a regulatory framework for the U-space (Text with EEA relevance), vol. 139. 2021. Accedido: 26 de junio de 2023. [En línea]. Disponible en: http://data.europa.eu/eli/reg_impl/2021/664/oj/eng
- [2] «F3548 Standard Specification for UAS Traffic Management (UTM) UAS Service Supplier (USS) Interoperability». <https://www.astm.org/f3548-21.html> (accedido 26 de junio de 2023).
- [3] Commission Implementing Regulation (EU) 2019/947 of 24 May 2019 on the rules and procedures for the operation of unmanned aircraft (Text with EEA relevance.), vol. 152. 2019. Accedido: 29 de junio de 2023. [En línea]. Disponible en: http://data.europa.eu/eli/reg_impl/2019/947/oj/eng
- [4] EASA, «AMC and GM to Implementing Regulation (EU) 2021/664», EASA, 2022.
- [5] Commission Delegated Regulation (EU) 2019/945 of 12 March 2019 on unmanned aircraft systems and on third-country operators of unmanned aircraft systems, vol. 152. 2019. Accedido: 29 de junio de 2023. [En línea]. Disponible en: http://data.europa.eu/eli/reg_del/2019/945/oj/eng
- [6] «Specific Operations Risk Assessment (SORA)», EASA, 28 de septiembre de 2022. <https://www.easa.europa.eu/en/domains/civil-drones-rpas/specific-category-civil-drones/specific-operations-risk-assessment-sora> (accedido 17 de julio de 2023).
- [7] «SESAR Joint Undertaking | Delivering the Digital European Sky». <https://www.sesarju.eu/> (accedido 4 de julio de 2023).
- [8] «SESAR Joint Undertaking | Concept of Operations for European UTM Systems - CORUS». <https://sesarju.eu/projects/corus> (accedido 17 de julio de 2023).
- [9] radan.pesic, «CORUS-XUAM Project - Urban Air Mobility for Europe», CORUS-XUAM - Concept of Operations for Urban Air Mobility in Europe. <https://corus-xuam.eu/> (accedido 17 de julio de 2023).
- [10] CORUS-XUAM, «U-space ConOps». 13 de julio de 2022.
- [11] «Rules of the Air Annex 2 to the Convention on International Civil Aviation». julio de 2005.
- [12] Bubbles, «Bubbles Project project», Nombre_sitio. <https://bubbles-project.eu/> (accedido 13 de julio de 2023).
- [13] J. V. Balbastre Tejedor, C. Claramunt Puchol, y N. Vera Velez, «BUBBLES CONCEPT FORMULATION», SESAR Joint Undertaking, D2.1, sep. 2022.
- [14] M. Nikolaeva, O. Aranda, O. Henrik Dahle, y M. Baena, «USEPE Concept of Operations Outline», D3.3, dic. 2022.
- [15] «SESAR Joint Undertaking | SPATIO- U-Space Separation Management».

<https://www.sesarju.eu/projects/SPATIO> (accedido 17 de julio de 2023).

- [16] H. R. Tiwary, «On the Hardness of Computing Intersection, Union and Minkowski Sum of Polytopes», *Discrete Comput. Geom.*, vol. 40, n.º 3, pp. 469-479, oct. 2008, doi: 10.1007/s00454-008-9097-3.
- [17] R. B. Ash, *Information Theory*. Dover Publications, 1990.
- [18] Champman & HALL/CRC, *Handbook of Data Structures and Applications*, 2.^a ed. CRC press.
- [19] D. Sui y K. Zhang, «A Tactical Conflict Detection and Resolution Method for En Route Conflicts in Trajectory-Based Operations», *J. Adv. Transp.*, vol. 2022, pp. 1-16, feb. 2022, doi: 10.1155/2022/9283143.

Anexo A Estructura de un U-plan

A continuación, se muestra un ejemplo de un U-plan en el que puede verse la estructura que sigue y la información que contiene:

```
{
  "UPlanID": 5,
  "TimeStart": "13:17:15 07/06/2023",
  "CruiseSpeed": 14,
  "CruiseAltitude": 359,
  "Category": "SAIL_III-IV",
  "Priority": 1,
  "Wps": [
    {
      "NumWp": 1,
      "Latitude": 39.4918,
      "Longitude": -0.3746,
      "Mode": "Inspection",
      "InspectionTime": 315,
      "InspectionRoofAltitude": 353,
      "InspectionFloorAltitude": 343,
      "InspectionRadio": 15
    },
    {
      "NumWp": 2,
      "Latitude": 39.4902,
      "Longitude": -0.4085,
      "Mode": "Pass"
    }
  ],
  "BatteryEndurance": 49,
  "State": "Pending",
  "Pilot": "John Doe",
  "Operator": "ACME Drones"
}
```

Anexo B Script para la generación de U-plans aleatorios

A continuación, se presenta el script utilizado para la generación de planes aleatorios:

```
import json
import random
import datetime
import os
Category=['A1','A2','SAIL_I-II','SAIL_III-IV','SAIL_V-VI','Certi_No_Pass']
def generate_json(num_jsons, lat_range, long_range, min_wps, max_wps,start_range)
    #os.makedirs("jsons_1000_2h", exist_ok=True)
    for i in range(num_jsons):
        json_data = {}
        json_data['UPlanID'] = i + 1
        time_start_sec = int(random.uniform((start_range[0] - datetime.datetime.now()).total_seconds(),
(start_range[1] - datetime.datetime.now()).total_seconds()))
        time_start = datetime.datetime.now() + datetime.timedelta(seconds=time_start_sec)
        json_data['TimeStart'] = time_start.strftime("%H:%M:%S %d/%m/%Y")
        json_data['CruiseSpeed'] = random.randint(10, 30)
        json_data['CruiseAltitude'] = random.randint(0, 500)
        json_data['Category'] = random.choice(Category)
        json_data['Priority'] = random.randint(1, 5)
        num_wps = random.randint(min_wps, max_wps)
        json_data['Wps'] = []
        for j in range(num_wps):
            wp = {}
            wp['NumWp'] = j + 1
            wp['Latitude'] = round(random.uniform(lat_range[0], lat_range[1]), 4)
            wp['Longitude'] = round(random.uniform(long_range[0], long_range[1]), 4)
            if random.randint(1, 10)<8:
                wp['Mode'] = 'Pass'
                wp['PassMode'] = 'Fb'
            else:
                wp['Mode'] = 'Inspection'
                wp['InspectionTime'] = random.randint(300, 900)
                wp['InspectionRoofAltitude'] = random.randint(350, 450)
                wp['InspectionFloorAltitude'] = random.randint(250, 350)
                wp['InspectionRadio'] = random.randint(10, 30)
            json_data['Wps'].append(wp)
        json_data['BatteryEndurance'] = random.randint(40, 80)
        json_data['State'] = 'Pending'
        json_data['Pilot'] = 'John Doe'
        json_data['Operator'] = 'ACME Drones'
        json_str = json.dumps(json_data)
        with open(f"jsons_2000_4h\plan{i+1}.json", 'w+') as f:
            f.write(json_str)
```

Ejemplo de uso:

```
start_range = [datetime.datetime.now(), datetime.datetime.now() +  
datetime.timedelta(hours=4)]  
generate_json(2000, [39.4988, 39.4167], [-0.4257,-0.3180], 2, 7, start_range)
```

II. PRESUPUESTO

Para finalizar este trabajo, se proporcionará información detallada sobre los costes aproximados necesarios para llevar a cabo este proyecto. Para establecer el presupuesto, se ha dividido en tres categorías. En primer lugar, se han considerado los costes relacionados con el personal involucrado. En segundo lugar, se han tomado en cuenta los costes de los materiales utilizados, incluyendo equipos y software. Por último, se han evaluado los costes indirectos.

Coste de personal

En cuanto al coste de personal, se refiere a la mano de obra requerida para llevar a cabo este proyecto. En este caso, el equipo está compuesto por un estudiante de Ingeniería Aeroespacial, un tutor con título en Ingeniería Aeroespacial y estudiante de doctorado y un doctor ingeniero de telecomunicación. Los detalles sobre el total de horas y el precio unitario de cada miembro se encuentran detallados en la siguiente tabla

Tabla 10-1 Coste de personal

Concepto	Cantidad (h)	Coste (€/h)	Total (€)
Mano de obra Estudiante	436	8,00	3.488,00
Mano de obra Cotutora	35	19,52	638,20
Mano de obra Tutor	15	39,10	586,50
Coste Total			4.712,70

Coste de materiales

En cuanto a los recursos necesarios, se considera el coste de los materiales utilizados en la ejecución del proyecto, incluyendo la depreciación de los equipos empleados y las licencias de software utilizadas. Además, se contemplan los materiales de oficina empleados, como papel, bolígrafos, calculadoras, entre otros. En la Tabla 10-2 se detalla el desglose de los costes y el total correspondiente a esta categoría.

Tabla 10-2 Coste de materiales

Concepto	Periodo de amortización	Coste (€)	Periodo de uso	Total (€)
Microsoft 365 Empresa Estándar	1 mes	11,70	5 meses	58,50
MATLAB	12 meses	875,76	5 meses	364,90
Lenovo thinkCenter M90s Intel Core i5 2.90 GHz 16GB RAM Intel UHD graphics 630	36 meses	964,50	5 meses	133,96
“PC ensamblado personalizado” Intel core i7-11700K @3.60 GHz 64 GB RAM intel UHD graphics 750	36 meses	1590,00	2 meses	88,33
Material de oficina				30,00
Coste Total				675,69

Costes indirectos

Los costes indirectos son aquellos gastos que no están directamente relacionados con una actividad o proyecto específico, pero que son necesarios para llevarlo a cabo. Estos costes se generan debido a la infraestructura, el personal de apoyo y otros gastos generales que son compartidos entre diferentes proyectos o actividades. Debido a su dificultad de cálculo, habitualmente se utilizan datos históricos o evaluación de expertos, sin embargo, como este análisis se escaba del objetivo de este trabajo, simplemente lo asumiremos como un 10% del total de los costes anteriores. El monto de esta cantidad se indica en la siguiente tabla.

Tabla 10-3 Costes indirectos

Concepto	Costes directos (€)	porcentaje	Total (€)
Costes indirectos	5.388,39	10%	538,84

Presupuesto total

Tomando en cuenta la descomposición del presupuesto diseñado, se llega al importe global del proyecto ejecutado, que abarca todos los aspectos relacionados con el estudio y progreso del proyecto presentado en el informe.

A dicho importe se le deberá agregar el Impuesto sobre el Valor Añadido (IVA) del 21%. Todos estos detalles quedan registrados en la Tabla 10-4.

Tabla 10-4 Presupuesto total del proyecto

Concepto	Total (€)
Costes de personal	4.712,70
Costes de material	675,69
Costes indirectos	538,84
Total (IVA exc)	5.927,23
IVA (21%)	1.244,72
Coste Total	7.171,95

Con todo esto se concluye que realizar este proyecto ha tenido un coste de **SIETE MIL CIENTO SETENTA Y UN EUROS con NOVENTA Y CINCO CÉNTIMOS.**

III. PLIEGO DE CONDICIONES

Para garantizar el correcto funcionamiento y rendimiento óptimo de la herramienta desarrollada en este Trabajo de Fin de Grado, es fundamental tener en cuenta ciertas condiciones y requisitos. Estas condiciones aseguran que el sistema opere de manera efectiva, obteniendo resultados precisos y confiables. En esta sección, se describirán las condiciones necesarias para el uso adecuado de la herramienta, tanto en términos de software como de hardware, así como la estructura requerida para el plan de vuelo propuesto.

Este sistema ha sido implementado en la versión R2021B de Matlab, por lo que no se asegura su correcto funcionamiento en versiones anteriores, a continuación, se explica cómo debe utilizarse.

Ejecución del programa

Para la ejecución del programa deberemos abrir MATLAB situando el Directorio en la carpeta SCRS, para que funcione correctamente esta carpeta debe tener al menos los siguientes archivos y carpetas:

- “SCRS.m” definición de la clase SCRS
- “Uplan.m” definición de la clase Uplan
- “waypoint.m” definición de la clase wayPoint
- “octTree.m” definición de la clase Octree
- functions carpeta con diversas funciones de apoyo que necesitan las clases
- jsons carpeta que almacena todos los archivos jsons a deconflitar
- separaciones_SCRS.cvs archivo que contiene las separaciones por categorías
- “main.m” archivo principal

Cuando se quiera ejecutar el código, abra que ejecutar el archivo “main.m”.

Entradas y modificaciones:

Los planes de vuelo que se deconflitarán serán los que se encuentren en la carpeta “jsons”, si se quiere realizar modificaciones sobre los parámetros del SCRS se deberán realizar en el archivo “main.m” como se muestra en la siguiente tabla:

Tabla 10-5 instrucciones para modificar los parametros de la herammienta

<i>paramtero</i>	<i>variable</i>	<i>linea</i>
<i>Latitud mínima de la zona de cobertura</i>	<i>“latMin”</i>	8
<i>Latitud máxima de la zona de cobertura</i>	<i>“latMax”</i>	9

paramtero	variable	línea
Longitud mínima de la zona de cobertura	"lonMin"	10
Longitud máxima de la zona de cobertura	"lonMax"	11
Nombre de la carpeta con los U-plans	"jsonsDirectory"	13
Tiempo de protección ¹	"Buffer"	14
dt	"dt"	15
Factor dt-Tiempo de protección	"factorDtBuff"	16

El tiempo de protección puede ser asimétrico, lo que significa que el tiempo de protección previo y posterior al inicio estimado del vuelo pueden ser diferentes, por lo que el tiempo de protección tiene dos componentes, la primera indica el tiempo previo al inicio y debe ser negativo o 0 (t-), y la segunda el tiempo posterior, que debe ser positivo o 0 (t+), de forma que la variable se asigna como Buffer=[t-, t+].

Outputs:

Actualmente el sistema no devuelve ningún archivo, el resultado final tras el deconflicto y la activación de los U-plans se guarda en el "workspace", bajo una variable llamada "service" de tipo SCRS, el formato de esta clase se encuentra en la memoria, esta puede ser guardada si se desea llamando al siguiente comando desde la línea de comandos.

```
save '[nombre_archivo].mat' service
```

sustituyendo [nombre_archivo] por el nombre del archivo con el que se desee guardar la variable.

Cuando un Uplan no es aceptado el sistema devuelve en la línea de comandos el Uplan con el que ha tenido conflicto, de igual forma si para aceptar un Uplan es necesario eliminar planes ya aceptados, estos se eliminan y se devuelven por la línea de comandos

Si el sistema se integra con otros servicios U-space, especialmente el CMS (Sistema de Gestión del Espacio Aéreo), pueden obtener la trayectoria como un vector de estructuras con los siguientes campos:

- "x" coordenada x
- "y" coordenada y
- "z" coordenada z

- “time” instante de tiempo

Características del sistema:

Para obtener resultados similares a los descritos en la memoria, se recomienda utilizar un ordenador con al menos las siguientes características:

Procesador:	Intel core i7-11700K @3.60 GHz
RAM	64 GB
Sistema operativo	Windows 10 Pro versión 22H2

Estructura del U-plan:

El U-plan, que representa el plan de vuelo propuesto, debe seguir la siguiente estructura:

Campo	Nombre	Descripción
ID:	"UPlanID"	Identificador único de U-plan
Fecha de inicio:	"TimeStart"	Fecha y hora de inicio del vuelo, en el siguiente formato: HH:mm:ss dd/mm/aaaa
Velocidad de crucero	"CruiseSpeed"	Velocidad de crucero en m/s
Altitud de crucero	"CruiseAltitude"	Altitud de crucero en m
Prioridad	"Priority"	Prioridad de la operación
categoría	"Category"	Categoría de la operación
Estado	"State"	Estado en el que se encuentra el <i>U-plan</i>
	"Wps"	Listado con los waypoints que conforman la trayectoria, este listado debe tener los siguientes campos:
Listado de waypoints	Campo	Nombre
	Latitud	"Latitude"
	Longitud	"Longitude"

Modo	“Modo”	Paso o altitud
Tiempo de inspección	“InspectionTime”	En segundos.
Radio de inspección	“InspectionRadio”	En metros.
Techo de inspección	“InspectionRoofAltitude”	Altitud máxima de inspección
Suelo de inspección	“InspectionFloorAltitude”	Altitud máxima de inspección.

El Json puede tener más información que el sistema es capaz de ignorar. Si se desea integrar el sistema con otros servicios U-space, el U-plan también debe cumplir con esta estructura.