



Universitat Autònoma de Barcelona

**ADVERTIMENT.** L'accés als continguts d'aquesta tesi queda condicionat a l'acceptació de les condicions d'ús establertes per la següent llicència Creative Commons:  [http://cat.creativecommons.org/?page\\_id=184](http://cat.creativecommons.org/?page_id=184)

**ADVERTENCIA.** El acceso a los contenidos de esta tesis queda condicionado a la aceptación de las condiciones de uso establecidas por la siguiente licencia Creative Commons:  <http://es.creativecommons.org/blog/licencias/>

**WARNING.** The access to the contents of this doctoral thesis it is limited to the acceptance of the use conditions set by the following Creative Commons license:  <https://creativecommons.org/licenses/?lang=en>



**Universitat Autònoma de Barcelona**

**Propuesta de sistema multi-UAV para aplicaciones de  
cobertura de área**

**Ernesto Emmanuel Santana Cruz**

**Tesis Doctoral**

**Programa de Estudios de Doctorado en Telecomunicación e  
Ingeniería de Sistemas**

**Director:**

**Dr. Romualdo Moreno Ortiz**

**Departamento de Telecomunicaciones e Ingeniería de Sistemas**

**Escuela de Ingeniería**

**2017**

Memòria presentada per aspirar al Grau de Doctor per

Ernesto Emmanuel Santana Cruz

Vist i plau

Dr. Romualdo Moreno Ortiz

Departament de Telecomunicació i d'Enginyeria de Sistemes

Escola d'Enginyeria

Universitat Autònoma de Barcelona

Bellaterra 16/06/2017

## **Agradecimientos**

Le doy gracias a Dios de todo corazón por el hermoso regalo de la vida, por todas las experiencias que me ha permitido vivir durante todos estos años, por siempre estar a mi lado y por darme el impulso para levantarme cada vez que lo he necesitado.

Mil gracia a toda mi familia. A mi esposa María, la que ha sabido estar a mi lado en todo momento, brindándome su amor y comprensión cuando más lo he necesitado. Gracias bobita. A mi pequeño Haziél, que con su pequeña sonrisa es capaz de mostrarme lo afortunado que soy por tenerle. Gracias mi niño por enseñarme a valorar las cosas que realmente importan. A mi mamá Leonidas y mi papá Luis que han hecho posible que hoy esté donde esté, sin ustedes nada sería. Gracias por haberme dado todo lo que estuvo en sus manos para que yo sea feliz, gracias por estar a mi lado a pesar de la distancia. Los quiero mucho mis viejitos.

Al Dr. Romualdo Moreno por abrirme las puertas del Lab. de UAVs, y darme la oportunidad de desarrollar esta tesis doctoral, y descubrir el apasionante mundo de los UAVs. Le estaré por siempre agradecido.

Gracias a todos mis compañeros de tesis, lo que están y los que ya se han ido. También muchas gracias a todos los profesores del grupo de investigación de Logística y Aeronáutica. Muchas gracias a todos por su comprensión, amabilidad y cordialidad para conmigo.

Ernesto Santana

# Contenido

Lista de Figuras.....	vii
Lista de Tablas.....	xi
Capítulo 1 Introducción.....	1
1.1 Motivación.....	2
1.2 Objetivos .....	3
1.3 Referencias.....	6
Capítulo 2 Fundamentos conceptuales.....	7
2.1 UAV.....	7
2.1.1 Tipos de UAVs .....	9
2.1.2 Subsistemas y/o elementos de un UAV.....	17
2.2 Subsistema de comunicación.....	22
2.3 Estación de control en tierra.....	24
2.3.1 Software de planificación, control y seguimiento de misiones.....	25
2.4 Carga útil .....	26
2.5 Referencias.....	28
Capítulo 3 Entorno de simulación multi-UAV .....	29
3.1 Introducción .....	29
3.1.1 Simulación Hardware in the Loop.....	29
3.1.2 Simulación Software in the Loop.....	30
3.2 Antecedentes .....	31
3.3 Arquitectura del entorno de simulación propuesto .....	33
3.3.1 Simulación de UAVs .....	35
3.3.2 Estación de Control en tierra (GCS) .....	39
3.3.3 Comunicación.....	42
3.4 Referencias.....	45
Capítulo 4 Arquitectura multi-UAV .....	47

4.1	Introducción .....	47
4.1.1	Clasificación de arquitecturas multi-UAV según el grado de centralización del sistema	48
4.1.2	Clasificación de arquitecturas multi-UAV según el nivel de interacción entre UAVs	50
4.2	Antecedentes .....	52
4.3	Arquitectura de control y cooperación propuesta .....	54
4.3.1	Capa centralizada .....	57
4.3.2	Capa Distribuida .....	63
4.4	Referencias .....	68
Capítulo 5	Planificación de misiones para aplicaciones de cobertura de área con sistemas multi-UAV	70
5.1	Introducción .....	70
5.2	Antecedentes .....	74
5.3	Formulación del problema .....	76
5.3.1	Escenario .....	78
5.3.2	Enfoque .....	81
5.4	Planificación de rutas para la cobertura de un área en tiempo mínimo .....	81
5.4.1	Dirección óptima de barrido .....	82
5.4.2	Particionado del área objetivo y planificación de rutas .....	85
5.5	Referencias .....	103
Capítulo 6	Capa descentralizada para la reasignación de tareas de forma cooperativa..	105
6.1	Introducción .....	105
6.2	Antecedentes .....	109
6.3	Formulación del Problema .....	111
6.3.1	Escenario .....	112
6.3.2	Enfoque .....	114
6.4	Estrategia propuesta para la reasignación de tareas de forma cooperativa basada en mecanismos de subastas .....	115
6.4.1	Disparadores del proceso de reasignación de waypoints .....	116
6.4.2	Algoritmo de reasignación de tramos de waypoints basado en mecanismos de subastas .....	121
6.5	Referencias .....	136
Capítulo 7	Conclusiones y Trabajo Futuro .....	138
7.1	Contribuciones de la presente tesis.....	<b>¡Error! Marcador no definido.</b>

7.2 Líneas de trabajo futuro.....¡Error! Marcador no definido.

## Lista de Figuras

Figura 2.1– Esquema general de un sistema de vehículos aéreos no tripulados.....	8
Figura 2.2– Clasificación en función del método de sustentación utilizado .....	9
Figura 2.3– UAV de ala fija .....	10
Figura 2.4– UAV con un rotor principal y uno de cola .....	11
Figura 2.5– Prototipo de un UAV con un único rotor .....	12
Figura 2.6– UAV con rotores en configuración coaxial.....	13
Figura 2.7– Micro UAV tipo helicóptero en configuración tándem .....	13
Figura 2.4– UAV quadróptero.....	14
Figura 2.9– UAV Híbrido.....	15
Figura 2.5– Burbuja de enlace de comunicación VLOS.....	23
Figura 2.6– Ejemplo de una Estación móvil de control en tierra, instalada en un furgón (Propiedad: Laboratorio de UAVS de la UAB).....	24
Figura 3.1– Esquema de la arquitectura del entorno de simulación multi-UAV .....	34
Figura 3.2– Entorno mixto, interacción de UAV real con un UAV simulado .....	35
Figura 3.3– Simulador SiL de un único UAV presente en cada máquina virtual.....	36
Figura 3.4– El software MAVProxy siendo ejecutado en dos máquinas virtuales distintas...37	
Figura 3.5– Esquema de conexión múltiple.....	41
Figura 3.6– Interfaz gráfica de usuario del módulo de conexión múltiple .....	41
Figura 3.7– Configuración de la tarjeta Ethernet PCI virtual .....	42
Figura 3.8– Infraestructura de red.....	43
Figura 4.1– Ejemplo de interacción física entre UAVs .....	50
Figura 4.2– Vuelo en formación.....	51
Figura 4.3– Enjambre de UAVs .....	51
Figura 4.4– Ejemplo de cooperación intencionada entre UAVs .....	52
Figura 4.5– Esquema general de la arquitectura multi-UAV propuesta.....	56
Figura 4.6– Esquema de la capa centralizada .....	57
Figura 4.7– Interfaz de usuario para ingresar los parámetros de vuelo.....	58

Figura 4.8- Escritura de waypoints .....	59
Figura 4.9 - Diagrama de flujo del proceso de planificación de misiones .....	59
Figura 4.10 - Pasos para iniciar la ejecución de la misión.....	61
Figura 4.11 - Esquema general del módulo de monitorización.....	62
Figura 4.12 - HUD ( <i>Head UP Display</i> ).....	63
Figura 4.13 - Arquitectura UAV embarcada.....	66
Figura 5.1 - Ejemplo de descomposición en celdas exacta.....	72
Figura 5.2 - Ejemplo de descomposición en celdas aproximada.....	72
Figura 5.3 - Patrón de movimiento <i>Back and forth</i> .....	74
Figura 5.4 - Patrón de movimiento <i>Spiral-like</i> .....	74
Figura 5.5 - Ejemplo del escenario planteado .....	78
Figura 5.6 - Campo visual de la cámara .....	81
Figura 5.7 - Visualización de la variación del número de giros dependiendo de la dirección de barrido adoptada .....	82
Figura 5.8 - Determinación de la anchura del polígono $P$ mediante el trazado de los pares antipodales.....	83
Figura 5.9 - Ejemplo de un tramo básico de ida y vuelta, <i>trb</i> .....	86
Figura 5.10 - Zona de lanzamiento definida para tres UAVs .....	87
Figura 5.11 - Situaciones que pueden presentarse cuando existe una única zona de lanzamiento .....	88
Figura 5.12 - Ejemplo de sobrevuelo de una subregión con camino de acceso.....	89
Figura 5.13 - Test para determinar si se necesita construir un camino de acceso.....	90
Figura 5.14 - Test para determinar qué punto se usará como referencia.....	91
Figura 5.15 - Construcción del tramo de ida.....	92
Figura 5.16 - Trazado del tramo de salida .....	93
Figura 5.17 - Caminos de acceso finalizado el proceso de particionado.....	94
Figura 5.18 - Estrategia de particionado y planificación de rutas .....	95
Figura 5.19 - Ejemplo de desequilibrio en la carga de trabajo .....	96
Figura 5.20 - Implementación en la estación de control en tierra de la estrategia de particionado de área y planificación de rutas presentada en este capítulo .....	98
Figura 5.21 - Región empleada para la simulación de cobertura de área.....	99
Figura 5.22 - Simulación de la cobertura del área con un único UAV .....	100
Figura 5.23 - Descomposición del área objetivo en 3 subregiones.....	101
Figura 6.1 - Disparador de proceso de reasignación por falla de batería.....	118
Figura 6.2 - Respuesta de la GCS al mensaje del disparador por problemas de batería.....	119

Figura 6.3 - Disparador por pérdida de la aeronave.....	120
Figura 6.4 - Disparador del proceso de reasignación por pérdida de un UAV ejecutado por la GCS.....	121
Figura 6.5 - Esquema de la subasta de un determinado waypoint desde el punto de vista del subastador.....	122
Figura 6.6 - Esquema de la subasta de un determinado waypoint desde el punto de vista de las aeronaves.....	123
Figura 6.7 - Ejemplo del mecanismo de replanificación de rutas ejecutado de forma local.....	126
Figura 6.8 - Sección de la interfaz de usuario que permite iniciar la estrategia de reasignación de waypoints de forma cooperativa.....	127
Figura 6.9 - Rutas de vuelo asignadas a cada UAV empleando el algoritmo de asignación presentado en el capítulo 5.....	128
Figura 6.10 - Caso 1 de Reasignación de waypoints.....	130
Figura 6.11 - Caso 2 de Reasignación de waypoints.....	130
Figura 6.12 - Caso 3 de Reasignación de waypoints.....	131
Figura 6.13 - Caso 4 de Reasignación de waypoints.....	131
Figura 6.14 - Caso 5 de Reasignación de waypoints.....	132
Figura 6.15 - Caso 6 de Reasignación de waypoints.....	132
Figura 6.16 - Caso 7 de Reasignación de waypoints.....	133
Figura 6.17 - Caso 8 de Reasignación de waypoints.....	133
Figura 6.18 - Caso 9 de Reasignación de waypoints.....	134
Figura 6.19 - Tiempos de vuelo para cada UAV fruto de la ejecución de la estrategia de reasignación de waypoints.....	134
Figura 6.20 - Incremento del tiempo global de misión debido al proceso de reasignación de waypoints.....	135



## Lista de Tablas

Table 1.1 – Some random maths .....	<b>¡Error! Marcador no definido.</b>
Table 1.2 – Some random maths .....	<b>¡Error! Marcador no definido.</b>
Tabla 2.1 – Clasificación en función del MTOW, rango de cobertura, altitud máxima y autonomía en tiempo de vuelo .....	16
Tabla 2.2 – Clasificación según el nivel de autonomía en la toma de decisiones de la aeronave .....	17
Tabla 2.3 – Autopilotos más populares del mercado .....	21
Tabla 2.4 – Comparación de los programas de planificación y control más usados.....	26
Tabla 5.1 – Parámetros de configuración .....	99
Tabla 5.2 – Resultado de la simulación para la flota de UAVs.....	102
Tabla 6.1 – Estados del indicador <i>flag</i> .....	116
Tabla 6.2 – Casos de reasignación de waypoints simulados .....	129



## Capítulo 1 Introducción

Hoy en día el uso de Sistemas multi-UAV en el ámbito civil está experimentando un notable aumento, debido a los grandes beneficios que aporta este tipo de sistemas. Y es que el empleo de varias aeronaves permite elevar sustancialmente el grado de eficiencia y seguridad [1] a la hora de realizar misiones relativamente complejas, extendiendo los límites que impone la utilización de un único UAV. La mejora de la eficiencia se evidencia esencialmente en la disminución de los tiempos de ejecución de tareas, ya que al distribuir las entre las distintas aeronaves, se genera una menor carga de trabajo para cada una de ellas [2]. Por otro lado, el incremento de la seguridad y robustez que ofrece los sistemas multi-UAV viene dado por el hecho de que al disponer de varias aeronaves se eleva el nivel de redundancia lo cual sumado a la posibilidad de implementar estrategias de cooperación, hace que puedan adaptarse mejor a entornos dinámicos aumentando con ello el nivel de tolerancia a fallos.

A su vez, los sistemas multi-UAV pueden estar conformados con aeronaves que posean diferentes características y/o capacidades, las cuales al complementarse generan una mayor eficiencia en la utilización de los recursos ya que permiten una amplia gama de configuraciones posibles. Por ejemplo, en el caso de un sistema multi-UAV compuesto por tres aeronaves que tienen como objetivo inspeccionar un área en busca de un grupo de personas que se encuentran en paradero desconocido, dos de las aeronaves son quadrópteros y la tercera es un aeroplano. En esta misión debido a que el aeroplano puede volar más rápido y cubrir más extensión de terreno empleando menor cantidad de energía, se podría configurar para que se encargue de realizar la inspección de toda el área, pero cuando este mediante su sistema de captura de imágenes detecta lo que podría ser un

grupo de personas, emite un mensaje de alerta a los dos quadrópteros al poder realizar un vuelo a menor altura y a baja velocidad podrán ir a la zona para inspeccionar el área de forma más exhaustiva y así confirmar o no el hallazgo.

Aplicaciones que implican realizar la cobertura de un área determinada como pueden ser la de búsqueda y rescate [4], detección y control de incendios forestales [5], vigilancia [6], y agricultura de precisión [7], pueden beneficiarse notablemente del uso de sistemas multi-UAV. Sin embargo para poder conseguir dichos beneficios se deben superar algunos retos en cuanto a coordinación y toma de decisiones.

### **1.1 Motivación**

La realización de esta tesis doctoral se enmarca en el proyecto de investigación “*Fire Guided Unmanned Aircrafts and Resources Distribution (FireGUARD)*” (TIN2011 - 29494 -C03), específicamente en la fase que comprende la integración de aeronaves no tripuladas en las tareas de prevención y control de incendios forestales. Es por ello que nuestra principal motivación para la elaboración de esta tesis ha sido el desarrollo de un sistema multi-UAV que permita obtener información fiable que ayude al proceso de toma de decisiones para la prevención y control de incendios.

Sin embargo, para conseguir esto se requiere el desarrollo de nuevas estrategias de particionado de área y de planificación de rutas, así como la implementación de estrategias de coordinación que, mediante el intercambio de información entre los UAVs acerca del desenvolvimiento de la misión, tengan como objetivo la realización de los ajustes necesarios para garantizar el cumplimiento de los objetivos establecidos.

## 1.2 Objetivos

El objetivo general de esta tesis es el diseño y desarrollo de un sistema multi-UAV robusto y escalable que permita la ejecución de aplicaciones de cobertura de área de forma eficiente. Dentro de este objetivo general podemos listar los siguientes objetivos secundarios:

- Desarrollar una plataforma de simulación multi-UAV empleando la técnica Software in the Loop
- Diseñar la arquitectura de toma de decisiones del sistema multi-UAV
- Desarrollar una nueva estrategia de planificación de tareas para aplicaciones de cobertura de área
- Diseñar e implementar una estrategia cooperativa que permita la reasignación de waypoints de forma descentralizada en caso de presentarse un accidente o algún fallo técnico

La fase de desarrollo de algoritmos para sistemas multi-UAV requiere de la realización de múltiples pruebas que podrían poner en riesgo la integridad de las aeronaves. Para evitar correr los riesgos que implican las pruebas reales, uno de los principales objetivos de la presente tesis es la creación de un entorno de simulación Software in the Loop que permita la realización de simulaciones tanto de multirrotores como de aviones. Este entorno de simulación deberá ser modular y deberá emplear el mismo software de planificación que el que emplee la arquitectura multi-UAV propuesta.

Se deberá diseñar una arquitectura híbrida de dos capas para llevar a cabo tanto los procesos de control como los de toma de decisiones, la primera capa poseerá una arquitectura centralizada, mientras que la arquitectura de la segunda capa será distribuida. En la capa centralizada se llevará a cabo todo el proceso de generación y asignación de las tareas de cada misión, descomposición del área objetivo y la posterior asignación de rutas. Todo ello teniendo en cuenta las capacidades de los distintos UAVs presentes en el sistema

multi-UAV. Una vez los UAVs tienen asignadas las tareas, regiones de observación y rutas, estos iniciarán la misión e inmediatamente entrarán en la capa distribuida, donde se llevará a cabo todo el proceso de cooperación.

Para garantizar la cobertura del área objetivo en el menor tiempo posible, se deberá desarrollar una nueva estrategia de particionado de área y planificación de rutas. Dicha estrategia deberá tener en cuenta que todos los UAVs serán lanzados desde una misma zona, y a su vez las rutas resultantes nunca deberán entrecruzarse para así evitar problemas de posibles interferencias físicas.

Para dotar al sistema multi-UAV de la capacidad de adaptación y reconfiguración ante posibles fallos e imprevistos, se deberá desarrollar una estrategia cooperativa de replanificación de rutas.

### 1.3 Estructura del documento

En el presente apartado se detalla la estructura de este documento, describiendo el objetivo principal de cada capítulo.

**Capítulo 1.** Describe el contexto y la motivación que llevan al desarrollo de esta tesis. Incluye un resumen de los objetivos marcados, y la descripción de la estructura del documento.

**Capítulo 2.** Introduce brevemente algunos conceptos que conforman la base sobre la que se elabora esta tesis. Para ello da una breve explicación de qué es un sistema de vehículos aéreos no tripulados así como de todos los elementos que lo constituyen.

**Capítulo 3.** Presenta el desarrollo de un entorno de simulación multi-UAV, el cual ha sido utilizado como entorno de pruebas para el desarrollo e implementación del sistema multi-UAV propuesto en la presente tesis.

**Capítulo 4.** Describe de forma detallada la arquitectura de toma de decisiones del sistema multi-UAV propuesto.

**Capítulo 5.** Explica la estrategia de planificación de misión para aplicaciones de cobertura de área en tiempo mínimo, propuesta en esta tesis. Incluye una exposición de los antecedentes encontrados en la literatura para la resolución del problema de cobertura utilizando UAVs, una descripción formal del problema, la descripción de la estrategia propuesta y su posterior implementación en el entorno de simulación.

**Capítulo 6.** Explica la estrategia de reasignación de tareas de forma cooperativa. Contiene una recopilación de algunos de las propuestas más destacados encontradas en la bibliografía científica para el problema de asignación de tareas de forma descentralizada en entornos multi-UAV, una descripción formal del problema, la presentación en detalle de la estrategia de reasignación propuesta y su implementación en la plataforma de simulación multi-UAV.

**Capítulo 7.** Presenta las aportaciones de esta tesis y las posibles líneas futuras de trabajo.

## 1.4 Referencias

- [1] Geramifard, A., Redding, J., Roy, N., & How, J. P. (2011, June). UAV cooperative control with stochastic risk models. In *American Control Conference (ACC), 2011* (pp. 3393-3398). IEEE.
- [2] Nemra, A., & Aouf, N. (2010, September). Robust cooperative UAV visual SLAM. In *Cybernetic Intelligent Systems (CIS), 2010 IEEE 9th International Conference on* (pp. 1-6). IEEE.
- [3] Morgenthaler, S., Braun, T., Zhao, Z., Staub, T., & Anwander, M. (2012, December). UAVNet: A mobile wireless mesh network using unmanned aerial vehicles. In *Globecom Workshops (GC Wkshps), 2012 IEEE* (pp. 1603-1608). IEEE.
- [4] Scherer, J., Yahyanejad, S., Hayat, S., Yanmaz, E., Andre, T., Khan, A., ... & Rinner, B. (2015, May). An autonomous multi-uav system for search and rescue. In *Proceedings of the First Workshop on Micro Aerial Vehicle Networks, Systems, and Applications for Civilian Use* (pp. 33-38). ACM.
- [5] Ollero, A., & Merino, L. (2006). Unmanned aerial vehicles as tools for forest-fire fighting. *Forest Ecology and Management*, 234(1), S263.
- [6] Adamey, E., & Ozguner, U. (2012, May). A decentralized approach for multi-UAV multitarget tracking and surveillance. In *SPIE Defense, Security, and Sensing* (pp. 838915-838915). International Society for Optics and Photonics.
- [7] Cao, J., Chen, Y., & Li, C. (2015, July). Multi-UAV-based optimal crop-dusting of anomalously diffusing infestation of crops. In *American Control Conference (ACC), 2015* (pp. 1278-1283). IEEE.

## Capítulo 2 Fundamentos conceptuales

En este capítulo se incluye una introducción a distintos conceptos básicos referentes a los sistemas de vehículos aéreos no tripulados.

La sección 2.1 comienza presentando qué es un UAS y los distintos elementos que lo conforman, los cuales son explicados en detalle en los sub-apartados de esta sección.

En 2.1.1 se aborda el concepto UAV, presentando a su vez las diferentes taxonomías empleadas para la clasificación de este tipo de aeronaves y los subsistemas y/o elementos propios de un UAV. Por otro lado, en 2.1.2 se exponen los conceptos básicos de la capa de comunicación de los UAS. Y por último en 2.1.2 se explica en detalle qué es la estación de control y el software que en ella se ejecuta.

### 2.1 UAS

Un sistema de vehículos aéreos no tripulados, también llamado UAS por sus siglas en inglés (*Unmanned Aerial Systems*), puede ser definido como el conjunto de elementos y subsistemas que hacen posible llevar a cabo operaciones aerotransportadas sin la utilización de un piloto a bordo de la aeronave. Por lo general, estos sistemas están compuestos por uno o más vehículos aéreos no tripulados (UAVs, *Unmanned Aerial Vehicles*), una o varias estaciones de control en tierra (GCS, *Ground Control Station*), elementos embarcados también conocidos como carga útil y un subsistema de comunicaciones para el intercambio de información entre los distintos elementos del sistema (ver figura 2.1).

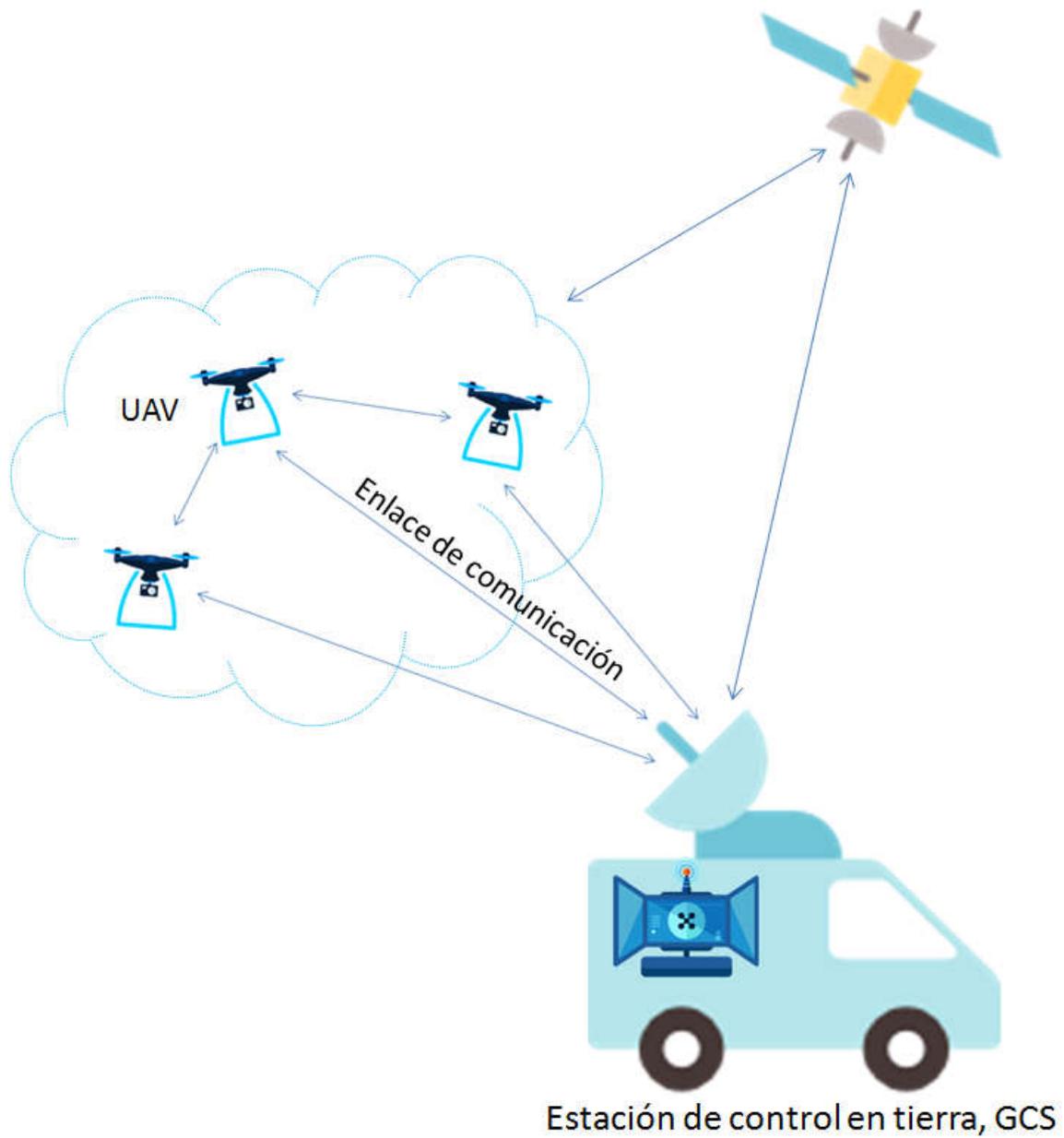


Figura 2.1– Esquema general de un sistema de vehículos aéreos no tripulados

Cuando el sistema consta de varias aeronaves, este es comúnmente llamado sistema multi-UAV.

En los siguientes sub-apartados explicaremos cada uno de los elementos principales que constituyen un UAS.

## 2.1.1 UAV

Mayoritariamente se define como UAV a toda aeronave capaz de volar sin necesidad de llevar un piloto humano a bordo. Los vehículos balísticos o semi balísticos, misiles de crucero y proyectiles de artillería no son considerados UAVs [1], sin importar de que estos lleven o no un sistema de guiado inteligente.

### 2.1.1.1 Tipos de UAVs

Debido a la gran diversidad de UAVs existente y a las múltiples misiones que éstos pueden ejecutar, no existe una clasificación universalmente aceptada, sino múltiples taxonomías que se acostumbra a entremezclar.

Algunos de los métodos de clasificación más empleados en la literatura son:

- Clasificación basada en los métodos de generación de sustentación:

Este criterio de clasificación agrupa a los UAVs en dos grandes grupos, en uno de ellos se encuentran todos los aerodinos (aeronaves más pesadas que el aire) y en un segundo grupo los aerostatos (aquellas aeronaves cuya suspensión en el aire se debe al empleo de un gas más ligero que el propio aire). En la figura 2.2 se presenta el esquema de dicha clasificación.



Figura 2.2– Clasificación en función del método de sustentación utilizado

Hoy en día, los UAVs más utilizados tanto en el ámbito civil como militar son aerodinos. Por ello a continuación definiremos los tres tipos de aerodinos más comunes dentro del campo de los UAVs:

- Ala fija: Son aquellos aerodinos en los cuales las alas se encuentran unidas/encastradas con el resto de elementos de la aeronave, y no poseen movimiento propio [4] (ver figura 2.3). En su conjunto, los UAVs de ala fija se caracterizan por poseer una estructura simple, con una elevada eficiencia aerodinámica, lo que permite a estas aeronaves poseer un alto rendimiento energético. Estas características les permiten presentar tiempos de vuelo relativamente elevados. Sin embargo, estas aeronaves requieren de elementos y/o infraestructuras externas para poder realizar el despegue y el posterior aterrizaje, lo cual eleva el nivel de complejidad en la ejecución de la misión de vuelo. El nivel de maniobrabilidad es mucho menor al que poseen las aeronaves de ala rotatoria, lo que les impide ser utilizados en espacios de alta complejidad y de poca extensión.



Figura 2.3– UAV de ala fija

- UAVs de ala rotatoria: Son aquellos aerodinos en los cuales las alas, en este caso también denominadas “palas”, giran alrededor de un eje, consiguiendo de este modo la sustentación [4]. Dependiendo del número de rotores y/o de su configuración, los UAVs de ala rotatoria pueden subdividirse en los siguientes tipos [5][6]:
  - Aeronaves con un rotor principal y un rotor de cola: En este tipo de aeronave la sustentación es generada por el rotor principal el cual está situado en la parte superior del aparato, mientras el rotor de cola compensa el par de torsión que el primero produce (ver figura 2.4). Este tipo de UAV tiene buena capacidad de control y maniobrabilidad. Sin embargo, la estructura mecánica es compleja.



Figura 2.4– UAV con un rotor principal y uno de cola

- Aeronaves con un único rotor: Este tipo de UAVs poseen un único rotor para generar la sustentación y un arreglo de alerones para compensar el torque del rotor. Ver ejemplo en la figura 2.5. Tienen una estructura mecánica relativamente simple, sin

embargo son difíciles de controlar aún para pilotos experimentados.



Figura 2.5– Prototipo de un UAV con un único rotor

- Aeronaves con dos rotores en configuración coaxial: En esta configuración posee dos rotores colocados uno encima del otro (ver figura 2.6). Ambos rotores giran en direcciones opuestas, consiguiendo el empuje del vuelo a partir de la diferencia generada de la velocidad angular de ambos rotores. La configuración coaxial permite la fabricación de aeronaves compactas, pero tiene como desventaja que una cantidad significativa de la energía se pierde debido a que los rotores interfieren entre sí.



Figura 2.6– UAV con rotores en configuración coaxial

- Aeronaves con dos rotores en configuración tándem: Este tipo de aeronaves poseen dos rotores ubicados en la parte superior del aparato en configuración tándem (ver figura 2.7). Ambos rotores giran en direcciones opuestas lo cual neutraliza el torque generado. La estructura mecánica de estos UAVS es extremadamente compleja.



Figura 2.7– Micro UAV tipo helicóptero en configuración tándem

- Multirrotores: Un multirrotor es una aeronave de ala rotatoria que posee tres o más rotores. Dependiendo del número de rotores y de su configuración, los multirrotores pueden subdividirse en diferentes tipos, yendo desde aeronaves con tres

rotores (tricópteros), cuatro rotores (quadcópteros) hasta configuraciones de 8 (octocópteros) o más rotores. Al igual que los otros tipos de aeronaves de ala rotatoria, los multirrotores generan la sustentación necesaria para el vuelo a través de sus rotores, aunque en este caso, sus hélices son de paso fijo (ángulo fijo) y de revoluciones variables. Las hélices son instaladas en sentidos de rotación opuestos de forma diametral, es decir, se alternan hélices de giro a derecha con hélices de giro a izquierdas resultando nula la suma de las fuerzas que generan. Este juego de sumas y restas de fuerzas es el que permite la maniobrabilidad de este tipo de aeronaves. Si todos los rotores producen la misma fuerza de sustentación y ésta se encuentra en equilibrio con el peso de la aeronave, entonces el aparato se mantendrá en vuelo estacionario. Si este equilibrio se ve alterado porque uno de los rotores presenta más o menos velocidad angular que el resto, entonces se producirá el balanceo del aparato.



Figura 2.8– UAV quadcóptero

En su conjunto, los UAVs de ala rotatoria pueden despegar o aterrizar de forma vertical sin requerir de infraestructuras externas. Este tipo de aeronaves presentan un alto nivel de maniobrabilidad, pudiendo realizar

desde vuelos estacionarios hasta la realización de maniobras de elevado nivel de precisión, lo que les permite llevar a cabo misiones tanto en interiores como en exteriores. También son capaces de volar a alturas muy bajas con respecto al suelo, lo que les permite captar imágenes con un elevado nivel de resolución. También debido a su mecánica de vuelo poseen una buena capacidad de carga de pago. Sin embargo, al necesitar que todos sus rotores estén funcionando permanentemente para lograr la sustentación necesaria, estas aeronaves presentan tiempos de autonomía de vuelo limitados, ya que poseen un elevado consumo energético. Esto hace que los UAVS de ala rotatoria estén limitados a sobrevolar áreas relativamente pequeñas.

- UAVs híbridos: Este tipo de UAVs son capaces de despegar y aterrizar de forma vertical, como las aeronaves de ala rotatoria, y de realizar vuelos a alta velocidad, como un ala fija tradicional (ver figura 2.9). Estas aeronaves poseen redundancia de mecanismos de sustentación, lo que convierte a esta solución en una opción robusta ante fallos inesperados. Sin embargo, su estructura mecánica y de control es compleja. Fruto de esta complejidad, actualmente existen muy pocas ofertas comerciales de este tipo de UAVs, y las aeronaves que ya se encuentran en el mercado poseen precios sumamente elevados.



Figura 2.9– UAV Híbrido

- Clasificación en función de diversas características de la aeronave, como peso al despegue (MTOW, *Maximum Take-Off Weight*), rango de operación, altitud máxima y autonomía[2]:

Esta forma de clasificación, ampliamente utilizada en la bibliografía, fue propuesta por Peter van Blyenburgh en el año 2008, e intenta abarcar a todos los UAVs agrupándolos en diferentes categorías tomando en cuenta para ello 4 características diferentes: peso al despegue, rango de cobertura, altitud máxima de vuelo y autonomía de tiempo de vuelo.

Tabla 2.1 – Clasificación en función del MTOW, rango de cobertura, altitud máxima y autonomía en tiempo de vuelo

Categoría	MTOW (kg)	Rango de operacion (km)	Altitud (m)	Tiempo de vuelo (h)
Nano ( $\eta$ )	$\leq 0,025$	<1	100	<1
Micro ( $\mu$ )	5	<10	250	1
Mini	25	<10	150 - 300	<2
Cobertura 1 ( <i>Close range</i> )	150	10 - 30	3000	2 - 4
Cobertura 2 ( <i>Short range</i> )	200	30 - 70	3000	3 - 6
Altitud baja y elevada autonomía (LALE por sus siglas en inglés)	<25	> 500	3000	> 24
Altitud media y elevada autonomía (MALE por sus siglas en inglés)	1500	> 500	14000	24 - 48
Altitud alta y elevada autonomía (HALE por sus siglas en inglés)	4500 - 12000	> 2000	20000	24 - 48
Estratosférico	25 – ?	> 2000	20000 - 30000	> 48

- Clasificación basada en el nivel de autonomía en la toma de decisiones [3]:

En esta clasificación se van otorgando valores al nivel de autonomía en la toma de decisiones (ACL, del inglés *Autonomous Control Levels*), los cuales van desde 0 para las aeronaves que dependen totalmente de un piloto remoto para la toma de decisiones hasta alcanzar un valor de 2 para aeronaves que son capaces de tomar toda clase de decisiones de forma autónoma.

Tabla 2.2 – Clasificación según el nivel de autonomía en la toma de decisiones de la aeronave

ACL	Nivel de autonomía
0	Dirigidos por control remoto: Un piloto certificado controla la aeronave de forma remota, pudiendo existir o no línea de vista directa con la aeronave.
1	Dirigidos por control remoto (semiautónomos): Un operador entrenado programa la aeronave con el plan de vuelo mediante comandos de alto nivel ( <i>waypoints</i> , altura de vuelo, velocidad, etc.), para luego monitorizar la ejecución de dicho plan. En este caso, la aeronave lleva a cabo el vuelo de forma autónoma, pero la toma de decisiones se delega a un operador humano.
2	Totalmente autónomo: En este caso, las aeronaves poseen total autonomía en la toma de decisiones, lo que les permite una interacción inteligente con su entorno. Cabe destacar que actualmente la implementación de este nivel de autonomía aún no es permitida por la reglamentación vigente.

### 2.1.1.2 Subsistemas y/o elementos de un UAV

A bordo de los UAVs se encuentran un conjunto de subsistemas y/o elementos que interactúan entre sí para hacer posible el vuelo de la aeronave. A continuación presentaremos una breve descripción de los subsistemas más importantes:

### 1) Subsistema mecánico:

Este subsistema engloba todos los elementos que determinan el comportamiento aerodinámico de la aeronave.

- **Frame:** Este elemento define la estructura aerodinámica, grados de libertad y las maniobras que la aeronave puede realizar. Los frames de UAVs más utilizados son los de ala fija y de ala rotatoria, ver ejemplo en la figura 10. Los materiales más usados para su fabricación son madera, aluminio, fibra de vidrio, plástico y fibra de carbono.
- **Hélices:** son uno de los elementos más importantes para que el UAV logre la sustentación necesaria para volar ya que determinan el empuje de vuelo y el ratio de respuesta. Los principales parámetros que a tomar en cuenta para realizar una buena elección de las hélices son el tipo de material de fabricación, diámetro, número de palas, ángulo de ataque y dirección de rotación.

### 2) Subsistema de propulsión

El subsistema de propulsión está compuesto principalmente por el o los motores, dependiendo del tipo de aeronave, y es el encargado de generar la fuerza de empuje que permite a la aeronave volar. En los UAVs podemos encontrar distintos tipos de motores dependiendo del tipo y de las características del aparato. A continuación presentaremos dos de los motores más usados por UAVs de uso civil:

- **Motores eléctricos:** la eficiencia y fiabilidad de los motores eléctricos ha mejorado significativamente en los últimos años. Por lo tanto, este tipo de motor ha sido una de las claves que explican el boom que hoy en día experimentan los UAVs de uso civil. Los nano y micro UAVs, así como también la gran mayoría de los multirrotores generalmente están equipados con este tipo de motor. La energía necesaria para hacer trabajar un motor eléctrico puede ser obtenida a partir de baterías, pilas

de combustible de hidrógeno o paneles solares. Existen diferentes tipos de motores eléctricos, pero los principales son los del tipo *Brushless* (sin escobillas) y los *Brushed* (con escobillas). Sin embargo, de estos dos tipos actualmente los más usados son los *Brushless* ya que poseen mayor eficiencia, generan menos calor, requieren menos mantenimiento y tienen mayor tiempo de vida en comparación de un motor con escobillas.

- Motores de combustión interna de 2 tiempos: principalmente utilizados en UAVs de ala fija, estos motores presentan una elevada fiabilidad. Los motores de combustión están especialmente pensados para UAVs que requieran una mayor autonomía y velocidad que las obtenidas con los motores eléctricos.

### 3) Fuente de energía

Este elemento es la fuente de donde el UAV extrae la energía necesaria para volar y es determinado por el tipo de propulsión empleado. Para los UAVs con motores eléctricos se utilizan comúnmente baterías de corriente directa, mientras que para los UAVs con motores de combustión se utilizan generalmente combustibles líquidos como por ejemplo la gasolina.

En el caso las baterías a continuación presentaremos los distintos tipos más usados en los UAVs:

- Ni-Cd (baterías de níquel-cadmio): El voltaje de una célula de Ni-Cd es de 1.2 V. Poseen efecto memoria. El efecto memoria se produce cuando cargamos las baterías sin haberlas previamente descargado por completo, se crean unos cristales en el interior de dichas baterías que hace que no se carguen en toda su capacidad. Para prevenir el efecto memoria basta con realizar de vez en cuando una descarga completa. Otro inconveniente es que el cadmio es muy tóxico.

- Ni-MH (baterías de níquel-metal-hidruro): En estas baterías, debido a la alta toxicidad del cadmio, éste se sustituye por hidruros metálicos. Tienen mayor densidad de carga (capacidad), no contienen Cd (menos contaminantes), no poseen efectos de pérdida de capacidad por mal uso (efecto memoria). Soportan un menor número de cargas durante su vida útil que las de Ni-Cd. Tienen una resistencia interna superior lo que las limita para alimentar motores de alta potencia. Aceptan cargas rápidas.
- Li-ion (baterías de iones de litio): La capacidad de una batería de Ion-Litio es aproximadamente el doble de la capacidad de una batería de Níquel-Cadmio. El litio es el metal más ligero que existe por lo que a igualdad de capacidad las baterías resultan mucho más ligeras. El voltaje de una célula de Ion-Litio es de 3.7 V. No poseen efecto memoria. Tienen como desventaja que requieren de un circuito de control que se emplea para limitar el voltaje máximo de cada célula de la batería.
- Li-Po (baterías de polímero de litio): Son una variación de las Baterías Litio-Ion (Li-ion). Tienen una densidad de energía entre 5 y 12 veces las de Ni-Cd o las de NiMH. Son baterías mucho más ligeras y que pueden adoptar cualquier forma. No poseen efecto memoria. El voltaje de cada elemento es de 3.7 V. La desventaja es que necesitan una carga mucho más lenta que las de NiMh. Este tipo de baterías son ideales para alimentar motores muy potentes. Se debe cuidar de que el consumo máximo del motor sea menor que la descarga máxima de la batería para evitar que la vida de ésta se acorte demasiado. Es el tipo de batería más usado actualmente en los UAVs de propulsión eléctrica.

#### 4) Subsistema de control

El subsistema de control tiene la función de evaluar continuamente el estado del UAV y a su vez es el encargado de gestionar las funciones de navegación y control. El

elemento principal de este subsistema recibe el nombre de autopiloto. El autopiloto o piloto automático es considerado el cerebro del UAV, ya que ejerce un control directo sobre su comportamiento. Este elemento es responsable de la estabilización y la navegación de la aeronave, tanto en modo de vuelo automático como en modo de vuelo manual. En la tabla 2.1 se presentan algunos de los autopilotos más populares del mercado.

Tabla 2.3<sup>i</sup> – Autopilotos más populares del mercado

Producto/Fabricante	Sensores	Peso (g)	Software	Plataforma
osFlexPilot/Airware	INS/GPS/AM/MM	300	Propietario	Ala Fija, Ala Rotatoria
osFlexQuad/Airware	INS/GPS/AM/MM	32	Propietario	Multirroto
MP2128/MicroPilot	INS/GPS/BA/AM	24	Propietario	Ala Fija, Ala Rotatoria
MP2028/MicroPilot	INS/GPS/BA/AM	28	Propietario	Ala Fija, Ala Rotatoria
PiccoloNano/CloudCap Technologies	GPS/AM/BA	65	Propietario	Ala Fija, Ala Rotatoria
Piccolo II/CloudCap Technologies	INS/GPS/BA/AM	226	Propietario	Ala Fija, Ala Rotatoria
VECTOR/UAV Navigation	INS/GPS/MM/BA/AM	180	Propietario	Ala Fija
ArduPilot Mega 2.5/3DRobotics	INS/MM/BA/AM	17	Código abierto	Ala Fija, Ala Rotatoria
Pixhawk/3DRobotics	INS/MM/BA/AM	38	Código abierto	Ala Fija, Ala Rotatoria

#### 5) Subsistema de comunicación embarcado

Este subsistema gestiona los enlaces de comunicación entre el autopiloto, la estación de control en tierra y otras fuentes de información como es el caso del módulo de Sistema de Posicionamiento Global, o GPS por sus siglas en inglés. A continuación presentaremos brevemente los elementos de comunicación comúnmente

<sup>i</sup> INS del inglés *inertial navigation system*; AM del inglés *accelerometer*; BA del inglés *baroaltimeter*; MM del inglés *magnetometer*; el peso no incluye la antena GPS.

embarcados en los UAVs para luego hablar con más detalle del sistema de comunicación de todo el sistema UAV.

- **Módulo GPS:** El Sistema de Posicionamiento Global, más conocido con las siglas en inglés GPS, es un Sistema de Navegación por Satélite que permite determinar en todo el mundo la posición de un objeto, con una gran precisión que puede llegar hasta los centímetros, si bien lo habitual son unos pocos metros. El módulo GPS a bordo del UAV es el encargado de recibir las señales de referencia de los satélites geoestacionarios y mediante triangulación determinar y comunicar en tiempo real la ubicación de la aeronave al autopiloto.
- **Radio Control:** Este elemento es el encargado del establecimiento del enlace de comunicación para el control manual de la aeronave desde una emisora de radio.
- **Telemetría:** el módulo de telemetría embarcado en el UAV interconecta la estación de control en tierra con el piloto automático para permitir la monitorización y/o control del estado de la aeronave. Generalmente las frecuencias de operación utilizadas son las 435 o 915 MHz, y para evitar posibles interferencias con otros sistemas se utiliza un protocolo especializado llamado MAVLink para la transferencia de datos. El rango de cobertura de los módulos de telemetría típicos está en el orden de kilómetros.

### **2.1.2 Subsistema de comunicación**

El subsistema de comunicación es uno de los elementos primordiales para la implementación de un sistema de UAVs, ya sea para llevar a cabo las tareas de control y/o monitorización de la misión desde la estación de control en tierra o para permitir el

establecimiento de estrategias cooperativas mediante el intercambio de información entre las aeronaves.

El enlace de comunicación entre la estación de control en tierra y el UAV es uno de los puntos más críticos de un sistema UAV. Este enlace puede subdividirse en dos tipos [7]:

1. Enlace de comunicación con línea de vista o VLOS (por sus siglas en inglés): Según la normativa española vigente en esta materia, se considera un enlace posee línea de vista directa, cuando existe una separación menor o igual a 500 metros de distancia horizontal y hasta 400 pies (122 m) de distancia vertical entre la aeronave y el operador (figura 2.5). Para este tipo de enlace se pueden emplear cualquier tipo de sistema de comunicaciones tanto analógico como digital que opere en las bandas de frecuencia UHF (del inglés *Ultra High Frequency*) y VHF, (del inglés *Very High Frequency*) así como también las tecnologías WiFi y Bluetooth.

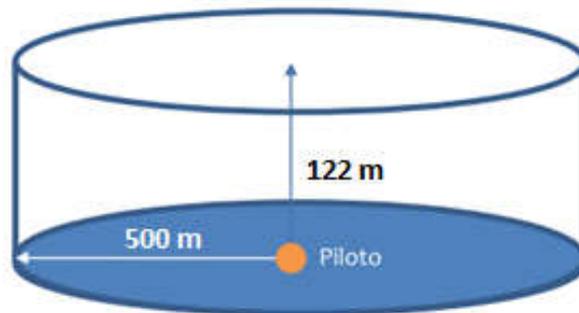


Figura 2.10– Burbuja de enlace de comunicación  
VLOS

2. Enlace de comunicación más allá de la línea de vista o BVLOS (por sus siglas en inglés): Todo enlace que exceda los límites anteriormente presentados es considerado como un enlace BVLOS. El establecimiento de este tipo de enlaces es posible gracias a la implementación de estrategias de enrutamiento y al uso de tecnologías que permiten garantizar la cobertura de comunicación más allá de línea

de vista como pueden ser la GSM (del inglés *Global System for Mobile Communications*), UTMS (del inglés *Universal Mobile Telecommunications System*), HSPA (del inglés *High Speed Packet Access*), LTE (del inglés *Long Term Evolution*), WIMAX (del inglés *Worldwide Interoperability for Microwave Access*) y SATCOM (del inglés *Satellite Communication*).

### 2.1.3 Estación de control en tierra

La estación de control es el centro de control operativo del sistema UAV y está formada por un conjunto de equipos y sistemas que asumen las tareas de planificación y control de la misión (control de vuelo, control de la carga útil), distribución o diseminado de la información a usuarios exteriores o finales.

La estación de tierra incluye paquetes software para la planificación, control y monitorización de misiones, sistemas de comunicaciones y enlaces de datos necesarios para acceder a la plataforma y a los sistemas embarcados en las aeronaves. En ella a parte de los elementos anteriormente mencionados están ubicados también los operadores humanos que dirigen y controlan la misión. La estación de control en tierra puede estar situada en un emplazamiento fijo o en una unidad móvil.

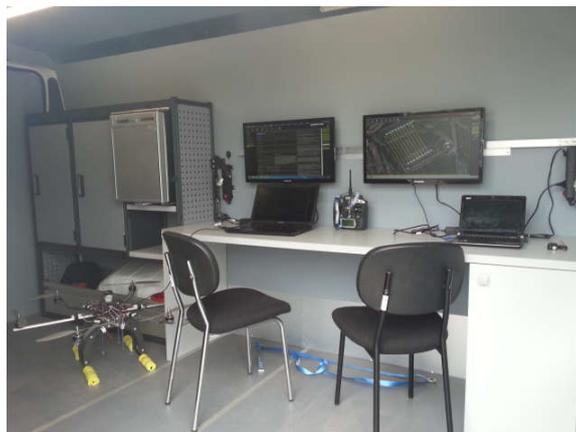


Figura 2.11– Ejemplo de una Estación móvil de control en tierra, instalada en un furgón (Propiedad: Laboratorio de UAVS de la UAB)

De todos los elementos que componen la estación de control, el software de planificación, control y monitorización de misiones es sin duda el corazón de la misma. Para abundar un poco más en tan importante elemento, presentamos a continuación un breve análisis de los paquetes software usados en la estación de control en tierra.

### **2.1.3.1 Software de planificación, control y seguimiento de misiones**

Como se mencionó anteriormente, el software de planificación, control y seguimiento de misiones es el corazón de la estación de control en tierra, ya que ofrece al operador todas las herramientas necesarias para llevar a cabo la planificación *offline* de las misiones de vuelo y a su vez es el encargado de gestionar todo el entramado de comunicaciones entre la estación de control y la o las aeronaves.

La gran mayoría de fabricantes de UAVs ofrecen un paquete software propietario para la planificación y control de misiones. Sin embargo, también existe una gran variedad de soluciones software de código abierto, que poseen la mayoría de funcionalidades que ofrecen los paquetes propietarios y, a su vez, permiten al operador crear nuevas funcionalidades mediante la modificación del código fuente.

Los paquetes software de planificación y control de misiones para UAVs de uso civil más usados actualmente son los siguientes:

- Mission Planner
- QGroundControl
- APM Planner
- Tower
- AndroPilot
- UGCS
- Drone Deploy
- OpenPilot
- DJI Ground Station
- FreeFlight3
- DJI Go
- Hover

A continuación se presenta una comparación, a modo de tabla, de los paquetes software anteriormente citados:

Tabla 2.4 – Comparación de los programas de planificación y control más usados

Software	Protocolo MavLink	Código abierto	Compatibilidad con APM <sup>i</sup> y/o PX4 <sup>ii</sup>	Sistema operativo
Mission Planner	✓	✓	✓	
QGroundControl	✓	✓	✓	
APM Planner	✓	✓	✓	
Tower	✓	✓	Solo con APM	
AndroPilot	✓	✓	✓	
UGCS	✓	✗	✓	
Drone Deploy	✓	✗	✓	
OpenPilot	✓	✓	✗	
DJI Ground Station	✓	✓	✗	
DroidPlanner2	✓	✓	✓	
FreeFlight3	✓	✓	Solo con APM	
DJI Go	✓	✓	✗	
Hover	✓	✓	✗	

### 2.1.4 Carga útil

Se define como carga útil a todo elemento embarcado que dotan al UAV de funcionalidades esenciales para la ejecución de una determinada misión. El tipo de carga útil es determinada

<sup>i</sup> Autopiloto Ardupilot del fabricante 3DRobotics

<sup>ii</sup> Autopiloto Pixhawk del fabricante 3DRobotics

por los objetivos y necesidades de la misión, así como también de la capacidad de carga de la aeronave.

Los elementos que se utilizan como carga útil en un UAV son generalmente de dos tipos diferentes:

- **Sensores** : Los sensores permiten a la aeronave percibir ciertas características del entorno, las cuales dependerán del tipo de sensor empleado. Los sensores electro-ópticos son los más usados como carga útil, ya que actualmente la mayoría de los UAVs vienen equipados con cámaras de espectro visible, las cuales son comúnmente empleadas para aplicaciones relacionadas con el sector audio visual (fotografía, vídeo y cine). Por otro lado, cámaras multiespectrales y térmicas son utilizadas en una gran variedad de aplicaciones, como pueden ser la agricultura de precisión, control y extinción de incendios, entre otras.
- **Actuadores**: Este tipo de carga útil consiste en elementos embarcados que permiten a la aeronave mediante mecanismos actuadores llevar a cabo una determinada acción, como pueden ser la entrega de paquetería, fumigación de cultivos, manipulación de objetos, etc.

## 2.2 Referencias

- [1] Weatherington, D., & Deputy, U. (2005). Unmanned aircraft systems roadmap, 2005-2030. *Deputy, UAV Planning Task Force, OUSD (AT&L)*.
- [2] Van Blyenburgh, P. (2008, February). Unmanned aircraft systems, the current situation. In *EASA Workshop on UAV, EASA*
- [3] Valavanis, K. P., & Vachtsevanos, G. J. (2014). Classification of UAVs. En *Handbook of Unmanned Aerial Vehicles(88)*. : Springer Publishing Company, Incorporated.
- [4] *CIAIAC, Informe Anual 2011*. (2011) Obtenido de <http://docplayer.es/17090596-Ciaiac-informe-anual-2011.html>
- [5] Angelov, P. (2012). *Sense and avoid in UAS: research and applications*. John Wiley & Sons.
- [6] Garcia, P. C., Lozano, R., & Dzul, A. E. (2006). *Modelling and control of mini-flying machines*. Springer Science & Business Media.
- [7] Okcu, H. (2016). Operational Requirements of Unmanned Aircraft Systems Data Link and Communication Systems. *Journal of Advances in Computer Networks*, 4(1).

## Capítulo 3 Entorno de simulación multi-UAV

### 3.1 Introducción

Las misiones llevadas a cabo por sistemas multi-UAV suelen implicar un alto nivel de complejidad. Por lo que el desarrollo de algoritmos para este tipo de sistemas requiere realizar pruebas que implican un alto nivel de riesgo para los UAVs.

Una solución comúnmente usada para contrarrestar este riesgo en la fase de desarrollo es la utilización de entornos de simulación que pueden ser implementados tanto mediante técnicas de *Hardware in the Loop* (HiL) como de *Software in the Loop* (SiL). Este tipo de entornos de simulación permiten poner a prueba diferentes aspectos del sistema sin necesidad de llevar a cabo experimentos en entornos reales.

#### 3.1.1 Simulación Hardware in the Loop

La simulación Hardware in the Loop (HiL) es una técnica que se utiliza en el desarrollo y prueba de sistemas con procesos de alto grado de complejidad. En este tipo de simulación los dispositivos *hardware* son conectados a un Simulador en Tiempo Real (RTS) que se encarga de ejecutar y recrear el resto del entorno de operación; el resultado es un entorno fiable de desarrollo para las pruebas más críticas de nuevas funciones.

Este mecanismo de simulación ofrece un método seguro para probar dispositivos físicos bajo diversas condiciones de operación en tiempo real. Se pueden recrear diversos escenarios de contingencia en un ambiente controlado para evaluar el rendimiento del dispositivo bajo prueba, antes de que éste sea conectado al sistema real.

Una de las principales ventajas de la simulación HiL es que la validación de los resultados de control es muy sencilla. En el área del control industrial, las técnicas de simulación Hardware in the Loop pueden reducir significativamente el tiempo requerido para diseñar controladores y pueden aumentar la confiabilidad de los sistemas[1]. Sin embargo, esta solución requiere que el software funcione en una unidad de control electrónico (ECU por sus siglas en inglés) y que los componentes del vehículo (por ejemplo, sensores y cargas) estén físicamente disponibles o sean simulados con precisión.

Las pruebas basadas en la simulación de HiL presentan las siguientes limitaciones:

- Tiempos de ejecución relativamente altos: la duración de las pruebas está condicionada por el tiempo de respuesta de los dispositivos físicos.
- Coste elevado: el uso de dispositivos *hardware* eleva los costes de la configuración de la simulación.
- Escalabilidad limitada: la escalabilidad de este tipo de entornos de simulación está limitada por el tipo de tecnología empleada, el espacio físico disponible y su consumo energético.

### **3.1.2 Simulación Software in the Loop**

A diferencia de la simulación HiL que está más enfocada a las tareas de desarrollo, pruebas y validación de componentes hardware, la técnica de simulación SiL está orientada a las tareas de desarrollo de software.

La técnica de simulación Software in the Loop ha surgido como un enfoque alternativo a las pruebas de software embebido. Básicamente consiste en la interacción de un código fuente compilado con la simulación de un modelo matemático, proporcionando así un entorno práctico y virtual de simulación para el desarrollo y prueba de estrategias de control detalladas para sistemas grandes y complejos.

El empleo de SiL como técnica de simulación facilita mucho el trabajo de desarrollo de software de control, ya que permite al desarrollador tener la posibilidad de probar y modificar de forma directa e iterativa el código fuente. Con esta técnica de simulación es posible probar la parte software de un sistema antes de la inicialización de la fase de prototipado *hardware*, lo que contribuye a disminuir considerablemente los tiempos y los costes de desarrollo.

En el caso de nuestro entorno de simulación multi-UAV hemos seleccionado la técnica de simulación SiL por los beneficios que aporta para el desarrollo de software de control.

### **3.2 Antecedentes**

A pesar del gran número de grupos de investigación que actualmente se encuentran trabajando en el desarrollo de nuevos algoritmos para sistemas multi-UAV, existen muy pocas propuestas de entornos de simulación multi-UAV en la literatura científica.

Una de las primeras propuestas encontradas en la bibliografía se llama MultiUAV2 presentada en [2] y [3]. El paquete de simulación MultiUAV2 es capaz de simular múltiples UAVs que cooperan para cumplir una misión predefinida. Este entorno de simulación está especialmente pensado para aplicaciones multi-UAV del ámbito militar, donde cada una de las aeronaves simuladas son capaces de buscar, reconocer, atacar y verificar la destrucción de objetivos. MultiUAV2 permite a los investigadores desarrollar y analizar algoritmos de control cooperativo. Sin embargo, no puede ser usado como entorno de prueba en la fase de desarrollo del software final que implementará los algoritmos de cooperación.

La mayor parte de los entornos de simulación encontrados en la bibliografía están basados en la técnica de simulación Hardware in the Loop, lo cual hace que sean soluciones poco escalables y con un coste de implementación relativamente alto como se especificó en el apartado 3.1.1. Por otro lado también existen algunos otros simuladores creados con el

objetivo de realizar pruebas de concepto de algoritmos cooperativos, pero que no pueden ser usados para el desarrollo del software que implementará dichos algoritmos.

Goktogan y Sukkarieh proponen en [4] un entorno de simulación multi-UAV empleando la técnica de simulación Hardware in the Loop. Esta propuesta permite llevar a cabo la verificación, validación y acreditación de cada uno de los elementos físicos de la arquitectura del sistema. Sin embargo, al requerir elementos hardware para su implementación, hace que sea un entorno poco escalable.

En [5] se presenta una plataforma de simulación Hardware in the Loop para sistemas multi-UAV especialmente pensado para pequeños helicópteros de pequeñas dimensiones. El funcionamiento de esta plataforma descansa esencialmente en 4 módulos principales, los cuales son: módulo hardware el cual es el encargado de activar los actuadores servo y los sensores de salida de cada uno de los helicópteros; módulo de control de vuelo responsable de ejecutar los algoritmos de control automático; módulo de la estación de control capaz de generar las ordenes y tareas para cada aeronave, así como de supervisar su ejecución mediante una interfaz de usuario que muestra los datos generados por cada aeronave; y módulo software responsable de integrar los 3 módulos anteriores para la ejecución de la simulación Hardware in the Loop.

García y Barnes [6] utilizan el simulador de vuelo comercialmente disponible X-Plane, junto con MATLAB, para crear un entorno de simulación para el estudio de algoritmos de control multi-UAV. Este entorno consiste en un grupo de ordenadores que actúan como UAVs y un ordenador central donde se ejecuta el simulador X-Plane. Esta configuración permite la inserción de UAVs reales en modo Hardware in the Loop.

Happe y Berger presentan en [7] un ambiente de simulación de eventos discretos para explorar y analizar modelos de decisión de gestión de recursos distribuidos y algoritmos para la planificación de trayectorias de búsqueda cooperativa multi-UAV en misiones de vigilancia y reconocimiento.

Wei en [8] presenta un entorno de simulación como prueba de concepto de la plataforma de planificación de misiones basado en DDDAS (Dynamic Data Driven Application System) propuesto. El entorno se implementa como un programa Java multithread, y tanto el controlador central como todos los UAVs están representados por subprocesos. Las comunicaciones entre el controlador central y los UAV se implementan como intercambios de mensajes.

Pujol, Cerquides y Meseguer en [9] proponen un simulador multi-agente para investigar la coordinación en sistemas multi-UAV. Esencialmente se trata de un entorno de simulación orientado a probar estrategias de coordinación para aquellos casos en que existe problemas de comunicación intermitente.

En [10] Odelga se presenta una entorno Hardware in the Loop para simulaciones de sistemas multi-UAV basado en Gazebo, un popular simulador habilitado para trabajar con ROS (Robot Operating System), el cual proporciona la simulación dinámica de uno o más UAVs y la lecturas de sus sensores.

### **3.3 Arquitectura del entorno de simulación propuesto**

Para el desarrollo de esta tesis doctoral hemos creado un entorno de simulación Software in the Loop para sistemas multi-UAV, que consta de dos partes principales (ver figura 3.1). La primera está formada por un grupo de máquinas virtuales que simulan la flota de UAVs mediante la técnica Software in the Loop. Y la segunda es la etapa de planificación, monitorización y control de misiones multi-UAV, construida a partir del software de código abierto *Mission Planner*. Para conseguir las funcionalidades requeridas para esta tesis se ha modificado el código fuente de *Mission Planner* para dotarlo de la capacidad de operar varias aeronaves de forma simultánea.

Esta arquitectura permite el intercambio de información entre todos los elementos del sistema gracias a una red virtual de comunicaciones implementada con la ayuda del software de virtualización de sistemas operativos Oracle VM Virtual Box. Esta característica permite el uso de esta plataforma para el desarrollo y prueba de algoritmos, tanto para sistemas centralizados como distribuidos.

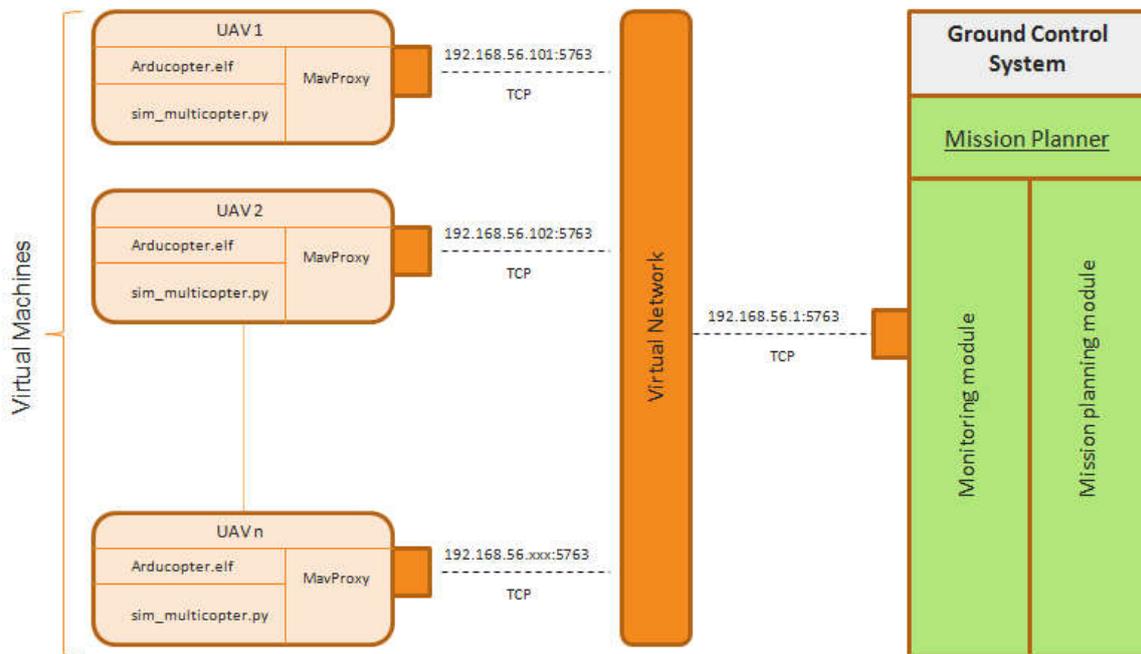


Figura 3.1– Esquema de la arquitectura del entorno de simulación multi-UAV

Cabe destacar que al haber basado su desarrollo en una versión modificada del software *Mission Planner*, nuestra arquitectura también permite crear entornos mixtos de simulación, en los que interactúen UAVs reales con UAVs simulados, lo cual aumenta el potencial de esta plataforma para la realización, en un ambiente controlado, de pruebas de algoritmos con UAVs reales (ver figura 3.2).

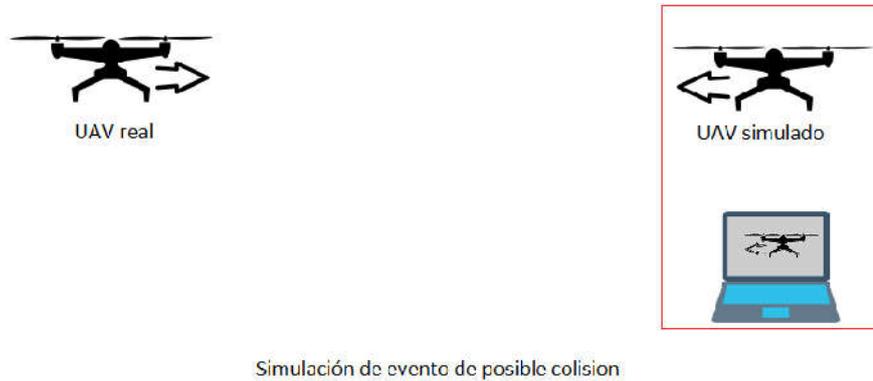


Figura 3.2– Entorno mixto, interacción de UAV real con un UAV simulado

A continuación pasaremos a explicar de forma detallada cada una de las partes de esta arquitectura.

### 3.3.1 Simulación de UAVs

La flota de UAVs es creada partir de un conjunto de máquinas virtuales en las que cada una ejecuta la simulación Software in the Loop de un UAV. Dichas máquinas virtuales emplean Ubuntu 14.04 como sistema operativo y poseen una asignación de memoria RAM de tan solo 250 MB.

El simulador SiL utilizado fue desarrollado por la comunidad de desarrolladores del famoso autopiloto de código abierto, ArduPilot Mega (APM), como una herramienta para el desarrollo y análisis de nuevos firmwares. Este simulador permite la ejecución de código fuente sin tener la necesidad de conectar ningún tipo de dispositivo físico, ya que es capaz de emular todos los módulos y sub-módulos presentes en UAV (drivers de bajo nivel de diferentes sensores, comunicación, unidad de procesamiento, etc.) (ver figura 3.3).

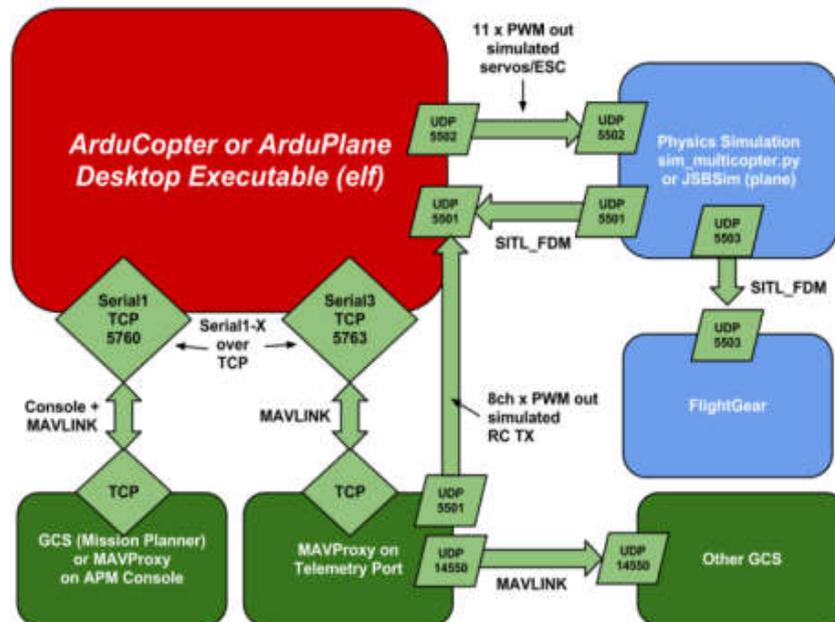


Figura 3.3– Simulador SiL de un único UAV presente en cada máquina virtual

Para conseguirlo los desarrolladores de APM compilaron el código fuente del autopiloto Ardupilot escrito en lenguaje C++, obteniendo con ello un ejecutable nativo con formato elf (*Executable and Linkable Format*) que recrea a la perfección el comportamiento de dicho autopiloto.

El ejecutable puede ser obtenido a partir de la compilación del firmware de un multirrotor, en cuyo caso es nombrado como *ArduCopter.elf*. En el caso de que se obtuviera a partir del firmware de un UAV de ala fija, entonces el fichero recibe el nombre de *ArduPlane.elf*.

Una vez contamos con la recreación virtual del autopiloto en forma de ejecutable, éste es asociado, en el caso de trabajar con el *ArduCopter.elf*, a otro fichero denominado *sim\_multicopter.py* escrito en el lenguaje interpretado Python, el cual se encarga de ejecutar el modelo de la dinámica de vuelo de un multirrotor.

Para el caso de utilizar *ArduPlane.elf*, éste será asociado al popular simulador de código abierto JSBSim, el cual ofrece una gran variedad de distintos modelos de dinámica de vuelo para aviones.

La asociación e interacción entre el ejecutable y el modelo de dinámica de vuelo se lleva a cabo mediante el intercambio de paquetes de datos a través de los puertos UDP 5501 y 5502.

Para las tareas de control, el simulador emplea el programa de código abierto MAVProxy, el cual es un potente software que proporciona las funcionalidades propias de una estación de control en tierra. Este programa se opera mediante comandos y es el responsable de la capa operativa del simulador, permitiendo mediante el protocolo de comunicación MavLink interactuar con la aeronave que está siendo simulada mediante el puerto TCP 5760.

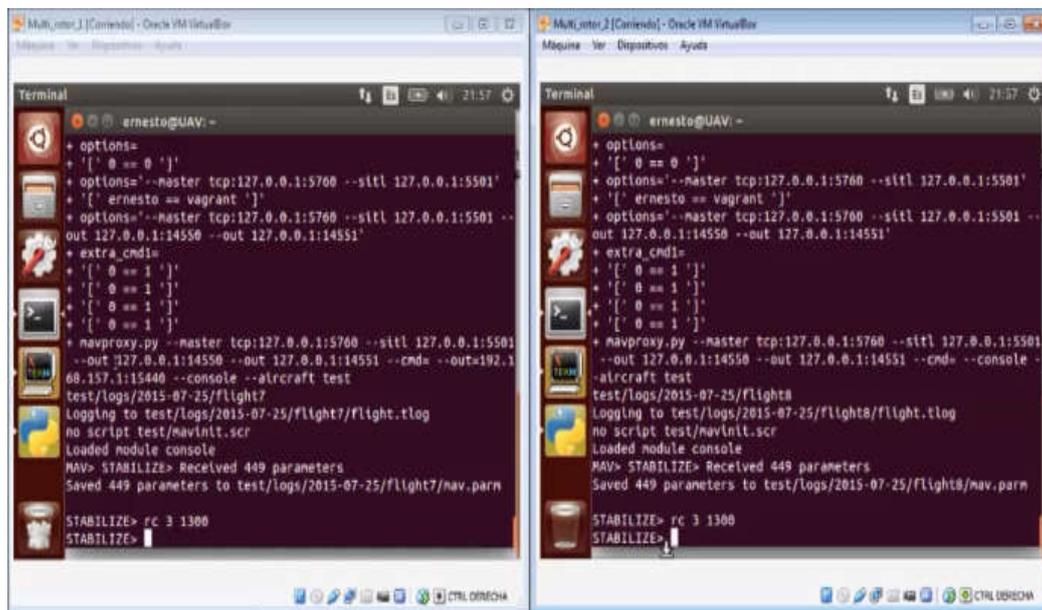


Figura 3.4– El software MAVProxy siendo ejecutado en dos máquinas virtuales distintas

Una de las grandes virtudes que presenta MAVProxy es que, además de ser un completo software de control en tierra, también realiza el papel de puente, permitiendo que a través

suyo se pueda establecer la comunicación entre el UAV simulado y un software de control externo mediante los protocolos TCP y UDP.

Para el desarrollo del entorno de simulación multi-UAV propuesto en esta tesis, la función de puente del software MAVProxy ha resultado de vital importancia, ya que permite, mediante el empleo de un software de control en tierra con función multi-UAV, realizar las tareas de control del conjunto de UAVs simulados desde una única GCS. Esto último constituye la capa centralizada de nuestro entorno de simulación multi-UAV.

MAVProxy también permite modificar, mediante la línea de comandos, los parámetros de configuración de la simulación. Se pueden manipular una gran cantidad de parámetros entre los cuales podemos mencionar:

- Localización
- Tipo de Frame de multirroto: Quad, hexa, X, +, Y, etc.
- Capacidad y número de baterías
- Velocidad y dirección del viento simulado
- Velocidad de crucero
- Velocidad de ascenso y descenso
- Todos los tipos de aceleración experimentados por la aeronave
- Parámetros del controlador PID
- Simular fallos de GPS
- Simular efectos de vibración
- Simular fallos de control remoto (RC)
- Verificar la calibración del Compass

Por otro lado, para agregar la capacidad a cada UAV simulado, tanto de ejecutar algoritmos como de intercambiar información con otros UAVs y con la estación de control, se ha creado un pequeño programa escrito en Python llamado *cooperacion.py*.

El programa *cooperación.py*, ha sido desarrollado empleando DroneKit, la cual es una api de código abierto creada por la empresa 3D Robotics con el objetivo de facilitar el desarrollo e implementación de aplicaciones de control para UAVs.

*cooperación.py* se conecta al UAV simulado a través de MAVProxy mediante el puerto UDP 14550. Una vez se ha establecido la conexión se ejecutan las líneas de código encargadas de establecer los mecanismos de comunicación, de seguimiento de la misión y de toma de decisiones. Este programa deberá ser modificado a la hora de implementar diferentes estrategias de actuación y de toma de decisiones.

Una de las ventajas que aporta la decisión de utilizar la técnica de simulación SiL para la creación de la plataforma de simulación multi-UAV es la flexibilidad de implementación. Las máquinas virtuales podrán estar todas alojadas en el mismo ordenador, o ser distribuidas en ordenadores distintos, todo dependerá de las prestaciones del ordenador usado, así como de los requisitos de ejecución de las máquinas virtuales.

### **3.3.2 Estación de Control en tierra (GCS)**

Para llevar a cabo las funciones de control, planificación y monitorización se ha seleccionado el software de código abierto Mission Planner. Este paquete software es, con mucho, el software de código abierto más utilizado para configurar y planificar misiones para UAVs.

Mission Planner, creado por Michael Osborne, es un paquete de software que nos permite configurar la mayoría de los parámetros del autopiloto APM, lo cual nos permite conseguir afinar al máximo los niveles de estabilidad y de maniobrabilidad. Este software también soporta la calibración de sensores como son la brújula y acelerómetros. A su vez, también nos permite verificar el estado de los motores.

Pero las funciones estrella de este completo software son la monitorización en tiempo real de la misión, control centralizado de la aeronave y la capacidad de planificar misiones con

*waypoints* para su posterior ejecución de forma autónoma. Para esto último utiliza una interfaz amigable y sencilla basada en la api GMaps. La planificación de misiones es muy sencilla: basta con hacer varios clicks en el mapa para seleccionar las distintas ubicaciones que se desea que el UAV visite y posteriormente enviar dicha planificación a la aeronave.

También tenemos la posibilidad de descargar los archivos de log de la misión para su posterior análisis. Este software GCS puede utilizarse como una utilidad de configuración o como un complemento de control dinámico para el vehículo autónomo.

Como este software está diseñado para funcionar con un solo UAV, para el desarrollo de esta tesis ha sido necesario modificar su código fuente para proporcionarle la capacidad de controlar y monitorizar múltiples UAV. Para esto se ha creado un módulo de conexión múltiple que permite la comunicación simultánea con varios UAV usando protocolos TCP / UDP y Serial.

Para establecer la conexión entre Mission Planner y cada UAV, este módulo realiza los siguientes pasos (ver figura 3.5):

1. Recibe la solicitud de conexión.
2. Identifica el protocolo de comunicación a emplear.
3. Se crea una interfaz MAVLink para este UAV (esta interfaz le permite interactuar con el UAV usando el protocolo MAVLink). Para su identificación se asigna un nombre según el orden de conexión, UAV<sub>i</sub>.
4. La interfaz se almacena en una lista llamada "*Comports*".
5. Todos los módulos de Mission Planner (Monitoreo, control y planificación) apuntan a la lista de "*Comports*" para interactuar con cada una de las Interfaces MAVLink almacenadas.

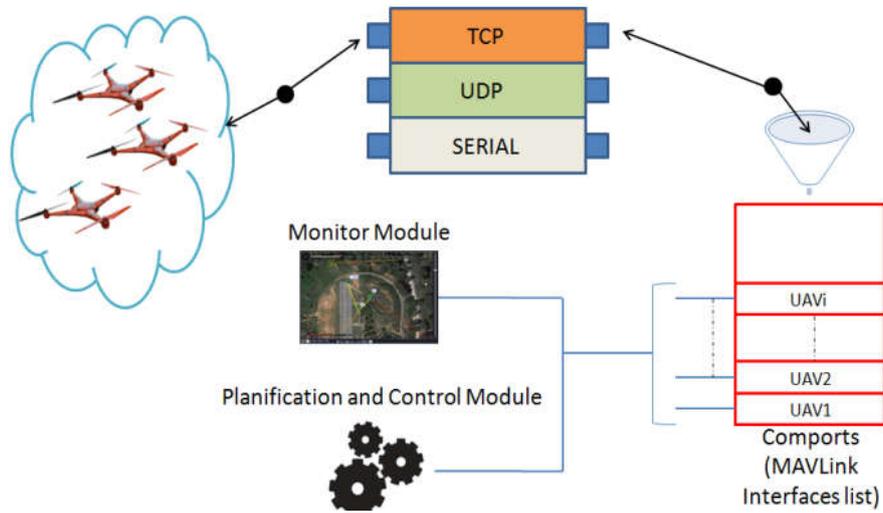


Figura 3.5– Esquema de conexión múltiple

En la figura 3.6 se muestra la interfaz gráfica de usuario del módulo de conexión múltiple.



Figura 3.6– Interfaz gráfica de usuario del módulo de conexión múltiple

Una gran ventaja de usar el software Mission Planner en el entorno de simulación multi-UAV propuesto es el hecho de que todos los algoritmos implementados pueden ser fácilmente probados en misiones reales simplemente reemplazando las máquinas virtuales por UAVs reales.

### 3.3.3 Comunicación

Para implementar la comunicación entre los distintos elementos del entorno de simulación utilizamos el VirtualBox Manager para establecer una red virtual. VirtualBox proporciona varias tarjetas Ethernet PCI virtuales para cada máquina virtual. Para cada tarjeta, se puede seleccionar individualmente el hardware que se va a virtualizar, así como el modo de virtualización en el que la tarjeta virtual va a operar.

En nuestro caso, seleccionamos el adaptador de escritorio Intel PRO / 1000 MT y el modo de red de Host Only para cada UAV simulado, como se muestra en la figura 3.7.

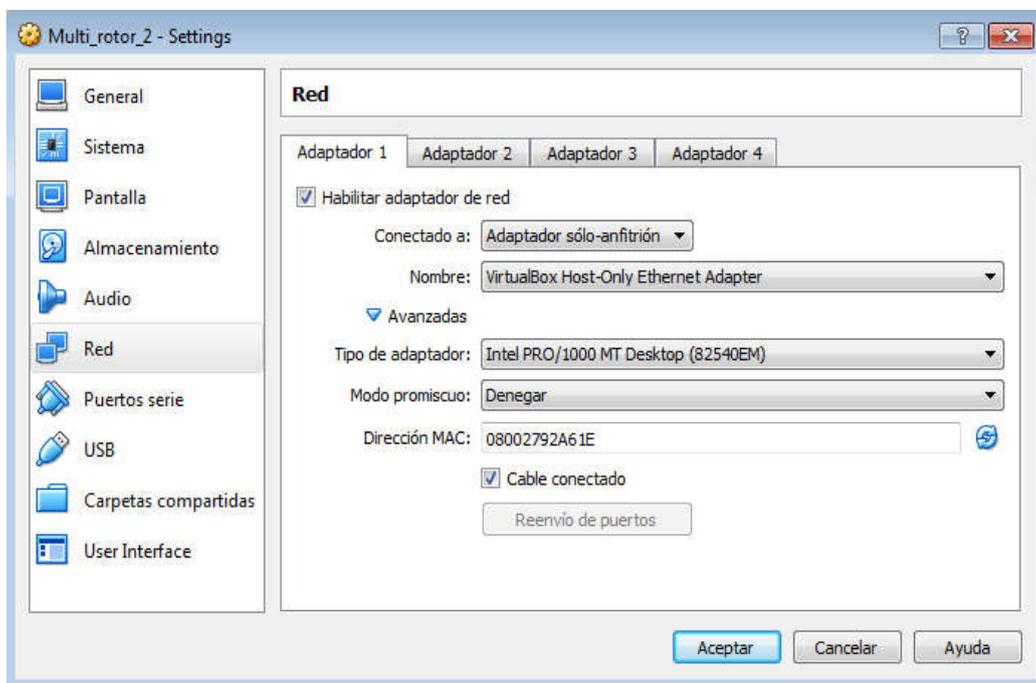


Figura 3.7– Configuración de la tarjeta Ethernet PCI virtual

Esta configuración es particularmente útil para aplicaciones virtuales preconfiguradas, donde múltiples máquinas virtuales están diseñadas para funcionar de forma cooperativa. Para esto en cada host se crea una interfaz de red virtual, proporcionando la capacidad de intercambio de información tanto UAV-UAV como UAV-GCS (ver figura 3.8).

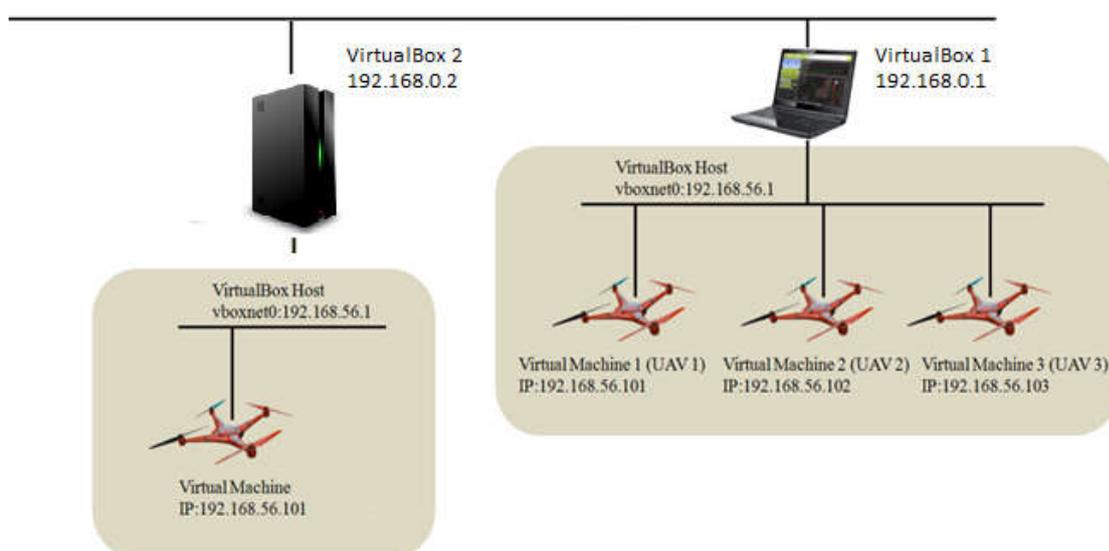


Figura 3.8– Infraestructura de red

Por otra parte, la arquitectura de comunicación empleada se subdivide en dos capas, una centralizada y otra distribuida. A continuación se explica...

### 3.3.3.1 Capa centralizada de comunicación

La capa de comunicación centralizada es la encargada de garantizar el intercambio de paquetes de datos entre la estación de control y los UAVs simulados, ignorando la información generada por los UAVs vecinos. Esta capa garantiza a la estación de control en tierra poseer en todo momento la información global del sistema.

La comunicación centralizada se establece en base a dos elementos fundamentales: en primer lugar la función de puente del software MAVProxy que permite a un software GCS

externo conectarse al UAV simulado, como se comentó en el apartado 1.3.1. Y en segundo lugar, el módulo de conexión múltiple desarrollado y agregado al software Mission Planner.

El intercambio de paquetes se realiza empleando el protocolo de comunicación MAVLink.

### **3.3.3.2 Capa distribuida de comunicación**

La capa de comunicación distribuida es aquella donde se llevan a cabo los intercambios de información tanto UAV-UAV como UAV-GCS, lo cual permite la ejecución de misiones que requieran un alto grado de cooperación entre las partes.

Para su implementación se ha elegido el protocolo UDP, ya que, al no requerir el establecimiento de conexión para el intercambio de paquetes, lo hace ideal para entornos de alta movilidad, como es el caso de los sistemas multi-UAV. Además, brinda la posibilidad enviar mensajes de tipo broadcast.

Para ello, cada UAV, al ejecutar el programa cooperación.py, crean un socket que empleará para comunicarse con toda la red. Lo mismo sucede en la estación de control en tierra al iniciar la ejecución de una misión.

El protocolo que describe el formato de los mensajes deberá ser establecido previamente, tanto en el archivo cooperación.py presente en los UAVs simulados como en el código fuente del programa Mission Planner.

### 3.4 Referencias

- [1] Xue, D., & Chen, Y. (2013). *System simulation techniques with MATLAB and Simulink*. John Wiley & Sons.
- [2] Chandler, S. P. R., & Rasmussen, S. J. (2002). MultiUAV: A Multiple UAV Simulation for Investigation of Cooperative Control. In *Winter Simulation Conference*.
- [3] Rasmussen, S., Mitchell, J., Schulz, C., Schumacher, C., & Chandler, P. (2003). A multiple UAV simulation for researchers. In *AIAA Modeling and Simulation Technologies Conference and Exhibit* (p. 5684).
- [4] Goktogan, A. H., & Sukkarieh, S. (2007, March). Simulation of multi-UAV missions in a real-time distributed hardware-in-the-loop simulator. In *4th International Symposium on Mechatronics and its Applications (ISMA07), Sharjah, UAE*.
- [5] Cai, G., Chen, B. M., Lee, T. H., & Dong, M. (2009). Design and implementation of a hardware-in-the-loop simulation system for small-scale UAV helicopters. *Mechatronics*, 19(7), 1057-1066.
- [6] Garcia, R., & Barnes, L. (2010). Multi-UAV simulator utilizing X-Plane. *Journal of Intelligent & Robotic Systems*, 57(1), 393-406.
- [7] Happe, J., & Berger, J. (2010, July). Couav: a multi-uav cooperative search path planning simulation environment. In *Proceedings of the 2010 Summer Computer Simulation Conference* (pp. 86-93). Society for Computer Simulation International.
- [8] Wei, Y., Blake, M. B., & Madey, G. R. (2013). An operation-time simulation framework for UAV swarm configuration and mission planning. *Procedia Computer Science*, 18, 1949-1958.
- [9] Pujol-Gonzalez, M., Cerquides, J., & Meseguer, P. (2014, May). MAS-planes: a multi-agent simulation environment to investigate decentralised coordination for teams of UAVs. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems* (pp. 1695-1696). International Foundation for Autonomous Agents and Multiagent Systems.

- [10] Odelga, M., Stegagno, P., Bühlhoff, H. H., & Ahmad, A. (2015, November). A Setup for multi-UAV hardware-in-the-loop simulations. In *Research, Education and Development of Unmanned Aerial Systems (RED-UAS), 2015 Workshop on* (pp. 204-210). IEEE.

## Capítulo 4 Arquitectura multi-UAV

### 4.1 Introducción

La arquitectura de un sistema multi-UAV determina y define las características de los procesos de toma de decisiones y de los mecanismos de interacción entre todos los elementos del sistema. Además, la arquitectura delimita el grado de robustez y de escalabilidad del sistema.

Cualquier arquitectura multi-UAV está compuesta principalmente por los siguientes tres módulos:

- Descomposición de la misión y generación de las tareas
- Asignación de tareas
- Planificación de rutas de vuelo

En la primera etapa se inicia el proceso de planificación generando la lista de tareas a ejecutar, luego éstas pasan a la fase de asignación en donde se decide cuál de los UAVs disponibles ejecutará cada una de las tareas, basándose para ello en las características de las mismas y en la capacidad de las aeronaves. Una vez se ha llevado a cabo la asignación de tareas, se generan rutas de vuelo para cada vehículo [1].

Para la clasificación de los distintos tipos de arquitecturas posibles se pueden emplear diferentes enfoques, a continuación presentamos los más utilizados en la bibliografía:

- Clasificación según el grado de centralización del sistema

- Clasificación según la forma de interacción entre UAVs [2]

#### **4.1.1 Clasificación de arquitecturas multi-UAV según el grado de centralización del sistema**

Basándonos en el grado de centralización en la ejecución de los distintos procesos de toma de decisiones, las arquitecturas pueden ser centralizadas, descentralizadas e híbridas.

- Arquitectura Centralizada: En este tipo de arquitectura la estación de control en tierra posee un control total sobre cada una de las funciones del sistema. Toda la información sobre los UAVs o sobre características del entorno es transmitida a la estación de control en tierra la cual es la encargada de llevar a cabo todos los procesos de toma de decisiones. Es decir, las funciones de recepción y procesamiento de datos, planificación de misiones, generación y asignación de tareas, así como la planificación de las rutas de vuelo se llevan a cabo en la estación de control en tierra.

Las principales ventajas de este tipo de arquitectura son su bajo nivel de complejidad y costes de implementación relativamente bajos al requerir UAVs menos especializados.

Sin embargo, los sistemas centralizados poseen un punto único de fallo y de saturación de las comunicaciones, lo cual supone un importante inconveniente en entornos dinámicos o con un elevado número de UAVs. Además, los algoritmos centralizados presentan tiempos de ejecución relativamente elevados.

- Arquitectura Distribuida: En este tipo de arquitecturas no existe un único elemento de control, de modo que todos los UAVs presentan un alto grado de autonomía en el proceso de toma de decisiones.

La arquitectura distribuida puede responder mejor a entornos desconocidos o cambiantes y, por lo general ofrecen un elevado nivel de fiabilidad, flexibilidad, adaptabilidad y robustez, en contraste con los sistemas centralizados.

Esta arquitectura es mucho más robusta ya que las aeronaves no están atadas a las decisiones de elementos externos, y en caso de existir un fallo, tanto en una aeronave como en la estación de control terrestre, el sistema podrá responder adecuadamente. Al no tener que estar constantemente enlazadas con la estación de control en tierra, el sistema podrá explotar mucho más sus capacidades, y cubrir mayor extensión de terreno. Para comunicarse con la estación en tierra las aeronaves emplean técnicas de enrutamiento, en donde el UAV que quiera enviar información a la estación de tierra deberá hacerlo a través de UAVs vecinos que harán el papel de repetidores.

En contra partida, el coste para su implementación es más elevado, ya que es necesario el uso de UAVs altamente especializados. Además, la arquitectura descentralizada requiere, entre otras cosas, de un intercambio mayor de información, tanto UAV-UAV como UAV-GCS, lo cual eleva el nivel de complejidad para el diseño e implementación de los sistemas de comunicación.

- Arquitectura Híbrida: Es el tipo de arquitectura más utilizada para la implementación de sistemas multi-UAV, ya que permite aprovechar las ventajas que ofrecen tanto la arquitectura centralizada como la descentralizada.

En los sistemas híbridos los distintos procesos de toma de decisiones son repartidos entre la estación de control en tierra y los UAVs. La capa centralizada del sistema se ejecuta en la GCS mientras que la capa descentralizada es llevada a cabo por la flota de aeronaves.

Un tipo de arquitectura híbrida es la jerárquica, en la que existen uno o más agentes de control central que organizan los UAVs en grupos, de acuerdo a la configuración maestro-esclavo.

### 4.1.2 Clasificación de arquitecturas multi-UAV según el nivel de interacción entre UAVs

Basándonos en la manera en que las aeronaves interactúan unas con otras, los sistemas multi-UAV pueden clasificarse de la siguiente forma [3]:

- Interacción física: en esta arquitectura los UAVs se encuentran enlazados físicamente, por lo que los movimientos de cada aeronave deben estar coordinados con las maniobras de las otras. Un ejemplo de este tipo de arquitectura lo vemos en aquellos sistemas multi-UAV empleados para levantar elementos pesados. En esta arquitectura, debido a que el número de aeronaves por lo general es bajo, se puede emplear una arquitectura de control tanto centralizada como distribuida.

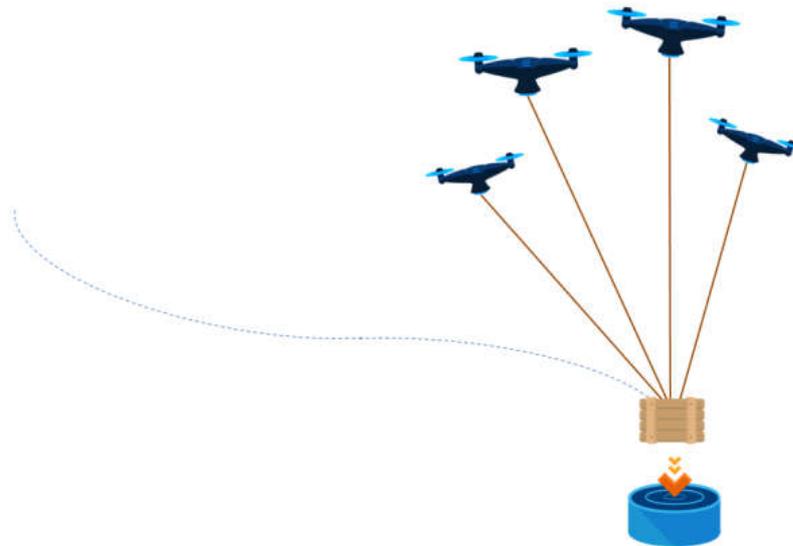


Figura 4.1– Ejemplo de interacción física entre UAVs

- Formación: En este caso no existe un enlace físico entre las aeronaves, pero los movimientos deben estar coordinados para mantener una formación de vuelo determinada. En este tipo de sistemas generalmente se emplean arquitecturas de control descentralizadas.

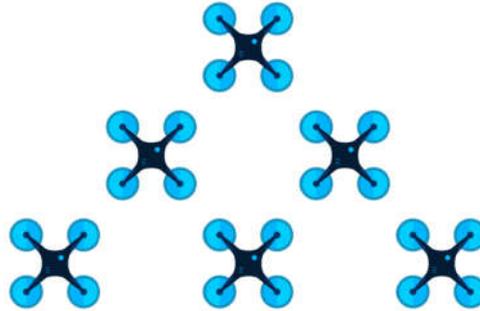


Figura 4.2– Vuelo en formación

- Enjambre (Swarm): En estos sistemas nos encontramos un conjunto homogéneo de aeronaves que interactúan entre sí con un esquema de vuelo dinámico. La escalabilidad es un aspecto crucial debido a que se pueden emplear un número relativamente elevado de aeronaves, por lo que es obligatorio emplear una arquitectura de control distribuida.



Figura 4.3– Enjambre de UAVs

- Cooperación intencionada: Los UAVs del equipo se mueven de acuerdo a las trayectorias definidas por las tareas individuales que son asignadas para la consecución de un objetivo global. En este caso, los problemas de asignación y planificación de tareas, y la resolución de conflictos deben resolverse teniendo en

cuenta las características de la misión global a ejecutar y las capacidades de los distintos UAVs involucrados.

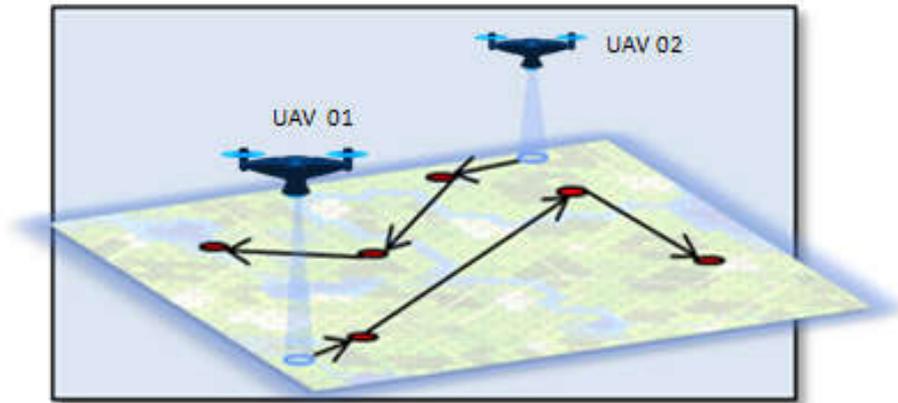


Figura 4.4– Ejemplo de cooperación intencionada entre UAVs

## 4.2 Antecedentes

En la bibliografía se encuentran una gran variedad de propuestas de arquitecturas de coordinación y control para sistemas multi-UAV, las cuales pueden ser desde soluciones totalmente centralizadas hasta aquellas en las que todo el proceso de toma de decisiones se lleva a cabo de forma distribuida por el segmento aire del sistema.

Gracias a la democratización de las tecnologías aplicables en el campo de las aeronaves no tripuladas, fruto del decremento de costes de producción experimentado en los últimos años, el desarrollo e implementación de nuevas arquitecturas multi-UAV ha mostrado un notable aumento.

En la actualidad existen muchos grupos de investigación en el mundo enfocados en el desarrollo de nuevas arquitecturas multi-UAV para diferentes tipos de aplicaciones. Como cada tipo de aplicación requiere de un nivel diferente de acoplamiento entre las aeronaves,

emplearemos el enfoque del nivel de interacción entre UAVs para realizar una revisión de las distintas soluciones presentes en la literatura.

Existen pocas propuestas de arquitecturas multi-UAV enfocadas en aplicaciones que implican la interacción física entre aeronaves, como por ejemplo el transporte de carga debido su elevado nivel de complejidad y riesgo. En [5] Iván Maza presenta una arquitectura distribuida para el transporte de cargas con varias aeronaves. En esta propuesta, cada UAV posee una capa deliberativa que se ocupa de la toma de decisiones de alto nivel de forma distribuida y una capa de ejecución encargada de la realización de las tareas. Para la toma de decisiones distribuidas, se realizan interacciones entre las capas deliberativas de los diferentes UAVs. La interfaz software permite al usuario especificar las misiones y tareas a ejecutar por la plataforma, así como supervisar el estado de ejecución de las tareas y el estado de los diferentes subsistemas. Gianluca Antonelli propone en [6] una arquitectura centralizada de control multi-UAV en el que las aeronaves están equipadas con brazos robóticos con el fin de realizar de forma sincronizada las tareas de traslado de objetos.

A diferencia del caso anterior, existen una amplia cantidad de propuestas para aplicaciones que requieren la realización de vuelos en formación [7][8][9]. Una arquitectura híbrida es propuesta en [9] por Hattenberger, en la cual la estación de control de tierra se encarga de forma centralizada de generar la ruta de vuelo, mientras que un módulo embarcado en los UAVs permite crear de forma distribuida la formación de vuelo. En [11] se presenta una estructura distribuida de control de formación basada en el software del autopiloto de código abierto Paparazzi. Una arquitectura centralizada de control de vuelos en formación de bajo coste es presentada en [12] para detectar radiación nuclear.

Las propuestas de arquitecturas basadas en Swarm han experimentado un auténtico boom en los últimos años, y todo ello como consecuencia al abaratamiento de los precios de los micro UAVs. Por ejemplo en [13] se presenta una estrategia basada en el uso de un Algoritmo Genético para tratar ciertos parámetros de control de un Swarm, tales como las

probabilidades de transición de un UAV a través de diferentes modos, la tasa de decaimiento de feromonas y los parámetros de atracción de feromonas de un UAV para búsqueda y supresión de misiones de defensa aérea enemiga. En [14] el mecanismo swarm para el reconocimiento automático de un target se basa en el mecanismo de comunicación de entre insectos mediante feromonas como elemento de refuerzo positivo para encontrar el rastro al objetivo.

### **4.3 Arquitectura de control y cooperación propuesta**

En el presente trabajo se propone una arquitectura multi-UAV híbrida de cooperación intencionada que permite la realización de forma eficiente tanto de los procesos de control como de los de toma de decisiones. Nuestra propuesta está especialmente pensada para aplicaciones que requieren la inspección de amplias zonas mediante un grupo de UAVs en el menor tiempo posible, como, por ejemplo, aplicaciones de búsqueda y rescate, vigilancia, levantamientos topográficos, búsqueda de *Hot Spots*, etc.

Esta arquitectura está compuesta por dos capas, la primera de ellas posee una estructura centralizada, mientras que la estructura de la segunda capa es distribuida. En la figura 5 se muestra un esquema general de la arquitectura propuesta.

La capa centralizada está implementada en su totalidad en la estación de control en tierra, y en ella se lleva a cabo todo el proceso de generación y asignación de tareas, así como la planificación de las rutas gracias al software de planificación Mission Planner. Todo ello teniendo en cuenta las características de los distintos vehículos disponibles en el sistema multi-UAV. Una vez han sido asignadas cada una de las tareas y las distintas rutas de vuelo, se procede a la ejecución de la misión. Esta capa es la encargada de establecer el enlace los UAVs y el operador del sistema, gracias a los módulos de monitorización y control. La

comunicación entre la estación de control en tierra y los UAVs se realiza mediante la implementación del protocolo de comunicaciones MAVLink.

En el momento en que los UAVs inician la ejecución de sus respectivos planes de vuelo, la capa distribuida toma el control del sistema. En esta etapa los UAVs inician el intercambio de información entre ellos acerca del desarrollo de la misión, lo que permitirá a cada UAV disponer del conocimiento necesario para iniciar un proceso de reasignación de tareas y de replanificación de rutas en caso de presentarse algún cambio que ponga en riesgo el cumplimiento de los objetivos de la misión.

La intercomunicación de datos se hace posible gracias a la implementación de una red ad-hoc entre los UAVs, la cual recibe el nombre de *Flying Ad-Hoc Network* (FANET) [3].

A continuación pasaremos a explicar cada uno de los procesos y funciones de la arquitectura propuesta.

# Arquitectura multi-UAV

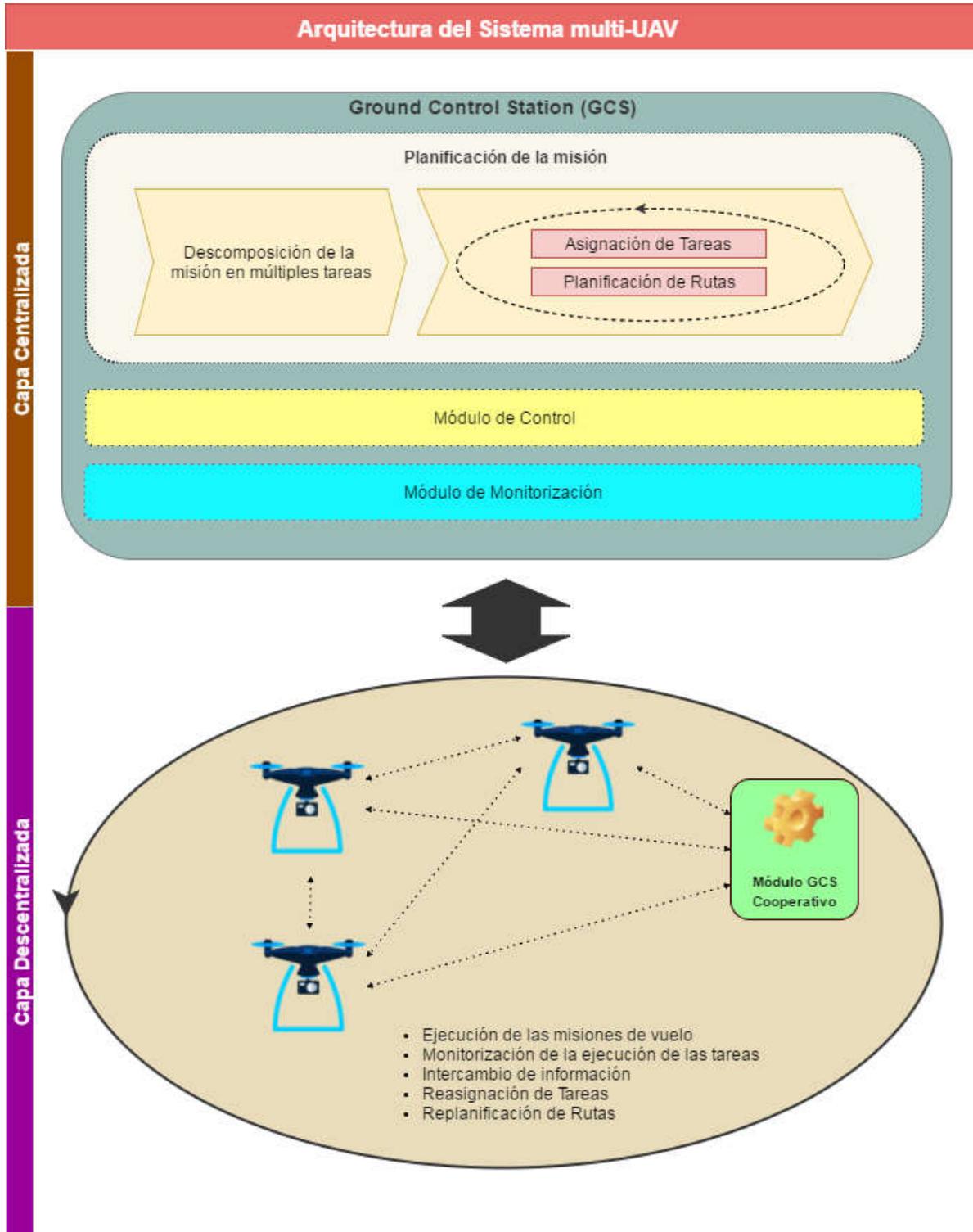


Figura 4.5– Esquema general de la arquitectura multi-UAV propuesta

### 4.3.1 Capa centralizada

En esta capa tienen lugar todos los procesos que requieren la participación del operador, como la monitorización, control y planificación de las misiones. Cada una de estas funciones permite al operador tener la capacidad de manipular y controlar en cualquier momento cada uno de los parámetros y elementos del sistema.

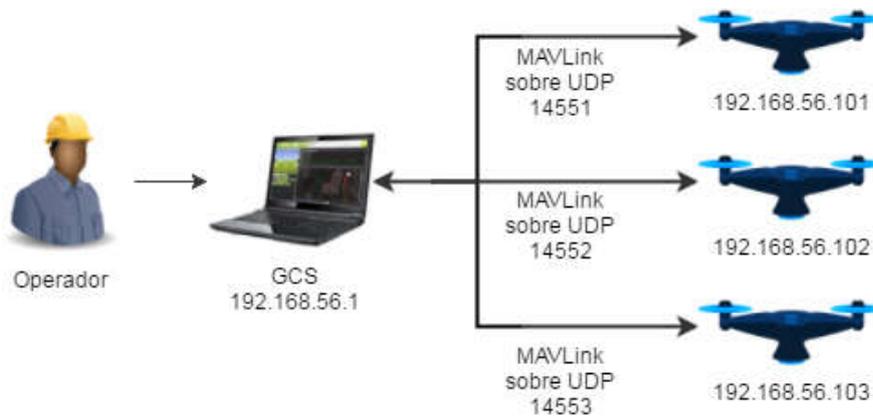


Figura 4.6– Esquema de la capa centralizada

#### 4.3.1.1 Módulo de Planificación

El primer paso del proceso de planificación es el establecimiento de la comunicación entre la estación de control en tierra y los distintos UAVs, luego el operador deberá indicar cada uno de los datos necesarios para la realización de la misión. Para ello el operador, con la ayuda de la interfaz del GCS, demarcará el perímetro de la zona que se desea cubrir.

Posteriormente el operador iniciará el proceso de planificación automatizado presionando un botón destinado a tal fin. Una vez hecho esto, el sistema pedirá al operador el ingreso para cada aeronave de algunos parámetros de vuelo, campo visual de la cámara y el tiempo de vuelo máximo (ver figura 4.7).

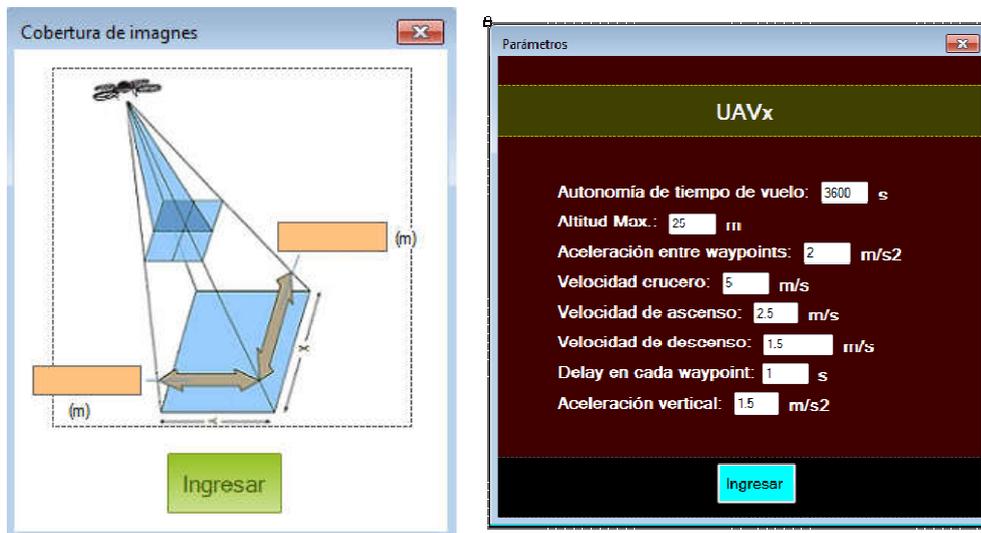


Figura 4.7– Interfaz de usuario para ingresar los parámetros de vuelo

Tras ingresar los datos anteriormente mencionados, el algoritmo de asignación de tareas y planificación de rutas descompone la zona objetivo en subregiones que luego son asignadas a cada UAV. A su vez estas subregiones son descompuestas en un conjunto de waypoints que definen la ruta de vuelo que deberá ejecutar para realizar el trabajo de cobertura. Los detalles detrás del funcionamiento de dicho algoritmo se abordarán en el capítulo 5.

Tras finalizar el proceso de asignación de tareas y planificación de rutas, cada UAV recibe un plan de vuelo que deberá ejecutar de forma autónoma. En la figura 4.8 se muestra cómo se lleva cabo el envío del plan de vuelo a cada UAV.

## Arquitectura multi-UAV

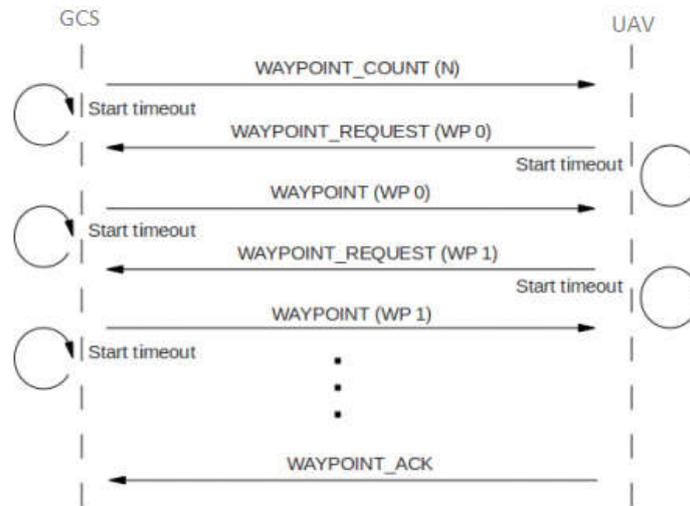


Figura 4.8- Escritura de waypoints

En la figura 4.9 se presenta, a modo de diagrama de flujo, la secuencia de pasos que se llevan a cabo en el proceso de planificación de la misión.

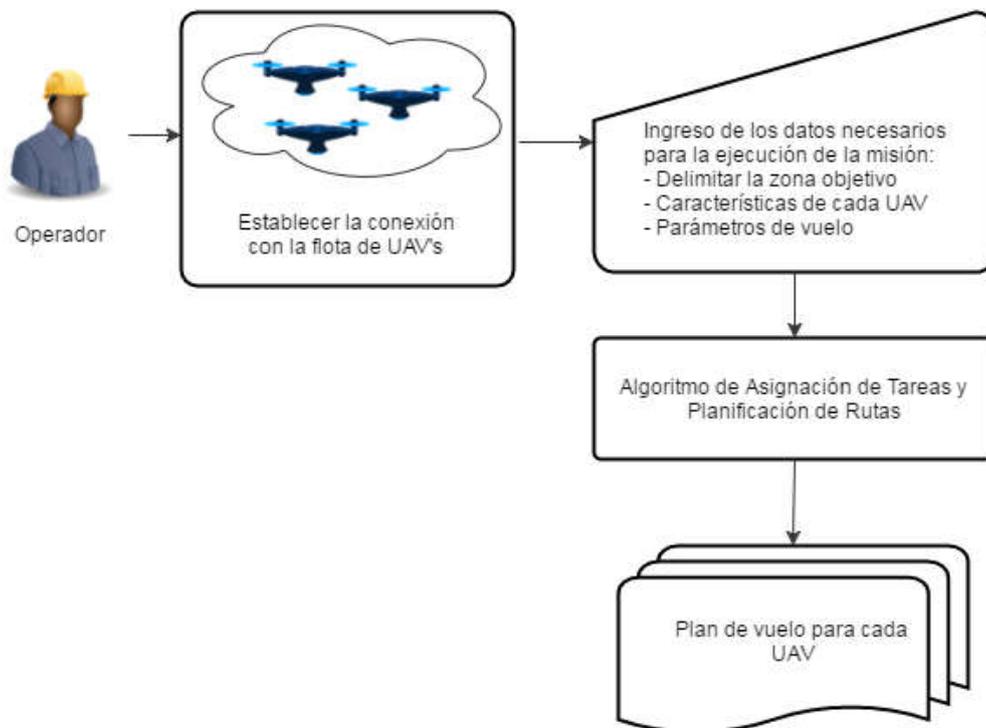


Figura 4.9 - Diagrama de flujo del proceso de planificación de misiones

#### 4.3.1.2 Módulo de Control

El módulo de control de la interfaz del software de planificación Mission Planner está compuesto por un conjunto de funciones que permiten al operador tener en todo momento el control total del sistema. Dichas funciones son:

- Establecer o modificar el modo de vuelo de cada aeronave: El operador podrá cambiar el modo de vuelo configurado para cada aeronave, pudiendo elegir entre *Auto*, *Stabilize*, *Guided*, *Loiter*, *Land* y *RTL (Return To Launch)*.
  - *Auto Mode*: Este modo permite que sean los propios UAVs los responsables de seguir una ruta preestablecida de forma autónoma. La implantación de esta arquitectura está basada en el uso este modo de vuelo.
  - *Stabilize Mode*: Con este modo uno o varios operadores pueden tomar el control manual de una o varias aeronaves, dejando sólo en manos del autopiloto la estabilización del cabeceo y el alabeo. En esta arquitectura este modo de vuelo está reservado sólo para los casos en que sea necesario tomar el control de la aeronave por la ocurrencia de un fallo en el autopiloto.
  - *Guided Mode*: Este modo se destina al caso en que el operador quisiera enviar un determinado UAV a unas coordenadas específicas.
  - *Loiter Mode*: Este modo permite paralizar la ejecución de la misión, manteniendo a uno o a todos los UAVs en su posición actual.,
  - *Land Mode*: Este modo permite hacer aterrizar a una o a todas las aeronaves sin importar en qué momento de la ejecución de la misión se encuentren.
  - *RTL*: Con este modo el operador podrá indicar a una o a todas las aeronaves que deben de regresar al punto de lanzamiento.
- Planificación de rutas de vuelo: El operador podrá, en todo momento, planificar nuevas rutas de vuelo o modificar las rutas preestablecidas, ya sea para el conjunto de aeronaves o sólo para un número determinado de ellas.

- Reiniciar la misión: Desde el interfaz de usuario el operador podrá reiniciar la misión en cualquier instante durante el desarrollo de la misma.
- Armar y desarmar motores: Para que la flota de UAVs inicie la ejecución de la misión deben primeramente armar motores. Por motivos de seguridad siempre será el operador el encargado de llevar a cabo esta tarea.
- Modificar parámetros de vuelo: Cuando lo estime oportuno, el operador podrá modificar cada uno de los parámetros de todos los UAVs o de un número de ellos en particular.

Por otro lado, el operador siempre será el encargado de iniciar la ejecución de la misión una vez se ha asignado una ruta de vuelo a cada UAV. Para esto se llevarán a cabo los pasos mostrados en la figura 4.10



Figura 4.10 - Pasos para iniciar la ejecución de la misión

### 4.3.1.3 Módulo de Monitorización

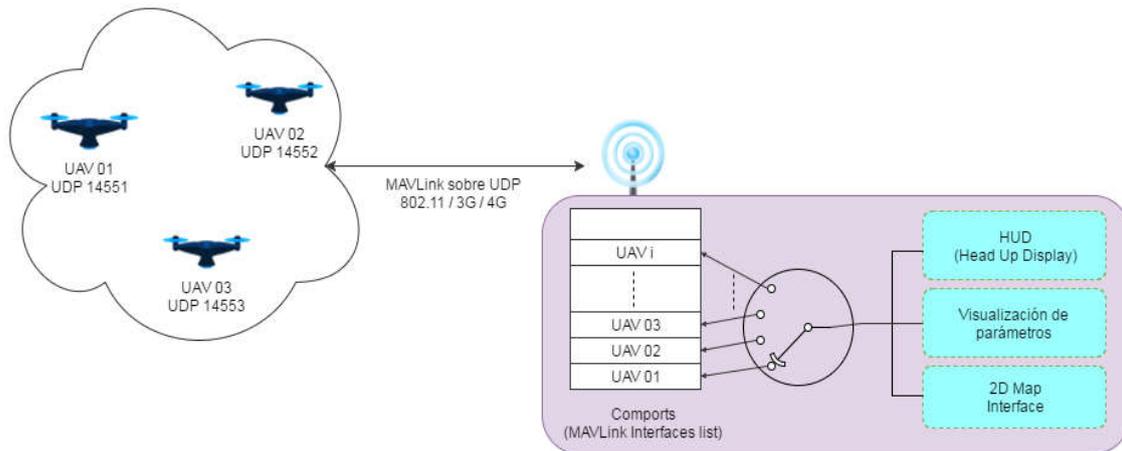


Figura 4.11 - Esquema general del módulo de monitorización

Desde el momento en que se establece la conexión entre los UAVs y la estación de control en tierra, el módulo de monitorización comienza a operar. Gracias a éste, el operador podrá conocer cada uno de los detalles del desarrollo de la misión.

Como se puede observar en la figura 4.11, el módulo de monitorización está dividido en tres partes:

- a) *2D Map Interface*: La interfaz de usuario ofrece un mapa 2D que permite visualizar las acciones y movimientos de cada uno de los UAVs en tiempo real.
- b) *Visualización de Parámetros*: El operador, gracias a esta funcionalidad, puede monitorizar en tiempo real cómo evolucionan cada uno de los parámetros de cada uno de los UAVs. A diferencia de la *2D Map Interface*, el operador no puede observar los parámetros de todos los UAVs de forma simultánea, por lo que deberá primeramente seleccionar la aeronave de la cual desea monitorizar sus parámetros.
- c) *HUD (Head UPDisplay)*: Es una interfaz que brinda información acerca de la evolución de algunos de los principales parámetros de un determinado UAV (ver

figura 4.12). Al igual que la función anterior, para usar el HUD para un determinado UAV, primero deberemos seleccionarlo.

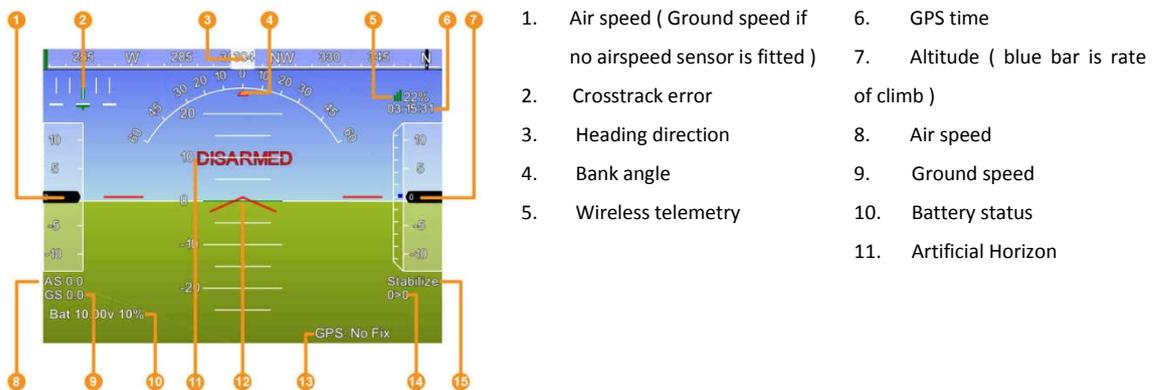


Figura 4.12 - HUD (Head UP Display)

### 4.3.2 Capa Distribuida

Cuando realizamos la asignación de tareas en sistemas multi-UAV en la fase de planificación de la misión, lo hacemos tomando en cuenta las características y capacidades de las aeronaves, y asumiendo ciertas condiciones del entorno donde se llevará a cabo la misión. Sin embargo, en ciertas ocasiones ocurren imprevistos que pueden poner en peligro la viabilidad de la misión.

Entre los distintos incidentes que pueden ocurrir durante el transcurso de una misión se encuentran los siguientes ejemplos:

- 1) Problemas de hardware en alguna de las aeronaves (fallo de motor, fallo del receptor GPS, baja autonomía de la batería, fallo del autopiloto, etc.)
- 2) Cambios atmosféricos (aumento de la velocidad del viento, tormentas, etc.)
- 3) Colisión de alguna aeronave con obstáculos móviles inesperados (aves, aeronaves ajenas a la misión, etc.)

Algunos de estos incidentes pueden dar al traste con toda la misión, en particular aquellos que afectan de forma simultánea a toda la flota de UAVs. Por ejemplo, si en el lugar en el que se desarrolla la misión se desata de forma repentina una fuerte tormenta, la mejor solución será abortar la misión y esperar a que pase el mal tiempo.

Sin embargo, para la gran mayoría de incidentes podemos llevar a cabo estrategias que permitan sortear o minimizar el daño que estos causan a la ejecución de la misión.

La capa distribuida de la arquitectura multi-UAV propuesta en el presente trabajo implementa una estrategia cooperativa de redistribución de las tareas y de replanificación de rutas, con el fin de atenuar los efectos negativos causados por eventos imprevistos.

La estrategia cooperativa de redistribución de tareas propuesta en esta tesis está basada en el sistema de subastas secuenciales de un solo ítem (*Sequential Single-item Auction, SSA*). Dicha estrategia se explicara ampliamente en el capítulo 6.

Como podemos ver en la figura 4.5, los actores principales de esta capa son la flota de UAVs y un módulo de la estación de control en tierra denominado Módulo GCS Cooperativo.

Con el inicio de la ejecución de la misión comienza la interacción de forma descentralizada entre todos los elementos del sistema. Para ello se ha establecido una FANET basada en la tecnología WiFi 802.11n. En caso de que se desee mayor rango de cobertura se podría implementar una red mallada basada en la tecnología 3G/4G.

Gracias al intercambio periódico de información, cada UAV y la GCS conocen cómo evoluciona la misión, pudiendo así detectar rápidamente cualquier imprevisto que altere la normal ejecución de cada una de las tareas.

El papel del módulo GCS Cooperativo es servir de soporte a la flota de UAVs en sus procesos de toma de decisión. Durante el transcurso normal de la misión, dicho módulo se mantiene observando de forma continua y automática el estado de cada uno de los UAVs y sólo

interviene en el proceso de redistribución de tareas en caso de ser necesario. En el capítulo 6 se explicará con más detalle la función de este módulo.

Por otra parte, desde el momento del despegue, los UAVs comienzan a realizar una serie de procesos que tienen como fin evaluar y determinar si es necesario realizar una redistribución de las tareas.

Dichos procesos son los siguientes:

- Intercambio de información: mediante esta función cada UAV puede tanto enviar como recibir información al resto de elementos del sistema, referente a la evolución en la realización de las tareas o al establecimiento de un proceso de redistribución de tareas.
- Evaluación de la ejecución de las tareas: Mediante este proceso, el UAV es capaz de detectar la aparición de problemas que le impidan completar de forma correcta las tareas que le fueron asignadas.
- Redistribución de tareas y replanificación de rutas: Al detectarse un problema que impida a un determinado UAV completar todas sus tareas, se inicia la estrategia de redistribución de tareas de forma descentralizada mediante mecanismos de subastas. Dicha estrategia implica un proceso de replanificación de ruta que se lleva a cabo en el propio UAV de forma local.

Cada uno de los procesos anteriormente mencionados forma parte de un programa escrito en Python llamado *Cooperacion.py* el cual se ejecuta en el computador embarcado en cada UAV. Dicho programa se conecta al autopiloto de la aeronave a través de MAVProxy mediante el puerto UDP 14550.

*Cooperacion.py* es el encargado de permitir el intercambio de mensajes entre las distintas aeronaves gracias a la creación de un Socket UDP cliente - servidor.

La arquitectura UAV embarcada puede observarse en el esquema de la figura 4.13.

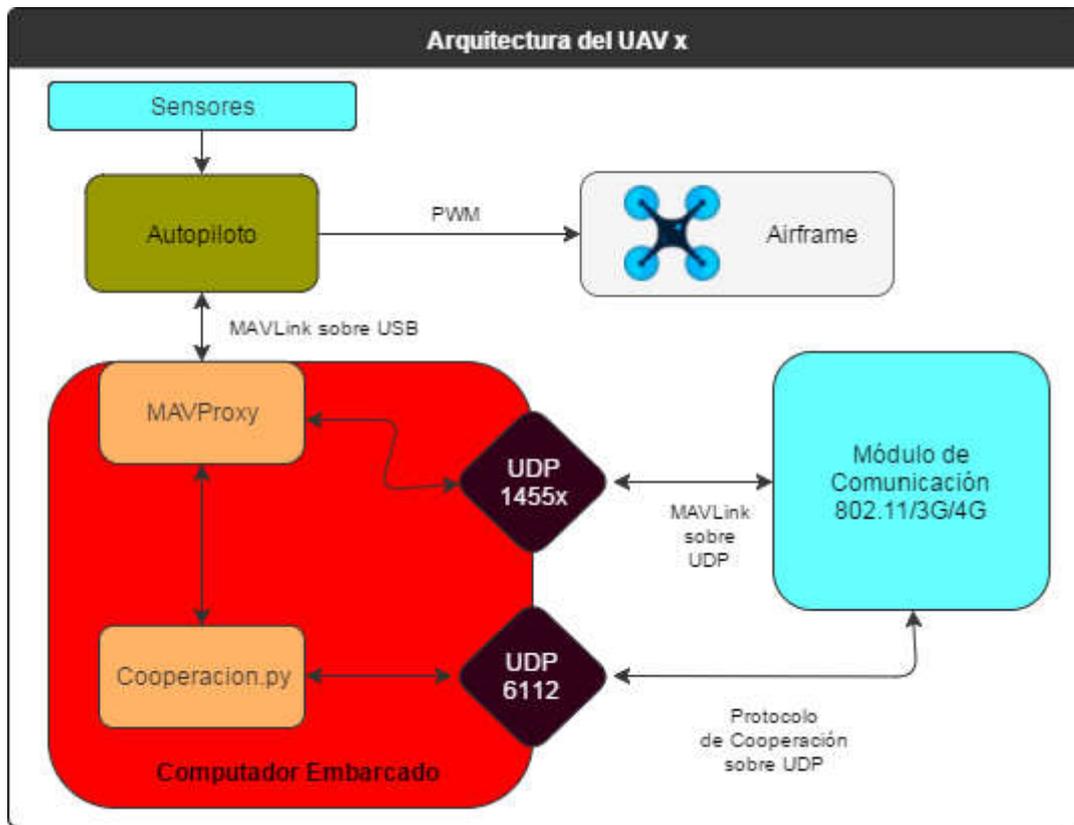


Figura 4.13 - Arquitectura UAV embarcada

Los elementos principales encontrados a bordo de las aeronaves son:

- Computador embarcado: Cada UAV posee un mini ordenador Raspberry Pi v3.0, el cual posee las siguientes características técnicas:
  - Procesador de 64 bits, a 1,2 GHz con cuatro núcleos ARMv8
  - 802.11n Wireless LAN
  - Bluetooth 4.1
  - Bluetooth LowEnergy (BLE)
  - 4 puertos USB
  - 40 pines GPIO (*General Purpose Input/Output*)

En este computador se ejecutan los programas *MAVProxy* y *Cooperacion*.

- Autopiloto: es el encargado de controlar el guiado autónomo de la aeronave. Todas las aeronaves de nuestro sistema multi-UAV están equipadas con un Ardupilot Mega v2.6, el cual está conectado al computador embarcado mediante el puerto USB.
- Sensores: son los elementos que generan los inputs para el autopiloto.

## 4.4 Referencias

- [1] Bethke, B., Valenti, M., & How, J. P. (2008). UAV task assignment. *IEEE Robotics & Automation Magazine*, 15(1).
- [2] Maza, I., Ollero, A., Casado, E., & Scarlatti, D. (2015). Classification of Multi-UAV Architectures. In *Handbook of Unmanned Aerial Vehicles* (pp. 953-975). Springer Netherlands.
- [3] Maza, I., Ollero, A., Casado, E., & Scarlatti, D. (2015). Classification of Multi-UAV Architectures. In *Handbook of Unmanned Aerial Vehicles* (pp. 953-975). Springer Netherlands.
- [4] Bekmezci, I., Sahingoz, O. K., & Temel, Ş. (2013). Flying ad-hoc networks (FANETs): A survey. *Ad Hoc Networks*, 11(3), 1254-1270.
- [5] Maza, I., Kondak, K., Bernard, M., & Ollero, A. (2010). Multi-UAV cooperation and control for load transportation and deployment. *Journal of Intelligent and Robotic Systems*, 57(1-4), 417.
- [6] Antonelli, G., Baizid, K., Caccavale, F., Giglio, G., Muscio, G., & Pierri, F. (2014). Control software architecture for cooperative multiple unmanned aerial vehicle-manipulator systems. *Journal of Software Engineering for Robotics*, 5, 1-12.
- [7] Chen, Y. Q., & Wang, Z. (2005, August). Formation control: a review and a new consideration. In *Intelligent Robots and Systems, 2005. (IROS 2005). 2005 IEEE/RSJ International Conference on* (pp. 3181-3186). IEEE.
- [8] Ren, W., & Beard, R. (2004). Decentralized scheme for spacecraft formation flying via the virtual structure approach. *Journal of Guidance, Control, and Dynamics*, 27(1), 73-82.
- [9] Giulietti, F., Innocenti, M., & Pollini, L. (2001). Formation flight control-A behavioral approach. In *AIAA Guidance, Navigation, and Control Conference and Exhibit* (p. 4239). Giulietti, F., Innocenti, M., & Pollini, L. (2001). Formation flight control-A

- behavioral approach. In *AIAA Guidance, Navigation, and Control Conference and Exhibit* (p. 4239).
- [10] Hattenberger, Gautier, RachidAlami, and Simon Lacroix. "Planning and control for unmanned air vehicle formation flight." *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*. IEEE, 2006.
- [11] Di, L., Chao, H., Han, J., & Chen, Y. (2011, January). Cognitive multi-UAV formation flight: principle, low-cost UAV testbed, controller tuning and experiments. In *ASME 2011 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference* (pp. 917-927). American Society of MechanicalEngineers.
- [12] Han, J., Xu, Y., Di, L., &Chen, Y. (2013). Low-cost multi-UAV technologies for contour mapping of nuclear radiation field. *Journal of Intelligent&RoboticSystems*, 1-10.
- [13] Gaudiano, P., Bonabeau, E., &Shargel, B. (2005, June). Evolving behaviors for a swarm of unmanned air vehicles. In *SwarmIntelligenceSymposium, 2005. SIS 2005. Proceedings 2005 IEEE* (pp. 317-324). IEEE.
- [14] Dasgupta, P. (2008). A multiagent swarming system for distributed automatic target recognition using unmanned aerial vehicles. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 38(3), 549-563.

# Capítulo 5 Planificación de misiones para aplicaciones de cobertura de área con sistemas multi-UAV

## 5.1 Introducción

En los últimos años se ha evidenciado un notable crecimiento del empleo de Sistemas Multi-UAV (Unmanned Aerial Vehicles) en aplicaciones civiles, debido a que éstos permiten minimizar el tiempo de ejecución de las tareas, así como elevar sustancialmente el grado de eficiencia y seguridad a la hora de realizar misiones relativamente complejas. Para la Una de las misiones que se enmarca en la mayoría de las aplicaciones civiles es la de cobertura de un área determinada. Este tipo de misión tiene como objetivo realizar un vuelo que permita cubrir toda el área en el menor tiempo posible. El problema que se plantea, por tanto, es la planificación de una ruta de vuelo que permita cumplir dicho objetivo.

Algunas de las aplicaciones cuyo objetivo principal es la cobertura de un área determinada son las siguientes:

- Agricultura de precisión
- Búsqueda y rescate
- Control de incendios y búsqueda de *Hot Spots*
- Vigilancia
- Reconocimiento y toma de datos en zonas de desastre
- Levantamiento de mapas

- Localización de recursos naturales
- Inspección de obra civil

En la presente tesis nos enfocaremos en este tipo de misión, tomando como ejemplo la aplicación de búsqueda de puntos calientes o *hot spots* con múltiples UAVs tras la extinción de un incendio forestal. Esto se debe a que la realización de este trabajo de investigación se enmarca, en buena parte, en el proyecto “*FIRE Guided Unmanned Aircrafts and Resources Distribution (FireGUARD)*” (TIN2011 – 29494 – C03), específicamente en la fase que comprende la integración de aeronaves no tripuladas en las tareas de prevención y control de incendios forestales.

El problema que se presenta en este tipo de misiones guarda relación con el *covering salesman problem*, una variante del problema del agente viajero, donde en vez de necesitar visitar cada ciudad, los clientes del vendedor especifican un barrio en el que están dispuestos a reunirse con éste [1]. Con la única diferencia de que el problema de cobertura de un área determinada equivaldría a tener que visitar todos los barrios de la ciudad asignada.

En la mayoría de las propuestas presentes en la bibliografía para la planificación de la misión en aplicaciones que requieran la cobertura de un área objetivo por sistemas multi-UAV, el proceso se lleva a cabo en dos etapas: primero se descompone el área en tantas subregiones como UAVs disponibles, y posteriormente se realiza la planificación de las rutas de vuelo de los UAVs asignados a cada una de estas subregiones.

El método de descomposición más empleado en las soluciones de planificación de misión para este tipo de aplicaciones es el basado en la estrategia de descomposición en celdas, bien sea de forma implícita o explícita. El objetivo que se persigue con dicha estrategia es subdividir la región en pequeñas celdas de fácil cobertura, lo cual permite una estructuración del problema más conveniente de cara al proceso de planificación de rutas.

La descomposición en celdas se puede dividir en dos tipos diferentes:

- Descomposición Exacta: el espacio libre  $X$  es representado por un conjunto de celdas no solapadas cuya unión es exactamente  $X$  (el espacio libre se divide enteramente en celdas). Este método de descomposición es completo ya que siempre será capaz de encontrar la solución si esta existe.

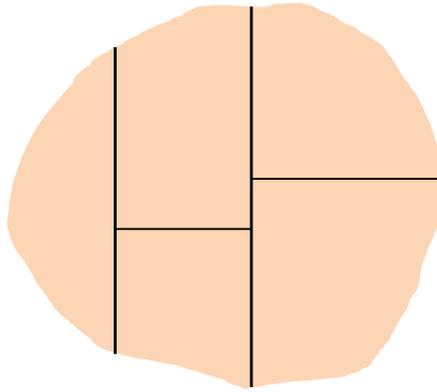


Figura 5.1 - Ejemplo de descomposición en celdas exacta

- Descomposición Aproximada: el espacio libre  $X$  es representado por un conjunto de celdas no solapadas cuya unión está incluida en  $X$  (el conjunto de todas las celdas no corresponde exactamente al total del espacio libre). Se trata de un método no completo pero que permite un cálculo más rápido de las trayectorias.

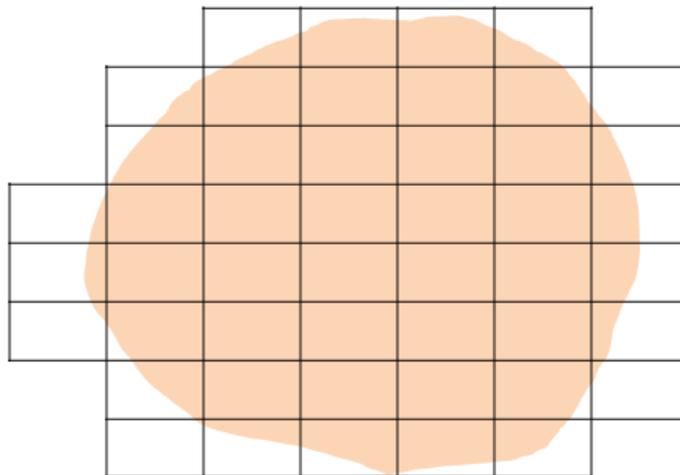


Figura 5.2 - Ejemplo de descomposición en celdas aproximada

Los tres elementos más importantes a tener en cuenta a la hora de abordar un problema de cobertura son los siguientes:

- **Región objetivo:** el área objetivo a cubrir deberá estar delimitada de forma clara. Estas deberán poder ser representadas como áreas poligonales, pudiendo ser convexas o cóncavas.
- **UAV:** se deben analizar las características y capacidades de cada una de las aeronaves que se emplearán para la cobertura.
- **Sensores:** la capacidad de cobertura que posee un UAV en cada pasada está directamente relacionada con las características del sensor empleado, por lo que se deberá prestar especial atención a este último a la hora del diseño del sistema multi-UAV.

Los procesos de planificación de rutas para la cobertura de un área objetivo deberán dar como resultado una secuencia de waypoints que permita barrer al 100% la región asignada.

Sin embargo, existen diferentes tipos de patrones de movimiento para llevar a cabo el recorrido a través de la zona objetivo. A continuación presentamos los más empleados:

- 1) *Back and forth:* Es una de los patrones de movimiento más intuitivos y sencillos. En esta técnica de movimiento los vehículos describen trayectorias de zigzag a modo de serpiente. Es especialmente empleada en misiones de coberturas de áreas en las que no existen obstáculos o zonas no accesibles. Cuando la dirección del recorrido coincide con la dirección óptima de barrido, este patrón permite minimizar el número de giros a realizar en el recorrido, lo cual lo convierte en una excelente opción para la cobertura de área.

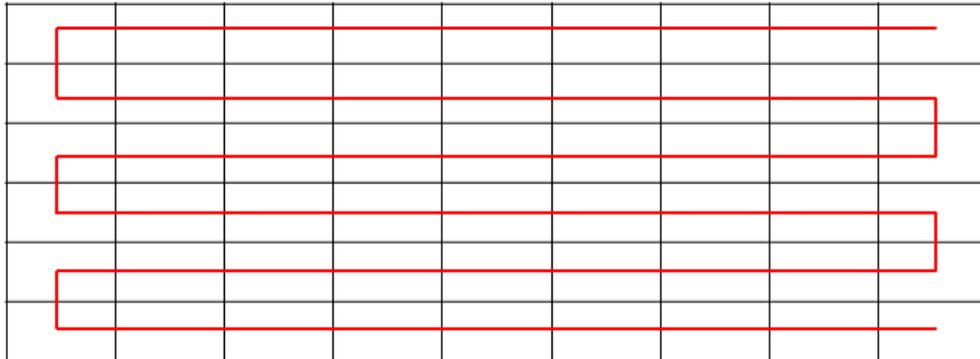


Figura 5.3 - Patrón de movimiento *Back and forth*

- 2) *Spiral-like*: La trayectoria que describe este patrón de movimiento se asemeja a una espiral, con la distinción de que posee lados y ángulos rectos. Este patrón de movimiento no ofrece la misma eficiencia que la del patrón *Back and forth*.

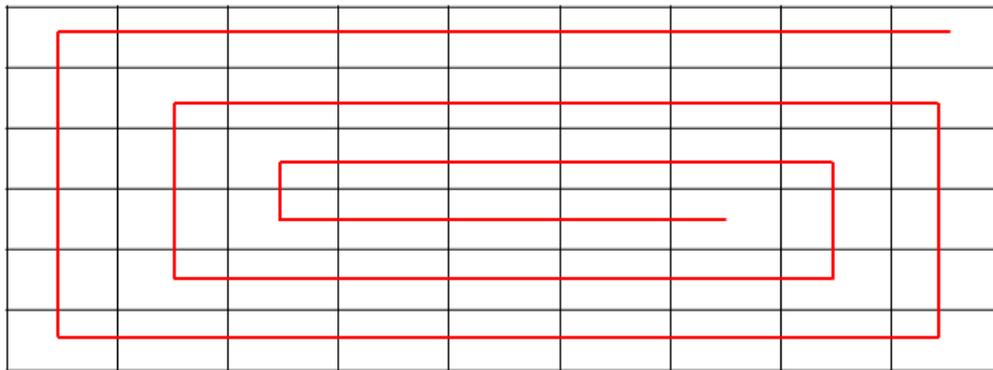


Figura 5.4 - Patrón de movimiento *Spiral-like*

## 5.2 Antecedentes

En la literatura podemos encontrar varias estrategias para solucionar el problema de cobertura de un área determinada.

Los sistemas multi-UAV son una excelente opción para aquellas aplicaciones en las que se busca cubrir una zona, y es que las soluciones basadas en un único UAV pueden resultar ineficientes, en términos de tiempo y disponibilidad. Por ejemplo, en [2] los autores presentan una solución para cubrir un área objetivo con un único UAV. En dicho trabajo se

muestra que debido a la autonomía limitada del vehículo, éste debe realizar varios vuelos para poder cubrir toda el área, lo que eleva considerablemente el tiempo de la misión. En [3] se presenta una solución especialmente pensada para la agricultura de precisión y basada en el uso de una única aeronave. Al igual que en [2], en [3] los autores en [4] abordan el problema de cobertura de área desde la perspectiva de un único UAV, proponiendo un nuevo algoritmo, basado en la técnica de descomposición celular de Boustrophedon, para calcular una ruta de cobertura eficiente para un entornos con obstáculos.

En la mayoría de las propuestas presentes en la bibliografía, la planificación de la misión de cobertura del área objetivo para sistemas Multi-UAV se realiza en dos etapas: primero se descompone el área en tantas subregiones como UAVs disponibles y luego se realiza la planificación de las rutas para los UAVs asignados a cada una de estas subdivisiones. En la referencia [5] se realiza un particionado teniendo en cuenta las capacidades de los UAVs, pero suponiendo que los puntos de lanzamiento de cada UAV se encuentran ubicados de manera que existen entre ellos distancias considerables, lo cual facilita la consecución de tiempo mínimo de vuelo de la misión global, pero aumenta la complejidad de la gestión global de la misión, pues hace más difícil el proceso de lanzamiento de los UAVs, con lo que, en general, no puede garantizarse que el tiempo de desarrollo de la misión global, considerando todos los aspectos que implica (no sólo el tiempo en que los UAVs están volando) sea menor, en relación a una estrategia de punto de lanzamiento único de la flota de UAVs.

En [6] la partición se lleva a cabo a lo largo de la dirección óptima de barrido, lo que permite minimizar el número de giros en las rutas de los UAVs, sin embargo, no se toma en cuenta la ubicación de la base de lanzamiento, por lo que esta solución pierde bastante eficiencia en aquellas situaciones en las que la accesibilidad del terreno permite una única zona de lanzamiento.

Avellar en [7], presenta una estrategia para la cobertura de un área en tiempo mínimo tomando en cuenta el número de operadores humanos disponibles. Si bien en este trabajo las soluciones están orientadas para su ejecución desde una única zona de lanzamiento, no se toma en cuenta el problema de posibles interferencias físicas entre aeronaves, ya que las rutas generadas se entrecruzan.

Los autores de [8] presentan un método de particionado para áreas poligonales cóncavas. Para ello dividen el área en subregiones convexas teniendo como objetivo minimizar la suma de las anchuras de todas las subregiones. Esta propuesta, al igual que la referencia [6], no toma en cuenta la posición de lanzamiento de los UAVs.

También existen propuestas de coberturas de áreas fuertemente acotadas, como son las avenidas y carreteras. En [9] Michael Dille, propone un método de cobertura de redes de carreteras basado en UAVs.

### **5.3 Formulación del problema**

Los incendios forestales, cada año dejan a su paso la desaparición de miles de hectáreas de bosques y reservas naturales. En los primeros nueve meses del año (2017), según datos estadísticos del Ministerio de Agricultura, Alimentación y Medio Ambiente, en España se han registrado 8.756 incendios forestales, afectando a 42.327,60 hectáreas de superficie forestal.

Esta situación pone en peligro vidas humanas, produce cuantiosas pérdidas materiales y provoca daños ambientales muy graves, ya que se producen serios trastornos en la biodiversidad, aumento del nivel de erosión de los suelos y un incremento significativo del efecto invernadero, fruto de las altas emisiones de gases.

Para combatir este terrible problema es necesario adoptar nuevas medidas y estrategias que permitan elevar el nivel de eficiencia en los trabajos de prevención, gestión y control de

incendios. Para ello debemos ser capaces de disponer de información precisa en el menor tiempo posible.

El empleo de vehículos aéreos no tripulados como herramienta en cada una de las etapas (antes, durante y después del incendio [3]) de la lucha contra los incendios forestales representa una excelente vía para elevar los niveles de eficiencia, ya que éstos permiten disponer de información fiable, en tiempo real y a muy bajo coste. En la literatura podemos encontrar varias experiencias del uso de UAVs en el combate de incendios, especialmente para tareas de monitorización de la propagación y el avance del frente de las llamas [4][5][6][7].

Una vez extinguido el incendio se procede a monitorizar y localizar las brasas existentes. Actualmente, una importante cantidad de recursos terrestres son dedicados a la ejecución de tareas posteriores a la extinción del incendio [8]. El personal de tierra tiene que barrer todo el terreno para localizar y sofocar los puntos calientes, lo cual es un procedimiento lento y poco eficiente.

Así, para mejorar la eficiencia de estos procesos, como parte del trabajo de esta tesis se desarrolló un algoritmo de planificación de rutas para la cobertura en tiempo mínimo del área afectada por un incendio mediante un sistema multi-UAV, con el objetivo de localización de puntos calientes.

No obstante, cabe comentar que la formulación del algoritmo, que se presenta a continuación, es general y, en consecuencia, podría aplicarse en cualquier otro escenario que implique cobertura de un área en tiempo mínimo mediante un sistema multi-UAV.

### 5.3.1 Escenario

Como se ha comentado, con la motivación de desarrollar un proceso eficiente de búsqueda de *hot spots* en un escenario de post-incendio, se plantea un problema más general, en los siguientes términos:

Un conjunto de  $n$  UAVs necesita cubrir, en el menor tiempo posible, un área poligonal convexa  $P$ , sin obstáculos en su interior. Todos los UAVs despegarán en el mismo instante de tiempo desde una única zona de lanzamiento  $B$  (ver figura 5.5).

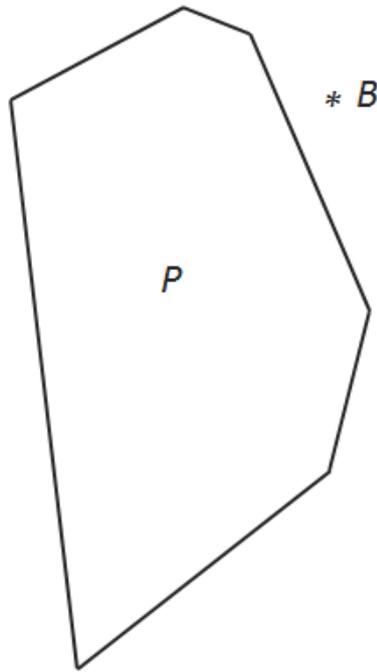


Figura 5.5 - Ejemplo del escenario planteado

Para realizar la cobertura, la región  $P$  será particionada en  $m$  subregiones, tantas como UAVs empleados. Cada  $UAV_i$  recibirá una ruta de vuelo con una duración en segundos  $\tau_i$  para realizar la cobertura de la subregión  $r_j$ . Cada subregión sólo podrá ser sobrevolada por un único UAV. La duración total de la misión la determinará aquel UAV con mayor tiempo de vuelo asignado.

Por tanto, el problema planteado consiste en determinar las rutas de vuelo a asignar a cada UAV de manera que la mayor de las duraciones de dichas rutas de vuelo resulte mínima.

Para dar una descripción formal del problema utilizamos la siguiente notación.

$U = \{UAV_1, \dots, UAV_n\}$	: Conjunto de UAVs disponibles
$T_A = \{A_1, \dots, A_n\}$	: Conjunto que contiene los tiempos de autonomía de vuelo de cada UAV
$A_1 \leq A_2 \dots \leq A_n$	: Los UAVs serán organizados de forma ascendente en función de su autonomía
$R = \{r_1, \dots, r_m\}$	: Conjunto que contiene las subregiones generadas
$tiempo_r(i)$	: Tiempo que tardaría el $UAV_i$ en volar la ruta que tiene asignada.
$W_i = \{w_1, \dots, w_l\}$	: Conjunto de $l$ waypoints que conforman la ruta de barrido asignada al $UAV_i$

Para el cálculo del  $tiempo_r(i)$  de cualquier  $UAV_i$  emplearemos la siguiente ecuación:

$$tiempo_r(i) = t_a + t_d + turns * delay + \sum_{k=0}^{l-1} t_r(\overline{w_k w_{k+1}})$$

Donde  $t_a$  es el tiempo de ascenso,  $t_d$  es el tiempo de descenso,  $turns * delay$  hace referencia a la cantidad de tiempo que tarda la aeronave en cada cambio de dirección y, por último,  $\sum_{k=0}^{l-1} t_r(\overline{w_k w_{k+1}})$  recoge el tiempo que tarda el  $UAV_i$  en volar todos los waypoints asignados ( $w_0$  indica la ubicación de la zona de lanzamiento).

En base a esta formulación, el problema se plantearía matemáticamente como un problema de optimización en los siguientes términos:

$$\text{Minimizar } t_{max} = \max_{i=1..n} \{tiempo_r(i)\} \quad (a)$$

Sujeto a

$$tiempo_r(i) \leq A_i \quad \forall i \in I = \{1, \dots, n\} \quad (b)$$

$$\sum_{i=1}^n x_{ij} \leq 1 \quad \forall j \in J = \{1, \dots, m\} \quad (c)$$

$$x_{ij} \in \{0, 1\}, \quad \forall (ij) \in I \times J \quad (d)$$

La función objetivo (a) expresa la duración mayor de las rutas de vuelo de todos los UAVs, que es lo que se pretende minimizar. La restricción (b) establece que el tiempo necesario para recorrer la subregión  $r$  asignada al  $UAV_i$  nunca podrá ser mayor a la autonomía de vuelo de la aeronave. Por otro lado, (c) establece que cada subregión  $r_j$  sólo podrá ser asignada a un único  $UAV_i$ , para lo cual se emplea la variable de decisión binaria  $x_{ij}$ . Cuando  $x_{ij} = 1$  indica que al  $UAV_i$  se le ha asignado la subregión  $r_j$ .

Se considera que todos los UAVs son multicopteros y deberán realizar el vuelo de cobertura a una misma altura  $h$  y a una misma velocidad de crucero  $v_c$ . A su vez, cada aeronave lleva a bordo una cámara capaz de capturar imágenes en la banda de infrarrojo y transmitir las en tiempo real a la estación de control en tierra. Cada cámara posee un campo visual capaz de cubrir un área de terreno igual a  $C_1 * C_2$  (ver figura 5.6).

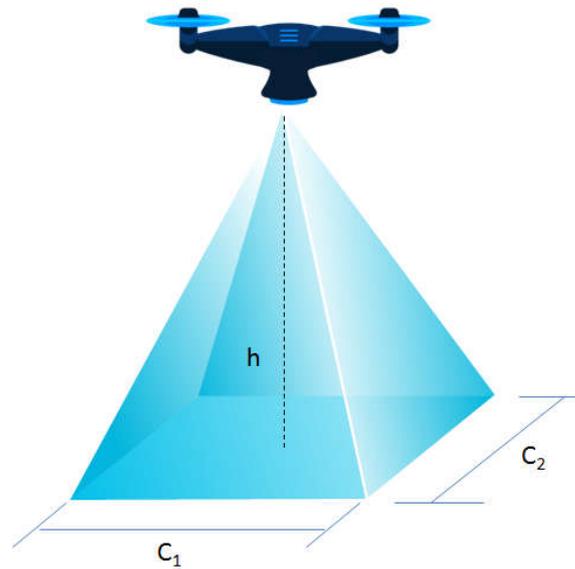


Figura 5.6 - Campo visual de la cámara

### 5.3.2 Enfoque

Como en este escenario disponemos de una única zona de lanzamiento, debemos adoptar una estrategia de particionado que nos permita establecer subregiones accesibles de forma directa desde la estación base y que, a su vez, se encuentren orientadas en la dirección de barrido óptimo, la cual permite realizar la cobertura del área con el mínimo de giros posibles. De este aspecto concreto nos ocupamos más adelante.

## 5.4 Planificación de rutas para la cobertura de un área en tiempo mínimo

Nuestra propuesta, a diferencia de la mayoría de los métodos planteados en la bibliografía, realiza de manera simultánea los procesos de particionado y de planificación de rutas.

Las áreas poligonales convexas pueden ser cubiertas de forma sencilla y eficiente empelando el patrón de movimiento *back and forth*, *presentado con anterioridad*, con



Para determinar la anchura de un área poligonal convexa debemos trazar todos los pares de antípodas del tipo V-L (vértice - lado) o del tipo L-L (lado - lado) que no es más que un caso especial del tipo V-L. El par antipodal que presente la menor distancia entre sus paralelas es el que delimitará la anchura del polígono (ver ejemplo de la figura 5.8). A partir de ahora nos referiremos a las paralelas que delimitan la anchura del polígono como líneas de referencia.

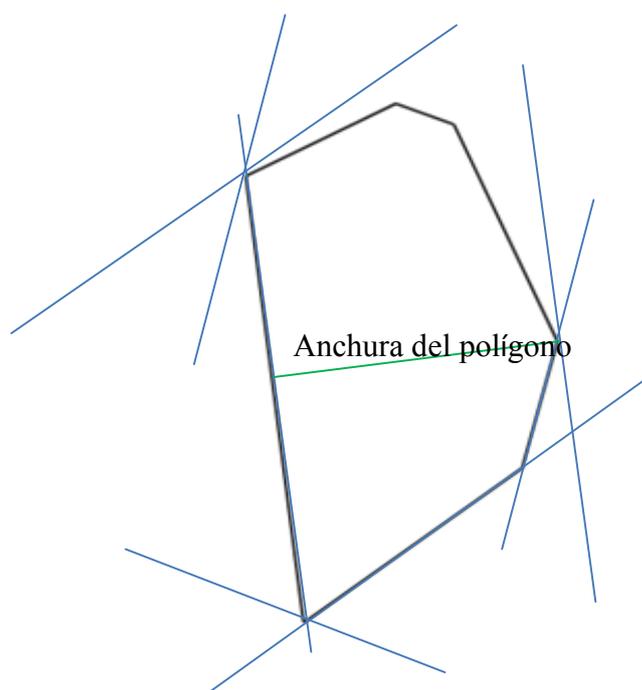


Figura 5.8 - Determinación de la anchura del polígono  $P$  mediante el trazado de los pares antipodales

Si un UAV vuela de forma paralela a las líneas de referencia, estará haciendo un barrido en la dirección óptima, lo que le permitirá minimizar el número de cambios de dirección y, por ende, minimizar el tiempo de vuelo.

Para encontrar el par antipodal que define la dirección óptima de barrido del área poligonal hemos implementado el siguiente algoritmo:

---

**Algoritmo 1** Búsqueda del Par antipodal que indica la dirección óptima de barrido

---

*Input*

---

```

    V = {p1, p2, ..., pn};
Output
    antipodalpair = [pvpl1pl2 ];
    width;
Begin
    width = ∞;
    antipodalpair = [000 ];
    foreach (p in V)
        count = 0;
        pnext = nextpoint(p);
        edge = linesegment(ppnext);
        angle = getlineangle(edge);
        foreach (g in V)
            if ((g ≠ p) ^ (g ≠ pnext))
                parallel = lineparallel(gangle );
                if ((~polinterseccion(puntosparallel) ))
                    count++;
                    disttemp = finddistancetosegment(gedge );
                    if (disttemp < width)
                        width = disttemp
                        antipodalpair[0] = g;
                        antipodalpair[1] = p;
                        antipodalpair[2] = pnext;
                    end if
                end if
            end if
        end foreach
    end foreach
    return antipodalpairwidth ;
End

```

---

### 5.4.2 Particionado del área objetivo y planificación de rutas

Una vez conocida la dirección de vuelo que ofrece el menor número de cambios de dirección, es posible iniciar el proceso de particionado del área en  $m$  subregiones.

En la bibliografía, comúnmente encontramos que el particionado se realiza mediante el cálculo estimado del área que puede volar cada UAV, según sus capacidades. En esta propuesta nosotros nos basaremos principalmente en la autonomía de los UAVs y en una estimación del tiempo que tardaría una aeronave en sobrevolar una determinada subregión mediante el uso de perfiles de velocidad.

Cada subregión puede ser vista como un conjunto de waypoints espaciados a una distancia igual a  $C$ , los mismos son generados mediante un proceso de descomposición aproximada en celdas. Los waypoints están situados en el centro de celdas rectangulares con área igual a  $C_1 * C_2$  (ver figura 6), cuyas dimensiones dependerán del campo visual de la cámara embarcada y de la altura de vuelo. Para facilitar la comprensión de futuras explicaciones, asumiremos que ambos lados de la celda son iguales, es decir:  $C = C_1 = C_2$ .

Al emplear el patrón de movimiento *back and forth* y al establecer una zona única de lanzamiento, necesitamos que cada ruta de vuelo tenga como mínimo un trayecto de ida y vuelta, que pasaremos a denominar tramo básico o  $tr_b$  (ver figura 9). Cada UAV deberá poseer la suficiente autonomía para cubrir una subregión con, al menos,  $1 tr_b$ . La anchura de un tramo básico es igual a  $2C$ .

El número máximo de subregiones que pueden ser generadas dependerá del número de tramos básicos que pueden ser trazados, que, a su vez, depende de la anchura del área poligonal  $P$  como se expresa en la siguiente ecuación:

$$m_{max} = \frac{anchura_P}{2C}$$

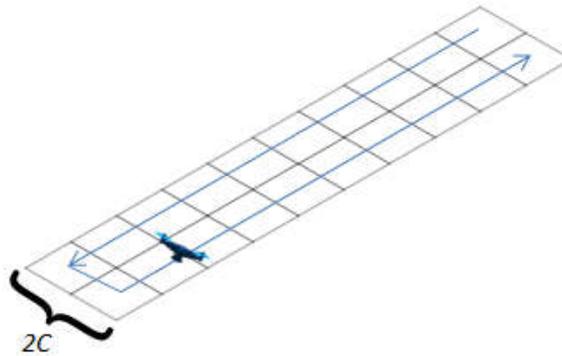


Figura 5.9 - Ejemplo de un tramo básico de ida y vuelta,  $tr_b$

La ecuación anterior también implica una restricción respecto al número de UAVs que pueden ser empleados, ya que, como se indicó en las restricciones presentadas en el apartado 5.3.1, una aeronave sólo podrá ser asignada a una única subregión, por lo que el número de UAVs a emplear nunca podrá ser mayor al número máximo de subregiones.

$$n_{max} \leq m_{max}$$

#### 5.4.2.1 Definición de la zona de lanzamiento

Generalmente, cuando se emplean sistemas multi-UAV para la ejecución de un conjunto determinado de tareas, se define una única zona de lanzamiento y aterrizaje, debido a que es la forma más eficiente para el despliegue operativo del sistema.

En la arquitectura multi-UAV propuesta en este trabajo, el proceso de planificación de la misión se inicia con la definición por parte del operador del área poligonal objetivo, la localización del punto central de la zona de lanzamiento  $B$  y el número de aeronaves a emplear.

Una vez se han indicado las coordenadas del punto central de  $B$  y el número de UAVs  $n$ , el sistema calcula el par antipodal que define la dirección óptima de barrido para luego pedir al operario que especifique la distancia de separación entre los puntos de lanzamiento de cada aeronave. Cuando el operario ingresa dicho dato, el sistema genera  $n$  puntos de

lanzamiento alineados de forma paralela a las líneas que forman el par antipodal anteriormente mencionado (ver figura 5.10).

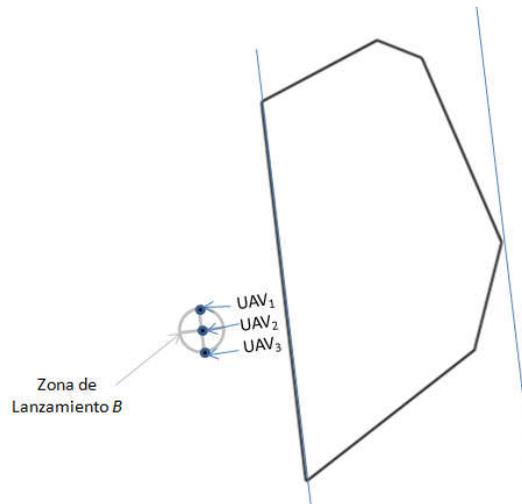


Figura 5.10 - Zona de lanzamiento definida para tres UAVs

El radio de la zona de lanzamiento se establece a partir del número de UAVs ( $n$ ) a emplear y de la separación ( $s$ ) entre los mismos.

$$radio_B = 0,5 * (n - 1) * s$$

Ahora bien, como partimos de la condición de que existe una única zona de lanzamiento, a la hora de realizar el particionado siguiendo el método de la dirección de barrido óptimo se nos presentarán dos casos:

- 1) La estación base está ubicada de tal forma que el  $UAV_i$  puede acceder a la subregión asignada de forma directa, sin necesidad de sobrevolar las regiones asignadas a otros UAVs.
- 2) La ubicación de la estación base dificulta el acceso de forma directa a la región asignada.

En la figura 5.11 se muestra un ejemplo cada uno de los casos anteriormente mencionados.

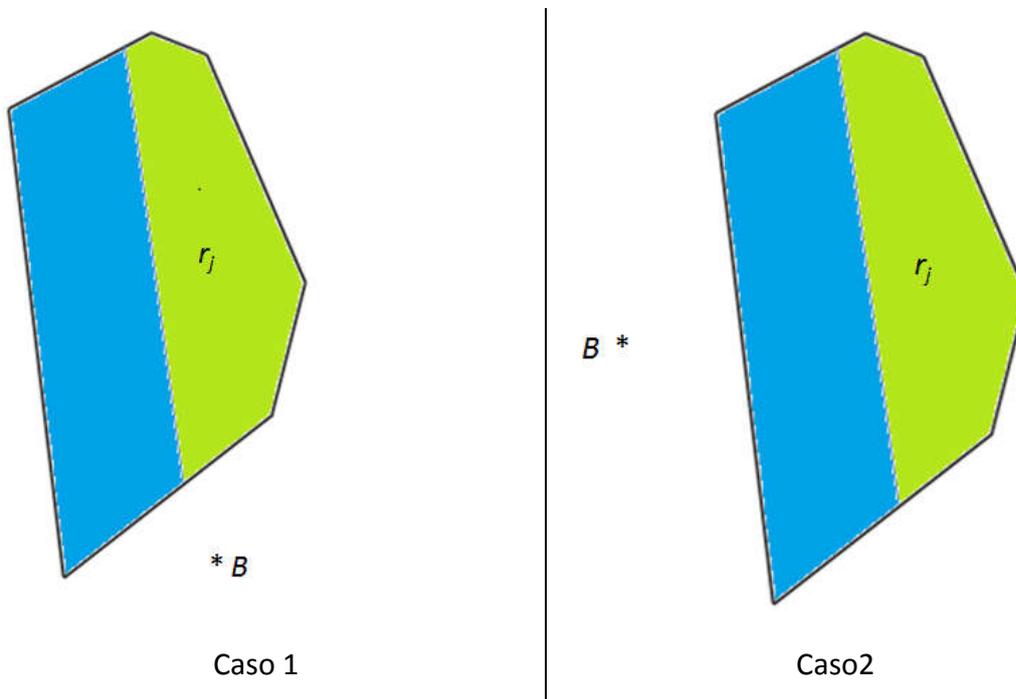


Figura 5.11 - Situaciones que pueden presentarse cuando existe una única zona de lanzamiento

Para continuar empleando el método de particionado siguiendo la dirección óptima de barrido, y cumplir con la restricción de no sobrevolar subregiones asignadas a otros UAVs, debemos crear caminos que permitan a los vehículos acceder a su subregión en los casos en los que no exista la posibilidad de acceder de forma directa desde la estación base. Esta estrategia altera muy poco el tiempo necesario para completar la misión y contribuye a elevar el tiempo de vuelo útil (tiempo de vuelo sobre la subregión asignada).

#### 5.4.2.2 Método de acceso indirecto a las subregiones

Como se comentó en el anterior apartado, el uso de una única zona de lanzamiento, si bien ayuda a simplificar la operativa de lanzamiento, debido a que implica el uso de menos recursos, también introduce un problema de posible interferencia entre aeronaves. Por ello es necesario diseñar una nueva estrategia de ingreso a las distintas subregiones que permita evitar la generación de posibles interferencias.

La solución que proponemos para resolver este problema es la creación de caminos de ingreso que se adapten a la geometría del contorno del área poligonal. En la figura 5.12 se muestra un ejemplo de una subregión con camino de ingreso, y cómo el UAV asignado a dicha subregión efectuaría el sobrevuelo de la misma.

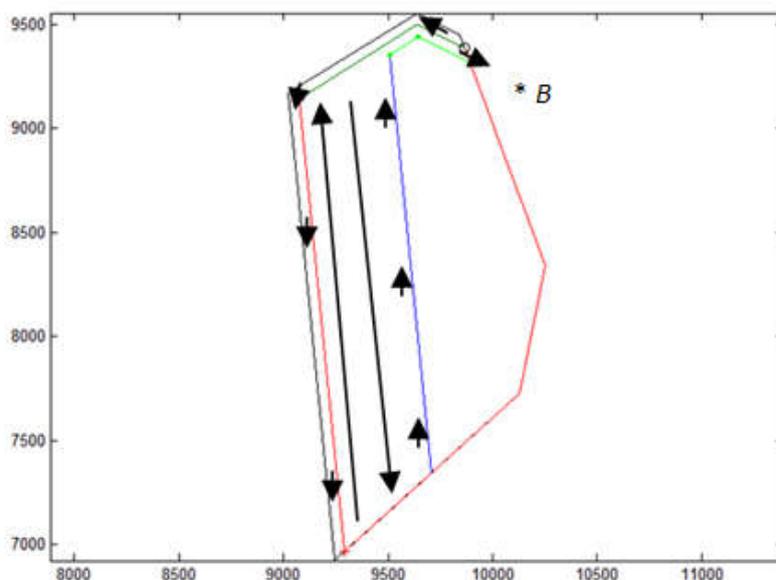


Figura 5.12 - Ejemplo de sobrevuelo de una subregión con camino de acceso

A continuación pasaremos a explicar, paso a paso, el proceso de creación de caminos de acceso indirectos:

**Paso 1** Determinar si es necesaria la creación de caminos de acceso indirecto:

Para evaluar si es necesario el trazado de un camino de acceso se utilizarán las líneas de referencia que indican la dirección óptima de barrido. De estas dos líneas tomaremos sólo aquella que esté más alejada de la estación base. Luego la desplazaremos hacia el interior del área poligonal una distancia igual a  $C$ , interceptándola en los puntos  $q_1$  y  $q_2$ . A continuación trazaremos una recta entre el punto central de la zona de lanzamiento a cada uno de los puntos ya mencionados. Si se observa que ambas rectas interceptan el área

poligonal entonces se dice que se requiere la construcción de caminos de acceso (ver figura 5.13).

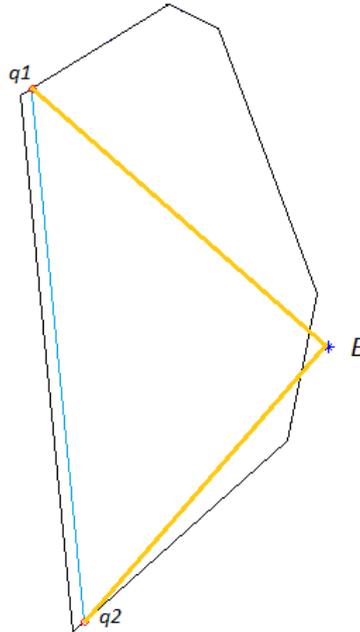


Figura 5.13 - Test para determinar si se necesita construir un camino de acceso

Paso 2 Determinar el punto de referencia para la construcción del camino: una vez superado el primer paso, procedemos determinar cuáles de los dos puntos,  $q_1$  y  $q_2$ , se tomará como referencia para la construcción del camino de acceso. Para ello tomamos los contornos comprendidos entre cada punto y el vértice al un UAV puede acceder de forma directa (ver figura 5.14).

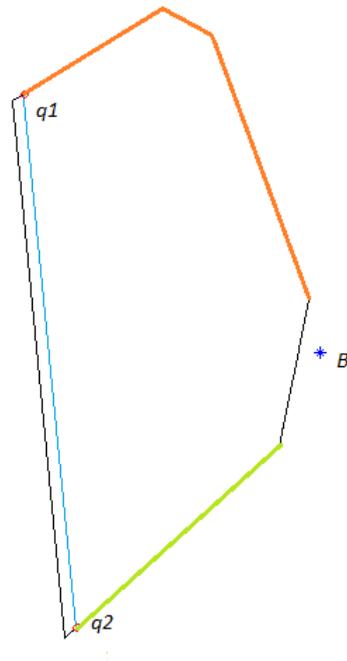


Figura 5.14 - Test para determinar qué punto se usará como referencia

Una vez hecho esto medimos la distancia a recorrer si siguiéramos el contorno marcado para cada punto. Aquél que presente una menor distancia será utilizado como punto de referencia. A partir de ahora llamaremos marco de referencia al contorno marcado para el punto de referencia.

- Paso 3 Construcción del tramo de ida: A continuación trazamos al interior del área poligonal un marco paralelo al marco de referencia, separados a una distancia igual a  $C$ . Al finalizar este paso ya tendríamos el trayecto de ida construido (ver figura 5.15).

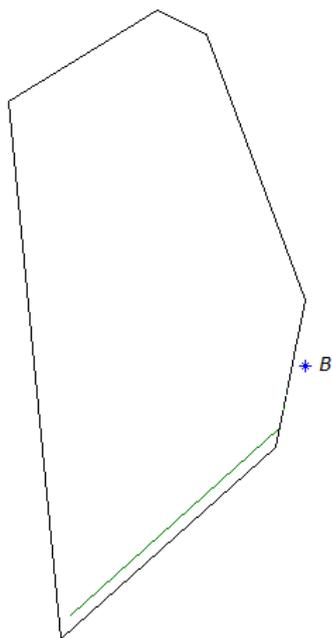


Figura 5.15 - Construcción del tramo de ida

- Paso 4 Construcción del tramo de regreso: Para la construcción del tramo de salida volveremos a trazar al interior del área poligonal un marco paralelo al marco de referencia, pero esta vez separado a una distancia igual a  $2C$ . Además el trazado de este segundo marco al interior del área poligonal estará limitado por el punto de intercepción de este con la línea que delimita cada subregión (ver figura 5.16).

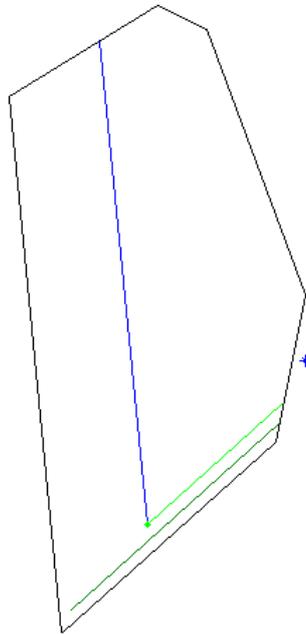


Figura 5.16 - Trazado del tramo de salida

Paso 5 Repetir los pasos 1, 3 y 4 para cada subregión: El proceso de creación de caminos de acceso va de la mano del proceso de particionado, ya que el tiempo que tarda el UAV en recorrer el camino de acceso se tiene en cuenta a la hora de calcular el tiempo necesario para recorrer cada subregión. Así, con cada iteración del mecanismo de particionado se verán modificados tanto los tramos de entrada como los de salida. Cuando se finaliza el proceso de particionado todas las subregiones poseen sus respectivos caminos, en caso de necesitarlos (ver figura 5.17).

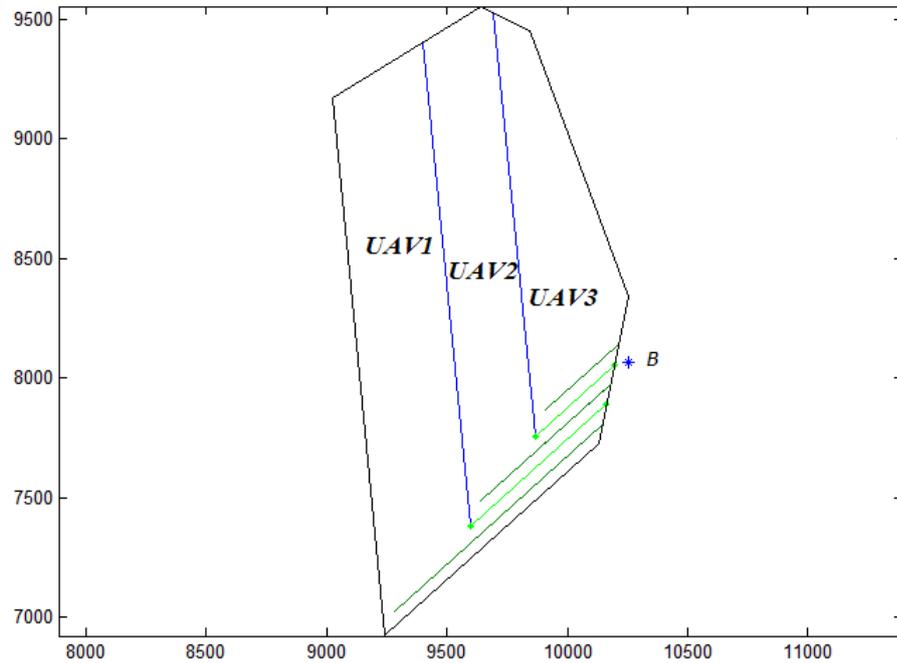


Figura 5.17 - Caminos de acceso finalizado el proceso de particionado

### 5.4.2.3 Estrategia de descomposición del área objetivo y de planificación de rutas

La estrategia adoptada para conseguir minimizar la duración global de la misión puede ser dividida en dos etapas. En una primera etapa se lleva a cabo un primer particionado asignación de subregiones. El objetivo de esta etapa es la realización de una primera inspección de la geometría del área objetivo. La asignación realizada en esta etapa se establece sólo como punto de partida.

En la segunda etapa se realiza un rebalanceo en la asignación de tramos básicos,  $tr_b$ . Con lo que se consigue equilibrar el máximo posible los tiempos de vuelo de las distintas aeronaves, minimizando así el tiempo global de la misión. En la figura 18 se muestra el diagrama de flujo que describe la estrategia adoptada.

Planificación de misión para aplicaciones de cobertura de área

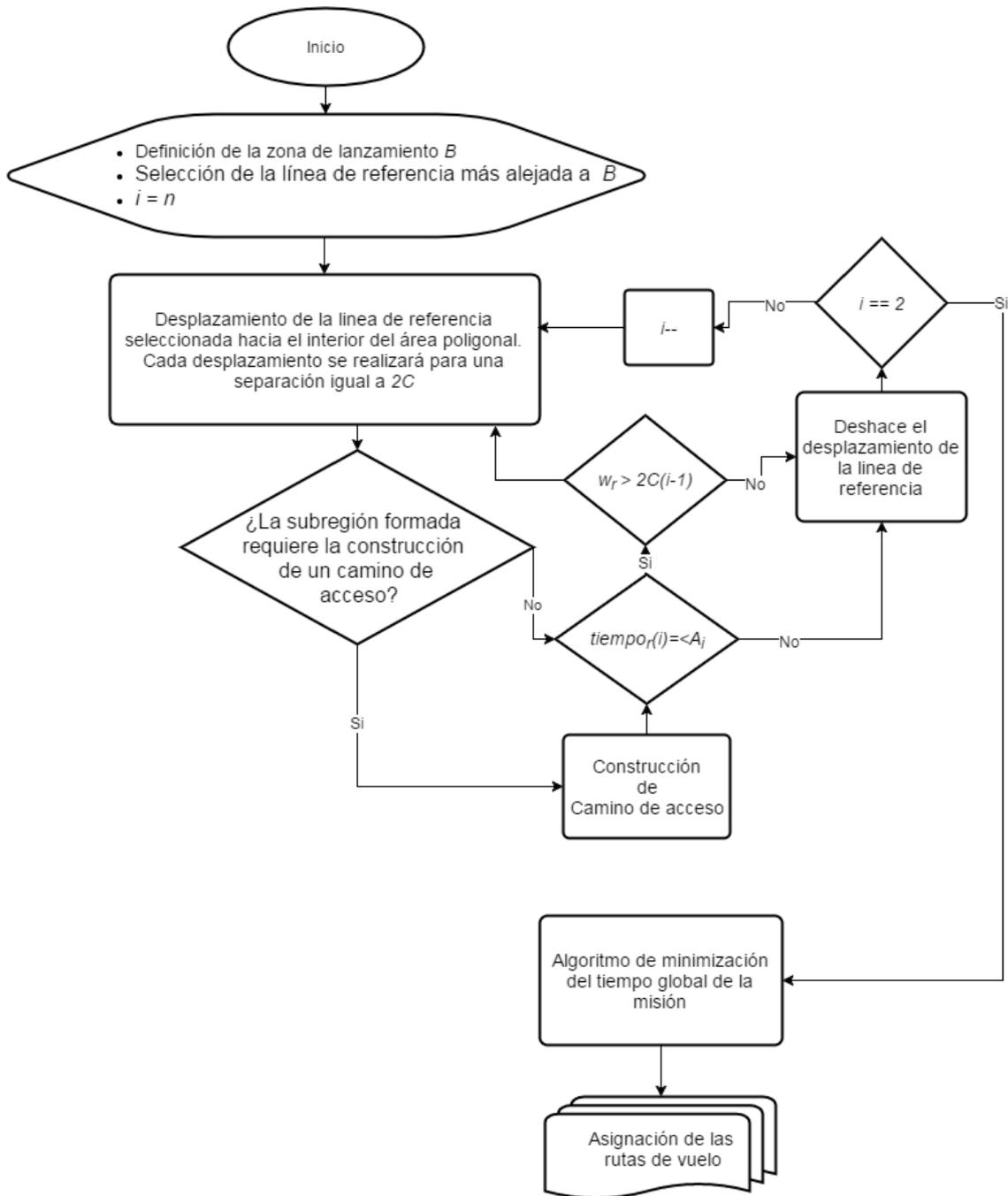


Figura 5.18 - Estrategia de particionado y planificación de rutas

#### 5.4.2.4 Algoritmo para minimizar el tiempo global de la misión

El problema de reequilibrio de los tiempos de vuelo asignados a cada UAV, puede ser planteado como un caso especial de VRP (Vehicle Routing Problem), donde cada UAV deberá visitar una determinada cantidad de tramos básicos,  $tr_b$ . El objetivo es reequilibrar la carga de trabajo entre los UAVs para así poder minimizar el máximo tiempo de recorrido y, como consecuencia, minimizar el tiempo global de la misión. Con el fin de mantener el patrón de movimiento *back and forth* y la conservación de subregiones estancas, el reequilibrio sólo puede darse entre UAVs que tengan tramos básicos adyacentes. En la figura 5.19 se presenta un caso de desequilibrio de carga de trabajo y por ende de los tiempos vuelo asignado.

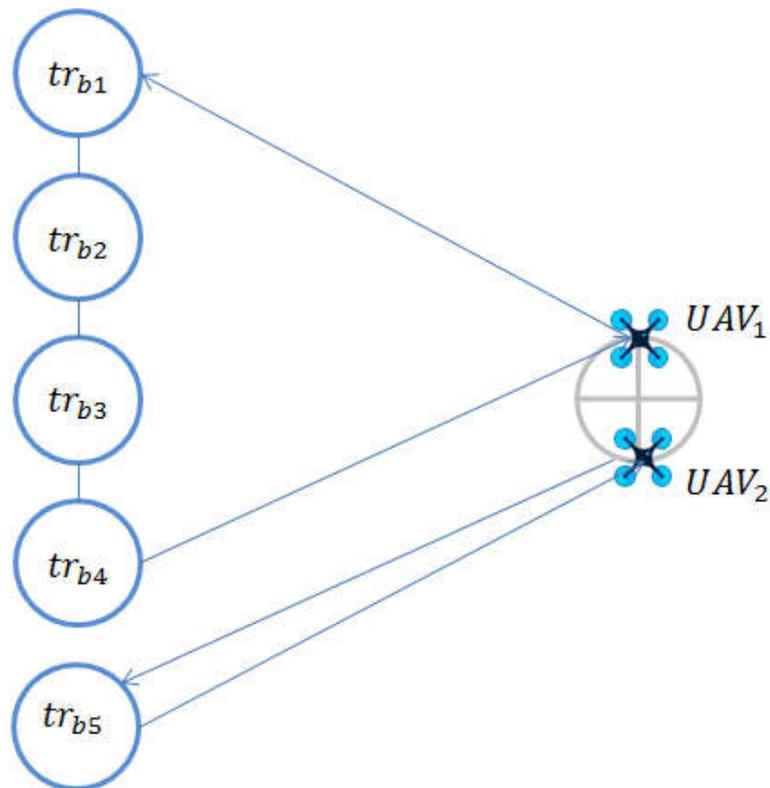


Figura 5.19 - Ejemplo de desequilibrio en la carga de trabajo

Para la resolución de este problema, nosotros proponemos un algoritmo heurístico que permite llevar a cabo el reequilibrio de los tiempos asignados manteniendo la naturaleza discreta del problema.

El algoritmo que proponemos busca minimizar de forma iterativa las diferencias entre los distintos tiempo de vuelo llevando a cabo reasignaciones de tramos básicos.

A continuación se presenta el mencionado algoritmo de minimización de tiempo global:

---

**Algoritmo 2** Minimización del tiempo global de misión

---

**Begin**

```
vcontrol = true;
q = indexoftmax(); // Guarda en q el índice del UAV que posee el mayor tiempo de ruta
tk = supposedtime(q); // Tiempo que tardaría UAVq-1 en sobrevolar el último trb asignado
                           al UAVq.

while ((abs(tiempor(q) - tiempor(q - 1)) > tk) ^ (vcontrol == true))
    changes = 0;
    i = n;
    changesexpected = true;
    while ((i ≥ 2) ^ (changesexpected == true))
        changesexpected = false;
        tk = supposedtime(i);
        while (abs(tiempor(i) - tiempor(i - 1)) > tk)
            dprev = abs(tiempor(i) - tiempor(i - 1));
            • El UAVi-1 toma el último trb del UAVi
            if (dprev - abs(tiempor(i) - tiempor(i - 1))) ≥ tolerance
                • Se conserva el cambio
                changes ++;
                changesexpected = true;
            else
                • Se desecha el cambio
                • Break;
            end
        end
    end
    i --;
```

```
end
if (changes == 0)
    vcontrol = false;
end
end
```

### 5.4.2.5 Implementación y resultados en simulación

En este apartado se ilustrará la estrategia propuesta en este capítulo mediante su implementación en la plataforma de pruebas presentada en el capítulo (3).

Para ello hemos habilitado un nuevo módulo de planificación para sistemas multi-UAV en el software de código abierto *Mission Planner*, utilizando para ello C# como lenguaje de programación y Microsoft Visual Studio 2012 como entorno de desarrollo. En la figura 5.20 se puede apreciar la interfaz de usuario creada para la ejecución de dicho módulo.

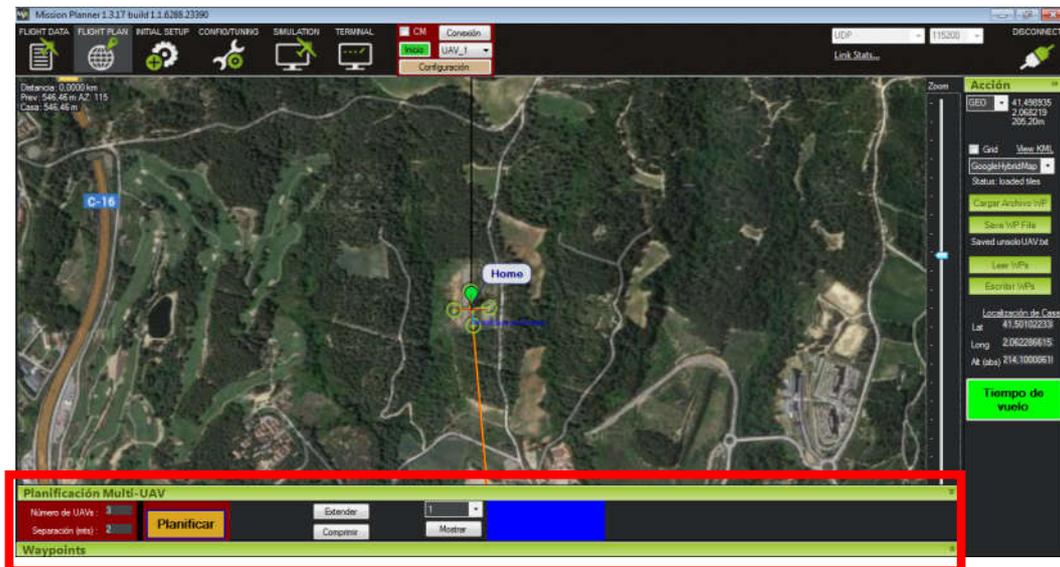


Figura 5.20 - Implementación en la estación de control en tierra de la estrategia de particionado de área y planificación de rutas presentada en este capítulo

Para mostrar el desempeño de nuestra propuesta realizaremos una simulación de cobertura de un área poligonal. Primero se mostrará el tiempo de misión para un único UAV y luego se mostrará el resultado de utilizar nuestra estrategia con una flota de 3 UAVs.

La región poligonal a cubrir tiene un área de 8,38 hectáreas (ver figura 5.21).



Figura 5.21 - Región empleada para la simulación de cobertura de área

Tabla 5.1 – Parámetros de configuración

		Posición en $B$	Sensor	$h^i$	$v_c^{ii}$	$delay_g^{iii}$	Autonomía
multi-UAV	UAV <sub>1</sub>	Lat: 41,501042 Lng: 2,062287	Tau2-13mm $C = 20,71$ m	25 m	5m/s	1 s	22 min
	UAV <sub>2</sub>	Lat: 41,501023 Lng: 2,062287	Tau2-13mm $C = 20,71$ m	25 m	5m/s	1 s	22 min
	UAV <sub>3</sub>	Lat: 41,501006 Lng: 2,062287	Tau2-13mm $C = 20,71$ m	25 m	5m/s	1 s	22 min
Single UAV		Lat: 41,501023 Lng: 2,062287	Tau2-13mm $C = 20,71$ m	25 m	5m/s	1 s	22 min

<sup>i</sup> Altura de vuelo

<sup>ii</sup> Velocidad de vuelo entre waypoints

<sup>iii</sup> Penalización de tiempo por cada cambio de dirección

Para llevar a cabo esta simulación, se ha seleccionado un grupo de quadrópteros homogéneos, con las mismas características técnicas, los mismos parámetros de configuración y el mismo modelo de cámara. Los vuelos se realizarán manteniendo una altura constante de 25 m y a una velocidad de crucero (velocidad entre waypoints) de 5 m/s. La penalización por cada cambio de dirección o giro será de 1 segundo. Mientras que la autonomía será de 22 minutos para todos los UAVs (ver tabla 5.1).

Al realizar la simulación de cobertura con un único UAV se obtiene un tiempo de ejecución de 1201,8 segundos (20,02 min), lo cual es bastante elevado tomando en cuenta que la autonomía de la aeronave es de solo 22 min. En la figura 5.22 se muestra la ruta de vuelo asignada.



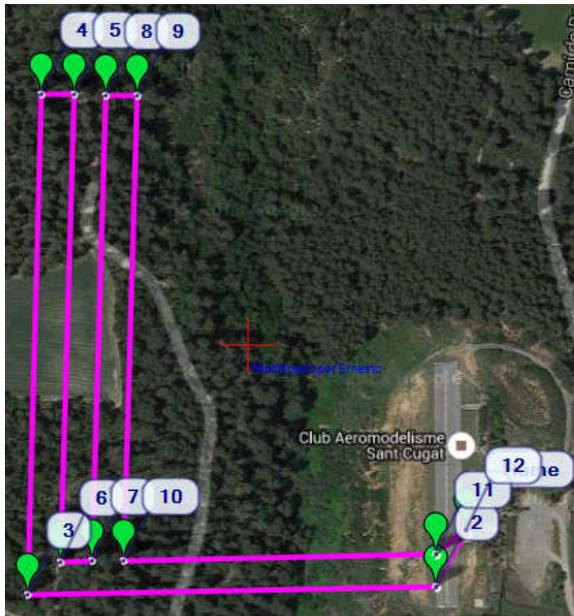
**Tiempo global de la misión:**

**20,02 min**

Figura 5.22 - Simulación de la cobertura del área con un único UAV

Utilizando la flota de tres UAVs y aplicando la estrategia explicada en el apartado 5.4.2.3, obtenemos las subregiones presentadas en la figura 5.23.

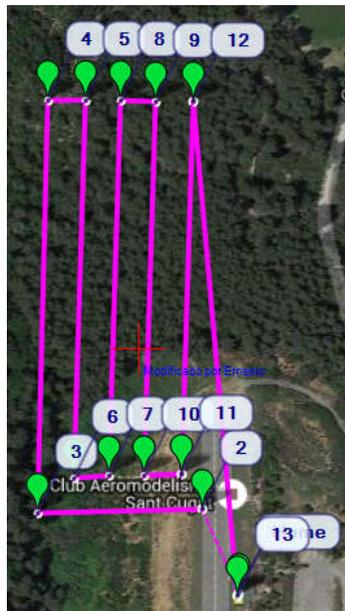
## Planificación de misión para aplicaciones de cobertura de área



Asignada al UAV<sub>1</sub>



Asignada al UAV<sub>2</sub>



Asignada al UAV<sub>3</sub>

Figura 5.23 - Descomposición del área objetivo en 3 subregiones

Se han generado tres subregiones, para las cuales se han asignado una rutas que siguen la dirección óptima de barrido, consiguiendo con ello minimizar el número de giros. A su vez, debido a la ubicación de la zona de lanzamiento  $B$ , las tres subregiones poseen caminos de acceso, garantizando con ello la no existencia de posibles interferencias físicas entre aeronaves.

Una vez se han asignado la ruta a cada UAV procedemos a la realización de la simulación para la cobertura con las tres aeronaves. Obteniendo los resultados expuestos en la siguiente tabla:

Tabla 5.2 – Resultado de la simulación para la flota de UAVs

	Tiempo de ejecución	Tiempo Global de la misión	$\sigma$
UAV <sub>1</sub>	454,00 s	454,00 s	26,23 s
UAV <sub>2</sub>	390,60 s		
UAV <sub>3</sub>	427,80 s		

Como se puede apreciar en la tabla 5.2, aplicando la estrategia de particionado propuesta en este capítulo hemos conseguido un tiempo global de misión de 454 segundos (7.56 minutos), logrando así reducir en un 62.23% el tiempo de ejecución conseguido con un único UAV.

La desviación estándar de 26,23 s, teniendo en cuenta el carácter fuertemente discreto del problema, muestra que existe un buen equilibrio en la asignación de tiempos. El tiempo de ejecución de cada UAV se encuentra muy por debajo del tiempo de autonomía de estos, lo cual es garantía de seguridad.

## 5.5 Referencias

- [1] Arkin, E. M., & Hassin, R. (1994). Approximation algorithms for the geomet covering salesman problem. *Discrete Applied Mathematics*, 55(3), 197-218.
- [2] Valente, J. (2011). An Aerial Robotic Framework to address Area Coverage in Precision Agriculture Practices. Escuela Técnica Superior de Ingenieros Industriales de la Universidad Politécnica de Madrid.
- [3] Sanz, D., Del Cerro, J., Valente, J., Garzón, M., Rossi, C., & Barrientos, A. (2011, November). Planificación de trayectorias bi-objetivo en robótica aérea para agricultura de precisión. In 3rd Workshop de Robótica: Robótica Experimental (Robot 2011), Sevilla, Spain (Vol. 4).
- [4] Xu, A., Viriyasuthee, C., & Rekleitis, I. (2014). Efficient complete coverage of a known arbitrary environment with applications to aerial operations. *Autonomous Robots*, 36(4), 365-381.
- [5] Maza, I., & Ollero, A. (2007). Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms. *InDistributed Autonomous Robotic Systems 6* (pp. 221-230). Springer Japan.
- [6] Araujo, J. F., Sujit, P. B., & Sousa, J. B. (2013, April). Multiple UAV area decomposition and coverage. In *Computational Intelligence for Security and Defense Applications (CISDA), 2013 IEEE Symposium on* (pp. 30-37). IEEE.
- [7] Avellar, G. S., Pereira, G. A., Pimenta, L. C., & Iscold, P. (2015). Multi-uav routing for area coverage and remote sensing with minimum time. *Sensors*, 15(11), 27783-27803.
- [8] Jiao, Y. S., Wang, X. M., Chen, H., & Li, Y. (2010, June). Research on the coverage path planning of uavs for polygon areas. In *Industrial Electronics and Applications (ICIEA), 2010 the 5th IEEE Conference on* (pp. 1467-1472). IEEE.
- [9] Dille, M., & Singh, S. (2013). Efficient aerial coverage search in road network

- In AIAA Guidance, Navigation, and Control (GNC) Conference (p. 5094).
- [10] Balzer, M. (2009, June). Capacity-constrained Voronoi diagrams in continuous spaces. In *Voronoi Diagrams, 2009. ISVD'09. Sixth International Symposium on* (pp. 79-88). IEEE.
- [11] Ollero, A., & Merino, L. (2006). Unmanned aerial vehicles as tools for forest-fire fighting. *Forest Ecology and Management*, 234(1), S263.
- [12] Casbeer, D. W., Beard, R. W., McLain, T. W., Li, S. M., & Mehra, R. K. (2005, June). Forest fire monitoring with multiple small UAVs. In *American Control Conference, 2005. Proceedings of the 2005* (pp. 3530-3535). IEEE.
- [13] Casbeer, D. W., Kingston, D. B., Beard, R. W., & McLain, T. W. (2006). Cooperative forest fire surveillance using a team of small unmanned air vehicles. *International Journal of Systems Science*, 37(6), 351-360.
- [14] Merino, L., Caballero, F., Martínez-de-Dios, J. R., Maza, I., & Ollero, A. (2012). An unmanned aircraft system for automatic forest fire monitoring and measurement. *Journal of Intelligent & Robotic Systems*, 65(1), 533-548.
- [15] Yuan, C., Zhang, Y., & Liu, Z. (2015). A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehicles and remote sensing techniques. *Canadian journal of forest research*, 45(7), 783-792.
- [16] Salamí, E., Pedre, S., Borensztein, P., Barrado, C., Stoliar, A., & Pastor, E. (2009, July). Decision Support System for Hot Spot Detection. In *Intelligent Environments* (pp. 277-284).
- [17] Jiao, Y. S., Wang, X. M., Chen, H., & Li, Y. (2010, June). Research on the coverage path planning of uavs for polygon areas. In *Industrial Electronics and Applications (ICIEA), 2010 the 5th IEEE Conference on* (pp. 1467-1472). IEEE.

# Capítulo 6 Capa descentralizada para la reasignación de tareas de forma cooperativa

## 6.1 Introducción

Para que un sistema multi-UAV pueda lograr los objetivos de una misión, adaptándose de forma dinámica a los cambios y eventos que surgen durante la ejecución de la misma, es necesario que las aeronaves sean capaces de coordinarse y cooperar, lo cual requiere que los UAVs posean cierto grado de autonomía en la toma de decisiones.

Las estructuras de toma de decisiones descentralizadas son muy útiles en sistemas multi-UAV en los que el intercambio de información se realiza exclusivamente a través de mensajes (las aeronaves están enlazadas entre sí a través de una red de comunicación, pero sin compartir memoria). Las soluciones descentralizadas son especialmente diseñadas para entornos en los que no existen restricciones rígidas en los retrasos de los mensajes, disponibilidad de la conectividad de red y fiabilidad de la llegada de mensajes. Además, este tipo de soluciones son capaces de manejar comunicaciones asíncronas de una forma mucho más natural que los enfoques distribuidos y centralizados.

La coordinación entre aeronaves es el principal elemento en un sistema multi-UAV cooperativo. El rendimiento general del sistema puede verse afectado directamente por la estrategia de coordinación adoptada.

La coordinación puede ser estática o dinámica. La coordinación estática (también conocida como coordinación deliberativa [1]) se refiere generalmente a la adopción de una serie de directrices que determinan cómo ejecutar una determinada tarea. En cambio, la

coordinación dinámica (también conocida como coordinación reactiva [1]) se produce en tiempo real, y se basa generalmente en el análisis y la síntesis de la información, la cual puede obtenerse a través de la infraestructura de comunicación desplegada.

La coordinación dinámica también se puede dividir en dos categorías: coordinación explícita y coordinación implícita. Gerkey y Mataric [2] definieron la coordinación explícita como aquel conjunto de técnicas que emplean métodos intencionales de comunicación y colaboración, y la coordinación implícita como aquel conjunto de técnicas que emplean métodos dinámicos de interacción entre las aeronaves y el entorno con el fin de lograr el rendimiento colectivo deseado. La coordinación explícita suele asociarse con la comunicación explícita, que es producida por el comportamiento activo de un agente. La coordinación implícita suele asociarse con la comunicación implícita, que requiere que el UAV perciba, modele y razone el comportamiento de otros.

La coordinación estática permite manejar tareas de elevada complejidad, pero es bastante deficiente al tratar tareas imprevistas, surgidas en tiempo real. Por el contrario, los métodos de coordinación dinámicos pueden tratar bien en tiempo real los cambios imprevistos, pero presentan dificultades a la hora de hacer frente a tareas de elevada complejidad.

Uno de los principales retos de cualquier sistema multi-UAV para resolver mediante esquemas de coordinación es la asignación de tareas de forma descentralizada y/o cooperativa. Para afrontar dicho reto se requiere el desarrollo e implementación de algoritmos especializados, que fruto al grado de robustez que demandan las arquitecturas descentralizadas, suelen ser algoritmos heurísticos; los cuales, a pesar de ofrecer soluciones sub-óptimas, responden muy bien a las incertidumbres del sistema y a los cambios del entorno.

Los algoritmos de asignación de tareas de forma descentralizada pueden subdividirse en 2 grandes grupos en función de cómo se distribuye el proceso de asignación entre los elementos del sistema:

- **Sistemas Auto-organizados o de *Swarm Intelligence*:** Los mecanismos basados en swarm intelligence generalmente emplean estrategias bio-inspiradas en el comportamiento de las colonias de insectos, tales como hormigas o abejas. En estos sistemas, a pesar de no existir ningún elemento coordinador central, se presenta un comportamiento cooperativo a partir de la interacción de acciones individuales muy sencillas implementadas por cada vehículo [3]. Los sistemas auto-organizados ofrecen las siguientes ventajas [4][5]:
  - **Sencillez y bajo coste:** Los sistemas basados en swarm intelligence suelen emplear aeronaves poco sofisticadas y, por ende, con un bajo coste.
  - **Alta escalabilidad:** el número de aeronaves del sistema multi-UAV puede ser fácilmente incrementado debido a la sencillez de los mecanismos de interacción y a que, por lo general, no se requiere la implementación de protocolos de comunicación entre los vehículos. Es por ello que este tipo de sistemas se caracterizan por poseer un alto número de aeronaves.
  - **Alto nivel de flexibilidad:** Los elementos del sistema pueden coordinar sus acciones de distintas formas, lo cual les permite ofrecer varias soluciones a un mismo problema en función de las condiciones del entorno.
  - **Elevado nivel de robustez:** debido a que estos sistemas cuentan con un gran número de aeronaves, el fallo de una o varias de éstas, no presenta un efecto crítico en el funcionamiento global del sistema.

Sin embargo, los sistemas auto-organizados, al no proveer a cada aeronave con información global del sistema, hacen que éstas ignoren las decisiones que han tomado los demás elementos del sistema. Ello provoca que las soluciones producidas por los sistemas basados en swarm intelligence suelen conducir en la mayoría de los casos a soluciones subóptimas.

- **Sistemas basados en mecanismos de subastas:** estos sistemas utilizan protocolos de comunicación entre los distintos elementos del mismo para llevar a cabo una

negociación que permita determinar qué tareas deberá ejecutar cada una de las aeronaves. Dichos protocolos están basados en el modelo Contract Net Protocol (CNP) propuesto en el año 1980 por Smith [6].

Los procesos involucrados en los mecanismos de subastas pueden ser resumidos en los siguientes tres pasos:

1. Se comunica a todos los miembros del sistema la apertura de un proceso de licitación para una determinada tarea. Dicha comunicación se lleva a cabo por un elemento del sistema identificado como el subastador, el cual, en muchas ocasiones, es una de las aeronaves.
2. Las aeronaves evalúan la tarea y, si están en capacidad de ejecutarla, comunican al subastador sus ofertas.
3. Después de evaluar las ofertas recibidas, el subastador adjudica la tarea a una determinada aeronave.

Las principales ventajas que ofrece el uso de mecanismos de subastas para la asignación de tareas de forma descentralizada son las siguientes:

- Elevada Flexibilidad: Siempre que las tareas puedan ser representadas mediante algún valor de utilidad, el sistema será capaz de gestionarlas.
- Simplicidad: Los protocolos empleados son sencillos e intuitivos y su implementación ofrece relativamente un bajo nivel de dificultad.
- Robustez: Los mecanismos de subastas permiten distribuir el proceso de decisión entre los diferentes agentes que componen el sistema utilizando protocolos de comunicación. De esta manera, se evitan buena parte de los problemas de los sistemas centralizados, como por ejemplo el punto único de fallo o la sobrecarga en las comunicaciones del agente central.
- Rendimiento: La solución obtenida mediante este tipo de mecanismos suele mejorar el rendimiento del sistema (incrementa el número de tareas finalizadas, reduce la energía consumida, etc.). Además, al incluir comunicación

explícita entre los distintos elementos del sistema, permite llevar a cabo tareas más complejas, como por ejemplo tareas fuertemente acopladas.

Sin embargo, los métodos de asignación de tareas basados en subastas resultan más complejos que los basados en swarm intelligence, ya que siempre requieren protocolos de comunicación para coordinar las aeronaves.

Por otro lado, en algunos problemas de asignación las tareas son generadas de forma dinámica y deben ser asignadas en tiempo real a sistemas que se encuentran ejecutando otras tareas. Este tipo de problemas recibe el nombre de *dynamic task insertion problem*. En los sistemas multi-UAV un caso de *dynamic task insertion* es el generado cuando, a causa de un fallo, alguna de las aeronaves no puede ejecutar una o varias de las tareas que le habían sido asignadas, lo cual genera la necesidad de implementar mecanismos de reasignación de dichas tareas.

En el presente capítulo propondremos una estrategia de reasignación de tareas de forma cooperativa, basándonos para ello en la implementación de mecanismos de subastas para establecer la coordinación entre las aeronaves. Dicha estrategia no garantiza una solución óptima para el problema de reasignación de tareas, pero es robusta para su uso en ambientes dinámicos, donde el tiempo de ejecución puede desviarse significativamente de las estimaciones iniciales, y donde la capacidad de adaptarse a las condiciones cambiantes es la clave del éxito.

## 6.2 Antecedentes

Desde principios del primer decenio del siglo XXI, distintos grupos de investigación en todo el mundo vienen trabajando activamente en el desarrollo de nuevas soluciones para la asignación de tareas de forma descentralizada.

La mayoría de los problemas de asignación de tareas de forma descentralizada son de tipo NP-hard y, por tanto, actualmente no existe ningún algoritmo que pueda generar una solución óptima en un tiempo acotado por un polinomio [7][8]. De esta manera, se evidencia la elevada complejidad de este tipo de problemas y la necesidad de afrontarlos mediante algoritmos heurísticos que, a pesar de no proporcionar una solución óptima, requieren tiempos de ejecución relativamente bajos.

Los problemas de asignación de tareas de forma descentralizada para sistemas multi-agente a menudo se resuelven utilizando mecanismos basados en enfoques de mercado, dado que las soluciones basadas en subastas se encuentran entre las más populares, por ser eficientes tanto en términos de coste computacional como de comunicación.

El enfoque más simple viene dado por el establecimiento de un único subastador, que introduce nuevas tareas y supervisa el proceso de licitación, como se presenta en [9]. Existen otras propuestas que permiten a cualquier agente del sistema tomar el rol de subastador, como se puede apreciar en [10], lo cual genera la posibilidad de conflictos en el caso de que múltiples agentes decidan subastar una misma tarea. Tomando en cuenta esto, la propuesta presentada en el presente capítulo establece unos mecanismos de control para la asignación del rol de subastador, impidiendo así que dicho rol esté vinculado a un único agente, a la vez de que se evita la ocurrencia de conflictos de subastas.

Por otro lado, se puede lograr una asignación casi óptima de un conjunto de tareas mediante la utilización de subastas combinatorias [11], [12]. En dichas subastas cada agente deberá realizar una oferta basada en lotes de tareas en lugar de realizar ofertas por tareas individuales. Cada oferta deberá indicar el coste que le generaría ejecutar todas las tareas pertenecientes a un mismo lote partiendo del estado actual. Sin embargo, esta estrategia es computacionalmente ineficiente. Por otro lado, las subastas paralelas de un solo ítem permiten a un agente licitar por varias tareas de forma independiente. Dicho mecanismo es muy eficiente, tanto en coste computacional como en coste de comunicación, sin embargo

ofrece soluciones muy subóptimas, ya que no tiene en cuenta ninguna sinergia entre las tareas. Para evitar dicho problema, se puede emplear el esquema de subastas secuenciales de un único ítem, lo cual permite realizar ofertas que ofrezcan cierto grado de sinergia basada en la distancia con el resto de tareas que el UAV tiene pendiente de ejecutar.

Otras soluciones para el problema de asignación de tareas emplean un esquema de subastas distribuido, en el cual se elimina el rol de subastador y, mediante algoritmos ejecutados en cada uno de los agentes, se establecen consensos. El algoritmo *consensus-basedbundle* (CBBA) propuesto en [13] ha despertado gran interés entre los investigadores. Éste realiza un proceso de iteración entre la fase de construcción de un único paquete de tareas que va actualizándose continuamente a medida que se realiza la subasta, y una fase de resolución de conflictos que tiene como fin lograr un consenso sobre las ofertas ganadoras. Recientemente, se han sugerido varias modificaciones para ampliar sus funciones y áreas de aplicación [14][15].

### **6.3 Formulación del Problema**

Cuando realizamos la asignación de tareas en sistemas multi-UAV en la fase de planificación de la misión, lo hacemos teniendo en cuenta las características y capacidades de las aeronaves, y asumiendo ciertas condiciones del entorno donde se llevará a cabo la misión. Sin embargo, en ciertas ocasiones ocurren imprevistos que pueden poner en peligro la viabilidad de la misión.

Entre los distintos incidentes que pueden ocurrir durante el transcurso de una misión se encuentran los siguientes ejemplos:

1. Problemas de hardware en alguna de las aeronaves (fallo de motor, fallo del receptor GPS, baja autonomía de la batería, fallo del autopiloto, etc.)
2. Cambios atmosféricos (aumento de la velocidad del viento, tormentas, etc.)

3. Colisión de alguna aeronave con obstáculos móviles inesperados (aves, aeronaves ajenas a la misión, etc.)

Algunos de estos incidentes pueden dar al traste con toda la misión, en particular aquellos que afectan de forma simultánea a toda la flota de UAVs. Por ejemplo, si en el lugar en el que se desarrolla la misión se desata de forma repentina una fuerte tormenta, la mejor solución será abortar la misión y esperar a que pase el mal tiempo.

Sin embargo, la gran mayoría afectará a aeronaves individuales y, en estos casos, es posible llevar a cabo estrategias que permitan redistribuir las tareas entre el resto de aeronaves, atenuando de este modo los efectos negativos en la consecución de los objetivos globales planteados en la misión.

### 6.3.1 Escenario

Para abordar el problema de reasignación de tareas de forma cooperativa, en este trabajo se tomará como punto de partida la solución dada al escenario planteado en el capítulo anterior, en el cual un conjunto de  $n$  UAVs se encuentra realizando la cobertura de un área poligonal convexa  $P$  sin obstáculos en su interior. Cada  $UAV_i$  tiene asignada una subregión  $r_j$ , compuesta por un conjunto de waypoints. Tras producirse un fallo en uno de los UAVs que le impide la llevar a cabo la ejecución total o parcial de su ruta de vuelo, se genera un conjunto de waypoints que deberán ser reasignados al resto de la flota para mitigar los efectos de dicho fallo en el logro del objetivo global de la misión.

Para dar una descripción formal del problema utilizamos la siguiente notación:

$U = \{UAV_1, \dots, UAV_{n-1}\}$  : Conjunto de UAVs disponibles que participarán en el proceso de reasignación

$W_f = \{w_1, \dots, w_k\}$  : Conjunto de waypoints que deberán ser reasignados

$T_A = \{A_1, \dots, A_{n-1}\}$	: Conjunto que contiene los tiempos de autonomía de vuelo de cada uno de los UAVs que participarán en el proceso de reasignación
$A_1 \leq A_2 \dots \leq A_{n-1}$	: Los UAVs serán organizados de forma ascendente según su autonomía
$R = \{r_1, \dots, r_m\}$	: Conjunto que contiene las subregiones generadas
$tiempo_r(i)$	: Tiempo que tardaría el $UAV_i$ en volar su ruta tras el proceso de reasignación de waypoints (para su cálculo se empleará la misma ecuación presentada en el capítulo anterior)
$W_i = \{w_1, \dots, w_l\}$	: Conjunto de $l$ waypoints que conforman la ruta de barrido asignada al $UAV_i$

El objetivo global continúa siendo la minimización del tiempo máximo de la misión (que viene dado por el mayor de los tiempos de vuelo de los UAVs que constituyen la flota). Es decir, se plantea el siguiente problema de optimización:

$$\text{Minimizar } t_{max} = \max_{i=1..n-1} \{tiempo_r(i)\} \quad (a)$$

Sujeto a

$$tiempo_r(i) \leq A_i \forall i \in I = \{1.. n - 1 \} \quad (b)$$

$$\sum_{i=1}^{n-1} x_{ig} \leq 1 \quad \forall g \in G = \{1.. k\} \quad (c)$$

$$x_{ig} \in \{0,1\}, \quad \forall (ig) \in I \times G \quad (d)$$

La función objetivo (a) se formula de manera que su minimización implique la atenuación, en la medida de lo posible, del efecto del fallo ocurrido sobre el tiempo de ejecución global de la misión de cobertura de área. La restricción (b) establece

que el tiempo necesario para recorrer la subregión  $r$  asignada al  $UAV_i$  nunca podrá ser mayor a la autonomía de vuelo de la aeronave. Por otro lado, (c) establece que cada waypoint  $w_g$  solo podrá ser asignado a un único  $UAV_i$ , para lo cual se emplea la variable de decisión binaria  $x_{ig}$ . Cuando  $x_{ig} = 1$  significa que al  $UAV_i$  se le ha asignado el waypoint  $w_g$ .

Debido a que debemos garantizar en la medida de lo posible la cobertura de toda el área poligonal, la reasignación de waypoints se llevará a cabo por tramos compuestos por dos waypoints, ya que así garantiza no dejar zonas sin cubrir. Por lo que si tenemos los waypoints  $\{w_1, \dots, w_k\}$  para ser reasignados, se conformaran pares  $[w_g, w_{g-1}] \forall g \in G = \{1, \dots, k-1\}$ .

Las características de las aeronaves, parámetros de vuelo y sensor son los mismos que los presentados en el escenario del capítulo anterior.

Como hipótesis de trabajo, se asumirá que cada UAV puede comunicarse en todo momento tanto con todas las demás aeronaves como con la estación de control en tierra.

### 6.3.2 Enfoque

Como el tiempo global de la misión es marcado por el UAV con mayor carga de trabajo, debemos adoptar una estrategia que permita reasignar los waypoints liberados tratando de atenuar su efecto negativo en la duración de la misión. Para ello tendremos en cuenta varios tipos de fallos diferentes y adoptaremos mecanismos que permitan evitar posibles problemas de interferencia física entre aeronaves.

## 6.4 Estrategia propuesta para la reasignación de tareas de forma cooperativa basada en mecanismos de subastas

En esta tesis proponemos una estrategia basada en subastas secuenciales de un solo ítem (*Sequential Single-item Auction, SSA* [16]) para llevar a cabo la reasignación de un conjunto de waypoints liberados por un UAV averiado. El proceso de replanificación de ruta será ejecutado de forma local y dinámica por cada aeronave.

Para ello se subastarán aquellos tramos  $[w_g w_{g+1}]$  que la aeronave averiada no puede sobrevolar, lo cual lleva a que las demás aeronaves modifiquen sus rutas de vuelo para incorporar los nuevos waypoints.

Cada UAV tendrá almacenada la matriz de asignación de waypoints  $M_w$ , la cual contiene todos los waypoints ubicados en el interior del área objetivo asignados a cada aeronave.

$$\begin{bmatrix} w_{11} & w_{12} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ w_{n1} & w_{n2} & \cdots & 0 \end{bmatrix}$$

El número de filas de  $M_w$  es igual al número de UAVs existentes en el sistema, mientras el número de columnas será igual al número total de waypoints ubicados en el interior del área a cubrir. La fila  $M_w(i, \cdot)$  contiene todos los waypoints asignados al UAV  $i$ . Los espacios de la matriz que no contienen waypoints contienen el valor cero.

Cada waypoint se representa como  $w_{ij} = [lat \ lng \ alt \ flag]$ , donde  $lat$ ,  $lng$  y  $alt$  hacen referencia a la latitud, longitud y altitud del waypoint, respectivamente, mientras que  $flag$  es un indicador que suministra información del estado del waypoint durante el transcurso de la misión.

El indicador  $flag$  puede adoptar los siguientes estados:

Tabla 6.1 – Estados del indicador *flag*

<i>flag</i>	Descripción
"o"	Waypoint asignado en la etapa de planificación, el cual aún no ha sido sobrevolado
"v"	Waypoint que ya ha sido sobrevolado
"r"	Waypoint reasignado
"s"	Waypoint apartado para subasta

Cada UAV, al llegar a un waypoint  $w_{ij}$  asignará el valor "v" al indicador *flag* en la matriz de asignación  $M_w$ , a su vez, le comunicará al resto de aeronaves y a la estación de control en tierra los subíndices  $i$  y  $j$  correspondientes al waypoint visitado, mediante el mensaje "Wv- $i:j$ ", con el objetivo de que éstos también modifiquen el estado de dicho indicador.

Esta continua actualización de la matriz  $M_w$  almacenada en cada aeronave permite que cada elemento del sistema posea toda la información necesaria sobre el desarrollo de la misión.

Además de la matriz de asignación, cada UAV llevará un registro de posición (latitud, longitud y altitud) y tiempo de vuelo restante para cada una de las rutas que en ese momento tienen asignadas cada uno de los UAV de la flota. Dicho registro se muestra a continuación:

$$E_r = [[lat\ lng\ alt\ t\ r_1] [lat\ lng\ alt\ t\ r_2] \dots [lat\ lng\ alt\ t\ r_n]]$$

Así, como se verá en el siguiente apartado, este registro se irá actualizando de forma continua durante el transcurso de la misión.

#### 6.4.1 Disparadores del proceso de reasignación de waypoints

El proceso de reasignación de waypoints se llevará a cabo sólo en el caso de que, por un incidente inesperado, un UAV no pueda completar en su totalidad la ruta asignada.

Los tipos de incidentes que pueden impedir a un UAV completar su misión los agrupamos en dos categorías y para cada una de ellas definimos un disparador, tal como se explica a continuación:

- Disparador por bajo nivel de batería: En ocasiones las baterías tienden a presentar fallos fruto del desgaste propio de su vida útil o a una mala gestión de los ciclos de carga y descarga. Para poder detectar estos posibles fallos cada aeronave deberá ir verificando cada  $t_v$  segundos que el nivel de carga de la batería le permite completar su ruta asignada. Si transcurrido ese tiempo  $t_v$  no se detecta el problema de bajo nivel de batería, se procede a enviar el mensaje "Ok -  $i$  - [ $lat$   $lng$   $alt$   $t_{ri}$ ]" el cual es usado para indicar al resto del sistema que el sobrevuelo de la ruta asignada se está efectuando sin ningún tipo de inconveniente. En dicho mensaje,  $i$  identifica el UAV que ha enviado el mensaje, y [ $lat$   $lng$   $alt$   $t_{ri}$ ] indica la posición, altura y tiempo de vuelo restante en el momento de enviar el mensaje. Este mensaje tiene como fin actualizar el registro de seguimiento  $E_r$  de cada UAV.

Por otro lado, si durante el proceso de verificación, un determinado UAV <sub>$i$</sub>  comprueba que la actual autonomía no le permite mantenerse en vuelo el tiempo necesario para sobrevolar todos los waypoints restantes de su plan de vuelo, comenzará a descontar waypoints de su ruta, comenzando por el último que tenía asignado en su plan de vuelo hasta quedarse solo con aquellos que su autonomía le permita cubrir. Los waypoints descontados serán puestos en subasta. Si el tiempo estimado de vuelo restante de la ruta modificada es superior a  $2 * t_s$  (dos veces la ventana de tiempo estipulada para las subastas), el propio UAV afectado podrá gestionar la subasta. En caso contrario, deberá enviar tanto a la estación de tierra como al UAV con mayor tiempo de vuelo restante,  $t_r$  (según los datos almacenados en el registro  $E_r$ ) el mensaje "Ps -  $i$ :  $j$ ", el cual indica que se debe realizar una

subasta de los tramos conformados por los waypoints asignados al  $UAV_i$  a partir del  $w_{ij}$  (ver figura 6.1).

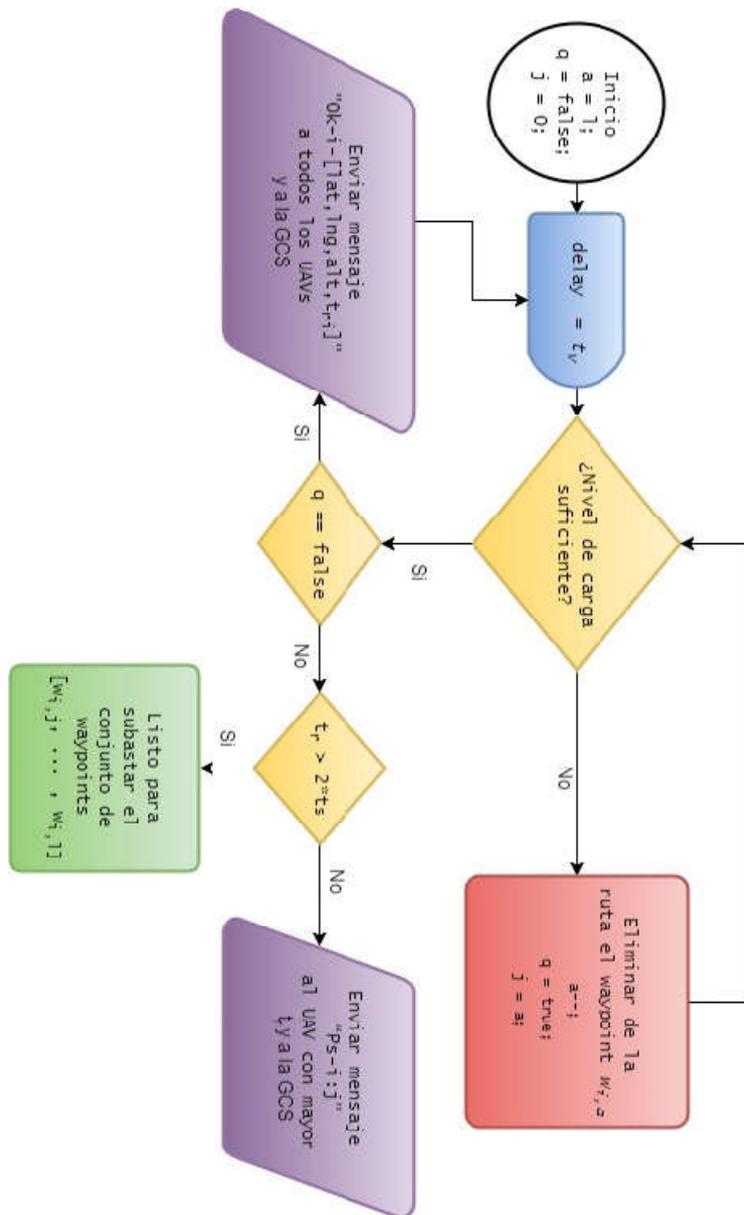


Figura 6.1 - Disparador de proceso de reasignación por falla de batería

El UAV con mayor  $t_r$ , tras recibir el mensaje que le indica que debe iniciar un proceso de reasignación de waypoints, deberá inmediatamente ejecutar el

algoritmo de subasta. Si transcurrido un tiempo equivalente a  $3 * t_s$  la subasta de los tramos de waypoints no ha sido iniciada, la estación de control en tierra tomará el rol de subastador (ver figura 6.2).

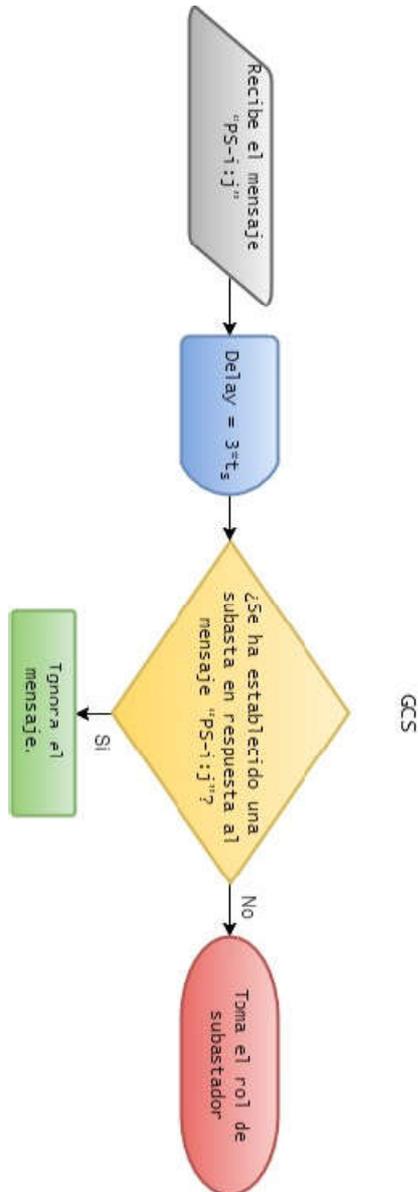


Figura 6.2 - Respuesta de la GCS al mensaje del disparador por problemas de batería

- Disparador por pérdida de un UAV: Algunos tipos de fallos (avería de motores, pérdida de la señal GPS, daño en el sistema de autopiloto, etc.) pueden causar la pérdida total de un UAV. En estos casos, puesto que el UAV afectado no podrá

comunicar la existencia del problema, se debe crear un mecanismo que permita al sistema identificar el incidente. Para ello se ha adoptado la siguiente política: la aeronave con mayor  $t_r$  deberá ir registrando el tiempo entre cada mensaje de "OK-i" proveniente de las demás aeronaves y en el caso de que haya transcurrido un tiempo mayor a  $2 * t_v$  sin que un determinado UAV haya enviado dicho mensaje, considerará a dicho UAV como perdido y pasará a poner a subasta los waypoints no visitados (ver figura 6.3). En este caso en particular, para poder completar todos los posibles tramos que pudo haber dejado de sobrevolar el UAV siniestrado, se incluirá dentro de los waypoints a subastar la última posición almacenada en el registro de seguimiento  $E_r$  (a partir de ahora esta posición pasará a ser el waypoint  $w_{i0}$ ), por lo que el primer tramo a subastar será el  $[w_{i0} w_{ij}]$ .

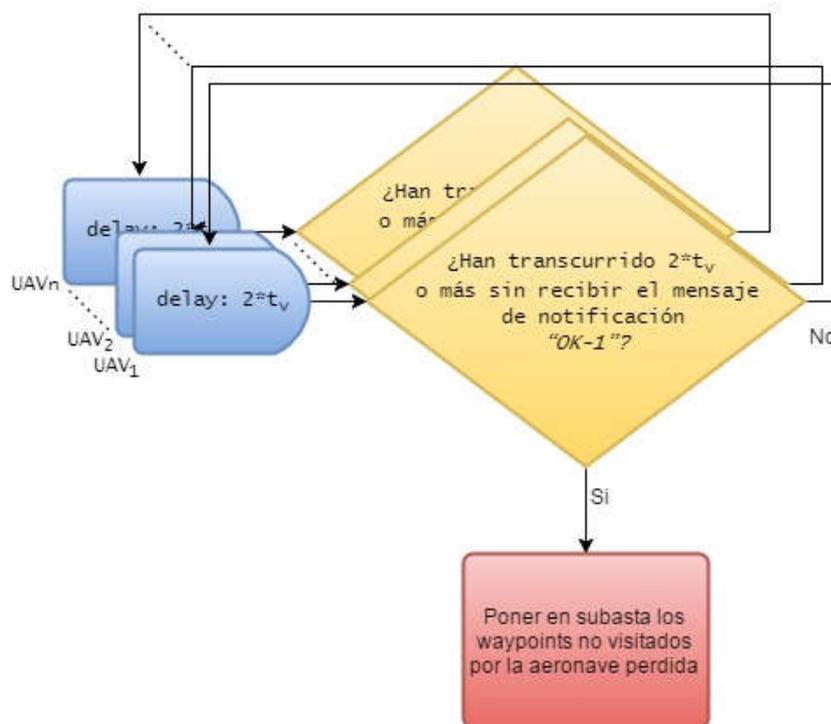


Figura 6.3 - Disparador por pérdida de la aeronave

Para hacer más robusto dicho mecanismo se ejecutará también con un cierto delay en la GCS, tal como muestra la figura 6.4.

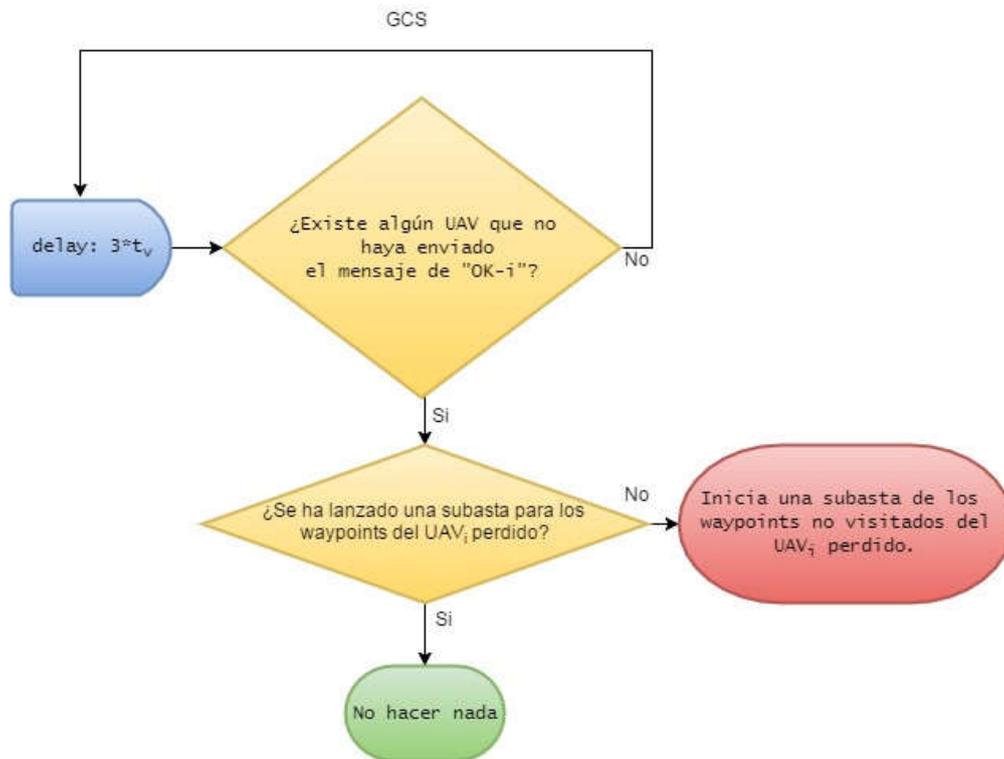


Figura 6.4 - Disparador del proceso de reasignación por pérdida de un UAV ejecutado por la GCS

### 6.4.2 Algoritmo de reasignación de tramos de waypoints basado en mecanismos de subastas

Una vez que uno de los dos tipos de disparadores anteriormente presentados ha liberado un grupo de waypoints para ser reasignados, se inicia el proceso de subasta.

En particular emplearemos el sistema de Subastas Secuenciales de un solo Item (*Sequential Single-item Auctions* - SSA). En dicho sistema se plantea un único agente que subasta las tareas de manera secuencial. Se trata de un mecanismo muy simple pero muy efectivo, tal como demuestran los resultados tanto teóricos como experimentales presentados en [7].

A continuación presentaremos dos diagramas de flujo que muestran la estrategia SSA propuesta, tanto desde el punto de vista del subastador como del de las aeronaves (ver figuras 6.5 y 6.6, respectivamente).

Dado que el rol de subastador será ejercido por el UAV al que se le haya asignado la ruta que requiera mayor tiempo de vuelo, y teniendo en cuenta que por efecto del propio proceso de reasignación las rutas pueden sufrir modificaciones, podemos concluir que nuestra estrategia establece un mecanismo dinámico de intercambio de rol basado en una jerarquización de las aeronaves a partir de su carga de trabajo. Esto cual evita que varias aeronaves puedan estar ejerciendo el rol de subastador para una misma tarea.

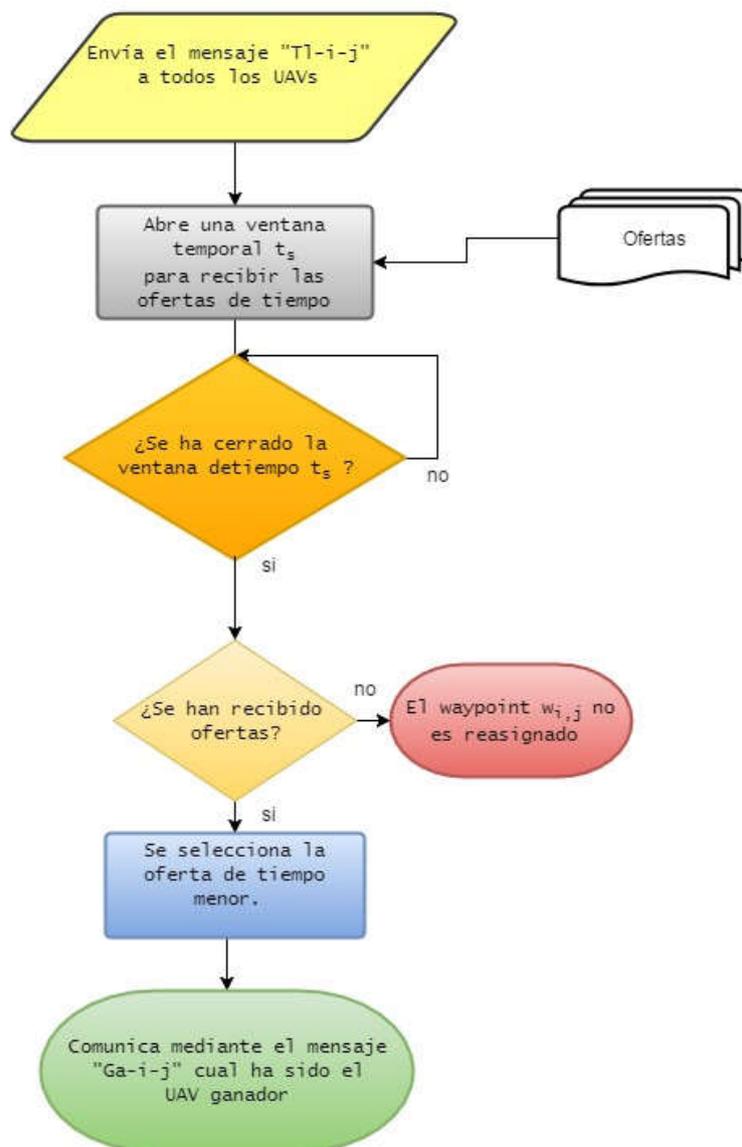


Figura 6.5 - Esquema de la subasta de un determinado waypoint desde el punto de vista del subastador

## Reasignación de tareas de forma cooperativa

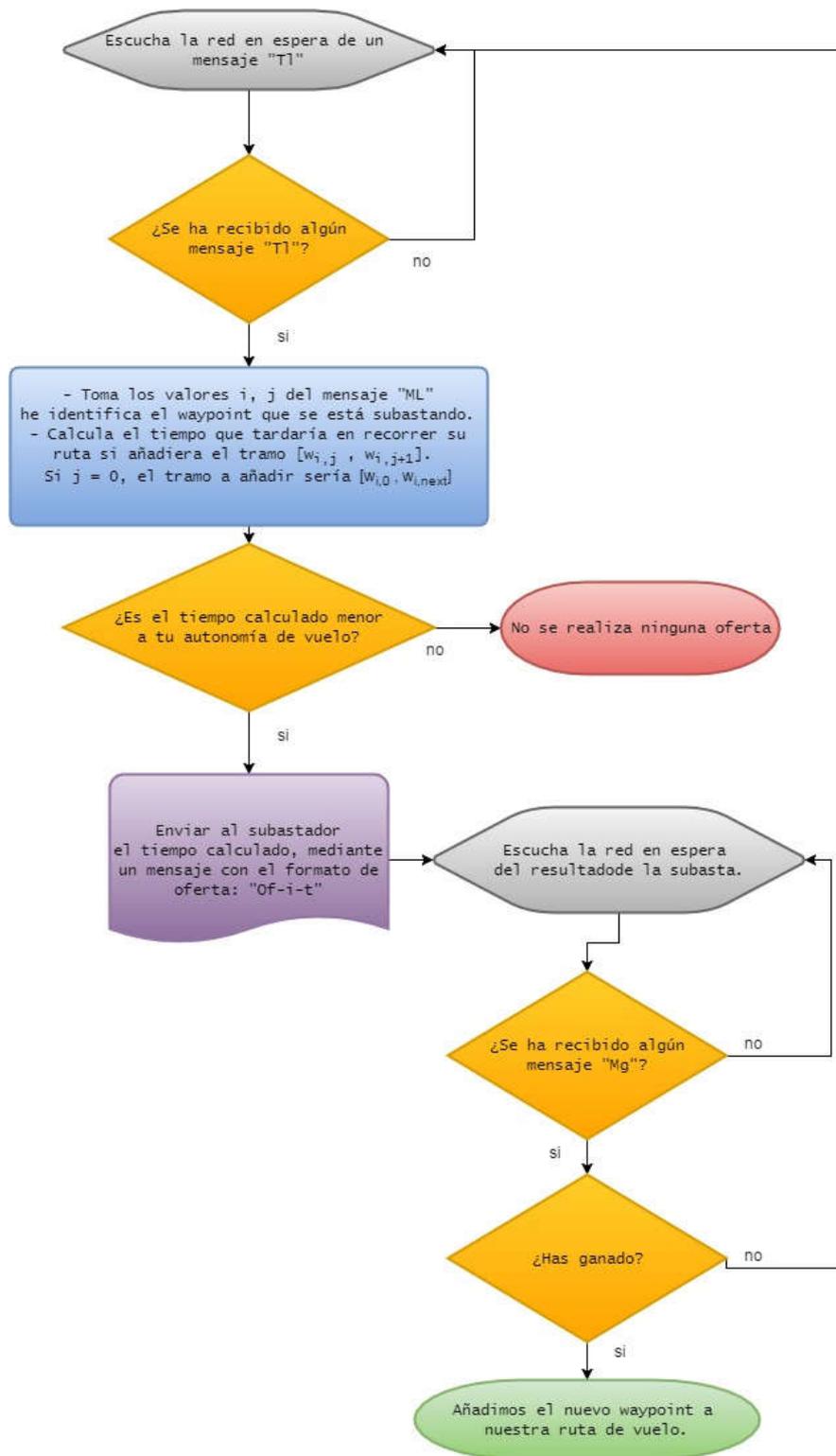


Figura 6.6 - Esquema de la subasta de un determinado waypoint desde el punto de vista de las aeronaves

En las figura 5 y 6 podemos ver cómo se produce la interacción entre el subastador y las aeronaves para llevar a cabo el proceso de subasta del waypoint  $w_{ij}$ . Para dar inicio a la licitación, el subastador envía a todos los UAVs del sistema el mensaje " $Tl - i: j$ ", el cual indica que se ha abierto un proceso de licitación para el  $w_{ij}$ . Inmediatamente después de realizar el envío del mensaje comienza a escuchar la red en espera de ofertas durante un tiempo igual a  $t_s$ . Los UAVs, al recibir el mensaje " $Tl - i: j$ " identifican el waypoint gracias a la matriz de asignación y conforma el tramo  $[w_{ij} w_{ij-1}]$  (si  $j = 0$ , entonces el tramo objetivo será  $[w_{ij} w_{next}]$ , donde el sufijo *next* hace referencia al siguiente waypoint del UAV<sub>*i*</sub>). Luego de esto se agrega a la ruta de vuelo el tramo anteriormente mencionado y se calcula el tiempo necesario para cubrir la nueva ruta de vuelo. Si dicho tiempo es mayor a la autonomía de la aeronave, no se realiza ninguna oferta, pero si se obtiene un tiempo menor, se envía dicho tiempo como oferta al subastador con el mensaje " $Of - i: t$ ". Del lado del subastador, al cerrarse la ventana de tiempo para escuchar ofertas, en el caso de que no se haya recibido ninguna oferta,  $w_{ij}$  no será reasignado, lo cual implica que el tramo correspondiente de la superficie a inspeccionar no será visitado. Pero si se reciben ofertas, el subastador elegirá la de aquél UAV que, incorporando el tramo en cuestión a su plan de vuelo previo, ofrezca el menor tiempo restante de vuelo. El ganador de la subasta es comunicado al resto del sistema mediante el mensaje " $Ga - i$ ".

Para llevar a cabo el cálculo del tiempo de vuelo a ofertar se emplea el siguiente algoritmo:

---

**Algoritmo 1** Cálculo del tiempo a ofertar para el waypoint  $w_{i,j}$

---

**Input**

$tramo = [w_{i_j} w_{i_{j-1}}];$

$ruta = [w_j w_{j+1} \dots w_l];$  //Solo se toman los waypoints aún no visitados

**Output**

$ruta\_mod;$

$tiempo\_de\_oferta;$

**Begin**

$pt = p\_central(w_{i_j} w_{i_{j-1}})$  //Punto central del tramo de ruta

$count = 0;$

$ref = [];$

$d = \infty;$

**foreach** ( $w$  in  $ruta$ )

$S = segment(w, w_{count+1});$  //segmento formado por  $w, w_{count+1}$

$pc = p\_segment(S, pt);$  //punto de  $S$  más cercano a  $pt$

**if** ( $distance(pc, pt) < d$ )

$ref = [count, pc];$

$d = distance(pc, pt);$

$count++;$

**if** ( $count == l$ )

**Break;**

$ruta\_mod = agregar(ruta, tramo, ref);$

$tiempo\_de\_oferta = tiempo_r(ruta\_mod);$

**End**

---

Dicho algoritmo recibe como inputs el tramo de ruta que se desea incorporar y el conjunto de waypoints de la ruta de la propia aeronave que aún no han sido visitados. Con dichos inputs determina el punto central del tramo a incorporar,  $pt$ , para luego determinar la posición dentro de la ruta original donde se ejecutará el desvío necesario para sobrevolar dicho tramo. Para ello, mediante métodos geométricos, se procede a la búsqueda del punto de la ruta ( $pc$ ) más cercano a  $pt$ .

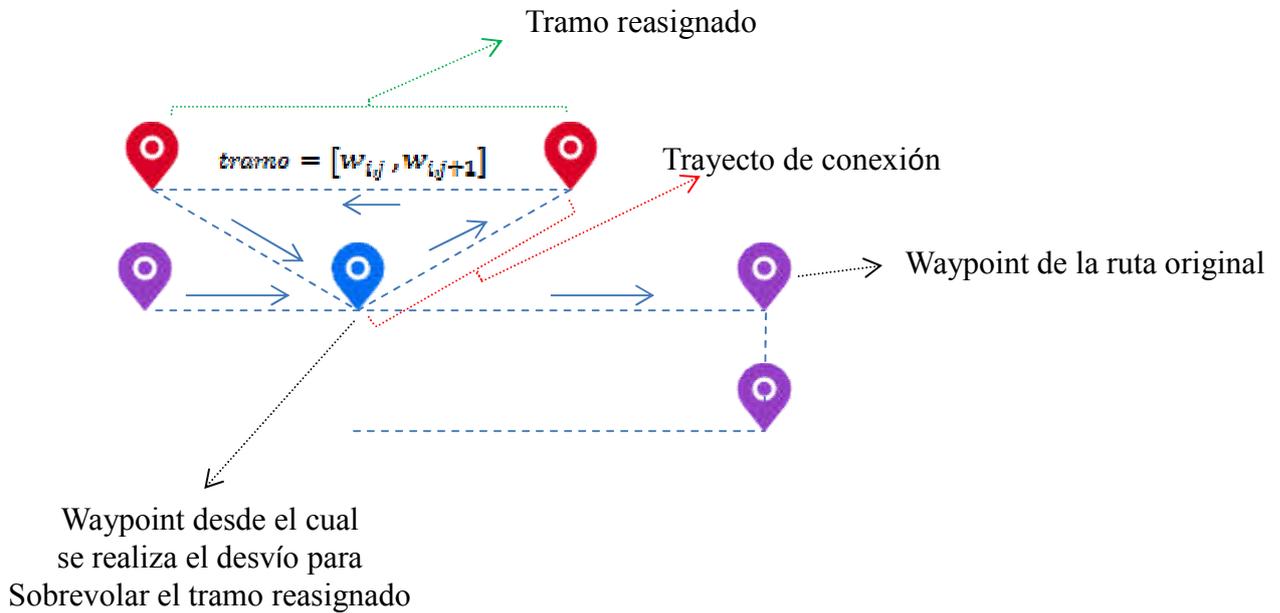


Figura 6.7 - Ejemplo del mecanismo de replanificación de rutas ejecutado de forma local

Una vez identificado el punto  $pc$  se procede a crear una ruta modificada (ver figura 6.7) y a calcular el tiempo requerido para sobrevolarla, el cual será el tiempo que se ofertará en la subasta. Este método de replanificación de ruta nos permite mantener el patrón de movimiento *forth and back*.

#### 6.4.2.1 Implementación y resultados en simulación

Basándonos en el esquema de toma de decisiones de forma distribuida de la arquitectura multi-UAV propuesta en el capítulo 3, en el presente apartado se mostrará la implementación de la estrategia de reasignación cooperativa de tareas y los resultados de las simulaciones realizadas. Dicha implementación se ha llevado a cabo en la plataforma de pruebas presentada en el capítulo 2.

La estrategia de reasignación de tareas de forma cooperativa presentada en el presente capítulo, requiere para su puesta en marcha la implementación de las distintas directivas de

actuación y de comunicación, tanto en la estación de control en tierra como en cada una de las aeronaves.

En el lado tierra del sistema, hemos programado el módulo GCS cooperativo en el software de código abierto *Mission Planner*, utilizando para ello C# como lenguaje de programación y Microsoft Visual Studio 2012 como entorno de desarrollo. Dicho módulo se encargará de monitorizar en tiempo real el desarrollo de la misión gracias a la información suministrada de forma periódica por cada UAV. En la estrategia cooperativa aquí presentada, la estación de control en tierra se convierte en un elemento capaz de tomar decisiones e interactuar con las aeronaves de forma autónoma, sin necesidad de la intervención del operador del sistema. Sin embargo, el módulo GCS cooperativo actúa sólo como un elemento de soporte en el proceso de coordinación mediante subastas, ya que únicamente tomará el rol de subastador en caso de que ninguna aeronave pueda asumir dicho rol. Este módulo es el encargado de enviar a cada aeronave la matriz de asignación  $M_w$  al inicio de la misión, lo cual sirve como indicador para los UAVs de que en la misión que está por iniciarse deberá ejecutar la estrategia de reasignación cooperativa de waypoints en caso de producirse algún fallo inesperado que afecte la normal ejecución de las rutas de vuelo planificadas. Dicho envío se realizará cuando el operador habilite el módulo GCS cooperativo, marcando el *checkbox* "CM" y presionando el botón de "Inicio", ubicados en el menú principal del software Mission Planner (ver figura 6.8). Es de suma importancia que el operador realice esta opción cuando todos los UAVs hayan establecido la conexión con la GCS y antes de que éstos inicien la ejecución de la misión.

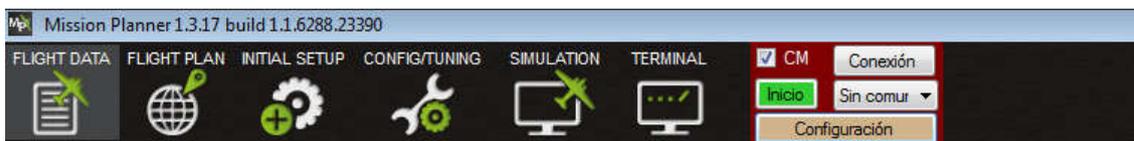


Figura 6.8 - Sección de la interfaz de usuario que permite iniciar la estrategia de reasignación de waypoints de forma cooperativa

## Reasignación de tareas de forma cooperativa

En el lado aire, la implementación de la estrategia se lleva a cabo mediante un programa escrito en Python llamado *Cooperacion.py* ubicado en cada UAV. Para su programación ha sido necesario el uso de la Api DroneKit-Python, la cual permite a los desarrolladores tener acceso mediante programación a toda la información que genera el UAV, como son la telemetría y los distintos parámetros de configuración. A su vez, gracias a esta poderosa Api, podemos ejercer un control directo sobre el movimiento y las operaciones de la aeronave, lo cual posibilita la creación de programas que permitan a la propia aeronave modificar su plan de vuelo.

También se ha empleado la librería *Shapely* para abordar todos los aspectos referentes a los cálculos geométricos necesarios para la implementación de nuestra propuesta.

Con el fin de ilustrar el desempeño de nuestra propuesta de reasignación de waypoints pasaremos a presentar los resultados de una serie de simulaciones llevadas a cabo en el entorno de simulación multi-UAV propuesto en la presente tesis.

Para ello partiremos del mismo escenario de simulación del apartado 5, por lo que cada UAV posee una ruta de vuelo ya asignada (ver figura 6.9).

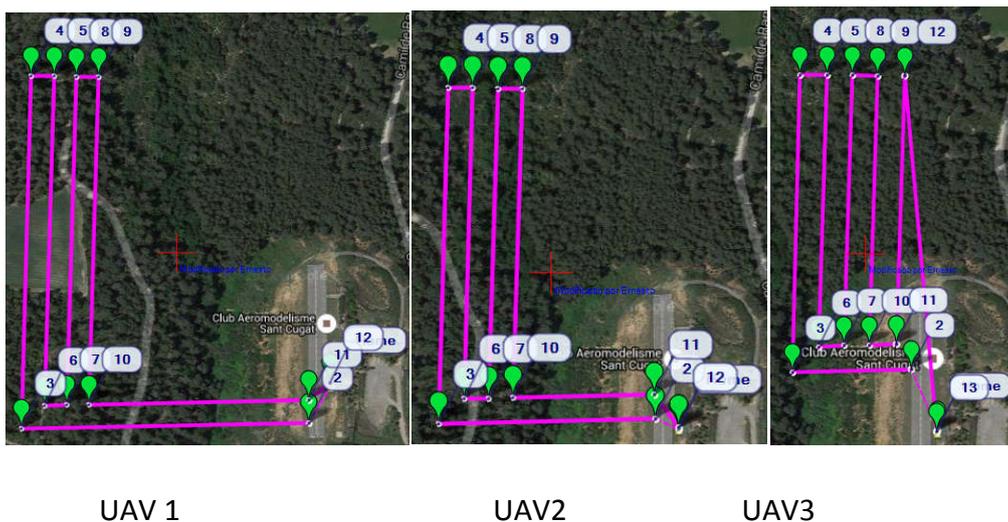


Figura 6.9 - Rutas de vuelo asignadas a cada UAV empleando el algoritmo de asignación presentado en el capítulo 5

Pero, en este caso, en el momento de iniciar la ejecución de la misión, simularemos que el UAV2, tras realizar el primer proceso de verificación (ver figura 6.1), descubre que debido a fallos inesperados de la batería el nivel de carga de ésta es mucho más bajo de lo esperado, lo que le obliga a poner en subasta algunos de sus waypoints para que sean reasignados.

Para ello realizaremos la simulación de 9 casos diferentes, que difieren en el número de waypoints que serán puestos en subasta. Yendo del caso 1 en el que sólo se pone en subasta el waypoint wp11 hasta el caso 9, en el que debido a un fallo total de la batería el UAV2 queda inoperativo a los pocos segundos de su despegue, por lo que se deben poner a subasta todos los waypoints ubicados dentro de la zona objetivo (ver tabla 6.2).

Para esta serie de simulaciones se ha adoptado un tiempo  $t_v = 20s$  y un tiempo  $t_s = 10s$

Tabla 6.2 – Casos de reasignación de waypoints simulados

	Caso 1	Caso 2	Caso 3	Caso 4	Caso 5	Caso 6	Caso 7	Caso 8	Caso 9
wp 11	Subasta								
wp 10	-	Subasta							
wp 09	-	-	Subasta						
wp 08	-	-	-	Subasta	Subasta	Subasta	Subasta	Subasta	Subasta
wp 07	-	-	-	-	Subasta	Subasta	Subasta	Subasta	Subasta
wp 06	-	-	-	-	-	Subasta	Subasta	Subasta	Subasta
wp 05	-	-	-	-	-	-	Subasta	Subasta	Subasta
wp 04	-	-	-	-	-	-	-	Subasta	Subasta
wp 03	-	-	-	-	-	-	-	-	Subasta

A continuación se presentan los resultados de las modificaciones generadas a las rutas a partir de la estrategia de reasignación de waypoints propuesta en este trabajo, basada en subastas SSA.

## Reasignación de tareas de forma cooperativa

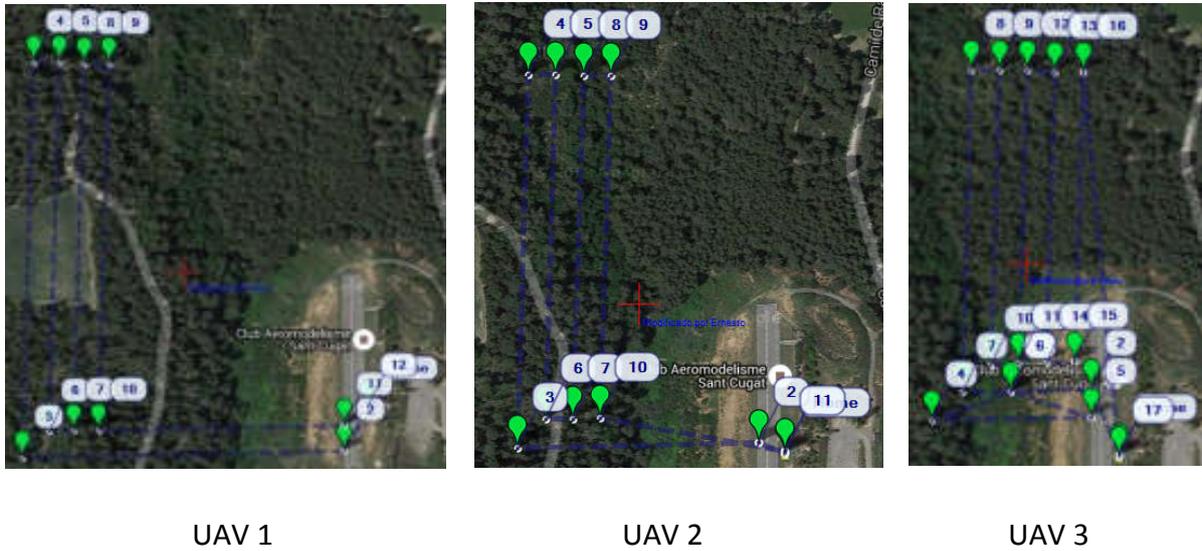


Figura 6.10 - Caso 1 de Reasignación de waypoints

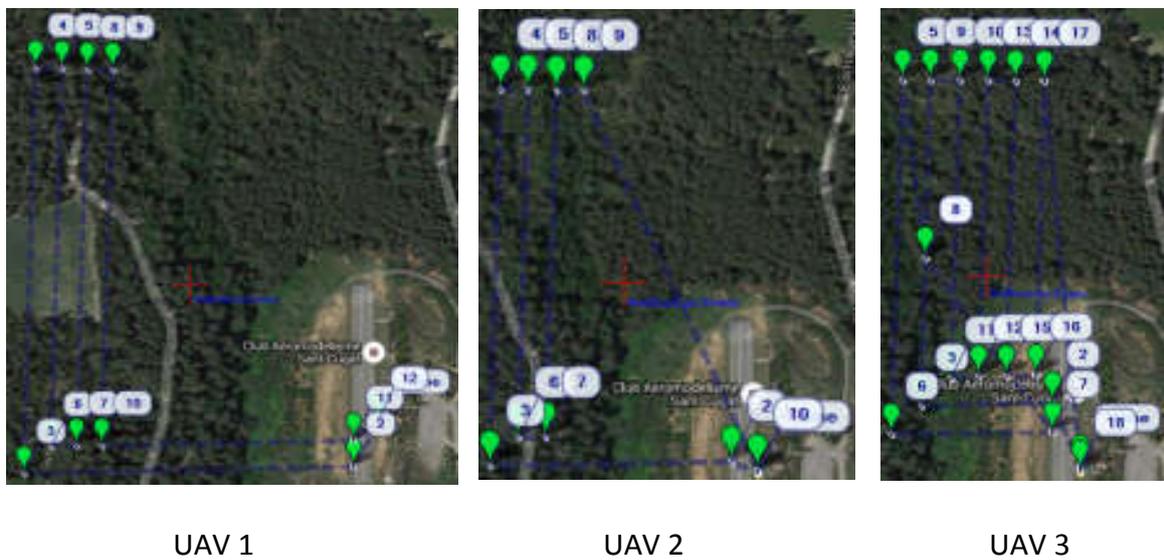


Figura 6.11 - Caso 2 de Reasignación de waypoints

Reasignación de tareas de forma cooperativa



UAV 1

UAV 2

UAV 3

Figura 6.12 - Caso 3 de Reasignación de waypoints



UAV 1

UAV 2

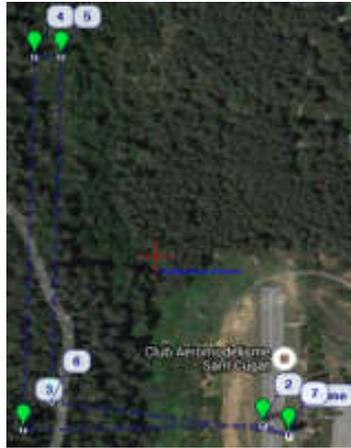
UAV 3

Figura 6.13 - Caso 4 de Reasignación de waypoints

Reasignación de tareas de forma cooperativa



UAV 1



UAV 2

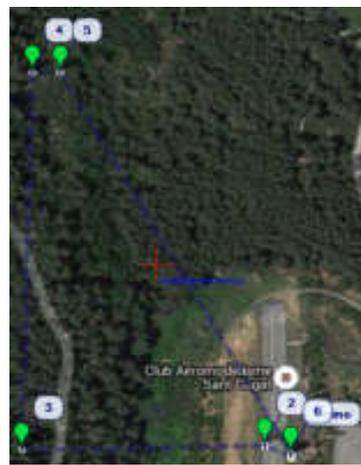


UAV 3

Figura 6.14 - Caso 5 de Reasignación de waypoints



UAV 1



UAV 2



UAV 3

Figura 6.15 - Caso 6 de Reasignación de waypoints

Reasignación de tareas de forma cooperativa

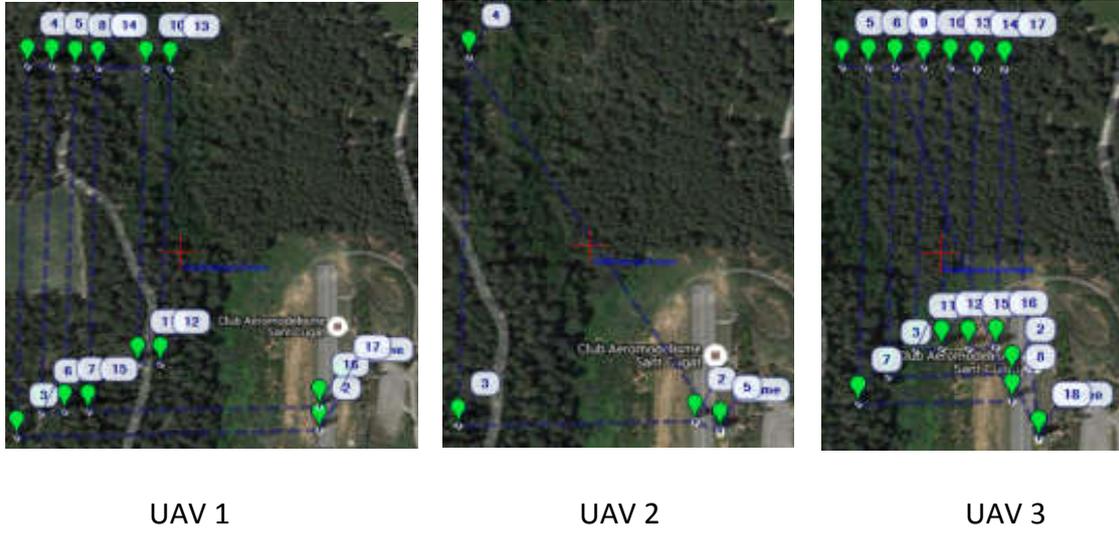


Figura 6.16 - Caso 7 de Reasignación de waypoints

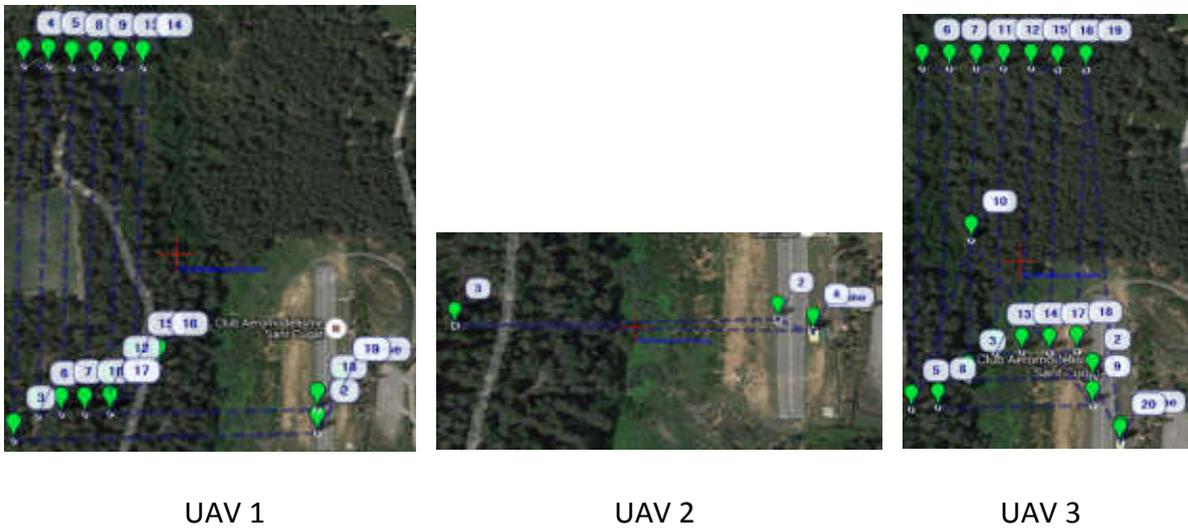
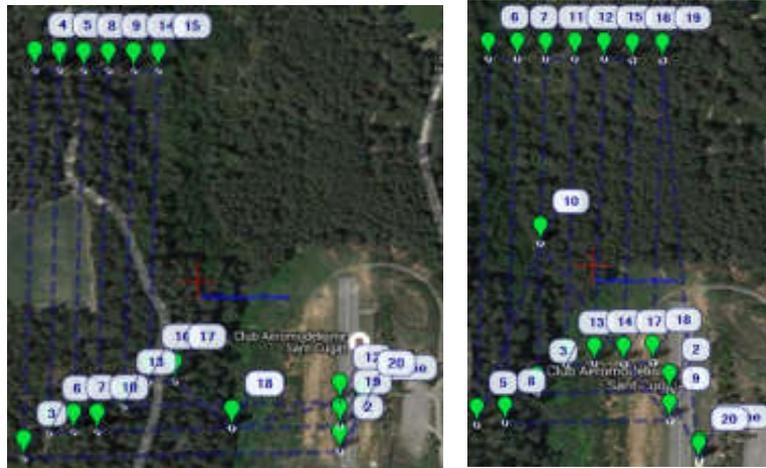


Figura 6.17 - Caso 8 de Reasignación de waypoints

## Reasignación de tareas de forma cooperativa



UAV 1

UAV 3

Figura 6.18 - Caso 9 de Reasignación de waypoints

En la figura 6.19 se presentan los tiempos de vuelo resultantes tras el proceso de reasignación de waypoints para cada uno de los 9 casos planteados.

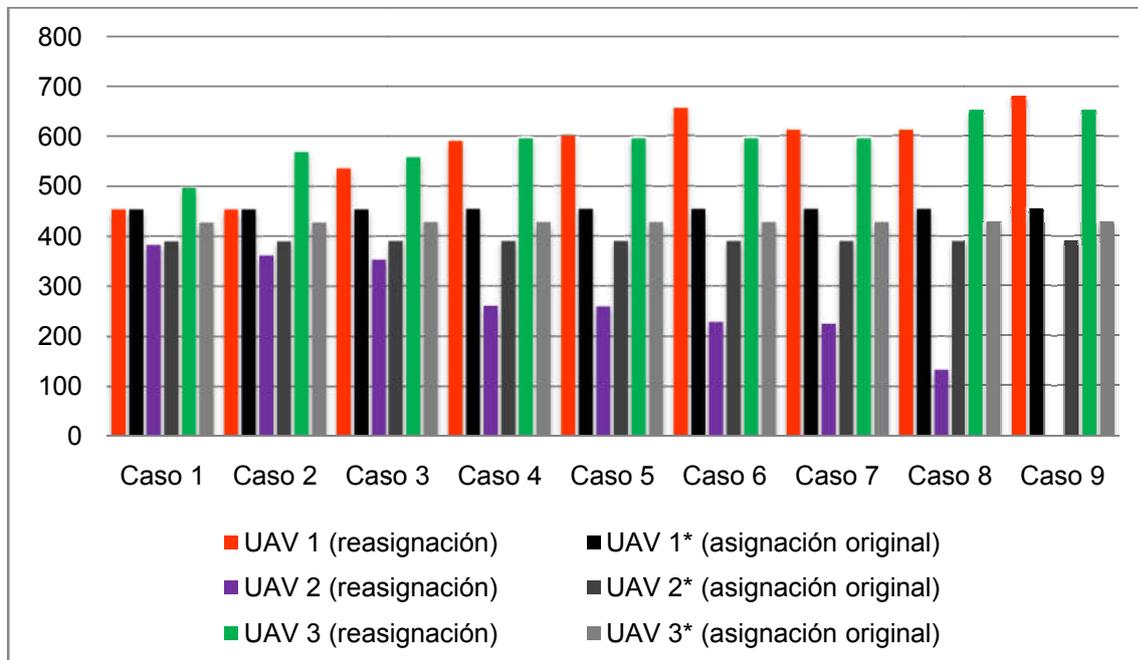


Figura 6.19 - Tiempos de vuelo para cada UAV fruto de la ejecución de la estrategia de reasignación de waypoints

En la figura 6.20 se presenta el incremento que experimenta el tiempo global de misión al aplicar la estrategia de reasignación.



Figura 6.20 - Incremento del tiempo global de misión debido al proceso de reasignación de waypoints

Cómo se puede apreciar en las figuras que ilustran los resultados del procesos de reasignación, la estrategia adoptada ha permitido reasignar todos los waypoints en cada uno de los 9 casos establecidos, logrando así cumplir con el principal objetivo de la misión de realizar la cobertura del área objetivo.

En la figura 6.19 y 6.20 se muestra como se modifican los tiempos de vuelo de cada UAV y el tiempo global de la misión respectivamente. La figura 6.20 revela el efecto negativo que provoca la alteración de la planificación original fruto a la ocurrencia de un fallo técnico en una de las aeronaves, llegando a registrarse un incremento de hasta 50.22% del tiempo global de misión en el caso 9. A pesar de ello, se observa en la figura 6.19 como en la mayoría de los casos la estrategia adoptada consigue atenuar dicho efecto gracias a una reasignación de waypoints que permite que la diferencia de tiempo de vuelo entre los UAVs no afectados sea menor a los 60 segundos en la mayoría de los casos.

## 6.5 Referencias

- [1] Iocchi, L., Nardi, D., & Salerno, M. (2000, August). Reactivity and deliberation: a survey on multi-robot systems. In *Workshop on Balancing Reactivity and Social Deliberation in Multi-Agent Systems* (pp. 9-32). SpringerBerlin Heidelberg.
- [2] Gerkey, B., & Mataric, M. J. (2004, March). Are (explicit) multi-robot coordination and multi-agent coordination really so different. In *Proceedings of the AAAI spring symposium on bridging the multi-agent and multi-robotic research gap* (pp. 1-3).
- [3] Cao, Y. U., Fukunaga, A. S., & Kahng, A. (1997). Cooperative mobile robotics: Antecedents and directions. *Autonomous robots*, 4(1), 7-27.
- [4] Bayindir, L., & Şahin, E. (2007). A review of studies in swarm robotics. *Turkish Journal of Electrical Engineering & Computer Sciences*, 15(2), 115-147.
- [5] Sahin, E., Girgin, S., Bayindir, L., & Turgut, A. (2008) *Swarm Intelligence: Introduction and Applications*, ser. Natural Computing. Springer Verlag, cap. Swarm Robotics, pp. 87–100.
- [6] R. G. Smith. 1980. The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver. *IEEE Trans. Comput.* 29, 12 (December 1980), 1104-1113.
- [7] Dias, M. B., Zlot, R., Kalra, N., & Stentz, A. (2006). Market-based multirobot coordination: A survey and analysis. *Proceedings of the IEEE*, 94(7), 1257-1270.
- [8] Sariel, S., & Balch, T. R. (2006, May). Efficient Bids on Task Allocation for Multi-Robot Exploration. In *FLAIRS Conference* (pp. 116-121).
- [9] Ajorlou, A., Homaifar, A., Esterline, A., Moore, J. G., & Bamberger, R. J. (2007). *Market-based coordination of UAVs for time-constrained remote data collection and relay*. PhD thesis, Citeseer.
- [10] Sujit, P. B., & Beard, R. (2007, July). Distributed sequential auctions for multiple UAV task allocation. In *American Control Conference, 2007. ACC'07* (pp. 3955-3960). IEEE.

- [11] Berhault, M., Huang, H., Keskinocak, P., Koenig, S., Elmaghraby, W., Griffin, P., & Kleywegt, A. (2003, October). Robot exploration with combinatorial auctions. In *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on* (Vol. 2, pp. 1957-1962). IEEE.
- [12] Zaman, S., & Grosu, D. (2013). A combinatorial auction-based mechanism for dynamic VM provisioning and allocation in clouds. *IEEE Transactions on Cloud Computing*, 1(2), 129-141.
- [13] Choi, H. L., Brunet, L., & How, J. P. (2009). Consensus-based decentralized auctions for robust task allocation. *IEEE transactions on robotics*, 25(4), 912-926.
- [14] Das, G. P., McGinnity, T. M., Coleman, S. A., & Behera, L. (2011, September). A fast distributed auction and consensus process using parallel task allocation and execution. In *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on* (pp. 4716-4721). IEEE.
- [15] Johnson, L., Ponda, S., Choi, H. L., & How, J. (2011). Asynchronous decentralized task allocation for dynamic environments. In *Infotech@ Aerospace 2011* (p. 1441).
- [16] Koenig, S., Tovey, C., Lagoudakis, M., Markakis, V., Kempe, D., Keskinocak, P., ... & Jain, S. (2006, July). The power of sequential single-item auctions for agent coordination. In *Proceedings of the National Conference on Artificial Intelligence* (Vol. 21, No. 2, p. 1625). Menlo Park, CA; Cambridge, MA; London; AAI Press; MIT Press; 1999.

## Capítulo 7 Conclusiones y Trabajo Futuro

En la presente tesis se ha conseguido el objetivo propuesto: diseñar una arquitectura de toma de decisiones completa para el diseño e implantación de sistemas multi-UAV híbridos en aplicaciones de cobertura de área, la cual cubre todas las fases de ejecución de este tipo de aplicaciones. En este capítulo enumeramos las contribuciones de este trabajo, así como las principales líneas futuras.

### 7.1 Contribuciones de la presente tesis

Las principales contribuciones de esta tesis son las siguientes:

- Se ha dotado al software Mission Planner de la capacidad de poder operar con múltiples aeronaves de forma simultánea.
- Se ha desarrollado un entorno de simulación multi-UAV Software in the Loop bastante versátil, ya que admite una gran variedad de configuraciones, pudiendo ser utilizado como un entorno de prueba enteramente virtual, hasta poder realizar pruebas mixtas, en la que interactúen UAVs reales con aeronaves simuladas. Dicha plataforma de simulación ha resultado de vital importancia en la realización de la presente tesis, ya que ha permitido desarrollar e implementar cada uno de los algoritmos aquí presentados.
- Se ha desarrollado un nuevo algoritmo de planificación de rutas para la cobertura de un área en tiempo mínimo con sistemas multi-UAV. Este algoritmo, a diferencia de los encontrados en la bibliografía, garantiza la cobertura de un área determinada con varios UAVs lanzados desde una única zona de lanzamientos y sin existir posibilidad de interferencia física entre las aeronaves, ya que ninguna de las rutas asignadas se

cruzan entre sí. Para conseguirlo, se ha creado una novedosa estrategia de construcción de caminos de ingreso y salida para los casos en los que el UAV no puede acceder directamente a la subregión que le fue asignada. Para garantizar la minimización del tiempo de vuelo se ha utilizado el patrón de movimiento *back and forth* en la dirección de barrido óptimo. A partir de los tiempos de vuelo observados en los resultados en la simulación, se puede concluir que el algoritmo de planificación propuesto realiza asignaciones de carga de trabajo bastante equilibradas, lo cual permite minimizar el tiempo de ejecución global de la misión.

- También se ha desarrollado una estrategia de cooperación basada en el mecanismo de subasta de un solo ítem (SSA) capaz de realizar procesos de reasignación de waypoints y replanificación de rutas de forma totalmente descentralizada. Para ello se ha creado toda una estrategia de interacción entre todos los elementos del sistema basada en el intercambio de mensajes periódicos. Cada UAV es capaz de estimar su tiempo restante de vuelo mediante la implementación de una función basada en los perfiles de velocidad. Los resultados de las simulaciones demuestran que dicha estrategia consigue atenuar el efecto negativo que producen la ocurrencia de fallos técnicos e imprevistos en el tiempo global de la misión.

## 7.2 Líneas de trabajo futuro

La primera línea de continuación de este trabajo de investigación es la realización de pruebas reales que permitan verificar los resultados obtenidos en el entorno de simulación.

También, el algoritmo de planificación de rutas para la cobertura de un área en tiempo mínimo ha sido creado para trabajar con áreas poligonales libres de obstáculos, por lo que una continuación de esta línea de investigación sería adaptar dicho algoritmo para que pueda operar con áreas que posean obstáculos en su interior.

## Conclusiones y Trabajo Futuro

---

En cuanto a la capa cooperativa, hemos adoptado la condición de que en todo momento todos los elementos del sistema mantienen la comunicación entre sí. Una extensión de esta línea de investigación debería llevar a crear nuevas estrategias de cooperación teniendo en cuenta condiciones de comunicación intermitente.

Y finalmente, sería dotar al sistema multi-UAV aquí presentado de la funcionalidad de evasión de colisión, lo que contribuirá a elevar el nivel de seguridad durante la ejecución de las misiones de vuelo.

