

UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingenieros Industriales



Planificación de Rutas de Cobertura de Área
y Gestión Dinámica de Misiones para
Sistemas Multi-UAV

TESIS DOCTORAL

Presentada para optar al título de Doctor por:

Marco Andrés Luna Aguirre

Máster en Automática y Robótica

Madrid, 2024



UNIVERSIDAD POLITÉCNICA DE MADRID
Escuela Técnica Superior de Ingenieros Industriales

Doctorado en Automática y Robótica

Planificación de Rutas de Cobertura de Área y Gestión Dinámica de Misiones para Sistemas Multi-UAV

TESIS DOCTORAL

Presentada para optar al título de Doctor por:

Marco Andrés Luna Aguirre
Máster en Automática y Robótica

Bajo la dirección de:
Dr. Pascual Campoy Cervera
Dr. Martín Molina González

Madrid, 2024

Título: Planificación de Rutas de Cobertura de Área y Gestión Dinámica de Misiones para Sistemas Multi-UAV

Autor: Marco Andrés Luna Aguirre

Programa de Doctorado: Automática y Robótica

Dirección de Tesis:

Dr. Pascual Campoy Cervera, Catedrático, Escuela Técnica Superior de Ingenieros Industriales

Dr. Martin Molina González, Catedrático, Escuela Técnica Superior de Ingenieros Informáticos

Revisores Externos:

Tribunal de Tesis:

Fecha de Defensa de Tesis:

Esta tesis ha sido parcialmente financiada por el programa de "Ayudas para la realización de doctorados industriales de la Comunidad de Madrid, con número de referencia: IND2020/IND-17478

*A Francy mi amada e incondicional esposa,
A mis hijos Agustín, Eva y Joaquín por su amor,
A mis padres, suegros y hermanos por su apoyo,
A Jheanyne que nos cuida desde el cielo.*

Agradecimientos

Me gustaría expresar sinceramente mi gratitud a todas las personas que me han ayudado y apoyado durante la realización de esta tesis.

En primer lugar, quiero agradecer a Dios por darme la fortaleza, la sabiduría y la paciencia necesarias para completar este trabajo. Sin su guía y bendiciones, este logro no habría sido posible.

Quiero agradecer a la Comunidad de Madrid por el soporte económico a través del programa de Becas de Doctorados Industriales (número de referencia: IND2020/IND-17478). Esta financiación me permitió avanzar significativamente en mi trabajo y contribuir al desarrollo científico y tecnológico impulsado por la colaboración entre el ámbito académico y el industrial, contribuyendo al progreso tecnológico de esta preciosa región.

Expreso mis más sincero agradecimiento a mi director, el Prof. Pascual Campoy por su apoyo académico y personal. Su orientación y amplio conocimiento en temas de robótica aérea junto con la libertad que me brindó para desarrollar mi trabajo me han sido muy útiles para el desarrollo de mi investigación. Además, su constante estímulo y confianza en mi trabajo me motivaron a seguir adelante, incluso en los momentos más difíciles.

También quiero expresar mi agradecimiento a mi director el Prof. Martin Molina. Su inestimable guía, dedicación y esfuerzo han sido fundamentales para la realización de esta tesis. Agradezco profundamente sus consejos y su paciencia a lo largo de este proceso. Su experiencia en desarrollo algorítmico, sus conocimientos de programación y su disponibilidad en tiempo y recursos han hecho posible que este trabajo salga adelante.

Gracias también a mis compañeros del Grupo de Visión por Computador y Robótica Aérea: Sadeq, Miguel, Rodrigo, Javier, Pedro, David y Rafael. Su colaboración fue esencial para la realización de esta tesis. Agradezco las numerosas discusiones académicas que enriquecieron mi trabajo y las valiosas sugerencias que contribuyeron al progreso de mi investigación. Su ayuda práctica en el laboratorio, su disposición para resolver problemas y su compañerismo crearon un ambiente de trabajo positivo y productivo. De la misma manera, agradezco a Beatriz por su ayuda esencial en las diferentes tareas económicas y administrativas relacionadas al programa de doctorado.

Quisiera también expresar mi gratitud a las compañías Drone-Hopper y Ahyres, en las que tuve la oportunidad de realizar mi trabajo de investigación. Gracias por introducirme en el mundo real de los UAVs, y por facilitarme sus equipos y compartir su vasta experiencia para el desarrollo de mi investigación. Agradezco también a mis colegas dentro del sector: Antonio, Olaya, Juan Carlos, Julio, Gonzalo y Aitor. Sus consejos a nivel profesional y personal a lo largo de estos años han sido muy valiosos. Además, quiero destacar la amistad que hemos cultivado a lo largo de este tiempo. Los momentos de camaradería, las risas compartidas y el apoyo mutuo en los momentos de estrés han sido invaluable.

Quiero agradecer de corazón a mis hijos Agustín, Eva y Joaquín por su comprensión y coraje durante los momentos más difíciles. A mis padres Marco y Mercedes, y mis suegros Wilman

y Patricia, por haber sido una fuente de inspiración. Sus constantes muestras de cariño y aliento fueron fundamentales para mantenerme enfocado y motivado a lo largo de este arduo camino. A mis hermanos David, Froilán y Juan por animarme de manera constante. Al resto de mi familia, por su ayuda, sus oraciones y buenos deseos que me dieron la fuerza para seguir adelante. Mi gratitud también a aquellos amigos que, aunque no están directamente involucrados en el ámbito académico, siempre estuvieron dispuestos a escuchar y ofrecer su apoyo incondicional.

Finalmente, quiero agradecer a mi esposa Francy, por su amor incondicional y por creer siempre en mí. Su apoyo constante en las decisiones más difíciles ha sido muy importante, gran parte de este trabajo le pertenece. Quiero recalcar que sin su esfuerzo y sacrificio esta tesis no habría sido posible, por lo cual, estoy infinitamente agradecido.

Abstract

In recent years, advancements in unmanned aerial systems (UAS) have significantly improved outdoor navigation, the integration of specialized payloads, and the ability to operate multiple vehicles simultaneously, resulting in their frequent utilization within industrial settings. Research areas such as area coverage path planning have garnered significant interest from the scientific community due to their numerous applications in precision agriculture, search and rescue operations, environmental monitoring, and other fields. Within this context, this doctoral thesis addresses the advanced management of multi-UAV systems in area coverage missions, presenting innovative advancements and techniques developed in three key stages: route planning, mission distribution among multiple UAVs, and dynamic in-flight system management with replanning capabilities.

Initially, the thesis introduces an original proposal for single spiral area coverage, which aims to reduce energy consumption and flight time associated with turns during navigation. This is achieved through an approach based on affine transformation matrices and clothoids. Subsequently, two novel solutions for efficient coverage of multiple areas are proposed: the first solution is based on the traveling salesman problem, while the second employs nearest neighbor techniques.

Furthermore, the thesis proposes two approaches for route distribution among multiple UAVs, tailored for homogeneous systems (drones with similar characteristics) and heterogeneous systems (drones with varying characteristics). These methods enhance efficiency in coverage tasks by minimizing flight time and mission processing time compared to existing techniques in the literature. This is accomplished through a combination of heuristic methods and optimization techniques to ensure efficient workload distribution among the drones. For homogeneous systems, the original Powell-BINPAT method is proposed, which performs task distribution analogous to the bin-packing problem, combined with Powell's optimization technique. For heterogeneous systems, a modified approach called CoP-BINPAT is introduced, featuring a novel CoP metric to evaluate each UAV's performance in area coverage tasks, thereby enabling efficient route distribution.

Lastly, the thesis presents a novel real-time monitoring architecture designed to supervise the status of UAVs and the ongoing mission. This architecture collects critical information for real-time analysis to detect potential failures or deviations from the pre-established flight plan. In cases of unplanned behaviors, an in-flight replanning module is activated to dynamically adjust the routes and tasks of the remaining drones.

The results, obtained through performance evaluation tests, simulation software, and real-world experiments in various operational scenarios, demonstrate the computational efficiency in task planning, mission cost minimization, and the system's rapid response to unplanned events when compared to other proposals in the literature.

Resumen

En los últimos años, los avances en los sistemas aéreos no tripulados (UAS) han mejorado significativamente la navegación en exteriores, la integración de cargas útiles especializadas y la capacidad de operar múltiples vehículos simultáneamente, resultando en su uso frecuente en entornos industriales. Áreas de investigación como la planificación de rutas de cobertura de áreas han captado un gran interés por parte de la comunidad científica debido a sus numerosas aplicaciones en la agricultura de precisión, operaciones de búsqueda y rescate, monitoreo ambiental y otros campos. En este contexto, esta tesis doctoral aborda la gestión avanzada de sistemas multi-UAV en misiones de cobertura de áreas, presentando avances y técnicas innovadoras desarrolladas en tres etapas clave: planificación de rutas, distribución de misiones entre múltiples UAVs y gestión dinámica en vuelo del sistema con capacidades de replanteamiento.

Inicialmente, la tesis introduce una propuesta original para la cobertura de áreas en espiral única, que tiene como objetivo reducir el consumo de energía y el tiempo de vuelo asociados con los giros durante la navegación. Esto se logra mediante un enfoque basado en matrices de transformación afines y clotoides. Posteriormente, se proponen dos soluciones novedosas para la cobertura eficiente de múltiples áreas: la primera solución se basa en el problema del viajante, mientras que la segunda emplea técnicas de vecinos cercanos.

Además, la tesis propone dos enfoques para la distribución de rutas entre múltiples UAVs, adaptados a sistemas homogéneos (drones con características similares) y sistemas heterogéneos (drones con características variadas). Estos métodos mejoran la eficiencia en las tareas de cobertura al minimizar el tiempo de vuelo y el tiempo de procesamiento de la misión en comparación con las técnicas existentes en la literatura. Esto se logra mediante una combinación de métodos heurísticos y técnicas de optimización para asegurar una distribución eficiente de la carga de trabajo entre los drones. Para sistemas homogéneos, se propone el método original Powell-BINPAT, que realiza la distribución de tareas de manera análoga al problema de bin-packing, combinado con la técnica de optimización de Powell. Para sistemas heterogéneos, se introduce un enfoque modificado llamado CoP-BINPAT, que presenta una métrica novedosa, CoP, para evaluar el rendimiento de cada UAV en tareas de cobertura de áreas, permitiendo así una distribución eficiente de las rutas.

Por último, la tesis presenta una arquitectura novedosa de monitoreo en tiempo real, diseñada para supervisar el estado de los UAVs y la misión en curso. Esta arquitectura recopila información crítica para su análisis en tiempo real con el fin de detectar posibles fallos o desviaciones del plan de vuelo preestablecido. En casos de comportamientos no planificados, se activa un módulo de replanificación en vuelo para ajustar dinámicamente las rutas y tareas de los drones restantes.

Los resultados, obtenidos mediante pruebas de evaluación de rendimiento, software de simulación y experimentos en escenarios operativos reales, demuestran la eficiencia computacional en la planificación de tareas, la minimización del costo de las misiones y la rápida respuesta del sistema frente a eventos no planificados en comparación con otras propuestas de la literatura.

Tabla de Contenido

Agradecimientos	v
Abstract	vii
Resumen	viii
Lista de Figuras	xi
Lista de Tablas	xv
Abreviaturas y acrónimos	xix
1 Introducción	1
1.1 Motivación de la tesis	2
1.2 Planteamiento del problema	3
1.3 Objetivos de la tesis	4
1.3.1 Objetivo principal	4
1.3.2 Objetivos específicos	4
1.4 Metodología	4
1.5 Esquema de la tesis	5
2 Estado del arte	7
2.1 Planificación de rutas de cobertura	7
2.2 Sistemas multi-UAV	9
2.2.1 Sistemas multi-UAV homogéneos	10
2.2.2 Sistemas multi-UAV heterogéneos	13
2.3 Gestión dinámica y replanificación de la misión	16
3 Planificación de trayectorias de cobertura de área para sistemas de UAV único	19
3.1 Introducción	19
3.2 Planificación en área única	21
3.2.1 Planteamiento del escenario	21
3.2.2 Algoritmo de cobertura de área en espiral	21
3.3 Planificación multi-área	25
3.3.1 Planteamiento del escenario	25
3.3.2 Cobertura multi-área basada en TSP	26
3.3.3 Cobertura multi-área según vecino más cercano	28
4 Distribución de trayectorias en sistemas multi-UAV	31

4.1	Introducción	31
4.2	Distribución de la trayectoria en sistemas multi-UAV homogéneos	34
4.2.1	Planteamiento del escenario	34
4.2.2	Algoritmo Powell-BINPAT	36
4.3	Distribución de la trayectoria en sistemas multi-UAV heterogéneos	38
4.3.1	Planteamiento del escenario	38
4.3.2	Algoritmo CoP-BINPAT	39
5	Gestión dinámica de trayectorias en sistemas multi-UAV	43
5.1	Introducción	43
5.2	Escenario planteado	45
5.3	Arquitectura de control propuesta	45
5.3.1	Planificación multi-UAV	46
5.3.2	Monitorización de ejecución	47
5.3.3	Planificación local	49
5.3.4	Replanificación en vuelo	50
5.3.5	Control de ejecución	52
6	Experimentos y discusión	53
6.1	Configuración de los experimentos	53
6.1.1	Experimentos de evaluación del rendimiento	53
6.1.2	Experimentos de simulación	54
6.1.3	Experimentos reales	56
6.2	Planificación de trayectorias de cobertura de área para sistemas de UAV único	59
6.2.1	Evaluación de rendimiento de los algoritmos	59
6.2.2	Resultados en simulación	63
6.2.3	Discusión	63
6.3	Algoritmo Powell-BINPAT	64
6.3.1	Diseño del experimento y métricas de evaluación	64
6.3.2	Evaluación de rendimiento del método propuesto	64
6.3.3	Pruebas en simulación	69
6.3.4	Pruebas en entornos reales	70
6.3.5	Discusión	72
6.4	Algoritmo CoP-BINPAT	72
6.4.1	Diseño del experimento y métricas de evaluación	72
6.4.2	Análisis de rendimiento del método propuesto	72
6.4.3	Resultados en simulación	78
6.4.4	Discusión	81
6.5	Gestión dinámica de trayectorias en sistemas multi-UAV	82
6.5.1	Diseño de experimentos y métricas de evaluación	82
6.5.2	Pruebas de simulación	82
6.5.3	Pruebas en entornos reales	85
6.5.4	Discusión	87
6.6	Discusión general	87

7	Conclusiones y trabajos futuros	89
7.1	Conclusiones	89
7.1.1	Generación de trayectorias de cobertura de área	89
7.1.2	Distribución de trayectorias en sistemas multi-UAV	89
7.1.3	Gestión dinámica de las trayectorias en sistemas multi-UAV	90
7.2	Trabajos futuros	90
	Referencias	93
	Anexos	103
A	Diseminación científica	103
A.1	Publicaciones	103
A.2	Proyectos de investigación	104
A.3	Transferencia tecnológica en el sector empresarial	107
B	Tablas de resultados algoritmo Powell-BINPAT	108
B.1	Resultados de costo en distancia para pruebas entre 2-10 drones . . .	108
B.2	Resultados de tiempo de vuelo en simulación del algoritmo Powell-BINPAT para múltiples pruebas en una sola área con algoritmo de planificación back-and-forth	111
C	Tablas de resultados algoritmo CoP-BINPAT	112
C.1	Resultados algoritmo CoP-BINPAT y planificación back and forth . .	112
C.2	Resultados algoritmo CoP-BINPAT planificación espiral	115
C.3	Pruebas de tiempo restante de vuelo	118
D	Resultados gráficos adicionales planificación dinámica	119

Lista de Figuras

2.1	Tipos de planificación de cobertura en 2D. a) Ejemplo de un método de descomposición. b) Ejemplo de un método de no descomposición.	8
2.2	Esquema tomado de (Luna et al., 2021) para ilustrar el funcionamiento de los algoritmos de asignación de tareas en sistemas homogéneos el el cual el número de UAVs disponibles se reparte entre las tareas de acuerdo al costo de ejecución de la misión.	10
2.3	Esquema control de formación en múltiples UAVs. a) Líder seguidor. b) Estructura virtual. c) Basado en comportamiento. d) Basado en consenso . . .	11
2.4	Esquema de los algoritmos de planificación en sistemas multi-UAV homogéneos. a) Planificación 3D estática. b) Planificación dinámica. c) Planificación de cobertura de áreas	12
2.5	Esquema de un sistema de control de formación líder-seguidor compuesto por UAVs heterogéneos.	13
2.6	Esquema de un sistema de asignación de tareas a UAVs heterogéneos. El tamaño de los círculos grises representa el costo de la tarea, y los UAVs son asignados de eficientemente de acuerdo a sus capacidades.	14
2.7	Esquema de un sistema de planificación de trayectorias con UAVs heterogéneos. Las rutas que optimizan el costo de la misión se generan de acuerdo con las capacidades de cada aeronave a partir de un conjunto desordenado de waypoints	15
2.8	Esquema de arquitecturas de control en sistemas multi-UAV. a) Centralizado. b) Descentralizado. c) Distribuido	16
3.1	Esquema de rutas generadas para ala rotativa y ala fija. Los primeros se pueden configurar para seguir una trayectoria sin realizar cambios de giro volando incluso hacia atrás	20
3.2	Esquema del problema de cobertura de área única.	22
3.3	Esquema de cálculo de sentido de giro de la espiral para el caso horario (CW)	23
3.4	Esquema de cálculo de sentido de giro de la espiral para el caso anti-horario (CCW)	23
3.5	Esquema de generación de puntos adicionales en la espiral. (a) Puntos originales del polígono. (b) Generación del primer punto de soporte. (c) Primera iteración de escalado. (d) Generación del segundo punto de soporte	25
3.6	Esquema del problema de cobertura de múltiples áreas	26

3.7	Esquema de la primera etapa del algoritmo de cobertura basado en TSP. a) Cálculo del centroide de las áreas. b) Resolución del TSP basada en los centroides y la posición inicial del UAV	27
3.8	Esquema de cobertura de área basado en región vecina más cercana	28
4.1	Figura tomada de (Y. Tan y Zheng, 2013) para ilustrar el comportamientos de inteligencia de enjambre de la naturaleza como las colonias de hormigas, bandadas de aves, cardúmenes de peces o jaurías de caza	32
4.2	Figura tomada de (Martinez-Rozas et al., 2021) para ilustrar uno de los retos formulados en la competición MBZIRC 2020 que proponía una colaboración UAV-UGV para extinción de incendios en un entorno urbano con GPS denegado.	33
4.3	Esquema de un sistema compuesto por UAVs heterogéneos que incluye comunicación con la estación de tierra y enlaces satelitales.	34
4.4	Esquema de la huella del sensor para un sistema homogéneo	35
4.5	Esquema de alturas de seguridad previas a la ejecución de la misión	35
4.6	Esquema de método de planificación back-and-forth para multi-UAVs basado en el centroide del polígono formado por la posición de los UAVs	36
4.7	Esquema del problema de cobertura de área con aeronaves heterogéneas. Cada UAV tiene una velocidad v , tiempo de suministro de energía t y área de la huella del sensor s diferente, con un porcentaje de superposición. Los puntos marcados con \otimes corresponden a los waypoints de cada aeronave	39
5.1	Figura tomada de (Peña et al., 2022) para demostrar un registro de rutas de vuelo de un sistema de UAV único con capacidad de replanificación dinámica de la misión	44
5.2	Arquitectura de control propuesta para planificación dinámica. Los rectángulos representan un proceso de información mientras que las flechas describen flujos de información.	46
5.3	Esquema de replanificación en vuelo. a) Los UAVs ejecutan la ruta normalmente. b) Pérdida del UAV morado en un momento determinado de la misión. c) Replanificación de la nueva ruta global. d) Distribución homogénea de tareas con el algoritmo Powell-BINPAT	51
6.1	Opciones de SITL de ardupilot en Mission Planner	55
6.2	Imagen del Robotics CAR Arena con tres UAVs en vuelo	57
6.3	Figura tomada de (Luna et al., 2022) para mostrar el esquema de configuración del prototipo SwarmHopper utilizado en pruebas reales en el aeródromo	58
6.4	Plataformas aéreas utilizadas en experimentos reales dentro del Robotics CAR Arena	58
6.5	Trayectorias de planificación espiral para UAV único	60
6.6	Comparación de costo de la misión en distancias del método de vecino más cercano (NN) y problema del viajante (TSP)	62
6.7	Comparación de tiempos de ejecución del método de vecino más cercano (NN) y problema del viajante (TSP)	62
6.8	Configuración del entorno SITL para Mission Planner. a) Planificación back-and-forth. b) Planificación espiral	63

6.9	Resultados gráficos del algoritmo Powell-BINPAT con planificación de trayectorias back-and-forth	65
6.10	Resultados gráficos del algoritmo Powell-BINPAT con planificación de trayectorias espiral	66
6.11	Resultados gráficos de comparación del algoritmo Powell-BINPAT con propuestas de la literatura. a) Propuesta presentada por (Avellar et al., 2015). b) Resultado Powell-BINPAT. c) Propuesta presentada por (Hong et al., 2021). d) Resultado Powell-BINPAT	67
6.12	Resultados de tiempo de CPU del algoritmo Powell-BINPAT para múltiples UAVs y múltiples regiones	68
6.13	Capturas de pantalla de la estación de control de tierra. (a) Planificación back-and-forth. (b) Planificación espiral	70
6.14	Pruebas del algoritmo Powell-BINPAT en entornos reales	70
6.15	Datos de trayectorias del algoritmo Powell-BINPAT obtenidos de datalog de vuelo	71
6.16	Tiempos de CPU del método CoP BINPAT, evaluados con diferentes números de UAVs y con los métodos de cobertura back-and-forth y espiral	73
6.17	Trayectorias generadas por el algoritmo CoP-BINPAT para 5 UAVs con el patrón back-and-forth	75
6.18	Trayectorias generadas por el algoritmo CoP-BINPAT para 5 UAVs con el patrón espiral	76
6.19	Resultados de coeficiente de variación para múltiples UAVs utilizando el algoritmo CoP-BINPAT	77
6.20	Visualización de la simulación en RViz	78
6.21	Resultados de simulación para 5 UAVs	80
6.22	Entorno de simulación 3D desarrollado en Flightmare para evaluación del sistema	82
6.23	Simulación de 3 UAVs en el entorno Flightmare para inspección de plantas fotovoltaicas repartidas en dos áreas, con visualización de las cámaras	83
6.24	Resultados de trayectorias recorridas para 3 UAVs con replanificación en vuelo	85
6.25	Trayectorias recorridas por los UAVs en un vuelo real con un evento de replanificación	86
A.1	Prueba de vuelo de un quadrotor de alta capacidad de carga con motores de combustión con control de paso variable	105
A.2	Interfaz del software de estación de control en tierra para operación de enjambres	106
A.3	Interfaz del software de estación de control en tierra para operación de enjambres	107
D.4	Trayectorias recorridas por los UAVs en pruebas de simulación de vuelo continuo sin fallos	119
D.5	Trayectorias recorridas por los UAVs en pruebas de simulación de vuelo con un fallo y una replanificación de los UAVs restantes	119
D.6	Trayectorias recorridas por los UAVs en pruebas de simulación de vuelo con dos fallos y dos replanificaciones de los UAVs restantes	120
D.7	Trayectorias recorridas por los UAVs en pruebas reales con dos fallos y replanificación de los dos UAVs restantes	120

Lista de Tablas

5.1	Requerimientos algorítmicos de la planificación Multi-UAV para el sistema dinámico	46
5.2	Eventos de replanificación	48
5.3	Ejemplos de acciones ejecutadas por el módulo de planificación local	50
6.1	Rendimiento del algoritmo espiral para un único UAV y una sola área en diferentes pruebas	59
6.2	Comparativa de métodos de planificación en espiral presentados en la literatura	60
6.3	Parámetros de la misión para evaluación de rendimiento de los algoritmos de planificación en múltiples áreas	61
6.4	Resultados de simulación de vuelo para planificación back-and-forth y espiral	63
6.5	Parámetros estandarizados de cada misión para evaluar el rendimiento del algoritmo	64
6.6	Rendimiento del algoritmo Powell-BINPAT utilizando la planificación de trayectorias espiral y back-and-forth para 2-10 drones	65
6.7	Resultados de comparación del CV en diferentes métodos presentados en la literatura	66
6.8	Resultados de comparación del tiempo de CPU en diferentes métodos presentados en la literatura para el algoritmo Powell-BINPAT	69
6.9	Resultados de tiempo de ejecución de la misión en segundos para simulación SITL con el algoritmo Powell-BINPAT	69
6.10	Resultados de tiempo de ejecución de la misión en segundos para prueba en entornos reales	71
6.11	Parámetros de la misión para experimentos de medición de tiempo de CPU .	73
6.12	Resultados de comparación del tiempo de CPU en diferentes métodos presentados en la literatura para el algoritmo CoP-BINPAT	74
6.13	Rendimiento del algoritmo CoP-BINPAT con planificación back-and-forth y 5 UAVs	75
6.14	Rendimiento del algoritmo CoP-BINPAT con planificación espiral y 5 UAVs	76
6.15	Resultados de pruebas de batería restante para múltiples UAVs	78
6.16	Tipos de UAVs utilizados en las pruebas de simulación	79
6.17	Número de UAVs de cada tipo utilizado en los diferentes escenarios de simulación	79
6.18	Resultados de coeficiente de variación para los tiempos de misión obtenidos en simulación en los diferentes escenarios de prueba	79

6.19	Escenarios propuestos para medir el rendimiento del algoritmo en tiempos de batería disponible	80
6.20	Resultados de batería restante en el entorno de simulación	81
6.21	Resultados de distancia reportados en la simulación para 3-5 UAVs y 2-4 áreas en vuelos continuos	83
6.22	Resultados de distancia reportados en la simulación para 3-5 UAVs con eventos de replanificación en vuelo	84
6.23	Tiempos de CPU para pruebas de simulación con 3-5 UAV y tres áreas de exploración	84
6.24	Resultados de distancia recorrida para 3 UAVs en las pruebas de vuelo reales	86
6.25	Tiempos de CPU para 3 UAVs en las pruebas de vuelo reales	86
6.26	Resultados de comparación del tiempo de CPU de la arquitectura propuesta con la propuesta por Hong et al., 2022	87

Abreviaturas y acrónimos

- 2D** Bidimensional
- 3D** Tridimensional
- ACO** Ant colony optimization (Optimización de colonia de hormigas)
- Av** Average (Promedio)
- B&F** Back-and-forth (Ida y vuelta)
- BINPAT** Bin Packing problem based Path Planner (Planificador de trayectorias basado en el problema de la mochila)
- CBBA** Consensus-based bundle algorithm (Algoritmo de paquete basado en consenso)
- CCW** Counter-clockwise (En sentido anti-horario)
- CoP** Coverage Flight Performance (Desempeño en vuelo de cobertura)
- CPP** Coverage Path Planning (Planificación de rutas de cobertura)
- CPU** Central Processing Unit (Unidad de procesamiento central)
- CV** Coeficiente de Variación
- CW** Clockwise (En sentido horario)
- DARPA** Defense Advanced Research Projects Agency
- DP** Dynamic Programming (Programación Dinámica)
- DRL** Deep Reinforcement Learning (Aprendizaje profundo por refuerzo)
- ERP** Enterprise Resource Planning (Planificación de recursos empresariales)
- FOV** Field of View (Campo de visión)
- GA** Genetic Algorithm (Algoritmo genético)
- GHz** Gigahercios
- Gb** Gigabyte
- GNSS** Global navigation satellite system (Sistema de navegación global por satélite)
- GSA** Gravitation Search Algorithm (Algoritmo de búsqueda por gravitación)
- GUI** Graphical User Interface (Interfaz gráfica de usuario)

LSO Life-cycle swarm optimization (Optimización de ciclo de vida de enjambre)

MBZIRC Mohamed Bin Zayed International Robotics Challenge

MILP Mixed Integer Linear Programming (Programación lineal de enteros mixtos)

MIQP Mixed Integer Quadratic Programming (Programación cuadrática de enteros mixtos)

MRTA Multi-robot task allocation (Asignación de tareas multi-robot)

NN Nearest neighbor (Vecino más cercano)

NP Nondeterministic polynomial time (Tiempo polinomial no determinístico)

ROS Robot Operating System (Sistema operativo robótico)

SD Standard deviation (Desviación estándar)

SITL Software in the loop (Software en el bucle)

TSP Travel Salesman Problem (Problema del viajante)

UAS Unmanned aerial systems (Sistemas aéreos no tripulados)

UAV Unmanned aerial vehicle (Vehículos aéreos no tripulados)

UGV Unmanned ground vehicle (Vehículo terrestre no tripulado)

USV Unmanned surface vehicle (Vehículo de superficie no tripulado)

UUV Unmanned underwater vehicle (Vehículo submarino no tripulado)

UV Unmanned vehicle (Vehículo no tripulado)

VRP Vehicle routing Problem (Problema de enrutado de vehículos)

VTOL Vertical Take-Off Landing (Despegue y aterrizaje vertical)

Capítulo 1

Introducción

En la última década, el uso de vehículos aéreos no tripulados (UAVs por sus siglas en inglés) ha experimentado un crecimiento significativo en una amplia gama de aplicaciones que incluyen la vigilancia, búsqueda y rescate, monitoreo, agricultura de precisión y otras. Los UAVs ofrecen varias capacidades como la ejecución de misiones en áreas de difícil acceso, operaciones en entornos peligrosos y la realización de tareas repetitivas con alta precisión. Además, en la naturaleza, los modelos de conducta colaborativa en la ejecución de tareas ayudan a los seres vivos a compensar sus deficiencias individuales, por lo cual, los sistemas multi-robot han llamado la atención de la comunidad científica, especialmente en el campo de la robótica aérea (Zhou et al., 2020). En este contexto, se han identificado en la literatura múltiples ventajas de los sistemas multi-UAV con respecto a los sistemas de una aeronave incluyendo la flexibilidad, escalabilidad, resiliencia y tolerancia a fallos (Chung et al., 2018). Así, a medida que las misiones se vuelven más complejas y abarcan mayores áreas geográficas, se hace necesario el uso de estos sistemas para asegurar la eficiencia y efectividad de las operaciones (Skorobogatov et al., 2020). De esta manera, los sistemas multi-UAV presentan una serie de desafíos técnicos y operativos en diferentes niveles de su arquitectura de control que incluyen la toma de decisiones, planificación de rutas, control de bajo nivel y comunicaciones (Boskovic et al., 2009, Zhou et al., 2020).

La planificación de rutas de cobertura (CPP por sus siglas en inglés) y la distribución equitativa de tareas, son esenciales para maximizar el uso de los recursos disponibles y garantizar el éxito de la misión. Para esto, es necesario disponer de métodos que minimicen el tiempo de operación considerando la naturaleza de las misiones y el entorno (Fevgas et al., 2022). Además, en sistemas heterogéneos, donde los UAVs pueden tener diferentes capacidades y características, estos desafíos se vuelven aún más complejos.

En la capa de toma de decisiones, la monitorización en tiempo real del estado de los UAVs y del progreso de la misión es otro aspecto crítico que impacta directamente en la seguridad y la eficiencia operativa (Sanchez-Lopez et al., 2016). Un sistema de monitorización robusto permite la detección temprana de problemas y facilita la toma de decisiones informadas, mejorando su resiliencia y adaptabilidad. La capacidad de replanificación en vuelo es también crucial, permitiendo al sistema adaptarse dinámicamente a cambios imprevistos, como el fallo de uno o varios UAVs.

Esta tesis se enfoca en abordar estos desafíos mediante el desarrollo de técnicas novedosas para la generación eficiente de rutas de cobertura de múltiples áreas, la distribución de tareas en sistemas homogéneos y heterogéneos, y la gestión dinámica del sistema con capacidad de replanificación de trayectorias frente a eventos inesperados. El enfoque integral que se desarrolla a lo largo de la tesis, une a todas las partes de manera que las propuestas presentadas durante la fase de planificación de rutas de cobertura se utilizan como base en el desarrollo de métodos de distribución homogénea y heterogénea de los planes de vuelo. De la misma manera, el sistema de gestión dinámica de trayectorias recopila las técnicas de planificación y distribución en una arquitectura con métodos propios para la detección de eventos anómalos y replanificación en vuelo.

1.1 Motivación de la tesis

La investigación actual en técnicas de generación de rutas de cobertura en sistemas dinámicos está contribuyendo al avance del conocimiento científico en varios campos de la robótica y la inteligencia artificial que requieren la coordinación de múltiples agentes autónomos. Específicamente, en el campo de la robótica aérea, el impacto potencial de esta tecnología abarca diferentes sectores industriales. En el ámbito de la seguridad y defensa, este tipo de técnicas se aplican en misiones de vigilancia, reconocimiento, búsqueda y rescate, y evaluación de los daños causados por desastres naturales para apoyo en las tareas de socorro. En el sector de la agricultura se están utilizando para la monitorización de cultivos, detección de plagas y enfermedades, y aplicación de pesticidas y fertilizantes de manera precisa. Otras aplicaciones industriales incluyen la inspección de infraestructuras y la cartografía.

En el entorno empresarial, se planteó esta propuesta en base a las necesidades de las compañías del sector aeroespacial Drone-Hopper y Ahyres en las cuales se ha desarrollado esta investigación (ver Anexo A). Empresas como estas requieren implementar estas capacidades en sus prototipos y aeronaves comerciales actuales para afrontar proyectos de investigación en el marco nacional y europeo y mejorar su propuesta de mercado con respecto a sus clientes. El desarrollo e implementación de estos métodos supone una ventaja competitiva considerando las múltiples aplicaciones ya mencionadas de esta tecnología.

Sin embargo, la gestión eficiente de estos sistemas es un reto complejo, especialmente cuando se trata de misiones que requieren la cobertura múltiples áreas en las cuales los drones son susceptibles a fallo. En este contexto, nos podemos encontrar algunos problemas en los algoritmos del estado del arte actuales como el alto costo computacional, la distribución ineficiente de las tareas o la incapacidad de reacción automática frente a fallos.

De esta manera, un método de alto costo computacional causa que el sistema se vea limitado en cuanto a su escalabilidad, ya que los tiempos de procesamiento pueden incrementar considerablemente con el aumento de UAVs al sistema. Por otra parte, la distribución ineficiente de las tareas provoca que se utilicen más recursos de los necesarios durante la ejecución de la misión. La incapacidad de reacción tendrá como consecuencia la repetición de toda la operación frente a eventos inesperados, lo cual a nivel industrial representa costos elevados en tiempo y dinero. Por este motivo, la presente investigación buscará mejorar los métodos actuales abordando cada uno de estos problemas con soluciones novedosas que realicen las

tareas de planificación, distribución y gestión dinámica de manera automática.

Así, en la tarea de planificación de rutas de cobertura se busca desarrollar métodos que contribuyan a minimizar el tiempo de toda la misión al reducir las distancias recorridas. En la fase de distribución de tareas se requiere minimizar la variación de los costos de la misión entre los UAVs que participan de la operación; y en la fase de gestión dinámica se necesita implementar una arquitectura de control capaz de reaccionar frente a cambios imprevistos y replanificar la misión para todo el enjambre. A nivel general, se buscará obtener tiempos de procesamiento eficientes en todos los algoritmos para minimizar el tiempo de reacción en la arquitectura de gestión dinámica y mantener una alta escalabilidad.

En resumen, esta tesis se motiva por las aplicaciones potenciales de esta tecnología y las necesidades reales del sector industrial reflejadas en dos compañías. De esta manera, nuestro enfoque intenta mejorar los tiempos de procesamiento, la distribución desigual de las tareas entre los UAVs, y la baja capacidad de reacción de los sistemas actuales de la literatura. Para esto, atacamos el problema en tres fases: planificación de rutas de cobertura, distribución de tareas entre los UAVs del sistema y gestión dinámica de la operación.

1.2 Planteamiento del problema

En base a la motivación presentada en la sección anterior, se puede mencionar que uno de los problemas principales de este estudio radica en la necesidad de desarrollar algoritmos que permitan la generación de rutas eficientes que garanticen la cobertura completa de un área, y su distribución equitativa para un conjunto de UAVs homogéneos y heterogéneos. En la siguiente fase, es crucial considerar la capacidad de estos sistemas para adaptarse dinámicamente a cambios inesperados, como la falla de uno o más aeronaves durante la misión. La ausencia de un UAV podría dejar áreas sin cubrir o requerir que otros UAVs asuman tareas adicionales realizadas en una misión posterior, lo cual podría afectar el rendimiento de la misión y provocar el consumo innecesario de recursos.

Por lo tanto, el problema central de esta tesis se enfoca en tres aspectos fundamentales:

1. **Necesidad de técnicas de generación de rutas de cobertura:** Se necesita disponer de algoritmos que permitan planificar rutas eficientes que garanticen una cobertura completa de múltiples áreas objetivo, minimizando el solapamiento y maximizando el uso de los recursos disponibles.
2. **Necesidad de métodos de distribución de tareas en sistemas homogéneos y heterogéneos:** Es necesario también desarrollar métodos de asignación de trayectorias que distribuyan de manera equitativa la tarea global independientemente del tipo de sistema, y que minimicen el tiempo de ejecución de la misión; por lo tanto, se requiere que estas técnicas tengan un bajo costo computacional para su aplicación en tiempo real durante la replanificación.
3. **Gestión dinámica del sistema y capacidad de replanificación en Vuelo:** Es necesario disponer de mecanismos que permitan monitorizar constantemente a los drones del sistema y el estado de la misión. En caso de fallo, se requiere la replanificación

dinámica sobre los puntos que no se han cubierto, redistribuyendo eficientemente las nuevas tareas entre los drones restantes para asegurar la continuidad y efectividad de la misión.

La solución propuesta en esta tesis trata de dar respuestas a estos problemas mediante el desarrollo de algoritmos avanzados de generación de trayectorias, planificación y replanificación de rutas que puedan ser implementados en tiempo real, mejorando así la resiliencia y eficiencia de los sistemas multi-UAV en operaciones críticas y en entornos potencialmente adversos.

1.3 Objetivos de la tesis

1.3.1 Objetivo principal

El objetivo central del trabajo de investigación de esta tesis doctoral es desarrollar métodos novedosos que mejoren el rendimiento de las propuestas actuales de la literatura en las tareas de planificación automática de rutas de cobertura y gestión dinámica de la misión para sistemas Multi-UAV, reduciendo el tiempo computacional, mejorando la eficiencia en la distribución de tareas y minimizando el tiempo de reacción de estos sistemas frente a entornos dinámicos complejos, con el fin de operar de manera eficiente en tiempo real y que estos métodos puedan ser utilizados en aplicaciones industriales reales.

1.3.2 Objetivos específicos

1. Proponer una solución computacional para la generación de rutas de cobertura eficientes que sean aplicables a un área única y a conjuntos de áreas separadas para un único UAV.
2. Estudiar y plantear métodos de distribución eficiente de las tareas en sistemas de múltiples UAVs homogéneos y heterogéneos en base a las capacidades individuales de cada uno.
3. Desarrollar un modelo de arquitectura de software para monitorizar el estado de la misión y de los UAVs con capacidad de detección automática de fallos y replanificación en vuelo frente a eventos no anticipados.
4. Evaluar las propuestas mediante la construcción de sistemas multi-UAV y su experimentación en entornos reales y simulados; comparando los resultados obtenidos con otros enfoques descritos en la literatura y comprobando su validez para su uso en aplicaciones industriales.

1.4 Metodología

La metodología que siguió el desarrollo de esta tesis se basa en el enfoque *Divide y Vencerás* que se basa en descomponer un problema complejo en subproblemas más pequeños y manejables, resolver cada subproblema de manera independiente y ordenada y luego integrar

las soluciones para abordar el problema global. La descomposición del problema facilita su análisis y resolución, permitiendo abordar la complejidad de manera sistemática y estructurada. Cabe recalcar que para llevar a cabo todas estas tareas, se contó además con la ayuda de un equipo multidisciplinario dentro del grupo de investigación y el entorno empresarial. A continuación, se presenta un resumen de las etapas:

1. **Planteamiento del problema global:** En esta primera etapa, se plantea el problema global en base a las necesidades del entorno industrial, sin embargo, el desafío ha sido auto-realimentado internamente en base a necesidades científicas, tecnológicas y operacionales.
2. **Descomposición en subproblemas:** Una vez que han finalizado las iteraciones anteriores y se ha definido el problema global, se definen formalmente los subproblemas en forma de hitos y tareas secuenciales a realizar
3. **Revisión de la literatura:** Se realiza un estudio y análisis de las soluciones actuales por cada subproblema de investigación, así como las ventajas y desventajas de cada propuesta. También se realizan pruebas de código libre disponible en repositorios para evaluación y comparación de resultados.
4. **Resolución de subproblemas:** En base al desafío propuesto y la revisión del estado del arte, se procede al desarrollo de las soluciones computacionales. En este punto, se busca que dichas soluciones sean novedosas y que aporten ventajas con respecto a las técnicas del estado del arte.
5. **Evaluación individual:** Se prueba de manera exhaustiva las soluciones individuales propuestas mediante tres tipos de pruebas: pruebas de escritorio, con software de simulación e implementación en entornos reales. Basándose en los resultados, se decide si es necesario regresar a la etapa de desarrollo para refinar los métodos propuestos. Esta etapa termina cuando se ha determinado que los resultados cumplen los objetivos de los subproblemas.
6. **Integración:** Si los resultados han cumplido con los objetivos, se integran las propuestas anteriores de tal manera que la propuesta resultante sea una solución al problema global. Si para uno de los subproblemas se ha encontrado más de una solución, en esta etapa se define qué solución se debe integrar para resolver el problema global.
7. **Evaluación global:** Se prueba la integración de la solución en entornos de simulación y entornos reales. En base a los resultados, se decide si es necesario, regresar a la etapa de integración para cambiar corregir errores o cambiar de estrategias de resolución del problema. En caso de que los resultados sean satisfactorios en esta etapa, se implementan como aplicaciones amigables para el usuario. También en esta etapa se definen los posibles trabajos futuros en base a nuevos desafíos identificados.

1.5 Esquema de la tesis

El presente documento muestra la evolución del trabajo de investigación de la presente tesis doctoral dividida en en siete capítulos y cuatro apéndices:

- El capítulo 1, que se corresponde con el presente, realiza una introducción al campo de la planificación de rutas de cobertura con sistemas multi-UAV, describe la motivación detrás de la tesis y realiza una descripción general de los subproblemas que componen el problema global de investigación.
- El capítulo 2 realiza una revisión del estado del arte relacionado con la tesis.
- El capítulo 3 describe las propuestas originales de planificación de trayectorias de cobertura de área propuestas considerando un solo UAV.
- El capítulo 4 extiende la planificación de trayectorias del capítulo anterior a sistemas con múltiples UAVs, presentando soluciones novedosas para la distribución de las trayectorias en sistemas homogéneos y heterogéneos.
- El capítulo 5 recopila las propuestas de los capítulos anteriores en una arquitectura con métodos originales para la monitorización del estado de la misión y los drones, y con capacidades de comportamiento dinámico reactivo frente a eventos no planificados.
- El capítulo 6 muestra el marco de experimentación y los resultados obtenidos.
- Finalmente, el capítulo 7 concluye el trabajo de investigación.

Capítulo 2

Estado del arte

Este capítulo presenta una descripción de los principales trabajos de la literatura relacionados con la presente investigación. El esquema en el cual se desarrolla sigue la línea de división del problema en tres partes principales. En primer lugar, realiza una revisión de los principales métodos de planificación de rutas de cobertura con enfoque en aplicaciones bidimensionales de múltiples áreas. Luego, realiza una descripción general de las técnicas aplicadas en sistemas multi-UAV homogéneos y heterogéneos con énfasis en la distribución de tareas. Finalmente, se describen las soluciones para la gestión del comportamiento dinámico del sistema y la replanificación en vuelo.

2.1 Planificación de rutas de cobertura

La planificación de rutas de cobertura (CPP por sus siglas en inglés) es una línea de investigación dentro del problema del enrutado de vehículos (VRP, por sus siglas en inglés) cuyo objetivo es generar rutas para un conjunto de robots móviles equipados con sensores de huella limitada para cubrir el espacio libre (Barrientos et al., 2011). Este es un campo multidisciplinario que sigue evolucionando con el avance de la tecnología y la creciente demanda de aplicaciones prácticas y que además combina diversas disciplinas, incluyendo robótica, inteligencia artificial, optimización y teoría de control.

En la literatura, se puede diferenciar principalmente los enfoques de cobertura bidimensionales (2D) y tridimensionales (3D). Por su parte, los algoritmos de cobertura 3D cuentan con aplicaciones prácticas que incluyen inspección visual de estructuras, seguimiento de terreno, reconstrucción, y otros. Por ejemplo, la investigación presentada por (Melo et al., 2021) genera la ruta en el espacio 3D utilizando métodos heurísticos y optimización lineal con enfoque en la inspección visual de infraestructuras. Otro enfoque práctico se puede encontrar en el trabajo presentado por (Mansouri et al., 2018) que divide el espacio en varias capas bidimensionales a diferentes alturas y luego genera rutas conectadas para cubrir todo el espacio. Otras propuestas como (Li et al., 2018; Mou et al., 2021) emplean técnicas más avanzadas como algoritmos genéticos (GA) y aprendizaje profundo por refuerzo (DRL) respectivamente, para la exploración de terrenos tridimensionales.

En el caso de cobertura 2D con UAVs, un trabajo muy completo de revisión del estado del arte se puede encontrar en (Cabreira et al., 2019), en el cual, los autores identifican dos tipos de métodos: descomposición y no descomposición (ver Figura 2.1).

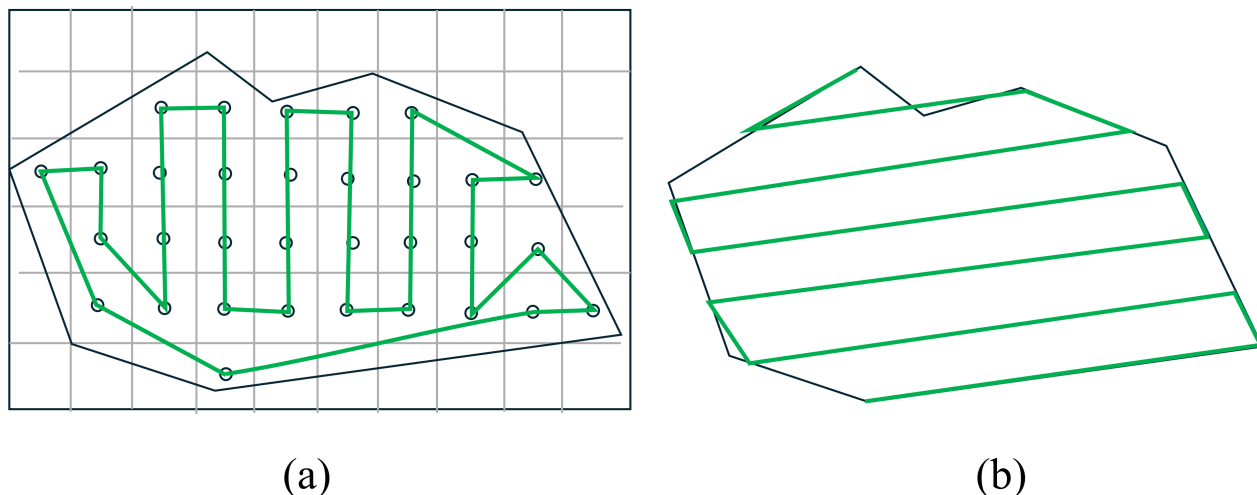


Figura 2.1: Tipos de planificación de cobertura en 2D. a) Ejemplo de un método de descomposición. b) Ejemplo de un método de no descomposición.

Como se puede esquematizar en la Figura 2.1, las técnicas de descomposición dividen el área de cobertura en sub-áreas para simplificar los movimientos del dron durante la misión. Algunos enfoques propuestos en la literatura se pueden encontrar en (Choi et al., 2020; Valente et al., 2013; Nedjati et al., 2016; Maza y Ollero, 2007). En este tipo de técnicas, los investigadores descomponen el área total en celdas y luego encuentran la ruta eficiente sin repeticiones que cubra todo el área. En el caso de sistemas multi-UAV, los algoritmos encuentran varias rutas, una para cada UAV. Sin embargo, este enfoque puede ser muy costoso computacionalmente dependiendo del método de optimización elegido. Por su parte, los métodos de no descomposición utilizan patrones geométricos clásicos para cubrir las áreas, como por ejemplo back-and-forth (Avellar et al., 2015; Hong et al., 2021) o espirales (Semsch et al., 2009; Balampanis et al., 2017; Zhu et al., 2021). Este tipo de algoritmos generan rutas de cobertura suponiendo que el entorno de trabajo es libre de obstáculos (lo cual es asumible considerando aplicaciones en las cuales a ciertas alturas de vuelo los obstáculos ya no representan un problema). En sistemas con múltiples aeronaves, las técnicas de este tipo generan una ruta de cobertura global que luego es repartida utilizando técnicas de asignación de tareas. En Torres et al., 2016, los autores combinan diferentes métodos utilizando un back-and-forth en áreas poligonales convexas, pero descomposición en celdas para áreas cóncavas.

Adicionalmente, en los últimos años, se ha planteado el problema de cobertura de múltiples áreas enunciado por los autores en (Xie et al., 2018) como TSP-CPP ya que combina la necesidad de cubrir completamente todas las área (CPP) al mismo tiempo que se optimiza el orden de exploración de las mismas (TSP). En este contexto, cabe destacar la investigación presentada por (Xie y Chen, 2022) en la cual los autores presentan múltiples enfoques para resolver este problema que incluyen programación dinámica (DP), un enfoque de vecino cercano (NN) basado en centroides y algoritmos genéticos (GA), aplicadas utilizando como

base el patrón de cobertura back-and-forth para la generación de la trayectoria, y obtuvieron resultados numéricos para uno y múltiples UAVs. Adicionalmente, los autores en Vasquez-Gomez et al., 2018 descomponen el problema resolviendo primero la generación de rutas de cobertura, y luego el cálculo del orden de exploración de las áreas. Por otro lado, (Yu et al., 2020) utiliza la técnica de optimización colonización de hormigas (ACO) para encontrar las rutas más eficientes de exploración de múltiples áreas. Sin embargo, se asume el mismo punto de despegue para cada UAV. De forma similar, los autores en (Khanam et al., 2020) generan rutas interregionales e intrarregionales con técnicas de programación lineal entera mixta (MILP) considerando la disposición de precedencia en la visita de regiones. Cabe recalcar que todos los enfoques descritos previamente han sido evaluados mediante simulaciones numéricas considerando que la aeronave tiene suficiente capacidad energética para completar toda la ruta.

2.2 Sistemas multi-UAV

Estos sistemas están formados por conjunto de UAVs que cooperan entre sí para completar misiones específicas como la cobertura de mayores áreas, la realización de tareas simultáneas y la mejora de la redundancia y la resiliencia operativa. Sin embargo, los sistemas actuales presentan varios desafíos en sus capas de control, comunicaciones, capacidades de toma de decisiones y aplicaciones prácticas que incluyen también a aquellas en las que el operador humano cierra el ciclo de control (Zhou et al., 2020). De esta manera, varias líneas de estudio se derivan de este campo incluyendo el comportamiento individual de cada plataforma aérea, su comunicación con otras aeronaves, la inteligencia de enjambre y factores externos como las restricciones legales, la complejidad del pilotaje, o los problemas de seguridad.

En este contexto, la asignación de tareas en sistemas de múltiples UAVs ha ganado interés en los últimos años como una línea de investigación y desarrollo que contribuye a la expansión y mejora de aplicaciones que requieren coordinación o cooperación entre varios drones. Su complejidad radica en la coordinación de las técnicas de navegación, la comunicación entre ellos y la adaptación a cambios en el entorno o en los objetivos de la misión. De esta manera, el propósito principal de este campo de estudio es la distribución eficiente de tareas y recursos entre los UAVs que componen el sistema para cumplir objetivos específicos considerando sus características y desempeño.

Dependiendo de los UAVs que los componen, estos sistemas se pueden clasificar en homogéneos y heterogéneos (Skorobogatov et al., 2020). Los sistemas multi-UAV homogéneos están formados por vehículos idénticos en términos de hardware, software y capacidades de vuelo, facilitando su coordinación y control. Por otro lado, los sistemas multi-UAV heterogéneos incluyen vehículos con diferentes características y capacidades, lo que puede ofrecer ventajas significativas en términos de flexibilidad y adaptabilidad a diversas misiones, pero también presenta desafíos adicionales en cuanto a la gestión de la heterogeneidad y la coordinación eficiente entre UAVs con capacidades disímiles (Bayındır, 2016).

De acuerdo con algunos autores en la literatura (Jing et al., 2020; J. Chen, Du et al., 2021) el problema de cobertura de área con UAVs es un problema NP-hard, es decir, que no se puede determinar una solución precisa de manera directa. Así, se ha propuesto un esquema de

resolución en dos etapas; la primera, se compone de la planificación de la trayectoria para una y múltiples áreas; y la segunda etapa es la asignación de estas trayectorias a cada aeronave. Este esquema se aplicará tanto en sistemas homogéneos como heterogéneos. A continuación, se realiza una revisión del estado del arte de cada uno de estos sistemas.

2.2.1 Sistemas multi-UAV homogéneos

Los sistemas homogéneos han sido el objeto de estudio de la mayor parte de trabajos de la literatura debido a sus ventajas con respecto a los sistemas de UAV único y su simplicidad comparada con sistemas heterogéneos. Por ejemplo en un resumen relativamente reciente presentado por (Zhou et al., 2020), se clasifican los trabajos realizados por enjambres de UAVs en siete categorías que incluyen: toma de decisiones, planificación de rutas, control, comunicaciones y aplicaciones prácticas. Los autores de este estudio presentan en su mayoría trabajos relacionados con sistemas homogéneos y plantean el uso de sistemas heterogéneos dentro de las tendencias futuras del estado del arte.

En el campo de la toma de decisiones (ver Figura 2.2), una de las líneas más importantes es la asignación de tareas, en la cual dado un conjunto de n UAVs, un conjunto de m tareas y una matriz de costos C , el objetivo es asignar a cada UAV la tarea o conjunto de tareas que minimice el costo global de la misión (Chung et al., 2018); estas técnicas encuentran sus aplicaciones más comunes en búsqueda y rescate, localización de objetivos, entrega de paquetes y otras.

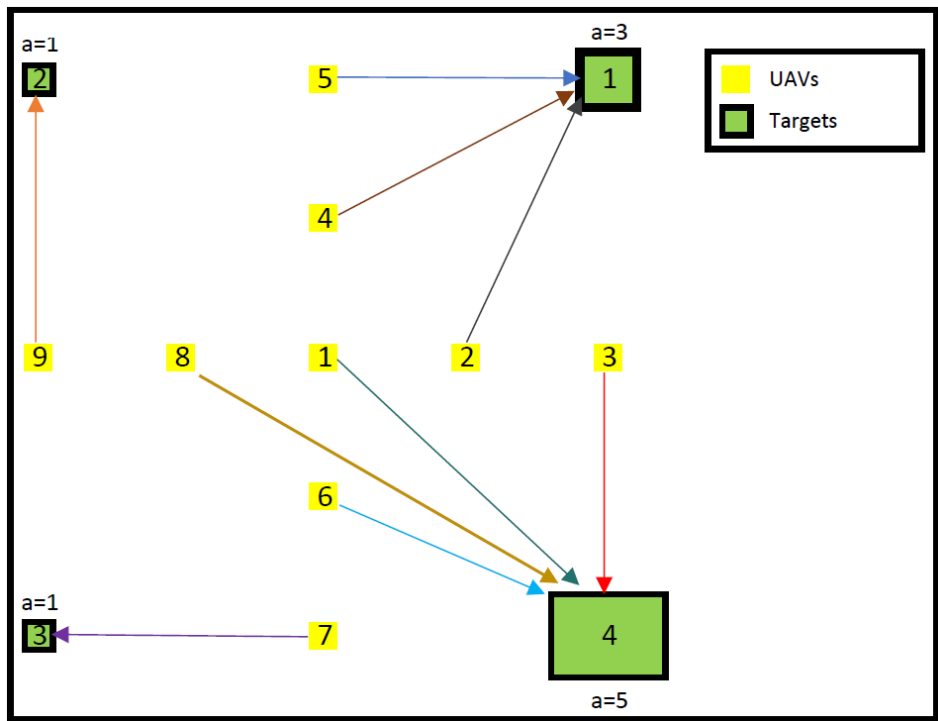


Figura 2.2: Esquema tomado de (Luna et al., 2021) para ilustrar el funcionamiento de los algoritmos de asignación de tareas en sistemas homogéneos el el cual el número de UAVs disponibles se reparte entre las tareas de acuerdo al costo de ejecución de la misión.

Así, destacan algunas técnicas basadas en subasta (Duan et al., 2019; Hu y Yang, 2018), técnicas de optimización bio-inspiradas (H. Kurdi et al., 2019; Y. Chen et al., 2018; H. A. Kurdi et al., 2018), o técnicas evolutivas (Ye et al., 2020; X. Wu et al., 2021). Las métricas de desempeño que se evalúan en este tipo de enfoque son el costo final de la misión y el tiempo de ejecución del algoritmo.

Por otra parte, entre los algoritmos de control destacan las aplicaciones de control de formación. De acuerdo a algunos autores en la literatura (Y. Liu y Bucknall, 2018; Do et al., 2021; Awasthi et al., 2023), estas técnicas se pueden clasificar en leader-follower (líder-seguidor), virtual structure (estructura virtual), behavior-based (basado en comportamiento) y consensus-based (basado en consenso), como se observa en la Figura 2.3.

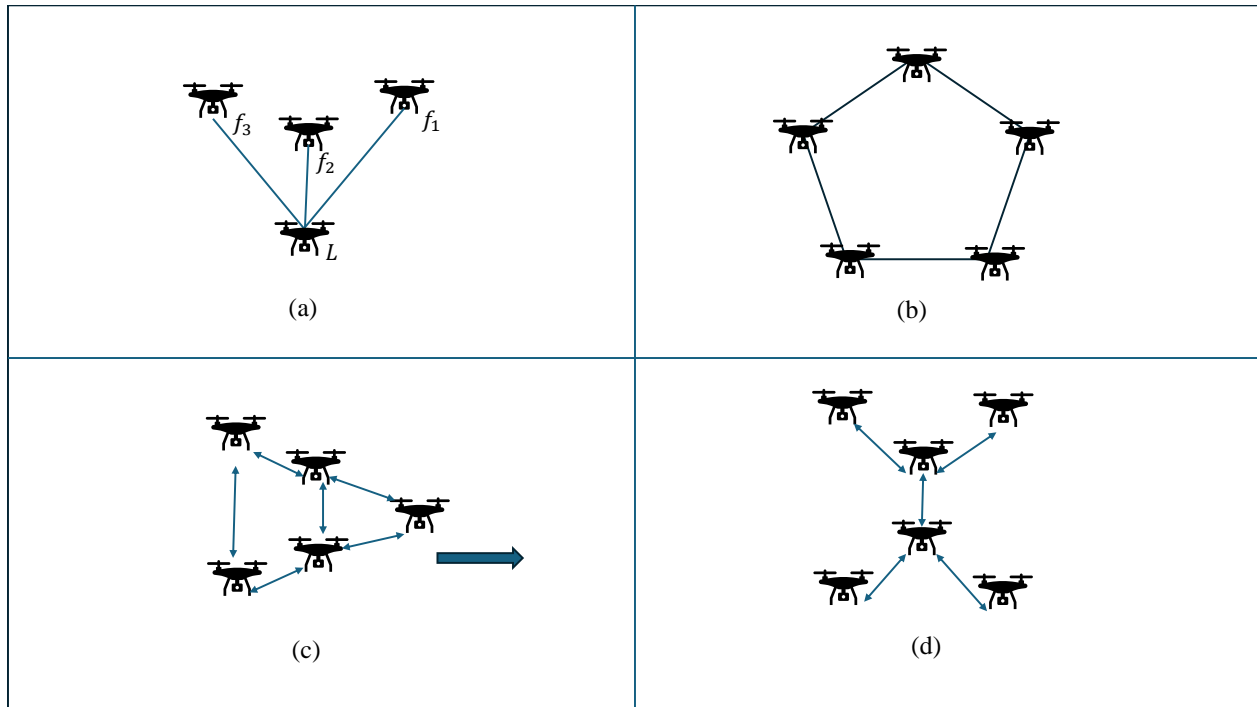


Figura 2.3: Esquema control de formación en múltiples UAVs. a) Líder seguidor. b) Estructura virtual. c) Basado en comportamiento. d) Basado en consenso

En la Figura 2.3, el control de formación líder-seguidor (Raffandi et al., 2019; Raja et al., 2022) es una estrategia de coordinación en la que una de las aeronaves actúa como líder y las demás actúan como seguidores. El objetivo es que los drones seguidores mantengan una posición relativa deseada con respecto al líder, formando así una formación específica. Una desventaja de este enfoque es su dependencia al líder. Estructura virtual (Low y San Ng, 2011; Cai et al., 2022) es una técnica de control que se utiliza para mantener la cohesión y la organización del enjambre mientras realiza una tarea en conjunto, en lugar de controlar cada UAV de manera individual, esta técnica trata al grupo de UAVs como un solo objeto rígido que se mueve y maniobra en el espacio, formando una geometría predefinida. El control basado en comportamiento (Fernando, 2021; X. Liu y Qiu, 2019) se refiere a la implementación de algoritmos y comportamientos inspirados en la naturaleza, particularmente en cómo se mueven y coordinan grupos de animales como bandadas de aves, cardúmenes de peces o

enjambres de insectos. Por ejemplo, en el comportamiento en bandada, los drones siguen reglas de separación, alineación y cohesión que dependen de la dirección y velocidad de sus vecinos inmediatos. Por su parte, el control basado en consenso (Kuriki y Namerikawa, 2015; Yan et al., 2017) se refiere a una estrategia en la cual múltiples drones alcanzan un acuerdo sobre ciertas variables clave (como posición, velocidad, o rumbo) a través de comunicaciones y algoritmos distribuidos. En este enfoque, no hay un único líder que controle toda la formación, sino que cada uno colabora y comparte información con sus vecinos para alcanzar un estado común deseado.

En la capa de planificación de trayectorias, los investigadores buscan establecer los caminos y movimientos que permitan al sistema Multi-UAV alcanzar un objetivo mientras navegan de manera autónoma por el espacio libre (Madridano et al., 2021). Los investigadores en (Zhou et al., 2020) clasifican este problema en planificación 3D estática, planificación dinámica y planificación de cobertura de área (ver Figura 2.4).

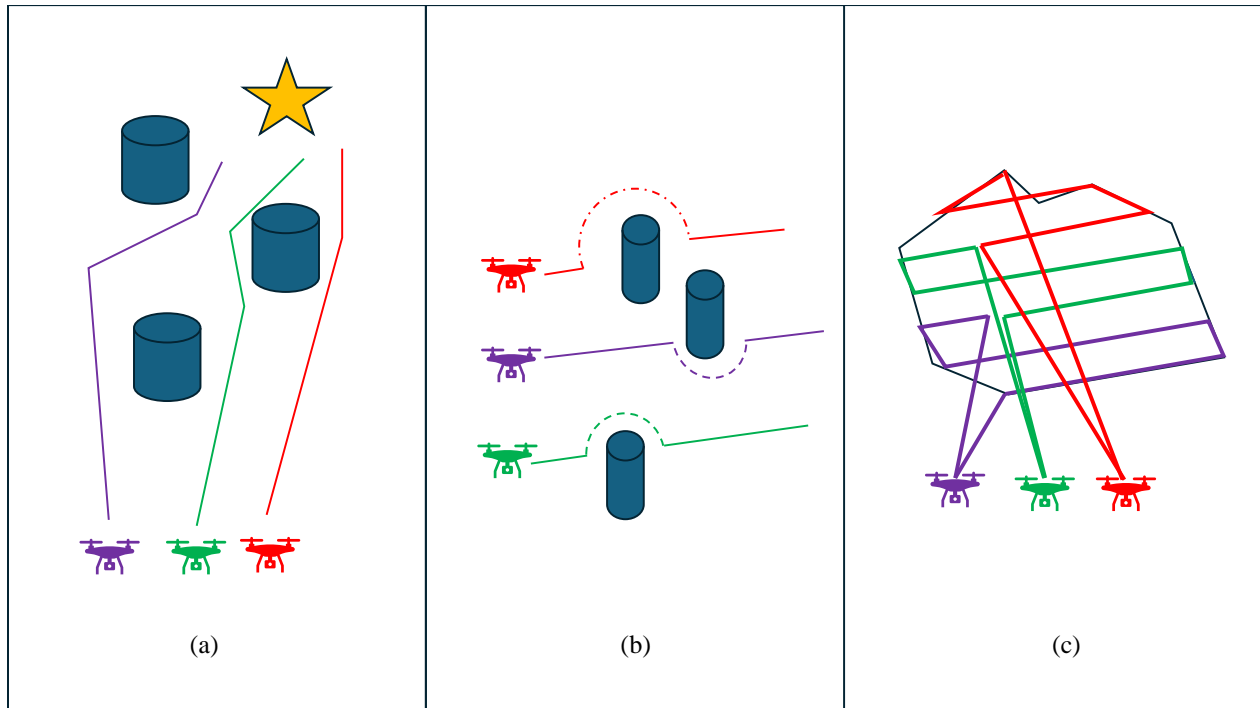


Figura 2.4: Esquema de los algoritmos de planificación en sistemas multi-UAV homogéneos. a) Planificación 3D estática. b) Planificación dinámica. c) Planificación de cobertura de áreas

En la planificación 3D estática (Dewangan et al., 2019; Madridano et al., 2020), el operador tiene un mapa 3D del entorno antes de iniciar el vuelo, por lo cual las rutas de cada aeronave se generan fuera de línea, alcanzando el punto objetivo mientras se optimiza el recorrido y se evaden los obstáculos. La planificación dinámica (X. Chen et al., 2020) es una extensión de la planificación estática, cuyo objetivo es que el grupo de UAVs genere constantemente trayectorias para esquivar obstáculos que van apareciendo durante el vuelo mientras intentan alcanzar su posición objetivo, calculando en todo momento las funciones de costo para elegir una ruta de entre un conjunto de rutas candidatas. Finalmente, en el caso de planificación

de cobertura de área con sistemas multi-UAV (Luna et al., 2022, Avellar et al., 2015), los investigadores buscan generar las trayectorias eficientes para que los UAVs puedan moverse sobre cada punto del espacio libre de las áreas de interés. En este contexto, existen múltiples enfoques en la literatura, que pueden ser divididos en dos grandes grupos: técnicas de optimización para planificación de la ruta de cada UAV y técnicas que planifican la ruta global y luego asignan sub-rutas entre el conjunto de aeronaves.

2.2.2 Sistemas multi-UAV heterogéneos

Estos sistemas compuestos por aeronaves de diferentes características tienen diversas aplicaciones en la literatura como comunicaciones robustas (Z. Xu et al., 2023), búsqueda y rescate (Arnold et al., 2020), agricultura de precisión (Mukhamediev et al., 2023), vigilancia (Orfanidis et al., 2019) y otros. A nivel algorítmico, se pueden distinguir tres áreas principales de investigación: control de vuelo en formación, asignación de tareas, y planificación de rutas.

En el campo de control de vuelo en formación de drones heterogéneos, una de las configuraciones típicas es la de "líder-seguidor". En esta configuración, el dron líder lleva equipo especializado para ejecutar tareas complejas y de navegación avanzada, mientras que los seguidores tienen el equipo necesario para seguir al líder y mantener la formación (como se puede observar en la Figura 2.5).

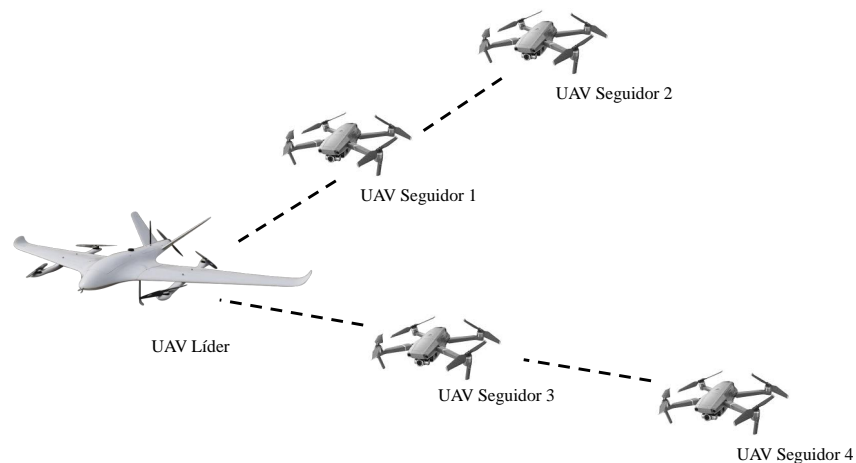


Figura 2.5: Esquema de un sistema de control de formación líder-seguidor compuesto por UAVs heterogéneos.

Por ejemplo, la investigación presentada en (Darush et al., 2023) describe un sistema de control para UAVs heterogéneos, en el cual el líder es un multirrotor con un tren de aterrizaje con capacidad de aterrizar en terrenos hostiles y moverse en tierra, la formación compuesta por seguidores más pequeños se controla en aire y tierra a través de la técnica de impedancia virtual. Por otro lado, (Gupta et al., 2022) proponen un sistema compuesto por una aeronave líder con capacidad de detección y aterrizaje en plataformas móviles, que al mismo tiempo guía en control de formación a sus seguidores con menores capacidades. Finalmente, los

investigadores en (Afghah et al., 2018) proponen un enjambre de distintas aeronaves para identificación de objetos, cuando uno de ellos ha encontrado el objetivo actúa como líder y coordina a los demás con la técnica de control de formación de coalición.

En el ámbito de la asignación de tareas, los investigadores utilizan diferentes métodos como técnicas de consenso, optimización, algoritmos evolucionarios o enfoques de aprendizaje de máquina para repartir varias tareas entre la flota de drones heterogéneos de acuerdo a las capacidades de cada uno. El objetivo principal es ejecutar el conjunto de tareas eficientemente (como se muestra en la Figura 2.6).

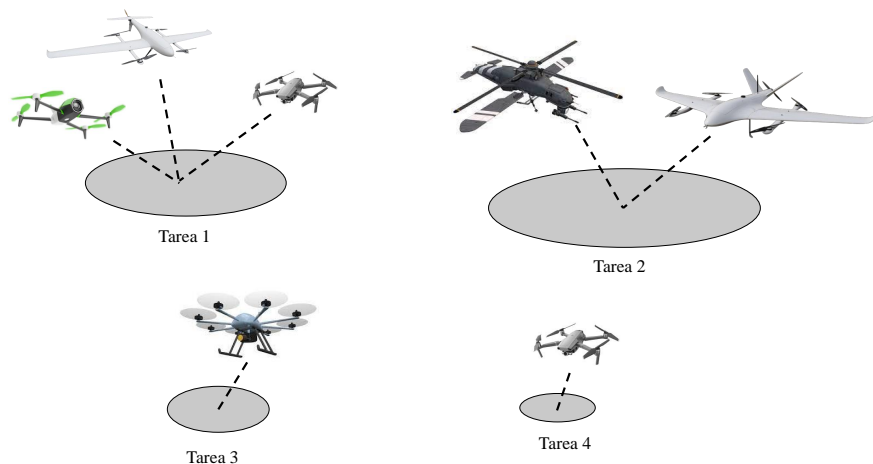


Figura 2.6: Esquema de un sistema de asignación de tareas a UAVs heterogéneos. El tamaño de los círculos grises representa el costo de la tarea, y los UAVs son asignados de eficientemente de acuerdo a sus capacidades.

Por su parte, los autores en (Zhang et al., 2016) utilizan una combinación de métodos de optimización y algoritmos evolutivos denominada GSA-GA para gestionar la repartición de tareas en un escenario con objetivos de combate. En esa investigación, los UAVs pueden ejecutar tareas de ataque y verificación y el algoritmo se encarga de minimizar el tiempo global de la misión. Igualmente, los investigadores en (Zhao et al., 2019) utilizan un enfoque de aprendizaje por refuerzo en un escenario de combate cuyos UAVs tienen capacidades de ataque, defensa o de interferencia electrónica.

Adicionalmente, los autores en (Ye et al., 2021) utilizan una versión extendida del algoritmo de CBBA (algoritmo de paquete basado en consenso) para realizar tareas de búsqueda y rescate; para ello consideran dos tipos de aeronaves: UAVs con capacidades de búsqueda y UAVs con capacidades de rescate, además, cada dron reporta diferencias en las velocidades crucero y un consumo de energía independiente del grupo al que pertenece. En general, todos estos trabajos reportan resultados en simulación y consiguen optimizar las tareas en los escenarios propuestos con bajos tiempos de procesamiento computacional.

La línea de planificación trayectorias ha ganado el interés de los investigadores en los últimos años. De la misma manera que en el caso homogéneo, el objetivo principal de la planificación

de trayectorias es generar rutas para cada UAV a partir de un conjunto de puntos de paso, que optimicen el costo de la misión (Como se presenta en la Figura 2.7).

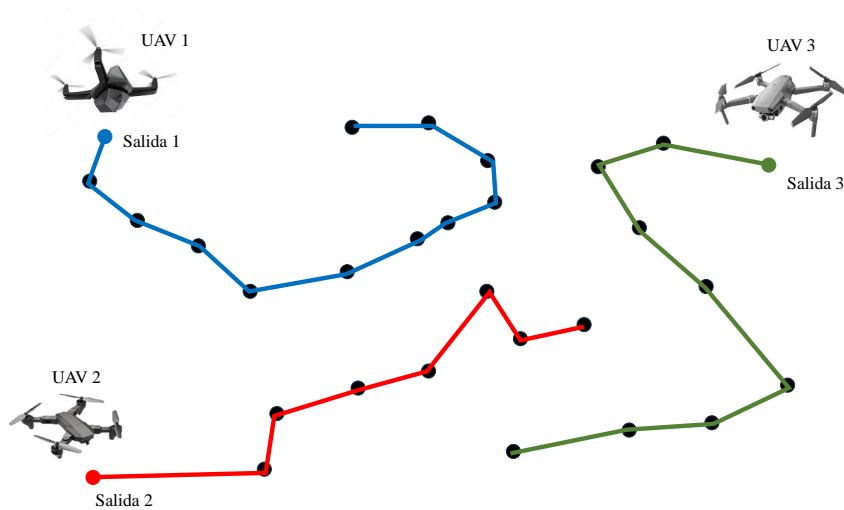


Figura 2.7: Esquema de un sistema de planificación de trayectorias con UAVs heterogéneos. Las rutas que optimizan el costo de la misión se generan de acuerdo con las capacidades de cada aeronave a partir de un conjunto desordenado de waypoints

En este contexto, la investigación desarrollada por (Bae y Chung, 2019) propone métodos heurísticos para planificación de trayectorias de UAVs heterogéneos basadas en el problema del TSP en los cuales el costo de la ruta asignada a cada aeronave se asigna en base a sus restricciones de movimiento. Los autores en (Zheng y Yuan, 2021) plantean un método optimización de búsqueda de vecinos para la planificación de la misión. Debido a que la heterogeneidad de su sistema depende de la distancia máxima que puede recorrer cada UAV, realizan un modelado matemático para determinar dicha distancia. Finalmente, el trabajo de investigación presentado en (H. Liu et al., 2020), propone un algoritmo de optimización basado en LSO (Lifecycle-Based Swarm Optimization) para realizar una planificación de trayectorias RRT tridimensional en un sistema heterogéneo, el escenario propuesto contempla dos tipos de UAVs cuyas diferencias se reflejan en su capacidad de ser detectados por un radar.

En aplicaciones de cobertura de área, varios trabajos en la literatura presentan diferentes planteamientos y enfoques del problema. La investigación presentada por (Deng et al., 2019), propone un enfoque de algoritmos genéticos y programación cuadrática de enteros mixtos (MIQP) para la división y exploración de una sola área. La heterogeneidad del sistema radica en el uso de UAVs y UGVs, no se consideran diferencias en el ancho de barrido. Otro enfoque, se puede encontrar en (Mukhamediev et al., 2023), en el cual los autores reúnen varias características de UAVs comerciales (velocidad, distancia máxima de vuelo, y precios por vuelo), para repartir una ruta de entre los UAVs utilizando algoritmos genéticos (GA), tampoco se consideran diferencias en el ancho de barrido. Adicionalmente, los autores en (J. Chen, Ling et al., 2022), abordan el problema de cobertura con tres variables de heterogeneidad: velocidad, ancho de barrido y tiempo de batería restante; sin embargo, el planteamiento inicial del problema consiste en varios UAVs para cubrir múltiples áreas relativamente pequeñas, los autores utilizan un enfoque basado en Optimización de colonia

de hormigas (ACO) para repartir las áreas entre el conjunto de UAVs. El mismo planteamiento del problema resuelto con técnicas de agrupación se puede encontrar en (J. Chen, Du et al., 2021; J. Chen, Zhang et al., 2021). De esta manera, en el planteamiento de la presente tesis se considera la exploración de una sola área bajo las tres variables de heterogeneidad presentadas previamente.

2.3 Gestión dinámica y replanificación de la misión

El diseño y la organización de los sistemas multi-UAV son cruciales para su correcto funcionamiento, múltiples trabajos en la literatura han propuesto varias arquitecturas para gestionar la cooperación y coordinación entre las aeronaves; pero se identifican principalmente tres tipos: centralizada, descentralizada y distribuida (como se muestra en la Figura 2.8).

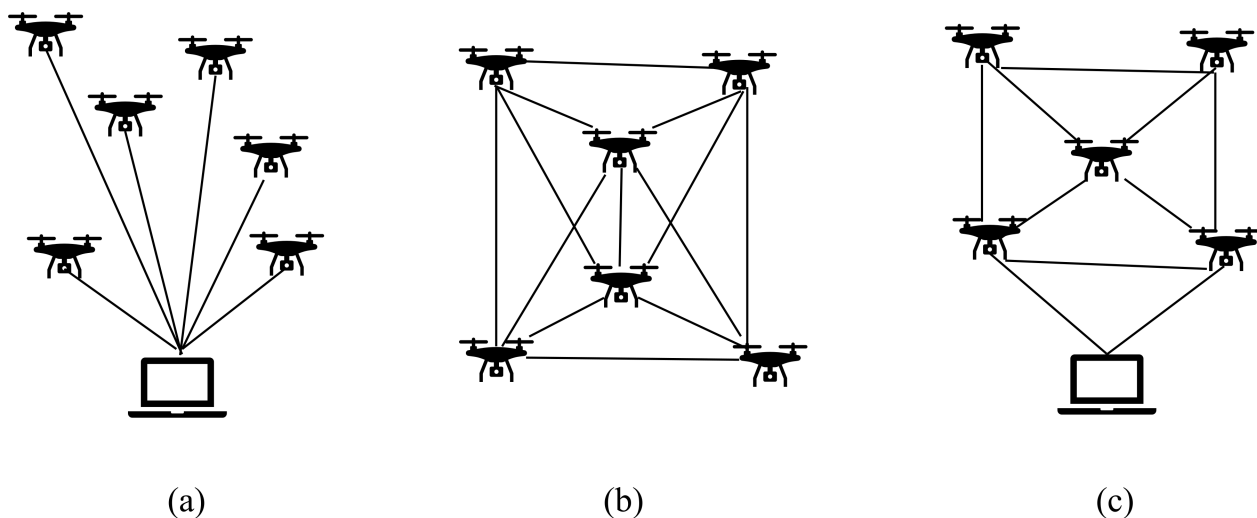


Figura 2.8: Esquema de arquitecturas de control en sistemas multi-UAV. a) Centralizado. b) Descentralizado. c) Distribuido

En una arquitectura centralizada (Sabino et al., 2018; Brandao et al., 2014), un nodo central gestiona y coordina todas las operaciones de los drones, facilitando la toma de decisiones y la planificación pero con vulnerabilidades a fallo de este nodo. Por su parte, en la arquitectura descentralizada (Bupe et al., 2015; Ye et al., 2021) el control se gestiona por varias aeronaves, aunque esto reduce el riesgo de fallo del nodo central, complica aún más la coordinación y la comunicación entre los UAV. Finalmente, en la arquitectura distribuida (Venturini et al., 2021) el control y la toma de decisiones no están centralizados en un único UAV o en una estación de control en tierra sino que se reparten entre varios drones que cooperan mediante la comunicación directa entre ellos.

Así, el desarrollo de arquitecturas de control en sistemas compuestos por varios drones ha sido el punto de mira de varios investigadores especializados en inteligencia de enjambre Zhou et al., 2020. Por ejemplo, en el trabajo presentado por (Pinto et al., 2021), los autores proponen una arquitectura cognitiva centralizada para plataformas aéreas con capacidades de toma de decisión colectiva. Por su parte, los investigadores en (Emel'yanov et al., 2016) describen

una arquitectura multi-capa para gestionar los enjambres de UAVs. Finalmente, los estudios presentados por (Sanchez-Lopez et al., 2016; Molina et al., 2019) proponen Aerostack¹, una arquitectura centralizada desarrollada en ROS (sistema operativo robótico por sus siglas en inglés) para coordinar una ejecución concurrente de las misiones. Este marco de desarrollo evolucionaría posteriormente a Aerostack2² como un software de arquitectura distribuida, disponible para ROS2 (Fernandez-Cortizas et al., 2023).

Por otro lado, durante el desarrollo de una misión en el mundo real, los algoritmos de control y monitorización deben considerar que las plataformas aéreas están sujetas múltiples tipos de fallo, lo cual se vuelve más crítico en un entorno de múltiples UAVs. Por este motivo, existen varios trabajos en la literatura que han estudiado a los sistemas Multi-UAV tolerantes a fallo (Ziquan et al., 2022; Furlas y Karras, 2021). Por ejemplo, en las investigaciones presentadas por (De Benedetti et al., 2017; Tahir et al., 2020; Hu et al., 2022) los autores abordan el problema del control de formación tolerante a fallos, cuya idea es mantener una formación geométrica regular tolerante a la pérdida de uno o más UAVs. La investigación presentada en (Wang et al., 2021) establece un modelo de asignación de tareas con reasignación en caso de fallos en las aeronaves. Además, múltiples autores proponen técnicas de replanificación de la misión en entornos dinámicos. Por ejemplo investigaciones como (J. Chen, Qing et al., 2022; Meng et al., 2009) planifican una ruta de bajo costo previo al desarrollo de la misión, la cual se recalcula frente a un conjunto de eventos de replanificación. En estos casos, el sistema determina que se han encontrado nuevos waypoints de interés y se busca el camino más corto para visitarlos. Finalmente, la investigación presentada por (Hong et al., 2022), aborda el problema de planificación de rutas de cobertura de área multi-UAV incluyendo replanificación en vuelo, en este trabajo, los investigadores proponen una técnica heurística denominada RHTA y su arquitectura contempla dos tipos de eventos de replanificación: la adición de un nuevo UAV al sistema o la adición de una área nueva de exploración, sin embargo, no se aborda el tema de la detección de fallos automática.

¹<https://github.com/cvar-upm/aerostack>

²<https://aerostack2.github.io/>

Capítulo 3

Planificación de trayectorias de cobertura de área para sistemas de UAV único

Este capítulo describe las soluciones desarrolladas para la planificación de rutas de cobertura en área única y múltiples áreas. En primer lugar, en la introducción, se describe el problema de la planificación de rutas de cobertura. A continuación, se detalla el método propuesto de planificación en espiral para áreas únicas, y finalmente, se extienden estos algoritmos con dos enfoques novedosos para resolver el problema de múltiples áreas.

3.1 Introducción

Los problemas de planificación de trayectorias han estado presentes en la robótica casi desde sus inicios, y su postulado más básico plantea hallar el camino más corto entre un punto inicial y un punto final esquivando obstáculos. Durante los primeros años de desarrollo de este campo, algunos trabajos de investigación como (Ó'Dúnlaing y Yap, 1985 ; Canny, 1988) proponen utilizar algoritmos basados en un mapa precargado del entorno, lo cual contribuyó a la visualización de nuevos desafíos que a su vez aportaron al desarrollo de nuevas líneas de investigación que cambiaron la concepción clásica en la planificación punto inicial- punto final (Choset, 2001).

Así, la planificación de rutas de cobertura (CPP por sus siglas en inglés) es una línea de investigación dentro de la planificación de movimientos, cuyo objetivo es la generación de trayectorias para que un robot con una huella de sensor limitada cubra todo el espacio libre (Barrientos et al., 2011). Este campo tiene múltiples usos en la robótica aplicada como robots limpiadores (Yasutomi et al., 1988; Farsi et al., 1994), UGV (Baras y Dasygenis, 2023), USV (P.-F. Xu et al., 2021), UUV(Luo et al., 2019, UAVs (Avellar et al., 2015) y otros. De acuerdo con (Galceran y Carreras, 2013), los algoritmos de CPP deben intentar cumplir algunos requisitos importantes como la generación de trayectorias que cubran toda el área, evitar la superposición de caminos, esquivar obstáculos si los hay, no repetir los puntos ya

visitados, entre otras. Algunos estudios recientes (Almadhoun et al., 2019; C. S. Tan et al., 2021; Cabreira et al., 2019) remarcan la relación entre el problema del CPP y el problema del viajante (TSP, por sus siglas en inglés) ya que ambos comparten requerimientos similares por lo que las mismas técnicas de optimización utilizadas en el TSP se han utilizado comúnmente para resolver problemas de CPP.

Uno de los retos actuales del CPP es la caracterización del vehículo y del área de interés. En el campo de la robótica aérea podemos diferenciar de manera general a vehículos de ala fija y ala rotativa, de los cuales se identifican cuatro subclases incluyendo multirrotores, ala fija, helicópteros y VTOL. Si consideramos los seis grados de libertad en los que se pueden mover este tipo de vehículos (movimiento longitudinal, movimiento lateral, movimiento vertical, cabeceo, guiñada y alabeo), solo algunos sistemas experimentales como el presentado por (Ryll et al., 2014) tienen mayor maniobrabilidad en los seis grados. En sistemas comerciales, se definen cuatro grados de maniobrabilidad para aeronaves de ala rotativa (movimientos en x , y , z y guiñada) y tres en ala fija (ejes x , y y z). En este contexto, tanto multirrotores como helicópteros tienen mayor maniobrabilidad al ejecutar trayectorias complejas comparadas con ala fija y VTOL. Por esta razón, los algoritmos de cobertura para ala fija contarán con más restricciones como el radio de giro y la dirección de navegación (Figura 3.1). Sin embargo, la

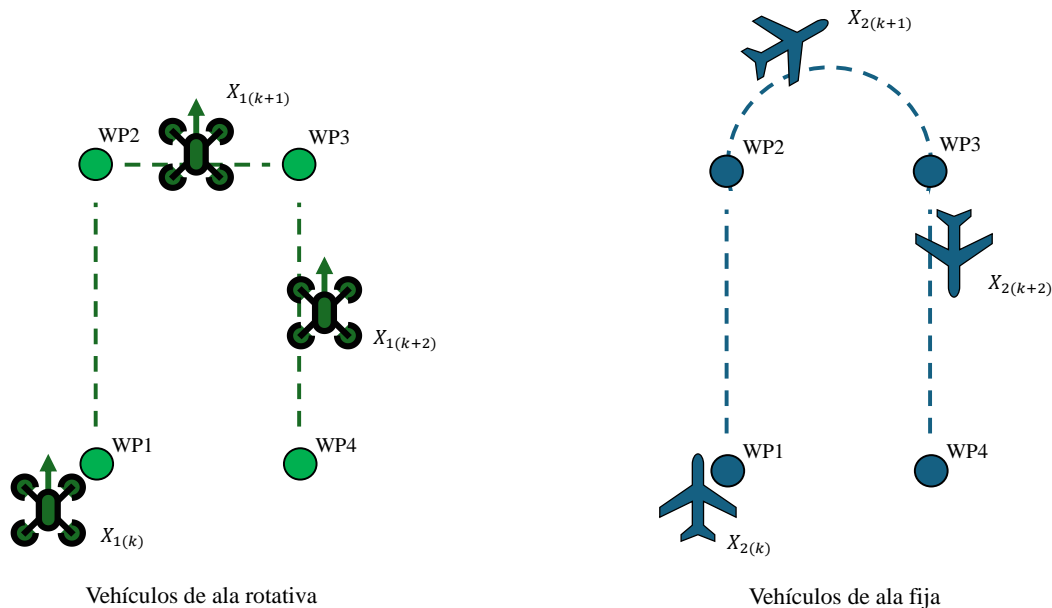


Figura 3.1: Esquema de rutas generadas para ala rotativa y ala fija. Los primeros se pueden configurar para seguir una trayectoria sin realizar cambios de giro volando incluso hacia atrás

restricción de dirección de giro se puede aplicar a los vehículos de ala rotativa considerando los casos en los que la carga de pago apunta hacia el frente. En estos casos es necesario configurar el sistema de control de vuelo para que el UAV dirija el rumbo apuntando a cada waypoint. Así, algunos patrones de cobertura de área pueden resultar ineficientes.

Además, la caracterización del área tiene retos asociados a la maniobrabilidad y a la aplicación. Por ejemplo, la cobertura de área 3D se aplica en inspección de estructuras sólidas (Cheng et al., 2008) y monitorización de terrenos con curvas de nivel (Mou et al., 2021). Por otro lado, el estudio de trayectorias de cobertura en áreas con obstáculos es más frecuente en aplicaciones de exploración urbana (Muñoz et al., 2021) donde los obstáculos vienen dado por las infraestructuras propias del entorno. sin embargo, sus aplicaciones aún son limitadas debido a las regulaciones legales de la mayor parte de países para realizar este tipo de vuelos. Finalmente, se puede mencionar la cobertura 2D de área única y múltiples áreas libres de obstáculos (Di Franco y Buttazzo, 2015; Avellar et al., 2015; J. Chen, Du et al., 2021); cuyas aplicaciones son más extensas y abarcan la agricultura de precisión, monitorización, búsqueda y rescate, fotogrametría y otras. En este tipo de técnicas se asume que las aeronaves vuelan lo suficientemente alto para esquivar los obstáculos del área de exploración.

En este contexto, el presente capítulo describen técnicas de planificación de trayectorias de cobertura de áreas 2D en entornos libres de obstáculos para una y múltiples regiones. En primer lugar, se propone una técnica de barrido en espiral que optimiza el costo de la misión para los casos en los que las aeronaves deben seguir el curso de los waypoints. A continuación, se proponen dos técnicas heurísticas de planificación en múltiples áreas que optimizan el costo de la misión y se ejecutan en tiempos de CPU relativamente bajos.

3.2 Planificación en área única

3.2.1 Planteamiento del escenario

Dada una región poligonal P definida como un conjunto de puntos en el espacio \mathbb{R}^2 , se requiere generar una trayectoria para cubrir dicha región con un vehículo de ala rotativa equipado con un sensor de huella limitada y configurado para navegar con rumbo hacia los waypoints debido a las características de la carga de pago. Además, se asume un entorno libre de obstáculos considerando que la altura de vuelo es constante y lo suficientemente elevada para sobrevolarlos (Figura 3.2).

De la Figura 3.2 se puede intuir que el objetivo propuesto es encontrar el conjunto de waypoints para cubrir la región de manera uniforme considerando el tamaño de la huella del sensor y la superposición deseada (necesaria para realizar emparejamiento de los datos durante el post-procesamiento). Por esta razón, se busca optimizar la dirección de barrido durante la planificación.

3.2.2 Algoritmo de cobertura de área en espiral

Según el estudio presentado por (C. Wu et al., 2019), una de las ventajas del tipo de cobertura en espiral es su eficiencia energética comparada con otros métodos, debido a que reduce los ciclos de aceleración y desaceleración durante las maniobras de giro. Sin embargo, la generación de este patrón puede ser costosa computacionalmente cuando se calculan los waypoints generando las rutas internas mediante el cálculo de distancias y ángulos de giro (Cabreira et al., 2018) o utilizando coordenadas polares (Hassan y Liu, 2018). El método propuesto en esta sección utiliza matrices de transformación afín que calculan los puntos

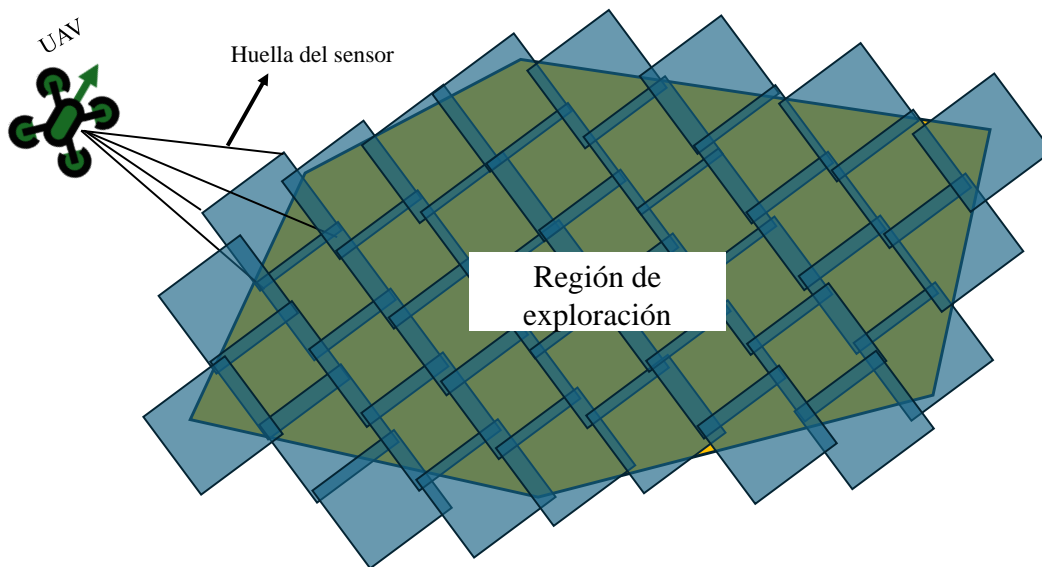


Figura 3.2: Esquema del problema de cobertura de área única.

mucho más rápido y presenta algunas mejoras con respecto a un enfoque similar presentado previamente (Luna et al., 2023). En primer lugar, se determina el orden de los puntos de la región a explorar y el sentido de giro de la espiral en base a la ubicación del UAV de acuerdo al análisis de las Figuras 3.3 y 3.4.

En la Figura 3.3, se utiliza como referencia el punto más cercano a la aeronave, luego se traza una línea recta y se calculan los ángulos entre esta recta y los puntos B y C, siendo el ángulo menor (θ_1) el formado por la recta \overline{AC} . En este caso la espiral comenzará en el punto A y continuará hacia C para generar el menor costo en ángulo de giro. Igualmente, en la Figura 3.4 el sentido de giro se corresponde con el menor ángulo θ_1 formado en este caso por las recta \overline{AB} . El resultado de aplicar este método es una lista ordenada de los puntos de la región que comienza en el punto más cercano al dron y continúa de acuerdo al sentido de giro óptimo. El resto de los pasos se describen en el Algoritmo 1.

Este método recibe como entradas la localización del UAV (U), las coordenadas relativas de la región a explorar (P), el ancho de las calles (lw), y la separación de los waypoints ($spacing$). Los parámetros lw y $spacing$ se obtienen de las características de la cámara y de la superposición deseada en la huella, sin embargo, en términos prácticos, hemos dejado su análisis fuera del alcance de este estudio.

Como primer paso se define una matriz vacía W en la cual se almacenarán los waypoints de la espiral. Después, la función *RegionProcessing* ordenará los puntos de la región de acuerdo al análisis planteado anteriormente en las Figuras 3.3 y 3.3 y los puntos resultantes se guardan en la matriz W . Luego, se calcula el centroide del polígono (c), y la menor distancia perpendicular (cp) entre el centroide (c) y cada uno de los ejes del polígono (e_i) de acuerdo

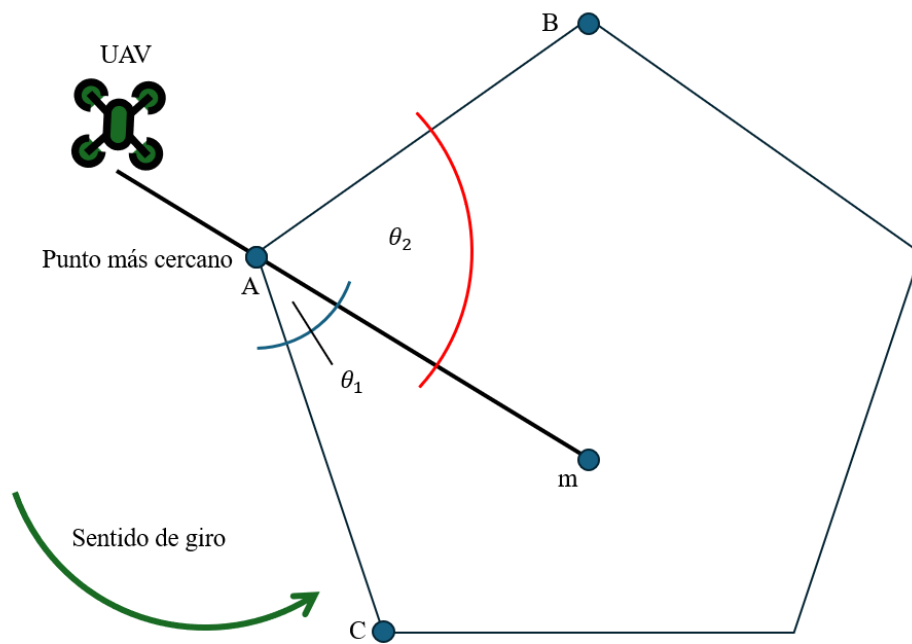


Figura 3.3: Esquema de cálculo de sentido de giro de la espiral para el caso horario (CW)

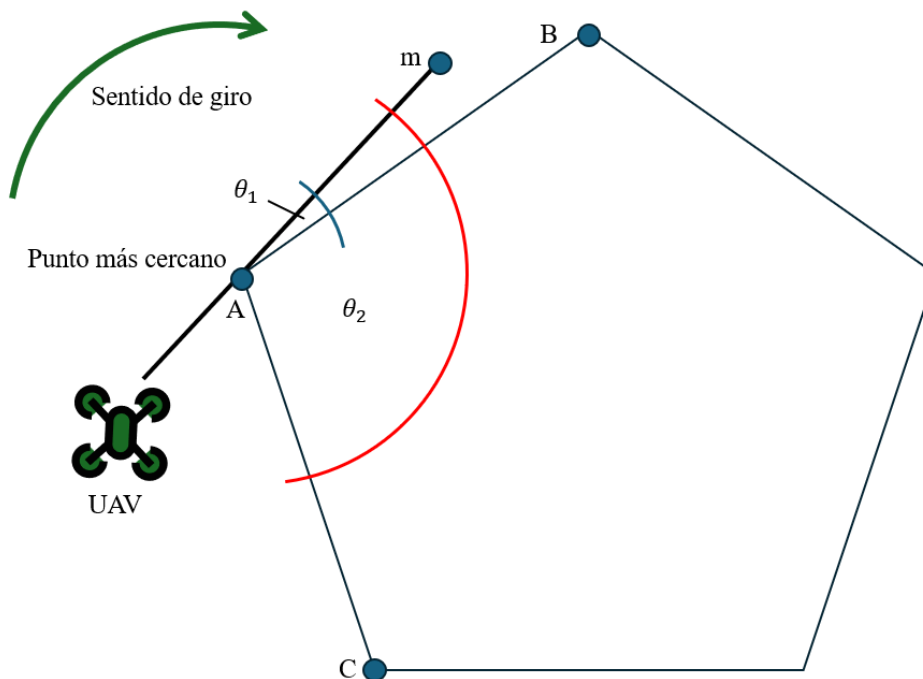


Figura 3.4: Esquema de cálculo de sentido de giro de la espiral para el caso anti-horario (CCW)

Algoritmo 1: Algoritmo de cobertura de área en espiral

Data: $U, P, lw, spacing$

Result: S

```

1  $W \leftarrow null$ 
2  $CurrentP \leftarrow RegionProcessing(U, P, lw)$ 
3  $W \leftarrow Append(CurrentP)$ 
4  $c \leftarrow Centroid(CurrentP)$ 
5  $cp \leftarrow GetdCP(CurrentP)$ 
6 while  $cp < lw$  do
7    $W \leftarrow AppendSupportPoint(CurrentP, lw)$ 
8    $InnerP \leftarrow Scale\&Translate(CurrentP, lw, c, cp)$ 
9    $W \leftarrow Append(InnerP(P_1^N))$ 
10   $CurrentP \leftarrow InnerP$ 
11   $cp \leftarrow GetdCP(InnerP, c)$ 
12 end
13  $S \leftarrow SampledClothoid(W, spacing)$ 

```

a la ecuación 3.1

$$cp = \min \|c - e_i\| \quad (3.1)$$

Este concepto de distancia fue introducido por (Cabreira et al., 2019) para determinar el número de anillos que debe tener una espiral en dividiendo cp para el ancho de calles lw . Sin embargo, esta expresión tiene que ser redondeada a un número entero, lo cual puede causar que el ancho de calles no coincida con el deseado o que no se cubran los tramos más cercanos al centroide. Por esta razón, nuestro enfoque utiliza un proceso iterativo con la condición de terminación $cp < lw$ que asegura que ambas condiciones se cumplan. A continuación, para la función *AppendSupportPoint* se genera un punto adicional que completa el enrutado de la espiral cómo se muestra en la Figura 3.5.

En la Figura 3.5, se puede observar que los puntos de soporte de la espiral se generan a partir del polígono actual y el ancho de calles. Básicamente, este punto es una transición entre cada uno de los polígonos internos que forman la espiral, el cual se añade de manera individual a la matriz W .

En el siguiente paso, se realiza el escalado y translación de la espiral. Primero, se calcula el factor de escala (sF) utilizando la ecuación 3.2.

$$sF = \frac{\|c\| - lw}{\|c\|} \quad (3.2)$$

Donde $\|c\|$ es la distancia del centroide a un punto cualquiera del polígono. A continuación, los puntos de coordenadas x e y del polígono interno generado se mueven para que el nuevo centroide coincida con el centroide original a través de una operación de translación.

En los pasos 9, 10 y 11 del Algoritmo 1, se agregan los puntos resultantes a la matriz W y se actualizan los valores de $CurrentP$ y cp . Todo el proceso se repite hasta cumplir la condición

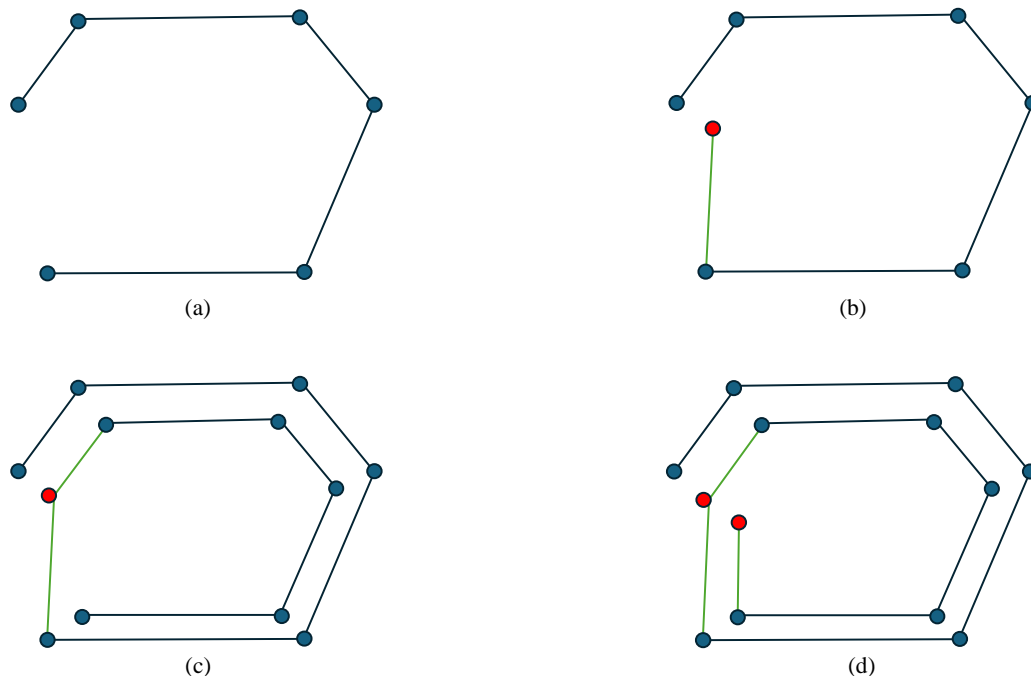


Figura 3.5: Esquema de generación de puntos adicionales en la espiral. (a) Puntos originales del polígono. (b) Generación del primer punto de soporte. (c) Primera iteración de escalado. (d) Generación del segundo punto de soporte

previamente mencionada.

Finalmente, la función *SampledClothoid* genera una clotoide para suavizar las rutas. Para esto utiliza el método "*G¹ fitting*", propuesto por (Bertolazzi y Frego, 2015). En primer lugar, se muestrea la espiral considerando la distancia entre waypoints deseada (*spacing*), y luego se generan clotoides iterativamente con el objetivo de atenuar el giro en las curvas. Los resultados de este algoritmo se presentan en el capítulo 6.

3.3 Planificación multi-área

3.3.1 Planteamiento del escenario

Dado un conjunto de áreas poligonales A compuestas por múltiples polígonos convexos definidos en el espacio \mathbb{R}^2 , el objetivo de los métodos de cobertura multiárea es encontrar la trayectoria eficiente de un UAV que cubra el espacio libre de todas las áreas. Con respecto al problema de planificación en área única, la planificación de trayectorias de cobertura en múltiples áreas incluye además la determinación del orden correcto de visita de las regiones para minimizar el costo global de la misión. Los autores en (Xie et al., 2018) han denominado a este problema como TSP-CPP, es decir, una combinación del problema del viajante con los problemas propios de la generación de trayectorias de cobertura, tal como se muestra en la Figura 3.6.

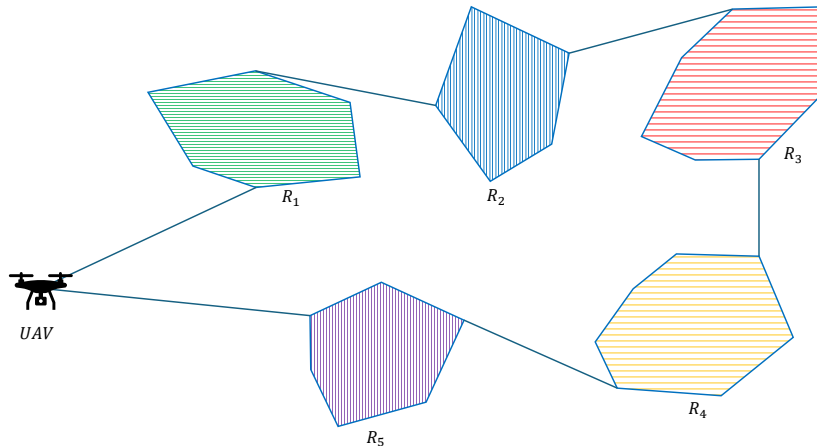


Figura 3.6: Esquema del problema de cobertura de múltiples áreas

Al igual que múltiples aplicaciones prácticas de la literatura como inspecciones fotovoltaicas (Luna et al., 2023), monitorización (Balampanis et al., 2017) o agricultura de precisión (Barrientos et al., 2011), hemos considerado que cada área es un entorno libre de obstáculos. Además, el número de regiones de exploración ha sido acotado entre uno y diez, considerando los escenarios comunes de las aplicaciones industriales.

Por otro lado, al igual que la cobertura de área única, no se abarca el cálculo de la huella basada en el FOV dentro del alcance de este estudio, sino que se trabaja directamente con los datos de altura de vuelo, ancho de calles y espacio entre waypoints. Adicionalmente, los resultados de los algoritmos de planificación de trayectorias en múltiples áreas se muestran en el capítulo 6.

3.3.2 Cobertura multi-área basada en TSP

Este algoritmo se ha descrito en las investigaciones presentadas en (Flores Peña et al., 2022); (Luna et al., 2023) y se compone de dos etapas: la determinación del orden de exploración de las áreas y el cálculo de las rutas de cobertura, de manera similar el trabajo presentado por (Vasquez-Gomez et al., 2018), cuyo enfoque utiliza una combinación de la estrategia back-and-forth para la cobertura y algoritmos genéticos (GA por sus siglas en inglés) para la determinación del orden de visita. En nuestra propuesta calculamos en primer lugar el centroide de cada área a explorar y los puntos obtenidos, además de la posición inicial del UAV, se utilizan como entrada del método de Cristofides (Christofides, 2022) para resolución del TSP, disponible en una librería de Python¹. Un esquema se presenta en la Figura 3.7.

A continuación, el siguiente paso consiste en generar las rutas de cobertura en cada región. Para esto, se toma como referencia los puntos iniciales y finales de las trayectorias de cada región de acuerdo al Algoritmo 2.

¹<https://pypi.org/project/Christofides/>

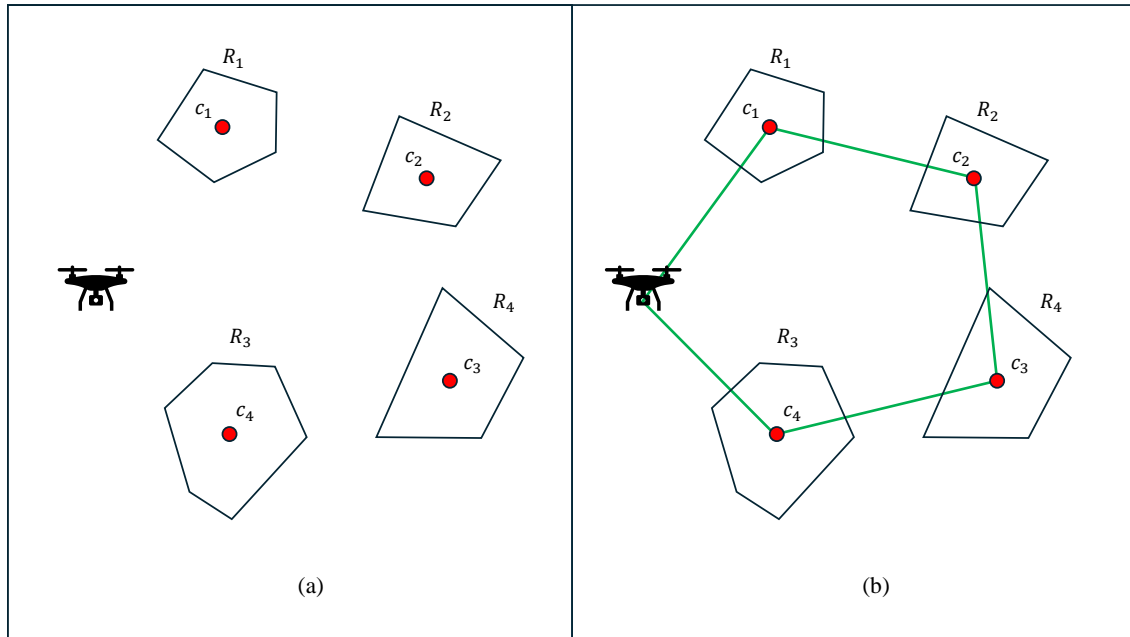


Figura 3.7: Esquema de la primera etapa del algoritmo de cobertura basado en TSP. a) Cálculo del centroide de las áreas. b) Resolución del TSP basada en los centroides y la posición inicial del UAV

Algoritmo 2: Algoritmo de generación de rutas de cobertura con el método TSP

Data: U, A

Result: W

```

1  $PrevPoint = U$ 
2 foreach  $a$  en  $A$  do
3    $C \leftarrow GenCoverageRoute(PrevPoint, a)$ 
4    $W \leftarrow Append(C)$ 
5    $PrevPoint \leftarrow GetRouteLastPoint(C)$ 
6 end
    
```

En el Algoritmo 2, U representa la posición inicial del UAV y A es el conjunto de polígonos ordenados de acuerdo al método de Cristofides de la etapa previa. En primer lugar, se inicializa $PrevPoint$ como la posición inicial del dron, después se inicia un proceso iterativo en cada área y se genera el conjunto de puntos C con las técnicas de planificación de ruta de cobertura (B&F o espiral) para cubrir de manera eficiente en base al polígono actual (a) y el punto previo ($PrevPoint$). Este paso se ha definido de forma general con la función $GenCoverageRoute()$. Después, se guardan cada una de las rutas en el vector W y finalmente, el nuevo punto previo se iguala al último punto de la ruta anterior generada, el cual se obtiene con la función $GetRouteLastPoint()$.

3.3.3 Cobertura multi-área según vecino más cercano

Este algoritmo heurístico, descrito en (Luna et al., 2024), recibe como entrada un conjunto de regiones y genera rutas de cobertura de áreas (B&F o espiral), al mismo tiempo que determina el orden de visita de manera iterativa. Para esto, toma como referencia las distancias más cortas entre los puntos de inicio y fin de cada ruta (como se observa en la Figura 3.8).

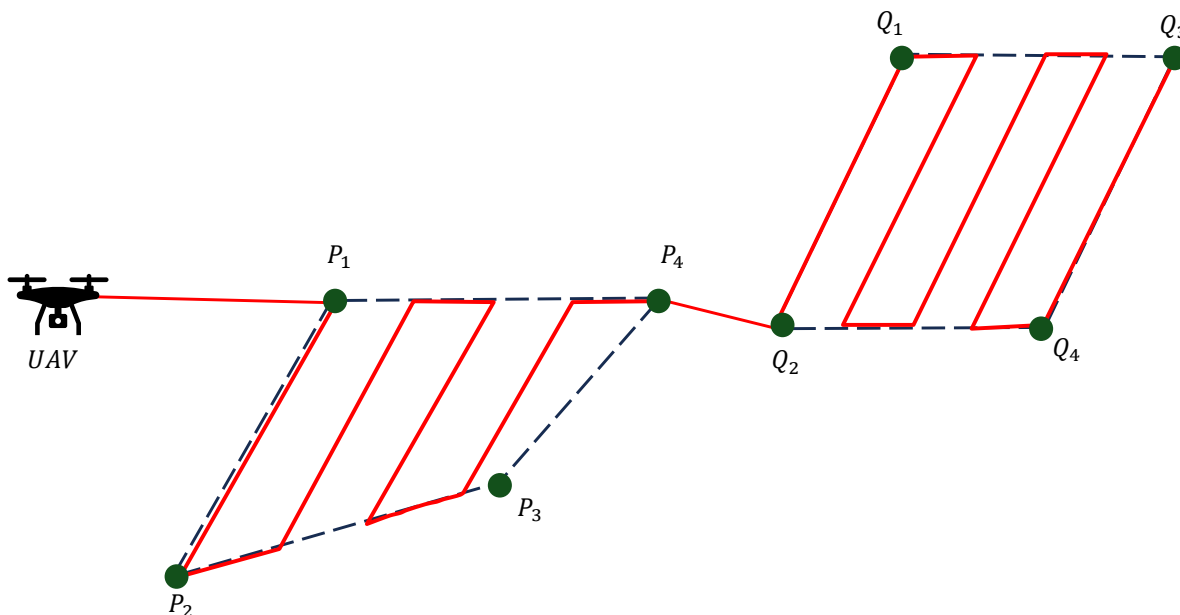


Figura 3.8: Esquema de cobertura de área basado en región vecina más cercana

La Figura 3.8, muestra un diagrama del método de cobertura multi-área para el supuesto de dos regiones. Para determinar la primera región en ser explorada, se toma como punto de referencia la posición inicial del UAV. De esta manera, se cubre el polígono formado por los puntos $P(P_{1-4})$. A continuación, se toma como referencia el punto P_4 , que es el final de la primera ruta y se explora el polígono $Q(Q_{1-4})$ siguiendo el criterio de distancia más cercana. Este proceso se describe formalmente en el Algoritmo 3.

Este método toma como entradas la posición inicial del UAV (U) y el conjunto R de regiones

Algoritmo 3: Algoritmo multi-área según vecino más cercano

Data: U, R

Result: W

```

1  $W = null$ 
2  $x^{ini} \leftarrow U$ 
3  $B \leftarrow GenCoverageRoute(R)$ 
4  $B^{ini} \leftarrow FirstPoints(B)$ 
5 while  $B \neq null$  do
6    $B^{min} = MinDist(x^{ini}, B, B^{ini})$ 
7    $W \leftarrow Append(B^{min})$ 
8    $B \leftarrow Remove(B^{min})$ 
9    $x^{ini} \leftarrow GetLastPoint(B^{min})$ 
10 end

```

a explorar y cuyo resultado es una matriz de waypoints W . En primer lugar, se designa como punto inicial x^{ini} a la posición inicial de la aeronave. Luego, se generan las rutas de cobertura con la función $GenCoverageRoute()$. A continuación, se obtienen los puntos iniciales de todas las rutas con la función $FirstPoints()$. Después, se inicia un proceso iterativo en el cual, primero se calcula la distancia mínima del punto inicial actual y los puntos iniciales de todas las rutas con la función $MinDist()$. Como resultado de esta etapa, se agregan los puntos de la ruta B^{min} al vector de waypoints W . Luego, se elimina B^{min} del vector B , y se utiliza la función $GetLastPoint()$ para calcular el nuevo x^{ini} que es el último punto de B^{min} . El algoritmo repite el proceso de forma iterativa hasta que todas las rutas se agreguen a W .

Capítulo 4

Distribución de trayectorias en sistemas multi-UAV

Este capítulo presenta los métodos desarrollados para la distribución de la trayectoria en sistemas multi-UAV en tres partes. En la introducción se presenta una visión general de los sistemas multi-UAV y las principales diferencias entre los sistemas homogéneos y heterogéneos incluyendo las ventajas y debilidades de cada uno. Después, en la siguiente sección se presenta el algoritmo original Powell-BINPAT para la distribución de tareas en sistemas homogéneos. Finalmente, la última sección, presenta una extensión del algoritmo BINPAT a sistemas heterogéneos con una variación denominada CoP-BINPAT que realiza una distribución de tareas basada en la métricas de rendimiento de cobertura de área propuesta denominada CoP.

4.1 Introducción

Los sistemas multi-UAV representan una evolución significativa en la tecnología de los UAS en la cual varias aeronaves colaboran para ejecutar una tarea particular. De acuerdo con (Zhou et al., 2020; Awasthi et al., 2023), múltiples investigaciones abordan el control de estos sistemas utilizando técnicas de comportamiento colaborativo inspiradas en la naturaleza como las colonias de hormigas, bandadas de pájaros, jaurías de caza y otras (como se puede observar en la Figura 4.1).

Por su parte, los sistemas homogéneos se componen de UAVs con características de rendimiento similares. Este tipo de sistemas presentan algunas ventajas con respecto a los heterogéneos. Debido a sus características y capacidades semejantes, los UAVs homogéneos pueden intercambiar información de forma eficiente y optimizar la coordinación y distribución de tareas de forma más simple. Además, la redundancia inherente en tener múltiples UAVs homogéneos proporciona una mayor fiabilidad en el sistema en su conjunto. Esta flexibilidad operativa resulta valiosa en entornos donde es necesario el despliegue rápido de la misión. Ejemplos de este requerimiento se pueden encontrar en proyectos de investigación para equipos de respues-

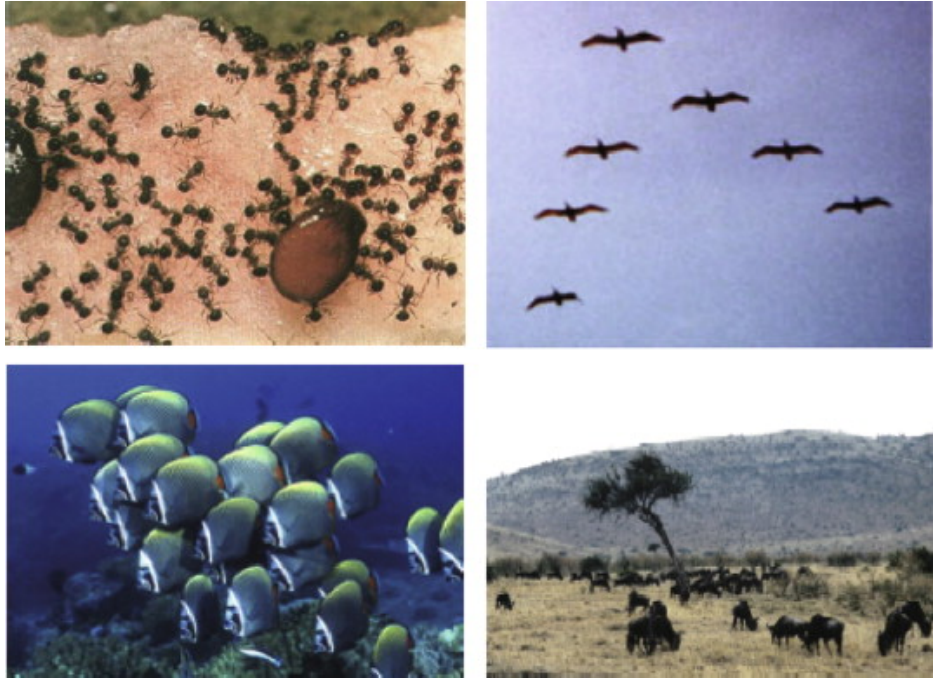


Figura 4.1: Figura tomada de (Y. Tan y Zheng, 2013) para ilustrar el comportamientos de inteligencia de enjambre de la naturaleza como las colonias de hormigas, bandadas de aves, cardúmenes de peces o jaurías de caza

ta rápida como IDEAL DRONE¹ o SAVER². Adicionalmente, estos sistemas tienen mejor escalabilidad ya que si la tarea cambia en términos de complejidad o tiempo de ejecución, basta con aumentar o disminuir aeronaves al sistema sin agregar mayores complejidades a los algoritmos de control o navegación.

Sin embargo, aunque las aeronaves de este tipo de sistemas tienen características similares, son sus inevitables diferencias lo que ha marcado la problemática de investigación en los últimos años. Por ejemplo, sabemos que los UAVs no pueden partir físicamente del mismo punto al mismo tiempo, y aunque lo hicieran, la distancia de ese punto de partida a la zona de trabajo específica de cada dron no será igual. Además, si consideramos que las aeronaves no están equipadas con sensores de detección de obstáculos tendríamos que planificar alturas de ascenso diferentes para evitar colisiones durante la misión.

Por otro lado, el surgimiento de nuevas tecnologías en las comunicaciones y los avances recientes en el campo de la robótica han permitido el desarrollo de sistemas multi-robóticos heterogéneos formados por diferentes tipos de robots diseñados para cooperar entre sí y con operadores humanos con el fin de desempeñar tareas complejas que requieren múltiples capacidades (Rizk et al., 2019). En este contexto, algunas competiciones internacionales como MBZIRC³ (ver Figura 4.2) o DARPA Subterranean challenge⁴ han propuesto a lo largo de su

¹<https://cordis.europa.eu/project/id/875183>

²<https://www.odenserobotics.dk/projects/saver-project/>

³<https://www.mbzirc.com/>

⁴<https://www.darpa.mil/program/darpa-subterranean-challenge>

historia varios retos para evaluar distintas aptitudes técnicas como la autonomía, percepción, comunicaciones y movilidad de sistemas compuestos por múltiples tipos de robots como UAVs, UGVs y USVs. Sin embargo, algunos autores en la literatura como (Y. Chen et al., 2018) destacan que un sistema robótico es heterogéneo incluso si está compuesto solo por aeronaves, siempre y cuando estas tengan diferentes prestaciones.



Figura 4.2: Figura tomada de (Martinez-Rozas et al., 2021) para ilustrar uno de los retos formulados en la competición MBZIRC 2020 que proponía una colaboración UAV-UGV para extinción de incendios en un entorno urbano con GPS denegado.

En algunos proyectos de investigación, como las propuestas de FASTER⁵ o MultiDrone⁶ los investigadores utilizan sistemas heterogéneos que incluyen aviones comerciales y prototipos propios. Además, en los últimos años se ha desarrollado en Europa el plan de despliegue del U-Space (Nafis et al., 2023), que consiste en un sistema que permita gestionar el tráfico de UAS de manera automática e integrada con la aviación tripulada, donde la gestión de aeronaves heterogéneas cumple un papel fundamental. También, algunos trabajos en la literatura (Göktoğan y Sukkarieh, 2009; Si et al., 2015) proponen arquitecturas de comunicación para coordinar helicópteros, multirotores y aeronaves de ala fija con la estación de control de tierra y con dispositivos satelitales (como se ilustra en la Figura 4.3).

Así, múltiples trabajos en la literatura apuestan por los sistemas heterogéneos porque permiten una mayor flexibilidad y reducen los costes de mantenimiento asociados (J. Chen, Zhang et al., 2021). Estas ventajas se observan especialmente en sistemas de gran escala. También puede demostrarse empíricamente que los sistemas homogéneos ideales no existen y que la diferencia entre las capacidades de cada aeronave, aunque sea mínima al principio, puede acentuarse en función de las condiciones del entorno de vuelo o del paso del tiempo. Por ejemplo, para un sistema homogéneo ejecutando una tarea de exploración con diferentes direcciones de barrido, los UAVs con ráfagas de viento favorables gastarán menos energía y alcanzarán la velocidad objetivo con mayor facilidad que aquellos que tengan ráfagas en contra, estos últimos incluso pueden llegar a reducir su velocidad por seguridad comprometiendo la homogeneidad en la distribución de la tarea.

Sin embargo, debido a la amplia gama de tipos de UAV existentes, se dificulta el diseño de un sistema que pueda aprovechar de manera efectiva las capacidades individuales de cada

⁵<https://www.faster-project.eu/>

⁶<https://multidrone.eu/>

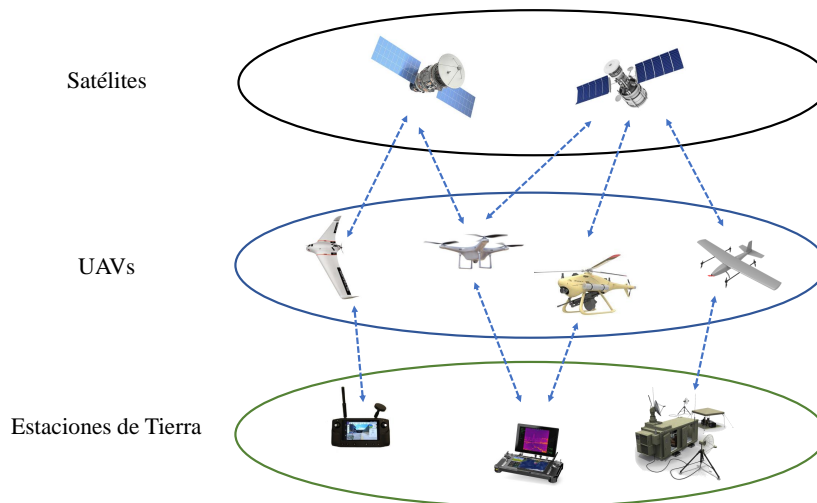


Figura 4.3: Esquema de un sistema compuesto por UAVs heterogéneos que incluye comunicación con la estación de tierra y enlaces satelitales.

aeronave. Este desafío se vuelve especialmente evidente en situaciones que requieren la toma de decisiones colaborativas entre múltiples UAVs, la planificación de misiones o vuelos de control de formación (Xiao et al., 2023). Por lo tanto, es importante acotar el problema dependiendo del campo de aplicación. Así, la planificación de rutas de cobertura de área para UAVs de ala rotativa tiene su complejidad en la planificación eficiente de las trayectorias considerando las diferentes capacidades de las aeronaves dentro de la flota. En este contexto, el estudio propuesto por (J. Chen, Du et al., 2021), caracteriza cada dron como un conjunto de tres variables de heterogeneidad que incluyen la velocidad de vuelo, ancho del barrido de los sensores a bordo y tiempo de vuelo antes de que se agote la batería.

4.2 Distribución de la trayectoria en sistemas multi-UAV homogéneos

4.2.1 Planteamiento del escenario

Se asume un grupo homogéneo de N UAVs de ala rotativa ubicados en diferentes posiciones iniciales con la misión de cubrir M áreas poligonales convexas en el espacio \mathbb{R}^2 definidas por un operador, donde $1 \leq M \leq 10$. Cada aeronave tiene cuatro grados de libertad, considerando que el barrido se realiza con una dirección fija de vuelo (sin giros en las curvas). Además, todos los UAVs cuentan con la misma estructura y equipamiento para las tareas de percepción aérea, por lo cual, la huella del sensor y la altura de vuelo serán iguales para todas las aeronaves durante la ejecución de la misión (como se observa en la Figura 4.4).

Además, todos los UAVs despegarán al mismo tiempo y a cada uno se le asignará una altura de seguridad de navegación antes de llegar a la zona de exploración. Una vez ahí descenderán a la altura de la misión de cobertura (de acuerdo al esquema de la Figura 4.5).

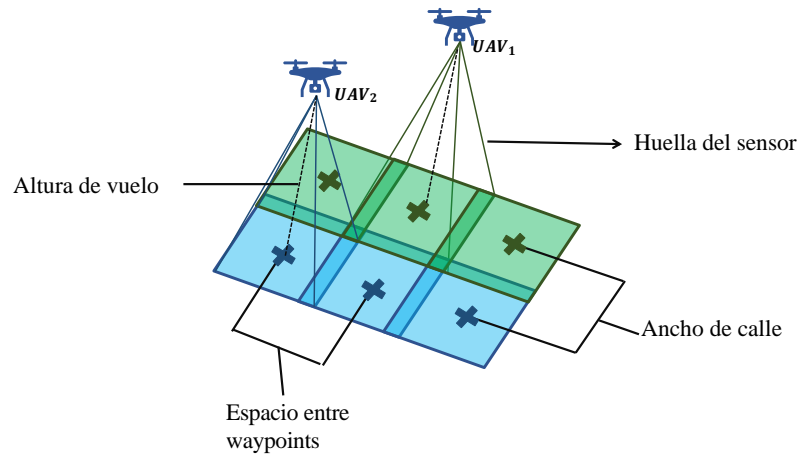


Figura 4.4: Esquema de la huella del sensor para un sistema homogéneo

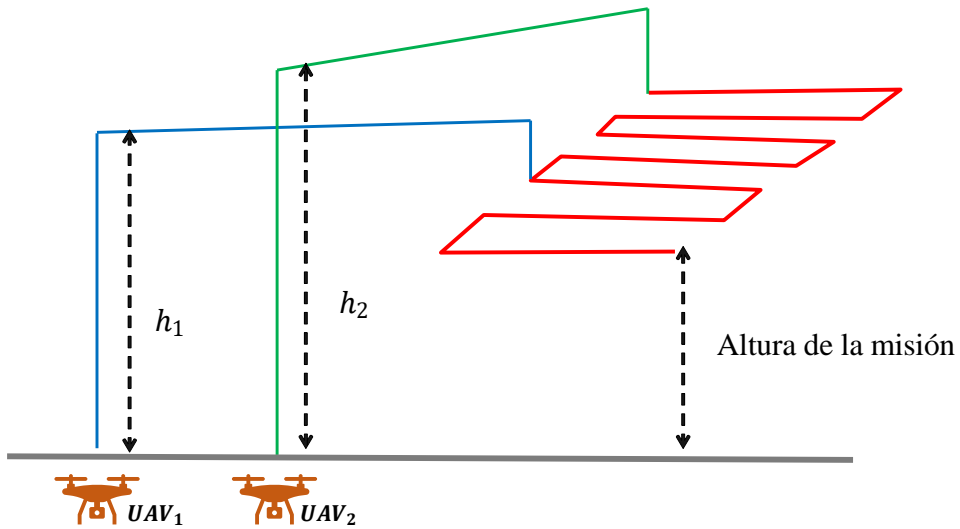


Figura 4.5: Esquema de alturas de seguridad previas a la ejecución de la misión

Finalmente, se asume que todos los UAVs tienen una velocidad constante y el tiempo de suministro de batería suficiente para completar la misión. También, siguiendo las directrices planteadas en el capítulo anterior 3, asumimos un entorno de trabajo libre de obstáculos. Así, el objetivo del algoritmo será encontrar la mejor distribución de las tareas de manera que se minimice el costo global de la misión.

4.2.2 Algoritmo Powell-BINPAT

El algoritmo de Powell-BINPAT presentado por primera vez en (Luna et al., 2022), se encarga de distribuir la ruta entre los múltiples agentes del sistema utilizando un enfoque de optimización ponderada de tareas (Gerkey y Matarić, 2004). Su objetivo principal es minimizar el costo global de la misión con un bajo costo computacional, y se divide en dos etapas: planificación de la trayectoria global y distribución de las rutas.

Para la planificación de la trayectoria global se utilizarán las estrategias propuestas en el capítulo 3 tanto para entornos de área simple como en múltiples áreas. Considerando que estos métodos utilizan como dato de entrada la ubicación una única aeronave, se utilizará para este caso el centroide del polígono formado por los UAVs (como se muestra en la Figura 4.6).

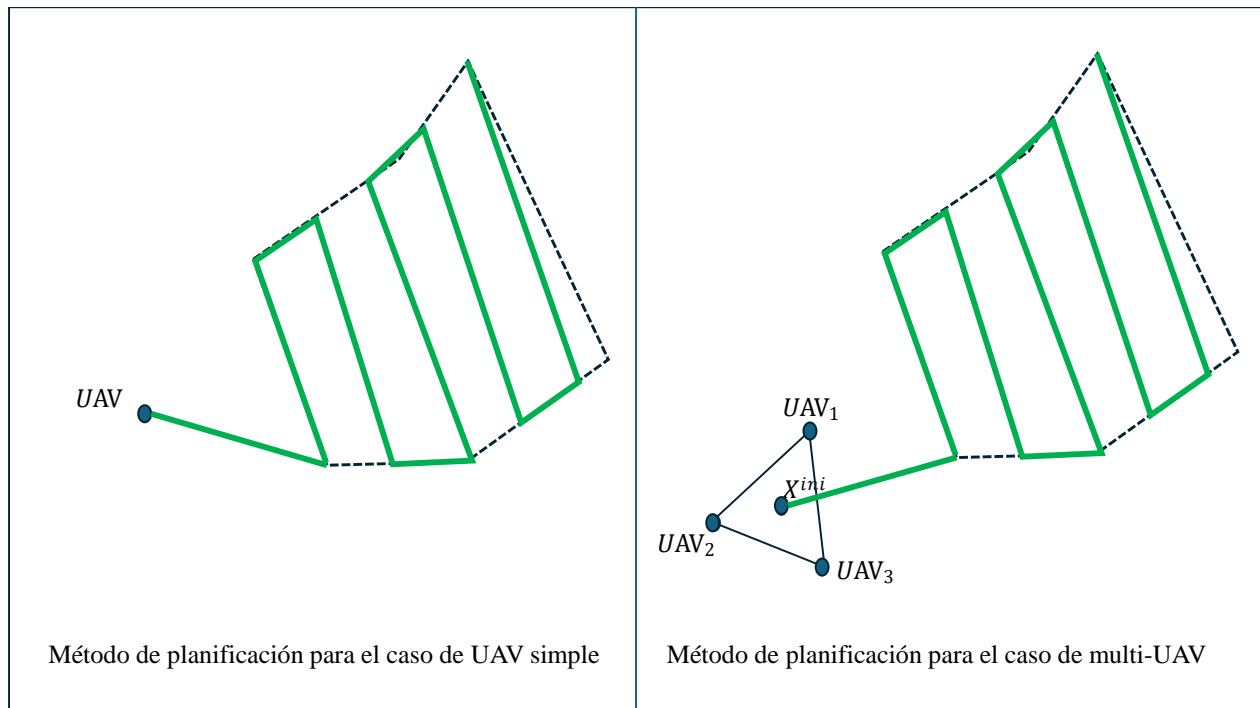


Figura 4.6: Esquema de método de planificación back-and-forth para multi-UAVs basado en el centroide del polígono formado por la posición de los UAVs

Sin embargo, este dato se utiliza únicamente como punto de partida para determinar las direcciones de barrido o el orden de exploración de las áreas, posteriormente, cada aeronave tendrá su propia ruta considerando su localización inicial, su altura de seguridad y su posición inicial.

Para la distribución de la ruta, dado un conjunto de waypoints W de una ruta planificada, se puede determinar el costo individual de cada aeronave durante la misión con la ecuación 4.1, asumiendo una velocidad constante.

$$C_m = \sum_{i=1}^{k-1} \|X_i - X_{i+1}\| \cdot T_m + Hd_m \quad (4.1)$$

En la ecuación 4.1, m representa el número de UAVs, k es el número de waypoints del vector W . De esta manera, C_m representa el costo de cada UAV como la suma de la distancia euclideana entre puntos consecutivos $\|X_i - X_{i+1}\|$ de la misión individual, T_m es un operador binario que ayuda a determinar el subconjunto correspondiente de la ruta de cada aeronave y Hd_m es el retraso causado por las diferencias en las localizaciones iniciales y alturas de seguridad. Powell-BINPAT se encarga de distribuir la misión para optimizar el costo global ajustando los costos individuales. Este proceso se describe en el Algoritmo 4.

Algoritmo 4: Algoritmo BINPAT

Data: U, W

Result: M

```

1  $GlobalCost \leftarrow GetCost(W)$ 
2  $Weights \leftarrow GetWeights(U)$ 
3  $C^{max} \leftarrow GlobalCost \cdot Weights$ 
4 foreach  $U_i \in U$  do
5      $Cost \leftarrow 0$ 
6     while  $Cost < C^{max}$  do
7          $W' \leftarrow GroupPoints(W)$ 
8          $Cost \leftarrow GetNewCost(W'_i)$ 
9     end
10     $CVec \leftarrow Append(Cost)$ 
11 end
12  $Assignments \leftarrow JV(W', U)$ 
13  $M \leftarrow SetAltitudes(Assignments, CVec)$ 
    
```

El Algoritmo 4 toma como datos de entrada el vector de posiciones de los UAVs U y el conjunto de waypoints de la ruta W y genera una matriz de waypoints M con los waypoints para cada aeronave. En primer lugar, se calcula el costo global de la misión con la función $GetCost()$ sumando la distancia total de la ruta previamente generada. A continuación, la función $GetWeights()$ obtiene una medida ponderada ω_k de las capacidades de los UAVs de forma que $0 \leq \omega \leq 1$. Para el caso homogéneo, esta medida será igual para todos los drones y se calculará como k^{-1} donde k es el número de UAVs, de tal manera que si existen 2 UAVs el peso de cada uno es 0.5, o 0.25 para el caso de 4 UAVs.

A continuación, se calcula el vector del costo máximo por UAV (C^{max}) como el producto punto entre el costo global ($GlobalCost$) y el vector de pesos ($Weights$). Este método considera a cada aeronave como un paquete ponderado utilizando una analogía con el problema de bin-packing (Garey y Johnson, 1990). De esta forma, los waypoints para cada dron se agrupan

en W' con la función $GroupPoints()$ mientras que el costo en distancia ($Cost$) de cada grupo sea menor que el costo máximo para ese UAV (C_i^{max}).

Considerando las diferentes posiciones iniciales en los drones, se aplica el algoritmo de asignación de Jonker-Volgenant ($JV()$) propuesto por (Jonker y Volgenant, 1988) para asignar los planes basados en las distancias de ida y vuelta para cada UAV. Finalmente, la función $SetAltitudes()$ calcula la altitud de seguridad para evitar colisiones. La salida de esta función es el conjunto final de waypoints para cada UAV.

Sin embargo, esta distribución no es completamente eficiente, por lo cual se se aplica el método de Powell (Powell, 1964) modificando los pesos iniciales de los drones en base a las restricciones de la ecuación 4.2 hasta minimizar el costo máximo individual de la misión (ecuación 4.3).

$$0 \leq W_m \leq 1, \forall m \in M \quad (4.2)$$

$$\text{mín} (\text{máx } C^m) \quad (4.3)$$

Esta técnica realiza un conjunto de operaciones de manera iterativa para encontrar el conjunto de pesos W_k mediante una búsqueda de línea bidireccional a lo largo de cada vector de búsqueda en el conjunto de soluciones N-dimensionales. Los resultados y la discusión del algoritmo Powell-BINPAT se presentan en el capítulo 6.

4.3 Distribución de la trayectoria en sistemas multi-UAV heterogéneos

4.3.1 Planteamiento del escenario

Para este planteamiento, se asume un un grupo de N UAVs heterogéneos de ala rotativa y guiados de manera autónoma con la misión de cubrir un área para sensado aéreo. De igual manera que para el caso homogéneo, se supone un entorno libre de obstáculos. En este caso se utilizarán las variaciones de heterogeneidad propuestas por J. Chen, Du et al., 2021. Es decir, cada dron tiene diferente velocidad de crucero y suministro de energía y aunque sus sensores a bordo son del mismo tipo, difieren en el tamaño de la huella. Por lo tanto, cada aeronave se caracteriza por el vector $U_i\langle x_i, y_i, v_i, s_i, t_i \rangle$, donde x_i e y_i son las coordenadas iniciales del UAV, v es la velocidad crucero, s es el área de la huella del sensor a bordo y t es el tiempo restante del suministro de energía (se muestra esquemáticamente en la Figura 4.7).

Es importante remarcar que el tiempo de vuelo y el desempeño de cada aeronave va a depender de otros factores como las condiciones de viento o el estado de los componentes de hardware. Sin embargo, al igual que otros estudios en la literatura para el caso heterogéneo (Berger et al., 2016, J. Chen, Zhang et al., 2021, y Xiao et al., 2023) se asume una velocidad constante y que el tiempo de suministro no cambia durante el vuelo. Además, las aeronaves del tipo ala rotativa pueden ser configuradas para volar marcha atrás por lo cual no se analizará el costo de girar en las curvas aunque también se realizarán pruebas con el algoritmo

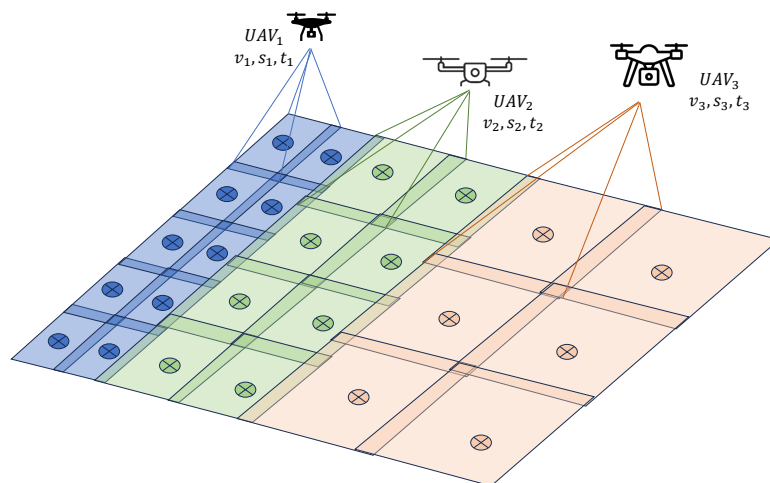


Figura 4.7: Esquema del problema de cobertura de área con aeronaves heterogéneas. Cada UAV tiene una velocidad v , tiempo de suministro de energía t y área de la huella del sensor s diferente, con un porcentaje de superposición. Los puntos marcados con \otimes corresponden a los waypoints de cada aeronave

de espiral presentado en el capítulo 3. Así, el objetivo será desarrollar las técnicas para determinar las trayectorias de vuelo eficientes para cada dron autónomo, con el fin de cubrir el área designada por el usuario en el menor tiempo posible, mientras se cumplen los límites de tiempo máximo de vuelo.

4.3.2 Algoritmo CoP-BINPAT

El algoritmo CoP BINPAT es una extensión del método BINPAT que resuelve el problema de distribución eficiente de las trayectorias en dos etapas que incluyen el cálculo de la métrica CoP y la distribución de las tareas.

Como se menciona en la sección anterior, la heurística utilizada por el algoritmo BINPAT consiste en tratar a cada aeronave como si fuera un paquete con determinadas capacidades para cumplir la trayectoria. Para UAVs homogéneos, se puede intuir que dicha trayectoria se repartirá de manera equitativa entre los agentes que conforman el sistema. Sin embargo, para el caso de heterogeneidad en tres características de vuelo diferentes (velocidad, huella y suministro de energía), es necesario encontrar una métrica que determine el aporte de cada paquete si se requiere aplicar esta misma heurística.

De esta manera se propone la métrica *CoP* (desempeño en vuelo de cobertura por sus siglas en inglés), que mide la capacidad de cada aeronave para realizar la misión global basada en sus tres variables de heterogeneidad. En primer lugar, tomando como base el tamaño de la huella del sensor, se determina el ancho de calles de la ruta. Este estudio no profundiza en el análisis del campo de visión del sensor para calcularlo, sino que utiliza el ancho de calle como un parámetro de entrada del algoritmo.

A continuación, suponiendo que tenemos m UAVs para explorar un área de n vértices, para

cada UAV se asume velocidad constante, por lo cual la velocidad de barrido de cada uno viene dada por la ecuación 4.4.

$$ScanVel_i = sw_i \cdot v_i \quad (4.4)$$

Donde sw_i (scan width) representa el ancho del barrido en metros de cada UAV y v_i la velocidad crucero en metros por segundo. De esta manera, la velocidad de barrido se expresa en m^2/s .

Luego, se calcula el área total de la región poligonal compuesta usando la fórmula de Gauss presentada en la ecuación 4.5.

$$Area = \frac{1}{2} \sum_{j=1}^n |X_{j+1}Y_j - X_jY_{j+1}| \quad (4.5)$$

Donde X_j y Y_j representan las coordenadas de cada vértice de la región a explorar. Además, se plantea que $X_{n+1} = X_1$ y $Y_{n+1} = Y_1$, para que la sumatoria concluya con el primer vértice (Persson et al., 2006).

Posteriormente, tomando en cuenta la velocidad de barrido y el área total, calculamos el tiempo que le tomaría a cada UAV realizar un barrido completo del área con la ecuación 4.6.

$$ScanTime_i = \frac{Area}{ScanVel_i} \quad (4.6)$$

Finalmente, obtenemos la calificación CoP_i de cada aeronave dividiendo el tiempo que le queda de suministro de batería sobre el tiempo que le tomaría a la aeronave cubrir toda el área (ecuación 4.7).

$$CoP_i = \frac{BattTime_i}{ScanTime_i} \quad (4.7)$$

Partiendo de la ecuación 4.7, se puede garantizar que el UAV cumple las restricciones de tiempo de vuelo máximo ya que su calificación CoP no permitirá que se le asigne una trayectoria mayor a la que puede recorrer.

Hasta este punto, hemos obtenido una medida de desempeño para cada UAV de acuerdo a su capacidad temporal para cubrir el área propuesta. A continuación, generamos el vector \vec{CoP} con los valores obtenidos para cada dron (ecuación 4.8).

$$\vec{CoP} = \{CoP_1, CoP_2, \dots, CoP_m\} \quad (4.8)$$

Sin embargo, es posible que de acuerdo a esta métrica, una aeronave esté lo suficientemente capacitada para cubrir por sí misma toda el área. Sin embargo, debido al requerimiento de utilizar múltiples UAVs, es necesario generar un vector normalizado \hat{CoP} , como se observa en la ecuación 4.9.

$$C\hat{o}P = \frac{C\vec{o}P}{|C\circ P|} \quad (4.9)$$

Donde

$$|C\circ P| = \sum_{k=1}^m C\circ P_k \quad (4.10)$$

Como se puede observar en las ecuaciones previas (4.4-4.10), el vector $C\vec{o}P$ involucra a todas las métricas de heterogeneidad planteadas y refleja la capacidad ponderada de cada aeronave para cubrir la misión.

Por otra parte, uno de los retos principales en la planificación de trayectorias para un sistema heterogéneo es la diferencia del área de la huella del sensor. Por este motivo, en la planificación de las trayectorias se propone dividir la región principal en subregiones, agrupando los UAVs según el ancho de barrido. Luego, para la asignación de trayectorias en cada subregión se aplica el algoritmo de BINPAT con las ponderaciones del vector normalizado $C\hat{o}P$. El proceso completo se describe en el Algoritmo 5.

Algoritmo 5: Algoritmo CoP-BINPAT

Data: $U, Region$

Result: M

- 1 $Area \leftarrow CalculateArea(Region)$
 - 2 $C\hat{o}P \leftarrow GetCoPNorm(U, Area)$
 - 3 $SortedU \leftarrow SortByScan(U)$
 - 4 $UWeights \leftarrow GetWeights(SortedU, C\hat{o}P)$
 - 5 $SubRegions \leftarrow PolygonDecomposition(Region, UWeights)$
 - 6 $Routes \leftarrow PlanRoutes(SubRegions, SortedU)$
 - 7 $Assingments \leftarrow BINPAT(Routes, UWeights, SortedU)$
 - 8 $WP \leftarrow GenerateWaypoints(Assingments)$
-

En el Algoritmo 5, el primer paso es calcular el área de la región con la fórmula propuesta previamente en la ecuación 4.5. A continuación, se obtiene el vector normalizado $C\vec{o}P$ utilizando las fórmulas planteadas en las ecuaciones 4.6-4.9.

Luego, se agrupan los drones del vector U de acuerdo al ancho de barrido generando un nuevo vector $SortedU$. Una vez obtenido este vector, se reordena también el vector $C\hat{o}P$ para obtener la ponderación de cada UAV para la tarea de exploración. Posteriormente se generan subregiones considerando la ponderación de cada UAV y agrupando los anchos de barrido iguales. Por ejemplo, suponiendo que tenemos 7 UAVs con 3 anchos de barrido diferentes, la región principal se dividiría en 3 subregiones, y el tamaño de cada subregión dependerá de la sumatoria del CoP de las aeronaves que lo componen, para mayor CoP , mayor área. Esta descomposición poligonal ponderada se realiza utilizando el algoritmo de Hert (Hert y Lumelsky, 1998). Como una etapa de procesamiento previo, es necesario que las subregiones sean convexas. Por este motivo, se utiliza la variación del algoritmo de Hert que utiliza triangulación Delaunay (Delaunay et al., 1934). De igual manera, si la región inicial en

cóncava, es necesario aplicar la técnica de convex hull (Barber et al., 1996) para transformarla en convexa.

Posteriormente, se generan las rutas de cobertura para cada subregión, en este caso, se puede utilizar la técnica de back-and-forth (Huang, 2001) o la espiral que se describe en el capítulo 3. Estos métodos toman como parámetros de entrada los vértices de las subregiones y el ancho de scan de las aeronaves previamente clasificadas para generar un vector de rutas.

Después, en cada ruta dentro de la subregión, se aplicará el método BINPAT (descrito en el Algoritmo 4) entre los UAVs con el mismo ancho de barrido. En este caso los pesos de entrada se determinarán en función de su valor de CoP . El resultado de este paso generará un vector de asignaciones de las rutas para cada dron, que finalmente se traducirán en waypoints en el último paso. Los experimentos y resultados de este algoritmo se presentan en el capítulo 6.

Capítulo 5

Gestión dinámica de trayectorias en sistemas multi-UAV

Este capítulo describe la arquitectura desarrollada para la monitorización del estado de los UAVs y de la misión y los métodos propuestos para la detección de fallos y la replanificación en vuelo. De esta manera, se ha dividido el capítulo en tres partes. En la introducción se plantea una visión general del problema y las necesidades que motivan a controlar a los multi-UAV de manera dinámica. A continuación, el planteamiento del escenario describe de manera específica el problema a resolver. Finalmente, en la arquitectura de control propuesta se presenta la arquitectura y soluciones desarrolladas para resolver dicho problema.

5.1 Introducción

En el problema de asignación de tareas multi-robot (MRTA, por sus siglas en inglés), dado un conjunto de n robots y un conjunto de m tareas, además de una estimación del rendimiento de cada robot en la ejecución en dichas tareas, el objetivo es encontrar la asignación correcta de tareas a cada robot de manera que se maximice el rendimiento. Sin embargo, de acuerdo con (Gerkey y Mataric, 2003), este desafío no puede tratarse de manera estática, ya que varía en el tiempo de acuerdo a distintos fenómenos que pueden incluir las condiciones del entorno o la naturaleza de la misión. Así, la investigación presentada por (Peng et al., 2021), muestra tres tipos de eventos que pueden dar lugar a comportamientos dinámicos en este tipo de sistemas. El primer tipo de eventos es el surgimiento de nuevas tareas conforme avanza la misión, muy común en tareas de exploración. El segundo tipo corresponde al fallo de uno o más UAVs que obligan al sistema a replanificar la misión para ejecutar las tareas que no se han finalizado. Y el tercer tipo corresponde a la actualización de las tareas, cuando la naturaleza de la misión cambia por requerimientos del operador.

Por otro lado, en múltiples aplicaciones industriales, incluso los sistemas de UAV único pueden estar sujetos a cambios en la misión sobre todo si se navega en entornos no controlados. Por ejemplo, el estudio presentado en (Peña et al., 2022), presenta una aeronave enfocada en respuesta rápida frente a incendios con la capacidad de cambiar el plan de vuelo mediante

comandos intuitivos para un operador como se observa en la Figura 5.1.

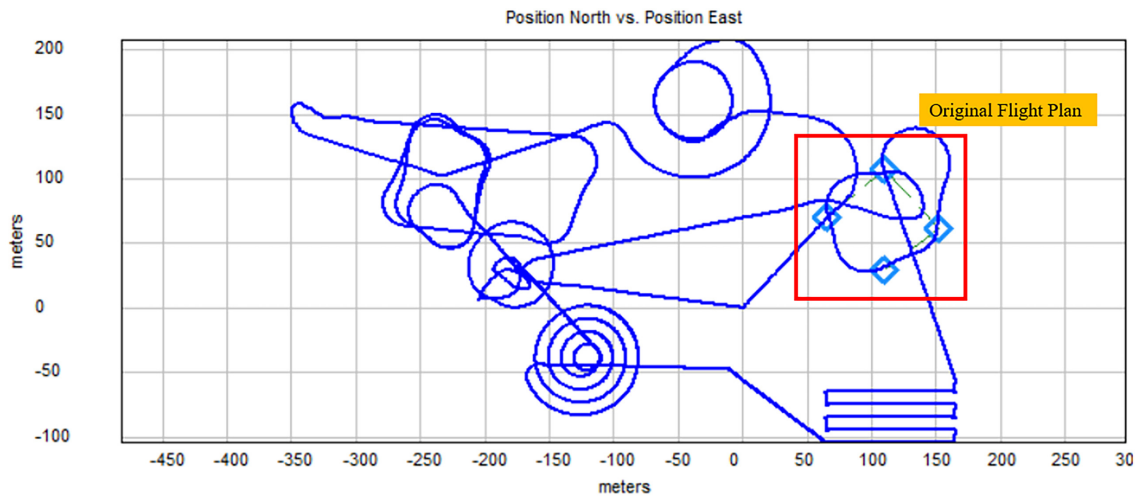


Figura 5.1: Figura tomada de (Peña et al., 2022) para demostrar un registro de rutas de vuelo de un sistema de UAV único con capacidad de replanificación dinámica de la misión

En la Figura 5.1, el dron recibe un plan de vuelo inicial formado por cuatro puntos. Sin embargo, esta ruta cambia durante el vuelo de acuerdo a los requerimientos del operador sin necesidad de aterrizar la aeronave y volver a generar el plan de vuelo.

Por otra parte, los estudios presentados por (Daramouskas et al., 2023; Hu et al., 2022), resaltan las capacidades de los sistemas compuestos por enjambres de vehículos no tripulados (UV, por sus siglas en inglés) con respecto a sistemas formados por un único vehículo. Una de estas capacidades es adaptación en tiempo real a situaciones que cambian constantemente, y otra, su redundancia y resiliencia cuando la pérdida de un vehículo no afecta a la efectividad global de la ejecución de la misión. Sin embargo, como se ha revisado en capítulos previos, no todas las investigaciones presentadas en la literatura presentan sistemas con la habilidad de ejecutar estos comportamientos, por lo cual es necesario plantear arquitecturas de control que puedan gestionar el comportamiento dinámico del sistema entre los drones del enjambre (Luna et al., 2021).

En este contexto, el problema CPP maneja sus propios desafíos en cuanto al comportamiento dinámico. Partiendo de la necesidad de que toda la región sea cubierta por el sistema multi-UAV, se asigna un primer conjunto de waypoints en la etapa de planificación. Si ocurre un evento no planificado durante la ejecución de la misión (como la pérdida o adición de un dron), es necesario que el resto de los agentes reorganicen su misión sin aterrizar para volver a replanificar la misión y sin cubrir puntos que ya han sido cubiertos. Por este motivo, este proceso debe realizarse en tiempo real contando con algoritmos que se ejecuten de con el menor tiempo de CPU posible. Así, se debe plantear una arquitectura de control capaz de controlar el sistema frente a eventos inesperados y eventos generados por el operador que no corresponden con la planificación inicial.

El sistema propuesto en esta investigación, presentado por primera vez en (Luna et al., 2024) propone una arquitectura computacional aplicable a problemas de cobertura de áreas separadas con sistemas multi-UAV homogéneos, y se describe como un conjunto de procesos inter-

conectados con sus algoritmos. Así, presentamos un enfoque híbrido centralizado-distribuido que maneja la planificación global de la misión de manera centralizada y la planificación local de manera distribuida. No se ha asumido el caso heterogéneo debido a que los algoritmos de replanificación de área resultan más complejos para este tipo de sistemas.

5.2 Escenario planteado

De manera similar al problema de asignación de trayectorias en sistemas homogéneos, se asume un grupo de N UAVs homogéneos de ala rotativa con diferentes posiciones iniciales. La misión de estos vehículos es cubrir M áreas poligonales convexas en el espacio \mathbb{R}^2 separadas entre sí, las cuales han sido previamente definidas por un operador. Por motivos prácticos el número de áreas a explorar estará limitado entre 1 y 10. También, cada aeronave tiene cuatro grados de libertad y mantiene la dirección de vuelo fija con respecto a la dirección de barrido, por lo cual, el costo de giro es mínimo. Por otro lado, debido a la naturaleza homogénea del sistema, la huella del sensor y la altura de vuelo será la misma para todas las aeronaves. Además, se requiere que el sistema tenga un comportamiento dinámico, es decir, que si uno o varios UAVs presentan un comportamiento inesperado, todo el sistema debe replanificar la misión.

5.3 Arquitectura de control propuesta

La arquitectura de control propuesta se observa en la Figura 5.2.

La Figura 5.2 describe el enfoque híbrido distribuido-centralizado. Así, el *Sistema de Planificación Global* genera un plan global para todos los UAVs, pero tiene la capacidad de modificarlo dinámicamente cuando ocurren determinados eventos. Por otro lado, el *Sistema de Ejecución Local* se encarga de la ejecución del plan de cada aeronave, monitoriza el desarrollo de la misión y la adapta localmente si el sistema global enfrenta algún evento.

De esta manera, el *Sistema de Planificación Global* recibe la información del operador humano y genera un plan global repartido entre todos los drones utilizando el módulo de *Planificación Multi-UAV*. A continuación, el *Sistema de Ejecución Local* de cada UAV recibe el plan individual correspondiente a través del módulo de *Planificación Local*. Luego, el módulo de *Control de Ejecución* lleva a cabo el plan de vuelo enviando los comandos adecuados al sistema de control de la aeronave.

Durante la ejecución del plan de vuelo, el módulo de *Monitor de Ejecución* opera de manera paralela al *Control de Ejecución* vigilando el comportamiento de cada plataforma aérea para detectar situaciones inesperadas o eventos relevantes fuera de la planificación inicial. Así, cuando el sistema detecta un evento se puede manejarlo de manera local a través del módulo de *Planificación Local* o de manera global con el módulo de *Re-planificación en vuelo*. De esta manera, se genera un nuevo plan de vuelo acorde a la situación que se experimente en ese momento.

En los siguientes apartados se explica cada uno de los módulos y su función dentro de todo el sistema. Además, los resultados de los experimentos del sistema dinámico se presentan en

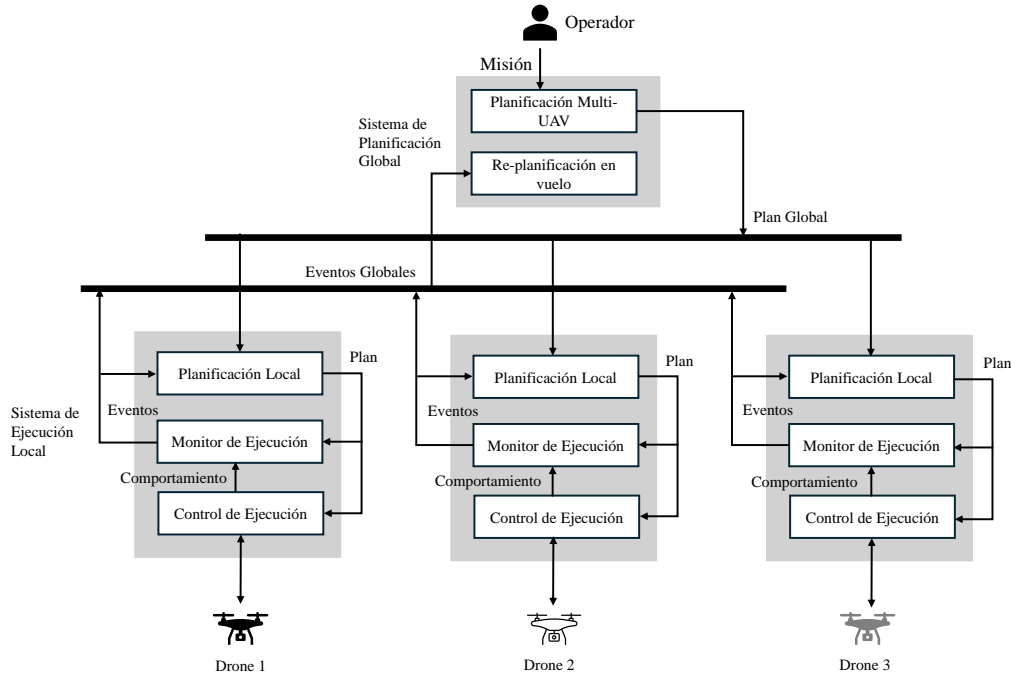


Figura 5.2: Arquitectura de control propuesta para planificación dinámica. Los rectángulos representan un proceso de información mientras que las flechas describen flujos de información.

el capítulo 6.

5.3.1 Planificación multi-UAV

Para la parte de planificación y asignación de trayectorias dentro de la arquitectura propuesta, se utiliza el mismo paradigma que se ha manejado a lo largo del documento de la tesis, con enfoque en las tres principales tareas que incluyen generación de la ruta de cobertura, planificación multi-área y asignación de tareas. Así, se han seleccionado diferentes técnicas de acuerdo a los requerimientos de la Tabla 5.1.

Tabla 5.1: Requerimientos algorítmicos de la planificación Multi-UAV para el sistema dinámico

Requerimientos generales	CPP		PMA		DT	
	SP	B&F	TSP	NN	P-BP	C-BP
Movimiento en los 4 grados de libertad	■	■				■
Sistema homogéneo					■	■
Rutas de menor costo	■	■	■	■		■
Tiempos de CPU bajos	■	■	■	■	■	■
Facilidad de implementación	■	■	■	■	■	■

En la Tabla 5.1, se ha dividido los requerimientos por tareas y por propuestas algorítmicas. CPP se refiere a la tarea de generación de trayectorias de cobertura, PMA es la planificación

en múltiples áreas y DT es la distribución de tareas. Se representan con color verde los algoritmos más adecuados para resolver el requerimiento, con color amarillo si el algoritmo puede resolverlo aunque de manera ineficiente y con color gris si el requerimiento no aplica para el algoritmo planteado.

En primer lugar, para la generación de trayectorias en cada área se utilizará el enfoque back-and-forth (B&F) en lugar de espiral (SP), considerando que según el planteamiento del problema las aeronaves se mueven en todas las direcciones de sus cuatro grados de libertad, es decir, no es necesario que el UAV navegue apuntando siempre hacia los waypoints. En este contexto, las rutas generadas con este método son más eficientes que las generadas en espiral.

A continuación, la exploración de múltiples áreas se lleva a cabo mediante el algoritmo de vecino más cercano (NN) en lugar del problema del viajante (TSP). Este método se ha escogido debido al requerimiento de minimizar los tiempos de ejecución de la operación. Además, el número de áreas a explorar comúnmente es menor a 10 en entornos industriales, condición bajo la cual las rutas generadas por este algoritmo tienen costos relativamente bajos. También, la implementación de este algoritmo en un nuevo entorno de desarrollo es mucho más rápida debido a que no depende de librerías externas que no podrían estar disponibles al migrar el programa a otro tipo de lenguaje.

Finalmente, la distribución de rutas entre los drones que participan en la misión se ha realizado mediante el algoritmo de Powell-BINPAT (P-BP) en lugar de CoP-BINPAT(C-BP), tanto en la etapa de planificación como re-planificación debido a su tiempo de CPU eficiente y a la caracterización homogénea del sistema (aunque teóricamente se podría aplicar CoP BINPAT en sistemas homogéneos). Otro factor que determinó la implementación de esta técnica dentro de la arquitectura es la optimización del costo de la misión del sistema multi-UAV reflejada en los coeficientes de variación bajos en las pruebas.

5.3.2 Monitorización de ejecución

El objetivo del sistema de monitorización es comprobar la ejecución del plan de vuelo de cada dron y reportar si la tarea se ha ejecutado con éxito o si existió un problema durante la ejecución por una situación inesperada que no corresponde a la planificación inicial. Estas situaciones incluyen cambios de velocidad durante el vuelo, pérdidas de batería, errores en la toma de datos o incluso la adición de una aeronave al sistema. En términos generales, estas situaciones se pueden presentar debido a cambios en el entorno o cambios en el UAV.

En el sistema propuesto, esta tarea se realiza de manera distribuida, esto es, que cada dron monitoriza su propio estado durante la ejecución del plan. Para esto, el módulo de monitorización recibe como entradas el plan de vuelo (codificado en tareas básicas para el UAV como despega, ve a este punto, aterriza, etc.), y mensajes periódicos sobre el estado del vehículo y el estado de ejecución de los comportamientos del plan de vuelo. Varios ejemplos se pueden visualizar en la Tabla 5.2.

La Tabla 5.2 muestra ejemplos de eventos que pueden generarse durante este proceso. Así, la categoría *UAV perdido* corresponde a una situación en la que el UAV no sigue la ruta planificada y se ha desviado de la planificación inicial. Por otro lado, el evento de UAV disponible,

Tabla 5.2: Eventos de replanificación

Evento	Descripción
UAV perdido	La aeronave se ha desviado de la ruta con mucha diferencia
Deterioro de batería	La batería del UAV ha tenido un descenso significativo de energía
Control Manual	El piloto humano ha debido tomar el mando de la aeronave por problemas técnicos
UAV disponible	Un nuevo UAV puede ser incorporado al sistema

contempla la adición de una nueva plataforma al sistema, para lo cual es necesario replanificar la misión y distribuirla para el nuevo conjunto de UAVs. Otros eventos incluyen el control manual y el deterioro de batería que son condiciones que no permiten a la aeronave continuar su misión y bajo las cuales también es necesario ejecutar el algoritmo de replanificación.

El proceso de monitorización de la ejecución de la misión se ejecuta continuamente en lazo y durante cada ciclo analiza la presencia de eventos potenciales por prioridades, este proceso se detiene en caso de detectar un evento. Si se producen múltiples eventos al mismo tiempo, el sistema presta atención al de mayor prioridad y maneja los demás eventos en los siguientes ciclos.

Además, los eventos se pueden detectar durante el vuelo de diferentes maneras. Por ejemplo, eventos como *UAV perdido* o *Deterioro de batería* se detectan de manera local mediante operaciones específicas del propio sistema. Otros eventos se activan al recibir una señal externa, como por ejemplo, *control manual* o *UAV disponible*.

Para verificar que la ruta que sigue el dron es correcta, el monitor de ejecución observa los waypoints visitados y por visitar (esta información se obtiene del plan de vuelo y la monitorización del comportamiento). En esta etapa, se asume que un waypoint ha sido visitado correctamente si el UAV ha volado sobre él. Para determinar si un UAV se ha perdido en vuelo, definimos γ como el umbral de distancia en metros que un UAV se puede desviar de su trayectoria. Para evaluar esta condición, calculamos la distancia a la trayectoria (D_{tr}) como la distancia del punto actual ($P_{current}$) a la recta formada por el waypoint siguiente y el anterior ($\overline{P_{last}P_{next}}$), dado por la ecuación 5.1

$$D_{tr} = \frac{(x_{next} - x_{last})(y_{last} - y_{current}) - (x_{last} - x_{current})(y_{next} - y_{last})}{\sqrt{(x_{next} - x_{last})^2 + (y_{next} - y_{last})^2}} \quad (5.1)$$

En la ecuación 5.1, $x_{current}$ e $y_{current}$ son coordenadas de la proyección 2D de la aeronave ($P_{current}$), mientras que $P_{last}(x_{last}, y_{last})$ son las coordenadas del punto visitado previo y $P_{next}(x_{next}, y_{next})$ son las coordenadas del punto por visitar. Esta condición se detecta por la función $dev(P_{current}, P_{last}, P_{next})$ (ver ecuación 5.2), que calcula cuando la aeronave se ha desviado de su trayectoria lo suficiente para considerarlo perdido.

$$dev(P_{current}, P_{last}, P_{next}) = \begin{cases} 1, & \text{if } D_{tr} > \gamma \\ 0, & \text{si no} \end{cases} \quad \text{Eq.(5.1)} \quad (5.2)$$

Adicionalmente, para determinar si el UAV está navegando en la dirección correcta, se calcula constantemente su distancia d_{curr} al siguiente waypoint utilizando la ecuación 5.3

$$d_{curr} = \|P_{next} - P_{current}\| \quad (5.3)$$

También se calcula la distancia de la aeronave al siguiente waypoint (d_{prev}) tomando como referencia la posición en el instante previo P_{prev} y la posición siguiente (P_{next}) utilizando la ecuación 5.4

$$d_{prev} = \|P_{next} - P_{prev}\| \quad (5.4)$$

En este sentido, si la distancia al siguiente waypoint aumenta en lugar de disminuir y este aumento supera el umbral γ , el sistema reporta que se ha perdido el UAV (Ver ecuación 5.5).

$$ret(d_{curr}, d_{prev}) = \begin{cases} 1, & \text{if } (d_{curr} > d_{prev}) \wedge \|d_{curr} - d_{prev}\| > \gamma \\ 0, & \text{caso contrario} \end{cases} \quad (5.5)$$

Finalmente, definimos δ como el umbral de distancia para determinar que un waypoint ha sido visitado, la cual se evalúa mediante la función *LblVis* que se muestra en la ecuación 5.6)

$$LblVis(P_{current}, P_{last}) = \begin{cases} 1, & \text{if } \|P_{current} - P_{last}\| > \delta \\ 0, & \text{caso contrario} \end{cases} \quad (5.6)$$

Resumiendo, el umbral γ nos ayuda a determinar si un UAV debe ser etiquetado como perdido mientras que δ nos ayuda a monitorizar si un waypoint ha sido ya visitado. En implementaciones reales, estos valores no son fijos y deben ser ajustados por el operador basado en rango de operación del plan de vuelo.

5.3.3 Planificación local

El objetivo del módulo de planificación local es generar un plan en detalle para cada vehículo particular. En el sistema propuesto, esta tarea se realiza de manera distribuida, esto es, un planificador local para cada dron. De esta manera, el objetivo de este módulo es extraer el plan global dado en waypoints y transformarlo en una representación de comportamientos de robot (despegar, navegar hacia un waypoint, aterrizar).

Además, este módulo se encarga de generar nuevas acciones locales diferentes a las del plan global, como respuesta a ciertos eventos. La Tabla 5.3 muestra ejemplos de este tipo de acciones locales.

Tabla 5.3: Ejemplos de acciones ejecutadas por el módulo de planificación local

Evento	Acciones locales para el UAV afectado	Acciones locales para el resto de UAVs
UAV perdido	Se comanda vuelta al punto de despegue	Esperar por un nuevo plan
Deterioro de la batería	Se comanda aterrizaje inmediato del UAV	Esperar por un nuevo plan
Control manual	Se borra el plan local para el UAV	Esperar por un nuevo plan
UAV disponible	Esperar por un nuevo plan	Esperar por un nuevo plan

En la Tabla 5.3, para cada evento, existen acciones generadas por el planificador local del UAV afectado y otras acciones generadas por los planificadores locales del resto de aeronaves. Por ejemplo, si se detecta deterioro en la batería, el planificador local de la aeronave afectada comanda un aterrizaje inmediato, mientras que el resto de planificadores esperan por el nuevo plan. La espera del nuevo plan global es una acción importante en la sincronización del proceso de replanificación.

En primer lugar, el vehículo v detiene la ejecución de su plan local actual p durante un lapso breve de tiempo δt . Esto se realiza para dar tiempo al planificador global de generar un plan nuevo, sin embargo, durante este tiempo es importante que la posición de los drones no cambie, ya que es la que toma el planificador global como referencia. Si el planificador global ha generado un nuevo plan p' para el vehículo v , dicho vehículo continuará con la misión después del tiempo determinado. Caso contrario continuará ejecutando el plan previo.

5.3.4 Replanificación en vuelo

El objetivo del módulo de replanificación adaptar la ejecución de las misiones del sistema durante situaciones que difieren de la planificación inicial. Cuando el módulo de monitor de ejecución detecta un evento significativo, el *planificador global* ejecuta el algoritmo de replanificación para cubrir los waypoints que no han sido visitados por las aeronaves. Así, el sistema etiqueta a los puntos de paso del plan de vuelo en dos categorías. La categoría de *waypoints de inspección* se refiere a aquellos puntos que están dentro de una de las áreas de cobertura y que forman parte de ruta a inspeccionar, y *waypoints de soporte* son aquellos que ayudan al dron a llegar a su destino pero que no forman parte de la inspección, es decir, están fuera del área de cobertura y cumplen otros objetivos como definir las alturas de seguridad para cada aeronave y que no exista riesgo de colisión entre ellas. Además, el monitor de ejecución etiquetará los waypoints como *visitados* o *no visitados* de acuerdo al criterio mencionado en la ecuación 5.6.

De esta manera, el planificador concatena los waypoints de *inspección no visitados* para generar una nueva ruta global, que luego se distribuirá de manera eficiente utilizando el algoritmo de Powell-BINPAT (como se observa en la Figura 5.3).

En la Figura 5.3a, los puntos visitados se han marcado con una "x" mientras que los puntos no visitados se clasifican en las dos categorías previamente mencionadas. De esta forma, los

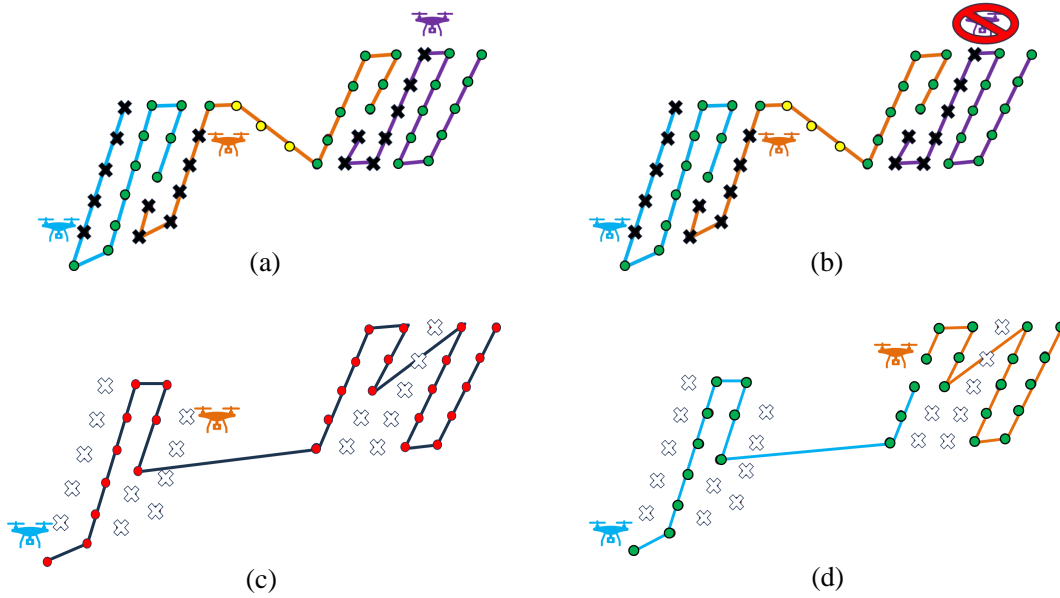


Figura 5.3: Esquema de replanificación en vuelo. a) Los UAVs ejecutan la ruta normalmente. b) Pérdida del UAV morado en un momento determinado de la misión. c) Replanificación de la nueva ruta global. d) Distribución homogénea de tareas con el algoritmo Powell-BINPAT

puntos verdes corresponden a los *puntos de inspección* y los puntos amarillos son los puntos de soporte. Durante el desarrollo, un evento inesperado es perder el dron morado (como se observa en la Figura 5.3b), para lo cual el sistema debe actuar de manera dinámica. En la Figura 5.3c, se puede observar la generación de la ruta global concatenando los waypoints de inspección *no visitados*. Finalmente, se aplica el algoritmo Powell-BINPAT para generar la nueva distribución de tareas (ver Figura 5.3d). Debido al bajo costo computacional de las técnicas aplicadas durante el proceso de replanificación, este proceso se ejecuta de manera casi inmediata.

En caso de añadir una nueva aeronave al sistema, el módulo de replanificación toma nuevamente los puntos de *inspección no visitados* y genera una misión repartida entre todos los UAVs (incluyendo al que ha ingresado), y para asegurar la eficiencia de la misión, el sistema calcula el costo de replanificar toda la misión. Así, siendo W^O una secuencia de waypoints de la trayectoria, el costo ponderado de la trayectoria se calcula como:

$$C\omega(W^O) = \text{máx}(length(W_i)); 1 \leq i \leq n \quad (5.7)$$

Es decir, la longitud de la ruta más larga en la secuencia. Finalmente, con el costo ponderado y tomando como referencia un umbral ϕ ; se genera una nueva secuencia W^N ; la cual se ejecutará solo sí:

$$\frac{C\omega(W^O)}{C\omega(W^N)} \leq \phi \quad (5.8)$$

Este proceso también se repite en los casos en los que un UAV ha terminado su misión y existen tareas pendientes que corresponden a los demás UAVs. En este escenario, si el costo para ejecutar la misión es menor, entonces el sistema replanifica con todos las aeronaves, caso contrario se mantiene la planificación inicial y el dron que ha terminado vuelve al punto de despegue.

5.3.5 Control de ejecución

La función del módulo de control de ejecución es llevar a cabo el plan asignado a cada UAV. Para esto, interpreta los comportamientos para generar comandos de control sobre la plataforma aérea. También, este proceso informa periódicamente el estado de la ejecución de la misión de cada dron lo cual se puede utilizar en el módulo de monitoreo de la ejecución.

En esta investigación, se ha implementado el control de ejecución utilizando el entorno de desarrollo de Aerostack2 (Fernandez-Cortizas et al., 2023) especializado en robótica aérea y brinda ventajas a la hora de su integración. Aerostack2 es modular y se puede aplicar a distintos tipos de plataforma tanto reales (por ejemplo, DJI Matrice, Crazyflie, etc.) como simuladas (Gazebo). Además, está desarrollado dentro del entorno de ROS2, lo cual facilita la integración de nuevos comportamientos y módulos de programación.

Aerostack2 incluye un componente de software especializado llamado *intérprete del plan de misión* que facilita la ejecución de planes. Este intérprete permite una adaptación incremental de los planes de misión durante su ejecución, que facilita el trabajo al módulo de replanificación descrito previamente. Así, durante la ejecución de un plan específico p , el intérprete puede recibir como entrada una orden (junto con un plan opcional p') que modifica la ejecución de p . Ejemplos de órdenes son: *insertar p'* (ejecutar p' y, luego, el resto de p), *agregar p'* (ejecutar p' al final de la ejecución de p), *pausar* (pausar la ejecución de p), etc.

Durante la ejecución de un plan de vuelo, el *intérprete del plan de misión* de Aerostack2 publica periódicamente datos sobre el estado de ejecución de la secuencia de comportamientos del robot. Estos datos incluyen el comportamiento que se está ejecutando actualmente, la retroalimentación de la ejecución del comportamiento (que es particular para cada comportamiento) y el número de comportamientos completados y pendientes en el plan.

Capítulo 6

Experimentos y discusión

En este capítulo se describen los experimentos realizados y los resultados obtenidos para evaluar el rendimiento de los métodos desarrollados a lo largo de la tesis doctoral. Para asegurar una validación exhaustiva de los algoritmos propuestos, se llevaron a cabo tres tipos de pruebas: pruebas de rendimiento, pruebas de simulación y pruebas en entornos reales. Las pruebas de rendimiento se realizan con el fin de obtener un análisis detallado de diversas variables de interés y compararlas con otras propuestas en el estado del arte. Los experimentos de simulación se realizan con software que emula el comportamiento físico de los UAVs para su observación en entornos controlados. Finalmente, las pruebas en entornos reales proporcionaron una evaluación práctica y robusta de la aplicabilidad y eficiencia de los algoritmos en condiciones del mundo real. A continuación, se detallan los procedimientos experimentales, los escenarios utilizados, los parámetros configurados y los resultados obtenidos, destacando las ventajas y limitaciones observadas en cada caso.

6.1 Configuración de los experimentos

6.1.1 Experimentos de evaluación del rendimiento

Para las pruebas de evaluación del rendimiento, se probaron los algoritmos de manera teórica, analizando variables fundamentales como el cálculo de distancias recorridas por los UAVs para determinar el costo de las misiones, o el tiempo de CPU obtenido al poner los diferentes métodos a prueba. Estas variables nos ayudarán a establecer métricas de rendimiento y comparar los resultados con propuestas similares presentadas en la literatura. Para estas pruebas, se utilizó una laptop con CPU Intel Core i7-7700HQ, 3.8 GHz, y 16 Gb de RAM ejecutando Ubuntu 22.04¹. Además, los algoritmos fueron desarrollados y probados en Python 3.10² utilizando múltiples librerías dentro de este entorno de las cuales se pueden destacar

¹<https://releases.ubuntu.com/jammy/>

²<https://www.python.org/downloads/release/python-3100/>

Numpy³, Matplotlib⁴, Scipy⁵, y otras. Este tipo de pruebas se aplicaron en los algoritmos de planificación de cobertura de área única y múltiples áreas, y en los métodos de distribución de tareas en sistemas homogéneos y heterogéneos.

6.1.2 Experimentos de simulación

Las pruebas en simulación fueron realizadas con software de emulación del comportamiento físico de las aeronaves. Así, estos experimentos permitieron realizar vuelos en entornos controlados para probar los algoritmos de manera segura para las personas y el equipo sobre todo en las primeras etapas de desarrollo. Otra ventaja de este tipo de experimentos fue la posibilidad de plantear múltiples escenarios ficticios y ejecutar varias pruebas sin que esto represente un costo económico elevado. Así, se utilizaron diferentes opciones de software in the loop (SITL), que simula las operaciones que ejecutaría un controlador de vuelo como algoritmos de control, navegación, gestión de la misión. También, algunos simuladores emulan la física de vuelo que incluye el comportamiento de los sensores y motores. A continuación, se detallan las configuraciones de SITL utilizadas para las pruebas en simulación.

- **SITL de Ardupilot⁶**

Esta opción permite simular vehículos de ala fija, multirrotores, vehículos terrestres tipo rover y helicópteros configurados para funcionar con autopilotos de código abierto con el firmware de Ardupilot⁷ dentro del software de código abierto de Mission Planner (Como se observa en la Figura 6.1).

Una de las ventajas que ofrece es el acceso a herramientas de desarrollo en C++ y una comunidad de desarrolladores de código libre que continúan incluyendo nuevas características y mejorando constantemente este entorno. Esta opción se escogió para evaluar el comportamiento del algoritmo de planificación de trayectorias en espiral (ver capítulo 3), debido a que la lógica de navegación de los multirrotores en se ha configurado por defecto en este entorno para que la dirección de navegación sea de frente hacia los waypoints

- **SITL de PX4 y software de simulación de Gazebo**

De la misma manera que Ardupilot, el controlador de vuelo PX4⁸, tiene una herramienta de SITL que replica de manera realista las dinámicas de vuelo y las respuestas de los sensores, que se integra con herramientas de simulación como Gazebo, jMAVSim y X-Plane. Por su parte, Gazebo es un software de simulación avanzado y de código abierto ampliamente utilizado en el desarrollo y prueba de aeronaves. Ofrece un entorno tridimensional realista donde se pueden simular diversas condiciones ambientales y físicas, lo que permite evaluar el comportamiento de los UAVs en situaciones variadas sin necesidad de pruebas físicas.

³<https://numpy.org/>

⁴<https://matplotlib.org/>

⁵<https://scipy.org/>

⁶<https://ardupilot.org/dev/docs/sitl-simulator-software-in-the-loop.html>

⁷<https://ardupilot.org/>

⁸<https://px4.io/>



Figura 6.1: Opciones de SITL de ardupilot en Mission Planner

Gazebo se integra perfectamente con sistemas como ROS, ROS2 (Robot Operating System) y PX4, facilitando la simulación precisa de sensores, dinámicas de vuelo y controladores de misión. Esta capacidad de emulación detallada acelera el desarrollo y mejora la seguridad y eficiencia de las pruebas de UAVs, permitiendo a los desarrolladores de código identificar y corregir problemas antes de implementar sus sistemas en el mundo real.

Gracias a estas ventajas, se utilizó este software en las pruebas de simulación de los algoritmos planificación de trayectorias en múltiples áreas (ver capítulo 3), y en la evaluación de la técnica Powell-BINPAT para la distribución de tareas en sistemas multi-UAV homogéneos (ver capítulo 4).

- **Multiple Drone Simulator⁹**

Este simulador desarrollado por (Muran-Trus, 2024), se conecta directamente con el entorno de ROS y ROS2, y está pensado para trabajar de manera eficiente con múltiples UAVs (con pruebas exitosas de simulación de hasta 30 drones) sin consumir demasiados recursos computacionales. Utiliza el entorno de RViz como interfaz gráfica, y sus aproximaciones matemáticas y el modelamiento del sistema hace que el movimiento de los drones se aproxime al movimiento real, de manera que durante la simulación, el software dispone de nodos que publican constantemente la posición de los UAVs en instantes sucesivos de tiempo. El enfoque principal de este software es la simulación de movimiento, por lo cual, no se emula el comportamiento de sensores o cargas de pago. Gracias a su eficiencia computacional y configuración rápida de los parámetros de las aeronaves, este simulador se utilizó para poner a prueba el método CoP BINPAT

⁹https://github.com/anastmur/multiple_drones_sim

(ver capítulo 4) para evaluar la distribución de trayectorias en sistemas multi-UAV heterogéneos.

- **Flightmare**¹⁰

Este software de simulación presentado en la investigación de (Song et al., 2020), es un software de simulación con desarrollo 3D basado en Unity¹¹, que permite una simulación fotorealista de entornos de vuelo, con capacidades de desarrollo en código abierto y con tiempos de ejecución de la simulación relativamente rápidos, lo cual presenta una ventaja al evaluar múltiples aeronaves.

Este software simula las dinámicas de vuelo de los UAVs basado en modelos matemáticos que se ejecutan en programación paralela. Posee herramientas de integración con entornos de ROS y ROS2, y también tiene la capacidad de emular sensores adicionales de carga de pago como cámaras RGB o RGB-D ofreciendo incluso efectos adversos como distorsiones o difuminación.

Debido a estas características, este entorno se eligió para las pruebas en simulación del sistema de gestión dinámica de trayectorias ya que fue posible simular fallos en la adquisición de la imagen en aplicaciones de inspección de plantas fotovoltaicas.

6.1.3 Experimentos reales

Para validar la eficacia y la aplicabilidad de los algoritmos descritos en esta tesis en escenarios prácticos, se llevaron a cabo pruebas exhaustivas en entornos reales. Estas pruebas son fundamentales para evaluar el rendimiento de estos métodos en condiciones que emulan situaciones operativas del mundo real, permitiendo así identificar y superar posibles limitaciones, y asegurar que las soluciones propuestas no solo son teóricamente sólidas, sino también robustas y eficientes en la práctica. A continuación, se detallan los entornos de prueba seleccionados proporcionando una visión integral del proceso de validación. Los escenarios seleccionados se detallan a continuación:

- **Robotics CAR Arena**

Es un área de experimentación robótica que se encuentra en la Escuela Técnica Superior de Ingenieros Industriales (ETSII) como parte del Centro de Automática y Robótica y el Consejo Superior de Investigaciones Científicas (CSIC) (ver Figura 6.2). Este espacio aproximado de 7m de largo por 7m de ancho por 6m de altura, se utiliza como un escenario de pruebas y desarrollo y se encuentra cerca a los laboratorios por lo cual permite realizar correcciones sobre la marcha. Para compensar la ausencia de señal GNSS en interiores, se utiliza el sistema de captura de movimiento Optitrack¹² como método de localización para realizar vuelos de precisión en interiores, evitando errores de medición de distancias.

- **Aeródromo Villanueva del Pardillo**¹³

¹⁰<https://github.com/uzh-rpg/flightmare>

¹¹<https://unity.com/es>

¹²<https://optitrack.com/>

¹³<https://aerodromoyeventos.es/el-aerodromo/>

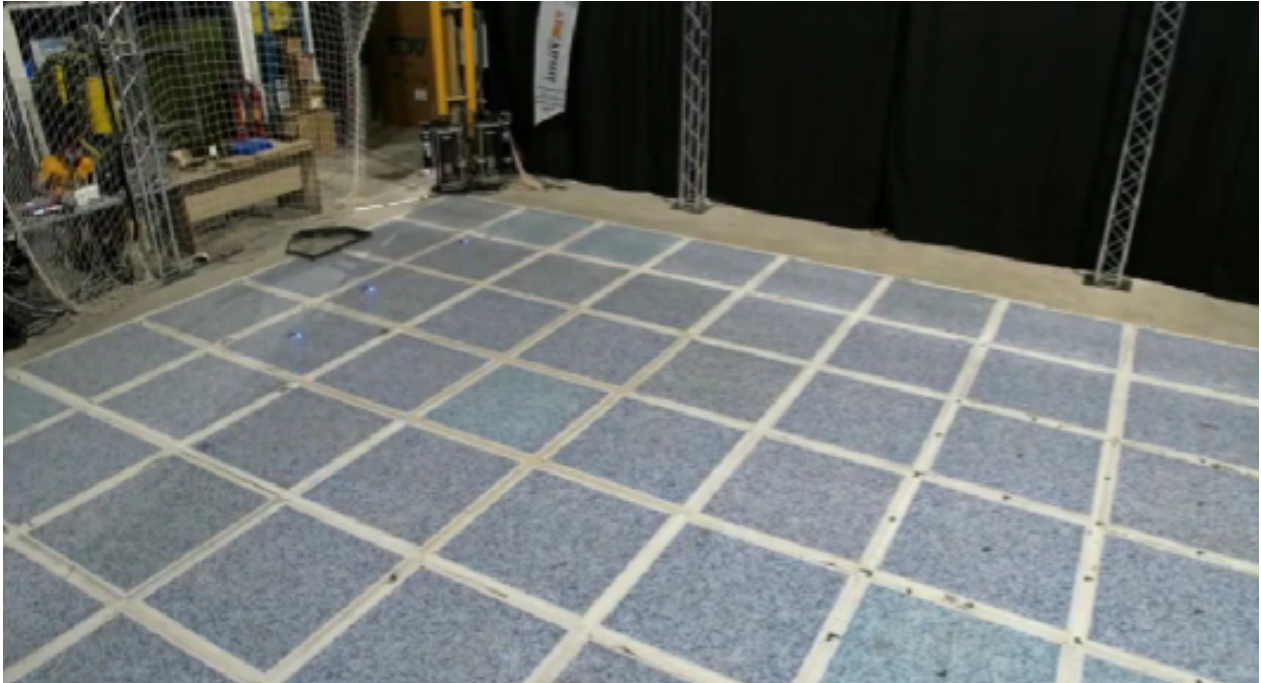


Figura 6.2: Imagen del Robotics CAR Arena con tres UAVs en vuelo

Es un aeródromo privado contratado por la empresa Drone-Hopper en el cual se pueden realizar pruebas con drones en un entorno privado y regulado de acuerdo a la legislación vigente en España. Se ubica en la cuenca del río Guadarrama y cuenta con un hangar de 300 m^2 y una pista de vuelo de 3000 m^2 . Cabe recalcar que aunque estas instalaciones facilitan las pruebas de vuelos reales, es necesario que la empresa que las coordina tenga al día toda la documentación relativa a las aeronaves y a los pilotos de apoyo.

- **Plataformas aéreas**

Se utilizaron dos tipos de plataformas aéreas para los experimentos en interiores y exteriores. Para vuelos en exteriores se utilizó el prototipo SwarmHopper de la empresa Drone-Hopper que se presenta en la Figura 6.3.

Este prototipo consta de una plataforma de hexarotor de amplia estabilidad equipada con un controlador de vuelo Pixhawk cube para ejecutar firmware de Ardupilot o PX4. Además, posee un sensor GNSS, un ordenador embarcado para procesamiento de las comunicaciones en malla, un sistema de comunicación Wifi, un receptor de radio y como carga de pago una cámara ZED2 para tomar datos durante el vuelo. Esta plataforma se desarrolló para el proyecto FASTER (ver Anexo A) con el fin de evaluar el comportamiento algoritmo Powell-BINPAT (capítulo 4) en entornos reales.

Para los vuelos en interior, se utilizaron las plataformas Bitcraze Crazyflie que son pequeñas y ligeras¹⁴ y permiten aprovechar el espacio disponible (dimensiones: $92\times 92\times 29\text{mm}$, peso: 27 gramos), como se muestra en la Figura 6.4.

Estas plataformas están equipadas con un controlador de vuelo y un sistema de co-

¹⁴<https://www.bitcraze.io/products/crazyflie-2-1/>



Figura 6.3: Figura tomada de (Luna et al., 2022) para mostrar el esquema de configuración del prototipo SwarmHopper utilizado en pruebas reales en el aeródromo

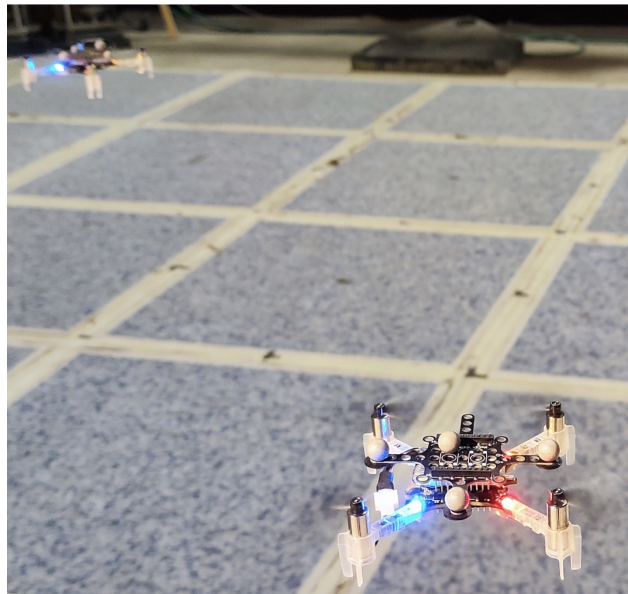


Figura 6.4: Plataformas aéreas utilizadas en experimentos reales dentro del Robotics CAR Arena

municaciones ligero. Para la captura de movimiento y estimación de la posición, cada UAV está equipado con 4 esferas reflectantes capturadas por el sistema Optitrack y el área de operación se ha limitado a $4 m^2$ para evitar colisiones.

6.2 Planificación de trayectorias de cobertura de área para sistemas de UAV único

Para este apartado se evaluarán las técnicas propuestas en sistemas de UAV único para compararlas con propuestas similares del estado del arte. Se realizaron experimentos computacionales evaluando dos métricas de rendimiento:

- Costo en distancia de la misión: En este tipo de experimentos se mide la distancia del plan de vuelo de manera que menores distancias representan planes más eficientes.
- Tiempo de CPU: Se analiza el tiempo de ejecución de los algoritmos en área única y múltiples áreas para determinar la eficiencia computacional de cada uno.

6.2.1 Evaluación de rendimiento de los algoritmos

Para evaluar el rendimiento del algoritmo de espiral en área única, se realizó una prueba con 10 áreas aleatorias de diferentes tamaños. Los resultados se tabularon en la Tabla 6.1 y se presentan gráficamente en la Figura 6.5.

Tabla 6.1: Rendimiento del algoritmo espiral para un único UAV y una sola área en diferentes pruebas

Nro. Test	Spiral	
	Costo(m)	T. CPU (ms)
1	521.80	10.22
2	777.07	11.32
3	846.93	13.51
4	1135.34	14.02
5	1224.70	13.57
6	1252.21	15.22
7	1514.39	17.54
8	1999.85	18.32
9	2007.74	20.04
10	2704.88	22.02

En la Tabla 6.1 y en la Figura 6.5, se puede observar que el algoritmo de generación de rutas en espiral genera rutas que cubren toda el área de manera eficiente y con un bajo costo computacional. En la Tabla 6.2, se presenta una comparación de nuestro método con propuestas similares en la literatura.

De acuerdo a los resultados de la Tabla 6.2, comparado con otras técnicas similares presentadas en la literatura (Cabreira et al., 2018; Balampanis et al., 2017) nuestro enfoque considera



Figura 6.5: Trayectorias de planificación espiral para UAV único

Tabla 6.2: Comparativa de métodos de planificación en espiral presentados en la literatura

Método	Técnica	Suavizado de la ruta	Tipo de área	Tiempo de CPU	Sentido óptimo de giro
Espiral Afín (nuestro)	Matriz de transformación afín y clotoides	Si	Polígonos regulares convexos	10 a 22 ms	Si
Cabreira et al., 2018	Cálculo de ángulos de barrido y líneas paralelas	No	Polígonos regulares convexos	No se reporta	No
Balampanis et al., 2017	Triangulación de Delaunay y suavizado de Lloyd	Si	Polígonos irregulares	200 a 1000 ms	No

un nuevo método de generación de la ruta basado en matrices de transformación que lo hace computacionalmente eficiente. Además, las técnicas de suavizado de la ruta permiten que los UAVs recorran la ruta más rápido sin cambios bruscos en el sentido de giro. Por otro lado, también utilizamos un enfoque de determinación del sentido óptimo de giro que no está presente en los trabajos previamente mencionados. No se encontraron repositorios ni código libre de estos métodos para realizar más experimentos. Además, como se menciona en el capítulo 3, esta técnica está enfocada para aeronaves con navegación en dirección hacia los waypoints (como se observa en las pruebas de simulación).

A continuación, para evaluar el rendimiento de las propuestas de exploración de múltiples áreas basada en el problema del viajante (TSP) y vecinos cercanos (NN) se estandarizarán los parámetros de la misión de manera que no se vean afectadas las métricas de costo y tiempo de CPU. Para esto se ajustan los parámetros de la Tabla 6.3.

Tabla 6.3: Parámetros de la misión para evaluación de rendimiento de los algoritmos de planificación en múltiples áreas

Parámetro	Valor
Área de cada región	2500 m ²
Altura de la misión	20 m
Espacio entre calles	10 m
Separación de waypoints	10 m
Técnica de planificación	Back-and-forth

En la Tabla 6.3, se utilizará la técnica back-and-forth propuesta por (Huang, 2001) como método de planificación debido sus ventajas en costo y eficiencia computacional cuando se aplica en aeronaves con navegación en los cuatro grados de libertad. También se evaluará el rendimiento entre 1 y 10 áreas considerando el enfoque práctico de esta tesis, ya que un mayor número de áreas disjuntas no tendría mayor aplicación en tareas del sector industrial (Luna et al., 2023). Los resultados de comparación de costo de la misión se muestran en la Figura 6.6, mientras que los resultados de tiempo de CPU se presentan en la Figura 6.7

En la Figura 6.6, se puede apreciar que el costo de la misión es similar utilizando los dos métodos. El método TSP busca optimizar la solución de manera global, mientras que NN busca una serie de soluciones optimizadas a nivel local de manera iterativa; por esta razón, la diferencia de costos por el cambio en el orden de visita se aprecia mejor cuando incrementa el número de áreas.

Adicionalmente, en la Figura 6.7, se observan cambios significativos en el tiempo de ejecución a partir de las siete áreas. Estos resultados muestran que aunque la técnica NN presenta un mayor costo de la misión es computacionalmente más eficiente que TSP. Además, su implementación es más simple debido al no depender de librerías externas. La solución NN se evaluará en con mayor profundidad en los próximos apartados de este capítulo.

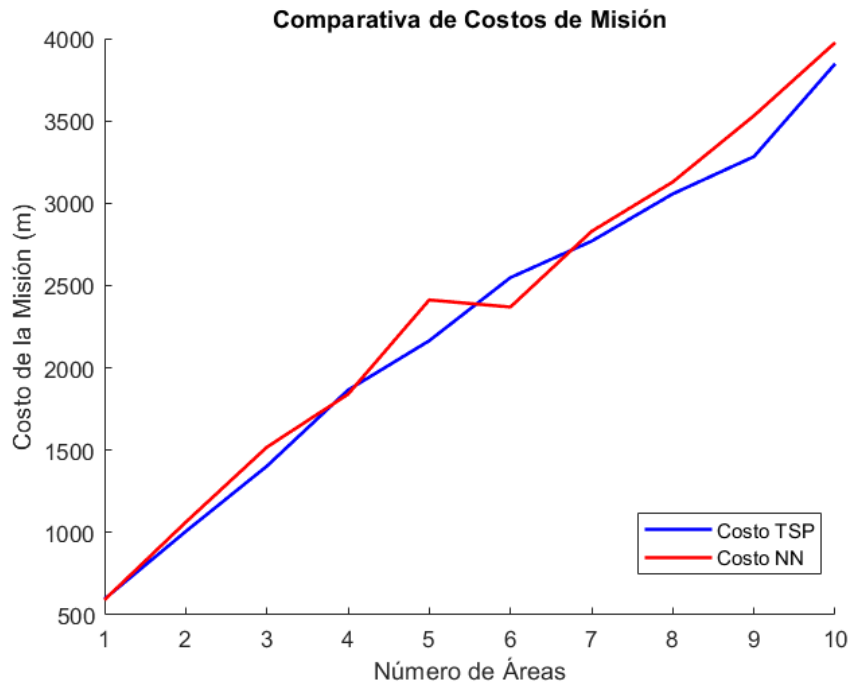


Figura 6.6: Comparación de costo de la misión en distancias del método de vecino más cercano (NN) y problema del viajante (TSP)

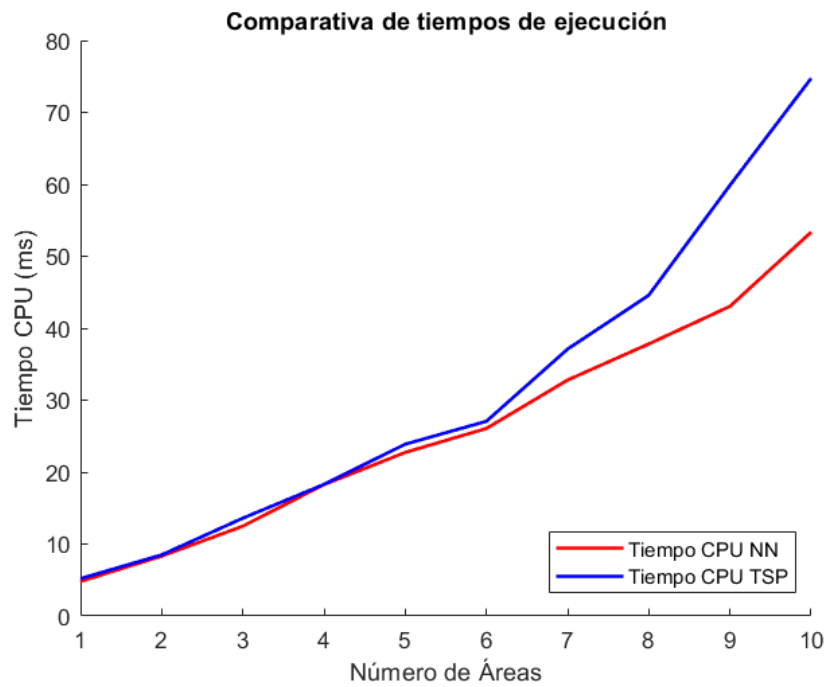


Figura 6.7: Comparación de tiempos de ejecución del método de vecino más cercano (NN) y problema del viajante (TSP)

6.2.2 Resultados en simulación

En esta sección se evaluará únicamente la diferencia del rendimiento de las técnicas de planificación de área única, simulando las características de navegación con rumbo hacia adelante para comprobar el rendimiento del método de planificación espiral. Para esto se utilizó el emulador SITL del entorno Mission Planner, seleccionando la misma región de exploración para back-and-forth y espiral como se muestra en la Figura 6.8, y los resultados para tres pruebas se muestran en la Tabla 6.4.



Figura 6.8: Configuración del entorno SITL para Mission Planner. a) Planificación back-and-forth. b) Planificación espiral

Tabla 6.4: Resultados de simulación de vuelo para planificación back-and-forth y espiral

Prueba	Tiempo de ejecución de la misión (s)	
	BF	Espiral
1	415	352
2	305	194
3	563	432

En la Tabla 6.4 se puede que el tiempo de ejecución de la misión para el patrón en espiral es menor que el back-and-forth. Esto se debe a que la configuración de vuelo de las aeronaves que en este caso se ha ajustado para navegar hacia los waypoints con rumbo hacia el frente, en otras configuraciones en las cuales las aeronaves pueden volar hacia atrás, el patrón de back-and-forth sigue siendo más eficiente ya que no necesita radios de giro para completar la trayectoria (esto se comprobará en las secciones posteriores de este capítulo).

6.2.3 Discusión

Como se muestra en la Tabla 6.2, el algoritmo de generación de trayectorias en espiral propuesto en esta investigación es más eficiente computacionalmente al compararlo con otras propuestas en la literatura, sus características adicionales como el suavizado de rutas y la determinación del sentido de giro reducen el costo de la misión. Sin embargo, como se puede comprobar en la tabla 6.4, solo es más eficiente que back-and-forth en vuelos con aeronaves que vuelan de frente hacia los waypoints, lo cual limita su uso en operaciones industriales.

Por otra parte, se han generado soluciones de generación de trayectorias para cobertura de múltiples áreas independientes del tipo de ruta de área única que se utilice (espiral o back-and-forth). De las dos propuestas presentadas, las Figuras 6.6 y 6.7 demuestran que el enfoque de vecino más cercano es más rápido en tiempos de CPU comparado con el método TSP, además su implementación no requiere realizar operaciones de optimización global, en cuanto a costos de distancia, TSP es más eficiente, aunque no es muy superior.

6.3 Algoritmo Powell-BINPAT

6.3.1 Diseño del experimento y métricas de evaluación

Para evaluar los resultados del método Powell-BINPAT se ejecutan pruebas de evaluación de rendimiento, pruebas en simulación y pruebas en entornos reales.

Además, para medir la eficiencia en la distribución de la misión, se mide el coeficiente de variación del costo individual de cada UAV, dado por la ecuación 6.1.

$$CV = \frac{SD}{Av} \times 100 \quad (6.1)$$

Donde SD es la desviación estándar, y Av es el promedio de los datos de costo tabulados.

En las pruebas de evaluación de rendimiento se comprobará el funcionamiento del algoritmo con las técnicas de generación de trayectorias descritas en el capítulo 3. También se analizará el costo de la misión calculado como el máximo costo reportado por un UAVs. Para la exploración multi-área se ha elegido el algoritmo de vecino más cercano (Ver capítulo 3, debido a que es más eficiente computacionalmente y en costo de la misión en aplicaciones de entre 2-10 áreas.

6.3.2 Evaluación de rendimiento del método propuesto

Los experimentos de esta sección se utilizarán para evaluar el rendimiento teórico del algoritmo. Para obtener resultados independientes de las condiciones de la misión, estas se han estandarizado de acuerdo a la Tabla 6.5.

Tabla 6.5: Parámetros estandarizados de cada misión para evaluar el rendimiento del algoritmo

Parámetro	Valor
Área de cada región	2500 m^2
Altura de la misión	20 m
Ancho de calles	10 m
Separación entre waypoints	10 m

En primer lugar, se realizan pruebas para múltiples UAVs y tres regiones de exploración utilizando las técnicas de espiral y back-and-forth. Los resultados de costo máximo de distancia y coeficiente de variación se presentan en la Tabla 6.6. Todos los datos se han tabulado en el Anexo B. Además, se presentan resultados gráficos en las Figuras 6.9 y 6.10.

Tabla 6.6: Rendimiento del algoritmo Powell-BINPAT utilizando la planificación de trayectorias espiral y back-and-forth para 2-10 drones

UAVs	Espiral		B&F	
	Costo	CV	Costo	CV
2	770.79	0.16	703.17	0.14
3	591.34	0.46	584.22	8.81
4	495.79	2.37	447.34	1.26
5	434.16	1.66	440.61	1.86
6	445.57	3.75	444.30	5.41
7	427.00	4.01	428.03	16.14
8	441.06	3.18	399.53	2.98
9	412.30	10.20	405.12	9.60
10	430.92	14.10	401.70	7.50

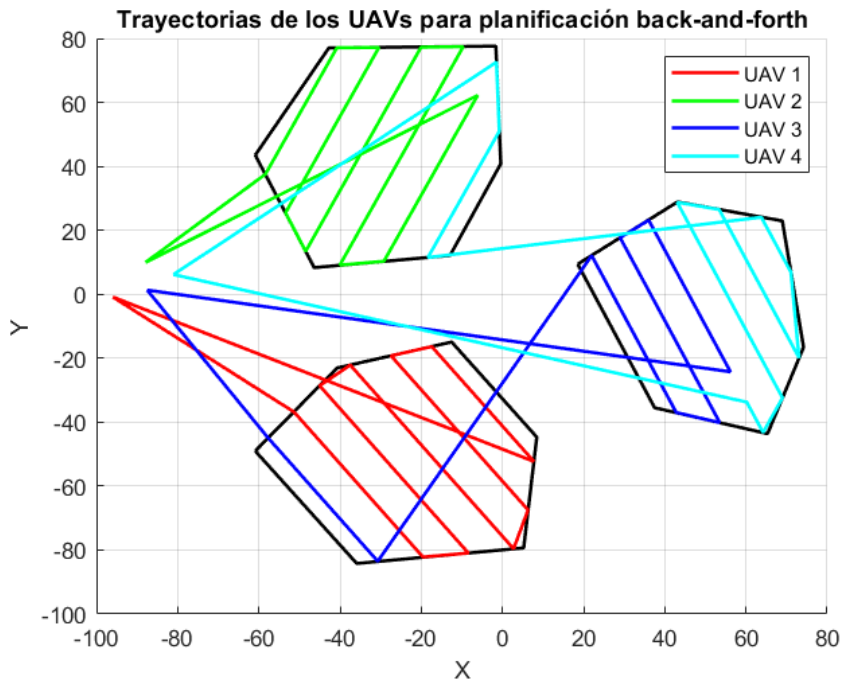


Figura 6.9: Resultados gráficos del algoritmo Powell-BINPAT con planificación de trayectorias back-and-forth

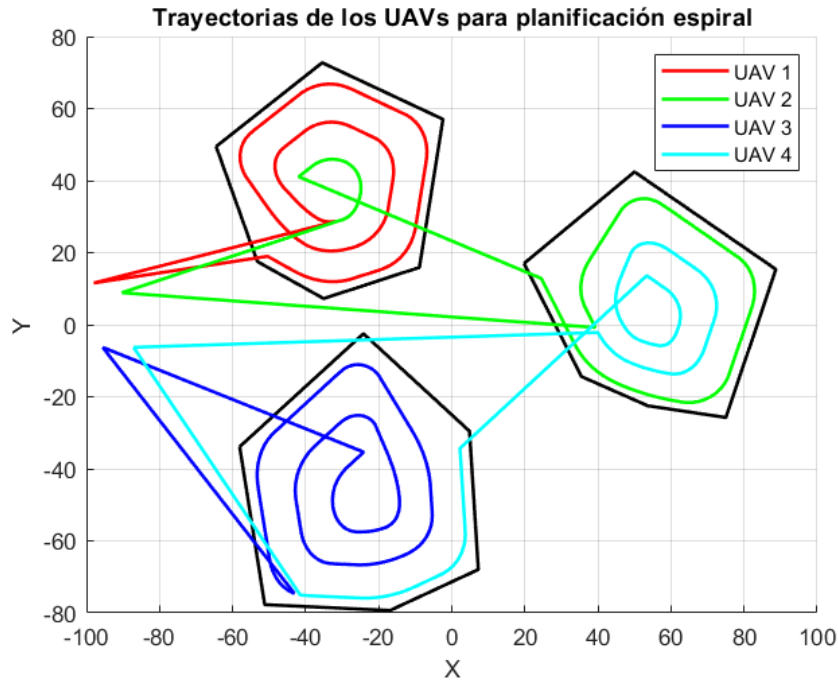


Figura 6.10: Resultados gráficos del algoritmo Powell-BINPAT con planificación de trayectorias espiral

En la Tabla 6.6, se muestran los resultados de costo de la misión. El algoritmo de planificación en back-and-forth consigue mejores resultados en costo ya que como se explica en el capítulo 3, está desarrollado considerando que el UAV se mueve incluso hacia atrás y no se considera el costo de giro en las curvas. También se puede apreciar que los resultados del coeficiente de variación son relativamente bajos cuando se comparan las dos técnicas, lo cual indica que el algoritmo Powell-BINPAT distribuye las misiones equitativamente.

Una comparación gráfica del algoritmo Powell-BINPAT con las propuestas presentadas por los autores en (Avellar et al., 2015) y (Hong et al., 2021) se presenta en la Figura 6.11.

En la Figura 6.11 se puede observar que Powell-BINPAT distribuye de manera más eficiente la misión que las propuestas presentadas en la literatura, ya que toma puntos intermedios en caso de ser necesario. Los resultados tabulados al comparar los tres métodos se presentan en la Tabla 6.7.

Tabla 6.7: Resultados de comparación del CV en diferentes métodos presentados en la literatura

Método	CV máximo reportado	Configuración
Powell-BINPAT (nuestro)	10.2	9 UAVs y 3 regiones de exploración
Hong et al., 2021	18.1	3 UAVs y 1 región de exploración
Avellar et al., 2015	13.96	3 UAVs y 1 región de exploración

En la Tabla 6.7, si calculamos el coeficiente de variación en las pruebas presentadas por los autores en (Hong et al., 2021) para tres UAVs y una sola área obtenemos un valor de

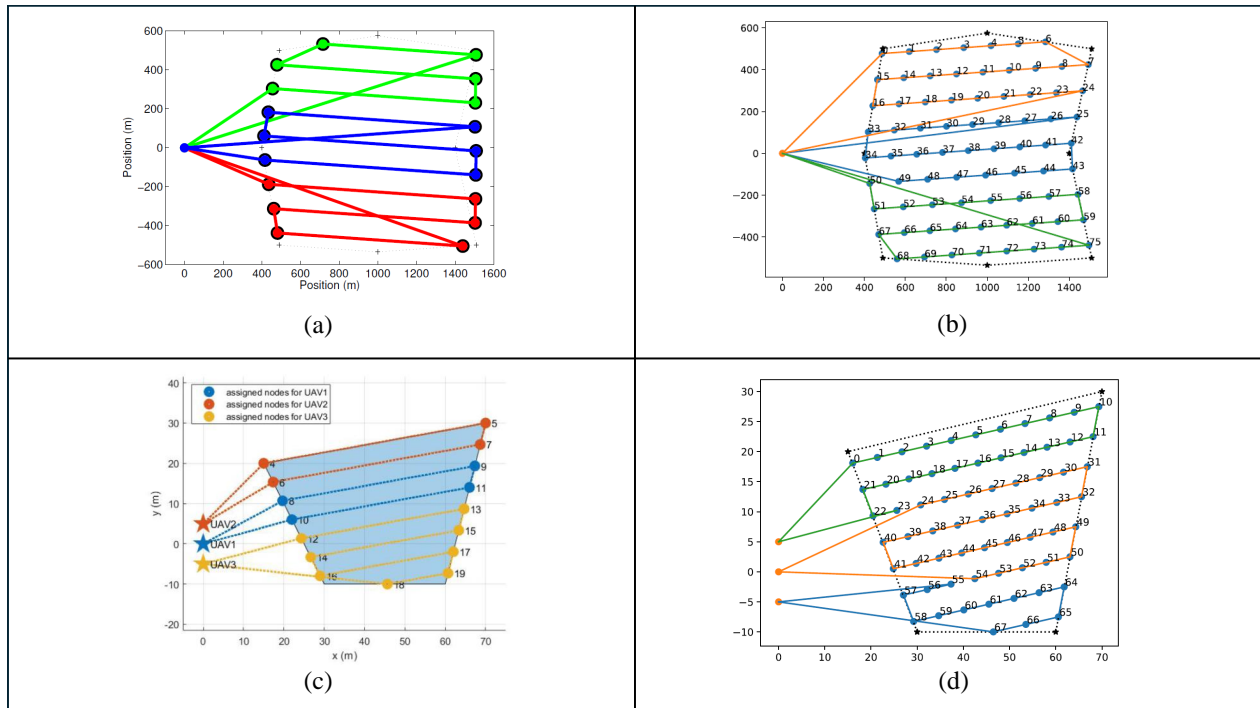


Figura 6.11: Resultados gráficos de comparación del algoritmo Powell-BINPAT con propuestas de la literatura. a) Propuesta presentada por (Avellar et al., 2015). b) Resultado Powell-BINPAT. c) Propuesta presentada por (Hong et al., 2021). d) Resultado Powell-BINPAT

18.01 y en la propuesta presentada en (Avellar et al., 2015) calculamos un valor de 13.96 bajo las mismas condiciones. En comparación con los resultados reportados, se observa que Powell-BINPAT distribuye de manera más eficiente la misión entre cada UAVs.

A continuación, se evaluará el tiempo de CPU del algoritmo utilizando como base los parámetros definidos en la Tabla 6.5. Así se estableció el número de UAVs entre 2 y 5 y el número de regiones a explorar entre 1 y 10. Los resultados se presentan en la Figura 6.12.

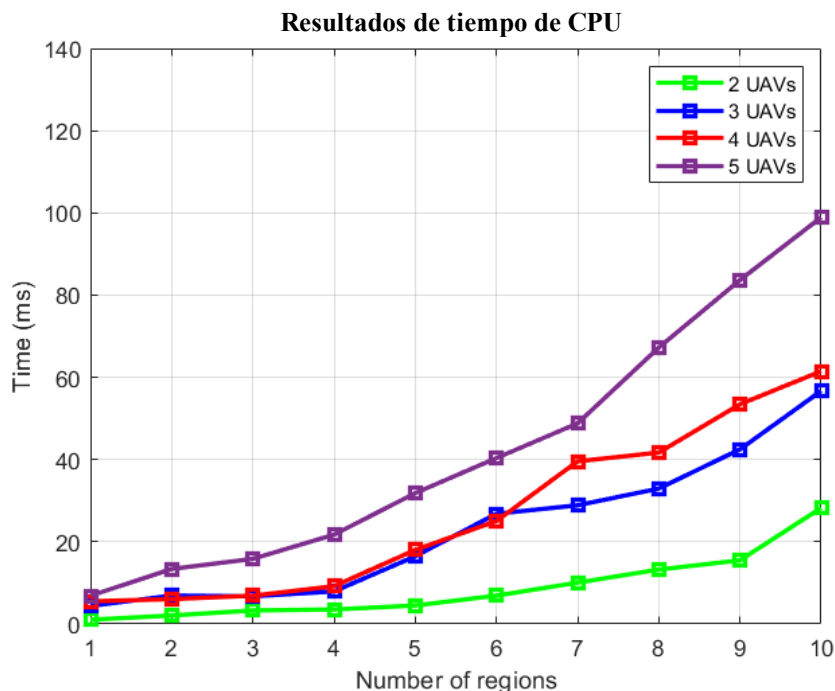


Figura 6.12: Resultados de tiempo de CPU del algoritmo Powell-BINPAT para múltiples UAVs y múltiples regiones

Una comparativa con las propuestas del estado del arte se presenta en la Tabla 6.8.

En comparación de propuestas de área única, podemos observar que los resultados presentados por (Hong et al., 2021) reportan tiempos de CPU de 0.8 segundos para tres UAVs y una región de 19 waypoints. En sistemas de múltiples áreas, en la investigación presentada por (Xie y Chen, 2022) los autores usan MATLAB en un MacBook Pro con un procesador Intel Core i7 de 6 núcleos a 2.6GHz y 16GB de memoria. Aplicando algoritmos genéticos obtuvieron tiempos de CPU entre 183.3 ms para una distancia de recorrido de 1275.1 m y 265.3 ms para una distancia de recorrido de 2110.8 m. Han reportado de uso de entre 1 a 6 UAVs, pero no se especifica el número exacto de regiones, aunque los autores afirman que el número de áreas para estos experimentos fue menor de 10. Por otro lado, los autores en (Vasquez-Gomez et al., 2018) realizan sus experimentos con Matlab en una máquina Intel Core i7 con 2GB de memoria, obteniendo 84 ms para 5 áreas de exploración y 89 ms para 10 regiones usando un UAV único. No se ha podido obtener código fuente de los métodos mencionados pero en comparación con otras propuestas, Powell-BINPAT obtiene tiempos de CPU más bajos para ordenadores con características de procesamiento similares, obteniendo

Tabla 6.8: Resultados de comparación del tiempo de CPU en diferentes métodos presentados en la literatura para el algoritmo Powell-BINPAT

Método	T. CPU (ms)	Configuración de la misión	Configuración del ordenador
Powell BINPAT	98.92	10 regiones y 5 UAVs	CPU Intel Core i7-7700HQ, 3.8 GHz, y 16 Gb de RAM
Hong et al., 2021	800	1 región y 3 UAVs	CPU Intel i7, 2.6 GHz y 16 Gb RAM
Xie y Chen, 2022	183.3	Menos de 10 regiones y 1-6 UAVs	Intel Core i7 de 6 núcleos, 2.6GHz, 16 Gb de RAM
Vasquez-Gomez et al., 2018	89	1 UAV y 10 regiones	Intel Core i7 con 2 GB de RAM

un máximo de 98.92 ms para 10 regiones y 5 UAVs.

6.3.3 Pruebas en simulación

Las pruebas de simulación para el algoritmo de Powell-BINPAT se realizaron con dos escenarios diferentes. En el primer escenario, se evaluó el rendimiento en cobertura de área única con 3 UAVs y en el segundo escenario se simularon 4 áreas con 3 UAVs. Para esto, se utilizó el simulador de código abierto Gazebo con SITL de PX4. Por otro lado, para monitorizar la misión y capturar los datos de rendimiento se utilizó el software de estación de control de tierra QGroundControl¹⁵. Los resultados de estas pruebas se presentan en la Tabla 6.9 y una captura de pantalla de la ejecución en QGroundControl se muestra en la Figura 6.13.

Tabla 6.9: Resultados de tiempo de ejecución de la misión en segundos para simulación SITL con el algoritmo Powell-BINPAT

Áreas	1 área		4 áreas		
	Planificación	BF (s)	Espiral (s)	BF (s)	Espiral (s)
UAV1		243	262	715	758
UAV2		258	284	719	746
UAV3		245	280	717	751
Costo		258	284	719	758
CV		3.32	4.19	0.23	0.65

Los resultados del entorno de simulación al medir el tiempo de ejecución de las misiones muestran un comportamiento similar al obtenido de manera teórica. Por lo tanto, se puede

¹⁵<http://qgroundcontrol.com/>

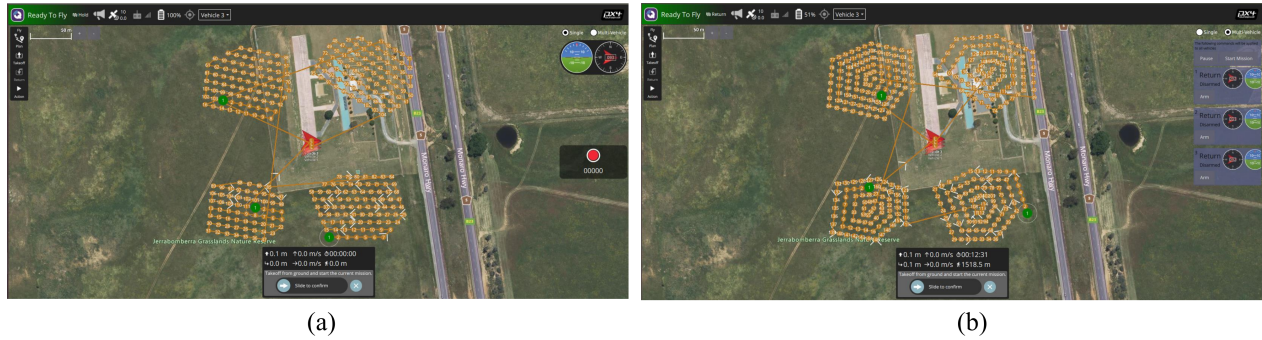


Figura 6.13: Capturas de pantalla de la estación de control de tierra. (a) Planificación back-and-forth. (b) Planificación espiral

observar que el método Powell-BINPAT mantiene la distribución eficiente de la misión incluso independientemente del método escogido de planificación (espiral o back-and-forth). Resultados adicionales de simulación obtenidos en (Luna et al., 2022) se tabularon en el Anexo B.

6.3.4 Pruebas en entornos reales

El método Powell-BINPAT se implementó en entornos reales para el proyecto FASTER financiado por la unión europea y para el proyecto Copilot-CM de la comunidad de Madrid (Ver Anexo A). Los vuelos de prueba se realizaron en el aeródromo de Villanueva de Pardillo como se muestra en la Figura 6.14.



Figura 6.14: Pruebas del algoritmo Powell-BINPAT en entornos reales

Los resultados de las trayectorias de tres UAVs se obtuvieron del archivo datalog del vuelo y se presentan en la Figura 6.15.

Finalmente, los resultados tabulados de tiempo de vuelo de cada UAV para las pruebas reales se presentan en la Tabla 6.10.

En la Tabla 6.10, se comprueba el mismo comportamiento de los coeficientes de variación obtenidos en simulación, ya que los tiempos están distribuidos equitativamente.

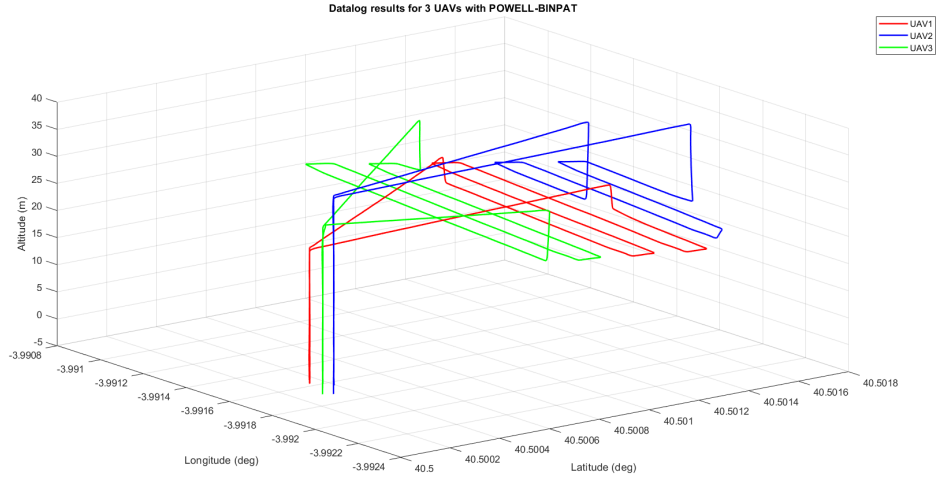


Figura 6.15: Datos de trayectorias del algoritmo Powell-BINPAT obtenidos de datalog de vuelo

Tabla 6.10: Resultados de tiempo de ejecución de la misión en segundos para prueba en entornos reales

UAV	Tiempo (s)
UAV1	243
UAV2	258
UAV3	245
Costo	258
CV	3.32

6.3.5 Discusión

El método propuesto para distribución de trayectorias en sistemas homogéneos denominado Powell-BINPAT utiliza un enfoque heurístico basado en el problema de bin-packing junto con la técnica de optimización de Powell. Este algoritmo demuestra dos características fundamentales al compararlo con algoritmos similares en la literatura. En la Tabla 6.7, se puede observar que el método es más eficiente en la distribución de misiones al compararlo con propuestas similares en la literatura, lo cual se ha comprobado midiendo el coeficiente de variación tanto en la distancia teórica como en tiempos de vuelo en entornos reales y simulados. Por otro lado, como se observa en la Tabla 6.8, al medir los tiempos de CPU del algoritmo en entornos de área única y entornos de múltiples áreas, se comprueba que es eficiente computacionalmente en comparación con propuestas similares presentadas en el estado del arte, incluso para configuraciones de 10 UAVs y 5 regiones de exploración.

6.4 Algoritmo CoP-BINPAT

6.4.1 Diseño del experimento y métricas de evaluación

Para evaluar los resultados del algoritmo CoP-BINPAT, se realizaron dos tipos de experimentos: experimentos de evaluación de rendimiento de los planes generados y pruebas en un entorno simulado. En ambos, se determina el balance de la distribución de la misión mediante la métrica del coeficiente de variación de los resultados de tiempo de ejecución de la misión dado por la ecuación 6.1.

Para los experimentos de rendimiento, se mide el coeficiente de variación del tiempo estimado de la misión (ecuación 6.2) ya que a diferencia del caso homogéneo, la medida del costo de distancia no determina el balance final, ya que las velocidades crucero de cada UAV son diferentes.

$$t_i = \frac{D_i}{v_i} \quad (6.2)$$

En la ecuación 6.2, el tiempo estimado de la misión viene dado por el costo de la distancia D_i dividido entre la velocidad crucero de cada UAV v_i asumiendo una velocidad constante. Además, esta métrica no solo se utiliza para determinar la distribución de la misión, también se comprueba que la misión asignada a cada UAV respete las restricciones de tiempo de vuelo restante.

6.4.2 Análisis de rendimiento del método propuesto

Estos experimentos tienen como objetivo evaluar el tiempo de ejecución y determinar de manera teórica el rendimiento del método CoP-BINPAT. En primer lugar, se mide el tiempo de CPU, para lo cual se prueba con múltiples UAVs explorando una región. De hecho, estos tiempos pueden variar dependiendo del tamaño del área y las variaciones en los tamaños de barrido. Si el tamaño del área es mayor, el tiempo de CPU aumentará. Por otro lado, con un mayor número de variaciones en los anchos de barrido, el algoritmo de descomposición

poligonal tardará más en generar las subregiones. Por esta razón, se ejecutaron las pruebas utilizando los parámetros de la Tabla 6.11

Tabla 6.11: Parámetros de la misión para experimentos de medición de tiempo de CPU

Parámetro	Valor
Area de la región	10000 m^2
Número de UAVs	3-10
Altura de la misión	20 m
Velocidades	(10, 20, 30, 40, 50) m/s
Anchos de barrido	(5, 10, 15, 20) m
Patrón de cobertura	Back-and-forth, espiral

En la Tabla 6.11, se ha definido un área de 10000 m^2 para ser explorada por 3 a 10 UAVs. Las velocidades, tiempos de suministro y anchos de barrido se ajustarán de manera aleatoria, pudiendo tomar únicamente los valores que se muestran en la misma tabla. En las pruebas iniciales, se asume que todos los UAVs tienen suficiente batería para ejecutar la misión y las planificaciones se realizan utilizando los métodos de cobertura de área back-and-forth y espiral. Las pruebas de tiempo de CPU se presentan gráficamente en la Figura 6.16.

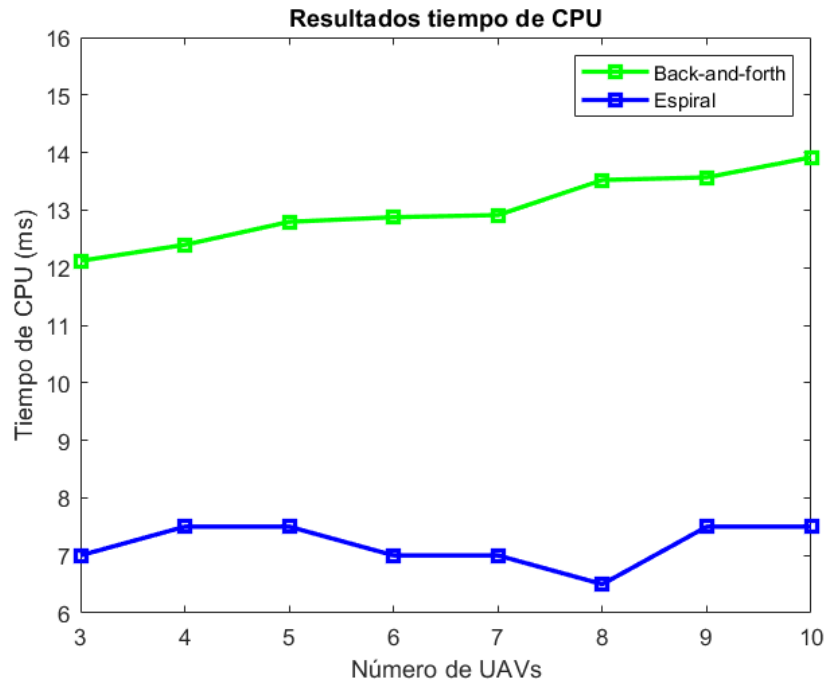


Figura 6.16: Tiempos de CPU del método CoP BINPAT, evaluados con diferentes números de UAVs y con los métodos de cobertura back-and-forth y espiral

En la Figura 6.16, se muestra que el patrón de cobertura en espiral reporta tiempos de ejecución más rápidos (menores a 8 ms). Sin embargo, el patrón de back-and-forth también mantiene tiempos de CPU bajos (menores a 15 ms). Una comparativa de los tiempos de CPU obtenidos con los métodos del estado del arte se presenta en la Tabla 6.12.

Tabla 6.12: Resultados de comparación del tiempo de CPU en diferentes métodos presentados en la literatura para el algoritmo CoP-BINPAT

Método	T. CPU (ms)	Configuración de la misión	Configuración del ordenador
CoP BINPAT	15	Una región, 5 UAVs	CPU Intel Core i7-7700HQ, 3.8 GHz, y 16 Gb de RAM
J. Chen, Du et al., 2021	20	Múltiples regiones, 3-9 UAVs	No se reporta
Xiao et al., 2023	850-1400	Múltiples regiones, 3-8 UAVs	No se reporta
Deng et al., 2019	280-2600	Una región, 1-7 UAVs	Intel Core i5 de 2.7 GHz

En la Tabla 6.12, el enfoque presentado en (J. Chen, Du et al., 2021), reporta tiempos de ejecución cercanos a 20 *ms* al variar el número de UAVs entre 3 y 9. Sin embargo, este método utiliza múltiples áreas de exploración. Similar a este método, los autores en (Xiao et al., 2023), reportan un método de asignación de múltiples regiones a UAVs heterogéneos, obteniendo tiempos de CPU de entre 0.85 y 1.4 s utilizando entre 3 y 8 UAVs. En ninguna de estas investigaciones se describen las características del ordenador utilizado, y no existe el código fuente para realizar más comparaciones. Cabe recalcar que la diferencia de estas técnicas con CoP-BINPAT es que no realizan una división ponderada de las áreas, sino que se asigna el área completa a los diferentes UAVs. En el trabajo presentado por (Deng et al., 2019) los autores utilizan un ordenador Intel Core i5 de 2.7 GHz obteniendo tiempos de ejecución de entre 280 y 2600 ms para pruebas entre 1 y 7 UAVs. En esta investigación se proponen algoritmos de división de área pero la heterogeneidad del sistema viene dada por la mezcla de UAVs y UGVs.

Además, se analizan los resultados de coeficiente de variación calculando los tiempos estimados de ejecución de la misión para los patrones de navegación back-and-forth y espiral. En la Tabla 6.13 se muestran los resultados del experimento para 5 UAVs. Adicionalmente, las rutas generadas se muestran en la Figura 6.17. A continuación, los resultados de rendimiento utilizando el patrón de espiral se muestran en la Tabla 6.14 y las rutas generadas se muestran en la Figura 6.18.

Como se puede observar en las Tablas 6.13 y 6.14, el método de planificación back-and-forth reporta mejores resultados de *CV* cuando se compara con el método espiral. Para comprobarlo se han realizado múltiples experimentos utilizando entre 3 y diez UAVs tabulados en los Anexos C.1 y C.2, y gráficamente en la Figura 6.19.

En la Figura 6.19, el patrón back-and-forth obtiene mejores resultados que el espiral en términos generales. El motivo principal de esta diferencia es que el método de CoP-BINPAT realiza sus cálculos basados en la partición del área previa a la asignación de la misión. De esta manera, las rutas generadas por el algoritmo back-and-forth son más uniformes que aquellas

Tabla 6.13: Rendimiento del algoritmo CoP-BINPAT con planificación back-and-forth y 5 UAVs

UAV	Barrido (m)	Vel (m/s)	Bat (s)	CoP	Dist (m)	Estimación t(s)
UAV 0	5	30	1600	0.14	1816.37	60.55
UAV 1	20	10	1600	0.18	861.35	86.14
UAV 2	5	50	1600	0.36	2996.59	59.93
UAV 3	20	20	1600	0.39	1698.84	84.94
UAV 4	10	10	1600	0.09	852.33	85.23

Prom	75.36
Desv. St	12.35
CV	16.39

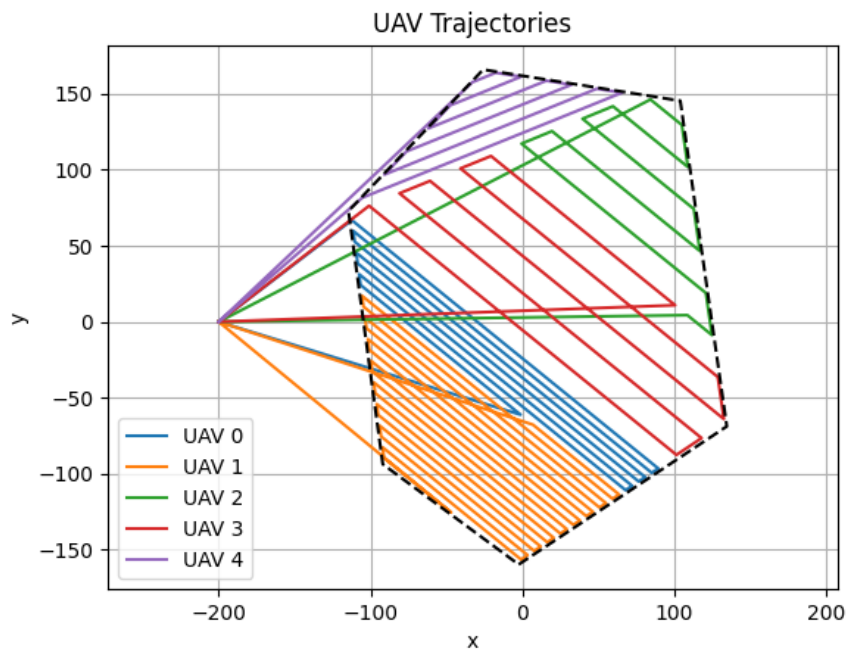


Figura 6.17: Trayectorias generadas por el algoritmo CoP-BINPAT para 5 UAVs con el patrón back-and-forth

Tabla 6.14: Rendimiento del algoritmo CoP-BINPAT con planificación espiral y 5 UAVs

UAV	Barrido (m)	Vel (m/s)	Bat (s)	CoP	Dist (m)	Estimación t(s)
UAV 0	5	40	1600	0.1	1110.61	27.77
UAV 1	15	20	1600	0.15	864.28	43.21
UAV 2	15	10	1600	0.08	765.22	76.52
UAV 3	20	40	1600	0.41	1459	36.48
UAV 4	10	50	1600	0.26	1570.38	31.41

Prom	43.08
Desv. St	17.51
CV	40.65

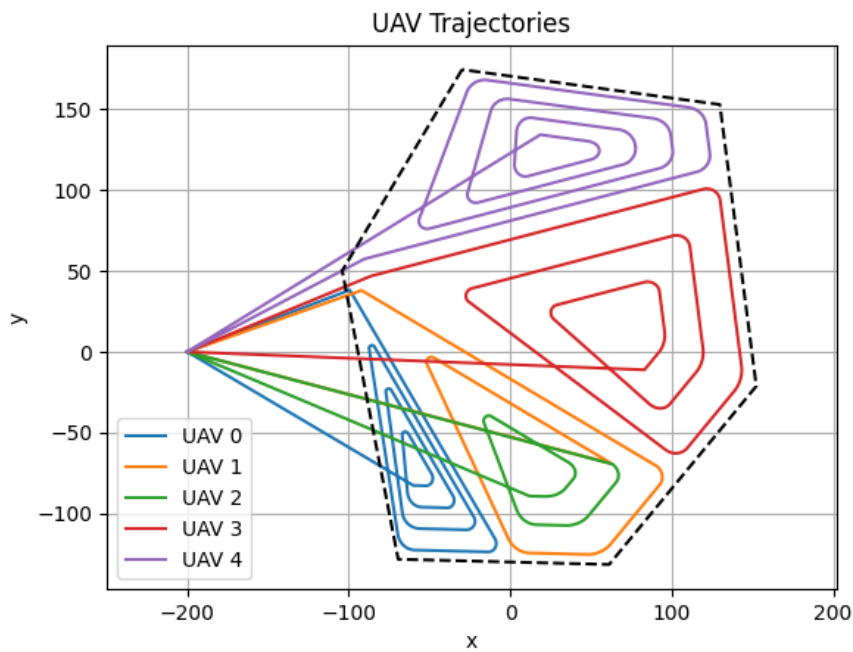


Figura 6.18: Trayectorias generadas por el algoritmo CoP-BINPAT para 5 UAVs con el patrón espiral

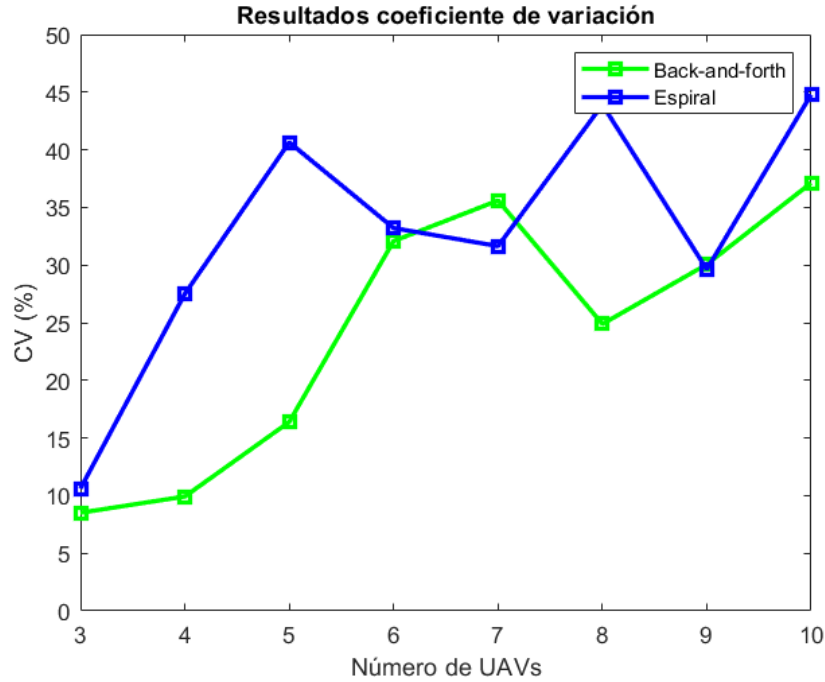


Figura 6.19: Resultados de coeficiente de variación para múltiples UAVs utilizando el algoritmo CoP-BINPAT

generadas con el método de la espiral. Por otra parte, se comprueba que los valores de CV son mucho mayores que otros que el método Powell-BINPAT debido a que el algoritmo no considera los costos de ida y vuelta de la misión para calcular la mejor distribución.

Finalmente, se comprueba que el algoritmo CoP-BINPAT cumple con las restricciones de máximo tiempo de vuelo. Para esto se han ajustado valores críticos de tiempo de batería restante y se calculó el tiempo estimado que toma a cada UAV ejecutar la misión. Después se obtuvieron los resultados del costo en tiempo de la misión sin tomar en cuenta el costo de ida y vuelta. Finalmente, se obtiene el tiempo disponible de cada UAV restando el valor consumido en la misión menos el valor inicial (Ecuación 6.3).

$$Fltime_f = Fltime_i - Fltime_m \quad (6.3)$$

Donde $Fltime_f$ es el tiempo de vuelo disponible teórico tras la misión, $Fltime_i$ es el tiempo restante inicial y $Fltime_m$ es el tiempo consumido en la misión. Las pruebas para 2-5 UAVs se han tabulado en las tablas del Anexo C.3, y un resumen se muestra en la Tabla 6.15.

Las pruebas de la Tabla 6.15 muestran el rendimiento del algoritmo con respecto al tiempo de vuelo restante. Se han tabulado el tiempo total disponible, el tiempo disponible promedio, el mínimo y el máximo. En todos los casos se observa que el algoritmo cumple con las restricciones de tiempo máximo y es independiente del número de UAVs. Sin embargo, la prueba con 3 UAVs sugiere que las restricciones de tiempo de vuelo no se cumplirán cuando no haya suficiente energía en todo el sistema. En este caso el promedio de tiempo disponible fue de 1.49 segundos siendo el máximo de 2.98 segundos. Adicionalmente, la diferencia entre

Tabla 6.15: Resultados de pruebas de batería restante para múltiples UAVs

UAVs	Total disp. (s)	Prom. disp. (s)	Min disp. (s)	Max disp. (s)
2	14.64	7.32	5.66	8.98
3	4.46	1.49	0.25	2.98
4	29.68	7.42	2.42	12.53
5	46.03	15.36	3.96	20.71

máximos y mínimos también se debe a la influencia de las otras variables de heterogeneidad, es decir, el ancho de barrido y la velocidad.

6.4.3 Resultados en simulación

Para evaluar el algoritmo en simulación, se utilizó el entorno Multiple Drone Simulator y el método de planificación de trayectorias back-and-forth ya que es más eficiente para las características de los drones simulados. Dado que este entorno funciona con un paquete de ROS2, permite realizar un seguimiento de la misión en herramientas como RViz¹⁶ como se puede observar en la Figura 6.20.

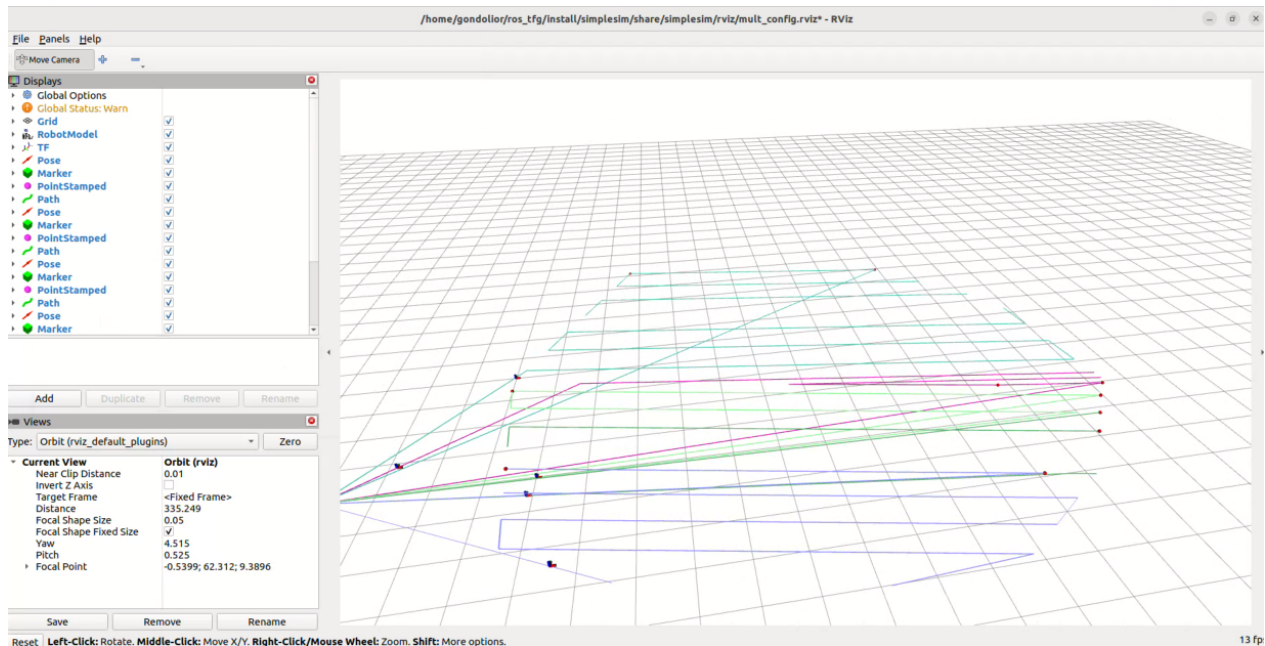


Figura 6.20: Visualización de la simulación en RViz

En la primera etapa de pruebas, se evaluará el rendimiento del algoritmo en base a los parámetros de velocidad y ancho de barrido. Luego, se evaluará por separado el rendimiento en base al tiempo de batería restante. En este contexto, para estandarizar los escenarios de simulación se han definido 5 tipos diferentes de UAVs que serán utilizados en simulaciones de entre 3 a 10 UAVs sobre una misma área estándar. Las características de cada aeronave se definen en la Tabla 6.16

¹⁶<http://wiki.ros.org/rviz>

Tabla 6.16: Tipos de UAVs utilizados en las pruebas de simulación

Tipo de UAV	Velocidad (m/s)	Ancho de barrido (m)
Tipo 1	20	15
Tipo 2	15	15
Tipo 3	20	20
Tipo 4	5	5
Tipo 5	10	10

Además, se definieron 5 escenarios de prueba de manera aleatoria, utilizando los diferentes tipos de aeronaves, la descripción de estos escenarios se muestra en la Tabla 6.17.

Tabla 6.17: Número de UAVs de cada tipo utilizado en los diferentes escenarios de simulación

Escenario	Tipo 1	Tipo 2	Tipo 3	Tipo 4	Tipo 5	Total
1	1	2	-	-	-	3
2	1	2	1	-	-	4
3	1	2	1	1	-	5
4	1	3	1	1	1	7
5	2	3	2	1	2	10

A continuación, se presentan los resultados del coeficiente de variación medidos sobre el tiempo de ejecución de la misión para cada uno de los escenarios en la Tabla 6.18.

Tabla 6.18: Resultados de coeficiente de variación para los tiempos de misión obtenidos en simulación en los diferentes escenarios de prueba

Escenario	Número de UAVs	CV
1	3	0.06
2	4	3.64
3	5	15.93
4	7	5.66
5	10	30.15

En la Tabla 6.18, se puede observar que el coeficiente de variación de los tiempos de ejecución de la misión depende de la variabilidad en las características de heterogeneidad más que en el número de UAVs. Así, en los escenarios 1,2 y 4 en los cuales predominan en mayor medida drones del tipo 2 (velocidad de 15 m/s y ancho de barrido de 15 m) en comparación con otros drones, se obtienen coeficientes de variación muy bajos y en los escenarios 3 y 5 este valor sube. Los resultados gráficos de simulación para 5 UAVs se presentan en la Figura 6.21.

Para evaluar el rendimiento del algoritmo bajo la limitación de batería, se han generado tres escenarios utilizando los mismos tipos de aeronaves de la Tabla 6.16 y añadiendo información del tiempo disponible de la batería, como se muestra en la Tabla 6.19.

Los resultados de tiempo de batería y coeficiente de variación para estas pruebas se muestran en la Tabla 6.20.

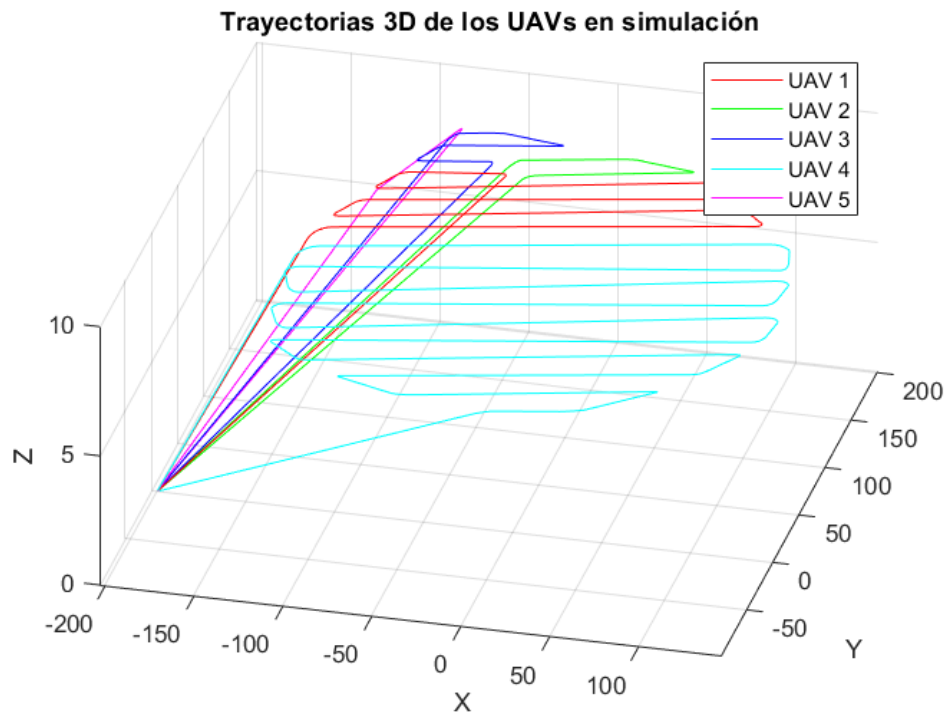


Figura 6.21: Resultados de simulación para 5 UAVs

Tabla 6.19: Escenarios propuestos para medir el rendimiento del algoritmo en tiempos de batería disponible

Escenario	Tipo UAV	Batería (s)
1	1	80
	2	200
	2	200
2	1	60
	2	60
	3	140
3	1	80
	2	60
	2	60
	3	210
	4	210

Tabla 6.20: Resultados de batería restante en el entorno de simulación

Escen.	Tipo	Bat. disp. (s)	Tiempo de vuelo	Bat. Rest. (s)	CV
1	1	80	72.30	7.70	31.39
	2	200	152.90	47.10	
	2	200	163.20	36.80	
2	1	60	63.20	-3.20	31.87
	2	60	64.50	-4.50	
	2	140	109.80	30.20	
	3	140	131.40	8.60	
3	1	80	48.60	31.40	54.31
	2	60	53.80	6.20	
	2	60	66.50	-6.50	
	3	210	161.80	48.20	
	4	210	172.10	37.90	

En la Tabla 6.20, se puede observar que en los escenarios 2 y 3 algunos UAVs no cumplen con las restricciones de tiempo de batería, esto es debido a que el algoritmo considera únicamente la batería suficiente para completar la misión y no toma en cuenta el tiempo de despegue, aterrizaje y llegada al punto de la misión. Sin embargo, no se reportan valores por encima de los 6.5 segundos, lo cual no sería un problema de fallo de batería en un entorno práctico. También se puede apreciar que el coeficiente de variación ha incrementado considerablemente respecto a las pruebas anteriores, esto demuestra que el algoritmo prioriza la restricción de batería sobre otras variables durante la asignación de la misión .

6.4.4 Discusión

El algoritmo de CoP-BINPAT presentado en esta sección responde a la necesidad de planificación de rutas de cobertura sobre una sola área con múltiples UAVs heterogéneos. Las variables de heterogeneidad se han elegido de acuerdo a criterios previos propuestos en el estado del arte. Los resultados de eficiencia computacional medidos en pruebas numéricas (Tabla 6.12) demuestran que CoP BINPAT es más eficiente computacionalmente al compararlo con propuestas similares de la literatura. También, de acuerdo con las Tablas 6.13 y 6.14 en pruebas numéricas y Tabla 6.18 en simulación, los resultados del coeficiente de variación en tiempos de ejecución son bajos, por lo cual, la misión se distribuye de manera eficiente. Finalmente, se ha evaluado las restricciones de batería, y se puede comprobar de acuerdo con las Tablas 6.15 y 6.20 que los UAVs cumplen estas restricciones únicamente para el desarrollo de la misión pero no toma en cuenta los tiempos de despegue ni el tiempo de transporte de la aeronave hacia los waypoints.

6.5 Gestión dinámica de trayectorias en sistemas multi-UAV

6.5.1 Diseño de experimentos y métricas de evaluación

La arquitectura propuesta junto con las técnicas de replanificación se evaluaron en diferentes escenarios tomando como referencia dos tipos de experimentos: simulación y pruebas reales.

Al igual que las técnicas anteriores, un factor importante de evaluación fue el balance de la distribución de la misión tanto en la etapa de planificación como en la replanificación medido sobre el costo de la misión a través del coeficiente de variación (ecuación 6.1). Finalmente, se tabulan los tiempos de CPU de los procesos de planificación y replanificación.

6.5.2 Pruebas de simulación

Para realizar las pruebas de simulación se ha implementado una primera versión del algoritmo en Flightmare, simulando un entorno de inspección de plantas fotovoltaicas, como se puede observar en la Figura 6.22.

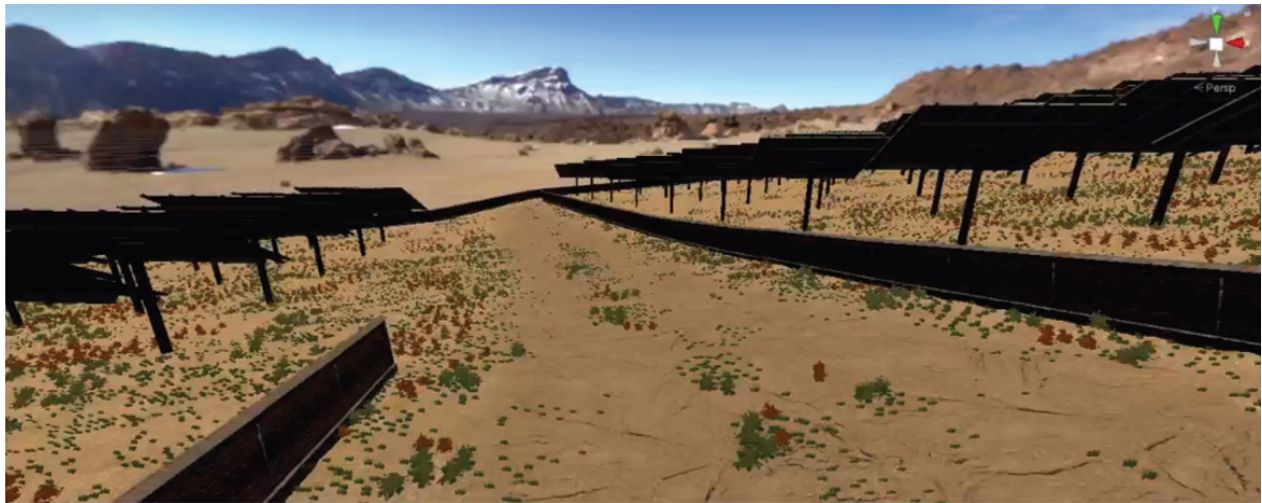


Figura 6.22: Entorno de simulación 3D desarrollado en Flightmare para evaluación del sistema

Para simular diferentes fallos se implementaron aeronaves simuladas con cámaras en la carga de pago, y con la misión de inspeccionar plantas fotovoltaicas en múltiples áreas como se puede observar en la Figura 6.23.

De esta manera se ejecutaron pruebas entre 3 y 5 UAVs en vuelos continuos (sin necesidad de replanificación) y para 2, 3 y 4 áreas de exploración, y los resultados se tabularon en la Tabla 6.21.

A continuación, se ejecutaron pruebas con uno y dos eventos inesperados durante el vuelo para lo cual el sistema ejecutó la tarea de replanificación. Los resultados se muestran en la Tabla 6.22.

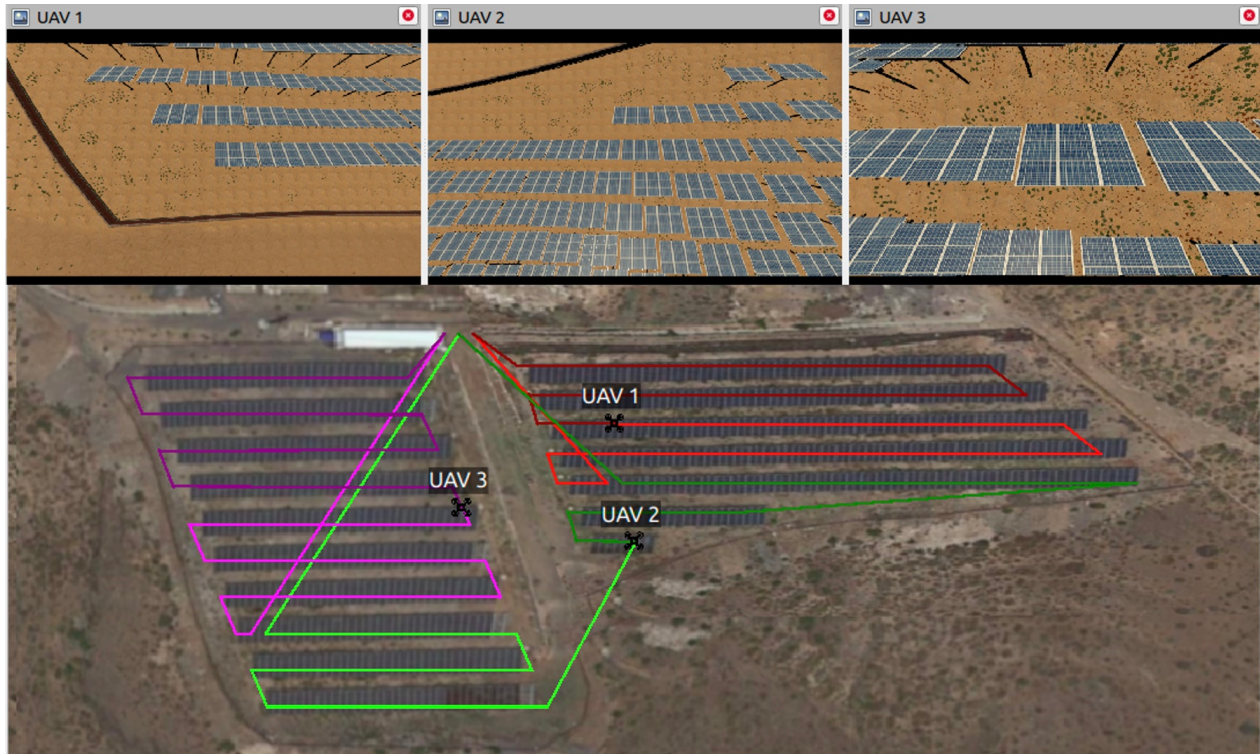


Figura 6.23: Simulación de 3 UAVs en el entorno Flightmare para inspección de plantas fotovoltaicas repartidas en dos áreas, con visualización de las cámaras

Tabla 6.21: Resultados de distancia reportados en la simulación para 3-5 UAVs y 2-4 áreas en vuelos continuos

Áreas	UAV1 (m)	UAV2 (m)	UAV3 (m)	UAV4 (m)	UAV5 (m)	CV
2	80.63	81.41	76.65	-	-	3.21
	67.14	71.73	66.44	67.19	-	3.56
	61.65	62.66	54.2	60.27	57.36	5.82
3	110.52	107.86	89.74	-	-	11.01
	92.56	92.59	94.26	86.32	-	3.83
	78.8	81.92	77.08	76.15	82.11	3.45
4	138.79	141.52	144.5	-	-	2.02
	113.03	114.72	109.17	102.68	-	4.86
	96.08	97.19	92.55	98.77	97.25	2.43

Tabla 6.22: Resultados de distancia reportados en la simulación para 3-5 UAVs con eventos de replanificación en vuelo

Eventos	UAV1 (m)	UAV2 (m)	UAV3 (m)	UAV4 (m)	UAV5 (m)	CV
1	11.53	161.44	164.95	-	-	1.52
	11.82	129.91	138.72	131.92	-	3.46
	11.97	115.44	118.93	105.62	108.71	5.43
2	152.74	32.16	164.78	-	-	5.36
	104.65	30.64	138.39	128.9	-	14.03
	107.81	34.83	115.3	119.49	130.06	7.86

Tabla 6.23: Tiempos de CPU para pruebas de simulación con 3-5 UAV y tres áreas de exploración

Eventos	UAVs	Tiempo de CPU (ms)			
		Plan	1er. Replan	2do. Replan	Promedio
1	3	13.61	11.30	-	12.45
	4	16.37	12.46	-	14.42
	5	13.32	12.60	-	12.96
2	3	15.24	11.70	14.12	13.69
	4	13.51	14.91	12.62	13.68
	5	13.15	14.08	15.10	14.11

Cabe recalcar que en la Tabla 6.22 durante los vuelos con un evento se excluye de los cálculos de *CV* al UAV perdido (representado con valores en negrita). Además, durante los vuelos con dos eventos, el UAV1 se pierde, luego, cuando el UAV2 se pierde, el UAV1 se recupera nuevamente para cumplir con el plan junto con el resto de las plataformas aéreas. En este caso, se ha excluido del cálculo del *CV* al UAV2 (valores en negrita) y se han mantenido los valores del UAV1.

De esta manera, las Tablas 6.21 y 6.22 muestran la distancia de cada aeronave al planificar y replanificar la misión, junto con los resultados del cálculo del *CV*. En vuelos continuos, el *CV* permanece en niveles bajos, lo cual corrobora los resultados del algoritmo de Powell-BINPAT. Sin embargo, en los planes en los que fue necesaria la replanificación, se puede observar cómo el *CV* aumenta relativamente poco entre los drones que han completado la misión pero permaneciendo relativamente bajo incluso en el caso excepcional en el que una aeronave se recupera y puede completar el plan de vuelo con el resto.

Para evaluar el tiempo de CPU de los algoritmos se han generado diferentes planes para tres regiones, simulando uno y dos eventos inesperados. Los resultados se presentan en la Tabla 6.23.

En la Tabla 6.23 se muestran tiempos computacionales bajos en las tareas de planificación y replanificación. Finalmente, resultados gráficos de las trayectorias recorridas por los UAVs en simulación se presentan en la Figura 6.24

La Figura 6.24 muestra las trayectorias de los UAVs en un vuelo con dos eventos inesperados. Al inicio de la operación, el UAV1 falla y regresa al punto inicial y el UAV2 junto con el

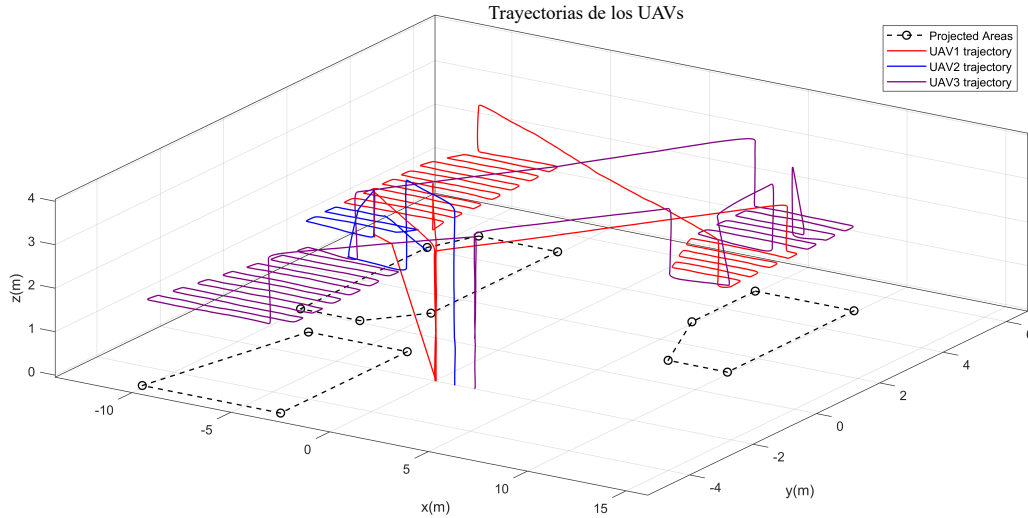


Figura 6.24: Resultados de trayectorias recorridas para 3 UAVs con replanificación en vuelo

UAV3 siguen la ejecución de la misión. En un segundo momento, el UAV2 falla y el UAV1 se recupera, por lo que el sistema replantea la misión para el UAV1 y el UAV3, tomando los caminos no completados del UAV2. También se pueden apreciar las transiciones de altura propuestas por el enfoque Powell-BINPAT para evitar colisiones. Algunos resultados gráficos adicionales se pueden observar en el Apéndice D.

6.5.3 Pruebas en entornos reales

Se realizaron pruebas con vuelos reales en el Robotics CAR Arena interiores utilizando plataformas Bitcraze Crazyflie. Para corroborar los resultados obtenidos en simulación, se realizaron tres pruebas de vuelo con tres UAVs con las siguientes características:

- **Vuelo 1:** Vuelo continuo sin necesidad de replanificación en vuelo.
- **Vuelo 2:** Se introduce un evento de fallo irrecuperable en medio del desarrollo de la misión.
- **Vuelo 3:** Uno de los UAVs regresa al punto de despegue en medio de la ejecución de la misión, luego uno de los UAVs restantes sufre un fallo irrecuperable por lo que el primer UAV entra nuevamente para a la misión para completarla.

Los resultados del costo de distancia se presentan en la Tabla 6.24, y los resultados de tiempo de CPU se encuentran en la Tabla 6.25.

En la Tabla 6.24, se han marcado en **negrita** los valores que no se consideraron para el cálculo del *CV*, debido a que estas distancias se corresponden con los drones que fallaron durante el vuelo, al igual que los resultados de simulación, en entornos reales se puede comprobar que el algoritmo distribuye las rutas equitativamente durante la planificación y replanificación. Además, de acuerdo a la Tabla 6.25 los tiempos de ejecución permanecen bajos, lo cual

Tabla 6.24: Resultados de distancia recorrida para 3 UAVs en las pruebas de vuelo reales

Número Vuelo	UAV1 (m)	UAV2 (m)	UAV3 (m)	Costo	CV
1	30.29	31.22	27.07	31.22	7.37
2	44.46	44.83	7.62	44.83	0.58
3	41.83	17.87	42.92	42.92	1.83

Tabla 6.25: Tiempos de CPU para 3 UAVs en las pruebas de vuelo reales

Número de vuelo	Tiempo de CPU (ms)		
	Plan	1er. Replan	2do. Replan
1	16.22	-	-
2	12.35	11.34	-
3	14.45	16.88	13.25

hace que la misión se ejecute de manera más fluida optimizando su tiempo de ejecución. Finalmente, las trayectorias recorridas en el vuelo 3 se presentan en la Figura 6.25.

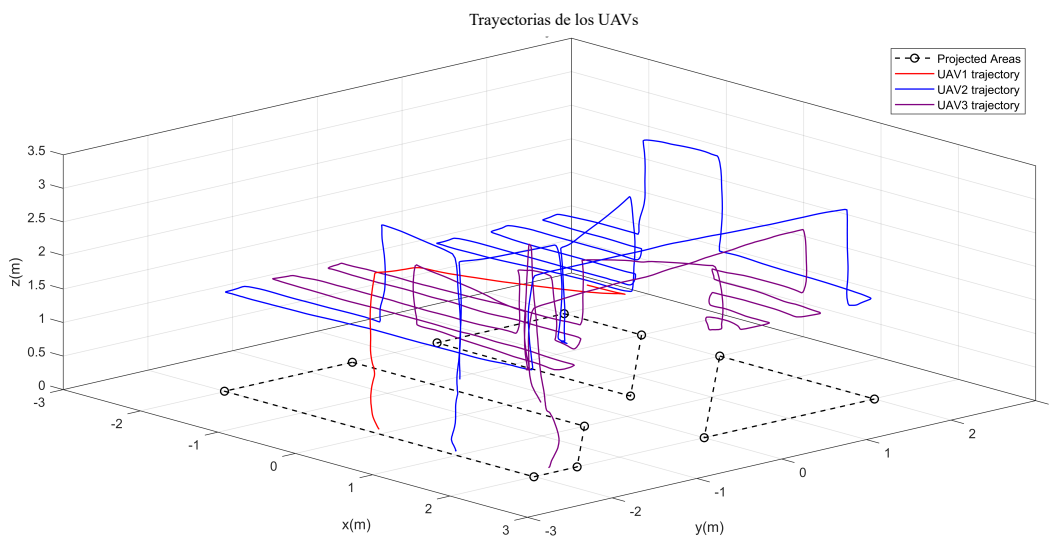


Figura 6.25: Trayectorias recorridas por los UAVs en un vuelo real con un evento de replanificación

Una comparativa con un sistema similar de la literatura propuesto por (Hong et al., 2022) se presentan en la Tabla 6.26.

En la Tabla 6.26, al contrastar los resultados con la investigación planteada en Hong et al., 2022, los autores proponen un sistema de replanificación dinámica utilizando un ordenador con CPU Intel i7 de 2.6 GHz y 16 GB de RAM para realizar sus experimentos, obteniendo de 260 a 710 ms en cada replanificación. Nuestro sistema por el contrario tiene capacidades de detección automática de fallos y obtiene un tiempo máximo de 14 ms en una CPU con características similares, también mantiene la continuidad de la misión manejando tiempos de replanificación más cortos. Además, nuestro algoritmo exhibe una mejor distribución de

Tabla 6.26: Resultados de comparación del tiempo de CPU de la arquitectura propuesta con la propuesta por Hong et al., 2022

Método	T. CPU (ms)	CV	Configuración de la misión	Configuración del ordenador
Gestión dinámica en Aerostack2 (nuestro)	14	7.37	3 UAVs, 2 áreas, replanificación automática	CPU Intel Core i7-7700HQ, 3.8 GHz, y 16 Gb de RAM
Hong et al., 2022	260-710	14.04	3 UAVs, 1 área, replanificación en base a una petición manual del operador	Intel i7 de 2.6 GHz y 16 GB

tareas entre los agentes (se ha calculado que dichos autores obtienen un CV de 14.03 en una planificación estática de 3 UAVs).

Resultados en video de estos experimentos reales se pueden encontrar en el canal de Vimeo del grupo CVAR-UPM ¹⁷.

6.5.4 Discusión

En este capítulo se propone una arquitectura de control para monitorizar el estado de los UAVs y el estado de la misión utilizando diferentes módulos de programación dentro del entorno de Aerostack2. De acuerdo con las Tablas 6.22, 6.23, 6.24 y 6.25, las estrategias implementadas para la detección temprana de errores de manera automática y las capacidades de replanificación de la misión en caso de fallo demostraron un rendimiento adecuado tanto en vuelos simulados como en implementaciones en entornos reales, demostrando una vez más eficiencia en la distribución de la misión y en el costo computacional. Esta eficiencia se mantiene incluso después de eventos de fallo. En comparación con otra propuesta similar presentada en la literatura (ver Tabla 6.26), nuestro sistema presenta características de detección automática y corrección de fallos, mayor eficiencia en la distribución de la misión, y tiempos de procesamiento más bajos.

6.6 Discusión general

A través de múltiples experimentos presentados en este capítulo, se puede demostrar que el enfoque de tres pasos para resolver el problema de gestión de misiones de cobertura de área, cuyos algoritmos se han integrado completamente en el capítulo 5 y cuyos resultados se observan en las Tablas 6.21-6.26, es eficiente computacionalmente, según demuestra el tiempo de CPU, y es eficiente en la distribución de la misión de acuerdo con los resultados del coeficiente de variación. También se han obtenido resultados favorables al comparar los

¹⁷<https://vimeo.com/847537551>

métodos propuestos con diferentes técnicas del estado del arte (Ver Tablas 6.2, 6.7, 6.8, 6.12 y 6.26). De esta manera se cumplen los objetivos de minimizar el costo computacional de la misión, distribuir las tareas de manera eficiente y generar un sistema reactivo en tiempo real frente a fallos. Esto demuestra el desarrollo secuencial de esta investigación, y la consecución de los objetivos mediante la metodología *divide y vencerás*. En términos generales, todas las técnicas presentadas en la presente tesis doctoral son principalmente heurísticas con alguna dependencia de técnicas de optimización que se ejecutan de manera eficiente; por lo cual el resultado es un sistema integral con capacidades de desempeñar las misiones de cobertura de área en tiempo real.

Capítulo 7

Conclusiones y trabajos futuros

Este capítulo presenta un resumen detallado de las conclusiones obtenidas a través de la investigación llevada a cabo durante el desarrollo de esta tesis. En él se analizan los resultados más significativos y se discute su impacto en el campo de estudio. Además, se sugieren direcciones futuras para continuar y expandir la investigación propuesta, destacando áreas que podrían beneficiarse de un análisis más profundo.

7.1 Conclusiones

7.1.1 Generación de trayectorias de cobertura de área

- Se propuso una solución novedosa de generación de trayectorias en espiral enfocada en aeronaves que navegan con rumbo hacia los waypoints. Comparado con otras propuestas de la literatura (Tabla 6.2), se demuestran algunas ventajas como la determinación eficiente del sentido de giro y la eficiencia computacional. En simulaciones con UAVs volando en dirección hacia los waypoints, la Tabla 6.4 demuestra que con esta configuración las trayectorias generadas se ejecutan más rápido que el enfoque back-and-forth. Sin embargo, esta propuesta continúa siendo ineficiente en aeronaves que se pueden mover en los cuatro grados de libertad para sobrevolar los waypoints.
- Se aportan métodos originales de planificación en múltiples áreas. El primero, basado en la solución del problema de viajante (TSP) y el segundo basado en vecinos más cercanos (NN). Al compararlos entre ellos (ver Figuras 6.6 y 6.7), TSP demuestra tener mejor resultado en eficiencia en el costo de la misión aunque NN obtuvo resultados similares. En tiempos de CPU, NN obtuvo resultados muy superiores a TSP.

7.1.2 Distribución de trayectorias en sistemas multi-UAV

- Para el caso homogéneo, se propone la técnica novedosa Powell-BINPAT, implementada junto con los métodos de planificación de trayectorias del capítulo 3. Los resultados medidos en pruebas de evaluación de rendimiento, simulaciones y experimentos reales (Tablas 6.6-6.10), demostraron que existe una mejora respecto a los sistemas actuales

en cuanto a la eficiencia en la distribución de la misión y que se ha logrado reducir el costo computacional.

- Para el caso heterogéneo, se extendió el método BINPAT proponiendo una métrica original denominada CoP para medir el rendimiento de un UAV en la misión de cobertura. De esta manera, se propuso el método CoP BINPAT. Esta solución se implementó con las técnicas de planificación de área única del capítulo 3, demostrando resultados superiores en eficiencia computacional comparada con otras propuestas de la literatura (Tabla 6.12). Los resultados medidos en pruebas de evaluación de rendimiento y simulación tabulados en las Tablas 6.13, 6.14 y 6.18, demostraron su eficiencia en la distribución de las misiones (aunque menor que Powell-BINPAT), y su eficiencia computacional. También se ejecutaron pruebas de restricción de tiempo de batería (Tablas 6.15, 6.19) comprobando que el algoritmo respeta las restricciones de tiempo restante de batería del grupo de UAVs.

7.1.3 Gestión dinámica de las trayectorias en sistemas multi-UAV

- Se implementó una arquitectura novedosa que recoge los principales resultados de los métodos desarrollados en las secciones anteriores (capítulos 3 y 4) y se desarrollaron nuevas soluciones que dieron lugar a un sistema de monitorización de la misión y estado de los UAVs en tiempo real, con capacidades de replanificación en caso de eventos inesperados. Las pruebas de rendimiento recogidas en las Tablas 6.21-6.25 demostraron la reactividad en tiempo real de este sistema gracias a la aplicación de algoritmos eficientes de detección de fallos y replanificación de la misión. Por otro lado, la Tabla 6.26 demuestra mayor eficiencia computacional y un mayor grado de automatización al compararla con una propuesta similar de la literatura.

7.2 Trabajos futuros

- **Evasión de obstáculos con procesamiento a bordo**

Esta línea de trabajo se propone con el fin de optimizar el desarrollo de la misión en la propuesta de Powell-BINPAT. Como se menciona en el capítulo ??, este método genera alturas de seguridad para cada UAV, si el número de UAVs aumenta, el costo de la misión se verá comprometido por estas alturas. Por lo tanto, es necesario implementar un sistema de evasión de obstáculos para eliminar la dependencia de estas alturas.

- **Planificación multi-área de sistemas heterogéneos**

La propuesta de asignación de tareas en sistemas heterogéneos tiene la desventaja de ser aplicable en área única. Por este motivo, se plantea extender la funcionalidad del algoritmo a múltiples áreas disjuntas sin comprometer la eficiencia del mismo.

- **Gestión dinámica de sistemas heterogéneos**

El problema de gestión dinámica de los sistemas heterogéneos radica en la dificultad de gestionar drones de diferentes características. Por ese motivo, una de las líneas de trabajo que se pretende estudiar es la gestión de este tipo de sistemas y los criterios para conseguir la replanificación en vuelo aprovechando las características de cada aeronave

de manera eficiente.

- **Técnicas de cobertura de área con sistemas multi-UAV volando en formación**

Un campo de investigación muy atractivo para la comunidad científica dentro de los sistemas Multi-UAV es el control de vuelo en formación, debido a sus múltiples aplicaciones. Por lo tanto, se propone estudiar la implementación de este tipo de sistemas en tareas de cobertura de área e identificar sus posibles ventajas y desventajas al compararlos con sistemas como los que propone esta tesis doctoral.

Referencias

- Afghah, F., Zaeri-Amirani, M., Razi, A., Chakareski, J., & Bentley, E. (2018). A coalition formation approach to coordinated task allocation in heterogeneous UAV networks. *2018 Annual American Control Conference (ACC)*, 5968-5975.
- Almadhoun, R., Taha, T., Seneviratne, L., & Zweiri, Y. (2019). A survey on multi-robot coverage path planning for model reconstruction and mapping. *SN Applied Sciences*, 1, 1-24.
- Arnold, R., Jablonski, J., Abruzzo, B., & Mezzacappa, E. (2020). Heterogeneous UAV multi-role swarming behaviors for search and rescue. *2020 IEEE Conference on Cognitive and Computational Aspects of Situation Management (CogSIMA)*, 122-128.
- 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.
- Awasthi, S., Fernandez-Cortizas, M., Reining, C., Arias-Perez, P., Luna, M. A., Perez-Saura, D., Roidl, M., Gramse, N., Klokowski, P., & Campoy, P. (2023). Micro UAV Swarm for industrial applications in indoor environment—A Systematic Literature Review. *Logistics Research*, 16(1).
- Bae, J., & Chung, W. (2019). Heuristics for two depot heterogeneous unmanned vehicle path planning to minimize maximum travel cost. *Sensors*, 19(11), 2461.
- Balampanis, F., Maza, I., & Ollero, A. (2017). Spiral-like coverage path planning for multiple heterogeneous UAS operating in coastal regions. *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*, 617-624.
- Baras, N., & Dasygenis, M. (2023). UGV Coverage Path Planning: An Energy-Efficient Approach through Turn Reduction. *Electronics*, 12(13), 2959.
- Barber, C. B., Dobkin, D. P., & Huhdanpaa, H. (1996). The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4), 469-483.
- Barrientos, A., Colorado, J., Cerro, J. d., Martinez, A., Rossi, C., Sanz, D., & Valente, J. (2011). Aerial remote sensing in agriculture: A practical approach to area coverage and path planning for fleets of mini aerial robots. *Journal of Field Robotics*, 28(5), 667-689.
- Bayındır, L. (2016). A review of swarm robotics tasks. *Neurocomputing*, 172, 292-321.
- Berger, C., Wzorek, M., Kvarnström, J., Conte, G., Doherty, P., & Eriksson, A. (2016). Area coverage with heterogeneous UAVs using scan patterns. *2016 IEEE international symposium on safety, security, and rescue robotics (SSRR)*, 342-349.
- Bertolazzi, E., & Frego, M. (2015). G1 fitting with clothoids. *Mathematical Methods in the Applied Sciences*, 38(5), 881-897.

- Boskovic, J., Knoebel, N., Moshtagh, N., Amin, J., & Larson, G. (2009). Collaborative mission planning & autonomous control technology (compact) system employing swarms of uavs. *AIAA Guidance, Navigation, and Control Conference*, 5653.
- Brandao, A. S., Barbosa, J. P., Mendoza, V., Sarcinelli-Filho, M., & Carelli, R. (2014). A multi-layer control scheme for a centralized uav formation. *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*, 1181-1187.
- Bupe, P., Haddad, R., & Rios-Gutierrez, F. (2015). Relief and emergency communication network based on an autonomous decentralized UAV clustering network. *SoutheastCon 2015*, 1-8.
- Cabreira, T. M., Di Franco, C., Ferreira, P. R., & Buttazzo, G. C. (2018). Energy-aware spiral coverage path planning for uav photogrammetric applications. *IEEE Robotics and automation letters*, 3(4), 3662-3668.
- Cabreira, T. M., Brisolará, L. B., & Paulo R, F. J. (2019). Survey on coverage path planning with unmanned aerial vehicles. *Drones*, 3(1), 4.
- Cai, Z., Liu, Y., Zhao, J., & Wang, Y. (2022). Virtual structure and artificial potential field-based cooperative control for uav formation. *International Conference on Guidance, Navigation and Control*, 366-375.
- Canny, J. (1988). *The complexity of robot motion planning*. MIT press.
- Chen, J., Qing, X., Ye, F., Xiao, K., You, K., & Sun, Q. (2022). Consensus-based bundle algorithm with local replanning for heterogeneous multi-UAV system in the time-sensitive and dynamic environment. *The Journal of Supercomputing*, 78(2), 1712-1740.
- Chen, J., Du, C., Zhang, Y., Han, P., & Wei, W. (2021). A clustering-based coverage path planning method for autonomous heterogeneous UAVs. *IEEE Transactions on Intelligent Transportation Systems*, 23(12), 25546-25556.
- Chen, J., Ling, F., Zhang, Y., You, T., Liu, Y., & Du, X. (2022). Coverage path planning of heterogeneous unmanned aerial vehicles based on ant colony system. *Swarm and Evolutionary Computation*, 69, 101005.
- Chen, J., Zhang, Y., Wu, L., You, T., & Ning, X. (2021). An adaptive clustering-based algorithm for automatic path planning of heterogeneous UAVs. *IEEE Transactions on Intelligent Transportation Systems*, 23(9), 16842-16853.
- Chen, X., Zhao, M., & Yin, L. (2020). Dynamic path planning of the UAV avoiding static and moving obstacles. *Journal of Intelligent & Robotic Systems*, 99, 909-931.
- Chen, Y., Yang, D., & Yu, J. (2018). Multi-UAV task assignment with parameter and time-sensitive uncertainties using modified two-part wolf pack search algorithm. *IEEE Transactions on Aerospace and Electronic Systems*, 54(6), 2853-2872.
- Cheng, P., Keller, J., & Kumar, V. (2008). Time-optimal UAV trajectory planning for 3D urban structure coverage. *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2750-2757.
- Choi, Y., Choi, Y., Briceno, S., & Mavris, D. N. (2020). Energy-constrained multi-UAV coverage path planning for an aerial imagery mission using column generation. *Journal of Intelligent & Robotic Systems*, 97(1), 125-139.
- Choset, H. (2001). Coverage for robotics—a survey of recent results. *Annals of mathematics and artificial intelligence*, 31, 113-126.
- Christofides, N. (2022). Worst-case analysis of a new heuristic for the travelling salesman problem. *Operations Research Forum*, 3(1), 20.

- Chung, S.-J., Paranjape, A. A., Dames, P., Shen, S., & Kumar, V. (2018). A survey on aerial swarm robotics. *IEEE Transactions on Robotics*, 34(4), 837-855.
- Daramouskas, I., Lappas, V., Patrinooulou, N., Meimetis, D., & Kostopoulos, V. (2023). Autonomous Systems for Defense Applications. En D. Kotarski & P. Piljek (Eds.), *Autonomous Vehicles*. IntechOpen.
- Darush, Z., Martynov, M., Fedoseev, A., Shcherbak, A., & Tsetserukou, D. (2023). Swarm-Gear: Heterogeneous Swarm of Drones with Reconfigurable Leader Drone and Virtual Impedance Links for Multi-Robot Inspection. *arXiv preprint arXiv:2304.02956*.
- De Benedetti, M., D'Urso, F., Fortino, G., Messina, F., Pappalardo, G., & Santoro, C. (2017). A fault-tolerant self-organizing flocking approach for UAV aerial survey. *Journal of Network and Computer Applications*, 96, 14-30.
- Delaunay, B., et al. (1934). Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7(793-800), 1-2.
- Deng, D., Jing, W., Fu, Y., Huang, Z., Liu, J., & Shimada, K. (2019). Constrained heterogeneous vehicle path planning for large-area coverage. *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 4113-4120.
- Dewangan, R. K., Shukla, A., & Godfrey, W. W. (2019). Three dimensional path planning using Grey wolf optimizer for UAVs. *Applied Intelligence*, 49, 2201-2217.
- Di Franco, C., & Buttazzo, G. (2015). Energy-aware coverage path planning of UAVs. *2015 IEEE international conference on autonomous robot systems and competitions*, 111-117.
- Do, H. T., Hua, H. T., Nguyen, M. T., Nguyen, C. V., Nguyen, H. T., Nguyen, H. T., & Nguyen, N. T. (2021). Formation control algorithms for multiple-uavs: a comprehensive survey. *EAI Endorsed Transactions on Industrial Networks and Intelligent Systems*, 8(27), e3-e3.
- Duan, X., Liu, H., Tang, H., Cai, Q., Zhang, F., & Han, X. (2019). A novel hybrid auction algorithm for multi-UAVs dynamic task assignment. *IEEE access*, 8, 86207-86222.
- Emel'yanov, S., Makarov, D., Panov, A. I., & Yakovlev, K. (2016). Multilayer cognitive architecture for UAV control. *Cognitive Systems Research*, 39, 58-72.
- Farsi, M., Ratcliff, K., Johnson, J., Allen, C., Karam, K., & Pawson, R. (1994). Robot control system for window cleaning. *Proceedings of 1994 American Control Conference-ACC'94*, 1, 994-995.
- Fernandez-Cortizas, M., Molina, M., Arias-Perez, P., Perez-Segui, R., Perez-Saura, D., & Campoy, P. (2023). Aerostack2: A Software Framework for Developing Multi-robot Aerial Systems. *arXiv preprint arXiv:2303.18237*.
- Fernando, M. (2021). Online flocking control of UAVs with mean-field approximation. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 8977-8983.
- Fevgas, G., Lagkas, T., Argyriou, V., & Sarigiannidis, P. (2022). Coverage path planning methods focusing on energy efficient and cooperative strategies for unmanned aerial vehicles. *Sensors*, 22(3), 1235.
- Flores Peña, P., Luna, M. A., Ale Isaac, M. S., Ragab, A. R., Elmenshawy, K., Martín Gómez, D., Campoy, P., & Molina, M. (2022). A proposed system for multi-UAVs in remote sensing operations. *Sensors*, 22(23), 9180.
- Fourlas, G. K., & Karras, G. C. (2021). A survey on fault diagnosis and fault-tolerant control methods for unmanned aerial vehicles. *Machines*, 9(9), 197.

- Galceran, E., & Carreras, M. (2013). A survey on coverage path planning for robotics. *Robotics and Autonomous systems*, 61(12), 1258-1276.
- Garey, M. R., & Johnson, D. S. (1990). *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co.
- Gerkey, B. P., & Mataric, M. J. (2003). Multi-robot task allocation: Analyzing the complexity and optimality of key architectures. *2003 IEEE international conference on robotics and automation (Cat. No. 03CH37422)*, 3, 3862-3868.
- Gerkey, B. P., & Mataric, M. J. (2004). A formal analysis and taxonomy of task allocation in multi-robot systems. *The International journal of robotics research*, 23(9), 939-954.
- Göktoğan, A. H., & Sukkarieh, S. (2009). Distributed simulation and middleware for networked UAS. *Unmanned Aircraft Systems: International Symposium On Unmanned Aerial Vehicles, UAV'08*, 331-357.
- Gupta, A., Baza, A., Dorzhieva, E., Alper, M., Makarova, M., Perminov, S., Fedoseev, A., & Tsetserukou, D. (2022). SwarmHive: Heterogeneous Swarm of Drones for Robust Autonomous Landing on Moving Robot. *2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring)*, 1-5.
- Hassan, M., & Liu, D. (2018). A deformable spiral based algorithm to smooth coverage path planning for marine growth removal. *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1913-1918.
- Hert, S., & Lumelsky, V. (1998). Polygon area decomposition for multiple-robot workspace division. *International Journal of Computational Geometry & Applications*, 8(04), 437-466.
- Hong, Y., Jung, S., Kim, S., & Cha, J. (2021). Autonomous mission of multi-uav for optimal area coverage. *Sensors*, 21(7), 2482.
- Hong, Y., Jung, S., Kim, S., & Cha, J. (2022). Extensions of Receding Horizon Task Assignment for Area Coverage in Dynamic Environments. *IEEE Transactions on Aerospace and Electronic Systems*.
- Hu, J., & Yang, J. (2018). Application of distributed auction to multi-UAV task assignment in agriculture. *International Journal of Precision Agricultural Aviation*, 1(1).
- Hu, J., Niu, H., Carrasco, J., Lennox, B., & Arvin, F. (2022). Fault-tolerant cooperative navigation of networked UAV swarms for forest fire monitoring. *Aerospace Science and Technology*, 123, 107494.
- Huang, W. H. (2001). Optimal line-sweep-based decompositions for coverage algorithms. *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, 1, 27-32.
- Jing, W., Deng, D., Wu, Y., & Shimada, K. (2020). Multi-UAV coverage path planning for the inspection of large and complex structures. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1480-1486.
- Jonker, R., & Volgenant, T. (1988). A shortest augmenting path algorithm for dense and sparse linear assignment problems. *DGOR/NSOR: Papers of the 16th Annual Meeting of DGOR in Cooperation with NSOR/Vorträge der 16. Jahrestagung der DGOR zusammen mit der NSOR*, 622-622.
- Khanam, Z., Saha, S., Ehsan, S., Stolkin, R., & Mcdonald-Maier, K. (2020). Coverage path planning techniques for inspection of disjoint regions with precedence provision. *IEEE Access*, 9, 5412-5427.

- Kurdi, H., AlDaood, M. F., Al-Megren, S., Aloboud, E., Aldawood, A. S., & Youcef-Toumi, K. (2019). Adaptive task allocation for multi-UAV systems based on bacteria foraging behaviour. *Applied Soft Computing*, *83*, 105643.
- Kurdi, H. A., Aloboud, E., Alalwan, M., Alhassan, S., Alotaibi, E., Bautista, G., & How, J. P. (2018). Autonomous task allocation for multi-UAV systems based on the locust elastic behavior. *Applied Soft Computing*, *71*, 110-126.
- Kuriki, Y., & Namerikawa, T. (2015). Formation control with collision avoidance for a multi-UAV system using decentralized MPC and consensus-based control. *SICE Journal of Control, Measurement, and System Integration*, *8*(4), 285-294.
- Li, J., Li, X., & Yu, L. (2018). Multi-UAV cooperative coverage path planning in plateau and mountain environment. *2018 33rd youth academic annual conference of Chinese association of automation (YAC)*, 820-824.
- Liu, H., Chen, Q., Pan, N., Sun, Y., & Yang, Y. (2020). Three-dimensional mountain complex terrain and heterogeneous multi-UAV cooperative combat mission planning. *IEEE Access*, *8*, 197407-197419.
- Liu, X., & Qiu, L. (2019). Bird flocking inspired control strategy for multi-UAV collective motion. *arXiv preprint arXiv:1912.00168*.
- Liu, Y., & Bucknall, R. (2018). A survey of formation control and motion planning of multiple unmanned vehicles. *Robotica*, *36*(7), 1019-1047.
- Low, C. B., & San Ng, Q. (2011). A flexible virtual structure formation keeping control for fixed-wing UAVs. *2011 9th IEEE international conference on control and automation (ICCA)*, 621-626.
- Luna, M. A., Ragab, A. R., Isac, M. S. A., Peña, P. F., & Cervera, P. C. (2021). A new algorithm using hybrid UAV swarm control system for firefighting dynamical task allocation. *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 655-660.
- Luna, M. A., Ale Isaac, M. S., Ragab, A. R., Campoy, P., Flores Peña, P., & Molina, M. (2022). Fast multi-UAV path planning for optimal area coverage in aerial sensing applications. *Sensors*, *22*(6), 2297.
- Luna, M. A., Isaac, M. S. A., Fernandez-Cortizas, M., Santos, C., Ragab, A. R., Molina, M., & Campoy, P. (2023). Spiral coverage path planning for Multi-UAV photovoltaic panel inspection applications. *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*, 679-686.
- Luna, M. A., Molina, M., Da-Silva-Gomez, R., Melero-Deza, J., Arias-Perez, P., & Campoy, P. (2024). A multi-UAV system for coverage path planning applications with in-flight re-planning capabilities. *Journal of Field Robotics*, *n/a*(n/a).
- Luo, H., Lin, H., Zhu, T., & Kang, Z. (2019). Complete coverage path planning of UUV for marine mine countermeasure using grid division and spanning tree. *2019 Chinese Control And Decision Conference (CCDC)*, 5016-5021.
- Madridano, Á., Al-Kaff, A., Martín, D., & De La Escalera, A. (2021). Trajectory planning for multi-robot systems: Methods and applications. *Expert Systems with Applications*, *173*, 114660.
- Madridano, Á., Al-Kaff, A., Martín, D., & de la Escalera, A. (2020). 3d trajectory planning method for uavs swarm in building emergencies. *Sensors*, *20*(3), 642.

- Mansouri, S. S., Kanellakis, C., Fresk, E., Kominiak, D., & Nikolakopoulos, G. (2018). Cooperative coverage path planning for visual inspection. *Control Engineering Practice*, *74*, 118-131.
- Martinez-Rozas, S., Rey, R., Alejo, D., Acedo, D., Cobano, J. A., Rodriguez-Ramos, A., Campoy, P., Merino, L., & Caballero, F. (2021). Skyeye team at MBZIRC 2020: A team of aerial and ground robots for GPS-denied autonomous fire extinguishing in an urban building scenario. *arXiv preprint arXiv:2104.01834*.
- Maza, I., & Ollero, A. (2007). Multiple UAV cooperative searching operation using polygon area decomposition and efficient coverage algorithms. En *Distributed Autonomous Robotic Systems 6* (pp. 221-230). Springer.
- Melo, A. G., Pinto, M. F., Marcato, A. L., Honório, L. M., & Coelho, F. O. (2021). Dynamic optimization and heuristics based online coverage path planning in 3D environment for UAVs. *Sensors*, *21*(4), 1108.
- Meng, B.-b., Gao, X., & Wang, Y. (2009). Multi-mission path re-planning for multiple unmanned aerial vehicles based on unexpected events. *2009 International Conference on Intelligent Human-Machine Systems and Cybernetics*, *1*, 423-426.
- Molina, M., Camporredondo, A., Bavle, H., Rodriguez-Ramos, A., & Campoy, P. (2019). An execution control method for the Aerostack aerial robotics framework. *Frontiers of Information Technology & Electronic Engineering*, *20*(1), 60-75.
- Mou, Z., Zhang, Y., Gao, F., Wang, H., Zhang, T., & Han, Z. (2021). Deep reinforcement learning based three-dimensional area coverage with UAV swarm. *IEEE Journal on Selected Areas in Communications*, *39*(10), 3160-3176.
- Mukhamediev, R. I., Yakunin, K., Aubakirov, M., Assanov, I., Kuchin, Y., Symagulov, A., Levashenko, V., Zaitseva, E., Sokolov, D., & Amirgaliyev, Y. (2023). Coverage path planning optimization of heterogeneous UAVs group for precision agriculture. *IEEE Access*, *11*, 5789-5803.
- Muñoz, J., López, B., Quevedo, F., Monje, C. A., Garrido, S., & Moreno, L. E. (2021). Multi UAV coverage path planning in urban environments. *Sensors*, *21*(21), 7365.
- Muran-Trus, A. (2024). *Entorno para Simulación de Sistemas de Robots Aéreos Autónomos* [Trabajo Fin de Grado de Ingeniería Informática]. Escuela Técnica Superior de Ingenieros Informáticos. Universidad Politécnica de Madrid.
- Nafis, F. R., et al. (2023). U-SPACE: El presente de las operaciones aéreas en el espacio Europeo. *Puente de Hierro*, *3*(1), 19-19.
- Nedjati, A., Izbirak, G., Vizvari, B., & Arkat, J. (2016). Complete coverage path planning for a multi-UAV response system in post-earthquake assessment. *Robotics*, *5*(4), 26.
- Ó'Dúnlaing, C., & Yap, C. K. (1985). A “retraction” method for planning the motion of a disc. *Journal of Algorithms*, *6*(1), 104-111.
- Orfanidis, G., Apostolidis, S., Kapoutsis, A., Ioannidis, K., Kosmatopoulos, E., Vrochidis, S., & Kompatsiaris, I. (2019). Autonomous swarm of heterogeneous robots for surveillance operations. *Computer Vision Systems: 12th International Conference, ICVS 2019, Thessaloniki, Greece, September 23–25, 2019, Proceedings 12*, 787-796.
- Peña, P. F., Ragab, A. R., Luna, M. A., Isaac, M. S. A., & Campoy, P. (2022). WILD HOPPER: A heavy-duty UAV for day and night firefighting operations. *Heliyon*, *8*(6).

- Peng, Q., Wu, H., & Xue, R. (2021). Review of dynamic task allocation methods for UAV swarms oriented to ground targets. *Complex System Modeling and Simulation*, 1(3), 163-175.
- Persson, A. H., Bondesson, L., & Börnin, N. (2006). Estimation of polygons and areas. *Scandinavian journal of statistics*, 33(3), 541-559.
- Pinto, M. F., Honório, L. M., Marcato, A. L., Dantas, M. A., Melo, A. G., Capretz, M., & Urdiales, C. (2021). Arcog: An aerial robotics cognitive architecture. *Robotica*, 39(3), 483-502.
- Powell, M. J. (1964). An efficient method for finding the minimum of a function of several variables without calculating derivatives. *The computer journal*, 7(2), 155-162.
- Rafifandi, R., Asri, D. L., Ekawati, E., & Budi, E. M. (2019). Leader-follower formation control of two quadrotor UAVs. *SN Applied Sciences*, 1, 1-12.
- Raja, G., Essaky, S., Ganapathisubramaniyan, A., & Baskar, Y. (2022). Nexus of deep reinforcement learning and leader-follower approach for aiot enabled aerial networks. *IEEE Transactions on Industrial Informatics*.
- Rizk, Y., Awad, M., & Tunstel, E. W. (2019). Cooperative heterogeneous multi-robot systems: A survey. *ACM Computing Surveys (CSUR)*, 52(2), 1-31.
- Ryll, M., Bühlhoff, H. H., & Giordano, P. R. (2014). A novel overactuated quadrotor unmanned aerial vehicle: Modeling, control, and experimental validation. *IEEE Transactions on Control Systems Technology*, 23(2), 540-556.
- Sabino, S., Horta, N., & Grilo, A. (2018). Centralized unmanned aerial vehicle mesh network placement scheme: A multi-objective evolutionary algorithm approach. *Sensors*, 18(12), 4387.
- Sanchez-Lopez, J. L., Pestana, J., De La Puente, P., & Campoy, P. (2016). A reliable open-source system architecture for the fast designing and prototyping of autonomous multi-uav systems: Simulation and experimentation. *Journal of Intelligent & Robotic Systems*, 84, 779-797.
- Semsch, E., Jakob, M., Pavlicek, D., & Pechoucek, M. (2009). Autonomous UAV surveillance in complex urban environments. *2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology*, 2, 82-85.
- Si, P., Yu, F. R., Yang, R., & Zhang, Y. (2015). Dynamic spectrum management for heterogeneous UAV networks with navigation data assistance. *2015 IEEE Wireless Communications and Networking Conference (WCNC)*, 1078-1083.
- Skorobogatov, G., Barrado, C., & Salamí, E. (2020). Multiple UAV systems: A survey. *Unmanned Systems*, 8(02), 149-169.
- Song, Y., Naji, S., Kaufmann, E., Loquercio, A., & Scaramuzza, D. (2020). Flightmare: A Flexible Quadrotor Simulator. *Conference on Robot Learning*.
- Tahir, A., Böling, J., Haghbayan, M.-H., & Plosila, J. (2020). Development of a fault-tolerant control system for a swarm of drones. *2020 International Symposium ELMAR*, 79-82.
- Tan, C. S., Mohd-Mokhtar, R., & Arshad, M. R. (2021). A comprehensive review of coverage path planning in robotics using classical and heuristic algorithms. *IEEE Access*, 9, 119310-119342.
- Tan, Y., & Zheng, Z.-y. (2013). Research advance in swarm robotics. *Defence Technology*, 9(1), 18-39.

- Torres, M., Pelta, D. A., Verdegay, J. L., & Torres, J. C. (2016). Coverage path planning with unmanned aerial vehicles for 3D terrain reconstruction. *Expert Systems with Applications*, 55, 441-451.
- Valente, J., Del Cerro, J., Barrientos, A., & Sanz, D. (2013). Aerial coverage optimization in precision agriculture management: A musical harmony inspired approach. *Computers and electronics in agriculture*, 99, 153-159.
- Vasquez-Gomez, J. I., Herrera-Lozada, J.-C., & Olguin-Carbajal, M. (2018). Coverage path planning for surveying disjoint areas. *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, 899-904.
- Venturini, F., Mason, F., Pase, F., Chiariotti, F., Testolin, A., Zanella, A., & Zorzi, M. (2021). Distributed reinforcement learning for flexible and efficient uav swarm control. *IEEE Transactions on Cognitive Communications and Networking*, 7(3), 955-969.
- Wang, H., Chen, M., & Fu, P. (2021). A distributed fault-tolerant mechanism for mission-oriented unmanned aerial vehicle swarms. *International Journal of Communication Systems*, 34(8), e4789.
- Wu, C., Dai, C., Gong, X., Liu, Y.-J., Wang, J., Gu, X. D., & Wang, C. C. (2019). Energy-efficient coverage path planning for general terrain surfaces. *IEEE Robotics and Automation Letters*, 4(3), 2584-2591.
- Wu, X., Yin, Y., Xu, L., Wu, X., Meng, F., & Zhen, R. (2021). Multi-UAV task allocation based on improved genetic algorithm. *IEEE Access*, 9, 100369-100379.
- Xiao, P., Li, N., Xie, F., Ni, H., Zhang, M., & Wang, B. (2023). Clustering-Based Multi-Region Coverage-Path Planning of Heterogeneous UAVs. *Drones*, 7(11), 664.
- Xie, J., Carrillo, L. R. G., & Jin, L. (2018). An integrated traveling salesman and coverage path planning problem for unmanned aircraft systems. *IEEE control systems letters*, 3(1), 67-72.
- Xie, J., & Chen, J. (2022). Multiregional Coverage Path Planning for Multiple Energy Constrained UAVs. *IEEE Transactions on Intelligent Transportation Systems*.
- Xu, P.-F., Ding, Y.-X., & Luo, J.-C. (2021). Complete coverage path planning of an unmanned surface vehicle based on a complete coverage neural network algorithm. *Journal of Marine Science and Engineering*, 9(11), 1163.
- Xu, Z., Zheng, X., & Zhou, J. (2023). Optimization design of collaborative beamforming for heterogeneous UAV swarm. *Physical Communication*, 61, 102202.
- Yan, M.-d., Zhu, X., Zhang, X.-x., & Qu, Y.-h. (2017). Consensus-based three-dimensional multi-UAV formation control strategy with high precision. *Frontiers of Information Technology & Electronic Engineering*, 18(7), 968-977.
- Yasutomi, F., Yamada, M., & Tsukamoto, K. (1988). Cleaning robot control. *Proceedings. 1988 IEEE International Conference on Robotics and Automation*, 1839-1841.
- Ye, F., Chen, J., Sun, Q., Tian, Y., & Jiang, T. (2021). Decentralized task allocation for heterogeneous multi-UAV system with task coupling constraints. *The Journal of supercomputing*, 77, 111-132.
- Ye, F., Chen, J., Tian, Y., & Jiang, T. (2020). Cooperative task assignment of a heterogeneous multi-UAV system using an adaptive genetic algorithm. *Electronics*, 9(4), 687.
- Yu, X., Jin, S., Shi, D., Li, L., Kang, Y., & Zou, J. (2020). Balanced multi-region coverage path planning for unmanned aerial vehicles. *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 3499-3506.

-
- Zhang, Y.-z., Hu, B., Li, J.-W., & Zhang, J.-D. (2016). Heterogeneous multi-UAVs cooperative task assignment based on GSA-GA. *2016 IEEE International Conference on Aircraft Utility Systems (AUS)*, 423-426.
- Zhao, X., Zong, Q., Tian, B., Zhang, B., & You, M. (2019). Fast task allocation for heterogeneous unmanned aerial vehicles through reinforcement learning. *Aerospace Science and Technology*, *92*, 588-594.
- Zheng, H., & Yuan, J. (2021). An Integrated Mission Planning Framework for Sensor Allocation and Path Planning of Heterogeneous Multi-UAV Systems. *Sensors*, *21*(10), 3557.
- Zhou, Y., Rao, B., & Wang, W. (2020). UAV swarm intelligence: Recent advances and future trends. *IEEE Access*, *8*, 183856-183878.
- Zhu, K., Han, B., & Zhang, T. (2021). Multi-UAV Distributed Collaborative Coverage for Target Search Using Heuristic Strategy. *Guidance, Navigation and Control*, *1*(01), 2150002.
- Ziquan, Y., Zhang, Y., Jiang, B., Jun, F., & Ying, J. (2022). A review on fault-tolerant cooperative control of multiple unmanned aerial vehicles. *Chinese Journal of Aeronautics*, *35*(1), 1-18.

Anexos

A Diseminación científica

A.1 Publicaciones

En este apartado se incluye una lista de publicaciones en revistas indexadas en JCR y conferencias relacionadas directamente con la presente tesis y una lista de publicaciones en revistas indexadas en JCR y conferencias relacionadas de manera indirecta.

Publicaciones en revistas indexadas en JCR relacionadas directamente con la tesis

- **Luna, M. A.**, Molina, M., Da-Silva-Gomez, R., Melero-Deza, J., Arias-Perez, P. & Campoy, P. (2024) A multi-UAV system for coverage path planning applications with in-flight re-planning capabilities. *Journal of Field Robotics*, 41, 1480–1497.
- **Luna, M. A.**, Ale Isaac, M. S., Ragab, A. R., Campoy, P., Flores Peña, P., & Molina, M. (2022). Fast multi-UAV path planning for optimal area coverage in aerial sensing applications. *Sensors*, 22(6), 2297.
- Awasthi, S., Fernandez-Cortizas, M., Reining, C., Arias-Perez, P., **Luna, M. A.**, Perez-Saura, D., ... & Campoy, P. (2023). Micro UAV Swarm for industrial applications in indoor environment—A Systematic Literature Review. *Logistics Research*, 16(1).
- Flores Peña, P., **Luna, M. A.**, Ale Isaac, M. S., Ragab, A. R., Elmenshawy, K., Martín Gómez, D., ... & Molina, M. (2022). A proposed system for multi-UAVs in remote sensing operations. *Sensors*, 22(23), 9180.

Publicaciones en conferencias relacionadas directamente con la tesis

- **Luna, M. A.**, Isaac, M. S. A., Fernandez-Cortizas, M., Santos, C., Ragab, A. R., Molina, M., & Campoy, P. (2023, June). Spiral coverage path planning for Multi-UAV photovoltaic panel inspection applications. In *2023 International Conference on Unmanned Aircraft Systems (ICUAS)* (pp. 679-686). IEEE.
- **Luna, M.A.**, Campoy, P., Molina, M. (2023). Evaluación de métodos de planificación de rutas de cobertura en sistemas multi-UAV para aplicaciones de teledetección. *Jornadas Nacionales de Robótica y Bioingeniería 2023*, 79-84.

Publicaciones en revistas indexadas en JCR relacionadas indirectamente con la

tesis

- Isaac, M. S. A., Peña, P. F., **Luna, M. A.**, Ragab, A. R., & Campoy, P. (n.d.). Sensing and Control Integration for Thrust Vectoring in Heavy UAVs: Real-World Implementation and Performance Analysis. *Unmanned Systems*, 1–23. doi:10.1142/S2301385025500396
- Ale Isaac, M. S., Ragab, A. R., **Luna, M. A.**, Ale Eshagh Khoeini, M. M., & Campoy, P. (2023). Thrust Vectoring Control for Heavy UAVs, Employing a Redundant Communication System. *Sensors*, 23(12), 5561.
- Ale Isaac, M. S., **Luna, M. A.**, Ragab, A. R., Ale Eshagh Khoeini, M. M., Kalra, R., Campoy, P., ... & Molina, M. (2022). Medium-scale uavs: A practical control system considering aerodynamics analysis. *Drones*, 6(9), 244.
- Isaac, M. S. A., Ragab, A. R., Garcés, E. C., **Luna, M. A.**, Peña, P. F., & Cervera, P. C. (2022). Mathematical modeling and designing a heavy hybrid-electric quadcopter, controlled by flaps. *Unmanned Systems*, 10(03), 241-253.
- Espinosa Peralta, P., **Luna, M. A.**, de la Puente, P., Campoy, P., Bavle, H., Carrio, A., & Cruz Ulloa, C. (2022). Performance analysis of localization algorithms for inspections in 2D and 3D unstructured environments using 3D laser sensors and UAVs. *Sensors*, 22(14), 5122.
- Peña, P. F., Ragab, A. R., **Luna, M. A.**, Isaac, M. S. A., & Campoy, P. (2022). WILD HOPPER: A heavy-duty UAV for day and night firefighting operations. *Heliyon*, 8(6).

Publicaciones en conferencias relacionadas indirectamente con la Tesis

- **Luna, M. A.**, Ragab, A. R., Isaac, M. S. A., Peña, P. F., & Cervera, P. C. (2021, October). A new algorithm using hybrid UAV swarm control system for firefighting dynamical task allocation. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (pp. 655-660). IEEE.
- Ragab, A. R., Isaac, M. S. A., **Luna, M. A.**, & Peña, P. F. (2021, October). Unmanned aerial vehicle swarming. In *2021 International Conference on Engineering and Emerging Technologies (ICEET)* (pp. 1-6). IEEE.

A.2 Proyectos de investigación

Wild-hopper: Heavy-duty UAV for day and night firefighting operations

Proyecto financiado por el programa Horizonte H2020 de la Unión Europea ¹ con número de referencia:880642, para desarrollar una plataforma de alta capacidad de carga para extinción de incendios.

- Actividad principal relacionada con la tesis:
 - Desarrollo de un sistema de replanificación en vuelo para gestión de emergencias tomando como base parte de su arquitectura para el desarrollo de técnicas de replanificación en vuelo para múltiples UAVs.

¹<https://cordis.europa.eu/project/id/880642/es>

- Actividades secundarias:
 - Integración de autopilotos en aeronaves prototipo, desarrollo de algoritmos de control y pruebas de vuelo. (Ver Figura A.1)



Figura A.1: Prueba de vuelo de un quadrotor de alta capacidad de carga con motores de combustión con control de paso variable

- Desarrollo de interfaces para software de control de estación de tierra para gestión de planes de vuelo y misión.
- Propuesta de arquitecturas de software para operación del sistema.

Faster: First responder advanced technologies for safe and efficient emergency response

Proyecto financiado por el programa Horizon H2020 de la Unión Europea ² con número de referencia 833507, cuyo objetivo fue el desarrollo de tecnologías de apoyo para equipos de respuesta rápida como policía, bomberos, protección civil y otros.

- Actividad principal relacionada con la tesis:
 - Desarrollo de un sistema multi-UAV para mapeo rápido de un área en una situación de emergencia. Una implementación temprana del algoritmo BINPAT para el proyecto se puede ver en Youtube ³
- Actividades secundarias:
 - Desarrollo de técnicas de cobertura de área para sistemas multi-UAV.
 - Implementación de una interfaz de control de tierra para monitorización de estado de los UAVs y carga de pago. (Ver Figura A.2)

²<https://cordis.europa.eu/project/id/833507/es>

³<https://www.youtube.com/watch?v=m6CssNmgwH0>

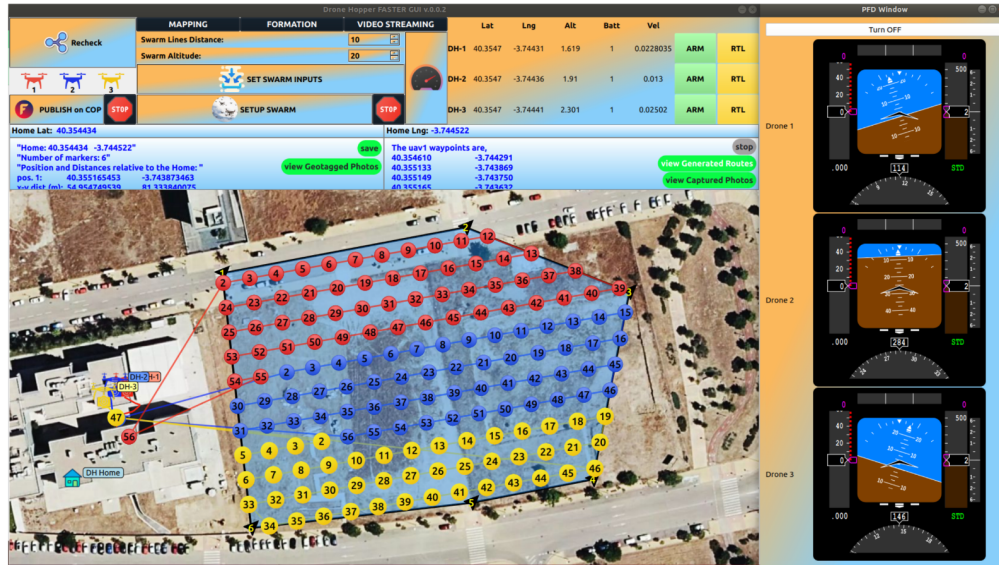


Figura A.2: Interfaz del software de estación de control en tierra para operación de enjambres

- Integración con un servidor en la nube para recibir solicitudes de de operación de manera remota

Copilot-CM: Control, supervisión y operación optimizada de plantas fotovoltaicas mediante integración sinérgica de drones, IoT y tecnologías avanzadas de comunicaciones

Proyecto financiado por la Comunidad de Madrid ⁴ con una ayuda para la realización de programas de actividades de I+D entre grupos de investigación de la Comunidad de Madrid, el objetivo de este proyecto fue la integración de varias tecnologías para la automatización de la inspección de paneles solares

- Actividad principal relacionada con la tesis:
 - Desarrollo de un sistema multi-UAV para inspección de paneles fotovoltaicos en múltiples áreas y con capacidades de replanificación en vuelo.
- Actividades secundarias:
 - Integración de algoritmos en el framework de investigación Aerostack2 ⁵ (Ver Figura A.3)
 - Implementación de una arquitectura de control y monitorización del estado de la misión.
 - Integración con un servidor en la nube para recibir solicitudes de de operación de manera remota

⁴<http://geiser.depeca.uah.es/copilot/index.php/es/>

⁵<https://aerostack2.github.io/>

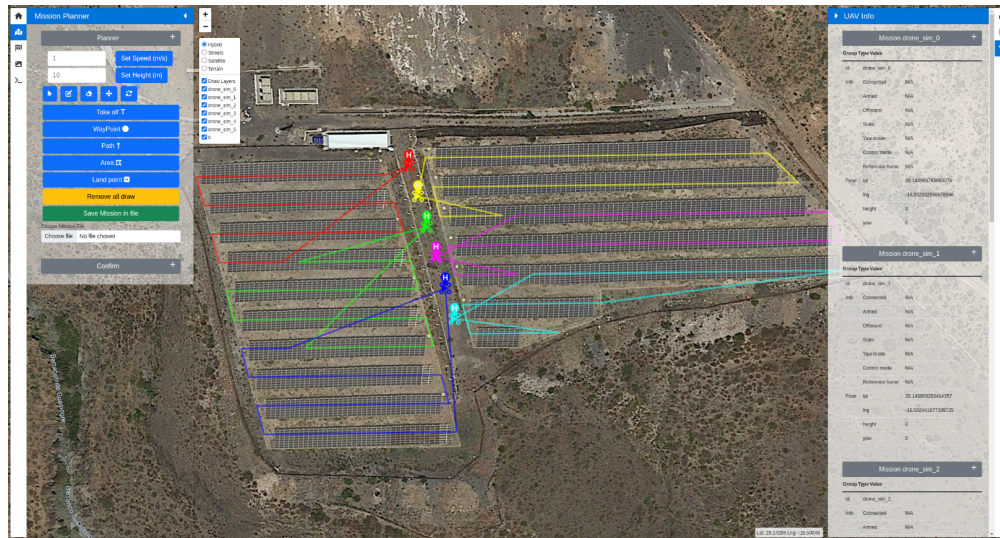


Figura A.3: Interfaz del software de estación de control en tierra para operación de enjambres

A.3 Transferencia tecnológica en el sector empresarial

Drone-Hopper

Es una empresa de del sector aeronáutico ⁶ fundada con el objetivo de diseñar, fabricar y comercializar sistemas aéreos no tripulados para aplicaciones de alta carga útil. Parte de la tesis doctoral se desarrolló en esta compañía aportando soluciones para proyectos de investigación, desarrollo de aplicaciones para control de estación de tierra e integración de algoritmos de control en plataformas experimentales. Por su parte, la compañía aportó plataformas de vuelo y pilotos para las pruebas en entornos reales durante la etapa de desarrollo de los algoritmos en sistemas Multi-UAV, así como sus instalaciones y laboratorios.

AHYRES Hardware and Software

Es una compañía del sector aeroespacial ⁷ que ofrece soluciones en el campo de los sistemas aéreos no tripulados (UAS) a nivel de hardware, diseñando electrónica, aviónica, comunicaciones y cargas de pago, y a nivel de software, para la gestión, control, navegación y sensores de UAS. Su principal mercado objetivo es la inspección de infraestructuras desatendidas. En esta compañía se desarrolló la última parte de la tesis doctoral referente a la arquitectura de control y monitorización para la gestión dinámica de misiones. Actualmente se está desarrollando un software para un sistema de planificación de recursos empresariales (ERP por sus siglas en inglés) enfocado en operadores y usuarios de UAVs con la funcionalidad de gestión Multi-UAV. La compañía ha aportado recursos de software, hardware y conocimientos prácticos en el desarrollo de la presente tesis.

⁶<https://www.drone-hopper.com/>

⁷<https://ahyres.com/>

B Tablas de resultados algoritmo Powell-BINPAT

B.1 Resultados de costo en distancia para pruebas entre 2-10 drones

UAV	Espiral	Back and forth
UAV1	770.80	701.24
UAV2	768.35	703.17
CV	0.16	0.14
Cost	770.80	703.17

UAV	Espiral	Back and forth
UAV1	584.78	481.16
UAV2	586.96	583.52
UAV3	591.34	584.22
CV	0.46	8.81
Cost	591.34	584.22

UAV	Espiral	Back and forth
UAV1	475.37	432.28
UAV2	495.79	447.34
UAV3	465.04	443.31
UAV4	484.97	442.35
CV	2.37	1.26
Cost	495.79	447.34

UAV	Espiral	Back and forth
UAV1	420.18	425.46
UAV2	434.16	440.61
UAV3	416.02	418.54
UAV4	416.93	436.72
UAV5	415.63	434.49
CV	1.66	1.86
Cost	434.16	440.61

UAV	Espiral	Back and forth
UAV1	444.75	385.83
UAV2	445.57	407.01
UAV3	398.91	433.95
UAV4	443.83	398.83
UAV5	433.57	444.30
UAV6	437.69	442.82
CV	3.75	5.41
Cost	445.57	444.30

UAV	Espiral	Back and forth
UAV1	427.00	428.03
UAV2	424.87	425.24
UAV3	422.15	401.93
UAV4	420.70	238.20
UAV5	425.96	411.90
UAV6	376.15	425.84
UAV7	414.91	402.67
CV	4.01	16.14
Cost	427.00	428.03

UAV	Espiral	Back and forth
UAV1	415.48	369.31
UAV2	441.06	385.30
UAV3	404.94	399.37
UAV4	438.93	393.71
UAV5	422.79	398.57
UAV6	437.52	399.53
UAV7	406.62	396.66
UAV8	419.82	372.20
CV	3.18	2.98
Cost	441.06	399.53

UAV	Espiral	Back and forth
UAV1	279.44	402.75
UAV2	402.30	401.19
UAV3	408.01	379.98
UAV4	368.49	280.04
UAV5	412.30	396.16
UAV6	403.93	405.12
UAV7	397.43	389.96
UAV8	371.16	387.14
UAV9	392.55	380.98
CV	10.20	9.60
Cost	412.30	405.12

UAV	Espiral	Back and forth
UAV1	410.95	367.51
UAV2	430.79	397.96
UAV3	286.38	382.80
UAV4	407.26	299.56
UAV5	430.92	372.80
UAV6	402.02	396.05
UAV7	278.00	401.70
UAV8	367.34	393.90
UAV9	409.49	391.81
UAV10	426.49	378.30
CV	14.10	7.50
Cost	430.92	401.70

B.2 Resultados de tiempo de vuelo en simulación del algoritmo Powell-BINPAT para múltiples pruebas en una sola área con algoritmo de planificación back-and-forth

UAV1	243.00	262.00	266.00	397.00	385.00	401.00
UAV2	258.00	284.00	286.00	400.00	413.00	426.00
UAV3	245.00	280.00	273.00	373.00	401.00	412.00
Costo	258.00	284.00	286.00	400.00	413.00	426.00
CV	3.32	4.19	3.72	3.97	3.50	3.04

UAV1	228.00	254.00	251.00	289.00	303.00	311.00
UAV2	213.00	226.00	239.00	315.00	324.00	355.00
UAV3	237.00	242.00	266.00	304.00	319.00	329.00
UAV4	240.00	245.00	266.00	325.00	338.00	338.00
UAV5	246.00	257.00	273.00	327.00	339.00	351.00
Costo	246.00	257.00	273.00	327.00	339.00	355.00
Av	232.80	244.80	259.00	312.00	324.60	336.80
CV	5.51	4.98	5.32	5.06	4.58	5.27

C Tablas de resultados algoritmo CoP-BINPAT

C.1 Resultados algoritmo CoP-BINPAT y planificación back and forth

UAV	Barrido (m)	Vel (m/s)	Bat (s)	CoP	Dist (m)	Estimación t(s)
UAV 0	10	20	1600	0.27	2258.37	112.9185
UAV 1	5	30	1600	0.2	3023.64	100.788
UAV 2	20	20	1600	0.53	2485.65	124.2825

Prom	112.66
Desv. St	9.59
CV	8.52

UAV	Barrido (m)	Vel (m/s)	Bat (s)	CoP	Dist (m)	Estimación t(s)
UAV 0	10	10	1600	0.06	603.14	60.31
UAV 1	15	50	1600	0.48	2402.18	48.04
UAV 2	5	20	1600	0.06	1143.52	57.17
UAV 3	20	30	1600	0.39	1893.95	63.13

Prom	57.16
Desv. St	5.67
CV	9.92

UAV	Barrido (m)	Vel (m/s)	Bat (s)	CoP	Dist (m)	Estimación t(s)
UAV 0	5	30	1600	0.14	1816.37	60.55
UAV 1	20	10	1600	0.18	861.35	86.14
UAV 2	5	50	1600	0.36	2996.59	59.93
UAV 3	20	20	1600	0.39	1698.84	84.94
UAV 4	10	10	1600	0.09	852.33	85.23

Prom	75.36
Desv. St	12.35
CV	16.39

UAV	Barrido (m)	Vel (m/s)	Bat (s)	CoP	Dist (m)	Estimación t(s)
UAV 0	20	40	1600	0.4	1699.5	42.49
UAV 1	10	20	1600	0.1	953.56	47.68
UAV 2	5	10	1600	0.05	806.33	80.63
UAV 3	15	50	1600	0.025	1796.84	35.94
UAV 4	10	10	1600	0.05	850.2	85.02
UAV 5	5	20	1600	0.375	1127.65	56.38

Prom	58.02
Desv. St	18.61
CV	32.08

UAV	Barrido (m)	Vel (m/s)	Bat (s)	CoP	Dist (m)	Estimación t(s)
UAV 0	20	20	1600	0.15	979.05	48.95
UAV 1	5	40	1600	0.07	1548.69	38.72
UAV 2	10	50	1600	0.19	2023.26	40.47
UAV 3	5	10	1600	0.02	908.31	90.83
UAV 4	10	20	1600	0.07	1086.53	54.33
UAV 5	10	50	1600	0.19	1917.72	38.35
UAV 6	20	40	1600	0.3	1489.13	37.23

Prom	49.84
Desv. St	17.74
CV	35.60

UAV	Barrido (m)	Vel (m/s)	Bat (s)	CoP	Dist (m)	Estimación t(s)
UAV 0	15	10	1600	0.04	539.87	53.99
UAV 1	10	40	1600	0.11	1288.43	32.21
UAV 2	15	20	1600	0.08	738.81	36.94
UAV 3	5	50	1600	0.07	1570.79	31.42
UAV 4	10	20	1600	0.06	875.3	43.77
UAV 5	20	50	1600	0.28	1486.21	29.72
UAV 6	15	50	1600	0.21	1271.18	25.42
UAV 7	10	50	1600	0.14	1446.21	28.92

Prom	35.30
Desv. St	8.79
CV	24.89

UAV	Barrido (m)	Vel (m/s)	Bat (s)	CoP	Dist (m)	Estimación t(s)
UAV 0	5	40	1600	0.06	1039.17	25.98
UAV 1	15	30	1600	0.13	1034.71	34.49
UAV 2	15	40	1600	0.18	1138.65	28.47
UAV 3	5	50	1600	0.07	1159.62	23.19
UAV 4	15	20	1600	0.09	846.82	42.34
UAV 5	15	10	1600	0.04	572.58	57.26
UAV 6	20	30	1600	0.18	952.74	31.76
UAV 7	10	20	1600	0.06	709.006	35.45
UAV 8	15	40	1600	0.18	972.01	24.30

Prom	33.69
Desv. St	10.13
CV	30.05

UAV	Barrido (m)	Vel (m/s)	Bat (s)	CoP	Dist (m)	Estimación t(s)
UAV 0	10	50	1600	0.15	1439.19	28.78
UAV 1	20	40	1600	0.24	1402.24	35.06
UAV 2	20	10	1600	0.06	666.33	66.63
UAV 3	20	20	1600	0.12	995.74	49.79
UAV 4	10	20	1600	0.06	648.74	32.44
UAV 5	15	40	1600	0.18	1135.58	28.39
UAV 6	15	20	1600	0.09	856.44	42.82
UAV 7	5	40	1600	0.06	1132.92	28.32
UAV 8	15	10	1600	0.04	762.62	76.26
UAV 9	5	10	1600	0.01	600.85	60.09

Prom	44.86
Desv. St	16.66
CV	37.13

C.2 Resultados algoritmo CoP-BINPAT planificación espiral

UAV	Barrido (m)	Vel (m/s)	Bat (s)	CoP	Dist (m)	Estimación t(s)
UAV 0	15	50	1600	0.6	3647.47	72.94
UAV 1	10	10	1600	0.08	942.34	94.23
UAV 2	20	20	1600	0.32	1618.2	80.91
Prom						82.7
Desv. St						8.78
CV						10.61

UAV	Barrido (m)	Vel (m/s)	Bat (s)	CoP	Dist (m)	Estimación t(s)
UAV 0	15	10	1600	0.09	820.05	82.01
UAV 1	10	50	1600	0.3	1974.75	39.49
UAV 2	20	40	1600	0.48	1956.51	48.91
UAV 3	10	20	1600	0.12	1203.39	60.17
Prom						57.64
Desv. St						15.85
CV						27.50

UAV	Barrido (m)	Vel (m/s)	Bat (s)	CoP	Dist (m)	Estimación t(s)
UAV 0	5	40	1600	0.1	1110.61	27.77
UAV 1	15	20	1600	0.15	864.28	43.21
UAV 2	15	10	1600	0.08	765.22	76.52
UAV 3	20	40	1600	0.41	1459	36.48
UAV 4	10	50	1600	0.26	1570.38	31.41
Prom						43.08
Desv. St						17.51
CV						40.65

UAV	Barrido (m)	Vel (m/s)	Bat (s)	CoP	Dist (m)	Estimación t(s)
UAV 0	15	20	1600	0.13	724.66	36.23
UAV 1	5	10	1600	0.02	668.9	66.89
UAV 2	15	50	1600	0.34	1624.86	32.50
UAV 3	10	10	1600	0.05	657.12	65.71
UAV 4	20	40	1600	0.36	1442.86	36.07
UAV 5	20	10	1600	0.09	741.92	74.19

Prom	51.93
Desv. St	17.25
CV	33.21

UAV	Barrido (m)	Vel (m/s)	Bat (s)	CoP	Dist (m)	Estimación t(s)
UAV 0	20	20	1600	0.15	979.05	48.95
UAV 1	5	40	1600	0.07	1548.69	38.72
UAV 2	10	50	1600	0.19	2023.26	40.47
UAV 3	5	10	1600	0.02	908.31	90.83
UAV 4	10	20	1600	0.07	1086.53	54.33
UAV 5	10	50	1600	0.19	1917.72	38.35
UAV 6	20	40	1600	0.3	1489.13	37.23

Prom	49.84
Desv. St	17.74
CV	35.60

UAV	Barrido (m)	Vel (m/s)	Bat (s)	CoP	Dist (m)	Estimación t(s)
UAV 0	5	40	1600	0.07	819.73	20.49
UAV 1	10	30	1600	0.11	1042.75	34.76
UAV 2	5	30	1600	0.05	921.64	30.72
UAV 3	20	50	1600	0.35	1280.61	25.61
UAV 4	10	50	1600	0.18	1159.93	23.20
UAV 5	20	10	1600	0.07	721.57	72.16
UAV 6	10	20	1600	0.07	742.36	37.12
UAV 7	15	20	1600	0.11	653.2	32.66

Prom	34.59
Desv. St	15.19
CV	43.91

UAV	Barrido (m)	Vel (m/s)	Bat (s)	CoP	Dist (m)	Estimación t(s)
UAV 0	5	40	1600	0.06	1039.17	25.98
UAV 1	15	30	1600	0.13	1034.71	34.49
UAV 2	15	40	1600	0.18	1138.65	28.47
UAV 3	5	50	1600	0.07	1159.62	23.19
UAV 4	15	20	1600	0.09	846.82	42.34
UAV 5	15	10	1600	0.04	572.58	57.26
UAV 6	20	30	1600	0.18	952.74	31.76
UAV 7	10	20	1600	0.06	709.006	35.45
UAV 8	15	40	1600	0.18	972.01	24.30

Prom	33.69
Desv. St	10.13
CV	30.05

UAV	Barrido (m)	Vel (m/s)	Bat (s)	CoP	Dist (m)	Estimación t(s)
UAV 0	5	10	1600	0.02	485.64	48.56
UAV 1	15	20	1600	0.09	838.65	41.93
UAV 2	15	10	1600	0.04	881.75	88.18
UAV 3	5	30	1600	0.04	1007.05	33.57
UAV 4	10	50	1600	0.15	1118	22.36
UAV 5	20	10	1600	0.06	575.32	57.53
UAV 6	5	30	1600	0.04	917.93	30.60
UAV 7	15	30	1600	0.13	934.19	31.14
UAV 8	20	30	1600	0.18	1002.18	33.41
UAV 9	20	40	1600	0.24	1056.64	26.42

Prom	41.37
Desv. St	18.53
CV	44.79

C.3 Pruebas de tiempo restante de vuelo

UAV	Barrido (m)	Vel (m/s)	Bat (s)	CoP	Dist (m)	Estimación t(s)
UAV 0	10	50	32	0.22	1316.81	26.34
UAV 1	20	40	70	0.78	2440.79	61.02

UAV	Barrido (m)	Vel (m/s)	Bat (s)	CoP	Dist (m)	Estimación t(s)
UAV 0	15	50	45	0.2	2101.23	42.02
UAV 1	10	20	60	0.46	1175.32	58.77
UAV 2	5 40	25	0.34	990.01	24.75	

UAV	Barrido (m)	Vel (m/s)	Bat (s)	CoP	Dist (m)	Estimación t(s)
UAV 0	20	30	16	0.27	407.33	13.58
UAV 1	10	40	41	0.46	1138.63	28.47
UAV 2	15	20	23	0.19	393.58	19.68
UAV 3	5	10	56	0.07	445.96	44.60

UAV	Barrido (m)	Vel (m/s)	Bat (s)	CoP	Dist (m)	Estimación t(s)
UAV 0	5	10	64	0.04	476.2	47.62
UAV 1	15	20	84	0.32	1293.9	64.70
UAV 2	15	10	18	0.03	140.41	14.04
UAV 3	5	30	62	0.12	1365.76	45.53
UAV 4	10	50	79	0.49	2914.31	58.29

D Resultados gráficos adicionales planificación dinámica

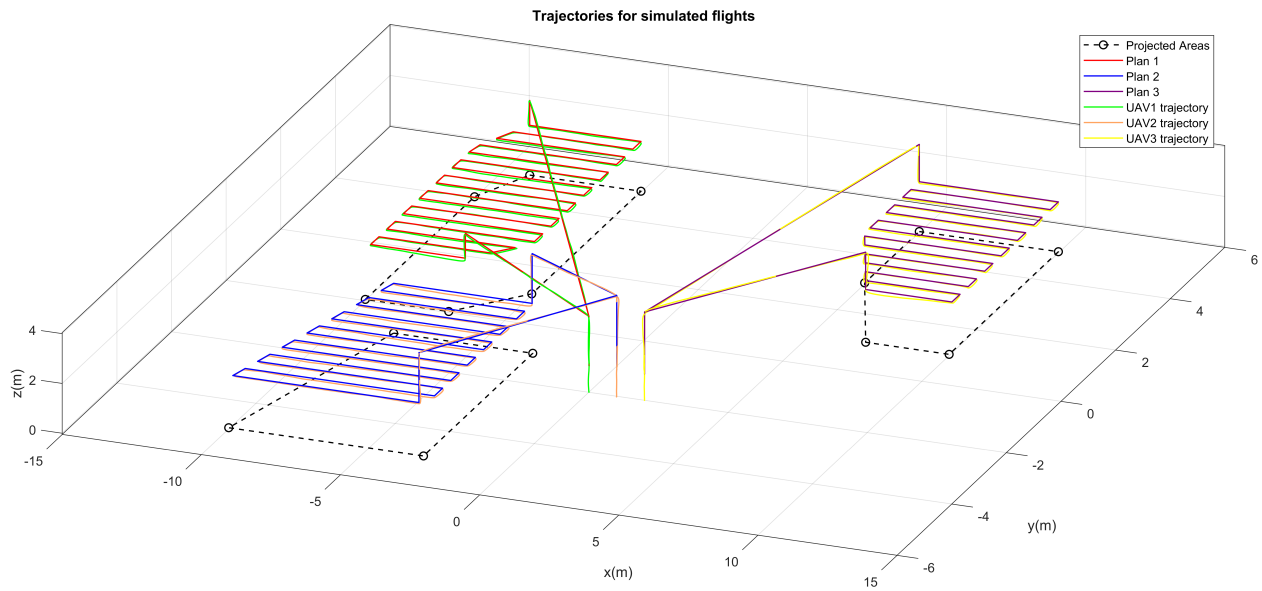


Figura D.4: Trayectorias recorridas por los UAVs en pruebas de simulación de vuelo continuo sin fallos

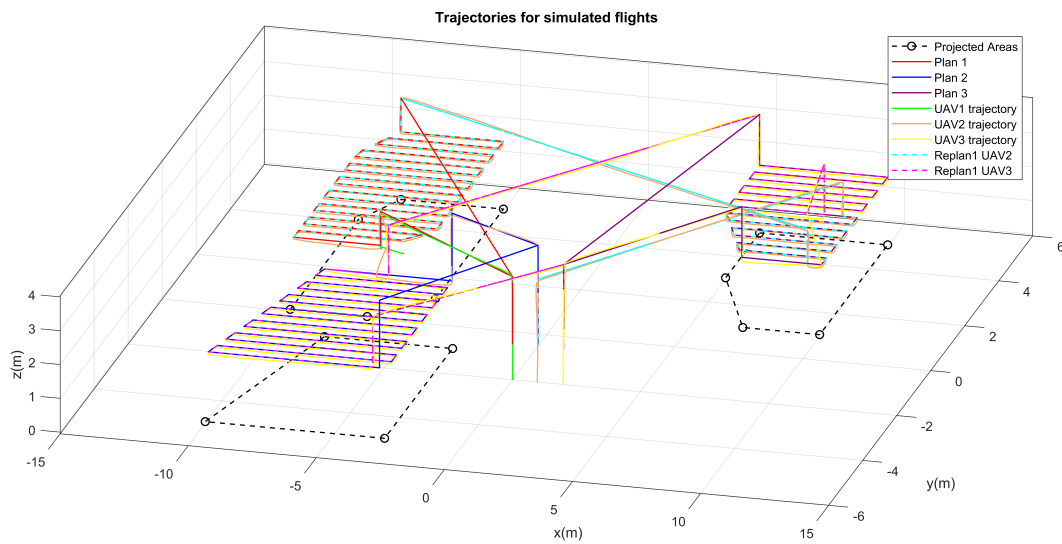


Figura D.5: Trayectorias recorridas por los UAVs en pruebas de simulación de vuelo con un fallo y una replanificación de los UAVs restantes

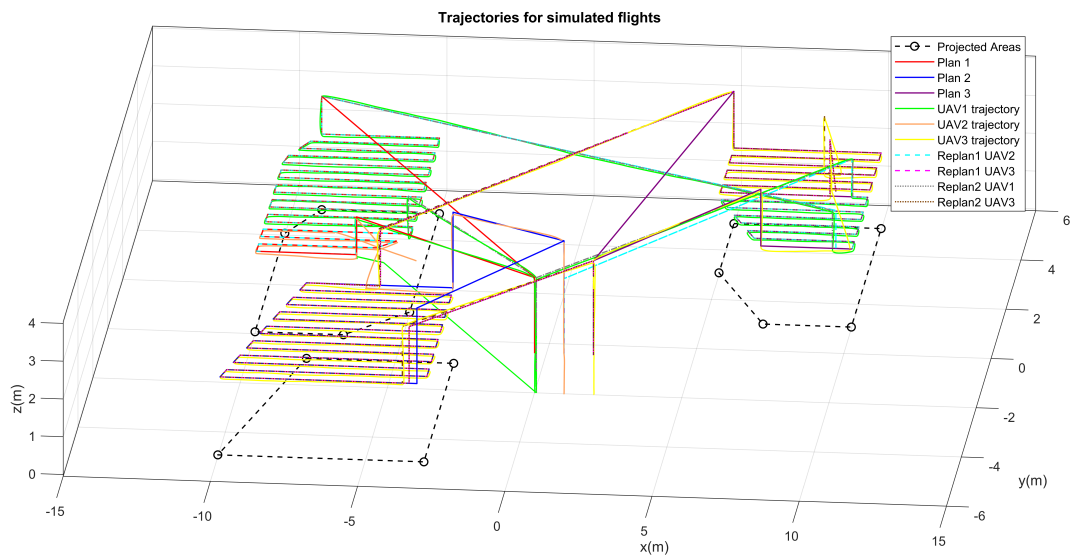


Figura D.6: Trayectorias recorridas por los UAVs en pruebas de simulación de vuelo con dos fallos y dos replanificaciones de los UAVs restantes

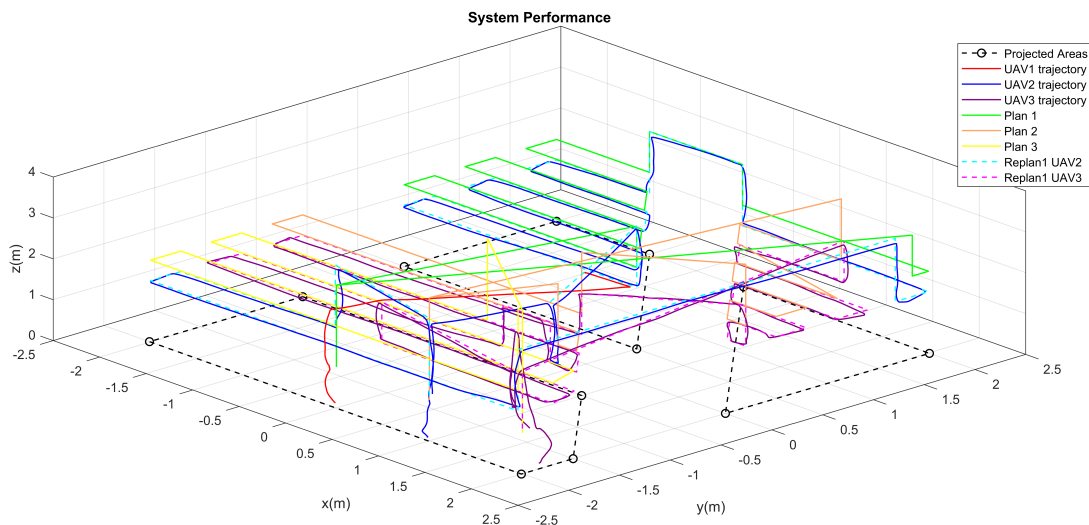


Figura D.7: Trayectorias recorridas por los UAVs en pruebas reales con dos fallos y replanificación de los dos UAVs restantes