

Trabajo Fin de Grado  
Grado en Ingeniería Aeroespacial  
Vehículos Aeroespaciales

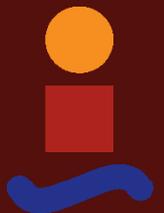
Estudio de la tolerancia a fallo de un sistema  
sincronizado de UAVs

Autor: Lourdes García Fernández

Tutor: José Miguel Díaz Báñez

Dep. de Matemática Aplicada II  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2015





Trabajo Fin de Grado  
Grado en Ingeniería Aeroespacial  
Vehículos Aeroespaciales

# **Estudio de la tolerancia a fallo de un sistema sincronizado de UAVs**

Realizado por:  
Lourdes García Fernández

Supervisado por:  
José Miguel Díaz Báñez  
Profesor Titular

Dep. de Matemática Aplicada II  
Escuela Técnica Superior de Ingeniería  
Universidad de Sevilla

Sevilla, 2015



Trabajo Fin de Grado: Estudio de la tolerancia a fallo de un sistema sincronizado de UAVs

Autor: Lourdes García Fernández

Tutor: José Miguel Díaz Báñez

El tribunal nombrado para juzgar el trabajo arriba indicado, compuesto por los siguientes profesores:

Presidente:

Vocal/es:

Secretario:

acuerdan otorgarle la calificación de:

El Secretario del Tribunal

Fecha:



# Agradecimientos

---

A mi familia por su constante apoyo y confianza, en especial a mi madre por su inmensa ilusión y por transmitirme su energía. Agradezco a mi tutor, Profesor J.M. Díaz Báñez y a E. Caraballo por su disponibilidad y su ayuda. Finalmente a mis compañeros de estudio, por su interés en el desarrollo de este trabajo.

*Lourdes García Fernández*  
*Sevilla, 2015*



# Resumen

---

El proyecto desarrollado se ha centrado en el estudio experimental de la tolerancia a fallo de un conjunto de vehículos formado por un sistema robots (considerados en principio como aéreos aunque pudieran ser terrestres) que deben mantenerse comunicados entre sí mientras completan una tarea sobre el territorio que sobrevuelan. Para ello, se analizará una serie de simulaciones realizadas en Matlab y se extraerán de ellas ciertos datos de interés que describen el grado de cumplimiento de las tareas asignadas.

Cada uno de los robots considerados posee un alcance limitado para realizar cualquier intercambio de información y vuela siguiendo una trayectoria que se le ha impuesto al inicio de la simulación. Cuando dos robots se encuentran suficientemente cerca, se establece un vínculo de comunicación que les permite intercambiar datos.

Con el fin de garantizar el máximo número de intercambios de información, el vuelo de los vehículos se programa de forma que el sistema pueda estar *sincronizado*, esto es, que cada par de vecinos se encuentren localizados periódicamente en un punto de la trayectoria, denominado enlace de comunicación, en el que se puede establecer un vínculo de comunicación viable. De esta forma, se maximiza la transferencia de información que permite a los vehículos actuar de modo descentralizado.

Nuestro estudio se enfoca a obtener resultados computacionales de las simulaciones en un sistema sincronizado cuando varios vehículos abandonan el área de trabajo y valoramos su repercusión en relación a varios parámetros de interés como la cantidad de área cubierta o el tiempo máximo que un vehículo vuela incomunicado.



# Índice

---

<i>Resumen</i>	III
<i>Índice de Figuras</i>	VII
<i>Índice de Tablas</i>	IX
<b>1. Introducción</b>	<b>1</b>
1.1. Campos de interés del estudio realizado	3
<b>2. Trabajos relacionados</b>	<b>5</b>
<b>3. Planteamiento del problema</b>	<b>7</b>
3.1. Modelo circular	9
3.2. Movimiento y sincronización de los robots	10
3.3. Percepción del estado del vecino y cambio de trayectoria	12
3.4. Datos a extraer de la simulación	13
<b>4. Implementación en Matlab</b>	<b>15</b>
<b>5. Resultados computacionales</b>	<b>19</b>
5.1. Estudio con cuatro vehículos en vuelo ( $N = 4$ )	19
5.2. Estudio con nueve vehículos en vuelo ( $N = 9$ )	28
<b>6. Análisis de resultados y propuestas de mejora</b>	<b>35</b>
6.1. Conclusiones	35
6.2. Generalizaciones del modelo	38
<b>Apéndice A. Código Matlab</b>	<b>43</b>
Referencias	51



# Índice de Figuras

---

3.1.	Representación de la línea de unión de trayectorias	8
3.2.	Criterio para nombrar a cada una de las trayectorias y vehículos (Grid)	9
3.3.	Definición del ángulo en la posición inicial, $\alpha_i$	10
3.4.	Definición de los ángulos formados por el eje x y la línea que une los centros de las trayectorias, $\beta_{ij}$	12
4.1.	Definición de los ángulos de posición, $\theta_i$	17
4.2.	Justificación de la distinción entre sentido horario y antihorario	17
5.1.	Primer cruce de vehículos en el caso de que ninguno se haya retirado	20
5.2.	Segundo cruce de vehículos en el caso de que ninguno se haya retirado	20
5.3.	Representación del instante final de la simulación	21
5.4.	Primer cruce de vehículos en el caso de que uno se haya retirado	22
5.5.	Segundo cruce de vehículos en el caso de que uno se haya retirado	23
5.6.	Situación al inicio de la simulación con fallo de vehículos de la misma fila	25
5.7.	Situación al inicio de la simulación con fallo de vehículos de la misma columna	26
5.8.	Representación del instante final de la simulación si mantienen el vuelo los vehículos 1 y 4	27
5.9.	Representación del instante final de la simulación si mantienen el vuelo los vehículos 2 y 3	27
5.10.	Representación del instante final de la simulación si mantienen el vuelo los vehículos 1 ó 4	28
5.11.	Representación del instante final de la simulación si mantienen el vuelo los vehículos 2 ó 3	28
5.12.	Cobertura del 100%	32
5.13.	Cobertura del 66,667%; configuración 1	32
5.14.	Cobertura del 66,667%; configuración 2	33
5.15.	Cobertura del 66,667%; configuración 3	33
5.16.	Cobertura del 33,333%; configuración 1	34
5.17.	Cobertura del 33,333%; configuración 2	34
6.1.	Cobertura en el vuelo de los vehículos 2 y 8	37
6.2.	Cobertura en el vuelo de los vehículos 1, 3, 5, 6, 8 y 9	38
6.3.	Comparación entre la trayectoria real (rojo) y la trayectoria del modelo (negro)	39
6.4.	Propuesta para el modelo de cambio de trayectoria	40



# Índice de Tablas

---

4.1.	Notación empleada en el planteamiento del problema	15
5.1.	Glosario de resultados obtenidos para $N = 4$	19
5.2.	Intervalos de tiempo entre que se produce la comunicación de uno de los vehículos con cualquiera de sus vecinos y con el siguiente vecino que se encuentra (se mantienen en vuelo todos los vehículos)	20
5.3.	Posiciones ocupadas por los vehículos en el caso de fallo del robot 1.	21
5.4.	Posiciones ocupadas por los vehículos en el caso de fallo del robot 2.	21
5.5.	Posiciones ocupadas por los vehículos en el caso de fallo del robot 3.	22
5.6.	Posiciones ocupadas por los vehículos en el caso de fallo del robot 4.	22
5.7.	Intervalos de tiempo entre que se produce la comunicación de uno de los vehículos con cualquiera de sus vecinos y con el siguiente vecino que se encuentra (se retira el vehículo 1)	23
5.8.	Posiciones ocupadas por los vehículos en el caso de fallo del 1 y el 2	24
5.9.	Posiciones ocupadas por los vehículos en el caso de fallo del 3 y el 4	24
5.10.	Posiciones ocupadas por los vehículos en el caso de fallo del 1 y el 3	24
5.11.	Posiciones ocupadas por los vehículos en el caso de fallo del 2 y el 4	24
5.12.	Posiciones ocupadas por los vehículos en el caso de fallo del 1 y el 4	24
5.13.	Posiciones ocupadas por los vehículos en el caso de fallo del 2 y el 3	24
5.14.	Intervalos de tiempo entre que se produce la comunicación de uno de los vehículos con cualquiera de sus vecinos y con el siguiente vecino que se encuentra (Fallo de vehículos de la misma fila)	25
5.15.	Intervalos de tiempo entre que se produce la comunicación de uno de los vehículos con cualquiera de sus vecinos y con el siguiente vecino que se encuentra (Fallo de vehículos de la misma columna)	25
5.16.	Situaciones a estudiar para el caso de $N = 9$	30
5.17.	Resultados de las situaciones estudiadas con $N = 9$	31
6.1.	Tabla resumen de la situación con $N = 4$	35
6.2.	Intervalos de tiempo entre que se produce la comunicación de uno de los vehículos con cualquiera de sus vecinos y con el siguiente vecino que se encuentra (vuelan 2 y 8)	37
6.3.	Posiciones ocupadas por los vehículos (vuelan 2 y 8)	37
6.4.	Intervalos de tiempo entre que se produce la comunicación de uno de los vehículos con cualquiera de sus vecinos y con el siguiente vecino que se encuentra (vuelan 1, 3, 5, 6, 8 y 9)	38
6.5.	Posiciones ocupadas por los vehículos (vuelan 1, 3, 5, 6, 8 y 9)	38



# 1 Introducción

---

Actualmente en la tecnología aeroespacial ha destacado el desarrollo de los Vehículos Aéreos No Tripulados, (en inglés Unmanned Aerial Vehicles, UAVs). Estas aeronaves destacan principalmente por ser altamente maniobrables, por su utilidad en el desempeño de trabajos cooperativos y por presentar costes inferiores a los costes típicos presentados por aeronaves tripuladas [1], [2]. Por otro lado, como ventaja adicional, con este tipo de vehículos se hace posible abordar tareas de mayor riesgo ya que no se exponen las vidas de los pilotos implicados, haciendo posible la operación en entornos hostiles, en condiciones de vuelo con escasa visibilidad o con condiciones climatológicas adversas.

El uso de un sistema de robots presenta ventajas en lo que respecta al tiempo de ejecución de las tareas debido a la realización simultánea de tareas, la tolerancia a fallo y el aumento de la precisión debido a la posibilidad de contrastar la información procedente de cada uno de los robots.

Tales ventajas hacen de esta novedosa tecnología un tema de estudio actual en la industria aeroespacial que va buscando mayores avances a medida que evoluciona. De este modo, estos vehículos son empleados actualmente en tareas como las de montaje de estructuras [3], inspección de instalaciones [4], vigilancia aérea [5], entre otras.

Un aspecto importante cuando se trabaja con varios de estos vehículos es la coordinación entre los mismos, lo cual es necesario tanto para evitar colisiones entre robots que operen en el mismo espacio como para mantener la proximidad necesaria para desarrollar las tareas asignadas [6]-[8].

El hecho de contar con varios vehículos en un mismo escenario es precisamente porque se busca que trabajen formando un conjunto asociado, cuyos miembros interaccionen entre sí. Para conseguir un sistema integrado que trabaje como tal, es necesario que exista comunicación entre sus miembros. Por ello, en todo sistema que desempeñe tareas de forma descentralizada se hace necesario un elevado nivel de comunicación. Asimismo, la tolerancia a fallos del sistema depende de la fiabilidad con la que se completen las comunicaciones realizadas entre los miembros del mismo.

En muchas situaciones, la comunicación directa entre todos los miembros no puede estar garantizada (por ejemplo casos en los que la zona de trabajo es mucho mayor que el alcance de comunicación de los robots). En estos casos la realización de tareas de forma cooperativa impone además ciertas restricciones para garantizar la comunicación por las cuales los vínculos que se establecen para el intercambio de información están disponibles sólo cuando los robots vecinos se encuentren dentro de su alcance de comunicación [9], [10]. Esta será la situación que se considere en el presente proyecto.

De este modo, se considera un sistema de robots que vuelan siguiendo trayectorias cerradas

de forma periódica y completando a su vez ciertas tareas asignadas. Cada uno de ellos necesita transmitir información sobre el estado de sus tareas asignadas al resto de miembros pero debe tenerse en cuenta que las conexiones de comunicación poseen un alcance limitado. Así, sólo se establece el vínculo de comunicación y se intercambia información cuando dos robots vecinos se encuentran dentro del rango de comunicación. Se considera que dos vecinos están sincronizados cuando son capaces de llegar al mismo tiempo al punto de sus respectivas trayectorias en el que pueden transmitirse información.

La necesidad del estudio del problema de la sincronización surge de forma natural en misiones como las de vigilancia y supervisión [11], [12] o en montaje de estructuras en las que los robots cargan y sitúan piezas de las mismas [13], por ejemplo. Además, el hecho de que dos vehículos aéreos vuelen e intercambien información de la forma que se ha planteado permite que no tengan que alterar su vuelo ni interrumpir la realización de sus tareas para comunicarse, ya que este método busca que el vínculo de comunicación se establezca de forma natural a la vez que los vehículos siguen su trayectoria.

El modelo que se ha tomado en el presente proyecto para asegurar la sincronización hace que se garantice el máximo intercambio posible de información teniendo en cuenta las características de los robots considerados y las restricciones que presentan. Además, el método permite que la sincronización se realice de un modo realista, sin necesidad de cambios rápidos de velocidad ni empleo de tiempo en esperas entre los vehículos que harían que el comportamiento del sistema se viera desmejorado.

Por otro lado, el hecho de considerar que el rango de comunicación de los robots estudiados sea pequeño resulta favorable desde el punto de vista de la seguridad de la información transmitida. De este modo se pueden evitar interferencias en la comunicación, lo cual representa un asunto importante en la aplicación práctica de vehículos aéreos no tripulados [14].

Otros casos posibles que no contemplan la necesidad de la sincronización que se ha elegido en el presente estudio podrían ser:

- Permitir que cada uno de los robots realice su tarea de forma autónoma e independiente de las condiciones en las que se encuentre el resto del sistema. Esto conllevaría que en caso de fallo de alguno de los robots, la tarea encomendada a éste quedara en todo momento sin completarse a partir del fallo.
- Emplear vehículos que puedan comunicarse en rangos de distancia más amplios o bien a tiempo real mediante tecnologías como el actual método de transferencia de datos en desarrollo, AFDX (Avionics Full Duplex Switched Ethernet). Tal y como se ha comentado previamente, esta solución podría dar lugar a interferencias en los intercambios de información realizados o a saturación de la información recibida que podrían dificultar la comunicación.

Durante la operación del sistema, cualquiera de los robots que integran el sistema podría verse obligado a abandonar el sistema, bien por avería o bien porque necesita retirarse temporalmente (para recargarse, por ejemplo), influyendo así en la sincronización del mismo. El objetivo en el diseño de este sistema sería el de buscar minimizar el efecto perjudicial que causa la marcha de alguno de los robots sobre el funcionamiento global del sistema. Esto se consigue considerando una serie de estrategias en el cambio de trayectorias que deben seguir los robots en caso de fallo de alguno de ellos.

Es por ello que en este proyecto se estudiarán determinadas situaciones experimentales en las que, partiendo de un sistema correctamente sincronizado, se retiren algunos de los miembros. Se busca comprobar la tolerancia a fallo del sistema descrito.

Con el objetivo de que ninguna zona del territorio considerado deje de estar sobrevolada, se impone la condición de que todo vehículo debe pasar a ocupar la trayectoria contigua si no recibe información de su vecino en el punto esperado.

Este trabajo se centra por tanto en el estudio de un sistema compuesto por un conjunto de vehículos aéreos no tripulados que vuela de forma sincronizada y completando una serie de tareas asignadas sobrevolando un determinado territorio. El objetivo del proyecto es el de comprobar cómo se ve afectado el sistema en el caso de que los vehículos considerados sufran daños durante el vuelo; en el presente proyecto se considerará el caso en el que el vehículo deja completamente de operar. Así, los resultados obtenidos ponderarán el efecto que se crea en la realización de las tareas cuando los miembros del sistema dejan de operar como deberían.

Para abordar el trabajo sobre el que versa el proyecto se llevará a cabo la implementación en Matlab de un algoritmo que simule el movimiento de los vehículos considerados, teniendo en cuenta restricciones que garantizan la coordinación del sistema y respetando la estrategia elegida para el cambio de trayectorias de los vehículos, que se presentará a lo largo del proyecto.

De este modo, el presente proyecto se divide en seis capítulos; en el capítulo 2 se presentan trabajos relacionados con el tema tratado en este estudio, de los cuales se ha tomado cierta información necesaria para la resolución. En el capítulo 3 se presenta el problema abordado y se detallan ciertas decisiones tomadas a la hora de resolverlo. En el capítulo 4 se explica la estrategia elegida para la implementación en Matlab del problema, definiendo una serie de variables necesarias para la comprensión del mismo y comentando el código que se ha diseñado. A continuación, el capítulo 5 recoge los resultados obtenidos con este método a partir de las simulaciones realizadas. Seguidamente, en el capítulo 6 se incluyen las conclusiones extraídas del trabajo y se proponen líneas de trabajo que podrían abordarse en una ampliación del proyecto. Finalmente, en el apéndice A se muestra el código que se ha desarrollado en el presente proyecto y que ha sido empleado para las simulaciones.

## 1.1 Campos de interés del estudio realizado

Debido a las características del problema considerado, basado en el empleo de un sistema sincronizado de vehículos para garantizar la máxima cobertura de una determinada zona en el espacio, se encuentran aplicaciones claras para el mismo, algunas de las cuales son:

- Vigilancia aérea
- Extinción de incendios
- Tareas de búsqueda y rescate
- Vaporización de productos líquidos sobre el terreno
- Realización de tareas independientes en una factoría

Para satisfacer en cada uno de los casos la tarea que se encomienda a los vehículos sería necesario adaptarlos para cada una de las aplicaciones incorporándoles el equipo necesario. En todas las situaciones que pudieran considerarse, el comportamiento del sistema se basará en la búsqueda del mismo objetivo: la zona que el conjunto de vehículos debe mantener cubierta tiene que permanecer así durante toda la maniobra, independientemente del número de fallos que se produzcan, y garantizando el mayor número de comunicaciones completas durante el vuelo.

De este modo, se garantiza, por ejemplo para el caso de la extinción de incendios, que a una zona

concreta del terreno le llegará un cierto caudal de líquido ignifugante en un periodo de tiempo medio que podría extraerse del presente estudio barajando incluso la posibilidad de fallo de alguno de los integrantes del sistema. Así, la utilidad del estudio en este caso podría ser la del dimensionamiento del sistema, es decir, la elección del número de vehículos que deberían incorporarse a la misión para lograr que el periodo de tiempo tras el que una zona del terreno recibe un determinado caudal sea siempre inferior a un valor fijado previamente, garantizando así la seguridad del terreno considerado.

## 2 Trabajos relacionados

---

El modelo que se ha considerado en el presente proyecto para garantizar la sincronización de los vehículos se ha obtenido de una serie de artículos en los que Díaz Báñez *et al.* [15], [12] estudian la problemática que se genera con respecto a la sincronización en la situación de que un grupo de robots aéreos (RAs) necesite comunicarse mientras llevan a cabo ciertas tareas asignadas en un cierto ambiente de cooperación. En estos artículos se considera el caso en el que cada uno de los robots posee un rango de alcance pequeño para establecer la comunicación y se encuentra siguiendo una trayectoria cerrada que se le ha asignado de antemano. De este modo, sólo cuando dos robots vecinos se encuentran lo suficientemente cerca, se podrá establecer la comunicación entre ellos permitiéndose así la transferencia de información.

Sin ánimo de ser exhaustivos, a continuación se van a detallar algunos artículos que se han centrado en el estudio de temas similares al tratado en el presente proyecto.

La coordinación descentralizada de un sistema de robots aéreos con ciertas restricciones en la comunicación ha sido estudiado en muchos casos específicos, especialmente en misiones de vigilancia y control. Casbeer *et al.* [16] y Alexis *et al.* [17] muestran estrategias para controlar de forma cooperativa el perímetro de un incendio mediante un sistema de pequeños vehículos aéreos no tripulados con restricciones en la comunicación entre ellos. En este caso, los vehículos operan a lo largo del perímetro del fuego y existe únicamente una trayectoria para todos ellos. De este modo, dos robots que vuelen en sentidos opuestos se cruzarán siempre, con lo que asignando sentidos de vuelo distintos a los miembros del sistema y haciendo que éstos lo cambien alternativamente cada vez que coincidan con uno de los demás, se garantizan los encuentros necesarios para la coordinación.

Franchi *et al.* [18] presentan una técnica para explorar áreas desconocidas usando un sistema de robots coordinados. La estrategia se basa en una base de datos en la que los robots almacenan información del entorno; así, los robots están continuamente explorando el territorio con el fin de añadir datos nuevos. Cuando casualmente dos de los miembros del sistema se encuentran, comparten la información que han almacenado y elaboran nuevas trayectorias que cada uno deberá seguir para explorar las zonas aún desconocidas.

Yoon and Qiao [19] plantean una estrategia llamada "*Sincronización X*" (XS) para usar sensores coordinados, autónomos y móviles en misiones de búsqueda. La idea consiste en dividir el área a explorar (un rectángulo como ejemplo por simplicidad) en  $n$  tramos, uno para cada robot. Los vínculos de comunicación se encuentran en los límites comunes entre dos tramos consecutivos. La información de los vehículos situados en los laterales va pasando hacia el centro y las decisiones de control fluyen desde el centro hacia el exterior. Se trata de una solución interesante pero no aplicable a vehículos con trayectorias irregulares.

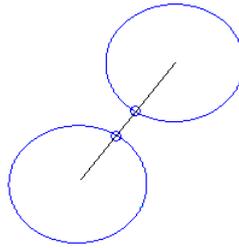
Es importante destacar que en el artículo de Díaz Báñez *et al.* [20] dedican un apartado al concepto de '*starvation*', según el cual, se dice que un robot aéreo se encuentra en este estado si, independientemente de cuánto tiempo permanezca en vuelo, no es capaz de encontrarse con alguno de los otros robots a través de ningún puente o enlace de comunicación. Esta variable será de interés en el presente estudio ya que se encuentra muy relacionada con el problema considerado. Se especificará en cada una de las simulaciones si el sistema se encuentra o no en '*starvation*', esto es, si todos los vehículos están en ese estado.

### 3 Planteamiento del problema

---

Se especificarán a continuación las condiciones del modelo que se considera en el presente proyecto:

- Se estudia un conjunto de  $N$  robots aéreos cuyos miembros necesitan intercambiarse información al mismo tiempo que cooperan en la realización de una cierta tarea que se les ha marcado.
- Cada uno de los robots posee un rango de comunicación fijo y limitado y vuela con altitud constante siguiendo una determinada trayectoria cerrada. Cada uno de ellos será designado según el nombre de la trayectoria que ocupe al inicio de la simulación. Se consideran trayectorias sin puntos en común con el fin de evitar colisiones entre los vehículos y aprovechar de forma óptima la cobertura ofrecida por el sistema.
- Los llamados *vínculos de comunicación* se establecen únicamente entre las trayectorias de dos vehículos que cumplan que la mínima distancia entre ellas no sobrepase el valor mínimo que toma el rango de comunicación de cada uno de los robots.
- Todos los vehículos poseen un mismo rango de comunicación, por lo que la condición que se impone es que la mínima distancia entre las trayectorias vecinas no sobrepase este valor. Se denomina, por tanto, *vínculo de comunicación* al segmento que une los dos puntos más próximos de cada par de trayectorias que cumple la condición anterior.
- Se designan dos robots como *vecinos* si existe un vínculo de comunicación entre las trayectorias en las que se hallan. Por su parte, se dice que dos robots vecinos se encuentran *sincronizados* pasan al mismo tiempo por el vínculo de comunicación.
- En cada una de las visitas al vínculo de comunicación los vehículos se intercambiarán la información correspondiente. Se asume que la comunicación se produce de forma automática cuando dos vehículos vecinos se encuentran sobre el vínculo de comunicación. Éste se sitúa sobre línea que une los centros de cada una de las trayectorias vecinas, representada en la figura 3.1. En ella las circunferencias azules en línea continua representan dos trayectorias vecinas y el vínculo de comunicación se corresponde con el segmento delimitado por las dos circunferencias pequeñas.
- Un sistema de robots está *sincronizado* si cada par de vecinos se encuentra sincronizado.



**Figura 3.1** Representación de la línea de unión de trayectorias.

En las condiciones anteriores, puede ocurrir que alguno de los vehículos deba abandonar el sistema con lo que sus vecinos no lograrán recibir información de él en el vínculo de comunicación. El primer vecino que se percate de esta situación, es decir, que pase por el vínculo de comunicación que los une, deberá cambiar de trayectoria ocupando la de robot retirado.

Este cambio de trayectoria debe realizarse de forma tal que la sincronización del sistema no se vea afectada; el robot que ha cambiado de posición debe ahora establecer vínculos de comunicación con sus nuevos vecinos. En el presente estudio se considera que como simplificación que la decisión y la actuación en el cambio de trayectoria se realiza de forma instantánea.

Puede darse el caso de que el número y la posición de los vehículos retirados del sistema lleve a una situación en la que ninguno de los robots restantes pueda cruzarse en ningún momento de la simulación con su robot vecino, es decir, puede darse 'starvation' (ya definida en el capítulo 2). Esta situación hace que el sistema deje de estar sincronizado por lo que este aspecto será objeto de estudio en los ensayos realizados.

A modo de glosario se detalla lo siguiente:

La mayor restricción que existe en la situación estudiada es el hecho de que se consideran robots que únicamente pueden intercambiar datos con otros vehículos en el caso de que se encuentren a una distancia menor o igual que su rango de comunicación. Este intercambio de información se hace estrictamente necesario ya que se busca que el estudio sea de un sistema interactivo en el que todos los miembros puedan llegar a conocer cuál es el estado del conjunto completo.

El hecho de que todos los vehículos estén comunicados permite entre otras cosas que, en el caso de que alguno de los robots se vea afectado y deje de funcionar, la tarea que éste tenía a su cargo pueda ser completada por cualquiera de sus vecinos. De este modo, es necesario conseguir que todos los vehículos pasen por ciertos puntos de sus trayectorias en los que sus vecinos se encuentren lo suficientemente cerca como para que la comunicación se complete con éxito. Para ello, es necesario que el conjunto se encuentre sincronizado, lo cual se consigue controlando las posiciones iniciales desde las que cada robot inicia su vuelo en un determinado instante.

Así, una vez conseguida la sincronización buscada, se pretende que cada robot pase a realizar la tarea de su vecino cada vez que se percate de que éste falle; deberá abandonar su trayectoria actual y pasar temporalmente a la de su vecino. Esto lleva a que la sincronización del sistema y la cobertura que el conjunto de robots ofrece se vea afectada, así como a posibles situaciones de 'starvation' en el conjunto. El presente estudio se centrará en comprobar los efectos que aparecen en estos aspectos a medida que algunos de los vehículos se retiran del conjunto.

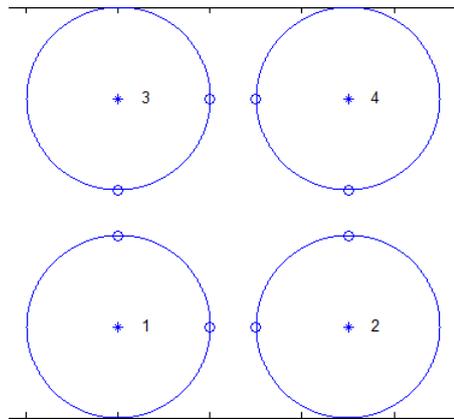
### 3.1 Modelo circular

El sistema estudiado podría en principio cubrir cualquier geometría presentada por el terreno pero en un primer caso se considerará un espacio cuya forma en planta será aproximada como cuadrada y dividida en un determinado número de sectores o cuadrículas del mismo tamaño según la cantidad de robots que vayan a considerarse en la simulación. De este modo, dado el número de robots involucrados en la simulación,  $N$ , se generarán  $N$  trayectorias distribuidas inicialmente en cuadrículas de dimensiones  $n \cdot n$  siendo

$$n = \sqrt{N} \quad (3.1)$$

Las trayectorias han sido definidas como circulares de radio unidad aunque el método puede ser extendido a cualquier perímetro que se necesite considerar siempre que pueda asumirse por parte del vehículo. Esto es, que las maniobras necesarias para seguir el recorrido fijado puedan ser realizadas por la aeronave manteniendo el vuelo en las condiciones requeridas.

Estas trayectorias se fijan en el espacio en una cuadrícula numerada desde la esquina inferior hasta la superior y de izquierda a derecha tal y como se representa en la figura 3.2 obtenida con  $N = 4$ . Por su parte, cada uno de los vehículos deberá ir desplazándose sobre las trayectorias marcadas y será nombrado según la posición que haya ocupado en el instante inicial.



**Figura 3.2** Criterio para nombrar a cada una de las trayectorias y vehículos (Grid).

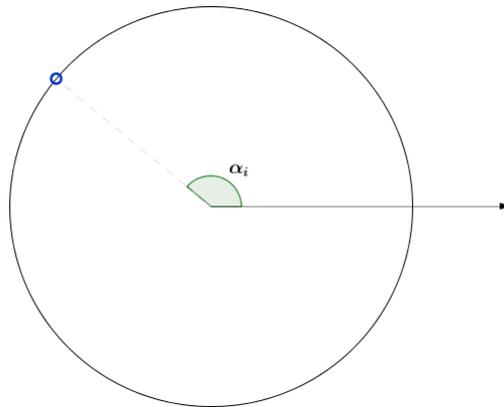
En la distribución de robots considerada se ha tenido en cuenta que no pueden existir intersecciones entre las trayectorias debido a que podría dar lugar a choques entre los vehículos considerados y a que si así fuera, una determinada región estaría sobrevolada por dos robots distintos lo cual no es óptimo a la hora de diseñar el sistema.

A la hora de diseñar la distribución de las trayectorias a seguir por los robots es necesario tener en cuenta que no pueden encontrarse demasiado cerca por el razonamiento presentado anteriormente ni tampoco demasiado lejos ya que debe considerarse el alcance máximo para la comunicación de cada uno de los robots. Es por ello que se ha incluido como dato de entrada al código una variable que se corresponde con el valor de la distancia máxima admisible entre cualquier pareja de puntos más cercanos pertenecientes a trayectorias vecinas ( $r$ ). Esta variable vendrá dada por el programador

ya que se debe determinar teniendo en cuenta cuál es la máxima distancia en la que es posible que la comunicación entre los robots se lleve a cabo de forma efectiva.

## 3.2 Movimiento y sincronización de los robots

Se ha considerado un movimiento circular y uniforme para simular el desplazamiento de los vehículos sobre cada una de las trayectorias, es decir, se considera una velocidad angular constante ( $\omega$ ) que se introduce como dato de entrada en el código ya que vendrá dada por las características de los robots elegidos.



**Figura 3.3** Definición del ángulo en la posición inicial,  $\alpha_i$ .

Debido a que todos los robots vuelan manteniendo velocidad angular constante en trayectorias circulares, la sincronización, si es posible, viene dada a partir de las posiciones iniciales que a cada uno de los vehículos se le indica. En este estudio, la posición inicial de cada vehículo se define de forma polar como en la figura 3.3 y para determinar la posición desde la que cada una de las aeronaves debería empezar se distinguen dos situaciones:

- Todos los vehículos vuelan en el mismo sentido
- Cada vehículo vuela en el sentido opuesto al que lo hacen sus vecinos.

### 3.2.1 Vuelo en el mismo sentido

En este apartado se establecen las condiciones que deben cumplirse para asegurar la completa sincronización del conjunto de robots durante el vuelo en el mismo sentido. Se asumirá que el sentido del vuelo es en sentido contrario al de giro de las agujas del reloj aunque este hecho no influye en las conclusiones obtenidas.

Se busca por tanto determinar los ángulos iniciales  $\alpha_1, \alpha_2, \dots, \alpha_N$ , correspondientes a cada uno de los vehículos, que hacen que una pareja de robots vecinos se encuentre sobre la línea que une los centros de sus respectivas trayectorias en el mismo momento tal y como se muestra en la figura 3.1. La definición de estos ángulos iniciales se realiza como se refleja en la figura 3.3.

Como ya se ha comentado en el capítulo 2, las condiciones necesarias y suficientes para que puedan sincronizarse el conjunto de robots se han extraído del artículo [15], que investiga precisamente este asunto y son las que siguen:

- El grafo generado por el conjunto de  $N$  trayectorias debe ser bipartito.
- Con respecto a la posición inicial de cada uno de los robots, se debe cumplir que  $\alpha_i = \pi + \alpha_j$  para cada pareja  $\{i,j\}$  de robots vecinos.

### 3.2.2 Vuelo en sentidos opuestos

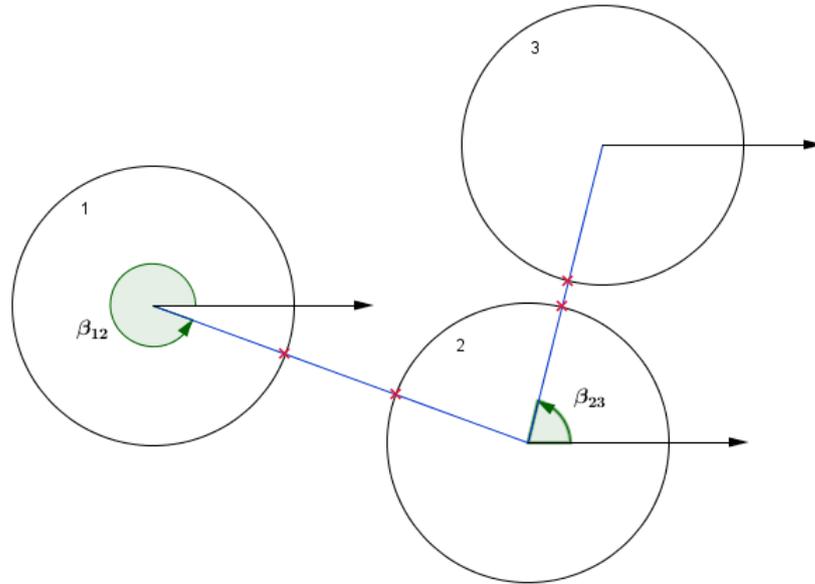
Veamos las condiciones que deben cumplirse para asegurar el vuelo sincronizado del conjunto de robots durante el vuelo en distintos sentidos. Es importante que se cumpla la condición de que cada uno de ellos debe encontrarse volando en el sentido opuesto al de todos sus vecinos.

De nuevo se busca determinar los ángulos iniciales  $\alpha_1, \alpha_2, \dots, \alpha_N$ , correspondientes a cada uno de los vehículos, que hacen que el sistema esté coordinado. La definición de estos ángulos iniciales se realiza igualmente como se refleja en la figura 3.3.

Del mismo modo, los resultados obtenidos del estudio se han extraído del artículo [15], detallado en el capítulo 2 y son los que siguen:

- El grafo generado por el conjunto de  $N$  trayectorias debe ser bipartito.
- Con respecto a la posición inicial de cada uno de los vehículos, se debe cumplir que  $\alpha_j = 2\beta_{ij} - \alpha_i \pm \pi$  para cada pareja  $\{i,j\}$  de robots vecinos.

Cabe indicar que el ángulo  $\beta_{ij}$  al que se hace referencia en las conclusiones anteriores se trata del ángulo formado por la línea que une los centros de las trayectorias de la pareja de vehículos  $\{i,j\}$  medido desde el eje horizontal positivo tal y como se representa en la figura 3.4.



**Figura 3.4** Definición de los ángulos formados por el eje x y la línea que une los centros de las trayectorias,  $\beta_{ij}$ .

### 3.3 Percepción del estado del vecino y cambio de trayectoria

Tal y como ya se ha indicado, el objetivo de este estudio consiste en comprobar la tolerancia a fallo de un sistema formado por un conjunto de robots que tiene como función la de sobrevolar una región determinada; esto es, se pretende verificar qué sucede en caso de fallo de alguno de tales robots.

Para que el sistema pueda ser considerado como tal debe existir comunicación entre los vehículos que lo forman. Esta razón es la que motiva que el vuelo de todos ellos esté sincronizado como se ha explicado previamente; debe garantizarse que en el momento en que uno de los vehículos se encuentre sobre la línea que une el centro de su trayectoria con la de cualquier otro vehículo, estos dos robots vecinos estarán a una distancia que permite la transmisión de datos entre ellos.

Si se da el caso de que alguno de los vehículos hubiera fallado y dejado, por tanto, de volar, el primero de sus vecinos que llegara a la línea de unión de trayectorias se percataría de este fallo en el sistema porque no se establecería la comunicación habitual y la función de éste sería la de asumir la tarea que le correspondía al que ha fallado ocupando la trayectoria del vehículo que falla. De este modo, es posible mantener el comportamiento deseado del sistema compartiendo las tareas asignadas a cada uno de los robots a medida que falla alguno de ellos.

Así, el vehículo consciente del fallo de su vehículo vecino deberá pasar a recorrer la trayectoria del robot dañado para completar así su tarea. La estrategia que se ha adoptado en la simulación para implementar el cruce de los vehículos es la de considerar que el robot cambiará de trayectoria siempre que pase por una zona en la que espera recibir información y no halle a su vecino. Realizará este cambio de forma instantánea; es decir, no se asumen aceleraciones que debería desarrollar para que el cambio de trayectoria no afectara en la realidad a la sincronización del sistema.

### 3.4 Datos a extraer de la simulación

Debido a que quiere comprobarse la fiabilidad del sistema ante diversas situaciones en las que se produce el fallo de una serie de robots, se ha considerado que resultaría interesante obtener de la simulación los siguientes parámetros:

- Tiempo máximo que puede llegar a encontrarse incomunicado un vehículo durante la simulación.
- Longitud máxima sobre las trayectorias que llega a quedarse sin sobrevolar durante la simulación.

Estos datos se extraerán para diversos casos que se generarán variando el número de integrantes del sistema y de fallos producidos en los mismos, así como modificando las posiciones de los vehículos retirados.

Además, en ciertos casos se especificará si la combinación de vehículos averiados elegida para la simulación lleva a la incomunicación total del sistema; es decir, se concretará si se produce o no *'starvation'*, ya detallada en el capítulo 2.



## 4 Implementación en Matlab

---

Con el fin de obtener los datos requeridos se procede a simular el vuelo de una serie de vehículos que cumplen con las restricciones que se han ido desarrollando a lo largo del presente proyecto. En primer lugar, se obtendrá el caso en el que todos los vehículos funcionan correctamente durante toda la simulación. Se considerarán distintas eliminaciones de vehículos buscando cubrir la mayor variedad casuística posible.

**Tabla 4.1** Notación empleada en el planteamiento del problema.

$ud$	Unidad de distancia	-
$ut$	Unidad de tiempo	-
$ua$	Unidad angular	-
$t$	Tiempo de duración de la simulación	$t=5ut$ o bien $t=8ut$
$N$	Número de vehículos que participarán en la simulación	$N = 4, N = 9$
$\omega$	Velocidad angular con la que volarán los vehículos, constante durante toda la simulación	$\omega = 5$ ua/ut
$r$	Rango de visibilidad que admite cada vehículo (Longitud del vínculo de comunicación)	$r = 0.5$ ud
$t_{max}(ut)$	Tiempo máximo que llega a permanecer un vehículo incomunicado durante la simulación	-
$t_{max}(\%)$	Porcentaje de tiempo de la simulación que llega a permanecer un vehículo incomunicado durante la simulación	-
$t_{ref}(ut)$	Tiempo en el que cualquiera de los vehículo completaría una vuelta completa a una de las trayectorias	-
$l_s(\%)$	Porcentaje de la longitud de todas las trayectorias que se sobrevuela durante la simulación	-

En la tabla 4.1 se presentan los parámetros que deben ser introducidos en la simulación y los valores elegidos, así como cierta notación necesaria para la interpretación de los resultados. Para el desarrollo del programa se ha definido y utilizado un gran número de variables auxiliares que no se incluirán a continuación aunque algunas de ellas se irán detallando a lo largo del presente capítulo.

Se procede a comentar el funcionamiento del código obtenido en Matlab para realizar las simulaciones, el cual se ha adjuntado al final de este trabajo para su consulta.

En primer lugar, cabe indicar que en su primera línea, el programa requiere de tres variables de entrada: número de vehículos que participarán en la simulación, velocidad angular a la que volarán los mismos y la longitud del vínculo de comunicación de cada par de vehículos ( $n$ ,  $\omega$  y  $r$ , respectivamente).

Una vez introducidos, se genera la cuadrícula correspondiente distribuyendo las trayectorias de cada uno de los vehículos de forma tal que todas ellas sean circunferencias de radio unidad, que no existan interferencias entre ellas y que se respete la distancia máxima entre los puntos más cercanos de trayectorias vecinas, correspondiente al rango de espacio máximo con el que cuenta una pareja de vehículos vecinos para transmitir y recibir información. Este proceso se va realizando por filas y va numerando las trayectorias tal y como se ha mostrado en la figura 3.2.

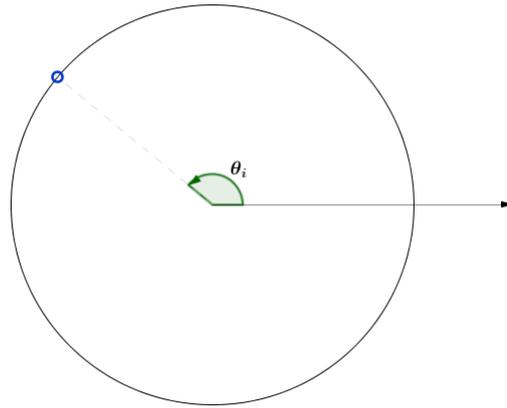
Se define el número de filas necesarias para contener en una malla cuadrada a todos los vehículos que se pretenden considerar; a continuación, por filas, se van obteniendo los centros de las circunferencias cumpliendo con las restricciones impuestas relativas a la distancia entre ellos. Finalmente se representan en una gráfica cada una de las circunferencias de radio unidad. Todo este proceso se corresponde con las líneas 14-56 del programa.

A continuación, a lo largo de las líneas 58-121 se genera una serie de celdas en Matlab en las que se almacenan las posiciones angulares donde los vehículos deberían encontrarse con cada uno de sus vecinos en caso de que éstos no hayan fallado. Estas posiciones se simbolizan con el símbolo "o" en la gráfica generada y pueden apreciarse también en la figura 3.2.

Se emplean diversas variables auxiliares (celdas  $A$ ,  $B$  y  $C$  para generar la celda  $\theta$ ), pero son los vectores  $\theta_{\text{tanum}}$ ,  $\theta_{\text{tanumvect}}$  y  $\theta_{\text{taveh}}$  los que resultan de interés. En primer lugar se sustituye el uso de la celda por el de vectores a través de la variable  $\theta_{\text{tanum}}$ , para ello se recurre a un bucle que recorre la celda por filas y almacena de forma temporal los datos buscados en la variable  $\theta_{\text{tanum}}$ , que contiene en cada bucle las posiciones angulares buscadas para una de las trayectorias; estos valores son almacenados en el vector  $\theta_{\text{taveh}}$ . Por otro lado, el vector  $\theta_{\text{tanumvect}}$  contiene los ángulos opuestos a los almacenados en  $\theta_{\text{taveh}}$ .

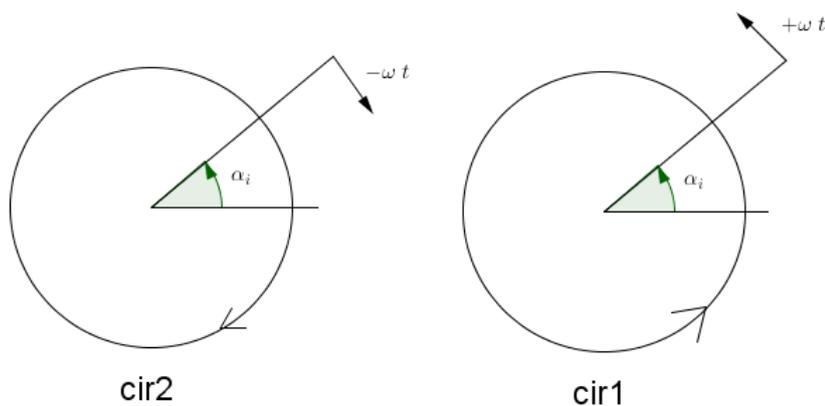
Seguidamente, entre las líneas 123 y 142 se imponen las condiciones iniciales en la posición con las que cada uno de los vehículos comenzará su vuelo. Estas posiciones de partida deben ser tales que garanticen la sincronización del sistema con lo que vienen dadas por el estudio realizado en la subsección dedicada al movimiento y sincronización de los robots, concretamente en el apartado 3.2.1. De este modo, se fija la posición inicial del vehículo que ocupa la trayectoria 1 y, a partir de él, las del resto de los vehículos según el modelo presentado en la sección referida.

Una vez generado el entorno en el que volarán los vehículos, y dadas las condiciones en las que lo harán, el programa solicita en la línea 145 que se le especifique el número de vehículos que fallarán a lo largo de la simulación, así como sus posiciones dentro del sistema. Para ello, se emplea la función auxiliar  $\text{caída\_vehículos}$ , que recibe los datos solicitados y hace que el sistema asuma la nueva situación (líneas 147-149).



**Figura 4.1** Definición de los ángulos de posición,  $\theta_i$ .

Posteriormente, entre las líneas 157 y 174 se identifican en el programa cuáles serán las trayectorias en las que los vehículos volarán en sentido horario (llamados en el programa *cir2*) y en cuáles lo harán en sentido antihorario (llamados en el programa *cir1*). Esto se hace necesario para poder implementar la evolución del ángulo theta (representado en la figura 4.1) en cada caso de forma independiente, tal y como se muestra de forma esquemática en la figura 4.2.



**Figura 4.2** Justificación de la distinción entre sentido horario y antihorario.

Se pasa ahora, a partir de la línea 176, a la implementación del movimiento de los vehículos. En primer lugar se introduce el tiempo durante el que debe realizarse la simulación, el cual es discretizado con el fin de que los robots realicen una serie de comprobaciones en cada uno de los intervalos obtenidos. Así, en cada intervalo los vehículos avanzarán en su trayectoria y realizarán la comprobación de si ha llegado a un punto en el que deba verificar la existencia de alguno de sus vecinos. En el caso de que se encontrara en alguno de tales puntos, deberá asimismo realizar la comprobación de si su vecino se encuentra o no en vuelo. Si no obtiene información de su vecino, deberá pasar a realizar la tarea de la trayectoria contigua y por el contrario, si se produce el encuentro entre ambos, se deben almacenar las variables requeridas para el estudio.

La evolución en el movimiento de los vehículos se simula variando el ángulo  $\theta$  en función de

la velocidad angular de entrada y en el sentido requerida según el vehículo se desplace de forma horaria o antihoraria (líneas 185-199).

Cada uno de los vehículos tiene información de en qué trayectoria se encuentran (línea 202), de cuáles son sus vecinos (se define la variable *vecinos* entre las líneas 204 y 226) así como de en qué valores del ángulo  $\theta$  debería comprobar si éstos se mantienen en vuelo (vector *thetaveh* ya comentado anteriormente). De este modo, una vez que el ángulo  $\theta$  de cualquiera de los vehículos coincide con el que lo sitúa en el punto de conexión con alguno de sus vecinos, el programa verifica si este vecino se encuentra o no en la posición correspondiente (líneas 229-243 y 315-329 para vehículos que se comporten como *cir2*; líneas 257-271 y 352-366 para vehículos que se comporten como *cir1*). Pueden darse dos situaciones:

- En el caso de que el vehículo verifique que su vecino se encuentra operando, el programa almacena el tiempo que ha pasado entre el actual encuentro de los vehículos y los encuentros previos que cada uno de ellos pudiera haber tenido. Esto es realizado entre las líneas 245 y 254 para vehículos que se comporten como *cir2* y entre las líneas 273-282 para vehículos que se comporten como *cir1*.
- En el caso de que el vehículo verifique que su vecino no está en funcionamiento, éste pasa a ocupar la trayectoria del UAV averiado, lo cual se programa en las líneas 331 y 349 para vehículos que se comporten como *cir2* y entre las líneas 368-386 para vehículos que se comporten como *cir1*.

De este modo, se extrae la matriz *tiempo\_cruce* en la que se recogen todos los intervalos de tiempo entre las transferencias de información entre vehículos vecinos y de la cual se obtiene el dato de  $t_{max}$  que se corresponde con el máximo de todos los términos que la componen.

Asimismo, del programa se extrae otra matriz de interés llamada *posic\_ocupadas* en la que se almacenan las trayectorias que cada uno de los vehículos ha ocupado a lo largo de la simulación.

Por otro lado, resulta de gran interés el hecho de que la simulación muestra en todo momento la posición de cada uno de los vehículos del sistema a través de un gráfica, con lo que resulta sencillo realizar un seguimiento de los ensayos. Finalmente, dado que las distintas combinaciones de vehículos caídos generan mapas de cobertura muy distintos entre sí, esta gráfica resulta de gran interés al final de la simulación ya que muestra cuáles serán las áreas que quedarían sin cubrir por los vehículos en las condiciones estudiadas.

## 5 Resultados computacionales

---

Debido a la dinámica propia de los vehículos estudiados se considera que existe mayor realismo en el caso de cambio de trayectoria con vuelo en sentidos opuestos, por lo que se analizará únicamente este caso en las simulaciones. De este modo, se presentan a continuación los resultados extraídos de la simulación para cada uno de los casos estudiados.

### 5.1 Estudio con cuatro vehículos en vuelo ( $N = 4$ )

Debido a la sencillez de la combinatoria de las posibles situaciones que podrían darse, se estudia el caso en el que no se cae ninguno de los vehículos y los casos en los que se caen uno, dos o tres de los vehículos. El tiempo tomado para la duración de la simulación ha sido de  $t = 5 \text{ ut}$  ya que es el tiempo suficiente que permite que en todas las simulaciones se llegue a completar un ciclo sin aportar demasiados datos redundantes. Así se llega a la tabla resumen 5.1, en la que los resultados se corresponden con la media de los datos obtenidos en todas las simulaciones. Esta tabla debería contrastarse con la tabla 6.1, reflejada en la subsección 6.1.1 en la que se detallan las conclusiones obtenidas.

**Tabla 5.1** Glosario de resultados obtenidos para  $N = 4$ .

Número de vehículos retirados	$t_{max}$ (ut)	$t_{max}$ (%)	$l_s$ (%)	'Starvation'
0	0.945	18.9	100	No
1	1.260	25.2	100	No
2	0.630	12.6	50	Sí
3	5.000	100	50	Sí

Se detallarán y comentarán a continuación los resultados obtenidos para cada uno de las situaciones recogidas en la tabla 5.1.

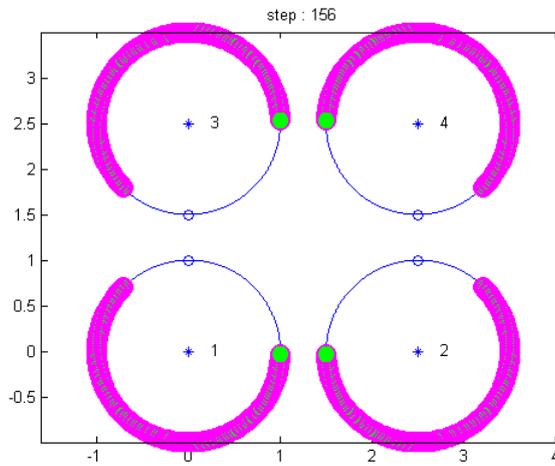
#### 5.1.1 No se produce fallo en ninguno de los vehículos

En este caso todos los vehículos mantienen sus correspondientes trayectorias ya que todos son capaces de continuar su vuelo en la zona que se les ha encomendado durante la duración de la simulación completa. De este modo, cada uno de ellos se cruzará siempre con los mismos vehículos

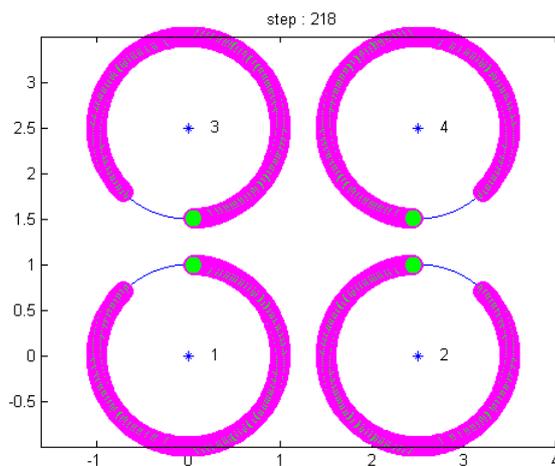
vecinos (figuras 5.1 y 5.2) y tras haber recorrido el mismo tramo de la trayectoria con lo que el tiempo obtenido es igual en cada tramo con errores en la centésima como se extrae de la tabla 5.2. Del mismo modo se concluye que el tiempo máximo que puede llegar a estar un vehículo sin comunicarse es de 0.9450 ut o lo que es lo mismo, un 18.9% de la duración total de la simulación (5 ut).

**Tabla 5.2** Intervalos de tiempo entre que se produce la comunicación de uno de los vehículos con cualquiera de sus vecinos y con el siguiente vecino que se encuentra (se mantienen en vuelo todos los vehículos).

Vehículo	tiempos de cruce (ut)							
1	0.7850	0.3100	0.9450	0.3150	0.9400	0.3150	0.9450	0.3100
2	0.7850	0.3100	0.9450	0.3150	0.9400	0.3150	0.9450	0.3100
3	0.7850	0.3100	0.9450	0.3150	0.9400	0.3150	0.9450	0.3100
4	0.7850	0.3100	0.9450	0.3150	0.9400	0.3150	0.9450	0.3100

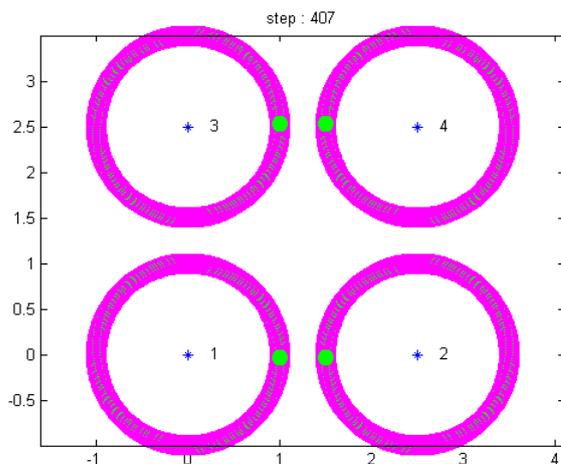


**Figura 5.1** Primer cruce de vehículos en el caso de que ninguno se haya retirado.



**Figura 5.2** Segundo cruce de vehículos en el caso de que ninguno se haya retirado.

Tal y como se extrae de la imagen 5.3 todo el espacio se mantiene cubierto por las aeronaves en este caso.



**Figura 5.3** Representación del instante final de la simulación.

### 5.1.2 Se produce el fallo de uno de los vehículos

Para este caso se han estudiado las cuatro posibilidades existentes; es decir, la situación en la que falla el vehículo 1, el 2, el 3 ó el 4 y en todas ellas se han obtenido idénticos resultados debido a la simetría del problema planteado.

En este caso los vehículos deben cambiar sus trayectorias a lo largo de la simulación ya que el hecho de que uno de ellos falle hace que el resto deba interactuar para compensar esta pérdida. Las posiciones ocupadas por cada uno de los vehículos a lo largo de las simulaciones realizadas se muestran a continuación:

**Tabla 5.3** Posiciones ocupadas por los vehículos en el caso de fallo del robot 1..

Vehículo	Posiciones		
2	1	3	4
3	4	2	-
4	2	1	-

**Tabla 5.4** Posiciones ocupadas por los vehículos en el caso de fallo del robot 2..

Vehículo	Posiciones		
1	2	4	3
3	1	2	4
4	3	1	-

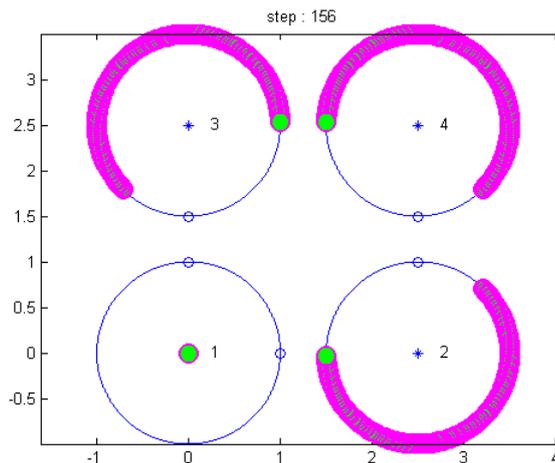
**Tabla 5.5** Posiciones ocupadas por los vehículos en el caso de fallo del robot 3..

Vehículo	Posiciones		
1	2	4	-
2	4	3	1
4	3	1	2

**Tabla 5.6** Posiciones ocupadas por los vehículos en el caso de fallo del robot 4..

Vehículo	Posiciones		
1	3	4	2
2	1	3	-
3	4	2	1

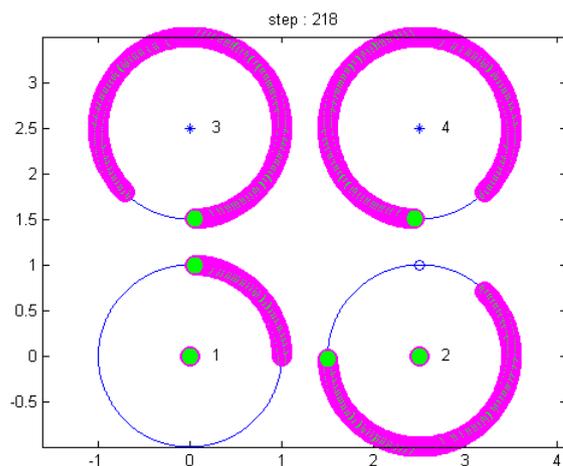
Se muestran algunos de los cruces obtenidos en el caso de que falle el vehículo 1 en las figuras 5.4 y 5.5. Puede comprobarse en la figura 5.5 como el vehículo 2 ha pasado a cubrir la trayectoria del vehículo 1 tal y como se extrae de la primera fila de la tabla 5.3.

**Figura 5.4** Primer cruce de vehículos en el caso de que uno se haya retirado.

El vehículo 2 llega al punto en el que se encuentra en la figura 5.4, donde comprueba que su vecino, el vehículo 1, se encuentra averiado por lo que pasa a ocupar su lugar tal y como se verifica en la figura 5.5.

En lo que respecta al tiempo de cruce entre dos de los vehículos, se extraen de la tabla 6.1 los resultados obtenidos para el caso de fallo del vehículo 1. Para el resto de los casos se ha obtenido una tabla análoga.

De todos ellos se obtiene que el tiempo máximo que puede llegar a estar un vehículo sin comunicarse es igual en los cuatro casos estudiados y toma el valor de 1.260 ut, es decir, un 25.2% de la duración total de la simulación (5 ut).



**Figura 5.5** Segundo cruce de vehículos en el caso de que uno se haya retirado.

**Tabla 5.7** Intervalos de tiempo entre que se produce la comunicación de uno de los vehículos con cualquiera de sus vecinos y con el siguiente vecino que se encuentra (se retira el vehículo 1).

Vehículo	tiempos de cruce (ut)					
2	1.0950	0.9450	1.2550	0.3150	1.2550	-
3	0.7850	0.3100	1.2600	0.9400	1.2600	0.3100
4	0.7850	1.2550	0.3150	1.2550	0.9450	-

Por otro lado, al igual que en el caso de la sección anterior en el que no se producía fallo alguno, en el caso de que falle uno de los vehículos, sea cual sea de todos ellos, el espacio total se mantiene cubierto por las aeronaves.

### 5.1.3 Se produce el fallo de dos de los vehículos

Para este caso se han estudiado también las seis combinaciones existentes; es decir:

- Fallan los vehículos 1 y 2.
- Fallan los vehículos 1 y 3.
- Fallan los vehículos 2 y 4.
- Fallan los vehículos 3 y 4.
- Fallan los vehículos 1 y 4.
- Fallan los vehículos 2 y 3.

En estas situaciones los vehículos en vuelo deben cambiar sus trayectorias a lo largo de la simulación ya que deben cubrir las zonas que eran sobrevoladas por los vehículos que han caído. Así, los cambios en las posiciones ocupadas por cada uno de los vehículos a lo largo de las simulaciones realizadas se muestran a continuación:

**Tabla 5.8** Posiciones ocupadas por los vehículos en el caso de fallo del 1 y el 2.

Vehículo	Posiciones				
3	1	3	1	3	
4	2	4	2	4	

**Tabla 5.9** Posiciones ocupadas por los vehículos en el caso de fallo del 3 y el 4.

Vehículo	Posiciones				
1	3	1	3	1	
2	4	2	4	2	

**Tabla 5.10** Posiciones ocupadas por los vehículos en el caso de fallo del 1 y el 3.

Vehículo	Posiciones				
2	1	2	1	2	
4	3	4	3	4	

**Tabla 5.11** Posiciones ocupadas por los vehículos en el caso de fallo del 2 y el 4.

Vehículo	Posiciones				
1	2	1	2	1	
3	4	3	4	3	

**Tabla 5.12** Posiciones ocupadas por los vehículos en el caso de fallo del 1 y el 4.

Vehículo	Posiciones									
2	1	3	4	2	1	3	4	2		
3	4	2	1	3	4	2	1	3		

**Tabla 5.13** Posiciones ocupadas por los vehículos en el caso de fallo del 2 y el 3.

Vehículo	Posiciones									
1	2	4	3	1	2	4	3	1		
4	3	1	2	4	3	1	2	4		

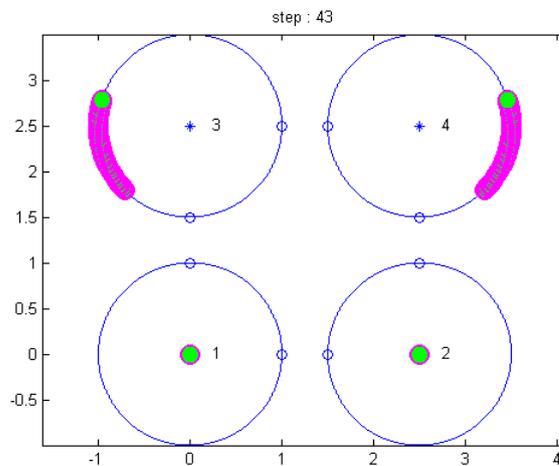
Tras la repetición de varias simulaciones considerando distintas posiciones iniciales de los vehículos, se llega a que en el caso de que sean dos los vehículos que fallen, se hace necesaria la distinción de dos casos en el estudio:

- Fallan dos vehículos pertenecientes a la misma fila o a la misma columna. Se corresponde con los casos recogidos en las tablas 5.8, 5.9, 5.10 y 5.11.
- Fallan dos vehículos pertenecientes a una fila y a una columna distintas; caso recogido en las tablas 5.12 y 5.13.

Se comenzará presentando la primera de las situaciones: Debido a la simetría del problema, se obtienen los mismos resultados en los dos casos en los que los vehículos pertenecen a la misma fila; así, tanto para el caso de fallo de los vehículos 1 y 2 como para el de fallo de 3 y 4 se obtienen los datos de intervalos de tiempo de cruce entre vehículos recogidos en la tabla 5.14. Se muestra en la figura 5.6 la situación en el caso de fallo de los vehículos 1 y 2 para orientar al lector.

**Tabla 5.14** Intervalos de tiempo entre que se produce la comunicación de uno de los vehículos con cualquiera de sus vecinos y con el siguiente vecino que se encuentra (Fallo de vehículos de la misma fila).

Vehículo averiado (falla la fila 1)	Vehículo averiado (falla la fila 2)	tiempos de cruce (ut)			
1	3	0.7850	1.2550	1.2550	1.2600
2	4	0.7850	1.2550	1.2550	1.2600



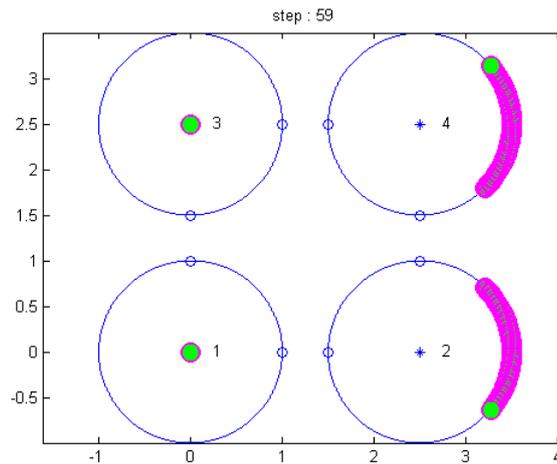
**Figura 5.6** Situación al inicio de la simulación con fallo de vehículos de la misma fila.

Análogamente, debido a la simetría del problema, se obtienen los mismos resultados en los dos casos en los que los vehículos pertenecen a la misma columna; así, tanto para el caso de fallo de los vehículos 1 y 3 como para el de fallo de 2 y 4 se obtienen los datos de intervalos de tiempo de cruce entre vehículos recogidos en la tabla 5.15. Se muestra en la figura 5.7 la situación en el caso de fallo de los vehículos 1 y 3 para orientar al lector.

**Tabla 5.15** Intervalos de tiempo entre que se produce la comunicación de uno de los vehículos con cualquiera de sus vecinos y con el siguiente vecino que se encuentra (Fallo de vehículos de la mismo columna).

Vehículo averiado (falla la columna 1)	Vehículo averiado (falla la columna 2)	tiempos de cruce (ut)			
1	2	1.0950	1.2550	1.2550	1.2550
3	4	1.0950	1.2550	1.2550	1.2550

De las cuatro combinaciones estudiadas se obtiene que el tiempo máximo que puede llegar a estar un vehículo sin comunicarse es igual en los cuatro casos estudiados y toma el valor de 1.260



**Figura 5.7** Situación al inicio de la simulación con fallo de vehículos de la misma columna.

ut, es decir, un 25.2% de la duración total de la simulación (5 ut).

Por otro lado, al igual que en los casos anteriores, en estos casos el espacio total se mantiene cubierto por las aeronaves ya que cada uno de los dos vehículos en funcionamiento se mantiene recorriendo de forma periódica su propia trayectoria inicial y la de su vecino averiado.

Para la segunda de las situaciones en la que fallan dos vehículos pertenecientes distinta fila y distinta columna, se obtienen resultados totalmente distintos que se recogen en lo que sigue:

Se produce en cualquiera de los dos casos el mencionado fenómeno de '*Starvation*' y se corresponde con la falta de información total por parte de los vehículos vecinos. Por ello, el parámetro relativo al tiempo de cruce entre dos de los vehículos carece de sentido en estos casos puesto que ninguno de los dos vehículos operativos se cruza con el otro a lo largo de la simulación. El tiempo máximo que puede llegar a estar un vehículo sin comunicarse toma por tanto el valor de 5 ut, es decir, un 100% de la duración total de la simulación (5 ut).

En los presentes casos, el espacio cubierto por las dos aeronaves en vuelo se corresponde con una cobertura que se reduce al 50%. El espacio recorrido sobre las trayectorias definidas puede ser como el que se muestra en la imagen 5.10, o bien como en la imagen 5.11 según cuál sea el par de vehículos caídos y la posición inicial tomada.

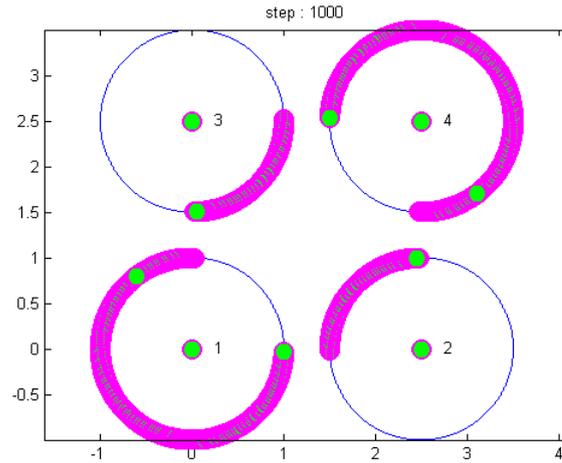
Cabe destacar que en este caso los vehículos en vuelo no mantienen comunicación con ninguno de los demás integrantes del conjunto con lo que dejaría de considerarse como un sistema sincronizado.

#### 5.1.4 Se produce el fallo de tres de los vehículos

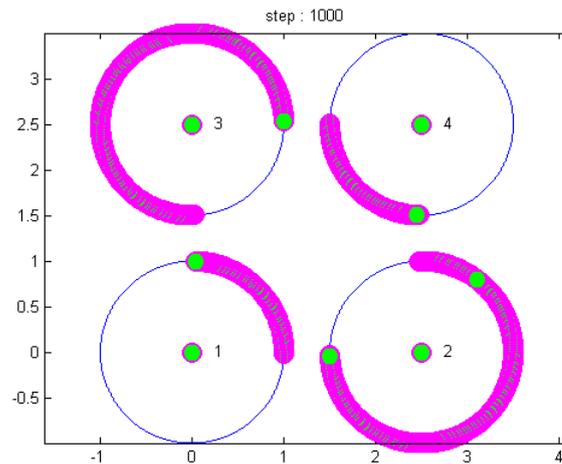
Para este caso se han estudiado también las cuatro posibilidades existentes, es decir:

- Fallan los vehículos 2, 3 y 4; vuela el vehículo 1.
- Fallan los vehículos 1, 3 y 4; vuela el vehículo 2.
- Fallan los vehículos 1, 2 y 4; vuela el vehículo 3.
- Fallan los vehículos 1, 2 y 3; vuela el vehículo 4.

En este caso el vehículo que mantiene el vuelo no coincidirá en ningún momento con los demás



**Figura 5.8** Representación del instante final de la simulación si mantienen el vuelo los vehículos 1 y 4.



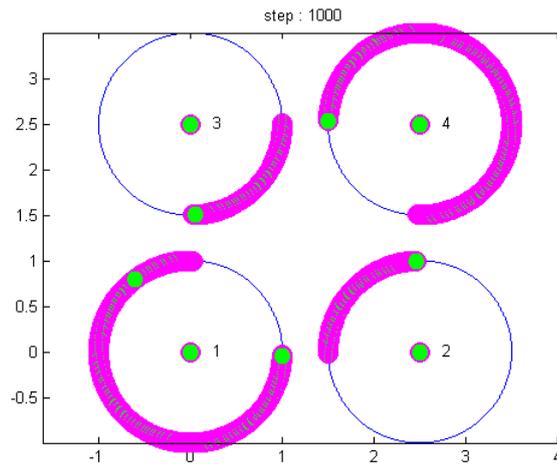
**Figura 5.9** Representación del instante final de la simulación si mantienen el vuelo los vehículos 2 y 3.

debiendo cambiar su trayectoria a lo largo de la simulación cada vez que pase por cualquiera de los puntos en los que debería encontrarse con alguno de sus vecinos. Es por ello que el estudio del tiempo de cruce entre dos de los vehículos carece de sentido en estos casos.

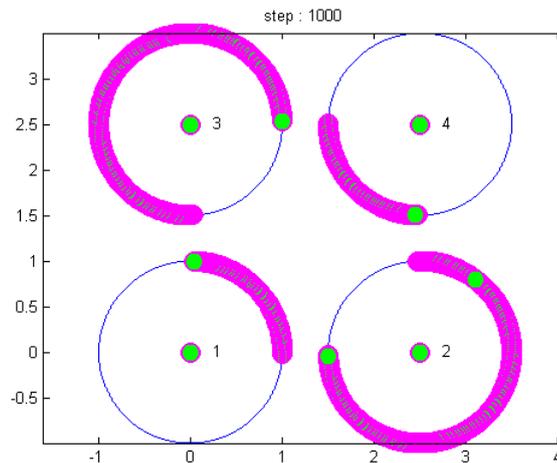
Se da en cualquiera de los casos el fenómeno de '*starvation*' ya comentado en la sección 2. Debido a que el vehículo operativo nunca se cruza con ningún otro, el tiempo máximo que puede llegar a estar un vehículo sin comunicarse es igual en los cuatro casos estudiados y toma el valor de 5 ut, es decir, un 100% de la duración total de la simulación (5 ut).

Cabe destacar que debido a que en estos casos el vehículo en vuelo no mantiene comunicación con ninguno de los demás robots, el conjunto dejaría de considerarse como un sistema sincronizado.

Sin embargo puede estudiarse el espacio cubierto por el robot que continúa el vuelo que, en todos los casos, se corresponde con que la cobertura se reduce al 50%. Si son los vehículos 1 ó 4 los que mantienen el vuelo, el espacio cubierto será el que se muestra en la imagen 5.10, mientras que si los que mantienen el vuelo son los vehículos 2 ó 3, el espacio cubierto quedará como en la imagen 5.11.



**Figura 5.10** Representación del instante final de la simulación si mantienen el vuelo los vehículos 1 ó 4.



**Figura 5.11** Representación del instante final de la simulación si mantienen el vuelo los vehículos 2 ó 3.

## 5.2 Estudio con nueve vehículos en vuelo ( $N = 9$ )

Una vez que se consideran nueve vehículos en vuelo, la combinatoria existente para el problema se hace excesivamente compleja lo que hace tedioso analizar cada uno de los casos que pudieran darse; es por ello que esta situación se estudiará con otro enfoque.

Tras realizar una serie de ensayos con el programa de simulación, se llega a la conclusión de que el sistema se ve afectado únicamente en el caso de que fallen vehículos con una determinada combinación en sus posiciones; lo cual se detallará en la sección 6.1.2 de análisis de resultados. Esto lleva a plantear el problema del siguiente modo: Se comprobará la tolerancia a fallo del sistema realizando un análisis estadístico de los datos extraídos tras realizar simulaciones con distintas configuraciones de vehículos caídos obtenidas aleatoriamente de un programa diseñado en Matlab.

En primer lugar, se muestran los casos extraídos de Matlab que han sido simulados. Como puede verse, del programa se obtiene tanto el número de vehículos que dejarán de funcionar como las

posiciones que ocupan en el conjunto los vehículos que mantienen su vuelo.

Cabe destacar que se han eliminado los casos en los que resultaba que el número de vehículos que fallan en la simulación es igual a nueve ya que el estudio carece de sentido en esta situación.

Número de vehículos que fallan en la simulación	Vehículos que continúan operando
3	[2 3 4 5 7 9]
2	[1 2 3 5 6 8 9]
0	[1 2 3 4 5 6 7 8 9]
4	[1 5 6 7 9]
6	[1 2 4]
1	[1 2 4 5 6 7 8 9]
5	[1 2 7 9]
6	[2 6 9]
2	[1 4 5 6 7 8 9]
2	[1 2 4 5 6 7 9]
4	[2 5 6 7 8]
8	4
3	[1 2 3 4 7 9]
3	[1 2 3 5 6 8]
5	[5 7 8 9]
0	[1 2 3 4 5 6 7 8 9]
8	9
4	[2 4 5 7 9]
8	8
6	[2 5 7]
4	[1 4 7 8 9]
7	[3 6]
5	[4 6 7 8]
4	[1 3 4 5 7]
1	[1 2 3 4 5 7 8 9]
8	9
7	[2 8]
1	[1 2 3 5 6 7 8 9]
6	[1 2 9]
0	[1 2 3 4 5 6 7 8 9]
8	9
4	[4 5 7 8 9]
1	[1 2 4 5 6 7 8 9]
7	[4 9]
5	[2 5 6 9]
6	[1 3 8]
2	[1 3 5 6 7 8 9]
1	[1 2 3 4 5 6 8 9]
7	[2 6]
5	[2 4 7 9]
7	[3 5]
1	[1 3 4 5 6 7 8 9]

Continúa en la página siguiente.

Número de vehículos que fallan en la simulación	Vehículos que continúan operando
3	[1 2 4 6 7 9]
1	[1 2 3 4 5 6 7 9]
2	[1 2 3 4 6 7 8]
3	[1 5 6 7 8 9]
2	[1 2 3 4 6 7 9]

**Tabla 5.16** Situaciones a estudiar para el caso de  $N = 9$ .

El tiempo tomado para la duración de la simulación ha sido de  $t = 8 ut$  ya que es el tiempo suficiente que permite que en la mayoría de las simulaciones se llegue a completar un ciclo sin aportar demasiados datos redundantes.

Así se llega a la tabla 5.17, siendo la variable  $\frac{t_{max}}{t_{ref}}$  representa la relación entre el  $t_{max} (ut)$  y  $t_{ref} (ut)$ , siendo éste último el tiempo que emplearía cualquiera de los vehículos en completar una vuelta sobre una misma trayectoria en las condiciones estudiadas,  $t_{ref} = 1.248 ut$ .

Número de vehículos retirados en la simulación	Posición de los vehículos retirados en la simulación	$t_{max} (ut)$	$t_{max} (\%)$	$\frac{t_{max}}{t_{ref}}$	$l_s (\%)$	'starvation'
0	-	0,944	11,800	0,756	100,000	No
0	-	0,944	11,800	0,756	100,000	No
0	-	0,944	11,800	0,756	100,000	No
1	[7]	0,944	11,800	0,756	100,000	No
1	[2]	0,944	11,800	0,756	100,000	No
1	[6]	0,944	11,800	0,756	100,000	No
1	[8]	1,256	15,700	1,006	100,000	No
1	[3]	1,264	15,800	1,013	100,000	No
1	[4]	1,264	15,800	1,013	100,000	No
1	[3]	1,264	15,800	1,013	100,000	No
2	[5 8]	1,256	15,700	1,006	100,000	No
2	[4 7]	1,260	15,750	1,010	100,000	No
2	[2 3]	1,264	15,800	1,013	100,000	No
2	[5 9]	1,576	19,700	1,263	100,000	No
2	[3 8]	1,576	19,700	1,263	100,000	No
2	[2 4]	1,576	19,700	1,263	100,000	No
3	[5 6 8]	1,576	19,700	1,263	100,000	No
3	[4 7 9]	1,576	19,700	1,263	100,000	No
3	[1 6 8]	1,885	23,563	1,510	100,000	No
3	[3 5 8]	1,888	23,600	1,513	100,000	No
3	[2 3 4]	1,888	23,600	1,513	100,000	No
4	[2 3 4 8]	4,712	58,900	3,776	66,667	Sí
4	[1 2 3 6]	1,256	15,700	1,006	100,000	No
4	[2 3 5 6]	1,264	15,800	1,013	100,000	No
4	[1 3 4 9]	1,888	23,600	1,513	100,000	No
4	[2 6 8 9]	1,888	23,600	1,513	100,000	No
4	[1 3 6 8]	2,200	27,500	1,763	100,000	No
5	[1 2 3 5 9]	1,888	23,600	1,513	66,667	Sí

Continúa en la página siguiente.

Número de vehículos retirados en la simulación	Posición de los vehículos retirados en la simulación	$t_{max}$ (ut)	$t_{max}$ (%)	$\frac{t_{max}}{t_{ref}}$	$l_s$ (%)	'starvation'
5	[3 4 5 6 8]	4,400	55,000	3,526	66,667	Sí
5	[1 3 4 7 8]	4,400	55,000	3,526	66,667	Sí
5	[1 2 3 4 6]	1,888	23,600	1,513	100,000	No
5	[1 3 5 6 8]	2,200	27,500	1,763	100,000	No
6	[2 4 5 6 7 9]	1,888	23,600	1,513	66,667	Sí
6	[3 4 5 6 7 8]	3,768	47,100	3,019	66,667	Sí
6	[1 3 4 5 7 8]	3,776	47,200	3,026	66,667	Sí
6	[1 3 4 6 8 9]	3,776	47,200	3,026	66,667	Sí
6	[3 5 6 7 8 9]	3,456	43,200	2,769	100,000	No
7	[1 3 4 5 7 8 9]	0,000	0,000	0,000	33,333	Sí
7	[1 2 4 6 7 8 9]	0,000	0,000	0,000	66,667	Sí
7	[1 2 3 5 6 7 8]	0,000	0,000	0,000	66,667	Sí
7	[1 2 4 5 7 8 9]	1,888	23,600	1,513	66,667	No
7	[1 3 4 5 6 7 9]	1,888	23,600	1,513	66,667	No
8	[1 2 3 5 6 7 8 9]	0,000	0,000	0,000	33,333	Sí
8	[1 2 3 4 5 6 7 8]	0,000	0,000	0,000	33,333	Sí
8	[1 2 3 4 5 6 7 8]	0,000	0,000	0,000	33,333	Sí
8	[1 2 3 4 5 6 7 8]	0,000	0,000	0,000	33,333	Sí
8	[1 2 3 4 5 6 7 9]	0,000	0,000	0,000	33,333	Sí

**Tabla 5.17** Resultados de las situaciones estudiadas con  $N = 9$ .

Los datos recogidos en el cuadro 5.17 y las conclusiones a las que se llegan a partir de ella se comentarán en la sección 6.1.2.

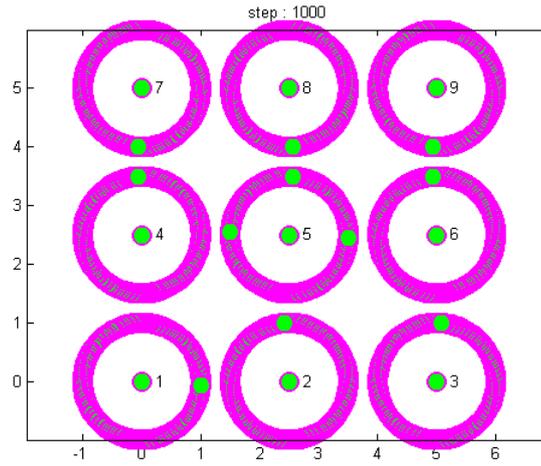
Por otro lado, al igual que para el caso de cuatro vehículos, en estas simulaciones se han obtenido las representaciones de las trayectorias seguidas por cada uno de los UAVs durante las mismas con lo que se dispone de un mapa esquemático en el que se representa la cobertura que aporta el sistema en cada caso. Cabe incluir en la presente sección las representaciones obtenidas para cada uno de los porcentajes que aparecen en la tabla 5.17.

Así, para el caso de que la cobertura sea del 100% se obtiene una representación como la que se muestra en la figura 5.12; para el caso del 66,667% se han obtenido tres variantes según cuáles sean los vehículos caídos, que se presentan en las figuras 5.13, 5.14 y 5.15. Finalmente para el caso de que la cobertura sea del 33,333% se obtienen las configuraciones presentadas en las figuras 5.16 y 5.17. En cualquiera de los casos se podrían obtener resultados simétricos si fallaran los vehículos apropiados pero manteniendo las geometrías aquí expuestas.

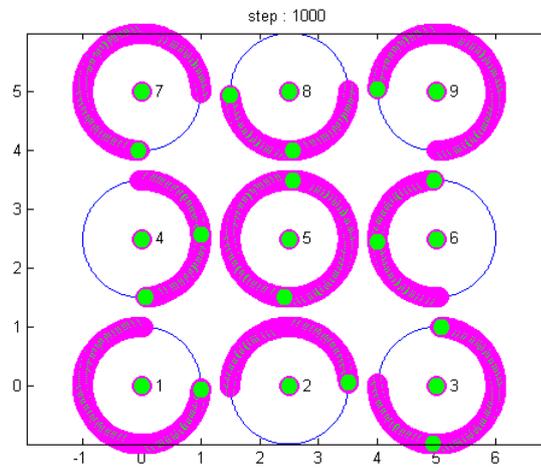
Se han obtenido asimismo para cada una de las simulaciones dos conjuntos de variables más que completan el estudio:

- Los intervalos de tiempo que pasan hasta que cada uno de los vehículos se encuentra con alguno de sus vecinos. No se incluye este estudio entre los datos recogidos ya que se considera que de todos ellos, el de mayor interés es el tiempo máximo, incluido en la tabla 5.17.
- Las trayectorias que va ocupando cada uno de los vehículos a lo largo de la simulación.

El análisis de estos dos conjuntos de variables resulta de gran interés a la hora de contrastar si los vehículos se han encontrado la mayor parte del tiempo en 'Starvation'. Así, si un determinado vehículo ha cambiado en muchas ocasiones de trayectoria es porque no se ha encontrado vecinos



**Figura 5.12** Cobertura del 100%.



**Figura 5.13** Cobertura del 66,667%; configuración 1.

en los cruces en los que los esperaba con lo que se concluye que se debe comprobar que se ha encontrado incomunicado durante más tiempo que otro vehículo que no haya cambiado tanto de trayectoria. Este hecho se detallará con algún ejemplo en la sección 6.1.2.

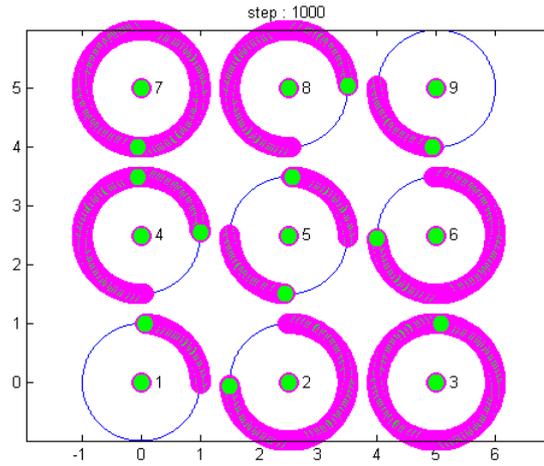


Figura 5.14 Cobertura del 66,667%; configuración 2.

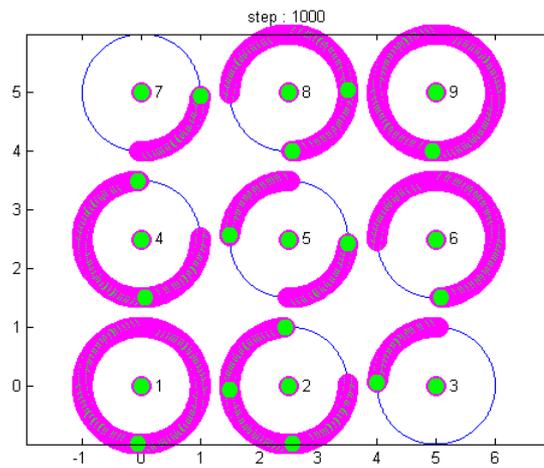


Figura 5.15 Cobertura del 66,667%; configuración 3.

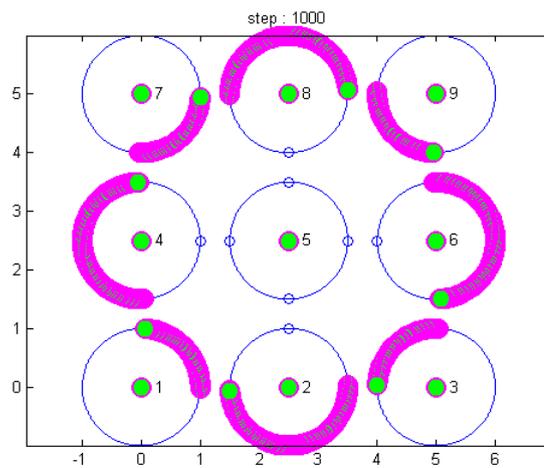


Figura 5.16 Cobertura del 33,333%; configuración 1.

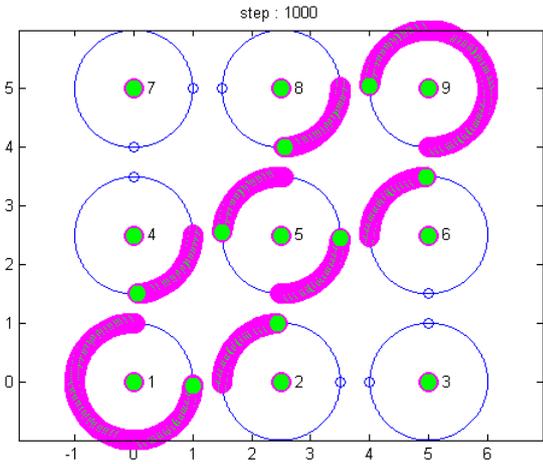


Figura 5.17 Cobertura del 33,333%; configuración 2.

# 6 Análisis de resultados y propuestas de mejora

---

Se detallarán a continuación los aspectos que se han extraído de las simulaciones realizadas así como propuestas de mejora en la línea de estudio para una futura ampliación del trabajo realizado.

## 6.1 Conclusiones

### 6.1.1 Estudio con 4 vehículos

En el caso de que se consideren 4 vehículos para el estudio se extrae que se produce 'Starvation' naturalmente si fallan tres de los vehículos, ya que sólo se mantendrá uno en vuelo y en algunas configuraciones resultantes si fallan 2 de los vehículos (se recuerda que esta configuración consiste en que los vehículos caídos pertenezcan a una fila y a una columna distintas). Esta conclusión hace que resulte bastante improbable que los vehículos vuelen incomunicados si empleamos cuatro, lo cual lleva a considerar que el sistema tratado es un sistema estable que permite evitar el fenómeno de la 'Starvation'.

Se muestra en la tabla 6.1 un glosario de las posibles situaciones relevantes que pueden darse para el caso de 4 vehículos en vuelo; se ha eliminado el estudio en el caso de fallo de tres de los vehículos por su trivialidad. Por otro lado, cabe recordar que el llamado  $t_{ref}$  es el que tarda un robot que dar una vuelta completa a una de las trayectorias y en este caso es igual a 1.26 segundos.

**Tabla 6.1** Tabla resumen de la situación con  $N = 4$ .

Número de vehículos retirados en la simulación	Posición de los vehículos retirados en la simulación	$t_{max}$ (ut)	$t_{max}$ (%)	$\frac{t_{max}}{t_{ref}}$	$l_s$ (%)	'starvation'
0	-	0,945	18,9	0,75	100,000	No
1	1, 2, 3 ó 4	1,26	25,5	1	100,000	No
2	Ambos de la misma fila ó misma columna	1,26	25,5	1	100,000	No
2	Pertenecen a distinta fila y columna	-	-	1	50,000	Sí

Se confirma a partir de la tabla anterior que el sistema es tolerante a fallo en lo relativo al concepto de 'Starvation' pues únicamente se da con una combinación muy particular de vehículos caídos.

En lo que respecta al tiempo que transcurre entre los intercambios de información, éste se ve más afectado si se compara con el llamado  $t_{ref}$  pues los vehículos llegan a encontrarse incomunicados durante el tiempo que tardarían en completar una vuelta en cualquiera de las trayectorias.

### 6.1.2 Estudio con 9 vehículos

A la vista de los datos recogidos en la tabla 5.17 puede argumentarse que el sistema estudiado se trata de un sistema estable en lo que se refiere a las variables analizadas. Se comprueba que el sistema sólo comienza a verse mermado a partir de que fallen 4 de los 9 vehículos. Además, en este caso de que fallen cuatro vehículos, el sistema se ve realmente afectado únicamente si los 4 vehículos que dejan de operar están dispuestos de una determinada manera que a la vista de la simulación no ocurre con frecuencia.

Por otro lado, en el caso de fallo de cinco de los vehículos, el sistema se ve afectado ante más configuraciones posibles de vehículos caídos. Cuando se caen los vehículos que llevan a las configuraciones que hacen que se deteriore el comportamiento del sistema, éste se ve afectado sobre todo porque los UAVs se encuentran incomunicados durante tiempo situados en torno al 23,600% y el 55,000% de la simulación pero se sigue manteniendo en cualquier caso la cobertura en un rango mayor o igual que el 66,667%.

Del mismo modo, en el caso de que sean seis los vehículos caídos, el sistema mantiene la cobertura en todo caso por encima del 66,667% y el tiempo máximo que se ha podido llegar a encontrar un vehículo incomunicado durante la simulación se sitúa entre el 23,600% y el 47,100%. Siguen dándose en este caso configuraciones que permiten que se cubra todo el espacio pero el sistema se ve afectado en el tiempo que tardan los vehículos en verse.

A partir ya de siete vehículos caídos la cobertura se mantiene entre un 33,333% y un 66,667% pero comienzan a aparecer casos de 'Starvation', lo cual significa que los dos vehículos que se mantienen en operación no se cruzan en ningún momento durante la simulación.

Finalmente, como es natural, se da la misma situación en el caso de que sólo sea un vehículo el que se mantiene en funcionamiento aunque la cobertura se reduce a un 33,333%.

Al inicio de esta sección se ha indicado que se trata de un sistema bastante estable y esto se justifica ya que de todos los casos estudiados, únicamente en el 34% de ellos se llega a dar la situación de que no exista comunicación entre los vehículos que se mantienen en operación.

En relación al análisis de los dos conjuntos de variables extraídos de la simulación (intervalos de tiempo entre cada cruce con dos vehículos vecinos y trayectorias ocupadas por los vehículos en funcionamiento) cabe destacar que el estudio de ambas constituye una herramienta fiable con la que determinar si los vehículos realizan muchos intercambios de información o no.

Si la tabla de intervalos de tiempo de cruce posee muchas columnas, entonces la tabla de posiciones ocupadas poseerá pocas y se interpreta como que los vehículos están viendo con frecuencia a sus vecinos. Por el contrario, si sucede a la inversa, los vehículos no localizan a sus vecinos y se ven, por tanto, obligados a cambiar de trayectoria con frecuencia.

Se mostrarán a continuación algunos ejemplos extremos con su correspondiente interpretación para resaltar la diferencia.

Para el caso en el que se mantienen en vuelo los vehículos 2 y 8 se obtienen los resultados contenidos en las tablas 6.2 y 6.3; puede comprobarse como los vehículos únicamente se cruzan cuatro veces a lo largo de la simulación mientras que han sobrevolado 13 trayectorias (incluyendo repeticiones de trayectorias) distintas a la suya describiendo la figura que se muestra en la imagen

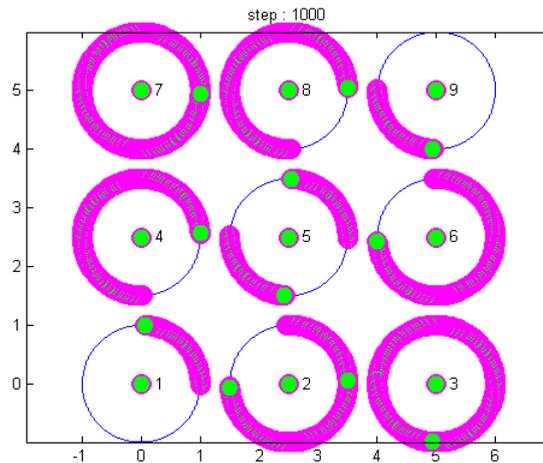
6.1. Los vehículos, al no cruzarse con ninguno de sus vecinos, cambian de trayectoria con el fin de completar la tarea de los vehículos caídos y compensar así la pérdida.

**Tabla 6.2** Intervalos de tiempo entre que se produce la comunicación de uno de los vehículos con cualquiera de sus vecinos y con el siguiente vecino que se encuentra (vuelan 2 y 8).

Vehículo	tiempos de cruce (ut)			
2	1.0960	1.8880	1.8800	1.8880
8	1.0960	1.8880	1.8880	1.8880

**Tabla 6.3** Posiciones ocupadas por los vehículos (vuelan 2 y 8).

Vehículo	Posiciones												
2	3	2	1	4	5	2	3	2	1	4	5	2	3
8	9	6	5	8	7	8	9	6	5	8	7	8	9



**Figura 6.1** Cobertura en el vuelo de los vehículos 2 y 8.

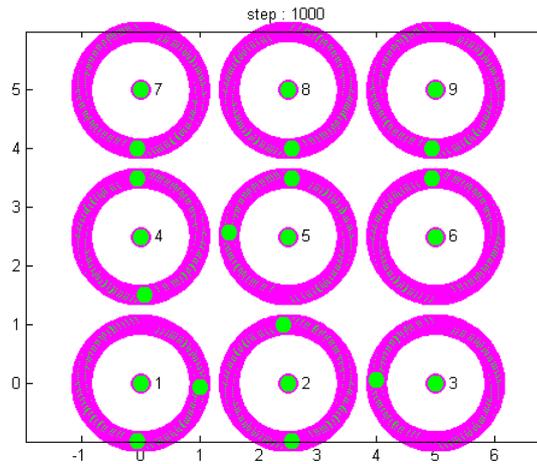
Se muestra a continuación el ejemplo opuesto en el que los vehículos están en contacto en un gran número de ocasiones, lo que hace que apenas se vean en la situación de tener que cambiar de trayectoria. Es el caso, por ejemplo, en el que dejan de operar los vehículos 2 y 4. A partir de los resultados contenidos en las tablas 6.4 y 6.5 puede comprobarse en la primera de las tablas como los vehículos se cruzan numerosas veces a lo largo de la simulación mientras que sobrevuelan 6 trayectorias distintas a la suya como máximo. Se muestra en la imagen 6.2 como se cubre todo el espacio en la simulación.

**Tabla 6.4** Intervalos de tiempo entre que se produce la comunicación de uno de los vehículos con cualquiera de sus vecinos y con el siguiente vecino que se encuentra (vuelan 1, 3, 5, 6, 8 y 9).

Vehículo	tiempos de cruce (ut)												
1	0.7840	0.3120	0.9440	0.3120	1.5680	0.3200	0.9360	1.2640	0.3120	0.9440	-	-	-
3	0.7840	0.3120	0.3120	0.6320	0.3120	1.2560	0.9440	0.3120	0.3120	0.6320	0.3120	1.2560	-
5	0.1520	0.3120	0.9440	1.2560	0.3200	0.9360	0.3200	1.5680	0.3120	0.9440	-	-	-
6	0.1520	0.3120	0.9440	1.2560	0.3200	0.6240	0.3120	0.3200	0.9360	1.2640	0.3120	0.6240	0.3200
7	0.7840	0.3120	0.6240	0.3200	0.3120	0.9440	1.2560	0.3120	0.6320	0.3120	0.3120	0.9440	-
8	0.1520	0.3120	1.2560	0.9440	0.3200	0.3120	0.6240	0.3200	1.2560	0.9440	0.3120	0.3120	0.6320
9	0.1520	0.3120	1.5760	0.3120	0.9440	1.2560	0.3120	0.9440	0.3120	1.5760	-	-	-

**Tabla 6.5** Posiciones ocupadas por los vehículos (vuelan 1, 3, 5, 6, 8 y 9).

Vehículo	Posiciones					
1	2	5	6	9	-	-
3	2	3	2	3	2	-
5	6	9	8	5	4	1
6	3	6	3	6	-	-
7	4	7	4	7	-	-
8	7	8	7	8	-	-
9	8	5	4	1	2	5



**Figura 6.2** Cobertura en el vuelo de los vehículos 1, 3, 5, 6, 8 y 9.

## 6.2 Generalizaciones del modelo

En el presente proyecto se ha tomado como objetivo la obtención de resultados computacionales consistentes y una simulación que proporcione resultados fiables basando el estudio en un modelo simplificado. Para extender el estudio a modelos más realistas, cabe plantearse posibles modificaciones y/o incorporaciones a las simulaciones realizadas.

### 6.2.1 Adición de variables de interés en el estudio

En el presente estudio se ha detallado que las variables a extraer de las simulaciones realizadas serían: el tiempo máximo que puede llegar a encontrarse incomunicado un vehículo durante la simulación y la longitud máxima sobre las trayectorias que puede llegar a quedarse sin sobrevolar durante la simulación. Asimismo, se ha manifestado que en trabajos relacionados se ha estudiado la tolerancia a fallo de un sistema similar con respecto al número de veces que se lleva a cabo la transferencia de información entre los vehículos. Se propone, por tanto, la ampliación de este estudio con la incorporación de nuevas variables de interés a considerar como por ejemplo el alcance máximo que puede recorrer una información, esto es, la profundidad de emisión de un determinado mensaje a lo largo del sistema.

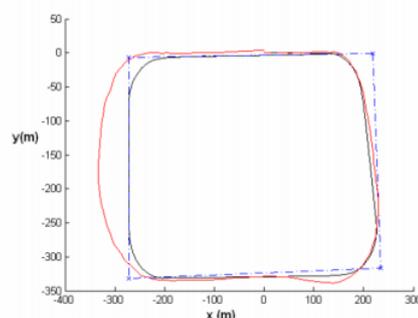
### 6.2.2 Incorporación de una ley que simule el efecto del viento

El viento es una variable que afecta en gran medida a todo vehículo aéreo en el desarrollo de su movimiento. Se generarán fuerzas sobre la aeronave que ésta deberá compensar con el fin de mantener la trayectoria que se le ha marcado. Este efecto lleva a que el movimiento de la aeronave no pueda ser considerado como uniforme ya que sufrirá ciertas aceleraciones dadas por los impulsos generados por el propio viento, que hacen que la hipótesis de vuelo circular con velocidad angular constante deba revisarse.

Los UAVs son vehículos que, en general, destacan por su ligereza. Se trata de vehículos difíciles de dirigir cuando hay viento, lo cual se agrava si la altitud de vuelo a la que debe mantenerse es elevada. De este modo se considera que este hecho realmente alterará de forma global a la sincronización de los vehículos estudiados ya que retrasará o adelanta la llegada de éstos a los puntos de encuentro definidos. Es por ello que, una vez considerado el viento, debería incorporarse a la simulación la modificación de que los vehículos adaptaran su velocidad de vuelo en función de las condiciones aportadas por el *sistema viento*.

Un estudio que muestra las modificaciones que llega a generar el viento en las trayectorias seguidas por un UAV se recoge en [21], trabajo que trata la planificación de trayectorias de estos vehículos.

Concretamente, en la sección 4.1.2. del proyecto citado en [21] se analiza el movimiento de un UAV que debe completar una trayectoria basada en el alcance de una serie de referencias que vienen dadas por el subsistema de seguimiento de *waypoints* (trayectoria azul en la figura 6.3). Todo ello sin contar a priori con el efecto del viento, que no se incorpora en las ecuaciones que modelan el estudio pero sí que se manifiesta en la simulación real tal y como se muestra en la figura 6.3.



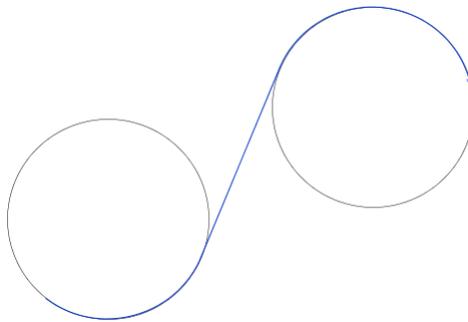
**Figura 6.3** Comparación entre la trayectoria real (rojo) y la trayectoria del modelo (negro).

### 6.2.3 Modelado de la maniobra para cambio de trayectoria

Los cambios de trayectoria se han modelado como instantáneos en el presente estudio, es decir, el vehículo que debe cambiar de trayectoria lo hace con la hipótesis de que ambas trayectorias poseen un punto común que sirve de nexo entre ambas, lo que permite que el UAV no deba incorporar aceleraciones a su movimiento. Sin embargo, esto no sucede de este modo en una maniobra real ya que el vehículo deberá adaptar su velocidad para completar los giros necesarios para alcanzar el nuevo patrón además de recorrer la distancia que separa ambas trayectorias, lo cual afecta directamente a la sincronización estudiada.

Es por ello que se debe incorporar tanto un modelo completo de la maniobra de cambio de trayectoria que se adapte a las condiciones de vuelo requeridas por el vehículo estudiado como un modelo que permita la adecuación de la velocidad de vuelo de cada uno de los componentes del sistema a la situación del mismo con el fin de garantizar la sincronización del sistema.

Se propone en la figura 6.4 un cambio de trayectoria basado en el seguimiento de las rectas tangentes entre ambas trayectorias.



**Figura 6.4** Propuesta para el modelo de cambio de trayectoria.

### 6.2.4 Nuevas estrategias en el cambio de trayectorias

La estrategia tomada para el análisis realizado en el presente proyecto ha sido que los vehículos cambien de trayectoria de forma instantánea. Conviene considerar en próximos estudios algunas alternativas de cambio de trayectoria distintas con el fin de comprobar si son favorables o no para el mantenimiento de la cobertura del sistema.

Una propuesta a estudiar podría ser la de permitir que el vehículo que detecta la existencia de fallo pudiera permanecer en su trayectoria o bien cambiar a la de su vecino mediante una variable de decisión aleatoria.

### 6.2.5 Consideración de velocidad angular variable a lo largo del vuelo y de trayectorias distintas

Tal y como se ha hecho ver en los razonamientos previos, recogidos en 6.2.2 y 6.2.3, era inmediata la propuesta de incorporar al problema la variación de velocidad de vuelo a lo largo del tiempo. Ya se ha reflexionado en las secciones mencionadas que para mantener la correcta sincronización será necesario adaptar la velocidad de los vehículos si el viento incide sobre éstos o si el cambio de trayectoria se modela de una forma más próxima a la realidad.

Por otro lado, también se hace necesaria la incorporación de tal variable en el caso de que las

trayectorias seguidas por los vehículos sean distintas entre sí. Esto se debe a que, en ese caso, habría vehículos que deberían recorrer distancias más largas que otros para llegar al punto de transferencia de información; no consiguiéndose por tanto la sincronización con velocidad angular constante. Los vehículos más alejados deberán aumentar su velocidad de vuelo mientras que los más próximos tendrían que disminuirla, siempre respetando los márgenes admisibles para que el vuelo se desarrolle en condiciones apropiadas.

#### **6.2.6 Incorporación de las variables propias del movimiento de los vehículos aéreos**

En el estudio actual se han modelado los vehículos como puntos en el espacio ya que el análisis se centraba en el resultado global de la tolerancia a fallo del sistema ante avería de cualquiera de ellos. Sin embargo, será necesaria la incorporación de los parámetros propios del movimiento de los vehículos aéreos en un estudio más avanzado y en profundidad de la situación.

El vuelo de estos vehículos debe modelarse en tal caso teniendo en cuenta las variables propias de la mecánica de vuelo de un UAV y las restricciones que aparecen cuando se trabaja con ellos, tales como velocidades mínimas de operación, tiempo de respuesta, maniobrabilidad, entre otros parámetros.



# Apéndice A

## Código Matlab

---

```
1 function [tmax,veh_tiempo_cruce]= sentido_contrario_sincronizado_concaida (N,w,r)
2 close all ;
3 % prelocated variables
4 dxc=zeros(1,n);
5 dyc=zeros(1,n);
6 xmat=zeros(n,length( theta ));
7 ymat=zeros(n,length( theta ));
8 A=cell(k);
9 B=cell(k);
10 C=cell(k);
11 thetas = cell ( k );
12 centros =zeros(n,2) ;
13
14 R=1;
15 theta =(0:0.0001:2*pi) ;
16
17 % numero de filas de la malla
18 k=0;
19 cont=2;
20 while k^2~=n
21     k=n/cont;
22     cont=cont+1;
23 end
24
25 % preparacion del entorno
26 % dibujo trayectorias
27 fila =1;
28 while fila <=k;
29     for ii =1:k
30         %definicion siguiente centro
31         daux=2*R+r;
32         xaux=daux*(ii-1);
33         yaux=daux*(fila-1);
34         plot (xaux,yaux, '*')
35         hold on
36         dxc(1, ii +k*( fila -1))=xaux;
37         dyc(1, ii +k*( fila -1))=yaux;
38         centros ( ii +k*( fila -1),:)=[dxc(1, ii +k*( fila -1)) dyc(1, ii +k*( fila -1))];
39
40         %nueva trayectoria
41         x=dxc(1, ii +k*( fila -1))+R*cos(theta);
```

```

43     y=dyc(1, ii +k*( fila -1))+R*sin(theta);
44
45     %almacenamiento puntos trayectorias
46     xmat(ii +k*( fila -1),:)=x;
47     ymat(ii +k*( fila -1),:)=y;
48
49     figure (1)
50     axis equal
51     plot(x,y)
52     hold on
53     str = int2str ( ii +k*( fila -1));
54     text (x(length(x)/2)+5*R/4,y(length(y)/2) , str );
55     end
56     fila = fila +1;
57 end
58
59 % creacion celdas auxiliares
60 for ll =1:k-1
61     A{ll, ll+1}=0;
62     A{ll+1, ll }=pi;
63 end
64
65 for ll =1:k
66     B{ll, ll }=pi/2;
67     C{ll, ll }=3*pi/2;
68 end
69
70 for ll =1:k
71     thetas { ll , ll }=A;
72 end
73 for ll =1:k-1
74     thetas { ll , ll+1}=B;
75     thetas { ll+1, ll }=C;
76 end
77
78 cont=0;
79 vect =[];
80 vectgrad=cell (n,2);
81 thetanumvect_a=cell (n,1);
82 thetanumvect_h=cell (n,1);
83 thetaveh_a=cell (n,1);
84 thetaveh_h=cell (n,1);
85
86 %betas
87 for tt =1:k
88     vect=[vect , ones (1,k)*tt ];
89 end
90
91 for kk=vect
92     cont=cont+1;
93     num=cont-k*(kk-1);
94     thetanum=[];
95     for ii =1:k
96         if (isempty(thetas {kk, ii })==0)
97             for pp=1:k
98                 if isempty(thetas {kk, ii }{num,pp})==0
99                     thetanum=[thetanum, thetas {kk, ii }{num,pp}];
100             end
101         end
102     end
103 end

```

```

105 thetanum2=thetanum+pi-2*pi*floor((thetanum+pi)/2/pi); %puntos de coincidencia  circunferencias
106 thetanum2(find (thetanum2==0))=2*pi;
107 thetanumvect_a { cont,1 }=thetanum2; %con 360
108 thetanumvect_h { cont,1 }=thetanum+pi-2*pi*floor((thetanum+pi)/2/pi); %con 0
109
110 thetanum3=thetanum-2*pi*floor(thetanum/2/pi);
111 thetanum3(find (thetanum3==0))=2*pi;
112 thetaveh_a { cont,1 }=thetanum3;
113 thetaveh_h { cont,1 }=thetanum-2*pi*floor(thetanum/2/pi);
114
115 vectgrad { cont,1 }=180* thetanumvect_a { cont }/pi; %vector col
116 vectgrad { cont,2 }=180* thetanumvect_h { cont }/pi; %vector col
117 xcorte=dx(1,cont)+R*cos(thetanum);
118 ycorte=dyc(1,cont)+R*sin(thetanum);
119 plot(xcorte, ycorte, 'o')
120 hold on
121 end
122
123 % condiciones  iniciales
124 beta=zeros(1,k-1);
125 aux=[pi/2 zeros (1,k-1)];
126 for tt=1:k-1
127     beta=[beta, aux];
128 end
129
130 alpha0=zeros(n,1);
131 alpha0 (1,1)=3*pi/4;
132 cont=1;
133 for r=2:1:n
134     if r<=cont*k
135         alpha0aux=2*beta(1,r-1)-alpha0(r-1,1)+pi; %posicion  inicial
136         alpha0 (r,1)=alpha0aux-2*pi*floor(alpha0aux/2/pi);
137     else % ultimo  vehiculo  de una fila
138         cont=cont+1;
139         alpha0aux=2*beta(1,r-1)-alpha0(r-k,1)+pi; %posicion  inicial
140         alpha0 (r,1)=alpha0aux-2*pi*floor(alpha0aux/2/pi);
141     end
142 end
143
144 % llamada funcion  auxiliar
145 [~, xplot, yplot, n_vvec, posic_ocupadas]= caida_vehiculos (xmat,ymat,dxc,dyc,n);
146
147 % Actualizacion de la informacion del sistema
148 posic_ocupadas_aux=posic_ocupadas
149 n_vvec_aux=n_vvec;
150
151 % Creacion de nuevas variables
152 col=ones(size (posic_ocupadas,1),1);
153 col1=ones(size (posic_ocupadas,1),1);
154 tiempo_cruce=zeros(size (posic_ocupadas,1),1);
155 tiempoaux=zeros(size (posic_ocupadas,1),1);
156
157 % sentido  trayectorias
158 cir1=[];
159 cir2=[];
160 if rem(n,2)~=0
161     cir1 =[1:2:n];
162     cir2 =[2:2:n-1];
163 else
164     for oo=1:2:k-1
165         cir1 =[ cir1, oo*oo:2:oo*oo+k-1];

```

```

    cir2=[ cir2 , oo*oo+1:2:oo*oo+k];
167   end
    cont=k/2;
169   for oo=2:2:k
        cir1 =[ cir1 , oo*cont+2:2:oo*cont+k];
171         cir2 =[ cir2 , oo*cont+1:2:oo*cont+k];
            cont=cont+1;
173   end
end
175
% Movimiento vehiculos
177 tiempo=8; %ut
    err=tiempo/1000*(w*R);
179 TOL=0.001;
    dt=tiempo*TOL;
181 step=-1;
    jj=length(theta);
183
for t=0:dt:tiempo
185     step=step+1;
        thetab_anti =alpha0+w*t;
187     b=find( thetab_anti >(2*pi) );
        thetab_anti (b)= thetab_anti (b)-2*pi*floor( thetab_anti (b)/2/pi );
189     thetab_hor=alpha0-w*t;
        a=find( thetab_hor <0 );
191     thetab_hor (a)=thetab_hor(a)+2*pi*-floor(thetab_hor(a)/2/pi );

193     thetabb_anti =round( thetab_anti*10000+1);
        thetabb_hor=round( thetab_hor*10000+1);
195
197     a1=find( thetabb_anti >length(theta) ); %vector
        a2=find( thetabb_hor >length(theta) );
        thetabb_anti (a1)= thetabb_anti (a1)-jj*floor( thetabb_anti (a1)/jj ); %para las trayectorias impares
199     thetabb_hor(a2)=thetabb_hor(a2)-jj*floor( thetabb_hor(a2)/jj ); %para las trayectorias pares

201     for pp=1:size(posic_ocupadas,1)
        veh=posic_ocupadas_aux(pp,1);%tray en la que se encuentra
203
        if length( thetaveh_h { veh } )==2 %esquinas
205             if veh==1
                    vecinos =[0,0, veh+1, veh+k];
207             elseif veh==k
                    vecinos =[0,0, veh-1, veh+k];
209             elseif veh==k*k-(k-1)
                    vecinos =[0,0, veh-k, veh+1];
211             elseif veh==n
                    vecinos =[0,0, veh-k, veh-1];
213             end
        elseif length( thetaveh_h { veh } )==3 %laterales
215             if veh<k %1 fila
                    vecinos =[0, veh-1, veh+1, veh+k];
217             elseif veh>k*(k-1) %ultima fila
                    vecinos =[0, veh-k, veh-1, veh+1];
219             elseif rem( veh,k)==0 %derecha
                    vecinos =[0, veh-k, veh-1, veh+k];
221             elseif rem( veh-1,k)==0 %izqda
                    vecinos =[0, veh-k, veh+1, veh+k];
223             end
        elseif length( thetaveh_h { veh } )==4 %zona central
225             vecinos =[veh-k, veh-1, veh+1, veh+k];
        end
227

```

```

229 if isempty(find(cir2==veh))==0 %oes un cir2
if length(thetaveh_h{veh})==2 %esquinas
    a=0;b=0;
231     d=thetab_hor(veh)>thetaveh_h{veh,1}(1,1) &&(thetab_hor(veh)-thetaveh_h{veh,1}(1,1))<err&&
        isempty(find(vecinos(3)==n_vvec))==1;
        e=thetab_hor(veh)>thetaveh_h{veh,1}(1,2) &&(thetab_hor(veh)-thetaveh_h{veh,1}(1,2))<err&&
        isempty(find(vecinos(4)==n_vvec))==1;
233 elseif length(thetaveh_h{veh})==3 %laterales
    a=0;
235     b=thetab_hor(veh)>thetaveh_h{veh,1}(1,1) &&(thetab_hor(veh)-thetaveh_h{veh,1}(1,1))<err&&
        isempty(find(vecinos(2)==n_vvec))==1;
        d=thetab_hor(veh)>thetaveh_h{veh,1}(1,2) &&(thetab_hor(veh)-thetaveh_h{veh,1}(1,2))<err&&
        isempty(find(vecinos(3)==n_vvec))==1;
237     e=thetab_hor(veh)>thetaveh_h{veh,1}(1,3) &&(thetab_hor(veh)-thetaveh_h{veh,1}(1,3))<err&&
        isempty(find(vecinos(4)==n_vvec))==1;
elseif length(thetaveh_h{veh})==4 %zona central
239     a=thetab_hor(veh)>thetaveh_h{veh,1}(1,1) &&(thetab_hor(veh)-thetaveh_h{veh,1}(1,1))<err&&
        isempty(find(vecinos(1)==n_vvec))==1;
        b=thetab_hor(veh)>thetaveh_h{veh,1}(1,2) &&(thetab_hor(veh)-thetaveh_h{veh,1}(1,2))<err&&
        isempty(find(vecinos(2)==n_vvec))==1;
241     d=thetab_hor(veh)>thetaveh_h{veh,1}(1,3) &&(thetab_hor(veh)-thetaveh_h{veh,1}(1,3))<err&&
        isempty(find(vecinos(3)==n_vvec))==1;
        e=thetab_hor(veh)>thetaveh_h{veh,1}(1,4) &&(thetab_hor(veh)-thetaveh_h{veh,1}(1,4))<err&&
        isempty(find(vecinos(4)==n_vvec))==1;
243 end

    if a==1||b==1||d==1||e==1
        c=[a==1 b==1 d==1 e==1];
247     f=find(c==1);
    nv_cruce=vecinos(f); %tray con la que se cruza
249 % tiempoaux(pp,1)=tiempoaux(pp,1)+tiempo_cruce(pp,col1(pp,1)-1)
        tiempo_cruce(pp,col1(pp,1))=t-tiempoaux(pp,1)
251     tiempoaux(pp,1)=t; %ttiempoaux(pp,1)+tiempo_cruce(pp,col1(pp,1));
        col1(pp,1)=col1(pp,1)+1;
253     c=[0 0 0 0];
end
255

elseif isempty(find(cir1==veh))==0 %cir1
257     if length(thet anumvect_a{veh})==2 %esquinas
        a=0;b=0;
259     d=thetab_anti(veh)<thetaveh_a{veh,1}(1,1) &&(thetaveh_a{veh,1}(1,1)-thetab_anti(veh))<err
            &&isempty(find(vecinos(3)==n_vvec))==1;
        e=thetab_anti(veh)<thetaveh_a{veh,1}(1,2) &&(thetaveh_a{veh,1}(1,2)-thetab_anti(veh))<err
            &&isempty(find(vecinos(4)==n_vvec))==1;
261 elseif length(thet anumvect_a{veh})==3 %laterales
        a=0;
263     b=thetab_anti(veh)<thetaveh_a{veh,1}(1,1) &&(thetaveh_a{veh,1}(1,1)-thetab_anti(veh))<err
            &&isempty(find(vecinos(2)==n_vvec_aux))==1;
        d=thetab_anti(veh)<thetaveh_a{veh,1}(1,2) &&(thetaveh_a{veh,1}(1,2)-thetab_anti(veh))<err
            &&isempty(find(vecinos(3)==n_vvec_aux))==1;
265     e=thetab_anti(veh)<thetaveh_a{veh,1}(1,3) &&(thetaveh_a{veh,1}(1,3)-thetab_anti(veh))<err
            &&isempty(find(vecinos(4)==n_vvec_aux))==1;
elseif length(thet anumvect_a{veh})==4 %zona central
267     a=thetab_anti(veh)<thetaveh_a{veh,1}(1,1) &&(thetaveh_a{veh,1}(1,1)-thetab_anti(veh))<err
            &&isempty(find(vecinos(1)==n_vvec_aux))==1;
        b=thetab_anti(veh)<thetaveh_a{veh,1}(1,2) &&(thetaveh_a{veh,1}(1,2)-thetab_anti(veh))<err
            &&isempty(find(vecinos(2)==n_vvec_aux))==1;
269     d=thetab_anti(veh)<thetaveh_a{veh,1}(1,3) &&(thetaveh_a{veh,1}(1,3)-thetab_anti(veh))<err
            &&isempty(find(vecinos(3)==n_vvec_aux))==1;
        e=thetab_anti(veh)<thetaveh_a{veh,1}(1,4) &&(thetaveh_a{veh,1}(1,4)-thetab_anti(veh))<err
            &&isempty(find(vecinos(4)==n_vvec_aux))==1;
271 end

```

```

273     if a==1||b==1||d==1||e==1
274         c=[a==1 b==1 d==1 e==1];
275         f=find(c==1);
276         nv_cruce=vecinos(f);%tray con la que se cruza
277     % tiempoaux(pp,1)=tiempoaux(pp,1)+tiempo_cruce(pp,col1(pp,1)-1)
278         tiempo_cruce(pp,col1(pp,1))=t-tiempoaux(pp,1)
279         tiempoaux(pp,1)=t; %tiempoaux(pp,1)+tiempo_cruce(pp,col1(pp,1));
280         col1(pp,1)=col1(pp,1)+1;
281         c=[0 0 0 0];
282     end
283 end
284 end
285
286 if n_vvec~=0
287     for pp=1:length(n_vvec)
288         n_v=n_vvec(pp);
289
290     if length(thetnumvect_h{n_v})==2 %esquinas
291         if n_v==1
292             posic_posibles =[0,0,n_v+1,n_v+k];
293         elseif n_v==k
294             posic_posibles =[0,0,n_v-1,n_v+k];
295         elseif n_v==k*k-(k-1)
296             posic_posibles =[0,0,n_v-k,n_v+1];
297         elseif n_v==n
298             posic_posibles =[0,0,n_v-k,n_v-1];
299         end
300     elseif length(thetnumvect_h{n_v})==3 %laterales
301         if n_v<k %1 fila
302             posic_posibles =[0,n_v-1,n_v+1,n_v+k];
303         elseif n_v>k*(k-1) %ultima fila
304             posic_posibles =[0,n_v-k,n_v-1,n_v+1];
305         elseif rem(n_v,k)==0 %derecha
306             posic_posibles =[0,n_v-k,n_v-1,n_v+k];
307         elseif rem(n_v-1,k)==0 %izqda
308             posic_posibles =[0,n_v-k,n_v+1,n_v+k];
309         end
310     elseif length(thetnumvect_h{n_v})==4 %zona central
311         posic_posibles =[n_v-k,n_v-1,n_v+1,n_v+k];
312     end
313
314     if isempty(find(cir1==n_v))==0 %se cae un cir1
315         if length(thetnumvect_h{n_v})==2 %esquinas
316             a=0;b=0;
317             d=thetab_hor( posic_posibles (3))>thetnumvect_h{n_v,1}(1,1)&&(thetab_hor(posic_posibles (3))
318                 -thetnumvect_h{n_v,1}(1,1))<err&&isempty(find(posic_posibles(3)==n_vvec))==1;
319             e=thetab_hor( posic_posibles (4))>thetnumvect_h{n_v,1}(1,2)&&(thetab_hor(posic_posibles
320                 (4))-thetnumvect_h{n_v,1}(1,2))<err&&isempty(find(posic_posibles(4)==n_vvec))
321                 ==1;
322         elseif length(thetnumvect_h{n_v})==3 %laterales
323             a=0;
324             b=thetab_hor( posic_posibles (2))>thetnumvect_h{n_v,1}(1,1)&&(thetab_hor(posic_posibles
325                 (2))-thetnumvect_h{n_v,1}(1,1))<err&&isempty(find(posic_posibles(2)==n_vvec))
326                 ==1;
327             d=thetab_hor( posic_posibles (3))>thetnumvect_h{n_v,1}(1,2)&&(thetab_hor(posic_posibles
328                 (3))-thetnumvect_h{n_v,1}(1,2))<err&&isempty(find(posic_posibles(3)==n_vvec))
329                 ==1;
330             e=thetab_hor( posic_posibles (4))>thetnumvect_h{n_v,1}(1,3)&&(thetab_hor(posic_posibles
331                 (4))-thetnumvect_h{n_v,1}(1,3))<err&&isempty(find(posic_posibles(4)==n_vvec))
332                 ==1;
333         elseif length(thetnumvect_h{n_v})==4 %zona central

```

```

325 a=thetab_hor( posic_posibles (1))>thetanutvect_h{n_v,1}(1,1)&&(thetab_hor(posic_posibles
      (1))-thetanutvect_h{n_v,1}(1,1)<err&&isempty(find(posic_posibles(1)==n_vvec))
      ==1;
      b=thetab_hor( posic_posibles (2))>thetanutvect_h{n_v,1}(1,2)&&(thetab_hor(posic_posibles
      (2))-thetanutvect_h{n_v,1}(1,2)<err&&isempty(find(posic_posibles(2)==n_vvec))
      ==1;
327 d=thetab_hor( posic_posibles (3))>thetanutvect_h{n_v,1}(1,3)&&(thetab_hor(posic_posibles
      (3))-thetanutvect_h{n_v,1}(1,3)<err&&isempty(find(posic_posibles(3)==n_vvec))
      ==1;
      e=thetab_hor( posic_posibles (4))>thetanutvect_h{n_v,1}(1,4)&&(thetab_hor(posic_posibles
      (4))-thetanutvect_h{n_v,1}(1,4)<err&&isempty(find(posic_posibles(4)==n_vvec))
      ==1;
329 end

      if a==1||b==1||d==1||e==1
      c=[a==1 b==1 d==1 e==1]
333 f=find(c==1);
      nv_cambia=posic_posibles(f);%numero de trayectoria que cambia
335 for ii =1: size (posic_ocupadas,1)
      if nv_cambia==posic_ocupadas(ii,col( ii ,1))
337 col( ii ,1)=col( ii ,1)+1;
      posic_ocupadas( ii , col( ii ,1 ))=n_v
339 posic_ocupadas_aux( ii ,1)=n_v;
      end
341 end
      c=[0 0 0 0];
343 xplot(n_v,:)=xmat(n_v,:);
      yplot(n_v,:)=ymat(n_v,:);
345 xplot(nv_cambia,:)=dxc(nv_cambia);
      yplot(nv_cambia,:)=dyc(nv_cambia);
347 n_vvec_aux(pp)=n_vvec(pp);
      n_vvec(pp)=nv_cambia;
349 end

elseif isempty(find( cir2==n_v))==0 %cir2 caido
      if length(thetanutvect_a{n_v})==2 %esquinas
353 a=0;b=0;
      d=thetab_anti( posic_posibles (3))<thetanutvect_a{n_v,1}(1,1)&&(thetanutvect_a{n_v
      ,1}(1,1)-thetab_anti( posic_posibles (3)))<err&&isempty(find(posic_posibles(3)==
      n_vvec))==1;
355 e=thetab_anti( posic_posibles (4))<thetanutvect_a{n_v,1}(1,2)&&(thetanutvect_a{n_v
      ,1}(1,2)-thetab_anti( posic_posibles (4)))<err&&isempty(find(posic_posibles(4)==
      n_vvec))==1;
elseif length(thetanutvect_a{n_v})==3 %laterales
357 a=0;
      b=thetab_anti( posic_posibles (2))<thetanutvect_a{n_v,1}(1,1)&&(thetanutvect_a{n_v
      ,1}(1,1)-thetab_anti( posic_posibles (2)))<err&&isempty(find(posic_posibles(2)==
      n_vvec))==1;
359 d=thetab_anti( posic_posibles (3))<thetanutvect_a{n_v,1}(1,2)&&(thetanutvect_a{n_v
      ,1}(1,2)-thetab_anti( posic_posibles (3)))<err&&isempty(find(posic_posibles(3)==
      n_vvec))==1;
      e=thetab_anti( posic_posibles (4))<thetanutvect_a{n_v,1}(1,3)&&(thetanutvect_a{n_v
      ,1}(1,3)-thetab_anti( posic_posibles (4)))<err&&isempty(find(posic_posibles(4)==
      n_vvec))==1;
361 elseif length(thetanutvect_a{n_v})==4 %zona central
      a= thetab_anti( posic_posibles (1))<thetanutvect_a{n_v,1}(1,1)&&(thetanutvect_a{n_v,1}(1,1)
      -thetab_anti( posic_posibles (1)))<err&&isempty(find(posic_posibles(1)==n_vvec))==1;
363 b= thetab_anti( posic_posibles (2))<thetanutvect_a{n_v,1}(1,2)&&(thetanutvect_a{n_v
      ,1}(1,2)-thetab_anti( posic_posibles (2)))<err&&isempty(find(posic_posibles(2)==
      n_vvec))==1;

```

```

d=thetab_anti ( posic_posibles (3))<thet anumvect_a{n_v,1}(1,3)&&(thet anumvect_a{n_v
,1}(1,3)-thetab_anti( posic_posibles (3)))<err&&isempty(find(posic_posibles(3)==
n_vvec))==1;
365 e=thetab_anti ( posic_posibles (4))<thet anumvect_a{n_v,1}(1,4)&&(thet anumvect_a{n_v
,1}(1,4)-thetab_anti( posic_posibles (4)))<err&&isempty(find(posic_posibles(4)==
n_vvec))==1;

end
367
if a==1||b==1||d==1||e==1
369 c=[a==1 b==1 d==1 e==1]
f=find(c==1);
371 nv_cambia=posic_posibles(f)%numero de trayectoria que cambia
for ii =1: size (posic_ocupadas,1)
373 if nv_cambia==posic_ocupadas(ii,col ( ii ,1))
col ( ii ,1)=col ( ii ,1)+1;
375 posic_ocupadas ( ii , col ( ii ,1))=n_v;
posic_ocupadas_aux(ii ,1)=n_v;
377 end
end
379 c=[0 0 0 0];
xplot (n_v ,:)=xmat(n_v,:);
381 yplot (n_v ,:)=ymat(n_v,:);
xplot (nv_cambia,:)=dxc(nv_cambia);
383 yplot (nv_cambia,:)=dyc(nv_cambia);
n_vvec_aux(pp)=n_vvec(pp);
385 n_vvec(pp)=nv_cambia;
end
387 end
end
389 end

for mm=cir1 %antih
391 xdib (round(mm/2),1)=xplot(mm,thetabb_anti(mm));
393 ydib (round(mm/2),1)=yplot(mm,thetabb_anti(mm));
end
395 for mm=cir2 %hor
xdib1 (round(mm/2),1)=xplot(mm,thetabb_hor(mm));
397 ydib1 (round(mm/2),1)=yplot(mm,thetabb_hor(mm));
end
399 plot (xdib,ydib,'or','linewidth',2,'markersize',12,'markeredgecolor','m','markerfacecolor','g'
)
plot (xdib1,ydib1,'or','linewidth',2,'markersize',12,'markeredgecolor','m','markerfacecolor','g'
)
401 title (['step : ' num2str(step)]);
axis equal;
403 pause(0.01);
end
405 vehiculos =posic_ocupadas (:,1) ;
tmax=max(max(tiempo_cruce));
407 veh_tiempo_cruce=[vehiculos, tiempo_cruce];
end

```

# Bibliografía

---

- [1] Y.U. Cao, A.S. Fukunaga, A. Kahng *Cooperative mobile robotics: Antecedents and directions*, in *Autonomous robots*, vol. 4, no. 1, pp.7-27, 1997.
- [2] A. Bicchi, A. Danesi, G. Dini, S. La Porta, L. Pallottino, I. Savino, R. Schiavi *Heterogeneous wireless multirobot system*, in *Robotics Automation Magazine*, IEEE, vol. 15, no. 1, pp.62-70, March 2008.
- [3] *Aerial Robotics Cooperative Assembly System* Proyecto ARCAS, 2015. [Online]. Available: <http://www.arcas-project.eu/>
- [4] [Online]. Available: <http://www.aracnicoptero.com/> 'Áracnóptero'. Empresa Arbórea
- [5] Instituto Nacional de Tecnología Aeroespacial [Online]. Available: <http://www.inta.es/programasAltaTecnologia.aspx?Id=1&SubId=3> INTA
- [6] C. Rizzo, D. Tardioli, D. Sicignano, L. Riazuelo, J.L. Villarroel, L. Montano *Signal-based deployment planning for robot teams in tunnel-like fading environments*, in *The International Journal of Robotics Research*, vol. 32, no. 12, pp.1381-1397, 2013. [Online]. Available: <http://ijr.sagepub.com/content/32/12/1381.abstract>
- [7] K.E. Bekris, D.K. Grady, M. Moll, L.E. Kavraki *Safe distributed motion coordination for second-order systems with different planning cycles* in *The International Journal of Robotics Research*, vol. 31, no. 2, pp.129-150, 2012. [Online]. Available: <http://ijr.sagepub.com/content/31/2/129.abstract>
- [8] R. Zlot y A. Stentz *Market-based Multirobot Coordination for Complex Tasks*, in *The International Journal of Robotics Research*, vol. 25, no. 1, pp.73-101, 2006. [Online]. Available: <http://ijr.sagepub.com/content/25/1/73>
- [9] W. Burgard, M. Moors, C. Stachniss, F.E. Schneider *Coordinated multi-robot exploration*, in *Robotics*, IEEE *Transactions on*, vol. 21, no. 3, pp.376-386, 2005.
- [10] W. Sheng, Q. Yang, J. Tan, N. Xi, *Distributed multi-robot coordination in area exploration*, in *Robotics and Autonomous Systems*, vol. 54, no. 12, pp.945-955, 2006.
- [11] F. Pasqualetti, A. Franchi, F. Bullo *On cooperative patrolling: Optimal trajectories, complexity analysis, and approximation algorithms*, in *Robotics*, IEEE *Transactions on*, vol. 28, no. 3, pp.592-606, 2012.
- [12] J. Acevedo, B.C. Arrue, J.M. Díaz Báñez, I. Ventura, I. Maza, A. Ollero *One-to-one coordination algorithm for decentralized area partition in surveillance missions with a team*

- of aerial robots*, in *Journal of Intelligent and Robotic Systems*, vol. 74, no. 1-2, pp.269-285, 2014.
- [13] M. Bernard, K. Kondak, G. Hommel *Load transportation system based on autonomous small size helicopters*, in *Aeronautical Journal*, vol. 114, no. 1153, pp.191-198, 2010.
- [14] K.P. Valavanis, G.J. Vachtsevanos *Handbook of Unmanned Aerial Vehicles*. Springer, 2015
- [15] J.M. Díaz Báñez, E. Caraballo, M.A. Lopez, S. Bereg, I. Maza, A. Ollero *The synchronization problem for information exchange between aerial robots under communication constraints*, in *Robotics and Automation (ICRA), 2015 International Conference on IEEE*, pp.4650-4655.
- [16] D.W. Casbeer, D.B. Kingston, R.W. Beard, T.W. McLain *Cooperative forest fire surveillance using a team of small unmanned air vehicles*, in *International Journal of Systems Science*, vol. 37, no. 6, pp.351-360, 2006.
- [17] K. Alexis, G. Nikolakopoulos, A. Tzes, L. Dritsas *Coordination of helicopter uavs for aerial forest-fire surveillance*, in *Applications of intelligent control to engineering systems*. Springer, 2009, pp. 169-193.
- [18] A. Franchi, L. Freda, G. Oriolo, M. Vendittelli *The sensor-based random graph method for cooperative robot exploration*, in *Mechatronics, (IEEE/ASME) Transactions on*, vol. 14, no. 2, pp.163-175, 2009.
- [19] S. Yoon, C. Qiao *A New Search Algorithm using Autonomous and Cooperative Multiple Sensor Nodes*, in *IEEE INFOCOM*, 2007, pp.937-945.
- [20] J.M. Díaz Báñez, E. Caraballo, M.A. Lopez, S. Bereg *The starvation problem*. Working paper, 2015. Science Journal, -, pp.12.
- [21] R. Conde Ojeda Proyecto Fin de Máster: *Filtros de partículas aplicados a la navegación y planificación de UAVs* vol. 1, Predicción de trayectorias, cap. 4, pp. 18-22, 2009.

