

INTRODUCCIÓN A LOS UAVs Y SISTEMAS DE ENJAMBRE CON ROS Y MAVLINK



Universitat d'Alacant
Universidad de Alicante



Ignacio Pérez Vilaplana

Resumen

El uso de vehículos aéreos no tripulados se ha popularizado enormemente en los últimos años. Dadas sus características de bajo costo, autonomía y el bajo riesgo humano, ofrecen un gran potencial para uso tanto civil como militar.

En ese sentido, son muchas las aplicaciones que pueden ser realizadas formando enjambres, ya que se puede sustituir un solo UAV (Unmanned Aerial Vehicle) con una cierta autonomía dada, por una flota de drones organizados de tal forma de cubrir esa misma tarea en un tiempo menor.

En este contexto, ROS (Robot Operating System) permite implementar acciones de alto nivel de cada dron, así como la red de comunicación entre ellos y con la computadora en tierra. Por otro, MAVLink es el protocolo que usa ArduPilot para la comunicación, tanto del vehículo a la estación de tierra como entre los vehículos. La combinación de ROS y el protocolo MAVLink de comunicación esta recogida en el paquete de ROS - MAVROS, el cual proporciona un controlador de comunicación para varios pilotos automáticos con el protocolo de comunicación MAVLink, siendo este paquete ideal para trabajar con sistemas robóticos de enjambre.

Durante el trabajo encontrará una introducción a los drones y sistemas de enjambre, así como al software típicamente utilizado en este tipo de proyectos. Tras esto se atacará superficialmente, comentando algunos métodos, el problema de la planificación de las trayectorias que tienen los drones aéreos, y se planteará un pseudo ejemplo práctico del cuál se comentará la estructura que sigue el enjambre así como la estructura de la lógica que deberían seguir los UAVs del enjambre para resolver un problema planteado. Además ROS proporciona un paquete de introducción, *HECTOR*, del cuál se hará una pequeña introducción. Por último se comentarán algunas de las aplicaciones más interesantes que se han encontrado durante la investigación de este trabajo.

Índice

1. Estado del arte	2
1.1. Introducción a los UAVs	2
1.2. Introducción a los sistemas de enjambre	2
1.2.1. Arquitecturas de los sistemas robóticos de enjambre	3
2. Software	3
2.1. ROS	3
2.2. MAVLink	4
2.3. SITL	4
2.4. MAVROS	5
3. Programación	5
3.1. Planificación de trayectorias en enjambres de UAVs	6
3.2. Estructura de enjambre. Líder y seguidor	7
3.3. Líder. Nodos líderes.	8
3.4. Seguidores. Nodos seguidores.	8
4. Ejemplo práctico y aproximación con ROS: HECTOR_QUADROTOR	10
4.1. Compilación de demostración de vuelo de quadrotor al aire libre: Paso a paso	11
5. Posibles aplicaciones de UAVs en sistemas de enjambre	12
6. Conclusiones	15

1. Estado del arte

1.1. Introducción a los UAVs

Los UAVs son aquellas aeronaves que navegan sin la presencia de un piloto a bordo [3]. Los UAVs pueden ser dirigidos por un piloto desde tierra, bien a través de un sistema de mando por radiofrecuencia (RPAS), o bien guiados de manera automática a través de una estación de control de tierra (Ground Control Station (GCS)), pero también pueden realizar navegación autónoma sin supervisión, empleando sensores embarcados en el mismo. Aunque esta terminología no se ciñe únicamente a la aeronave, si no que incluye todo el software de apoyo, incluyendo sensores, controladoras de vuelo o autopilotos, estaciones terrestres, sistemas de comunicación e interfaces de usuario [4].

Según el principio de vuelo, los UAVs se pueden clasificar en dos tipos, según la necesidad de un rotor o sistema de empuje para volar o, si por el contrario su navegación depende de la fuerza de rotación. En caso de que el UAV tenga una masa sustancial y, por tanto, necesite de un sistema de empuje, se dividen en UAVs de tipo ala o UAVs de tipo rotor, mientras que aquellos UAVs que sean más livianos del aire, son conocidos como dirigibles o globos. Dentro de los primeros se puede establecer un escalón más, así pues, los de ala pueden dividirse en tres grupos: ala fija, ala voladora y ala flotante, mientras que los de tipo rotor se diferencian en helicópteros o multirrotores, siendo estos últimos clasificables en función de la cantidad de rotores que compongan el sistema de propulsión [11].

El control autónomo de los robots se presenta como una dificultad en la actualidad y en el futuro de muchas aplicaciones como, por ejemplo, exploración de áreas de difícil acceso humano o ambientes contaminados, que exigen el desarrollo de plataformas móviles autónomas capaces de maniobrar de una manera ágil y estable en lugares donde no es posible la asistencia humana. Por esta razón, es válido trabajar con helicópteros de cuatro rotores de despegue y aterrizaje vertical, o como se conocen por sus siglas en inglés, “VTOL” (Vertical takeoff and landing). Estos helicópteros, también llamados “quadrotors”, son muy prácticos para trabajos donde se requiere gran capacidad de maniobra a bajas velocidades, lo que se puede lograr debido a su reducido peso y tamaño.

1.2. Introducción a los sistemas de enjambre

La robótica de enjambre es un campo de investigación dentro del área de la robótica que estudia la coordinación de un gran número de robots simples [14]. Este campo de investigación se inspira en el comportamiento observado en los insectos sociales, los cuales son grandes ejemplos de cómo un gran número de individuos simples pueden interactuar para crear sistemas inteligentes colectivos. En estos sistemas el comportamiento colectivo emerge de forma auto-organizada a partir de las interacciones entre los individuos y de estos con el entorno. De la misma manera en los sistemas de enjambre artificiales, la inteligencia es

una propiedad emergente a partir del comportamiento global del enjambre. En estos sistemas, el enjambre es capaz de llevar a cabo tareas, de manera global, que están fuera de las capacidades de un robot individual. Los sistemas de enjambre deben cumplir una serie de características que los diferencian de otros sistemas multirobóticos:

- Los robots que forman el enjambre deben ser robots autónomos situados en el entorno.
- El enjambre debe estar formado por un gran número de robots que podrán dividirse en grupos más pequeños de robots homogéneos.
- Los robots deberán disponer de sensores locales y capacidades de comunicación limitadas

Estas características aseguran la coordinación entre los robots será distribuida, que el sistema tendrá una alta tolerancia a fallos ya que debido a la redundancia de robots cada uno de los agentes que forman el sistema es prescindible pudiendo ser sustituido por otro agente, y que el sistema será escalable, permitiendo añadir o eliminar más agentes según la tarea lo requiera.

1.2.1. Arquitecturas de los sistemas robóticos de enjambre

Las arquitecturas existentes pueden ser divididas de manera general en dos grupos: las *arquitecturas individuales*; que se centran en las decisiones y comportamiento de un único robot (follow the leader), y las *arquitecturas multirobóticas*; que prestan más atención a la gestión de las relaciones y flujos de información entre los diferentes componentes para poder llevar a cabo tareas que necesiten mecanismo cooperativos.

2. Software

A continuación se va a introducir el software necesario para trabajar con drones y sistemas de enjambre.

2.1. ROS

¿Qué es ROS? y ¿Por qué su uso es interesante en la implementación de robotica de enjambres?

ROS (Robot operating system) es un software específicamente creado para el desarrollo de software en robots. Es un software libre, desarrollado originalmente en el Laboratorio de Inteligencia Artificial de *Stanford University*, y posteriormente mantenido por la empresa *Willow Garage* y la organización *Open Robotics*. Su funcionamiento puede describirse como una estructura de grafos, donde los nodos son rutinas o programas e interactúan entre sí mediante *topics*. Todo el registro y comunicación entre nodos es monitorizado por un ROS master. Este tipo de arquitectura modular de software se acompaña con la arquitectura

de hardware de los robots, que generalmente consiste en una red de sensores y controladores que se comunican entre sí. ROS permite comunicarse con los sensores y actuadores del robot, posibilitando un fácil acceso a la información ya procesada de estos dispositivos, generalmente en forma de *topics*.

Por todo esto, el entorno ROS se cree que es el mejor para implementar los comportamientos de cada integrante del enjambre, así como la comunicación y el flujo de información entre los mismos. Además cuenta con una variedad de herramientas de desarrollo como compiladores, simuladores, visualizadores que permiten automatizar tareas y analizar los datos entregados por el robot en tiempo real. En resumen, el motivo por el cual es útil emplear ROS en este tipo de proyectos es por la implementación de las acciones de control de alto nivel de cada dron, así como la red de comunicación entre ellos y con la computadora en tierra.

2.2. MAVLink

MAVLink es el protocolo que usa ArduPilot (software autopilot open-source) para la comunicación, tanto del vehículo a la estación de tierra como entre los vehículos y es el protocolo usado en todos o casi todos los drones del mercado actuales. Este protocolo serie tiene muchos mensajes que se usan para transmitir entre otros datos la posición obtenida con el sistema Global Positioning System (GPS), la actitud y la velocidad del vehículo. Los mensajes de este software tienen un formato diferente al que utiliza ROS, como es lógico, es por eso, a primera vista, estos dos software parecen incompatibles.

2.3. SITL

¿Qué es SITL?

SITL (software in the loop) es un simulador que permite emular el firmware de una unidad de control de vuelo (FCU - Flight Control Unit) en una computadora sin necesidad de ningún hardware adicional.

Entonces, ¿es imprescindible utilizar SITL si se va a trabajar con drones y sistemas de enjambre?

La respuesta corta es que no necesariamente. Sin embargo, como todos los programas, es preferible hacer una simulación y asegurarse de que todo va bien antes de aplicarlo con hardware real. Es por ello que aunque no es obligatorio con este simulador, es altamente recomendable, más en proyectos de este tipo. Entonces básicamente, es una compilación del código de la controladora de vuelo que utiliza un compilador C++ ordinario, lo que brinda un ejecutable que permite probar el comportamiento del código desarrollado, interactuando con el firmware emulado y con el modelo simulado en Gazebo, sin el hardware de la controladora real.

2.4. MAVROS

¿Como se enlaza entonces el protocolo MAVLink de Ardupilot con ROS?

Se ha dicho las ventajas que proporciona utilizar ROS en la comunicación entre drones y lo útil que resulta utilizarlo con software principal a la hora de crear proyectos con drones y sobre todo, con sistemas de enjambre, pero, como se ha dicho, los UAVs, utilizan un protocolo de comunicación llamado MAVLink, cuya salida de mensajes no esta en el mismo formato que los mensajes de ROS. Por lo tanto, se necesita un puente que uno los dos. Este puente es MAVROS [7].

MAVROS es un paquete de ROS que permite la comunicación MAVLink entre computadoras que ejecutan ROS, controladoras de vuelo y estaciones de control terrestre que sean compatibles con el protocolo MAVLink. El paquete contiene las bibliotecas que definen los mensajes, así como un conjunto de launches para establecer la conexión con diferentes controladoras y ejecutar comandos particulares, y un set de nodos ejemplificando la simulación en Gazebo.

Luego de que se ejecuta un roscore, se debe realizar la conexión con la controladora de vuelo (ya sea una controladora real o una simulada en Gazebo) utilizando el archivo launch adecuado. Al establecer la conexión, se ejecuta el nodo de nombre mavros, y se ponen a disposición una larga lista de *topics*, algunos sobre los cuales el sistema comienza a publicar el estado actual reportado por la controladora, y otros a los que se suscribe, esperando que un agente externo (otro nodo) publique en ellos algún comando. Las definiciones de los *topics* disponibles, así como los tipos de mensajes y servicios, se encuentran en la web oficial del paquete.

En resumen, para que un nodo interactúe con la aeronave, basta con importar las bibliotecas de mensajes MAVROS que se deseen escribir o leer, y utilizar las funciones de publicación y suscripción a *topics* que provee el paquete. La sintaxis que se utiliza varía según el lenguaje elegido (Los nodos de ROS pueden ser programados en python o C++) pero el funcionamiento es el mismo ya que el paquete MAVROS soporta ambos.

3. Programación

En los últimos años se ha extendido el uso de sistemas de enjambre para acometer una amplia variedad de tareas tanto en entornos estructurados y seguros como en ambientes abiertos, dinámicos y hostiles [5]. Uno de los aspectos más importantes a tratar dentro de los sistemas multi-robot es la planificación, definida como la tarea de crear una secuencia de acciones que permita a los diferentes vehículos alcanzar una meta. Este aspecto incluye dos fases distintas: **planificación de tareas**, donde el robot debe ejecutar cada tarea, y **planificación de movimiento**, que busca generar una trayectoria segura para cada vehículo.

Los métodos para la planificación de trayectorias permiten dotar a los enjambres de autonomía a la hora de acometer misiones, además de, garantizar la seguridad

de todos los elementos presentes en el entorno al planificar las rutas considerando colisiones tanto con los obstáculos estáticos como con los dinámicos presentes en el entorno.

Actualmente, existe una amplia gama de tecnologías disponibles para resolver el problema de la planificación de trayectorias, tales como métodos basados en gráficos, métodos basados en campos potenciales, métodos probabilísticos o métodos basados en el aprendizaje profundo. Dentro de las distintas formas de abordar este problema, los métodos probabilísticos permiten explorar grandes espacios sin aumentar exponencialmente el tiempo de cálculo y, además, en el caso de la planificación para enjambres, permiten reutilizar el gráfico generado tras la exploración del entorno tantas veces como agentes forman el sistema, reduciendo así el gasto computacional derivado del uso de este tipo de métodos.

3.1. Planificación de trayectorias en enjambres de UAVs

Los métodos probabilísticos agrupan dos grandes métodos de planificación, clasificados como pasivos y activos, dependiendo de si es necesario utilizar un segundo tipo de algoritmo para establecer el camino óptimo entre la configuración inicial y final. Los métodos pasivos incluyen las Hojas de Ruta Probabilísticas (PRM), mientras que los métodos activos incluyen la Exploración Rápida de Árboles Aleatorios (RRT). Este trabajo se centra en estudiar un método basado en PRM, que permita establecer como combinación de trayectorias óptimas para un sistema de enjambre que trabaja en un entorno estructurado [6].

PRM posee la ventaja de poder explorar entornos de grandes dimensiones en un corto periodo de tiempo y, generar un grafo con el conjunto total de posibles trayectorias. Esta primera fase se combina con una segunda fase en la que, a través de un algoritmo descomposición de grafos como el algoritmo A^* , se establecen, de entre todas las soluciones posibles, aquellas que optimiza la distancia recorrida para cada uno de los UAV del enjambre. El desarrollo propuesto en 2D muestra que, a través de dicha combinación, se puede obtener una solución global eficaz al problema de la planificación de trayectorias, para un enjambre de UAVs, formado por un número variable de drones. Junto a esta validación, la implementación de métodos en 2D se establece, en primer lugar, como un paso previo al desarrollo de un planificador global de trayectorias en 3D. En segundo lugar, de cara a la aplicación directa de este planificador en entornos abiertos, caracterizados por no poseer obstáculos solapados en a distintas alturas, como sí sucedería en el caso de un entorno urbano o un edificio de viviendas, dónde el solapamiento en alturas de diferentes obstáculos obliga a realizar la planificación global sobre un modelo tridimensional del entorno.

Si bien la solución presentada en esta sección es aplicable a distintas situaciones, las cuales poseen desarrollos diferentes y adicionales a la hora de generar las trayectorias óptimas y, cuyas modificaciones son detalladas posteriormente, el método de planificación en 2D comparte un esquema general para todos los casos formado por un conjunto de fases, como las mostradas en la Figura [1].



Figura 1: Esquema para la planificación de trayectorias en 2D

Ahora bien, estos desarrollos establecidos en 2D, sirven como paso previo a la implementación de un método que permita obtener un planificador de trayectorias global a partir de información tridimensional del entorno. La necesidad de trabajar con información del entorno en 3D, como puede ser el caso de los incendios en entornos urbanos o edificios de viviendas, conlleva la introducción de cambios en el método de planificación de trayectorias presentado. La idea entonces establecer una extensión del método PRM de 2D a 3D que genere una solución óptima y escalable, de cara a ser aplicada en enjambres de UAVs, que puedan operar en diferentes entornos y bajo distintos planteamientos.

Como en el caso de la planificación en 2D, al trabajar con un algoritmo basado en muestreo pasivo, es necesario combinar la exploración probabilística del entorno con el algoritmo A* para poder establecer la solución óptima. Al igual que en la planificación en 2D, el método propuesto se conforma de un conjunto de fases, tal y como se sintetiza en la Figura [2]. Como se puede observar, el esquema seguido es similar al caso de la planificación en 2D, pero con diferencias significativas, tanto en la primera fase de generación de mapa del entorno, como en la implementación del algoritmo de PRM para una exploración del área de en 3D.



Figura 2: Esquema para la planificación de trayectorias en 3D

3.2. Estructura de enjambre. Líder y seguidor

Una dinámica muy utilizada en los sistemas de enjambre de drones es la *Follow the leader*. Esta estrategia se basa en el seguimiento de un subconjunto de UAVs a uno líder. Estos "seguidores" calculan la trayectoria y evasión de

obstáculos en función del comportamiento del "líder". A continuación se va a explicar un poco la estructura que tienen o debería de tener los nodos de los drones líder y la de los nodos de los seguidores. Para ello se plantea una misión: *"Se dispone de un enjambre con 3 drones cuadricópteros que implementan una dinámica de Follow the leader donde uno de ellos actúa como líder y al cuál se le asigna una trayectoria circular, mientras que los otros dos seguidores tienen la misión de seguir al líder manteniendo cierta formación y evitando colisiones (estrategia mixta [7])."*

3.3. Líder. Nodos líderes.

El diagrama de la Figura [3] describe de manera concisa una posible implementación para que un dron realice una trayectoria circular. Para este caso, teniendo en cuenta que se trabaja con ROS, los nodos de los líderes reciben como parámetros desde el launch únicamente valores que determinan la trayectoria a realizar, como son por ejemplo el tiempo total de vuelo, y parámetros geométricos como la altura de vuelo y similares. Por ejemplo, para este caso en concreto que la trayectoria es un círculo, el parámetro correspondiente al radio de este se proporciona desde los archivos del launch. El nodo líder es el encargado de publicar la posición para el sistema de coordenadas cartesiano absoluto utilizado por los tres drones. Para esto, el líder publicará ciertos valores en diferentes topics para indicar esta posición. Una vez el líder tenga una estimación relativamente precisa de su latitud y longitud publicará estos valores en el mismo topic. Así mismo, es el líder el que se encarga de indicar cuando se aterriza (variable modo LAND).

3.4. Seguidores. Nodos seguidores.

Por otro lado, para aplicar los nodos de los drones seguidores, lo que se necesitaría es que estos nodos recibieran desde el launch parámetros que indican la altura de vuelo a la que se posicionarán luego de despegar, y antes de aterrizar, la diferencia de altura que deben mantener con el dron líder, el ángulo con el que deben mantener su posición relativa al líder con el vector velocidad de líder, un número identificador del seguidor (1 o 2 en este caso), y condiciones de armado, despegue y aterrizaje (que pueden depender de condiciones de tiempo o de la posición y comportamiento que tome el líder, según la implementación que se haga).

Como se muestra en la Figura [4], al inicio el dron seguidor espera a que se cumpla su condición de inicio antes de .armarse". Una vez .armado", antes de despegar, el dron espera (si es necesario) a que el líder publique la posición del sistema de coordenadas cartesiano para tener un marco de referencia absoluto compartido entre los drones. Una vez que recibe esta posición, la dinámica es similar a la del dron líder, en la que a partir de su longitud, latitud y altura, publica su posición en un topic. El dron seguidor recibe las velocidades de los otros drones (líder y el resto de seguidores) y sus posiciones en el marco absoluto y usando esta información es que puede seguir al líder y evitar obstáculos

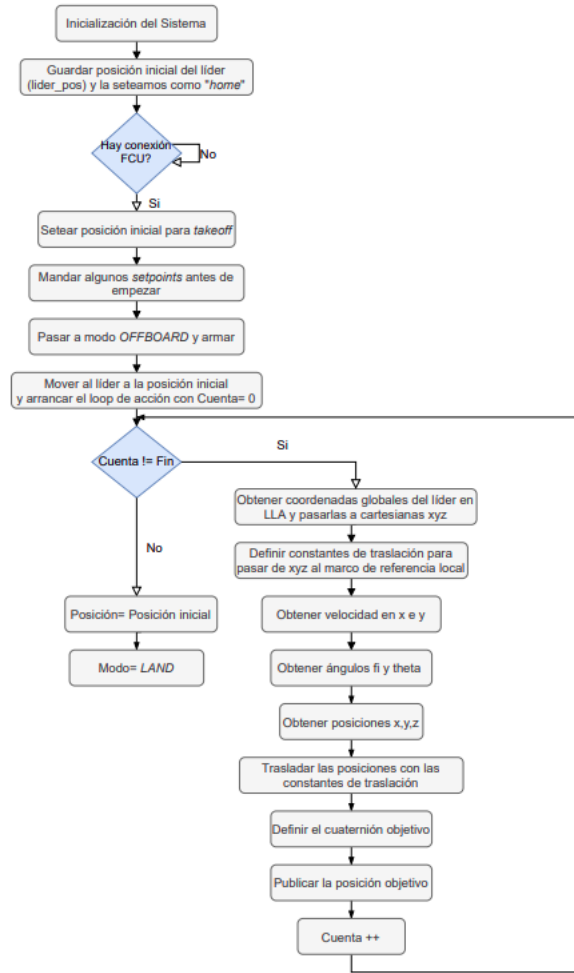


Figura 3: Diagrama de flujo del nodo que implementa que el líder siga una trayectoria circular

y/o otros drones. Cabe destacar que es cada dron seguidor el encargado de convertir su latitud, longitud y altura al sistema cartesiano absoluto y publicarla para que los demás drones dispongan de la información. Por último, los seguidores están atentos siempre de la variable de aterrizaje (LAND) del líder.

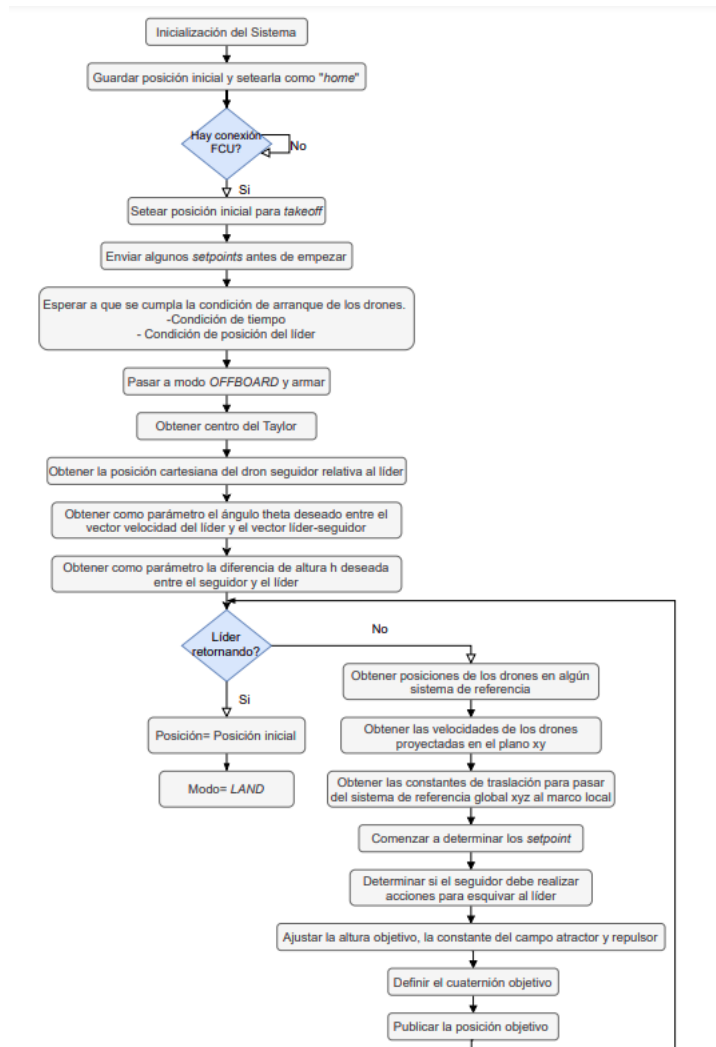


Figura 4: Diagrama de flujo del nodo que implementa un dron seguidor para poder seguir al líder y/o esquivar los obstáculos que surjan.

4. Ejemplo práctico y aproximación con ROS: HECTOR_QUADROTOR

Hector_quadrotor contiene paquetes relacionados con el modelado, control y simulación de sistemas quadrotor UAV [8]. Los paquetes disponibles son:

- `hector_quadrotor_description`
- `hector_quadrotor_teleop`

- `hector_quarotor_gazebo`
- `hector_quarotor_gazebo_plugins`

Se pueden encontrar varios tutoriales en la Wiki de ROS, entre ellos:

- Demostración de vuelo en quadrotor al aire libre [10]. En este tutorial se describe cómo simular un UAV quadrotor volando en un escenario al aire libre usando el paquete *hector_quadrotor*.
- Demostración SLAM interior de quadrotor [9].

4.1. Compilación de demostración de vuelo de quadrotor al aire libre: Paso a paso

Para esto se necesitan los paquetes: `hector_quadrotor`, `hector_localization`, `hector_gazebo` y `hector_models` para ejecutar esta demostración.

```
sudo apt-get update
sudo apt-get install ros-kinetic-hector-quadrotor-demo
```

Instalar desde la fuente:

Si se desea compilar desde el origen para editar los archivos de configuración o de origen para satisfacer sus necesidades, todos los paquetes requeridos se pueden encontrar en el repositorio de GitHub [2].

```
mkdir ~/hector_quadrotor_tutorial
cd ~/hector_quadrotor_tutorial
wstool init src wstool init src
https://raw.githubusercontent.com/tu-darmstadt-ros-pkg/
hector_quadrotor/kinetic-devel/tutorials.rosinstall
```

Instalación de paquetes adicionales requeridos:

Quizás tenga que instalar además algunos paquetes. (Sustituya `kinetic` por la versión de ROS que utilice).

```
sudo apt-get install ros-kinetic-geographic-info
sudo apt-get install ros-kinetic-ros-control
sudo apt-get install ros-kinetic-gazebo-ros-control
sudo apt-get install ros-kinetic-joy
sudo apt-get install ros-kinetic-teleop-twist-keyboard
```

Compilación del workspace:

```
catkin_make
source devel/setup.bash
```

Ejecutar Gazebo y RVIZ:

```
roslaunch hector\_quadrotor\_demo outdoor\_flight\_gazebo.launch
```

Control:

Dependiendo de la versión del paquete que utilice, es posible que primero deba habilitar los motores a través de un servicio:

```
rosservice call /enable\_motors "enable: true"
```

Por otro lado, si se quiere controlar el quadrotor por teclado, se puede hacer con:

```
roslaunch teleop\_twist\_keyboard teleop\_twist\_keyboard.py
```

5. Posibles aplicaciones de UAVs en sistemas de enjambre

Las aplicaciones de este tipo de tecnología son virtualmente infinitas. No solo pueden adaptarse una buena cantidad de tareas actuales a estos sistemas si no que también se están planteando nuevas aplicaciones pensadas en un inicio para actuar en forma de enjambre como se verá a continuación. Por supuesto no se van a repasar todas, solo algunas que se cree son innovadoras, curiosas o con mucho potencial.

La planificación de lucha de incendios forestales y urbanos [3]. Gracias al avance de los últimos años en los sistemas de enjambre, a la posibilidad de emplear equipos heterogéneos dentro de los enjambres, la posibilidad de acometer diferentes tareas de manera simultánea, etc, convierte a los enjambres de drones en una herramienta tecnológica de alto valor añadido en trabajos relacionados con la lucha contra el fuego.

Otra aplicación interesante es la *vigilancia, control y mantenimiento de campos solares* [13]. Donde se integran enjambres en un sistema de control con el objetivo de mejorar los procesos de integración de la energía solar en los sistemas que suministran energía.

Pero también se puede encontrar aplicaciones en campos como el de la *construcción mediante robots aéreos*, es de especial relevancia el trabajo llevado a cabo en los últimos años por el Instituto Técnico de Zurich (ETH). Por un lado, se ha demostrado la viabilidad de levantar puentes colgantes de cuerdas entre dos puntos separados por unos 10 m. Un grupo de tres cuadricópteros van tejiendo el puente entre dos estructuras tubulares a ambos lados de forma completamente autónoma, haciendo nudos con las cuerdas para que la estructura resultante sea segura. Una vez finalizado, se comprueba que el puente es capaz de aguantar el peso de una persona, que puede pasar a través de él con seguridad. Por otro lado, también se propone la construcción de estructuras cerradas compuestas por bloques similares a los ladrillos. Cuatro cuadricópteros vuelan a una zona de recogida de los ladrillos y los colocan ordenadamente para construir la fachada de un edificio de 6 metros de altura. [1].

En el *entorno espacial*, el despliegue de sistemas multirobot ha sido propuesto para múltiples tareas. Dichos sistemas espaciales presentan las mismas ventajas e inconvenientes que sus homólogos aéreos, pero con el problema añadido de su alto coste de despliegue, aunque actualmente se han realizado despliegues de decenas e incluso cientos de satélites con un mismo lanzador. Algunos trabajos incluso proponen el uso de centenares o incluso miles de femtosatélites (con un peso menor de 100 gramos) que podrían ser usados para crear sensores de un tamaño mucho más grande que el que se conseguiría con un solo satélite, crear redes de comunicaciones (proyecto actualmente en marcha por SpaceX denominado Starlink, Figura [5]), o como redes de sensores distribuidos. [1]



Figura 5: Enjambre de satélites Starlink desde la tierra

Otra aplicación relativamente famosa es el del transporte de una carga. No en un contexto de toneladas, si no de pequeñas mercancías. Un claro ejemplo es Amazon con su servicio Prime Air. Ahora bien, si la carga es demasiado pesada para un solo vehículo aéreo, o se desea disponer de cierta robustez en el transporte, el uso de más de un vehículo puede ser necesario [1]. Existen algunos trabajos muy interesantes (Sreenath, K. & Kumar, V. , 2013) [12] donde se trata de resolver el problema de transportar un cuerpo rígido suspendido por cables

mediante n cuadricópteros.

Aunque este último sea el que quizás tenga menos relevancia en el aspecto de avance y contribución social, también es interesante la integración de esta tecnología en el *aspecto visual y de ocio: los espectáculos visuales con drones*. En estas exhibiciones, como la de Tokio 2020 que se puede ver en la Figura [6], que se han hecho virales en los últimos 5 años, un conjunto de cientos de drones equipados con leds se mueven de forma coordinada para formar imágenes en el cielo que son visible desde kilómetros de distancia. Debido a que estos sistemas son desarrollados y explotados por empresas privadas, no hay mucha información disponible sobre como se consiguen coordinar tales cantidades de agentes ni como se posicionan en el espacio con tanta precisión. [1]

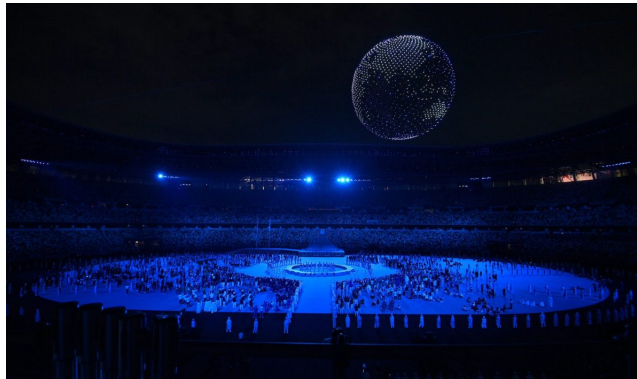


Figura 6: Espectáculo con enjambre de drones en las olimpiadas Tokio 2020

Por supuesto hay empresas dedicadas a esta tecnología y que han conseguido hacer avances interesantes y puestas en marcha de aplicaciones muy útiles. Por ejemplo la empresa **Drone Hopper S.L** [3] cuya labor se destina al diseño y fabricación de drones de alta capacidad de carga y autonomía destinados a la lucha contra el fuego han provocado que dichas plataformas aéreas se posicionen como una potente y eficaz herramienta en el campo de las emergencias y la seguridad. Un proyecto muy interesante que desarrollaron durante 2020 fue **Wild Hopper** capaz de trasladar hasta 600 litros de carga útil y realizar maniobras eficaces en labores de extinción de incendios, gracias, en gran parte, a su sistema patentado de liberación de líquidos.

Embention, que cuenta con 15 años de experiencia en la industria de los vehículos autónomos aéreos ha desarrollado formaciones de vuelo para distintos casos, como puede ser una formación en cuadrícula o línea; para el reconocimiento, mapeo y/o misiones de búsqueda, o la formación de acompañamiento y escolta; para brindar apoyo a un piloto de combate sin interrumpir su movimiento y proporcionándole apoyo de carga útil, etc, o una formación de perímetro; con la cual se puede configurar un perímetro de vigilancia en cuestión de minutos (según la empresa).

Un caso interesante también es el de la empresa española **Escribano Mechanical & Engineering**, quienes han diseñado un sistema de enjambre de drones con inteligencia artificial que es capaz de realizar misiones complejas de salvamento, vigilancia costera, recolección de inteligencia, etc, sin intervención humana. Este programa es vanguardista, no solo para España sino también dentro de la Unión Europea, porque incorporará inteligencia artificial, lo que habilitará al “enjambre” a tomar sus propias decisiones.

6. Conclusiones

Con la robótica abriéndose camino en todos los ámbitos de nuestras vidas cada vez con más fuerza, es inevitable que los sistemas multirobot acaben por ser parte normal en cualquier entorno laboral. Se ha visto, no solo el marco teórico de como funcionan y se programan estos sistemas para drones, si no también un abanico de aplicaciones y empresas que están ofreciendo sistemas para una buena cantidad de situaciones. Si se combinan estas redes de drones con inteligencia artificial como ya esta haciendo la empresa **Escribano Mechanical & Engineering**, las posibilidades se vuelven todavía mayores. En un futuro no muy lejano, probablemente, la sociedad este completamente adaptada a trabajar con ellos.

Referencias

- [1] Ángel Barrientos. “Aplicaciones prácticas de los sistemas multi-UAV y enjambres aéreos”. En: *Centro de Automática y Robótica (CAR), Universidad Politécnica de Madrid* (2021).
- [2] basavarajnavalgund. *hector-quadrotor*. [Online; accessed 13. Jan. 2022]. Ene. de 2022. URL: <https://github.com/basavarajnavalgund/hector-quadrotor>.
- [3] Ángel Madridano Carrasco. “Arquitectura de software para navegación autónoma y coordinada de enjambres de drones en labores de lucha contra incendios forestales y urbanos”. En: *Universidad Carlos III de Madrid* (nov. de 2020).
- [4] Nuria Martínez & Esther Vera David Mataix Victoria Frutos. “Introducción a los drones en ROS”. En: *Universidad de Alicante* ().
- [5] Ángel Madridano Abdulla Al-Kaff David Martín Arturo de la Escalera & J.M. Armingol. “Método de planificación de trayectorias múltiples para enjambre de UAVs”. En: *Laboratorio de sistemas inteligentes (LSI), Universidad Carlos III de Madrid* (2019).
- [6] Daniel Ruiz Restrepo & Pedro Felipe Rizo Gonzales. “Planeación de trayectorias para un robot aéreo AR.Drone 2.0 usando GPS”. En: *Departamento de Electrónica, Facultad de ingeniería de la Pontificia Universidad Javeriana de Bogotá* (2014).
- [7] Agustín Castro & Sofia Salmini Guillermo Cabrera. “ENDRA: Enjambre de Drones Autónomos”. En: *Universidad de la República* (jun. de 2021).
- [8] *hector-quadrotor - ROS Wiki*. [Online; accessed 13. Jan. 2022]. Ene. de 2022. URL: <http://wiki.ros.org/hector-quadrotor>.
- [9] *hector-quadrotor/Tutorials/Quadrotor indoor SLAM demo - ROS Wiki*. [Online; accessed 14. Jan. 2022]. Ene. de 2022. URL: <http://wiki.ros.org/hector-quadrotor/Tutorials/Quadrotor%5C%20indoor%5C%20SLAM%5C%20demo>.
- [10] *hector-quadrotor/Tutorials/Quadrotor outdoor flight demo - ROS Wiki*. [Online; accessed 13. Jan. 2022]. Ene. de 2022. URL: <http://wiki.ros.org/action/fullsearch/hector-quadrotor/Tutorials/Quadrotor%5C%20outdoor%5C%20flight%5C%20demo?action=fullsearch&context=180&value=linkto%5C%3A%5C%22hector-quadrotor%5C%2FTutorials%5C%2FQuadrotor+outdoor+flight+demo%5C%22>.
- [11] Eros Piera & Jaume Díaz Israel Alcañiz. “Robots Voladores. Drones”. En: *Universidad de Alicante* ().
- [12] Vijay Kumar Kousil Sreenath. “Dynamics, Control and Planning for Cooperative Manipulation of Payloads Suspended by Cables from Multiple Quadrotor Robots”. En: *Department of Mechanical Engineering and Applied Mechanics, University of Pennsylvania Philadelphia* (2013).

- [13] Jenifer Delgado Rich. “Integración de enjambre de drones en sistemas de control de campos solares”. En: *Escuela Técnica Superior de Ingeniería, Universidad de Sevilla* (2021).
- [14] Mireia Luisa Sempere Tortosa. “Agentes y enjambres artificiales: modelado y comportamiento para sistemas de enjambre robóticos”. En: *Universidad de Alicante* (2013).