

Trabajo Fin de Máster

Máster en Ingeniería Electrónica, Robótica y Automática

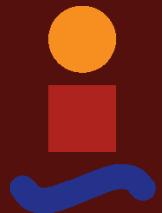
Aerial co-workers: a task planning approach
for multi-drone teams supporting inspection
operations

Autor: Álvaro Calvo Matos

Tutor: Jesús Capitán Fernandez

Dpto. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2021



Trabajo Fin de Máster
Máster en Ingeniería Electrónica, Robótica y Automática

Aerial co-workers: a task planning approach for multi-drone teams supporting inspection operations

Autor:
Álvaro Calvo Matos

Tutor:
Jesús Capitán Fernández
Associate Professor

Dpto. Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2021

Trabajo Fin de Máster: Aerial co-workers: a task planning approach for multi-drone teams supporting inspection operations

Autor: Álvaro Calvo Matos
Tutor: Jesús Capitán Fernandez

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:

Índice

1	Introducción	1
1.1	Motivación	1
1.2	Objetivos	2
2	Formulación del problema	3
2.1	Descripción de las tareas	4
2.1.1	Tareas de inspección	4
2.1.2	Tareas de monitorización	4
2.1.3	Tareas de entrega de herramienta	5
2.2	Recargas de batería	6
2.3	Connection losses	6
2.4	Situaciones de replanificación de tareas	6
3	Diseño de la solución propuesta	7
3.1	Diagrama de bloques	7
3.2	Módulo centralizado: Planificador de alto nivel	8
3.3	Módulo distribuido: Gestor de comportamiento	9
3.3.1	Árbol principal	10
3.3.2	Inspection task tree	10
3.3.3	Monitoring task tree	11
3.3.4	Tool delivery task tree	11
4	Results	13
5	Conclusiones and trabajo futuro	15
5.1	Conclusiones	15
5.2	Trabajo futuro	16

1 Introducción

El uso de los vehículos aéreos no tripulados (Unmanned Aerial Vehicle (UAV)) ha crecido considerablemente en los últimos años para numerosas aplicaciones civiles, como la supervisión en tiempo real, la búsqueda y el rescate, la provisión de cobertura inalámbrica, la seguridad y la vigilancia, la agricultura de precisión, la entrega de paquetes y la inspección de infraestructuras. Con el rápido desarrollo de la tecnología en este ámbito, y las demostraciones de lo que pueden hacer los UAV, cada vez se hacen más esfuerzos para llevar esta tecnología a otras aplicaciones. Con el aumento previsto de las aplicaciones de esta tecnología, surgen nuevos problemas y desafíos, como la autonomía, la seguridad, la evitación de obstáculos y la coordinación de equipos multi-UAV. El desarrollo de la tecnología para resolver estos problemas supone un gran esfuerzo, pero como los UAV han demostrado ser fundamentales en situaciones en las que los humanos corren un gran riesgo o son muy ineficaces, y han demostrado su capacidad para evolucionar y desarrollar aún más su potencial a corto plazo, las empresas están invirtiendo en el desarrollo de todo tipo de soluciones basadas en los UAV.

1.1 Motivación

Con el aumento de la demanda mundial de electricidad, ha surgido el reto para las empresas de suministro eléctrico que consiste en mantener y reparar las redes eléctricas de forma que se minimice la frecuencia de los cortes. Una de las principales causas de los cortes de electricidad son los daños en las líneas de transmisión debidos al mal tiempo o a campañas de inspección ineficaces.

La estrategia que suelen utilizar las compañías eléctricas para reducir los cortes de energía es programar operaciones periódicas de mantenimiento en las líneas activas. Este es el método más adecuado si se quiere asegurar el correcto funcionamiento del sistema y cuando la sustitución de un circuito es inaceptable. Estas misiones de mantenimiento son realizadas por tripulaciones experimentadas a bordo de helicópteros y equipadas con trajes de seguridad y arneses, entre otras cosas, que evitan que los operarios reciban una descarga eléctrica. El problema de esta solución es que estas actividades son peligrosas para los operarios, ya que trabajan a gran altura y en líneas electrificadas, consumen mucho tiempo, son muy caras (1.500 dólares por hora) y están sujetas a errores humanos.

Estas son las razones por las que las empresas de distribución tienen la necesidad de desarrollar métodos de mantenimiento más eficientes y seguros. Se han propuesto múltiples soluciones para automatizar esta tarea, pero la mejor parece ser el uso de UAVs, debido a su flexibilidad y capacidad para inspeccionar a diferentes niveles. Para ello, todavía hay que superar algunas barreras importantes, como la limitada autonomía de estos aparatos, las fuertes interferencias electromagnéticas a las que estarían sometidos por estar cerca de las líneas eléctricas, y la capacidad de detectar y evitar los obstáculos de distinta naturaleza que se podrían encontrar en este tipo de entornos. Dotar

a los UAVs de la capacidad cognitiva necesaria para operar de forma autónoma en entornos tan dinámicos y con presencia humana, y dotarles de un método de planificación rápida en línea, es clave para hacer frente a estas complejidades y cumplir con seguridad y éxito la misión asignada con las flotas de UAV.

Una arquitectura de software versátil y fiable es esencial para integrar e interconectar todos los componentes heterogéneos que componen estos sistemas cognitivos multi-UAV. En el proyecto europeo AERIAL-CORE se presenta una arquitectura de software multicapa para llevar a cabo este tipo de misiones de forma cooperativa entre operadores humanos y una flota de quadrotors. Uno de los componentes de software implicados es un planificador de tareas de alto nivel. Su función es coordinar toda la flota de UAVs para generar comportamientos de alto nivel con el fin de completar de forma eficiente, segura y exitosa la misión de mantenimiento o inspección. Este tipo de trabajos tienen la característica de ser dinámicos, ya que no es posible conocer de antemano cuál será el resultado de la inspección como tal para planificarla fuera de línea, sino que, a medida que se desarrolla la misión, surgirán nuevas tareas que la flota deberá atender. Por lo tanto, el planificador de tareas debe ser capaz de reaccionar ante eventos inesperados (nuevas tareas, fallo de un UAV, pérdida de conexión, menos autonomía de la calculada, etc.) y volver a planificar en línea. Así, este planificador de alto nivel será el principal bloque cognitivo del sistema.

1.2 Objetivos

El objetivo general de este proyecto es desarrollar un planificador cognitivo de tareas encargado de gobernar el comportamiento de equipos multi-UAV para la inspección y mantenimiento de líneas eléctricas de forma colaborativa con operadores humanos, siendo una de las capas de software que componen la mencionada arquitectura de software desarrollada para el proyecto europeo AERIAL-CORE. La flota de UAVs gobernada actúa como co-trabajadores aéreos y puede realizar diversas tareas, como entregar una herramienta a un operario, inspeccionar regiones de la línea eléctrica o vigilar a un trabajador mientras opera para garantizar su seguridad. El planificador recibe tanto información de alto nivel como de las distintas plataformas que componen la flota, y procesa toda la información para elaborar un plan que permita gestionar el equipo de UAV o modificarlo como reacción a un imprevisto. Para ello, se definieron los siguientes objetivos:

- Garantizar la utilización de los recursos y la ejecución eficaz de las tareas.
- Cumplir con todos los requisitos de seguridad y garantizar la integridad de las plataformas aéreas y el éxito de la misión.
- Ser capaz de volver a planificar en línea para reaccionar ante acontecimientos imprevistos.
- Implementar la capa de software en Robot Operating System (ROS) y gestionar la comunicación necesaria con el resto de capas y módulos de software que conforman la arquitectura completa.
- Realizar simulaciones de software en el bucle (Software In The Loop (SITL)) para demostrar que el algoritmo es capaz de gobernar el comportamiento de la flota de forma eficiente y segura, y que es capaz de reaccionar ante imprevistos de forma dinámica, demostrando capacidades cognitivas.
- Diseñar el planificador de tareas de forma que sea fácil de mantener, modificar o ampliar, buscando que sea modular y reutilizable para que pueda servir de base para la construcción de planificadores para otras aplicaciones.

2 Formulación del problema

El proyecto AERIAL-CORE pretende desarrollar diferentes tecnologías para el uso de sistemas multi-UAV en tareas de inspección y mantenimiento en instalaciones eléctricas de alta tensión. En concreto, una de las tecnologías propuestas es el uso de co-trabajadores aéreos (Aerial Co-Worker), es decir, pequeños equipos de UAVs cooperativos para apoyar de forma segura a los trabajadores de mantenimiento mientras trabajan en altura en las líneas eléctricas (véase la figura 2.1).



Figura 2.1 Equipo multi-UAV apoyando a un trabajador. Fuente: Página web de Aerial-Core.

Se hace referencia a tres tipos de ACWs, cada uno destinado a proporcionar una funcionalidad diferente: *Inspection-ACW*, *Safety-ACW*, y *Physical-ACW*. Aunque existe un tipo de ACW específico para cada una de las tareas (inspección, monitorización y entrega de herramientas), esto no significa que un UAV pueda realizar en un momento dado una tarea para la que no es el mejor.

Este problema de planificación de misiones con múltiples UAVs con limitaciones de batería puede plantearse como un problema de optimización, cuya solución indica la forma más eficiente de asignar las diferentes tareas y planificar las recargas. Para reaccionar ante posibles fallos, una de las opciones más extendidas es la de plantear métodos dinámicos que puedan replanificar en tiempo real cuando se produzcan determinados eventos.

En este contexto se desarrolló el planificador de tareas cognitivas de esta tesis. Como este planificador cognitivo es un módulo que forma parte de una arquitectura de software más grande para abordar todo el problema, se presenta brevemente la imagen completa de esa arquitectura.

2.1 Descripción de las tareas

El planificador recibe comunicaciones asíncronas desde la capa superior con las especificaciones de nuevas tareas y se encarga de procesar la nueva información junto con la que ya tenía para elaborar y ejecutar un nuevo plan. También se encarga de llamar a los controladores de bajo nivel cuando es necesario y de garantizar la seguridad del UAVs y el cumplimiento de la misión. Cada tarea se explica en detalle en los siguientes apartados.

2.1.1 Tareas de inspección

Esta tarea puede ser realizada por los tres tipos de ACWs. Es la segunda tarea más prioritaria, siendo la tarea de entrega de herramientas la única que la supera. Consiste en realizar una inspección detallada de las zonas especificadas de las líneas de alta tensión (ver Fig. 2.2). La capa inmediatamente superior al planificador de tareas se encarga de pasarle una lista de waypoints (WPs) que definen la tarea de inspección, y el planificador se encarga de decidir cuántos ACWs recluta para ejecutar la tarea y a cuáles de los ACWs disponibles se la asigna. Dividir la lista total de WP a inspeccionar en subconjuntos y asignar cada uno a uno de los ACWs seleccionados para la tarea es el trabajo de un controlador de bajo nivel. Por tanto, una vez ejecutada la planificación, las tareas de este tipo se transmiten a las capas inferiores con la lista total de WPs a inspeccionar y una lista con las identificaciones (identifications (IDs)) de los UAVs seleccionados.

Todas las comunicaciones mencionadas se realizarán de forma asíncrona, ya que la creación de la tarea por parte de los trabajadores, que desencadena toda la secuencia de acciones, se realiza de esta forma.

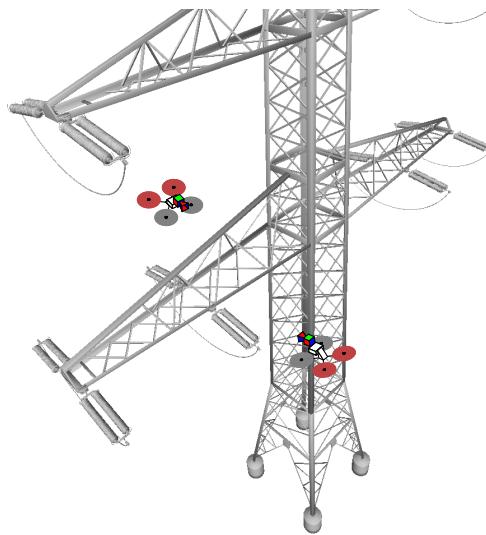


Figura 2.2 *Inspection-ACW* llevando a cabo una tarea de inspección.

2.1.2 Tareas de monitorización

Esta tarea también puede ser ejecutada por los tres tipos de ACWs. Es la tarea de menor prioridad. Monitorizar la seguridad de los trabajadores consiste en proporcionar al equipo de supervisión una visión de las personas que trabajan en la torre de energía para controlar su estado y garantizar su seguridad (ver Fig. 2.3). La capa inmediatamente superior al planificador de tareas comunica en esta ocasión el ID del trabajador a vigilar, el número de UAVs deseado y la distancia que deben mantener con el trabajador. Es responsabilidad del planificador de la tarea decidir de nuevo a cuál de los ACWs disponibles asignar a esta tarea y la formación que deben mantener durante el vuelo.

Una vez realizada la planificación, las tareas de este tipo se transmiten a los estratos inferiores tanto con la información original como con la resultante de la planificación.

Las comunicaciones mencionadas también se realizarán de forma asíncrona por el mismo motivo.

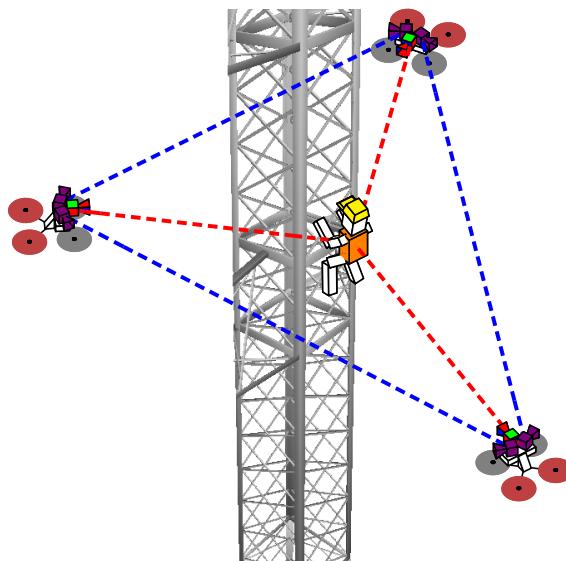


Figura 2.3 Safety-ACW llevando a cabo una tarea de monitorización.

2.1.3 Tareas de entrega de herramienta

Esta tarea sólo puede ser realizada por UAVs de tipo *Physical-ACW*, ya que se requiere un hardware especial para realizar la interacción física con los objetos de bajo peso y el humano. Esta es la tarea más prioritaria. La entrega de una herramienta consiste en recoger una herramienta y transportarla hasta el trabajador, con el que se producirá una interacción física a través de la cual se realizará la entrega de la herramienta (véase la Fig. 2.4). Los controladores de bajo nivel tendrán que ser especialmente precisos y cuidadosos para no dañar al trabajador. Esta vez, la capa inmediatamente superior al planificador de tareas comunica el ID del trabajador al que hay que entregar la herramienta y el ID de la herramienta solicitada. De nuevo, la misión del planificador de tareas es decidir a cuál de los ACWs disponibles asignar esta tarea. Una vez realizada la planificación, las tareas de este tipo pasan a las capas de nivel inferior con la misma información que en un principio.

Las mencionadas comunicaciones, una vez más, se realizarán de forma asíncrona.



Figura 2.4 Physical-ACW llevando a cabo una tarea de entrega de herramienta.

2.2 Recargas de batería

Eventualmente cada uno de los ACWs que participan en la misión se quedará sin batería. El planificador puede tenerlo en cuenta a la hora de distribuir las tareas para que el propio ACW se anticipa a este evento. La recarga no tiene por qué producirse cuando el UAV esté a punto de quedarse sin batería, ni tampoco hasta que ésta llegue al máximo, por lo que ambos serán parámetros a tener en cuenta durante el proceso de planificación y optimización de la misión.

Además, es posible que los cálculos fallen por algún motivo y la batería se agote antes de lo previsto. Por lo tanto, será necesario leer periódicamente el estado de la batería y realizar una recarga de emergencia y una replanificación si es necesario, reaccionando ante fallos inesperados. Debería haber un módulo de comprobación de la batería a bordo de cada vehículo aéreo, y un protocolo de emergencia en caso de que esto ocurra.

2.3 Connection losses

Otra consideración importante es la posible pérdida de conexión entre el planificador centralizado, donde se concentra la mayor parte de la capacidad cognitiva, y alguno de los ACWs. Esta es una situación potencialmente peligrosa, ya que el UAV desconectado podría actuar de forma autónoma según su último plan y provocar un accidente con el resto de agentes que aún están en línea. Por ello, es importante: (i) implementar un sistema de detección de desconexiones desde ambos lados de la comunicación, y (ii) establecer un protocolo de actuación común para que ambos módulos sepan cómo va a actuar el otro, garantizando así la integridad de todos los vehículos y la seguridad de los trabajadores

2.4 Situaciones de replanificación de tareas

Una vez que la misión está en marcha, cualquier acontecimiento imprevisto tiene el potencial de cambiar por completo cuál es el plan óptimo. Por lo tanto, aunque exista la posibilidad de que el evento no afecte en absoluto a la misión, siempre será necesario ejecutar una replanificación de la misión en caso de que se produzca un imprevisto.

A continuación se enumeran los imprevistos que se han contemplado en este trabajo:

- Llegada de una nueva tarea.
- Modificación de los parámetros de una tarea.
- Conexión de un nuevo ACW.
- Desconexión de un ACW.
- Batería insuficiente imprevista en uno de los ACWs.
- La batería de algún ACW se consume más rápido de lo esperado y por lo tanto no será suficiente para el plan actual.
- Un ACW termina de recargar antes de lo esperado.
- Una tarea termina exitosamente.
- Una tarea termina no exitosamente.

3 Diseño de la solución propuesta

La solución propuesta sigue un enfoque jerárquico, con un planificador de alto nivel encargado de activar diferentes controladores de bajo nivel. El planificador de alto nivel detecta las tareas requeridas por los operadores, y las distribuye desde tierra de forma centralizada entre los ACWs disponibles, planificando las recargas necesarias a lo largo de la misión. Además, este planificador reacciona en tiempo real ante posibles eventos reasignando las tareas. Los planificadores de bajo nivel están a bordo de cada UAV y se encargan de ejecutar los planes de contingencia para estos eventos mientras el planificador central calcula y comunica el nuevo plan. También se encargarán de controlar el movimiento del ACWs para ejecutar las diferentes tareas asignadas por el módulo de nivel superior.

3.1 Diagrama de bloques

La figura 3.1 muestra un esquema de la arquitectura software desde la perspectiva del módulo implementado en esta tesis, incluyendo los diferentes bloques y sus interfaces. La capa de software correspondiente a esta tesis, encargada de la toma de decisiones de alto nivel, está marcada en azul-verde. Está compuesta por el *Planificador de alto nivel*, que está centralizado y se ejecuta en una estación terrestre (en naranja) y el *Gestor de comportamiento*, distribuido a bordo de cada ACW (en color lima).

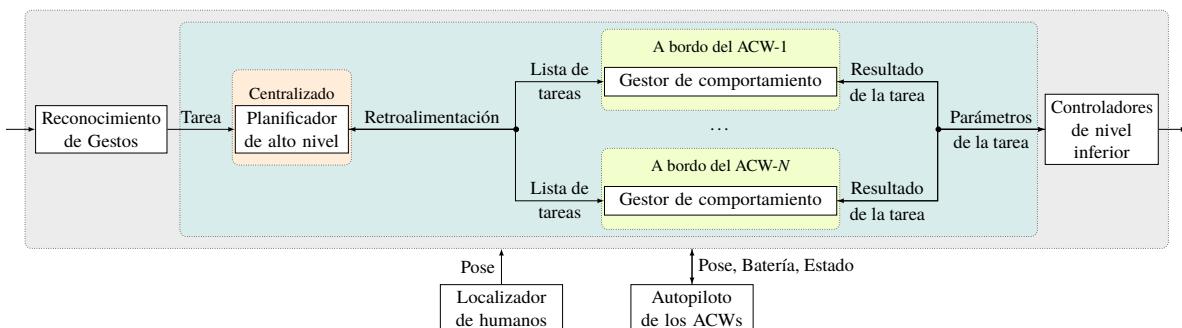


Figura 3.1 Arquitectura de software: bloques e interfaces. Diagrama de bloques desde la perspectiva del planificador de tareas cognitivas de alto nivel.

El primer módulo comprueba constantemente las imágenes captadas por el UAVs en busca de un gesto que indique una nueva tarea o la modificación de una tarea existente. Cuando esto ocurre, envía asíncronamente una tarea, que será recogida por el planificador centralizado. El *Planificador de alto nivel*, cuando recibe esta información, procede a reevaluar el plan óptimo teniendo en cuenta la tarea recibida, la información que recibe de los *Autopilotos de los ACWs*, y la posición de los

operarios, que es publicada periódicamente por el *Localizador de humanos*. A bordo de cada ACW hay un *Gestor de comportamiento* que recoge la correspondiente lista de tareas proporcionada por el planificador centralizado, la información procedente del *Localizador de humanos* y la del *Autopiloto del ACW* y se encarga de llamar a los *Controladores de nivel inferior* para llevar a cabo la ejecución del plan asignado. La información emitida por el *Autopiloto del ACW* se utiliza también para comprobar que todo funciona correctamente y para ejecutar los protocolos de seguridad en caso de que sean necesarios. Si esto ocurriera, se emitiría la correspondiente comunicación de vuelta al *Planificador de alto nivel* para calcular un nuevo plan. Estos módulos también reciben el resultado de los *Controladores de nivel inferior* después de llamar a cada uno de ellos, y lo publican de vuelta al *Planificador de alto nivel* como retroalimentación. Además de estas comunicaciones, los módulos *Planificador de alto nivel* y *Gestor de comportamiento* de agente intercambian periódicamente balizas que sirven para detectar tanto la conexión de un nuevo ACW como su desconexión en caso de fallo.

3.2 Módulo centralizado: Planificador de alto nivel

El *Planificador de alto nivel* es un módulo centralizado que se ejecuta en una estación terrestre y constituye el principal módulo cognitivo de la arquitectura de software. Su objetivo es planificar la misión de forma óptima teniendo en cuenta el tiempo que se tarda en completar cada una, el tipo de cada UAVs, la distancia que tendrá que recorrer cada uno, la batería que tienen disponible, la tarea que estaba ejecutando cada uno, la prioridad de cada tarea, la batería consumida por cada tarea, las recargas que serán necesarias, y cuándo es mejor realizar esas recargas.

El pseudocódigo general de este componente, desde el lanzamiento hasta la terminación, está representado en el código 3.1.

Salida 3.1 General operation of *High-Level Planner's* code.

1. Leer de un ros::param la dirección del archivo de configuración.
2. Leer del archivo de configuración toda la información necesaria.
3. Configurar las comunicaciones ROS (Publishers, Subscribers y ActionServers).
4. Configurar la frecuencia de bucle.
5. Bucle "while" principal". Mientras que ros::ok() y no misión terminada hacer:
 - 5.1. Comprobar el tiempo transcurrido de las balizas de los Agentes.
 - 5.2. Publicar una nueva baliza del Planificador.
 - 5.3. Comprobar si hay comunicaciones entrantes pendientes (ros::spinOnce).
 - 5.4. Dormir el tiempo restante para enviar la siguiente baliza.
6. Esperar a que todos los UAVs terminen y se desconecten. Mientras que haya algún agente conectado hacer:
 - 6.1. Comprobar el tiempo transcurrido de las balizas de los agentes.
 - 6.2. Comprobar si hay comunicaciones entrantes pendientes (ros::spinOnce).
 - 6.3. Dormir un rato.

3.3 Módulo distribuido: Gestor de comportamiento

Este componente se encarga de ejecutar el plan asignado por el *Planificador de alto nivel*, comprobando en todo momento la seguridad de los UAVs, detectando los imprevistos y comunicándolos al nodo centralizado para que realice un cambio de planes en caso necesario. El *Gestor de comportamiento* se comunicará con los controladores de bajo nivel, cediendo el control cuando sea necesario para completar el plan asignado.

La estructura general de este módulo es bastante similar a la del módulo central. El pseudocódigo se resume en el Código 3.2. Tras la inicialización, el *Gestor de comportamiento* prepara la información necesaria para comenzar su funcionamiento, configura las comunicaciones necesarias, declara e inicializa el árbol de comportamiento y, una vez que el UAV ha terminado de inicializarse, comienza a enviar balizas al nodo central para notificar que se incorpora a la misión. Una vez que el código termina de inicializarse y llega al bucle while principal, la actividad del *Gestor de comportamiento* se concentra en la ejecución de funciones en respuesta a los mensajes entrantes, como las de el *Planificador de alto nivel*, y en la ejecución del árbol de comportamiento, que dirige y supervisa el movimiento del UAV.

Salida 3.2 Pseudocódigo del bloque *Gestor de comportamiento*.

1. Leer de un ros::param el contenido de la baliza (ID y tipo de ACW).
2. Leer de un ros::param la dirección del archivo de configuración.
3. Leer del archivo de configuración toda la información necesaria.
4. Configurar las comunicaciones de ROS (Publishers, Subscribers y ActionServers).
5. Configurar la frecuencia del bucle.
6. Declarar el árbol de comportamiento.
7. Inicializar cada nodo del BT.
8. Iniciar los registradores del BT para facilitar la depuración y el seguimiento del rendimiento del nodo.
9. Esperar a que el ACW se inicialice completamente.
10. Bucle "while" principal. Mientras ros::ok() y el estado de BT es EN EJECUCIÓN:
 - 10.1. Si no se ha producido un timeout de las balizas del Planificador:
 - 10.1.1. Publicar una nueva baliza del Agente.
 - 10.2. Comprobar si la batería es suficiente para la tarea actual.
 - 10.3. Comprobar si hay comunicaciones entrantes pendientes (ros::spinOnce).
 - 10.4. Dormir el tiempo restante para enviar la siguiente baliza.

Los Behaviour Trees (BTs) son quienes gobiernan a los ACWs para realizar cada una de las tareas asignadas. Cada BT monitoriza el estado de la batería y de las tareas de su ACW y reacciona ante cualquier posible fallo o imprevisto, solicitando una nueva replanificación al *Planificador de alto nivel* en caso de necesidad. Un BT puede definirse como una máquina finita de estados (Finite State Machine (FSM)) mejorada. Son un mecanismo más avanzado para implementar comportamientos, especialmente por sus ventajas en términos de escalabilidad, modularidad, legibilidad y reutilización, facilitando la creación de comportamientos más complejos con menos esfuerzo.

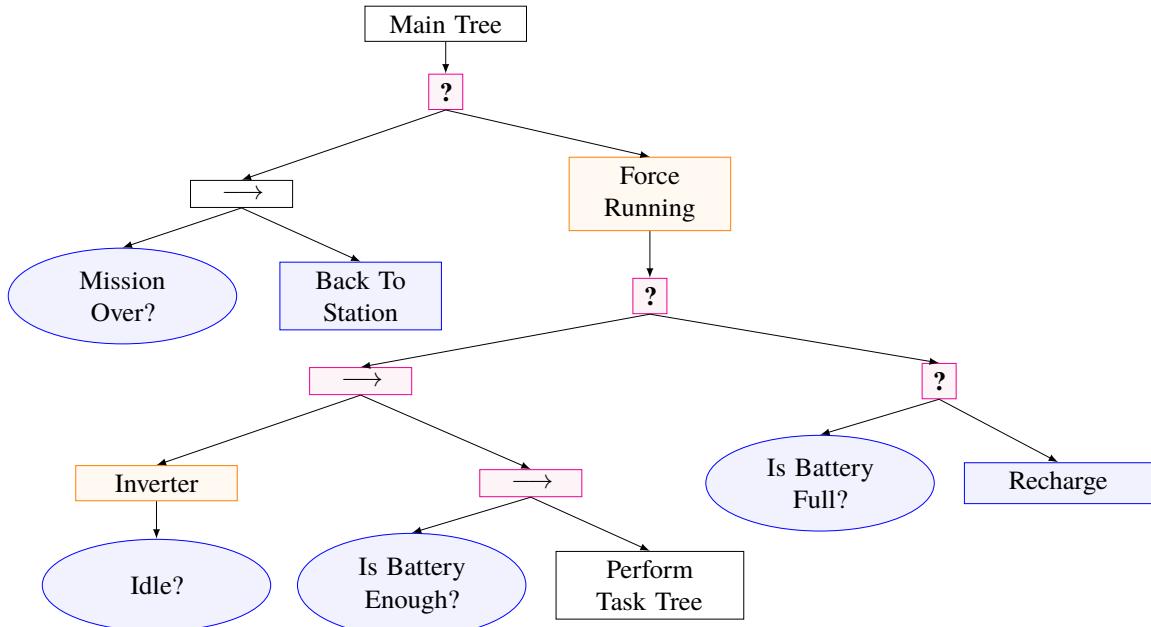


Figura 3.2 Árbol de comportamiento: Árbol principal.

3.3.1 Árbol principal

Este BT comprueba si la misión ha terminado y si es así dirige el ACW de vuelta a la estación base. Si no es así se comprueba si hay alguna tarea asignada. Si resulta que el ACW está ocioso y la batería no está al cien por cien, el ACW es guiado a una estación de recarga (aquí es donde está codificado el protocolo de emergencia). Si se asigna una tarea y la batería es suficiente, se entra en el *Árbol de ejecutar tarea* (subárbol representado en la Figura 3.3).

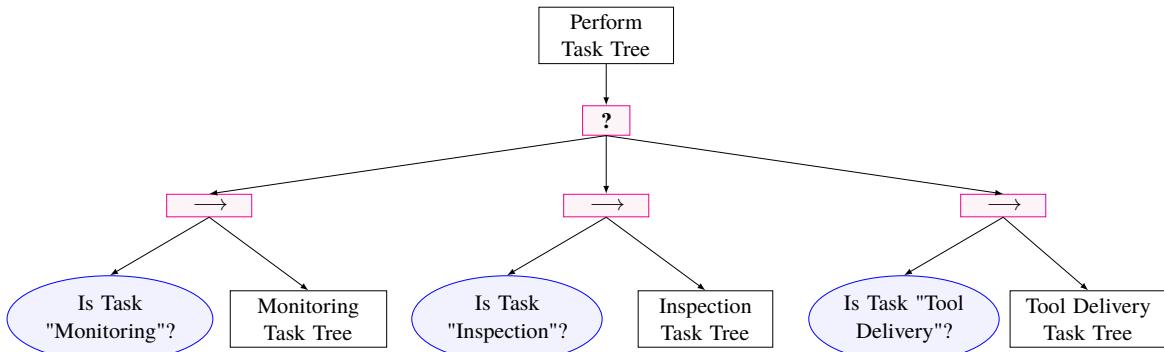


Figura 3.3 Behaviour Tree: Perform Task Tree.

El *Árbol de ejecutar tarea* comprueba cuál es la primera tarea de la cola, que es la que debe ejecutarse en ese momento. En las figuras 3.5, 3.4 y 3.6 se representan los subárboles que ejecutan las tareas. Se decidió que el control total no se daría a los controladores de bajo nivel hasta que el ACW estuviera lo suficientemente cerca de la zona donde se desarrolla la tarea.

3.3.2 Inspection task tree

Este BT es bastante sencillo, ya que la tarea consiste simplemente en visitar una serie de puntos y detenerse en cada uno de ellos para tomar fotografías (véase la 3.4). El BT comprueba si el ACW

está cerca de los puntos a inspeccionar, si es necesario, se ejecuta el nodo para ir cerca de WP y una vez allí, se llama al controlador de bajo nivel que ejecuta esta tarea.

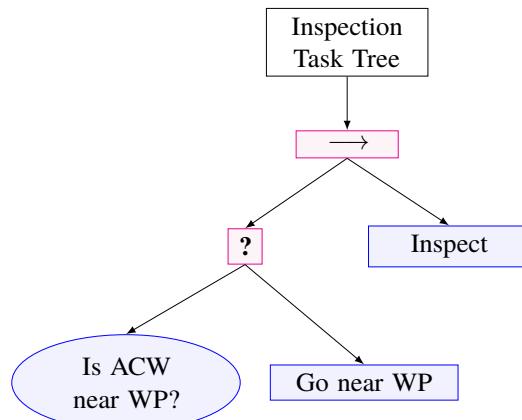


Figura 3.4 Behaviour Tree: sub-tree that controls the inspection tasks.

3.3.3 Monitoring task tree

Como se puede ver en la Figura 3.5, este BT sigue la misma estructura y funcionamiento que el anterior. Las únicas diferencias son los nodos *Hoja*.

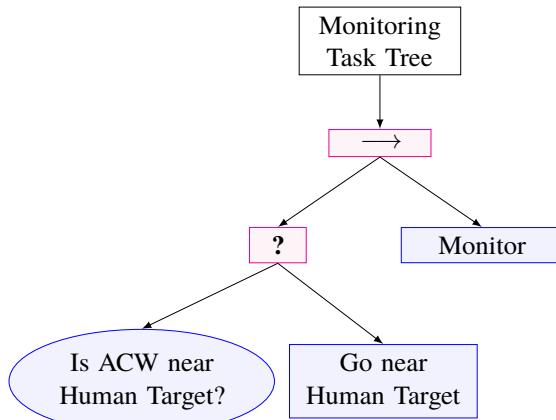


Figura 3.5 Behaviour Tree: sub-tree that controls the safety monitoring tasks.

3.3.4 Tool delivery task tree

Este árbol es un poco más complejo, ya que la tarea implica varios pasos (ver Fig. 3.6). En primer lugar, recoge la herramienta solicitada en caso de que el ACW no la tenga todavía, luego se realiza la aproximación al operario, y por último el controlador de bajo nivel de esta tarea puede ser activado.

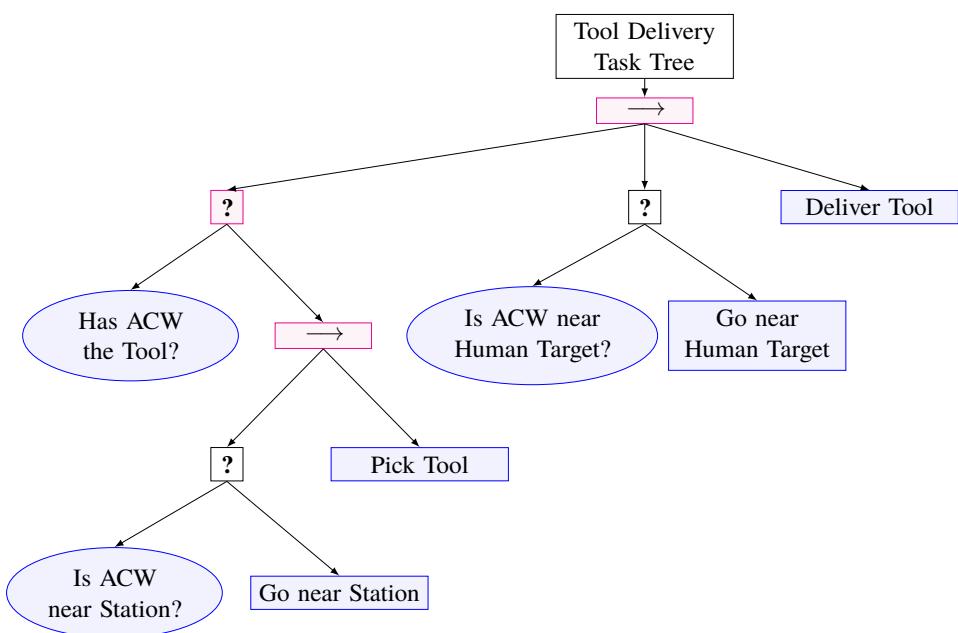


Figura 3.6 Behaviour Tree: sub-tree that controls the tool delivery tasks.

4 Results

En este capítulo se exponen los experimentos realizados para validar la capa de software construida. Los experimentos se dividieron en dos fases. En la primera fase, se realizaron simulaciones con un solo ACW para probar el rendimiento de cada elemento del sistema de forma controlada. Por un lado, se comprobó que el *Planificador de alto nivel* realizaba la planificación de la misión correctamente, asignando las tareas como se esperaba según las especificaciones y restricciones, y por otro lado, se comprobó que el *Gestor de comportamiento* realizaba su función correctamente, tanto la ejecución de las tareas individuales como la capacidad de detectar y actuar en caso de imprevistos. Durante esta fase, la validación del bloque distribuido cobra protagonismo. En la segunda fase, las simulaciones consistieron en incluir múltiples ACWs y realizar pruebas en diferentes escenarios. Esta fase se centra menos en la validación del BT, que estaría totalmente validado durante la primera fase, y más en la evaluación de las capacidades del *Planificador de alto nivel*. Las situaciones a las que se enfrenta el sistema en esta fase implican desconexiones, reconexiones, entrada de nuevas tareas, modificaciones del nivel de batería, etc.

Los BTs funcionaron perfectamente. En Gazebo, se observó el movimiento de los ACW de un lado a otro mientras los BT recorrían los nodos del subárbol de cada tarea (ver Fig. 4.1). Además, se comprobó que el resultado de las tareas se comunicaba correctamente con el bloque centralizado.

Salida 4.1 Mensajes impresos por el Gestor de comportamiento durante la ejecución del plan completo.

```
[ INFO] [/uav_1/agent]: [newTaskList] Received a NewTaskList Action
[ INFO] [/uav_1/agent]: task_1: DeliverTool
[ INFO] [/uav_1/agent]: [Recharge] halt requested
[ INFO] [/uav_1/agent]: [GoNearStation] Moving near Tool...
[ INFO] [/uav_1/agent]: [PickTool] Calling Lower-level controllers...
[ INFO] [/uav_1/agent]: [PickTool] PICK TOOL FINISHED
[ INFO] [/uav_1/agent]: [GoNearHumanTarget] Moving near HT...
[ INFO] [/uav_1/agent]: [DeliverTool] Calling Lower-level controllers
...
[ INFO] [/uav_1/agent]: [DeliverTool] DELIVER TOOL TASK FINISHED (
    true)
[ INFO] [/uav_1/agent]: task_2: Inspect
[ INFO] [/uav_1/agent]: [GoNearWP] Moving near WP...
[ INFO] [/uav_1/agent]: [TakeImage] Calling Lower-level controllers...

[ INFO] [/uav_1/agent]: [TakeImage] INSPECT TASK FINISHED (true)
```

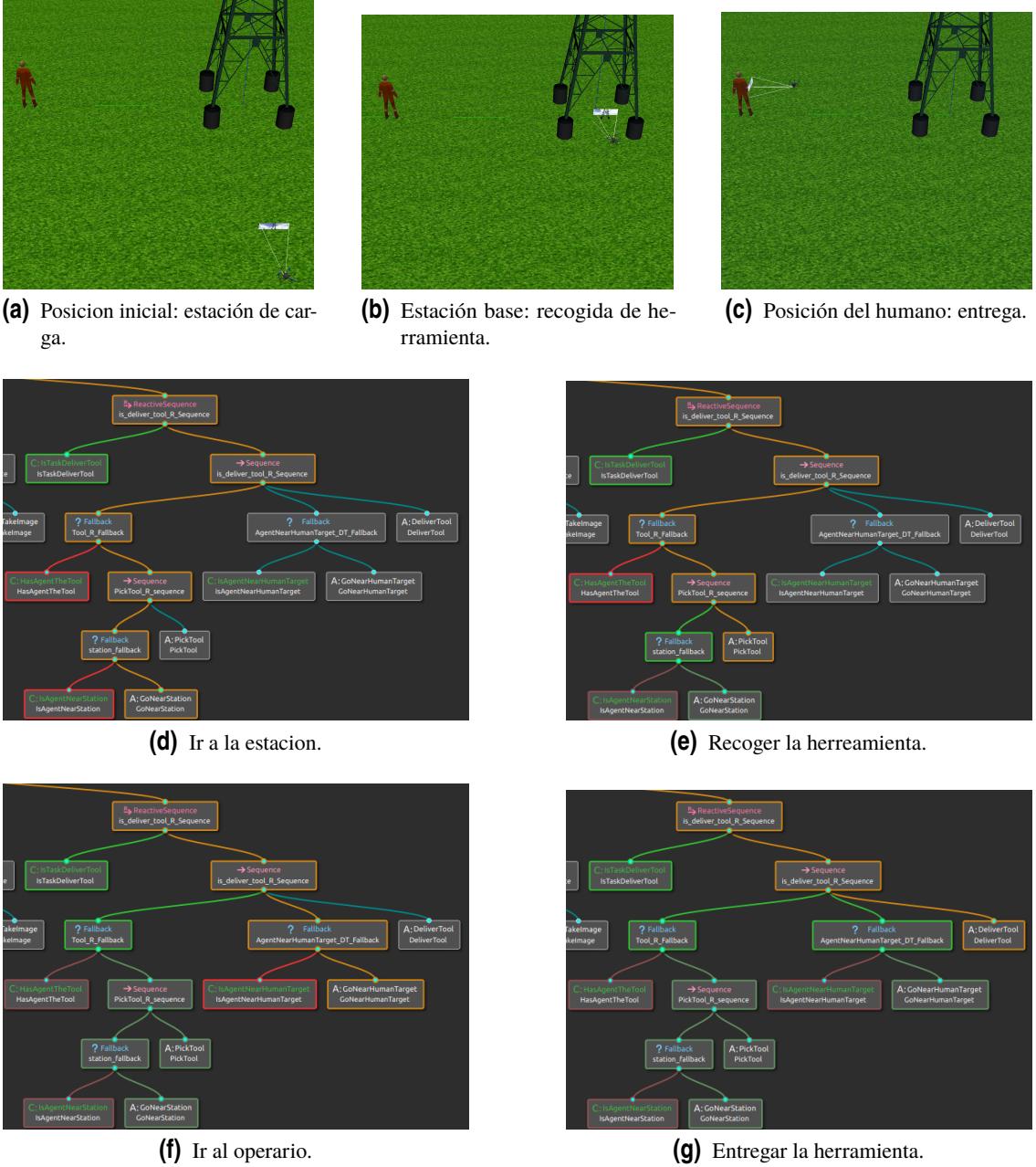


Figura 4.1 Evolución de la simulación durante la ejecución de la tarea de entrega.

```
[ INFO] [/uav_1/agent]: task_3: Monitor
[ INFO] [/uav_1/agent]: [GoNearHumanTarget] Moving near HT...
[ INFO] [/uav_1/agent]: [MonitorHumanTarget] Calling Lower-level
 controllers...
```

Una vez comprobado que el sistema funcionaba bien en ausencia de imprevistos, se probó el sistema en todo tipo de situaciones (capacidad del BT para detener la tarea actual y pasar a la ejecución de la nueva tarea en caso de cambio de planes, llegada de una nueva tarea con un ID repetido, pruebas de caída repentina de la batería, desconexiones, etc). El resultado de todas estas pruebas fue muy positivo. En la segunda fase se puso a prueba al planificador centralizado con cinco UAVs distintos y una gran variedad de situaciones tanto imprevistas como planificadas. Este bloque también superó con éxito todas las pruebas.

5 Conclusiones and trabajo futuro

5.1 Conclusiones

En este trabajo se ha desarrollado un planificador de tareas con capacidad para realizar la planificación de misiones para equipos multi-UAV. El sistema tiene la capacidad cognitiva suficiente para controlar a múltiples UAVs que funcionen como co-trabajadores en entornos dinámicos de forma segura. Las simulaciones realizadas han demostrado la capacidad que tiene el sistema para detectar situaciones de emergencia y actuar de forma segura ejecutando planes de contingencia de forma autónoma mientras se calcula un nuevo plan a seguir que tenga en cuenta los imprevistos que hayan acontecido. El diseño del sistema dividido en un bloque centralizado en tierra encargado de realizar la planificación óptima de la misión y de bloques distribuidos a bordo de cada uno de los ACWs permite que el sistema presente robustez ante fallos y capacidad cognitiva suficiente para reaccionar ante eventos imprevistos recalculando el plan óptimo. De esta forma, se consigue una ejecución eficiente de las tareas y un mejor aprovechamiento de los recursos, que se traduce en una mayor autonomía conjunta del equipo de UAVs.

El sistema se ha diseñado en ROS y las comunicaciones entre las diferentes capas de software y los diferentes bloques de cada capa se han realizado empleando las herramientas que este ofrece. Esto facilita la integración del sistema desarrollado en otros proyectos de robótica que necesiten de un planificador de tareas con estas características o similares. El uso de BT para el diseño del UAV behaviour manager presenta grandes ventajas frente a las FSM convencionales. Esta técnica permite generar comportamientos complejos con numerosos estados sin que haya que preocuparse por tener en cuenta cada una de las transiciones entre esos estados como pasa con las FSM, en las que el número de transiciones crece exponencialmente con el número de estados. Las características de la librería empleada para programar esta parte del sistema hacen que sea fácil de mantener, de modificar o de ampliar. Además, gracias a su carácter modular, este bloque puede ser reutilizado tanto a partes como en su totalidad en cualquier otro proyecto. El BT diseñado, aunque mejorable, ha demostrado en las simulaciones realizadas que funciona bastante bien, sentando las bases para programar comportamientos más complejos en el futuro y sirviendo de ejemplo para la comunidad de robótica aérea, que podrá emplearlo como punto de partida para otras aplicaciones.

Respecto al bloque que se encarga de la planificación de la misión, el High-Level Planner, decir que hasta el momento ha demostrado tener la capacidad para generar planes con sentido en las condiciones en las que se le ha puesto a prueba y que además ha demostrado ser capaz de recalcular dichos planes en línea como reacción a eventos imprevistos de diferente naturaleza. La solución alcanzada es capaz de planificar la misión teniendo en cuenta restricciones impuestas como el tipo de cada ACW, la prioridad de cada una de las tareas y el nivel de autonomía de cada uno de los UAVs, siendo capaz de calcular planes para misiones formadas por una cantidad indefinida de tareas y ACWs. Respecto a la optimalidad de los planes generados por este bloque hay que decir que no

se ha implementado ningún algoritmo que realice una búsqueda o aproximación del plan óptimo, sino que se ha diseñado una solución basada en una función de costes que se calcula para cada uno de los ACWs con cada una de las tareas. Sin embargo, este trabajo forma parte de un proyecto que tendrá aplicaciones reales en unas condiciones definidas. Por tanto, está justificado alejarse de análisis académicos en los que se busca aproximar el plan óptimo en misiones con un número indeterminado de tareas y de ACWs y analizar en su lugar al planificador en escenarios compuestos por pocas tareas y ACWs. La solución alcanzada, en este contexto, es una aproximación válida hacia un algoritmo de planificación que genere un plan próximo al óptimo.

5.2 Trabajo futuro

Como parte del trabajo futuro se realizará una validación en un entorno real con equipos reales de las técnicas desarrolladas en este trabajo. Además, el sistema desarrollado en este trabajo se empleará como punto de partida de una tesis doctoral en la que se tratará de pulir y mejorar el diseño del Agent Behaviour Manager, así como de desarrollar un algoritmo de planificación que genere una aproximación real al plan óptimo de cada situación. Para ello se introducirán al sistema algoritmos heurísticos aleatorios, así como la capacidad para aprender en tiempo real características como el consumo de la batería de los UAVs, anticipándose de esta forma a eventos imprevistos aplicando planes de contingencia, consiguiendo así una mayor robustez ante fallos y extendiendo la autonomía del sistema aún más.

Una primera mejora para el planificador respecto a la versión actual podría ser la incorporación de tareas de tipo Recarga que, en vez de ser solicitadas por operarios humanos como el resto de las tareas, serían incorporadas por el High-Level Planner a la cola de tareas, separando de esta forma las recargas de emergencia o las recargas que se ejecutan cuando un agente se encuentra ocioso, de las recargas llevadas a cabo como parte del plan. Implementar este cambio en el BT supondría modificar el Perform Task Tree para contemplar esta nueva tarea en el diseño del árbol, cambio que se podría llevar a cabo reutilizando y adaptando ligeramente los árboles empleados para las tareas de Inspection y Safety Monitoring aprovechando la reusabilidad de los BTs.

Además, durante la futura tesis, se pretende investigar el uso de tecnologías de realidad mixta también para aplicaciones de inspección con equipos multi-UAV, combinando vistas tomadas desde distintos puntos para recrear entornos visuales más completos para el operario, y mejorando la interacción hombre-máquina del sistema durante tareas colaborativas.