https://doi.org/10.17979/spudc.9788497498081.0848

Coordinación UAV-UGV para tareas de Búsqueda y Rescate

Pablo Rodríguez Palafox, Mario Garzón Oviedo y Antonio Barrientos Cruz
Centro de Automática y Robótica UPM-CSIC
c/ José Gutierrez Abascal, 2, 28006, Madrid, España
pablo.rodriguez.palafox@alumnos.upm.es, ma.garzon@upm.es, antonio.barrientos@upm.es

Resumen

Se presenta un sistema autónomo de Búsqueda y Rescate basado en la coordinación entre un UAV y un UGV (este último funcionando como plataforma móvil de aterrizaje). El vehículo aéreo debe localizar el objetivo en base a una imagen aérea, seguirlo y aterrizar sobre él cuando así le sea indicado. El algoritmo de seguimiento y aterrizaje está basado en la combinación de un regulador PID con un algoritmo de predicción (basado este último, a su vez, en un filtro de Kalman) con el que se van obteniendo las posiciones futuras de la plataforma. Ello facilita el seguimiento y, principalmente, el aterrizaje, ya que el UAV puede posicionarse ligeramente por delante de la plataforma de aterrizaje momentos antes de la aproximación final y así poder completar con éxito el aterrizaje.

Palabras clave: Aterrizaje autónomo, Plataforma móvil, Seguimiento, Filtro de Kalman, Control en altura, UAV (Unmanned Aerial Vehicle), UGV (Unmanned Ground Vehicle), Visión por Computador, PID.

1. INTRODUCCIÓN

Día tras día, la robótica va tomando una mayor importancia en nuestras vidas. De sus múltiples aplicaciones, en este proyecto se abordará la cooperación entre plataformas robóticas para tareas de Búsqueda y Rescate.

Los robots destinados a este tipo de tareas (SaR robots - Search and Rescue robots) deben operar en muchas ocasiones en entornos desconocidos, sobre superficies no estables y con múltiples dificultades para efectuar su misión. Esta, por lo general, incluye la obtención de un mapa del entorno para facilitar la intervención posterior de las brigadas de rescate. Para un robot terrestre, este tipo de condiciones no resultan las mejores para operar. Una posible solución es la de constituir equipos mixtos de un robot terrestre (UGV - Unmanned Ground Vehicle) y uno aéreo (UAV - Unmanned Aerial Vehicle) para realizar estas tareas de Búsqueda y Rescate. La elección de este tipo de cooperación UAV-UGV queda justificada por la posibilidad de compensar las carencias del uno con los puntos fuertes del otro.

Efectivamente, mientras que los robots aéreos poseen una capacidad única de obtener vistas en altura y de desplazarse sin verse obstaculizados por los elementos que pueda haber sobre el terreno tras un derrumbamiento, su reducida autonomía de vuelo limita su tiempo de uso a pocas decenas de minutos y su capacidad de carga a poco más de 1 Kg, generalmente.

Por el contrario, los robots terrestres sí son capaces de superar, por lo general, los requisitos de autonomía energética y capacidad de carga de pago. Además, pueden actuar como repetidores para los sistemas de comunicaciones, así como proveer elevadas capacidades de cómputo y almacenamiento de datos al sistema. Sus puntos débiles aparecen a la hora de superar obstáculos o situaciones complicadas, como puentes estrechos o planos inclinados; adicionalmente, su capacidad de obtención de información acerca de su entorno también puede quedar limitada por los elementos del escenario o por las propias limitaciones de los sensores.

De manera que se presenta como muy buena alternativa el empleo de un sistema mixto, en el que un robot terrestre provea la autonomía energética a un robot aéreo, así como el transporte de la carga de pago y del propio robot aéreo. Este útlimo, a su vez, podrá, si la situación así lo requiere, proveer imágenes u otras medidas que permitan al controlador del sistema mejorar su percepción de la situación en la que se encuentra. Este trabajo se propone como un paso hacia la obtención de un sistema de dichas características, para lo cual se aborda la automatización de dos tareas: el seguimiento mediante un robot aéreo de los movimientos realizados por un robot terrestre y el aterrizaje autónomo sobre una plataforma de aterrizaje a bordo de dicho UGV.

2. ESTADO DEL ARTE

En los últimos años, el desarrollo de aplicaciones relacionadas con el uso de drones está siendo muy elevado; en concreto, existe un enorme interés en

poder emplear estos como herramientas de detección y seguimiento de objetos móviles terrestres. Sin embargo, como ya se ha visto en la sección anterior, su baja autonomía hace imposible el poder realizar tareas complejas de larga duración. Es por ello que gran parte de la investigación hasta ahora se ha centrado en conseguir aterrizar UAVs sobre plataformas móviles, dotándolos así de una mayor versatilidad, ya que en muchos escenarios es imposible asegurar una zona de aterrizaje estacionaria.

Ling *et al.* [4] intentan dar solución a los problemas que surgían en la toma de fotografías de icebergs empleando drones, los cuales eran lanzados desde un barco. Los vehículos aéreos debían ser posteriormente rescatados semi-manualmente entre dos operarios: el primero debía encargarse de pilotarlo hasta situarlo cerca de la embarcación, mientras que un segundo operario era el encargado de recuperar el vehículo aéreo de forma manual, con el peligro que ello conllevaba. Ling consigue finalemente desarrollar un algoritmo de aterrizaje de precisión con el que eliminar por completo la actuación humana en este tipo de situaciones.

Existen algunos otros ejemplos de aterrizajes autónomos de VTOL UAVs (Vertical Take-Off and Landing UAVs) sobre plataformas móviles. Lee, Ryan y Kim [3] se centran en el uso de cámaras verticales y algoritmos IBVS (Image-Based Visual Servoing) para el seguimiento de una plataforma en un espacio bidimensional; obtienen después la velocidad con la que se mueve la plataforma y, finalmente, usan este dato como referencia para realizar un control adaptativo de movimiento deslizante. Comparado con otros algoritmos de control basados en visión que reconstruyen una representación completa en 3D del objetivo (los cuales requieren estimaciones de profundidad precisas), los algoritmos IBVS son menos costosos computacionalmente.

Anteriormente a estos dos trabajos, Saripalli [6] ya trabaja con algoritmos de visión para el aterrizaje autónomo de un helicótero sobre una plataforma móvil. Saripalli emplea los momentos de inercia de Hu para una detección precisa del objetivo y un filtro de Kalman para el seguimiento del mismo. Basándose en la salida del algoritmo de tracking, es capaz de implementar un controlador de trayectoria que asegura el aterrizaje sobre el objetivo móvil.

Es interesante observar que en la gran mayoría de los proyectos que tratan el tema de la detección y localización de objetos móviles desde quadrotors se asumen ciertas simplificaciones. La primera de ellas es asumir como conocida la forma y/o color del objetivo a localizar, y es que la discriminación de este último era hasta hace unos años la tarea más compleja a realizar. Con la mejora y aparición de nuevos algoritmos de visión (librería OpenCV), hoy en día, la dificultad de esta tarea es notablemente menor. Asimismo, se suele asumir que la velocidad del objetivo móvil es lo suficientemente baja como para que el dron sea capaz de aterrizar sobre él con la mayor facilidad posible. Sin embargo, en trabajos realizados por el Centro Aeroespacial Alemán, se ha conseguido que el UAV sea capaz de aterrizar sobre una red colocada encima de un coche circulando a una velocidad cercana a los 70 km/h, si bien es cierto que el aterrizaje se realiza en línea recta.

Este último es un aspecto relevante a considerar, y es que existen diferencias sustaciales entre aterrizajes en trayectoria rectilínea, circular o completamente aleatoria. La mayoría de las investigaciones hasta el momento se centran únicamente en un aterrizaje en línea recta, sin asumir en ningún caso que el movimiento del objetivo pueda ser aleatorio. Esta es, por tanto, una línea interesante de trabajo, ya que en tareas de búsqueda y rescate la plataforma móvil terrestre (generalemente, un UGV) tendría total libertad de movimiento, al ser capaz el UAV de realizar el aterrizaje en cualquier situación, por muy aleatoria que fuera la trayectoria seguida por dicha plataforma de aterrizaje.

3. PLANTEAMIENTO Y ENFOQUE DEL PROBLEMA

El aterrizaje sobre una plataforma móvil puede descomponerse en tres etapas. La primera consiste en la **detección y localización** del objetivo. Para ello se emplea un algoritmo de visión desarrollado a partir del que propone Baira [1].

La segunda fase consiste en el **seguimiento** de la plataforma de aterrizaje, para lo cual, en un primer acercamiento al problema, se empleó un algoritmo de seguimiento basado en un regulador PID. Posteriormente, se quiso implementar un algoritmo de predicción basado en un filtro de Kalman desarrollado por Garzón *et al.* [2]. En este último caso, el valor de referencia para el regulador PID es la posición futura del centroide de la plataforma de aterrizaje, calculado gracias al filtro de Kalman.

La etapa final es la de **aterrizaje**. Durante ella, el algoritmo de predicción sigue funcionado, aunque con un pequeño matiz: a medida que el UAV desciende, los parámetros de configuración del algoritmo de predicción habrán de irse modificando, con el fin de obtener los mejores resultados de dicha predicción.

Paralelamente a todo ello, se realiza un **control**

en altura con el que se persiguen dos objetivos: primero, que en caso de que el aterrizaje no sea correcto o la plataforma salga del campo visual de la cámara, el UAV recupere altura y vuelva a intentar detectar y localizar la plataforma de aterrizaje; y segundo, que la velocidad de descenso del UAV varíe en función de la franja de altura en la que se encuentre.

3.1. PLATAFORMAS ROBÓTICAS

Esta sección describre las plataformas utilizadas para resolución del problema. Se emplea como plataforma robótica aérea el AR Drone 2.0; en cuanto a la terrestre, se ha utilizado el Summit XL (véase Figura 1).

Figura 1: Plataformas robóticas reales.

En cuanto a los modelos empleados en las simulaciones, estos han sido obtenidos de la Universidad de Darmstadt (modelo del UAV) y de la empresa Robotnik, fabricante del Summit XL (véase Figura 2).

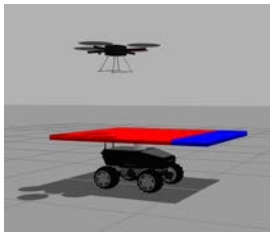

Figura 2: Plataformas robóticas empleadas en el simulador.

3.2. SIMPLIFICACIONES DEL MODELO

Es necesario asumir ciertas simplificaciones de cara a que el problema de seguimiento y aterrizaje de una plataforma móvil sea abordable:

1. Tanto en la simulación como en las plataformas robóticas reales, se supone conocido el color y forma de la plataforma de aterrizaje.
2. Se asume que la plataforma de aterrizaje se mueve suficientemente lenta como para que el UAV sea capaz de seguirla y aterrizar sobre ella.
3. Se asume también que la plataforma no realizará maniobras evasivas o maliciosas, sino, como mucho, trayectorias aleatorias.

4. DETECCIÓN Y LOCALIZACIÓN

Esta sección describe la sub-tarea de detección y localización de un objetivo móvil, en este caso la plataforma de aterrizaje. Estos algoritmos emplean información tomada por la cámara ventral y el sensor de altura, incorporados ambos en el UAV AR Drone 2.0.

4.1. DETECCIÓN

En esta sección se ha de hacer notar una diferencia fundamental entre el trabajo realizado en el simulador (Gazebo) y sobre las plataformas robóticas reales. De cara a la implementación de los algoritmos en las plataformas reales, la detección se lleva a cabo empleando un marcador de visión fácilmente detectable con las librerías existentes de OpenCV (véase Figura 3).

Figura 3: Marcador visual empleado para la detección de la platforma de aterrizaje en el sistema real.

Por el contrario, en el simulador no se han empleado dichas etiquetas visuales, sino que se ha optado por diseñar una plataforma de aterrizaje rectangular de medidas 1.2x1.6m^2 de color fácilmente detectable. Posteriormente, esta plataforma ha sido situada sobre el modelo del robot terrestre modificando ligeramente los archivos originales de definición del modelo simulado Summit XL.

Con independencia del método de detección empleado, una vez detectada la plataforma de aterrizaje se generará el centroide geométrico de la misma, dando entonces paso a la etapa de localización.

4.2. LOCALIZACIÓN

Esta etapa tiene como entrada el centroide anteriormente calculado, el cual representa al objetivo móvil. Dicho centroide está definido como un punto dentro de la matriz de píxeles que se obtiene

con la cámara. Este dato, como se puede deducir fácilmente, no es fácilmente manejable, de manera que es necesario tratarlo. El objetivo será obtener la posición de la plataforma de aterrizaje respecto del sistema de referencia de la cámara.

Para ello, se emplea un algoritmo basado en la inversión del modelo Pinhole, el cual describe la relación matemática entre un punto en el espacio y su proyección sobre el plano imagen de una cámara (véase Figura 4).

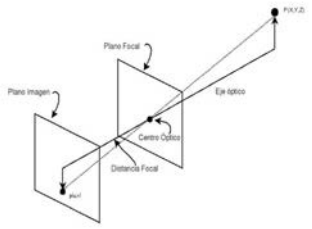

Figura 4: Proyección del punto objetivo P(X,Y,Z) sobre los planos usados en el modelo.

De la tesis de Yannick Morvan [5] se puede obtener el siguiente marco teórico.

El modelo de Pinhole en su forma matricial se puede observar en la Ecuación 1.

$$\lambda \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} f & \tau & \sigma_x \\ 0 & \eta f & \sigma_y \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (1)$$

Donde λ es el factor de escala, $(u, v)^T$ es la proyección del punto sobre el plano imagen expresada en píxeles, f es la distancia focal en píxeles, τ es el parámetro que indica si los píxeles están torcidos o sesgados (valiendo 0 cuando no lo están), $(\sigma_x, \sigma_y)^T$ son las coordenadas del centro óptico de la cámara. Por último, η indica la forma del píxel, siendo estos cuadrados cuando η vale 1 (véase Figura 5), y (X,Y,Z) las coordenadas de la plataforma de aterrizaje respecto de la cámara del UAV.

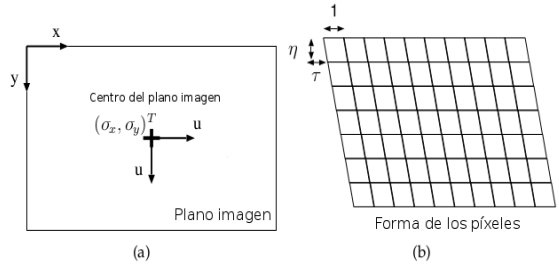

Figura 5: (a) Sistema coordenado de la imagen (x,y) y de la cámara (u,v). (b) Representación de píxeles no cuadrados y sesgados.

Una vez obtenidos los parámetros de la cámara a través de un proceso de calibración y teniendo en cuenta el valor de la altura del UAV, se pasa a resolver el sistema de ecuaciones. Se invierte la matriz, se sustituye $(u, v)^T$ por las coordenadas del centroide calculadas en la etapa anterior (detección) y, finalmente, se obtienen las coordenadas buscadas P(X,Y,Z), *posición del centroide de la plataforma de aterrizaje respecto de la cámara del UAV.*

Los resultados de esta sub-tarea de detección-localización para la simulación pueden observarse en la Figura 6. Nótese en dicha figura la existencia de dos puntos rojos, uno dentro de un recuadro verde y otro dentro de un cuadro negro recuadrado por bordes azules. El primero es el centroide de la plataforma de aterrizaje, entendiendo como tal el punto que caracteriza al objetivo. Este es el punto que el algoritmo de seguimiento empleará como valor de referencia. El otro punto es un indicador que ayuda a determinar la dirección de avance de la plataforma terrestre.

En la Figura 7 se muestra una vista aérea de la plataforma de aterrizaje, la cual ha sido situada encima del Summit XL. En el centro de la misma se encuetra el marcador visual que el AR Drone 2.0. es capaz de detectar a través de su cámara ventral.

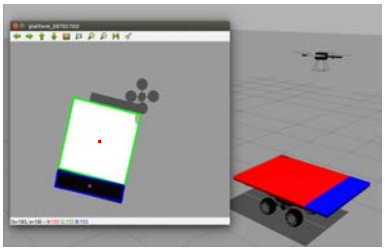

Figura 6: Detección de la plataforma de aterrizaje y cálculo del centroide de la misma, así como de un indicador de la dirección que lleva el objetivo móvil.

Figura 7: Vista aérea de la plataforma de aterrizaje real desde la cámara ventral del AR Drone 2.0. La plataforma se ha colocado encima del Summit XL.

5. SEGUIMIENTO Y ATERRIZAJE

En este momento se ha de diferenciar entre un algoritmo de seguimiento-aterrizaje *sin predicción* y otro que, por el contrario, *se ayuda del algoritmo de predicción* antes mencionado desarrollado por Garzón *et al.* [2] y que pretende ser una mejora del primero. Se ha de hacer incapié, sin embargo, en que lo único que los diferencia es el dato de entrada: mientras que el primero recibe la posición del centroide de la plataforma respecto de la cámara, el segundo tiene como *input* la posición futura de dicho centroide referenciada al sistema del mundo.

De hecho, ambos comparten la misma estructura basada en **variables de estado**. Se pueden distinguir 4 estados distintos: *IDLE*, *TAKINGOFF*, *TRACKING* y *LANDING*. A continuación, se explican en detalle los distintos estados.

5.1. DIAGRAMA DE ESTADOS

El primero, *IDLE*, es el estado de reposo del quadrotor y es su estado por defecto. Asimismo, el UAV pasará también encontrarse en este estado una vez haya aterrizado correctamente.

Posteriormente, y solo si el usuario así lo desea (pusación botón de despegue), se podrá pasar al estado de *TAKINGOFF*, que representa el periodo durante el cual el vehículo aéreo está ganando altura. Durante este estado, independientemente de si la plataforma de aterrizaje entrara o no en el campo de visión del UAV, no se atiende a los mensajes provenientes del algoritmo de localización.

Una vez el dron ha alcanzado una altura determinada (prefijada por el usuario), el estado del quadrotor cambia automáticamente a *TRACKING*. Es a partir de este momento cuando se dará luz verde a que el regulador PID empiece a calcular la velocidad que es necesario aplicar al vehículo aéreo (velocidad tanto en x como en y) para que este reduzca su distancia con el centroide de la plataforma de aterrizaje. Es importante destacar que durante el estado de TRACKING el dron se mantiene a una altura constante.

Finalmente, y solo si el usuario lo considera oportuno (pulsación botón de aterrizaje), el estado del UAV cambiará a *LANDING*.

Tras haberse presentado el esquema general de ambos algoritmos (sin y con predicción), se presentarán a continuación brevemente las diferencias básicas entre ellos. Se quiere hacer notar que el motivo de mantener esta distinción es que el algoritmo de seguimiento-aterrizaje sin predicción presenta muy buenos resultados en aterrizajes con trayectorias rectilíneas de la plataforma (véase Sección 6), siendo innecesaria la carga computacional extra que supondría la introducción del algoritmo de predicción en este tipo de aterrizaje. Sin embargo, la respuesta de dicho algoritmo sin predicción no es buena ante movimientos circulares o aleatorios de la plataforma de aterrizaje, algo que se corrige al introducir un filtro de Kalman.

5.2. SEGUIMIENTO-ATERRIZAJE SIN PREDICCIÓN

5.2.1. Seguimiento sin predicción

Como ya se ha mencionado antes, el punto obtenido en la etapa de localización está referenciado respecto de la cámara. Por ello, si lo que se quiere es emplearlo como posición de referencia para el control PID, lo primero que se ha de hacer es transformar sus coordenadas al sistema de referencia del cuerpo del quadrotor.

Una vez realizada esta transformación, se está en disposición de calcular el error de posición, que se define en este caso como la distancia entre el UAV y el centroide de la plataforma de aterrizaje, referenciado este último punto al sistema de ejes del quadrotor.

El ajuste de los parámetros del PID (Kp, Ki y Kd) se ha llevado a cabo a través de reglas heurísticas, esto es: buscando primero la Kp que proporciona la respuesta deseada (manteniendo Ki y Kd a cero); incrementando después de manera progresiva la Ki para anular el error de posición (y disminuyendo ligeramente la Kp para que el sistema no se haga inestable); y, finalmente, aumentando la Kd para que la respuesta del sistema sea más rápida (manteniendo los valores de Kp y Ki obtenidos en los dos pasos anteriores). Los valores obtenidos para la simulación se muestran en el Cuadro 1. Nótese que no se ha hecho distinción alguna entre los ejes x e y del UAV, ya que las dinámicas del vehículo respecto de dichos ejes son prácticamente iguales.

Cuadro 1: Constantes del PID empleadas en la simulación.

Constantes del PID	
Kp	0.1045
Ki	0.0015
Kd	0.008

Con respecto a las constantes empleadas en la plataforma real (AR Drone 2.0), estas no son las mismas que para la simulación (véase Cuadro 2), debido a que la inercia del quadrotor real es mayor que la que posee el modelo de la simulación. En

concreto, se ha reducido la acción integral, aumentado la derivativa (para conseguir una respuesta más rápida) y reducido la proporcional (de manera que esté asegurada la estabilidad del sistema).

Cuadro 2: Constantes del PID empleadas en el AR Drone 2.0.

Constantes del PID	
Kp	0.04
Ki	0.001
Kd	0.01

5.2.2. Aterrizaje sin predicción

Cuando el UAV pasa al estado de LANDING, el control PID sigue funcionando. La única diferencia ahora es la adición de un control de altura (véase Sección 5.4), que permitirá al UAV ir descendiendo hasta la plataforma de aterrizaje a una velocidad determinada por la franja de altura en la que se encuentre.

En conreto, y para el caso de la *plataforma aérea real*, la estrategia a seguir es la siguiente. El UAV descenderá desde la altura característica del estado de TRACKING a una velocidad de 0.4 m/s hasta encontrarse a 0.5 m por encima de la plataforma de aterrizaje. En ese momento se apagarán los motores y el UAV caerá sobre ella.

Para el caso de la *simulación*, la estrategia es la misma, aunque se han de controlar los motores de forma distinta con el objetivo de simular la realidad. Ahora, cuando el quadrotor se encuentre a 0.5 m de la plataforma de aterrizaje, su velocidad será mayor (de 1 m/s), de manera que se pueda recrear el apagado de los motores que tiene lugar en la plataforma real. De cara a poder *decidir cuándo se ha aterrizado*, se verifica si la diferencia entre dos posiciones consecutivas de la plataforma de aterrizaje (respecto al cuerpo del UAV) es muy próxima a cero. En tal caso, se pondrán a cero las velocidades del UAV (x,y,z) y pasará a estado IDLE, esperando a que el usuario indique el comienzo de un nuevo ciclo de rastreo (despegue - seguimiento de la plataforma de aterrizaje a la vez que realiza una posible toma de datos del terreno - aterrizaje).

5.3. INTEGRACIÓN DEL ALGORITMO DE PREDICCIÓN

Como ya se vio al comienzo de esta sección, el algoritmo de seguimiento-aterrizaje con predicción recibe como entrada la posición futura de la plataforma de aterrizaje referida al sistema mundo.

Sin embargo, antes de ser recibida dicha posición, la información proveniente del algoritmo de detección-localización fluye por distintos nodos ROS, que se encargan de tratarla correctamente. A continuación se recuerda cuál es este flujo de información.

Inicialmente, y como ya se ha visto, el algoritmo de detección-localización genera un punto, que es la posición del objetivo móvil con respecto de la cámara. Después, esta información es recogida por uno de estos nodos ROS de control anteriormente citados, que se encarga de referenciar dicha posición al sistema de referencia del mundo. Esto es necesario debido a que el algoritmo de predición necesita trabajar con posiciones referidas a sistemas de referencia inerciales.

Con respecto al algoritmo de detección, este se basa en la generación de un *path* o vector de posiciones futuras ordenadas en el tiempo. Los parámetros de configuración del algoritmo son los siguientes:

- ***path time*:** indica al algoritmo el tiempo total en segundos de la predicción, es decir, cuánto tiempo por delante se necesita que prediga el algoritmo. Cuanto mayor sea este valor, menor será la fiabilidad del último valor del *path*, a no ser que la trayectoria seguida por el objetivo sea totalmente rectilínea. En cualquier otro caso, este valor debería ser cercano a los 3-4 s.
- ***step*:** tiempo en segundos entre las posiciones del vector del *path* para un mismo ciclo de predicción, cumpliéndose: *step* $<$ *path time*.
- ***pub freq*:** representa la frecuencia en Hz a la cual el algoritmo de predicción publicará la información calculada.

Tras realizar los cálculos necesarios, este nodo ROS de predicción habrá generado el mencionado vector de posiciones futuras. En este caso, y de cara al algoritmo de seguimiento-aterrizaje, se tomará solamente el último valor de dicho vector, que corresponderá a la posición futura de la plataforma de aterrizaje dentro de un tiempo definido por el parámetro *path time*. Es importante hacer notar que esta posición está referida al sistema de referencia del mundo.

5.3.1. Seguimiento con predicción

Una vez generada dicha posición futura del objetivo y tras referirla al sistema de referencia del cuerpo del quadrotor, será con esta posición con la que el control PID trabaje para generar una velocidad en x y otra en y para el UAV. De esta manera, durante el estado de TRACKING, este

se mantendrá siempre ligeramente por delante del objetivo terrestre.

5.3.2. Aterrizaje con predicción

Tras pulsar el usuario el botón de aterrizaje, el quadrotor pasará al estado de IDLE y, de la misma manera que ocurría con el algoritmo sin predicción, el control PID en el plano xy del UAV seguirá funcionando paralelamente al control en altura en el eje z.

La diferencia ahora es que a medida que el dron descienda, será necesario que el sistema vaya actualizando automáticamente el parámetro *path time*. En concreto, se deberá ir reduciendo. Esto es así debido a que, cuanto más bajo vuele el UAV, menor campo visual tendrá, y, en consecuencia, la predicción solo será fiable del orden de 0.5 a 2 s por delante en el tiempo. De manera que este valor de *path time* irá reduciéndose desde un máximo de 5-6 s hasta un mínimo de 0.5 s, que coincidirá con el momento en que el UAV apague sus motores (en en el caso de la plataforma real) y concluya, por fin, su aterrizaje con éxito.

5.4. CONTROL DE ALTURA

Englobado dentro del algoritmo del seguimiento-aterrizaje y corriendo paralelamente a la "máquina de estados", el quadrotor cuenta con un control de altura.

Principalmente, asegura que, si el aterrizaje no fue correcto, el sistema se reinicie. Es decir, si el dron pierde la plataforma de aterrizaje de su campo de visión debido a un aterrizaje fallido y el sistema pasa una determinada cantidad de tiempo sin volver a detectar la plataforma de aterrizaje, el estado del quadrotor pasará de LANDING a TAKINGOFF de forma automática (sin necesidad de intervención por parte del usuario). Esto es, dejará de intentar aterrizar y comenzará a ascender. Al aumentar progresivamente la altitud del vehículo, su campo de visión crecerá, pudiendo entonces, muy probablemente, volver a detectar el objetivo móvil.

Asimismo, este control de altura es el encargado de asignar una velocidad de descenso para el quadrotor en función de la franja de altitud en la que se encuentre.

6. RESULTADOS

Se presentan aquí los resultados obtenidos en la simulación. Actualmente se están llevando a cabo las pruebas sobre las plataformas robóticas reales, pero la dificultad que entraña la preparación de cada prueba ha impedido que se esté en disposición de suficientes datos como para sacar conclusiones al respecto.

6.1. RESULTADOS EN LA SIMULACIÓN

Se diferencia entre los resultados obtenidos sin y con algoritmo de predicción.

6.1.1. Sin predicción

A grandes rasgos, el sistema es muy robusto y funciona correctamente siempre y cuando la trayectoria seguida por la plataforma de aterrizaje sea rectilínea. Sin embargo, cuando esta se mueve describiendo movimientos circulares, los resultados no son buenos. En este segundo caso, el quadrotor, aun siendo capaz de seguir el objetivo de manera muy aceptable y sin perderlo nunca de vista durante el estado de TRACKING, no es capaz de aterrizar de manera satisfactoria sobre la plataforma, posándose la mayoría de las veces de manera inestable sobre ella. Normalmente, consigue apoyar solo parte de las patas, cayéndose después o, en ocasiones, activándose el sistema de recuperación de altura y retomando posteriormente el seguimiento del objetivo.

Los resultados obtenidos tras realizar las pruebas en el simulador se muestran en el Cuadro 3. Se han realizado 50 ciclos de despegue-seguimiento-aterrizaje tanto para el caso en que la plataforma de aterrizaje sigue una trayectoria rectilínea como para cuando su movimiento es aleatorio.

Cuadro 3: Resultados tras 50 ciclos de aterrizaje empleando el algoritimo de seguimiento-aterrizaje **sin predicción**. Velocidad máxima admitible para la plataforma de aterrizaje: 1 m/s.

	Rectilínea	**Aleatoria**
Aterrizajes Exitosos	49	7
Aterrizajes Fallidos	1	43
Total realizados	50	50
% de éxito	98 %	14 %

6.1.2. Con predicción

En el Cuadro 4 se observa cómo la introducción del algoritmo de predicción ha conseguido arreglar los problemas del aterrizaje cuando la trayectoria seguida por la plataforma de aterrizaje no es rectilínea.

Cuadro 4: Resultados tras 50 ciclos de aterrizaje empleando el algoritimo de seguimiento-aterrizaje **con predicción**. Velocidad máxima admitible para la plataforma de aterrizaje: 0.6 m/s.

	Rectilínea	**Aleatoria**
Aterrizajes Exitosos	50	46
Aterrizajes Fallidos	0	4
Total realizados	50	50
% de éxito	100 %	92 %

7. CONCLUSIONES

En este artículo se ha propuesto un sistema mixto de Búsqueda y Rescate basado en visión que permite a un UAV realizar el despegue, seguimiento y aterrizaje autónomo sobre una plataforma terrestre en movimiento. Esto permite a dicho vehículo aéreo la recogida de información sobre el entorno sin que su autonomía de vuelo suponga un problema, al poder recargarse en una hipotética base de carga situada en el robot terrestre.

Se ha observado que un control en posición basado en un PID es suficiente para aterrizajar satisfactoriamente cuando el objetivo terrestre describe trayectorias rectilíneas, fallando en caso de que la trayectoria sea circular o aleatoria.

Un control basado en un filtro de Kalman consigue dar solución a este problema, al predecir de forma satisfactoria las posiciones futuras de la plataforma y poder entonces el UAV adelantarse ligeramente a la misma para efectuar el aterrizaje con éxito. Las ocasiones en las que el aterrizaje no se completa con éxito aun empleando el algoritmo de predicción se deben a giros demasiado abruptos por parte de la plataforma de aterrizaje en el momento justo en que el UAV ha pasado la franja de 0.5 m/s de altura, momento en el que se corta el control *PID-Filtro de Kalman* y se deja que el UAV caiga sobre el objetivo.

8. TRABAJOS FUTUROS

Sería interesante mejorar el sistema de Control de Altura añadiendo una funcionalidad de *rastreo* en caso de que no bastara con recuperar altura para volver a detectar la plataforma de aterrizaje. Ello podría enfocarse como un nuevo estado, SEARCHING, durante el cual el UAV comenzara a describir círculos, aumentando progresivamente el diámetro de los mismos, hasta encontrar de nuevo el objetivo.

Una mejora quizás más elegante al respecto sería recordar por qué punto de la imagen se vió desaparecer a la plataforma y retomar la búsqueda apuntando en esa dirección.

Agradecimientos

Esta investigación ha recibido fondos del proyecto RoboCity2030-III-CM (Robotica aplicada a la mejora de la calidad de vida de los ciudadanos, fase III; S2013/MIT-2748), financiado por los Programas de Actividades I+D en la Comunidad de Madrid y los Fondos Estructurales de la Unión Europea, y del proyecto DPI2014-56985-R (Protección robotizada de infraestructuras críticas), financiado por el Ministerio de Economía y Competitividad del Gobierno de España.

Referencias

[1] Baira Ojeda, I., Garzón Oviedo, M. y Barrientos Cruz, A. (2016). "Detecting, Localizing and Following Dynamic Objects with a Mini-UAV". En *RoboCity16 Open Conference on Future Trends in Robotics, May 26 - 27th 2016, Madrid, Span. ISBN 978-84-608-8452-1. pp. 275.*

[2] Garzón, M., Garzón-Ramos, D., Barrientos, A. and del Cerro, J., (2016) "Pedestrian Trajectory Prediction in Large Infrastructures: A long-term approach based on path planning". In *Proceedings of the 13th International Conference on Informatics in Control.*

[3] Lee, D., Ryan, T. and Kim, H.J., (2012) "Autonomous landing of a VTOL UAV on a moving platform using image-based visual servoing". In *Robotics and Automation (ICRA), 2012 IEEE International Conference*, pp 971-976.

[4] Ling, K., (2014) "Precision Landing of a Quadrotor UAV on a Moving Target Using Low-cost Sensors". In *University of Waterloo.*

[5] Morvan, Y., Farin, D. and de With, P., (2016) "Depth-Image compression based on an RD optimized quadtree decomposition for the transmission of multiview images". In *IEEE International Conference on Image Processing*, pp 87-105.

[6] Saripalli, S. and Sukhatme, G.S., (2003) "Landing on a moving target using an autonomous helicopter". In *Field and service robotics*, pp 277-286.