

Plataforma para la implementación y validación de algoritmos de control de tiempo real en mini-helicópteros de varios rotores

Luis Ródenas
luidelo@aii.upv.es

Ricardo Sanz
rsanz@aii.upv.es

Pablo Albiol
pabalgr@etsii.upv.es

Alberto Castillo
alcafra@etsii.upv.es

Daniel Verdú
davertor@etsii.upv.es

Pedro García
pggil@isa.upv.es

Instituto de Automática e Informática Industrial. Universidad Politécnica de Valencia

Resumen

En este trabajo se presenta una plataforma para la implementación y validación de algoritmos de control en mini-helicópteros de varios rotores. La plataforma, se compone de una unidad de control terrestre (UCT) y un quadrotor con un sistema empotrado de control. La configuración del sistema empotrado, permite la ejecución de las tareas de control más críticas sobre un Sistema Operativo de Tiempo Real (SOTR). Para validar la plataforma, se han implementado varios algoritmos de control. Los resultados experimentales muestran el buen desempeño del sistema propuesto así como una fácil implementación y ajuste de los algoritmos de control considerados.

Palabras clave: Plataforma de desarrollo para quadrotors, Sistemas de tiempo real, PIDs y control no lineal por saturaciones.

1. Introducción

Un vehículo aéreo remotamente pilotado, “Remotely Piloted Aircraft Systems” (RPAS) es un vehículo aéreo capaz de realizar misiones en modo (semi) autónomo. Los mini-helicópteros con varios motores, son una clase de RPAS que han ocupando la atención de los investigadores durante los últimos años [4, 7].

Recientemente ha empezando a plantearse su uso en múltiples aplicaciones civiles de todo tipo. Cabe destacar que, en todo el mundo, se esta empezando a regular de manera más estricta el vuelo de UAVs; en España, concretamente, su uso con fines comerciales y civiles acaba de ser regulado en el BOE del 5 de julio de 2014 (página 59 de [29]).

En cualquier caso, la proyección que se prevé para este sector, y la necesidad de desarrollar nuevas estrategias de control que puedan adaptarse fácil y rápidamente a la futura normativa, justifica el desarrollo de nuevas plataformas para la

implementación y validación de nuevos algoritmos de control. La Unión Europea también considera, en su programa HORIZONTE 2020, los RPAS, y en particular los helicópteros de varios rotores, un sector estratégico de cara a su uso en aplicaciones civiles.

El desarrollo de un sistema de control para un RPAS no es trivial, debido principalmente a la compleja dinámica inherente a los sistemas aéreos, los cuales son multivariables, normalmente subactuados y no lineales [20, 21]. Equipado, normalmente, con un sistema empotrado de control, este vehículo puede emplear diferentes sensores, como cámaras de video, GPS, láser, detectores infra-rojos, detectores ultra-sonidos, etc., para poder navegar en un ambiente desconocido (ver por ejemplo, [6] [24]). El desarrollo del sistema de control para este tipo de vehículos no sólo implica un adecuado ajuste de las leyes de control, sino, la ejecución de las mismas de forma segura, y en el menor tiempo de muestreo posible [19]. Igual de importante es que la tarea de control que implemente el algoritmo de control presente el menor retardo (jitter) posible [3]. Como es sabido, este objetivo se puede cumplir fácilmente realizando la implementación sobre un micro-controlador y/o en un mini-PC con un Sistema Operativo de Tiempo Real (SOTR).

Un problema crucial, cuando se trabaja con este tipo de sistemas, es la necesidad de disponer de una herramienta que nos permita interactuar con el sistema empotrado en la fase de ajuste on-line de los parámetros de control [1, 21, 5].

El trabajo presentado, consiste en el desarrollo de una plataforma genérica para la implementación y validación de un sistema empotrado de control en un quadrotor. La plataforma esta concebida para permitir una fácil implementación y comparación entre diferentes algoritmos de control.

El resto del artículo está estructurado en las siguientes secciones: en la sección 2 se describe la herramienta desarrollada para la solución adopta-

da, mientras que en la sección 3 se plantea tanto el modelado matemático como las leyes de control básicas del quadrotor. En la sección 4 se presentan los resultados experimentales, y en la sección 5 las conclusiones y futuros trabajos.

2. Descripción de la plataforma

En esta sección se realiza una descripción de la plataforma desarrollada. Ésta consta de una Unidad de Control Terrestre (UCT) que se utilizará como interfaz (Human Machine Interface, HMI) entre el usuario y el sistema empotrado en el quadrotor. En el UCT se ha desarrollado una herramienta que permite un ajuste on-line de los parámetros de control y facilita el hecho de implementar y comparar diversos algoritmos de control.

Con respecto al sistema empotrado, se ha desarrollado una arquitectura en dos capas. El objetivo es que los algoritmos de control más críticos (orientación) se ejecuten sobre un μC (emulando una tarea crítica de un SOTR), dejando las tareas de mayor carga computacional (control de posición, flujo óptico, etc.) y comunicación con la UCT a un mini-PC.

2.1. Arquitectura del sistema empotrado

Los prototipos construidos se componen de tres elementos, que denominaremos: unidades de control, actuadores y elementos de sensorización (figura 1). Se han realizado varias iteraciones en el diseño de la plataforma (dispositivos HW y SO) hasta conseguir una arquitectura lo más simple y sencilla de configurar pero que, a la vez, ofrezca las mejores prestaciones posibles para realizar el control integral de un quadrotor tanto en interiores como en exteriores.

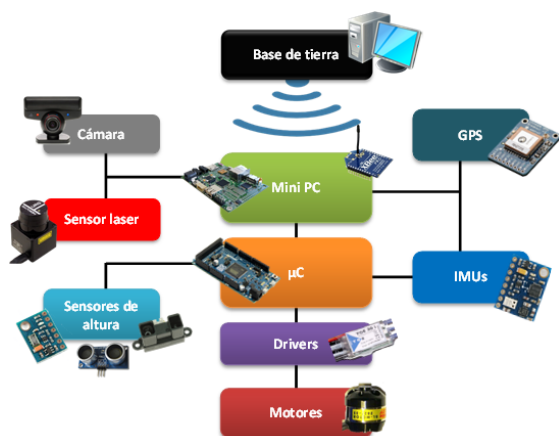


Figura 1: *Sistema empotrado*

Unidades de control

Las unidades de control del sistema empotrado son dos, un mini-PC y un μC . En la versión actual de la plataforma, el mini-PC utilizado es una placa IGEP V2 [10], mientras que el μC es un Arduino Due [9].

La IGEP V2 está provista de un procesador ARM Cortex A8 a 1 GHz, 512MB SDRAM, 512MB NAND Flash, lector de tarjetas micro-SD, USB 2.0, mini USB OTG 2.0, HDMI, Ethernet 10/100, Wi-Fi, RS-232, RS-485, I2C y SPI. Pero lo más interesante, tal vez, es que cuenta con un DSP (digital signal processing), el cual puede resultar muy útil a la hora de realizar el tratamiento de imágenes para la implementación del control de posición x-y mediante flujo óptico, por ejemplo, dado que descarga al procesador de todo el algoritmo de tratamiento de imágenes.

Como Sistema Operativo se ha instalado Linux Ubuntu + Xenomai [12]. El Arduino Due cuenta con un microcontrolador de 32 bits ARM Cortex M3 a 84 MHz, acompañado de 54 pines de entrada/salida digitales, 12 pines de entrada analógicos, 2 pines de salida analógicos, así como periféricos que se encargan de las comunicaciones serie, USB OTG, I2C, SPI e incluso bus CAN.

La figura 2 muestra los diferentes protocolos de comunicación utilizados para conectar los componentes del sistema empotrado entre ellos y con la UCT.

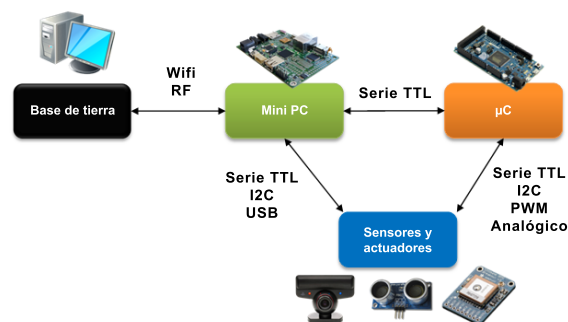


Figura 2: *Arquitectura de las unidades de control*

Actuadores

La plataforma utiliza motores brushless (sin escobillas) de la marca Robbe Roxxy, modelo 2827-35, especialmente pensados para quadrotors. Junto a unas helices 1045 flexibles, cada motor tiene un empuje máximo de unos 600 gramos (5.9 N) y un consumo máximo de 10 amperios.

Para variar la frecuencia de giro de los motores, y en consecuencia la fuerza de empuje que genera cada uno de ellos, se utilizan drivers o ESC (elec-

tronic speed controller). Los drivers utilizados en la plataforma son de la marca YGE [11], modelo 25i.

El diseño actual de la plataforma utiliza señales PWM (pulse-width modulation) para controlar los drivers. Se utiliza un pulso alto entre 1 y 2 ms de ancho, a una frecuencia de entre 200 y 500 Hz, actualmente fijada en 400 Hz. Por lo que esta es la frecuencia máxima de operación de los motores.

Sensorización.

Para poder obtener las distintas variables de estado, se utilizarán diferentes tipos de sensores. Para medir la orientación se utilizan IMUs, equipadas con acelerómetros, giroscopios y magnetómetros. La IMU hace el tratamiento de los datos, proporcionando directamente la medida de los ángulos de Euler (ϕ, θ, ψ) y sus derivadas. Para este proyecto se ha utilizado la IMU MPU 6050 de InvenSense Inc [13]. El quadrotor incorpora dos IMUs puesto que un fallo en la medida de los ángulos sería una situación crítica y puede llegar a provocar un fallo en la ley de control. En una de ellas los ángulos de Euler los proporciona directamente el procesador interno de la IMU. En la otra, se ha implementado un código que lee directamente las medidas en bruto de los sensores y aplica un filtro de Kalman con las medidas de giroscopios y acelerómetros para obtener el pitch y el roll y otro filtro de Kalman con las medidas del giroscopo y del magnetómetro [17] para obtener el yaw.

Para el control de posición, es necesario obtener una medida rápida y fiable tanto de la posición en el plano xy como de las respectivas velocidades. El GPS presenta el problema que en interiores no es funcional, ya que no hay visibilidad de los satélites. En el caso de interiores se pueden utilizar distintos sistemas de posicionamiento, la mayoría se basan en el principio de triangularización (Vicon, kinematics, etc.), si bien también es posible realizar un control de posición a partir de dispositivos de barrido laser (SLAM, [2]), o a partir de algoritmos de visión, como es el caso del control x-y por flujo óptico [22], a partir del cual es posible obtener las variables \dot{x}, \dot{y} , y a partir de ellas, estimar la posición relativa x-y.

Se ha dotado a la plataforma con un sistema de posicionamiento GNSS/INS. Dicho sistema combina las ventajas del *Sistema Global de Navegación por Satélite* (Global Navigation Satellite System: GNSS) con las ventajas de los *Sistemas de Navegación Inerciales* (Inertial Navigation Systems: INS).

Por un lado los sistemas GNSS dan una posición absoluta que no está sometida a deriva, sin embargo son lentos en cuanto a actualizaciones. Por

otro lado los sistemas INS detectan muy bien las variaciones rápidas de posición o velocidad, pero su mayor inconveniente es que están sometidos a una deriva temporal ya que obtienen la posición y velocidad por integración de la aceleración.

Actualmente el sistema GNSS más utilizado es el Sistema de Posicionamiento Global (GPS) de los Estados Unidos. El sensor GPS que se ha incorporado es el *Ultimate GPS Breakout v3* de Adafruit [14]. Se ha comprobado experimentalmente que el error de medida de dicho sensor, en un día despejado, con visibilidad directa al cielo y recibiendo correcciones DGPS está en torno a los 2 m para la posición y a los 0.1 m/s para la velocidad.

El GPS emite la información de posición en formato NMEA, por lo que da medidas de latitud, longitud y de la velocidad en módulo. Las señales de latitud y longitud se han transformado a medidas de posición local x, y (m). También se ha desarrollado un algoritmo que observa las variaciones de x, y y obtiene un coeficiente que pondera la medida del módulo de la velocidad. De esta forma se obtienen también las medidas v_x, v_y . Con esto logramos que el GPS proporcione las 4 coordenadas (x, y, \dot{x}, \dot{y}) , necesarias para el correcto desarrollo de cualquier algoritmo de control x-y.

Para completar el sistema GNSS/INS se ha implementado un filtro de Kalman, en el cual se estiman las medidas del GPS $(\hat{x}, \hat{y}, \hat{\dot{x}}, \hat{\dot{y}})$ integrando las aceleraciones $\ddot{x}, \ddot{y}, \ddot{z}$ dadas por la IMU, y se corrigen cuando llega una señal GPS útil. El algoritmo de Kalman, como resultado, nos ofrece una estimación de los bias $(\delta a_x, \delta a_y)$ de los acelerómetros ya que han sido debidamente incluidos en el modelo.

Para la medición de la altura “z”, los tipos de sensores más usuales son los infrarrojos, ultrasonidos o sensores barométricos. Los primeros se usan para hacer medidas menores de 1,5 metros. Los ultrasonidos proporcionan medidas hasta los 3 o 6 metros dependiendo del tipo del sensor. Su desventaja es su reducido alcance y que dependiendo del tipo de suelo que haya debajo (cesped, una alfombra, etc.) se pueden obtener medidas erróneas.

Por último los sensores barométricos, pueden medir casi cualquier altura, ya que utilizan el cambio de presión en la atmósfera, este tipo de sensores es apto para exteriores, el inconveniente es que no puede detectar los obstáculos que hay debajo del vehículo y que cualquier ráfaga de viento puede generar una sobrepresión o una depresión que genere una medida errónea de altura. Es por esto que la mejor medida consiste en la integración sensorial de distintos dispositivos.

En este proyecto se ha optado por dotar a la plataforma con un sensor barométrico [16] junto a un

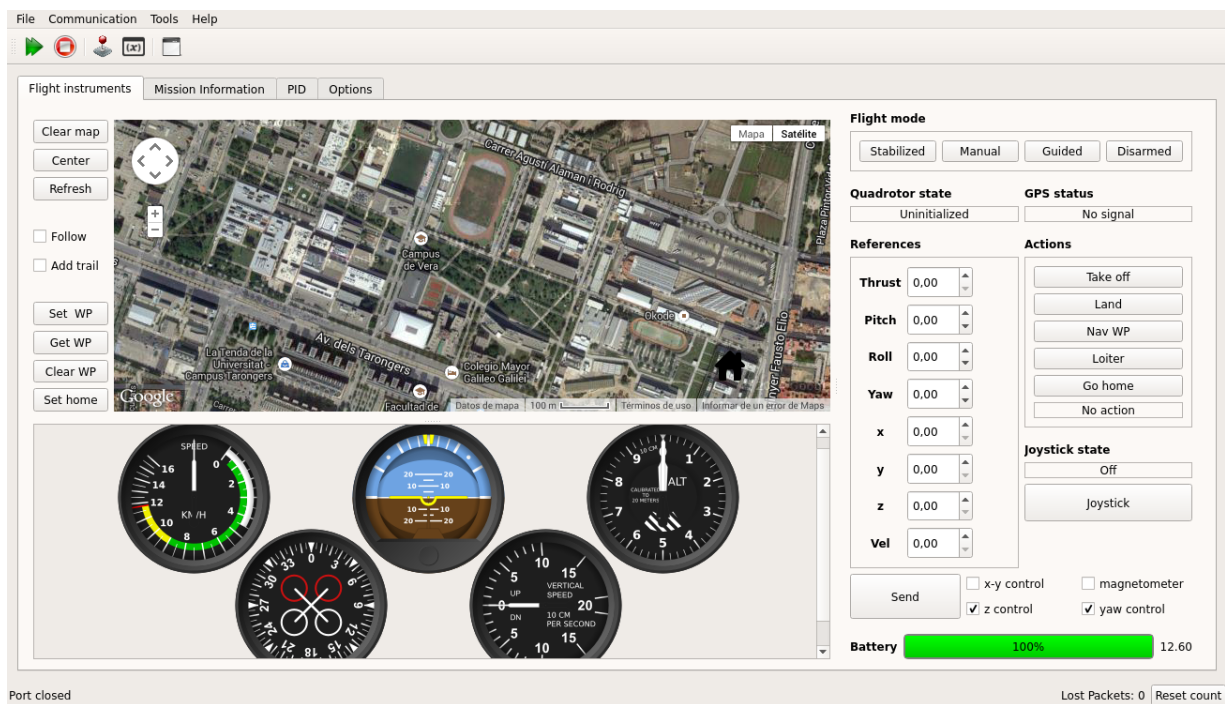


Figura 3: Interface (HMI) para la configuración del control en vuelo libre de un quadrotor.

sensor de ultrasonidos [15] y fusionar las medidas. Se podría haber realizado el control de altura de la plataforma solo con el sensor barométrico pero se optó por incorporar también un sensor de ultrasonidos por dar una medida más precisa y fiable que el barómetro a cotas bajas, y así facilitar las maniobras de despegue y aterrizaje.

Como sensor de ultrasonidos se utiliza el *Ultrasonic Distance Sensor* de Parallax. El rango de funcionamiento de este sensor es de entre 2 cm y 3 m.

Para el sensor barométrico se ha optado por el modelo *MS5611-01BA03* de Measurements Specialties. Una de las ventajas que presenta este sensor es que incorpora un sensor de temperatura interno para corregir la medida de presión. Este sensor funciona para rangos de presión entre 10 y 1200 mbar y rangos de temperatura entre -45 y 80 °C. Es la solución más empleada en quadrotors concebidos para volar en exteriores.

2.2. Comunicación mini PC - UCT e interfaz (HMI)

Además de las unidades de control y sensores presentados anteriormente, resulta imprescindible disponer de un sistema de telemetría entre el mini PC de a bordo y un PC en la unidad de control terrestre (UCT). Este enlace se utiliza tanto para la monitorización de las distintas variables e informa-

ción proporcionada por los sensores del vehículo, como para el control del mismo.

La plataforma puede comunicarse con la UCT mediante WiFi y radiofrecuencia. Para el enlace WiFi se aprovecha el wifi integrado de la IGEP V2. La transmisión de información se lleva a cabo mediante sockets a través del protocolo TCP. El principal problema es que el alcance está limitado a 10-15 metros y por tanto no sirve para vuelos en exteriores. Para la comunicación por radio se han utilizado dos *Xbee PRO S1* [18] (una conectada por serie al mini PC y la otra directamente conectada a un PC en la UCT). Las Xbee son módulos de radiofrecuencia de bajo coste, bajo consumo y que además permiten crear redes extensibles. La principal ventaja respecto a la comunicación por WiFi es que el alcance puede, teóricamente, llegar a 1,5 kilómetros.

La visualización de todas las variables y constantes de interés, además del control del UAV, se realiza por medio de una interfaz hombre máquina (HMI). Este software ha sido desarrollado en Qt, una biblioteca multiplataforma para desarrollar aplicaciones en C++ con interfaz gráfica de usuario o sin ella y que además cuenta con IDE propio.

Se trata de una interfaz sencilla que cuenta con una ventana principal donde se pueden visualizar las variables más importantes del vehículo y va-

rios bloques para realizar funciones adicionales. A grandes rasgos, la herramienta dispone de las siguientes funcionalidades:

Sección central con mapa de Google Maps, instrumentos de vuelo en la parte inferior (horizonte artificial, altímetro, brújula, indicador de velocidad en el plano horizontal e indicador de velocidad vertical). Asimismo, en el bloque Mission Information es posible exportar y visualizar estos parámetros en forma de gráficos. En la parte derecha hay un panel para enviar referencias al vehículo, así como cambiar su estado de vuelo. La interfaz también está preparada para recibir información sobre el estado de vuelo o de la cobertura GPS.

Mención aparte merece el bloque dedicado al ajuste online de las constantes de control (proporcional, derivativa e integral). Notar que, dado que los parámetros físicos del quadrotor pueden variar para diferentes configuraciones, el ajuste de los parámetros de control se debe realizar a partir de la experiencia del ingeniero de control realizando pequeños ensayos hasta alcanzar las prestaciones de control deseadas.

Por último, la interfaz también permite al usuario el control del vehículo a través del teclado o de un joystick. En particular, el offset de velocidad de los motores podrá ser controlado mediante teclado, así como las referencias sobre el roll y pitch.

3. Modelado dinámico y algoritmo de control

3.1. Modelo dinámico

Para el cálculo del modelo matemático se utiliza la aproximación de las ecuaciones de Euler-Lagrange [8]. Las coordenadas generalizadas del helicóptero se presentan como:

$$q = (x, y, z, \psi, \theta, \phi) \in R^6 \quad (1)$$

donde (x, y, z) denotan las posición del centro de masas del helicóptero, y (ψ, θ, ϕ) describen los tres ángulos de Euler, que representan la orientación del helicóptero. Después de una serie de cálculos y simplificaciones, el modelo matemático se puede representar como [7]:

$$m\ddot{x} = -u \sin \theta \quad (2a)$$

$$m\ddot{y} = u \cos \theta \sin \phi \quad (2b)$$

$$m\ddot{z} = u \cos \theta \cos \phi - mg \quad (2c)$$

$$\ddot{\psi} = u_\psi \quad (2d)$$

$$\ddot{\theta} = u_\theta \quad (2e)$$

$$\ddot{\phi} = u_\phi \quad (2f)$$

donde m es la masa del helicóptero, g es la aceleración gravitacional, y u el empuje total. Las va-

riables u_ψ , u_θ y u_ϕ se definen como las estradas de control de los ángulos (ψ, θ, ϕ) , tal que:

$$u_\psi = k_\psi(V_1 - V_2 + V_3 - V_4) \quad (3a)$$

$$u_\theta = k_\theta(V_1 + V_2 - V_3 - V_4) \quad (3b)$$

$$u_\phi = k_\phi(V_1 - V_2 - V_3 + V_4) \quad (3c)$$

Donde k_ψ, k_θ, k_ϕ , son los parámetros de momentos de inercia (normalmente de valor desconocido a priori), y V_1, V_2, V_3, V_4 , las tensiones aplicadas a los motores, numerados en orden horario.

3.2. Algoritmo de control

La estrategia de control considerada en este trabajo se ha obtenido como un caso particular de la ley de control propuesta en [23]. La utilización de esta propuesta no lineal, tiene como ventaja que, a parte de su simplicidad, utiliza funciones de saturación para cada estado, lo cual proporciona un fácil ajuste de las ganancias.

De forma general, y para los tres ángulos se propone la siguiente ley de control:

$$\begin{aligned} u_{PD} &= f_s(k_d \dot{\alpha}, \zeta_1) - f_s(k_p(\alpha - \alpha_d), \zeta_2) \\ u_I &= f_s(u_I + k_i(\alpha - \alpha_d), \zeta_3) \\ u_\alpha &= u_{PD} + u_I \end{aligned} \quad (4)$$

donde $k_d, k_p, k_i, \zeta_1, \zeta_2, \zeta_3$ son constantes positivas, α es el ángulo considerado, y α_d es el valor del ángulo deseado. f_s se define como una función de saturación acotada por los valores $\pm \zeta_i$.

Para asegurar teóricamente la estabilidad global, y para cada una de las variables se tienen que cumplir las siguientes desigualdades $\zeta_2 > \zeta_1, k_d^2 > k_p$ (para más detalles ver [23]).

Para pequeñas variaciones en los ángulos de orientación, el control de altura “ z ” se puede considerar desacoplado del resto de variables, siendo el modelo equivalente un doble integrador, la ley de control utilizada será la misma que se propone en la ecuación 4. Para el diseño del control de posición, es posible reescribir el modelo (2b)-(2c) obteniéndose un desacoplamiento del desplazamiento lateral “ y ” con respecto al avance “ x ”. Se ha diseñado un algoritmo de control para la posición que permita al quadrotor desplazarse de un punto a otro en línea recta y a una velocidad constante que se le envía como parámetro de entrada. Además independientemente de la orientación ψ .

Si $\mathbf{X}_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$ representa una medida de coordenadas cualquiera, en un sistema de referencia con x apuntando al norte e y apuntando al este; y $\mathbf{R} = \begin{pmatrix} \cos \psi & \sin \psi \\ -\sin \psi & \cos \psi \end{pmatrix}$, con ψ definido en el eje z

que apunta al centro de la tierra. Entonces:

$$\begin{pmatrix} \mathbf{X}'_d \\ \mathbf{X}'_{gps} \\ \dot{\mathbf{X}}'_{gps} \\ \ddot{\mathbf{X}}'_{imu} \end{pmatrix} = \begin{pmatrix} \mathbf{R} & 0 & 0 & 0 \\ 0 & \mathbf{R} & 0 & 0 \\ 0 & 0 & \mathbf{R} & 0 \\ 0 & 0 & 0 & \mathbf{R} \end{pmatrix} \cdot \begin{pmatrix} \mathbf{X}_d \\ \mathbf{X}_{gps} \\ \dot{\mathbf{X}}_{gps} \\ \ddot{\mathbf{X}}_{imu} \end{pmatrix} \quad (5)$$

Es la transformación que hace que el control x-y sea independiente del yaw. Por otro lado, se definen los valores de saturación de la velocidad según:

$$\zeta_V = \begin{pmatrix} \zeta_{V_x} \\ \zeta_{V_y} \end{pmatrix} = V \cdot \begin{pmatrix} \sqrt{\frac{1}{1 + \left(\frac{y'_d - y'}{x'_d - x'}\right)^2}} \\ \sqrt{\frac{1}{1 + \left(\frac{x'_d - x'}{y'_d - y'}\right)^2}} \end{pmatrix} \quad (6)$$

De esta forma se consigue que las referencias de velocidad en cada eje se posicionen en un valor tal que el quadrotor se desplace de (x, y) a (x_d, y_d) en línea recta. Las siguientes ecuaciones muestran la ley de control aplicada:

$$e_V(k) = f_s(k_p \cdot [\mathbf{X}'_d(k) - \mathbf{X}'_{gps}(k)], \zeta_V) - \dot{\mathbf{X}}'_{gps}(k)$$

$$u_I(k) = \frac{h}{2} \cdot (e_V(k) + e_V(k-1)) + u_I(k-1)$$

$$u_{xy} = \left[k_{dv} \cdot \ddot{\mathbf{X}}'_{gps}(k) + k_{pv} \cdot e_V(k) + k_{iv} \cdot u_I(k) \right]$$

$$\begin{pmatrix} \theta_d \\ \phi_d \end{pmatrix} = f_s(u_{xy}, \zeta_\alpha)$$

donde las constantes $k_p, k_{dv}, k_{pv}, k_{iv}, \zeta_\alpha$ se corresponden con los parámetros de control y $\mathbf{X}_d, \mathbf{X}_{gps}, \dot{\mathbf{X}}_{gps}, \ddot{\mathbf{X}}_{imu}$ son las coordenadas de posición deseada, de posición y velocidad dadas por el gps y de aceleración dada por la IMU, respectivamente.

La tensión aplicada a los motores, será el resultado de la suma de las distintas acciones de control sobre los ángulos (ϕ, θ, ψ) y la acción para el control en z:

$$V_1 = u + u_\psi + u_\phi + u_\theta \quad (7a)$$

$$V_2 = u - u_\psi - u_\phi + u_\theta \quad (7b)$$

$$V_3 = u + u_\psi - u_\phi - u_\theta \quad (7c)$$

$$V_4 = u - u_\psi + u_\phi - u_\theta \quad (7d)$$

4. Resultados experimentales

El objetivo de esta sección es la validación y evaluación experimental de la plataforma desarrollada y sus prestaciones.

Las figuras 4 y 5 muestran la evolución de los ángulos de orientación Pitch y Roll, y sus respectivas velocidades angulares, durante un vuelo libre a punto fijo a una altura aproximada de un metro. En esta ocasión el algoritmo de control utilizado

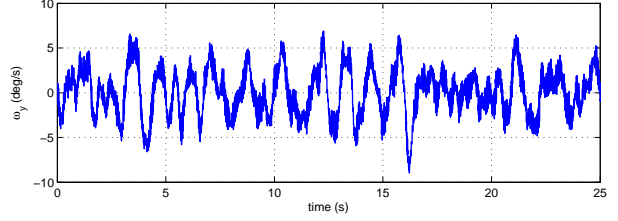
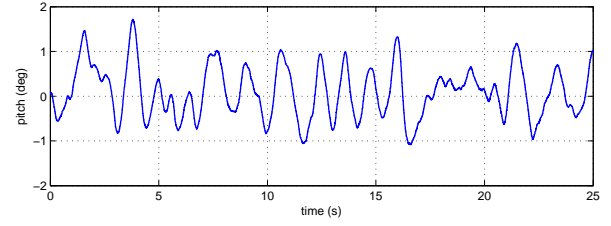


Figura 4: Control del pitch en hover

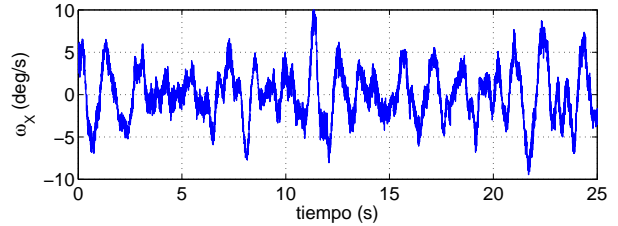
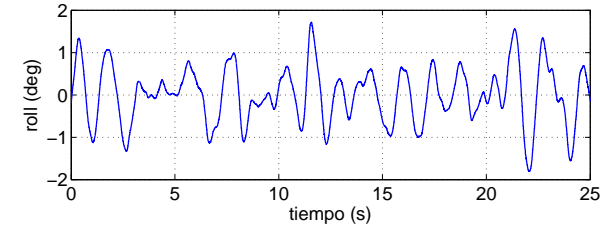


Figura 5: Control del roll en hover

para ambos ángulos es un PID, convenientemente ajustado utilizando la interfaz HMI.

Se observa el buen comportamiento de la plataforma, consiguiendo que ambos ángulos estén durante todo el vuelo dentro de un rango de ± 2 grados.

Las figuras 6 y 7 muestran la respuesta de los ángulos Pitch y Roll frente a perturbaciones. En este caso el quadrotor está realizando un vuelo libre a punto fijo cuando se le introducen perturbaciones en los ángulos manualmente. El algoritmo de control utilizado para ambos ángulos es un UDE y las figuras destacan su rápida respuesta para volver al vuelo a punto fijo y su gran estabilidad. La figura 8 muestra la evolución de la medida de altura y el valor de la acción de control durante un vuelo a punto fijo durante el cual se ha seguido un patrón de referencias de altura para evaluar el algoritmo de control, que en este caso es un UDE. Se observa que se siguen las referencias de forma bastante precisa y rápida.

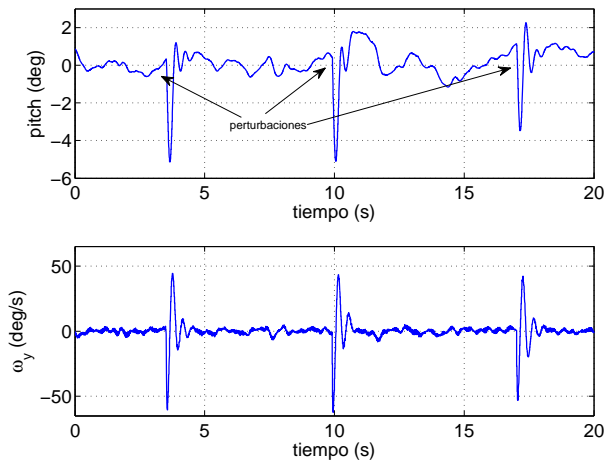


Figura 6: Respuesta en pitch ante perturbaciones

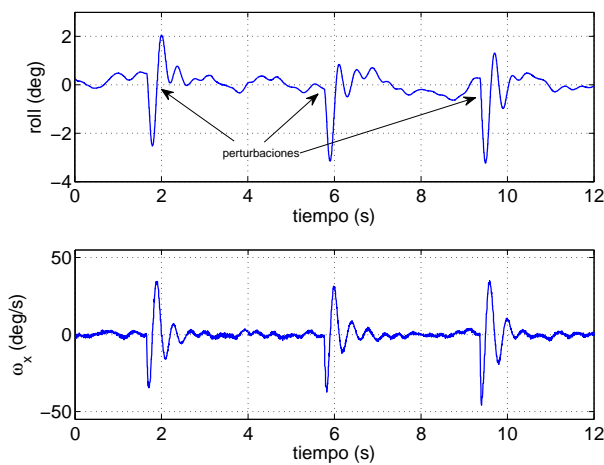


Figura 7: Respuesta en roll ante perturbaciones

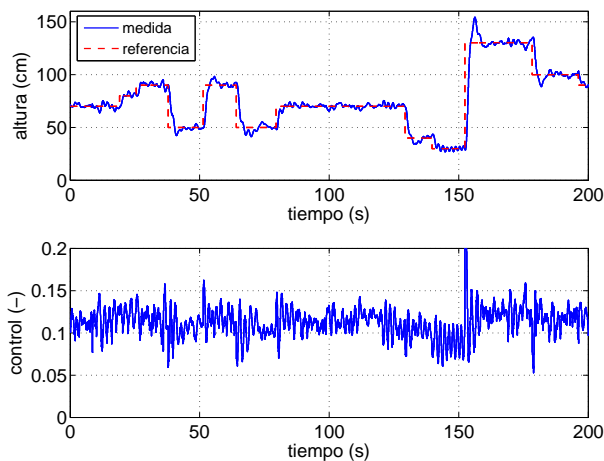


Figura 8: Control de altura siguiendo un patrón de referencias

Las figuras 9 y 10 muestran las medidas obtenidas tras someter al quadrotor a un desplazamiento de unos 25 m en dirección norte-sur.

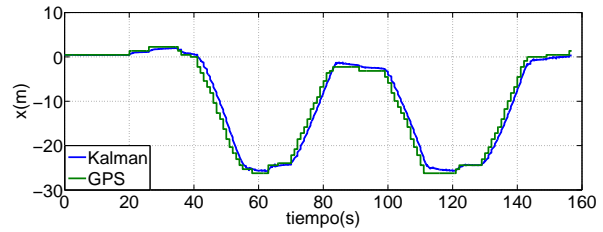


Figura 9: Medida experimental de posición tras efectuar un desplazamiento.

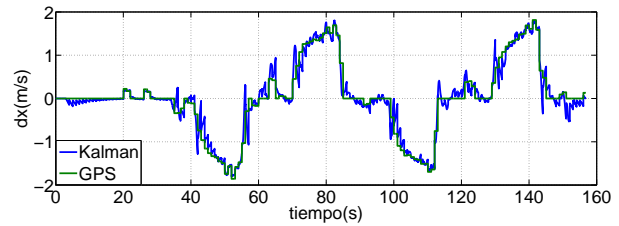


Figura 10: Medida experimental de velocidad tras efectuar un desplazamiento.

Se puede consultar un video del funcionamiento de la plataforma en [30].

5. Conclusiones y trabajos futuros

Para validar la plataforma, se ha desarrollado el control de vuelo tanto en interiores como en exteriores de varios quadrotors, los resultados experimentales, muestran el buen desempeño del sistema propuesto, así como una fácil y sistemática sintonización y evaluación de los algoritmos de control.

Agradecimientos

Este trabajo ha sido financiado con fondos pertenecientes al proyecto PROMETEOII/2013/004 de la Generalitat Valenciana, y por el Vicerrectorado de Investigación de la UPV bajo el programa PAID-06-12 ref. SP20120896.

Referencias

- [1] L. Acosta, J. Toledo, G.N. Marichal, S. Hernández, J.N. Rodríguez, and S. Torres. Diseño e implementación de una maqueta de un helicóptero de 4 rotores para un laboratorio de control. In *XXV Jornadas de Automática*, Ciudad Real, España, 2004.
- [2] P. Agarwal and T. Brady. Slam strategy for an autonomous quadrotor.
- [3] P. Balbastre, M. Lluesma, and I. Ripoll. Análisis y compensación de los retardos de planificación en sistemas de control. *RIAI*, 3(2):40–49, 2010.

- [4] M. Béjar and A. Ollero. Modelado y control de helicópteros autónomos. revisión del estado de la técnica. *RIAI*, 5(4):5–16, 2008.
- [5] A. Berna, J. Muñoz, G. Sanahuja, and P. Garcia. Desarrollo de una plataforma para la implementación en tiempo real del control de un mini-helicóptero de 4-rotores. 2010.
- [6] J. Borenstein. The hoverbot: An electrically powered flying robot, "<http://www-personal.umich.edu/~johannb/hoverbot.htm>".
- [7] P. Castillo, R. Lozano, and A. Dzul. *Modelling and Control of Mini-Flying Machines*. Springer-Verlag in Advances in Industrial Control, julio 2005.
- [8] I. Fantoni and R. Lozano. *Nonlinear Control for Underactuated Mechanical Systems*. Springer-Verlag, 2002.
- [9] Site internet de *Arduino*. "<http://www.arduino.cc>".
- [10] Site internet de la *IGEP*. "<http://www.isee.biz>".
- [11] Site internet de *YGE*. "<http://www.yge.de/artikel.php?search=yge30>".
- [12] Site internet *Xenomai*. "http://www.xenomai.org/index.php/Main_Page".
- [13] Site internet *ImuMPU6050*. "<http://www.invensense.com/mems/gyro/mpu6050.html>".
- [14] Site internet *Gps*. "<http://www.adafruit.com/products/746>".
- [15] Site internet *Ultrasonidos*. "<http://www.parallax.com/product/28015>".
- [16] Site internet *Barometro*. "<http://www.meas-spec.com>".
- [17] Site internet *Magnetometro*. "<http://www.aerospace.honeywell.com>".
- [18] Site internet *Xbee PRO S1*. "<http://www.digi.com>".
- [19] R. Lozano, editor. *Unmanned Aerial Vehicles, Embedded Control*. ISTE-WILEY, mayo 2010.
- [20] G.V. Raffo. *Modelado y Control de un Helicóptero Quad-Rotor*. PhD thesis, Master's thesis, Universidad de Sevilla, Dept. Ing. Sistemas y Automática, Sevilla, Spain, 2007.
- [21] G.V. Raffo, M.G. Ortega, and F.R. Rubio. Plataforma de pruebas para un vehículo aéreo no tripulado utilizando labview. Dept. Ingeniería de Sistemas y Automática, Escuela Superior de Ingenieros. Universidad de Sevilla, 2009.
- [22] H. Romero, S. Salazar, J. Escareño, and R. Lozano. Estabilización de un mini helicóptero de cuatro rotores basada en flujo óptico y sensores inerciales. *Revista Iberoamericana de Automática e Informática Industrial RIAI*, 7(2):49–56, 2010.
- [23] G. Sanahuja, P. Castillo, and A. Sanchez. Stabilization of n integrators in cascade with bounded input with experimental application to a VTOL laboratory system. *International Journal of Robust and Nonlinear Control*, 2009. DOI: 10.1002/rnc.1494.
- [24] M. Simons. *Model aircraft aerodynamics*. Model and Allied Publications, 1978.
- [25] H.J. Sussmann, E.D. Sontag, and Y. Yang. A general result on the stabilization of linear systems using bounded controls. *IEEE Transactions on Automatic Control*, 39:2411–2425, 1994.
- [26] A.R. Teel. Global stabilization and restricted tracking for multiple integrators with bounded controls. *Systems and Control Letters*, 18:165–171, 1992.
- [27] A.R. Teel. Semi-global stabilization of minimum phase nonlinear systems in special normal forms. *Systems and Control Letters*, 19:187–192, 1992.
- [28] A.R. Teel. Semi-global stabilization of the ball and beam using output feedback. In *American Control Conference*, San Francisco, Estados Unidos, 1993.
- [29] Boletín Oficial del Estado *25 de julio de 2014*. "<https://www.boe.es/boe/dias/2014/07/05/pdfs/BOE-A-2014-7064.pdf>".
- [30] Video del funcionamiento de la plataforma - *DEMO*. "<http://disicom.ai2.upv.es/images/xeon/portfolio/quadrotor-JA.mp4>".