

Identificación y control en cascada mediante inversión de no linealidades del cuatrirrotor para el Concurso de Ingeniería de Control CEA IFAC 2012

Lucía Hernández-Hernández^{*}, Jesús Pestana, Daniel Casares-Palomeque, Pascual Campoy, Jose Luis Sanchez-Lopez

Centro de Automática y Robótica. Centro mixto CSIC-UPM

Escuela Técnica Superior de Ingenieros Industriales, C/ José Gutiérrez Abascal, nº2, 28006, Madrid, España.

Resumen

En este trabajo se describen los análisis y los procesos realizados para obtener la identificación del modelo y el diseño del controlador, así como su posterior implementación, para el cuatrirrotor de tipo Parrot AR-Drone. El objetivo de este trabajo es el control del seguimiento autónomo de trayectorias en el plano horizontal XY, definidas de forma flexible por el usuario. La fase de identificación se caracteriza por la separación del modelo en una parte lineal y otra no-lineal, dando lugar a un modelo de tipo Hammerstein. El control está caracterizado por la utilización de un controlador en cascada, con lazo interno de control de velocidad y lazo externo de control de posición. Como elementos característicos, resaltan que: el controlador de velocidad incluye una linealización estática del modelo, y el lazo de posición incluye un bloque no lineal para disminuir el error de seguimiento de la trayectoria pero utilizando una lógica sencilla de tipo controlador de posición. Este estudio tiene como marco la participación en el Concurso de Ingeniería de Control 2012, “Control autónomo del seguimiento de trayectorias de un vehículo cuatrirrotor” organizado por el Comité Español de Automática y que tuvo lugar en las Jornadas de Automática de Vigo, el pasado septiembre de 2012. El trabajo presentado fue premiado en la primera y la segunda fase del concurso con la primera y tercera posición, respectivamente. Copyright © 2013 CEA. Publicado por Elsevier España, S.L. Todos los derechos reservados.

Palabras Clave:

Unmanned Aerial Vehicles, cuatrirrotor, identificación, control de la trayectoria, ingeniería de control.

1. Introducción

El cuatrirrotor es una plataforma VTOL (Vertical TakeOff and Landing), cuya característica principal, es la posesión de 4 hélices de paso fijo para su impulsión, que logran la estabilización del vehículo variando sus velocidades de giro. Este tipo de plataformas ofrece grandes posibilidades de aplicación práctica en ámbitos donde se requieren vehículos autónomos no tripulados que deben realizar trayectorias con rapidez y precisión en entornos con obstáculos cercanos. Ejemplos de estas aplicaciones son la inspección de edificios y puentes dañados, extinción de incendios, monitorización de paquetes en almacenes a través de la tecnología RFID (Radio Frequency IDentification), seguimiento de la vida salvaje y de especial importancia, la búsquedas de supervivientes entre los escombros de los edificios tras catástrofes humanitarias (Hoffmann *et al.*, 2008) (Michael *et al.*, 2010).

1.1. Avances en la materia

El problema que se plantea con esta clase de vehículos, es la necesidad de diseñar un buen sistema de control que permita gobernar y supervisar los movimientos que realiza en la ejecución de sus tareas. Un grupo de investigadores de la Universidad de Pennsylvania han sido capaces de ejecutar, con rendimientos

excelentes, trayectorias precisas usando leyes de control con PID en cascada y funciones de desacoplamiento no lineales (Mellinger *et al.*, 2010). Además, en esta investigación han empleado sistemas de captura de movimiento (por ejemplo, el sistema Vicon) para mejorar el control a partir de mediciones de la posición. Destaca el interés que están suscitando los multirrotors en la exploración de edificios, por ejemplo, para realizar trabajos de auxilio en escenarios de catástrofes humanitarias; donde los trabajos centrados en leyes de control PID combinadas con ecuaciones no lineales que desvinculan las acciones de comando del vehículo, han alcanzado resultados muy fructíferos. El inconveniente es que las leyes de control dependen de la precisión de la medida de posición y de los datos cinemáticos suministrados por estos sensores de captura de movimiento.

Por otro lado, hay grupos de investigación que han demostrado una buena capacidad de navegación autónoma utilizando los datos de posicionamiento GPS. El proyecto STARMAC (The Stanford Testbed of Autonomous Rotorcraft for Multi-Agent Control), de la Universidad de Stanford muestra el rendimiento en el seguimiento de la trayectoria al aire libre (Hoffmann *et al.*, 2008) basándose sólo en GPS. Su trabajo también se ha centrado en el estudio de las perturbaciones aerodinámicas que están presentes durante el vuelo, por ejemplo

^{*} Autor en correspondencia.

Correo electrónico: luciahhdez@gmail.com (Lucía Hernández-Hernández)

en la maniobra rápida de parada y marcha atrás (Hoffmann *et al.*, 2009), donde el vehículo tiene que cambiar su dirección de navegación para moverse hacia atrás. Otro estudio interesante tanto teórico como experimental fue hecho en el trabajo (Gillula *et al.*, 2010), donde la plataforma STARMAC era capaz de dar una vuelta completa al aire libre alrededor de uno de sus ejes horizontales.

1.2. Características del concurso CEA-IFAC 2012

Por su tamaño, manejabilidad y prestaciones (comunicaciones Wifi para interactuar con el PC, un sistema de guiado inercial compuesto por acelerómetros y giroscopios, un altímetro ultrasónico y dos cámaras de video), el cuatrirrotor AR-Drone, de la compañía Parrot, reúne todas las características necesarias para trabajar en el modelizado y control de este tipo de vehículos, en los ámbitos de la docencia e investigación.



Figura 1: El AR-Drone ejecutando, de modo autónomo, parte de la trayectoria en la prueba final de la segunda fase del Concurso en Ingeniería de Control 2012, “Control autónomo del seguimiento de trayectorias de un vehículo cuatrirrotor”

El Concurso en Ingeniería de Control 2012, “Control autónomo del seguimiento de trayectorias de un vehículo cuatrirrotor” organizado por el CEA, propuso el uso del AR-Drone para el desarrollo de los objetivos en las dos fases de la competición:

- Fase 1: Partiendo de un paquete de datos suministrado por la organización, que incluía variables de posición (x y y) de velocidad (v_x y v_y) y las manipuladas (las referencias *RefPitch* y *RefRoll*), obtuvimos, en primer lugar, un modelo que reproduce el comportamiento dinámico de las variables de posición frente a variaciones en las variables manipuladas. Posteriormente, se diseñó un controlador de seguimiento de trayectorias sobre el plano horizontal XY. Los organizadores simplificaron el problema de control ignorando la rotación del cuatrirrotor en el ángulo *yaw*. El controlador implementado puede ser fácilmente aumentado para rotaciones en *yaw* mediante el bloque de desacoplamiento presentado en (Mellinger *et al.*, 2010). La evaluación de la calidad del modelo y del controlador en simulación se facilitó gracias al suministro, por parte de la organización, de unas funciones en

Matlab©/Simulink© que puntuaban el desempeño de ambos componentes (Blasco *et al.*, 2012).

- Fase 2: Consistió en la implementación real y ajuste del controlador, utilizando una plataforma software, TrackDrone Lite® (García-Nieto *et al.*, 2012), suministrada, una vez más, por la organización. El software permite la realización de experimentos a través de una sencilla interfaz gráfica y la inclusión del controlador a través de una librería dinámica, o “dll”, programada por los concursantes.

Los trabajos y resultados de la identificación del modelo, que se presentaron en la primera fase se detallan en la sección 2 de este documento. Consecutivamente, en la sección 3, se especifica las características del controlador que se diseñó para la fase 1 y que posteriormente se implementó en el AR-Drone para participar en la segunda fase de la competición. En la cuarta sección se analizan diversos ensayos experimentales realizados tras la citada competición.

2. Identificación, modelizado del comportamiento

2.1. Modelo teórico

El cuatrirrotor se puede modelizar como un sólido rígido al que se le añaden los cuatro propulsores (hélices). Cada hélice genera una fuerza T_i y un momento M_i , verticales de dirección fija, sobre el cuerpo del vehículo. El impulso (1) y el momento (2) se calculan a partir de la velocidad de las hélices w_i , y de dos constantes K_t y K_m , respectivamente (Michael *et al.*, 2010):

$$T_i = K_t \cdot w_i^2 \quad (1)$$

$$M_i = K_m \cdot w_i^2 \quad (2)$$

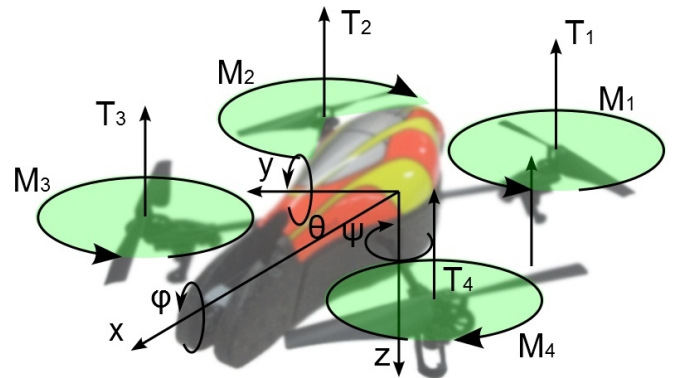


Figura 2: Esquema de fuerzas (T), momentos (M) y del sistema de referencia ligado al AR-Drone, indicándose los ángulos de Euler, *yaw* (\mathbb{E}), *pitch* ($^\circ$) y *roll* ($\{\}$).

Por tanto, haciendo un esquema general de fuerzas, el diagrama de cuerpo libre del cuatrirrotor es el siguiente (Figura 2). El vehículo realiza giros entorno a los ejes x , giro en *pitch* ($^\circ$), y y , giro en *roll* ($\{\}$), aprovechando los pares de fuerzas formados por los impulsos T_i . Los giros alrededor del eje z , giro en *yaw* (\mathbb{E}), se obtienen aprovechando los momentos M_i , cuya acción promedio es nula cuando el vehículo está quieto en el aire. La altura se controla mediante el valor promedio de los impulsos

T_i . Todos estos lazos de control se cierran en la tarjeta controladora a bordo del vehículo (o autopiloto) cuyo firmware no es posible modificar, y que es suministrado por el fabricante del AR-Drone.

El movimiento en el plano horizontal se obtiene inclinando el vehículo en *pitch* y en *roll* de manera que el impulso que contrarresta el peso, genera una aceleración horizontal. En los multirrotores, estos lazos de control, junto a la selección de las referencias de *yaw* y de altura, se realizan habitualmente en un computador a parte, externo al autopiloto. En base al diagrama de la figura 2, las ecuaciones de la dinámica del vehículo son:

$$I_{xx} \cdot \ddot{\phi} = \mathcal{E} \cdot \dot{\phi} \cdot (I_{yy} - I_{zz}) + l \cdot (T_1 + T_2 - T_3 - T_4) \quad (3)$$

$$I_{yy} \cdot \ddot{\psi} = \mathcal{E} \cdot \dot{\psi} \cdot (I_{zz} - I_{xx}) + l \cdot (T_1 + T_4 - T_2 - T_3) \quad (4)$$

$$I_{zz} \cdot \ddot{\theta} = \mathcal{E} \cdot \dot{\theta} \cdot (I_{zz} - I_{xx}) - \sum M_i \quad (5)$$

$$m \cdot a_x = (-\sin \phi \cdot \sin \mathcal{E} - \sin \psi \cdot \cos \phi \cdot \cos \mathcal{E}) \cdot \sum T_i - k_{aero} \cdot v_x^2 \quad (6)$$

$$m \cdot a_y = (-\cos \phi \cdot \sin \psi \cdot \sin \mathcal{E} + \sin \phi \cdot \cos \mathcal{E}) \cdot \sum T_i - k_{aero} \cdot v_y^2 \quad (7)$$

$$m \cdot a_z = m \cdot g - \cos \phi \cdot \cos \psi \cdot \sum T_i \quad (8)$$

En el caso de la competición el movimiento se restringe al plano XY, ya que la altura es constante, de donde $d T_i \approx mg$; y el giro en *yaw* es siempre nulo. Teniendo en cuenta, además, que los ángulos de *pitch* y de *roll* están controlados por el autopiloto interno del vehículo y realizando la aproximación para ángulos de *pitch* y de *roll* pequeños, el modelo se simplifica a (9), (10), (13) y (14):

$$\Phi(s) \approx G_\phi \cdot \Phi_r(s) \quad (9)$$

$$\Theta(s) \approx G_\theta \cdot \Theta_r(s) \quad (10)$$

$$m \cdot a_x = (-\psi \cdot \cos \mathcal{E} - \phi \cdot \sin \mathcal{E}) \cdot \sum T_i - k_{aero} \cdot v_x^2 \quad (11)$$

$$m \cdot a_y = (-\psi \cdot \sin \mathcal{E} + \phi \cdot \cos \mathcal{E}) \cdot \sum T_i - k_{aero} \cdot v_y^2 \quad (12)$$

$$m \cdot a_x = -mg \cdot \psi - k_{aero} \cdot v_x^2 \quad (13)$$

$$m \cdot a_y = mg \cdot \phi - k_{aero} \cdot v_y^2 \quad (14)$$

En las anteriores ecuaciones: I_{xx} , I_{yy} e I_{zz} representan los momentos de inercia del cuerpo del vehículo, l es de la distancia entre los centros de dos hélices consecutivas. La masa del cuatrirrotor, es decir, la suma de las masas de todos sus componentes, se simboliza por m (Michael *et al.*, 2010). La constante k_{aero} es la constante de fricción aerodinámica del vehículo con el aire. Por último, G y G aproximan el desempeño del control interno del autopiloto para controlar los ángulos de *pitch* (ϕ) y de *roll* (ψ) a partir de sus valores de referencia $RefPitch(\phi_r)$ y $RefRoll(\psi_r)$.

2.2. Identificación del modelo teórico

El modelo teórico, representado por (9), (10), (13) y (14) requiere la obtención de los siguientes parámetros: la masa del vehículo, la constante de fricción aerodinámica y las funciones de transferencia que modelizan la respuesta del vehículo en *pitch* y en *roll*. Estos parámetros se pueden calcular como se explica a continuación:

- Pesando el vehículo con una balanza, se determina que la masa del mismo es 374 g.

- El tiempo de respuesta en *pitch* y en *roll* del AR-Drone está en torno a los 150-300 ms, dependiendo de la amplitud del comando y de la velocidad del vehículo. La ganancia estática de este lazo de control es aproximadamente 1.
- La constante de fricción aerodinámica se calcula en base a datos de velocidad en régimen estacionario para las variables (v_r y v_x) y (ϕ_r y ψ_r) en ensayos con entrada en escalón de $RefPitch(\phi_r)$ y $RefRoll(\psi_r)$ constantes. Se puede estimar un valor medio de $0.119 \text{ N}\cdot\text{s}^2/\text{m}^2$. Sin embargo, la curva de velocidad obtenida no se ajusta adecuadamente a los datos experimentales. Además, como se expone en la sección 2.3 esta constante determina un polo del sistema, al mismo tiempo que la mencionada correspondencia en régimen estacionario. Por estas dos razones se optó por utilizar el modelo presentado en la sección 2.3.

2.3. Modelo simplificado para la identificación

Para hallar un modelo simple del control se deben tomar sólo las características principales para el control (las que más limitan al controlador). De esta forma se asegura un ajuste adecuado del comportamiento del vehículo simulado respecto al vehículo real.

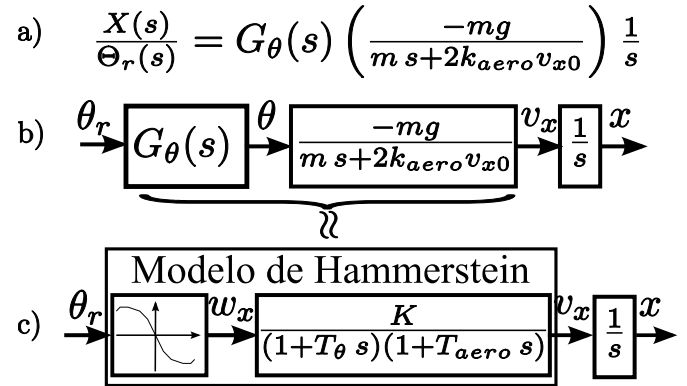


Figura 3: (a) Modelo dinámico simplificado del AR-Drone para el movimiento a lo largo del eje x , deducido de (10) y (13) a través de linealización. (b) Diagrama de bloques inducido de la ecuación de la figura 3 (a). (c) Diagrama de bloques deducido a partir de la figura 3 (b), donde se incluye un modelo de Hammerstein seguido de un integrador. El modelo de Hammerstein consta de una función no lineal en serie con un modelo ARX. La función no lineal permite fijar el valor entrada-salida en régimen permanente.

Para obtener un diagrama de bloques del movimiento en x a partir de (10) y (13), se linealiza el término en v_x^2 , y mediante la transformada de Laplace se llega a la ecuación de la figura 3 (a). De dicha ecuación se obtiene el diagrama de bloques de la figura 3 (b), donde cabe resaltar que el valor de k_{aero} determina uno de los polos principales del sistema y la ganancia estática entre ψ_r y v_x . La inclusión de un bloque no lineal en el modelo permite fijar estos parámetros de manera independiente, de donde se deduce el diagrama de bloques de la figura 3 (c). Como se muestra en dicha figura esto conlleva modelizar al vehículo con un modelo de Hammerstein seguido de un integrador. Este tipo de modelos (Biagiola and Figueroa, 2009) permiten identificar las no linealidades estáticas que suelen aparecer debido a la saturación de los actuadores, a sensores con características no lineales y a rozamientos aerodinámicos, junto la dinámica lineal del vehículo.

Como se puede observar en la figura 2, el perfil aerodinámico del AR-Drone es diferente según la dirección de avance. El movimiento en las direcciones de los ejes x y y presentan el perfil aerodinámico mínimo y máximo respectivamente. Esto tiene como consecuencia que la respuesta en velocidad es diferente en ambas direcciones, tanto en términos de tiempo de respuesta como de ganancia o velocidad en régimen estacionario.

La utilización de dos modelos del tipo mostrado en la figura 3(c) permite modelizar esta asimetría en el perfil aerodinámico del vehículo. Al mismo tiempo permite capturar las características principales del movimiento del AR-Drone, las cuales son enumeradas a continuación:

- La respuesta del autopiloto del vehículo en *pitch* y *roll* se modeliza dentro de la función de transferencia ARX del modelo de Hammerstein.
- Rozamiento aerodinámico en x e y : la función no lineal del modelo de Hammerstein permite fijar la relación entre los valores en régimen permanente de las variables u_r y v_x para el movimiento en x , y $\{r$ y v_y para el movimiento en y . El polo debido al rozamiento se modeliza dentro de los modelos ARX del modelo de Hammerstein.
- En la competición, la configuración del Parrot satura internamente las acciones de control, de *pitch* y de *roll*, a una inclinación máxima de 20°-24°. Esto es un parámetro de configuración habitual en multirrotores, y se modela incluyendo una saturación a la entrada del modelo.
- El nivel de ruido de medida, ya que puede afectar a la acción derivativa del controlador.

De la exposición anterior se deduce que se pueden simplificar los polos rápidos presentes en (9-10, 13-14), tomando principalmente el tiempo de respuesta en *pitch* y *roll*, y los polos en la respuesta en v_x y v_y debidos al rozamiento aerodinámico.

2.4 Método de Identificación

El modelizado de un sistema puede realizarse por identificación tipo caja-negra, o por modelización teórica (usando ecuaciones físicas). El enfoque que proporcionó mejores resultados fue la identificación tipo caja-negra; donde se tuvieron en cuenta las características del vehículo expuestas en la sección 2.3.

El suministro directo de un paquete de datos muestreados cada 0.06 segundos, por parte de la organización del concurso, permitió realizar la identificación utilizando la “System Identification Tool” de Matlab®. Utilizando un diagrama de Simulink® y una función de evaluación (Blasco *et al.*, 2012) programada en Matlab®, suministrados por la organización, se pudo simular una serie de posibles modelos, compararlos con el paquete de datos y evaluar su calidad.

Los datos de identificación contienen información sobre la dinámica de las variables de posición x , y , v_x y v_y ante cambios en las referencias de los ángulos *RefPitch* y *RefRoll*. En concreto, la trayectoria consiste en una sucesión de ensayos en escalón de una duración total de 720 segundos. Los datos de identificación muestran un comportamiento desacoplado en las direcciones x e y . Es decir, los datos muestran que el *pitch* controla la velocidad del desplazamiento en x , y el *roll* la velocidad del desplazamiento

en y . Esto refuerza el uso de dos modelos separados para los movimientos en x y en y . En base a la experiencia adquirida, esta hipótesis es válida a velocidades menores que 1-1.5 m/s, y cuando la suma de las referencias de *pitch* y de *roll* se encuentra por debajo del límite de saturación impuesto por la configuración del autopiloto del vehículo.

Para la obtención de una identificación simple pero adecuada, se deben ajustar el número de polos, n_p , y ceros, n_z , de la parte lineal del modelo de Hammerstein. De la exposición anterior, principalmente de la sección 2.3, se observa que probablemente el sistema no tiene ceros, y que tiene dos polos o más. Los valores elegidos para n_p , y n_z se obtuvieron comparando múltiples modelos identificados, y se observó que el valor óptimo era $n_z = 0$, y $n_p = 2$. Los polos adicionales resultaban demasiado rápidos como para corresponder a una dinámica visible en los datos de identificación, sobrepasando el límite impuesto por el tiempo de muestreo calculable mediante el Teorema de Shannon.

2.5. Modelo completo

Tal y como se mencionó con anterioridad, el comportamiento del vehículo cuando se mueve en x está desacoplado con el comportamiento en y , por esta razón, el modelo que relaciona el *RefPitch* con v_x y el que relaciona *RefRoll* con v_y deben de obtenerse, y así se hizo, de forma independiente pero siguiendo procedimientos análogos. A partir de ahora los denominaremos *modelo en x* y *modelo en y* respectivamente. El modelo completo es el que se observa en la figura 4, pudiéndose apreciar que además de los bloques lineales y no lineales, se ha añadido una acción que satura las variables de control *RefPitch* y *RefRoll* entre -1 y 1, es decir, el valor en tanto por uno de los límites de inclinación máximos y mínimos fijados por la configuración del AR-Drone. Cabe señalar que los bloques de Simulink® usados para representar las partes no lineales son de tipo “Lookup Table”, cada uno de los cuales está definido por dos parámetros: un vector de valores de entrada y un vector de valores de salida, a partir de los cuales realiza una interpolación y extrapolación lineal. A continuación se muestran los vectores de entrada y salida (inU_x y $outW_x$) correspondientes al *modelo en x* y los correspondientes al *modelo en y* (inU_y y $outW_y$), cuyos valores afectan al modelo de Hammerstein obtenido con la “System Identification Tool”:

$inU_x = [-1, -0.6198, -0.5998, -0.3370, -0.3207, -0.3010, 0.1325,$

$0.3192, 0.6495, 0.7795, 1];$

$outW_x = [0.0262, 0.0301, 0.0302, 0.0247, 0.0238, 0.0232, -0.0102,$

$-0.0244, -0.0318, -0.0273, -0.0267];$

$inU_y = [-1, -0.7836, -0.6142, -0.2991, -0.2773, -0.3003, 0.7231, 1];$

$outW_y = [-0.0496, -0.0549, -0.0539, -0.0397, -0.0364, 0.0397,$

$0.0555, 0.0501];$

En el proceso de identificación se desarrollaron varios modelos hasta obtener el definitivo (figura 4). Los modelos se simulaban y evaluaban con unas funciones de Matlab®/Simulink® (Blasco *et al.*, 2012) y el paquete de datos suministrado por los organizadores. Dichos datos, provienen de ensayos sobre un modelo no-lineal del vehículo, con el que los organizadores evaluaron y puntuaron el modelo y el controlador durante la fase I del concurso y que no estaba a disposición de los concursantes. Las funciones devolvían una serie de indicadores numéricos, cuyo cálculo ha sido descrito y detallado por los

organizadores (Blasco *et al.*, 2012). De esos indicadores, se destaca: el rendimiento global (*GPP*) y los errores absolutos medios ($e_{m\ x}$ y $e_{m\ y}$) y máximos ($e_{max\ x}$ y $e_{max\ y}$) de las variables de posición. Los valores de los indicadores permitieron ir descartando modelos hasta obtener el definitivo.

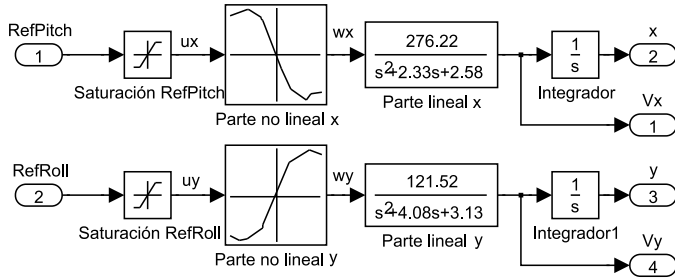


Figura 4: Modelo completo en Simulink© resultante de la identificación del cuatrirrotor. El *modelo en x* y el *modelo en y*, se componen, individualmente, de dos bloques, uno no lineal y otro lineal, que conjuntamente constituyen el modelo Hammerstein, precedidos de un bloque que satura las variables de control *RefPitch* y *RefRoll*, y con un integrador.

La tabla 1 muestra los valores obtenidos con el modelo de la figura 4 para dos evaluaciones. En la primera, realizada por los autores de este documento durante la etapa de modelizado, se compararon los resultados de la simulación del modelo, con los datos del paquete suministrado. En la segunda, realizada por los organizadores del concurso durante la evaluación de la fase I, se compararon los resultados de la simulación del modelo de la figura 4 con los del modelo no-lineal de los organizadores.

Tabla1: Comparativa de rendimientos globales (*GPP*) y errores absolutos [m] medios ($e_{m\ x}$ y $e_{m\ y}$) y máximos ($e_{max\ x}$ y $e_{max\ y}$) de las variables de posición del modelo de la figura 4, tras las evaluaciones realizadas por los autores de este documento y las realizadas por los organizadores del concurso en la fase I.

Evaluadores	<i>GPP</i>	$e_{m\ x}$	$e_{m\ y}$	$e_{max\ x}$	$e_{max\ y}$
Autores	19.0831	0.5017	0.4674	1.7607	2.6031
Organizadores	5.6347	0.5836	0.4477	1.7179	1.7478

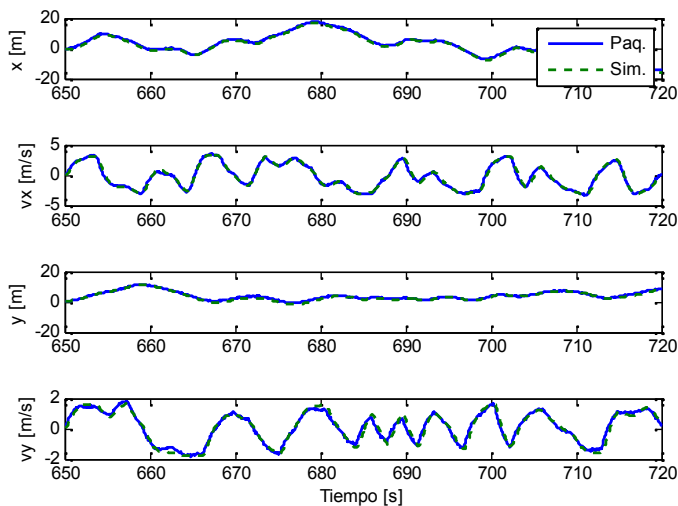


Figura 5: Comparativa entre una porción de los resultados de la simulación del modelo de la figura 4 (Sim.) y los datos del paquete suministrado (Paq.).

El modelo completo final (figura 4), es el resultado de la identificación de un modelo no-lineal del AR-Drone y no el

resultado de la identificación del modelo real. El objetivo del mismo, era usarlo para diseñar un controlador. Sin embargo, la calidad del modelo se ve respaldada por el hecho de que el controlador definitivamente implantado en experimentos reales con el AR-Drone y el controlador diseñado a partir del modelo completo, son idénticos estructuralmente y prácticamente iguales numéricamente.

3. Control para el seguimiento de trayectorias sobre el plano XY

3.1. Diseño de la arquitectura de control

El objetivo consiste en obtener un controlador de trayectoria que permita dar al cuatrirrotor las órdenes en forma de una secuencia de coordenadas que ubican puntos de referencia en el espacio, llamados “waypoints”, los cuales deben ser alcanzados por el vehículo. Con las medidas disponibles de posición (x y y) y velocidad (v_x y v_y) y actuando sobre los ángulos (*RefPitch* y *RefRoll*) debe conseguir el mejor seguimiento posible de trayectorias sobre el plano XY.

La complejidad del control radica en que el cuatrirrotor es un sistema con dos variables de entrada y cuatro de salida, y tal y como se ha mencionado anteriormente, el sistema se puede dividir en dos subsistemas cada uno de una entrada y dos salidas, los llamados *modelo en x* y *modelo en y*. Por lo tanto, disponemos en cada modelo de dos variables controlables (x y v_x para uno, y y v_y para el otro) que dependen de una variable manipulada (*RefPitch* y *RefRoll* respectivamente).

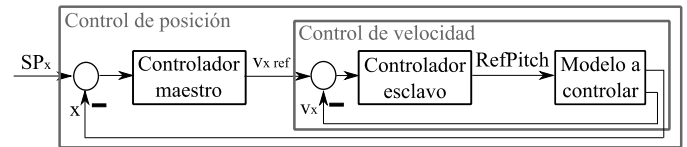


Figura 6: Control en cascada para el *modelo en x* (análogo al del *modelo en y*) que consta de un lazo interno con un controlador esclavo, encargado de controlar la velocidad, y de un lazo externo con un controlador maestro para la posición.

Son numerosos los procesos de control avanzado que se pueden aplicar a sistemas reales, pero entre ellos, el que consideramos más adecuado para el cuatrirrotor, en función del número de variables manipuladas y controladas, es un control en cascada, el cual permite controlar variables internas para, a través de ellas, poder controlar más fácilmente variables externas (Dominguez *et al.*, 2006). El control en cascada de cada *modelo en x* y en *y* consta pues de un lazo interno con un controlador esclavo, encargado de controlar la velocidad, y de un lazo externo con un controlador maestro para la posición (Figura 6). El motivo por el que considerar más apropiada esta configuración se debe a que al disponer, gracias a la cámara cenital que incorpora el vehículo, de las velocidades a las que viaja, es posible detectar y eliminar los efectos de las perturbaciones menores haciendo la respuesta de regulación del sistema más rápida y estable.

Los controladores calculados para esta estructura mediante la herramienta “Sisotool” de Matlab© son de tipo PID, es decir, controladores diseñados para modelos lineales, sin embargo, el cuatrirrotor tiene características no lineales, por lo que estos controladores podrían no funcionar adecuadamente, es por ello que se haya realizado una linealización. En el controlador del lazo

interno, se ha incluido un bloque correspondiente a la inversión estática de la parte no lineal del modelo identificado del cuatrirrotor (Figura 7), de tal modo que al unirlo en serie con este, se anula, aproximadamente, dicha no linealidad.

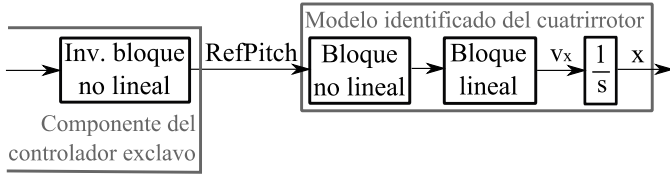


Figura 7: Linealización del modelo en x del cuatrirrotor por la adición, en el controlador, de un bloque inverso a la no linealidad del modelo identificado.

Es preciso mencionar que, la variable de entrada al lazo interno (v_{ref}) debe saturarse para evitar que tome valores superiores a los máximos y mínimos admisibles, correspondientes a las velocidades (v_x y v_y) máximas y mínimas del vehículo. Limitar su valor repercute en un aumento del tiempo de recorrido pero por el contrario mejora el seguimiento de la trayectoria con menores desviaciones respecto a la misma.

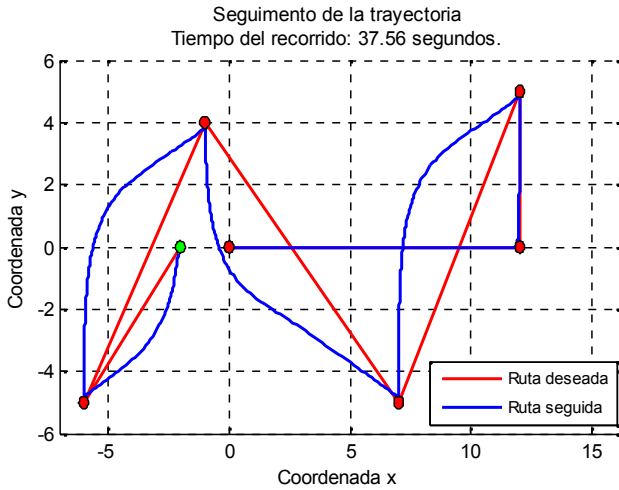


Figura 8: Simulación del seguimiento de la trayectoria (en metros) del cuatrirrotor, con reguladores P para el control de posición y reguladores PID para el control de velocidad.

Mediante una simulación con Maltab®, se ha observado el comportamiento del cuatrirrotor, cuando el controlador implementado es el descrito anteriormente. Se puede apreciar cómo se desvía la trayectoria cuando este parte de un “waypoint” para llegar a otro, es decir, el cuatrirrotor no sigue la trayectoria óptima en forma de línea recta (Figura 8).

$$\begin{cases} \vec{P} = (SP_x - x, SP_y - y); \\ \vec{v}_{ref} = (v_{x ref}, v_{y ref}); \\ \vec{v}_o = \frac{\vec{v}_{ref} \cdot \vec{P}}{\|\vec{P}\|^2} \cdot \vec{P} = (v_{ox}, v_{oy}); \end{cases} \quad (15)$$

Con el fin de evitar este comportamiento, se ha diseñado otro bloque de control que adapta las referencias de velocidad que suministran al controlador esclavo. Éste se encarga de tomar la

referencia de la velocidad a la salida del controlador maestro, tratarla y suministrar la nueva velocidad de referencia al lazo de control interno. Las operaciones que realiza, descritas en (15), son las siguientes: calcula el vector de proyección v_o de la velocidad v_{ref} sobre la recta que une su posición actual $[x, y]$ con la siguiente que se desea alcanzar $[SP_x, SP_y]$. De este modo se obtiene la velocidad corregida (Figura 9).

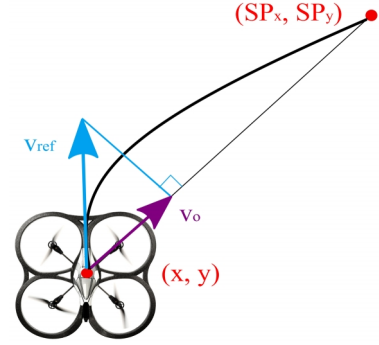


Figura 9: Rectificado de la velocidad inicial (v_{ref}) calculando su proyección (v_o) sobre el vector desplazamiento del punto inicial $[x, y]$ al final $[SP_x, SP_y]$.

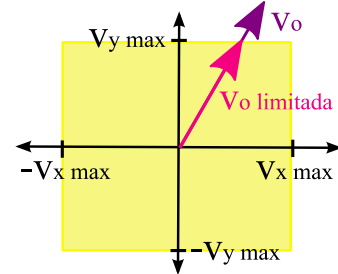


Figura 10: Reducción del valor de v_o , resultante del rectificado de la velocidad inicial (v_{ref}) (Figura 9), por sobrepasar un máximo ($v_{y max}$) impuesto por el vehículo.

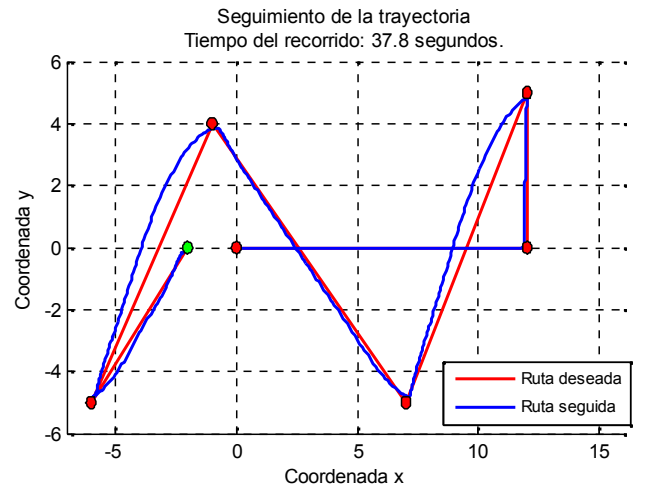


Figura 11: Simulación del seguimiento de la trayectoria (en metros) con la implementación del control corrector de la velocidad (Figura 9 y 10 y ecuaciones (15)), precedido por el regulador PD para el control de posición.

Además, tras el cálculo de la citada velocidad es necesario vigilar, de nuevo, que no supere, para ambas coordenadas, las restricciones de valores máximos o mínimos admisibles por el cuatrirrotor, y reducirla en el caso de que los sobrepasara

(Figura 10). Se comprobó en simulación, que este nuevo bloque de control mejora el seguimiento de la trayectoria, entendida como las líneas rectas que unen la secuencia de “waypoint” (Figura 11). Se ha considerado que se alcanza el “waypoint” cuando la distancia a él es igual o inferior a 0.2 metros.

3.2. Implementación del control

Tras los trabajos de simulación realizados con Matlab®, se ha procedido a implementar el controlador en el AR-Drone. La implementación se ha realizado en lenguaje C++, generando una librería dinámica o “dll” que posteriormente se utiliza con el programa TrackDrone Lite® (García-Nieto *et al.*, 2012). Dicho programa, facilitado por la organización, gestiona las comunicaciones con el AR-Drone. El programa recibe los paquetes de telemetría del vehículo y envía los comandos de control de manera cíclica. La organización facilitó la programación de la “dll” en C++, proporcionando a los equipos un proyecto de Visual Studio® con un patrón de la misma y un ejemplo sencillo de controlador.

TrackDrone Lite® ejecuta cada 0.06 segundos el controlador, en forma de una función llamada “void_cdecl Control”, la cual tiene un patrón entrada-salida prefijado por la organización y está contenida en la “dll”. Los parámetros de entrada del controlador son: los “waypoints” de la trayectoria, y la actual posición (x e y) y velocidad (v_x y v_y) del vehículo. Los n puntos que se desean alcanzar, los “waypoints”, vienen definidos por sus componentes en x por el vector SP_x y por sus componentes en y por el vector SP_y . Los parámetros de salida proporcionados a la herramienta TrackDrone Lite® son el valor de las acciones de control, es decir, los ángulos $RefPitch$ (°) y $RefRoll$ (°) puesto que las particularidades del concurso requieren obviar el ángulo de yaw . Además, dentro de “void_cdecl Control”, es necesario implementar la estrategia de seguimiento de la trayectoria. Al haber implementado un controlador posición, la estrategia optada fue simplemente recorrer el vector de “waypoints” secuencialmente, cambiando la referencia de posición cuando el vehículo está a 0.2 metros del “waypoint” que desea alcanzar.

La programación derivada de estos aspectos y las numerosas pruebas experimentales han determinado la obtención del controlador definitivo empleado en la competición. La figura 12 muestra el diagrama de bloques del algoritmo de control y seguimiento de la trayectoria programado en C++.

En los ensayos se ha prestado especial atención a minimizar el error de seguimiento y el tiempo de recorrido de la trayectoria, fijándose en parámetros como los que se describen en la tabla 2.

El controlador de posición en x de la figura 12 está detallado en la ecuación (16) así como el de y (17); y los controladores de velocidad en (18) y (19). Los bloques de la inversión de las no

linealidades, operan de modo inverso a los bloques no lineales descritos en el apartado 2.5.

$$PD_x = 1.2 \left[1 + 0.9 \frac{de_x}{dt} \right] \quad (16)$$

$$PD_y = 1.2 \left[1 + 0.9 \frac{de_y}{dt} \right] \quad (17)$$

$$PI_{v_x} = 0.097 \left[1 + \frac{1}{0.975} \int_0^{0.06} e_x dt \right] \quad (18)$$

$$PID_{v_y} = 0.019 \left[1 + 0.243 \frac{de_y}{dt} + \frac{1}{0.974} \int_0^{0.06} e_y dt \right] \quad (19)$$

4. Pruebas experimentales

Con el objetivo de mostrar el comportamiento del AR-Drone con el controlador diseñado implementado, se han realizado varios ensayos experimentales sometiéndolo a la ejecución de tres trayectorias diferentes. Dos de ellas tienen forma de cruz inscrita en un cuadrado, es decir, un cuadrado con sus dos diagonales, diferenciándose únicamente por su tamaño, con ellas se pretende comparar el comportamiento del vehículo, y en concreto su velocidad, en trayectos idénticos pero de mayor o menor longitud. Además las *cruces inscritas*, como a partir de ahora se denominarán, muestran cómo reacciona en trayectos con ángulos muy cerrados. La tercera trayectoria consiste en la ejecución de un hexágono, finalizando con un movimiento de ida al centro del hexágono y vuelta al punto anterior. Más adelante, se compararán numéricamente los resultados de las pruebas en la tabla 2.

Los ensayos se realizaron en el interior de un complejo deportivo con ausencia de viento. Las longitudes de las trayectorias diseñadas se compararon con las longitudes trazadas durante los vuelos reales tomando como referencia las líneas trazadas en las pistas deportivas. No obstante, la ausencia de magnetómetro en el AR-Drone provoca pequeñas desviaciones en el trazado de los vuelos.

4.1. Trayectoria de cruz inscrita pequeña

La trayectoria de referencia (ref en la figura 13) que debe seguir el cuatrirrotor comienza en la coordenada nula y continua con el recorrido del cuadrado en sentido contrario a las agujas del reloj. Una vez llegado al vértice inferior izquierdo, tiene que recorrer la diagonal, el lado superior, la otra diagonal y finalmente llegar al punto inicial de partida. Las posiciones que ha tomado el AR-Drone, es decir, la trayectoria real (pos en la figura 13) que ha

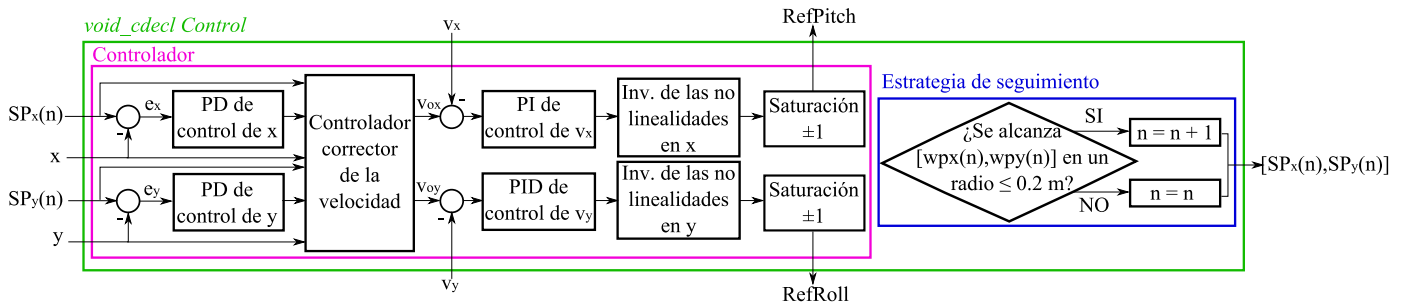


Figura 12: Controlador completo en cascada, compuesto por reguladores PI (18) y PID (19) y la inversión de la no linealidad del modelo, para el controlador de velocidad, y para el controlador de la posición, reguladores PD (16) y (17), saturaciones de la velocidad y un controlador corrector de la velocidad.

ejecutado en el ensayo, se pueden observar y comparar con la de referencia en las figuras 13 y 15.

El error de posición durante el seguimiento de la trayectoria, entendido como la distancia mínima a la trayectoria deseada, apenas supera los 25 cm (Figura 14). Si bien es cierto que no es un seguimiento exacto de la trayectoria, si tenemos en cuenta que el AR-Drone mide unos 50x50 cm y que a 20 cm del “waypoint” se considera que ha llegado a él, este error es mínimo.

Para estudiar el comportamiento del controlador de posición y apreciar que no existen sobreoscilaciones en la posición, se ha elaborado la gráfica de la figura 15. En ningún momento, las coordenadas de posición $[x, y]$ superan a las que deben alcanzar $[SP_x, SP_y]$, ya que el cuatrirrotor frena antes de sobrepasar el “waypoint”.

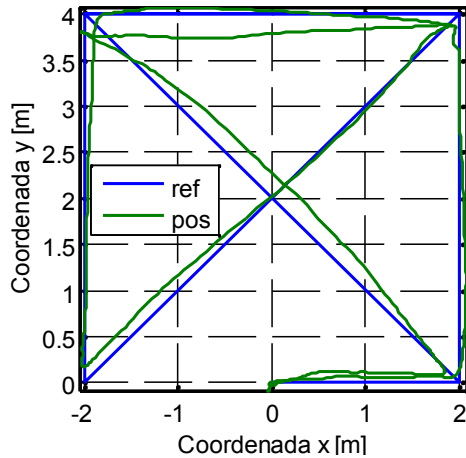


Figura 13: Superposición de la trayectoria de referencia y la real con el controlador implementado y comandándole al AR-Drone un recorrido experimental en forma de cruz inscrita pequeña. El controlador corrector de la velocidad permite el correcto trazado de las diagonales observándose escasas desviaciones.

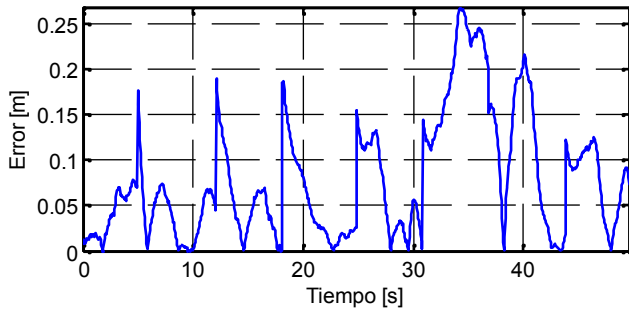


Figura 14: Error en el seguimiento de la trayectoria experimental de cruz inscrita pequeña. El error apenas supera los 25 cm, por lo que se puede considerar muy pequeño si se tiene en cuenta que el AR-Drone mide unos 50x50 cm y que a 20 cm del “waypoint” se ha establecido que ha llegado a él.

Por otro lado, recordamos que, el controlador corrector de la velocidad, interviene en evitar que la velocidad comandada (V_o) al controlador de velocidad, no supere unos valores máximos establecidos como $v_{x \max}$ y $v_{y \max}$ (Figura 10). El correcto funcionamiento del mismo se puede ver en la figura 16, donde sus componentes (v_{ox} y v_{oy}) se saturan en el valor establecido, 1 m/s. Dicho valor se estableció para evitar que el AR-Drone pasara de largo el “waypoint” por no darle tiempo a frenar.

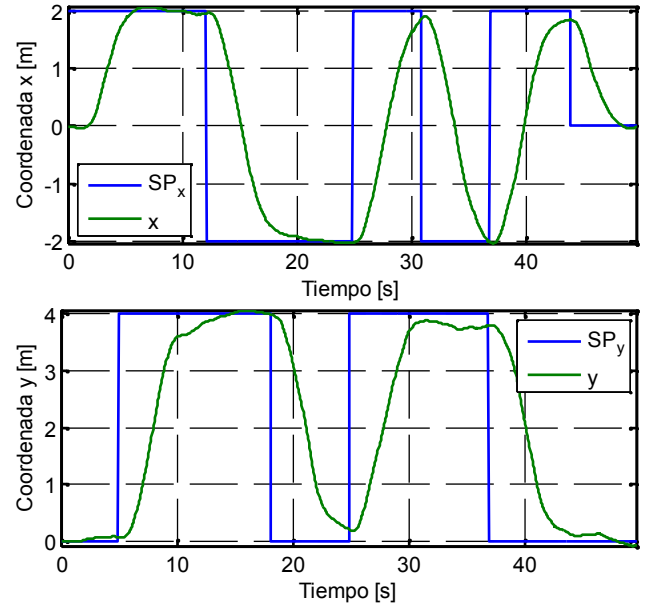


Figura 15: Comparativa entre la referencia de posición que debe seguir el vehículo $[SP_x, SP_y]$ y las posiciones que toma en la prueba real $[x, y]$. Durante los 50 segundos de recorrido experimental no se aprecia ninguna sobreoscilación y el AR-Drone no supera la posición de referencia ya que frena antes de sobrepasar el “waypoint”.

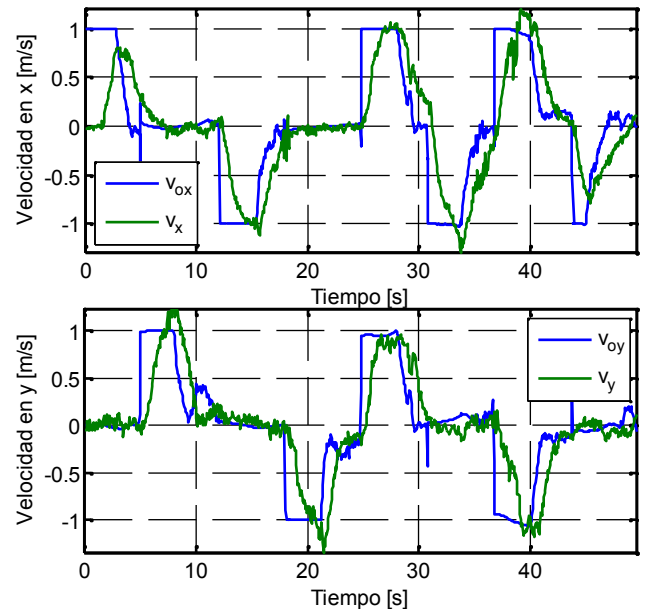


Figura 16: Comparativa entre la velocidad referenciada a la salida del controlador corrector de la velocidad (v_o) y la velocidad del vehículo durante el vuelo experimental (v), para cada componente x e y . Se observa la saturación en 1 m/s de v_{ox} y v_{oy} . Estos datos reales son los mandados por telemetría desde el AR-Drone al programa TrackDrone Lite®. Se observa un tiempo de respuesta en velocidad de uno o dos segundos.

Finalmente, en la figura 17 se puede ver la evolución de la salida del controlador, es decir, las variables manipuladas cuando actúa la función no lineal invertida y cuando la función no lineal se ha sustituido por la linealización de dicho bloque en torno a velocidad cero. Se puede observar como los valores de referencia de $pitch$ y de $roll$ contienen una cantidad de ruido limitada, más presente en $RefRoll$ que en $RefPitch$, y que los comandos varían

con suavidad, especialmente el comando de *pitch*. Esto indica que el controlador está bien calibrado, y que debería tener cierta robustez a pequeños cambios en la dinámica del vehículo.

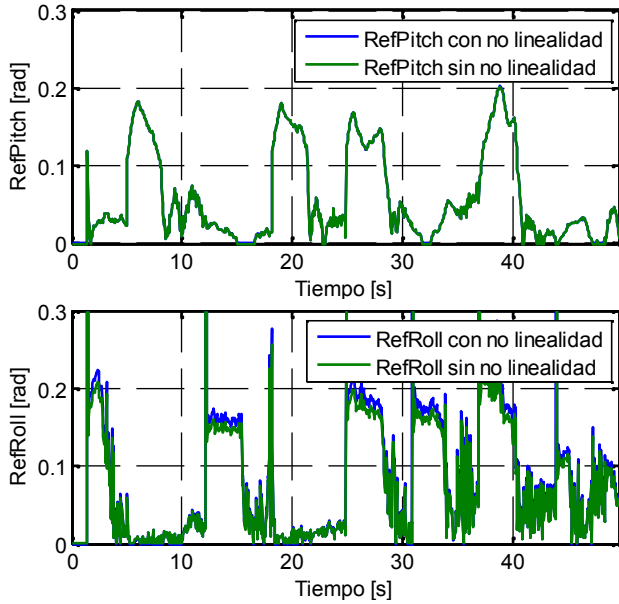


Figura 17: Comparativa entre las variables manipuladas *RefPitch* y *RefRoll* durante la trayectoria experimental de *cruz inscrita* pequeña, cuando actúa la función no lineal y cuando ésta se ha sustituido por la linealización del bloque de la inversión de la no linealidad en torno a velocidad cero. Se puede observar una cantidad de ruido limitada en *RefRoll* y *RefPitch*.

La superposición en *RefPitch* en la figura 17 demuestra que la no linealidad no tiene ningún efecto en esta acción, sin embargo no ocurre exactamente lo mismo con *RefRoll*. Para observar mayores diferencias entre las dos señales habría que comandar velocidades más altas al vehículo.

4.2. Trayectoria de cruz inscrita grande

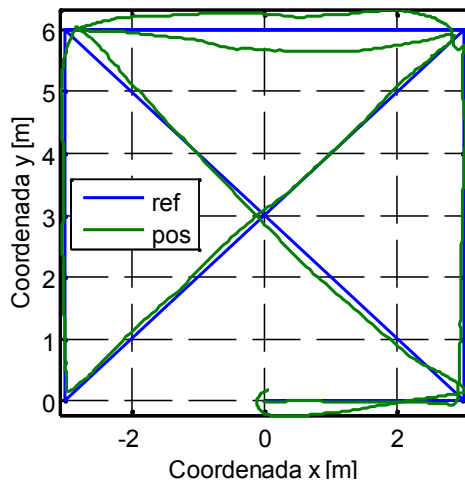


Figura 18: Superposición de la trayectoria de referencia y la real con el controlador implementado y comandándole al AR-Drone un recorrido experimental en forma de *cruz inscrita* grande. El controlador corrector de la velocidad permite el correcto trazado de las diagonales observándose menores desviaciones que en tramos en las direcciones de eje *y*.

En este caso, la trayectoria de referencia (*ref* en la figura 18) que sigue el AR-Drone, es idéntica a la del apartado anterior exceptuando sus dimensiones. En esta cruz inscrita, las distancias entre “waypoints” son mayores con el objetivo de comparar como le afecta al comportamiento del controlador el tamaño de la trayectoria.

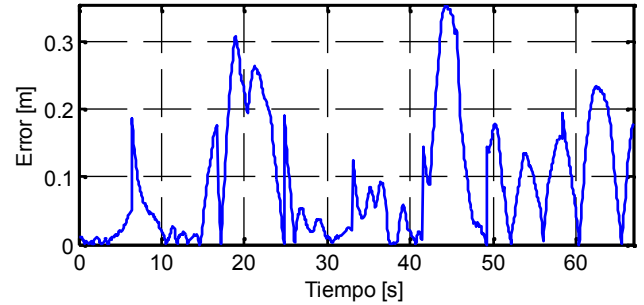


Figura 19: Error en el seguimiento de la trayectoria experimental de *cruz inscrita* grande. El error máximo se puede considerar muy pequeño si se tiene en cuenta que el AR-Drone mide unos 50x50 cm y que a 20 cm del “waypoint” se ha establecido que ha llegado a él.

El comportamiento de las variables de posición y de velocidad en este ensayo, son análogas a las de la *cruz inscrita* pequeña, no recalándose ningún comportamiento destacable salvo por el ligero aumento del error de seguimiento de la trayectoria (Figura 19).

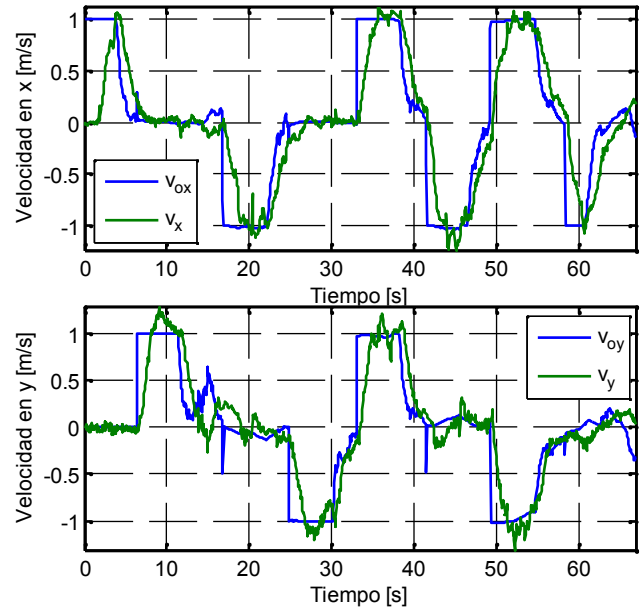


Figura 20: Comparativa entre la velocidad referenciada a la salida del controlador corrector de la velocidad (v_r) y la velocidad del vehículo durante el vuelo experimental (v), para cada componente *x* e *y*. Se observa la saturación en 1 m/s de v_{ox} y v_{oy} . Estos datos reales son los mandados por telemetría desde el AR-Drone al programa TrackDrone Lite®. Se observa un tiempo de respuesta en velocidad de uno o dos segundos.

Las dos trayectorias en forma de cruz inscrita en un cuadrado, permiten analizar el comportamiento del cuatrirrotor en diagonales y en direcciones que requieren sólo la acción de *pitch* o de *roll*, es decir, en direcciones sólo en *x* o en *y*, respectivamente. Si comparamos las figuras 13 y 18 podemos observar cómo, pese a ser más complicada, las diagonales las

ejecuta mucho mejor que aquellas que son en las direcciones del eje y .

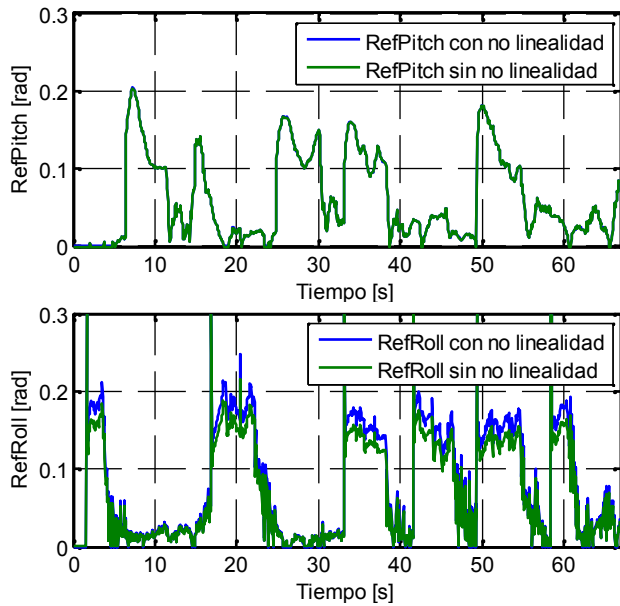


Figura 21: Comparativa entre las variables manipuladas $RefPitch$ y $RefRoll$ durante la trayectoria experimental de cruz inscrita grande, cuando actúa la función no lineal y cuando ésta se ha sustituido por la linealización del bloque de la inversión de la no linealidad en torno a velocidad cero. Se puede observar una cantidad de ruido limitada en $RefRoll$ y $RefPitch$.

También es posible confrontar las velocidades alcanzadas en ambas *cruces inscritas* (Figura 16 y 20) y los valores de las variables manipuladas (Figura 17 y 21), en tramos idénticos pero de diferente longitud. La velocidad media (Tabla 2) adquirida en los tramos de la *cruz inscrita* grande es algo mayor que en los del pequeño, ya que el aumento de la longitud le permite al AR-Drone alcanzar mayor velocidad, pero, se puede comprobar, que en ningún momento se sale de la trayectoria porque no le dé tiempo a frenar al llegar al “waypoint”.

4.3. Trayectoria hexagonal

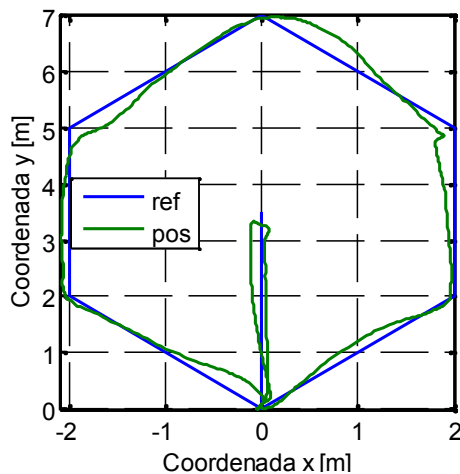


Figura 22: Superposición de la trayectoria de referencia y la real con el controlador implementado y comandándole al AR-Drone un recorrido en forma hexagonal. Se observa un excelente comportamiento del vehículo ante los cambios bruscos de trayectoria durante todo el vuelo experimental.

La trayectoria de referencia (ref en la figura 22) que debe describir el cuatrirrotor en esta prueba consiste en partir de un origen de coordenadas y trazar un hexágono hasta volver a esa posición. Tras esto, tiene que dirigirse al centro del hexágono y retroceder de nuevo a las coordenadas $[0,0]$, un movimiento que se suele denominar “stall turn”. El objetivo de este último tramo de retroceso es analizar cómo de bien se comporta ante esos cambios bruscos de trayectoria. Existen estudios (Hoffmann *et al.*, 2009) que prestan especial atención a este tipo de movimientos, porque introducen perturbaciones aerodinámicas al movimiento del vehículo.

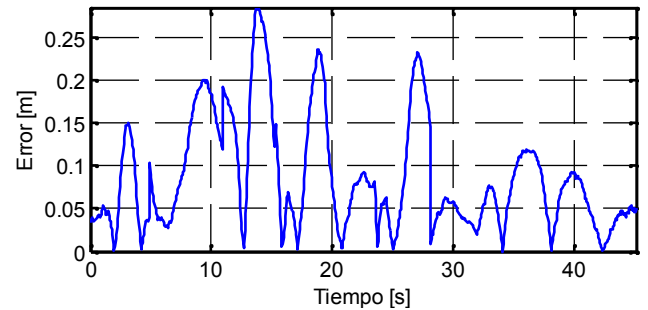


Figura 23: Error en el seguimiento de la trayectoria experimental hexagonal. El error máximo se puede considerar muy pequeño si se tiene en cuenta que el AR-Drone mide unos 50×50 cm y que a 20 cm del “waypoint” se ha establecido que ha llegado a él.

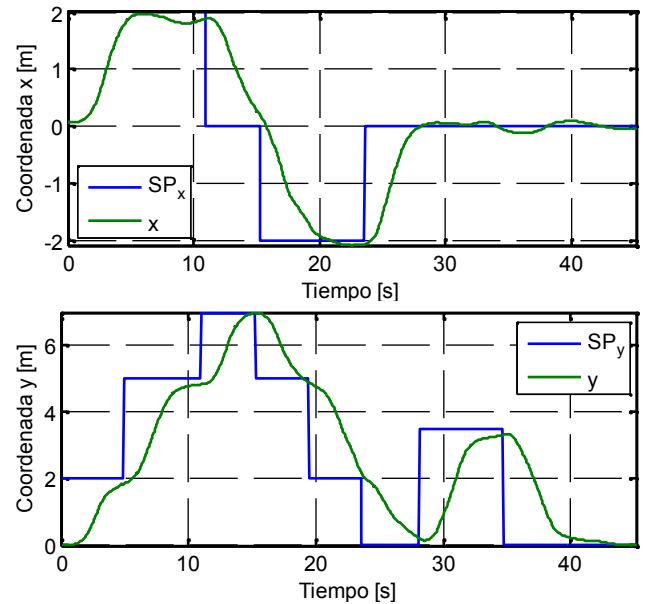


Figura 24: Comparativa entre la referencia de posición que debe seguir el vehículo $[SP_x, SP_y]$ y las posiciones que toma en la prueba real $[x, y]$. Durante los 45 segundos de recorrido experimental no se aprecia ninguna sobreoscilación y el AR-Drone no supera la posición de referencia ya que frena antes de sobrepasar el “waypoint”.

Pese a que este nuevo recorrido tiene cambios de dirección con ángulos más suavizados que en los dos anteriores, el efecto sobre el error de posición (Figura 23) es prácticamente igual, ya que el valor del mismo continúa siendo pequeño. Por lo tanto el controlador actúa igual de bien sin importar la trayectoria. El vuelo real del AR-Drone al ejecutar este recorrido se puede ver en el video (Hernández, 2012). En el minuto 00:11 se puede apreciar como el vehículo, antes de llegar a la coordenada $[2, 5]$ se frena y

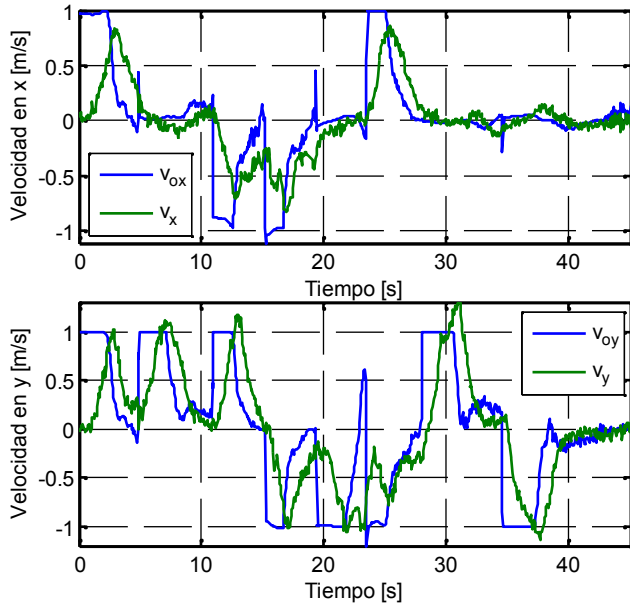


Figura 25: Comparativa entre la velocidad referenciada a la salida del controlador corrector de la velocidad (v_o) y la velocidad del vehículo durante el vuelo experimental (v), para cada componente x e y . Se observa la saturación en 1 m/s de v_{ox} y v_{oy} . Estos datos reales son los mandados por telemetría desde el AR-Drone al programa TrackDrone Lite®. Se observa un tiempo de respuesta en velocidad de uno o dos segundos.

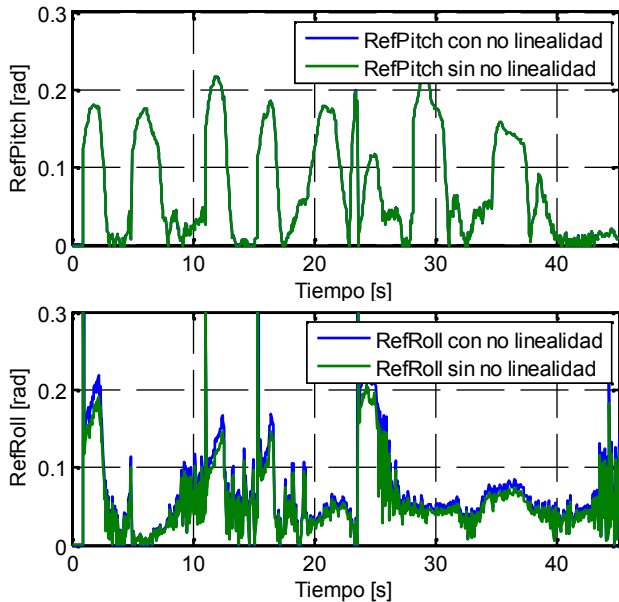


Figura 26: Comparativa entre las variables manipuladas $RefPitch$ y $RefRoll$ durante la trayectoria experimental hexagonal, cuando actúa la función no lineal y cuando ésta se ha sustituido por la linealización del bloque de la inversión de la no linealidad en torno a velocidad cero. Se puede observar una cantidad de ruido limitada en $RefRoll$ y $RefPitch$.

corrige la trayectoria desplazándose ligeramente hacia la derecha, cambio que se manifiesta en la figura 24 en la coordenada x a los 15 segundos aproximadamente. El movimiento de “stall turn” se inicia en el minuto 00:34 desde la posición [0,0]. Visualmente se aprecia que la posición nula desde la que parte al comienzo del video no es igual a la posición donde comienza a ejecutar el “stall turn”, esto se debe a que la estimación por odometría visual que realiza el programa TrackDrone Lite®, ha acumulado error en la

estimación de la posición. Se trata de un problema que puede ser resuelto mediante técnicas de medición de la posición sin odometría, y en todo caso, ajeno al propio diseño del controlador.

Al realizar medidas de flujo óptico el controlador puede afectar a la medida si introduce excesivo ruido en los comandos de *pitch* y de *roll*, pero como se observa en figura 26, estos comandos son relativamente suaves y el nivel de ruido en el comando de *roll* es limitado.

Siguiendo la misma línea, las respuestas en posición (Figura 24) y en velocidad (Figura 25) y las variables manipuladas (Figura 26), no tienen un comportamiento muy diferente a las del primer ensayo con la *cruz inscrita* pequeña. Los tiempos de respuesta y la saturación en la velocidad se observan entre los mismos valores que los citados en el apartado 4.1, lo que mantiene la afirmación de que el controlador es robusto para cualquier trayectoria de referencia.

5. Conclusiones

En este artículo se han descrito los análisis y resultados alcanzados en la obtención de un modelo y de un controlador para el Parrot AR-Drone.

En la etapa de identificación, caracterizada por el diseño de un modelo Hammerstein con parte lineal y parte no lineal, la sistemática seguida es fácilmente reproducible gracias a que se basa en la realización de ensayos de tipo escalón a altura y *yaw* constantes. El modelo obtenido, mostrado en la figura 4, fue evaluado por los organizadores del concurso comparándolo con su modelo no-lineal, obteniéndose la mejor puntuación durante la fase I. Los pequeños errores entre ambos modelos (tabla 1), demuestran la buena aproximación realizada durante la etapa de identificación, permitiendo que el controlador se haya elaborado partiendo de una base sólida. Esta afirmación se ve respaldada por el hecho de que el controlador definitivamente implantado en el AR-Drone y el simulado a partir del modelo, tienen la misma estructura y los parámetros numéricos son prácticamente iguales.

Tabla 2: Comparativa de errores [m] y velocidades [m/s] obtenidas en las diferentes pruebas experimentales

Trayectoria	Cruz inscrita pequeña	Cruz inscrita grande	Hexágono
Error máximo	0.2680	0.3560	0.2843
Error medio	0.0781	0.0921	0.0861
Velocidad máxima	1.6560	1.6420	1.3711
Velocidad media	0.6134	0.7009	0.5352

El controlador en cascada tiene tres aspectos fundamentales. El primero de ellos es la presencia de dos bloques que linealizan el modelo, que representan la inversa de las funciones no lineales del modelo, anulando así la falta de linealidad del mismo. Pese a que este bloque no lineal ayuda a una mejor identificación del modelo, su influencia sobre el controlador al invertir dicho bloque no es tan notable, pudiéndose sustituir por una ganancia equivalente a la linealización del bloque no lineal invertido. En segundo lugar, el controlador corrector de la velocidad, permite corregir las desviaciones de la posición con respecto a la trayectoria de referencia, sin la complejidad de controlar el error de seguimiento de la trayectoria en todo momento. Este hecho permite que no se sobrecontrole el vehículo, reduciendo muy sensiblemente los tiempos en los que realiza el trayecto. En tercer

y último lugar, las saturaciones que han sido aplicadas a las acciones de control y variables internas del controlador, permiten controlar de primera mano que sus valores no excedan los límites deseados y requeridos por el vehículo; redundando en un controlador más robusto y fiable.

Implementando el controlador diseñado, el AR-Drone es capaz de volar autónomamente controlado en posición sin cometer grandes desviaciones en la trayectoria. Los resultados de las pruebas experimentales realizadas y descritas en los apartados 4.1, 4.2 y 4.3, se muestran resumidas en los datos de la tabla 2. Se puede comparar como dichos datos se mantienen dentro de unos valores constantes independientemente de la trayectoria de cada ensayo experimental, siendo un índice del excelente comportamiento del controlador frente a este tipo de cambios. Los comportamientos, que ha mostrado en la realidad el vehículo, se ajustan, en todo momento, a lo esperado en las sesiones de simulación previas.

Todo el trabajo presentado en este artículo se ha visto recompensado y reconocido con la obtención del primer y tercer premio en la primera y segunda fase, respectivamente, del Concurso en Ingeniería de Control 2012, “Control autónomo del seguimiento de trayectorias de un vehículo cuatrirrotor” organizado por el Comité Español de Automática.

English Summary

Identification and cascade control by nonlinearities reversion of a quadrotor for the Control Engineering Competition CEA IFAC 2012.

Abstract

This paper describes and analyzes the processes carried out to obtain the model identification and the controller design, as well as its posterior implementation, for the quadrotor Parrot AR-Drone. The results are intended to control the autonomous tracking of that quadrotor on a path in the XY plane, flexible defined by the user. The identification phase is characterized by the division of the model into a linear and a non-linear part, resulting in a Hammerstein’s model. The control is characterized by the use of a cascade control with an inner speed control loop and an outer position control loop. As characteristic features, it is highlighted that: the speed controller includes a static linearization of the model, and the position loop includes a nonlinear block to reduce the tracking error of the trajectory but using a simple position controller logic. The context of this study is the Control Engineering Competition 2012, "Autonomous trajectory tracking control of a quadrotor vehicle " organized by

the Spanish Committee for Automation, which took place at the Automation Symposium in Vigo, last September 2012. The work presented was awarded in the first and second phase of the competition with the first and third prize, respectively.

Keywords:

Unmanned Aerial Vehicles, quadrotor, identification, trajectory control, control engineering.

Agradecimientos

Este trabajo está enmarcado dentro del proyecto CICYT DPI2010- 20751-C02-01 y también dentro de las investigaciones de varios autores en el marco de sus becas CSIC-JAE.

Referencias

- Biagiola, S. I., Figueroa, J. L. (2009), "Identificación Robusta de Modelos Wiener y Hammerstein", Revista Iberoamericana de Automática e Informática industrial, Vol. 6, Núm. 2, pp. 98-107.
- Blasco, F.X., García-Nieto, S., Reynoso-Meza, G. (2012), "Control autónomo del seguimiento de trayectorias de un vehículo cuatrirrotor. Simulación y evaluación de propuestas" Revista Iberoamericana de Automática e Informática industrial, Vol. 9, Num. 2, pp 194-199.
- Domínguez, S., Campoy, P., Sebastián, J. M., Jiménez, A. (2006), Control en el espacio de estado, (2ª edición). Pearson, Prentice Hall. Madrid.
- García-Nieto, S., Blasco, F.X., Sanchís, J., Herrero, J.M., Reynoso-Meza, G., Martínez, M.A., (2012), Software "TrackDrone Lite®" <http://hdl.handle.net/10251/16427>
- Gillula, J. H., Huang, H., Vitus, M. P., and Tomlin, C. J. (2010), "Design of Guaranteed Safe Maneuvers Using Reachable Sets : Autonomous Quadrotor Aerobatics in Theory and Practice". Robotics and Automation ICRA 2010 IEEE International Conference, pp 1649–1654.
- Hernández, L. (2012), "AR-Drone: trayectoria hexagonal con "stall turn"", Youtube video, disponible en <https://www.youtube.com/watch?v=3XCI3Tg28Yo>
- Hoffmann, G.M, Huang, H., Waslander, S., and Tomlin, C. (2009) "Aerodynamics and control of autonomous quadrotor helicopters in aggressive maneuvering". IEEE International Conference on Robotics and Automation, pp 3277–3282.
- Hoffmann, G.M., Waslander, S. L., Tomlin C. J. (2008), "Quadrotor Helicopter Trajectory Tracking Control", American Institute of Aeronautics and Astronautics, Vol. 44 pp 1-14.
- Mellinger, D., Michael, N., and Kumar, V. (2010), "Trajectory Generation and Control for Precise Aggressive Maneuvers with Quadrotors". The International Journal of Robotics Research, Vol. 31 Num. 5, pp 27-27.
- Michael, N., Mellinger, D., Lindsey, Q., Kumar, V. (2010), "The GRASP Multiple Micro UAV Testbed", IEEE Robotics & Automation Magazine, Vol. 17, Num. 3, pp. 56–65