

Control robusto \mathcal{H}_∞ para la velocidad de desplazamiento de un UAV en base a estimación de flujo óptico

Ghersin, A.S.^{†a}, Giribet, J. I.^{†b,*}, Luiso, J.^c, Tournour, A.^b

^aInstituto Tecnológico de Buenos Aires, Argentina; [†]CONICET, Argentina

^bDepartamento de Ingeniería Electrónica, Universidad de Buenos Aires, Argentina.

^cNeural-Labs, Calle Altimira 48, PL 2 08290 Cerdanyola del Valles, Barcelona - España.

To cite this article: Ghersin, A.S., Giribet, J.I., Luiso, J., Tournour, A. 2021. H_∞ robust displacement velocity control of a UAV based upon optical flow estimation. Revista Iberoamericana de Automática e Informática Industrial 18, 242-253. <https://doi.org/10.4995/riai.2021.14370>

Resumen

Se presenta el diseño del control para la velocidad de desplazamiento de un vehículo aéreo de seis rotores. La técnica de diseño utilizada es el control óptimo en H -infinito con el objetivo de conseguir rendimiento robusto ante la incertidumbre en el modelo de la dinámica de la velocidad de desplazamiento. Se considera que buena parte de la incertidumbre es atribuible a retardos de tiempo inciertos que introduce el propio algoritmo que se utiliza para estimar la velocidad de desplazamiento. El vehículo realiza a bordo la estimación de esta última a través de un sensor de flujo óptico implementado con una cámara y un procesador de alto nivel en el cual además se implementa la ley de control. Junto con el diseño del control, se muestra el procedimiento de identificación de sistemas utilizado para conseguir una descripción de la dinámica a través de una familia de plantas con incertidumbre dinámica global a través de la toma de datos experimentales. Finalmente se exhiben resultados experimentales con la implementación del sistema de control completo.

Palabras clave: Control robusto H -infinito, flujo óptico, identificación, UAV, incertidumbre dinámica.

H_∞ robust displacement velocity control of a UAV based upon optical flow estimation

Abstract

The design of a displacement velocity controller is presented for a six rotor aerial vehicle. H -infinity control is proposed in order to achieve robust performance in presence of dynamic model uncertainty. It is assumed that a considerable amount of uncertainty is due to time delays introduced by the algorithms employed. The estimation of the vehicle's displacement velocity is carried out on-board through an optical flow sensor implemented employing a camera and a high level processor as well as the H -infinity controller. Through experimental data, the system's identification procedure used to obtain a description of the plant as a family of models with global dynamic uncertainty is also presented as part of the design process. The implemented optical flow estimation methods are also presented as well as the tuning procedures employed which may affect the results of the system's identification and the control performance. Experimental results are presented with details regarding the implementation phase.

Keywords: H -infinity robust control, optical flow, identification, UAV, dynamic uncertainty.

1. Introducción

En los últimos años, los vehículos aéreos no tripulados (UAV) se han vuelto una herramienta de uso cada vez más fre-

cuente. La reducción en el costo de los componentes electrónicos, el desarrollo de nuevas baterías de mayor capacidad, menor peso y tamaño, han fomentado la proliferación de estos vehículos.

*Autor para correspondencia: jgiribet@fi.uba.ar

Uno de los subsistemas más importantes para lograr esta autonomía es el de navegación, guiado y control (NGC). El sistema de navegación utiliza información de diversos sensores y mediante un algoritmo estima la posición, velocidad y orientación del vehículo en cualquier instante de tiempo. Esta información es suministrada a un algoritmo de control que calcula los comandos para los actuadores, que permiten seguir la trayectoria establecida por el algoritmo de guiado, y así cumplir los objetivos de la misión.

En el diseño de los algoritmos de NGC existen requerimientos que deben tenerse en cuenta. Ciertas tareas requieren garantías estrictas en cuanto a los tiempos de ejecución mientras que otras requieren una capacidad de cómputo significativa para poder llevarse a cabo. Por ejemplo, el algoritmo de control puede no requerir una elevada capacidad de cómputo pero sí garantías en cuanto al tiempo de ejecución para lograr estabilizar al vehículo. Por otro lado, un algoritmo de navegación por imágenes puede requerir una elevada capacidad de cómputo para el procesamiento de las mismas, relegando cierta precisión en cuanto al período de ejecución. Frecuentemente, la arquitectura del sistema de NGC utiliza dos capas o niveles de *hardware* (Zhang et al. (2015), Azkarate et al. (2020)), que se destinan a tareas de bajo y alto nivel. Aquellas tareas que requieren garantías de tiempo de ejecución se alojan en una computadora de bajo nivel, mientras que las tareas computacionalmente más demandantes pueden ser ejecutadas en una computadora de alto nivel.

En este trabajo se presenta el sistema de NGC para un UAV de tipo multi-rotor, el cual utiliza una arquitectura basada en dos procesadores. El procesador de bajo nivel se encarga de ejecutar las rutinas de navegación y control que permiten estabilizar el vehículo en orientación y altura, mientras que en el procesador de alto nivel se ejecutan las rutinas que permiten controlar la velocidad de desplazamiento del UAV. Esto se debe a que la estimación de la velocidad de desplazamiento del vehículo (que en este caso se basa en navegación por imágenes) requiere una capacidad de cómputo mayor a la que ofrece el procesador de bajo nivel. La tarea de control de la velocidad de desplazamiento queda alojada en la computadora de alto nivel, que no solo ejecuta los algoritmos de navegación por imágenes sino que además ejecuta otras tareas como el guiado, la monitorización, el control de la misión, y las comunicaciones, entre otras.

La interacción entre el sistema físico y el software que ejecuta los diversos algoritmos, es uno de los objetos de estudio en el control de sistemas ciberfísicos. Muchas veces se recurre técnicas específicas para su resolución. Sin embargo en casos como el del presente trabajo, técnicas estándar como el control robusto ofrecen soluciones prácticas.

El diseño del algoritmo de control debe garantizar cierta robustez ante las perturbaciones, no sólo físicas sino también propias del software, por ejemplo los retardos provocados por tareas que se ejecutan concurrentemente en el procesador. Para lograr esto, en este trabajo se utilizó un diseño robusto para el control de la velocidad de desplazamiento del UAV. Debe mencionarse el trabajo de Noormohammadi Asl et al. (2020), donde se propone un control H_∞ para lidiar con las incertidumbres del modelo que representa la orientación del UAV. En el presente trabajo se muestra que esta técnica puede ser

utilizada para implementar el control de velocidad de desplazamiento del vehículo utilizando información de flujo óptico. El flujo óptico es frecuentemente utilizado para estimar el desplazamiento de un UAV, en particular en ambientes cerrados (ver Grabe et al. (2012), Honegger et al. (2013), Lim et al. (2012), Ma et al.(2019)). El flujo óptico también es susceptible de ser estimado en exteriores para aplicaciones de misión crítica en plataformas como la que se reporta en Madridano et al. (2020). Por su parte en Zhang et al. (2015), se muestra que de ser posible, para mejorar el rendimiento del algoritmo de control, es necesario sincronizar las mediciones de los distintos sensores de navegación, compensando en particular los retardos en el cálculo del flujo óptico. A diferencia de lo que ocurre en el mencionado trabajo, aquí se supone que no es posible sincronizar la información de los sensores de navegación, debido a que no se pueden determinar los retardos provocados por tareas de software que se ejecutan concurrentemente en el procesador de alto nivel junto con otras varias que son de una demanda computacional variable e incierta. Además, atento a limitaciones de capacidad computacional disponible de alto nivel, la fusión de los datos de sensores tampoco resulta conveniente. Por lo tanto se muestra la utilidad de una técnica basada en H_∞ que da garantías sobre la estabilidad del sistema aún ante estas incertidumbres lo cual constituye la principal contribución del presente trabajo junto con los resultados experimentales.

Técnicas basadas en aprendizaje de máquina (ver Bithas et al. (2019), Choi y Cha (2019), Fonnegra et al. (2019)) han demostrado ser de utilidad para lidiar con este tipo de problemas, no obstante lo cual son estrategias computacionalmente más costosas por lo que resulta más conveniente implementar la estrategia aquí propuesta teniendo en cuenta el hardware disponible en este diseño.

Se presentará la arquitectura del sistema de NGC y la metodología para el diseño del control del UAV. El trabajo está ordenado de la siguiente manera. En la Sección 2 se explica el funcionamiento y la dinámica del UAV multi-rotor, lo cual resultará útil para explicar cómo es posible dividir las tareas de control del UAV. En la Sección 3 se presentan los sensores de navegación utilizados, explicando las ventajas y limitaciones del sensor de flujo óptico que permite detectar el desplazamiento del UAV. En la Sección 4 se detalla la arquitectura del sistema de NGC y la estrategia utilizada para la implementación del control del vehículo. En la Sección 5 se presenta el vehículo utilizado, con el cual se realizaron los ensayos que permiten determinar su modelo dinámico. La Sección 6 presenta los resultados de la identificación de la dinámica que describe el desplazamiento del multi-rotor y la ley de control que permite garantizar la estabilidad del sistema considerando la incertidumbre en el modelo dinámico. La estrategia de control propuesta se valida experimentalmente en la Sección 7. El trabajo finaliza con las conclusiones en la Sección 8.

2. Funcionamiento de un multi-rotor

Existen diversos tipos de UAV. En particular los denominados multi-rotor han sido una de las plataformas más adoptadas tanto en el ámbito académico como en las aplicaciones. Esto se debe a que su mecánica es muy simple y poseen una gran maniobrabilidad, ofreciendo un uso muy versátil. Otro de

los atractivos de esta plataforma es su facilidad para pilotar. Para poder lograr un comportamiento adecuado, es necesario tener un algoritmo de control que se ejecute en la computadora que se encuentra a bordo del vehículo, que suele denominarse *auto-piloto*. Este algoritmo se encarga primero de mantener estable la orientación y la altura del UAV mediante un control por realimentación, y luego interpreta los comandos provenientes de un nivel superior que pueden tratarse tanto de comandos del piloto como de señales de referencia relacionadas con un lazo de control automático de posición o velocidad. De esta forma, este nivel superior se maneja con un sistema estable de respuesta adecuada a los comandos de orientación y altura.

Un multi-rotor está compuesto por una estructura en la cual se montan los motores con sus hélices, la batería que provee energía al sistema y la computadora de vuelo, que suele incluir los sensores de navegación (sensores inerciales, GPS, magnetómetros, barómetro) y los procesadores encargados de ejecutar los algoritmos de navegación, guiado y control.

En la Figura 1 se puede apreciar el vehículo utilizado para este trabajo, compuesto por: (1) Estructura del vehículo. (2) Batería. (3) Placa de la computadora de vuelo. (4) Placa de la computadora de vuelo utilizada para procesar información de las cámaras y ejecutar algoritmos que requieran mayor capacidad de cómputo. (5) Placa utilizada para el envío de datos de telemetría con el fin de monitorizar el estado del vehículo e información del vuelo para post-procesamiento. (6) Magnetómetro utilizado para medir el campo magnético terrestre y en general para obtener información de *yaw*. (7) Cámara instalada en la parte inferior del vehículo. (8) Sensor de ultrasonido que permite obtener información de la altura a la que vuela el vehículo, bajo ciertas condiciones.

Los actuadores de este sistema son los rotores (motores y hélices) que generan el empuje y los pares de fuerza necesarios para controlar el vehículo. Si bien todos los motores ejercen una fuerza ascendente, las hélices se configuran de forma tal que tres de los rotores giren en sentido horario y tres en sentido anti-horario, de forma tal que el par de fuerza neto nominal en el eje z del vehículo sea nulo (ver la Figura 2).

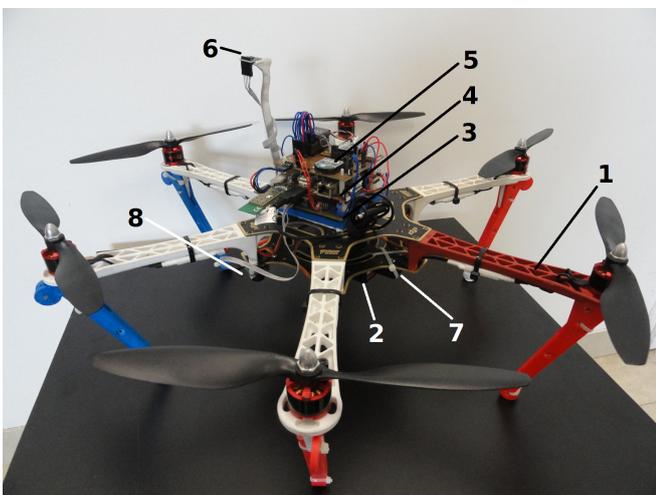


Figura 1: Vehículo hexa-rotor desarrollado en la UBA para la evaluación de algoritmos de control y navegación.

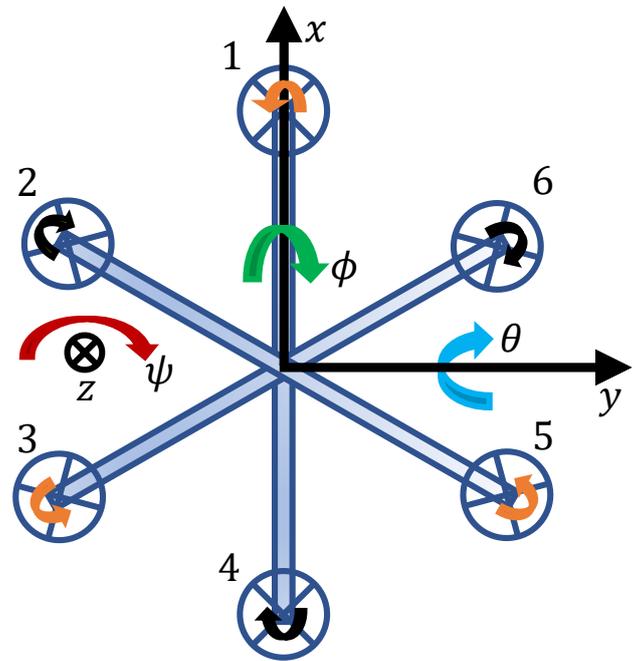


Figura 2: Definición de los ejes del hexa-rotor, sentidos de giro de los rotores y ángulos de Euler de *roll* (ϕ), *pitch* (θ) y *yaw* (ψ).

Siendo el vehículo en estudio un sistema mecánico subactuado, se busca primero estabilizarlo en los cuatro grados de libertad que se controlan directamente es decir altura y orientación. En la Figura 2 se puede observar una vista superior del vehículo con los ejes $x - y - z$ de la terna fija al cuerpo del mismo marcados. La terna se define con el eje x hacia el frente, el eje y hacia la derecha y el eje z apuntando al suelo. También están marcados los ángulos de Euler que describen la rotación del vehículo. Ángulo de *roll* (ϕ) en verde, rotación respecto del eje x ; ángulo de *pitch* (θ) en celeste, rotación respecto del eje y ; y ángulo de *yaw* (ψ) en rojo, rotación respecto del eje z . En la misma Figura están marcados en naranja los sentidos de giro antihorarios de los rotores 1, 3 y 5 con flechas naranjas mientras que los sentidos de giro horarios de los rotores 2, 4, y 6 se marcan con flechas negras.

Controlando la velocidad de giro de los motores dentro del rango de trabajo de los mismos, se pueden generar una fuerza de empuje vertical y tres pares de fuerza respecto de cada uno de los ejes principales de inercia del vehículo. De esta forma se pueden controlar directamente cada uno de los grados de libertad mencionados. En base a ensayos para los conjuntos motor/hélice que se utilizan (ver Szafranski et al. (2014), Jakubowski et al. (2015)), se puede mostrar que un modelo que aproxima adecuadamente el comportamiento del motor i establece que la fuerza ejercida por el mismo será $f_i = \kappa \omega_i^2$ donde ω_i es la velocidad de giro del motor correspondiente. La constante $\kappa > 0$ depende del motor y las hélices elegidas y su valor se obtiene a partir de ensayos.

De la geometría del vehículo puede verse que la relación entre las fuerzas que ejercen los motores, y el empuje total más los pares de fuerza respecto de cada uno de los ejes principales, estará dada por la siguiente expresión:

$$\begin{bmatrix} f_z \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & -L\frac{\sqrt{3}}{2} & -L\frac{\sqrt{3}}{2} & 0 & L\frac{\sqrt{3}}{2} & L\frac{\sqrt{3}}{2} \\ L & \frac{L}{2} & -\frac{L}{2} & -L & -\frac{L}{2} & \frac{L}{2} \\ -k_T & k_T & -k_T & k_T & -k_T & k_T \end{bmatrix}}_M \cdot \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \end{bmatrix} \quad (1)$$

donde f_z es el empuje resultante en el eje z mientras τ_x, τ_y, τ_z son los pares de fuerza en cada uno de los ejes correspondientes. La fuerza realizada por cada uno de los motores es f_i , con $i = 1, \dots, 6$. La distancia desde cada rotor al centro del vehículo es $L > 0$ y $k_T > 0$ es una constante, que depende del motor y las hélices elegidas, la cual también se obtiene en base a ensayos.

De acuerdo a lo anterior, los algoritmos pueden comandar acciones de control τ_x, τ_y, τ_z y f_z para cada uno de los grados de libertad controlados del vehículo es decir *roll*, *pitch*, *yaw* y altura. De acuerdo con Li y Li (2011), la respuesta del vehículo al aplicar estas acciones de control estará dada por :

$$\ddot{\phi} = \frac{I_y - I_z}{I_x} \dot{\theta} \dot{\psi} + \frac{\tau_x}{I_x} \quad (2)$$

$$\ddot{\theta} = \frac{I_z - I_x}{I_x} \dot{\phi} \dot{\psi} + \frac{\tau_y}{I_y} \quad (3)$$

$$\ddot{\psi} = \frac{I_x - I_y}{I_z} \dot{\phi} \dot{\theta} + \frac{\tau_z}{I_z} \quad (4)$$

$$\ddot{h} = g - \frac{f_z}{m} \cos(\phi) \cos(\theta) \quad (5)$$

donde I_x, I_y, I_z son los momentos de inercia en cada uno de los ejes, m su masa, h es la altura del vehículo y g es la gravedad.

Este modelo dinámico se obtiene a partir de ciertas simplificaciones. Se toma el tensor de inercia del vehículo como una matriz diagonal, se desprecia el momento de inercia de los rotores, y se supone que los ángulos ϕ y θ son suficientemente pequeños como para aproximar las velocidades angulares en cada uno de los ejes por $\dot{\phi}, \dot{\theta}$ y $\dot{\psi}$. Además, se desprecian los modos de flexión en la estructura y las hélices. Otros fenómenos que no son tenidos en cuenta en este modelo, están dados por la dinámica pasabajos de los rotores y por los ruidos de alta frecuencia presentes en los sensores inerciales y el magnetómetro en base a los cuales se determinan los ángulos ϕ, θ y ψ . El filtrado de estos sensores es realizado en la computadora de bajo nivel y es admisible no tenerlo en cuenta en este modelo simplificado. Como se verá más adelante, la incertidumbre introducida por estas dinámicas no modeladas será tenida en cuenta en el lazo de control de la velocidad de desplazamiento.

Por otro lado, despreciando los términos del efecto giroscópico y suponiendo que los ángulos del vehículo no se apartan significativamente de la posición de equilibrio $\theta = \phi = 0$ (situación de *hovering*), se obtienen los cuatro modelos lineales correspondientes simplificados y desacoplados:

$$\begin{aligned} \ddot{\phi} = \frac{\tau_x}{I_x} &\Rightarrow \phi(s) = \frac{1}{I_x s^2} \tau_x(s) \\ \ddot{\theta} = \frac{\tau_y}{I_y} &\Rightarrow \theta(s) = \frac{1}{I_y s^2} \tau_y(s) \\ \ddot{\psi} = \frac{\tau_z}{I_z} &\Rightarrow \psi(s) = \frac{1}{I_z s^2} \tau_z(s) \\ \ddot{h} = g - \frac{f_z}{m} &\Rightarrow h(s) = \frac{1}{m s^2} (g m - f_z(s)) \\ &\Rightarrow h(s) = \frac{1}{m s^2} \tilde{f}_z(s) \end{aligned} \quad (6)$$

Este es un modelo útil para un primer diseño del control de estabilidad en orientación y altura, pero con incertidumbre proveniente de las simplificaciones respecto del modelo completo. Para estabilizar el vehículo en *hovering*, se implementa un control PID para cada uno de los cuatro grados de libertad mencionados. Como se ve arriba, las transferencias simplificadas de cada eje y de la altura de vuelo del vehículo quedan dadas por un doble integrador $\frac{\alpha}{s^2}$ donde α puede ser igual a $\frac{1}{I_x}, \frac{1}{I_y}, \frac{1}{I_z}$ o $\frac{1}{m}$. Para cualquiera de estos casos al cerrar el lazo con un controlador PID se obtiene la siguiente función transferencia nominal:

$$T(s) = \frac{\alpha(k_d s^2 + k_p s + k_i)}{s^3 + \alpha(k_d s^2 + k_p s + k_i)} \quad (7)$$

De la ecuación (7), será útil denotar a las funciones de transferencia $T_\phi(s)$ y $T_\theta(s)$ como las relaciones correspondientes entre los ángulos comandados ϕ_c y θ_c con los ángulos realizados ϕ y θ . Las constantes k_p, k_i, k_d son las ganancias proporcional, integral y derivativa del PID, y para cada eje (*roll*, *pitch*, *yaw* o *altura*) tendrán los valores que resulten de llevar a cabo una sintonización adecuada. La salida de los controles PID de cada eje, define los pares de fuerza τ_x, τ_y, τ_z y la fuerza f_z necesarias. La incertidumbre presente en estos modelos hace que los valores calculados para las ganancias de los PID's que se utilizan para cerrar los cuatro lazos, sirvan inicialmente. Luego, en base a pruebas de vuelo, se suele llevar a cabo una sintonización adicional de dichas ganancias a fin de obtener una respuesta dinámica adecuada.

Basados en el modelo (6) y mediante la ecuación (1), se procede a obtener las señales que se deben comandar a cada uno de los motores para lograr el empuje y los pares de fuerza necesarios. En este caso, mediante la pseudo-inversa de Moore-Penrose de la matriz M , puede obtenerse la solución de menor energía que logra el empuje y los pares de fuerza deseados mediante:

$$\begin{bmatrix} f_1 & f_2 & \dots & f_6 \end{bmatrix}^T = M^\dagger \begin{bmatrix} f_z & \tau_x & \tau_y & \tau_z \end{bmatrix}^T$$

Siguiendo a Li y Li (2011), la posición del vehículo con respecto a una terna local con el plano xy fijo al suelo estará dada por:

$$\begin{aligned} \ddot{x} &= -(\cos(\phi) \cos(\psi) \sin(\theta) + \sin(\phi) \sin(\psi)) \frac{f_z}{m}, \\ \ddot{y} &= -(\cos(\phi) \sin(\psi) \sin(\theta) - \cos(\psi) \sin(\phi)) \frac{f_z}{m}. \end{aligned} \quad (8)$$

Al igual que para las dinámicas de orientación y altura, estas ecuaciones representan un modelo que relaciona los ángulos y el empuje con las aceleraciones de manera simplificada. Se desprecian efectos aerodinámicos suponiendo que el vehículo estará en una situación cercana al *hovering*. Considerando que el vehículo es controlado en 4 grados de libertad, se desprende de la ecuación (8) que para conseguir desplazamientos en el plano local xy deberán comandarse cambios en la orientación del vehículo. El modelo bajo las hipótesis de que ϕ y θ son pequeños con f_z constante, permite simplificar más aún obteniéndose una dinámica lineal. Además se puede asumir sin pérdida de generalidad que $\psi = 0$. No suponer esto último redundaría en un cálculo más elaborado de las inclinaciones necesarias que se deberían imprimir al vehículo para controlar su velocidad de desplazamiento y se evita por simplicidad. De esta forma con $\psi = 0$, resulta la siguiente descripción lineal:

$$\ddot{x} = -\frac{f_z}{m}\theta \quad \ddot{y} = \frac{f_z}{m}\phi \quad (9)$$

Si llamamos v_x y v_y a las respectivas velocidades y si además incluimos las funciones de transferencia T_θ y T_ϕ , tendremos como modelo nominal una relación de ángulos comandados a velocidades realizadas dada por las siguientes funciones de transferencia:

$$V_x(s) = \underbrace{\left(-\frac{f_z}{ms}T_\theta(s)\right)}_{H_x(s)}\theta_c(s) \quad V_y(s) = \underbrace{\left(\frac{f_z}{ms}T_\phi(s)\right)}_{H_y(s)}\phi_c(s) \quad (10)$$

En la ecuación anterior, $V_x(s)$ y $V_y(s)$ denotan las transformadas de Laplace de las velocidades v_x y v_y . De la misma forma $\theta_c(s)$ y $\phi_c(s)$ son los ángulos de “pitch” y “roll” comandados como acciones de control de velocidad. Debe notarse que nominalmente estos modelos tienen partes estables dadas por $T_\theta(s)$ y $T_\phi(s)$ más un polo en el origen cada uno.

Los modelos (7) y (10) son habituales y las simplificaciones son necesarias para una estrategia de control simple que pueda implementarse en un procesador con una capacidad moderada. Sin embargo, es posible diseñar en una capa superior un control de velocidad para la dinámica de desplazamiento en el plano xy local, que contemple estas diferencias entre el modelo de la ecuación (10) y el sistema real del multi-rotor. Esto se logra a partir de ensayos experimentales para la identificación del mismo como se verá en la sección 6.

3. Sensores de navegación

Para poder controlar el multi-rotor, la computadora de vuelo debe ejecutar varias tareas. Primero, para estabilizar el vehículo en *hovering*, los datos de los tres acelerómetros y giróscopos (Unidad de Medida Inercial - IMU) son adquiridos y procesados para obtener una estimación filtrada de los ángulos ϕ y θ . Los acelerómetros están afectados por ruido de alta frecuencia, pero en una situación cercana al *hovering*, entregan una buena medición de la dirección del vector de gravedad. Por otro lado, los giróscopos tienen una buena respuesta en alta frecuencia, pero suelen estar afectados por sesgos en sus mediciones. A su vez, un magnetómetro puede utilizarse para obtener el ángulo de yaw ψ . Para extraer las mejores características de cada uno

de los sensores, se utiliza un filtro que fusione la información de estos sensores estimando la orientación del UAV (θ, ϕ, ψ).

Para mantener fija la posición del vehículo es necesario estimar la posición o velocidad. Integrando numéricamente la información de la IMU podría estimarse la velocidad del vehículo y su posición (España (2019)). Sin embargo, las unidades de medidas inerciales de tecnología MEMS que son las que se utilizan frecuentemente en pequeños UAV, no son lo suficientemente precisas para esta tarea. Los sesgos, derivas térmicas y otros factores que estos sensores presentan, hacen que el error en la estimación de velocidad y posición se degrade en el tiempo muy rápidamente y de manera no acotada. Es por esto que es necesario utilizar información adicional para resolver el problema de control de posición del vehículo.

Para poder utilizar el vehículo en ambientes cerrados, el posicionamiento utilizando GPS u otro sistema satelital de posicionamiento global, no es una alternativa viable. Si bien existen soluciones para posicionamiento en lugares cerrados que utilizan Wi-Fi, Bluetooth o ultrasonido, estos sensores requieren la instalación de *beacons* en distintos puntos del entorno en donde se estará moviendo el vehículo. Algo similar ocurre con sistemas de captura de movimiento basados en cámaras, como Optitrack o VICON (Gomes et al. (2016)).

Dado que se pretende que el vehículo pueda ser capaz de posicionarse en cualquier entorno, sin una infraestructura instalada a priori, debe utilizarse algún sensor a bordo del vehículo capaz de adquirir información relativa al ambiente en el que está navegando. Una alternativa adecuada podría ser dotar al vehículo con un LIDAR, sensor láser que permite calcular con alta precisión la distancia relativa del vehículo a los objetos alrededor de sí mismo. Este posicionamiento relativo podría utilizarse para estabilizar el UAV en posición. Sin embargo, los LIDAR son sensores costosos y pesados que tienen un consumo considerable, por lo cual no suelen estar presentes salvo en aplicaciones que requieran muy alta precisión. Como alternativa más económica, podría considerarse el posicionamiento relativo utilizando sensores de ultrasonido. Sin embargo, su precisión, falta de direccionalidad y la poca robustez de sus medidas, hacen que estos sensores no sean muy utilizados en UAV. Sensores de infrarrojo, como los TeraRanger Evo resultan una alternativa adecuada para posicionar el UAV en ambientes cerrados, pero son considerablemente más costosos que otros sensores.

Utilizar una cámara a bordo del vehículo parece una buena opción, debido a su bajo costo, consumo, tamaño y peso. Sin embargo, hay que considerar que la navegación por visión requiere una capacidad de cómputo considerable. Existen diversas alternativas, técnicas y sensores, para resolver la navegación por imágenes. La elección del algoritmo para procesar la información de la cámara, dependerá, en parte, de la capacidad de cómputo disponible. Es para una tarea como ésta que la arquitectura de la computadora de vuelo basada en un procesador de alto nivel y uno de bajo nivel resulta útil.

La fusión de información provista por una IMU junto con información de cámaras, ha demostrado ser de utilidad en diversas aplicaciones de navegación y control de vehículos, en particular en UAV como se ve en Armesto et al. (2007), Zul Azfar y Hazry (2011), Fossen et al. (2017), Kanellakis y Nikolakopoulos (2017), Chao et al. (2014) y en Giribet et al. (2018).

En el caso de este trabajo, la Intel Edison es la computadora encargada de resolver el procesamiento de la cámara, la fusión de la información de visión con la información de la IMU y el control de la velocidad de desplazamiento del UAV. Más aún, esta computadora también se encarga de realizar algoritmos de guiado del vehículo, así como la monitorización del mismo. Ejecutar todas estas tareas de manera simultánea impone restricciones sobre los algoritmos que pueden ejecutarse y el código debe ser optimizado para la arquitectura elegida, siempre que sea posible. El flujo óptico es una de las formas más simples de realizar navegación por visión, e implementado de manera eficiente, requiere una capacidad de cómputo moderada.

3.1. Flujo óptico

La imagen vista o capturada por un sensor de imagen, es el resultado de la proyección de una porción del espacio sobre un plano, que es el propio sensor de imagen. Cualquier movimiento que describa un objeto dentro de la zona de visión del sensor tendrá su correlato como un movimiento en dos dimensiones de una región de la imagen. Dicha región es la proyección del objeto sobre el plano del sensor. Si es posible detectar el desplazamiento que una región presenta entre cuadros consecutivos de una secuencia de imágenes, entonces se podrá definir una magnitud que represente la velocidad del movimiento de dicha región. Esta velocidad caracteriza entonces al movimiento que se detecta en el plano del sensor. En caso de que se detecte un movimiento también es posible determinar cuál es su dirección.

Este campo de velocidades cuyo dominio es toda la imagen, es lo que se denomina “flujo óptico” de la imagen o también “campo de velocidades” de la imagen. El flujo óptico constituye entonces un *estimador del movimiento 2D* presente en la imagen.

Existen diversos enfoques para la estimación del flujo óptico, en particular en Horn y Schunck (1981), Tai et al. (2007), se toman en consideración dos restricciones sobre la señal de luminancia de la imagen, que permiten calcular el campo de velocidades en cada píxel de la imagen. Dado un instante de tiempo $t \in \mathbb{R}_{\geq 0}$, sea $\mathbf{x}(t) = (x(t), y(t)) \in \mathbb{R}^2$ un punto de la imagen respecto del sensor (la cámara). Para $t \geq 0$, se denota la intensidad de la imagen correspondiente al punto $\mathbf{x}(t)$ como $I(\mathbf{x}(t), t) = I(x(t), y(t), t)$. La primera hipótesis que se considerará es que la intensidad permanece constante a lo largo de la trayectoria que describe $\mathbf{x}(t)$, lo cual se expresa en la ecuación (11), donde $\langle \cdot, \cdot \rangle$ representa el producto interno canónico en \mathbb{R}^2 , $\mathbf{v} = (\frac{dx}{dt}, \frac{dy}{dt})$ e $I_t(\mathbf{x}, t)$ es la derivada parcial de $I(\mathbf{x}, t)$ respecto del tiempo t . De esta forma resulta

$$\langle \nabla I(\mathbf{x}, t), \mathbf{v} \rangle + I_t(\mathbf{x}, t) = 0 \quad , \quad (11)$$

$$\nabla^2 u + \nabla^2 v = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} = 0 \quad . \quad (12)$$

La condición de la ecuación (12), impone condiciones de regularidad de los movimientos de la superficie de los objetos respecto del sensor. El vector $\mathbf{v}(t) = (u(t), v(t))$ representa la velocidad del punto de la imagen $\mathbf{x} = (x, y)$ en el instante t . El campo de velocidades sobre todos los puntos de la imagen conforma el *flujo óptico* (OF). Para poder calcular el flujo óptico a partir de la información obtenida en las imágenes, es necesario

estimar las derivadas parciales I_x, I_y, I_t y los laplacianos $\nabla^2 u, \nabla^2 v$.

En Horn y Schunck (1981) y en Tai et al. (2007), se propone un algoritmo recursivo para calcular los valores de (u, v) para lo cual se define la cantidad de iteraciones $k = 0, 1, 2, \dots$ utilizadas en el refinamiento del cálculo. El parámetro k se configura de acuerdo a las condiciones del entorno, por ejemplo la luminosidad. En condiciones de baja luminosidad es necesario una mayor cantidad de iteraciones del algoritmo, pero esto tiene un impacto significativo en el tiempo de cómputo. En Luiso y Giribet (2017) se propone una implementación eficiente de este algoritmo, para lo cual se hace uso de las capacidades de procesamiento SIMD (Single Instruction Multiple Data) disponibles en un procesador Edison de Intel, que es el utilizado en este trabajo para resolver las tareas de alto nivel. Para analizar el tiempo de cómputo, en este mismo trabajo se analizó el porcentaje de ocupación del procesador para distintos valores de k , que permitieron obtener un procesamiento de 25 fps (cuadros de 160x120 px) utilizando un 7,5 % del procesador para un número de iteraciones $k = 1$, un 32,5 % para $k = 3$, y un 57,5 % del uso del procesador para $k = 5$.

A partir de la información de flujo óptico es posible obtener la velocidad del vehículo respecto a la terna local. Sin embargo, dado que el flujo óptico es el movimiento relativo entre un objeto y la cámara, además del desplazamiento del vehículo, el flujo óptico se ve afectado por los movimientos de rotación (ver la Figura 3). Dado que el vehículo cuenta con una IMU, es posible compensar el flujo óptico para reducir la componente debida a la rotación y así obtener la componente correspondiente al desplazamiento.

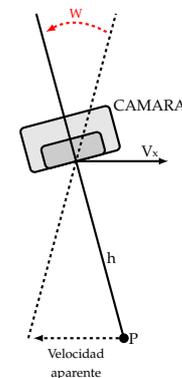


Figura 3: Movimiento aparente de un punto observado por la cámara.

Considerando las componentes de traslación (T_{OF}) y rotación (R_{OF}), el flujo óptico estará dado por:

$$\begin{bmatrix} \overline{OF}_x \\ \overline{OF}_y \end{bmatrix} = T_{OF} + R_{OF} \quad (13)$$

siendo \overline{OF}_x y \overline{OF}_y el promedio del flujo óptico en los ejes “x” e “y” respectivamente, calculados como:

$$\overline{OF}_k = \frac{\sum_{i=1}^{ph} \sum_{j=1}^{pw} v p_{ij}^k}{ph \cdot pw} \quad \text{con } k = \{x, y\} \quad (14)$$

donde p_h es la altura del vehículo (convertida en píxeles de la imagen), p_w el ancho de la imagen en píxeles y $v p_{ij}^k$ la velocidad del píxel ij en la componente $k = x$ o $k = y$. Esta es una aproximación válida si se considera que el vehículo mantiene su altura y ángulo de yaw constantes.

Para rotaciones bajas (menores que 60° aproximadamente) se puede modelar el flujo rotacional como:

$$R_{OFi} = kr_j \cdot w_j \cdot h \quad \text{con} \quad i, j = \{x, y\} \text{ y } j \neq i \quad (15)$$

donde h será la altura del UAV, w_i su velocidad angular en el eje $i = \{x, y\}$, y kr_i una constante de proporcionalidad entre la velocidad de los píxeles en la imagen y la velocidad angular (rotación) del vehículo, abarcando características propias de la cámara como ser la distancia focal y el ángulo de apertura de la misma. Por otro lado, para la componente trasversal del flujo, se tiene:

$$T_{OFi} = kt_i \cdot v_i \quad \text{con} \quad i = \{x, y\} \quad (16)$$

siendo v_i la velocidad de traslación del vehículo en el eje i y kt_i una constante de proporcionalidad entre la velocidad de los píxeles en la imagen y la velocidad de traslación del vehículo. De esta forma:

$$\begin{bmatrix} \overline{OF}_x \\ \overline{OF}_y \end{bmatrix} = \begin{bmatrix} 0 & kr_y \\ kr_x & 0 \end{bmatrix} \cdot \begin{bmatrix} w_x \\ w_y \end{bmatrix} \cdot h + \begin{bmatrix} kt_x & 0 \\ 0 & kt_y \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \end{bmatrix} \quad (17)$$

Las constantes kr_i y kt_i pueden ser determinadas mediante ensayos experimentales, e incluso pueden ser estimadas en vuelo (Tournour (2018)).

De esta forma la velocidad de traslación del UAV estimada por el algoritmo de flujo óptico estará dada por $v_{OF} = (v_x, v_y)$, obtenida a partir de la ecuación (17). Esta información puede ser fusionada con la de la IMU en un Filtro de Kalman Extendido para estimar la velocidad del vehículo con respecto al plano local.

4. Arquitectura del sistema de NGC

Cuando se implementa el sistema de control de un multirotor hay dos problemas que deben ser resueltos. El primero es la estabilización de la orientación y la altura del vehículo. El segundo problema es mantener fija la posición del vehículo respecto del plano (local) xy .

Como se mencionó en la sección 2, el sistema es subactuado ya que para desplazarse en el plano xy , es necesario modificar la orientación del vehículo en $pitch$ o $roll$. Si bien como problema de control el diseño podría abordarse de manera integral, este suele separarse en el diseño del control de orientación y altura por un lado, y el de la velocidad de desplazamiento en xy por otro, debido a que los requerimientos de cómputo de los algoritmos son distintos. Los controles de orientación y altura deben ejecutarse a una tasa de refresco mayor que el control de la velocidad de desplazamiento porque su dinámica es más rápida. Por otro lado, para la estimación de la velocidad de desplazamiento del vehículo se necesita una capacidad de cómputo mayor.

4.1. Lazo interno para la estabilización del UAV

Como se mencionó en la sección 2, para estabilizar el vehículo en *hovering*, se utilizan tres controladores PID independientes para cada uno de sus ángulos. Para poder cerrar el lazo de control de altura y calcular la fuerza f_z necesaria para mantener el vehículo en una altura fija, es necesario contar con información provista por un barómetro o un sensor de ultrasonido, ya que son económicos y de peso y tamaño reducidos.

Los lazos de control de orientación, así como el control de altura se ejecutan en el procesador de bajo nivel de la computadora de vuelo. Con este lazo es posible contar con un vehículo capaz de mantenerse estabilizado y a una altura fija. Sin embargo, cualquier perturbación puede provocar un desplazamiento en el plano local xy .

4.2. Control de desplazamiento del UAV

Un lazo de control adicional se encarga de mantener fija la posición del vehículo en el plano xy . Es decir que el lazo de control busca que la velocidad del vehículo en el plano xy sea nula. Para poder estimar la velocidad del vehículo respecto del plano, resulta fundamental el algoritmo que calcula el flujo óptico por lo cual el lazo de control de desplazamiento se implementa en la computadora de alto nivel.

La arquitectura del sistema de control se conforma mediante un control de desplazamiento de *alto nivel* que envía como comandos, referencias de ángulo y altura a la computadora de *bajo nivel* para regular la velocidad. El esquema propuesto puede verse en la Figura 4.

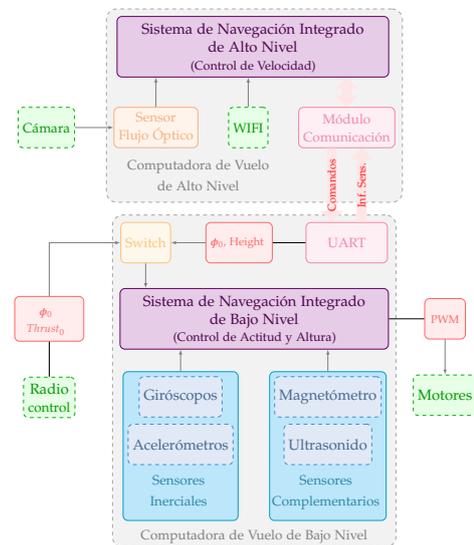


Figura 4: Diagrama en bloques de sistema de navegación propuesto.

En esta computadora de alto nivel se ejecutan concurrentemente diversas tareas, no sólo el cálculo de flujo óptico y el algoritmo de control, sino además el manejo de las comunicaciones, la monitorización del estado de la misión y los comandos enviados por el control remoto entre otras propias del sistema operativo. Todas estas tareas inciden en la dinámica del vehículo y resulta muy complejo establecer un modelo que contemple todas estas interacciones. Ya de por sí, la tarea de la estimación

de flujo óptico depende de la cantidad de iteraciones necesarias para estimar correctamente la velocidad del vehículo, ante determinadas condiciones del entorno.

Para establecer un modelo dinámico que relacione los comandos de *pitch* y *roll* con la velocidad de desplazamiento (v_x, v_y) del vehículo puede recurrirse a técnicas de *identificación* que permitan establecer un modelo dinámico con ciertas dispersiones, que corresponderán a diversos factores, principalmente las demoras provocadas en el hardware de alto nivel por las diversas tareas que se ejecutan y a la discrepancia entre los modelos teóricos y la realidad, que están relacionadas con la linealización del modelo y con las simplificaciones que llevan a desacoplar las dinámicas de los lazos de *pitch* y *roll*. Será necesario entonces que el controlador sea robusto ante las incertidumbres de modelo.

Para lograr identificar el sistema es necesario realizar experimentos, por lo cual resulta conveniente primero especificar las características principales del vehículo y el hardware utilizado.

5. Especificaciones del UAV y hardware del autopiloto



Figura 5: Computadora de navegación y control. Izquierda: Placa de alto nivel (Intel Edison). Derecha: Placa de bajo nivel (LPC-1769 y sensores de navegación).

La computadora de vuelo está dividida en dos subsistemas (ver la Figura 5). Por una lado, una computadora de bajo nivel, que se encarga de resolver la adquisición y sincronización de las mediciones de los sensores, mientras que al mismo tiempo resuelve un algoritmo de control de bajo nivel, con el cual se pueden garantizar tiempos estrictos de ejecución. En las pruebas que se presentan en este trabajo se utilizó como computadora de bajo nivel una placa de desarrollo propio. Por otro lado, el sistema se conecta con una computadora de alto nivel. El propósito de la computadora de alto nivel es ejecutar algoritmos que requieran una carga computacional más elevada. Esto permite dividir el algoritmo de control en un proceso que se ejecuta en la computadora de bajo nivel con garantías en los tiempos de ejecución, mientras que en la computadora de alto nivel se ejecuta un lazo de control basado en un algoritmo más elaborado, entre otras tareas.

Para este trabajo la computadora de bajo nivel cuenta con un microcontrolador LPC-1769 tipo ARM Cortex M3 con un reloj de 120MHz. La placa a su vez contiene los sensores de navegación necesarios, una IMU MPU-6000, un magnetómetro

HMC5883L y un barómetro BMP180. En esta computadora se ejecuta el lazo de control de orientación a 200Hz y se calculan las señales para comandar los actuadores del vehículo. Por su lado, la computadora de alto nivel es una Intel Edison que funciona con un sistema operativo Linux.

El multi-rotor utilizado para las pruebas se basa en una estructura DJI-F550, con una distancia máxima entre motores de 550 mm, una masa $m = 2,5\text{kg}$ y momentos de inercia estimados $I_x = I_y = 0,00915\text{ kg}\cdot\text{m}^2$ y $I_z = 0,01186\text{ kg}\cdot\text{m}^2$. Los actuadores instalados en esta estructura son motores T-Motor 2212-920KV, con hélices de plástico 9545, accionadas por controladores de velocidad electrónicos (ESC) de 20A, con una estimada $k_T = 1,576 \cdot 10^{-2}$. La batería utilizada es una 4S 5000mAh 20C LiPo que permite aproximadamente 15 minutos de vuelo. A la computadora de alto nivel se le conecta una webcam Logitech C920 a través de un puerto USB. La computadora permite conexiones mediante Wi-Fi y ZigBee, por lo cual en todo momento se puede monitorizar el estado de vuelo. Además, la información de interés para evaluar el desempeño del control es guardada en una tarjeta de memoria microSD. Detalles de las controladoras de vuelo de diseño propio se reportan en Garberoglio et al. (2019).

6. Identificación y diseño del control de desplazamiento

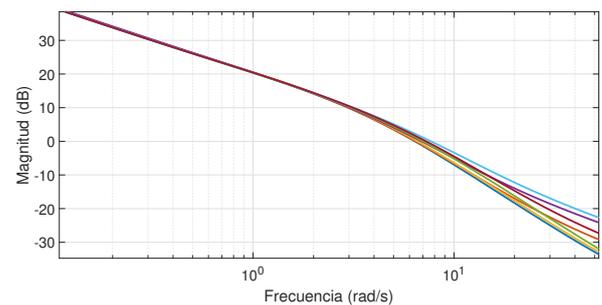


Figura 6: Respuesta en frecuencia para el eje de *roll* para los distintos ensayos realizados.

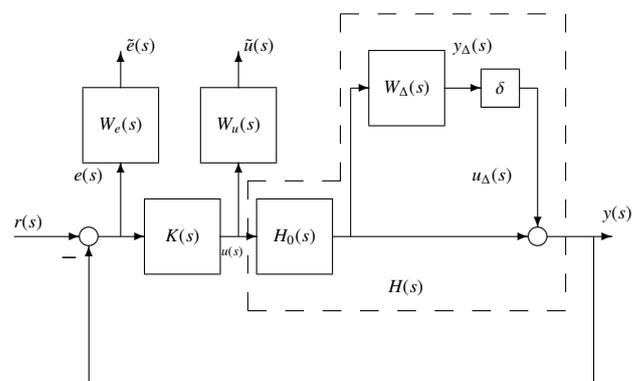


Figura 7: Diagrama del lazo cerrado para cada una de las transferencias $H_x(s) = \frac{v_x(s)}{\theta(s)}$ y $H_y(s) = \frac{v_y(s)}{\phi(s)}$.

En la computadora de alto nivel, se estima la velocidad del UAV y se calculan los comandos de *pitch* y *roll* necesarios para

regular la velocidad del vehículo. Las variaciones en la componente de velocidad v_x estará ligada a variaciones en el ángulo de *pitch*, mientras que la componente v_y estará asociada a variaciones en *roll*. Es por este motivo que se desea determinar las transferencias

$$H_x(s) = \frac{V_x(s)}{\theta_c(s)} \quad \text{y} \quad H_y(s) = \frac{V_y(s)}{\phi_c(s)}. \quad (18)$$

$V_x(s)$, $V_y(s)$, $\phi_c(s)$, $\theta_c(s)$ son señales en el dominio de Laplace que fueron definidas en la sección 2.

Con una estrategia similar a la propuesta en Noormohammedi Asl et al. (2020), para estimar $H_x(s)$ y $H_y(s)$ se procede de la siguiente manera. Desde la computadora de alto nivel, se envían a la computadora de bajo nivel comandos *pitch* y *roll* y se estiman las velocidades resultantes del vehículo v_x y v_y . Para obtener las respuestas, las referencias son de tipo *chirp*. Con estos pares entrada-salida fue posible estimar funciones de transferencia que caractericen la dinámica utilizando herramientas de identificación tradicionales (Ljung (1999)). Cabe aclarar que esta función de transferencia tiene en cuenta el comportamiento dado por la dinámica del vehículo junto con los lazos cerrados de control de *pitch* y *roll* correspondientes al nivel inferior. Además, quedan incluidos en este proceso de identificación los retardos adicionales que tienen lugar debido al procesamiento de alto nivel.

Con el propósito de poder garantizar la robustez ante distintas condiciones del ambiente, se decidió variar las iteraciones en el cálculo del flujo óptico modificando así la carga computacional. Se realizaron $M = 7$ ensayos y se estimaron las funciones de transferencia correspondientes. La respuesta en frecuencia de magnitud de cada una de estas transferencias ajustadas mediante la identificación puede observarse en la Figura 6. Se presenta solo el eje correspondiente a las respuestas en *roll* habiendo resultado respuestas similares para el eje de *pitch*.

La dispersión en las respuestas en frecuencia ajustadas que se observa entre los distintos ensayos puede ser utilizada como base para un diseño robusto en \mathcal{H}_∞ . A tal efecto, resulta útil el esquema de la Figura 7. Los experimentos de identificación realizados permiten tener una idea del comportamiento del sistema $H(s)$ encerrado en línea punteada. En rigor, $H(s)$ constituye una familia de plantas con incertidumbre multiplicativa de salida parametrizada por el bloque “ δ ” (Sánchez Peña y Sznaier (1998)).

La salida del lazo $y(s)$ corresponderá a la velocidad $V_x(s)$ o $V_y(s)$ del UAV, mientras que la entrada de referencia $r(s)$ corresponderá a la velocidad deseada (que será $r(s) = 0$ para que el vehículo no se desplace). La señal de control comandada $u(s)$ corresponde a los ángulos de *pitch* $\theta_c(s)$ o *roll* $\phi_c(s)$ que se enviarán a la placa de bajo nivel como comandos a ser realizados. $W_e(s)$ y $W_u(s)$ son funciones de peso útiles para especificar el desempeño deseado del sistema de control mientras que $W_\Delta(s)$ es utilizada a los efectos de caracterizar la incertidumbre dinámica en el modelo identificado lo cual permite dar garantías en relación a la robustez del control.

A través de la norma \mathcal{H}_∞ se pueden resolver problemas de diseño de controles donde se consiga cumplir con ciertos objetivos de robustez y rendimiento. En pos de obtener un diseño que consiga un balance adecuado entre el rendimiento en la respuesta, con una acción de control acorde a requerimientos además

de robustez, se suele buscar como objetivo la minimización de la norma \mathcal{H}_∞ de la siguiente transferencia que mezcla dos objetivos:

$$\left\| \left\| \begin{array}{c} W_e(s)S(s) \\ W_u(s)K(s)S(s) \end{array} \right\| \right\|_\infty < 1 \quad (19)$$

siendo

$$S(s) = [1 + L(s)]^{-1} \quad \text{con} \quad L(s) = H(s)K(s)$$

y $K(s)$ el controlador. Por un lado el término $W_e(s)S(s)$ se relaciona con el error de seguimiento mientras que el término $W_u(s)K(s)S(s)$ se incluye en pos de conseguir una acción de control moderada. En lo relacionado a los pesos se puede decir que además de conseguir error de seguimiento nulo en estado estacionario a comandos de velocidad constantes, el peso $W_e(s)$ se usará para lograr un ancho de banda deseado a lazo cerrado. Por su parte, el peso $W_u(s)$ permite limitar la acción de control (Sánchez Peña y Sznaier (1998)).

En cuanto al peso $W_\Delta(s)$, el mismo se utiliza para describir la incertidumbre para modelo. El control $K(s)$ debe ser capaz de satisfacer las especificaciones no solo del modelo nominal $H_0(s)$ sino para cualquier miembro de la familia de plantas parametrizada por δ de la forma

$$H(s) = H_0(s) + \underbrace{\delta W_\Delta(s)H_0(s)}_{\text{incerteza}}, \quad (20)$$

con $|\delta| \leq 1$. Si $\delta = 0$ significaría que tenemos la certeza de que el sistema real se comporta exactamente como el modelo $H_0(s)$, lo cual es una suposición muy fuerte. Más realista es suponer que $\delta \neq 0$ (con incerteza acotada $|\delta| \leq 1$), y que la función de peso $W_\Delta(s)$ describe cómo se distribuye el desconocimiento del modelo real respecto al nominal en el espectro de frecuencias.

A partir de la descripción de la familia de plantas con incertidumbre multiplicativa dada por (20), puede verse que para cualquier $|\delta| \leq 1$, cualquiera de las $H(s)$ de la familia de plantas debe ser tal que para toda ω

$$\left| \frac{H(j\omega) - H_0(j\omega)}{H_0(j\omega)} \right| \leq |W_\Delta(j\omega)|. \quad (21)$$

Para asegurar la estabilidad robusta del sistema, es decir garantizar que el controlador $K(s)$ estabiliza cualquiera de las posibles $H(s)$ descriptas con la familia de modelos, se debe satisfacer $\|W_\Delta(s)T_0(s)\|_\infty < 1$, donde

$$T_0(s) = 1 - S_0(s) \quad \text{con} \quad S_0(s) = [1 + H_0(s)K(s)]^{-1}$$

es la sensibilidad complementaria nominal del lazo (Sánchez Peña y Sznaier (1998)).

Para obtener un controlador de forma tal de satisfacer los requerimientos de rendimiento robusto (19) que involucran a todas las posibles plantas de la familia (20), se recurre al criterio de sensibilidades mixtas nominales. Es decir que se buscará un controlador $K(s)$ que minimice la norma \mathcal{H}_∞ de la siguiente

transferencia:

$$\begin{Bmatrix} W_e S_o \\ W_u K S_o \\ W_{\Delta} T_o \end{Bmatrix}_{\infty}$$

Esto puede resolverse utilizando herramientas de control robusto tradicionales (Sánchez Peña y Sznaiar (1998)).

6.1. Identificación del sistema y determinación de los pesos de rendimiento y robustez

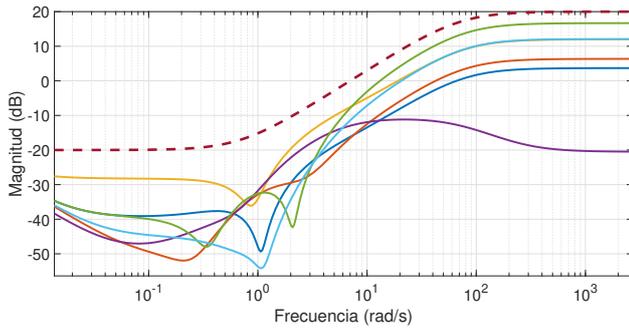


Figura 8: Distribución en frecuencia de la incerteza dinámica. En trazo punteado se puede observar el peso W_{Δ} que cubre la incerteza.

Analicemos por simplicidad el eje “x”, puesto que para el otro eje puede procederse de forma análoga. De los ensayos realizados cuyos resultados se reflejan en la Figura 6, es posible establecer un modelo nominal y cubrir la dispersión obtenida en los experimentos con la incerteza del modelo. Más precisamente, se toma el eje “x” y se define el modelo nominal como:

$$H_{x_0}(s) = \frac{0,6239(s + 76,48)(s + 1,044)}{(s + 3,788)(s + 1,343)(s + 0,0001966)} \quad (22)$$

para luego compararlo con la diferencia relativa de cada uno de las transferencias obtenidas en los experimentos, respecto al modelo nominal. Esto es, si $H_{x_i}(s)$ es la transferencia correspondiente al experimento $i = 1, 2, \dots$, graficamos $\left| \frac{H_{x_i}(j\omega) - H_{x_0}(j\omega)}{H_{x_0}(j\omega)} \right|$, como muestra la Figura 8. En esta misma Figura puede observarse la respuesta de la transferencia:

$$W_{\Delta_x}(s) = \frac{10s + 7}{s + 70},$$

que satisface la condición dada en la ecuación (21).

En lo relativo al peso W_e , siguiendo un procedimiento similar al utilizado en Noormohammadi Asl et al. (2020) se fijó el mismo como

$$W_e(s) = \frac{s^2 + 1,8s + 9}{(s + 1000)(s + 1,8)(s + 0,001)}. \quad (23)$$

En cuanto al peso W_u , se lo fijó como una constante $W_u(s) = 0,05$ a fin de limitar la acción de control en todo el espectro de frecuencias. Esto se puede hacer de esta forma en tanto la robustez del diseño y la limitación en el ancho de banda se logran mayormente mediante los pesos W_e y W_{Δ} . Debe notarse que el W_e consigue el efecto de que el controlador resultante tenga acción cuasi-integral.

7. Resultados Experimentales

Con el fin de evaluar experimentalmente el algoritmo de control de la velocidad de desplazamiento utilizando información de flujo óptico, se implementó el control descrito anteriormente en la computadora de vuelo abordo del vehículo. En Giribet y Luiso (2020) se ve el vídeo de un vuelo en donde puede apreciarse el comportamiento del vehículo cuando se cierra el lazo de control de la velocidad de desplazamiento.

Para validar los resultados se utilizó el sistema de posicionamiento Marvelmind, el cual permite obtener información con una precisión de $\pm 2cm$. Esta información no es utilizada en el algoritmo de control, sino solamente para tener una referencia del error en el control de velocidad de desplazamiento implementado. Debe notarse además que la computadora de bajo nivel controla la altura del vehículo en una referencia fija, la cual es establecida desde la computadora de alto nivel.

Si bien se realizaron varios ensayos de validación, aquí se presenta uno en particular para mostrar el funcionamiento. Se fijó la referencia de altura en $1,5m$ respecto del punto de despegue del vehículo. A su vez, la computadora de bajo nivel controla el ángulo de *yaw* utilizando información de un magnetómetro, con el objetivo que el vehículo mantenga un ángulo fijo. Además, controla los ángulos de *pitch* y *roll* para que sigan referencias dadas. Las referencias de *pitch* y *roll* pueden estar dadas por la computadora de alto nivel (cuando se cierra el lazo de control de velocidad de desplazamiento) o por los movimientos ejecutados por el piloto, quien comanda al vehículo mediante un radio-control. De esta forma, el piloto puede activar o desactivar el control de velocidad de desplazamiento por flujo óptico durante el vuelo. Cuando este control se activa, el piloto ya no tiene control del vehículo y éste funciona de manera autónoma, hasta el momento en el que se desactive el control por flujo óptico, para lo cual se utiliza un interruptor en el radio-control.

En la primera prueba, el piloto guía al vehículo hasta un punto determinado y deja de enviar comandos de *pitch* y *roll*, por lo cual se envían referencias de ángulo (*pitch* y *roll*) cero al control que ejecuta la computadora de bajo nivel. En las Figuras 9 y 10 se indica con líneas a trazos los instantes en los cuales el piloto ejecuta maniobras de *pitch* o *roll* utilizando el radio-control. Con una línea roja a trazos se indica cuándo el piloto envía referencias de ángulos *pitch* y *roll* iguales a cero y con una línea negra cuándo vuelve a enviar comandos. Puede apreciarse que cuando el piloto deja de enviar comandos de *pitch* o *roll*, el vehículo se desplaza, llegando a derivar hasta $2m$ en el eje *y*, y $1m$ en el eje *x*, respecto de su posición inicial. Este desplazamiento se debe a varios factores, principalmente perturbaciones que actúan sobre el vehículo, como podría ser cualquier corriente de aire.

En las mismas Figuras puede verse con una línea a trazos y puntos roja, que a los 35 segundos (aproximadamente) se activa el control utilizando el flujo óptico. Puede apreciarse que el vehículo mantiene la posición dentro de un rango $< \pm 15cm$, hasta que el control de flujo óptico se desactiva a los 130 segundos. Esto está indicado en la Figura con una línea a trazos y puntos negra.

En el siguiente ensayo se perturbó al vehículo durante el vuelo, con el objetivo evaluar la respuesta del control y analizar su desempeño. Se introdujeron perturbaciones tanto en el eje *x* – *pitch* como en el eje *y* – *roll*. El tipo de perturbaciones

aplicadas al vehículo puede apreciarse en el vídeo que se ve en Giribet y Luiso (2020). Estando el vehículo estabilizado a 1,5 metros de altura respecto del suelo, las perturbaciones consisten de pequeños empujes moderados aplicados con la mano sobre el vehículo a fin de acelerarlo en el plano xy requiriendo al control de velocidad de desplazamiento que lleve la velocidad a cero.

En las Figuras 12 y 11 se encuentran las magnitudes de velocidad y posición del vehículo a lo largo del vuelo. Se observan

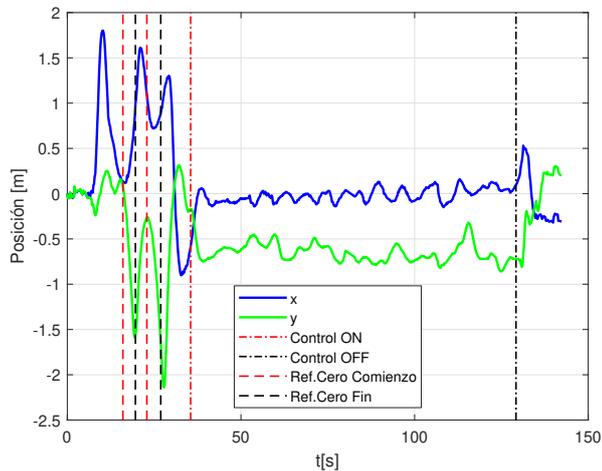


Figura 9: Vuelo con y sin el control de velocidad a cero.

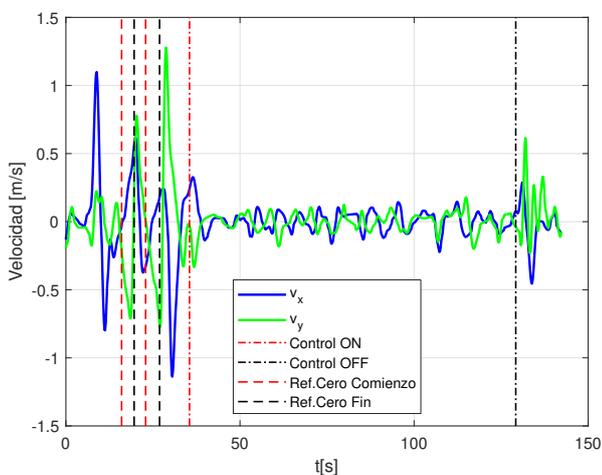


Figura 10: Vuelo con y sin control de velocidad a cero.

8. Conclusiones

En este trabajo se mostró que una estrategia de control en \mathcal{H}_∞ permite controlar el desplazamiento de un multi-rotor utilizando información de flujo óptico. Este diseño permite implementar un control en presencia de incertidumbre en el modelo, en particular la incertidumbre en el modelo provocada por los retardos debidos al software que se ejecuta en la computadora de alto nivel. Resulta difícil en general compensar estos retardos porque se deben a diversos factores que pueden cambiar durante el vuelo, por lo cual se recurrió a una técnica de identificación que permite estimar la incertidumbre del modelo y diseñar un algoritmo que sea capaz de controlar el vehículo ante el peor caso.

las perturbaciones introducidas en los segundos 28 y 45 para *roll*, y en los segundos 61 y 78 para *pitch*.

En la Figura 11 se ve claramente que luego de la perturbación el vehículo se desplaza 1,5m aproximadamente a una nueva posición y se mantiene en torno a esta posición hasta la siguiente perturbación. Esto se debe a que el lazo no controla la posición sino que busca mantener en cero la velocidad. De hecho, el vehículo no utiliza información absoluta de posición para el control.

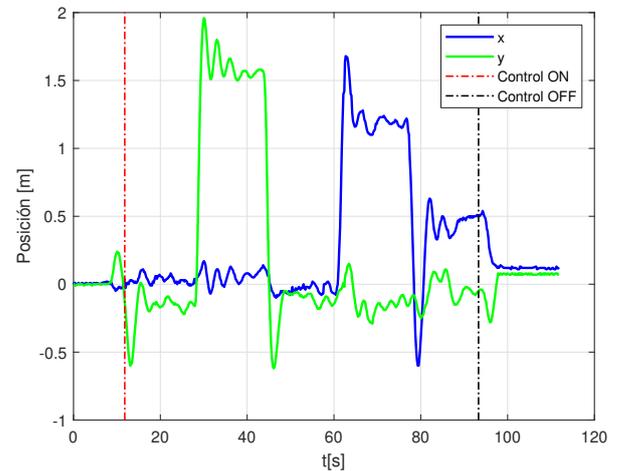


Figura 11: Vuelo con perturbaciones.

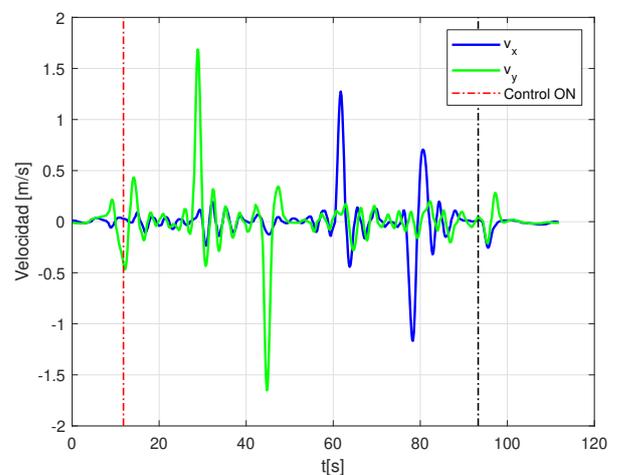


Figura 12: Vuelo con perturbaciones.

Podría argumentarse que es posible utilizar un procesador destinado exclusivamente al cálculo del flujo óptico, o aumentar la capacidad de cómputo del procesador de alto nivel, pero esto incrementa el costo, consumo, tamaño y peso del sistema. Además, debe resaltarse como resultado de este trabajo es que fue posible aprovechar la arquitectura y capacidad de cómputo existente utilizando una técnica de control adecuada.

Otra pregunta que surge es cuánto más complejo resulta el algoritmo de control robusto propuesto desde el punto de vista computacional en comparación a una técnica tradicional como el control proporcional-integral-derivativo (PID). Debe decirse que la carga computacional generada por el control propuesto

resulta comparable a la de un PID, medido desde el porcentaje de ocupación del procesador.

Una limitación de la técnica empleada es que el diseño de la ley de control se basa en resultados experimentales realizados en una etapa de identificación y diseño por lo cual no se aprovecha toda la información de los vuelos que ocurren posteriormente. Sería interesante adoptar una técnica que vaya aprendiendo la dinámica del sistema con los sucesivos vuelos. Ciertas técnicas basadas en control adaptativo buscan lograr esto, y en los últimos años se han propuesto diversas técnicas basadas en aprendizaje de máquina con el mismo propósito. Sin embargo, la complejidad computacional para implementar este tipo de técnicas es considerablemente mayor.

En este trabajo se utilizó la información de flujo óptico solo para obtener información de desplazamiento del vehículo, sin embargo el flujo óptico puede ser utilizado para obtener información de altura e incluso de rotación. Esto no alteraría los resultados presentados aquí, aunque sí permitiría mejorar la estimación de *yaw* en situaciones en donde el vehículo navegue en entornos que pudieran afectar la medición del magnetómetro. Esto puede ser un aspecto a mejorar en el futuro. También sería interesante para realizar en el futuro un análisis más detallado de los retardos de tiempo en el procesador, para ver si es posible incluir esta información en el diseño del control.

Agradecimientos

Los autores agradecen la tarea de los revisores del trabajo. Sus observaciones y correcciones han contribuido a la introducción de significativas mejoras en el mismo.

Este trabajo ha sido realizado parcialmente gracias al apoyo de la Universidad de Buenos Aires a través del proyecto UBA-PDE2019 y de la Universidad Tecnológica Nacional a través del proyecto CCUT-7731TC.

Referencias

- Armesto, L., Tornero, J., Vincze, M., 06 2007. Fast ego-motion estimation with multi-rate fusion of inertial and vision. *I. J. Robotic Res.* 26, 577–589. DOI: 10.1177/0278364907079283
- Azkarate, M., Gerdes, L., Joudrier, L., J. Pérez-del Pulgar, C., 2020. A GNC architecture for planetary rovers with autonomous navigation. En: *International Conference on Robotics and Automation (ICRA)*. IEEE, pp. 1–6.
- Bithas, P. S., Michailidis, E. T., Nomikos, N., Vouyioukas, D., Kanatas, A. G., 2019. A survey on machine-learning techniques for UAV-based communications. *Sensors* 19 (23). URL: <https://www.mdpi.com/1424-8220/19/23/5170> DOI: 10.3390/s19235170
- Chao, H., Gu, Y., Napolitano, M., Jan 2014. A survey of optical flow techniques for robotics navigation applications. *Journal of Intelligent & Robotic Systems* 73 (1), 361–372.
- Choi, S. Y., Cha, D., 03 2019. Unmanned aerial vehicles using machine learning for autonomous flight; state-of-the-art. *Advanced Robotics*, 1–13. DOI: 10.1080/01691864.2019.1586760
- España, M. D., 2019. *Sistemas de Navegación Integrada con Aplicaciones*, 2do Edición. CONAE. URL: https://www.argentina.gob.ar/sites/default/files/me_nav_integ_libro_2019.pdf
- Fonnegra, R., Goez, G., Tobón, A., 2019. Estimación de orientación de un vehículo aéreo no modelado usando fusión de sensores inerciales y aprendizaje de máquina. *Revista Iberoamericana de Automática e Informática Industrial* 16(4), 415–422.
- Fossen, T., Pettersen, K., Nijmeijer, H., 2017. *Sensing and Control for Autonomous Vehicles*. Springer.
- Garberoglio, L., Pose, C., Mas, I., Giribet, J., 2019. Diseño de un autopiloto para pequeños vehículos no tripulados. *Elektron* 3 (1).
- Giribet, J., Mas, I., Moreno, P., 2018. Vision-based integrated navigation system and optimal allocation in formation flying. En: *Proceedings of International Conference on Unmanned Aerial Aircrafts*, Dallas, USA. pp. 52–61.
- Giribet, J. I., Luiso, J., 2020. Vuelo experimental - control de flujo óptico - LAR-GPSIC. URL: <https://youtu.be/9YuDyu21pvA>
- Gomes, L. L., Leal, L., Oliveira, T. R., Cunha, J. P. V. S., Revoredo, T. C., Aug 2016. Unmanned quadcopter control using a motion capture system. *IEEE Latin America Transactions* 14 (8), 3606–3613.
- Grabe, V., Bühlhoff, H. H., Giordano, P. R., 2012. On-board velocity estimation and closed-loop control of a quadrotor UAV based on optical flow. En: *2012 IEEE International Conference on Robotics and Automation*. pp. 491–497.
- Honegger, D., Meier, L., Tanskanen, P., Pollefeys, M., 2013. An open source and open hardware embedded metric optical flow CMOS camera for indoor and outdoor applications. En: *2013 IEEE International Conference on Robotics and Automation*. pp. 1736–1741.
- Horn, B. K. P., Schunck, B. G., 1981. Determining optical flow. *Artificial Intelligence* 17, 185–203.
- Jakubowski, A., Kubacki, A., Minorowicz, B., Nowak, A., 2015. Analysis thrust for different kind of propellers. En: *Advances in Intelligent Systems and Computing*. Springer International Publishing, pp. 85–90.
- Kanellakis, C., Nikolakopoulos, G., Jul 2017. Survey on computer vision for UAVs: Current developments and trends. *Journal of Intelligent & Robotic Systems* 87 (1), 141–168.
- Li, J., Li, Y., 2011. Dynamic analysis and pid control for a quadrotor. En: *2011 IEEE International Conference on Mechatronics and Automation*. pp. 573–578.
- Lim, H., Lee, H., Kim, H. J., 2012. Onboard flight control of a micro quadrotor using single strapdown optical flow sensor. En: *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. pp. 495–500.
- Ljung, L., 1999. *System Identification: Theory for the User*. Prentice Hall information and system sciences series. Prentice Hall PTR.
- Luiso, J. E., Giribet, J. I., 2017. Sensor de flujo óptico. En: *Actas de la Reunión de Procesamiento de la Información y Control*. pp. 1–6.
- Ma, Z., Li, H., Gu, Y., Li, Z., Li, Q., 2019. Flight and hover control system design for a mini-quadrotor based on multi-sensors. *International Journal of Control, Automation and Systems* 17, 486–499.
- Madridano, A., Campos, S., Al-Kaff, A., García, F., Martín, D., Escalera, A., 2020. Vehículo aéreo no tripulado para vigilancia y monitorización de incendios. *Revista Iberoamericana de Automática e Informática industrial* 17 (3), 254–263. DOI: 10.4995/riai.2020.11806
- Noormohammadi Asl, A., Esrafilian, O., Arzati, M. A., Taghirad, H. D., 2020. System identification and \mathcal{H}_∞ -based control of quadrotor attitude. *Mechanical Systems and Signal Processing* 135.
- Sánchez Peña, R. S., Szaiaier, M., 1998. *Robust Systems Theory and Applications*. John Wiley & Sons, Inc.
- Szafranski, G., Czyba, R., Błachuta, M., 2014. Modeling and identification of electric propulsion system for multirotor unmanned aerial vehicle design. En: *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. pp. 470–476.
- Tai, X.-C., Lie, K.-A., Chan, T. F., Osher, S. (Eds.), 2007. *Image Processing Based on Partial Differential Equations*. Springer-Verlag Berlin Heidelberg, proceedings of the International Conference on PDE-Based Image Processing and Related Inverse Problems, CMA, Oslo, August 8-12, 2005. DOI: 10.1007/978-3-540-33267-1
- Tournour, A., 2018. Control de un vehículo aéreo no tripulado utilizando información de flujo óptico. Tesis de Ingeniería Electrónica. Facultad de Ingeniería, Universidad de Buenos Aires. URL: <http://bibliotecadigital.fi.uba.ar/items/show/18231>
- Zhang, X., Xian, B., Zhao, B., Zhang, Y., 2015. Autonomous flight control of a nano quadrotor helicopter in a gps-denied environment using on-board vision. *IEEE Transaction on Industrial Electronics* 62, 10.
- Zul Azfar, A., Hazry, D., 2011. A simple approach on implementing imu sensor fusion in pid controller for stabilizing quadrotor flight control. En: *IEEE 7th Int. Colloquium on Signal Processing and its Applications*. pp. 28–32.