



FACULTAD DE INGENIERÍA
UNIVERSIDAD DE BUENOS AIRES

TESIS DE GRADO DE INGENIERÍA ELECTRÓNICA

**Diseño e implementación de un sistema
embebido de control de actitud para
aeronaves no tripuladas**

Alan Kharsansky

Director: Dr. Ing. Ariel Lutenberg

31 de mayo de 2013

Prefacio

En este informe se presenta el trabajo de investigación y desarrollo llevado a cabo para lograr el diseño y la implementación de una computadora¹ de vuelo para aeronaves no tripuladas. La idea surge del estudio de dos ramas de la ingeniería electrónica: la teoría de control y los sistemas embebidos.

La teoría de control trata el comportamiento de sistemas dinámicos y en particular busca controlar diferentes variables de estos. Es inherentemente un campo interdisciplinario que abarca áreas como la electrónica, la física y la mecánica. Su campo de aplicación tradicional, surge del ambiente industrial con el control de temperaturas, niveles y caudales, aunque en la actualidad son cada vez más las aplicaciones de la teoría de control en industrias como la automotriz, la aeroespacial y la del transporte.

Los sistemas embebidos, son equipos electrónicos que incluyen un procesamiento de datos pero que, a diferencia de una computadora personal, están diseñados para satisfacer una función específica, como en el caso de un reproductor de música portátil, un teléfono celular, un router o la computadora de abordo de un automóvil o un satélite. Su diseño está optimizado para reducir su tamaño y su costo, aumentar su confiabilidad y mejorar su desempeño.

Estas dos áreas pueden combinarse, logrando así los denominados “sistemas de control embebidos”. En este trabajo, se utilizará como pretexto a la aeronave no tripulada “quadrotor”, con el fin de estudiar todos los problemas que involucran el desarrollo de un sistema de estas características. Se comenzará con la investigación del estado del arte de los quadrotores y las áreas que involucran. Luego se desarrollará un modelo matemático del quadrotor que permitirá estudiarlo como sistema dinámico. A continuación se hará un estudio detallado de uno de los controladores más utilizados hoy en dia, el PID, y se presentaran los aspectos fundamentales para adaptar estos a sistemas digitales discretos. Una vez obtenido esto, se tendrá una primera estimación de los requerimientos que deberá tener la computadora de vuelo y se detallaran los aspectos claves del diseño de la misma. Finalmente se comprobaran las estrategias de control elegidas y se validarán los modelos matemáticos mediante pruebas de campo con un quadrotor en vuelo bajo diferentes condiciones.

En síntesis, en este trabajo se pretende demostrar que es posible resolver un problema aparentemente complejo, como es el vuelo de una aeronave, dividiéndolo en problemas más sencillos y luego aplicando fundamentos básicos de la teoría de control a la implementación de un sistema de control embebido.

¹Aquí el término computadora se hará extenso a todo el equipo y no solo el microprocesador o microcontrolador y se lo utilizará como sinónimo de “sistema embebido”.

Índice general

Índice general	3
1. Introducción	7
1.1. Principio de funcionamiento del Quadrotor	9
1.1.1. Funcionamiento básico	9
1.1.2. Formas de vuelo y sistemas de referencia	10
1.1.3. Dinámica del quadrotor	12
1.2. Navegación, control y guiado	13
1.2.1. Navegación	13
1.2.2. Control	15
1.2.3. Guiado	16
1.3. Estado del arte	17
1.3.1. Quadrotores recreativos	17
1.3.2. Quadrotores profesionales	18
1.3.3. Quadrotores de investigación	19
1.4. Resumen del capítulo	20
2. Modelado de la planta	21
2.1. Dinámica del quadrotor	22
2.2. Fuerzas y momentos sobre el centro de masa: \bar{M}_1	23
2.3. Actuadores	25
2.3.1. Empuje y torque producido por una hélice	28
2.3.2. Banco de pruebas de actuadores	28
2.3.3. Velocidad angular	30
2.3.4. Empuje estático	31
2.3.5. Torque estático	32
2.3.6. Constantes de las hélices	32
2.3.7. Identificación paramétrica	33
2.3.8. Linealización	35
2.3.9. Respuesta transitoria del actuador	36
2.3.10. Modelo completo del actuador	38
2.4. Matriz del quadrotor: \bar{M}_Q	39
2.5. Matriz de control: M_{MIX}	39

2.6. Parámetros constantes de las ecuaciones	41
2.7. Perturbaciones y efectos de segundo orden	43
2.7.1. Blade flapping	43
2.7.2. Efectos giroscópicos	44
2.7.3. Efectos aerodinámicos	44
2.7.4. Efecto suelo	45
2.8. Separación por ejes	45
2.9. Resumen del capítulo	46
3. Controlador PID	47
3.1. Arquitecturas de implementación de controladores PID	48
3.2. Mejoras en el algoritmo	49
3.2.1. Setpoint ponderado	49
3.2.2. Derivación de la variable de proceso	50
3.2.3. Ganancia limitada	51
3.2.4. Saturación de la acción integral (Windup)	52
3.2.5. Transición Manual/Automático	53
3.3. Implementación digital	53
3.4. Modelo matemático del controlador	55
3.5. Validación del controlador PID digital	55
3.5.1. Planta de prueba	55
3.5.2. Respuesta al escalón	56
3.5.3. Transición manual automático	56
3.5.4. Anti Windup	57
3.6. Resumen del capítulo	58
4. Plataforma de validación	59
4.1. Simulador	60
4.2. Estructura mecánica y actuadores	62
4.3. Diseño e implementación de la computadora de vuelo	62
4.3.1. Selección de componentes	63
4.3.2. Arquitectura	65
4.3.3. Firmware	65
4.3.4. Arquitectura de Comunicaciones	68
4.4. Segmento de tierra	69
4.5. Resumen del capítulo	69
5. Implementación	71
5.1. Implementación digital del controlador	72
5.2. Diseño en cascada	72
5.3. Diseño del SAS	73
5.3.1. Análisis de la planta	73
5.3.2. Ajuste del controlador	74
5.3.3. Validación	77
5.4. Diseño del CAS	78
5.4.1. Análisis de la planta	78
5.4.2. Ajuste del controlador	79
5.4.3. Validación	81
5.5. Modos de vuelo	81

5.6.	Limitaciones del modelo	83
5.6.1.	Dependencias del modelo	83
5.6.2.	Dependencia del sistema de navegación	85
5.7.	Resumen del capítulo	88
6.	Resultados	89
6.1.	Modos de vuelo	90
6.2.	Vuelo en modo “acrobático”	92
6.3.	Vuelo en modo “estabilizado”	94
6.4.	Vuelo en modo “actitud”	96
6.5.	Hovering con perturbaciones en modo “estabilizado”	98
6.6.	Computadora de vuelo	100
6.6.1.	Consumo de recursos	100
6.6.2.	Costo estimado de fabricación	101
6.7.	Resumen del capítulo	102
7.	Conclusiones y trabajo futuro	103
7.1.	Conclusiones	103
7.2.	Trabajo futuro	104
7.2.1.	Navegación	104
7.2.2.	Modelado	105
7.2.3.	Control	105
7.2.4.	Guiado	105
7.2.5.	Computadora de vuelo	106
7.2.6.	Mejoras directas de este trabajo	106
Bibliografía		107
A. Diagramas esquemáticos de la Computadora de Vuelo		109

Nomenclatura

ϕ, θ, ψ	Ángulo de rotación sobre los ejes X, Y, Z respectivamente. Equivalente a roll, pitch, yaw.
p, q, r	Velocidad angular sobre los ejes X, Y, Z respectivamente.
x_b, y_b, z_b	Posición de la aeronave respecto de la terna fija en tierra.
$v_{x_b}, v_{y_b}, v_{z_b}$	Velocidad de desplazamiento de la aeronave respecto de la terna fija en tierra.
F_{aero}	Fuerzas sobre el rígido producto de efectos aerodinámicos.
M_{aero}	Momentos sobre el rígido producto de efectos aerodinámicos.
F_{act}	Fuerzas producidas sobre el centro de masa de la aeronave por los actuadores del sistema.
M_{act}	Momentos producidos sobre el centro de masa de la aeronave por los actuadores del sistema.
M_i	Momento generado por el actuador i .
T_i	Empuje generado por el actuador i .
y_i	Señal de control de actuador i .
Z_c	Acción de control colectiva de los actuadores.
ϕ_c	Acción de control para producir torque neto en el eje X de la aeronave.
θ_c	Acción de control para producir torque neto en el eje Y de la aeronave.
ψ_c	Acción de control para producir torque neto en el eje Z de la aeronave.
SAS	Stability agumentation system.
CAS	Control augmentation system.

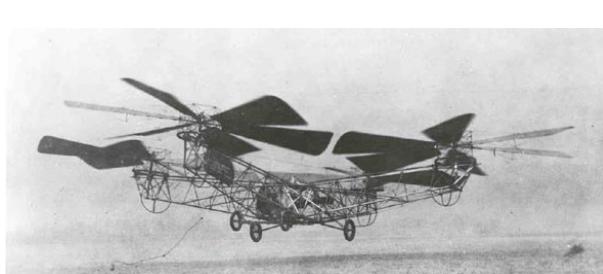
CAPÍTULO 1

Introducción

Las aeronaves no tripuladas, UAVs por sus siglas en inglés, se utilizan para tareas de investigación, aplicaciones comerciales, aplicaciones civiles de búsqueda y rescate, así también como para seguridad. En general estas aeronaves suelen agruparse en aquellas capaces de realizar una misión sin intervención humana, llamadas autónomas, y aquellas controladas de manera remota por un operador humano.

Dentro de los UAVs más populares, se encuentra la aeronave de cuatro rotos denominada quadrotor por su nombre en inglés que significa "de cuatro rotos". En los comienzos de la aviación, este tipo de aeronave fue de las primeras en aparecer, inclusive previamente al helicóptero convencional, pero sus problemas mecánicos y su dificultad de control por parte del piloto la hicieron inviable para la época. En la actualidad, gracias a los avances de la electrónica, tanto en capacidad de procesamiento como en miniaturización, resulta posible hacer pequeñas aeronaves de este tipo, no tripuladas, capaces de volar y realizar tareas de utilidad. Es posible observar en la figura 1.1 una comparación entre el primer quadrotor diseñado por el hombre y el que se utilizó para este trabajo.

Hoy en día es posible encontrar este tipo de aeronaves en aplicaciones tales como la agri-mensura y la agricultura, y que permiten sobrevolar campos de manera autónoma y sin riesgo para un humano; en seguridad e ingeniería civil, equipado con cámaras de monitoreo remoto



(a) Quadrotor tripulado de De Bothezat - 1923



(b) Quadrotor autónomo - 2013

Figura 1.1: Unos de los primeros quadrotores tripulados vs. un quadrotor autónomo moderno

así también como en la industria cinematográfica para la realización de tomas aéreas que eran imposibles de realizar con helicópteros convencionales una década atrás.

Este tipo de aeronaves se han popularizado en los últimos años dentro del ámbito académico debido a su bajo costo, su relativa facilidad de construcción y su reducida cantidad de partes móviles, lo que permitió que laboratorios de diferentes universidades del mundo [1, 2] lo utilicen como plataforma de estudio para diferentes áreas de investigación entre las que se destacan:

- Modelado de sistemas físicos
- Teoría de control
- Sistemas Embebidos
- Procesamiento de señales
- Procesamiento de imágenes y video

Estas diferentes disciplinas se suelen encontrar también dentro de una rama de la ingeniería conocida como "Navegación, control y guiado", que estudia y diseña sistemas para controlar los movimientos de automóviles, barcos, aviones, etc. El significado de cada uno de estos términos es el siguiente:

- **Navegación:** determinación en un determinado momento de la posición y actitud¹ del vehículo, así también como sus derivadas.
- **Control:** manipulación de fuerzas de control sobre el vehículo para lograr mantener la estabilidad de la misma y cumplir con los requerimientos de guiado.
- **Guiado:** determinación y control del camino o recorrido que el vehículo debe realizar.

Si bien en muchos casos, estas funciones pueden ser realizadas por humanos, existen sistemas donde las velocidades de reacción necesarias están por encima de las capacidades humanas. Es por eso que se recurre a sistemas electrónicos de alta velocidad, mayormente computadoras o microcontroladores, para su implementación.

Este trabajo se enfoca principalmente en el área de control y en el diseño e implementación de una computadora capaz de estabilizar y controlar el vuelo del quadrotor para luego poder usarlo junto a técnicas de navegación y guiado en aplicaciones de más alto nivel. Se analizará el quadrotor como sistema físico y se buscará desarrollar un enfoque simple que permita abordar el problema del diseño de control desde una perspectiva clásica. En particular se busca diseñar un sistema de control en tiempo real que pueda mantener a un quadrotor volando paralelamente al piso a una altura considerable del mismo, sin intervención externa, lo cual se conoce como hovering. Aunque en algunas aplicaciones se denomina hovering a la acción de quedarse sobrevolando un mismo lugar, en este caso, no se hará referencia a la posición sino solamente a la actitud; es decir que el quadrotor se encontrará en hovering cuando el plano que conforman las hélices se encuentre paralelo al plano de la superficie terrestre.

En las siguientes secciones se presentarán los conceptos fundamentales que forman la base de estudio de este trabajo: se comenzará primero con los fundamentos del quadrotor como aeronave y luego se presenta una introducción a los conceptos de navegación, control y guiado.

¹Se entiende por actitud a la orientación de un objeto con respecto a un marco de referencia inercial u otro objeto

1.1. Principio de funcionamiento del Quadrotor

1.1.1. Funcionamiento básico

Un quadrotor como el que se observa en la figura 1.1b es un sistema simple, que puede resumirse con el diagrama en bloques de la figura 1.2. en donde:

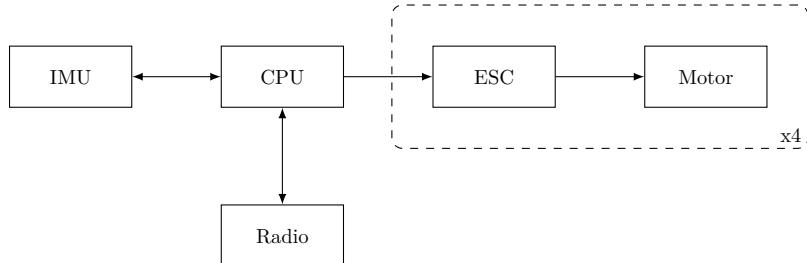


Figura 1.2: Diagrama de bloques mínimo de un quadrotor

- **IMU:**² unidad determinación de la actitud, posición y velocidad de la aeronave utilizando sensores como ser giróscopos, acelerómetros, magnetómetros, etc.
- **ESC:** del inglés ”Controlador electrónico de velocidad”, recibe comandos y apartir de ellos permite variar la velocidad de los motores.
- **Motor:** motores sin escobillas trifásicos que entregan energía a las hélices para generar fuerzas y momentos en los 4 extremos del quadrotor.
- **CPU:** es la encargada de ejecutar los algoritmos de control que, a partir de los datos de la IMU y del recorrido o la actitud configurada, comandan los ESC para efectuar fuerzas de control sobre la aeronave.
- **Radio:** interfaz de comunicación con tierra que, si bien no es imprescindible, suele ser utilizada para la descarga de telemetría así también como para enviar información de navegación y guiado.

Durante el vuelo la computadora recibe el estado actual de la aeronave en cuanto a posición, velocidad y/o actitud de la IMU y otros sensores como ser: barómetros, sensores ultrasónicos de distancia, gps, etc. y efectúa acciones correctivas utilizando los actuadores. Los actuadores, compuestos por motores eléctricos y hélices generan tanto empuje en la dirección axial, como así también momento ³ en sentido contrario a la rotación de las mismas. La plataforma de validación diseñada para este trabajo incorpora en una misma placa electrónica la CPU, la IMU y la radio, lo cual provee un entorno de desarrollo apropiado.

Es posible combinar las fuerzas y momentos generados por los actuadores de manera tal de generar hasta cuatro diferentes movimientos como se observa en la figura 1.3.

Si los cuatro actuadores ejercen la misma fuerza hacia arriba (y suponemos que el sistema es perfectamente simétrico), entonces la aeronave se trasladará hacia arriba. Este control será denominado en el resto del trabajo como el control “colectivo”, concepto que proviene de la rama de la jerga de los helicópteros y determina el control que hace que la aeronave ascienda o

²Unidad de medición inercial de sus siglas en inglés

³Se utilizará la palabra *momento* como sinónimo de torque

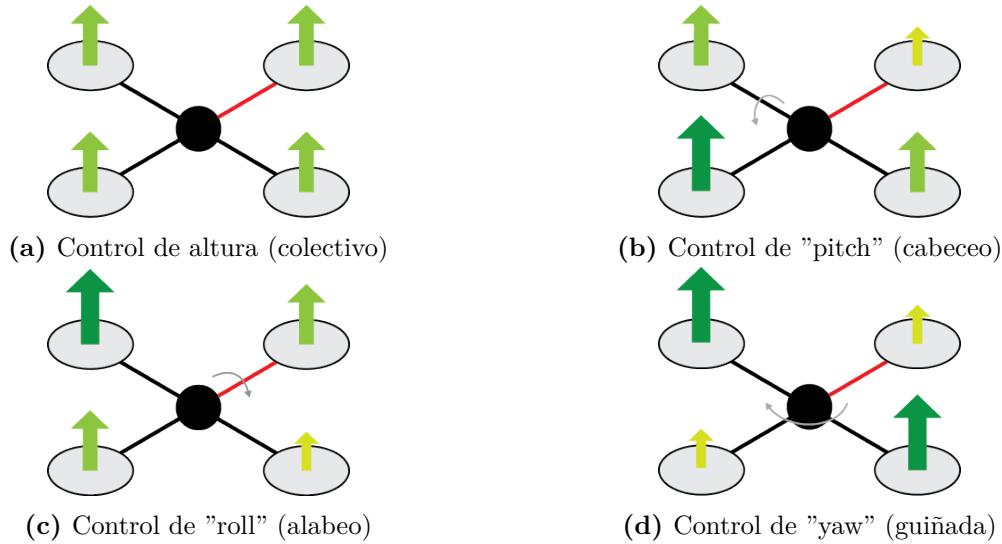


Figura 1.3: Posibles mezclas de las fuerzas de los actuadores para obtener diferentes movimientos espaciales

descienda. Si en cambio, se aplica empuje diferencial, es decir que algunos actuadores generan más fuerza que otros, entonces, se generan momentos en diferentes ejes provocando así que éste rote. Se puede inferir que dada esta arquitectura, es imposible que la aeronave se desplace para adelante, atrás o para los lados sin tener que cambiar su actitud. Esto es denominado: un sistema subactuado, ya que el mismo se encuentra en un entorno de 6 grados de libertad (DOF) pero solo puede actuar en cuatro de estos.

1.1.2. Formas de vuelo y sistemas de referencia

En general, los quadrotores pueden ser utilizados en dos formas de vuelo diferentes. Estas se conocen comúnmente como vuelo en "X" y vuelo en "+", cuya diferencia radica en el lado de la estructura que define el avance. En la figura 1.4 se observa la diferencia entre ambos conceptos. En este trabajo, dada la forma del chasis del quadrotor que se utilizó en los experimentos, se eligió el vuelo en el modo "X" y se mantuvo esta estructura durante todo el trabajo.

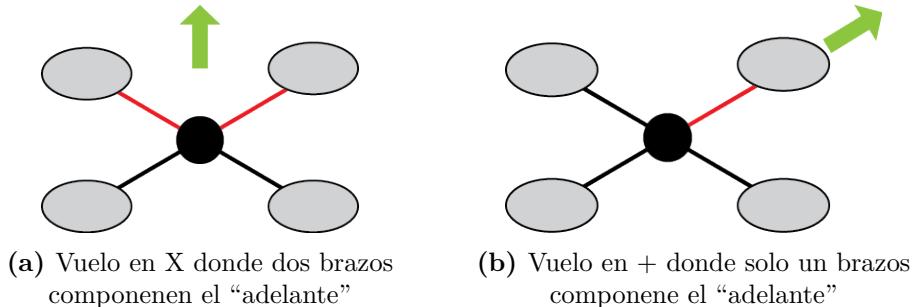


Figura 1.4: Modos de vuelo de un quadrotor. La flecha verde indica el sentido de avance hacia adelante de la aeronaves

Resulta imprescindible elegir los sistemas de referencias para analizar el problema. El primero que se utilizará es el denominado “de cuerpo”, y se encuentra fijo en la aeronave y ubicado en su centro de gravedad. Se utilizará el subíndice b para denominar coordenadas o rotaciones en este sistema de referencia por la palabra inglesa “body”. En la figura 1.5 se puede observar la orientación de la terna elegida. Notar que el origen de la terna no es la que muestra la figura sino que esta está en el centro de masa del quadrotor.

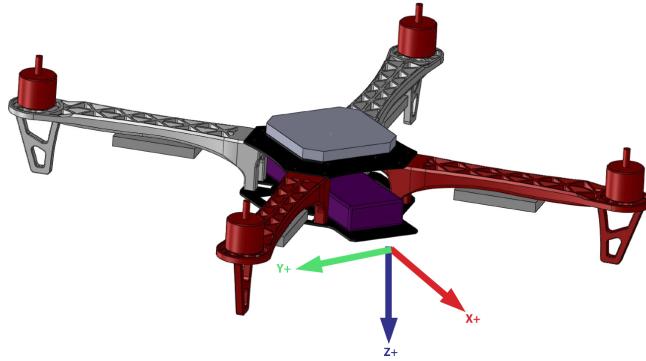


Figura 1.5: Perspectiva del modelo CAD del quadrotor.

Para expresar las rotaciones en el sistema cuerpo, se usarán los ángulos ϕ , θ y ψ que representan rotaciones en los ejes x_b , y_b y z_b respectivamente, según la regla de “la mano derecha” teniendo así, rotaciones antihorarias positivas en cada uno de los ejes. Estos ángulos suelen ser llamados en la nomenclatura aeronáutica como roll, pitch y yaw. El ángulo $\phi = 0^\circ$ y $\theta = 0^\circ$ se corresponde con la aeronave paralela al piso, y el angulo $\psi = 0^\circ$ con el norte geográfico. Las derivadas de estos ángulos, llamadas velocidades angulares, se representarán con el vector $\omega = [p, q, r] = [\dot{\phi}, \dot{\theta}, \dot{\psi}]$.

La segunda terna utilizada es una terna fija en algún punto de la tierra y será utilizada para determinar la posición del cuerpo de la aeronave. Dadas las dimensiones del problema que se estudia, se considerará un modelo de la tierra plana, infinitamente extenso y que no rota. Esta terna es útil para describir desplazamientos de la aeronave y fuerzas externas como, por ejemplo, la fuerza de gravedad o las producidas por el viento. En la figura 1.6 se pueden observar estas dos ternas.

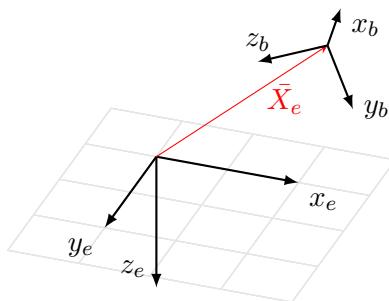


Figura 1.6: Ternas utilizadas en el sistema de referencia. b para la terna fija en la aeronave y e para la fija en la tierra.

1.1.3. Dinámica del quadrotor

Si bien la aeronave bajo estudio no es estrictamente un cuerpo rígido, resulta suficiente considerarla como tal, suponiendo que:

1. La estructura es perfectamente rígida.
2. La estructura es perfectamente simétrica.
3. No se considerán efectos giroscópicos de los motores y hélices debido a su pequeña inercia en comparación con la aeronave.

De la segunda ley de Newton se sabe que la relación entre las fuerzas y aceleraciones en un sistema de referencia inercial están dados por la ecuación:

$$\sum \bar{F}_e = m \cdot \bar{a}_e \quad (1.1)$$

donde \bar{F}_e es la suma de todas las fuerzas que actúan sobre el centro de gravedad del cuerpo, m representa la masa del rígido y \bar{a}_e la aceleración. Para el caso de un sistema de referencia no inercial, como es el sólido rígido en movimiento, se obtiene la expresión:

$$\sum \bar{F}_b = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = m(\bar{a}_b + \bar{\omega} \times \bar{v}_b) \quad (1.2)$$

que al reemplazar por sus componentes vectoriales se convierte en:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = m \left(\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} \right) \quad (1.3)$$

Expandiendo el producto vectorial se logran las primeras tres ecuaciones de la dinámica del cuerpo rígido:

$$\begin{aligned} F_x &= m(\dot{u} + qw - rv) \\ F_y &= m(\dot{v} + ru - pw) \\ F_z &= m(\dot{w} + pv - qu) \end{aligned} \quad (1.4)$$

Luego se deberán considerar las dinámicas de rotación del sistema. A partir de la segunda ley de rotación de Euler se tiene que:

$$\sum \bar{M}_e = \frac{dL_e}{dt} = \frac{d(I_e \bar{\omega})}{dt} \quad (1.5)$$

que puede ser expresado en términos de velocidad, aceleraciones angulares y momentos expresados en la terna de cuerpo como:

$$\sum M_b = \begin{bmatrix} L \\ M \\ N \end{bmatrix} = I \cdot \dot{\bar{\omega}}_b + \bar{\omega}_b \times (I \cdot \bar{\omega}_b) \quad (1.6)$$

Como se verá más adelante, si el tensor de inercia I es diagonal, se puede entonces simplificar el desarrollo de la ecuación 1.6 para obtener las 3 ecuaciones de rotación del rígido:

$$\begin{aligned} L &= I_{xx}\dot{p} + qr(I_{zz} - I_{yy}) \\ M &= I_{yy}\dot{q} + pr(I_{xx} - I_{zz}) \\ N &= I_{zz}\dot{r} + pq(I_{yy} - I_{xx}) \end{aligned} \quad (1.7)$$

Las ecuaciones 1.4 y 1.7 determinan el modelo matemático completo que describe un cuerpo rígido libre y de manera general representan la dinámica de la aeronave. Conociendo los parámetros constantes ($m, I_{xx}, I_{yy}, I_{zz}$), el estado inicial, las entradas de fuerza (F_x, F_y, F_z) y momentos (L, M, N) sobre el centro de masa del rígido, es posible estudiar su evolución a lo largo del tiempo.

Se puede observar en estas ecuaciones que los ejes están acoplados. Esto significa que un torque aplicado en el eje \hat{x} provoca un movimiento tanto en el eje \hat{x} como en los ejes \hat{y} y/o \hat{z} lo cual hace que diseñar un sistema de control sea una tarea compleja. En los siguientes capítulos se estudiará con detalle este sistema de ecuaciones para obtener modelos lineales más simples que permitan simplificar el estudio del problema y se determinará el origen y la forma de cada uno de los términos que componen estas ecuaciones.

1.2. Navegación, control y guiado

1.2.1. Navegación

Se denomina navegación al área de la ciencia que busca determinar variables como posición, actitud y velocidades tanto lineales como rotativas de objetos que se mueven dentro de un entorno mediante diferentes técnicas. Algunas permiten determinar el valor absoluto de los parámetros de navegación del vehículo mediante mediciones referidas a cuerpos exteriores al mismo, como es el caso de navegación con brújulas magnéticas, referencias visuales o inclusive mediante señales de radio conocidas en diferentes puntos del entorno (por ejemplo los sistemas VOR de la aeronáutica).

En cambio, otras técnicas utilizan mediciones de parámetros que permiten inferir los parámetros de navegación de manera indirecta, por ejemplo propagandolos. Estos sistemas, utilizados desde que el hombre comenzó a trasladarse por los mares (propagando a lo largo del tiempo la velocidad de desplazamiento), por ejemplo están limitados a las condiciones del medio y a los errores en los métodos de medición utilizados.

Con el avance de la tecnología se logró diseñar y fabricar sensores iniciales que fijos en el vehículo pueden medir sus aceleraciones y velocidades angulares de manera muy precisa y con acotado error, pero aún así, los métodos de propagación con datos que se integran en el tiempo suelen diverger. Es por eso que muchos sistemas combinan ambos tipos de mediciones, las internas y las externas para corregir durante largos períodos de tiempo los parámetros de navegación estimados. Este tipo de sistemas de navegación es ideal para aplicaciones en entornos abiertos y para vuelos autónomos, ya que por ejemplo, implementados en un quadrocoptero permiten que este conozca su entorno y parámetros de navegación, sin contacto con una estación de tierra en particular.

Otro tipo de sistemas de navegación que se utiliza para quadrocopteros en entornos cerrados son aquellos basados en visión global que consta de cámaras de video, en general dentro del espectro infrarrojo, que detectan marcas reflectivas que se suelen adherir a los quadrocopteros para poder determinar su posición, actitud y velocidad. Este método, que proviene de la industria del cine y los videojuegos para la captura de movimiento humano, puede capturar a una altísima tasa de datos estos parámetros con excelente resolución y muy bajo nivel de error. En general los

datos de este tipo de sistemas son ingresados a una computadora externa que luego mediante comandos de radio controla a los quadrotores.

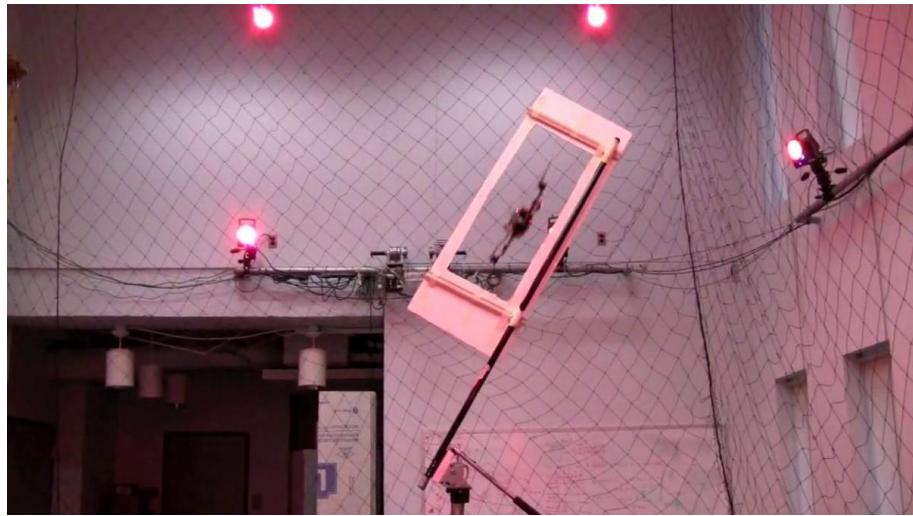


Figura 1.7: Sistema de visión para determinación de parámetros de navegación Vicon utilizado por la universidad de Pensilvannia para sus trabajos de investigación

En la figura 1.7 se puede observar un labotorio de la universidad de Pensilvania equipado con un sistema de captura de movimiento de la firma Vicon que determina la posición y velocidad de un quadrotor que atraviesa una abertura. En esta imagen también es posible ver las tres cámaras infrarrojas utilizadas por el sistema de visión. Si bien estos sistemas resultan muy efectivos para entornos cerrados, requieren de una calibración y un entorno controlado para operar, por lo que se vuelven inutilizables en exteriores o ambientes no preparados especialmente con este fin. Durante el resto de este trabajo se hará mención solamente a los sistemas de navegación y control que pueden ser utilizados tanto para entornos cerrados como abiertos y es por eso que los sistemas de visión global o de sensores externos al quadrotor no se estudiarán.

Dentro de los sistemas de navegación inercial, integrados generalmente en una unidad denominada IMU, se suelen encontrar sensores como giróscopos que miden la velocidad angular, acelerómetros que miden la fuerza específica a la que están sometidos [3], junto con magnetómetros para medir el campo magnético de la tierra y obtener así una referencia del norte magnético y barómetros para medir las diferencias de altura. En general, estos sensores pueden proveer datos útiles, pero su verdadero valor surge al utilizar técnicas de fusión de datos como ser, por ejemplo, filtros complementarios o filtros de Kalman y pueden ser además, fusionados con datos de sensores con referencias externas para aprovechar al máximo las características de cada uno de los sensores de diferentes maneras. Este tema se desprende como una de las áreas de investigación más importantes en lo relacionado con UAVs y dada su extensión y complejidad no se abordará un estudio detallado sobre este tema. Si bien resulta imprescindible contar con datos de navegación para poder realizar el control correctamente, se verá a lo largo de este trabajo que, bajo ciertas circunstancias, estos pueden ser obtenidos de circuitos integrados de muy bajo costo que proporcionan la información necesaria de navegación para cumplir con el objetivo propuesto.

1.2.2. Control

Los sistemas de control son aquellos que permiten controlar alguna cantidad o variable física en un sistema o proceso. Se los suele encontrar en aplicaciones industriales para controlar: la temperatura de hornos y calderas, el caudal o nivel de líquido en un tanque de almacenamiento, como así también en aplicaciones domésticas como la temperatura de la heladera o el aire acondicionado. También es posible encontrarlos en los pilotos automáticos de aviones y helicópteros.

En su expresión más sencilla, un sistema de control, o lazo de control (denominado así por su representación de diagrama en bloques como se ve en la figura 1.8), se compone de una planta o sistema representada por la ecuación de transferencia $H_p(t)$ la cual es controlada mediante una señal $u(t)$ generada por el controlador $H_c(t)$ en función del error $e(t)$ entre el valor deseado de la salida $sp(t)$ y el valor actual $y(t)$. Es común considerar perturbaciones en el sistema que, en este caso, puede estar representada por la señal $d(t)$. La función de transferencia $H_p(t)$ es una función matemática, en general una ecuación diferencial, o en el espacio de la transformada de Laplace por un cociente de polinomios y funciones exponenciales, que dadas las condiciones iniciales y la entrada independiente determina el valor de la variable a lo largo del tiempo. Por ejemplo, si suponemos que la planta es un motor eléctrico, y queremos controlar la velocidad del mismo, entonces la función de transferencia $H_p(t)$ se vuelve una función expresada en rpm/volt que determina la velocidad de rotación en función de la tensión aplicada y el estado inicial.

En general, es necesario expresarlo como una ecuación diferencial o en diferencias porque se deben tener en cuenta la dinámica de los sistemas. Es decir que el motor no alcanza automáticamente su velocidad final al aplicarle tensión, sino que, el mismo debe acelerar para llegar a ese valor.

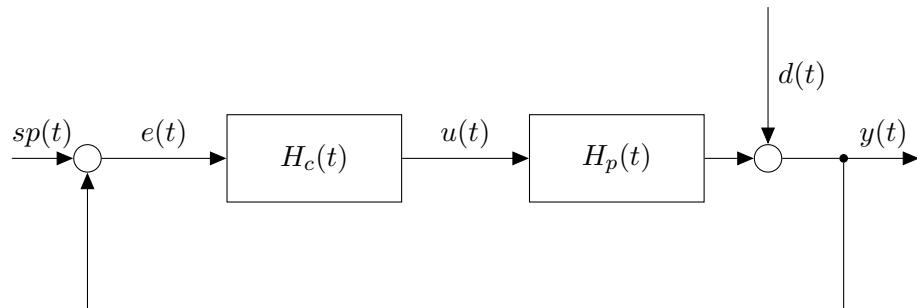


Figura 1.8: Diagrama de bloques de un sistema de control

Si ahora consideramos el sistema de control de un quadrotor, es posible proponer diferentes objetivos: controlar la actitud, la velocidad de rotación en algún eje, la velocidad de traslación o la altura. Si tomamos el caso del control de altura, el piloto (o el controlador) determina, la altura a la que quiere volar mediante la entrada de referencia $sp(t)$ y el controlador de altura $H_c(t)$ determina el empuje, $u(t)$ que deberán hacer los actuadores en conjunto para corregir la altura en función del error que se está cometiendo, $e(t)$, calculado como la diferencia entre el valor deseado y la altura real de la aeronave, $y(t)$, medida por algún sensor. En este caso, la función de transferencia $H_p(t)$ determina cómo evolucionará la altura de la aeronave en función del tiempo, las condiciones iniciales y el empuje aplicado. La ley de control que rige, $H_c(t)$, no es única y es el trabajo del diseñador del sistema elegirla para que esta efectúe acciones correctivas de acuerdo con las características deseadas.

A lo largo de este trabajo se estudiará en profundidad este tipo de controladores y se los implementará de manera digital en un sistema embebido que permitirá controlar en vuelo dife-

rentes variables del quadrotor. En particular este trabajo se destaca porque no se utilizará un controlador comercial, sino que se diseñara un sistema completo partiendo de la selección de componentes, el armado del circuito impreso, la programación de los algoritmos de control, etc.

1.2.3. Guiado

El guiado es la rama del control que se encarga de ubicar a un vehículo en una determinada posición o hacer que este siga trayectorias de posición a lo largo del tiempo. Estas pueden ser puntos claves en un mapa, o trayectorias precisas de actitud y posición que el vehículo debe tomar a lo largo de un camino en determinado momento. En general, para el primero de los casos, utilizando un sistema de posicionamiento, como por ejemplo GPS, es posible hacer que un quadrotor autónomo recorra puntos marcados en un mapa como se muestra en la figura 1.9a. En este caso es posible que la actitud que el quadrotor tome durante el recorrido no sea precisamente determinada, sino que, lo que se busca es la posición del vehículo.

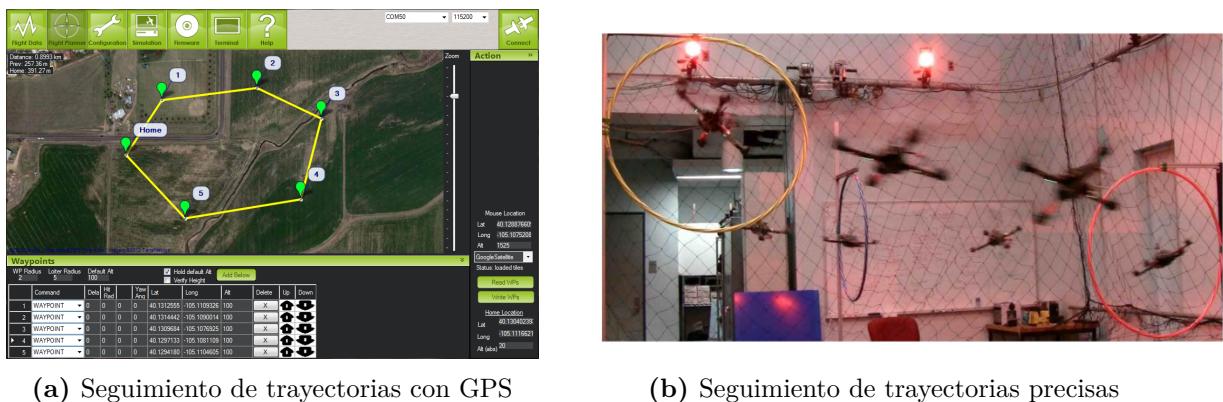


Figura 1.9: Diferentes métodos de guiado para seguimiento de trayectoria

Para el segundo de los casos en general se utilizan sistemas de visión externos para determinar la posición de la aeronave ya que la propagación de mediciones iniciales no resulta ser lo suficientemente precisa. En la figura 1.9b se puede ver un ejemplo de guiado en donde un quadrotor debe realizar un recorrido preciso para pasar por dentro de obstáculos fijos.

En este trabajo en particular no se estudiará el problema de guiado y, de ser necesario, se realizará manualmente por un piloto mediante una realimentación visual.

1.3. Estado del arte

Existen hoy en día decenas de quadrotores que se comercializan en todo el mundo. Dada la gran variedad de usos que se le suelen dar, se propone una categorización en función del propósito final para el que han sido desarrollados que se presenta en la figura 1.10. A continuación se presenta un descripción breve de lo que hoy se entiende como estado del arte en cada una de estas categorías.

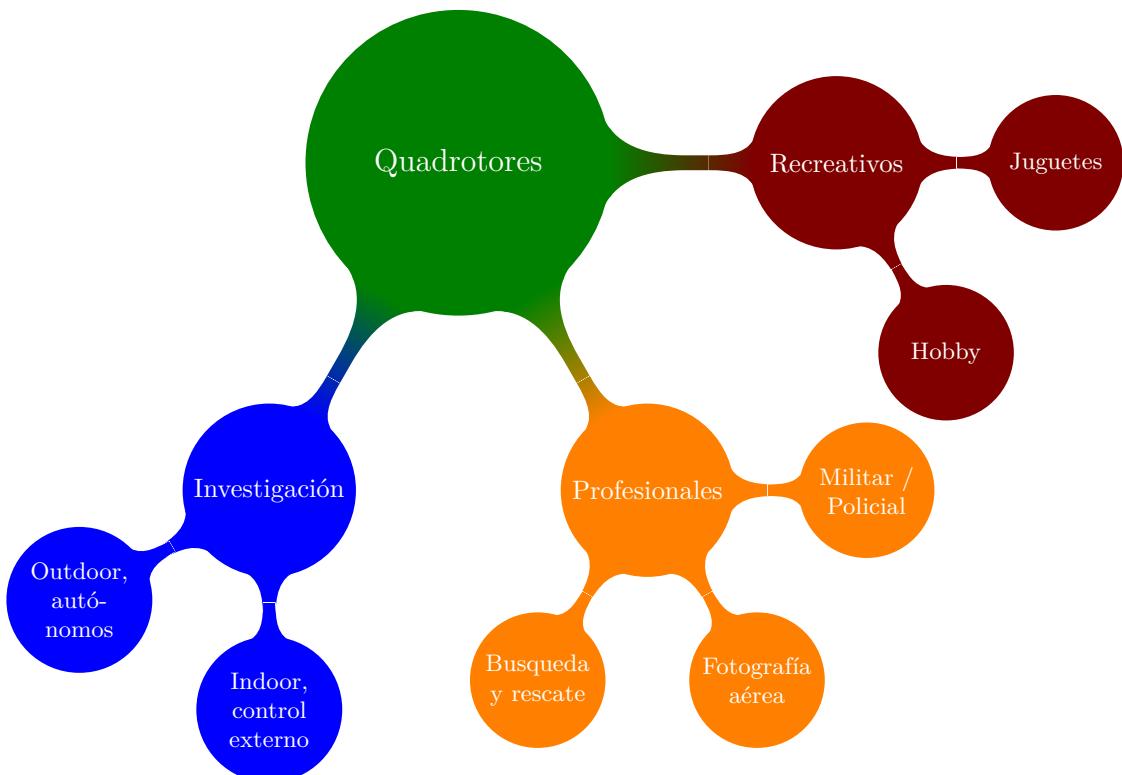


Figura 1.10: Clasificación de quadrotores

1.3.1. Quadrotores recreativos

En esta categoría se agrupan dos tipos de quadrotores y en ambos casos se caracterizan por su bajo costo y su facilidad de uso.

Los quadrotores tipo juguete comenzaron con el popular “Parrot AR Drone”. Este quadrotoor es capaz de volar en un entorno cerrado y ser controlado mediante un teléfono inteligente. Posee cámaras de video en la parte delantera e inferior que permiten manejarlo con facilidad. No tienen capacidad de ser mejorado o expandido mediante cargas utiles y su finalidad es puramente recreativa. Su precio actual es de USD 300 aproximadamente [4].

Por otra parte, uno de los más nuevos en el mercado es el quadrotoor “Phantom” que permite volar en entornos abiertos y puede cargar una cámara de alta resolución. Esta equipado con un receptor de GPS y es capaz de volar en trayectorias planificadas previamente desde una computadora o teléfono celular y en caso de perdida de enlace de radio tiene la capacidad de volver automáticamente al punto de partida. Su precio es de USD 680 [5].

En la figura 1.11 se pueden ver estos dos modelos.



(a) Parrot Ar Drone

(b) Phantom

Figura 1.11: Quadrotores de la categoría “juegos”

En la categoría de hobby resulta más difícil citar ejemplos ya que en general estos quadrotores son armados por aficionados que construyen sus propias partes y compran otras. El punto clave de estos quadrotores es la computadora de abordo y en general se usan computadoras basadas en Arduino, o plataformas similares, para su realización [6, 7]. Se caracterizan por ser basados en software y hardware libre y son capaces de funcionar con diferentes configuraciones. Algunos han sido utilizados como plataformas de investigación pero en general su uso es de un público aficionado. Dentro de las computadoras de vuelo más populares de este segmento se encuentra el Arducopter, Openpilot, Papparazzi, MultiWii y Mikrokopter [?, 6–8]. En general suelen estar más orientados a los accesorios que se les pueden poner (cámaras, sensores, radios, etc.) y se deja el control de vuelo y el software para los desarrolladores del proyecto lo cuál los hace difícil de modificar para aplicaciones de investigación. Su poder de procesamiento suele ser limitado ya que en general utilizan microcontroladores pequeños de 8 bits que apenas satisfacen los requerimientos del proyecto. El precio de estas computadoras se encuentra en USD 200 y USD 400 y se les debe agregar luego el quadrotor y los equipos de radio.

1.3.2. Quadrotores profesionales

Los quadrotores profesionales se comercializan desde hace algunos años y son, a diferencia de los de aficionados, orientados a la carga útil. Es decir, su diseño ha sido pensado para llevar diferentes cargas como ser cámaras de alta definición, cámaras térmicas, sensores, etc. Son en general diseños propietarios y el usuario solo puede usarlos con el objetivo que fueron diseñados. Generalmente vienen provistos de estaciones terrenas, antenas y de todo el soporte necesario para utilizarlo directamente en la aplicación deseada. Se caracterizan también por ser aeronaves muy confiables ya que la carga que transportan suele ser valiosa. El precio de este tipo de quadrotores ronda los 5 a 20 mil dólares.

Dentro de la categoría de seguridad o militares, suelen estar equipados con cámaras de vigilancia y son utilizados para monitorear zonas peligrosas. Resultan ser una alternativa mucho más económica al helicóptero convencional y como también es más silencioso puede sobrevolar zonas a menor altura sin ser detectado. En la figura 1.12a se pueden observar los que recientemente fueron adquiridos por el departamento de policía de la localidad de Tigre.

La más popular de las aplicaciones de hoy en día es para la fotografía aérea. Estos quadrotores suelen estar equipados con cámaras fotográficas o de video profesionales y son utilizados para la cobertura de eventos deportivos, películas y comerciales. El costo de operación es una fracción con el alquiler de un equipo de filmación aérea en helicóptero. En general estos pueden ser

comprados o alquilados, y su precio esta dada por la cámara que utiliza más que por el quadrotor. En la figura 1.12b se puede ver un hexacoptero (de seis rotores en vez de cuatro) equipado con una camara profesional de fotografía. Dado que estas aplicaciones buscan tomas precisas, la camara suele encontrarse dentro de un cardán que se puede controlar independientemente de la estructura del quadrotor permitiendo independizar los movimientos de la camara de los de la aeronave.

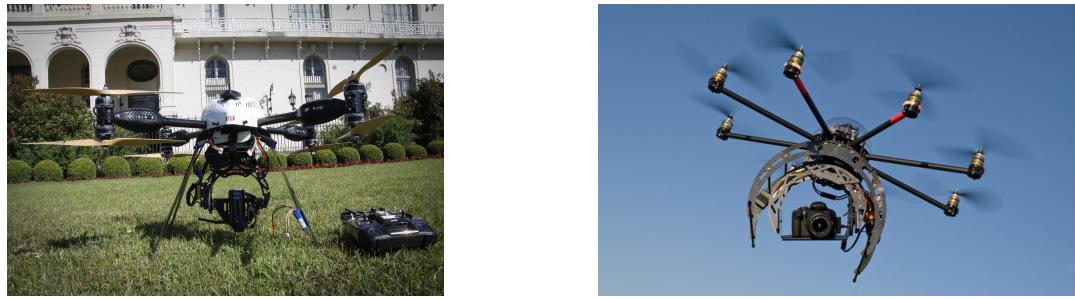


Figura 1.12: Quadrotores de la categoría “profesionales”

1.3.3. Quadrotores de investigación

Los quadrotores de investigación son en general realizados en universidades. Algunos de los temas que se encuentran en desarrollo en la actualidad son: vuelo en formación, enjambres de aeronaves, trabajo colaborativo, nano quadrotores (del tamaño de la palma de una mano o inclusive más pequeños), vuelo autonomo, aprendizaje, etc. [1,2,9]. En general estas plataformas no se encuentran abiertas para su uso y muchas de ellas dependen de un sistema de navegación basado en visión que lo hace muy costoso. En las figuras 1.13a y 1.13b se pueden ver dos de estos proyectos.



Figura 1.13: Quadrotores de la categoría “investigación”

1.4. Resumen del capítulo

En este primer capítulo se presentaron los conceptos generales de los quadrotores como caso particular de los UAVs y se introdujeron las nociones básicas de la navegación, el control y el guiado. Se presentaron, también, ejemplos de la aplicación de los quadrotores y una idea de las innovaciones en esta tecnología. El presente trabajo se enfoca en estudiar el quadrotor de forma detallada y en base a este estudio diseñar e implementar una computadora de vuelo capaz de mantener al sistema en “hovering”. En la actualidad la denominación para este tipo de computadoras no es uniforme y se las suele llamar también “computadora de abordo”, “sistema de control de vuelo”, “piloto automático”, etc. [5, 6]. En este trabajo se la llamará de ahora en adelante “computadora de vuelo”.

Se presentaron soluciones comerciales y libres de este tipo de computadoras, pero se observó que en general ninguna logra satisfacer los requerimientos propuestos, ya sea en el caso de las computadoras para aficionados, por su limitada potencia de cálculo y capacidad de expansión, o bien por el alto costo y/o diseño cerrado característico de las computadoras profesionales.

Se presentará a lo largo del trabajo el diseño de una computadora de vuelo de capaz de implementar los algoritmos de control de actitud para un quadrotor, y luego de implementar dicho control se mostrará que es posible considerar que en trabajos futuros podrían implementarse en la misma computadora también aplicaciones de navegación y guiado simples, dejando incluso algunos de recursos disponibles para aplicaciones de alto nivel. Si bien esto es posible en la actualidad con algunas de las computadoras de vuelo para aficionados, en general estas no disponen de recursos adicionales ni fueron diseñadas para ser expandidos.

CAPÍTULO 2

Modelado de la planta

El primer paso para diseñar un sistema de control es tener un buen conocimiento del modelo matemático de la planta que se busca controlar. Este modelo debe representar los diferentes aspectos se que buscan controlar en una planta, es decir, su dinámica. La búsqueda de este modelo forma una parte considerable del trabajo del diseñador de un sistema de control, ya que por lo general se desconoce. Como este caso no es la excepción, el modelo matemático del quadrotor se desconoce. En este capítulo se presenta el trabajo desarrollado en cuanto al modelado del quadrotor que incluyó análisis mecánico, experimentación en bancos de pruebas con los actuadores y ajustes empíricos.

Se pretende encontrar un modelo que describa la dinámica del quadrotor según el diagrama de bloques de la figura 2.1. Éste debe tener cuatro entradas que representen las señales de control, y salidas que se corresponden con los parámetros observables de la aeronave, como ser actitud y posición en función del tiempo como así también sus derivadas.

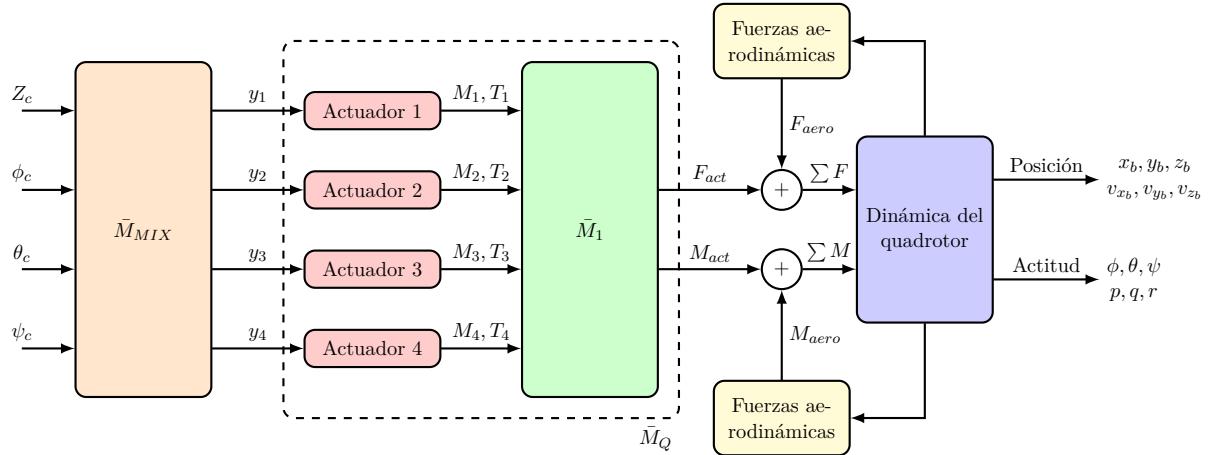


Figura 2.1: Modelo matemático completo del quadrotor

El modelo comienza con las cuatro entradas de control del sistema Z_C , ϕ_C , θ_C y ψ_c , asociadas directamente con estos cuatro grados de libertad. Estas entradas, que luego serán comandadas por el piloto automático, se deberán combinar de manera adecuada para actuar en cada uno de los cuatro actuadores según la disposición y sentido de giro de los mismos. Esta mezcla de controles se verá representada por la matriz M_{mix} , de donde se pueden obtener las entradas de control a cada actuador, que como se verá a continuación, proporcionan momentos y fuerzas al sistema. Luego estas fuerzas y momentos, tomadas desde el marco de referencia del actuador, son combinadas y llevadas al marco del quadrotor, acción realizada por la matriz M_1 . Se verá también que es posible combinar estos dos últimos conceptos, la matriz M_1 y el modelo de los actuadores, en una matriz simplificada de ganancias denominada M_Q marcada en puntos en el diagrama de bloques. A la salida de estos bloques, se encuentran las fuerzas y momentos internos producidos por los actuadores, que sumado a las generadas por los efectos aerodinámicos, componen el total de fuerzas y momentos que actúan sobre el cuerpo rígido. Aplicando la teoría general de la dinámica del cuerpo rígido, se obtendrá un modelo simplificado y linealizado que representará la dinámica del quadrotor en vuelo “hovering”.

A lo largo de este capítulo se detallarán y estudiarán cada uno de los bloques que componen el diagrama y se finalizará con un modelo matemático que describe el comportamiento de cada eje de la aeronave por separado.

2.1. Dinámica del quadrotor

Como se vio en las ecuaciones 1.4 y 1.7, el sistema de ecuaciones que representa la dinámica del quadrotor es no lineal y se encuentra acoplado. Si bien este sistema describirá con detalle la evolución del vuelo, resulta de utilidad disponer de un sistema lineal y desacoplado que describa de forma aproximada la misma dinámica en un determinado punto de trabajo, debido a la gran cantidad de herramientas desarrolladas para el análisis y el diseño de algoritmos de control para sistemas lineales. El punto de trabajo escogido, como se vio en el capítulo 1, debe ser

el de “hovering” y se busca mantener la actitud del quadrotor dentro de su entorno. Es decir que el sistema busca mantener la actitud de la aeronave en pitch y roll lo más cercanas a 0° posible. Es posible entonces suponer que la aeronave solo se moverá en pequeñas cantidades y se puede entonces asumir un modelo de desviaciones para linealizar la planta. Esto se verá reflejado en que las variables de velocidad angular p, q y r serán lo suficientemente pequeñas en todo momento como para descartar sus productos de las ecuaciones 1.4 y 1.7, obteniendo así un modelo simplificado según:

$$\begin{aligned} F_x &= m\dot{u} & (2.1) \\ F_y &= m\dot{v} \\ F_z &= m\dot{w} \\ L &= I_{xx}\dot{p} \\ M &= I_{yy}\dot{q} \\ N &= I_{zz}\dot{r} \end{aligned}$$

Las ecuaciones linealizadas en el punto de trabajo, ya no presentan acoplos entre ejes, lo cuál permitirá desarrollar una estrategia de control “por eje” que se desarrollará a lo largo del trabajo. Dado que el objetivo es solamente realizar un controlador de actitud de la aeronave, las primeras tres ecuaciones no se estudiarán en detalle, en cambio, las últimas tres conformarán la base de estudio.

2.2. Fuerzas y momentos sobre el centro de masa: \bar{M}_1

Para definir el sentido de las fuerzas y momentos aplicados sobre el cuerpo, es necesario determinar el sentido de giro de los motores y su ubicación relativa al sistema de referencia de cuerpo. En la figura 2.2 se puede observar la definición utilizada:

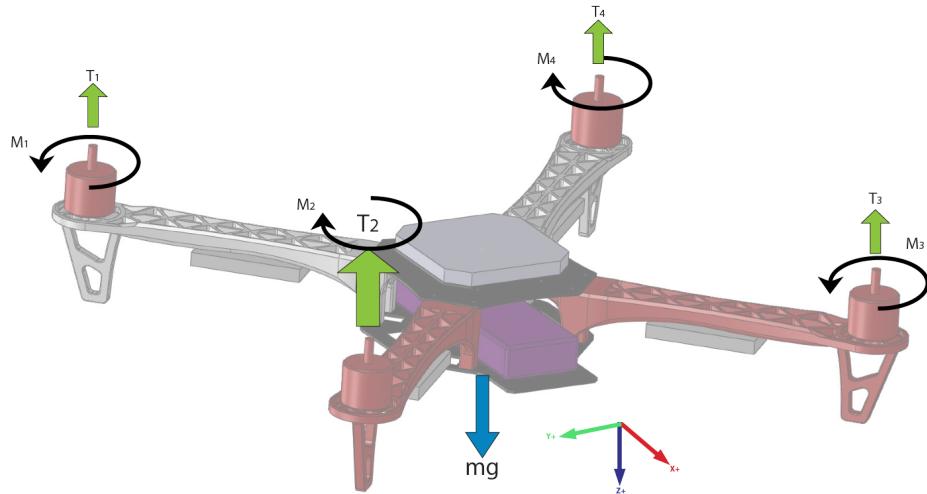


Figura 2.2: Diagrama de cuerpo libre

donde las fuerzas y momentos presentes son:

- Las fuerzas de cada uno de los motores en el eje $-\hat{z}$ de la terna fija en la aeronave: T_1, T_2, T_3 y T_4 .
- La fuerza de gravedad en la dirección \hat{z}_e de la terna fija en la tierra: mg
- El torque generado por el desplazamiento de las hélices en sentido contrario al de rotación de las mismas: M_1, M_2, M_3 y M_4 .
- El torque generado por el brazo de palanca de las fuerzas de empuje de cada actuador.

Del diagrama de cuerpo libre de la figura 2.2 se pueden reformular las ecuaciones de fuerzas en forma matricial como:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ -T_1 - T_2 - T_3 - T_4 + mg \end{bmatrix} \quad (2.2)$$

Donde T_1, T_2, T_3 y T_4 son el empuje producido por cada uno de los actuadores. Mientras que la ecuación de momentos se transforma en:

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} -T_1d - T_2d + T_3d + T_4d \\ -T_1d + T_2d + T_3d - T_4d \\ -M_1 + M_2 - M_3 + M_4 \end{bmatrix} \quad (2.3)$$

donde M_1, M_2, M_3 y M_4 representan el contra torque generado por cada uno de los actuadores y d representa el brazo de palanca medido como la mínima distancia a la línea que pasa por el centro de masa del cuerpo. Las ecuaciones 2.2 y 2.3 conforman las entradas a las ecuaciones 2.1 y pueden ser resumidas con una única matriz según:

$$\begin{bmatrix} F_z \\ L \\ M \\ N \end{bmatrix} = M_1 \cdot \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ M_1 \\ M_2 \\ M_3 \\ M_4 \end{bmatrix} + \begin{bmatrix} mg \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.4)$$

donde

$$M_1 = \begin{bmatrix} -1 & -1 & -1 & -1 & 0 & 0 & 0 & 0 \\ -d & -d & d & d & 0 & 0 & 0 & 0 \\ -d & d & d & -d & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & -1 & 1 \end{bmatrix} \quad (2.5)$$

En la siguiente sección se estudiarán los valores y formas que toman el empuje T_i y los momentos M_i producidos por cada actuador para poder utilizar la matriz \bar{M}_1 para hallar las fuerzas y momentos sobre el centro de masa.

2.3. Actuadores

Los actuadores del sistema, de los cuales se tienen cuatro iguales, están compuesto por tres partes: el ESC, el motor y la hélice, como se muestra en la figura 2.3.

Las hélices son las encargadas de generar empuje a partir de la rotación causada por el momento aplicado por el motor eléctrico. Estas en general son diseñadas con diferentes cantidades de aspas, desde dos hasta seis para diferentes aplicaciones y pueden tener varias formas y tamaños. Su principio de funcionamiento se basa en los teoremas de Bernoulli y Venturí, al igual que las alas de un avión [10] pero en este caso, para generar viento aparente que luego genere sustentación.



Figura 2.3: Fotografía de las partes que componen al subsistema de actuadores

Las hélices son caracterizadas por su forma (en inglés conocido como “aerodynamic airfoil”) el cual puede resumirse en dos parámetros constructivos básicos: el paso y el diámetro. El paso determina cuánto avanzaría la hélice si estuviera en un medio que no deslizara (concepto similar al paso de un tornillo o tuerca) y el diámetro es el tamaño del disco barrido, como se muestra en la figura 2.4. Existen aplicaciones en las que el paso de la hélice puede ser variado por el usuario a fin de lograr mejor eficiencia en el sistema o casos aún más complejos, como el helicóptero moderno, en donde el paso de la hélice es variado a lo largo de una misma revolución para generar diferente empuje en diferentes lugares y así orientar el empuje neto para producir los movimientos de la aeronave.

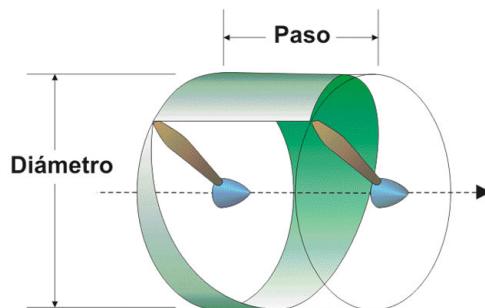


Figura 2.4: Paso y diámetro de una hélice convencional

Para el caso particular de los quadrotores se encontró que las hélices plásticas de dos palas y de paso fijo suelen ser las más eficientes, pero existen casos interesantes en donde se utilizaron hélices de paso variable [11] como así también hélices de tres y cuatro palas [12]. Como conclusión general de estos trabajos se obtiene que, para realizar maniobras agresivas o vuelos acrobáticos, resultan ser más eficientes en cuanto a capacidad de control las hélices de paso variable; pero estas implican un sistema mecánico de control complejo que no está dentro de los objetivos. Por otra parte, no se han encontrado referencias que demuestren que la utilización de hélices de

más de dos palas sean mejores o peores para este tipo de aeronaves. Pero de manera general se rescata que estas producen más empuje al mismo diámetro que una de dos palas lo cual permite reducir el tamaño total de la aeronave. Al ser estas menos utilizadas, se prefirió optar para este trabajo por hélices de dos palas. La elección del tamaño y forma de las hélices es un proceso complejo, por eso se prefirió utilizar las hélices propuestas por el fabricante del chasis. Estas son hélices plásticas de 8.0" de diámetro y 4.5" de paso.

Los motores eléctricos comúnmente utilizados en este tipo de aeronaves son los llamados BLDC por sus siglas en inglés que significa: motor de corriente continua sin escobillas. Son motores trifásicos, y son hoy en día, los motores eléctricos más eficientes. Dado su bajo peso y su gran confiabilidad son los preferidos para estas aplicaciones. Para controlarlos, es necesario comutar las fases en función de la posición del rotor y por ende, es necesario sensar de alguna manera la posición del rotor. Para ello se suelen usar sensores de "efecto hall", o la medición de la tensión generada por la fuerza contra electromotriz (Back EMF en inglés) al desconectar cada fase. A fin de variar la velocidad se debe modular esta señal trifásica con una señal de PWM, que cambiando su ciclo de funcionamiento, varía la tensión media entregada a cada fase. En la figura 2.5 es posible observar la señal de la fuerza electro motriz que se puede medir en dos de las fases de un motor BLDC controlado por PWM.

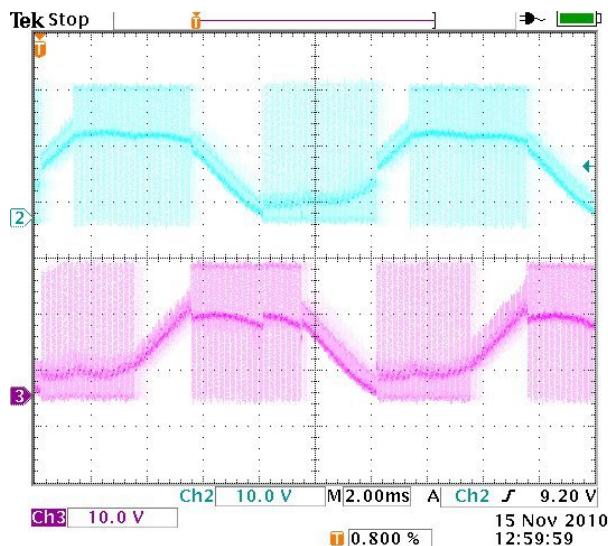


Figura 2.5: Voltaje generado por la fuerza contra electromotriz de un motor BLDC típico controlador por PWM. Imagen tomada de Bodin Electric Company.

También es importante destacar que estos motores suelen consumir una corriente de entre 15A y 30A dependiendo de su tamaño. Por ende, el sistema de variación de velocidad no suele ser sencillo de realizar. Es por eso que se prefirió utilizar motores y variadores (ESC) comerciales que ya se encuentren probados para este tipo de aplicaciones. El variador incorpora un microcontrolador digital que acepta señales de control de hasta 400Hz de actualización en forma de señal de PWM. Si bien la frecuencia de la señal de 2.5ms es fija, el ciclo activo, variable entre 1ms y 2ms, como se puede ver en la figura 2.6, determina la tensión aplicada al motor. En el resto del trabajo se hará referencia a esta entrada de control como una señal porcentual adimensional donde el 0% se corresponde con una señal cuyo período activo es de 1ms y el 100% con una de 2ms.

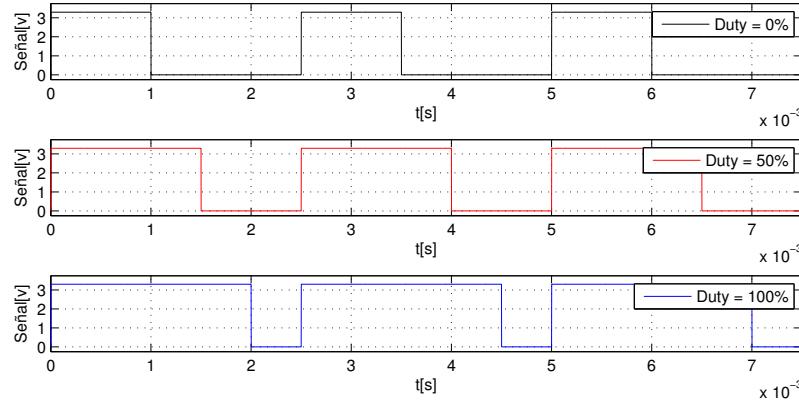


Figura 2.6: Señal de control de entrada para los ESC utilizados.

Utilizando las tres partes en conjunto se forma lo que en este trabajo será llamado actuador. El diagrama en bloques de este se presenta en la figura 2.7.

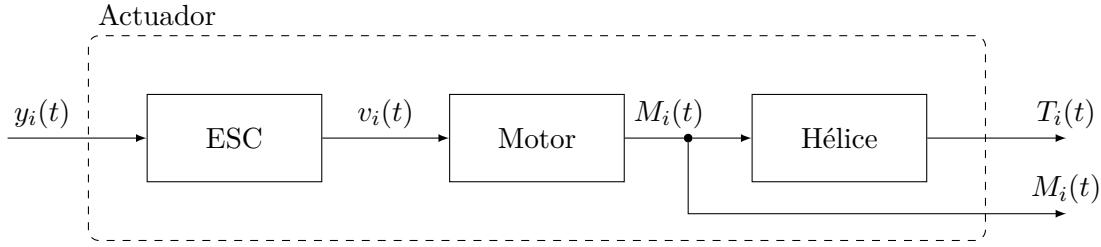


Figura 2.7: Diagrama en bloques del actuador

en donde:

- $y_i(t)$: Es la entrada al actuador i y es una señal de PWM con un duty cycle variable entre 0 % - 100 %.
- $v_i(t)$: Es la señal de control que envía el ESC al motor, en este caso: una señal de tensión trifásica modulada por PWM para variar la velocidad.
- $M_i(t)$: Es el torque generado por el motor y que es aplicado sobre la hélice para generar el empuje. Por el principio de acción y reacción este torque también se aplica, en sentido contrario sobre el centro de masa de la aeronave y también se lo considera una salida del subsistema .
- $T_i(t)$: Es el empuje generado por la hélice debido a su perfil aerodinámico y el torque de entrada $\tau_i(t)$.

Se busca entonces, encontrar las ecuaciones de transferencia (en el espacio de la transformada de Laplace):

$$H_T(s) = \frac{T(S)}{y(s)} \quad (2.6)$$

y

$$H_M(s) = \frac{M(S)}{y(s)} \quad (2.7)$$

que caracteriza al torque ejercido por el motor en función de esta misma entrada. Para poder obtener dichas transferencias, es necesaria una caracterización de estado estacionario (sin incluir transitorios) comúnmente llamada prueba de banco estática, en donde se observan las relaciones del motor luego de los transitorios, y una caracterización dinámica en donde se puede observar cómo llega el actuador a ese estado estacionario.

2.3.1. Empuje y torque producido por una hélice

El empuje estático generado por una hélice i en aire libre [10] está dado por:

$$T_i = C_T \rho A_r r^2 \omega_i^2 \quad (2.8)$$

Donde T es el empuje (o fuerza) generada por el rotor, A_r es el área barrida, ρ la densidad del aire, r el radio de la hélice, ω la velocidad angular y C_T una constante propia de la hélice dada por su geometría.

Dado que para todas las hélices utilizadas las constantes A_r , ρ , r y C_T son prácticamente iguales, es posible concentrar estos parámetros en solo uno, teniendo así la ecuación 2.9 que relaciona la velocidad angular con el empuje generado.

$$T(\omega_i) = K_T \omega_i^2 \quad (2.9)$$

En general, la constante K_T suele ser difícil de estimar sin conocer los perfiles aerodinámicos de una hélice con exactitud y si no es provista por el fabricante de la misma, como sucedió en este caso, es necesario determinarla mediante experimentos en bancos de prueba.

De manera similar al empuje, el torque generado por una hélice en rotación se puede determinar con la ecuación 2.10 [10].

$$M(\omega_i) = K_M \omega_i^2 \quad (2.10)$$

De manera similar al caso del empuje, la constante K_M , de no ser provista por el fabricante, debe ser encontrada mediante experimentos.

Como se describe en las ecuaciones 2.6 y 2.7 lo que se busca en realidad es la relación entre empuje y torque vs. entrada al controlador, que es la única variable manipulable (no así la velocidad angular). Es por eso que es necesario encontrar la relación entre la velocidad angular y la entrada de control $y(t)$ para poder incluirla en las ecuaciones 2.9 y 2.10 en reemplazo de ω_i .

2.3.2. Banco de pruebas de actuadores

Para caracterizar los actuadores se realizaron dos diferentes bancos de prueba, uno para medición de empuje y otro para medición de torque. En ambos casos, también fue necesario la medición de la velocidad angular. Para la medición de empuje y torque se utilizó una celda de carga PS-2008A del fabricante PASCO, mientras que para la medición de velocidad angular se utilizó un sensor reflectivo infrarrojo como tacómetro óptico y se realizaron las mediciones sobre

el rotor del motor ya que resultó más sencillo que tomarlas en las aspas. El banco utilizado fue tipo balancín como se puede ver en la figura 2.8.



Figura 2.8: Banco de pruebas de empuje de actuadores

Para medir empuje se utilizó la configuración de banco mostrada en la figura 2.9a donde el actuador ejerce empuje hacia arriba, lo que se traslada mediante el brazo de palanca (simétrico) en peso sobre la celda de carga. En cambio, para la medición de torque, se rota el motor de forma tal de generar empuje en el mismo plano del balancín y así poder traducir el contra torque a peso en la celda de carga como se observa en la figura 2.9b.

Para las pruebas estáticas, en donde lo que se buscaba era la relación entre diferentes parámetros, como por ejemplo, entrada de control, velocidad angular, empuje y/o torque, se hizo un ensayo en dónde se realizó un barrido de la señal de entrada entre 0 % y 100 % para determinar el empuje y la velocidad angular de los diferentes actuadores. Para evitar el efecto transitorio, luego de cada incremento se esperó hasta que las RPM se estabilizaran para comenzar la toma de datos. Por cada valor de input se tomaron 100 muestras de la velocidad angular y 100 muestras del empuje que luego se promediaron. Esto resultó fundamental ya que la señal proveniente de la celda de carga resultó ser extremadamente ruidosa debido a las vibraciones del banco de pruebas como se puede ver en la figura 2.10.

Cabe destacar que en todos los casos se realizaron las pruebas con los cuatro actuadores para determinar las diferencias entre estos. A continuación se presentan los experimentos y resultados realizados para determinar las diferentes características de los actuadores.

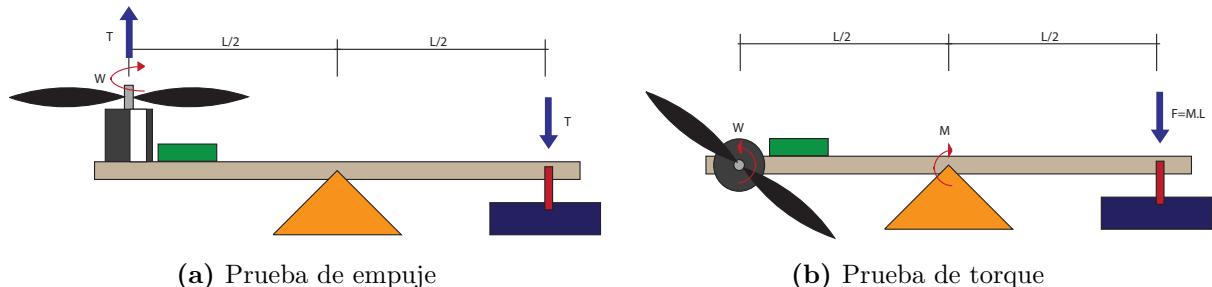


Figura 2.9: Esquemas de los bancos de prueba para empuje y torque

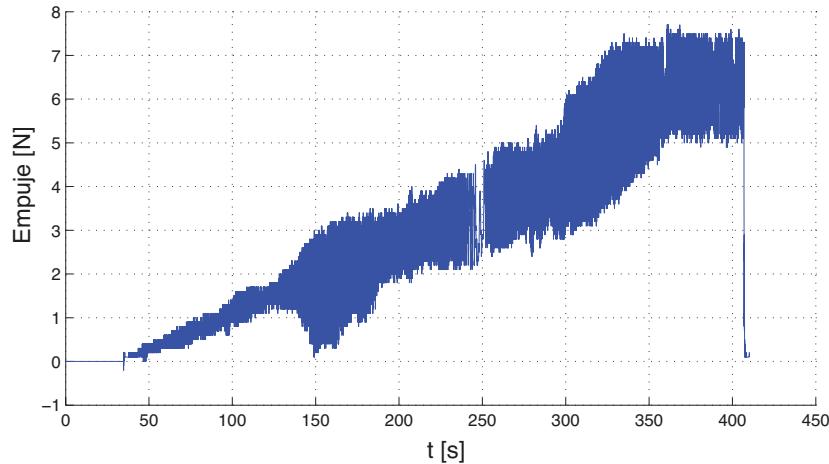


Figura 2.10: Señal de la celda de carga en una prueba de empuje estático

2.3.3. Velocidad angular

La primera variable que se analiza es la velocidad angular de la hélice en función de la entrada de control obteniendo los resultados del gráfico 2.11.

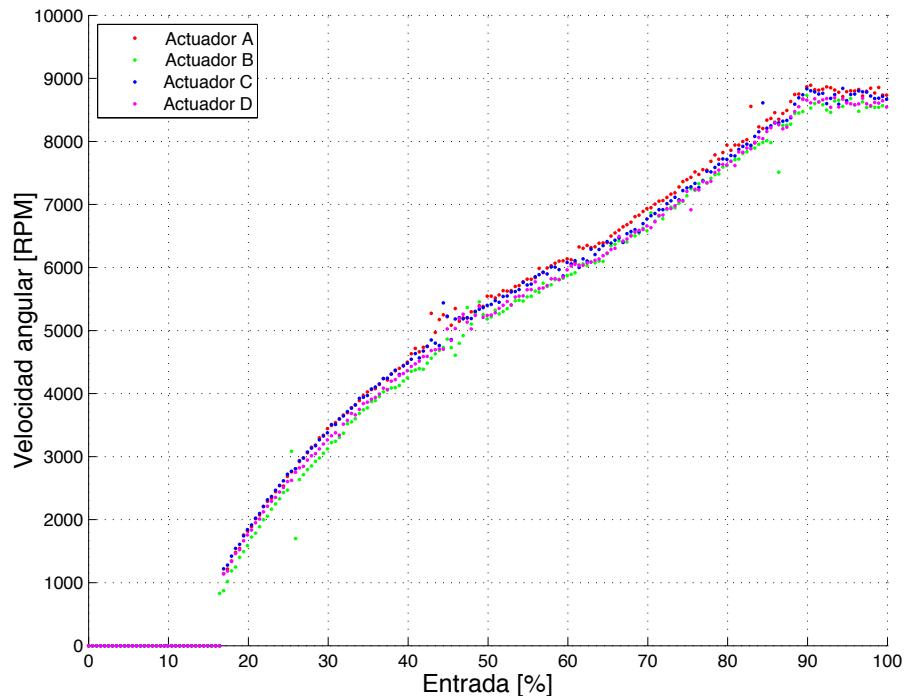


Figura 2.11: Velocidad angular vs. Entrada

Se puede observar en los resultados de este experimento algunos fenómenos no ideales reseñables, como ser la dispersión de los resultados de empuje como así también la presencia de zonas muertas o de saturación. Las principales conclusiones de la figura 2.11 se pueden resumir en la siguiente lista:

- Presencia de una zona muerta por debajo del 17% al igual que para el empuje.
- Hasta el 60% de la entrada, las RPM parecen crecer en forma de $\sqrt{y_i}$. Luego se asemejan a una recta.
- Se puede observar la saturación luego del 90%.

2.3.4. Empuje estático

Se analizó el empuje generado por cada actuador en función de la entrada de control que se varió desde el 0% hasta el 100% a fin de determinar el empuje en todo el rango de operación, dando el resultado que se muestra en la figura 2.12.

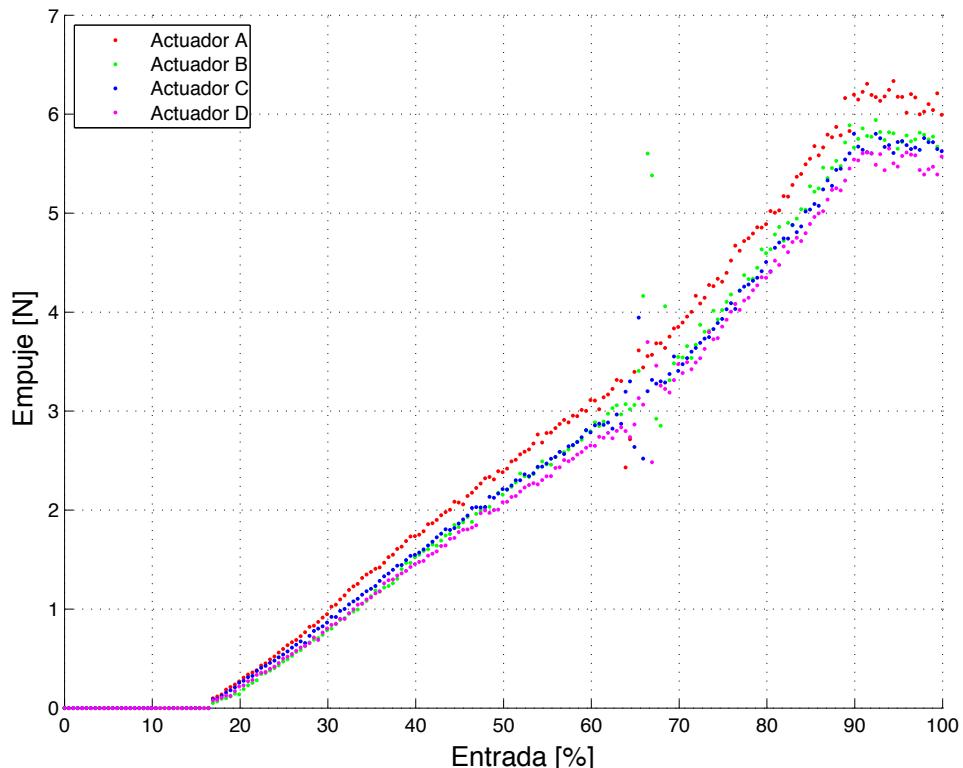


Figura 2.12: Empuje vs. Entrada

De manera similar a la velocidad angular, se obtienen las siguientes conclusiones:

- Existencia de una zona muerta de entrada por debajo del 17% de la entrada de control en donde el actuador no produce empuje.
- Comportamiento lineal en la respuesta del empuje de los actuadores por lo menos hasta el 60%. Luego la respuesta parece volverse cuadrática.
- Todos los actuadores saturan al 90%.

2.3.5. Torque estático

Utilizando el banco de ensayos de torque, se midió el torque que generaba el actuador en sentido contrario a la rotación de la hélice. En la figura 2.13 se pueden observar los resultados de los experimentos.

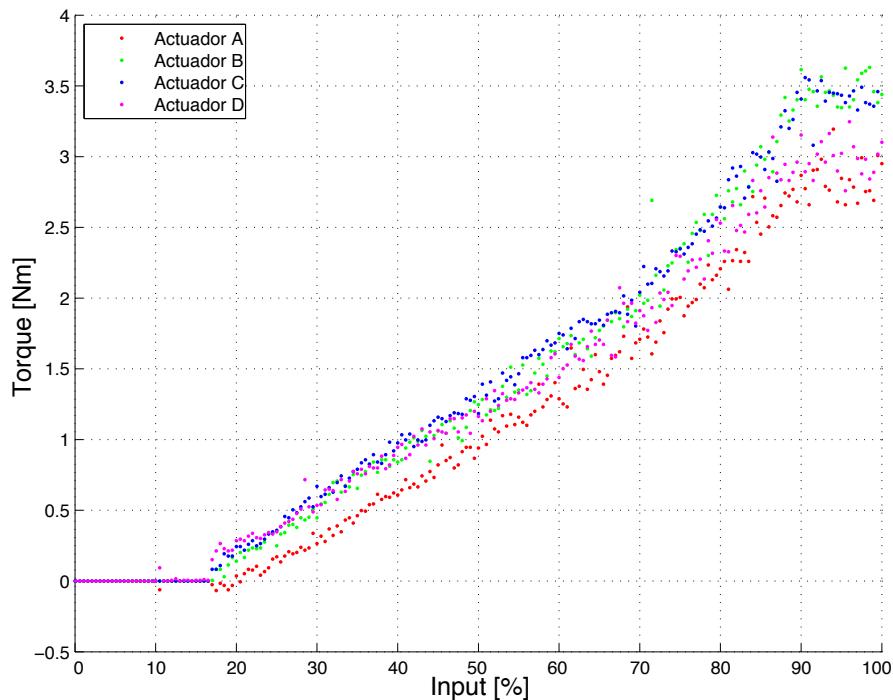


Figura 2.13: Torque vs entrada de control

Debido a los efectos indeseados en la respuesta del motor en la sección anterior tambien es posible observar que luego del 65 % aproximadamente, la respuesta cambia su pendiente.

2.3.6. Constantes de las hélices

Como se vio en las secciones anteriores, para desarrollar el modelo matemático del quadrotor sólo es necesario encontrar la relación entre entrada de control y el empuje o torque. Por ende, las constantes K_T y K_M de las ecuaciones 2.9 y 2.10 respectivamente no resultan de gran interés ya que estas solamente sirven si se conoce la velocidad angular de las mismas. Como también se demostró que los ESC dan una respuesta lineal de empuje y torque, no resulta necesario conocer la relación entre velocidad angular y estas variables. De todas maneras esto puede resultar de suma utilidad si se desea realizar un control a lazo cerrado de la velocidad angular de los motores para mejorar la performance total del sistema, o bien porque se cambia el ESC.

Si se relacionan los resultados de empuje y velocidad angular se pueden obtener las curvas de empuje vs velocidad angular como se muestra en la figura 2.14 como puntos de colores (el amontonamiento de los resultados en los extremos del gráfico se debe a la saturación y a la zona muerta que proveen una velocidad angular que no resulta monótonamente creciente). Se puede observar que la relación es cuadrática, tal como se vio en la ecuación 2.9 y en líneas de colores los ajustes por cuadrados mínimos de estas curvas.

La constante encontrada K_T para cada uno de los actuadores se detalla en la tabla 2.1

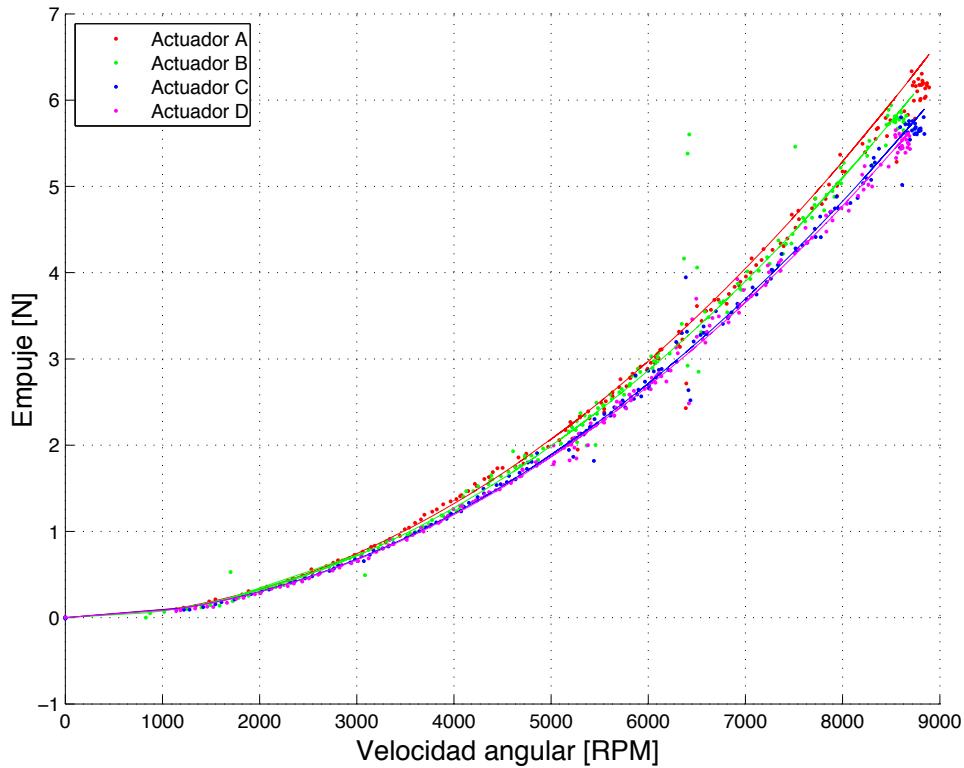


Figura 2.14: Regresión para la función de empuje vs. velocidad angular

Actuador	$K_T \left[\frac{N}{RPM^2} \right]$
K_{T_A}	$8,263 * 10^{-8}$
K_{T_B}	$7,961 * 10^{-8}$
K_{T_C}	$7,543 * 10^{-8}$
K_{T_D}	$7,460 * 10^{-8}$

Cuadro 2.1: Constantes K_T de los diferentes actuadores

2.3.7. Identificación paramétrica

Como sucedió en el caso del modelo dinámico del quadrotor, también se busca acá obtener modelos lineales para los actuadores. Para eso es necesario conocer el punto de trabajo en donde se linearizará el modelo. En este caso se escogió un punto de trabajo que coincide con la condición de “hovering” en la que cada actuador aporte un empuje neto de 1/4 del peso total de la aeronave. Dado que la masa del mismo es de 1037gr, como se detalla en la sección 2.6, el empuje necesario para mantenerla en vuelo es de 10.16 Newtons y por ende cada actuador debe aportar 2.54N. Este valor se encuentra con una entrada de control y_i de aproximadamente 55 % y será el punto en donde se linearizará el actuador, que se denominará como y_{bias} . De ahora en más, las entradas de control y_i serán tomadas como desviaciones respecto de y_{bias} .

Partiendo de esta idea es posible determinar un modelo lineal para el empuje generado por el actuador según se muestra en el gráfico 2.15. El mismo se corresponde con una recta que no

pasa por el origen para cada uno de los actuadores, y que se corresponde con la ecuación:

$$T(y_i)_i = K_{T\omega} (y_i - c_{dz}) \quad (2.11)$$

Se puede observar en la figura 2.15 estas rectas superpuestas con los datos obtenidos mediante los experimentos donde los parámetros de cada una se presentan en la tabla 2.2.

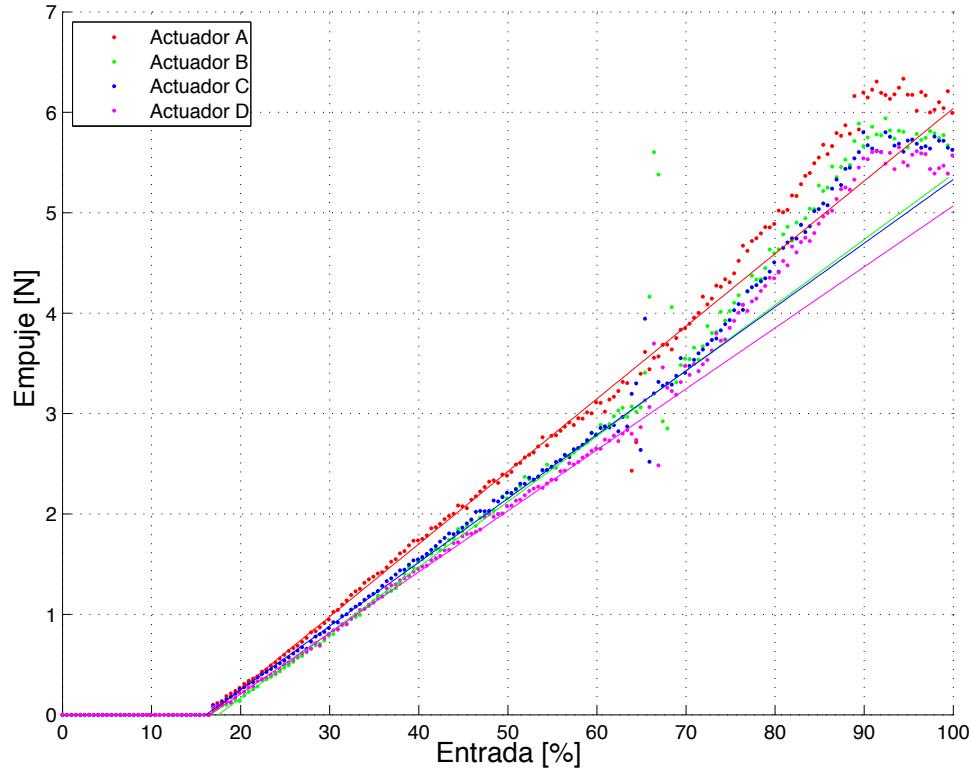


Figura 2.15: Regresión de la función de Empuje vs. Entrada de control

Actuador	$K_{T\omega} [\frac{N}{\%}]$	$C_{dz} [\%]$
Actuador A	0.072	16.37
Actuador B	0.065	17.37
Actuador C	0.064	16.00
Actuador D	0.061	16.47

Cuadro 2.2: Constantes para la transferencia de velocidad de los diferentes actuadores

De manera analógica se puede representar el torque generado por cada actuador como una función lineal que se corresponde con la ecuación:

$$M(y_i) = K_{M\omega} (y_i - c_{dz}) \quad (2.12)$$

Haciendo un ajuste por cuadrados mínimos de la parte lineal se obtienen los resultados mostrados en el gráfico 2.16 y el cuadro 2.3 resume los parámetros del mismo.

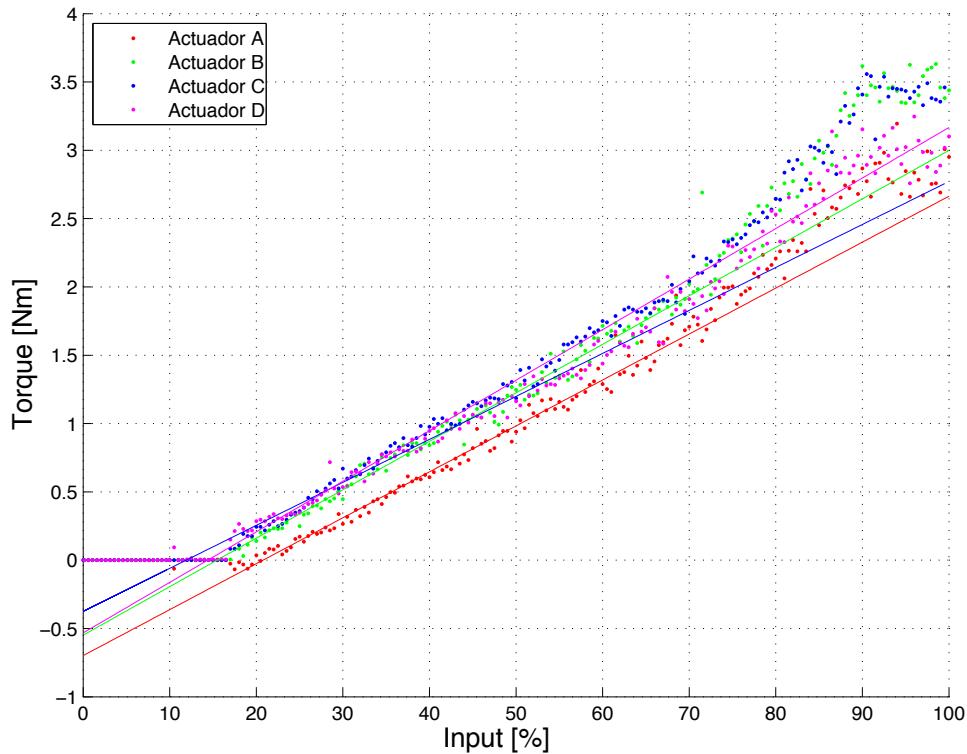


Figura 2.16: Ajuste de la función de torque vs la entrada de control

Actuador	$K_{M\omega} [\frac{Nm}{\%}]$	$C_{dz} [\%]$
Actuador A	0.034	16.37
Actuador B	0.035	17.37
Actuador C	0.031	16.00
Actuador D	0.037	16.47

Cuadro 2.3: Constantes para la transferencia de torque de los diferentes actuadores

2.3.8. Linealización

Si bien se pueden obtener diferentes funciones para el empuje y el torque generado por cada actuador, resulta de gran utilidad disponer de un modelo único de actuador promedio entre todos que luego pueda ser linealizado para obtener un modelo lineal. Se realizó un ajuste por cuadrados mínimos en donde se utilizaron los puntos de los cuatro actuadores promediados para cada valor de entrada y_i . Se obtuvo entonces un único conjunto de parámetros promedio para los actuadores según se ve en la tabla 2.4 para el caso del empuje y 2.5 en el caso del torque.

$K_{T\omega} [\frac{N}{\%}]$	$C_{dz} [\%]$
0.0667	17.03

Cuadro 2.4: Constantes para la transferencia de empuje promedio

Las ecuaciones 2.11 y 2.12 son del tipo afín, es decir, funciones lineales con ordenada al

$K_{M\omega} \left[\frac{Nm}{\%} \right]$	$C_{dz} [\%]$
0.0346	15.90

Cuadro 2.5: Constantes para la transferencia de torque promedio

origen distinta de cero. Aún así, todavía no se puede trabajar con ellas como si fuesen funciones puramente lineales, como se busca para expresar funciones de transferencia. Es por eso que se utilizó el teorema de Taylor para linealizar estas ecuaciones en el punto de trabajo deseado. De esta manera se puede obtener un modelo lineal para el actuador según:

$$\Delta T_i(y_i) = K_{T\omega} \Delta y_i \quad (2.13)$$

$$\Delta M_i(y_i) = K_{M\omega} \Delta y_i \quad (2.14)$$

Como se vio anteriormente, es necesario que cada actuador entregue un total de 2.54N de empuje para poder mantener a la aeronave en hovering. Para ello se encuentra que la entrada de control que se necesita para el actuador promedio es $y_i = 55\%$ que será el punto de trabajo en el cual será válida la linealización y un entorno del mismo, como ya se mencionó anteriormente. De ahora en adelante se omitirá el símbolo Δ de las ecuaciones para simplificar la nomenclatura, pero siempre se estará haciendo mención a desviaciones respecto del punto de trabajo.

2.3.9. Respuesta transitoria del actuador

Para ensayar la respuesta dinámica del actuador se realizó un experimento en el cual se introdujeron diferentes valores de entradas tipo escalón para ver la respuesta transitoria. Dado que el banco de pruebas utilizado para la medición de empuje estático no tenía la suficiente resolución temporal como para hacer las mediciones, se utilizó un tacómetro óptico y se midió solamente la velocidad angular. Los valores fueron ajustados al punto de trabajo y luego transformados de velocidad a angular en empuje utilizando la ecuación 2.9 con las constantes encontradas en los experimentos previamente descriptos. Es por eso que los mismos se encuentran centrados en cero. Es posible observar los resultados de este ensayo en la figura 2.17.

Luego se realizó un ajuste por cuadrados mínimos de la transferencia de la planta para diferentes tipos de planta. Se propusieron tres diferentes tipos de respuesta y se ajustó la respuesta para cada caso. En la figura 2.18 se pueden observar los resultados para sistemas de primer orden, de primer orden con tiempo de retardo y de segundo orden donde la función de transferencia para cada modelo se puede ver en la tabla 2.6.

Modelo	Ecuación	T_1	T_2	T_d
FO	$G(S) = \frac{1}{(T_1 s + 1)}$	0.080		
FOPDT	$G(S) = \frac{1}{(T_1 s + 1)} e^{-T_d s}$	0.047		0.031
SOPDT	$G(S) = \frac{1}{(T_1 s + 1)(T_2 s + 1)}$	0.0385	0.0385	

Cuadro 2.6: Constantes actuadores

Los resultados de los tres tipos de sistemas son similares y por ende se propuso utilizar únicamente el modelo de primer orden para el resto del trabajo por ser este el más sencillo de

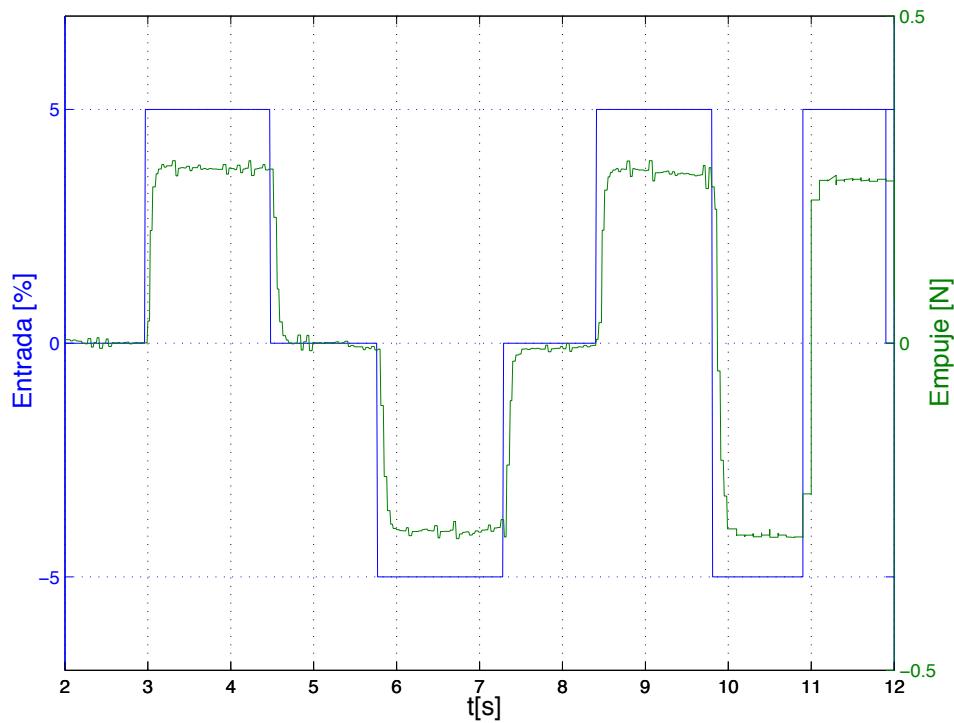


Figura 2.17: Respuesta transitoria al escalón

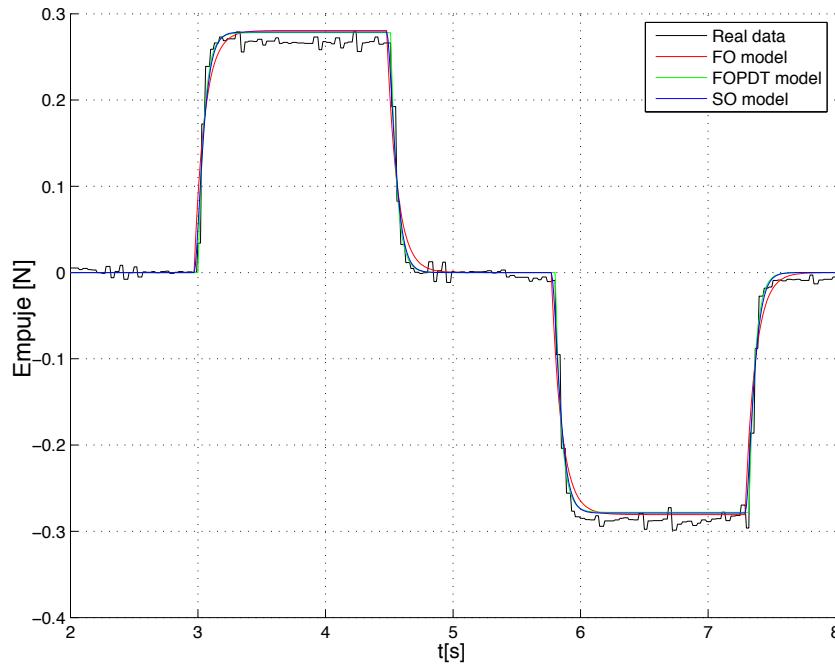


Figura 2.18: Respuesta transitoria al escalón

los tres. Si se toma en cuenta esta transferencia, una simulación del subsistema del actuador completo verifica el modelo utilizado como se puede ver en la figura 2.19

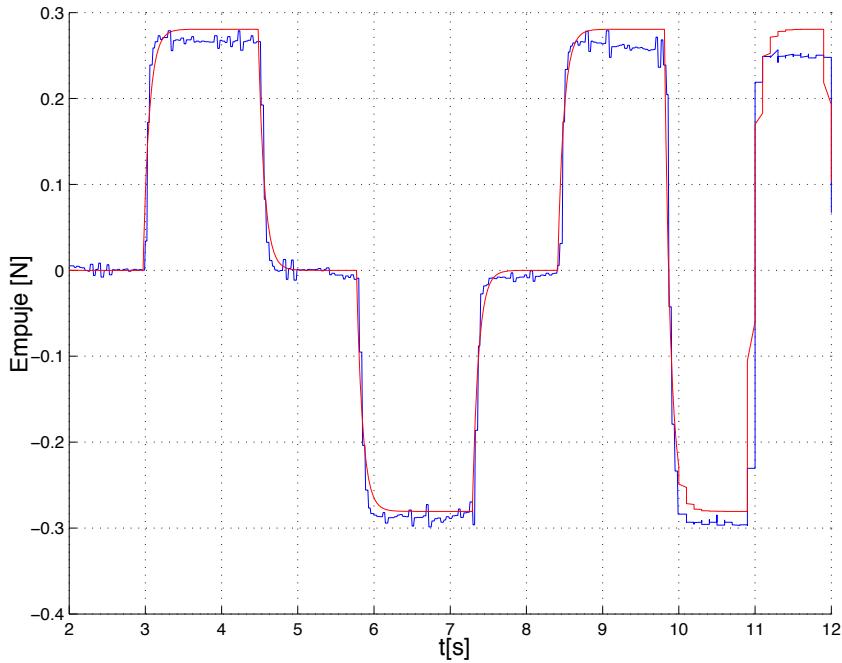


Figura 2.19: Respuesta transitoria al escalón

El tiempo de crecimiento obtenido $T_1 = 0,08\text{s}$ en este experimento, se encuentra dentro del orden del error cometido en las mediciones de tiempo y velocidad angular por lo que es difícil de asegurar que este sea el valor real. Dado que en el resto del trabajo se verificó que con un valor de 0.05 segundos como máximo, el modelo coincide lo suficientemente bien con los experimentos de campo, es posible corregir esta aproximación a un valor dentro de este rango. Más aún, dentro de este rango es posible eliminar de la ecuación la constante de tiempo y obtener el mismo resultado. Entonces resulta válido suponer que el actuador responde instantáneamente a la señal de control, o por lo menos, mucho más rápido que el resto de las dinámicas involucradas. De este modo es posible descartar su efecto y en lo que resta del trabajo se lo considerará nulo. Esta conclusión es coincidente con resultados de otros trabajos de control con quadrotores que han demostrado que es posible no considerar este tiempo en el modelo del actuador [13] [14].

2.3.10. Modelo completo del actuador

Finalmente es posible encontrar un modelo completo y lineal del actuador expresado como una función de transferencia en el espacio de la transformada de Laplace definido por:

$$\begin{cases} H_T(s)|_{y_{bias}} = \frac{T(s)}{y(s)} = K_{T\omega} \\ H_M(s)|_{y_{bias}} = \frac{M(s)}{y(s)} = K_{M\omega} \end{cases} \quad (2.15)$$

donde

- $K_{T\omega} = 0,0667 \frac{N}{\%}$
- $K_{M\omega} = 0,0346 \frac{Nm}{\%}$

2.4. Matriz del quadrotor: \bar{M}_Q

Como se vio en la sección anterior, el torque y momento producido por los actuadores son proporcionales a las entradas de control y por ende es posible resumir la matriz \bar{M}_1 utilizada hasta ahora junto con el modelo lineal de los actuadores en una nueva matriz \bar{M}_Q tal que:

$$\begin{bmatrix} F_z \\ L \\ M \\ N \end{bmatrix} = \bar{M}_Q \cdot \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} + \begin{bmatrix} mg \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (2.16)$$

donde

$$\bar{M}_Q = \begin{bmatrix} -K_{T_w} & -K_{T_w} & -K_{T_w} & -K_{T_w} \\ -dK_{T_w} & -dK_{T_w} & dK_{T_w} & dK_{T_w} \\ -dK_{T_w} & dK_{T_w} & dK_{T_w} & -dK_{T_w} \\ -K_{M_w} & K_{M_w} & -K_{M_w} & K_{M_w} \end{bmatrix} \quad (2.17)$$

Para que esta simplificación sea válida, la constante de tiempo de los motores debe ser considerada nula. Esta nueva matriz que será utilizada de ahora en más en vez de las matrices \bar{M}_1 y \bar{M}_Q , permite determinar de manera aproximada si el quadrotor va a poder alcanzar la condición de hovering o no y cuáles serán las entradas de control de cada uno de los actuadores. Se debe entonces resolver la ecuación:

$$\begin{bmatrix} F_z \\ L \\ M \\ N \end{bmatrix} = \begin{bmatrix} -mg \\ 0 \\ 0 \\ 0 \end{bmatrix} = \bar{M}_Q \cdot \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \quad (2.18)$$

que tiene solución si y solo si el determinante de \bar{M}_Q es distinto que cero.

2.5. Matriz de control: M_{MIX}

Si se determinan las entradas y_i de cada uno de los actuadores, se puede entonces propagar esta entrada hasta las fuerzas y momentos sobre el centro de masa de la aeronave que finalmente describirán la evolución de la misma. Pero, como se observa en el diagrama de bloques de la figura 1.2 se busca diseñar un sistema de cuatro entradas, que no son directamente la entrada de control de cada actuador sino las que controlen cada uno de los cuatro grados de libertad de la aeronave. El enfoque que se propone no es el único posible, pero resulta ser muy sencillo de comprender. Los cuatro grados de libertad del quadrotor son: “roll”, “pitch”, “yaw” y el empuje en dirección $-\hat{z}$ denominado colectivo. Para poder vincular estas entradas con las señales de control de cada uno de los actuadores se propone un enfoque matricial tal que:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \bar{M}_{Mix} \cdot \begin{bmatrix} Z_c \\ \phi_c \\ \theta_c \\ \psi_c \end{bmatrix} \quad (2.19)$$

Si se observa el diagrama de bloques de la figura 1.2 la matriz \bar{M}_{Mix} deberá tener una forma

similar a la inversa de la matriz 2.16, es decir:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} K_z & -K_\phi & -K_\theta & -K_\psi \\ K_z & -K_\phi & K_\theta & K_\psi \\ K_z & K_\phi & K_\theta & -K_\psi \\ K_z & K_\phi & -K_\theta & K_\psi \end{bmatrix} \cdot \begin{bmatrix} Z_c \\ \phi_c \\ \theta_c \\ \psi_c \end{bmatrix} \quad (2.20)$$

donde K_z , K_ϕ , K_θ , K_ψ son ganancias que se deberán elegir de manera adecuada para poder hacer la mezcla de los controles. Estas constantes son un parámetro de diseño en el sistema de control y se las puede pensar como la sensibilidad o agresividad de cada uno de los comandos de la aeronave que también proporcionan un ajuste de unidades para el control. Se eligió usar entradas de control entre -1.0 y 1.0 para los controles angulares y entre 0.0 y 1.0 para el control colectivo.

La selección de las ganancias de cada control es sin duda un punto clave en el diseño del sistema de control. Si por ejemplo suponemos que $K_z = 100\%$ y la entrada de control $Z_c = 0.7$, se tiene, entonces, que la componente de control colectivo es 70 %. Si suponemos que un controlador de velocidad angular esta actuando sobre el control de roll, y $K_\phi = 50$, puede darse el caso en el cual la salida de este controlador esté cercana a la saturación por ejemplo 0.9. En ese caso las entradas de control a cada uno de los actuadores quedarían:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} 25\% \\ 25\% \\ 115\% \\ 115\% \end{bmatrix} \quad (2.21)$$

lo cual claramente satura la entrada de control. Si bien esto no es algo inaceptable, la acción de control diferencial ya no es simétrica y el sistema se vuelve alineal. Se proponen entonces diferentes alternativas para solucionar este problema:

- Elegir las ganancias K_z , K_ϕ , K_θ , K_ψ de forma tal que la suma de las mismas (con signo según la ecuación 2.20) nunca supere el 100 %.
- Elegir las ganancias para obtener la mayor reacción de control y, en consecuencia, tener que lidiar con las alinealidades en el empuje y en el modelo del sistema.

En este trabajo se optó por la solución conservadora ya que el resto estaba fuera del alcance del mismo. Se obtuvieron resultados favorables tanto en las simulaciones como en los experimentos sobre la aeronave real con las siguientes ganancias:

Ganancia	Valor
K_z	70 %
K_ϕ	20 %
K_θ	20 %
K_ψ	30 %

Cuadro 2.7: Ganancias de los controladores actuadores

Como se verá más adelante, estas decisiones influyen directamente en los resultados que se podrán obtener del control. Dado que estas constantes controlan directamente la cantidad de energía que se destinará a cada una de las cuatro acciones de control, dependiendo de la aplicación será posible reajustarlas para lograr diferentes objetivos. Por supuesto esto también

dependerá de los actuadores elegidos ya que tener un actuador más potente permite reducir el valor de K_z y aumentar el del resto de los controles.

2.6. Parámetros constantes de las ecuaciones

La terna fija en la aeronave se eligió de forma tal de coincidir con los ejes principales de inercia logrando así un tensor de inercia I_b diagonal cuyos productos de inercia se vuelven nulos. Se utilizó un diseño asistido por computadora (CAD) para obtener los tensores de inercia y la posición del centro de gravedad. Para hacer el cálculo de las propiedades físicas, como ser momento de inercia, masa y centro de gravedad, se utilizaron mediciones reales de masa de las diferentes piezas del quadrotor para luego integrar al diseño CAD. En el cuadro 2.8 se muestran los principales componentes que componen el modelo real del quadrotor y sus respectivos pesos. En este solo se consideran aquellas piezas significativas en masa que fueron consideradas para el cálculo de las propiedades físicas. No fueron considerados tornillos, tuercas, adhesivos ni cables.

Cant.	Pieza	Descripción	Peso[gr]	Subtotal[gr]
4	Brazos	Nylon 6/6	54	216
4	ESC	30A Opto 400Hz	24	96
4	Motor	2212/920 Kv BLDC	56	225
1	Base inferior		42	42
1	Base superior		27	27
1	Bateria	3S1P 4500mAh 25/C	339	339
4	Hélices	12x4.5	12	48
1	FLC V2	Computadora de vuelo	44	44
Peso total [gr]:				1037 ± 10

Cuadro 2.8: Presupuesto de masa del quadrotor

En la figura 2.20 se puede observar la ubicación del centro de gravedad de la aeronave calculado automáticamente con el software de simulación a partir del modelo 3D.

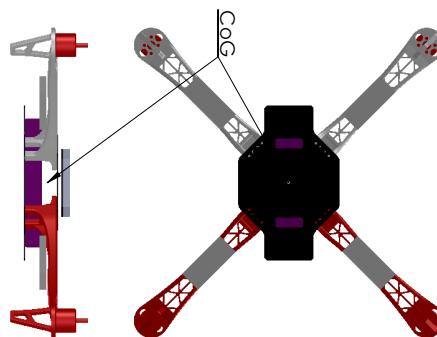


Figura 2.20: Perspectiva del modelo CAD del quadrotor. El bloque gris en la parte superior representa un modelo exterior de la computadora de vuelo

Tomado desde el centro de gravedad en la terna fija a la aeronave y orientado como se observa en las figuras previas, se obtiene el tensor de inercia diagonal:

$$I = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} = \begin{bmatrix} 9400 & 0 & 0 \\ 0 & 10000 & 0 \\ 0 & 0 & 18700 \end{bmatrix} \cdot 10^{-6} \cdot \text{Kg} \cdot \text{m}^2 \quad (2.22)$$

siendo la masa total del modelo:

$$m = 1037 \text{gr} \quad (2.23)$$

Para validar el tensor de inercia obtenido mediante el software CAD se realizó un experimento de pendulo bifilar. Este consiste en suspender de dos hilos paralelos suficientemente largos el objeto, de forma tal de medir su inercia en el eje perpendicular al plano de oscilación como se puede observar en la figura 2.21.

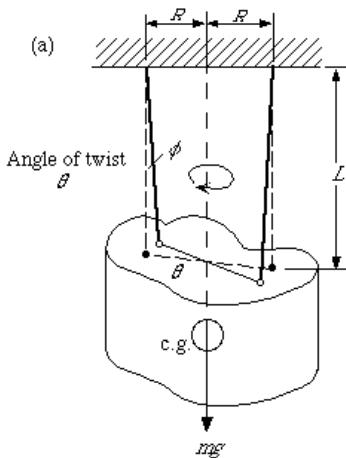


Figura 2.21: Experimento del pendulo bifilar. Imagen propiedad del departamento de ingeniería mecánica de la universidad de Texas

En este experimento, se hace oscilar al objeto bajo estudio a partir de una perturbación pequeña y se mide el tiempo de oscilación. En este trabajo se utilizó el acelerómetro incorporado en la computadora de abordo para determinar este periodo. Como en este caso lo único que se busca es medir tiempo, no resulta relevante la posición del sensor ni su orientación respecto al objeto a medir. Se realizó el experimento para cada uno de los ejes principales y se utilizó la formula:

$$I = \left(\frac{T}{2\pi} \right)^2 \frac{mgR^2}{L} \quad (2.24)$$

para calcular el momento en cada eje. Esto solo se realizó para los ejes principales y el resto de los productos de inercia se los consideró cero. Los resultados obtenidos mediante este experimento fueron:

$$I_{exp} = \begin{bmatrix} 0,0074 & 0 & 0 \\ 0 & 0,0085 & 0 \\ 0 & 0 & 0,0209 \end{bmatrix} \quad (2.25)$$

Se puede observar que la diferencia entre cada eje es del orden del 11 % salvo en el caso del momento I_{xx} que fue de aproximadamente el 27 %. Aun con estas diferencias, este es un resultado aceptable ya que el modelo real no presenta la alineación perfecta del CAD y la batería y sus cables proporcionan fuentes de error considerables. Resulta también interesante notar que si bien

los resultados fueron ligeramente diferentes, la relación $I_{ZZ} > I_{YY} > I_{XX}$ se cumple en ambos casos y dada la forma de la estructura, eso debe ser así. Como se verá en el capítulo de 5, el sistema es suficientemente robusto a cambios en el tensor de inercia pequeños como para que el error cometido en este cálculo sea despreciable.

2.7. Perturbaciones y efectos de segundo orden

Ciertas hipótesis fueron tomadas en cuenta a lo largo del desarrollo para simplificar el modelo del sistema. Estas fueron:

1. La estructura es perfectamente rígida.
2. La estructura es perfectamente simétrica.
3. No se consideraran efectos giroscópicos de los motores y hélices debido a su pequeña inercia en comparación con la aeronave.
4. El empuje y torque generado por las hélices es siempre en la dirección $-\hat{z}$.
5. El quadrotor se encuentra en hovering.
6. No existen fuerzas aerodinámica que provoquen fuerzas y/o momentos sobre el sistema.

La hipótesis número 5 es la base de este trabajo y es, sin dudas, la más difícil de cumplir. Para que el quadrotor se encuentre en hovering, este debe despegar del piso y mantenerse a una altura considerable, esto en la práctica no sucede, pero las alinealidades y efectos secundarios cuando esto no se cumple son fácilmente evitables por un piloto con poca experiencia. Es por eso que se considerará valida la hipótesis de hovering para el resto del trabajo. Por otra parte resulta curioso analizar el hecho de que para que la hipótesis de hovering se cumpla, el controlador debe poder funcionar y este hecho se cumple si estas hipótesis son válidas. En realidad, lo que ocurre es que si bien durante el aterrizaje y el despegue las hipótesis no se cumplen, el sistema se encuentra cerca de la condición de hovering y es posible lograr el vuelo con pese a esto.

Por otro lado, las hipótesis 1 y 2 seguirán suponiéndose válidas ya que el análisis de estas se encuentra fuera del alcance de este trabajo. Pero, en cambio, resulta interesante dedicar unas palabras para las hipótesis 3, 4 y 6.

2.7.1. Blade flapping

Si bien existe una gran cantidad de efectos aerodinámicos en cualquier aeronave con rotores, la mayoría de estos sólo influyen en aeronaves de gran tamaño y por eso no se estudian en detalle. Pero el caso del “blade flapping” es importante estudiarlo. Existen otros trabajos [15], en donde estos efectos fueron considerados en el modelo, pero dado que en este trabajo solo se busca el hovering, se demostrará que no es necesario incluirlos.

Las hélices utilizadas en este tipo de aeronave son en general plásticas y flexibles. Esta flexibilidad hace que las mismas sufran modificaciones en su forma durante el vuelo y si bien esto es un efecto deseado, ya que ayuda a reducir el impacto de las fuerzas aerodinámicas sobre los motores; también pueden ser una fuente de fuerzas y momentos no deseados.

Cuando las hélices se trasladan horizontalmente por el aire, el aspa que se encuentra avanzando en el mismo sentido que la aeronave siente por el viento aparente que girá más rápido que el aspa del lado opuesto. Esto produce que un lado de la hélice genere ligeramente más sustención que el otro y por ende incline el plano de rotación de la hélice del eje $-\hat{z}$. Esto produce que el

empuje no sea sólo en esta dirección, violando la hipótesis 4. Es posible observar este efecto en la figura 2.22.

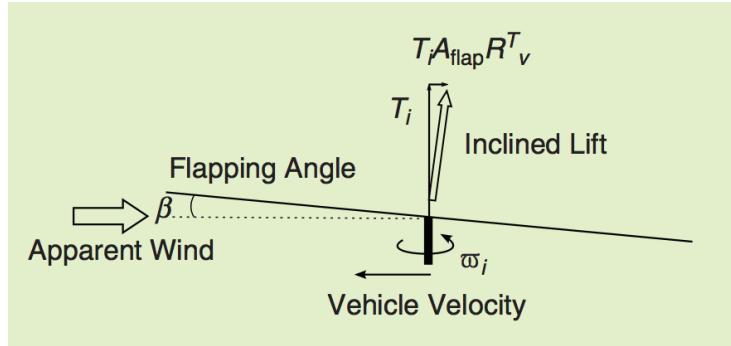


Figura 2.22: Efecto del flapping de las hélices. Imagen propiedad de Robotics and Automation Magazine - IEEE

Dado que solo se busca mantener en condición de hovering a la aeronave, este efecto no resulta de interés y por ende no se lo considera en el modelo del mismo. Es importante descartar, que de buscar un modelo para translaciones a altas velocidades, este efecto juega un rol fundamental en el modelado.

2.7.2. Efectos giroscópicos

La hipótesis de considerar un cuerpo rígido no resulta estrictamente válida si se tiene en cuenta que en su interior existen cuatro rotores girando a altas velocidades. Debido a la conservación del momento angular, estos rotores generan un torque en la dirección opuesta a la de rotación que le impone la aeronave para intentar conservar su momento angular. Esto es conocido en inglés como “gyroscopic stiffness”. Si bien esto puede parecer de gran ayuda para mantener al quadrotor en hovering, ya que los cuatro rotores intentarán quedarse en el mismo plano de rotación ante perturbaciones o cambios requeridos por las referencias impuestas por el piloto, esto puede provocar momentos en distintos ejes que, de no considerarse, pueden desestabilizar la aeronave. Dado que las hélices utilizadas en este tipo de aeronaves en general tienen un momento de inercia muy pequeño, inclusive a altas velocidades estas no generan suficiente torque para ser considerados los efectos giroscópicos en el modelo [16].

2.7.3. Efectos aerodinámicos

Los efectos aerodinámicos son aquellos que, dada la geometría del quadrotor y el flujo de aire circundante, generan fuerzas y momentos sobre el quadrotor. Pueden ser considerados como perturbaciones o bien como parámetros del medio que se pueden tener en cuenta en el modelado del sistema. Por ejemplo si el quadrotor es capaz de girar en el eje de yaw por el torque generado por los motores, de no existir estos efectos, entonces sería posible que el quadrotor acelere en este eje indefinidamente y si no se aplica momento alguno, entonces, este continuara girando indefinidamente. Esta situación no es válida dado que el aire ejerce una fuerza de rozamiento que hace que el sistema pierda energía y se tenga, por ejemplo, una velocidad terminal de giro, o que eventualmente esta se reduzca. Pero dado que en las condiciones de hovering el quadrotor no se traslada ni rota en cantidades significativas, estos efectos son perfectamente despreciables. En el capítulo 6 se verá que esto no afecta al modelo ni el desempeño de la aeronave real.

2.7.4. Efecto suelo

Otro efecto interesante a tener en cuenta es el llamado “efecto suelo”, a veces conocido como “colchón de aire”. Este ocurre cuando una aeronave con un rotor está a una distancia del piso menor a un diámetro de su rotor y se debe a un aumento de presión en la parte inferior de la aeronave debido a que el flujo de aire que sale del rotor se ve frenado por el piso [17]. Este efecto hace que el quadrotor “rebote” del piso casi sin tocarlo con bastante facilidad y también ayuda a mejorar la estabilidad ya que en el caso de que este se incline hacia algún lado, este lado va a estar más cerca del piso. Como el aumento de presión aumenta cuando disminuye la distancia al piso, entonces este lado sufre un empuje aumentado de este mismo lado que actúa para corregir la inclinación. Si bien esto es favorable, este efecto hace que el modelo matemático simplificado que se utiliza en este trabajo sea invalido ya que aparecen fuerzas externas que no son consideradas y la dinámica del sistema cambia. Es por eso importante destacar que la condición de hovering propuesta también exige que la aeronave se encuentre a por lo menos un diámetro del rotor (9 pulgadas) del piso para no sufrir estos fenómenos. En el caso de querer trabajar en pilotos automáticos para aterrizaje o despegue autónomo, o bien vuelo a muy baja altura, estos efectos deberán ser tenidos en cuenta. En la figura 2.23 se puede observar como la cercanía con el suelo interrumpe el flujo de aire generado por el rotor de un helicóptero tradicional.

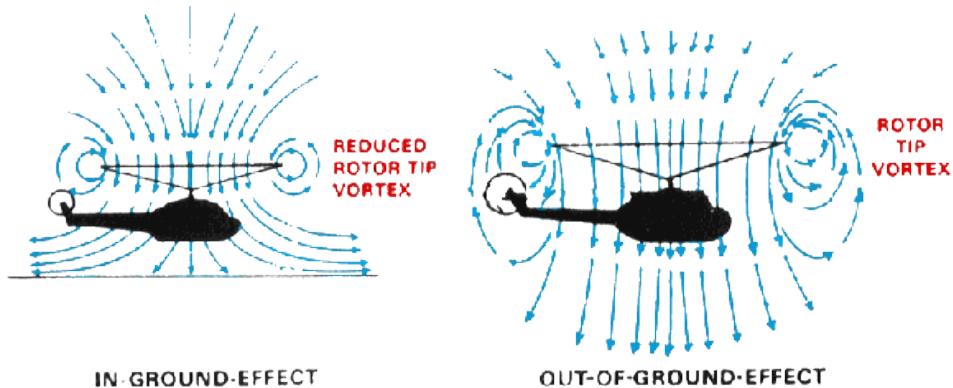


Figura 2.23: Representación gráfica del efecto suelo

2.8. Separación por ejes

Dadas las condiciones del problema y las linealizaciones utilizadas, se concluye que es posible separar cada eje o grado de libertad de manera independiente y desacoplado del resto para su análisis. Por ejemplo, si tomamos el eje de “roll” se tiene que de la ecuación 2.16 para el eje \hat{x} :

$$L = dK_{T_w} (-y_1 - y_2 + -y_3 + y_4) \quad (2.26)$$

Si ahora se utiliza la ecuación 2.20 para encontrar los valores de y_i solo con $\phi_c \neq 0$ se obtiene que:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} -K_\phi \phi_c \\ -K_\phi \phi_c \\ K_\phi \phi_c \\ K_\phi \phi_c \end{bmatrix} \quad (2.27)$$

que se puede simplificar como:

$$L = 4dK_{T_w}K_\phi\phi_c \quad (2.28)$$

Con ϕ_c entre -1 y 1. Si se agrega ahora el efecto de la función de transferencia para el actuador obtenido en 2.6, se obtiene entonces el modelo final para el eje de “roll” según la siguiente ecuación:

$$H_{roll}(s) = \frac{p(s)}{\phi_c(s)} = 4dK_\phi \frac{K_{T\omega}}{sI_{xx}} \quad (2.29)$$

Y análogamente resulta para el eje de pitch:

$$H_{pitch}(s) = \frac{q(s)}{\theta_c(s)} = 4dK_\theta \frac{K_{T\omega}}{sI_{yy}} \quad (2.30)$$

y para el eje de yaw:

$$H_{yaw}(s) = \frac{r(s)}{\psi_c(s)} = 4K_\psi \frac{K_{M\omega}}{sI_{zz}} \quad (2.31)$$

Se obtienen así las transferencias de cada uno de los ejes por separado, que permitirá realizar un control independiente para cada uno, siempre que las hipótesis que permitieron la separación se cumplan.

2.9. Resumen del capítulo

En este capítulo se hizo un estudio detallado del modelo del quadrotor, que permitió obtener, bajo ciertas condiciones, un modelo simple de cada eje que puede ser utilizado independientemente de los otros. Esta solución permite resolver de manera sencilla el diseño del sistema de control permitiendo trabajar a cada eje como un problema independiente. Mejor aún, la planta para cada eje resultó ser un sistema de primer orden que es ampliamente estudiado en la literatura de control. Si bien esto requiere que se cumplan las condiciones de hovering para ser válido, se demostrará que también es posible utilizar este modelo para el control de ángulo pequeño y bajas velocidades en el que los efectos aerodinámicos no son significativos. De querer realizar pruebas con ángulos grandes o velocidades de rotación y/o translación considerable, entonces este enfoque desacoplado dejará de ser válido y se deberá utilizar un modelo diferente.

CAPÍTULO 3

Controlador PID

El control PID es hoy en día utilizado a nivel industrial en el 95 % de los sistemas de control de procesos [18]. En un principio, se empleó para el guiado automático de barcos, aunque su formalización matemática no se conoció hasta 1922 [19]. La idea básica del controlador PID es proporcionar una acción de control proporcional al error, a su integral y también a su derivada mediante la realimentación negativa. Si bien su implementación original fue mediante sistemas hidráulicos y neumáticos, en la actualidad los mismos son implementados casi exclusivamente de forma electrónica, y más aún de forma digital, es decir que estos controladores son implementados por microcontroladores, PLC's o sistemas distribuidos. En este capítulo se realizará un análisis detallado de este controlador y se terminará con su implementación digital capaz de ser ejecutada en la computadora de vuelo del quadrotor.

3.1. ARQUITECTURAS DE IMPLEMENTACIÓN DE CONTROLADORES PID

La ecuación que determina la acción de control realizada por un controlador tipo PID continuo es en general:

$$u(t) = K \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de(t)}{dt} \right] \quad (3.1)$$

Donde los diferentes términos son:

- $u(t)$: Salida del controlador
- $e(t)$: Error entre la variable de proceso y el punto de trabajo deseado, $y(t) - y_{sp}(t)$
- K, T_i, T_d : Constantes del controlador

y en el dominio de la transformada de Laplace se la puede expresar como:

$$u(s) = K \left[[e(s) + \frac{1}{sT_i} e(s) + sT_d * e(s)] \right] \quad (3.2)$$

Partiendo del diagrama básico del lazo de control de la figura 1.8 es posible reemplazar al bloque del controlador por la implementación particular del PID de la ecuación 3.2 mediante el diagrama en bloques de la figura 3.1.

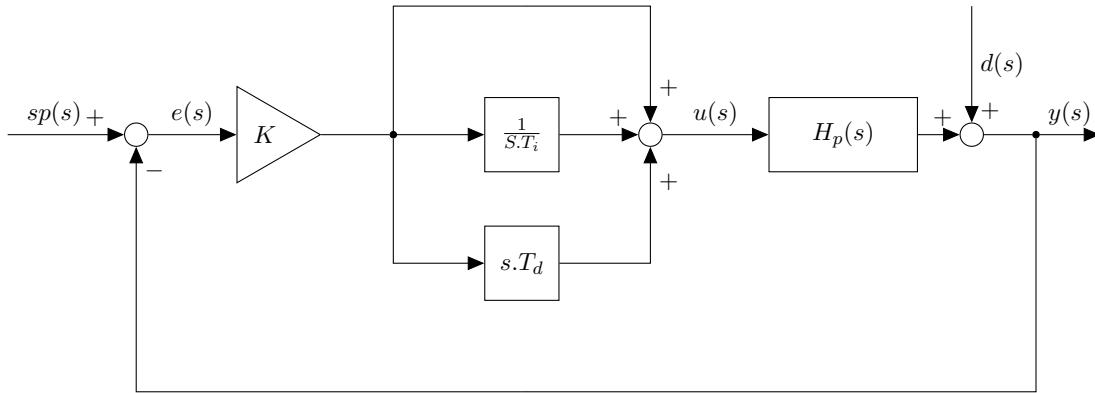


Figura 3.1: Controlador PID

El algoritmo así descripto presenta algunos inconvenientes para su implementación [20] y es posible, mediante algunas modificaciones, mejorarla significativamente. Por ejemplo, ante la presencia de ruido en la variable de proceso, $y(s)$, el término derivador puede comportarse de forma no deseada. También pueden darse condiciones donde la salida del controlador, $u(s)$, exceda los límites del actuador. En las siguientes subsecciones se detallarán las mejoras y modificaciones que se le puede realizar al mismo.

3.1. Arquitecturas de implementación de controladores PID

Existen por lo menos tres diferentes arquitecturas para la implementación de un controlador PID [18] conocidas como: “Standard non Interacting (ISA)”, “Series - Interacting” y “Parallel”. En general todas son utilizadas en la literatura de manera indistinta, pero en lo que a la implementación práctica se refiere, el más utilizado es el tipo Series [18] dado que en la época cuando los controladores eran implementados mediante máquinas neumáticas, este era el más sencillo de realizar. A continuación se detallan brevemente las diferencias entre las distintas arquitecturas.

Standard, non interacting

La transferencia ($G(s) = u(s)/e(s)$) para este controlador surge directamente de la ecuación 3.2 y es:

$$G(s) = K \left(1 + \frac{1}{sT_i} + sT_d \right) \quad (3.3)$$

Series, interacting

En esta arquitectura se puede ver que la constante integral T_i interactúa con la constante T_d . Se suele decir que esta arquitectura es más fácil de ajustar manualmente y tiene su origen en los controladores pneumáticos.

$$G(s) = K' \left(1 + \frac{1}{sT'_i} \right) (1 + sT'_d) \quad (3.4)$$

Parallel

Esta arquitectura es la más sencilla y permite tener controladores que solo tengan un término proporcional, integral y/o derivativo. Esto es imposible de lograr en las otras arquitecturas.

$$G(s) = K_p + \frac{Ki}{s} + sKd \quad (3.5)$$

Como se puede observar, el diagrama de bloques de la figura 3.1 se corresponde con la arquitectura “Standard”. Y si bien estos controladores se pueden relacionar con sus constantes convirtiendo cada uno en el otro, resulta de extrema importancia determinar qué arquitectura se utiliza ya que las constantes de ajuste de una a otra deben ser modificadas.

Para este trabajo se optó por utilizar la arquitectura Standard que se corresponde con el diagrama en bloques de la figura 3.1 y por la ecuación 3.3. Esto se debe a que suele ser la más comúnmente encontrada en la literatura de control. En las siguientes secciones se detallarán las modificaciones introducidas en la ecuación 3.3 para mejorar el desempeño del controlador así también como lo necesario para implementarla en un controlador digital.

3.2. Mejoras en el algoritmo

Con el uso de estos controladores a lo largo de los últimos años, se han realizado modificaciones en el algoritmo original que permitieron adaptarlo de una manera más eficiente a condiciones de uso real. En esta sección se presentan algunas de estas mejoras que marcaron la diferencia a lo largo del tiempo y que fueron introducidas a nivel industrial por la mayoría de los fabricantes [18].

3.2.1. Setpoint ponderado

Cuando se habla de realimentación de error se entiende que todos las partes del controlador (P, I y D por ejemplo) reciben la misma señal de error. Pero en un caso más general se podría tener una modificación de la ecuación 3.1:

$$u(t) = K \left[e_p(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d \frac{de_d(t)}{dt} \right] \quad (3.6)$$

que a diferencia de la ecuación original, la señal de error $e(t)$ se encuentra dividida en:

$$e_p = b \cdot y_{sp} - y \quad (3.7)$$

$$e_d = c \cdot y_{sp} - y$$

$$e = y_{sp} - y$$

(3.8)

El efecto de las constantes b y c afectan directamente el modo en el que el sistema responde para un cambio en una perturbación, un cambio de setpoint o ruidos en la señal de entrada.

Como se explica en ??, los controladores comerciales suelen usar $b = 1$ y $c = 0$ en sus algoritmos de control. El efecto de estas constantes se verá en las próximas secciones, pero es importante destacar que cada término del controlador podrá tener diferentes caminos de realimentación, a diferencia de lo que se presentó en el diagrama preliminar de la figura 3.1.

3.2.2. Derivación de la variable de proceso

Una mejora que se puede aplicar en la acción derivativa surge directamente de la idea del Setpoint ponderado. En esta, se utiliza una señal de error diferente para cada término del controlador PID. Para el caso del término derivativo se tiene de la ecuación 3.6 que:

$$e_d = c \cdot y_{sp} - y \quad (3.9)$$

Donde:

- y = Salida de la planta o Variable de proceso (PV)
- y_{sp} = Salida deseada o SetPoint

Como se ve en la ecuación 3.1 el término derivador toma en cuenta la derivada del error. Es decir:

$$\frac{d(e_d)}{dt} = \frac{d(c \cdot y_{sp} - y)}{dt} = c \frac{d(y_{sp})}{dt} - \frac{d(y)}{dt} \quad (3.10)$$

Si el Setpoint se mantiene constante, entonces $e_d = -\frac{d(y)}{dt}$. Pero si el setpoint cambia, por ejemplo como un escalón, entonces a la salida del controlador se produce una acción de control que teóricamente debería ser una delta de Dirac, pero que está limitada por la dinámica y la respuesta del sistema. Más allá de que esta salida no sea una delta de Dirac, la misma puede tomar valores muy elevados y entonces afectar el desempeño de la planta, inclusive dañándola. Este comportamiento es conocido como "Derivative Kick" [21] y es por este motivo que los controladores comerciales traen incorporado $c = 0$. A esta arquitectura a veces se la suele llamar PI-D por que el término derivador tiene otro camino de realimentación. Este cambio se puede observar en el diagrama de bloques de la figura 3.2. En el caso de utilizar $c = 1$, el diagrama en bloques del sistema es el mismo que el de la figura 3.1.

El efecto de la constante $c = 0$, conocido como "derivada de la variable de proceso", o $c = 1$ conocido como "derivada del error" se puede ver en la simulación de la figura 3.3. En esta simulación la constante c es cambiada de $c = 0$ a $c = 1$ en $t = 100s$. Se observa que la acción de control del segundo gráfico presenta picos cuando la señal de referencia cambia rápidamente y luego, al cambiar el camino de la acción derivativa, esto se ve disminuido notablemente.

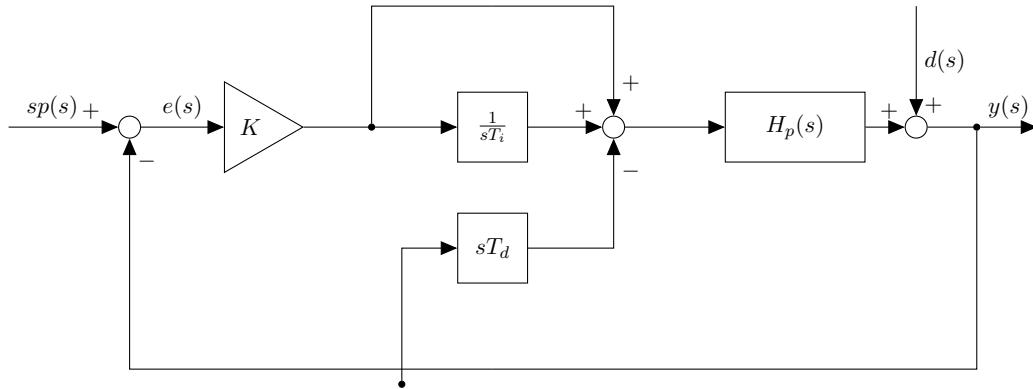


Figura 3.2: Diagrama de bloques del controlador con $c = 0,0$

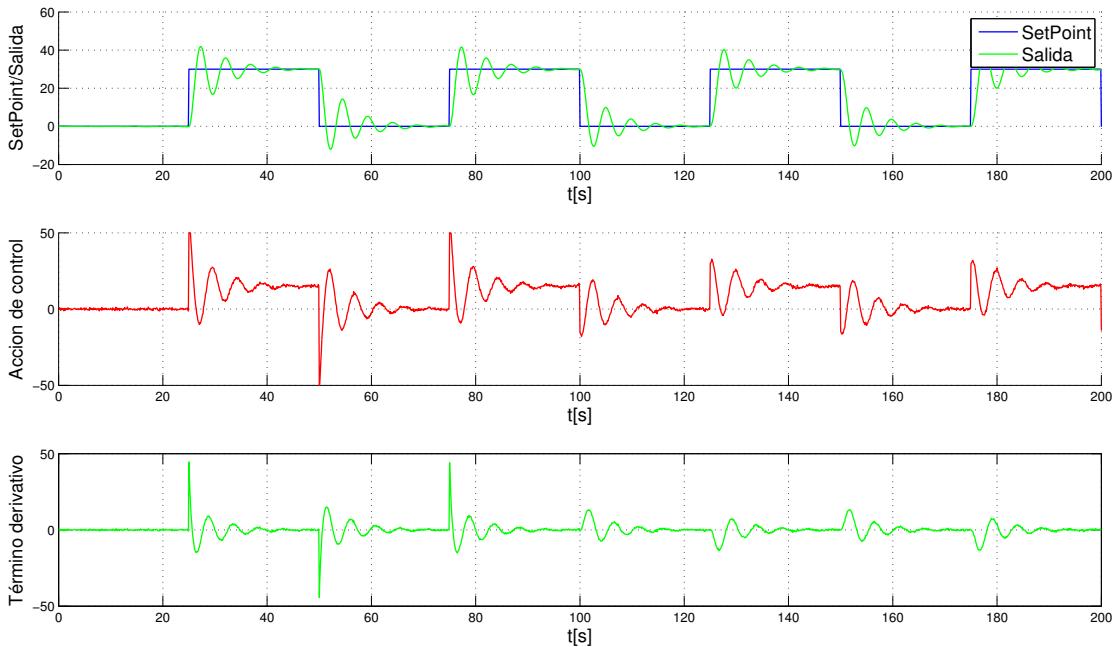


Figura 3.3: Respuesta al cambiar el setpoint con $c = 1$ y $c = 0$

3.2.3. Ganancia limitada

La señal que ingresa al controlador se toma directamente de la salida de la planta. Esta señal tomada por un sensor suele ser transportada hasta el controlador por algún medio que puede agregar ruido a la señal. Si se piensa en una descomposición en Fourier de este ruido y se trata una frecuencia determinada se tiene que:

$$N = a_0 \cdot \sin(\omega_n \cdot t) \quad (3.11)$$

al derivar esta señal, este mismo componente se convierte en:

$$N = \omega_n a_0 \cdot \cos(\omega_n \cdot t) \quad (3.12)$$

Se observa que para ω_n grandes la potencia de ruido en la variable de proceso aumenta pro-

porcionalmente con la frecuencia por lo que si el ruido presente en la planta es de alta frecuencia, este afectará notablemente el desempeño del controlador ya que será pasado directamente a la salida del mismo.

Para evitar esto se suele utilizar un filtro pasabajos junto con el derivador con un coeficiente N que limite la ganancia del ruido para altas frecuencias, pero que se comporta como un derivador para frecuencias bajas (que son las de operación de la planta). Este filtro se lo suele implementar como se muestra en la ecuación 3.13:

$$H_D(s) = -\frac{KsT_d}{1 + sT_d/N} \quad (3.13)$$

El problema con la ecuación 3.13, comúnmente encontrada en los libros de control, es que la misma considera que:

$$H_D(s) = \frac{CO_D(s)}{PV(s)} \quad (3.14)$$

donde $CO_D(s)$ representa el aporte del término derivativo a la acción de control total generada por el controlador. Es decir que solo contempla el caso de la constante de ponderación $c = 0$. Por eso en este trabajo se modificó esta ecuación tal que:

$$H'_D(s) = \frac{CO_D(s)}{c.Y(s) - PV(s)} \quad (3.15)$$

Y se obtiene entonces la respuesta temporal de la parte derivativa como:

$$CO_D(t) = -\frac{T_d}{N} \cdot \frac{dD}{dt} - KT_d \frac{dy}{dt} + cKT_d \frac{dy_{sp}}{dt} \quad (3.16)$$

De esta manera resulta un algoritmo que incluye los beneficios de utilizar diferentes caminos para la señal de error a la vez que proporciona un filtro en el término derivativo que resulta de suma importancia en la aplicación del algoritmo en entornos reales.

3.2.4. Saturación de la acción integral (Windup)

Hasta aquí se ha considerado que la salida del control puede tomar cualquier valor real, que luego actúa como entrada del actuador. Por ejemplo, una válvula, puede ir de abierta a cerrada (medida en porcentaje, por ejemplo) pero no más que eso, así como un motor tiene una máxima tensión posible de operación que es la provista por la fuente de alimentación. Si este es el caso y el controlador impusiese una acción de control por fuera de los límites de operación del actuador, entonces el actuador estaría saturado y es muy posible que la planta nunca pudiera llegar a la referencia impuesta por el usuario. Si esto ocurre, el error $e(s)$ se mantiene constante y el término integral crece indefinidamente hasta que el sistema vuelve a seguir la referencia. Este efecto se lo conoce en inglés como "Windup".

Si la salida del controlador es mayor que los límites del actuador, este no estará efectuando ninguna acción correctiva y por ende el sistema estará funcionando a lazo abierto. Si el término integrador continuó subiendo y el sistema vuelve a los valores normales de operación, demorará en volver a operar ya que el término integral debe disminuir tanto como creció. Esto también puede ocurrir ante la presencia de perturbaciones.

En el controlador implementado, se optó por indicarle los límites de operación de la salida mediante parámetros de configuración. Es decir, que el controlador tiene información acerca del actuador o la planta. Y se implementó el antiwindup mediante un integrador saturado en estos límites. De esta manera el término integral y la salida del controlador no pueden ir más allá de

lo que el actuador admite como entrada y por ende su acción de control, cuando la planta vuelve a los rangos de operación, es instantánea.

3.2.5. Transición Manual/Automático

En muchas ocasiones resulta útil poder ingresar un valor manual a la planta sin tener que utilizar el controlador. Para esto se suele incluir una entrada manual adicional. En este diseño se utiliza la misma entrada del setpoint como entrada manual y una variable binaria del mismo permite seleccionar si se utiliza en modo manual o automático. En la figura 3.4 se puede observar el diagrama en bloques del sistema donde el controlador representa el algoritmo del PID utilizado.

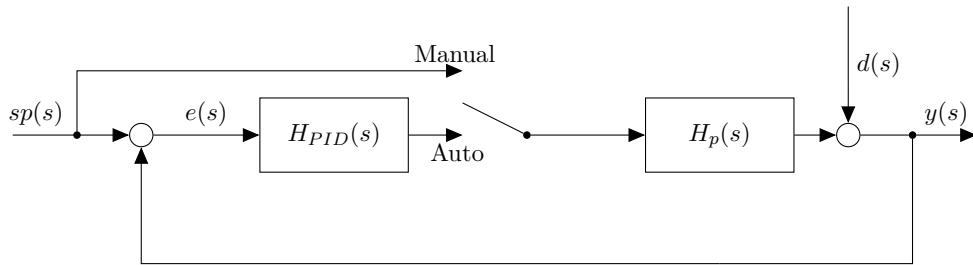


Figura 3.4: Controlador con modo manual y automático

En general, el problema de este sistema es que si se pasa de modo manual a modo automático, la planta sufre una perturbación ya que el controlador impone una acción diferente a la que se imponía en modo manual. Es por eso que se debe implementar un sistema que permita hacer una transición suave, conocida en inglés como "Bumpless". En este trabajo este método se implementó haciendo que en modo manual el controlador actualice el valor del integrador (que funciona como auto bias) con el valor de la operación manual. De esta manera, cuando el sistema pasa a modo automático, la salida permanece constante y no existe discontinuidad. La acción derivativa (al tener memoria) también necesita un sistema similar de seguimiento. El efecto se apreciaría si la derivada de la entrada manual fuese distinta que cero y en ese momento se hiciese la transición a modo automático. Dado que esto pocas veces ocurre durante condiciones de vuelo estable en hovering, no se implementó el sistema de tracking para la acción derivativa.

Finalmente, del diagrama de bloques de la figura 3.4 resulta evidente que es necesario que el controlador tenga conocimiento de la acción de control manual más allá de que no está operando en modo automático ya que en caso contrario, no podría hacer el seguimiento de la misma.

3.3. Implementación digital

El estudio de las secciones anteriores se basó exclusivamente en el controlador PID para variable continua y dado que para este trabajo se busca una implementación en un microcontrolador, es necesario transformarlo a un controlador digital representado como ecuaciones en diferencia. En esta sección se detallan los pasos que fueron realizados donde el controlador que se implementó busca incluir todas las características y mejoras detalladas en las secciones anteriores que se corresponden con la ley de control de la ecuación 3.17:

$$\begin{aligned}
 CO(s) &= K [CO_P(s) + CO_I(s) + CO_D(s)] \\
 &= K[(b.y_{sp}(s) - y(s)) \\
 &\quad + \left(\frac{1}{T_i.s} (y_{sp}(s) - y(s)) \right) \\
 &\quad + \left(\frac{T_d.s}{1 + T_d.s/N} \cdot (c.y_{sp}(s) - y(s)) \right)]
 \end{aligned} \tag{3.17}$$

donde:

- K, T_i, T_d, N, b, c son constantes de ajuste.
- $y_{sp}(s)$ es la entrada de referencia del controlador
- $y(s)$ es la variable de proceso o salida de la planta.

Otros parámetros que se deberán ajustar del controlador, pero que no se reflejan en la ecuación 3.17, son:

- CO_{max} y CO_{min} para poner los límites del controlador e implementar el anti windup.
- Modo: manual o automático con transferencia bumpless.

En la versión digital, la ecuación 3.17 es reemplazada por:

$$CO(k) = P(k) + I(k) + D(k) \tag{3.18}$$

En donde k es el instante en donde se calcula la acción que corresponde a un momento de muestreo. La ganancia proporcional en su versión discreta tiene una implementación directa como:

$$P(k) = K(b.y_{sp}(k) - y(k)) \tag{3.19}$$

Luego la integral discreta se puede calcular como una diferencia en adelanto:

$$I(k+1) = I(k) + \frac{KT_s}{T_i} (y_{sp}(k) - y(k)) \tag{3.20}$$

y por último de la ecuación 3.13 se puede obtener una versión discreta como una diferencia en atraso tal que:

$$\begin{aligned}
 D(k) &= \frac{T_d}{T_d + NT_s} D(k-1) \\
 &\quad - \frac{KT_d N}{T_d + NT_s} (y(k) - y(k-1)) \\
 &\quad + \frac{cKT_d N}{T_d + NT_s} (y_{sp}(k) - y_{sp}(k-1))
 \end{aligned} \tag{3.21}$$

Finalmente las ecuaciones 3.19, 3.20 y 3.21 pueden ser transformadas en un código fuente de computadora que realizará de manera digital el controlador PID.

3.4. Modelo matemático del controlador

Para poder caracterizar y estudiar el controlador diseñado se partió del algoritmo implementado y se llevó el mismo a una función de transferencia en el dominio de la transformada Z . Para ello la única manera de obtener una ecuación de transferencia es utilizando las constantes del controlador $b = 1$ y $c = 1$. Entonces para el siguiente desarrollo se considera:

$$E(z) = y_{sp}(z) - y(z)$$

Y, transformando la ecuación 3.19 se obtiene que:

$$P(z) = K \cdot E(z)$$

De la misma forma con la ecuación 3.20 se obtiene:

$$I(z) = K \cdot \frac{\frac{T_s}{T_i}}{z - 1} E(z)$$

Y por último de la ecuación 3.21 se obtiene:

$$D(z) = \frac{\frac{T_d \cdot K \cdot N}{T_d + N \cdot T_s} (z - 1)}{z - \frac{T_d}{T_d + N \cdot T_s}} E(z)$$

Entonces la transferencia del controlador $H_C(z)$ es:

$$H_C(z) = \frac{CO(Z)}{E(Z)} = K + K \cdot \frac{\frac{T_s}{T_i}}{z - 1} + \frac{\frac{T_d \cdot K \cdot N}{T_d + N \cdot T_s} (z - 1)}{z - \frac{T_d}{T_d + N \cdot T_s}}$$

Desarrollando la ecuación anterior se obtiene que:

$$H_C(z) = \frac{(\alpha \cdot N + 1)z^2 - \left(\alpha(2N + 1) + 1 - \frac{T_s}{T_i} \right) K \cdot z + \alpha \cdot K \left(N + 1 - \frac{T_s}{T_i} \right)}{z^2 - (\alpha + 1)z + \alpha} \quad (3.22)$$

con:

$$\alpha = \frac{T_d}{T_d + N \cdot T_s}$$

La ecuación 3.22 caracteriza por completo al controlador. Se comprobó mediante simulación que la respuesta del controlador digital y de la transferencia $H_C(z)$ responden exactamente igual, pero estos resultados no se incluyen aquí para que el trabajo no resulte excesivamente largo. No obstante, en la sección a continuación se muestra mediante un ejemplo el resultado de la implementación de la ecuación 3.22.

3.5. Validación del controlador PID digital

3.5.1. Planta de prueba

Para validar la implementación del controlador se utilizó una planta de prueba dada por la ecuación 3.23. Como se explicó en el capítulo 2, esta planta es equivalente, o incluso de mayor orden que la dinámica representativa de los ejes desacoplados del quadrotor.

$$H(s) = \frac{2}{2s^2 + 0,5s + 1} \quad (3.23)$$

Se ajustó mediante el método de Ziegler Nichols, los parámetros del controlador a partir de los valores de la ganancia crítica y el período crítico [18]. Los valores hallados para esta planta fueron:

- $K_c = 1.682$
- $T_c = 4.7$ segundos

Luego se ajusto el controlador con los valores:

- $K_p = K_c/1,7$
- $T_i = (T_c/2)$
- $T_d = (T_c/8)$

3.5.2. Respuesta al escalón

La respuesta del sistema ante un escalón de valor arbitrario, como por ejemplo 50, se observa en la figura 3.5.

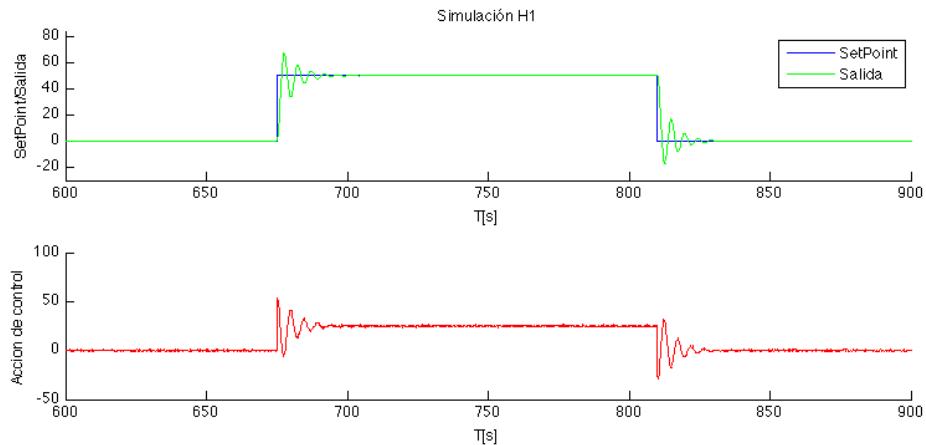


Figura 3.5: Respuesta a un escalón de la planta $H(s)$

Se puede observar que la salida del controlador, $u(t)$, tiene un aspecto ruidoso. Esto se debe a que la entrada al controlador, $y(t)$, proveniente de la planta fue modificada y se le agregó ruido para simular una señal real. Al implementar el término derivador solo con la variable de proceso y sin tomar el SetPoint no se produce el "derivative kick". También se verifica que el sistema sigue perfectamente la referencia introducida.

3.5.3. Transición manual automático

En la siguiente prueba se introdujo una señal cuadrada como entrada. Durante la mitad de la simulación se operó el controlador en modo manual y durante la otra mitad se lo activó en modo automático. Se observa en la figura 3.6 que hay una discontinuidad muy grande en la acción de control al momento del cambio de modo.

Aproximadamente a los 180 segundos de la simulación, se activa el modo automático que trata de llevar la planta a la referencia. Previo a esto, la planta operaba en lazo abierto y su punto de operación no seguía el setpoint sino una entrada manual. A los 180 segundos se observa un brusco pulso en la acción de control. En la figura 3.7 se puede ver la misma simulación pero con la función Bumpless activada. Se observa que la transición es ahora mucho más suave.

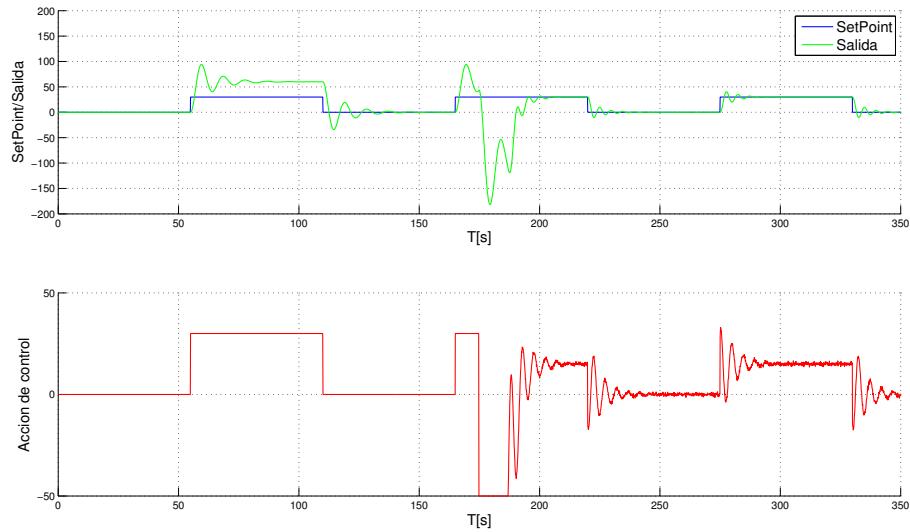


Figura 3.6: Respuesta agresiva al activar el modo automático

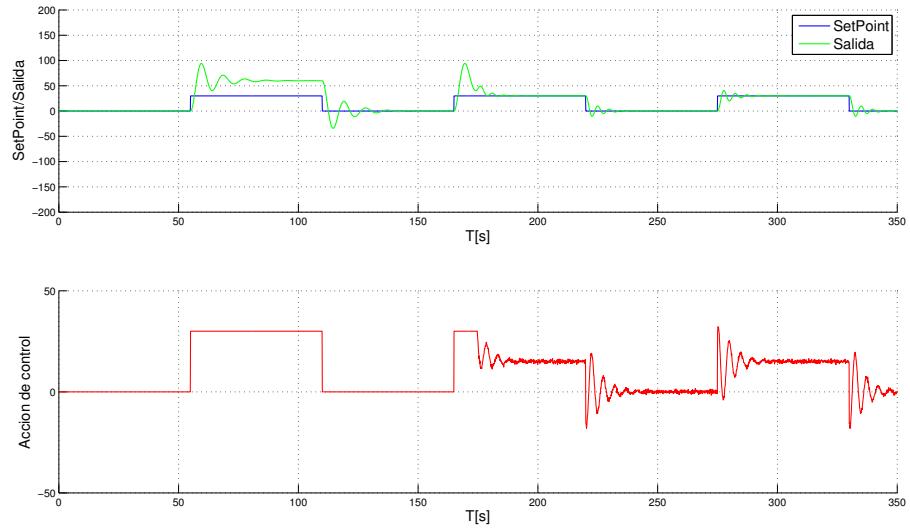


Figura 3.7: Respuesta suave al activar el modo automático

3.5.4. Anti Windup

Si el controlador alimenta a un actuador real, es seguro que el mismo tendrá límites de operación. En la figura 3.8 se muestran los resultados de una simulación en la que se supuso que la salida del controlador alimenta un actuador cuyo máximo de operación de salida es de 10 (en unidades arbitrarias). Se observa en la última subfigura que el término integrador continúa aumentando por que el sistema no llegó al Setpoint. Cuando el sistema vuelve a un estado posible, el integrador debe disminuir su valor y esto lo hace a la velocidad T_i . Entonces se observa una demora en la acción de control muy grande. La línea punteada detalla el momento en que el controlador debería volver a responder. En la figura 3.9 se observa la misma simulación pero en este caso, el controlador tiene conocimiento de los límites máximos y mínimos del actuador (introducidos por el usuario). De esta manera se ve que el integrador satura y por ende su

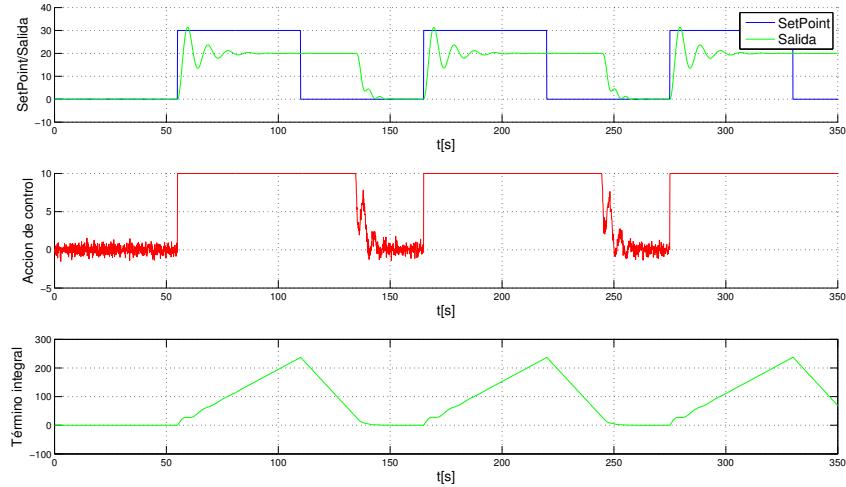


Figura 3.8: Controlador con Anti-windup desactivado

respuesta al volver al modo de operación normal es instantánea.

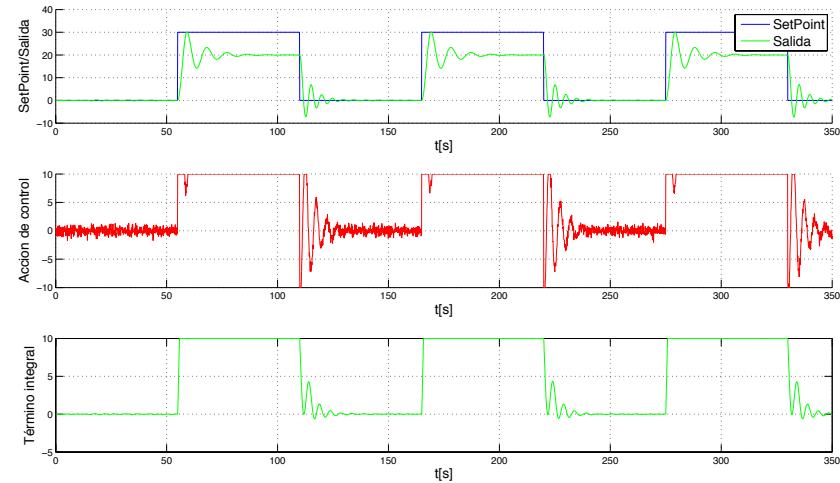


Figura 3.9: Controlador con Anti-windup activado

3.6. Resumen del capítulo

En este capítulo se presentaron los fundamentos del controlador PID. Se explicó que, el diseño original del controlador no es apto para su utilización en problemas de control real en los cuales los actuadores poseen entradas limitadas, las señales de los sensores se encuentran contaminadas de ruido y deben ser implementados de forma digital en microprocesadores. Se mostró ademas que la gran mayoría de estos inconvenientes se resuelven de forma sencilla.

Como trabajo adicional a este capítulo se desarrolló una versión escrita en lenguaje C de este controlador que luego se utilizó en los controladores que se presentarán en el capítulo 5. Esta fue probada y comparada con el controlador PID de Matlab dando los mismos resultados. En <http://github.com/akharsa/qPID> es posible encontrar el código fuente desarrollado, apto para ser utilizado en cualquier circunstancia en la que se requiera este tipo de solución.

CAPÍTULO 4

Plataforma de validación

Durante el desarrollo de este trabajo se diseñaron herramientas que permitieron validar las diferentes etapas del diseño de la computadora de vuelo.

En una primera instancia se diseñó un simulador completo en Simulink en donde se volcaron los datos obtenidos de los modelos de los actuadores y se utilizaron los conceptos de las matrices M_{mix} y M_Q para verificar el diseño del sistema. Esto permitió entender la dinámica del sistema y realizar pruebas valiosas antes de realizar simplificaciones.

Luego se hicieron simulaciones con un modelo simplificado en las cuales se utilizó el modelo de transferencia por eje que se presentó en el capítulo 2. Se utilizaron los controladores digitales que provee Matlab y cuya función de transferencia ya fue correspondida con el controlador escrito en lenguaje “C” introducido en el capítulo 3, para ajustar la dinámica del sistema y obtener así un primer conjunto de parámetros para configurar el sistema de control final que se utilizó en el modelo de vuelo.

Finalmente, se diseñó una computadora de vuelo compacta y capaz de implementar los diferentes controladores para probar su desempeño en condiciones de vuelo real. Esta se utilizó junto a una estructura de quadrotor comercial y se validó la estrategia propuesta de controladores en cascada para el control de un quadrotor en hovering que se presentará en el capítulo 5.

En este capítulo se detallarán los diferentes aspectos de cada una de estas etapas del desarrollo. final.

4.1. Simulador

El primer simulador que se desarrolló como herramienta de estudio fue un simulador con motor de renderización tridimensional para probar de manera manual los conceptos básicos de funcionamiento de un quadrotor. Este simulador se basó en herramientas de Simulink para simulación de cuerpos rígidos y se obtuvo un primer diseño de un simulador no lineal complejo que permitió validar los conceptos básicos del vuelo de un quadrotor. Mediante la utilización de un joystick de computadora se pudieron realizar vuelos simulados a fin de comprender las limitaciones del diseño de un quadrotor. Inclusive, al incluir las disparidades entre los actuadores se pudo observar de manera cualitativa la necesidad de un sistema de control para volarlo. En la figura 4.1 se puede observar una escena del simulador 3D en donde se aprecia al modelo del quadrotor en un entorno virtual.

Este primer simulador incluyó:

- La matriz de mezcla de controles M_{mix}
- La matriz del quadrotor M_Q
- Los modelos de los actuadores obtenidos en los experimentos
- Un simulador de cuerpo rígido de 6 grados de libertad no lineal

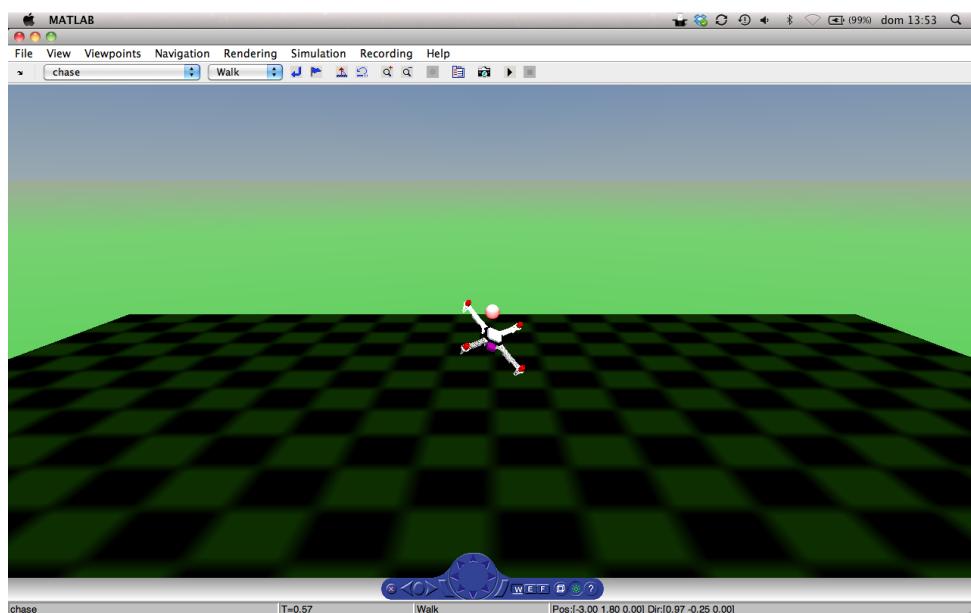


Figura 4.1: Visualización tridimensional del primer simulador

En la figura 4.2 es posible observar un fragmento del simulador que incluye los actuadores y la dinámica del rígido como así también la acción de la fuerza de gravedad. El bloque “Quadrotor forces and moments” se puede ver en detalle en la figura 4.3 donde se puede observar la matriz del quadrotor implementada en forma de diagrama de bloques y en la izquierda el bloque que implementa el modelo del actuador con la saturación y la zona muerta.

Si bien este primer simulador no aportó información cuantitativa para el diseño de los algoritmos de control, sirvió como aproximación al quadrotor como aeronave y permitió estudiar las limitaciones.

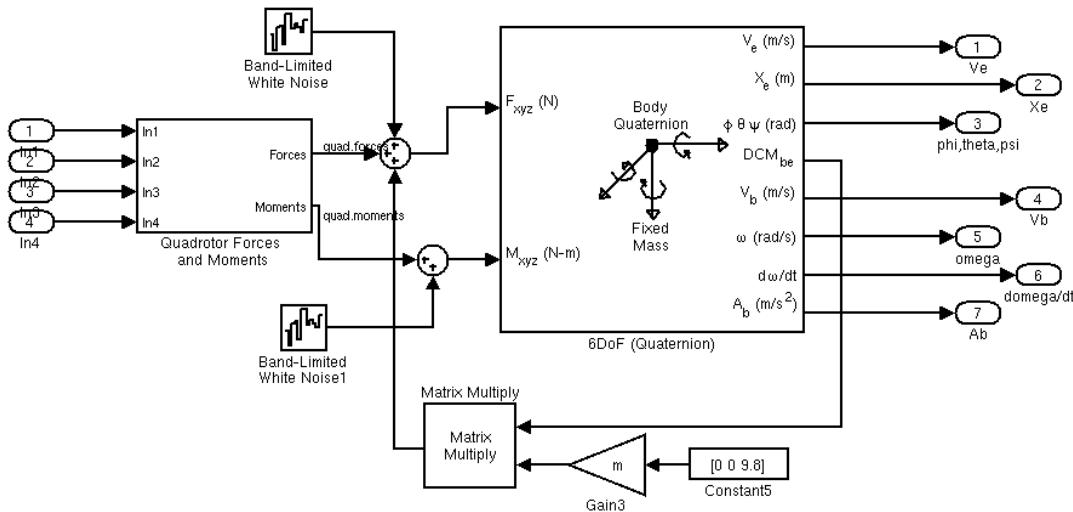


Figura 4.2: Modelo general del simulador implementado en Simulink.

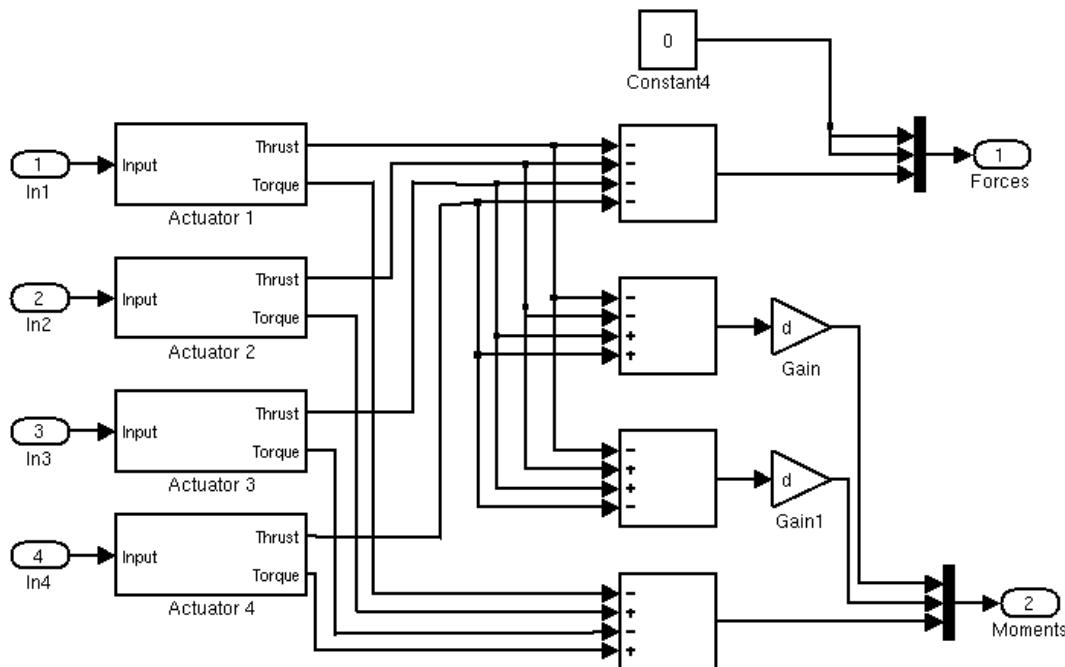


Figura 4.3: Detalle del bloque “Quadrrotor forces and moments”.

Para la implementación de la simulación una vez realizada la linealización y la separación de ejes, se utilizaron las herramientas clásicas de Simulink como funciones de transferencia, controladores y señales. Este simulador mucho más simple permitió utilizar el modelo lineal por eje desarrollado en el capítulo 2 de manera mucho más precisa para ajustar luego los controladores que integran el piloto automático. Por cuestiones de espacio aquí no se ahondará en detalles sobre este modelo de simulación. En el capítulo 5 se presentan estos simuladores, y se sugiere al

lector interesado visitar laboratorios.fi.uba.ar/lse/tesis.html para más información.

4.2. Estructura mecánica y actuadores

La estructura mecánica del quadrotor elegida para pruebas de vuelo fue el modelo F450 de la firma DJI. Como se ve en la figura 4.4, el mismo cuenta con las piezas estructurales, motores, hélices y los variadores de velocidad. Se prefirió la utilización de un modelo comercial operativo para no tener que abocarse en la selección de partes mecánicas, tarea fuera del alcance de este trabajo. Como computadora de abordo se utilizó únicamente la diseñada en este trabajo y no se utilizaron computadoras comerciales durante ninguna etapa del desarrollo. En la tabla 4.1 se presentan las características principales de este modelo.



Figura 4.4: Estructura mecánica y actuadores del quadrotor F450 de DJI-Innovations utilizado.

Modelo	F450
Peso de la estructura	282gr
Diagonal mayor	450mm
Peso de despegue	800gr - 1600gr
Hélices recomendada	8"x4.5"
Batería recomendada	3S, 11.1v
Motor recomendado	22x12mm
Corriente ESC	30A

Cuadro 4.1: Características principales del quadrotor utilizado.

Es importante destacar, que estas hélices, motores y variadores de velocidad fueron los seleccionados para la caracterización del capítulo 2.

4.3. Diseño e implementación de la computadora de vuelo

Una de las partes fundamentales fue el diseño e implementación de una computadora de vuelo para la aeronave. Para ello se buscó diseñar una placa electrónica compacta que alojase todas las funcionalidades necesarias y que pudiese servir como plataforma de validación para este y también para futuros trabajos sobre el tema.

4.3. DISEÑO E IMPLEMENTACIÓN DE LA COMPUTADORA DE VUELO

Dentro de los requerimientos más importantes que se establecieron para la computadora de vuelo se encuentran:

- Micrcontrolador de 32 bits con capacidad de debug
- Radio integrada para telemetría y telecomandos
- Unidad de medición inercial (IMU) integrada con 3 giróscopos y 3 acelerómetros
- Sensores extra: barómetro, magnetómetro, sensor de temperatura, medidor de batería
- Alimentación directa de la batería del quadrotor
- Almacenamiento no volátil: memoria SD y/o EEPROM.
- Indicadores luminosos de alta potencia

En la siguiente sección se presentan las diferentes opciones estudiadas y los criterios de selección para cada una de las partes de la computadora.

4.3.1. Selección de componentes

4.3.1.1. Microcontrolador

Para la selección de este componente no se analizaron en profundidad las diversas alternativas disponibles en el mercado sino que se optó por utilizar el mismo micrcontrolador que se utiliza en el dictado de cursos y en trabajos de investigación en el Laboratorio de Sistemas Embibidos. El microcontrolador elegido fue entonces el LPC1769 de la firma NXP Semiconductors y este probó ser una excelente alternativa como microcontrolador económico de 32 bits. Algunas de sus características principales son:

- Núcleo Cortex M3 @ 120 Mhz
- 512 KB de memoria Flash
- 64 KB de memoria RAM
- 70 pines de entrada y salida
- 8 canales de ADC de 12 bits
- Conversar DAC
- Interfaces de datos: SPI, UART, I2C, I2C, Ethernet, USB, CAN,
- 4 timers de 32 bits
- Modulos de PWM

Por otra parte, este microcontrolador ya ha sido evaluado previamente por los firmantes como plataforma para cálculo numérico en punto flotante, demostrando ser una excelente alternativa, más allá de la falta de unidad aritmética de punto flotante [22], y su gran cantidad de interfaces de entrada y salida lo hacen ideal para aplicaciones de control digital. Por otra parte este microcontrolador presenta una relación precio/desempeño adecuada, ya que el precio por unidad es de USD 11.75 y en cantidades de 1000 su costo se reduce a USD 6.4 por unidad [23].

4.3.1.2. Sensores

El sensor más importante en la computadora de vuelo es sin duda la IMU, que permite medir las velocidades angulares y las aceleraciones a las que es sometido el quadrotor. En muchas aplicaciones, este sensor es el más costoso y complejo. Dado que en este trabajo se buscó la utilización de sensores integrados, la tecnología que más se adapta a este tipo de diseños es la tecnología MEMS (Micro electro mecanichal sensors). Como se mencionaba anteriormente, las dos variables que como mínimo debe medir este sensor son la aceleración y la velocidad angular y para ello se suelen usar acelerómetros y giróscopos respectivamente. Estos pueden estar integrados en el mismo chip, y es común utilizarlos en grupos de tres, uno por cada eje. Un aspecto fundamental de la IMU es que por sí solos, los datos de aceleración y velocidad angular no proveen una estimación de actitud y se deben utilizar filtros de fusión de datos para obtener estas variables [24]. En las tablas 4.2 y 4.3 se pueden observar las comparaciones entre los distintos modelos utilizados comúnmente en quadrotores [25].

Chip	Interfaz	Ejes	Alimentación	Rango	Ancho de banda	No linealidad	Precio
ADXL330	Analógica	3	1.8v-3.6V @ 0.32 mA	±3.6G	1.6 KHz	±0,3 %	USD 14.89
SCA3100	SPI	3	3.0V-3.6V @ 3mA	±2G	2 KHz	±2 %	USD 50.82
LIS344ALH	Analógica	3	2.4V-3.6V @ 0.68mA	±6G	1.8 KHz	±0,5 %	USD 5.67
MPU6050	I2C	3	2.4V-3.46V @ 0.5mA	±16G	1 KHz	±0,5 %	USD 15

Cuadro 4.2: Comparación de acelerómetros comerciales

Chip	Interfaz	Ejes	Alimentación	Rango	Ancho de banda	Precio
ENC-03	Analógica	1	2.7v-5.5V @ 5 mA	±300°/sec	50 Hz	USD 19.77
IDG1215	Analógica	2	2.7-3.3V @ 7 mA	±67°/sec	140 Hz	USD 24.50
ADXRS620	Analógica	1	4.75v-5.25V @ 3.5mA	±300°/sec	2.5 KHz	USD 45.35
MPU6050	I2C	3	2.4v-3.46V @ 3.6 mA	±2000°/sec	256 Hz	USD 15

Cuadro 4.3: Comparación de giróscopos comerciales

El sensor MPU6050 integra al mismo tiempo 3 acelerómetros y 3 giróscopos, y sus características son suficientemente buenas para esta aplicación. Por otro lado, el mismo chip integra un procesador que fusiona los datos de los 6 sensores brindando de manera digital y a una tasa de actualización de hasta 256Hz una estimación de actitud y velocidad angular que probó ser adecuada para esta aplicación. Sumado a eso, el precio es muy inferior al resto de las opciones contempladas e implica utilizar un solo chip en lugar de dos, como sería el caso en las otras opciones. Mejor aún, este sensor permite conectarle un magnetómetro externo para hacer un filtro de fusión de datos de 9DOF con referencia de norte verdadero. Si bien esta función todavía no se encuentra disponible en el firmware del sensor, resulta una alternativa interesante a futuro para la fusión de datos. Es por estos motivos que se seleccionó al MPU6050 como sensor principal de la computadora.

Otros sensores fueron también incorporados en la computadora de vuelo a fin de hacer más versátil el diseño aunque no se utilizaron en el presente trabajo. Estos fueron:

- Magnetómetro de 3 ejes HMC5883 utilizado como brújula digital (conectado como dispositivo esclavo a la IMU)
- Sensor barométrico de presión BMP085 con resolución de ±0,01 hPa (0.5 metros aprox.)
- Sensor de temperatura LM60 con resolución de ±2°C

En este diseño se optó también por incluir puertos de comunicaciones digitales UART, I2C y SPI en los que se puede conectar un sonar o un GPS por ejemplo. Dada la gran cantidad de alternativas para este tipo de sensores y que su selección estaba fuera del alcance de este trabajo, se prefirió por no incorporar uno en el diseño.

4.3.1.3. Radio

Para este trabajo se buscó una radio con las siguientes características:

- Enlace de datos bidireccional
- Alcance mayor a 100 metros
- Velocidad de transmisión mayor a 56 Kbps
- Interfaz digital
- Posibilidad de trabajo en redes tipo mesh
- Antena integrada
- Funcionamiento en bandas ISM.

Si bien existen varias radios que reúnen estas características, se optó por la radio XBee Pro 2.4Ghz de la marca Digi por su amplia penetración en el mercado, su excelente precio y modularidad. Además, es posible seleccionar radios de la misma marca y con la misma interfaz para diferentes distancias, con diferentes tipos de antena y suelen ser muy versátiles en cuanto a configuración. Permiten trabajar en redes mesh y tienen una gran cantidad de funciones adicionales como corrección de errores, medición de potencia de recepción, etc. que las hacen ideales para esta aplicación. Su costo ronda los USD 20 para una versión con un alcance de aproximadamente 1 km al aire libre, una tasa de datos máxima de 115.2 Kbps y antena incorporada.

4.3.2. Arquitectura

El diseño final de la computadora de vuelo se basó en el diagrama en bloques de la figura 4.5. En este se puede observar que el núcleo de la computadora es el microprocesador LPC1769 que está conectado a todos los dispositivos mencionados anteriormente. También se puede observar que gran parte de los sensores y periféricos de la placa son conectados mediante un bus de comunicaciones I2C lo que permitió un desarrollo más ágil del firmware y la fácil comprobación externa de la placa luego del ensamblado.

En la figura 4.5 también se puede observar la presencia de un regulador de tensión DC-DC para generar la tensión necesaria para alimentar el circuito como así también un zócalo para memorias uSD que permite registrar la telemetría del vuelo. El diseño final se presenta en la figura 4.6 y en el apéndice A se presentan los esquemáticos correspondientes.

4.3.3. Firmware

El firmware de la computadora de vuelo fue escrito íntegramente en el lenguaje de programación C y se realizó exclusivamente para este trabajo desde cero, con el fin de mantener el programa lo más simple posible. El desarrollo del sistema se basó en el sistema operativo de tiempo real FreeRTOS v7.1.0 el cual permite realizar aplicaciones con múltiples tareas en paralelo de manera simple y sencilla.

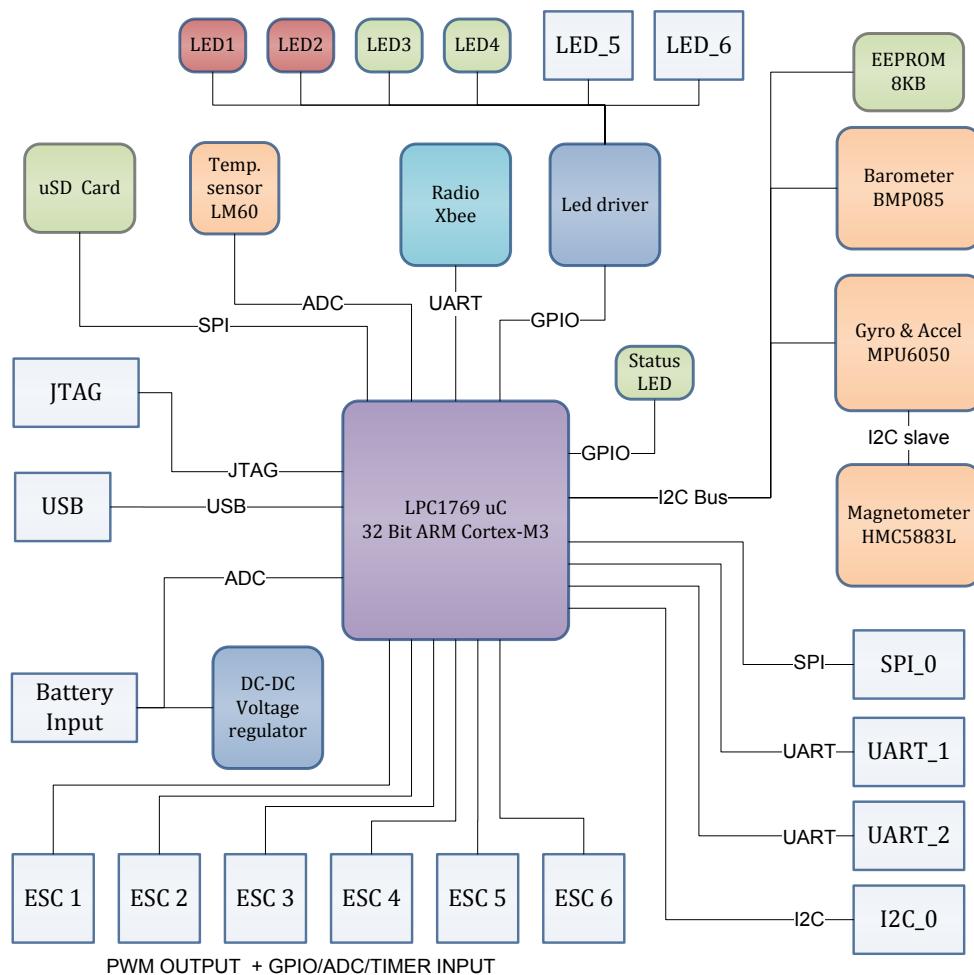


Figura 4.5: Diagrama en bloques de la computadora de vuelo. Los bloques rectangulares con fondo gris representan conectores en la placa.

Como en la mayoría de los sistemas embebidos, resulta importante diseñar el firmware de forma tal que este no dependa estrechamente del hardware donde es utilizado, concepto denominado “abstracción de hardware”. Esto permite obtener un software sencillo de modificar y que puede ser fácilmente portado a otras plataformas. Con el fin de lograr este objetivo, se diagramó una estructura de software embebido compuesta por las siguientes partes:

- HAL (Hardware Abstraction Layer)
- API (Application Programming Interface)
- Aplicación

El HAL es la capa encargada de controlar el hardware, siendo la única específica para el mismo y que aporta funciones de manejo de los periféricos de microcontrolador. Dado que solo el HAL es dependiente del hardware, ante una eventual modificación de hardware solo se debe modificar esta capa y el resto de la aplicación seguirá siendo válida.

4.3. DISEÑO E IMPLEMENTACIÓN DE LA COMPUTADORA DE VUELO

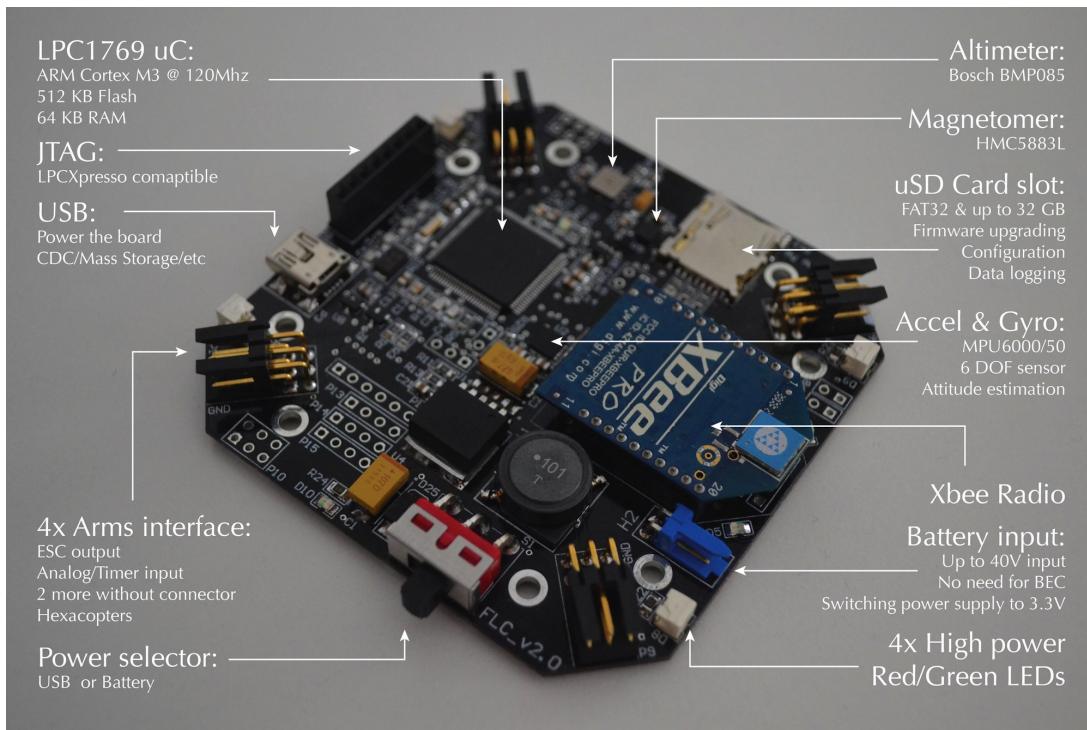


Figura 4.6: Computadora de vuelo diseñada en el presente trabajo.

La API es la capa que provee al usuario los métodos de programación. Esta capa es intermedia entre el usuario y el HAL, y no tiene acceso al hardware sino que se ve limitada al uso de las funciones otorgadas por el HAL.

La capa de aplicación es la de mayor nivel de abstracción y es donde se programa el software de control de vuelo. Esta hace uso solamente de la API, no pudiendo acceder directamente al HAL o al hardware. En este caso, la aplicación será el algoritmo de control, navegación o cualquier otra funcionalidad que se necesite agregar.

En Fig. 4.7 se muestra un diagrama de la implementación en las capas previamente descriptas.

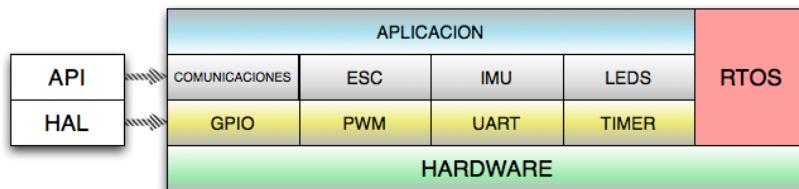


Figura 4.7: Arquitectura del firmware

Esta división por capas, si bien implica una mayor complejidad en la programación y aumenta el tiempo de desarrollo de las etapas iniciales, permite luego una muy rápida evolución del código, con un tiempo de desarrollo muy corto.

4.3.4. Arquitectura de Comunicaciones

Para aumentar la confiabilidad y versatilidad del sistema se desarrolló una arquitectura completa para las comunicaciones. Se diseñó un protocolo capaz de detectar errores en las comunicaciones y de transportar paquetes de información entre nodos. Este subsistema se subdividió en cuatro capas: Aplicación, Transporte, Enlace, Física.

Las capas físicas y de enlace son íntegramente controladas por los módulos XBee, dejando los modos de direccionamiento a las capas de nivel superior. Se eligió esto ya que estos módulos poseen probada efectividad en este tipo de aplicaciones [26].

La capa de transporte se encarga de que tanto los datos salientes como entrantes cumplan con un determinado formato. Esta conformación del paquete permite la detección de errores y la separación de paquetes en función de su utilidad. El módulo de comunicaciones de la capa API de la estructura de software se ocupa de este rol mediante la implementación de una máquina de estados. La comunicación con los módulos XBee la realiza a través del módulo UART de la capa HAL de la estructura de software.

En Fig. 4.8 se muestra el formato de la trama establecida por el protocolo implementado. En la misma se aprecia que la carga útil de la misma es de 0-255 Bytes, dejando encargada del fragmentado de los paquetes a la capa de aplicación.

HEADER	SRC	DST	TIME/SEQ	TYPE	LEN	PAYOUT	CHECKSUM
1B	1B	1B	1B	1B	1B	0-255B	1B

Figura 4.8: Trama del protocolo de comunicaciones

La capa de transporte es además la encargada de verificar la trama mediante las direcciones de fuente y destino, número de secuencia y “checksum”. Dado que el protocolo permite interconexión entre diferentes nodos, en ambos sentidos, se requiere un tipo de dato que especifique el carácter de la transmisión (control, telemetría, “uplink”, “downlink”). El contenido del “payload” es presentado a la aplicación de la capa superior que requiere este servicio mediante un patrón publisher/subscriber [27]. La capa de aplicación implementa la utilidad del protocolo de comunicaciones. Es posible implementar hasta 256 tipos diferentes de datos, diferenciándolos en su propósito. Los tipos de datos implementados actualmente son:

- System: transmite comandos de sistema, como el cambio entre modos de funcionamiento del vehículo, reajuste de parámetros, etc.
- Control: transmite información acerca del control de vuelo, tales como cambio en la posición, navegación, acciones, etc. En este caso en particular, se transmite el estado de las referencias impuestas por el joystick.
- Debug: transmite mensajes de baja prioridad, a interpretarse desde la consola terrestre como caracteres ASCII.
- Telemetry: transmite paquetes de datos relacionados con la telemetría. Estos son los que se utilizarán en los próximos capítulos para contrastar con las simulaciones.

Finalmente el protocolo diseñado probó ser adecuado y se lo utilizó extensamente durante todo el desarrollo del trabajo. Los experimentos y resultados que se presentan en los capítulos 5 y 6 fueron medidos mediante los sensores incorporados en la computadora de vuelo y descargado vía telemetría utilizando el sistema de comunicaciones presentados en este capítulo.

4.4. Segmento de tierra

Como se aprecia en la descripción del protocolo de comunicaciones, por el canal de comunicaciones vía radio se podrán tener diferentes tipos de paquetes, transportando información que será útil a distintos puestos de trabajo, como ser: la interfaz de control para un piloto, visualizadores de telemetría, grabadoras de datos, representaciones en tiempo real, etc. Dado que estos datos son recibidos y/o generados por diferentes aplicaciones, que en general están escritas en diferentes lenguajes y posiblemente ejecutándose en diferentes computadoras de tierra, se buscó un protocolo de comunicaciones entre procesos que sea potente y a su vez sencillo de implementar para las diferentes aplicaciones.

El RadioServer es un servidor desarrollado por el autor exclusivamente para este trabajo que corre en tierra y se encarga de distribuir las comunicaciones entre los dispositivos de tierra y el quadrotor. Ante el arribo de un mensaje, se dirigirá el mismo al usuario en cuestión, utilizando puertos TCP. Esta estructura de comunicaciones brinda una gran flexibilidad en cuanto al espectro de utilización del dispositivo, pudiendo desplegar simultáneamente funciones en campo y en laboratorio. En Fig. 4.9 se ilustra el funcionamiento del servidor, junto con algunos ejemplos de clientes que se comunican con el cuadricóptero.

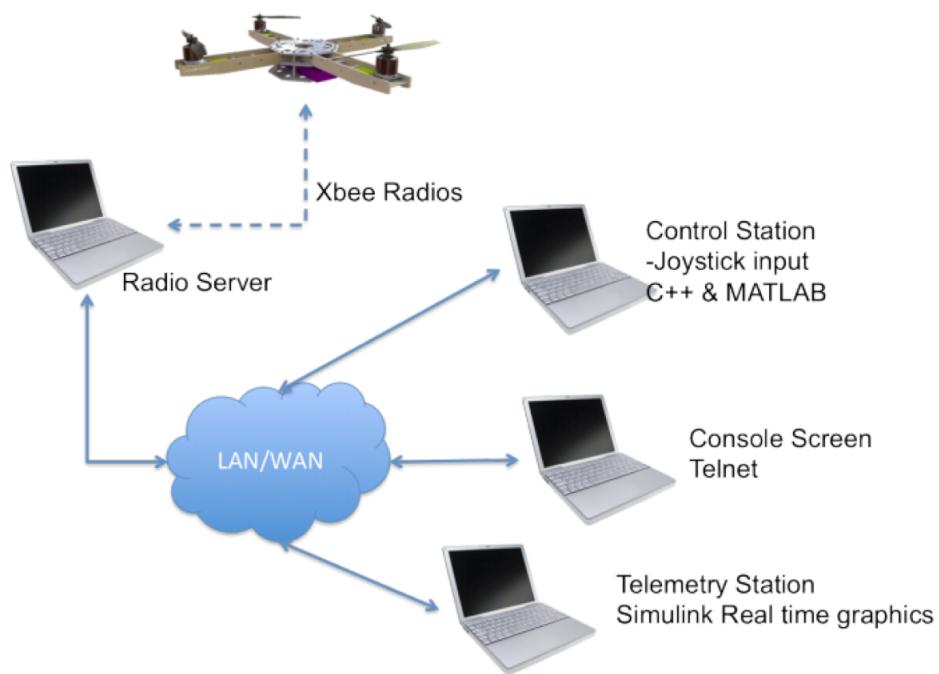


Figura 4.9: Esquema de funcionamiento del servidor

Por falta de espacio aquí no se describe con más detalle el servidor RadioServer desarrollado. Para más referencias el lector puede visitar <https://github.com/QUADLSE/RadioServer> donde se encuentra el repositorio en línea del código.

4.5. Resumen del capítulo

En este capítulo se presentaron las herramientas que fueron diseñadas y utilizadas a lo largo del desarrollo del trabajo. Los simuladores fueron una herramienta fundamental del trabajo y permitieron tener un primer acercamiento al concepto de quadrotor. Incluso se pudo im-

plementar un simulador 3D en Simulink que era controlado con el mismo joystick con el que despues se controlará el modelo de vuelo. Esto resultó en una herramienta muy poderosa ya que se puede considerar la realización de simuladores de este tipo para el entrenamiento de pilotos y la simulación de misiones.

El quadrotor utilizado probó ser resistente y se utilizó el mismo durante todas las pruebas. Los variadores de velocidad que incorpora el kit resultaron ser excelentes y su frecuencia de actualización de hasta 425 Hz lo ubican dentro de los mejores del mercado. Finalmente la computadora de vuelo junto con el segmento de tierra probaron ser una solución práctica para trabajar y permitieron realizar todos los ensayos y experimentos que se detallan en el capítulo 6 donde también se presentan los resultados de performance de la computadora de vuelo.

CAPÍTULO 5

Implementación

En los capítulos anteriores se estudió el modelo de la planta y el diseño del controlador PID a utilizar. Estos constituyen los bloques fundamentales del sistema de control, y para su utilización debe diseñarse una estrategia de control, basada en este modelo y en la implementación del controlador para cada uno de los ejes. En este capítulo se presentará primero un análisis de la validez de utilizar controladores digitales en plantas continuas. Luego se presentará la estrategía de control elegida y los distintos modos de vuelo que ésta permite. Por último se realizará un análisis de aquellos factores no ideales que pueden perjudicar el desempeño del sistema de control implementado, para concluir que dentro de margenes acotados el diseño es robusto ante diversos factores, como por ejemplo imprecisiones en el modelado, o en el funcionamiento de los sensores.

5.1. Implementación digital del controlador

Como se vio en los capítulos 2 y 3, el modelo de la planta que se utilizó en las simulaciones esta representado en variable continua, mientras que el controlador PID, al ser ejecutado en la computadora de abordo, es inherentemente digital. Si bien todos los sistemas reales son continuos, más alla del modelo utilizado, el controlador siempre será digital y esto trae asociado un tema crítico en el estudio de los sistemas de control digitales que es la selección del tiempo de muestreo.

Cuando los microcontroladores no eran lo suficientemente rápidos, este era un tema de investigación muy activo ya que no era trivial decir que la frecuencia de muestreo estaba correctamente elegida. Hoy en día, con procesadores cada vez más veloces, para el caso de sistemas de control de procesos industriales, con constantes de tiempo muy grandes, se podría decir que no es necesario dedicarle tiempo a la selección de este parámetro sino que elegir una frecuencia de muestreo de 50 Hz, por ejemplo, siempre es suficiente.

Un criterio ampliamente utilizado para la selección de la frecuencia de muestreo es el propuesto por Karl J. Astrom en su libro “Computer Controlled Systems” [20] donde se compara el tiempo de muestreo del controlador T_s con el tiempo de crecimiento T_r de un sistema continuo, dado que este es un parámetro fácilmente observable. El criterio dice que se debe seleccionar una frecuencia de muestreo T_s tal que el número de muestras dentro del tiempo de crecimiento

$$N_r = \frac{T_r}{T_s} \quad (5.1)$$

sea de entre 4 y 10 veces. En el caso de un sistema de primer orden el tiempo de crecimiento es igual a la constante de tiempo del mismo pero dado que la planta de cada eje del quadrotor prácticamente es solo un integrador no se posible determinar el tiempo de crecimiento a lazo abierto. Dado que a lazo cerrado se espera que el tiempo de crecimiento se encuentre en el orden de los 200ms, se obtiene tomando el mejor caso del criterio de Astrom

$$T_s = \frac{200ms}{10} = 20ms \quad (5.2)$$

que equivale a una frecuencia de muestreo de 50 Hz. Para evitar problemas de muestreo, se escogió de forma árbitraria una frecuencia de muestreo, aún mayor que la recomendada por este criterio, de 200 Hz. Si bien aumentar la frecuencia de muestreo aumenta la carga de trabajo de la CPU que debe procesar los controladores, este valor no resultó excesivo para la computadora de vuelo. Más detalles sobre esto se pueden encontrar en el capítulo 6.

5.2. Diseño en cascada

En la aeronáutica se suele utilizar un sistema de control dado por dos lazos en cascada [28]. El primero, denominado SAS (Stability augmentation system), es el encargado de controlar y estabilizar la velocidad angular en uno o más ejes, y el segundo, comúnmente conocido como “piloto automático” se denomina CAS (Control augmentation system) y es el encargado de asegurar una actitud determinada, que puede ser en este caso, vuelo en hovering. En la figura 5.1 se puede observar un diagrama de bloques del sistema de piloto automático completo para un eje de la aeronave.

Es importante destacar que sin el SAS mantener a la aeronave en hovering demandaría un esfuerzo del piloto enorme ya que las asimetrías en la estructura o las disparidades entre actuadores y hélices harían que la aeronave sea muy inestable. En este caso, donde la aeronave tiene un tensor de inercia y masa pequeñas, la dinámica de la misma se vuelve muy rápida. Esto

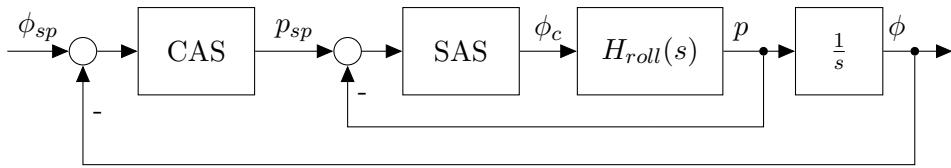


Figura 5.1: Sistema de control del piloto automático completo

imposibilita que un piloto humano vuela directamente la aeronave sin asistencia por parte de un controlador tipo SAS como mínimo. En cambio el CAS es un agregado optativo y un piloto puede volar sin este, pero al estar activado es posible hacer vuelos controlados con intervención mínima o nula por parte del piloto.

5.3. Diseño del SAS

El SAS debe controlar la velocidad angular de cada uno de los ejes. Se busca entonces diseñarlo de forma tal que el mismo pueda seguir de la mejor manera posible las referencias introducidas por el piloto, ya sea de manera manual o mediante la salida del CAS, y que pueda, también eliminar perturbaciones externas. Dado que los tres ejes del quadrotor tienen una dinámica similar (salvo en el valor de las constantes), es posible utilizar el mismo tipo de controlador para cada eje. En esta sección se presenta el método de diseño utilizado para el ajuste de las constantes de cada uno de los ejes. Dado que el eje más sencillo de ensayar en pruebas de banco resultó ser el eje de yaw (es posible colgar el quadrotor de una estructura suspendida y permitir que este rote libremente sobre el eje \hat{z}), se presenta el procedimiento para este eje. Cabe aclarar que si bien los experimentos no se realizaron con la aeronave estrictamente volando por sus propios medios, se puso un valor de colectivo $Z_c = 0,57$ que representa una entrada del 40 % en los actuadores. Esto permitió estar en un punto válido de la linealización de los actuadores a la vez que evitó que la aeronave levantara vuelo y complicara la realización del experimento. En todos los casos, las mediciones de velocidad angular y la actitud fueron realizadas con la IMU incluida en la computadora de a bordo diseñada, que se presentó en el capítulo 4, y fueron descargados mediante telemetría por radio a una tasa de 20Hz.

5.3.1. Análisis de la planta

Se parte de la función de transferencia de la planta (eje yaw del quadrotor):

$$H_{yaw}(s) = \frac{r(s)}{\psi_c(s)} = 4K_\psi \frac{K_{M\omega}}{(T_1 s + 1)(sI_{zz})} \quad (5.3)$$

donde los parámetros utilizados son los encontrados en el capítulo 2 con la diferencia que se ajustó la constante de torque del actuador a $K_{M\omega} = 0,05 \frac{Nm}{\%}$. Este resultó ser un valor más adecuado para el modelo y se ajustó de forma experimental. También, como se propuso en el capítulo 2, la constante de tiempo T_1 se consideró nula.

Dadas estas modificaciones, se encontró que la planta tiene un solo polo en $p_1 = 0 + j0$ producto del término integral de la ecuación de momento linealizada 2.1. Entonces es posible entonces utilizar un controlador tipo PI [29] que pudiera satisfacer las dos condiciones impuestas al control SAS:

- Ser capaz de seguir referencias.

- Ser tolerante a perturbaciones.

En la siguiente subsección se presenta el ajuste de este controlador.

5.3.2. Ajuste del controlador

El controlador PI utilizado se ilustra en la figura 5.2. Mediante el asistente de calibración de controladores de Simulink se ajustaron los valores de las constantes del controlador PI de manera iterativa hasta obtener un conjunto de parámetros que cumpliesen el objetivo de seguir referencias y tolerar perturbaciones con un desempeño adecuado.

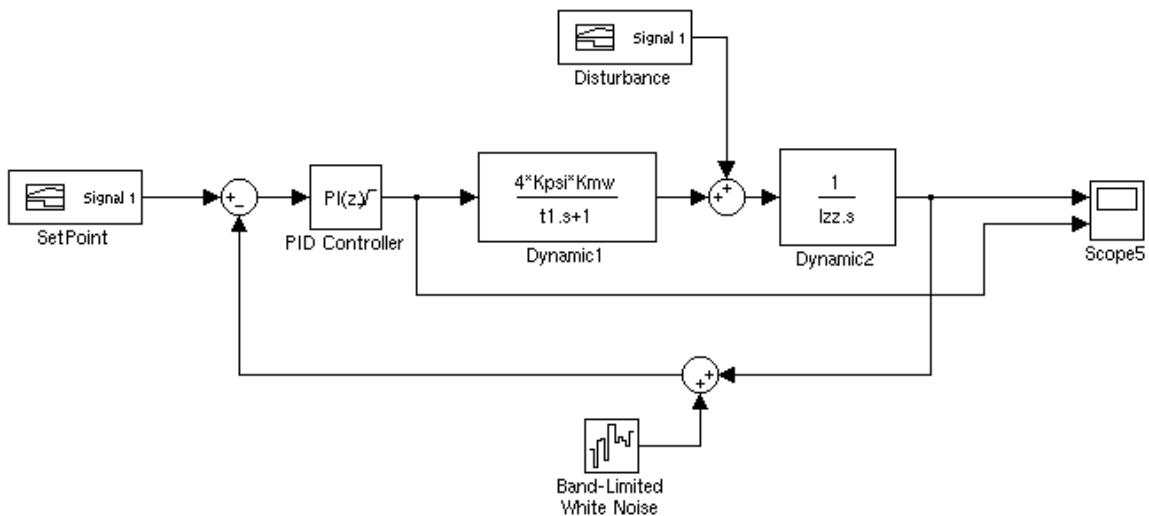


Figura 5.2: Modelo de simulación en MATLAB Simulink

En la figura 5.2 es posible observar que la planta se encuentra dividida por un lado en todo lo que genera torques sobre el eje \hat{z} , y por otro, en la inercia del eje y su término integral. También es importante destacar que la planta se simuló en tiempo continuo mientras que el controlador PI(Z) se implementó de manera digital con la frecuencia de muestreo seleccionada de 200 Hz lo que no generó ningún inconveniente. Finalmente se encontró que con las constantes $K = 0,05$ y $T_i = 10,0$ el sistema es estable y puede seguir una referencia de entrada. El controlador fue configurado con Anti Windup y la salida fue restringida a $[-1,0, 1,0]$ para ser compatible con el modelo diseñado. Como se presentó en los capítulos anteriores, el PID digital implementado para este trabajo, es ejecutado en la computadora vuelo diseñada y se corresponde directamente con el controlador PID discreto de Matlab. Es por eso que fue posible luego de encontrar estas constantes introducirlas en la computadora de vuelo y verificar el correcto funcionamiento del SAS en este eje. Se pudo comprobar la correspondencia entre el simulador y la planta real como así también la estabilidad de ambas. Este resultado se muestra en la figura 5.3.

Es posible observar en la figura 5.3 ambas respuestas superpuestas como así también la entrada de referencia. Para el experimento se utilizó exactamente la misma entrada tanto para el modelo de ejes desacoplados en Simulink como para la planta real y los resultados fueron prácticamente idénticos sólo que la señal proveniente del modelo real es ligeramente más ruidosa. Este ruido no es un ruido eléctrico o de muestreo sino que se debe a las vibraciones propias del banco de pruebas. También es posible comparar la salida de control generada por el controlador simulado y por el controlador real en el gráfico inferior. Se observa que la misma nuevamente

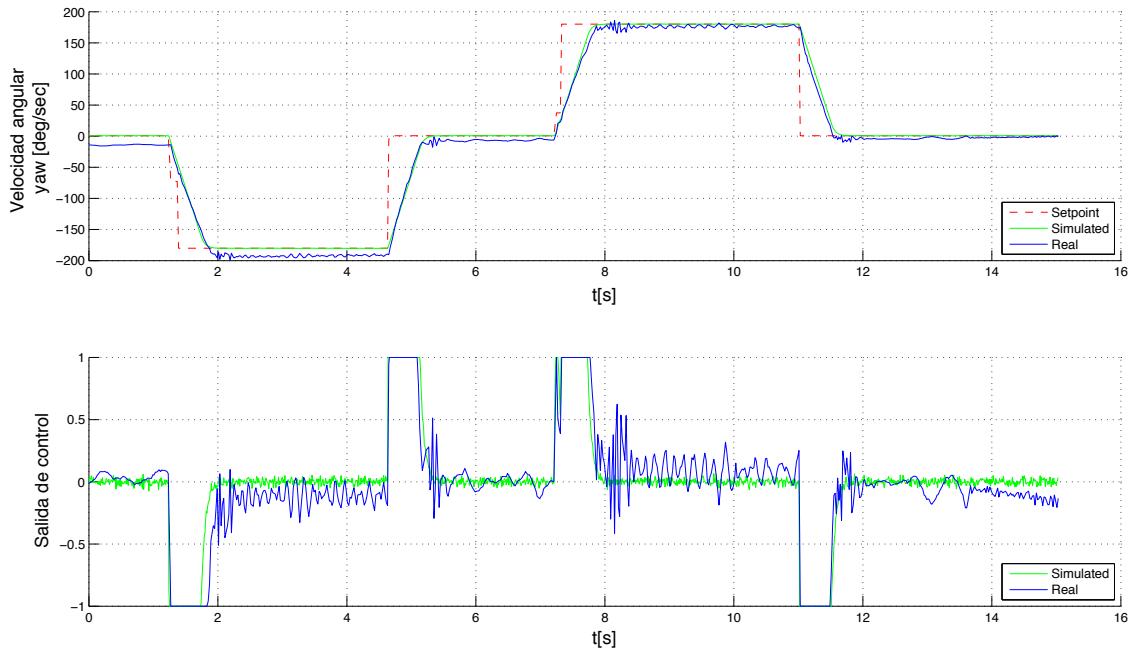


Figura 5.3: Resultado de la simulación del SAS ante una referencia del tipo escalon en la velocidad angular del eje de yaw. Se utilizaron los parámetros $K = 0,05$ y $T_i = 10,0$ en el controlador con una frecuencia de muestreo de 200 Hz.

contiene un nivel de ruido mayor debido a que el término proporcional lo amplifica de la señal de entrada. En el gráfico 5.4 se puede observar la diferencia entre la velocidad angular del experimento con el quadroto real y el resultado de la simulación. Es posible observar que la diferencia entre ambas velocidades angulares se mantienen dentro de un rango de los 20 grados por segundo como cota superior. En la mayoría de los casos, donde la diferencia es grande, esto se debe al momento de aceleración, pero en los períodos de velocidad angular constante se puede observar que la diferencia está dentro de los 5 grados por segundo de error.

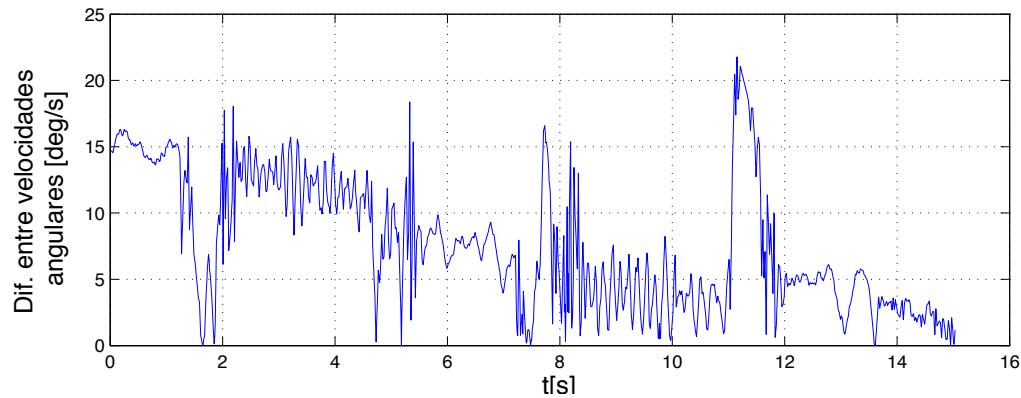


Figura 5.4: Comparación entre la velocidad angular real medida y la simulación (en valor absoluto)

Por otra parte es posible extraer conclusiones de este resultado que resultan fundamentales para el diseño del sistema. Primero se puede observar que el actuador esta saturado durante el

tiempo que le toma a la planta llegar a la velocidad angular deseada. Este tiempo “de crecimiento”, denominado así por analogía con los circuitos electrónicos, esta determinado exclusivamente por el torque máximo generado en el eje de yaw y por la planta, y no por el controlador elegido ya que, como se observa, el mismo ya está saturado, y ejerce el máximo posible de acción correctiva. Como los parámetros de inercia del rígido y torque de los motores se consideraron constantes para este trabajo, el único parámetro de ajuste que queda es el componente $K\psi$ de la matriz M_{mix} . Esta constante define cuánto es el máximo de control que el eje de yaw puede ejercer sobre los actuadores. Si se modifica este valor, que en el primer experimento era de 30 % se logran los resultados mostrados en la figura 5.5.

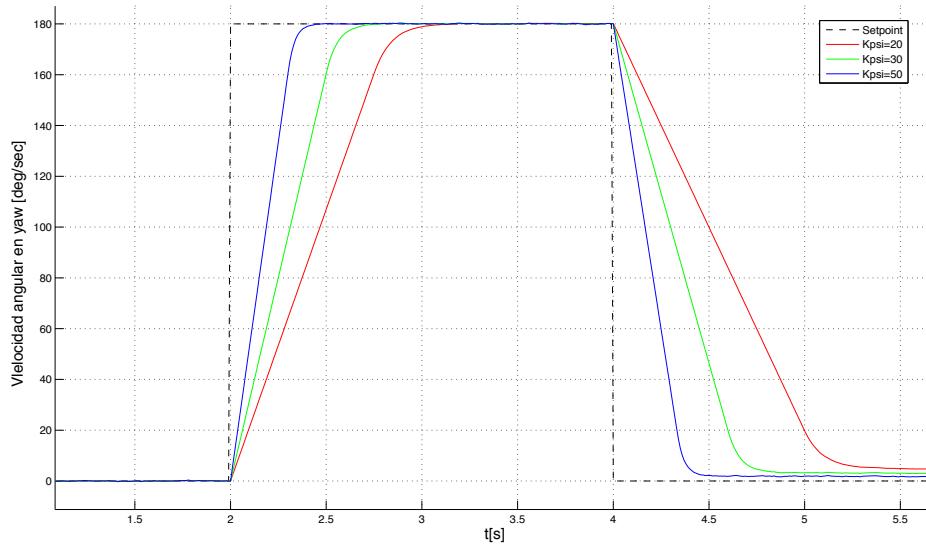


Figura 5.5: Comparación de la aceleración angular para diferentes valores de $K\psi = \{20, 30, 50\}$

Es decir que la constante $K\psi$ determina la aceleración angular máxima que podrá lograr el quadrotor. Este valor está finalmente dado por:

$$\dot{r}_{max} = \frac{4K\psi K_{Mw}}{I_{zz}} \quad (5.4)$$

y de manera análoga para los ejes de pitch y roll se obtiene:

$$\dot{p}_{max} = \frac{4dK\phi K_{Tw}}{I_{xx}} \quad (5.5)$$

$$\dot{q}_{max} = \frac{4dK\theta K_{Tw}}{I_{yy}} \quad (5.6)$$

Como se presentó en el capítulo 2, los valores K_ϕ , K_θ y K_ψ se deberán elegir entonces para lograr las aceleraciones de rotación deseadas teniendo en cuenta el límite de la mezcla de todos los controladores.

Por último es importante mencionar que el SAS diseñado presenta un desempeño muy satisfactorio, ya que otros ajustes más agresivos provocaban sobrepicos o por el contrario controladores más suaves hacían la respuesta aún más lenta, por lo que se prefirió la utilización de una solución subamortiguada.

En los ejes de pitch y de roll se aplicó el mismo procedimiento de ajuste de las constantes. En el cuadro 5.1 se presenta un resumen de los parámetros de ajuste utilizados para cada uno de los ejes.

Eje	K	T_i
Roll	0.01	0.1
Pitch	0.01	0.1
Yaw	0.05	0.1

Cuadro 5.1: Constante de ajuste para los controladores SAS de los diferentes ejes.

5.3.3. Validación

Finalmente para validar el controlador desarrollado para el eje de yaw con una entrada de control real, se utilizó el mismo banco de pruebas presentado que para el ajuste y se controló el setpoint mediante un joystick externo. Luego se utilizó este mismo setpoint como entrada en la simulación para correlacionar ambos resultados. En la figura 5.6 se observa que el sistema puede seguir la referencia de velocidad angular sin problemas y que la simulación y el quadrotor real se comportan de manera similar. En esta no se incluye un gráfico de la posición angular absoluta ya que como no era la variable controlada, no aporta información al experimento.

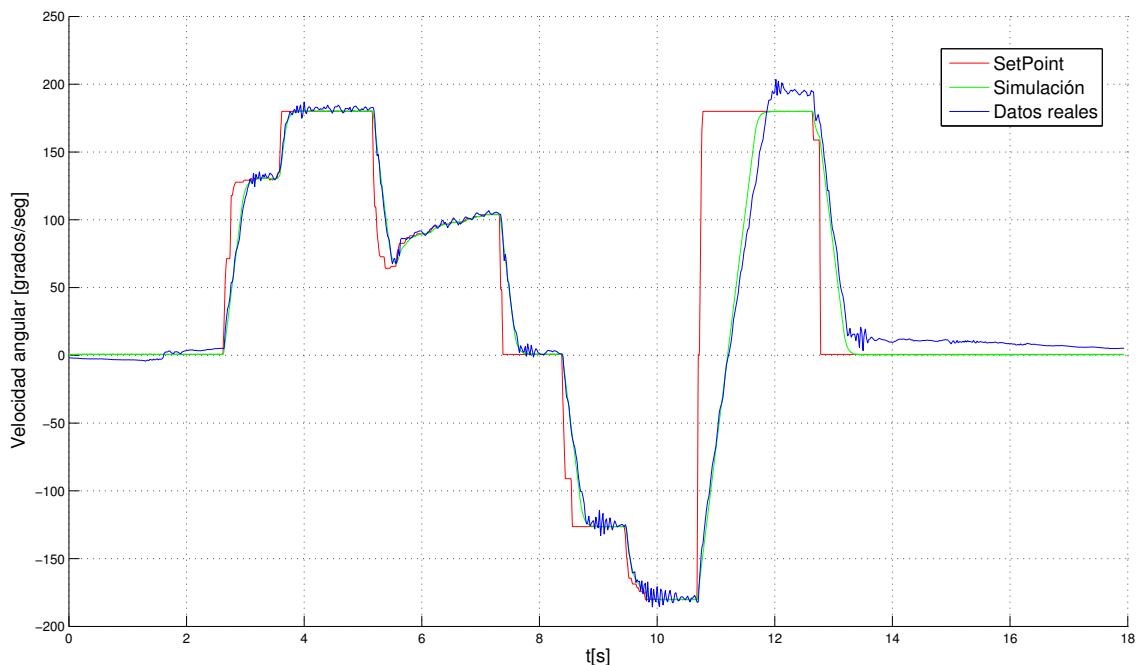


Figura 5.6: Resultados de utilizar el SAS de control de yaw para mantener la velocidad angular según el setpoint. Las constantes del PID del SAS utilizadas fueron: $K = 0,05$, $T_i = 10,0$ y $T_d = 0,0$.

5.4. Diseño del CAS

El diseño del CAS se realizó de manera similar al del SAS. En este caso, la entrada al controlador en vez de ser de referencia de velocidad angular es directamente el ángulo en el que se quiere que el eje de la aeronave se mantenga. En el caso del hovering esta entrada resulta 0° para pitch y roll mientras que el eje de yaw puede tomar cualquier valor arbitrario. La salida del CAS es una velocidad angular, es decir que el sistema CAS controla la velocidad angular que es la entrada del SAS.

5.4.1. Análisis de la planta

En este caso, la planta que debe controlar el CAS no es simplemente el quadrotor y su transferencia por eje, sino la transferencia que determina el ángulo de la aeronave en función de una entrada de control de velocidad angular. Esto incluye, entonces, la transferencia original por eje, un término integrador que pasa de velocidad angular a ángulo y por último, el lazo del controlador PID como se puede ver en la figura 5.7 marcado en color.

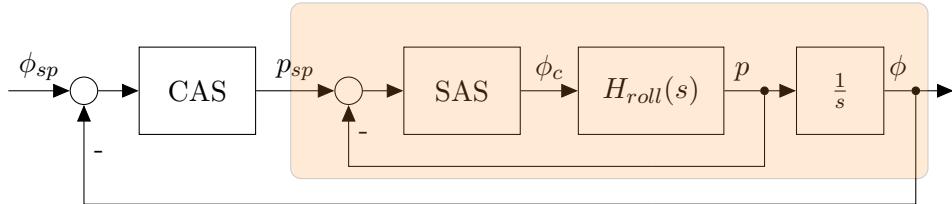


Figura 5.7: Sistema de control en donde se detalla en color cual es la planta que debe controlar el CAS.

Resulta entonces de utilidad encontrar una función de transferencia única que permita caracterizar a toda lo que el CAS ve como planta. Para eso primero se deben combinar las transferencias del controlador SAS y la planta. Luego se debe simplificar el lazo de realimentación para luego poder combinarlo con el integrador. Como se observa en la figura 5.8, para un lazo de control con realimentación negativa, se puede encontrar una transferencia equivalente que simplifique la función de transferencia.

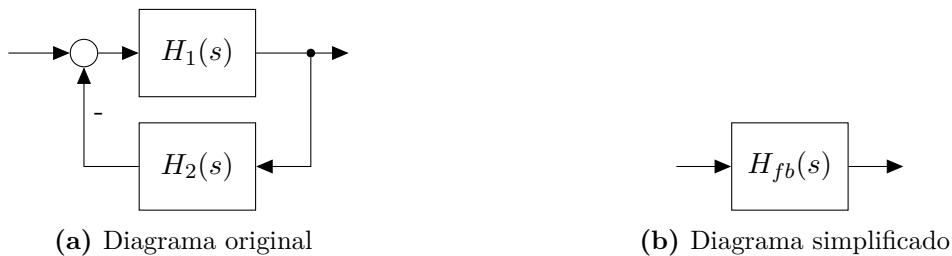


Figura 5.8: Simplificación del diagrama de bloques de un sistema con realimentación negativa.

donde la transferencia equivalente resulta:

$$H_{fb}(s) = \frac{H_1(s)}{1 + H_1(s)H_2(s)} \quad (5.7)$$

De este modo, desde el punto de vista del CAS la transferencia de la planta se convierte en:

$$H_\psi(s) = \frac{H_{yaw}(s)H_{PID}(s)}{1 + H_{yaw}(s)H_{PID}(s)} \frac{1}{s} \quad (5.8)$$

Como no se utilizó el término derivativo y la constante para el camino del error fue $b = 1$, se obtiene que:

$$H_{PID}(s) = Ke(s) \left(1 + \frac{1}{T_i s} \right) \quad (5.9)$$

Simplificando la ecuación 5.8 y modificando el valor del término integral del PID para contemplar que este es digital es posible obtener la función de transferencia del eje con el SAS como:

$$H_\psi(s) = \frac{s + T_i T_s}{\frac{I_{zz}}{4KK_\psi K_{M_w}} s^3 + s^2 + T_i T_s s} \quad (5.10)$$

donde K , T_i y T_s son los parámetros de ganancia, tiempo de integración y tiempo de sampleo del controlador PID del SAS y el resto de los parámetros corresponden al modelo del eje. Este modelo de “planta controlada en velocidad angular” solo es válido si el controlador digital PID de su interior no satura. Si este satura, entonces esta función de transferencia lineal deja de ser válida y no es correcta su utilización. Dado que las herramientas de Simulink permiten ajustar los parámetros del controlador, incluso si la planta a controlar no es lineal, se utilizó el modelo completo con el PID del SAS.

5.4.2. Ajuste del controlador

De manera similar al ajuste para el CAS, se buscó un controlador que permita seguir referencias aún en presencia de perturbaciones. Como se mencionó en la sección anterior, dado que el controlador del SAS satura en muchos casos, se prefirió dejar el modelo completo y no la simplificación de la ecuación 5.10. Como criterio para el ajuste de las constantes se buscó que el sistema sea lo más rápido posible, a fin de minimizar el tiempo de establecimiento, manteniendo al sistema sobreamortiguado para que el mismo no presente sobrepicos. Esto resulta de suma importancia si el piloto que lo utiliza es humano ya que el sobreceso tiende a que el piloto intente corregirlo manualmente, lo cual perjudica el desempeño. En la figura 5.9 se puede observar el seguimiento de referencias del CAS sin sobrepicos.

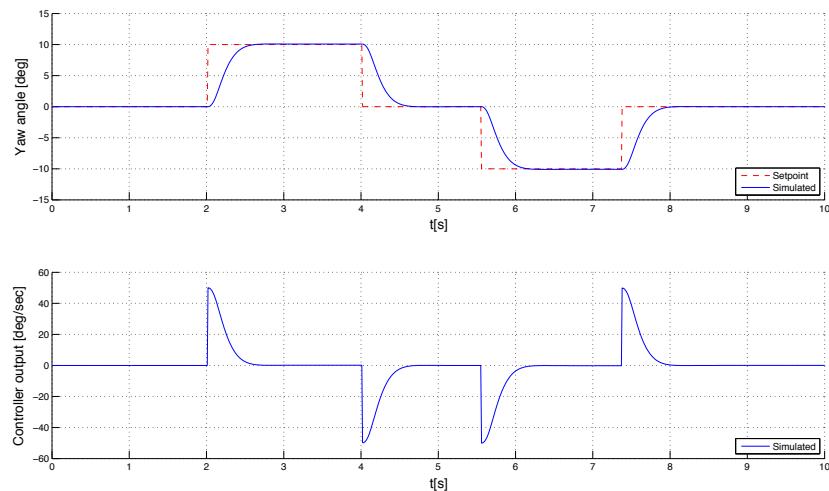


Figura 5.9: Respuesta del quadrotor (eje yaw) con el controlador CAS y SAS. Los ctes. de ajuste fueron $K = 5$ y $T_i = 0,05$

Si bien el SAS tenía como salida la variable de control ψ_c cuyos valores máximos y mínimos son -1 y 1, para el caso del CAS, la salida del controlador es un comando de referencia de velocidad angular. Entonces es posible, en principio, ajustar los límites del controlador CAS según la máxima velocidad angular que el eje pueda tomar. Por supuesto, el límite superior va a estar dado por la dinámica del sistema, pero esta variable de ajuste permite cambiar la dinámica de la planta sin afectar la estabilidad. En la figura 5.10 se puede observar el efecto de cambiar este parámetro en el controlador. Como en este caso, el controlador solo llega a generar una salida de aproximadamente $50^\circ/s$ sólo valores máximos por debajo de este valor cambian la respuesta de la planta pudiéndola hacer más o menos lenta.

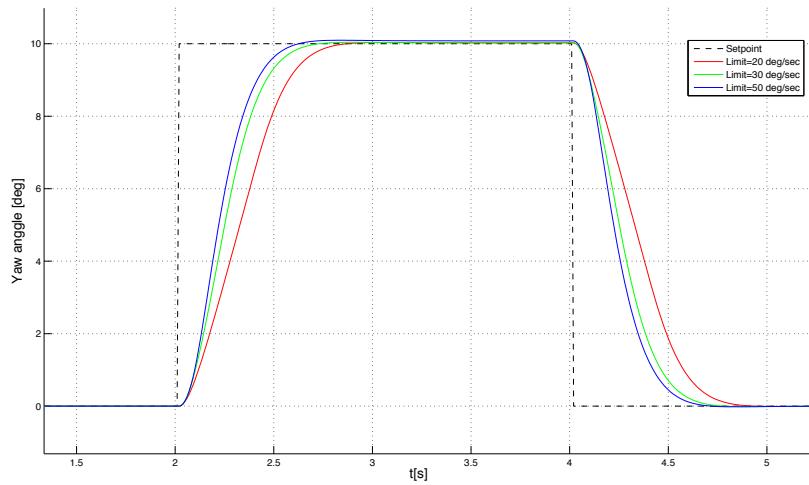


Figura 5.10: Efecto del límite del controlador PID del CAS para control de yaw con el controlador CAS y SAS activados. El CAS fue ajustado con $K = 5$ y $T_i = 0,05$.

En los ejes de pitch y de roll se aplicó el mismo procedimiento de ajuste de las constantes. En el cuadro 5.2 se presenta un resumen de los parámetros de ajuste utilizados para cada uno de los ejes.

Eje	K	T_i
Roll	1.0	0.02
Pitch	1.0	0.02
Yaw	5.0	0.05

Cuadro 5.2: Constante de ajuste para los controladores CAS de los diferentes ejes.

5.4.3. Validación

En un experimento similar a la validación del SAS, pero esta vez utilizando también el CAS para control de actitud en yaw, se obtuvieron resultados satisfactorios que se pueden observar en la figura 5.11. Ambos, la simulación y la aeronave real, se comportan de manera similar. Se observa también un primer pico en la actitud real que se debe a una perturbación externa que el sistema rápidamente elimina a los dos segundos del experimento. Por otra parte, se observa la presencia de sobrepicos en la respuesta, esto se debe a que se buscó un ajuste del controlador que diera una respuesta subamortiguada a fin de evaluar la concordancia entre el modelo simulado y el real, lo cual se logró con éxito. En el resto de los experimentos se ajustó el controlador más suavemente para evitar este sobrepico y lograr una respuesta sobreamortiguada como se puede observar en la figura 5.9.

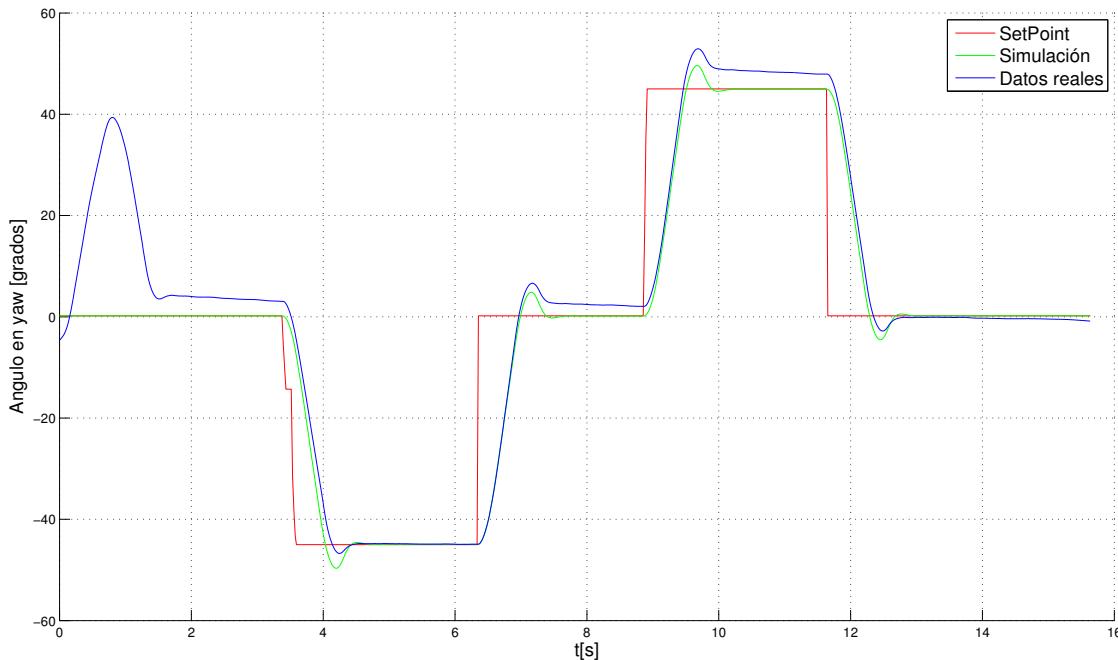


Figura 5.11: Resultados de utilizar el CAS de control de yaw para mantener la orientación en yaw según el setpoint. Las constantes del PID del CAS utilizadas fueron: $K = 11,0$, $T_i = 5,90$ y $T_d = 0,0$.

5.5. Modos de vuelo

Combinando el sistema de control SAS y CAS de diferentes maneras es posible definir distintos modos de vuelo que pueden resultar útiles en distintas situaciones. Se diseñaron para los experimentos tres modos de vuelo cuyos diagramas de bloques se pueden observar en las figuras 5.12, 5.13 y 5.14.

El primero de estos modos, llamado modo acrobático (“acrobatic” de ahora en adelante), solo implementa el SAS en cada uno de los ejes del quadrotor. Esto permite que el mismo sea relativamente estable en vuelo. Este tipo de esquema no garantiza en absoluto la actitud de la aeronave y, por ende, permite al piloto controlarla manualmente. Como se ve en la figura 5.12, el usuario proporciona entradas de velocidad angular en cada eje para controlar la aeronave, es decir, que el piloto libera los controles (dejando todos en 0) la aeronave intentará no rotar

en ninguno de sus ejes. Como único conjunto de parámetros de configuración, se deben ajustar las velocidades angulares máximas en cada eje que se desea obtener cuando el piloto utiliza al máximo cada uno de los controles.

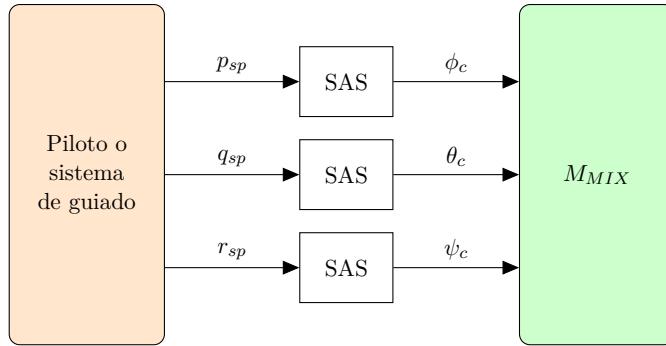


Figura 5.12: Diagrama en bloques del sistema de control para el modo de vuelo acrobático

En el segundo modo de vuelo, denominado modo “estabilizado”, se agregan dos CAS para poder mantener una actitud determinada en los ejes de pitch y roll. En este caso, el piloto ingresa ángulos de actitud para estos ejes y de velocidad angular para el eje de yaw como se puede observar en la figura 5.13. Esto resulta en un modo sencillo de utilizar ya que permite que el quadrotor gire libremente en el eje de yaw. No así en los ejes de pitch y roll. En este modo, el usuario debe configurar los ángulos máximos en los que se puede inclinar el quadrotor en los ejes de pitch y roll y la velocidad máxima que se puede lograr en el eje de yaw. También es necesario configurar los límites de velocidad angular que este puede imponer sobre los SAS de pitch y roll respectivamente.

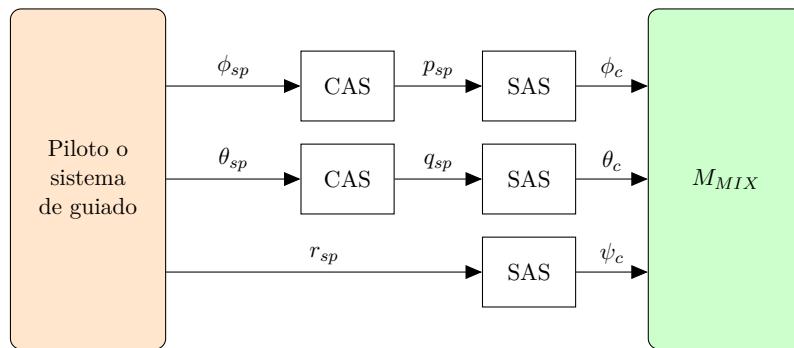


Figura 5.13: Diagrama en bloques del sistema de control para el modo de vuelo “stabilized”

Por último, se puede plantear también el tercer modo de vuelo, denominado “actitud”. En este se agrega al modo de tail lock un CAS para controlar la actitud en yaw. Esto permite que el usuario indique hacia donde debe apuntar el quadrotor mediante la entrada de control, como se observa en la figura 5.14. Este modo, si bien no resulta práctico para el usuario, ya que él debe mantener todo el tiempo el control de yaw en la posición deseada, suele ser el más utilizado en aplicaciones de seguimiento de trayectorias o en sistemas de guiado. Por otro lado, este modo requiere de un componente extra en el sistema de navegación que es una referencia de norte externa, como ser una brújula electrónica o un gps ya que el eje de yaw suele ser el más propenso a la deriva a lo largo del tiempo [30].

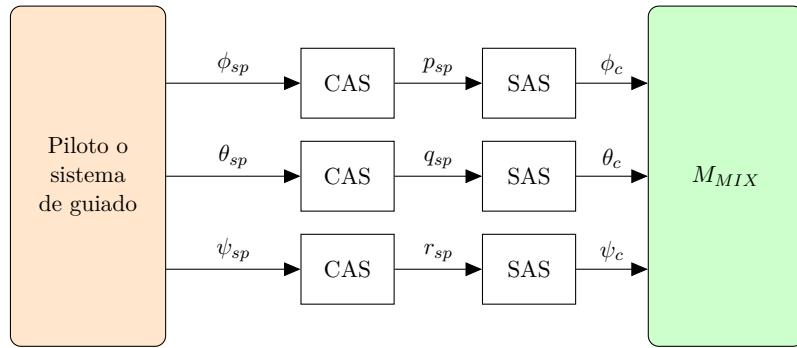


Figura 5.14: Diagrama en bloques del sistema de control para el modo de vuelo “atti”.

Como se observa en los gráficos 5.12, 5.13 y 5.14, la señal de control colectivo Z_c no se incluyó en estos diagramas ya que en este trabajo no se implementó un controlador de altura por estar fuera de alcance. Es por eso que se optó por dejar este control en forma manual y deberá ser controlado por un piloto para lograr mantener el quadrotor a cierta altura y así evitar el efecto suelo mencionado en el capítulo 2.1. De implementarse un control de altura utilizando un sensor barométrico o de distancia, es posible implementar también un sistema tipo SAS-CAS para controlar por un lado, la aceleración vertical y luego, la altura absoluta.

5.6. Limitaciones del modelo

El modelo matemático utilizado hasta aquí para estudiar al quadrotor se basa en una serie de hipótesis que permiten simplificar el problema hasta obtener una planta de primer orden. Si bien estas hipótesis son válidas para la condición de hovering, resulta interesante estudiar cuán robusto es el sistema de control para variaciones de los parámetros del modelo causado por variaciones en el quadrotor, como así también la influencia de los errores del sistema de navegación que hasta ahora se consideraron ideales.

5.6.1. Dependencias del modelo

Lo primero que se analizará es la respuesta del sistema ante cambios en los diferentes parámetros del modelo. En general los problemas que puede traer aparejado un cambio en algún parámetro se refleja en cambios en la dinámica de la planta a lazo cerrado. Por ejemplo, un aumento en la constante K_{T_ω} por utilizar actuadores más potentes. Se analizarán diferentes casos por separado en las siguientes secciones para determinar cualitativamente estas dependencias.

5.6.1.1. Dependencia del tensor de inercia

Si por ejemplo se simula el SAS para el eje de yaw con momentos de inercia variando entre un 5 % y 150 % del valor nominal, se obtienen las diferentes respuestas para una entrada escalón de velocidad angular.

Como se puede observar en la figura 5.15, una variación de ±50 % en el momento de inercia del eje no afecta de manera considerable la dinámica. Pero, si el momento de inercia del eje disminuye considerablemente, entonces, el sistema comienza a variar su dinámica. En una primera instancia empezarán a aparecer sobrepicos hasta que finalmente, el sistema puede volverse oscilatorio o divergente como es el caso de una reducción del tensor de inercia al 5 % del valor original del trabajo.

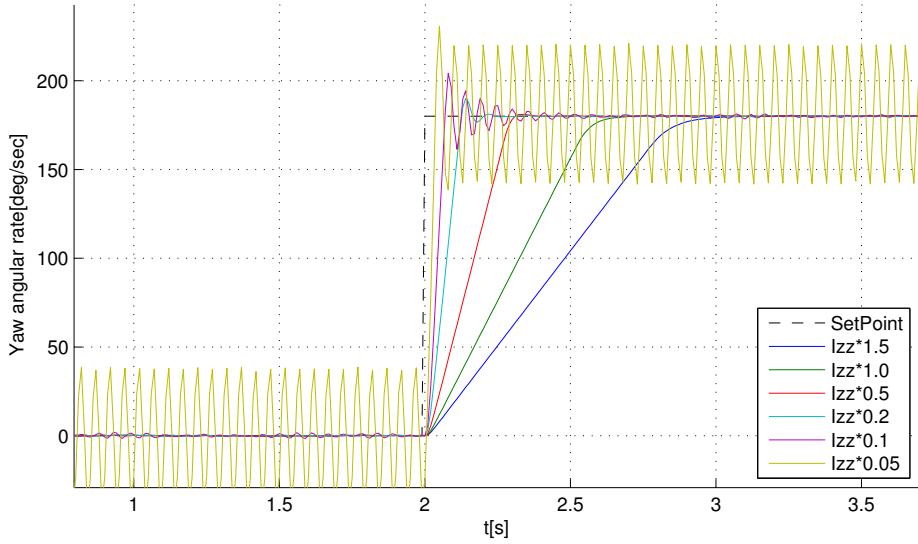


Figura 5.15: Respuesta del SAS ante variaciones del momento de inercia I_{zz}

Resulta interesante evaluar qué sucede si a este mismo sistema se le agrega el CAS para controlar la actitud. En este caso, como se puede ver en la figura 5.16, el sistema es tolerante a cambios en la inercia en comparación con utilizar solo el SAS.

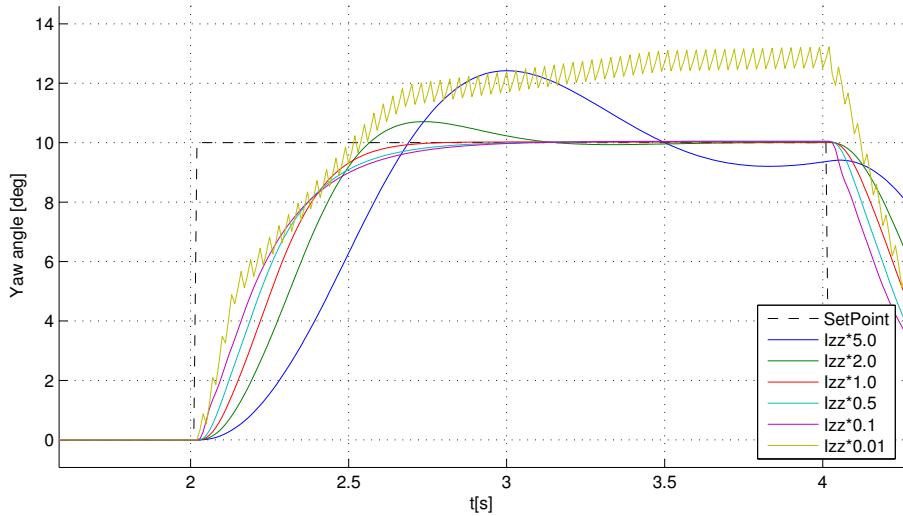


Figura 5.16: Respuesta del CAS+SAS ante variaciones del momento de inercia I_{zz}

Como se observa en estos resultados, si bien el sistema tiene sus límites en cuanto al tensor de inercia con el que es compatible, éste resulta ser suficientemente robusto como para permitir variaciones considerables en el quadrotor. Por supuesto, si se desea la misma dinámica para dos quadrotores diferentes, entonces se deberán ajustar los controladores.

5.6.1.2. Dependencia de los actuadores

En el caso de una variación en la ganancia de los actuadores, es decir la fuerza y momento que generan a una misma entrada de control, el efecto que se observa es inversamente proporcional al efecto de cambiar el momento de inercia en ese eje. Un aumento de 10 veces en la constante $K_{T\omega}$ es equivalente a una disminución de 10 veces en el momento de inercia en ese eje. Esto es apreciable en la ecuación del eje 5.3 por ejemplo.

Pero en cambio, el tiempo de crecimiento de la respuesta del actuador resulta un punto clave en la selección del mismo. Como se vio en el capítulo 2, se encontró que el tiempo de crecimiento de motor estaba en el orden de los 0.05 segundos. Tiempos más bajos resultan en un actuador cada vez más rápido que tiende al actuador ideal en donde la acción de control es instantánea. Pero incrementar este valor de manera considerable resulta en un deterioro de la respuesta dinámica del sistema, como se puede observar en la figura 5.17.

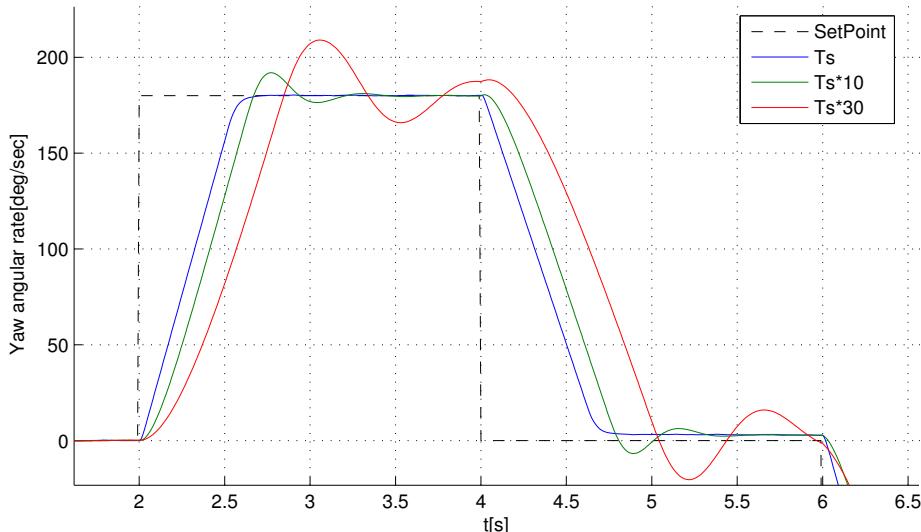


Figura 5.17: Influencia del tiempo de crecimiento del motor en la dinámica de la planta con SAS

Otro aspecto importante a tener en cuenta en la elección del actuador resulta ser la frecuencia de muestreo del mismo. Se ha demostrado empíricamente que pasar de actuadores con una frecuencia de sampleo en su señal de entrada de 200Hz a 500Hz mejora considerablemente la performance de vuelo en hovering [11]. Para este trabajo se optó por utilizar variadores comerciales con una frecuencia de actualización de 400Hz a fin de evitar los problemas que puede traer asociado una baja frecuencia de actualización.

5.6.2. Dependencia del sistema de navegación

Un aspecto fundamental a considerar es el error que se comete al obtener las velocidades angular y la actitud de la aeronave ya que la medición de las variables a controlar afectan directamente la performance del sistema. En las siguientes secciones se analizarán brevemente algunos factores que influyen considerablemente.

5.6.2.1. Representación de las rotaciones

Una rotación en un espacio tridimensional puede ser representada de diversas formas, por ejemplo, mediante matrices de rotación, ángulos de Euler, vector de dirección y rotación o quaterniones, y en general cada tipo de representación tiene ventajas y desventajas que la hacen adecuada para diferentes problemas. Como el objetivo de este trabajo es realizar controladores independientes por cada eje se eligió como método de representación los ángulos de Euler que son de fácil comprensión y naturalmente representan tres ángulos discociados. Cada controlador entonces debe controlar un ángulo sin importar el estado de los demás, condición válida solo para pequeños desvíos y para la maniobra de hovering. En cualquier otro caso es fundamental el conocimiento de la rotación completa en el espacio tridimensional y, por ende, la utilización de ángulos de Euler no sería correcta. El primer problema que traen asociados las rotaciones es la no conmutatividad. Es decir, que si un objeto es rotado en sus tres ejes pero en diferente orden, el resultado no es el mismo. Esto se puede ver en la figura 5.18. Por lo que, conocer el ángulo

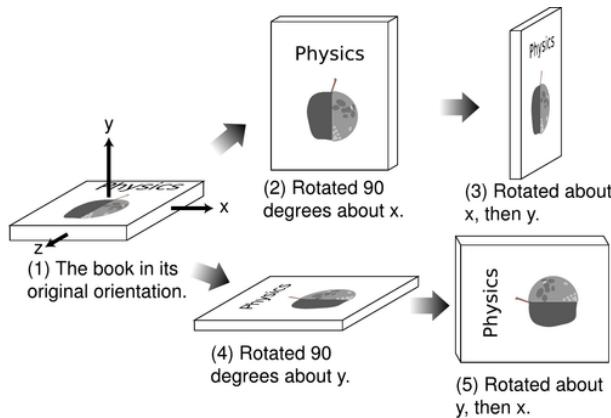


Figura 5.18: No conmutatividad de las rotaciones

de rotación en los tres ejes sin conocer el orden en que fueron calculados puede llevar a una orientación equivocada. De manera inversa, si se desea que el quadrotor tenga una determinada actitud en el espacio, es necesario determinar el orden o la forma en la que se efectuarán las rotaciones para llegar a ese punto, existiendo siempre más de una solución posible. Este problema es resuelto comúnmente mediante la utilización de quaternios para representar las rotaciones. Estos son una extensión de los números complejos y mediante cuatro elementos permiten representar una rotación en el espacio de manera inequívoca. Es por eso que en general las IMUs, de calcular internamente la actitud, suelen entregar este dato en forma de quaternion como es el caso de la IMU integrada MPU6050 que se utilizó en la computadora de vuelo de este trabajo. Dado que esta representación es incompatible con el sistema de control desacoplado por ejes, se debió convertir de quaternios a ángulos de Euler utilizando la transformación:

$$\begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} = \begin{bmatrix} \text{atan2}(2(q_0q_1 + q_2q_3), 1 - 2(q_1^2 + q_2^2)) \\ \arcsin(2(q_0q_2 - q_3q_1)) \\ \text{atan2}(2(q_0q_3 + q_1q_2), 1 - 2(q_2^2 + q_3^2)) \end{bmatrix} \quad (5.11)$$

donde q_0, q_1, q_2 y q_3 son los elementos del quaternion. Además, surge un problema conocido como “Gimbal lock” que se debe a la perdida de un grado de libertad en la rotación. Esto ocurre en sistemas mecánicos como cardán pero también matemáticamente, y se ve reflejado que rotaciones grandes en un eje empiezan a verse acopladas en otros. Esto sucede para ángulos

cercanos a los 90° aunque su efecto se comienza a ver en ángulos ligeramente menores también. Dado que en este trabajo se busca siempre estar cerca de los 0° en todos los ejes, este efecto resulta despreciable.

5.6.2.2. Errores en la actitud

Por último, pero no menos importante, resulta interesante estudiar la estabilidad del sistema respecto a errores en la actitud. En general, los sensores como giróscopos o acelerómetros, como así también sistemas integrados como las IMU, tienen dos tipos de errores que los caracterizan: el bias y el factor de escala [31]. El primero representa un corrimiento en el valor sensado mientras que el segundo representa un error proporcional en la medición.

Como en este trabajo se busca realizar un controlador para hovering, en donde las velocidades angulares y la actitud siempre estarán cercanas a cero, el factor de escala no representa un problema importante. Por supuesto lo que ocurrirá es que si el controlador CAS impone una rotación a $90^\circ/s$ esta podrá estar ocurriendo realmente a más o menos que ese valor, lo que cambiaría ligeramente la dinámica pero el objetivo real que es volver a los 0° , se cumpliría igual. En la simulación de la figura 5.19 se observan los resultados de un experimento en donde se simuló el eje de yaw con CAS+SAS que es perturbado para alejarlo de la condición de hovering. A este se le agregó también ruido que proviene de las imperfecciones del modelo y que es similar al obtenido en las pruebas de vuelo.

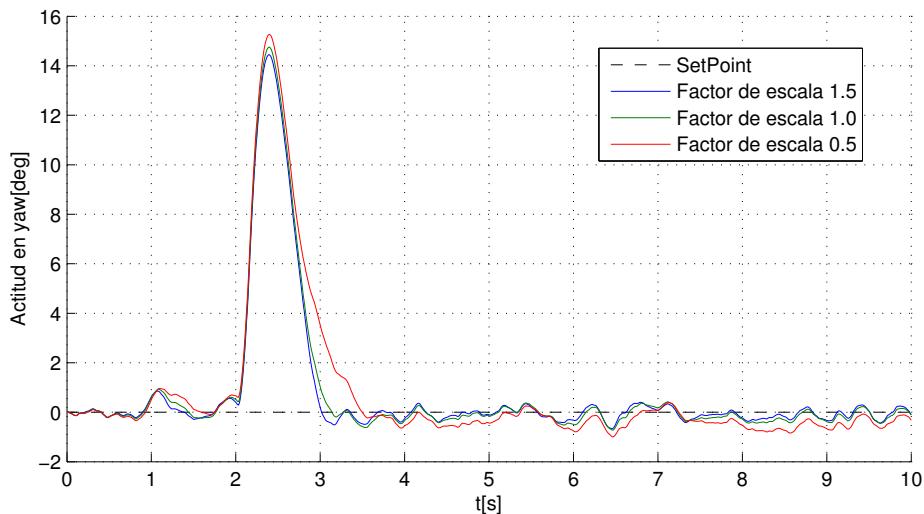


Figura 5.19: Simulación del control de hovering con perturbaciones para diferentes valores en el factor de escala de la IMU

Se puede observar que el sistema responde de manera muy similar para factor de escala con un $\pm 50\%$ de error y que pese a esto, el sistema puede volver a la condición de hovering sin mayores problemas. Notar que un error del 50% es un valor absurdo para IMUs comerciales y solo se lo utilizó como ejemplo.

Por otra parte, también se estudió el efecto del bias en la medición. Esto como es esperable introducirá un corrimiento en la señal medida que el sistema de control no podrá eliminar e intentará ajustar a cada eje según este valor erróneo. Se puede ver en la figura 5.20 en donde el sistema, con una referencia de 0° no logra nunca llegar a este punto realmente (aunque internamente su señal de error le indica que sí llegó al “setpoint”).

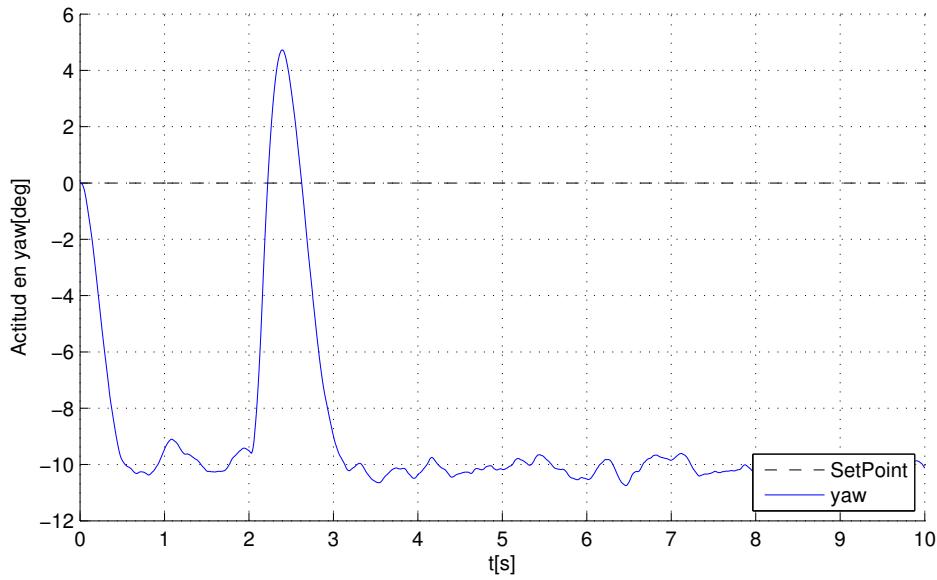


Figura 5.20: Simulación de la influencia del bias en los sensores de actitud durante un vuelo en hovering. También se lo perturbo para entender la respuesta a esta con efecto del bias.

Esta es sin duda la limitación más importante del sistema. Dado que a lo largo de las pruebas se obtuvieron resultados muy buenos utilizando la IMU de la computadora de vuelo y que el análisis o mejoras sobre este problema estaban fuera de los alcances del trabajo se optó por no considerar la influencia del bias en los experimentos. Como con el tiempo el bias suele incrementarse, es necesario que un piloto externo compense el error y dado que este puede ser considerado un lazo externo de control con realimentación visual, esto es fácilmente corregible. De existir un sistema de guiado que controle automáticamente la posición del quadrotor, este bias se vería fácilmente corregido ya que si este sistema intentase dejar al quadrotor en alguna posición determinada y este se corre por estar inclinado debido al bias, el sistema de guiado debería corregir la actitud y forzarle una referencia de igual valor pero signo contrario al bias para mantener a la aeronave en posición horizontal.

5.7. Resumen del capítulo

En este capítulo se presentó la estrategia de control SAS+CAS que se utiliza en cada eje del quadrotor. A lo largo de este, se mostró que los resultados de las simulaciones del sistema de control coinciden con las pruebas de banco realizadas sobre el modelo real. Por otra parte, el análisis de las limitaciones del sistema de control determinó que, salvo en errores de biasing en la IMU, el sistema es tolerante a variaciones y errores de diversos tipos, lo que lo hace una solución viable para este problema.

En el capítulo siguiente se estudiará la solución propuesta en condiciones de vuelo reales.

CAPÍTULO 6

Resultados

Los experimentos que se presentan a continuación fueron realizados con la computadora de vuelo presentada en el capítulo 4 y en todos los casos el quadrotor completo fue probado en condiciones de vuelo real. Dado que el objetivo del trabajo es únicamente el control de actitud para mantener la condición de hovering, el control de altura se realizó de manera manual mediante un joystick en el segmento de tierra. Se logró así que el quadrotor despegue, realize el experimento y luego aterrice sin mayores inconvenientes.

En las siguientes secciones se presentan los resultados de estos experimentos donde los datos presentados fueron descargados en tiempo real por el sistema de telemetría por radio. Es posible encontrar los videos de muchas de estas pruebas en el siguiente vínculo: <http://laboratorios.fi.uba.ar/lse/tesis.html>

6.1. Modos de vuelo

Para comparar la performance general del sistema utilizando los diferentes modos de vuelo se realizó un experimento en el que se hizo despegar al quadrotor y luego se lo mantuvo a una altura constante durante algunos segundos, hasta que finalmente se lo aterrizó. Durante el vuelo solo se efectuaron acciones correctivas en la señal Z_C para mantener la altura estable, pero no se modificaron los setpoints de ninguno de los controles de roll, pitch ni yaw, que se mantuvieron durante todo el vuelo en 0° . En la figura 6.1 se pueden ver los resultados del angulo de yaw en cada uno de los tres casos.

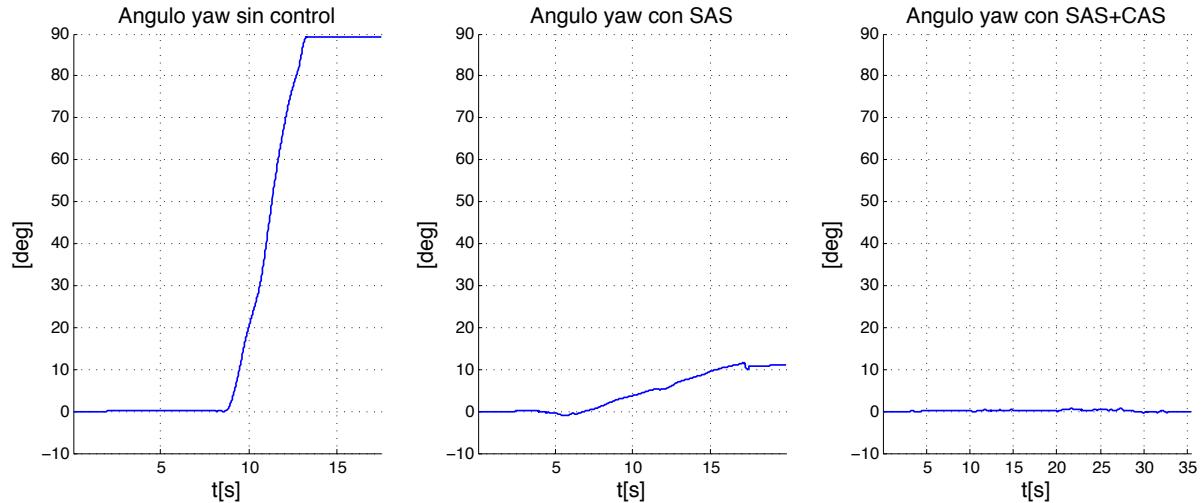


Figura 6.1: Comparación de los diferentes modos de control en una maniobra de despegue, hovering y aterrizaje con todas las referencias en 0°

En el primer vuelo se desactivó por completo el control de yaw, es decir que, sobre este eje, no había ni control de velocidad angular ni de ángulo. En cambio, en el experimento, si se controlaron los ejes de pitch y roll, con el objetivo de mantener al quadrotor en posición horizontal. Se puede ver que el quadrotor rota a velocidad casi constante sobre el eje de yaw, acumulando en pocos segundos aproximadamente unos 90° . Luego en el segundo experimento, se activó el SAS en yaw, lo que sería equivalente al modo “acrobático” y esto redujo la deriva de este eje notablemente. Como en experimentos anteriores, el pequeño error acumulado en la velocidad angular hizo que el quadrotor tenga una deriva acumulada de unos 10° durante todo el vuelo. Finalmente en el último de los vuelos se activó el modo estabilizado en donde se utilizó el SAS y el CAS. Se observa que el angulo de yaw se mantiene muy cercano a los 0° durante todo el vuelo, inclusive durante el aterrizaje y el despegue.

En la figura 6.2 es posible ver un detalle en otra escala de este último resultado en donde se puede apreciar que la deriva en este eje es menor a 1° durante todo el vuelo. Se concluye así que los modelos de vuelo funcionan de acuerdo a lo esperado. A continuación se realizará un análisis más detallado de los resultados en cada uno de estos modos de vuelo.

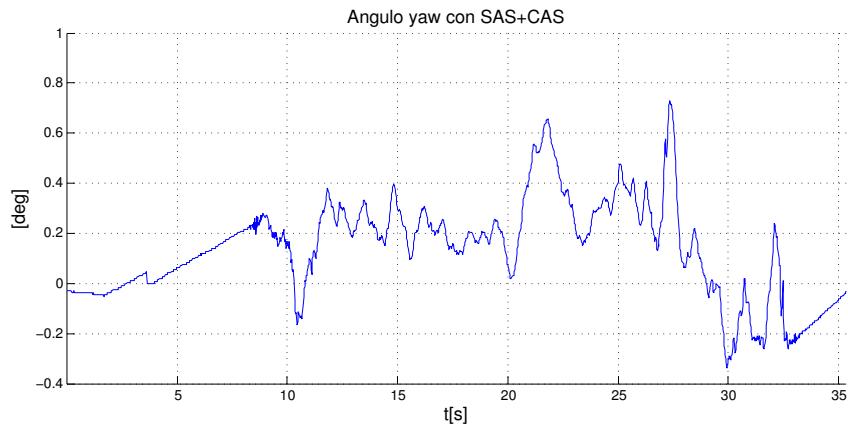


Figura 6.2: Detalle del angulo de yaw con SAS+CAS correspondiente a la figura 6.1

6.2. Vuelo en modo “acrobático”

En el modo de vuelo acrobático el joystick controla las velocidades angulares en cada uno de los ejes del quadrotor. En la figura 6.3 se presenta la telemetría de un vuelo de prueba para probar el este modo.

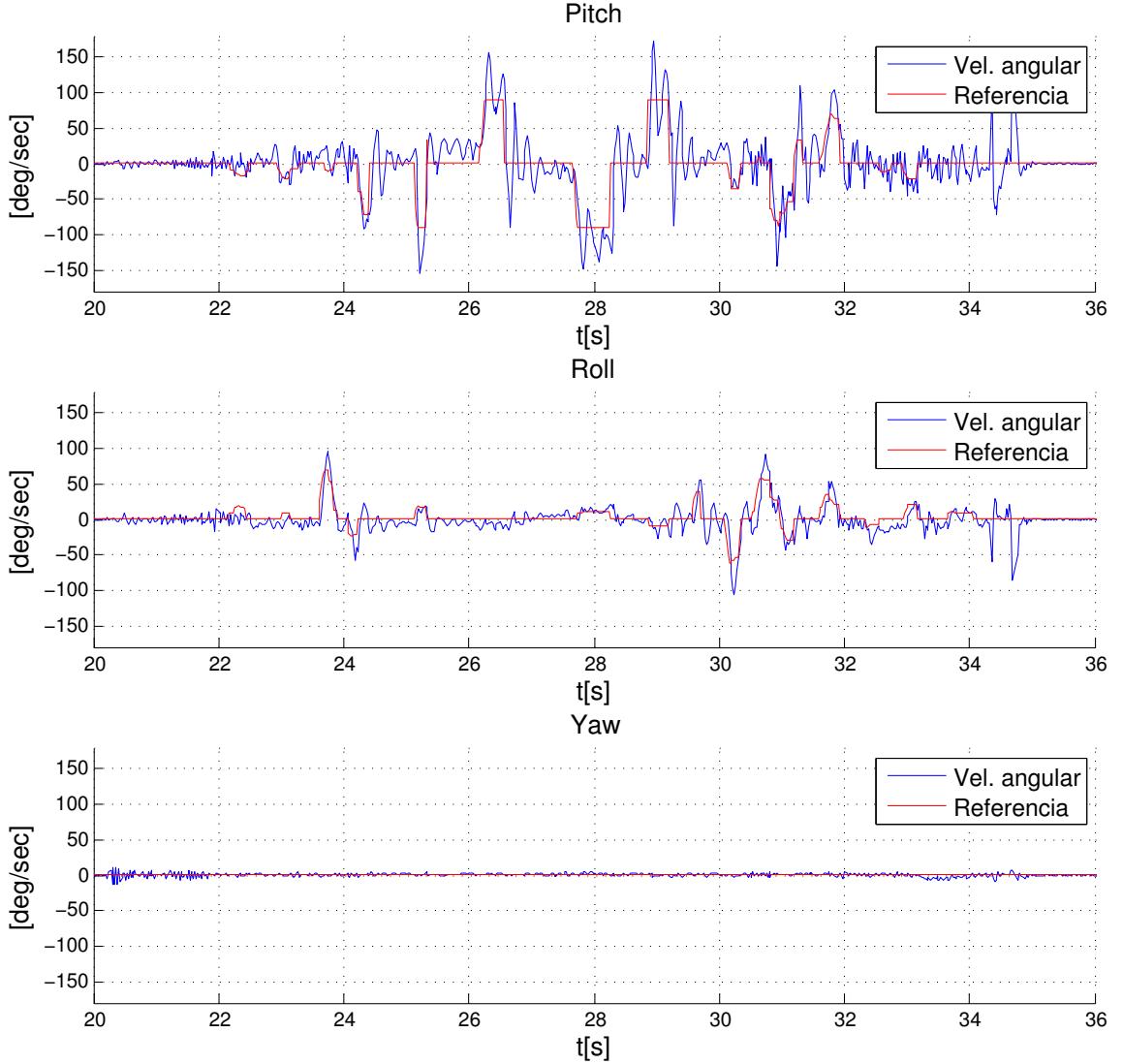


Figura 6.3: Telemetría de un vuelo en modo acrobático que muestra las velocidades angulares y las referencias introducidas por el piloto.

En este en se pueden ver las velocidades angulares de cada uno de los ejes junto con las referencias impuestas por el piloto. Durante el experimento se puso a volar al quadrotor a una altura de 2m del piso y una vez en hovering se introdujeron diferentes referencias de manera manual para mover al quadrotor de un lado al otro. El eje de yaw se mantuvo durante todo el experimento fijo en $0^\circ/s$.

Se puede observar que el quadrotor sigue de manera aproximada las referencias en los ejes de pitch y roll. Para mantener la cohesión del gráfico los tres subgráficos se muestran en la misma escala, pero en el eje de yaw, la velocidad angular se mantuvo siempre dentro de los $\pm 5^\circ/s$ pese a

los grandes desplazamientos y velocidades obtenidos durante el vuelo. Esto también es causan de las señales poco suaves de los ejes de pitch y roll. En este caso el quadrotor se mantuvo fuera de la condición de hovering en casi todo el vuelo. También se debe tener en cuenta que los actuadores producen una vibración considerable en la estructura, y al estar la computadora junto con la IMU las vibraciones son medidas por el giróscopo, lo que introduce cierto nivel de ruido. A partir de este experimento se concluye que el sistema es capaz de desacoplar perfectamente la dinámica de los tres ejes, incluso en condiciones que no son exactamente de hovering, validando así el modo de vuelo acrobático.

6.3. Vuelo en modo “estabilizado”

En el modo de vuelo estabilizado el joystick controla las referencias de angulo para los ejes de pitch y roll mientras que controla la velocidad angular en el eje de yaw. En el siguiente experimento se realizó una prueba en un entorno cerrado libre de perturbaciones en donde se hizo despegar a la aeronave, luego mantenerse en hovering por unos segundos y finalmente aterrizar. Las referencias de los ejes de pitch y roll se ajustaron ligeramente para mantener a la aeronave en posición. En la figura 6.4 se pueden observar los resultados de este experimento.

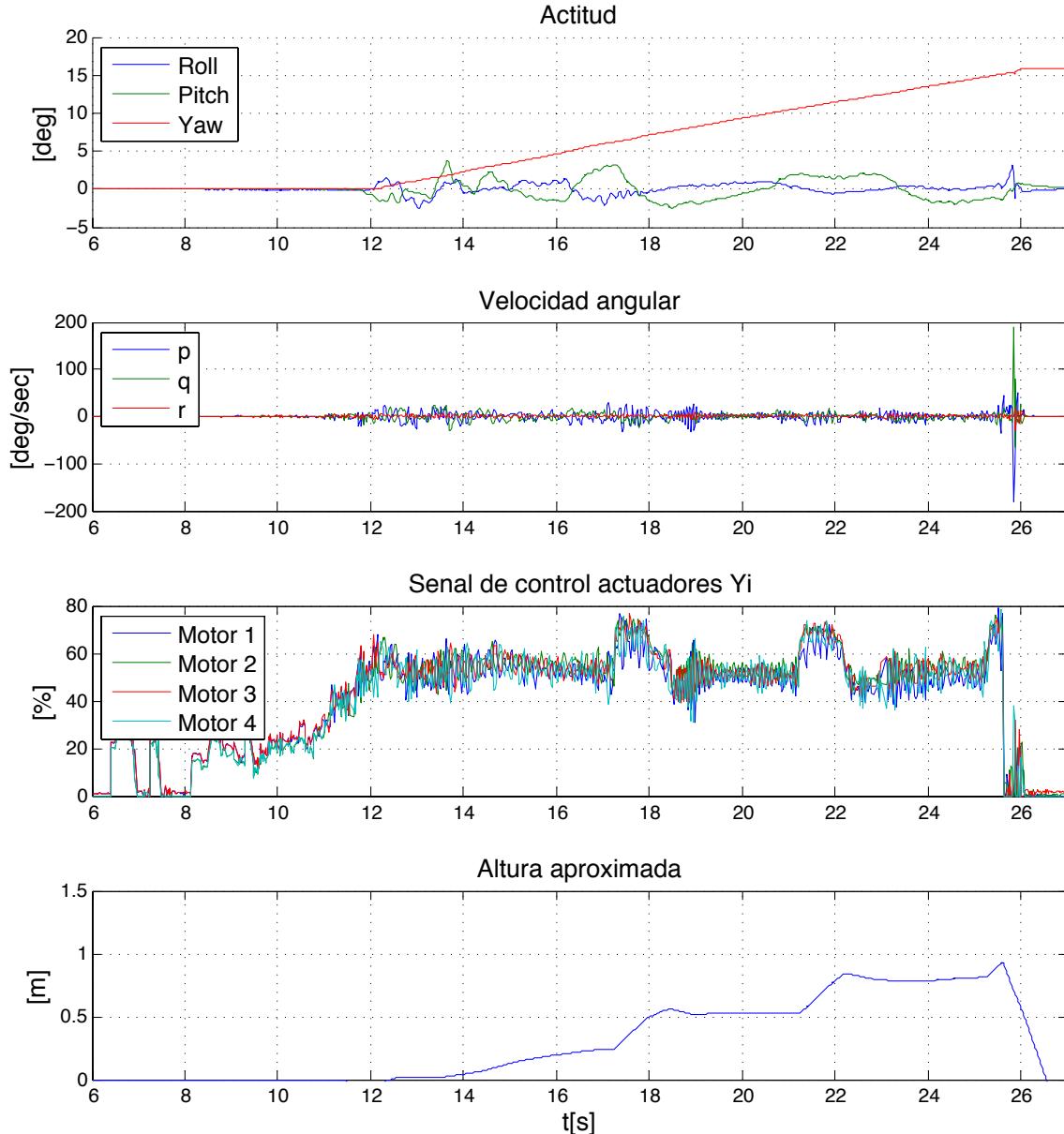


Figura 6.4: Telemetría de un vuelo en hovering sin perturbaciones. El gráfico de altura fue calculado posteriormente y solo se presenta de manera ilustrativa.

Se puede ver en el subgráfico superior la actitud del quadrotor, luego las velocidades angulares en cada eje seguido de las señales de control y_i que reciben los actuadores y por último una

estimación de la altura aproximada que alcanzó el quadrotor en vuelo. Esta último subgráfico se calculó luego del vuelo a partir de los valores que tomaron las entradas de los actuadores, su ganancia y la masa de la aeronave y se presenta solo a modo ilustrativo para que el lector pueda visualizar mejor el experimento.

El subgráfico inferior no representa la altura sino la acción de control sobre cada uno de los motores. El valor medio de estas señales es el valor comandado por la acción de control manual Z_C , mientras que la variación sobre el valor medio es la acción que imponen los controladores SAS y CAS según corresponda en cada eje. En el subgráfico inferior también es posible observar hasta $t = 12s$ la maniobra de despegue, luego hovering entre $t = 12s$ y $t = 25s$, donde ocurren algunos cambios de altura, y luego a partir de $t = 25s$ la maniobra de aterrizaje.

El sistema de control de actitud CAS en los ejes de pitch y roll mantiene la actitud de la aeronave dentro de un margen de $\pm 5^\circ$ durante todo el vuelo, inclusive en el despegue y aterrizaje. En cambio, el eje de yaw presenta un corrimiento temporal que luego de los 15 segundos de vuelo alcanza los 15° aproximadamente. Es importante recordar que esto se debe a que no se activó el control CAS en el eje de yaw y por ende mínimas variaciones de velocidad angular a lo largo del tiempo producen un corrimiento en el ángulo. Pese a esto, el error es de aproximadamente $1^\circ/s$ que es un valor muy acotado. Por otra parte, de este gráfico es posible validar la elección del punto de trabajo en el 55% ya que si se observa el período entre $t = 14s$ y $t = 17s$ se obtiene que el valor promedio de la acción de control (establecido de manera manual para mantener a la aeronave a altura constante) es de 54.3%. También resulta interesante analizar lo que ocurre con las velocidades angulares de cada eje. Como se observa, las velocidades angulares en pitch y roll se encuentran dentro de un rango de $\pm 20^\circ/s$, mientras que el eje de yaw dentro de los $\pm 5^\circ/s$, lo cual es consistente con los experimentos en banco de pruebas presentados en el capítulo 5.

6.4. Vuelo en modo “actitud”

Luego de comprobar que el sistema efectivamente puede mantener a la aeronave en hovering se realizaron experimentos de seguimiento de referencias en modo “actitud”. En la figura 6.5 es posible observar un recorte de la telemetría del vuelo en donde se mantuvo a la aeronave volando a una altura de aproximadamente 2m en forma manual, y luego mediante el joystick se le varió la referencia de ángulo en el eje de yaw donde se puede observar que el quadrotor sigue de manera precisa las referencias impuestas externamente.

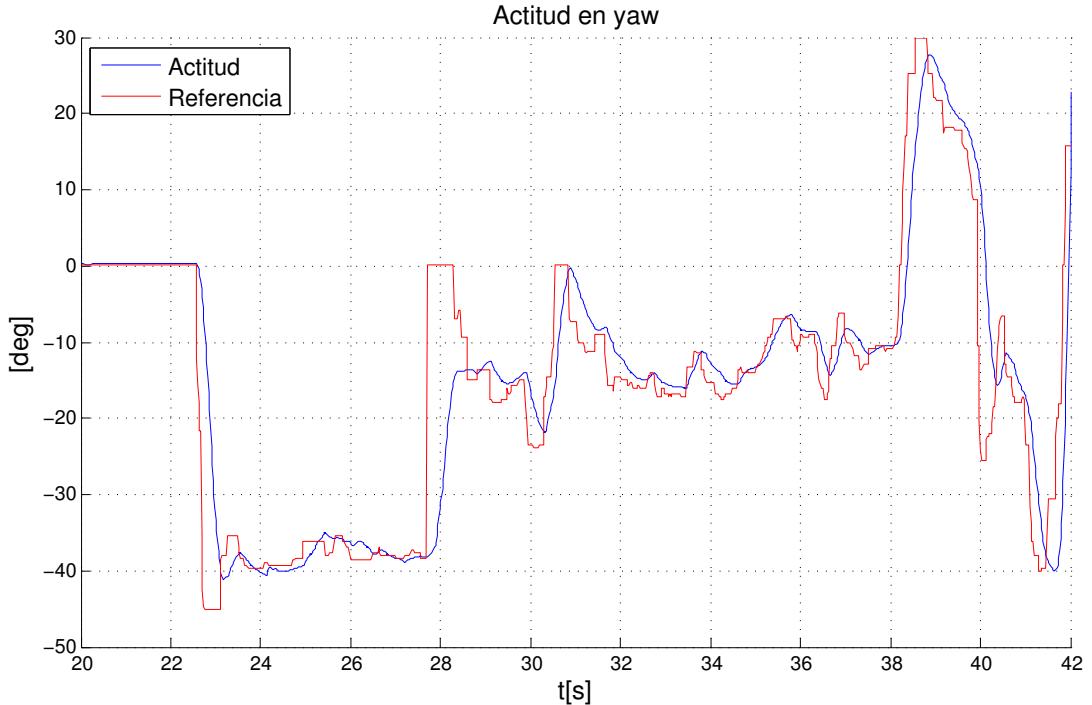


Figura 6.5: Telemetría de un vuelo en hovering con referencias controladas por joystick.

También resulta interesante realizar este experimento en los ejes de pitch y roll, pero el hecho de hacer esto trae acompañado ciertos inconvenientes: el primer efecto de esto es que la aeronave comienza a trasladarse, perdiendo la condición de hovering y aumentando los efectos aerodinámicos causados por el viento aparente, y el segundo es que la aeronave pierde sustentación al utilizar parte del empuje en avance dado que el Z_C se mantiene constante, pero el empuje de los actuadores ya no se encuentra perfectamente orientado en dirección vertical. Esto, como era de esperar, saca al quadrotor de la condición de hovering y en consecuencia ya no es válido el modelo matemático desarrollado. Si bien esto no es condición necesaria para determinar que el sistema dejará de ser estable, no es posible evaluarlo con los modelos disponibles hasta el momento.

En la imagen de la figura 6.6 se puede observar una composición de imágenes tomadas cada 1/10 de segundo mediante una cámara con lente tipo ojo de pez. Se observa cómo la aeronave parte de hovering hasta alcanzar el valor de referencia de 40° en el eje de pitch y luego comienza a desplazarse a la vez que desciende. Las imágenes utilizadas no están equiespaciadas en tiempo por lo que no se debe entender a la velocidad del quadrotor como la distancia entre tomas. Solo se seleccionaron aquellas que demuestran la idea de manera clara. A partir de este experimento es posible determinar que los controladores diseñados logran controlar la actitud en el eje de

pitch. Mediante un ensayo similar se comprobó que el controlador también es capaz de controlar la actitud en el eje de roll.



Figura 6.6: Composición de imágenes de un vuelo que comienza en hovering y luego se introduce una referencia en el pitch de 40° aproximadamente

6.5. Hovering con perturbaciones en modo “estabilizado”

Otro de los objetivos del control de actitud es tolerar perturbaciones externas. En general estas se deben al viento y su efecto sobre la geometría del quadrotor. A fin de evaluar el desempeño se mantuvo al quadrotor en hovering a una altura controlada de aproximadamente 2 metros y se lo golpeó desde abajo con la mano, aplicando así perturbaciones considerables. En la figura 6.4 se presenta un recorte de la telemetría en donde las perturbaciones ocurren a los $t = 47,2\text{s}$ y $t = 50,8\text{s}$ segundos aproximadamente. Se muestra en la primera subfigura la actitud del quadrotor en todos sus ejes, luego la salida del control del SAS y por último la del CAS. Se observa que ante la ausencia de perturbaciones, el sistema de control de actitud CAS

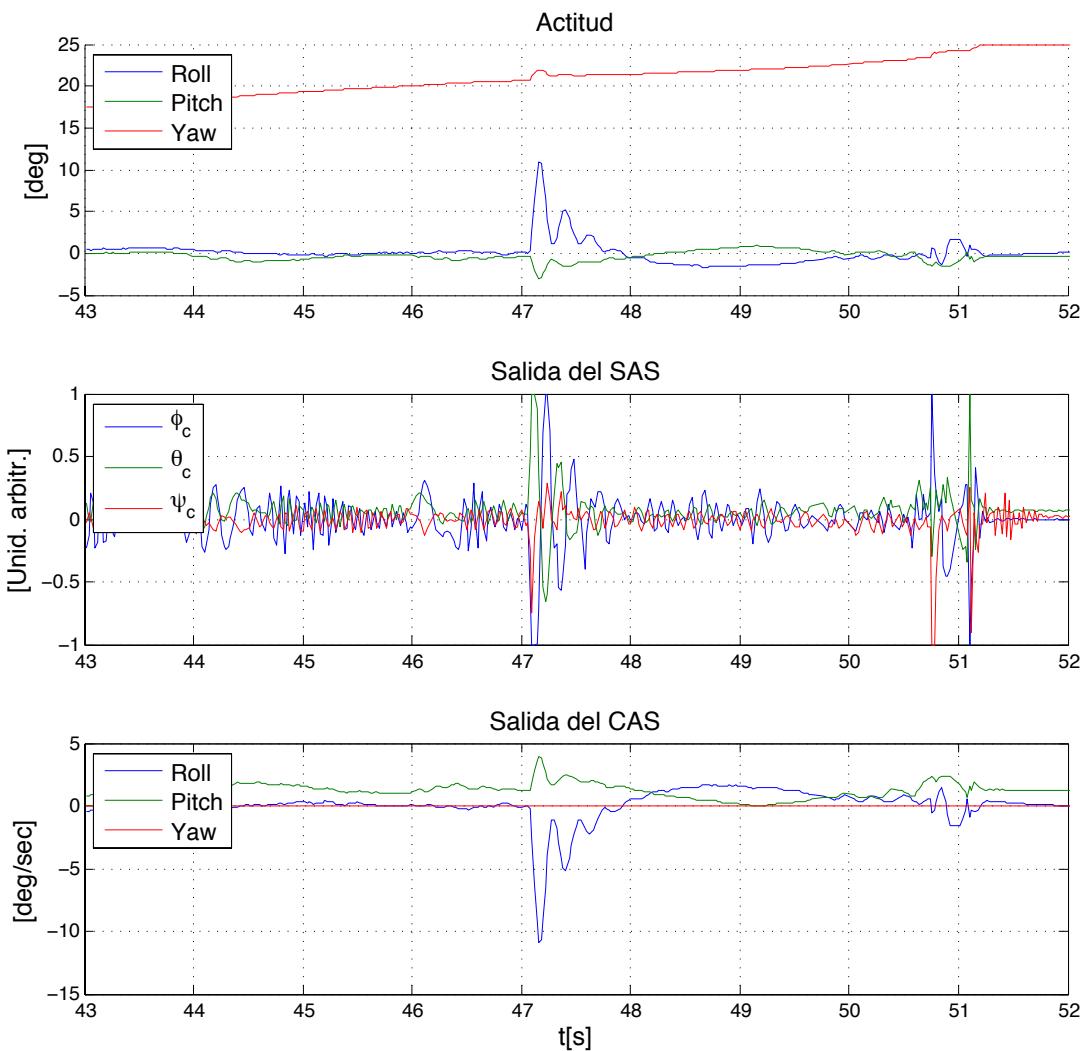


Figura 6.7: Telemetría de un vuelo en hovering con perturbaciones a los $t = 47,2\text{s}$ y $t = 50,8\text{s}$

logra mantener los ejes de pitch y roll en un rango de $\pm 2^\circ$. También es posible observar que el eje de yaw al solo tener SAS tiene una pequeña deriva causada por la acumulación de velocidad angular muy pequeña, pero que no afecta el vuelo. Luego, al momento de las perturbaciones el controlador CAS actúa corrigiendo la actitud en menos de 1 segundo. Esto demuestra que inclusive ante golpes, que son en general perturbaciones mucho más fuertes que las que puede

6.5. HOVERING CON PERTURBACIONES EN MODO “ESTABILIZADO”

sufrir el quadrotor en vuelo, el sistema de control mantiene al quadrotor en una posición de hovering estable.

Tamén se realizaron pruebas en entornos abiertos con excelentes resultados. En la figura 6.8 es posible ver una fotografía de una de estas pruebas donde el quadrotor vuela por encima de campos de soja a una altura de aproximadamente 20 metros en un día con vientos considerablemente fuertes que continuamente perturbaban a la aeronave.



Figura 6.8: Fotografía de vuelo en altura. Es posible ver el video de un vuelo similar en modo acrobático en <http://www.youtube.com/watch?v=ErMcxXlnN10>.

En estas pruebas se observó que el sistema es capaz de soportar estos vientos, manteniéndose en hovering, sin mayores problemas. En próximos trabajos se podría hacer un análisis más riguroso, para cuantificar la estabilidad del vuelo ante estas perturbaciones.

6.6. Computadora de vuelo

6.6.1. Consumo de recursos

Los experimentos que se presentaron en los capítulos 5 y 6 fueron realizados con la computadora de vuelo que se presentó en el capítulo 4. Esta demostró ser una plataforma apta para el control del quadrotor. En este capítulo se presentan resultados de desempeño de la computadora de vuelo que se pudieron medir a lo largo de este trabajo.

El software final de la computadora de vuelo cuenta con un sistema operativo de tiempo real FreeRTOS que ejecuta de manera concurrente las siguientes tareas:

- Seis lazos de control PID a 200 Hz cada uno (3 CAS + 3 SAS, todos en punto flotante).
- Adquisición de parámetros de navegación a 200 Hz.
- Recepción de setpoints vía radio a 50 Hz.
- Descarga de telemetría vía radio a 20 Hz.

Este firmware, incluyendo sistema operativo, drivers y aplicaciones de control, ocupó solo 79KB de la memoria no volátil de los 512KB disponibles en el sistema, es decir un 15% del total disponible. Por su parte el controlador PID resultó eficiente, considerando que la computadora de vuelo no dispone de unidad aritmética de punto flotante. Algunos resultados de la implementación del controlador PID son los siguientes:

- Tamaño de compilación del algoritmo: 984 Bytes de flash
- Tamaño del objeto controlador por cada instancia: 56 Bytes de RAM
- Tiempo de ejecución de cada instancia: aproximadamente $24 \mu\text{s}$

Es decir que la ejecución de los 6 lazos de control solo toma 124uS.

Por último para determinar la tasa de uso de CPU se utilizó la herramienta de análisis FreeRTOS+Trace durante una prueba de vuelo y se obtuvo el resultado que se presenta en la figura 6.9. En la figura se puede observar en colores las diferentes tareas y la cantidad de tiempo de CPU que consumen. El resultado muestra que, aún sin haber realizado esfuerzos para optimizar el código de vuelo, la utilización de CPU se encuentra por debajo del 30%, lo que demuestra que la computadora tiene capacidad suficiente para realizar tareas adicionales, como por ejemplo aplicaciones elementales de guiado o procesamiento de datos para navegación integrada. Si bien, se muestra el consumo de CPU solo para un fragmento del experimento, este resultado es representativo en general, ya que el tiempo de ejecución del controlador PID siempre es el mismo.

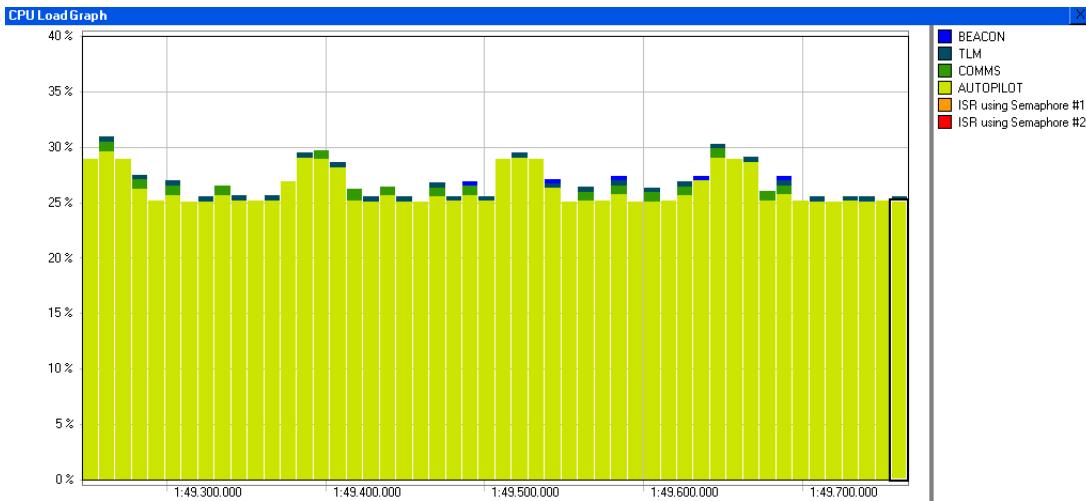


Figura 6.9: Uso de la cpu durante un vuelo

6.6.2. Costo estimado de fabricación

Finalmente, se presentan algunos datos indicativos del costo estimado de fabricación. Vale mencionar que el circuito impreso fue fabricado en el exterior, y también los componentes de la computadora de vuelo fueron comprados en el extranjero. Se debe tener en cuenta además que sólo se adquirieron los componentes electrónicos para dos de estas placas, por lo que los precios son poco representativos, ya que si se quisiera fabricar esta placa en forma masiva los costos bajarían considerablemente. Habiendo hecho estas salvedades se tiene que el costo de fabricación del circuito impreso por unidad, fabricando sólo diez placas, e incluyendo los costos fijos, fue de USD 14 por placa, mientras que el costo de los componentes fue de USD 127 por placa. Es decir que el costo total de cada computadora de vuelo, sin contar el montaje, es aproximadamente USD 150. A modo de referencia, vale decir que este importe es comparable con el precio de la computadora “Arducopter” [6], por ejemplo, disponible actualmente en el mercado. Se la comparó con este producto ya que el mismo es en la actualidad el más vendido en la categoría de “aficionados” y la misma tiene un desempeño muy similar al de la computadora de vuelo diseñada en este trabajo, pero tiene una limitada capacidad de expansión ya que la mayoría de sus recursos son utilizados para ejecutar el software que trae incorporado.

En el cuadro 6.1 se presenta una comparación entre los diferentes quadrotores comerciales y el desarrollado en este trabajo. Solo se presentan las características más destacables de cada uno de los modelos. Dada la falta de información disponible acerca de los quadrotores de la categoría “investigación”, estos no se incluyeron en la comparación. El modelo de Arducopter que se eligió, cuenta con el kit de telemetría que funciona de manera similar al implementado en este trabajo.

Se observa que el sistema desarrollado en este trabajo tiene características muy interesantes si se lo compara con varios de los quadrotores comerciales líderes del mercado. En particular, vale aclarar que el importe indicado de USD 376 corresponda a la computadora de vuelo, mas los motores, las hélices, los ESC, las baterías, la estructura y todo lo necesario para volar. El diseño de este trabajo es superior al Arducopter en cuanto a capacidad de procesamiento y memoria, y en cuanto a que corre un sistema operativo. El AR Drone tiene mayor poder de cálculo, pero es un sistema de código cerrado, mientras que el diseño presentado es de código abierto.

	Arducopter	Parrot AR-Drone	Phantom	Este trabajo
Procesador	ATMega 2560	TMS320DMC64x Cortex-A8	N/D	LPC1769 Cortex-M3
Ancho de palabra	8 bits	32 bits	N/D	32 bits
MIPS	16 MIP-S/16MHz	4000 MIP-S/500MHz	N/D	150 MPIS/120 MHz
Memoria	256 KB Flash, 8KB RAM	288KB RAM	N/D	512 KB Flash, 32 KB RAM
Radio	2.4 Ghz XBee	WiFi	2.4 GHz	2.4 GHz XBee
IMU DOFs	9 + Alt	9 + Alt	9 + Alt	9 + Alt
Telemetría	si	si	no	si
Código abierto	si	no	no	si
Sistema operativo	No	Linux	N/D	FreeRTOS
GPS/waypoints	Si	No	Si	No (possible)
Precio [USD]	765	299	699	376

Cuadro 6.1: Cuadro comparativo entre diferentes quadrotores comerciales y el desarrollado en este trabajo.

6.7. Resumen del capítulo

En este capítulo se presentaron los resultados de pruebas realizadas en los diferentes modos de vuelo y se realizó un análisis cualitativo del desempeño en cada uno de estos. Algunos de los experimentos se realizaron en condiciones de entornos cerrados, mientras que otros fueron realizados en entornos abiertos con perturbaciones. En todos los casos el sistemas de control demostró funcionar según lo esperado.

Por otra parte, si se observan los resultados de performance de la computadora de vuelo, se puede ver que esta es capaz de resolver el problema planteado utilizando menos del 30 % de su capacidad de computo. Esto abre las puertas a utilizar esta plataforma para trabajos futuros de mayor complejidad.

Por último se presentaron datos respecto al costo de computadora y del proyecto en general, y se los comparó con otros productos similares, obteniendo resultados muy alentadores.

CAPÍTULO 7

Conclusiones y trabajo futuro

7.1. Conclusiones

A lo largo de este trabajo se presentó el desarrollo de una computadora de vuelo capaz de controlar el vuelo en hovering de un quadrotor. Si bien es posible definir “vuelo” de muchas maneras, se impuso un objetivo claro y simple: lograr que la aeronave pueda mantenerse en hovering sin intervención de un piloto, pese a las asimetrías en las hélices, los motores, etc., a las perturbaciones del medio y otros factores. También fue uno de los objetivos, que la computadora diseñada pueda ser utilizada en trabajos futuros para aplicaciones de más alto nivel, incluyendo algoritmos de navegación y/o guiado elementales.

El diseño de la computadora no comenzó sino hasta la determinación de un modelo matemático del sistema y un posterior análisis estimativo de los requerimientos en cuanto a poder de computo necesario. Para desarrollar el modelo matemático se realizaron caracterizaciones experimentales que en muchos casos requirieron la construcción de herramientas adicionales. Finalmente se obtuvo un modelo del sistema para la condición de hovering de muy simple utilización ya que se consideró que en cierto entorno de uso era posible desacoplar la dinámica de los ejes. Este modelo fue implementado en un entorno de simulación desarrollado en MATLAB para estudiar su dinámica.

Para el diseño del sistema de control de actitud y velocidad angular de cada eje se propuso un sistema en cascada flexible que probó su efectividad. El primero de estos controladores, el SAS, es el encargado de estabilizar la planta en lo que se refiere a la velocidad angular. Luego el controlador CAS comanda al SAS imponiendo velocidades angulares que hacen posible seguir una referencia de ángulo. Este sistema permite utilizar al quadrotor en diferentes modos de vuelo. Estos controladores fueron implementados internamente mediante PIDs que para su mejor funcionamiento incorporan protección “anti-windup”, transición “bumpless”, etc.

Finalmente se desarrolló la computadora de vuelo, que incluyó el microprocesador, los sensores de navegación, la radio y las salidas de control necesarias para sentir y controlar los diferentes aspectos de la aeronave. Sobre esta se implementaron los diferentes PIDs digitales que fueron escritos en lenguaje C. El firmware de la computadora se desarrolló sobre un sistema operativo de tiempo real que permite fácilmente el diseño de software multitarea, como así también el fácil agregado de nuevas funcionalidades. Esto resultó una novedad, ya que ningún otro sistema de control de quadrotores de código abierto utiliza un sistema operativo como base de su

arquitectura de firmware.

Se implementaron diferentes modos de vuelo que resultaron de gran utilidad y se pudo probar por separado la efectividad del SAS y luego del sistema completo CAS+SAS. Si bien el sistema original fue diseñado para “hovering”, mediante las diferentes referencias y señales de control se pudo maniobrar el quadrotor para que este se traslade y vuele en diferentes sentidos y a grandes velocidades. Aunque el sistema resultó estable, fue necesario un control manual de las referencias de ángulo para mantener a la aeronave en posición, ya que perturbaciones pequeñas como brisas hacían que este se traslade rápidamente. El control de altura se realizó de manera manual durante todo el trabajo y se experimentaron algunas dificultades para mantenerlo a una altura constante a lo largo del tiempo. Como prueba final se realizaron pruebas a campo abierto, con viento, pudiendo volar al quadrotor a más de 40 metros del piso.

Ademas, se observó la robustez del sistema respecto a perturbaciones y también ante cambios en el quadrotor producidos por roturas y desgaste, lo que fue un resultado inesperado. Por ejemplo, durante las primeras pruebas se tomó el recaudo de balancear manualmente las hélices (agregando o quitando material de las mismas) para reducir las vibraciones, mientras que en las últimas, estas se encontraban completamente deformadas por los golpes y cubiertas de tierra, situación que no afectó a la estabilidad del sistema.

En resumen, se pudo demostrar que un problema aparentemente complejo, como es el control de una aeronave, puede ser resuelto mediante la descomposición de este en pequeños problemas más sencillos. Inclusive se demostró que es posible solucionar cada uno de estos pequeños problemas utilizando las técnicas de la teoría de control clásico. Por último, el diseño de computadora de vuelo propuesto probó ser efectivo como plataforma de control y permitió implementar la solución a estos problemas. Se comprobó también, que dada la gran capacidad de computo con la que fue diseñada, esta puede alojar perfectamente el control de actitud dejando suficiente capacidad de computo y almacenamiento para futuras aplicaciones.

7.2. Trabajo futuro

Del desarrollo matemático y los diferentes experimentos que se realizaron durante el trabajo se observaron diferentes tareas que resultan de gran interés para trabajos futuros. A continuación se detallan algunas de estas y se proponen diferentes caminos para su solución.

7.2.1. Navegación

Como se explicó a lo largo del trabajo, no se estudiaron en profundidad los problemas de navegación. Resulta, entonces, evidente que mucho trabajo a futuro se puede desarrollar en este área. Algunas ideas interesantes al respecto son:

- Desarrollo de un algoritmo de fusión de datos que incluya los valores obtenidos por el magnetómetro de la computadora de vuelo para así determinar con precisión el ángulo de yaw respecto del norte magnético.
- Implementación de un sistema de navegación que permita determinar la posición de la aeronave en el espacio utilizando propagación y datos provenientes de sensores externos como ser GPS o tecnologías similares para exteriores, o bien señales de radio para interiores.
- Diseñar un sistema que a partir del sensor barométrico, incluido en la computadora de vuelo, y mediante un sensor externo de distancia (sonar, laser, etc.) permita determinar la altura de vuelo del quadrotor.

7.2.2. Modelado

El modelado, como se mencionó en reiteradas oportunidades, juega un papel fundamental en el desarrollo del trabajo de control ya que permite determinar en una primera instancia la estrategia de control y luego da un punto de partida para el ajuste de los controladores. Respecto a esto surgen algunas áreas que no fueron exploradas en este trabajo y que resultan de gran interés:

- Encontrar un modelo para ángulos grandes o velocidades grandes que contemple los efectos aerodinámicos de la resistencia del aire y el “blade flapping”.
- Estudiar el “efecto suelo” para aplicaciones de despegue y aterrizaje autónomo.
- Diseñar un sistema de identificación paramétrica en vuelo del sistema.

7.2.3. Control

El área de control, si bien fue la más desarrollada a lo largo del trabajo, también presenta una gran variedad de posibilidades para trabajos futuros. Algunas de estas son:

- Diseñar un sistema de control de altitud basado en datos del barómetro integrado o de un sonar que permita mantener el vuelo a una altura constante.
- Implementar un sistema de control adaptativo, que en función de un modelo identificado en línea, permita adaptar las constantes de los controladores.
- Diseñar un sistema de control dependiendo del estado del quadrotor. Este permitiría cambiar la estrategia de control según el estado de vuelo o la misión. Por ejemplo para intercambiar entre despegue, aterrizaje y vuelo, o bien hovering y desplazamiento.
- Implementar algoritmos de control óptimo para control de actitud que permita minimizar la cantidad de energía necesaria para pasar de un estado al otro.
- Diseñar un sistema de control de velocidad de traslación.
- Ensayar algoritmos de control moderno, como LQR, DMC, LPV, etc.

7.2.4. Guiado

Se presentaron algunos trabajos relacionados con el guiado en la introducción y fue un tema que no se abordó en absoluto durante el desarrollo de este trabajo. Se listan a continuación algunas de las áreas de investigación que podrían desarrollarse sobre el tema:

- Diseñar un sistema de guiado que permita mantener estable la posición de la aeronave en términos de latitud y longitud.
- Diseñar un controlador que permita que el quadrotor recorra puntos en un mapa georeferenciado y/o realice vuelos en trayectorias precisas.

7.2.5. Computadora de vuelo

Si bien los resultados de la computadora de vuelo fueron muy buenos, se detectaron algunos aspectos que se pueden estudiar para una futura versión de esta. A continuación se presentan algunos de estos:

- Rediseño del circuito impreso utilizando técnicas avanzadas de PCB's multicapas a fin de reducir el tamaño y costo de cada unidad.
- Evaluar la integración de un “transceiver” de radio en la placa con el fin de de reducir el costo y/o mejorar el enlace de comunicaciones.
- Analizar la posibilidad de usar un procesador más potente, como un Cortex-M4, o un multicore M4-M0, por ejemplo, para lograr mejor desempeño, y estudiar también la conveniencia de utilizar un procesador de la gama del Cortex-R4 para morar la confiabilidad del sistema.
- Estudiar la viabilidad de utilizar un procesador de alta gama, como Cortex-A8 o A9, single core o multicore, para poder correr algoritmos de control, guiado y navegación más potentes, a la vez que aplicaciones de alto nivel más avanzadas.
- Incorporar un receptor de GPS.

7.2.6. Mejoras directas de este trabajo

A lo largo del trabajo se encontraron problemas específicos de la implementación elegida que en algunos casos se resolvieron de manera sencilla y sin ahondar en el tema, mientras que en otros se prefirió dejarlos para un trabajo futuro. A continuación se listan brevemente estos trabajos que surgen directamente como mejoras del presente sistema:

- Compensar la acción colectiva, Z_C , de manera automática a fin de evitar que el quadrotor pierda altura cuando su referencia de ángulo en pitch o roll es distinta que cero. Esto se podría resolver utilizando un control de altura en conjunto con un determinador de altitud o bien compensando la acción colectiva en función del ángulo de ataque mediante la aplicación de funciones trigonométricas para la descomposición de la fuerza total.
- Analizar los efectos de la utilización de una matriz de ganancias M_{mix} que permita saturación en los actuadores.
- Implementar una matriz de ganancias dinámica que permita variar las ganancias de cada control en función del objetivo deseado.
- Diseñar un sistema que permita determinar de manera automática el modo de vuelo adecuado según los comandos del piloto. Por ejemplo, que el modo acrobático se desactive cuando el piloto suelta los controles pasando a modo actitud y manteniendo a la aeronave estabilizada.

En resumen se observa que este trabajo puede servir de inspiración para futuras tesis de grado, maestría o doctorado o eventualmente trabajos finales de carreras de grado o de especialización. También se destaca que puede ser utilizado como material de lectura complementaria a los fines de motivar a estudiantes de grado que estén cursando las materías básicas del área de teoría de control.

Bibliografía

- [1] Aerospace controls laboratory - mit. "<http://acl.mit.edu/projects/>".
- [2] Grasp - general robotics, automation, sensing and perception laboratory, upenn university. "<https://www.grasp.upenn.edu/>".
- [3] Paul G Savage. What do accelerometers measure?
- [4] Parrot ardrone. <http://ardrone2.parrot.com/>.
- [5] Dji innovations phantom. <http://www.dji-innovations.com/product/phantom/>.
- [6] Arducopter project. <https://code.google.com/p/arducopter/>.
- [7] Multiwii project. <http://www.multiwii.com/>.
- [8] Openpilot project. <http://www.openpilot.org/>.
- [9] Starmac laboratory, berkeley university. "<http://hybrid.eecs.berkeley.edu/starmac/>".
- [10] J.G. Leishman. *Principles of Helicopter Aerodynamics*. Cambridge Aerospace Series. Cambridge University Press, 2006.
- [11] Bernard Michiniy Jonathan P. Howz Mark Cutler, N. Kemal Urey. Comparison of fixed and variable pitch actuators for agile quadrotors. http://acl.mit.edu/papers/GNC11_Cutler_uber.pdf.
- [12] V. Moreau N. Guenard, T. Hamel. Dynamic modeling and intuitive control strategy for an x4-yer. http://www2.irccyn.ec-nantes.fr/Robvolint/files/ICCA_05.pdf.
- [13] Aleksandr Kushleyev Bruce Kothmann Vijay Kumar Caitlin Powers, Daniel Mellinger. Inuence of aerodynamics and proximity effects in quadrotor flight. http://www.seas.upenn.edu/~dmel/ISER2012_powers.pdf.
- [14] Vijay Kumar Alex Kushleyev, Daniel Mellinger. Towards a swarm of agile micro quadrotors. <http://www.roboticsproceedings.org/rss08/p28.pdf>.
- [15] Robert Mahony Moses Bangura. Nonlinear dynamic modeling for high performance control of a quadrotor.

- [16] Derek Scott Miller. Open loop system identification of a micro quadrotor helicopter from closed loop data.
- [17] D. I. Bazov. *Helicopter Aerodynamics*. NASA, 1972.
- [18] K.J. Åström and T. Hägglund. *Pid controllers*. Setting the standard for automation. International Society for Measurement and Control, 1995.
- [19] Nicolas Minorsky. *Directional stability of automatically steered bodies*. J. Amer. Soc. Naval Eng. 34 (2): 280-309, 1922.
- [20] K.J. Åström and Bjorn Wittenmark. *Computer-Controlled Systems: Theory and Design (3rd Edition)* /. Prentice Hall, 1996.
- [21] Dr M.J. Willis. Proportional-integral-derivative control. <http://lorien.ncl.ac.uk/ming/pid/PID.pdf>.
- [22] Alan Kharsansky. Implementación y benchmarking de blas en microcontroladores 32 bits.
- [23] Lpc1769, nxp semiconductors. http://www.nxp.com/products/microcontrollers/cortex_m3/LPC1769FBD100.html#ordering.
- [24] Gerard Lachapelle Jared B. Bancroft. Data fusion algorithms for multiple inertial measurement units.
- [25] Robotics automation magazine, vol 19, no. 3. september 2012. http://www.nxp.com/products/microcontrollers/cortex_m3/LPC1769FBD100.html#ordering.
- [26] Hasan Ghazi Al-Khakani Samer Kombarji Zaid Al-Khatib, Jaime Yu. A wireless multivariable control scheme for a quadrotor hovering robotic platform using ieee® 802.15.4.
- [27] Bobby Woolf Gregor Hohpe. *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley Professional;, 2003.
- [28] R.P.G. Collinson. *Introduction to Avionics Systems*. SpringerLink : Bücher. Springer London, Limited, 2011.
- [29] Finn Haugen. Pid control of dynamic systems.
- [30] Patrick Bouffard. On-board model predictive control of a quadrotor helicopter: Design, implementation, and experiments.
- [31] David M. Bevly Warren S. Flenniken IV, John H. Wall. Characterization of various imu error sources and the effect on navigation performance.

APÉNDICE A

Diagramas esquemáticos de la Computadora de Vuelo

