



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCUELA TÉCNICA  
SUPERIOR INGENIEROS  
INDUSTRIALES VALENCIA

Curso Académico:



*Dedicado a mi abuela por su valentía.*



# Agradecimientos

Dar las gracias a mis padres por toda la confianza depositada en mi durante todos estos años. Ha sido un camino largo y lleno de dificultades que he ido salvando gracias a vuestro apoyo. También agradecer a mi hermana por ser el referente cuando el camino se complica y a Inma por tener la paciencia necesaria todos estos meses.

*Gracias abuelo por enseñarnos a esforzarnos cada uno de nuestros días hasta el final.*

Como no, agradecer a mi tutor Javier que me ha introducido más profundamente en el mundo del control y me ha guiado. Agradecer también a Sergio y Jesús sus consejos y ayudas prestadas durante la elaboración de este trabajo.



# Resumen

Dirigibles están siendo objeto de estudio y desarrollo por su potencial para desarrollar misiones aéreas con bajo coste. Este proyecto aborda el modelado matemático de un dirigible de pequeña escala haciendo uso de las ecuaciones de Newton. Para su desarrollo se ha considerado el conjunto de fuerzas y momentos formado por : las fuerzas aerodinámicas, fuerza de la gravedad, fuerza de flotabilidad, fuerza de propulsión y los momentos generados por estos. Del modelado no lineal se obtiene mediante transformadas de Laplace el modelo lineal MIMO sobre el cual se ha diseñado un sistema de control. Empleando un PI para el control de la velocidad en  $x$ , un control en cascada constituido por una ganancia y un PI para el control de la velocidad de ascenso y un PID para el control de la guiñada o Yaw. Se ha implementado un sistema de navegación simple que le permite navegar de un punto otro. Todo el funcionamiento se ha verificado mediante software y se ha empleado el entorno de simulación de Flightgear.

**Palabras Clave:** Dirigible, Autónomo, UAV, Control, modelado.



# Índice general

Resumen	I
Índice general	VII
1 Objetivos y alcance	1
2 Estado del Arte	3
2.1 Antecedentes históricos . . . . .	3
2.2 El dirigible y el control . . . . .	4
3 Cálculo del modelo	9
3.1 Introducción . . . . .	9
3.2 Ejes y sistema de referencia . . . . .	10
3.3 Definición de fuerzas . . . . .	12
3.4 Definición de los momentos . . . . .	15
4 Implementación del modelo	19
5 Validación del modelo	25
5.1 Implementación de un nuevo modelo no lineal. . . . .	25
5.2 Simulación y comparación . . . . .	25
6 Linearización del modelo	35
7 Implementación del sistema de control	43
7.1 Diseño . . . . .	43
7.2 Validación. . . . .	46

8 Navegación	55
9 Entorno de Simulación 3D	63
10 Conclusiones y trabajos futuros	65
A Parámetros del modelo	67
B Código Matlab: Linearización y FDT	69
C Ajuste de los controladores	77
Bibliografía	79

## Capítulo 1

# Objetivos y alcance

El trabajo realizado tiene como objetivo la implementación de un sistema de control capaz de permitir volar de forma autónoma un dirigible de pequeñas dimensiones. Permitiendo así su uso en nuevos ambientes y aplicaciones tales como la vigilancia, la obtención de datos meteorológicos o la grabación de eventos en interiores.

El proyecto está disgregado en objetivos menores que en su conjunto satisfacen el vuelo autónomo del dirigible. Entre ellos se encuentra la creación del modelo matemático del dirigible que englobe los principales comportamientos dinámicos y a su vez sea fácil de implementar. El desarrollo del control de los movimientos de la aeronave, la navegación y la integración en un entorno de simulación virtual.



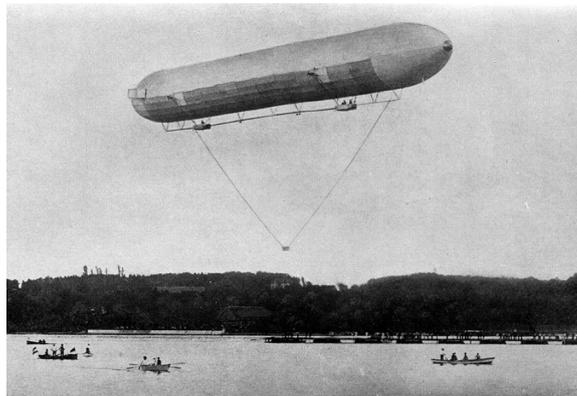
## Capítulo 2

# Estado del Arte

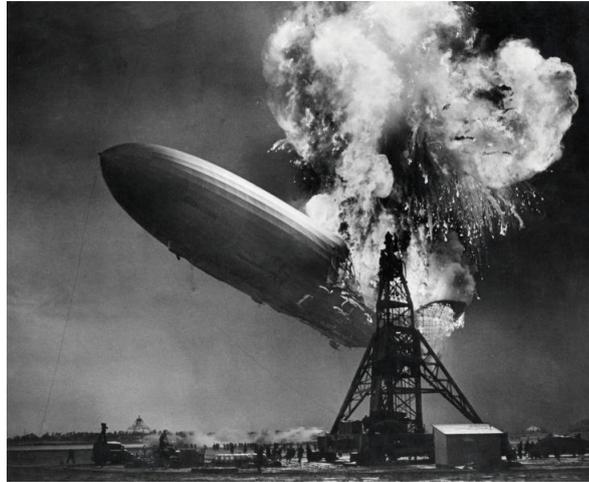
### 2.1 Antecedentes históricos

El nacimiento de los dirigibles se remonta a mediados del siglo XIX donde a los ya utilizados globos aerostáticos se les incorporó un medio de propulsión que les permitía ser controlados y navegar por el cielo. Por primera vez Henri Giffard, en 1852, incorporó una máquina de vapor en un dirigible propulsado y fue capaz de recorrer 27 Km.

Su desarrollo se vio impulsado pocos años antes de la Primera Guerra Mundial donde aparecieron los primeros dirigibles capaces de transportar un gran número de pasajeros. La edad de oro llegaría con con la fabricación del dirigible LZ1, el primer Zeppelin.



**Figura 2.1:** LZ1 , primer dirigible de Luftschiff Zeppelin



**Figura 2.2:** Accidente del Hindenburg

Durante la Gran Guerra su fueron utilizados para bombardear y como observatorios de vigilancia. Tras la guerra se empezaron a utilizar como transporte de pasajeros pero tras la tragedia del Hindenburg, en Nueva Jersey el 6 de mayo del 1937, su popularidad se precipitó coincidiendo con un auge de los aviones. Actualmente su uso se limita, por lo general, a la monitorización del medio ambiente, vigilancia y uso publicitario en eventos.

## 2.2 El dirigible y el control

Este apartado pretende ilustrar al lector de otros proyectos de mayor escala donde también se desarrolla el modelado y control de estas aeronaves.

### 2.2.1 *YEZ-2A* [5]

En este proyecto se llevaron a cabo pruebas en un túnel de viento que encaminaron al desarrollo de un modelo dinámico de alta fidelidad. El dirigible empleado fue el YEZ-2A ( $129.5\text{ m}$  y  $107.42\text{ m}^3$ ) construido por Airship Industries. El modelo que se desarrolló fue un modelo de 6 grados de libertad no lineal escrito en el lenguaje de simulación ACSL (Advanced Continuous Simulation Language) y que empleaba los datos obtenidos del túnel de viento almacenados en una base de datos. El modelo permitió un estudio con mayor profundidad de los modos de estabilidad y estableció un precedente para aquellas aeronaves con el ratio longitud/diámetro similar ( $r=4.2$ ) que pueden hacer uso de las pruebas de viento realizadas.



**Figura 2.3:** Góndola del YEZ-2A

### **2.2.2 Project AURORA [4][2]**

El proyecto AURORA, desarrollado por Information Technology Institute of Campinas (Brazil), se trata de un proyecto multifase que pretende desarrollar un dirigible no tripulado capaz de realizar vuelos autónomos sobre localizaciones definidas por el usuario para inspecciones visuales y sistemas de monitorización medio ambiental.

La aeronave empleada es el AS-800 cuyo modelo dinámico está basado en el que se desarrolló para el YEZ-2A por Gomes y Ramos [5] ya que presenta un ratio longitud/diámetro similar. El modelo toma ciertas consideraciones como por ejemplo: considera masas y inercias virtuales, un sistema de ejes ligado al cuerpo y se asume el dirigible como un sólido rígido. Un resultado importante que se extrae al linealizar el modelo es la separación de los movimientos vertical y longitudinal.



**Figura 2.4:** Dirigible del proyecto Aurora: AS-800

El diseño inicial del controlador consistía en un controlador de cuatro niveles. El primer nivel engloba los actuadores que controlan la aeronave. En el segundo nivel se sitúan los controladores de los actuadores que actúan en función de las órdenes del tercer nivel. En este nivel se decide qué valores y comandos pasar a los actuadores en función de las órdenes del cuarto nivel, que establece la trayectoria a seguir.

En [8] se implementa el controlador lateral que consiste de dos controladores. Un controlador proporcional integral (PI) para el control de la trayectoria, el cual toma el error entre la trayectoria actual y la deseada y obtiene el ángulo de orientación. Dicho ángulo es utilizado por el segundo controlador, un proporcional derivativo (PD), para encauzar la aeronave en la trayectoria deseada. Las ganancias de se obtuvieron por prueba y error o por optimización H-Infinito.

Alexandra Bento en su Phd. Thesis [7] propone una solución no lineal robusta para el control que permita tener una única ley de control para cualquier escenario de vuelo.

### ***2.2.3 Proyecto de dirigible autónomo por LAAS-CNRS***

Laboratoire d'analyse et d'architecture des systèmes (LAAS) centra su investigación en el desarrollo de una aeronave autónoma que sea capaz de obtener información actualizada del terreno.

La aeronave empleada en esta proyecto tiene las siguientes características:

**Tabla 2.1:** Especificaciones Karma

<b>Modelo:</b>	AS-500
<b>Nombre:</b>	Karma
<b>Longitud:</b>	8 m
<b>Diámetro máximo:</b>	1.9 m
<b>Volumen:</b>	15 $m^3$
<b>Ratio L/D:</b>	4.25
<b>Carga máxima;</b>	3.5 kg

El modelo se deduce mediante las leyes de Newton tomando como referencia la nariz de la aeronave por ser independiente a la variación de parámetros de la aeronave y los coeficientes aerodinámicos son obtenidos en [1].

En [6] se detalla la estrategia de control empleada. En ella se describen tres fases de vuelo: despegue, navegación longitudinal y lateral y aterrizaje. Se plantea un modelo de control con los movimientos laterales y longitudinales desasociados. Reduciendo, por tanto, la complejidad del control.

En [6] también se hace hincapié en el análisis del terreno mediante técnicas SLAM y su incorporación al control que permita no solo realizar mapas de alta fidelidad sino también determinar con mayor precisión la posición del dirigible.

#### **2.2.4 Proyecto Titan Aerobot de NASA/JPL**

La NASA junto con JPL (Jet Propulsion Laboratory) han estado investigando la posibilidad de desplegar dirigibles (aerobot) para la exploración de planetas o lunas con atmósfera gaseosa tal que permita cubrir mayor superficie de terreno explorado. En concreto en [3] para la luna de Saturno , Titan.

Por tanto en [3] se exponen los requisitos que debe de poseer la aeronave. Al estar desplegada en un ambiente desconocido, el dirigible, debe de ser capaz de asegurar su propia integridad, poseer un control preciso y robusto que permita un vuelo autónomo. Incluyendo el despliegue, despegue, toma de muestras, largas travesías y cernerse en una posición. Además de autodetección de peligros, seguimiento y planificación de misiones.

La aeronave empleada se trata del dirigible AS-800B cuyas especificaciones son:

**Tabla 2.2:** Especificaciones AS-800B

<b>Modelo:</b>	AS-800B
<b>Longitud:</b>	11 m
<b>Diámetro máximo:</b>	2.5 m
<b>Volumen:</b>	34 m <sup>3</sup>
<b>Ratio L/D:</b>	4.4
<b>Carga máxima;</b>	12 kg
<b>Velocidad máxima:</b>	13 m/s

El modelo aerodinámico de aerobot se realizó mediante en el modelado de sistemas, incluyendo datos reales de sensores y actuadores. Además se identificaron los parámetros aerodinámicos para posteriormente validar el modelo y el control mediante la simulación y la utilización de un campo de pruebas. El control autónomo del vuelo está dividido en distintas capas de supervisión y consta de tres modos de funcionamiento: 'cruise', 'hover' y 'loiter'. Cada uno de estos modos emplea los controladores de ascenso, descenso, giro y altitud. Y estos a su vez dirigen los controladores de propulsión y actitud.

En [9] se detallan las técnicas empleadas para la generación de trayectorias de vuelo. El proceso de planificación de estas trayectorias tiene en cuenta las características concretas de Aerobot así como radio mínimo de giro que este presenta. De modo que las trayectorias son calculadas para optimizar el

camino más corto dentro de las limitaciones de maniobra de la aeronave. Además se añade la capacidad de planificar las trayectorias mediante el uso de cámaras.

## Capítulo 3

# Cálculo del modelo

### 3.1 Introducción

El presente capítulo tiene como objetivo realizar un modelo matemático lo suficientemente preciso y que a su vez se adapte a las necesidades de uso del dirigible. Para ello deberá contemplar los siguientes puntos:

- Incluir las principales fuerzas que afectan al dirigible.
- Tener una fácil interpretación física.
- Estar basado en el centro de gravedad.

Asumiendo además las siguientes simplificaciones:

- La masa del dirigible permanece constante durante el vuelo.
- El globo del dirigible se considera un sólido rígido.
- El dirigible es simétrico respecto del plano  $oxz$
- El dirigible tiene el centro de gravedad y volumen en el plano  $oxz$
- El dirigible opera bajo velocidades de viento nulas.

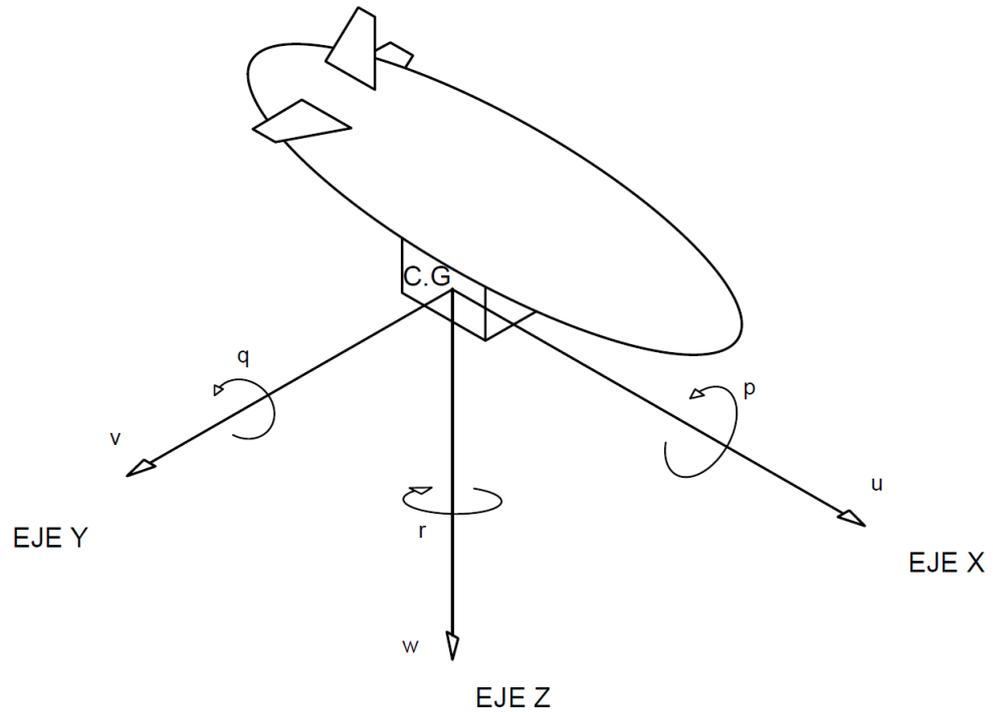


Figura 3.1: Sistema de referencia ligado al cuerpo

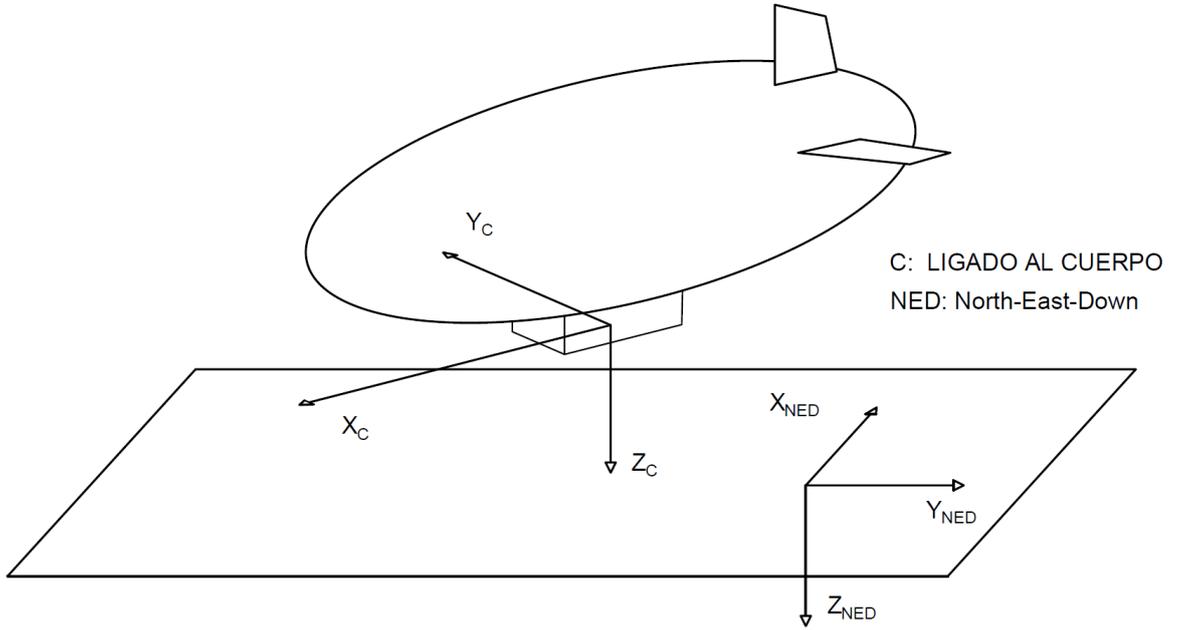
## 3.2 Ejes y sistema de referencia

### 3.2.1 Sistema de referencia ligado al cuerpo

El sistema de referencia ligado al cuerpo queda definido en el centro de gravedad (C.G) del dirigible con el EJE-X orientado longitudinalmente de modo que señala donde mira el dirigible. Siguiendo la regla de la mano derecha el EJE-Y esta orientado transversalmente y el EJE-Z positivo hacia abajo. La Figura 3.1 muestra esta configuración. Por tanto quedan también definidas las velocidades lineales y angulares siendo  $[u, v, w]$  y  $[p, q, r]$  respectivamente.

### 3.2.2 Sistema de referencia North-East-Down (NED)

Con el fin de poder determinar la posición y la orientación de la aeronave es necesario un sistema de referencia inercial. Y para ello se supone que la tierra es plana y no tiene rotación. De modo que el sistema NED cumple este propósito siendo en este sistema de referencia el EJE-X positivo hacia el norte, el EJE-Y positivo hacia el este y el EJE-Z positivo hacia el centro de la tierra y perpendicular al plano formado por los ejes X e Y. La figura 3.2 muestra en una misma ilustración tanto el sistema de referencia NED como el ligado al cuerpo.



**Figura 3.2:** Comparación sistema de referencia ligado al cuerpo y NED

### 3.2.3 Rotación de los sistema de referencia

Dependiendo de la naturaleza de las fuerzas, algunas de ellas deben ser transformadas para ser expresadas en un mismo sistema de referencia. En el caso particular que se expone, la unidad de medición inercial va montada en la góndola del dirigible y por tanto es de interés rotar todas las fuerzas del sistema de referencia NED (sistema inercial) al sistema de referencia ligado al cuerpo. Para ello se emplea una matriz de transformación conocida como matriz de cosenos directores,  $T_{MCD}$ . Esta matriz queda definida como:

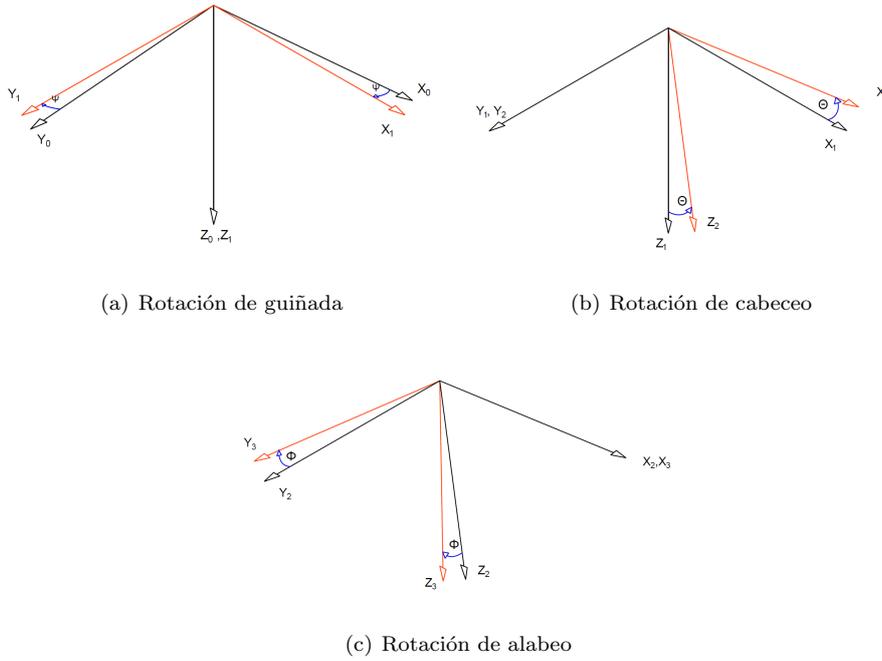
$$T_{MCD} = T_{\phi} T_{\theta} T_{\psi} \quad (3.1)$$

donde  $T_{\phi}$ ,  $T_{\theta}$  y  $T_{\psi}$  son las rotaciones respecto de los ejes  $x$ ,  $y$  y  $z$  respectivamente. En la Figura 3.3 se observa las rotaciones que experimenta un sistema de referencia al aplicar la transformación de la Eq. 3.1. Primero una rotación de guiñada (Eq. 3.2), seguida de una rotación de cabeceo (Eq. 3.3) y finaliza con una rotación de alabeo (Eq. 3.4).

$$\begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \quad (3.2)$$

$$\begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & -\sin \theta \\ 0 & 1 & 0 \\ \sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \\ z_1 \end{bmatrix} \quad (3.3)$$

$$\begin{bmatrix} x_3 \\ y_3 \\ z_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & \sin \phi \\ 0 & -\sin \phi & \cos \phi \end{bmatrix} \begin{bmatrix} x_2 \\ y_2 \\ z_2 \end{bmatrix} \quad (3.4)$$



**Figura 3.3:** Rotaciones de Euler

### 3.3 Definición de fuerzas

Mediante la segunda ley de Newton es posible obtener un modelo matemático sencillo y que además facilite la interpretación de los resultados obtenidos a partir de él. De modo que se puede escribir:

$$\begin{cases} m\dot{V} = F'_g + F'_f + F_{aero} + F_{prop} \\ J\dot{W} = M_f + M_{aero} + M_{prop} + M_{viscosidad} \end{cases} \quad (3.5)$$

siendo  $m$  la masa del dirigible,  $\dot{V}$  el vector aceleraciones lineales,  $F_g$  vector de la fuerza de la gravedad,  $F_f$  el vector de la fuerza de flotabilidad,  $F_{aero}$  el vector de las fuerzas aerodinámicas y  $F_{prop}$  el vector de la fuerza de propulsión.  $\dot{W}$  es el vector de aceleraciones angulares y  $J$  es la matriz de inercia definida como:

$$J = \begin{vmatrix} \frac{2}{3}mr_b^2 & 0 & 0 \\ 0 & \frac{1}{3}m(r_a^2 + r_b^2) & 0 \\ 0 & 0 & \frac{1}{3}m(r_a^2 + r_b^2) \end{vmatrix}$$

siendo  $r_a$  y  $r_b$  los semiejes de un elipsoide similar al globo del dirigible.  $M_f$  y  $M_{prop}$  son el momento causado por la flotabilidad aplicada en el centro de volumen (CV) y el momento debido a las fuerzas de propulsión. Y  $M_{aero}$  y  $M_{viscosidad}$  son los momentos generados por las fuerzas aerodinámicas y el rozamiento viscoso generado al rotar la aeronave.

### 3.3.1 Fuerza de gravedad y flotabilidad

La fuerza gravedad actúa sobre el dirigible haciéndolo descender de forma continuada y su punto de aplicación es el centro de gravedad. Esta fuerza se define como:

$$F_g = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (3.6)$$

siendo  $g$  la aceleración de la gravedad. Por el Principio de Arquímedes se calcula la fuerza de flotabilidad que contrarresta el efecto que ejerce la gravedad. Esta fuerza aparece al rellenar el globo del dirigible con un gas de menor gravedad que el aire que lo envuelve. De modo que se puede escribir:

$$F_f = \begin{bmatrix} 0 \\ 0 \\ -(\rho_a - \rho_{he})V_{globo}g \end{bmatrix} \quad (3.7)$$

donde  $\rho_a$  es la densidad del aire que envuelve al dirigible,  $\rho_{he}$  la densidad del helio,  $V_{globo}$  el volumen relleno con helio dentro del globo y  $g$  la aceleración de la gravedad. Esta fuerza se aplica en el centro de volumen (CV).

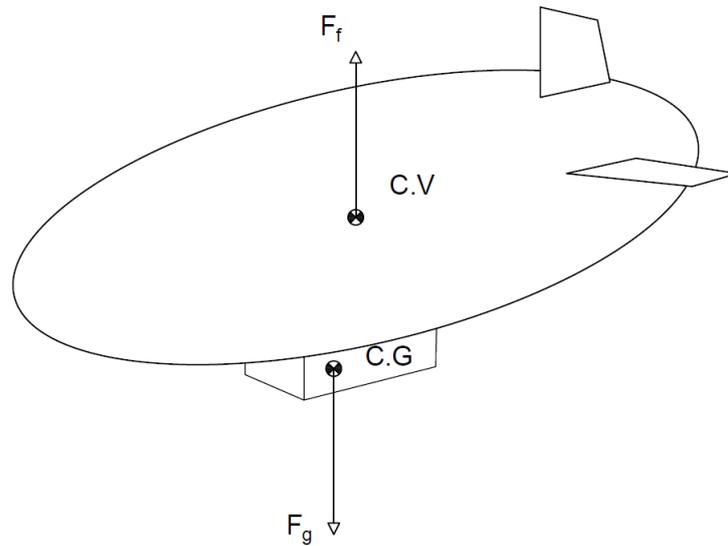
Cabe destacar que la fuerza de flotabilidad sufre variaciones debido a cambios en la densidad del aire y del helio. Viéndose esta reducida por el aumento de la presión atmosférica así como por un aumento de la humedad relativa. No obstante es mayor si la densidad del helio disminuye por efectos como el sobrecalentamiento.

### 3.3.2 Fuerzas aerodinámicas

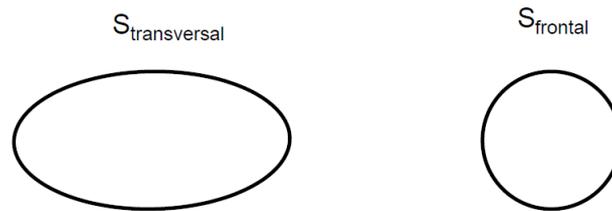
Con el fin de facilitar el entendimiento y el modelo matemático las fuerzas aerodinámicas quedan definidas como:

$$F_{aero} = \begin{bmatrix} -\frac{1}{2}\rho_a V_x^2 C_{dx} S_{frontal} \\ -\frac{1}{2}\rho_a V_y^2 C_{dy} S_{transversal} \\ -\frac{1}{2}\rho_a V_x^2 C_l S_{frontal} - \frac{1}{2}\rho_a V_z^2 C_{dz} S_{transversal} \end{bmatrix} \quad (3.8)$$

donde  $V_x$ ,  $V_y$ ,  $V_z$  son las velocidades relativas en los ejes  $x$ ,  $y$  y  $z$ .  $C_{dx}$ ,  $C_{dy}$ ,  $C_{dz}$  son los coeficientes aerodinámicos y  $\rho_a$  la densidad del aire.



**Figura 3.4:** Representación de la fuerza de la gravedad y la flotabilidad



**Figura 3.5:** Detalle de las superficies del dirigible

Se ha supuesto que en el entorno de actuación del dirigible la velocidad del viento es despreciable y por tanto las velocidades relativas corresponden a las velocidades de la aeronave aplicadas en el centro de volumen (CV).

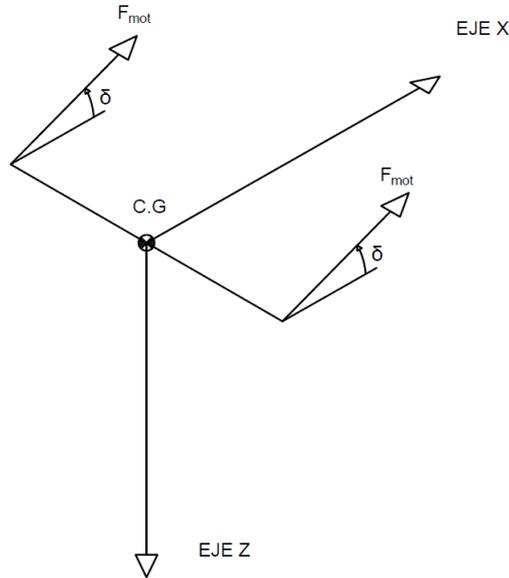
$$\vec{V}_{relativas} = \vec{V}_{viento} + \vec{V}_{dirigible}$$

### 3.3.3 Fuerza de propulsión

Para el cálculo de la fuerza de propulsión se ha tomado las siguientes premisas: los motores que propulsan la aeronave son iguales, están equidistes respecto al plano  $oxz$  y están a la misma altura que el centro de gravedad. La Fig. 3.6 muestra esta simplificación.

Por lo tanto podemos definir la fuerza de propulsión aplicada en el centro de gravedad como:

$$F_{prop} = \begin{bmatrix} 2F_{motor} \cos(\delta) \\ 0 \\ -2F_{motor} \sin(\delta) \end{bmatrix} \quad (3.9)$$



**Figura 3.6:** Detalle de la fuerza de propulsión

Para poder orientar la aeronave se hace uso de un motor situado en la parte trasera que permite el control de la guiñada del dirigible. Esta fuerza esta definida como:

$$F_{lat} = \begin{bmatrix} 0 \\ F_{mot_{lateral}} \\ 0 \end{bmatrix} \quad (3.10)$$

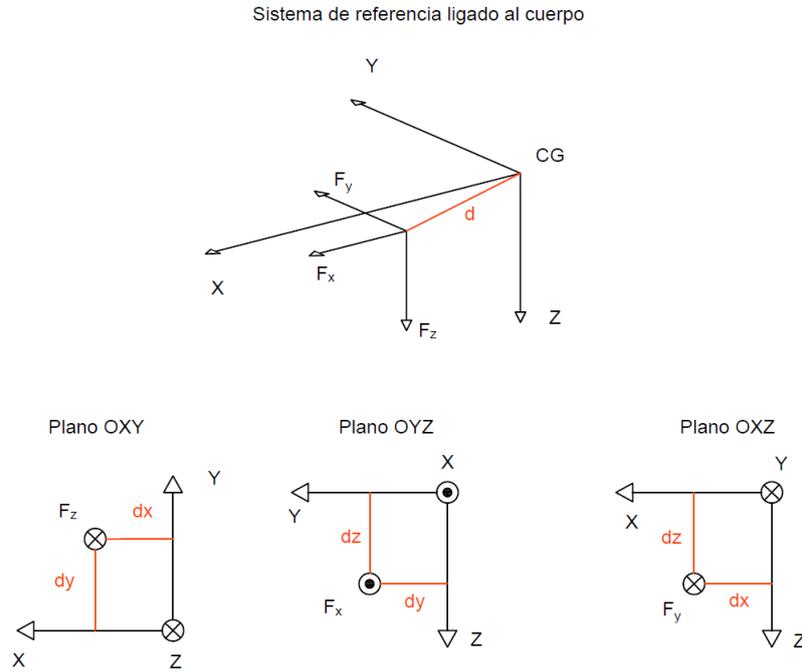
el punto de aplicación es la cola del dirigible.

### 3.4 Definición de los momentos

Todas aquellas fuerzas que no actúan en el centro de gravedad generan un momento. Con el objetivo de mejorar el entendimiento del lector de estos momentos se expone a continuación un caso genérico. La Fig. 3.7 ilustra como se han descompuesto las fuerzas utilizadas en el desarrollo genérico de los momentos.

El momento generado por un conjunto de fuerzas a una distancia  $d$  del centro de gravedad del dirigible viene determinado por:

$$M = \begin{bmatrix} F_z dy - F_y dz \\ -F_z dx + F_x dz \\ -F_x dy + F_y dx \end{bmatrix} \quad (3.11)$$



**Figura 3.7:** Detalle de la descomposición de fuerzas

Por tanto con esto es posible definir el momento generado por las fuerzas aerodinámicas, el momento generado por la flotabilidad y el momento generado por la fuerza de propulsión lateral.

La fuerza de flotabilidad que se ha definido en 3.3.1 está expresada en el sistema de referencia NED, por tanto para el cálculo es necesario premultiplicarla por la matriz de rotación definida en el punto 3.2.3.

$$F'_f = T_{MCD}F_f$$

Así el momento debido a la flotabilidad queda:

$$M_f = \begin{bmatrix} F'_{fz}y_v - F'_{fy}z_v \\ -F'_{fz}x_v + F'_{fx}z_v \\ -F'_{fx}y_v + F'_{fy}x_v \end{bmatrix} \quad (3.12)$$

donde las coordenadas del centro de volumen son  $(x_v, y_v, z_v)$ .

A diferencia de la fuerza de flotación, las fuerzas aerodinámicas vienen ya expresadas en el sistema de referencia ligado al cuerpo ya que se asume que la velocidad relativa del dirigible es la misma velocidad a la que se desplaza la aeronave. Por tanto el momento generado por las fuerzas aerodinámicas se puede escribir como:

$$M_{aero} = \begin{bmatrix} F_{aero_z} y_v - F_{aero_y} z_v \\ -F_{aero_z} x_v + F_{aero_x} z_v \\ -F_{aero_x} y_v + F_{aero_y} x_v \end{bmatrix} \quad (3.13)$$

En cuanto al momento generado por la fuerza de propulsión se define como:

$$M_{prop} = \begin{bmatrix} -F_{mot_{lateral}} r_x \\ 0 \\ F_{mot_{lateral}} r_z \end{bmatrix} \quad (3.14)$$

donde  $r_x, r_z$  son la distancia desde el punto de aplicación de la fuerza al eje  $x$  y  $z$  respectivamente. Oponiéndose al momento generado por la fuerza de propulsión se encuentra un momento generado por las fuerzas viscosas del aire el cual puede expresarse como:

$$M_{viscosidad} = \begin{bmatrix} -K_{viscosidad} \cdot p \\ -K_{viscosidad} \cdot q \\ -K_{viscosidad} \cdot r \end{bmatrix} \quad (3.15)$$

donde  $p, q$  y  $r$  son las velocidades angulares en los ejes  $x, y$  y  $z$  y  $K_{viscosidad}$  un coeficiente empírico .

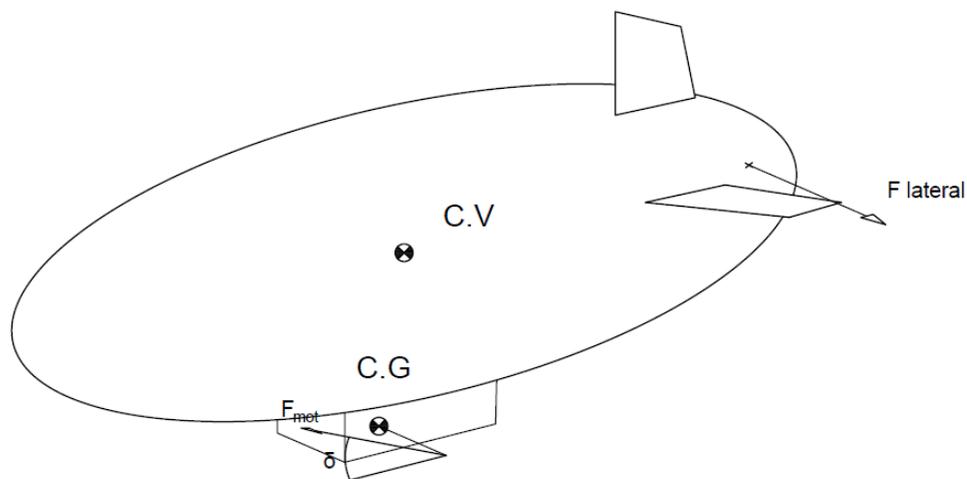


## Capítulo 4

# Implementación del modelo

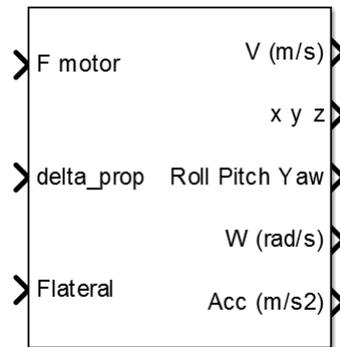
Para llevar a cabo la implementación del modelo descrito en el Capítulo 3 se emplea el software Matlab y el paquete Simulink. Simulink es un entorno de diagramas de bloque para la simulación multidominio. El modelo no lineal implementado tiene tres variables de control que permiten variar las variables de estado deseadas. Estas tres entradas de control son (Figura 4.1):

- $F_{\text{motor}}$ , que representa la fuerza realizada por cada motor de forma individual.
- Delta prop,  $\delta$ , indica la inclinación del vector de propulsión.
- Flateral, fuerza aplicada por el motor de cola del dirigible



**Figura 4.1:** Representación de las variables de control

Como variables de estado tenemos las velocidades lineales,  $V(m/s)$ , así como las velocidades angulares,  $W(rad/s)$ , de las cuales se deriva el desplazamiento en los tres ejes y los ángulos de euler. La Figura 4.2 muestra en simbología de Simulink para el modelo no lineal.



**Figura 4.2:** Representación en bloque del modelo no lineal del dirigible

La Figura 4.3 muestra el interior del bloque de la Figura 4.2. En el se relacionan las distintas fuerzas y momentos. Se puede diferenciar dos partes, por un lado la definición de las fuerzas y momentos, en azul, y por otro lado la resolución de las ecuaciones de Newton y la obtención de las variables de estado, en naranja.

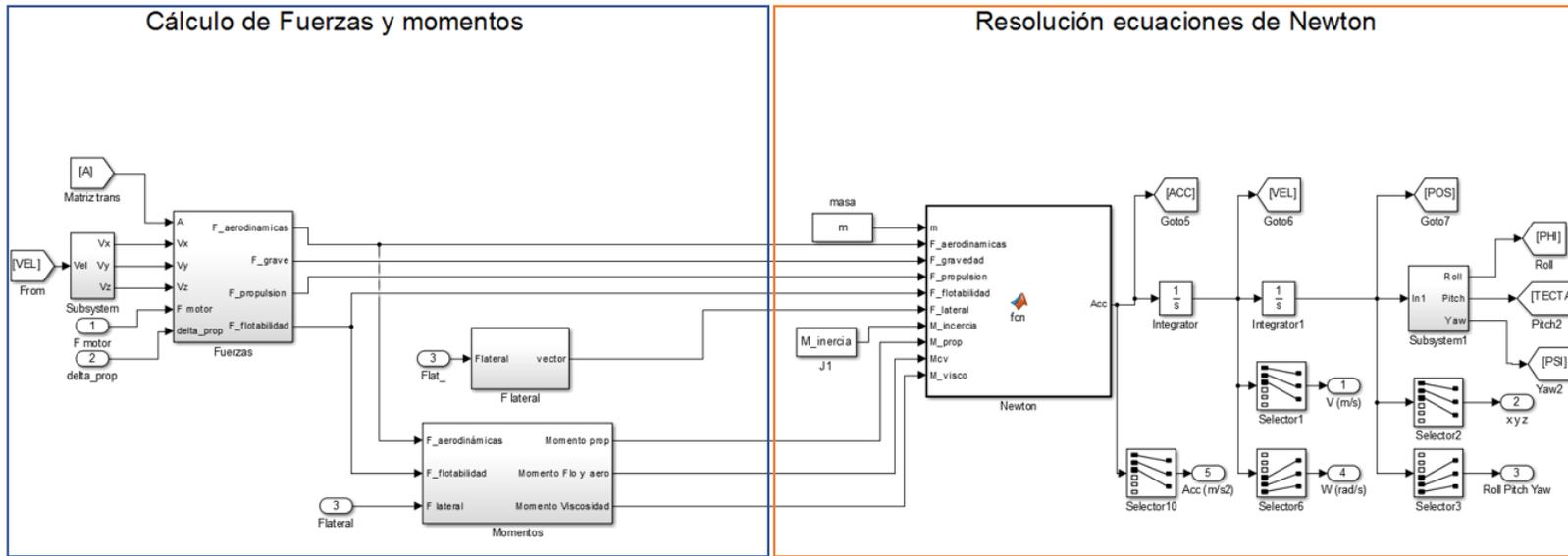


Figura 4.3: Implementación esquemática del modelo

En el recuadro del fuerzas y momentos conviven tres bloques, los dos superiores hacen referencia a las fuerzas y el bloque inferior al cálculo de los momentos. Dentro del bloque de 'fuerzas', situado en la parte superior izquierda de la Figura 4.3, se definen las fuerzas tal y como se han descrito en el Capítulo 3 (Figura 4.4). Cada uno de estos bloques tiene unas entradas determinadas que son los parámetros utilizados para los cálculos.

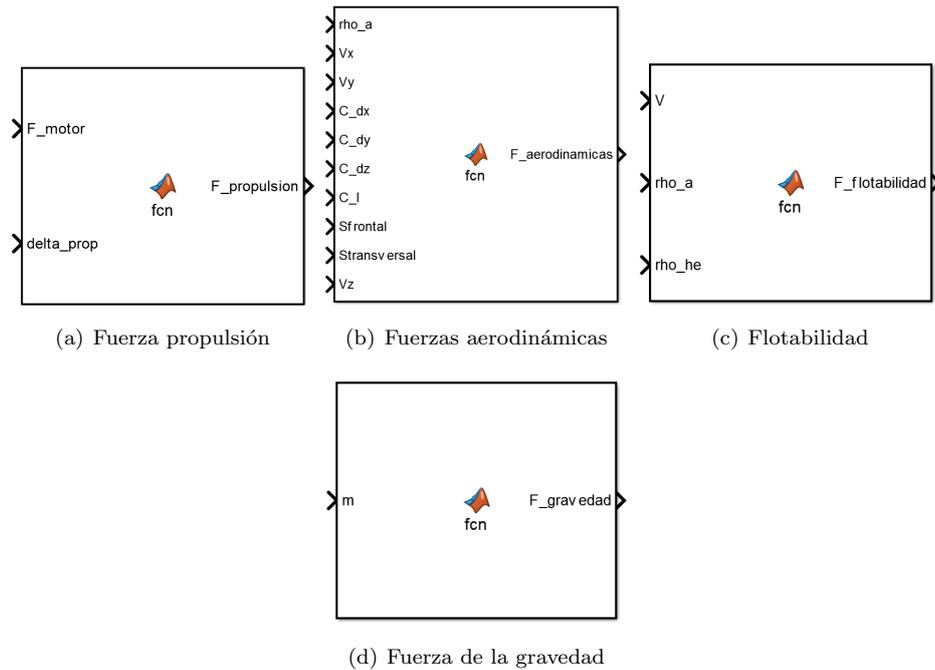


Figura 4.4: Interior bloque de fuerzas

Figura 4.4a, contiene implementada la fuerza de propulsión y tiene el siguiente código:

```
1 function F_propulsion = fcn(F_motor, delta_prop)
2 F_propulsion = [2*F_motor*cos(delta_prop) 0 -2*F_motor*sin(delta_prop)
3               ]';
```

Para las fuerzas aerodinámicas (Figura 4.4 b):

```
1 function F_aerodinamicas = fcn(rho_a, Vx, Vy, C_dx, C_dy, C_dz, C_l,
2                               Sfrontal, Stransversal, Vz)
3
4
5 F_aerodinamicas = [-1/2*rho_a*Vx^(2)*C_dx*Sfrontal ;
6                   -1/2*rho_a*Vy^(2)*C_dy*Stransversal*sign(Vy);
7                   -1/2*rho_a*Vx^(2)*C_l*Sfrontal -1/2*rho_a*Vz^(2)*
8                   C_dz*Stransversal*sign(Vz) ];
```

Del mismo modo para la flotabilidad (Figura 4.4 c):

```

1 function F_flotabilidad = fcn(V, rho_a, rho_he)
2 g=9.81;
3 F_flotabilidad =[0 0 -(rho_a-rho_he)*V*g]';

```

Y finalmente para la gravedad Figura 4.4 d:

```

1 function F_gravedad = fcn(m)
2 g=9.81;
3 F_gravedad =[0 0 m*g]';

```

Aunque se ha hecho hincapié en el capítulo anterior, se debe recordar que cada una de las fuerzas viene representada en sus respectivos ejes de coordenadas y que deben ser transformadas al sistema de referencia ligado al cuerpo. Por ello y teniendo en cuenta que la velocidad del viento es nula, es necesario premultiplicar la fuerza de la gravedad (Figura 4.4 d) y la flotabilidad (Figura 4.4 c) por la matriz de transformación.

Del mismo modo ocurre con el bloque de momentos, situado en la parte inferior izquierda de la Figura 4.3. En su interior se encuentran los bloques que definen los momentos (Figura 4.5).

El bloque de la Figura 4.5 a incluye el siguiente código:

```

1 function Momento_prop = fcn(F_lateral, rz, rx)
2
3 Momento_prop =[-F_lateral*rx 0 F_lateral*rz]';

```

por otro lado el momento viscoso queda definido como:

```

1 function M_visco = fcn(K_visc, pqr)
2
3 M1=-K_visc*pqr(1);
4 M2=-K_visc*pqr(2);
5 M3=-K_visc*pqr(3);
6
7 M_visco =[M1 M2 M3]';

```

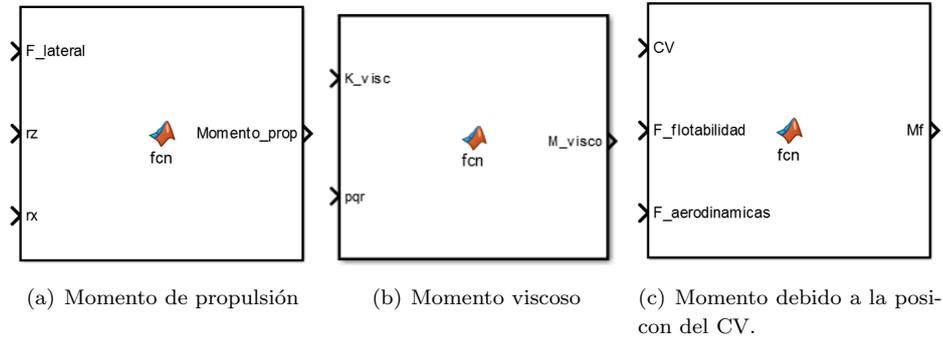
y finalmente el momento generado por la posición del centro de volumen se ha implementado del siguiente modo:

```

1 function Mf = fcn(CV, F_flotabilidad, F_aerodinamicas)
2
3 M1=F_flotabilidad(3)*CV(2)-F_flotabilidad(2)*CV(3) +F_aerodinamicas
    (3)*CV(2)-F_aerodinamicas(2)*CV(3);
4 M2=-F_flotabilidad(3)*CV(1)+F_flotabilidad(1)*CV(3)- F_aerodinamicas
    (3)*CV(1)+F_aerodinamicas(1)*CV(3);
5 M3=-F_flotabilidad(1)*CV(2)+F_flotabilidad(2)*CV(1)- F_aerodinamicas
    (1)*CV(2)+F_aerodinamicas(2)*CV(1);
6
7 Mf =[M1 M2 M3]';

```

La tabla aquí mostrada presenta los valores de entrada de los bloques de las figuras 4.4 y 4.5.



**Figura 4.5:** Interior bloque de momentos

**Tabla 4.1:** Constantes y parámetros de la aeronave

Masa:	0.4508 Kg	$C_{dx}$ :	0.041
Volumen:	$0.4362 m^3$	$C_{dy}$ :	0.041
$\rho_{he}$ :	$0.1664 kg/m^3$	$C_{dz}$ :	0.041
$K_{visc}$ :	0.1	$C_l$ :	0
Ratio L/D:	2.42	L:	1.7 m
$r_x$ :	0 m	$r_z$ :	0.7 m

Donde  $C_{dx}$ ,  $C_{dy}$ ,  $C_{dz}$  son los coeficientes de resistencia del aire considerando los valores obtenidos en [11]. Se ha considerado además que la sustentación aportada por la geometría  $C_l$  con el avance de la aeronave es nula.  $r_z$  hace referencia al brazo del motor lateral hasta el eje z y del mismo modo  $r_x$  es la longitud del brazo del motor lateral hasta el eje x.

En el recuadro naranja se ejecuta la resolución de las ecuaciones de Newton obteniendo como resultado un vector columna con las aceleraciones lineales y angulares. Por tanto con el uso de integradores es posible obtener las respectivas velocidades y posiciones. En el interior de la Figura 4.6 se encuentra:

$$\left\{ \begin{array}{l} \dot{V}_{lineal} = \frac{1}{m}(F_{aero} + F_g + F_{prop} + F_{flot} + F_{lateral}) \\ \dot{\omega}_{angular} = M_{inercia}^{-1}(M_{prop} + M_{cv} + M_{viscoso}) \\ Acc = \begin{bmatrix} \dot{V}_{lineal} \\ \dot{\omega}_{angular} \end{bmatrix} \end{array} \right.$$

**Figura 4.6:** Bloque para el cálculo de las ecuaciones de Newton.

## Capítulo 5

# Validación del modelo

En este capítulo se pretende ilustrar al lector cómo se ha llevado a cabo la validación del modelo implementado en el Capítulo 4.

### 5.1 Implementación de un nuevo modelo no lineal

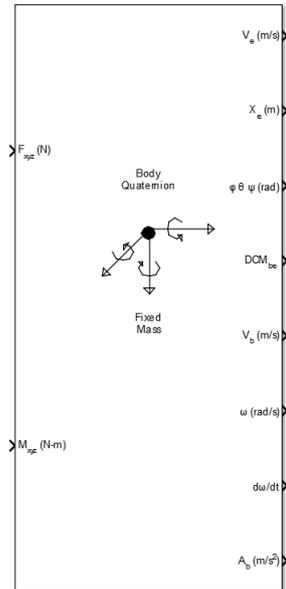
Por su simplicidad, la validación del modelo no lineal se ha realizado haciendo uso de un nuevo modelo no lineal que emplea un bloque ya definido de Simulink (Figura 5.1): “6DOF (Quaternion)”. Este bloque se encarga de resolver el modelo no lineal teniendo como entradas las fuerzas y los momentos.

Así pues el nuevo modelo presenta el mismo conjunto de fuerzas y momentos descritos en el Capítulo 3 y por tanto solamente difiere del modelo no lineal del Capítulo 4 en la resolución de las ecuaciones.

### 5.2 Simulación y comparación

Para llevar a cabo la validación se ha realizado un conjunto de simulaciones y se han obtenido los errores relativos al modelo con el bloque ya definido de Simulink. Un nuevo esquema en Simulink (Figura 5.6) ha sido necesario para simular ambos modelos y el proceso de simulación ha consistido en :

- Establecer un punto de estabilidad.
- Aplicar variaciones sobre el punto de estabilidad.
- Comparar resultados y determinar el error relativo.



**Figura 5.1:** Bloque de Simulink empleado para la resolución del nuevo modelo

### 5.2.1 Variación de ángulos de Euler despreciable

El punto de estabilidad seleccionado ha sido un punto de flotación neutra con un avance constante de  $1\text{ m/s}$ . Entendiéndose por flotación neutra el estado que adquiere la aeronave al tener compensadas las fuerzas verticales  $y$ , por tanto, la resultante nula. Además se ha seleccionado un punto donde los ángulos de Euler son muy pequeños y prácticamente constantes.

Siguiendo el procedimiento descrito se obtienen los resultados de las Figuras 5.2, 5.3, 5.4, 5.5 . La leyenda de cada una de las simulaciones se debe interpretar del siguiente modo:  $V4$  hace referencia a la versión 4 del modelo no lineal y por tanto el modelo a validar. Así pues  $V4 : x$ ,  $V4 : y$ ,  $V4 : z$  son las componentes del vector ilustrado en cada una de las figuras. Del mismo modo ocurre con la versión 4.1, que es el modelo referencia con el bloque de Simulink.  $V4.1 : x$ ,  $V4.1 : y$ ,  $V4.1 : z$ , por tanto, son las componentes en  $x$ ,  $y$ ,  $z$  del vector representado respectivamente.

La Figura 5.2 muestra el vector de las aceleraciones lineales de ambos modelos. En el primer gráfico se observa como el sistema evoluciona hasta llegar al punto de equilibrio descrito. Se mantiene estacionario hasta el instante  $t = 400\text{ s}$  donde se aplica una variación en la inclinación del vector de propulsión,  $\delta$ , de  $0,3\text{ rad}$  ( $17^\circ$  aproximadamente). En el segundo gráfico se aprecia como las mayores diferencias de los resultados obtenidos son del orden de las milésimas y por tanto se puede considerar que ambos modelos razonablemente iguales.

Procediendo del mismo modo, la Figura 5.3, representa los vectores velocidad de cada uno de los modelos. Tras llegar al punto de equilibrio se aplica la

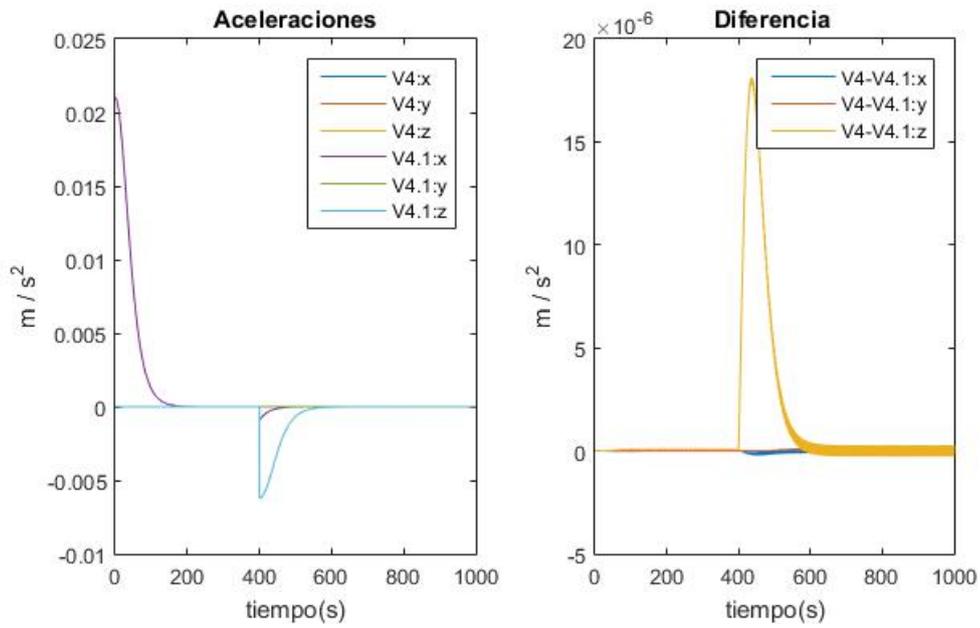


Figura 5.2: Respuesta de las aceleraciones de los modelos

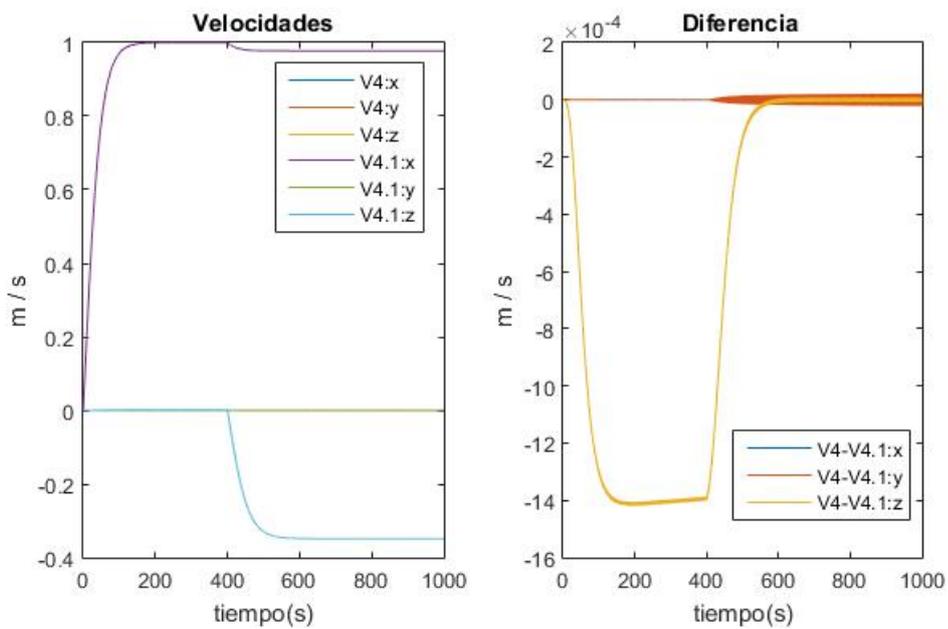


Figura 5.3: Respuesta de las velocidades de los modelos

variación anterior en la inclinación del vector de propulsión y se observa como tras el cambio la velocidad de avance,  $Vx$ , es menor y el módulo de la componente de  $Vz$  crece, indicando con el signo negativo que la aeronave aumenta su altura ya que el eje  $z$  ha sido definido positivo hacia abajo. Esta situación corresponde al comportamiento esperado de la aeronave y de nuevo es apreciable que la diferencia de los dos modelos no lineales es muy pequeña.

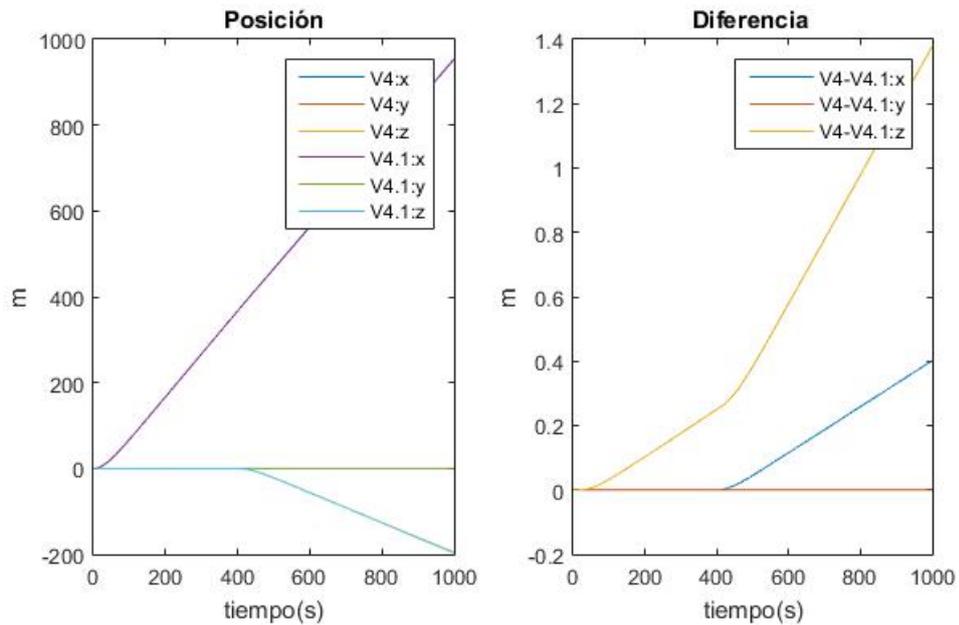


Figura 5.4: Respuesta de las posiciones de los modelos

En la Figura 5.4, en cambio, se aprecia una mayor diferencia en los resultados obtenidos. Esta figura muestra el resultado del vector posición de ambos modelos y, tal y como era de esperar, existe un cambio de tendencia en el instante  $t = 400s$  debido a la variación de la inclinación del vector de propulsión. No obstante el error acumulado final es aproximadamente de  $1,4m$  en el caso de la componente  $z$  y de  $0,4m$  en el caso de la componente  $x$ .

Para determinar si la diferencia el cálculo es de importancia se ha evaluado la cantidad de metros recorridos en cada componente del vector de posiciones frente al error absoluto cometido. Y se obtienen los siguientes resultados:

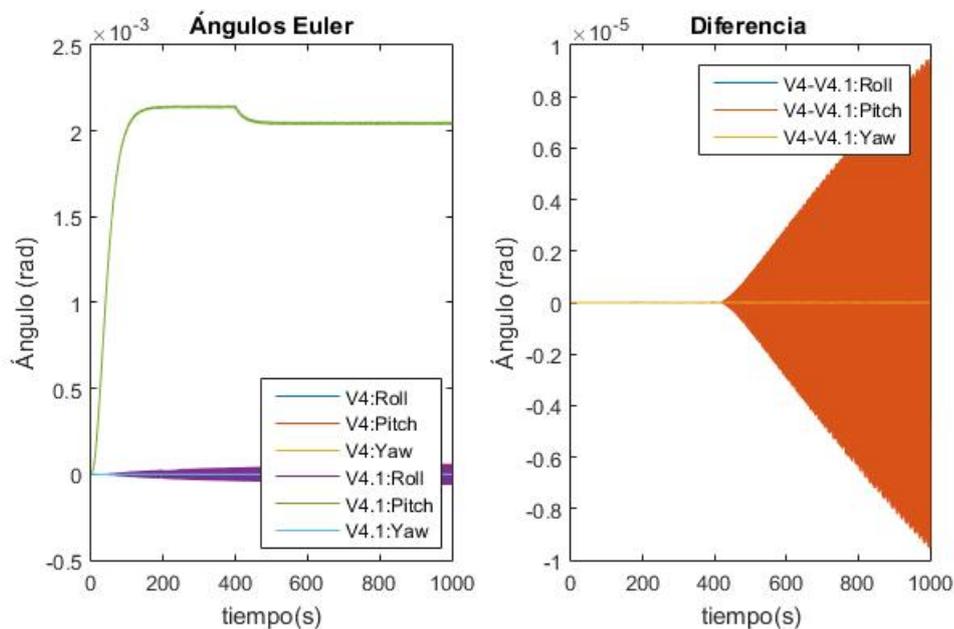
$$\epsilon_x = \frac{|\Delta x|}{x_{total}} = \frac{0,4}{953,6} 100 = 0,042 \%$$

$$\epsilon_z = \frac{|\Delta z|}{z_{total}} = \frac{1,378}{197,1} 100 = 0,7 \%$$

Se asume por tanto que las diferencias, en estas condiciones, no suponen un peso importante teniendo en cuenta los metros desplazados en cada caso.

Recalcar que los denominadores son las distancias recorridas por el modelo V4.1.

El error generado se acentúa con velocidades de desplazamiento mayores, no obstante puesto que el dirigible aquí considerado es de pequeñas dimensiones y se desplaza lentamente no se ha considerado este efecto. El error se genera con la presencia de diferencias en el cálculo de los ángulos de Euler. Estos están presentes en la matriz de transformación y por tanto el error se acumula a medida que se integran los resultados con el tiempo.



**Figura 5.5:** Respuesta de los ángulos de Euler de los modelos

La Figura 5.5 evidencia pequeñas diferencias en el cálculo de los ángulos de Euler que en este caso no son especialmente significantes.

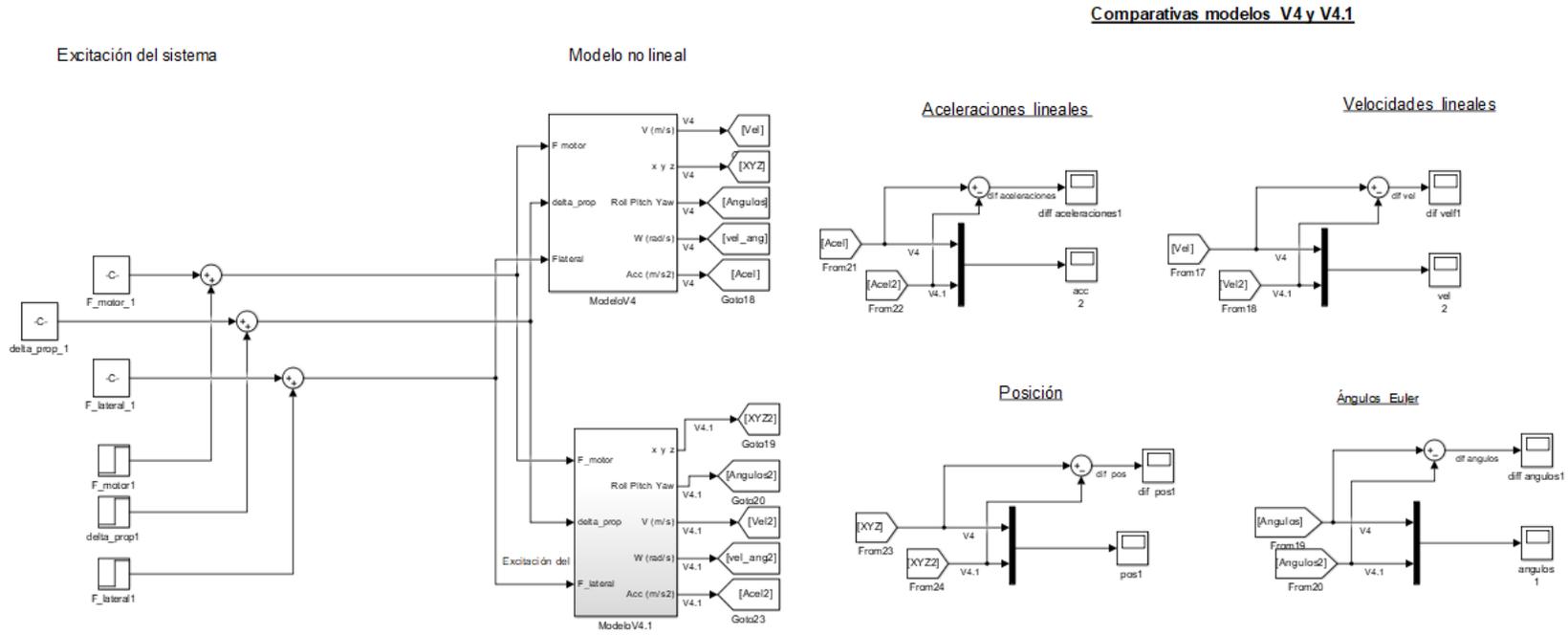


Figura 5.6: Esquema de Simulink para la validación

### 5.2.2 Variación de ángulos de Euler

Los anteriores resultados están calculados integrando directamente las aceleraciones que se obtienen de la resolución de las ecuaciones lo cual solamente es correcto si no se tiene en cuenta la rotación del sistema de referencia.

Por tanto lo que ocurre si se implementa con el esquema de la Figura 5.6 y se intenta seguir una ruta de waypoints, es que los resultados obtenidos son incorrectos. Y por ello el bloque "6DOF (Quaternion)" (Figura 5.1) no solamente incluye la integración, sino también el efecto de la fuerza de Coriolis entre otros cálculos.

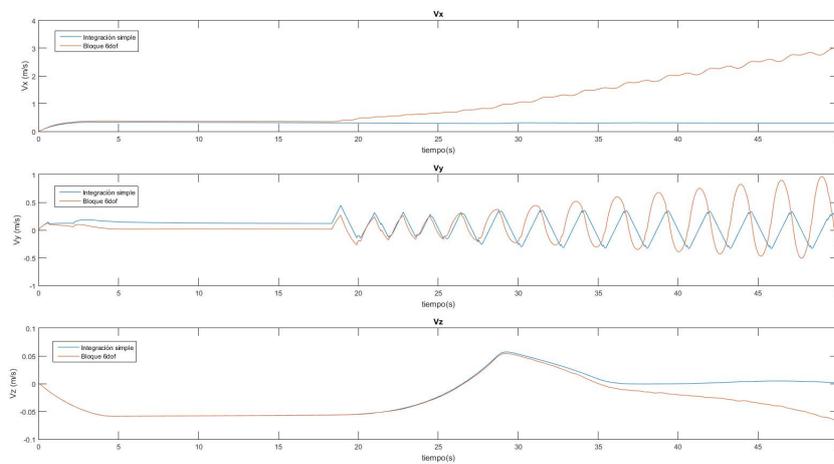


Figura 5.7: Comparativa de las velocidades obtenidas con los dos métodos

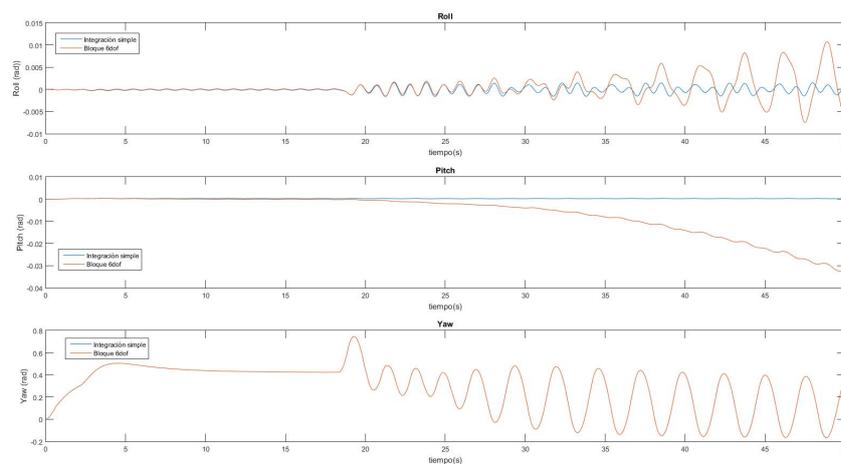


Figura 5.8: Comparativa de ángulos obtenidos con los dos métodos

Las Figuras 5.7 y 5.8 ilustra como los cálculos obtenidos en cada caso son diferentes, siendo correctos los obtenidos con el bloque de Simulink. En azul

se presenta el resultado de la integración pura y en naranja la salida del bloque “6DOF (Quaternion)”. Así pues para la implementación del control y la navegación se hace uso de dicho bloque.

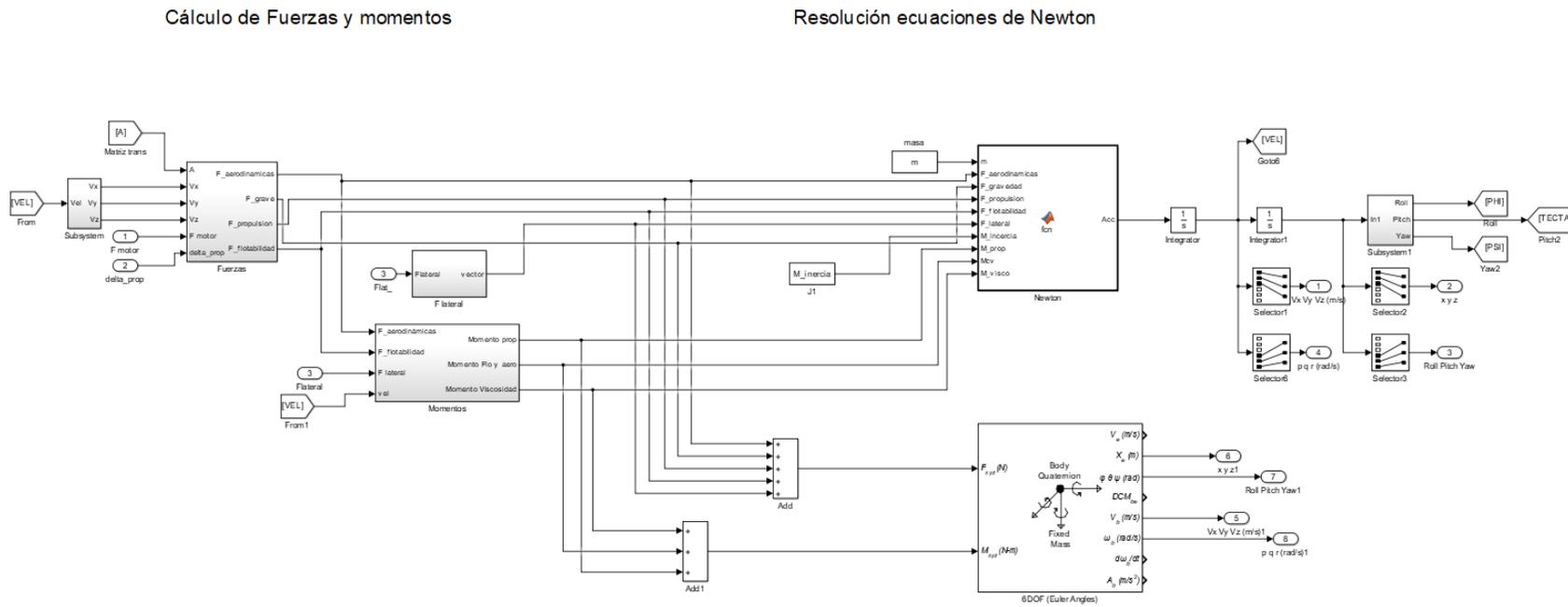


Figura 5.9: Esquema de Simulink para la validación



## Capítulo 6

# Linearización del modelo

Una vez implementado el modelo teórico y validado es necesario obtener un modelo lineal del dirigible que facilite el diseño del control. Existen diferentes métodos para linealizar un modelo matemático. Un método sería , por ejemplo, extender mediante series de Taylor entorno a un punto de funcionamiento cada una de las ecuaciones de Newton utilizadas para la resolución del sistema no lineal y resolver el sistema lineal generado. En cambio en sistemas donde se tiene una caja negra, es decir se conoce como varían las salidas en función de las entradas pero no se conoce la totalidad del proceso. Pues en estos casos se suele aproximar los datos obtenidos en el punto de equilibrio mediante rectas.

Como en el Capítulo 3 se han descrito cada una de las ecuaciones la linealización se realizara haciendo uso de las series de Taylor. Aunque también es posible hacerlo considerando el modelo desarrollado como una caja negra.

Sea el modelo no lineal descrito en el Capítulo 3 definido como:

$$\begin{cases} m\dot{V} = F'_g + F'_f + F_{aero} + F_{prop} \\ J\dot{W} = M_f + M_{aero} + M_{prop} + M_{viscosidad} \end{cases} \quad (6.1)$$

y teniendo en cuenta las fuerzas definidas en 3.6, 3.7, 3.8, 3.9, 3.10 y los momentos definidos en 3.12, 3.13, 3.14 y 3.15 se obtiene:

$$\begin{cases} m\dot{V} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}' + \begin{bmatrix} 0 \\ 0 \\ -(\rho_a - \rho_{he})V_{globog} \end{bmatrix}' + \begin{bmatrix} -\frac{1}{2}\rho_a V_x^2 C_{dx} S_{front} \\ -\frac{1}{2}\rho_a V_y^2 C_{dy} S_{trans} \\ -\frac{1}{2}\rho_a V_x^2 C_l S_{front} - \frac{1}{2}\rho_a V_z^2 C_{dz} S_{trans} \end{bmatrix} + \begin{bmatrix} 2F_{mot} \cos(\delta) \\ 0 \\ -2F_{mot} \sin(\delta) \end{bmatrix} \\ J\dot{W} = \begin{bmatrix} F'_{fz} y_v - F'_{fy} z_v \\ -F'_{fz} x_v + F'_{fx} z_v \\ -F'_{fx} y_v + F'_{fy} x_v \end{bmatrix} + \begin{bmatrix} F_{aero_z} y_v - F_{aero_y} z_v \\ -F_{aero_z} x_v + F_{aero_x} z_v \\ -F_{aero_x} y_v + F_{aero_y} x_v \end{bmatrix} + \begin{bmatrix} -F_{mot_{lateral}} r_x \\ 0 \\ F_{mot_{lateral}} r_z \end{bmatrix} + \begin{bmatrix} -K_{viscosidad} \cdot p \\ -K_{viscosidad} \cdot q \\ -K_{viscosidad} \cdot r \end{bmatrix} \end{cases} \quad (6.2)$$

donde  $m$  es la masa,  $g$  la gravedad,  $\rho_a$  y  $\rho_{he}$  son las densidades del aire y helio respectivamente,  $C_{dx}$ ,  $C_{dy}$ ,  $C_{dz}$ ,  $C_l$  son coeficientes aerodinámicos,  $S_{front}$  y  $S_{trans}$  son la superficie frontal y transversal,  $J$  es la matriz de inercia,  $(x_v, y_v, z_v)$  son las coordenadas del centro de volumen,  $r_x, r_z$  son la distancia desde el punto de aplicación de la fuerza al eje  $x$  y  $z$  respectivamente,  $r$  es la velocidad angular en el eje  $z$  y  $K_{viscosidad}$  es un coeficiente empírico.

Como se va a linealizar respecto de las velocidades se ha considerado las aceleraciones angulares y lineales nulas. Además se ha considerado que la aeronave esta orientada en su forma natural y que se encuentra avanzando a  $0.35 \text{ m/s}$ , elevándose a  $0.2 \text{ m/s}$  y girando sobre si mismo a  $0.005 \text{ rad/s}$ . El resto de variables se obtienen resolviendo el sistema de ecuaciones no lineal (ec. 6.2) y se obtiene:

**Tabla 6.1:** Punto de equilibrio

$\ddot{V}_x$	0	$\dot{V}_x$	0.35 m/s
$\ddot{V}_y$	0	$\dot{V}_y$	0
$\ddot{V}_z$	0	$\dot{V}_z$	-0.2 m/s
$\dot{p}$	0	$p$	0
$\dot{q}$	0	$q$	0
$\dot{r}$	0	$r$	0.0015 rad/s
$F_{mot}$	$7,4 \cdot 10^{-4} \text{ N}$	$\delta$	0.6705 rad
$F_{lateral}$	$2,14 \cdot 10^{-4} \text{ N}$	Pitch	$2,6 \cdot 10^{-4} \text{ rad}$
Roll	0		

Una vez obtenido el punto de operación se aproxima el sistema empleando una expansión en series de Taylor alrededor del punto de operación.

$$f(x(t)) = f(x_{op}) + \left. \frac{df(\cdot)}{dx} \right|_{x_{op}} (x - x_{op})$$

La implementación en Matlab de la linearización se ha realizado utilizando el código del Apéndice B. A continuación se muestra la linearización de una de las ecuaciones:

```

% Linealiza y transformada de Laplace en el punto de equilibrio
syms s xpunto dxpunto ;
% %%% EQ1 %%%
% cambio de variable
eq1=subs(eq1, [d2x, dx],[dxpunto, xpunto]);

Variables_eq1=[delta_prop F_motor dxpunto xpunto];
pto_func_eq1=[delta_prop_0 F_motor_0 d2x_0 dx_0];

% Derivadas parciales en el punto de operación
coefl_dxpunto = subs(diff(eq1, dxpunto), Variables_eq1, pto_func_eq1);

```

```

coef1_xpunto = subs(diff(eq1, xpunto), Variables_eq1, pto_func_eq1);
coef1_F_motor = subs(diff(eq1, F_motor), Variables_eq1, pto_func_eq1);
coef1_delta_prop = subs(diff(eq1, delta_prop), Variables_eq1,
    pto_func_eq1);

%1ª Ecuación linealizada
eq1l = coef1_dxpunto*s*xpunto + coef1_xpunto*xpunto + coef1_F_motor*
    F_motor + coef1_delta_prop*delta_prop;

```

Tras obtener el sistema lineal es posible aplicar la transformada de Laplace y obtener las funciones de transferencia que relacionan cada una de las variables de entrada con las salidas.

Por ejemplo para  $V_x$  se implementa de tal modo:

```

%% Resolución de las salidas xpunto, ypunto, zpunto, Rollpunto,
    Pitchpunto, Yawpunto
fdts=solve(eq1l, eq2l, eq3l, eq4l, eq5l, eq6l, xpunto, ypunto, zpunto,
    Rollpunto, Pitchpunto, Yawpunto);

s=tf('s');

escalado_mot=0.2644; %[N] motor comercial (propulsión máxima)
escalado_ang= 1.57; %[rad] ángulo máximo (90°).
% VX
VxFlat=collect(subs(fdts.xpunto, [F_motor, delta_prop], [0 0]),
    F_lateral_mot)/F_lateral_mot;
GVxFlat=eval(char(VxFlat));
GVxFlat=escalado_mot*GVxFlat;

VxFprop=collect(subs(fdts.xpunto, [delta_prop, F_lateral_mot], [0 0]),
    F_motor)/F_motor;
GVxFprop=eval(char(VxFprop));
GVxFprop=GVxFprop*escalado_mot;

VxDelta=collect(subs(fdts.xpunto, [F_lateral_mot, F_motor], [0 0]),
    delta_prop)/delta_prop;
GVxDelta=eval(char(VxDelta));
GVxDelta=GVxDelta*escalado_ang;

Gxyz=[];
Gxyz=[Gxyz; GVxFprop GVxDelta GVxFlat];

```

Del mismo modo para la velocidad angular roll:

```

% Velocidad PHI - ROLL
VRollFlat=collect(subs(fdts.Rollpunto, [F_motor, delta_prop], [0 0]),
    F_lateral_mot)/F_lateral_mot;
GVRollFlat=eval(char(VRollFlat));
GVRollFlat=GVRollFlat*escalado_mot;

VRollFprop=collect(subs(fdts.Rollpunto, [delta_prop, F_lateral_mot], [0
    0]), F_motor)/F_motor;
GVRollFprop=eval(char(VRollFprop));
GVRollFprop=GVRollFprop*escalado_mot;

```

$$\begin{bmatrix} V_x \\ V_y \\ V_z \\ p \\ q \\ Yaw \end{bmatrix} = \begin{bmatrix} \frac{V_x}{F_{prop}} = \frac{0,91908}{s+0,0147} & \frac{V_x}{\delta} = \frac{-0,0032028}{s+0,0147} & \frac{V_x}{F_{lateral}} = 0 \\ \frac{V_y}{F_{prop}} = 0 & \frac{V_y}{\delta} = 0 & \frac{V_y}{F_{lateral}} = \frac{0,5865}{s} \\ \frac{V_z}{F_{prop}} = \frac{-0,72883}{s+0,0204} & \frac{V_z}{\delta} = \frac{-0,0040389}{s+0,0204} & \frac{V_z}{F_{lateral}} = 0 \\ \frac{p}{F_{prop}} = 0 & \frac{p}{\delta} = 0 & \frac{p}{F_{lateral}} = 0 \\ \frac{q}{F_{prop}} = 0 & \frac{q}{\delta} = 0 & \frac{q}{F_{lateral}} = 0 \\ \frac{Yaw}{F_{prop}} = 0 & \frac{Yaw}{\delta} = 0 & \frac{Yaw}{F_{lateral}} = \frac{1,4576}{s(s+0,7875)} \end{bmatrix} \cdot \begin{bmatrix} F_{prop} \\ \delta \\ F_{lateral} \end{bmatrix}$$

**Figura 6.1:** Matriz de funciones de transferencia

```

VRollDelta=collect(subs(fdts.Rollpunto,[F_lateral_mot,F_motor],[0
0]),delta_prop)/delta_prop;
GVRollDelta=eval(char(VRollDelta));
GVRollDelta=GVRollDelta*escalado_ang;

Grpy=[];
Grpy=[Grpy; GVRollFprop GVRollDelta GVRollFlat];
    
```

En el Apéndice B se encuentra el resto de código para obtener las funciones de transferencia.

Se obtiene por tanto una matriz de funciones de transferencia, Figura 6.1, que relaciona las entradas,  $F_{prop}$ ,  $\delta$ ,  $F_{lateral}$ , con las salidas,  $V_x$ ,  $V_y$ ,  $V_z$ ,  $p$ ,  $q$  y  $Yaw$ .

Siendo las variables de interés  $V_x$ ,  $V_z$  y  $Yaw$ . Es posible apreciar en la Figura 6.1 que estamos antes un sistema MIMO aunque se asume para el control que es un sistema SISO desacoplado. Así pues las funciones de transferencia de interés son:

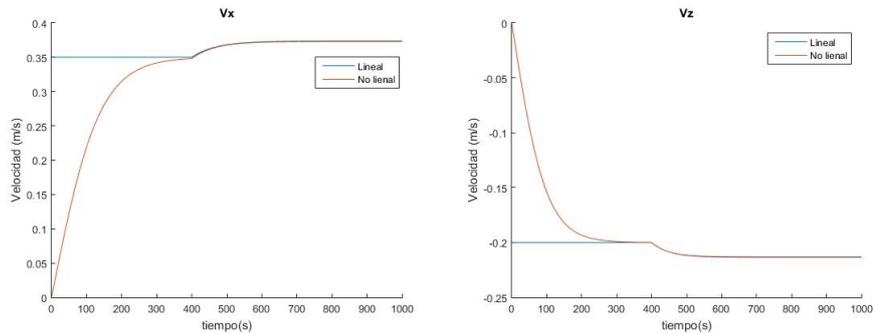
$$\frac{V_x}{F_{prop}} = \frac{0,91908}{s + 0,0147}$$

$$\frac{V_z}{\delta} = \frac{-0,0040389}{s + 0,0204}$$

y

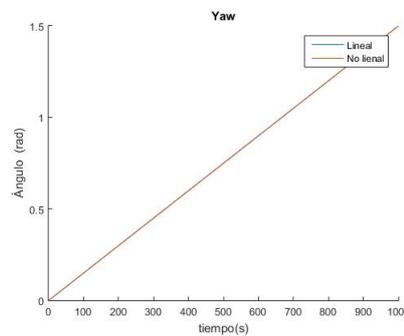
$$\frac{Yaw}{F_{lateral}} = \frac{1,4576}{s(s + 0,7875)}$$

La Figura 6.2 muestra la respuesta del modelo no lineal y lineal tras aplicar un escalón de  $1 \cdot 10^{-4}$  N en la fuerza del motor en el instante  $t = 400s$ . Se aprecia con claridad que la aproximación realizada bastante buena.



(a) Respuesta de la velocidad de avance

(b) Respuesta de la velocidad de elevación



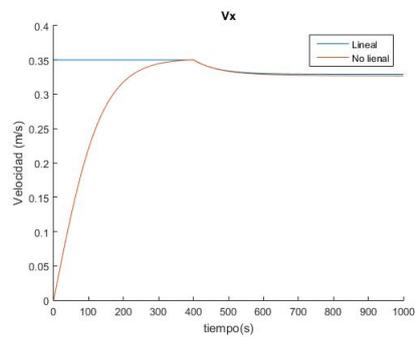
(c) Respuesta del Yaw

**Figura 6.2:** Respuesta ante un escalón de la fuerza del motor

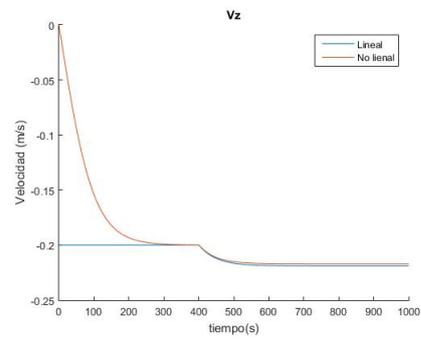
De nuevo se realiza otra simulación pero esta vez con otra variable de control, concretamente  $\delta$ . Se aplica un escalón de 0.15 rad en el instante  $t = 400s$  y se obtienen los resultados de la Figura 6.3.

Por último se comprueba la última variable controlada,  $F_{lateral}$ . Procediendo del mismo modo se obtienen ante un escalón de  $1 \cdot 10^{-4}$  N en el instante  $t = 400s$  los resultados de la Figura 6.4.

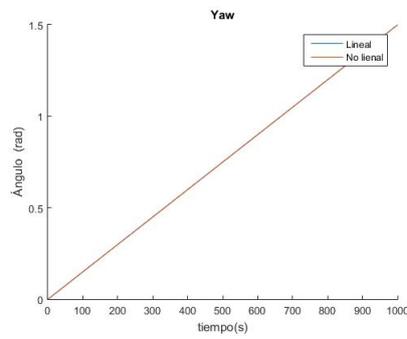
Se observa como también en esta simulación el modelo lineal responde de la misma forma que el modelo no lineal y por tanto con estas tres simulaciones se puede concluir que el modelo lineal se aproxima al modelo no lineal lo suficiente como para poder realizar el control basándose en este.



(a) Respuesta de la velocidad de avance



(b) Respuesta de la velocidad de elevación



(c) Respuesta del Yaw

**Figura 6.3:** Respuesta ante un escalón en la inclinación de los motores

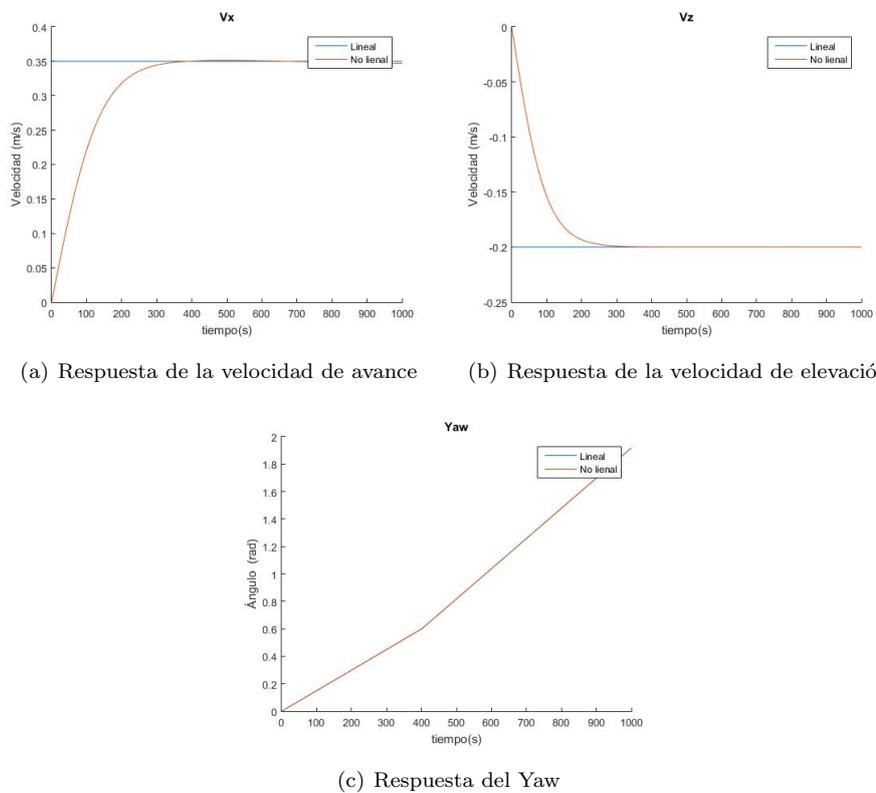


Figura 6.4: Respuesta ante un escalón en la fuerza del motor lateral o de cola



## Capítulo 7

# Implementación del sistema de control

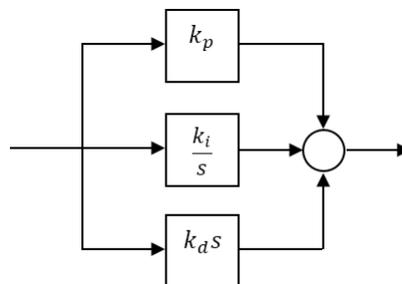
Este capítulo pretende ilustrar la estrategia de control adoptada para el control de las variables de estado de interés:  $V_x$ ,  $z$  y  $Yaw$ .

### 7.1 Diseño

Por su simplicidad el control es llevado a cabo mediante PID, cuya expresión matemática en forma paralelo viene expresada como:

$$C_{paralelo} = k_p + \frac{k_i}{s} + k_d s$$

y siendo su representación la mostrada en la Figura 7.1



**Figura 7.1:** PID en forma paralelo

Una vez seleccionado el tipo de controladores es necesario ajustar cada uno de los parámetros de los controladores. Para ello se ha seguido el método

SIMC [10]. Este método tiene la ventaja que sólo es necesario un parámetro de diseño,  $\tau_c$ , para obtener los valores de  $K_c$ ,  $\tau_i$  y  $\tau_d$ .

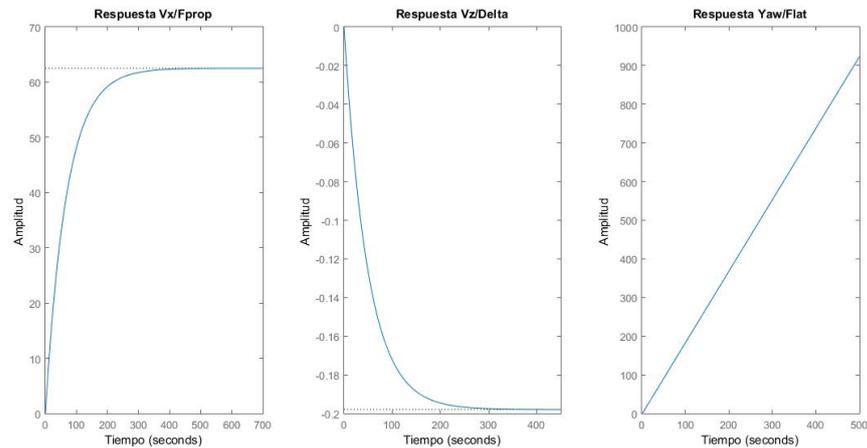
En el Capítulo 6 se han definido las funciones de transferencia que sintetizan las dinámicas de interés:

$$\frac{V_x}{F_{prop}} = \frac{0,91908}{s + 0,0147}$$

$$\frac{V_z}{\delta} = \frac{-0,0040389}{s + 0,0204}$$

y

$$\frac{Yaw}{F_{lateral}} = \frac{1,4576}{s(s + 0,7875)}$$



**Figura 7.2:** Respuesta de las funciones de transferencia ante un escalón .

Estas funciones de transferencias están escaladas, de modo que se ha tenido en cuenta datos de motores comerciales para aeronaves de radio control y el ángulo máximo de inclinación de estos. Se ha establecido una relación lineal donde los motores proporcionarán como máximo un empuje de  $0,2644 N$  y el ángulo máximo de inclinación que pueden alcanzar los motores será de  $1,57 rad (90^\circ)$ .

Limitando el dominio de actuación de los controladores,  $[0, 1]$  en el caso del PI que controla  $V_x$ ,  $[-1, 1]$  en el caso del PI que controla  $V_z$  y  $[-0,35, 0,35]$  para el PID que controla el  $Yaw$  se consigue una interpretación en tanto por uno mucho más clara de la acción de control. El dominio del motor lateral es menor ya que este, además de afectar directamente al  $Yaw$  también afecta a  $V_y$ , la cual no va a ser controlada.

Una vez obtenidas las funciones de transferencia se procede al diseño de los controladores considerando el sistema desacoplado. Para ello de [10] se obtiene que para un sistema de primer orden de la forma:

$$g_1(s) = \frac{k}{\tau_1 s + 1} e^{-\theta s} = \frac{k'}{s + 1/\tau_1} e^{-\theta s}$$

Al aplicar el método SIMC resulta en un controlador PI con los siguientes parámetros:

$$K_c = \frac{1}{k} \frac{\tau_1}{\tau_c + \theta} = \frac{1}{k'} \frac{1}{\tau_c + \theta}$$

$$\tau_I = \min \{ \tau_1, 4(\tau_c + \theta) \}$$

donde  $\tau_c$  es la constante de tiempo del lazo cerrado que se desea. Así se obtienen los siguientes parámetros para las funciones de transferencia  $\frac{V_x}{F_{prop}}$  y  $\frac{V_z}{\delta}$ :

$PI_{V_x}$		$PI_{V_z}$	
$K_c$	1.0880	$K_c$	-247.5922
$\tau_I$	4	$\tau_I$	4
$\tau_c$	1	$\tau_c$	1

**Tabla 7.1:** Parámetros obtenidos con el método SIMC para un sistema de primer orden.

De la misma forma que para un sistema de primer orden en [10] también establece una guía para obtener los parámetros del controlador para el caso de un sistema con integrador expresado como:

$$g_2(s) = \frac{k'}{s(\tau_2 s + 1)} e^{-\theta s}$$

por tanto siguiendo el método se obtiene un PID como controlado estando sus parámetros definidos por:

$$K_c = \frac{1}{k'} \frac{1}{\tau_c + \theta}$$

$$\tau_I = 4(\tau_c + \theta)$$

$$\tau_D = \tau_2$$

Así para para controlar el *Yaw* se obtienen los siguientes valores del Cuadro 7.2.

Los valores que se obtienen con este método son para la forma serie del PID, no obstante Simulink necesita los del controlador PID en forma paralelo. Teniendo en cuenta que la forma serie viene expresada como:

$$C_{serie} = K_c \left( \frac{\tau_I s + 1}{\tau_I s} \right) (\tau_D s + 1)$$

$PID_{Yaw}$	
$K_c$	0.5403
$\tau_I$	4
$\tau_D$	1.2698
$\tau_c$	1

**Tabla 7.2:** Parámetros obtenidos con el método SIMC para un sistema con integrador.

se pueden obtener los valores para el PID en forma paralelo aplicando las siguientes expresiones:

$$k_p = K_c(1 + \tau_D/\tau_I)$$

$$k_i = K_c/\tau_I$$

$$k_d = K_c\tau_D$$

$PI_{V_x}$		$PI_{V_z}$		$PID_{Yaw}$	
$k_p$	1.0880	$k_p$	-247.5922	$k_p$	0.7118
$k_i$	0.2720	$k_i$	-61.8980	$k_i$	0.1351
$k_d$	0	$k_d$	0	$k_d$	0.6861

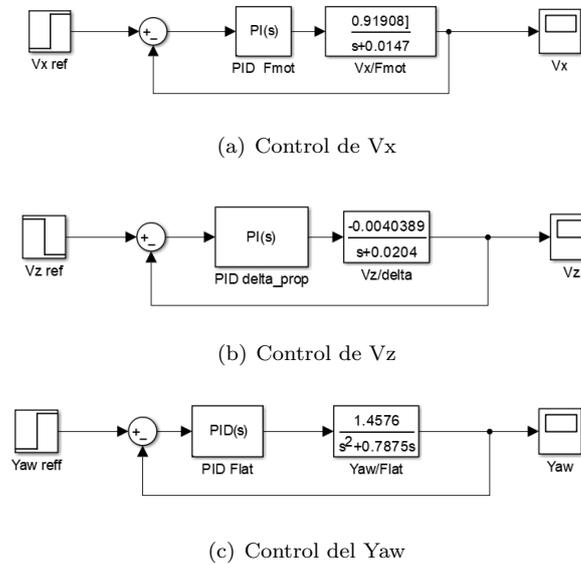
**Tabla 7.3:** Valores de los controladores en forma paralelo

## 7.2 Validación

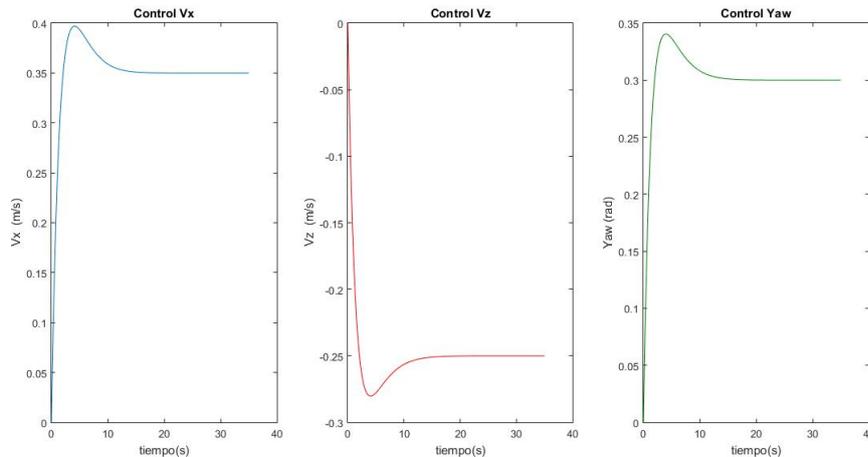
### 7.2.1 Sistema SISO

Llegados a este punto se está en disposición de validar si los controladores están correctamente ajustados. Para ello se ha analizado el control en lazo cerrado simple (Figura 7.3) para cada una de las funciones de transferencia obtenidas con anterioridad.

Es posible observar como con la primera aproximación para ajustar los parámetros de los controladores mediante el método SIMC es posible controlar  $V_x, V_z$  y  $Yaw$  (Figura 7.4). Nótese la existencia de sobreoscilaciones en cada una de las respuestas. Puesto que el sistema real del dirigible que se ha desarrollado en los capítulos es multivariable acoplado estos resultados son suficientes como primera aproximación para continuar con el desarrollo del control teniendo en cuenta esta característica.



**Figura 7.3:** Control desacoplado de lazo cerrado



**Figura 7.4:** Respuestas del control desacoplado de  $V_x$ ,  $V_z$  y  $Yaw$  respectivamente

### 7.2.2 Sistema MIMO lineal

En el Capítulo 6 se hizo evidente que el sistema está acoplado y prueba de ello es la matriz de funciones de transferencia que se obtuvo en el capítulo (Figura 6.1). Por ello se ha llevado a cabo un nuevo esquema de control (Figura 7.8) donde se incluyen todas las dinámicas simultáneamente.

En esta nueva alternativa se ha apostado por una configuración en cascada para el control de la altura de modo que el lazo externo actúa sobre la altura e marca la referencia al lazo interno que controla la velocidad de  $z$ . El controlador proporcional introducido en el control en cascada se ha ajustado haciendo uso de la herramienta 'rltool' proporcionada por Matlab. Por otro

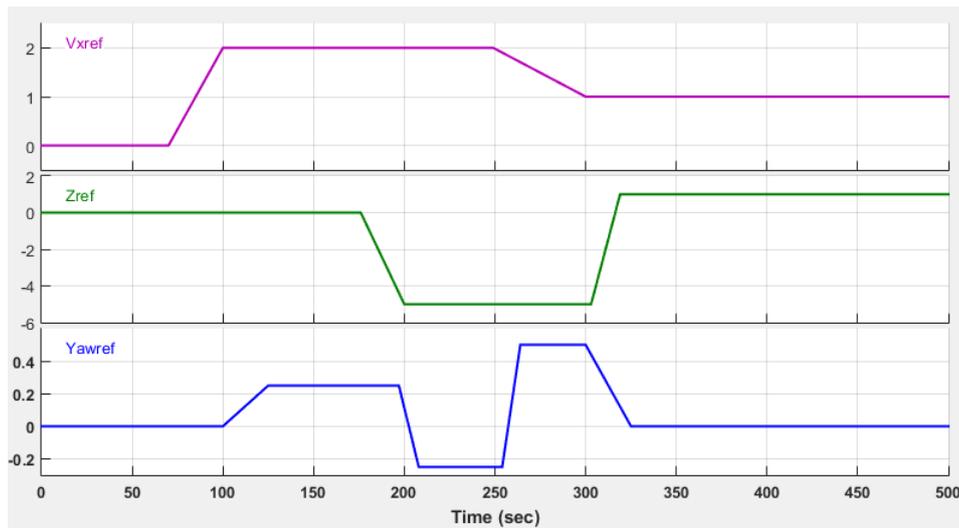
lado, el control de  $V_x$  y  $Yaw$  se ha procedido con la misma configuración que en sistema SISO.

El modelo que se utilizó para las simulaciones se trata del modelo lineal en su forma de espacio de estado que viene representada de forma general tal que así:

$$\begin{aligned}\dot{x}(t) &= A x(t) + B u(t) \\ y(t) &= C x(t) + D u(t)\end{aligned}$$

siendo  $A$  la matriz de estados,  $B$  la matriz de entradas,  $C$  la matriz de salidas y  $D$  la matriz de transmisión directa.

En la simulación se han generado un conjunto de referencias que varían con el tiempo para así ver las posibles interferencias (Figura 7.5) causadas por los controladores.



**Figura 7.5:** Referencias utilizadas en la simulación para el modelo lineal

Es posible observar en la Figura 7.6 como el sistema de control es capaz de alcanzar las distintas consignas durante toda la simulación. Remarcar que los resultados mostrados son variaciones sobre el punto de funcionamiento considerado.

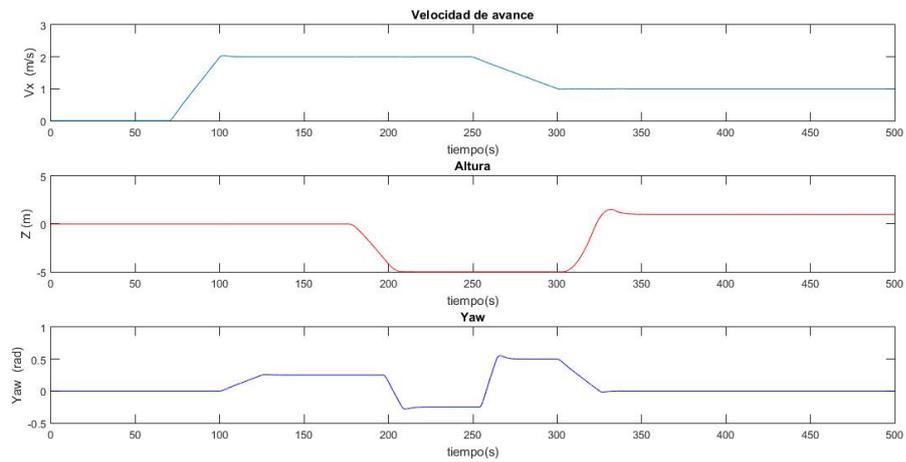


Figura 7.6: Resultado de la simulación con el modelo MIMO lineal

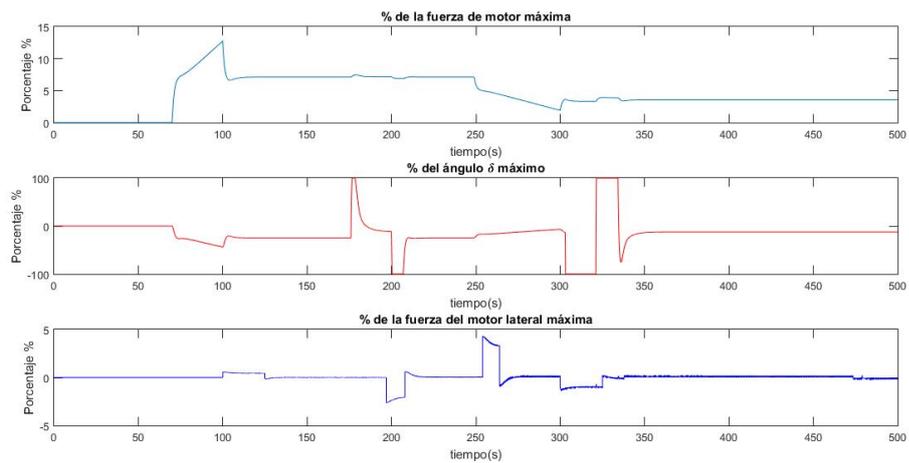


Figura 7.7: Acciones de control de la simulación con el modelo MIMO lineal

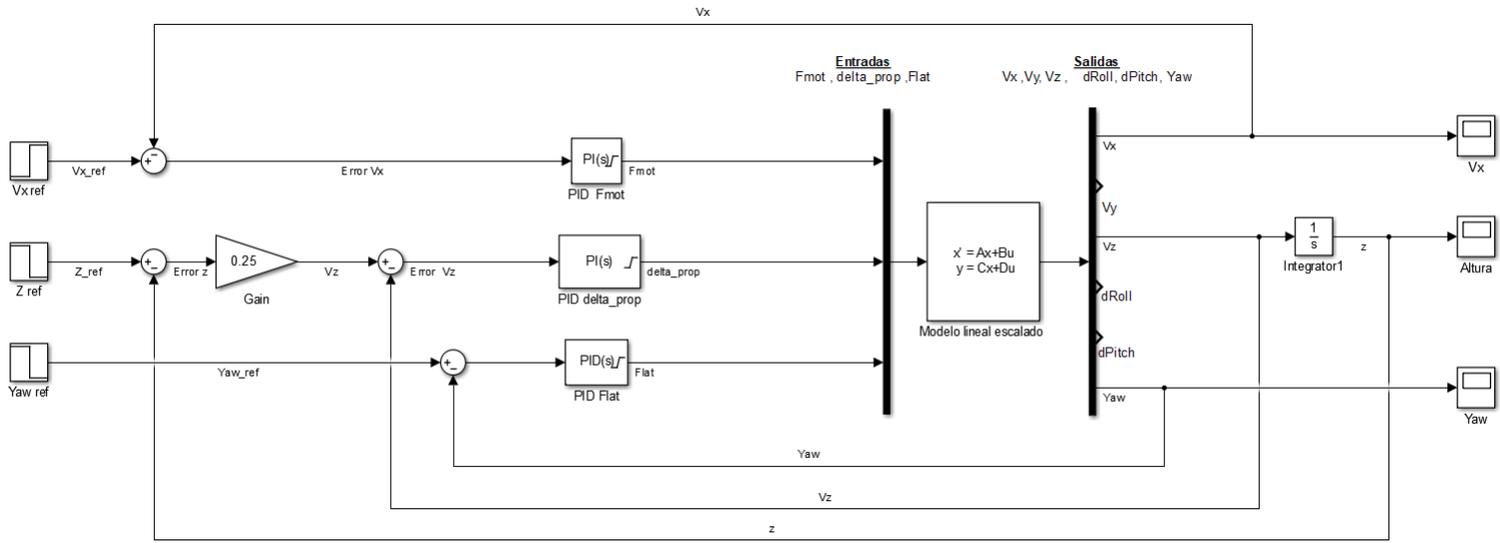
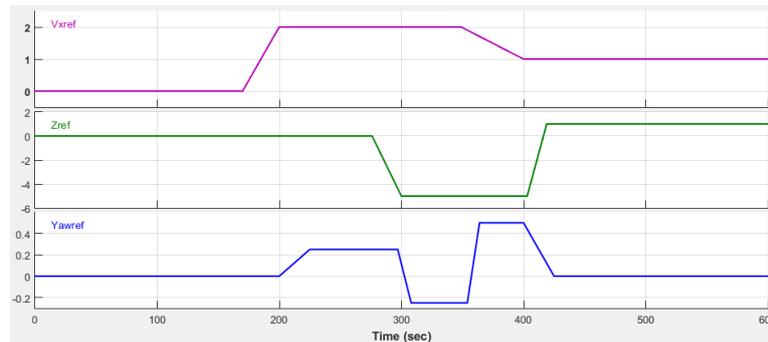


Figura 7.8: Esquema de Simulink para el control del sistema lineal

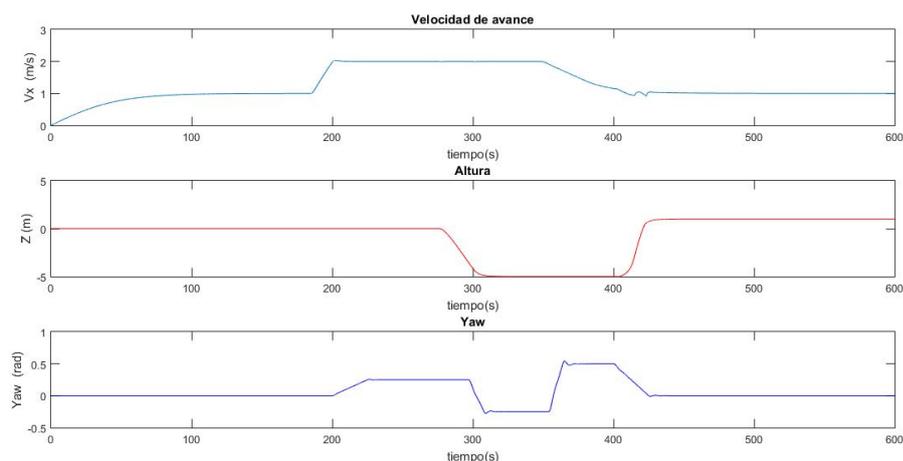
### 7.2.3 Sistema MIMO no lineal

Aplicando la misma estrategia (Figura 7.15) de control pero esta vez introduciendo el modelo no lineal es posible obtener resultados muy similares. En esta simulación se mantienen las magnitudes de las referencias pero se retrasa su aplicación en el tiempo para dejar que el sistema evolucione previamente hasta el punto de funcionamiento.



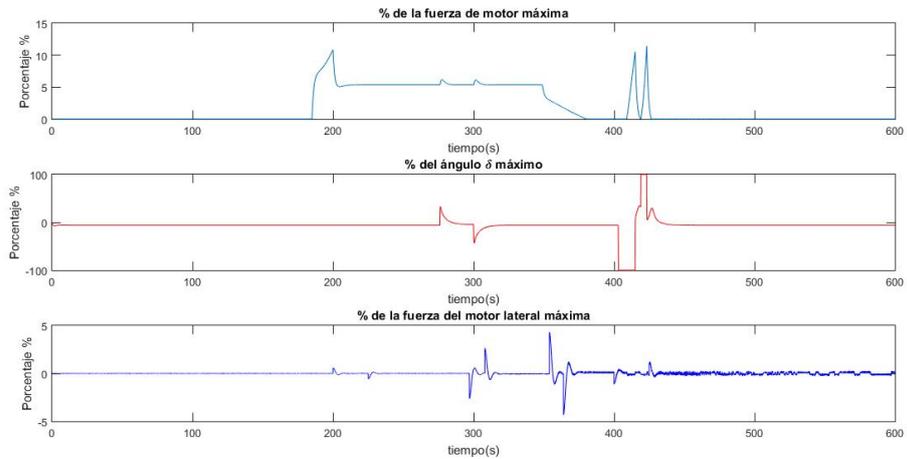
**Figura 7.9:** Referencias utilizadas en la simulación para el modelo no lineal

Comparando los resultados simulados con los obtenidos con el modelo lineal es posible identificar una gran similitud. No obstante las acciones de control son ligeramente menos exigentes en el caso del modelo no lineal. La naturaleza multivariable del sistema se hace visible en el instante  $T = 420s$  cuando el controlador en cascada, que controla la altura, afecta a la velocidad de avance de la aeronave, no dando tiempo al controlador de la velocidad de mantener la referencia.



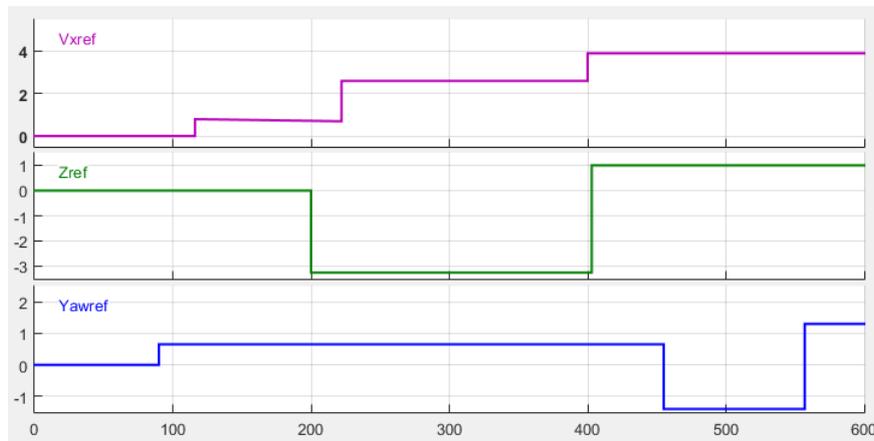
**Figura 7.10:** Resultado de la simulación con el modelo MIMO no lineal

Se realiza otra simulación para verificar que la respuesta del control. En la simulación de la Figura 7.13 y 7.14 se ha aumentado la exigencia al contro-



**Figura 7.11:** Acciones de control de la simulación con el modelo MIMO no lineal

lador realizando cambios de referencias más bruscos y elevados con el fin de observar comportamientos con mayor inestabilidad (Figura 7.12).



**Figura 7.12:** Referencias utilizadas en la segunda simulación para el modelo no lineal

A la vista de los resultados de esta simulación se aprecia como el control de la altura tiene dificultades para alcanzar la referencia. En la Figura 7.14 se aprecia a partir del instante  $t = 400s$  como la acción de control oscila entre los valores máximos y mínimos debido al salto de  $4 m$  introducido en la referencia. Es posible apreciar también como el control de  $V_x$  va contrarrestando las oscilaciones derivadas del control de altura debido a la naturaleza multivariable del sistema.

En definitiva, el control implementado en este capítulo es válido en situaciones con pequeñas variaciones en las referencias ya que se ha visto como el control de la altura oscila con cambios bruscos.

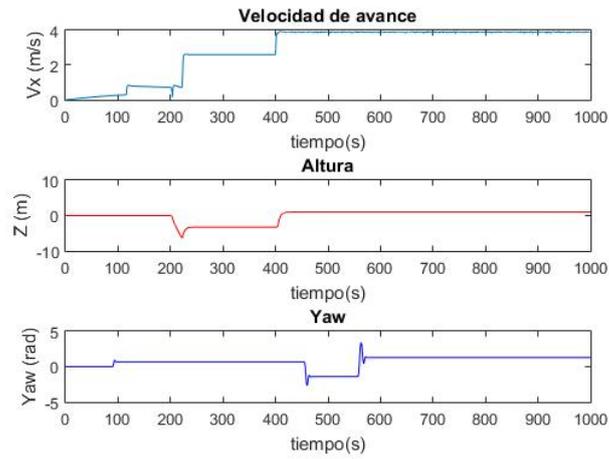


Figura 7.13: Resultado de la segunda simulación con el modelo MIMO no lineal

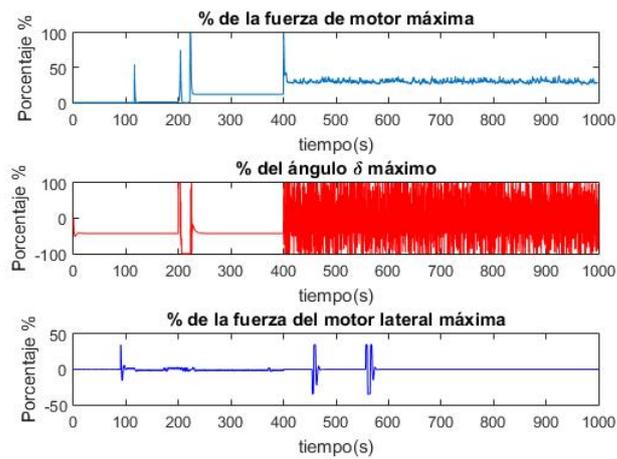


Figura 7.14: Acciones de control de la segunda simulación con el modelo MIMO no lineal

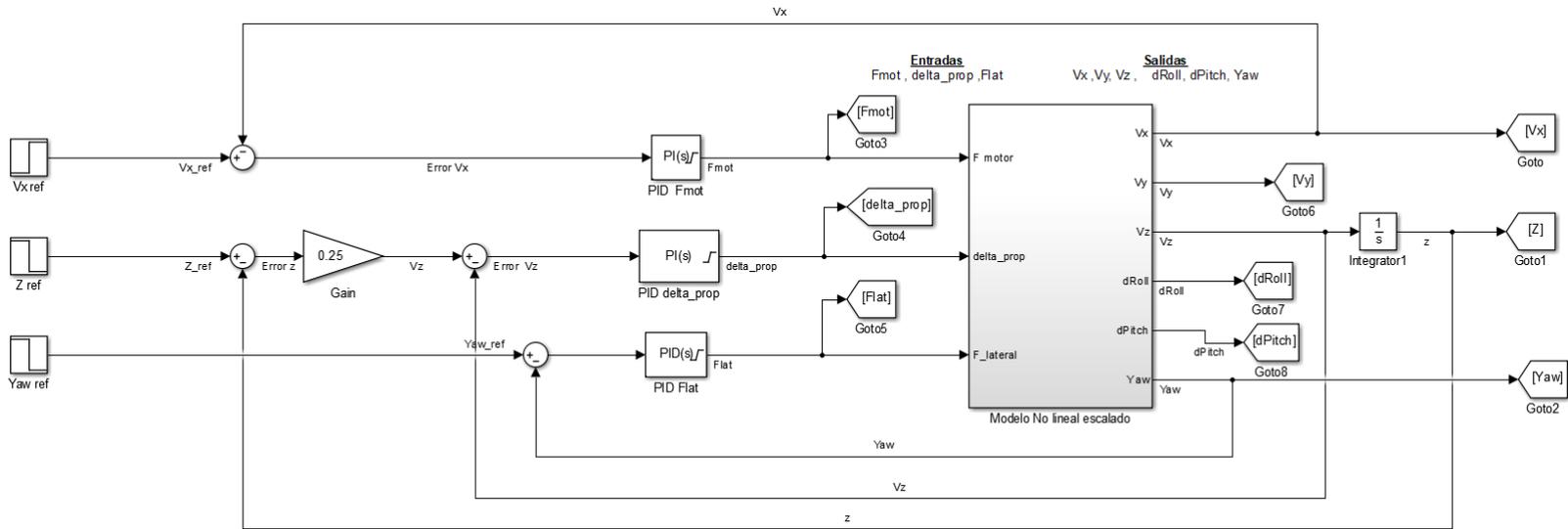


Figura 7.15: Esquema de Simulink para el control del sistema no lineal

## Capítulo 8

# Navegación

Este capítulo pretende ilustrar al lector como se ha implementado el algoritmo de navegación.

Se desea que la aeronave se desplace de un punto origen a un punto destino. Es necesario, por tanto, tener conocimiento de estos puntos los cuales conforman la trayectoria a seguir. Así el vector de la trayectoria a seguir queda definido como:

$$\vec{AB} = (x_B - x_A, y_B - y_A)$$

- A: Coordenadas del punto de origen.
- B: Coordenadas del punto de destino.
- C: Coordenadas de la posición de la aeronave.

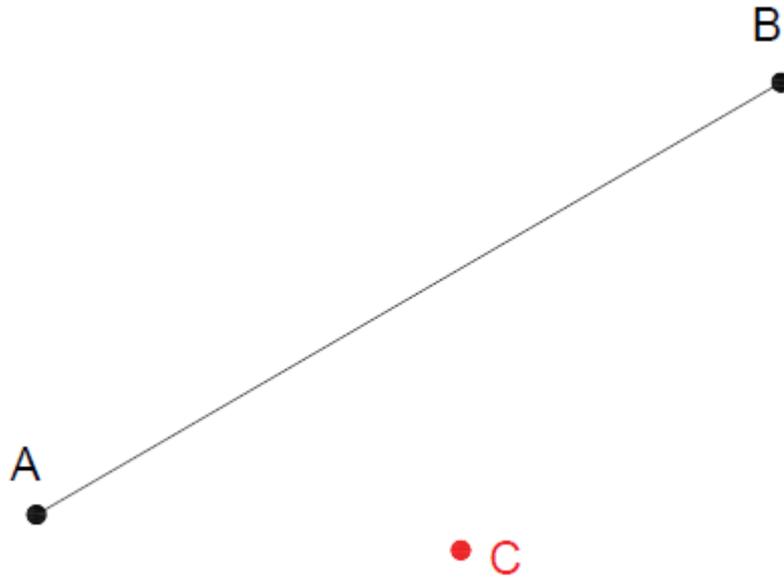
Tomemos por tanto la proyección del punto C sobre la trayectoria AB :

$$D_0 = \frac{\vec{AB} \cdot \vec{AC}}{|\vec{AB}|} \frac{\vec{AB}}{|\vec{AB}|}$$

de este modo se obtiene el vector libre el cual se referencia al punto de origen, A. Así se obtiene el punto D:

$$D = D_0 + A$$

Con este procedimiento es posible orientar la aeronave para alcanzar la trayectoria deseada, y una vez sobre ella, seguirla. Este guiado se conoce como guiado elemental.



**Figura 8.1:** Trayectoria AB a seguir y posición de la aeronave

Existe otro guiado que a la vez que sitúa la aeronave sobre la trayectoria deseada avanza hacia el punto de destino. A este guiado se le conoce como guiado arroyado y consiste en generar un nuevo punto E como un punto ligeramente más avanzado que la proyección D. De modo que E se define como:

$$E = D + k \cdot \overrightarrow{u_{AB}}$$

donde  $\overrightarrow{u_{AB}}$  es el vector unitario definido como:

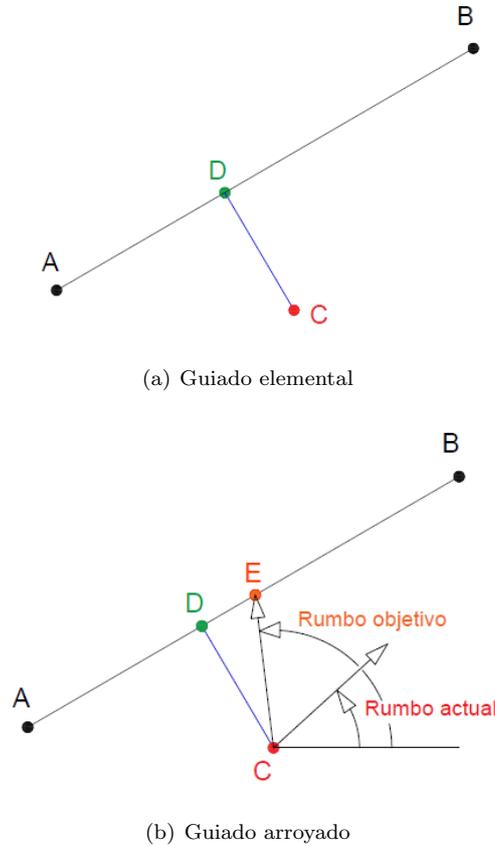
$$\overrightarrow{u_{AB}} = \frac{\overrightarrow{AB}}{|\overrightarrow{AB}|}$$

y siendo  $k$  el porcentaje del vector unitario a añadir al punto D. De modo que para valores de  $k$  pequeños se obtienen resultados en la navegación más agresivos y para valores de  $k$  mayores la respuesta es más suave. En el caso particular que aquí se presenta se ha tomado:

$$k = 0,25$$

Tomando por tanto el guiado arroyado podemos expresar el rumbo como:

$$\text{Rumbo objetivo} = \arctan\left(\frac{y_{\overrightarrow{CE}}}{x_{\overrightarrow{CE}}}\right)$$



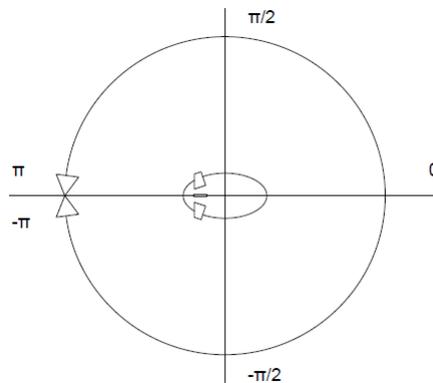
**Figura 8.2:** Estrategias de guiado

Como es conocido la función  $arctan$  devuelve valores de ángulos entre  $\pi/2$  y  $-\pi/2$  los cuales deben ser trasladados en función de la posición relativa de los puntos C y E para su correcta interpretación. Por ello en la implementación se emplea una función conocida como  $atan2$  y que define como:

$$atan2(y, x) = \begin{cases} arctan(\frac{y}{x}) & si\ x > 0 \\ arctan(\frac{y}{x}) + \pi & si\ x < 0\ e\ y \geq 0 \\ arctan(\frac{y}{x}) - \pi & si\ x < 0\ e\ y < 0 \\ +\frac{\pi}{2} & si\ x = 0\ e\ y > 0 \\ -\frac{\pi}{2} & si\ x = 0\ e\ y < 0 \\ no\ definida & si\ x = 0\ e\ y = 0 \end{cases}$$

cuyo dominio es  $(-\pi, \pi]$ . Por tanto la señal de referencia del controlador de Yaw puede tomar valores entre  $(-\pi, \pi]$ .

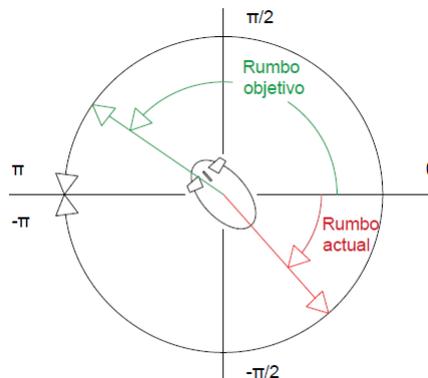
No obstante el error obtenido al restar la referencia y la variable controlada genera un nuevo ángulo que no tiene por qué estar en este dominio de acción. El ángulo error se debe de transformar para que esté en el dominio. Para ello se aplica:



**Figura 8.3:** Dominio del rumbo

$$Error_{(-\pi, \pi]} = \arctan2(\text{sen}(\text{error}), \text{cos}(\text{error}))$$

y así es posible obtener el error en el dominio  $(-\pi, \pi]$ .

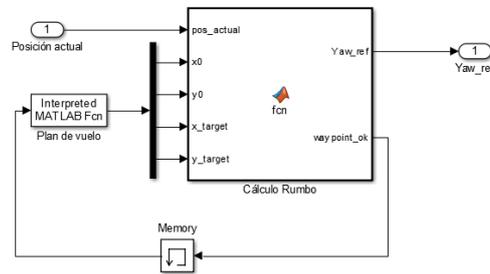


**Figura 8.4:** Error fuera de dominio

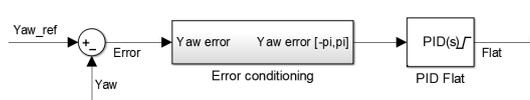
La Figura 8.5 muestra como se ha construido el cálculo del rumbo en Simulink. Aclarar que se está asumiendo que el vector velocidad y el ángulo Yaw están siempre orientados de la mismo modo. En el caso de tener deriva el vector velocidad no está alineado con Yaw y por tanto no sería correcta la implementación aquí mostrada. No obstante por su simplicidad se ha adoptado esta simplificación.

El bloque “Plan de vuelo ” contiene los waypoints a los que, uno tras otro, el dirigible se irá dirigiendo. Mientras que el bloque “ Cálculo de Rumbo ” incorpora el procedimiento matemático anteriormente desarrollado.

En la Figura 8.6 se presenta la trayectoria seguida por el dirigible aplicando el algoritmo de navegación. Se aprecia como el dirigible es capaz de alcanzar

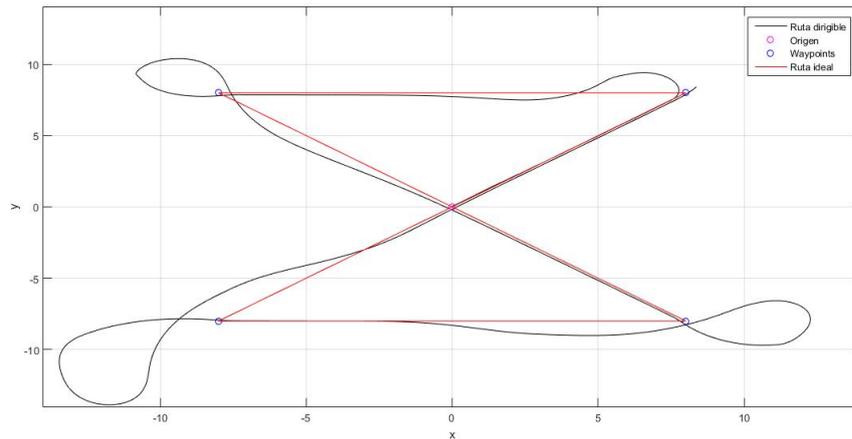


(a) Cálculo del rumbo



(b) Adecuación del error

**Figura 8.5:** Implementación del algoritmo de navegación



**Figura 8.6:** Trayectoria realizada por el dirigible

las coordenadas teóricas no obstante existen dificultades en mantener las consignas de altura y velocidad (Figura 8.7) con lo cual es necesario revisar los controladores para mejorar su actuación.

La Figura 8.8 muestra una ruta menos exigente para la aeronave y diferencia de la anterior trayectoria, en este caso las variaciones en el rumbo son menores y esto se traduce en movimientos más suaves. Consecuentemente el controlador de la altura y el de la velocidad son capaces de mantener ligeramente mejor las consignas tal y como se aprecia en la Figura 8.9.

Así pues la Figura 8.10 muestra la ruta real en tres dimensiones llevada a cabo por el dirigible. Remarcar como la sobreoscilación que introduce el control de la altura hace que la aeronave quede muy alejada de los waypoints

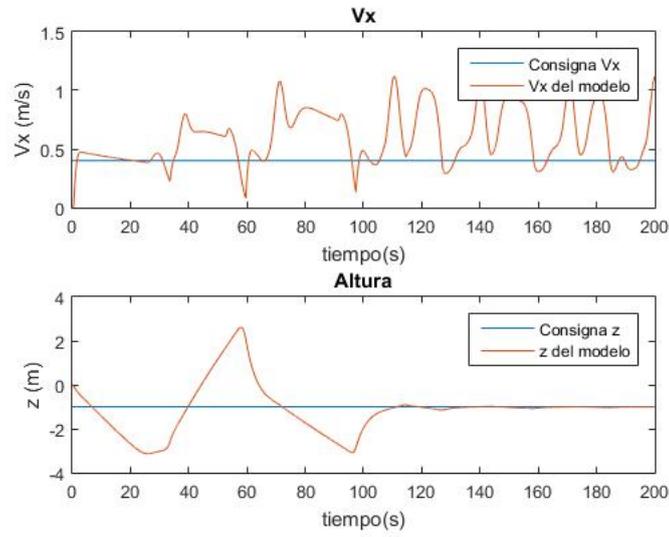


Figura 8.7:  $V_x$  y  $z$  durante la navegación

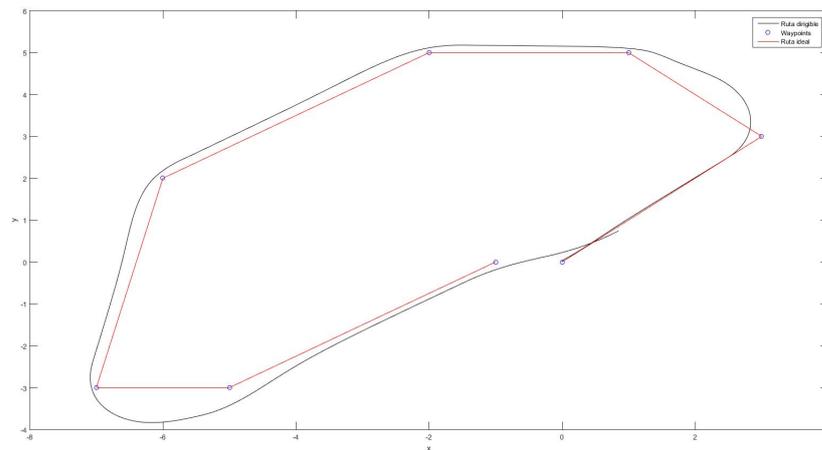


Figura 8.8: Trayectoria menos exigente para el dirigible

2 y 3. No obstante posteriormente consigue mantenerse en ruta hasta el final.

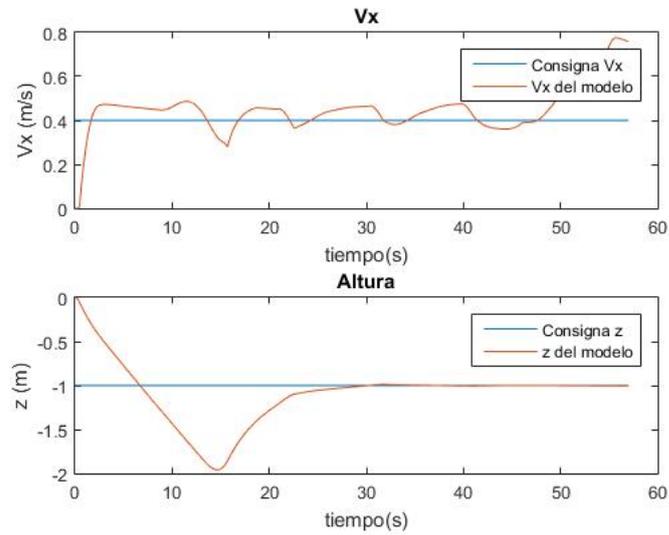


Figura 8.9:  $V_x$  y  $z$  durante una navegación más suave

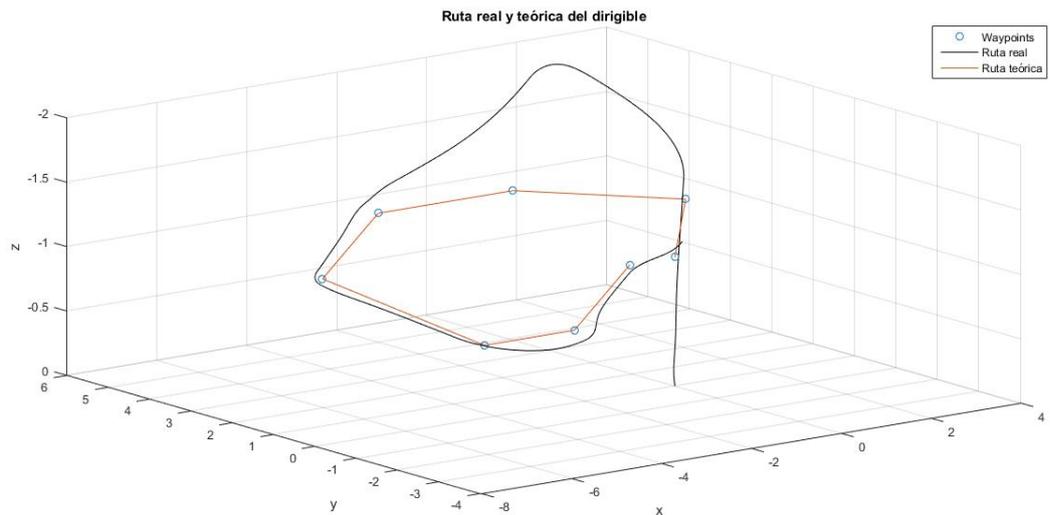


Figura 8.10: Ruta del dirigible en el espacio xyz

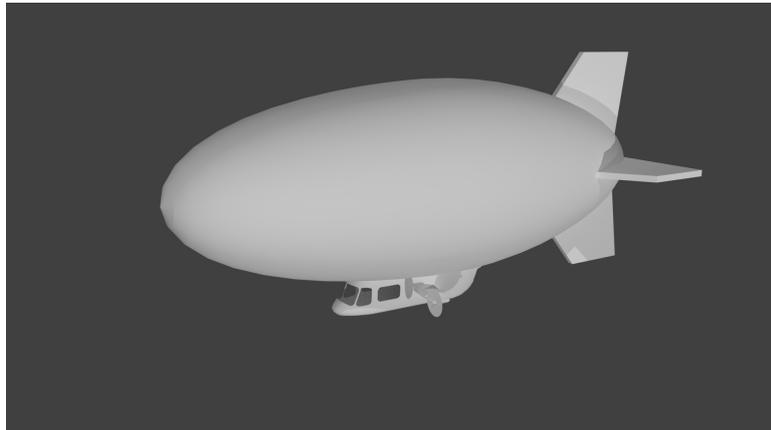


## Capítulo 9

# Entorno de Simulación 3D

El capítulo presenta el entorno de 3D de FlightGear.

Flightgear es un simulador de vuelo abierto que soporta todo tipo de sistemas operativos y ha sido desarrollado por voluntarios de todo el mundo. Este software es libre y está abierto a cualquier tipo de modificaciones y por ello ha sido empleado para simular el entorno 3D del modelo del dirigible aquí implementado.



**Figura 9.1:** Modelo 3D del dirigible

Primero se ha diseñado el modelo 3D de la Figura 9.1, mediante un software de CAD, respetando las dimensiones físicas de la aeronave y posteriormente se ha añadido a los modelos de Flightgear.

Una vez añadido modelo 3D se ha realizado la conexión de Flightgear y Simulink. La conexión entre ambos programas se ha realizado mediante un esquema ya incorporado en el paquete de Simulink (Figura 9.2). En esta

conexión se transmite la latitud, longitud y altura, además de los ángulos de Euler. Y así FlightGear es capaz de representar la posición del dirigible en cualquier lugar del mundo de su entorno 3D. De modo que Flightgear no hace uso de sus modelos dinámicos existentes en el programa, sino que la información se obtiene directamente de los cálculos del modelo no lineal implementado en el Capítulo 4.

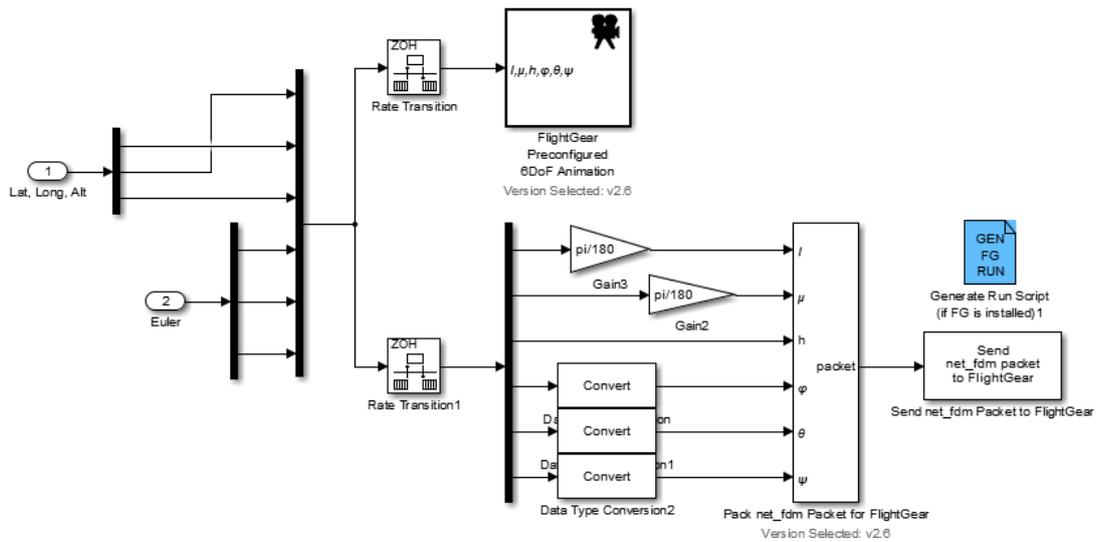


Figura 9.2: Simulación en el entorno gráfico de Flightgear



Figura 9.3: Simulación en el entorno gráfico de Flightgear

## Capítulo 10

# Conclusiones y trabajos futuros

En el Capítulo 1 se marcaba como objetivo implementar un sistema de control capaz de hacer volar de forma autónoma un dirigible de pequeñas dimensiones.

Unos de los puntos conseguidos para satisfacer el principal objetivo ha sido modelar el dirigible mediante un modelo sencillo que permita interpretar el estado de la aeronave. Sobre este modelo se ha desarrollado un sistema de control que permite desplazar el dirigible haciendo uso de los motores de propulsión, su inclinación y un motor de cola. Se han incluido además magnitudes reales de motores y ,en esa fase del proyecto, se concluía que, con los datos obtenidos, los resultados eran satisfactorios.

La implementación de la navegación entre distintos waypoints (Capítulo 8) ha sido posible gracias al empleo de un sencillo algoritmo de navegación y la asunción que el vector velocidad y el ángulo de Yaw están igualmente orientados . No obstante se ha revelado que la primera aproximación para el control de  $V_x$  y  $z$  no es suficiente para mantener consignas durante la ruta descrita por los waypoints.

Por ello directamente de los resultados obtenidos en este trabajo, en un futuro, sería necesario modificar la primera aproximación del control del dirigible aquí realizada. Las respuestas de los controladores implementados han resultado tener demasiada sobreoscilación y no ser optimas para la navegación. Además no se ha tenido en cuenta el control de  $V_y$  y por tanto no es posible compensarla de ninguna manera. Por ello con el fin de reducir su efecto se ha limitado su acción. No obstante la mejora sería necesaria.

En términos generales, el proyecto es muy abierto y es posible llevar a cabo su desarrollo hasta su implementación en una plataforma física. Además de pasar de un entorno de trabajo en interior al exterior donde el nivel de complejidad del modelo incrementa y la consideración de controladores robustos es necesaria.

## Apéndice A

# Código Matlab: Parámetros del modelo

```
% Modelo dirigible V.4 Abril 2017
% Autor: Jose Luis Carratalá Broch

clc

%% Constantes del Modelo
% dimensiones a=0.85 , b=0.35 , c=0.35 (semiejes de un elipsoide)
rb=0.35;
ra=0.85;
rc=0.35;

% Variables para el cálculo de las fuerzas aerodinámicas

rho_a=1.2; % Densidad del aire

C_dx= 0.041; % coeficientes de resistendencia aerodinamica y
            sustentación
C_dy= 0.041;
C_dz= 0.041;
C_l=0;
p=8/5; % constante
S=4*pi*((ra^(p)*rb^(p)+ra^(p)*rc^(p)+rb^(p)*rc^(p))/3)^(1/p); %
    Superficie del dirigible

Sfrontal=pi*rb^2; % Superficie frontal
Stransversal=pi*ra*rb; % Superficie del elipsoide visto desde un alzado
    (elipse)

% Fuerza de la gravedad y Flotabilidad
```

```
rho_he=0.1664;    %[kg/m3] -
V=4/3*pi*ra*rb*rc;% Volumen dirigible [m3]

g= 9.81; % [m/s2]
m=(rho_a-rho_he)*V;

% Fuerzas de propulsión y variables de control

rx= 0;%brazo del motor de cola hasta el eje x [m]
rz= 0.7; %brazo del motor de cola hasta el eje Z [m]

%% Momentos %%
K_visc=0.1; %coeficiente de viscosidad

CG=[0, 0 ,0]'; %centro de gravedad [m]
CV=[0, 0, -0.3]'; %centro de volumen desde el CdG[m]

% Momento inercia elipsoide

% Globo

Jxx_hull=1/3*m*(2*rb^2); %[kgm2]
Jyy_hull=1/3*m*(ra^2+rb^2);
Jzz_hull=1/3*m*(ra^2+rb^2);
M_inercia= [diag([Jxx_hull , Jyy_hull , Jzz_hull])];
J=Jzz_hull;
```

## Apéndice B

# Código Matlab: Linearización y FDT

```
clear all
clc
%%

% LINEARIZACION DEL MODELO V4

% Parámetros del modelo
CondicionesInicialesV4
disp('%%%%%%%%%Cargando datos para el MODELO V4 %%%%%%%%%%')

syms Yaw Roll Pitch %Yaw, Roll, Pitch respectivamente
Rx=[1 0 0;0 cos(Roll) sin(Roll);0 -sin(Roll) cos(Roll)];
Rz=[cos(Yaw) -sin(Yaw) 0;sin(Yaw) cos(Yaw) 0;0 0 1];
Ry=[cos(Pitch) 0 -sin(Pitch);0 1 0;sin(Pitch) 0 cos(Pitch)];

TMCD=Rx*Ry*Rz; %from NED frame to body frame

% Fuerzas

disp('Definiendo fuerzas y momentos')

%F de gravedad - F de flotabilidad %%%
F_gravedad=[0 0 m*g]';
F_gravedad=TMCD*F_gravedad;

F_flotabilidad =-[0 0 (rho_a-rho_he)*V*g]';
F_flotabilidad= TMCD*F_flotabilidad;

% Fuerzas aerodinámicas
syms x y z dx dy dz d2x d2y d2z dRoll dPitch dYaw d2Roll d2Pitch
d2Yaw
```

```

F_aerodinamicas = [-1/2*rho_a*dx^(2)*C_dx*Sfrontal ;
                  -1/2*rho_a*dy^(2)*C_dy*Stransversal*sign(dy);
                  -1/2*rho_a*dx^(2)*C_l*Sfrontal -1/2*rho_a*dz^(2)*
                  C_dz*Stransversal*sign(dz)];

% Fuerzas de propulsión y control

syms F_motor delta_prop F_lateral_mot
F_propulsion =[2*F_motor*cos(delta_prop) 0 -2*F_motor*sin(delta_prop)
              ]';
F_lateral=[0 F_lateral_mot 0]';

% Momentos

% Momentos debidos a la flotabilidad y fuerzas aerodinámicas
M1= F_flotabilidad(3)*CV(2)-F_flotabilidad(2)*CV(3)+F_aerodinamicas(3)
    *CV(2)-F_aerodinamicas(2)*CV(3);
M2=-F_flotabilidad(3)*CV(1)+F_flotabilidad(1)*CV(3)-F_aerodinamicas(3)
    *CV(1)+F_aerodinamicas(1)*CV(3);
M3=-F_flotabilidad(1)*CV(2)+F_flotabilidad(2)*CV(1)-F_aerodinamicas(1)
    *CV(2)+F_aerodinamicas(2)*CV(1);

M =[M1 M2 M3]';

% Momento debido a las fuerzas laterales
Momento_prop =[-F_lateral_mot*rx 0 F_lateral_mot*rz]';

% Momento debido a la fuerza viscosa
Momento_visco=[-K_visco*dRoll -K_visco*dPitch -K_visco*dYaw]; %Este
momento se opone al giro del dirigible

disp('Definición de fuerzas y momentos completada')

% Definición de las ecuaciones de movimiento

disp('Construyendo las ecuaciones')

eq1=F_aerodinamicas(1)+F_gravedad(1)+F_flotabilidad(1)+F_propulsion(1)
    +F_lateral(1)-m*d2x ;
eq2=F_aerodinamicas(2)+F_gravedad(2)+F_flotabilidad(2)+F_propulsion(2)
    +F_lateral(2)-m*d2y ;
eq3=F_aerodinamicas(3)+F_gravedad(3)+F_flotabilidad(3)+F_propulsion(3)
    +F_lateral(3)-m*d2z ;
eq4=M(1)+Momento_prop(1)+Momento_visco(1)-M_inercia(1,1)*d2Roll;
eq5=M(2)+Momento_prop(2)+Momento_visco(2)-M_inercia(2,2)*d2Pitch;
eq6=M(3)+Momento_prop(3)+Momento_visco(3)-M_inercia(3,3)*d2Yaw;

```

```

% punto de equilibrio

d2x_0=0;
d2y_0=0;
d2z_0=0;

d2Roll_0=0;
d2Pitch_0=0;
d2Yaw_0=0;

dx_0=1;
dy_0=1e-12;
dz_0=-0.2;

dYaw_0=0;
dRoll_0=0;
dPitch_0=0;

eq10 = subs(eq1,[d2x dx],[d2x_0 dx_0]);
eq20 = subs(eq2,[d2y dy],[d2y_0 dy_0]);
eq30 = subs(eq3,[d2z dz],[d2z_0 dz_0]);
eq40 = subs(eq4,[d2Roll dRoll dy],[d2Roll_0 dRoll_0 dy_0]);
eq50 = subs(eq5,[d2Pitch dPitch dx],[d2Pitch_0 dPitch_0 dx_0]);
eq60 = subs(eq6,[d2Yaw dYaw],[d2Yaw_0 dYaw_0]);

% Resolvemos

Resultado=solve(eq10,eq20,eq30,eq40,eq50) ;
F_motor_0=Resultado.F_motor;
F_lateral_0=Resultado.F_lateral_mot;
delta_prop_0=Resultado.delta_prop;

Roll_0=Resultado.Roll;
Pitch_0=Resultado.Pitch;

% Resto de variables en el punto de equilibrio
F_motor_0=double(F_motor_0);
F_lateral_0=double(F_lateral_0);
delta_prop_0=double(delta_prop_0);
Roll_0=double(Roll_0);
Pitch_0=double(Pitch_0);

disp('Inicio linearización')
% Linealiza y transformada de Laplace en el punto de equilibrio
syms s xpunto dxpunto ;
% EQ1
% cambio de variable
eq1=subs(eq1,[d2x, dx],[dxpunto, xpunto]);

Variables_eq1=[delta_prop F_motor dxpunto xpunto];
pto_func_eq1=[delta_prop_0 F_motor_0 d2x_0 dx_0];

```

```

% Derivadas parciales en el punto de operación
coef1_dxpunto = subs(diff(eq1, dxpunto), Variables_eq1, pto_func_eq1);
coef1_xpunto = subs(diff(eq1, xpunto), Variables_eq1, pto_func_eq1);
coef1_F_motor = subs(diff(eq1, F_motor), Variables_eq1, pto_func_eq1);
coef1_delta_prop = subs(diff(eq1, delta_prop), Variables_eq1,
    pto_func_eq1);

% 1ª Ecuación linealizada
eq1l = coef1_dxpunto*s*xpunto + coef1_xpunto*xpunto + coef1_F_motor*
    F_motor + coef1_delta_prop*delta_prop;

% EQ2

% cambio de variable
syms dypunto ypunto
eq2=subs(eq2, [d2y, dy],[dypunto, ypunto]);

Variables_eq2=[F_lateral_mot dypunto ypunto];
pto_func_eq2=[F_lateral_0 d2y_0 dy_0];

% Derivadas parciales en el punto de operación
coef2_F_lateral = subs(diff(eq2, F_lateral_mot), Variables_eq2,
    pto_func_eq2);
coef2_dypunto = subs(diff(eq2, dypunto), Variables_eq2, pto_func_eq2);
coef2_ypunto = subs(diff(eq2, ypunto), Variables_eq2, pto_func_eq2);

% 2ª Ecuación linealizada
eq2l = coef2_dypunto*s*ypunto+coef2_ypunto*ypunto+coef2_F_lateral*
    F_lateral_mot;

% EQ3

% cambio de variable
syms dzpunto zpunto
eq3=subs(eq3, [d2z, dz],[dzpunto, zpunto]);

Variables_eq3=[z punto dzpunto delta_prop F_motor];
pto_func_eq3=[dz_0 d2z_0 delta_prop_0 F_motor_0];

% Derivadas parciales en el punto de operación
coef3_zpunto=subs(diff(eq3, zpunto), Variables_eq3, pto_func_eq3);
coef3_dzpunto=subs(diff(eq3, dzpunto), Variables_eq3, pto_func_eq3);
coef3_delta_prop=subs(diff(eq3, delta_prop), Variables_eq3, pto_func_eq3);
;
coef3_F_motor=subs(diff(eq3, F_motor), Variables_eq3, pto_func_eq3);

% 3ª Ecuación linealizada
eq3l=coef3_dzpunto*s*zpunto+ coef3_zpunto*zpunto+coef3_delta_prop*
    delta_prop+coef3_F_motor*F_motor;

% EQ4
syms dRollpunto Rollpunto Pitchpunto
% cambio de variable
eq4=subs(eq4, [d2Roll, dRoll],[dRollpunto, Rollpunto]);

Variables_eq4=[F_lateral_mot dRollpunto Pitch Roll ypunto];
pto_func_eq4=[F_lateral_0 d2Roll_0 Pitch_0 Roll_0 dy_0];

% Derivadas parciales en el punto de operación

```

```

coef4_F_lateral=subs( diff(eq4, F_lateral_mot), Variables_eq4,
    pto_func_eq4);
coef4_dRollpunto=subs( diff(eq4, dRollpunto), Variables_eq4, pto_func_eq4)
;
coef4_Pitch=subs( diff(eq4, Pitch), Variables_eq4, pto_func_eq4);
coef4_Roll=subs( diff(eq4, Roll), Variables_eq4, pto_func_eq4);

% 4ª Ecuación linealizada
eq4l=coef4_dRollpunto*s*Rollpunto+coef4_Pitch*Pitchpunto/s+coef4_Roll*
    Rollpunto/s+coef4_F_lateral*F_lateral_mot;

% EQ5

syms dPitchpunto
%cambio de variable
eq5=subs(eq5,[dx d2Pitch, dPitch],[xpunto dPitchpunto Pitchpunto]);

Variables_eq5=[xpunto Pitch dPitchpunto];
pto_func_eq5=[dx_0 Pitch_0 d2Pitch_0];

% Derivadas parciales en el punto de operación
coef5_xpunto=subs( diff(eq5, dx), Variables_eq5, pto_func_eq5);
coef5_Pitch=subs( diff(eq5, Pitch), Variables_eq5, pto_func_eq5);
coef5_dPitchpunto=subs( diff(eq5, d2Pitch), Variables_eq5, pto_func_eq5);

% 5ª Ecuación linealizada
eq5l=coef5_dPitchpunto*s*Pitchpunto+coef5_xpunto*xpunto+coef5_Pitch*
    Pitchpunto/s;

% EQ6

syms dYawpunto Yawpunto
%cambio de variable
eq6=subs(eq6,[d2Yaw, dYaw],[dYawpunto Yawpunto]);

Variables_eq6=[F_lateral_mot dYawpunto Yawpunto];
pto_func_eq6=[F_lateral_0 d2Yaw_0 dYaw_0];

% Derivadas parciales en el punto de operación
coef6_F_lateral_mot=subs( diff(eq6, F_lateral_mot), Variables_eq6,
    pto_func_eq6);
coef6_dYawpunto=subs( diff(eq6, dYawpunto), Variables_eq6, pto_func_eq6);
coef6_Yawpunto=subs( diff(eq6, Yawpunto), Variables_eq6, pto_func_eq6);

% 6ª Ecuación linealizada
eq6l=coef6_dYawpunto*s*Yawpunto+coef6_Yawpunto*Yawpunto+
    coef6_F_lateral_mot*F_lateral_mot;

% Resolución de las salidas xpunto, ypunto, zpunto, Rollpunto, Pitchpunto
, Yawpunto
fdts=solve( eq11, eq21, eq31, eq41, eq51, eq61, xpunto, ypunto, zpunto,
    Rollpunto, Pitchpunto, Yawpunto);

disp('Ecuaciones resueltas')
% Funciones de transferencia de cada una de las salida con respecto
a cada una de las entradas

```

```

s=tf('s');

escalado_mot=0.2644; %[N] valor obtenido de un motor comercial (
    propulsión máxima)
escalado_ang= 1.57; %[rad] ángulo máximo que puede tomar los motores
    (90°).
% VX

VxFlat=collect(subs(fdts.xpunto,[F_motor, delta_prop],[0 0]),
    F_lateral_mot)/F_lateral_mot;
GVxFlat=eval(char(VxFlat));
GVxFlat=escalado_mot*GVxFlat;

VxFprop=collect(subs(fdts.xpunto,[delta_prop, F_lateral_mot],[0 0]),
    F_motor)/F_motor;
GVxFprop=eval(char(VxFprop));
GVxFprop=GVxFprop*escalado_mot;

VxDelta=collect(subs(fdts.xpunto,[F_lateral_mot, F_motor],[0 0]),
    delta_prop)/delta_prop;
GVxDelta=eval(char(VxDelta));
GVxDelta=GVxDelta*escalado_ang;

Gxyz=[];
Gxyz=[Gxyz; GVxFprop GVxDelta GVxFlat];

% VY

VyFlat=collect(subs(fdts.y punto,[F_motor, delta_prop],[0 0]),
    F_lateral_mot)/F_lateral_mot;
GVyFlat=eval(char(VyFlat));
GVyFlat=GVyFlat*escalado_mot;

VyFprop=collect(subs(fdts.y punto,[delta_prop, F_lateral_mot],[0 0]),
    F_motor)/F_motor;
GVyFprop=eval(char(VyFprop));
GVyFprop=GVyFprop*escalado_mot;

VyDelta=collect(subs(fdts.y punto,[F_lateral_mot, F_motor],[0 0]),
    delta_prop)/delta_prop;
GVyDelta=eval(char(VyDelta));
GVyDelta=GVyDelta*escalado_ang;

Gxyz=[Gxyz; GVyFprop GVyDelta GVyFlat];

% VZ

VzFlat=collect(subs(fdts.z punto,[F_motor, delta_prop],[0 0]),
    F_lateral_mot)/F_lateral_mot;
GVzFlat=eval(char(VzFlat));
GVzFlat=GVzFlat*escalado_mot;

VzFprop=collect(subs(fdts.z punto,[delta_prop, F_lateral_mot],[0 0]),
    F_motor)/F_motor;
GVzFprop=eval(char(VzFprop));
GVzFprop=GVzFprop*escalado_mot;

```

```

VzDelta=collect(subs(fdts.zpunto,[F_lateral_mot,F_motor],[0 0]),
    delta_prop)/delta_prop;
GVzDelta=eval(char(VzDelta));
GVzDelta=GVzDelta*escalado_ang;

Gxyz=[Gxyz; GVzFprop GVzDelta GVzFlat];

% Velocidad PHI – ROLL

VRollFlat=collect(subs(fdts.Rollpunto,[F_motor, delta_prop],[0 0]),
    F_lateral_mot)/F_lateral_mot;
GVRollFlat=eval(char(VRollFlat));
GVRollFlat=GVRollFlat*escalado_mot;

VRollFprop=collect(subs(fdts.Rollpunto,[delta_prop,F_lateral_mot],[0
    0]),F_motor)/F_motor;
GVRollFprop=eval(char(VRollFprop));
GVRollFprop=GVRollFprop*escalado_mot;

VRollDelta=collect(subs(fdts.Rollpunto,[F_lateral_mot,F_motor],[0 0]),
    delta_prop)/delta_prop;
GVRollDelta=eval(char(VRollDelta));
GVRollDelta=GVRollDelta*escalado_ang;

Grpy=[];
Grpy=[Grpy; GVRollFprop GVRollDelta GVRollFlat];

% Velocidad TECTA – PITCH

VPitchFlat=collect(subs(fdts.Pitchpunto,[F_motor, delta_prop],[0 0]),
    F_lateral_mot)/F_lateral_mot;
GVPitchFlat=eval(char(VPitchFlat));
GVPitchFlat=GVPitchFlat*escalado_mot;

VPitchFprop=collect(subs(fdts.Pitchpunto,[delta_prop,F_lateral_mot],[0
    0]),F_motor)/F_motor;
GVPitchFprop=eval(char(VPitchFprop));
GVPitchFprop=GVPitchFprop*escalado_mot;

VPitchDelta=collect(subs(fdts.Pitchpunto,[F_lateral_mot,F_motor],[0
    0]),delta_prop)/delta_prop;
GVPitchDelta=eval(char(VPitchDelta));
GVPitchDelta=GVPitchDelta*escalado_ang;

Grpy=[Grpy; GVPitchFprop GVPitchDelta GVPitchFlat];

% Velocidad PSI – YAW

VYawFlat=collect(subs(fdts.Yawpunto,[F_motor, delta_prop],[0 0]),
    F_lateral_mot)/F_lateral_mot;
GVYawFlat=eval(char(VYawFlat));
GVYawFlat=GVYawFlat*escalado_mot;

VYawFprop=collect(subs(fdts.Yawpunto,[delta_prop,F_lateral_mot],[0 0])
    ,F_motor)/F_motor;
GVYawFprop=eval(char(VYawFprop));
GVYawFprop=GVYawFprop*escalado_mot;

VYawDelta=collect(subs(fdts.Yawpunto,[F_lateral_mot,F_motor],[0 0]),
    delta_prop)/delta_prop;

```

```
GVYawDelta=eval(char(VYawDelta));
GVYawDelta=GVYawDelta*escalado_ang;

% Las fdt para Yaw seran:

GYawFprop=1/s*GVYawFprop;
GYawDelta=1/s*GVYawDelta;
GYawFlat=1/s*GVYawFlat;

Grpy=[Grpy; GYawFprop GYawDelta GYawFlat];

disp('Linearización completada')

% Conversion a State Space
disp('Creando espacio de estados ')

GG_escalado=ss([Gxyz; Grpy]);
disp('Creación espacio de estados finalizada')

% disp('Transformación a espacio de estados completada')
```

## Apéndice C

# Código Matlab: Ajuste de los controladores

```
% Ajuste de los PID mediante SIMC
% 11 mayo 2017 por Jose Luis Carratalá

LinearizacionV4_Vx_Vz_Yaw_Escalado;
% Principales FDT de nuestro proceso (dirigible)
%%
GVxFprop=zpk(GVxFprop);
GVzDelta=zpk(GVzDelta);
GYawFlat=zpk(GYawFlat);
%%

disp('Cálculo PI Vx/Fprop ')

%LAZO INTERNO

% Las fdt del modelo deben de estar definidas como
%          k
%          ----- * e^(-tecta*s)   ó bien , ----- * e^(-
%          tecta*s)                  tp*s+1          s+1/tp
%
% tp es la tau del proceso.

%PID en forma SERIE

% Por tanto para GVxFprop

kp_VxFprop= 0.91908/0.0147;
Tp_VxFprop=1/0.0147;
Tdeseada_VxFprop=1;

Kpid_VxFprop= 1/(kp_VxFprop)*Tp_VxFprop/Tdeseada_VxFprop;
Ti_VxFprop=min([Tp_VxFprop, 4*Tdeseada_VxFprop]);
```

```

Td_VxFprop=0;

%pasamos las constantes del PID a paralelo
Kpid_VxFprop_p= Kpid_VxFprop*(1+Td_VxFprop/Ti_VxFprop);
Ti_VxFprop_p=Kpid_VxFprop/Ti_VxFprop;
Td_VxFprop_p=Kpid_VxFprop*Td_VxFprop;

disp('Cálculo PI delta_prop/Vz ')

% Para GVzDelta

kp_VzDelta = -0.0040389/0.0204;
Tp_VzDelta = 1/0.0204;
Tdeseada_VzDelta = 1;

Kpid_VzDelta = 1/(kp_VzDelta)*Tp_VzDelta/Tdeseada_VzDelta;
Ti_VzDelta = min([Tp_VzDelta, 4*Tdeseada_VzDelta]);
Td_VzDelta = 0;

%pasamos las constantes del PID a paralelo
Kpid_VzDelta_p= Kpid_VzDelta*(1+Td_VzDelta/Ti_VzDelta);
Ti_VzDelta_p=Kpid_VzDelta/Ti_VzDelta;
Td_VzDelta_p=Kpid_VzDelta*Td_VzDelta;

disp('Cálculo PID Yaw/Flat ')
% Para GYawFlat

kp_YawFlat = 1.4576/0.7875;
Tp_YawFlat = 1/0.7875;
Tdeseada_YawFlat = 1;

Kpid_YawFlat = 1/(kp_YawFlat)*1/Tdeseada_YawFlat;
Ti_YawFlat = 4*Tdeseada_YawFlat;
Td_YawFlat = Tp_YawFlat;

%pasamos las constantes del PID a paralelo
Kpid_YawFlat_p= Kpid_YawFlat*(1+Td_YawFlat/Ti_YawFlat);
Ti_YawFlat_p=Kpid_YawFlat/Ti_YawFlat;
Td_YawFlat_p=Kpid_YawFlat*Td_YawFlat;

%% variable global para waypoints
global index
index=0;

disp(' Fin ')

```

# Bibliografía

- [1] A Bonnet y J Luneau. “Identification des coefficients aérodynamiques du dirigeable AS500 du LAAS”. En: *Rapport technique, Etude Hydro-Aérodynamique, LAAS/CNRS* (2003) (vid. pág. 6).
- [2] Eli Carneiro de Paiva y col. “Project AURORA: Infrastructure and Flight Control Experiments for a Robotic Airship”. en. En: *Journal of Field Robotics* (mar. de 2006) (vid. pág. 5).
- [3] Alberto Elfes y col. “Autonomous flight control for a titan exploration aerobot”. En: (2005) (vid. pág. 7).
- [4] Alberto Elfes y col. “Project AURORA: Development of an Autonomous Unmanned Remote Monitoring Robotic Airship”. en. En: *Journal of the Brazilian Computer Society* 4 (abr. de 1998). ISSN: 0104-6500 (vid. pág. 5).
- [5] S. B. V. Gomes. “An Investigation of the Flight Dynamics of Airships with Application to the YEZ-2A”. En: 1990 (vid. págs. 4, 5).
- [6] Emmanuel Hygounenc y col. “The autonomous blimp project of laas-cnrs: Achievements in flight control and terrain mapping”. En: *The International Journal of Robotics Research* 23.4-5 (2004), págs. 473-511 (vid. págs. 6, 7).
- [7] Alexandra Bento Moutinho. “Modeling and nonlinear control for airship autonomous flight”. Tesis doct. Universidade Técnica de Lisboa, 2007 (vid. pág. 6).

- [8] José Raul Azinheira y col. “Lateral/directional control for an autonomous, unmanned airship”. En: *Aircraft Engineering and Aerospace Technology* 73.5 (2001), págs. 453-459 (vid. pág. 6).
- [9] Shivanjli Sharma, Eric A Kulczycki y Alberto Elfes. “Trajectory generation and path planning for autonomous aerobots”. En: (2007) (vid. pág. 7).
- [10] Sigurd Skogestad y Chriss Grimholt. “The SIMC method for smooth PID controller tuning”. En: *PID Control in the Third Millennium*. Springer, 2012, págs. 147-175 (vid. págs. 44, 45).
- [11] FL Thompson y HW Kirschbaum. “The drag characteristics of several airships determined by deceleration tests”. En: (1932) (vid. pág. 24).