



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

Departamento de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería del Diseño

Universitat Politècnica de València

Diseño, modelado y control del sistema de propulsión para un vehículo aéreo no tripulado basado en globo aerostático dirigible

Presupuesto

Trabajo Fin de Grado

Grado en Ingeniería Aeroespacial

Autor: Rodrigo Penadés Mañes

Tutor: Sergio García-Nieto Rodríguez

2016/2017

1.	Introducción	1
2.	Precio de la mano de obra	2
3.	Precio de los materiales	3
3.1	Materiales para el desarrollo.....	3
3.2	Materiales del proyecto	3
4.	Precio unitario	4
5.	Precios descompuestos.....	5
5.1	Horas de mano de obra por unidad de obra	5
5.2	Precios descompuestos	6
6.	Presupuesto parcial	8
7.	Presupuesto total	9

1. Introducción

En el presente documento se procede a detallar los costes de ejecución del proyecto “Diseño, modelado y control del sistema de propulsión para un vehículo aéreo no tripulado basado en globo aerostático dirigible”. Se detallarán los costes de mano de obra y los materiales empleados en las unidades de obra y sus tareas que componen el proyecto.

El proyecto se organiza en cuatro unidades de obra:

- Estudio y aplicación del modelo dinámico del sistema.
- Diseño y modelado del sistema de propulsión.
- Desarrollo del *software* en *Mathematica*.
- Diseño, desarrollo e implementación del control.

2. Precio de la mano de obra

El coste de mano de obra se ha estimado a partir de una cantidad por hora para un recién graduado en ingeniería aeroespacial. Se ha supuesto un contrato indefinido a tiempo completo con catorce pagas mensuales tomando 22 días laborales efectivos al mes. Se aplica al bruto un 40% de cotización a la Seguridad Social, que incluye: 23,6% contingencias comunes de la empresa, 4,7% contingencias comunes del trabajador y el resto que incluye otras retenciones como el IRPF. El precio final por hora se calcula sobre 11 meses.

En el cuadro se resume el cálculo del precio por hora del profesional contratado:

Graduado en Aeroespacial	
Salario (€/h)	12
Salario bruto mensual (€/mes)	2112
Salario anual (€/año)	29568
Cotización S.S. 40% (€/mes)	844,8
Seguridad Social (€/año)	11827,2
Coste anual	41395,2
Precio (€/h)	21,38

3. Precio de los materiales

Se detallan ahora los precios de los materiales empleados en la ejecución del proyecto.

3.1 Materiales para el desarrollo

Materiales	Precio (€/ud)
Ordenador portátil Lenovo G50	299,95
Licencia Estudiante <i>Mathematica</i>	0
Licencia Estudiante <i>Matlab</i>	0
Licencia Estudiante <i>Microsoft Office</i>	0
Licencia Estudiante <i>Solidworks</i>	0
Software libre <i>FlightGear</i>	0

3.2 Materiales del proyecto

Materiales	Precio (€/ud)
Globo RC Publizepelines modelo 1,7m 500L	480
ESC YGE 8S	34
Motor RotorX RX1105	14,3
Pack 4 Hélices RX3020	3,3
Batería Li-Po Turnigy nano-tech 850mah 2S	5,41
Servomotor	7
Controladora OpenPilot CC3D	40,49
Estación tierra OPLink	17,02
Modulo GPS Ublox Neo-M8N	30,15
Anclaje del motor	5

4. Precio unitario

Precios unitarios de cada unidad de obra

Unidad de obra	Coste (€)
Estudio y aplicación del modelo dinámico del sistema	1623,31
Diseño y modelado del sistema de propulsión	1318,43
Desarrollo del <i>software</i> en <i>Mathematica</i>	956,19
Diseño, desarrollo e implementación del control	2179,23

5. Precios descompuestos

Primero se presenta un desglose por tareas de las horas empleadas en cada unidad de obra. Después, se presenta la descomposición de los precios de cada unidad de obra.

5.1 Horas de mano de obra por unidad de obra

UNIDAD DE OBRA 1: Estudio y aplicación del modelo dinámico del sistema

Tareas	Horas
Estudio del diseño de globos dirigibles	20
Revisión bibliográfica de proyectos similares	8
Estudio pormenorizado del modelo dinámico	25
Estudio, calculo y estimación de los efectos aerodinámicos	20
TOTAL	73

UNIDAD DE OBRA 2: Diseño y modelado del sistema de propulsión

Tareas	Horas
Estudio de propuestas	8
Diseño en detalle de la propuesta	40
Diseño en en detalle del CAD de la propuesta	5
Modelado de la propulsión	0,5
Obtención de datos y conclusión del modelo dinámico	0,5
TOTAL	54

UNIDAD DE OBRA 3: Desarrollo del *software* en *Mathematica*

Tareas	Horas
Implementación del modelo en <i>Mathematica</i>	2
Cálculo del punto de funcionamiento y otros	16
Linealización del sistema. Matrices completas y linealización numérica	5
Diseño de simulaciones y estudio del modelo	20
Programación del documento ToMatlab.m	5
TOTAL	48

UNIDAD DE OBRA 4: Diseño, desarrollo e implementación del control

Tareas	Horas
Implementación del modelo, completo y lineal, en <i>Matlab</i> y <i>Simulink</i>	1
Estudio previo dinámico del modelo lineal	2
Estudio, diseño e implementación del control. LQR.	24
Estudio, diseño e implementación del módulo de navegación	24
Conexión al simulador de vuelo	1
Diseño y simulación de pruebas de vuelo	8
Ajuste y verificación del control	30
Validación del diseño	8
TOTAL	98

5.2 Precios descompuestos

UNIDAD DE OBRA 1: Estudio y aplicación del modelo dinámico del sistema			
Concepto	Rendimiento (h o ud.)	Precio (€)	Importe (€)
Mano de obra ingeniero	73	21,38	1560,87
Total antes de costes			1560,87
Costes directos (1%)			15,61
Costes indirectos (3%)			46,83
TOTAL			1623,31

UNIDAD DE OBRA 2: Diseño y modelado del sistema de propulsión			
Concepto	Rendimiento (h o ud.)	Precio (€)	Importe (€)
Mano de obra ingeniero	54	21,38	1154,62
Licencia estudiante <i>Mathematica</i>	1	0	0,00
Licencia <i>Microsoft Office</i>	1	49,9	49,90
Licencia estudiante <i>Solidworks</i>	1	0	0,00
Globo RC <i>Publizeppelines</i> modelo 1,7m 500L	1	480	480,00
ESC YGE 8S	2	34	68,00
Motor RotorX RX1105	2	14,3	28,60
Pack Hélices RX3020	1	3,3	3,30
Batería Li-Po <i>Turnigy nano-tech</i> 850mah 2S	1	5,41	5,41
Servomotor	1	7	7,00
Controladora OpenPilot CC3D	1	40,49	40,49
Estación tierra OPLink	1	17,02	17,02
Modulo GPS Ublox Neo-M8N	1	30,15	30,15
Anclaje del motor	2	5	10,00

Total antes de costes	1894,49
Costes directos (1%)	18,94
Costes indirectos (3%)	56,83
TOTAL	1970,27

UNIDAD DE OBRA 3: Desarrollo del <i>software</i> en <i>Mathematica</i>			
Concepto	Rendimiento (h o ud.)	Precio (€)	Importe (€)
Mano de obra ingeniero	48	21,38	1026,33
Licencia estudiante <i>Mathematica</i>	1	0	0,00
Total antes de costes			1026,33
Costes directos (1%)			10,26
Costes indirectos (3%)			30,79
TOTAL			1067,38

UNIDAD DE OBRA 4: Diseño, desarrollo e implementación del control			
Concepto	Rendimiento (h o ud.)	Precio (€)	Importe (€)
Mano de obra ingeniero	98	21,38	2095,42
Licencia estudiante <i>Mathematica</i>	1	0	0,00
Licencia estudiante <i>Matlab</i>	1	0	0,00
Software <i>FlightGear</i>	1	0	0,00
Total antes de costes			2095,42
Costes directos (1%)			20,95
Costes indirectos (3%)			62,86
TOTAL			2179,23

6. Presupuesto parcial

Precios de las unidades de obra junto a la cantidad requerida

Unidad de obra	Cantidad	Coste (€)
Estudio y aplicación del modelo dinámico del sistema	1	1623,31
Diseño y modelado del sistema de propulsión	1	1970,27
Desarrollo del <i>software</i> en <i>Mathematica</i>	1	1067,38
Diseño, desarrollo e implementación del control	1	2179,23

7. Presupuesto total

A continuación se expone el coste total de ejecución material, de inversión y base de licitación.

Concepto	Coste (€)
UNIDAD DE OBRA 1: Estudio y aplicación del modelo dinámico del sistema	1623,31
UNIDAD DE OBRA 2: Diseño y modelado del sistema de propulsión	1970,27
UNIDAD DE OBRA 3: Desarrollo del <i>software</i> en <i>Mathematica</i>	1067,38
UNIDAD DE OBRA 4: Diseño, desarrollo e implementación del control	2179,23
Ordenador Portátil	299,95
Total Ejecución Material	7140,14
6% Beneficio Industrial	428,41
Total Presupuesto Inversión	7568,55
21% IVA	1589,40
Total presupuesto base de licitación	9157,94

- El Presupuesto de Ejecución Material asciende a la cantidad expresada en EUROS:
SIETE MIL CIENTO CUARENTA CON CATORCE
- El Presupuesto de Inversión asciende a la cantidad expresada en EUROS:
SIETE MIL QUINIENTOS SESENTA Y OCHO CON CINCUENTA Y CINCO
- El Presupuesto base de Licitación asciende a la cantidad expresada en EUROS:
NUEVE MIL CIENTO CINCUENTA Y SIETE CON NOVENTA Y CUATRO



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

Departamento de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería del Diseño

Universitat Politècnica de València

Diseño, modelado y control del sistema de propulsión para un vehículo aéreo no tripulado basado en globo aerostático dirigible

Pliego de condiciones

Trabajo Fin de Grado

Grado en Ingeniería Aeroespacial

Autor: Rodrigo Penadés Mañes

Tutor: Sergio García-Nieto Rodríguez

2016/2017

1.	Condiciones generales de seguridad	1
1.1	General	1
1.2	Real Decreto 488/1997 de 14 de abril	2
2.	Condiciones Particulares	7
2.1	Especificaciones de materiales y equipos	7
2.1.1	Hardware	7
2.1.2	Software	7
2.1.3	Materiales para el diseño	8
2.2	Características del personal	8
2.2.1	Conocimientos previos	8
2.3	Unidades de obra	8
2.3.1	Estudio y aplicación del modelo dinámico del sistema	8
2.3.1.1	Tareas a realizar	9
2.3.2	Diseño y modelado del sistema de propulsión	9
2.3.2.1	Tareas a realizar	9
2.3.3	Desarrollo del <i>software</i> en <i>Mathematica</i>	9
2.3.3.1	Tareas a realizar	9
2.3.4	Diseño, desarrollo e implementación del control	10
2.3.4.1	Tareas a realizar	10

1. Condiciones Generales de Seguridad

1.1 General

En cualquier puesto de trabajo en el que, habitualmente y durante una parte relevante del trabajo se haga uso de un equipo con pantalla de visualización, ya bien se trate de un ordenador o vigilancia de pantallas, está sujeto a unos riesgos bien definidos que se deben prevenir. Este tipo de trabajo constituye, probablemente, el ejemplo más característico de cómo una nueva tecnología puede suponer la introducción de nuevos riesgos tales como problemas en ojos y visión, posturales y lesiones por movimientos repetidos y estrés.

Para mitigar dichos efectos, el Real Decreto 488/1997 de 14 de Abril establece las disposiciones mínimas de seguridad y de salud para la utilización por los trabajadores de equipos que incluyan pantallas de visualización donde se aplican las disposiciones de la Ley 31/1995, de 8 de Noviembre, de Prevención de Riesgos laborales.

Las variables principales a las que se debe prestar especial atención para la prevención del tipo de riesgos a los que el trabajador puede enfrentarse en la realización del Trabajo de Fin de Grado son:

- Tiempo de permanencia requerida ante la pantalla, el cual puede ser de manera continua o discontinua.
- Tiempo de trabajo con la pantalla de visualización.
- Exigencia y grado de complejidad de la tarea que el operario tiene que realizar ante la pantalla de visualización.
- Necesidad de obtener información de manera rápida y precisa.

Los factores generales que se deben tener en cuenta para la prevención de riesgos laborales son:

- Seguridad (debido a contactos eléctricos, caída o golpes en el puesto de trabajo).
- Higiene industrial (iluminación, ruido y condiciones termo-higrométricas, transmisiones de virus en el ambiente de trabajo o bacterias nocivas a través del contacto en el teclado que podría acumular gran cantidad de estas).
- Ergonomía (fatiga visual, picores, percepción borrosa. Fatiga física y mental: ansiedad, irritabilidad o insomnio).

1.2 Real Decreto 488/1997 de 14 de abril

Artículo 1. Objeto.

1. El presente Real Decreto establece las disposiciones mínimas de seguridad y de salud para la utilización por los trabajadores de equipos que incluyan pantallas de visualización.
2. Las disposiciones de la Ley 31/1995, de 8 de noviembre, de Prevención de Riesgos Laborales, se aplicarán plenamente al conjunto del ámbito contemplado en el apartado anterior.
3. Quedan excluidos del ámbito de aplicación de este Real Decreto:
 - a. Los puestos de conducción de vehículos o máquinas.
 - b. Los sistemas informáticos embarcados en un medio de transporte.
 - c. Los sistemas informáticos destinados prioritariamente a ser utilizados por el público.
 - d. Los sistemas llamados portátiles, siempre y cuando no se utilicen de modo continuado en un puesto de trabajo.
 - e. Las calculadoras, cajas registradoras y todos aquellos equipos que tengan un pequeño dispositivo de visualización de datos o medidas necesario para la utilización directa de dichos equipos.
 - f. Las máquinas de escribir de diseño clásico, conocidas como máquinas de ventanilla.

Artículo 2. Definiciones.

A efectos de este Real Decreto se entenderá por:

1. **Pantalla de visualización:** una pantalla alfanumérica o gráfica, independientemente del método de representación visual utilizado.
2. **Puesto de trabajo:** el constituido por un equipo con pantalla de visualización provisto, en su caso, de un teclado o dispositivo de adquisición de datos, de un programa para la interconexión persona/máquina, de accesorios ofimáticos y de un asiento y mesa o superficie de trabajo, así como el entorno laboral inmediato.
3. **Trabajador:** cualquier trabajador que habitualmente y durante una parte relevante de su trabajo normal utilice un equipo con pantalla de visualización.

Artículo 3. Obligaciones generales del empresario.

1. El empresario adoptará las medidas necesarias para que la utilización por los trabajadores de equipos con pantallas de visualización no suponga riesgos para su seguridad o salud o, si ello no fuera posible, para que tales riesgos se reduzcan al mínimo.

En cualquier caso, los puestos de trabajo a que se refiere el presente Real Decreto deberán cumplir las disposiciones mínimas establecidas en el anexo del mismo.

2. A efectos de lo dispuesto en el primer párrafo del apartado anterior, el empresario deberá evaluar los riesgos para la seguridad y salud de los trabajadores, teniendo en cuenta en particular los posibles riesgos para la vista y los problemas físicos y de carga mental, así como el posible efecto añadido o combinado de los mismos.

La evaluación se realizará tomando en consideración las características propias del puesto de trabajo y las exigencias de la tarea y entre éstas, especialmente, las siguientes:

- a. El tiempo promedio de utilización diaria del equipo.
 - b. El tiempo máximo de atención continua a la pantalla requerido por la tarea habitual.
 - c. El grado de atención que exija dicha tarea.
3. Si la evaluación pone de manifiesto que la utilización por los trabajadores de quipos con pantallas de visualización supone o puede suponer un riesgo para su seguridad o salud, el empresario adoptará las medidas técnicas u organizativas necesarias para eliminar o reducir el riesgo al mínimo posible. En particular, deberá reducir la duración máxima del trabajo continuado en pantalla, organizando la actividad diaria de forma que esta tarea se alterne con otras o estableciendo las pausas necesarias cuando la alternancia de tareas no sea posible o no baste para disminuir el riesgo suficientemente.
 4. En los convenios colectivos podrá acordarse la periodicidad, duración y condiciones de organización de los cambios de actividad y pausas a que se refiere el apartado anterior.

Artículo 4. Vigilancia de la salud.

1. El empresario garantizará el derecho de los trabajadores a una vigilancia adecuada de su salud, teniendo en cuenta en particular los riesgos para la vista y los problemas físicos y de carga mental, el posible efecto añadido o combinado de los mismos, y la eventual patología acompañante. Tal vigilancia será realizada por personal sanitario competente y según determinen las autoridades sanitarias en las pautas y protocolos que se elaboren, de conformidad con lo dispuesto en el apartado 3 del artículo 37 del Real Decreto 39/1997, de 17 de enero, por el que se aprueba el Reglamento de los servicios de prevención. Dicha vigilancia deberá ofrecerse a los trabajadores en las siguientes ocasiones:
 - a. Antes de comenzar a trabajar con una pantalla de visualización.
 - b. Posteriormente, con una periodicidad ajustada al nivel de riesgo a juicio del médico responsable.
 - c. Cuando aparezcan trastornos que pudieran deberse a este tipo de trabajo.
2. Cuando los resultados de la vigilancia de la salud a que se refiere el apartado 1 lo hiciese necesario, los trabajadores tendrán derecho a un reconocimiento oftalmológico.
3. El empresario proporcionará gratuitamente a los trabajadores dispositivos correctores especiales para la protección de la vista adecuados al trabajo con el equipo de que se trate, si los resultados de la vigilancia de la salud a que se refieren los apartados anteriores demuestran su necesidad y no pueden utilizarse dispositivos correctores normales.

Artículo 5. Obligaciones en materia de información y formación.

1. De conformidad con los artículos 18 y 19 de la Ley de Prevención de Riesgos Laborales, el empresario deberá garantizar que los trabajadores y los representantes de los trabajadores reciban una formación e información adecuadas sobre los riesgos derivados de la utilización de los equipos que incluyan pantallas de visualización, así como sobre las medidas de prevención y protección que hayan de adoptarse en aplicación del presente Real Decreto,
2. El empresario deberá informar a los trabajadores sobre todos los aspectos relacionados con la seguridad y la salud en su puesto de trabajo y sobre las medidas llevadas a cabo de conformidad con lo dispuesto en los artículos 3 y 4 de este Real Decreto.
3. El empresario deberá garantizar que cada trabajador reciba una formación adecuada sobre las modalidades de uso de los equipos con pantallas de visualización, antes de comenzar este tipo de trabajo y cada vez que la organización del puesto de trabajo se modifique de manera apreciable.

Artículo 6. Consulta y participación de los trabajadores.

La consulta y participación de los trabajadores o sus representantes sobre las cuestiones a que se refiere este Real Decreto se realizarán de conformidad con lo dispuesto en el apartado 2 del artículo 18 de la Ley de Prevención de Riesgos Laborales.

Anexo

Disposiciones mínimas

Por todo lo dispuesto anteriormente, se va a definir el puesto de trabajo específico para la prevención de riesgos laborales y seguridad.

1. Equipo.
 - a. Observación general.

La utilización en sí misma del equipo no debe ser una fuente de riesgo para los trabajadores.
 - b. Pantalla.

Los caracteres de la pantalla deberán estar bien definidos y configurados de forma clara, y tener una dimensión suficiente, disponiendo de un espacio adecuado entre los caracteres y los renglones. La imagen de la pantalla deberá ser estable, sin fenómenos de destellos, centelleos u otras formas de inestabilidad. El usuario de terminales con pantalla deberá poder ajustar fácilmente la luminosidad y el contraste entre los caracteres y el fondo de la pantalla, y adaptarlos fácilmente a las condiciones del entorno. La pantalla deberá ser orientable e inclinable a voluntad, con facilidad para adaptarse a las necesidades del usuario. Podrá utilizarse un pedestal independiente o

una mesa regulable para la pantalla. La pantalla no deberá tener reflejos ni reverberaciones que puedan molestar al usuario.

c. Teclado.

El teclado deberá ser inclinable e independiente de la pantalla para permitir que el trabajador adopte una postura cómoda que no provoque cansancio en los brazos o las manos. Tendrá que haber espacio suficiente delante del teclado para que el usuario pueda apoyar los brazos y las manos. La superficie del teclado deberá ser mate para evitar los reflejos. La disposición del teclado y las características de las teclas deberán tender a facilitar su utilización. Los símbolos de las teclas deberán resaltar suficientemente y ser legibles desde la posición normal de trabajo.

d. Mesa o superficie de trabajo.

La mesa o superficie de trabajo deberán ser poco reflectantes, tener dimensiones suficientes y permitir una colocación flexible de la pantalla, del teclado, de los documentos y del material accesorio. El soporte de los documentos deberá ser estable y regulable y estará colocado de tal modo que se reduzcan al mínimo los movimientos incómodos de la cabeza y ojos. El espacio deberá ser suficiente para permitir a los trabajadores una posición cómoda.

e. Asiento de trabajo.

El asiento de trabajo deberá ser estable, proporcionando al usuario libertad de movimiento y procurándole una postura confortable. La altura del mismo deberá ser regulable, el respaldo deberá ser reclinable y su altura ajustable. Se pondrá un reposapiés a disposición de quienes lo deseen.

2. Entorno

a. Espacio.

El puesto de trabajo deberá tener una dimensión suficiente y estar acondicionado de tal manera que haya espacio para permitir los cambios de postura y movimientos de trabajo.

b. Iluminación.

La iluminación general y la iluminación especial (lámparas de trabajo), cuando sea necesaria, deberán garantizar unos niveles adecuados de iluminación y unas relaciones adecuadas de luminancias entre la pantalla y su entorno, habida cuenta del carácter del trabajo, de las necesidades visuales del usuario y del tipo de pantalla utilizado. El acondicionamiento del lugar de trabajo y del puesto de trabajo, así como la situación y las características técnicas de las fuentes de luz artificial, deberán coordinarse de tal manera que se eviten los deslumbramientos y los reflejos molestos en la pantalla u otras partes del equipo.

c. Reflejos y deslumbramientos.

Los puestos de trabajo deberían instalarse de tal forma que las fuentes de luz, tales como ventanas y otras aberturas, los tabiques transparentes o translúcidos y los equipos o tabiques de color claro no provoquen deslumbramiento directo ni produzcan reflejos molestos en la pantalla. Las ventanas deberán ir equipadas con un dispositivo de cobertura adecuado y regulable para atenuar la luz del día que ilumine el puesto de trabajo.

d. Ruido.

El ruido producido por los equipos instalados en el puesto de trabajo deberá tenerse en cuenta al diseñar el mismo, en especial para que no se perturbe la atención ni la palabra.

e. Calor.

Los equipos instalados en el puesto de trabajo no deberán producir un calor adicional que pueda ocasionar molestias a los trabajadores.

f. Emisiones.

Toda radiación, excepción hecha de la parte visible del espectro electromagnético, deberá reducirse a niveles insignificantes desde el punto de vista de la protección de la seguridad y de la salud de los trabajadores.

g. Humedad.

Deberá crearse y mantenerse una humedad aceptable.

3. Interconexión ordenador/persona.

Para la elaboración, la elección, la compra y la modificación de programas, así como para la definición de las tareas que requieran pantallas de visualización, el empresario tendrá en cuenta los siguientes factores:

a. El programa habrá de estar adaptado a la tarea que deba realizarse.

b. El programa habrá de ser fácil de utilizar y deberá, en su caso, poder adaptarse al nivel de conocimientos y de experiencia del usuario; no deberá utilizarse ningún dispositivo cuantitativo o cualitativo de control sin que los trabajadores hayan sido informados y previa consulta con sus representantes.

c. Los sistemas deberán proporcionar a los trabajadores indicaciones sobre su desarrollo.

d. Los sistemas deberán mostrar la información en un formato y ritmo adaptados a los operadores.

e. Los principios de ergonomía deberán aplicarse en particular al tratamiento de la información por parte de la persona.

2. Condiciones Particulares

En este punto se describen las condiciones técnicas a las que debe someterse el proyecto. Es decir, las características que ha de tener el personal contratado y el material adquirido para la realización correcta del proyecto.

2.1 Especificaciones de materiales y equipos

2.1.1 Hardware

Para gran parte del proyecto es imprescindible un equipo que sea capaz de soportar el software empleado. Dicho equipo deberá ser capaz de soportar cómodamente los procesos, teniendo en cuenta que se deben realizar numerosas simulaciones y cálculos en simbólico que ocupan una cantidad de tiempo moderada.

Los dos software principales, *Mathematica* y *Matlab Simulink*, son los que presentan los requisitos de Hardware superiores. Se recomienda, para el proyecto, emplear un procesador Intel Core i7, 40 Gb de disco duro para las instalaciones y archivos generados y 1Gb de memoria RAM disponibles. Para una mayor velocidad de cálculo se puede ampliar la RAM.

Se ha empleado un ordenador portátil Lenovo G50 que cumplía estas especificaciones.

2.1.2 Software

Los programas empleados para la elaboración del proyecto:

- *Wolfram Mathematica 10.2*
- *Matlab y Simulink*
- *Catia V5*
- Simulador de vuelo libre *FlightGear 2.10*
- *Microsoft Word*
- *Microsoft Excel*
- *Microsoft PowerPoint*
- *PDF Reader*
- Navegador web

De los programas listados es necesario contar con el módulo *Simulink Aeronautics*, presente en la licencia para estudiantes de la UPV.

También ha sido necesario el paquete *ToMatlab.m* para la programación en *Mathematica* de un documento que facilitaba la implementación del modelo dinámico del dirigible en *Simulink*.

La conexión internet ha sido necesaria y el empleo de las credenciales de estudiante la universidad vital para el acceso gratuito a la mayoría de los documentos de investigación citados en la bibliografía,

su consulta y su descarga. Sin el acceso a estos documentos los avances en el proyecto se hubieran visto muy comprometidos.

2.1.3 Materiales para el diseño

Los materiales que se empleen en la ejecución del diseño deben ser los indicados en la memoria y presupuesto del presente proyecto. Cualquier cambio en la elección de las piezas o del diseño debe estar debidamente justificado.

Los precios indicados en el presupuesto se corresponden con los precios de los proveedores y las facturas emitidas en el proceso de adquisición de las mismas a la fecha actual del proyecto.

2.2 Características del personal

2.2.1 Conocimientos previos

Para el desarrollo de este Trabajo Fin de Grado se ha requerido de conocimientos previos en aeronáutica, mecánica y tecnología de control automático. También es muy importante tener una experiencia alta en los softwares listados en el apartado anterior. Por tanto, el personal contratado debe tener experiencia probada en el uso de dichos programas, principalmente *Mathematica* y *Matlab*, y tener una completa formación como ingeniero.

La gran mayoría de los documentos técnicos consultados están en inglés. Por tanto, el conocimiento del idioma y del vocabulario técnico específico en dicho idioma de las áreas nombradas en el apartado anterior es indispensable.

2.3 Unidades de obra

Las obras y sus especificaciones se describirán a continuación en detalle.

El orden aquí expuesto de las obras es el que más se acerca a la línea temporal de trabajo seguida y la recomendada a seguir. No siendo incompatible la realización simultánea de las tres primeras por estar interrelacionadas.

Las unidades de obra que componen el proyecto son las siguientes:

2.3.1 Estudio y aplicación del modelo dinámico del sistema

Esta unidad de obra comprende un importante trabajo de estudio e investigación previos al diseño del sistema propulsivo. Se llevó a cabo un estudio de nociones de diseño generales sobre este tipo de plataformas. Se consultaron las memorias proyectos académicos, o de colaboración, actuales muy similares. Se estudiaron distintos modelos dinámicos desarrollados, que diferían principalmente en la aerodinámica y puntos menores, se compararon las diferencias y, fruto de este trabajo, se desarrolló un modelo aplicado al globo disponible.

Es vital la estimación de los distintos efectos aerodinámicos para el correcto dimensionamiento de la propulsión. En la actualidad dichos efectos se encuentran bajo discusión siendo necesarias diseñar y realizar pruebas de vuelo con el sistema para confirmar o corregir los valores estimados.

2.3.1.1 Tareas a realizar

A continuación se especifican las tareas a realizar en esta unidad de obra:

- Estudio del diseño de globos dirigibles.
- Revisión bibliográfica de proyectos.
- Estudio pormenorizado del modelo dinámico.
- Estudio, cálculo y estimación de los efectos aerodinámicos.

2.3.2 Diseño y modelado del sistema de propulsión

Esta unidad de obra comprende el trabajo realizado de diseño del sistema propulsivo y la electrónica del sistema para dotar al sistema de la capacidad de vuelo autónomo.

Está constituido por el desarrollo conceptual de prototipos en CAD, su discusión, y el diseño en detalle de la propuesta escogida.

Las piezas y materiales empleados no podrán ser modificados sin la justificación pertinente.

2.3.2.1 Tareas a realizar

A continuación se especifican las tareas a realizar en esta unidad de obra:

- Estudio de propuestas.
- Diseño en detalle de la propuesta.
- Modelado en detalle de un CAD de la propuesta.
- Modelado matemático de la propulsión.
- Obtención de datos geométricos y conclusión del modelo dinámico.

2.3.3 Desarrollo del *software* en *Mathematica*

Esta unidad de obra recoge el trabajo realizado con el *software* matemático *Mathematica*. Está constituida por todos los documentos creados en la realización del proyecto.

Esta unidad de obra es transversal al proyecto. Todas las tareas que la componen están relacionadas con el desarrollo de las otras unidades de obra en alguna tarea.

2.3.3.1 Tareas a realizar

A continuación se especifican las tareas a realizar en esta unidad de obra:

- Implementación del modelo dinámico completo.
- Cálculos de puntos de funcionamiento habituales.

- Linealización del sistema. Obtención de las matrices del sistema en espacio de estados.
- Diseño de simulaciones y estudio completo del modelo con los datos estimados.
- Programación de un documento para la implementación en *Simulink*.

2.3.4 Diseño, desarrollo e implementación del control

Esta unidad de obra recoge el trabajo sobre el control. Está constituido por la implementación del modelo en Simulink y de todas las funciones programadas en el proyecto para dotarle de capacidad de vuelo autónomo.

2.3.4.1 Tareas a realizar

A continuación se especifican las tareas a realizar en esta unidad de obra:

- Implementación del modelo dinámico, completo y lineal, en *Simulink*.
- Estudio previo dinámico del modelo lineal
- Estudio, diseño e implementación del servocontrol. LQR.
- Estudio, diseño e implementación del módulo de navegación
- Conexión al simulador de vuelo
- Diseño y simulación de pruebas de vuelo
- Ajuste fino y verificación del control
- Validación del diseño



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

Departamento de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería del Diseño

Universitat Politècnica de València

Diseño, modelado y control del sistema de propulsión para un vehículo aéreo no tripulado basado en globo aerostático dirigible

Manual de programación

Trabajo Fin de Grado

Grado en Ingeniería Aeroespacial

Autor: Rodrigo Penadés Mañes

Tutor: Sergio García-Nieto Rodríguez

2016/2017

1. Introducción	1
2. . Mathematica	2
2.1 Matriz de transformación ejes horizonte local a ejes cuerpo	2
2.2 Modelo matemático	2
2.2.1 Matrices de masa y matriz de masa añadida	2
2.2.2 Contribuciones del modelo	3
2.3 Relación Ángulos de Euler y ecuaciones cinemáticas.....	5
2.4 Ecuaciones diferenciales reordenadas	5
2.5 Expresiones para los bloques Simulink	5
2.6 Constantes del proyecto.....	7
2.7 Punto de funcionamiento. Vuelo horizontal.	8
2.8 Punto de funcionamiento. Vuelo espiral.	9
2.9 Linealización del modelo. Matrices analíticas	10
A.9 Linealización del modelo. Resultado numérico	12
3. Matlab	15
3.1 Archivo principal.....	15
3.2 Etapa de navegación.....	23

1. Introducción

En el presente documento se procede a mostrar el código programado en la ejecución del proyecto. Primero se incluye gran parte del trabajo realizado con *Mathematica*. De éste, se ha excluido el código de algunas simulaciones numéricas y la mayoría de gráficos. Y después se incluye gran parte del código en Matlab: el archivo principal donde se ha realiza la mayor parte del trabajo de diseño del control y la función completa de la etapa de navegación.

2. . Mathematica

Código programado en Mathematica. La declaración de las constantes y las variables refleja la notación empleada en este trabajo en la medida de lo posible.

2.1 Matriz de transformación ejes horizonte local a ejes cuerpo

```
Giroδ3 = {{Cos[δ3], Sin[δ3], 0}, {-Sin[δ3], Cos[δ3], 0}, {0, 0, 1}};
```

Giro alrededor del eje Y (Asiento Longitudinal)

```
Giroδ2 = {{Cos[δ2], 0, -Sin[δ2]}, {0, 1, 0}, {Sin[δ2], 0, Cos[δ2]}};
```

Giro alrededor del eje X (Asiento Lateral)

```
Giroδ1 = {{1, 0, 0}, {0, Cos[δ1], Sin[δ1]}, {0, -Sin[δ1], Cos[δ1]}};
```

Matriz Resultante

```
M321 = Giroδ1.Giroδ2.Giroδ3;
```

De Horizonte Local a Ejes Cuerpo $\delta_3 \rightarrow \psi$, $\delta_2 \rightarrow \theta$, $\delta_1 \rightarrow \phi$

```
MBodyLocal = Mbl = M321 /. {δ3 → ψ[t], δ2 → θ[t], δ1 → φ[t]};
```

2.2 Modelo matemático

```
vel[t_] = {u[t], v[t], w[t]};
```

```
ω[t_] = {p[t], q[t], r[t]};
```

```
V[t_] = Join[vel[t], ω[t]};
```

$$\text{velx}[t_] = \begin{pmatrix} 0 & -w[t] & v[t] \\ w[t] & 0 & -u[t] \\ -v[t] & u[t] & 0 \end{pmatrix}$$

2.2.1 Matrices de masa y matriz de masa añadida

- **M**

```
Mm = ArrayFlatten[{{ m IdentityMatrix[3] -m rgx}, {m rgx, Icb}}];
```

$$\text{rgx} = \begin{pmatrix} 0 & -z_g & y_g \\ z_g & 0 & -x_g \\ -y_g & x_g & 0 \end{pmatrix}; \quad \omega_x[t_] = \begin{pmatrix} 0 & -r[t] & q[t] \\ r[t] & 0 & -p[t] \\ -q[t] & p[t] & 0 \end{pmatrix}; \quad \text{Icb} = \begin{pmatrix} I_{xxcb} & I_{xycb} & I_{xzcb} \\ I_{yxcb} & I_{yycb} & I_{yzcb} \\ I_{zxcb} & I_{zycb} & I_{zzcb} \end{pmatrix};$$

- M_A

```
Madd = ArrayFlatten[{{M11 M12},
                    {M21 M22}}];
```

$$M11 = \begin{pmatrix} Xup1 & 0 & 0 \\ 0 & Yvp1 & 0 \\ 0 & 0 & Zwp1 \end{pmatrix}; \quad M12 = M21 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}; \quad M22 = \begin{pmatrix} Kpp1 & 0 & 0 \\ 0 & Nrp1 & 0 \\ 0 & 0 & Mqp1 \end{pmatrix};$$

2.2.2 Contribuciones del modelo

- τ_I

```
 $\tau_I = Flatten[{{-m \omega x[t].vel[t] + m \omega x[t].rgx.\omega[t]},
                {-m rgx.\omega x[t].vel[t] - \omega x[t].Icb.\omega[t]}}$ 
```

- τ_G

```
GravityLocal = {0, 0, g};
```

(*Eje Z definido positivo hacia el centro de la tierra*)

```
GravityBody = MBodyLocal.GravityLocal;
```

```
Fg = m GravityBody
```

```
{-g m Sin[ $\theta[t]$ ], g m Cos[ $\theta[t]$ ] Sin[ $\phi[t]$ ], g m Cos[ $\theta[t]$ ] Cos[ $\phi[t]$ ]}
```

```
 $\tau_g = Join[Fg, rgx.Fg];$ 
```

- τ_B

AS: Aerostatics.

```
Fas = LBooyant;
```

```
ras = {0, 0, 0}; rasx =  $\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix};$ 
```

(*posición de la fuerza as *)

```
ASLocal = {0, 0, -Fas};
```

```
ASBody = MBodyLocal.ASLocal;
```

```
 $\tau_{AS} = Join[ASBody, rasx.ASBody];$ 
```

- τ_A

$$\tau_{add} = -\text{Flatten}\left[\left(\begin{array}{l} \omega x[t].M11.vel[t] + \omega x[t].M12.\omega[t] \\ velx[t].(M11.vel[t] + M12.\omega[t]) + \omega x[t].(M21.vel[t] + M22.\omega[t]) \end{array}\right)\right]$$

- τ_C

$$FTizq = \begin{pmatrix} F1[\delta T1] \text{Cos}[\delta] \\ 0 \\ F1[\delta T1] \text{Sin}[\delta] \end{pmatrix}; \quad FTder = \begin{pmatrix} F2[\delta T2] \text{Cos}[\delta] \\ 0 \\ F2[\delta T2] \text{Sin}[\delta] \end{pmatrix};$$

$$rTxizq = \begin{pmatrix} 0 & -zT & -yT \\ zT & 0 & -xT \\ yT & xT & 0 \end{pmatrix}; \quad rTxder = \begin{pmatrix} 0 & -zT & yT \\ zT & 0 & -xT \\ -yT & xT & 0 \end{pmatrix};$$

$$\tau T = \text{Join}[FTizq + FTder, rTxizq.FTizq + rTxder.FTder]$$

- $\tau_{Vis} + \tau_{Axial}$

$$Velm[t_] = \sqrt{u[t]^2 + v[t]^2 + w[t]^2};$$

$$q0[t_] = \frac{1}{2} \rho_{air0} Velm[t]^2;$$

$$\{uv[t_], vv[t_], wv[t_]\} = vel[t] + \text{Cross}[\omega[t], \{xv, 0, 0\}];$$

$$\gamma[t_] = \text{ArcTan}\left[\frac{\sqrt{vv[t]^2 + wv[t]^2}}{uv[t]}\right];$$

$$\alpha[t_] = \text{ArcTan}\left[\frac{\sqrt{v[t]^2 + w[t]^2}}{u[t]}\right];$$

$$FVN = \text{CteVis2} q0[t] \text{Sin}[\gamma[t]]^2 + \text{CteVis1} q0[t] \text{Sin}[2 \gamma[t]];$$

$$MVN = \text{CteVis4} q0[t] \text{Sin}[\gamma[t]]^2 + \text{CteVis3} q0[t] \text{Sin}[2 \gamma[t]];$$

$$\tau_{vis} = \begin{pmatrix} -q0[t] CD0 S \text{Cos}[\alpha[t]]^2 \\ FVN * \frac{-w[t]}{\sqrt{vv[t]^2 + wv[t]^2}} \\ FVN * \frac{-w[t]}{\sqrt{vv[t]^2 + wv[t]^2}} \\ 0 \\ MVN * \frac{w[t]}{\sqrt{vv[t]^2 + wv[t]^2}} \\ MVN * \frac{-w[t]}{\sqrt{vv[t]^2 + wv[t]^2}} \end{pmatrix};$$

2.3 Relación Ángulos de Euler y ecuaciones cinemáticas

```

ec4 =  $\theta'$ [t] == q[t] Cos[ $\phi$ [t]] - r[t] Sin[ $\phi$ [t]];

ec8 =  $\phi'$ [t] == p[t] + r[t] Cos[ $\phi$ [t]] Tan[ $\theta$ [t]] + q[t] Sin[ $\phi$ [t]] Tan[ $\theta$ [t]];

ec9 =  $\psi'$ [t] == Sec[ $\theta$ [t]] (r[t] Cos[ $\phi$ [t]] + q[t] Sin[ $\phi$ [t]]);

ec10 =  $x'$ [t] == Cos[ $\theta$ [t]] Cos[ $\psi$ [t]] u[t] + (Cos[ $\psi$ [t]] Sin[ $\theta$ [t]] Sin[ $\phi$ [t]] - Cos[ $\phi$ [t]] Sin[ $\psi$ [t]]) v[t] +
(Cos[ $\phi$ [t]] Cos[ $\psi$ [t]] Sin[ $\theta$ [t]] + Sin[ $\phi$ [t]] Sin[ $\psi$ [t]]) w[t];

ec11 =  $y'$ [t] == Cos[ $\theta$ [t]] Sin[ $\psi$ [t]] u[t] + (Cos[ $\phi$ [t]] Cos[ $\psi$ [t]] + Sin[ $\theta$ [t]] Sin[ $\phi$ [t]] Sin[ $\psi$ [t]]) v[t] +
(-Cos[ $\psi$ [t]] Sin[ $\phi$ [t]] + Cos[ $\phi$ [t]] Sin[ $\theta$ [t]] Sin[ $\psi$ [t]]) w[t];

ec12 =  $z'$ [t] == -Sin[ $\theta$ [t]] u[t] + Cos[ $\theta$ [t]] Sin[ $\phi$ [t]] v[t] + Cos[ $\theta$ [t]] Cos[ $\phi$ [t]] w[t];

```

2.4 Ecuaciones diferenciales reordenadas

Nota: $\mathbf{M} = (\mathbf{M}_m + \mathbf{M}_A)$

```

ec1 = (M.V'[t])[{1}] == Flatten[( $\tau_I + \tau_g + \tau_{AS} + \tau_{add} + \tau_T + \tau_{vis2}$ )][{1}] // Simplify;

ec5 = (M.V'[t])[{2}] == Flatten[( $\tau_I + \tau_g + \tau_{AS} + \tau_{add} + \tau_T + \tau_{vis2}$ )][{2}] // Simplify;

ec2 = (M.V'[t])[{3}] == Flatten[( $\tau_I + \tau_g + \tau_{AS} + \tau_{add} + \tau_T + \tau_{vis2}$ )][{3}] // Simplify;

ec6 = (M.V'[t])[{4}] == Flatten[( $\tau_I + \tau_g + \tau_{AS} + \tau_{add} + \tau_T + \tau_{vis2}$ )][{4}] // Simplify;

ec3 = (M.V'[t])[{5}] == Flatten[( $\tau_I + \tau_g + \tau_{AS} + \tau_{add} + \tau_T + \tau_{vis2}$ )][{5}] // Simplify;

ec7 = (M.V'[t])[{6}] == Flatten[( $\tau_I + \tau_g + \tau_{AS} + \tau_{add} + \tau_T + \tau_{vis2}$ )][{6}] // Simplify;

SistemaEcuaciones = {ec1, ec2, ec4, ec5, ec6, ec7, ec8, ec9};

```

2.5 Expresiones para los bloques Simulink

```
M1 = Inverse[M] // FullSimplify;
```

Nota: Se ha separado la contribución del momento de Munk de los términos de masa aparente:

```

 $\tau_{add} = -\text{Flatten}\left[\left(\begin{array}{c} \omega_x.M11.vel + \omega_x.M12.\omega \\ vel_x.(M12.\omega) + \omega_x.(M21.vel + M22.\omega) \end{array}\right)\right];$ 

 $\tau_{Munk} = -\text{Join}\{0, 0, 0\}, vel_x.(M11.vel)\};$ 

```

El vector u del bloque *Fcn* y cada una de las contribuciones despejando la matriz de masas:

```

U = {u → u01, w → u02, q → u03, Theta → u04, v → u05, p → u06, r → u07, Phi → u08,
Psi → u09, F1 → u13, F2 → u14, delta → u15};

```

```

ecsI = M1.rI // FullSimplify;
ecsg = M1.rg // FullSimplify;
ecsAS = M1.rAS // FullSimplify;
ecsT = M1.rT // FullSimplify;
ecsAdd = M1.radd // FullSimplify;
ecsMunk = M1.rMunk // FullSimplify;
ecsVis = M1.rvis // FullSimplify;

```

Los denominadores de las dos dinámicas, por observación:

```

DenLong =
(-m (Iyycb + Nrp1 - m xg^2) (m + Xup1) + m^3 zg^2 -
((Iyycb + Nrp1) (m + Xup1) - m^2 zg^2) Zvp1);
ToMatlab[DenLong]

```

```

DenLatDir =
(-Ixxcb m (Izzcb + Mqp1 - m xg^2) - Ixxcb (Izzcb + Mqp1) Yvp1 + Ixzcb^2 (m + Yvp1) +
2 Ixzcb m^2 xg zg + m^2 (Izzcb + Mqp1) zg^2);
ToMatlab[DenLatDir]

```

Ecuaciones de la dinámica longitudinal. La variable *variablenum* se cambia según se requieran las expresiones de u, w, y q siendo 1,3 y 5 respectivamente.

```

variablenum = 5;

ec111 = (ecsI[[variablenum]] * DenLong // Simplify) /. U
ec112 = (ecsg[[variablenum]] * DenLong // Simplify) /. U
ec13 = (ecsAS[[variablenum]] * DenLong // Simplify) /. U
ec14 = (ecsT[[variablenum]] * DenLong // Simplify) /. U
ec15 = (ecsAdd[[variablenum]] * DenLong // Simplify) /. U
ec16 = (ecsMunk[[variablenum]] * DenLong // Simplify) /. U
ec17 = (ecsVis[[variablenum]] * DenLong // Simplify) /. U

```

Para la dinámica lateral-direccional se procede de la misma manera pero cambiando el denominador. En este caso las variables son v, p y r con 2, 4 y 6 respectivamente.

```

variablenum = 6;

ec111 = (ecsI[[variablenum]] * DenLatDir // Simplify) /. U
ec112 = (ecsg[[variablenum]] * DenLatDir // Simplify) /. U
ec13 = (ecsAS[[variablenum]] * DenLatDir // Simplify) /. U
ec14 = (ecsT[[variablenum]] * DenLatDir // Simplify) /. U
ec15 = (ecsAdd[[variablenum]] * DenLatDir // Simplify) /. U
ec16 = (ecsMunk[[variablenum]] * DenLatDir // Simplify) /. U
ec17 = (ecsVis[[variablenum]] * DenLatDir // Simplify) /. U

```

Finalmente, se obtienen las expresiones en Matlab. El *string* obtenido es modificado para adaptarlo al formato del bloque Simulink. Un ejemplo:

Diseño, modelado y control del sistema de propulsión para un vehículo aéreo no tripulado basado en globo aerostático dirigible

```
StringReplace[ToMatlab[ec111, 1500],  
{ "u01" → "u(1)", "u02" → "u(2)", "u03" → "u(3)", "u04" → "u(4)",  
  "u05" → "u(5)", "u06" → "u(6)", "u07" → "u(7)", "u08" → "u(8)",  
  "u09" → "u(9)", "u13" → "u(13)", "u14" → "u(14)", "u15" → "u(15)", "." → ""}]
```

Las expresiones resultantes se pueden copiar dentro del bloque *Fcn*.

2.6 Constantes del proyecto

```
a = 0.85; b = 0.374741;  
Vol = 0.5;  
Sref = S = Vol2/3;  
  
zg = 119.52 / 1000;      Ixxcb = 0.068;  
                          Iyycb = 0.128;  
zT = 402 / 1000;      Izzcb = 0.098;  
YT =  $\frac{320}{2 \times 1000}$ ;      Ixzcb =  $9.564 \times 10^{-4}$ ;  
  
(* Aire y Helio *)  
rhoair0 = 1.223; (* Atmósfera Standard ISA a 15°C *)  
rho100He = 0.1785;  
rho90He = 0.9 rho100He + 0.1 rhoair0;  
  
g = 9.8;  
LBuoyant = g * Vol * (rhoair0); (* Flotación *)  
m = LBuoyant / 9.8; (* 611.5 g *)  
(*Masa aparente*)  
  
k1 = 0.177761; (*coeficientes *)  
k2 = 0.737723;  
k3 = 0.310559;  
  
Voldes = (4 / 3) pi a b2; (*Simplific. Esferoide Prolato*)  
Ides = (4 / 15) pi a b2 (a2 + b2);  
  
Xup1 = k1 rhoair0 Voldes;  
(*Una constante por la masa de aire desplazado*)  
Yvp1 = Zwpl = k2 rhoair0 Voldes;  
Nrp1 = Mqp1 = k3 rhoair0 Ides;
```


Diseño, modelado y control del sistema de propulsión para un vehículo aéreo no tripulado basado en globo aerostático dirigible

```
(* Resistencia aero *)

CD0 = 0.315662;
(*Estas están integradas desde  $\epsilon_v$  retirando parte de a contribución
potencial*)

CteVis1 = 0.084957867498;
CteVis2 = 0.033037609478;
CteVis3 = -0.065593888575;
CteVis4 = -0.024899016470;

xv = -0.6885; (*este es el punto donde empiezan a predominar los
efectos viscosos*)
```

2.7 Punto de funcionamiento. Vuelo horizontal.

Primero se toman las ecuaciones de la dinámica longitudinal y se introduce el vector de equilibrio. Por simetría se reduce el término propulsivo a una sola entrada de fuerza, *Prop*, y al ángulo de inclinación de los motores, delta:

```
ecs = Inverse[M].(τI + τg + τAS + τadd + τT + τvis);
Xeq0 = {u[t] → u0, v[t] → 0, w[t] → 0.000001, p[t] → 0, q[t] → 0, r[t] → 0,
  θ[t] → θ0, φ[t] → 0, (F1[δT1] + F2[δT2]) → Prop};

ec10 = ((ecs[[1, 1]] /. Xeq0) // FullSimplify) // Chop
ec20 = ((ecs[[3, 1]] /. Xeq0) // FullSimplify) // Chop
ec30 = ((ecs[[5, 1]] /. Xeq0) // FullSimplify) /. (F1[δT1] + F2[δT2]) → Prop //
  FullSimplify // Chop
ecs0 = {ec10, ec20, ec30};
```

Ahora se calcula la relación entre la propulsión y la posición del centro de gravedad en el eje x para generar un momento tal que se pueda alcanzar las condiciones del punto de funcionamiento del vector X_{eq} .

```
Prop0 = Prop /. Flatten[Solve[ecs0[[1]] = 0 /. {δ → 0, θ0 → 0}, Prop] // FullSimplify];
(*y la posición del cdg*)
xg[u0_] = xg /. Solve[({ecs0[[3]] /. {Prop → Prop0, δ → 0, θ0 → 0}) // Chop // FullSimplify] = 0, xg];
```

El resultado de estas operaciones se ha ilustrado en el apartado 3.5 en la **¡Error! No se encuentra el origen de la referencia..**

Si se quiere especificar para una velocidad en concreto, por ejemplo 1 m/s:

```
Clear[u0, θ0, Prop0, xg]
Prop0 = Prop /. Flatten[Solve[ecs0[[1]] = 0 /. {δ → 0, u0 → 1, θ0 → 0}, Prop] // FullSimplify]
(*y la posición del cdg*)
Solve[({ecs0[[2]] // FullSimplify) /. {Prop → Prop0, δ → 0, u0 → 1, θ0 → 0}) = 0, xg]
Solve[({ecs0[[3]] // FullSimplify) /. {Prop → Prop0, δ → 0, u0 → 1, θ0 → 0}) = 0, xg]
```

De donde se obtiene que para 1 m/s el centro de gravedad debe estar adelantado 0,00815 m

Imponiendo la hipótesis de ángulo pequeños se puede establecer el valor de la propulsión, de delta y del valor de Theta que equilibran el modelo en un rango de velocidades:

```
SistemaTrim01 =
  Table[ecs0[[i]] == 0, {i, 1, 3}] /.
  {Cos[θ0] → 1, Sin[θ0] → θ0, Cos[δ] → 1, Sin[δ] → δ} // FullSimplify;

{Prop01, δ01, θ01} =
  {Prop, δ, θ0} /.
  Flatten[Solve[SistemaTrim01, {Prop, δ, θ0}] // Simplify]
```

Con lo que se obtienen 3 ecuaciones, de las cuales se han graficado la propulsión y theta en la Figura 25 de la Memoria del TFG.

2.8 Punto de funcionamiento. Vuelo espiral.

Primero, los datos de la espiral a realizar y fijar dos condiciones iniciales.

```
Radio = 50; (*m*)
u0 = 1; (*m/s*)

θasc = ArcTan[(2 Pi) / (2 Pi Radio)] // N (*2Pi / 2Pi*Radio*)

θ0 = 0.0031075444781329255` (* dato obtenido de una simulación *)

w0 = -(u0 * θasc);
w0 = -(u0 * θasc) + θ0;
```

Se calcula ahora el valor del resto de estados para lograr el equilibrio. Los pasos previos al cálculo son los siguientes:

```
EqGiro = {u'[t] → 0, v'[t] → 0, w'[t] → 0,
  u[t] → u0, v[t] → vTrim, w[t] → w0,

  p'[t] → 0, q'[t] → 0, r'[t] → 0,
  p[t] → pTrim, q[t] → qTrim, r[t] →  $\frac{-u0}{Radio}$ ,
  θ[t] → θ0, φ[t] → φTrim, ψ[t] → ψpTrim * t,
  θ'[t] → 0, φ'[t] → 0, ψ'[t] → ψpTrim,
  δ → δTrim, F1[δT1] → F1Trim, F2[δT2] → F2Trim};

SistemaEquilibrio = Table[(SistemaEcuaciones[[i]] /. EqGiro), {i, 1, 8}];
```

Y el cálculo:

```
SistemaEquilibrioSol = FindRoot[SistemaEquilibrio, {
  {F1Trim, 0.01, 0, 1}, {F2Trim, 0.01, 0, 1}, {δTrim, 0},
  {vTrim, 0}, {pTrim, 0.0001},
  {qTrim, 0.001}, {ψpTrim, 1/Radio}, {φTrim, -0.01}},
  WorkingPrecision → 20, AccuracyGoal → Infinity, PrecisionGoal → 10];
```

Para simular el vuelo en espiral hay que emplear el siguiente código.

```

PropTrim = {δ → δTrim, F1[δT1] → F1Trim, F2[δT2] → F2Trim} /.
  SistemaEquilibrioSol;

SistemaCompleto =
  ( {ec1, ec2, ec3, ec4, ec5, ec6, ec7, ec8, ec9, ec10, ec11, ec12,

    x[0] == 0, y[0] == 0, z[0] == 0,

    u[0] == u0, v[0] == vTrim, w[0] == w0,

    p[0] == pTrim, q[0] == qTrim, r[0] ==  $\frac{u0}{Radio}$ ,

    θ[0] == θ0, φ[0] == φTrim, ψ[0] == 0 } /. PropTrim ) /.
  SistemaEquilibrioSol;

system2 = NDSolve[SistemaCompleto,
  {x[t], y[t], z[t], u[t], v[t], w[t], p[t], q[t], r[t], θ[t], φ[t], ψ[t]},
  {t, 0, TF}];

```

Como se ha comentado al inicio de este punto, los resultados de las simulaciones para los valores de algunos estados diferían de algunos de los valores calculados con *FindRoot*. Se modificó el código para calcular otros estados y dar como valores iniciales algunos de los resultados obtenidos. Por tanto, el proceso es iterativo.

2.9 Linealización del modelo. Matrices analíticas

Este proceso para linealizar las ecuaciones puede ser optimizado si se realiza contribución a contribución, al estilo de la implementación. Sería necesario simplificar el término de efectos viscosos aplicando, como se realiza con el resto de las ecuaciones, para que la linealización fuese satisfactoria. El resultado de este código, si bien correcto, es muy farragoso y por ello se desechó emplearlo en el proyecto:

```
ecuaciones = {ec1, ec2, ec3, ec4, ec5, ec6, ec7, ec8};
```

Hipótesis de ángulos pequeños (no siempre realista. Mejor en actuaciones cerca del vuelo en el plano horizontal):

```

ecuaciones =
  ecuaciones /. {Cos[φ[t]] → 1, Sin[φ[t]] → φ[t], Tan[φ[t]] → φ[t], Cos[θ[t]] → 1,
    Sin[θ[t]] → θ[t], Tan[θ[t]] → θ[t], Cos[δ] → 1, Sin[δ] → δ, Tan[δ] → δ};

```

Punto de vuelo genérico:

Diseño, modelado y control del sistema de propulsión para un vehículo aéreo no tripulado basado en globo aerostático dirigible

```
X0 = {u[t] → u0, v[t] → v0, w[t] → w0, p[t] → p0, q[t] → q0, r[t] → r0, θ[t] → θ0,
      φ[t] → φ0};
```

```
U0 = {F1[δT1] → F10, F2[δT2] → F20, δ → δ0};
```

```
ecuaciones0 = (ecuaciones /. X0) /. U0;
```

Ecuaciones en incrementos:

```
XΔ = {u[t] → u0 + Δu, v[t] → v0 + Δv, w[t] → w0 + Δw, p[t] → p0 + Δp, q[t] → q0 + Δq,
      r[t] → r0 + Δr, θ[t] → θ0 + Δθ, φ[t] → φ0 + Δφ};
```

```
UΔ = {F1[δT1] → F10 + ΔF1, F2[δT2] → F20 + ΔF2, δ → δ0 + Δδ};
```

```
ecuacionesΔ = ((ecuaciones /. XΔ) /. UΔ) - ecuaciones0;
```

Se eliminan los términos de segundo orden:

```
Incrementos = {Δu, Δv, Δw, Δp, Δq, Δr, Δθ, Δφ, ΔF1, ΔF2, Δδ};
```

```
IncrementosSegundoOrden = Times@@# & /@Tuples[Incrementos, 2];
```

```
igualdades = Thread[IncrementosSegundoOrden == 0];
```

```
ecuacionesΔLin1 = Simplify[ecuacionesΔ[[1]], igualdades] // FullSimplify // Flatten;
```

```
ecuacionesΔLin2 = Simplify[ecuacionesΔ[[2]], igualdades] // FullSimplify // Flatten;
```

```
ecuacionesΔLin3 = Simplify[ecuacionesΔ[[3]], igualdades] // FullSimplify // Flatten;
```

```
ecuacionesΔLin4 = Simplify[ecuacionesΔ[[4]], igualdades] // FullSimplify // Flatten;
```

```
ecuacionesΔLin5 = Simplify[ecuacionesΔ[[5]], igualdades] // FullSimplify // Flatten;
```

```
ecuacionesΔLin6 = Simplify[ecuacionesΔ[[6]], igualdades] // FullSimplify // Flatten;
```

```
ecuacionesΔLin7 = Simplify[ecuacionesΔ[[7]], igualdades] // FullSimplify // Flatten;
```

```
ecuacionesΔLin8 = Simplify[ecuacionesΔ[[8]], igualdades] // FullSimplify // Flatten;
```

Y se obtienen las matrices:

```
ΔX = {Δu, Δw, Δq, Δθ, Δv, Δp, Δr, Δφ};
```

```
ΔU = {ΔF1, ΔF2, Δδ};
```

```

ecuacion1ΔLinA = Table[D[ecuacionesΔLin1, ΔX[[j]]], {j, 1, 8}] // FullSimplify;
ecuacion1ΔLinB = Table[D[ecuacionesΔLin1, ΔU[[j]]], {j, 1, 3}] // FullSimplify;

ecuacion2ΔLinA = Table[D[ecuacionesΔLin2, ΔX[[j]]], {j, 1, 8}] // FullSimplify;
ecuacion2ΔLinB = Table[D[ecuacionesΔLin2, ΔU[[j]]], {j, 1, 3}] // FullSimplify;

ecuacion3ΔLinA = Table[D[ecuacionesΔLin3, ΔX[[j]]], {j, 1, 8}] // FullSimplify;
ecuacion3ΔLinB = Table[D[ecuacionesΔLin3, ΔU[[j]]], {j, 1, 3}] // FullSimplify;

ecuacion4ΔLinA = Table[D[ecuacionesΔLin4, ΔX[[j]]], {j, 1, 8}] // FullSimplify;
ecuacion4ΔLinB = Table[D[ecuacionesΔLin4, ΔU[[j]]], {j, 1, 3}] // FullSimplify;

ecuacion5ΔLinA = Table[D[ecuacionesΔLin5, ΔX[[j]]], {j, 1, 8}] // FullSimplify;
ecuacion5ΔLinB = Table[D[ecuacionesΔLin5, ΔU[[j]]], {j, 1, 3}] // FullSimplify;

ecuacion6ΔLinA = Table[D[ecuacionesΔLin6, ΔX[[j]]], {j, 1, 8}] // FullSimplify;
ecuacion6ΔLinB = Table[D[ecuacionesΔLin6, ΔU[[j]]], {j, 1, 3}] // FullSimplify;

ecuacion7ΔLinA = Table[D[ecuacionesΔLin7, ΔX[[j]]], {j, 1, 8}] // FullSimplify;
ecuacion7ΔLinB = Table[D[ecuacionesΔLin7, ΔU[[j]]], {j, 1, 3}] // FullSimplify;

ecuacion8ΔLinA = Table[D[ecuacionesΔLin8, ΔX[[j]]], {j, 1, 8}] // FullSimplify;
ecuacion8ΔLinB = Table[D[ecuacionesΔLin8, ΔU[[j]]], {j, 1, 3}] // FullSimplify;

A = {ecuacion1ΔLinA, ecuacion2ΔLinA, ecuacion3ΔLinA, ecuacion4ΔLinA, ecuacion5ΔLinA,
     ecuacion6ΔLinA, ecuacion7ΔLinA, ecuacion8ΔLinA};

B = {ecuacion1ΔLinB, ecuacion2ΔLinB, ecuacion3ΔLinB, ecuacion4ΔLinB, ecuacion5ΔLinB,
     ecuacion6ΔLinB, ecuacion7ΔLinB, ecuacion8ΔLinB};

```

A.9 Linealización del modelo. Resultado numérico

Primero es necesario definir varios vectores. El código puede incluir o extraer el momento de Munk comentando o descomentando las líneas necesarias:

```

Import["...\ToMatlab.m"]

(*sin Momento de Munk*)
τadd = -Flatten[{{
    ωx[t].M11.vel[t] + ωx[t].M12.ω[t]
    velx[t].(M12.ω[t]) + ωx[t].(M21.vel[t] + M22.ω[t])
}}];

(*CON Momento de Munk*)
(*τadd=-Flatten[{{
    ωx[t].M11.vel[t] + ωx[t].M12.ω[t]
    velx[t].(M11.vel[t]+M12.ω[t]) + ωx[t].(M21.vel[t]+M22.ω[t])
}}];*)
(*Vector de vectores*)
τ = {τI, τg, τAS, τadd, τT, τvis};

(*Vector de SS*)
X = {u[t], w[t], q[t], θ[t], v[t], p[t], r[t], φ[t], ψ[t]};
(*Vector de control*)
U = {F1[δT1], F2[δT2], δ};

```

En el vector de estados se ha incluido el rumbo por si se quisiera incluir como estado a controlar, aunque lo habitual sería dejarlo fuera y controlar dicho ángulo con seguimiento de referencias.

```
(*Punto de equilibrio genérico*)
X0 = {u[t] → u0, v[t] → v0, w[t] → w0,
      p[t] → p0, q[t] → q0, r[t] → r0, θ[t] → θ0, φ[t] → φ0, ψ[t] → ψ0};
U0 = {F1[δT1] → F10, F2[δT2] → F20, δ → δ0};
```

Se define el punto de equilibrio del sistema. Para obtener los resultados es necesario que todos los datos del sistema estén introducidos.

```
(*Datos Equilibrio*)
Xeq = {u0 → 1, v0 → 0, w0 → 1 × 10-10, p0 → 0, q0 → 0, r0 → 0, θ0 → 0, φ0 → 0, ψ0 → 0};
Ueq = {F10 → Prop0 / 2, F20 → Prop0 / 2, δ0 → 0};
```

Se incluyen las ecuaciones de las derivadas de los ángulos de Euler y las cinemáticas. Dos necesarias, por ser estados del sistema, y las otras ecuaciones se incluyen por si se quiere ampliar la matriz de estados:

```
(* Ángulos de Euler y ecuaciones cinemáticas *)
ec4 = q[t] Cos[φ[t]] - r[t] Sin[φ[t]]; (*θ' [t]*)
ec8 = p[t] + r[t] Cos[φ[t]] Tan[θ[t]] + q[t] Sin[φ[t]] Tan[θ[t]] (*φ' [t]*) ;
ec9 = Sec[θ[t]] (r[t] Cos[φ[t]] + q[t] Sin[φ[t]]) (*ψ' [t]*) ;
ec10 = Cos[θ[t]] Cos[ψ[t]] u[t] +
      (Cos[ψ[t]] Sin[θ[t]] Sin[φ[t]] - Cos[φ[t]] Sin[ψ[t]]) v[t] +
      (Cos[φ[t]] Cos[ψ[t]] Sin[θ[t]] + Sin[φ[t]] Sin[ψ[t]]) w[t] (*x' [t]*) ;
ec11 = Cos[θ[t]] Sin[ψ[t]] u[t] +
      (Cos[φ[t]] Cos[ψ[t]] + Sin[θ[t]] Sin[φ[t]] Sin[ψ[t]]) v[t] +
      (-Cos[ψ[t]] Sin[φ[t]] + Cos[φ[t]] Sin[θ[t]] Sin[ψ[t]]) w[t] (*y' [t]*) ;
ec12 = -Sin[θ[t]] u[t] + Cos[θ[t]] Sin[φ[t]] v[t] + Cos[θ[t]] Cos[φ[t]] w[t] (* z' [t]*) ;

AngPosEcs = {ec4, ec8, ec9, ec10, ec11, ec12};

TerminosAAngPos =
  Table[{{D[AngPosEcs[[i]], X[[j]]] /. X0) /. U0}, {i, 1, 6}, {j, 1, 9}];
```

Las dos matrices, para los 8 estados, se obtendrían así:

```
TerminosA = Table[Flatten[
  Total[
    Table[{{D[{{Inverse[M].(τ[[i]])}][nEcu] // FullSimplify), X[[j]]] /. X0) /. U0},
    {i, 1, 6}, {j, 1, 8}] // Simplify
  , {1}]]
, {nEcu, 1, 6}];

TerminosA4 = Take[TerminosAAngPos[[1]], 8];
TerminosA8 = Take[TerminosAAngPos[[2]], 8];

A8 = {TerminosA[[1]], TerminosA[[3]], TerminosA[[5]], TerminosA4,
      TerminosA[[2]], TerminosA[[4]], TerminosA[[6]], TerminosA8};
```

```
TerminosB = Table[Flatten[
  Total[
    Table[(D[(Inverse[M].(τ[[i]])][[nEcu]] // FullSimplify), U[[j]]] /. X0) /.
      U0), {i, 1, 6}, {j, 1, 3}] // Simplify
  , {1}]]
, {nEcu, 1, 6}];

B8 = {TerminosB[[1]], TerminosB[[3]], TerminosB[[5]], TerminosBANGPos[[1]],
  TerminosB[[2]], TerminosB[[4]], TerminosB[[6]], TerminosBANGPos[[2]]};
```

Finalmente, para trasladar las ecuaciones a *Matlab*:

```
ToMatlab[(A8 /. Xeq // Chop), 15 000]
```

```
ToMatlab[(B8 /. Ueq // Chop), 15 000]
```

3. Matlab

3.1 Archivo principal

```
% Datos y Constantes del proyecto

a = 0.85;    b = 0.374741;
Vol = 0.5;  S= Vol^(2/3); %Sref
%Pos. C.G.
zg = 119.52/1000;  xg = 0.0081571;    yg = 0;

%Pos motores
zT = 402/1000;    yT = 320/(2*1000);  xT = -0.01;

% Inercia
Ixxcb = 0.068;    Iyycb = 0.128;    Izzcb = 0.098;
Ixzcb = 9.564*10^-4;
ICB=[Ixxcb 0 Ixzcb, 0 Iyycb 0,Ixzcb 0 Izzcb];

% Gravedad y aeroestática
rhoair0 = 1.223;
rho100He = 0.1785;
rho90He = 0.9*rho100He + 0.1*rhoair0;
g = 9.8;

LBuoyant = g*Vol*(rhoair0);
m=LBuoyant/g; %Flotabilidad neutra

% Masa aparente
k1 = 0.177761;    k2 = 0.737723;    k3 = 0.310559;

Voldes = (4/3)*pi*a*b^2 ;
Ides = (4/15)*(pi*a*b^2)*(a^2 + b^2);

Xupl = k1*rhoair0*Voldes;
Yvpl = k2*rhoair0*Voldes;
Zwpl = k2*rhoair0*Voldes;
Nrpl = k3*rhoair0*Ides;
Mqpl = k3*rhoair0*Ides;

% Aero
CD0 = 0.315662;

CteVis1 = 0.084957867498;
CteVis2 = 0.033037609478;
CteVis3 = -0.065593888575;
CteVis4 = -0.024899016470;

XC=-0.6885; %posición efectos viscosos
```



```

%% Modelo no lineal - Simulink

% Al introducir las fórmulas distinguiendo entre dinámica longitudinal y
% lateral-direccional se puede observar que cada conjunto está dividido por
% la misma expresión.
%
% Para acortar la entrada de los bloques FCN de simulink calculamos su
% valor y lo introduciremos en el modelo con un bloque Gain antes del
% integrador.

% Las expresiones de los bloques FCN han sido calculadas con Mathematica y
% transformadas con al formato de simulink y matlab empleando funciones
% de Mathematica.

%Divisores - Completos

DLong=(-1)*m.*(Iyycb+Nrp1+(-1).*m.*xg.^2).*(m+Xup1)+m.^3.*zg.^2+(-1).*((Iyycb+Nrp1)...
.*(m+Xup1)+(-1).*m.^2.*zg.^2).*Zwpl;
DLongl=1/DLong;

DLatDir=(-1).*Ixxcb.*m.*(Izzcb+Mqp1+(-1).*m.*xg.^2)+(-1).*Ixxcb.*(Izzcb+Mqp1)...
.*Yvpl+Ixxcb.^2.*(m+Yvpl)+2.*Ixxcb.*m.^2.*xg.*zg+m.^2.*(Izzcb+Mqp1).*zg.^2;
DLatDir1=1/DLatDir;

%% En espacio de estados:

% Sin M. Munk [8]
A8= [(-0.354016E0),0.534151E-1,0.378401E-1,0.474029E0,0,0,0,0;...
0.755534E-3,(-0.100251E0),0.608689E0,(-0.219266E-1),0,0,0,0;...
0.160955E0,(-0.526358E0),(-0.37288E0),(-0.467113E1),0,0,0,0;...
0,0,1,0,0,0,0,0;0,0,0,0,(-0.109367E0),0.1E-9,(-0.576641E0),(-0.785267E0);...
0,0,0,0,(-0.126247E0),0,0.461242E0,(-0.113839E2);...
0,0,0,0,0.618524E0,0,(-0.441867E0),0.486995E0;0,0,0,0,0,1,0,0];

B8= [0.118961E1,0.118961E1,(-0.118254E-2);...
0.919986E-2,0.919986E-2,0.114489E0;...
0.195989E1,0.195989E1,0.116529E-1;...
0,0,0;(-0.748241E-2),0.748241E-2,0;...
(-0.252565E-1),0.252565E-1,0;...
0.122394E1,(-0.122394E1),0;...
0,0,0];
C8=eye(8);
D8=zeros(8,3);

```

%% Análisis del Sistema en Espacio de Estados

% Polos del sistema

```
Polos=eig(A8);
```

% Controlabilidad Total

```
fprintf('Controlabilidad total, número de estados incontrolables:')
```

```
Co=ctrb(A8,B8);
```

```
unCon=length(A8)-rank(Co) % 0: es controlable
```

% Controlabilidad por entradas:

```
    fprintf('Estados controlables con F1 y esfuerzo de control:')
```

```
rank(ctrb(A8,B8(:,1)))
```

```
det(ctrb(A8,B8(:,1)))
```

```
fprintf('Estados controlables con F2 y esfuerzo de control:')
```

```
rank(ctrb(A8,B8(:,2)))
```

```
det(ctrb(A8,B8(:,2)))
```

```
fprintf('Estados controlables con delta y esfuerzo de control:')
```

```
rank(ctrb(A8,B8(:,3)))
```

```
det(ctrb(A8,B8(:,3)))
```

% Índices de controlabilidad:

```
W=[B8(:,1) A8*B8(:,1) A8^2*B8(:,1) ...
```

```
    B8(:,2) A8*B8(:,2) A8^2*B8(:,2) A8^3*B8(:,2)...
```

```
    B8(:,3) ];
```

```
rank(W)
```

```
fprintf('Índices de controlabilidad:')
```

```
mu_F1=[4 3]
```

```
mu_F2=[3 4]
```

```
mu_delta=1
```

```
fprintf('Esfuerzo de control:')
```

```
det(W)
```

```
W2=[B8(:,1) A8*B8(:,1) A8^2*B8(:,1) A8^3*B8(:,1)...
```

```
    B8(:,2) A8*B8(:,2) A8^2*B8(:,2) ...
```

```
    B8(:,3) ];
```

```
det(W2)
```

%% LQR básico para realimentación de los estados

%Diseño básico

```
Qu=1000;  
Qw=2000;  
Qq=2000;  
QTh=10;  
Qv=30000;  
Qp=1000;  
Qr=10;  
QPh=100;
```

```
Q = diag([Qu, Qw, Qq , QTh, Qv, Qp, Qr, QPh]);
```

%% LQR para seguimiento de la referencia: Yaw altura y Theta

```
%
```

```
% Esquema
```

```
%
```

```
% Aa=| A 0|
```

```
%   |-C 0|
```

```
Ce=[1 0 0 0 0 0 0 0;...
```

```
    0 0 0 0 0 0 1 0;...
```

```
    0 0 0 1 0 0 0 0];
```

```
Aa=[A8 zeros(8,3);-Ce zeros(3,3)];
```

```
Ba=[B8;zeros(3,3)] ;
```

```
% El sistema sigue siendo controlable
```

```
Co=ctrb(Aa,Ba);
```

```
unCon=length(Aa)-rank(Co)
```

```
QuE=500; QPsE=2000; QThE=2000;
```

```
Qerr=diag([QuE, QPsE, QThE]);
```

```
Qa=blkdiag(Q,Qerr);
```

```
rho=400;
```

```
dR=20;
```

```
R=diag([rho, rho, dR]);
```

```
K=lqr(Aa,Ba,Qa,R);
```

```
%% para la simulación:
```

```
% Condiciones inicio dinámicas
```

```
u0=0.0001;      v0=0;      w0=0.00001;  
p0=0;          q0=0;      r0=0;  
Theta0=-0.11964; Phi0=0;
```

```
%referencias del control en equilibrio para u = 1 m/s en vuelo horiz.
```

```
F10=0.1216/2; F20=F10; delta0=0;
```

```
Pos=[];
```

Un ejemplo de una de las rutas: el “ocho”.

```
% 8
```

```
Psi0=90*G2R;    x0=50;   y0=-50;   z0=0;
```

```
p1=[50,  50 ,5];  
p2=[-50, 50, 10];  
p3=[-50,-50, 15];  
p4=[50, -50, 20];
```

```
p5=[50,  50 ,25];  
p6=[-50, 50, 30];  
p7=[-50,-50, 35];  
p8=[50, -50, 40];
```

```
p9= [150, -50 ,40];  
p10=[150,-150, 30];  
p11=[50, -150, 20];  
p12=[50, -50, 10];
```

```
p13=[150, -50 , 0];  
p14=[150,-150,-10];  
p15=[50, -150,-20];  
p16=[50, -50,-30];
```

```
Pos.Pos0=[x0 y0 z0];
```

```
Pos.Ruta=[p1;p2;p3;p4;p5;p6;p7;p8;p9;p10;p11;p12;p13;p14;p15;p16];
```

```
%% Simulación
```

```
tsim=2000;
```

```
sim('Dirigible_General_Final.slx')
```

Para realizar los gráficos mostrados en la memoria se ha programado el siguiente código:

```
%% gráficos

% para plotear la ruta

[num nad]=size(Pos.Ruta)
xplot=[ ];
yplot=[ ];
zplot=[ ];

j=1;
while j <= num

    xplot(j,1)=Pos.Ruta(j,1);
    yplot(j,1)=Pos.Ruta(j,2);
    zplot(j,1)=Pos.Ruta(j,3);

    j=j+1;
end

% Trazado de la simulación
figure

hold on
grid on

plot3([yplot(1:(num-1),1)], [xplot(1:(num-1),1)], -[zplot(1:(num-1),1)], ...
    'ob', 'DisplayName', '\itWaypoints')
plot3(y0,x0,-z0, '>g', 'LineWidth',1, 'DisplayName', 'Punto Inicial')
plot3(yplot(num), xplot(num), -zplot(num), '<r', 'LineWidth',1, ...
    'DisplayName', 'Meta')

plot3( PosEspacio.Data(:,2), PosEspacio.Data(:,1), -PosEspacio.Data(:,3), ...
    'r', 'LineWidth',1, 'DisplayName', 'Resultado Simulación')

legend('show', 'AutoUpdate', 'off')

plot3([y0; yplot], [x0; xplot], -[z0; zplot], ':b', 'LineWidth',1) % Ruta

xlabel('y (m)', 'FontName', 'Calibri', 'FontSize',12)
ylabel('x (m)', 'FontName', 'Calibri', 'FontSize',12)
zlabel('h (m)', 'FontName', 'Calibri', 'FontSize',12)

hold off
```

```
%% Plot Estados

% Din. Long

figure

hold on

subplot(4,1,1)
plot(EstadosSim.Time,EstadosSim.Data(:,1),'LineWidth',1)
axis([0 inf min(EstadosSim.Data(:,1)) max(EstadosSim.Data(:,1))])
grid on
ylabel('u [m/s]','FontName','Calibri','FontSize',12)

subplot(4,1,2)
plot(EstadosSim.Time,EstadosSim.Data(:,2),'LineWidth',1)
axis([0 inf min(EstadosSim.Data(:,2)) max(EstadosSim.Data(:,2))])
grid on
ylabel('w [m/s]','FontName','Calibri','FontSize',12)

subplot(4,1,3)
plot(EstadosSim.Time,EstadosSim.Data(:,3),'LineWidth',1)
axis([0 inf min(EstadosSim.Data(:,3)) max(EstadosSim.Data(:,3))])
grid on
ylabel('q [rad/s]','FontName','Calibri','FontSize',12)

subplot(4,1,4)
plot(EstadosSim.Time,EstadosSim.Data(:,4)*R2G,'LineWidth',1)
axis([0 inf min(EstadosSim.Data(:,4)*R2G) max(EstadosSim.Data(:,4)*R2G)])
grid on
xlabel('t [s]','FontName','Calibri','FontSize',12)
ylabel('\theta [\circ]','Interpreter','tex','FontName','Calibri','FontSize',12)

hold off

% Din. Lat - Dir

figure

hold on

subplot(4,1,1)
plot(EstadosSim.Time,EstadosSim.Data(:,5),'LineWidth',1)
axis([0 inf min(EstadosSim.Data(:,5)) max(EstadosSim.Data(:,5))])
grid on
ylabel('v [m/s]','FontName','Calibri','FontSize',12)

subplot(4,1,2)
plot(EstadosSim.Time,EstadosSim.Data(:,6),'LineWidth',1)
axis([0 inf min(EstadosSim.Data(:,6)) max(EstadosSim.Data(:,6))])
grid on
ylabel('p [rad/s]','FontName','Calibri','FontSize',12)
```

```
subplot(4,1,3)
plot(EstadosSim.Time,EstadosSim.Data(:,7), 'LineWidth',1)
axis([0 inf min(EstadosSim.Data(:,7)) max(EstadosSim.Data(:,7))])
grid on
ylabel('r [rad/s]', 'FontName', 'Calibri', 'FontSize', 12)

subplot(4,1,4)
plot(EstadosSim.Time,EstadosSim.Data(:,8)*R2G, 'LineWidth',1)
axis([0 inf min(EstadosSim.Data(:,8)*R2G) max(EstadosSim.Data(:,8)*R2G)])
grid on
xlabel('t [s]', 'FontName', 'Calibri', 'FontSize', 12)
ylabel('\phi [\circ]', 'Interpreter', 'tex', 'FontName', 'Calibri', 'FontSize', 12)

hold off

% Acc Control

figure

hold on

subplot(3,1,1)
ti=AccControl.Time;
y=AccControl.Data(1,1,:);

MX=max(max(AccControl.Data(1,1:)),max(AccControl.Data(1,2:))); %Dato Máximo de los motores

plot(ti,y(1,:), 'LineWidth',1)
axis([0 inf 0-0.01 MX+0.1])
grid on
ylabel('F_1 [N]', 'FontName', 'Calibri', 'FontSize', 12)

subplot(3,1,2)
y=AccControl.Data(1,2,:);
plot(AccControl.Time,y(1,:), 'LineWidth',1)
axis([0 inf 0-0.01 MX+0.1])
grid on
ylabel('F_2 [N]', 'FontName', 'Calibri', 'FontSize', 12)

subplot(3,1,3)
y=AccControl.Data(1,3:)*R2G ;
plot(AccControl.Time,y(1,:), 'LineWidth',1)
axis([0 inf -90 90])
grid on
xlabel('t [s]', 'FontName', 'Calibri', 'FontSize', 12)
ylabel('\delta [\circ]', 'FontName', 'Calibri', 'FontSize', 12)
```

3.2 Etapa de navegación

Código de la función en Matlab de la etapa de navegación.

```
function [PosRef,Refs,Stop]= ControlNavegacion(Waypoints,PosActual)

%Creamos Constantes

Stop=0;
G2R=pi/180;           %conversiones
R2G=180/pi;

persistent j xE yE zE meta YA YN n u m

if isempty(j)% Inicializamos variables

    [m 1]=size(Waypoints);% Contadores
    j=1;
    xE=0; yE=0; zE=0;

    YA=0; YN=0; n=0; % Control del Yaw...

    YawRef=0; ThetaRef=0; meta=0;
    Refs=[0; 0; 0];
    PosRef=[0; 0; 0];
    u=1;

end

%% Punto Referencia

if (j<m) && (meta==1)           % se alcanza un punto de la ruta
    j=j+1;
    meta=0;
end

if (j==m) && (meta==1)           % Se alcanza el último punto
    Stop=1;
end

x=Waypoints(j,1);
y=Waypoints(j,2);
z=Waypoints(j,3);

PosRef=[x; y; z];
```



```
%% Cálculo Referencias

% Error referencia
xE=PosRef(1,1)-PosActual(1,1);
yE=PosRef(2,1)-PosActual(2,1);
zE=PosRef(3,1)-PosActual(3,1);

%% Yaw Ref

YN=atan2(yE,xE);      % Se calcula el nuevo Yaw

    if      ((YN + (n)*2*pi)- YA) < (-350*G2R)
        n=n+1;
    elseif  ((YN + (n)*2*pi)- YA) > (350*G2R)
        n=n-1;
    end

YN=YN+n*2*pi;        % Se corrige
YA=YN;                % Se guarda el dato para comparar

YawRef=YN;

%% Theta Ref

ThetaRef=atan2(-zE,sqrt(xE^2+yE^2));

    if (abs(zE)<=1)
        ThetaRef=0;
    end

%% Vel Ref

% La velocidad de referencia siempre será 1, el punto de funcionamiento
u=1;

%% meta

if (abs(xE)<1) && (abs(yE)<1) && (abs(zE)<=1)
    meta=1;
else
    meta=0;
end

%% Referencias

Refs=[u; YawRef; ThetaRef];
```