



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

INGENIERÍA ELECTROMECÁNICA

ESPECIALIDAD ELECTRÓNICA

NAVEGACIÓN AUTÓNOMA DE UNA AERONAVE NO TRIPULADA DE ALA FIJA

Autor: Carlos Martín Parages

Directores: Juan Luis Zamora Macho

José Porras Galán

Madrid

Julio 2016

AUTORIZACIÓN PARA LA DIGITALIZACIÓN, DEPÓSITO Y DIVULGACIÓN EN RED DE PROYECTOS FIN DE GRADO, FIN DE MÁSTER, TESINAS O MEMORIAS DE BACHILLERATO

1º. Declaración de la autoría y acreditación de la misma.

El autor D. CARLOS MARTÍN PARAGES DECLARA ser el titular de los derechos de propiedad intelectual de la obra: NAVEGACIÓN AUTÓNOMA DE UNA AERONAVE NO TRIPULADA DE ALA FIJA, que ésta es una obra original, y que ostenta la condición de autor en el sentido que otorga la Ley de Propiedad Intelectual.

2º. Objeto y fines de la cesión.

Con el fin de dar la máxima difusión a la obra citada a través del Repositorio institucional de la Universidad, el autor CEDE a la Universidad Pontificia Comillas, de forma gratuita y no exclusiva, por el máximo plazo legal y con ámbito universal, los derechos de digitalización, de archivo, de reproducción, de distribución y de comunicación pública, incluido el derecho de puesta a disposición electrónica, tal y como se describen en la Ley de Propiedad Intelectual. El derecho de transformación se cede a los únicos efectos de lo dispuesto en la letra a) del apartado siguiente.

3º. Condiciones de la cesión y acceso

Sin perjuicio de la titularidad de la obra, que sigue correspondiendo a su autor, la cesión de derechos contemplada en esta licencia habilita para:

1. Transformarla con el fin de adaptarla a cualquier tecnología que permita incorporarla a internet y hacerla accesible; incorporar metadatos para realizar el registro de la obra e incorporar “marcas de agua” o cualquier otro sistema de seguridad o de protección.
2. Reproducirla en un soporte digital para su incorporación a una base de datos electrónica, incluyendo el derecho de reproducir y almacenar la obra en servidores, a los efectos de garantizar su seguridad, conservación y preservar el formato.
3. Comunicarla, por defecto, a través de un archivo institucional abierto, accesible de modo libre y gratuito a través de internet.
4. Cualquier otra forma de acceso (restringido, embargado, cerrado) deberá solicitarse expresamente y obedecer a causas justificadas.
5. Asignar por defecto a estos trabajos una licencia Creative Commons.
6. Asignar por defecto a estos trabajos un HANDLE (URL *persistente*).

4º. Derechos del autor.

El autor, en tanto que titular de una obra tiene derecho a:

1. Que la Universidad identifique claramente su nombre como autor de la misma
2. Comunicar y dar publicidad a la obra en la versión que ceda y en otras posteriores a través de cualquier medio.
3. Solicitar la retirada de la obra del repositorio por causa justificada.
4. Recibir notificación fehaciente de cualquier reclamación que puedan formular terceras personas en relación con la obra y, en particular, de reclamaciones relativas a los derechos de propiedad intelectual sobre ella.

5º. Deberes del autor.

El autor se compromete a:

1. Garantizar que el compromiso que adquiere mediante el presente escrito no infringe ningún derecho de terceros, ya sean de propiedad industrial, intelectual o cualquier otro.
2. Garantizar que el contenido de las obras no atenta contra los derechos al honor, a la intimidad y a la imagen de terceros.
3. Asumir toda reclamación o responsabilidad, incluyendo las indemnizaciones por daños, que pudieran ejercitarse contra la Universidad por terceros que vieran infringidos sus derechos e intereses a causa de la cesión.
4. Asumir la responsabilidad en el caso de que las instituciones fueran condenadas por infracción de derechos derivada de las obras objeto de la cesión.

6º. Fines y funcionamiento del Repositorio Institucional.

La obra se pondrá a disposición de los usuarios para que hagan de ella un uso justo y respetuoso con los derechos del autor, según lo permitido por la legislación aplicable, y con fines de estudio, investigación, o cualquier otro fin lícito. Con dicha finalidad, la Universidad asume los siguientes deberes y se reserva las siguientes facultades:

- La Universidad informará a los usuarios del archivo sobre los usos permitidos, y no garantiza ni asume responsabilidad alguna por otras formas en que los usuarios hagan un uso posterior de las obras no conforme con la legislación vigente. El uso posterior, más allá de la copia privada, requerirá que se cite la fuente y se reconozca la autoría, que no se obtenga beneficio comercial, y que no se realicen obras derivadas.
- La Universidad no revisará el contenido de las obras, que en todo caso permanecerá bajo la responsabilidad exclusiva del autor y no estará obligada a ejercitar acciones legales en nombre del autor en el supuesto de infracciones a derechos de propiedad intelectual derivados del depósito y archivo de las obras. El autor renuncia a cualquier reclamación frente a la Universidad por las formas no ajustadas a la legislación vigente en que los usuarios hagan uso de las obras.
- La Universidad adoptará las medidas necesarias para la preservación de la obra en un futuro.
- La Universidad se reserva la facultad de retirar la obra, previa notificación al autor, en supuestos suficientemente justificados, o en caso de reclamaciones de terceros.

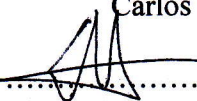
Madrid, a 20 de Julio de 2016

ACEPTA

Fdo.....

Proyecto realizado por el alumno/a:

Carlos Martín Parages

Fdo.:  Fecha: 20./07./2016

Autorizada la entrega del proyecto cuya información no es de carácter confidencial

LOS DIRECTORES DEL PROYECTO

Juan Luis Zamora Macho

Fdo.:  Fecha: 20./07./2016

José Porras Galán

Fdo.:  Fecha: 20./07./2016

Vº Bº del Coordinador de Proyectos

Álvaro Sánchez Miralles

Fdo.: Fecha:/...../.....



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

INGENIERÍA ELECTROMECÁNICA

ESPECIALIDAD ELECTRÓNICA

NAVEGACIÓN AUTÓNOMA DE UNA AERONAVE NO TRIPULADA DE ALA FIJA

Autor: Carlos Martín Parages

Directores: Juan Luis Zamora Macho

José Porras Galán

Madrid

Julio 2016

Navegación autónoma de una aeronave no tripulada de ala fija

Autor: Martín Parages, Carlos

Directores: Zamora Macho, Juan Luis; Porras Galán, José

Entidad Colaboradora: ICAI – Universidad Pontificia de Comillas

INTRODUCCIÓN

En este proyecto se ha diseñado unos controles que permitan a un avión a escala mantener un vuelo estable. El objetivo final de este proyecto es el de crear unos controles que permitan de manera eficaz a una aeronave de ala fija realizar un vuelo estabilizado en exterior.

Para el diseño de controles con buenos resultados era preciso obtener un modelo matemático preciso del avión. Diferentes proyectos desarrollados con anterioridad han obtenido resultados satisfactorios para objetivos similares a los de este proyecto. Estos proyectos utilizaron diferentes métodos a la hora de realizar este modelado del avión, ya fuese considerando el avión como un sólido rígido y mediante la aplicación de las leyes de Newton [1] o mediante la realización de ensayos que permitiesen conocer los cambios en las variables de control según los cambios en los mandos [2]. Debido a la complejidad de la geometría del avión utilizado para este proyecto, se optó por la utilización del segundo método.

El proyecto cuenta con la obtención de controles lo más eficaces posibles basándose para ello en los proyectos realizados con anterioridad sobre el tema, así como los conocimientos del director y autor del proyecto de aeromodelismo y electrónica.

La razón de la existencia de este proyecto es la voluntad de crear un vehículo con la capacidad para el transporte de mercancías a poblaciones de difícil acceso. Este proyecto se centra en el vuelo horizontal de la aeronave, pero sin olvidar que pretende ser dotada

de despegue y aterrizaje vertical que le permitan la toma de tierra en un espacio reducido.

El proceso para desarrollar el proyecto constó de las siguientes partes:

- Preparación previa
- Modelado del avión
- Diseño teórico de controles
- Resultados

PREPARACION PREVIA

Para posibilitar la ejecución del proyecto, se tuvo que preparar el entorno de software, que consta en este caso, de los programas *Matlab* y *Simulink*, así como la instalación de una serie de archivos que permitiesen a estos comunicarse con el microprocesador que se utilizaría en el modelo, el Ardupilot 2.6 [3].

De manera paralela a la instalación del software necesario, se realizó el montaje del avión, por un lado, el ensamblaje del conjunto del modelo y por otro la instalación de los elementos electrónicos. Estos últimos constan, del sistema de propulsión formado por un motor brushless y un ESC, de los servos que controlan las superficies móviles del avión, el receptor y emisora que permiten la comunicación a distancia para el control manual, el conjunto de sensores, que en su mayoría se encuentran en el microprocesador y por último dos baterías LiPo para alimentar el conjunto de los sistemas electrónicos.

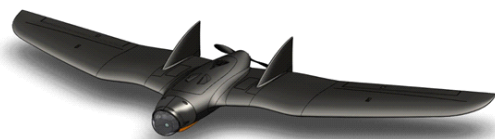


Ilustración 1: Avión utilizado

Como último paso de la preparación a los ensayos se tuvo que tomar la decisión del lugar en el que estos se realizarían, se optó por un campo de vuelo destinado a avionetas por el conocimiento del sitio y facilidad de acceso para el autor del proyecto.

MODELADO DEL AVIÓN

El modelado del avión consiste en la obtención de las ecuaciones que relacionen el mando que recibe el avión desde la emisora con los ángulos de Euler de este. Pese a la existencia de varios sistemas que permiten la obtención de estas funciones, como por ejemplo el uso de ecuaciones en las que se tiene en cuenta todas las características físicas del avión (masa, momentos de inercia, potencia, aerodinámica...), se decidió realizar el modelado del avión mediante unos ensayos. La elección de este método con respecto al expuesto mediante ecuaciones se debe a la complejidad de la aerodinámica del avión.

Los parámetros que se tuvieron en cuenta para el modelado fueron, las medidas de los sensores y los mandos aplicados. Las medidas de los sensores utilizados (giroscopios, acelerómetros, barómetro y medida del tubo de pitot), que una vez transformadas por funciones definidas en *Simulink* proporcionaban, por un lado, los ángulos de Euler y por otro la altura y la velocidad del avión con respecto al aire. Los mandos aplicados consistieron en la suma del mando enviado por el piloto y una señal *PRBS* tanto en la profundidad como en el alabeo, cada uno de estos dos ensayos se realizó por separado. El registro de la velocidad del avión se hizo con intención de realizar un control que a la hora de realizar vuelos automáticos mantuviese una velocidad lo más cercana posible a la de los ensayos.

Dado que el mando de profundidad afecta únicamente al

ángulo de *pitch*, y el de alabeo únicamente al ángulo de *roll*, las funciones de transferencia serían SISO (single input single output), es decir, de una única entrada y una única salida.

Una vez registrados los valores de las variables se obtuvieron las diferentes funciones de transferencia entre, ángulo de *roll* con respecto al mando de alabeo, ángulo de *pitch* con respecto al mando de profundidad y altura con respecto a ángulo de *pitch*.

El procedimiento para obtener las funciones de transferencia consiste en pasar los datos guardados en la memoria del Ardupilot al *Matlab*. Una vez se tienen los datos como variables en *Matlab*, se le aplica la herramienta *Identification Toolbox* del programa. Esta herramienta permite relacionar una entrada y una salida, definidas por el usuario (datos guardados), mediante una función de transferencia, que será la utilizada para el diseño del control correspondiente.

Antes del diseño se debe realizar una comprobación de que la planta estimada es correcta, para ello se compara la salida real (medida), con la que se obtendría aplicándole a la planta estimada la entrada que se midió para esa salida medida. Para ajustar los valores obtenidos de las funciones de transferencias, se calculó el error cuadrático entre las variables anteriores y se ajustó la planta hasta reducirlo al máximo.

DISEÑO TEÓRICO DE CONTROLES

Para cada una de las variables que se deseaban controlar, y conociendo las respectivas plantas, se realizó un control lo más robusto posible manteniendo la máxima simplicidad posible para reducir la potencia de cálculo necesaria.

Los controles que se realizaron fueron por un lado uno en cascada para controlar la altura del vehículo y un

control proporcional integral para el ángulo de *roll*. Para realizar el control en cascada, primero se diseñó un control proporcional para el ángulo de *pitch* y luego un control externo proporcional integral para eliminar el error en régimen permanente a la referencia de altura. Para el diseño del control de *roll* se realizó un único control proporcional integral para evitar errores en régimen permanente.

Todos los controles diseñados se comprobaron realizando simulaciones y en los casos en los que se requiriese se reajustaron tanto como fuese necesario hasta conseguir las simulaciones más favorables.

RESULTADOS

Los controles que se diseñaron de manera teórica se comprobaron realizando unos vuelos en los que en todo momento se tenía la capacidad de volver a tener el control manual del avión. Estos controles se comprobaron primero por separado y después conjuntamente para obtener un vuelo estable sin intervención del piloto. Dado que la obtención de las distintas plantas no era completamente fiable, se realizaron pequeños ajustes sobre los controles comprobados según las sensaciones que se obtuvo a partir del vuelo autónomo con los controles teóricos.

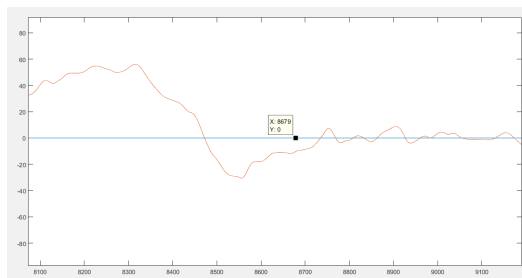


Ilustración 2: Control de *roll*

En la ilustración se muestra el resultado de la activación del control de *roll*.

Los resultados guardados en la memoria muestran que el control de *roll* resultó efectivo, sin embargo, se puede

observar como en algún caso la respuesta era algo oscilante, esto se debe en parte al viento racheado que había el día de la comprobación de los controles, pero también podría ser debido a alguna imprecisión en la obtención de la planta.

En cuanto a los datos obtenidos en la implementación del control de altura, muestran que el control resultaba, si no erróneo, demasiado lento de manera que la comprobación en vuelo resultaba complicada ya que se perdía de vista antes de alcanzar la altura de referencia.

CONCLUSIONES

A la vista de los resultados obtenidos podemos afirmar que el procedimiento para la obtención de la planta del avión, así como el del diseño de los controles son válidos, si bien se considera que pueden ser mejorados. Pese a no haber realizado los objetivos de incluir un sistema GPS que permitiese al vehículo realizar una ruta programada, se consiguió obtener un vehículo con la capacidad de mantener un vuelo controlado manteniendo el rumbo y altura. Es por esto que se puede considerar el proyecto como un éxito, pero siempre pensando en que es posible realizar mejoras y aportes.

Los aportes que se consideran realizables e importantes para la evolución de este proyecto sería la incorporación del dispositivo GPS y la modificación estructural del avión para que obtuviese la capacidad de volar no solo horizontalmente, pero también vertical de manera que el espacio requerido para su despegue y aterrizaje fuesen mínimos. Estos aportes permitirían que el UAV de este proyecto tuviese cualidades que lo convertirían en un vehículo idóneo para actividades de reparto, vigilancia o incluso para misiones de ayuda humanitaria.

REFERENCIAS

- [A. D. M. L. Tadeo Espinoza,
1] «Linear and nonlinear controllers
applied to fixed-wing UAV,»
Instituto Tecnológico de la Laguna,
2012.
- [A. Noth, «Dynamic Modeling of
2] Fixed-Wing UAV,» Swiss Federal
Institute of Technology Zurich,
2006.
- [F. Hugon, «How to Guide:
3] ArduPilot 2.0 Simulink Blockset,»
Embry-Riddle Aeronautical
University, 2012, 2012.

Control system design of a fixed wing UAV for autonomous outdoors flights

Author: Martín Parages, Carlos

Directors: Zamora Macho, Juan Luis; Porras Galán, José

Colaborator Entity: ICAI – Universidad Pontificia de Comillas

INTRODUCTION

In this project some controls systems have been design to allow a scale airplane to have a stable flight. The ultimate goal of this project is for a fixed-wing aircraft to be able to make a stable flight outdoors.

To make the theoretical design of the controls it was necessary to obtain a precise mathematical model of the plane. Different projects have been developed around this subject and different ways have been used to obtain it. Those methods were either considering the aircraft as a rigid body and applying the Newton's laws [1] or by performing several tests to obtain how variables of the airplane were affected by the manipulated variable [2]. Due to the complexity of the first method if applied to the geometry of the plane used in this project, it was decided to use the second method.

This project has been carried out to obtain controls as reliable as possible basing in previous projects performed about this subject as well as the project director and author on aeromodelling and control systems.

The reason of the existence of this project is the will to create a vehicle with the skill to transport freights to towns with difficult access. This project is centred in the horizontal flight of the aircraft while other parallels projects are focussed on the ability of the aircraft to take off and land vertically.

The process of developing this project, features the following parts:

- Previous preparation
- Modelling of the plane
- Control systems design
- Results

PREVIOUS PREPARATION

To make it possible to carry out this project, it was necessary to prepare the software, which in this case was the installation of *Matlab* and Simulink, as well as the installation of a number of files to allow the communication with microprocessor used in the model, the ArduPilot 2.6 [3].

As well as the installation of all the software needed, it was performed the mounting of the airplane, first the assembly of the model and then the installation of the electronic elements. Those electronic elements consist in the propeller system formed by a brushless engine and ESC, servos to control movable surfaces, receiver and emitter to permit the communication from distance between pilot and aircraft in manual control, all the sensors, which most are in the microprocessor and finally two Lipo batteries to feed all systems.

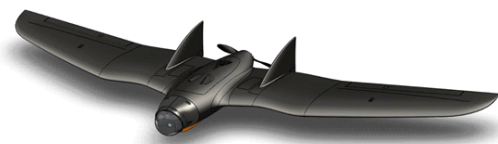


Ilustración 3: Plane used

As last step to prepare the project it was necessary to decide where to perform all tests, it was chosen to use a flying field destined to full scale planes due to the knowledge and its facility for the author of the project to access it.

MODELLING OF THE PLANE

The modelling of any vehicle consists on the obtaining of the equations which relate the manipulated variables with Euler's angles of the plane. Even if there are several ways to obtain those functions, as the one which considers all the characteristics physics (mass, inertia moments, power, aerodynamics...), it was decided to obtain them by performing tests. That decision was made because of the complexity of the aerodynamics of that plane.

Parameters considered in the modelling were, the measurements of sensors and manipulated variables. The measurements of the sensors (gyroscope, accelerometers, barometer and measured of pitot tube), once transformed by *Simulink* functions provided Euler's angles, altitude and speed of the plane relative to the air. Manipulated variables applied consisted on a PRBS signal summed to the received signal both in depth and ailerons in separated tests. All variables were recorded in the flash memory. Airplane speed was recorded to make a control system so the plane worked at the same speed during tests and with automatic controls.

As the depth manipulated variable affects just the *pitch* angle, and aileron just *roll* angle, functions used were SISO (single input, single output).

Once registered all values of the variables during the tests, transfer functions were obtained between *roll* angle and ailerons, *pitch* angle and depth, and finally height with *pitch* angle.

The procedure to obtain those transfer functions consists on taking the values recorded to the *Matlab* workspace, and applying the *Identification Toolbox* to them, organising variables as explained before. This tool allows to relate an input with an output, defined by the user

(saved data), with a transfer function, that will be later used to design the corresponding control systems.

Before designing, all transfer functions were checked to be sure they were correct, to make that, a *Simulink* model was created to calculate the square error between the inputs applied to the transfer functions obtained and the measured outputs and values in the transfer functions were modified to decrease that error to minimum.

CONTROL SYSTEM DESIGN

For each variable willing to control, and knowing the respective plant, a control system was made, as robust as possible maintaining maximum simplicity to reduce the power needed to calculation.

Controls designed were, for altitude a cascade control and a proportional integral (PI) for *roll* angle. To perform the cascade control, first a proportional (P) control was created for *pitch* angle and then a PI to remove permanent regime error. To design the *roll* control, a PI was created to remove as well that error.

All controls designed were checked by simulations and in some cases some adjusts were made until those results were satisfying.

RESULTS

All controls designed theoretically were checked performing flights in which the pilot had always the choice to disable the automatic controls prioritizing security. Each control was checked first individually, then all together to obtain a stable without the pilot's intervention. Due to the not exact plant obtained, some adjusts were made in the values of the controls until satisfactory results.

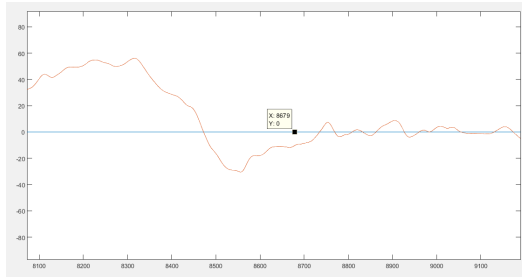


Ilustración 4: Roll control

The previous picture shows the activation of the *roll* control.

All results were recorded on the flash memory and show how the *roll* control is quite effective, even if in some cases the results were a little oscillating, that may be due to gusty wind on the day of the checking or to an imprecision of the obtained plant.

Results of the height control show that if maybe not wrong, it was too slow to check without any risk, and making it faster would probably make it too oscillating which, in this control was not desirable.

CONCLUSIONS

In view of the results, we can say that the procedure for obtaining the aircraft plant, as well as the design controls are valid, even though it is considered that they may be improved. Despite not having made the objectives such as the GPS system allowing the vehicle to carry out a scheduled route, a vehicle with the ability to maintain a controlled flight while keeping height,

direction and speed was made. This is why the project is considered as a success, but as improvable.

The contributions that may be adequate to this project may be the incorporation of GPS device and aircraft structural modification so that it would have the ability to fly, not only horizontally, but also vertically allowing it to take off and land in minimal surface. These contributions would allow this project to have qualities that would turn it into a vehicle suitable for distribution activities, surveillance or even humanitarian missions.

REFERENCES

- [A. D. M. L. Tadeo Espinoza,
1] «Linear and nonlinear controllers applied to fixed-wing UAV,» Instituto Tecnológico de la Laguna, 2012.
- [A. Noth, «Dynamic Modeling of
2] Fixed-Wing UAV,» Swiss Federal Institute of Technology Zurich, 2006.
- [F. Hugon, «How to Guide:
3] ArduPilot 2.0 Simulink Blockset,» Embry-Riddle Aeronautical University, 2012, 2012.



Índice de la memoria

Parte I	Memoria.....	5
Capítulo 1	Introducción	7
1.1	Estado de la cuestión	7
1.2	Motivación del proyecto.....	10
1.3	Objetivos del proyecto.....	11
1.4	Metodología / Solución desarrollada	11
1.5	Recursos / Herramientas empleadas.....	12
Capítulo 2	Preparación previa	13
2.1	Software.....	13
2.2	Hardware	14
2.2.1	Montaje	14
2.2.2	Vuelo manual	16
2.2.3	Vuelo autónomo y ensayos	16
2.3	Lugar de ensayos	17
Capítulo 3	Modelado del avión.....	19
3.1	Procedimiento	19
3.2	Preparación.....	21
3.2.1	Lectura y escritura.....	21
3.2.2	Señal PRBS.....	23
3.2.3	Mezcla alerones para la cola en V.....	25
3.2.4	Sensores	26



3.3	Ensayos.....	28
3.3.1	Tubo de Pitot.....	29
3.3.2	Memoria flash y vuelos.....	33
3.4	Obtención de la planta	35
Capítulo 4	<i>Diseño teórico de los controles.....</i>	39
4.1	Diseño	39
4.1.1	Control en Cascada	39
4.1.1.1	Control interno: Pitch	39
4.1.1.2	Control externo: Altura	40
4.1.2	Control Roll	41
4.2	Simulación.....	42
4.2.1	Simulación control de altura	42
4.2.1.1	Escalón en la referencia.....	42
4.2.1.2	Tren de escalones como perturbación	43
4.2.2	Simulación control de roll.....	44
Capítulo 5	<i>Comprobación de los controles diseñados.....</i>	47
5.1	Implementación de los controles	47
5.1.1	Comprobación de los controles	47
5.2	Resultados	48
5.2.1	Control de altura.....	49
5.2.2	Control de alabeo	50
5.2.3	Control completo.....	51
Capítulo 6	<i>Conclusiones.....</i>	53
Capítulo 7	<i>Futuros desarrollos</i>	55
Bibliografía	57	
Parte II	<i>Estudio económico.....</i>	59
Capítulo 1	<i>Estudio económico.....</i>	61
Parte III	<i>Manual de usuario</i>	63
Capítulo 1	<i>Vuelo autónomo.....</i>	65



Parte IV	Código fuente.....	67
Capítulo 1	Ficheros de Matlab.....	69
1.1	Sensor_fusion.....	69
1.2	Filtro	72
1.3	Funciones memoria flash	72
1.4	Función Pitot.....	76
1.5	Control velocidad	77
1.6	Control de altura	77
1.7	Control de roll.....	78
1.8	Fichero Maestro.....	78
1.9	Diagramas de Fourier	80
Capítulo 2	Ficheros de Simulink	81
2.1	Ficheros	81
2.1.1	Ensayo profundidad	81
2.1.2	Ensayo alabeo	82
2.1.3	Control altura	82
2.1.4	Control alabeo	83
2.1.5	Control completo.....	83
2.2	Subsistemas	84
2.2.1	Lectura y escritura de canales	84
2.2.2	IMU.....	85
2.2.3	Altura barómetro	85
2.2.4	Reloj.....	85
2.2.5	Ángulos avión	86
2.2.6	Mezcla alerones.....	87
2.2.7	Pitot.....	87
2.2.8	Activación profundidad/alabeo	87
2.2.9	Memoria flash	88
2.2.10	Referencia de altura.....	89
2.2.11	Referencia de alabeo	89
2.2.12	Control velocidad.....	90



2.2.13 Control altura	90
2.2.14 Control alabeo	90
Parte V Planos.....	91
Capítulo 1 Centro de gravedad.....	93
Capítulo 2 Vista tridimensional.....	95
Capítulo 3 Pieza hélice	97
Parte VI Presupuesto.....	99
Capítulo 1 Componentes principales	101
Capítulo 2 Equipo y herramientas	103
Capítulo 3 Software	105
Capítulo 4 Mano de obra y desplazamientos	107
Capítulo 5 Total	109
Parte VII Datasheets.....	111
Capítulo 1 Avión	113
Capítulo 2 Emisora.....	115
Capítulo 3 Receptor	117
Capítulo 4 Motor.....	119
Capítulo 5 ESC	121
Capítulo 6 Placa de presión para pitot.....	123
Capítulo 7 APM 2.6	125



Índice de figuras

Ilustración 1: Ecuaciones de cabeceo	8
Ilustración 2: Ecuaciones de alabeo	8
Ilustración 3: Avión utilizado para la realización del proyecto.....	10
Ilustración 4: Antes y después de introducir la modificación en la hélice	15
Ilustración 5: Campo de vuelo.....	18
Ilustración 6: Señales PWM.....	22
Ilustración 7: Bloque de lectura.....	22
Ilustración 8: Bloque de escritura.....	23
Ilustración 9: Señal de mando en ensayos.....	24
Ilustración 10: Diagrama de Fourier de la señal PRBS.....	25
Ilustración 11: Mezcla alerones para cola en V	26
Ilustración 12: IMU acelerómetros y giroscopios	27
Ilustración 13: Función filtro de Kalman	27
Ilustración 14: Altura, sensor barométrico	28
Ilustración 15: Ensayo tubo de pitot.....	29
Ilustración 16: Lectura tubo de pitot	30
Ilustración 17: Gráfica de velocidad del pitot	32
Ilustración 18: Velocidad del avión.....	33
Ilustración 19: Memoria flash	34
Ilustración 20: Montaje del control de velocidad.....	35
Ilustración 21: Simulink para calcular el error cuadrático de la función de transferencia	37



Ilustración 22: Control de altura en cascada	40
Ilustración 23: Control de Roll PI	41
Ilustración 24: Respuesta a un escalón en la referencia del control de altura	43
Ilustración 25: Esquema para una perturbación en el ángulo en el control de altura	43
Ilustración 26: Respuesta a un tren de escalones en la perturbación del control de altura	44
Ilustración 27: Respuesta a un tren de escalones a modo de perturbación en el control de roll	45
Ilustración 28: Implementación del control de altura.....	47
Ilustración 29: Implementación del control de alabeo	48
Ilustración 30: Resultado ensayo control de altura	49
Ilustración 31: Resultado ensayo control de alabeo	50
Ilustración 32: Resultado ensayo control completo	51
Ilustración 33: Diagrama de Bode del filtro	72
Ilustración 34: Simulink del ensayo de profundidad.....	81
Ilustración 35: Simulink del ensayo de alabeo	82
Ilustración 36: Simulink del control de altura	82
Ilustración 37: Simulink del control de alabeo.....	83
Ilustración 38: Simulink del control completo	83
Ilustración 39: Subsistema lectura y escritura	84
Ilustración 40: Subsistema IMU	85
Ilustración 41: Subsistema altura barómetro	85
Ilustración 42: Subsistema reloj	85
Ilustración 43: Subsistema ángulos avión	86
Ilustración 44: Subsistema mezcla alerones	87
Ilustración 45: Subsistema pitot	87



Ilustración 46: Subsistema activación profundidad.....	88
Ilustración 47: Subsistema memoria flash.....	88
Ilustración 48: Subsistema referencia de altura.....	89
Ilustración 49: Subsistema referencia de alabeo	89
Ilustración 50: Subsistema control velocidad.....	90
Ilustración 51: Subsistema control altura	90
Ilustración 52: Subsistema control alabeo.....	90

Índice de tablas

Tabla 1: Resultados del ensayo del tubo de Pitot.....	31
Tabla 2: Presupuesto Componentes principales	101
Tabla 3: Presupuesto Equipo y herramientas	103
Tabla 4: Presupuesto Software	105
Tabla 5: Presupuesto Mano de obra y desplazamientos.....	107

Índice de planos

Plano 1: Centro de gravedad del avión	93
Plano 2: Vista tridimensional del avión	95
Plano 3: Pieza hélice	97





Parte I MEMORIA





Capítulo 1 INTRODUCCIÓN

La electrónica es una de las ramas de la ingeniería que más evoluciona hoy en día. Esta evolución se debe en gran parte al desarrollo que existe alrededor de los microprocesadores, que cada vez son más pequeños y potentes. Para los sistemas de control estas mejoras han supuesto un gran avance, ya que permiten controlar sistemas que por razones de potencia, peso o tamaño antes habría resultado imposible. En todos los aspectos, estas han sido unas mejoras importantes, pero en el ámbito del transporte aéreo, la reducción de peso y tamaño ha sido decisiva, para tener mejor rendimiento y así optimizar la autonomía. Esto ha permitido que los vehículos aéreos no tripulados (UAV) sean una realidad.

Existen muchos transportes en los que el proyecto podía realizarse, ya que cualquier vehículo es susceptible de implantarse un control de pilotaje, pero en cuanto a los aéreos, a pesar de la existencia de otras alternativas, los multicopteros y aviones son los que aportan mayores ventajas. Cada una de estas opciones tiene cualidades que le faltan a la otra, mientras que los multicopteros tienen la capacidad de vuelo estático y despegue vertical, los aviones tienen la ventaja de poder recorrer distancias mayores. Es por esto que se ha optado por desarrollar el sistema de control de vuelo horizontal de un avión con el objetivo futuro de implantarle la opción de despegue vertical, para obtener un vehículo con capacidad de despegue vertical y capaz de recorrer grandes distancias.

1.1 ESTADO DE LA CUESTIÓN

El problema que se pretende resolver en el proyecto es como desarrollar un sistema de vuelo autónomo para un avión. A la hora de modelar un UAV, se requiere definir dos sistemas de ejes, uno fijo a la tierra y otro solidario al UAV, para así tener su posición y orientación.

Una simplificación que se puede hacer a la hora de sacar las ecuaciones de un UAV de ala fija a esta escala es, como se dice en el artículo “*Linear and Nonlinear Controllers Applied to Fixed-Wing UAV*” [1], obviar la flexibilidad de la estructura, considerando el UAV como un cuerpo rígido, así como no considerar la curvatura de la tierra.

A la hora de realizar el modelado del avión generalmente existen tres ejes donde estudiar las ecuaciones dinámicas, respecto a los tres ejes solidarios al avión. En el caso del avión que se empleará debido a su configuración, carece de la posibilidad de giro respecto al eje solidario z. Los parámetros necesarios para el estudio de las ecuaciones dinámicas de cada uno de los otros ejes se muestran en las figuras siguientes junto a las ecuaciones resultantes de la aplicación de las leyes de Newton.

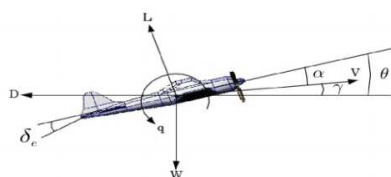


Ilustración 1: Ecuaciones de cabeceo

$$\dot{V} = \frac{1}{m}(-D + T \cos \alpha - mg \sin \gamma)$$

$$\dot{\gamma} = \frac{1}{mV}(L + T \sin \alpha - mg \sin \gamma)$$

$$\dot{\theta} = q$$

$$\dot{q} = \frac{M}{I_{yy}}$$

$$\dot{h} = V \sin \gamma$$

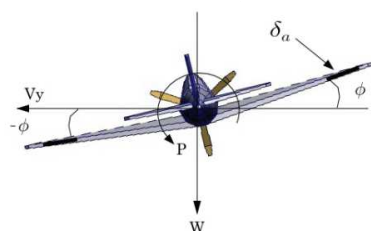


Ilustración 2: Ecuaciones de alabeo

$$\dot{p} = \frac{L}{I_{xx}}$$

$$\dot{\phi} = p$$

$$\dot{V}_y = \frac{F_y}{m} + pV_x$$

$$\dot{V}_x = \frac{F_x}{m} - pV_y$$

Existen dos tipos de UAV, aquellos que son controlados de manera remota, y aquellos que vuelan de manera autónoma con un plan de vuelo programado previamente.

Existen diferentes áreas en los que se usan estas modalidades de UAV, sobre todo en ámbitos en los que por la existencia de riesgos se prefiere la ausencia de



personas. A continuación, se presentan diferentes ámbitos en los que los UAV son utilizados de manera habitual.

Hoy en día se utilizan en el ámbito militar tanto para misiones de reconocimiento como de ataque. El funcionamiento de estos UAVs puede basarse por un lado en sensores y cámaras que permiten situar de manera precisa el dron para el control manual desde la base, o la utilización de sistemas GPS para la localización, junto con sensores que permitan hacer redundantes la mayor parte de los sistemas, y seguimiento de la ruta programada. La primera vez que se usó un UAV para uso militar se dio a durante la segunda guerra mundial para en entrenamiento de los operarios de cañones antiaéreos.

Los UAVs también son utilizados con frecuencia en actividades de vigilancia, para prevención de incendios o vigilancia de zonas conflictivas. Estos drones suelen realizar vuelos programados y estén dotados de cámaras o sensores de distinto tipo, según lo que se desea comprobar, que transmiten al centro de control. La utilización de estos UAVs supone, con una reducción notable de personal, tener un radio de acción mucho mayor.

En el ámbito del ocio se ha comenzado desde hace unos pocos años a implementar sistemas de navegación autónomos para eliminar riesgos en caso de pérdida de vista del avión, en cuyo caso se requiere únicamente activar el control y el avión volverá a una distancia programada del receptor. Cabe destacar en este ámbito el sistema de eagletreesystem [6], que requiere únicamente que se ajusten unas ganancias de los controles que incluye.

Recientemente se han comenzado a hablar del uso de UAVs para el reparto de mercancías. Es el caso de Amazon que desde hace pocos años trabaja en que esto sea posible, a pesar de todo, todavía no se han puesto en marcha debido a problemas burocráticos. Pese a que no se estén utilizando aún, cabe esperar que en pocos años esto se convierta en una realidad y que aumenten su radio de acción, actualmente estimado en 16km para una carga de 2.5kg.

La particularidad del modelo que se utilizará en este proyecto es que se trata de un avión al que se le realizarán en un futuro una serie de modificaciones para



permitir que tenga la capacidad de despegue vertical. Este proyecto se centrará en la parte de vuelo horizontal de este vehículo.

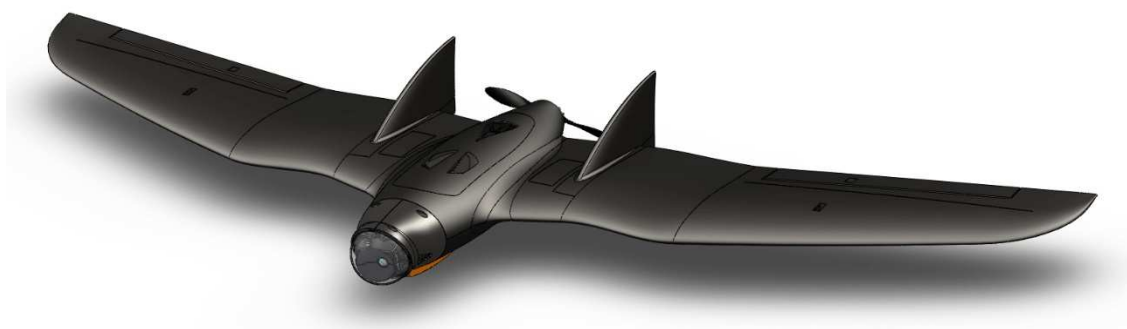


Ilustración 3: Avión utilizado para la realización del proyecto

1.2 MOTIVACIÓN DEL PROYECTO

A pesar de que el proyecto consiste en el vuelo autónomo de la aeronave, tiene como peculiaridad el ser parte de un proyecto mayor con objetivo el de una aeronave de despegue y aterrizaje vertical y vuelo horizontal que la dotaría de las cualidades no solo de los aviones, pero también de las de los multicópteros. Estas ventajas harían de este UAV un vehículo con grandes ventajas frente a otros en muchos de los ámbitos en los que ahora se utilizan.

Este UAV proporciona, por un lado, al tener vuelo horizontal, un alcance mucho mayor del que se obtiene actualmente con multicópteros y por otro, el aterrizaje vertical permitiría su llegada a lugares de difícil acceso. Como ejemplo de uso, el reparto de mercancías podría ser mucho más eficaz gracias a dotarlo de vuelo horizontal, y puede ser de especial importancia a la hora de realizar misiones de ayuda humanitaria a zonas en las que, por geografía o desastres naturales, las condiciones de acceso sean difíciles. En otros ámbitos como el militar, el despegue



vertical permitiría el despliegue de los UAV desde lugares sin pista de aterrizaje, sin disminuir el radio de alcance de los UAVs utilizados actualmente.

1.3 OBJETIVOS DEL PROYECTO

El proyecto puede dividirse en diferentes objetivos que según la disponibilidad de tiempo podrán realizarse en mayor o menor medida.

Aquellos objetivos que se considera que han de ser cumplidos son los siguientes:

- Obtención del modelo físico del avión,
- Diseño e implantación del control,
- Vuelo completamente autónomo.

Por otro lado, si el tiempo lo permite y el resultado de los objetivos anteriores es favorable, sería interesante la obtención de los siguientes objetivos secundarios:

- Despegue horizontal del avión,
- Aterrizaje horizontal del avión.

1.4 METODOLOGÍA / SOLUCIÓN DESARROLLADA

A continuación, se explica cómo se pretende realizar y las tareas que se pretenden llevar a cabo.

- Dejar el avión listo para volar, así como la electrónica instalada para realizar el registro de datos;
- Instalación de todo el software necesario para el procesamiento de los datos, y comprobación del buen funcionamiento de los mismos;
- Obtención de los datos necesarios para conseguir el modelo físico del avión, realizando vuelos controlados por radio;



- Obtención del modelo físico del avión a partir de los ensayos realizados previamente;
- Diseño en Matlab de los controles teóricos para el vuelo autónomo;
- Comprobación de los controles diseñados previamente y realización de los ajustes necesarios para su buen funcionamiento;
- Comienzo de la redacción de la memoria;
- Redacción de la memoria.

1.5 RECURSOS / HERRAMIENTAS EMPLEADAS

Los recursos que se utilizarán en este proyecto son los siguientes:

- Hardware:
 - o Avión: De ala fija con motor de propulsión. Modelo RV Jet de Range Video,
 - o Emisora turnigy TGY-i10 y receptor,
 - o Motor brushless y ESC,
 - o Batería 8000 mAh 3S,
 - o Batería 950 mAh 2S,
 - o Conector USB-micro USB,
 - o UBEC hextronic,
 - o APM: Dispositivo microcontrolador para procesamiento de los controles y recopilación de datos para el diseño del modelo físico del avión.
- Software:
 - o Matlab: El diseño de los controles se realizará en este programa,
 - o Simulink: Se utilizará para la obtención del modelo físico del avión, así como para la comprobación teórica de los controles.



Capítulo 2 PREPARACIÓN PREVIA

2.1 SOFTWARE

La comunicación y programación de los distintos sistemas electrónicos que iban a ser utilizados a lo largo del proyecto requería la instalación de ciertas herramientas informáticas. A pesar de existir diferentes posibilidades en cuanto a la elección de estas, se optó, por tratarse de unos programas conocidos previamente, por la utilización de Matlab y Simulink. De las distintas versiones que existen del Matlab se decidió utilizar la más actual que permitiese el uso de las librerías creadas para el uso de APM, la versión que abarcaba estas características resultó ser la 2015a, que por otro lado había sido utilizada con anterioridad y había demostrado ser completamente funcional.

El microprocesador utilizado a lo largo del proyecto es el ArduPilot Mega 2.6, un microcontrolador diseñado por Arduino enfocado al uso en aeronaves. Existe un conjunto de bloques diseñados para Simulink, que permiten acceder a diferentes funciones del APM, se decidió instalarlo para facilitar la comunicación con el microprocesador. La instalación del blockset requiere seguir una serie de pasos que especifican en la guía, “How To Guide: ArduPilot 2.0 Simulink Blockset”[3]. Una vez realizados los pasos expuestos en la guía se tiene acceso a los bloques preparados para el uso del APM en la librería de Simulink.

El uso del bloque de Simulink destinado al registro de datos en la memoria flash del microprocesador, requería la utilización de un programa que permitiese por un lado el borrado de la memoria, dar la orden de iniciar el registro de datos y el acceso a estos. Se barajó el utilizar para ello el programa Putty, pero este presentaba el problema de resultar fácil el paso de los datos a Matlab, el programa principal de desarrollo del proyecto. Este problema había sido solventado con anterioridad en el proyecto “control de un cuadricóptero para vuelos autónomos en interiores”[4], mediante el uso de las funciones *ByA* y *ArduData*, creadas en



Matlab, que permiten borrar y comenzar un nuevo registro de datos, y acceder a los datos guardados en la memoria.

2.2 *HARDWARE*

La instalación del hardware se dividió en varias etapas, en un primer lugar se realizó el montaje del avión, y más adelante, la instalación de la electrónica en él, primera aquella necesaria para un vuelo manual, y se fue añadiendo todo aquello que fuese necesario para el vuelo autónomo del avión.

2.2.1 MONTAJE

Pese a tratarse de un avión comprado con la categoría ARF (Almost Ready to Fly), este requería bastantes horas en cuanto al montaje. Todo el montaje se realizó siguiendo las instrucciones proporcionadas por el fabricante, y solo se realizaron unas pequeñas modificaciones con respecto a las indicaciones de este.

El motor, hélice y buje no se adquirió de la misma casa del avión y, al no estar preparadas para ser montadas en un avión de propulsión como es el que se utiliza en este proyecto, esto planteaba dos problemas. El primero suponía que la hélice se plegara demasiado sobre el buje, esto hacía que el centro de gravedad de una de las palas pasase del otro lado del eje de rotación, de manera que, al comenzar a girar el motor, esta no se desplegaba y además de no proporcionar el empuje deseado, hacía que el avión comenzase a vibrar. Esto se resolvió añadiendo una placa metálica que impidiese que las palas se replegasen completamente (Ilustración 4). El segundo problema implicaba que al aumentar repentinamente la velocidad del motor la hélice al desplegarse avanzase demasiado golpeando el fuselaje del avión, este problema habría podido solucionarse con otra placa metálica que impidiese ese movimiento de las palas, pero en este caso, conociendo el



problema, se optó por al realizar los vuelos no aumentar de manera súbita la velocidad.



Ilustración 4: Antes y después de introducir la modificación en la hélice



2.2.2 VUELO MANUAL

Para comprobar que el montaje del avión era correcto se realizó un vuelo completamente manual y sin incluir aquello que no fuese necesario para este. En este vuelo la electrónica del avión era únicamente el receptor, el ESC (electronic speed controller), motor, servos y batería. Las conexiones se realizaron de la manera habitual, la batería conectada al ESC y este al motor y receptor, los servos igualmente conectados al receptor. Por la configuración de cola en V, era necesario configurar la emisora para que realizase la mezcla de alerones para este tipo de avión.

La configuración de la emisora para avión de cola en V se realiza desde el menú de la emisora accediendo a la pestaña *Airplane structure* y seleccionando las opciones *Engine*, *Elevator*, *Rudder*, y dentro de la opción *Rudder* seleccionando *V-tail*. Esta opción realiza la mezcla necesaria para mover los alerones de manera correcta, dado que, por la configuración del avión, los dos canales que permiten orientar el avión están acoplados, dos servos son accionados para los dos mandos, la profundidad y el alabeo.

2.2.3 VUELO AUTÓNOMO Y ENSAYOS

Para el montaje del modo autónomo, así como para los vuelos de ensayos, las conexiones que se realizaron consistieron por un lado en conectar cada uno de los canales del receptor a los inputs del APM utilizando para ello cables hembra-hembra de servos. Los alerones se conectaron a los outputs 1 y 2 del APM, reservando el 1 para el izquierdo y el 2 para el derecho. El ESC se conectó a la batería, al motor y al output 3 del APM.

Para los ensayos y vuelos autónomos, se realizó la mezcla de los alerones dentro del APM por lo que se configuró la emisora para un avión convencional, con



únicamente profundidad y dirección. Esta configuración de la emisora se hizo en la pestaña *Airplane structure*, seleccionando las opciones *Engine*, *Rudder* y *Elevator*.

Para la realización de los ensayos, fue necesario habilitar canales suplementarios que activasen o desactivasen funciones programadas en el APM. Esto se configura en la emisora en la pestaña *Auxiliary channels*, eligiendo para cada una de las funciones programadas el que resultase más cómodo. Para los ensayos se añadieron 2 canales suplementarios, para activar las señales en los mandos de profundidad y de alabeo. Para el vuelo autónomo, se incluyó únicamente un canal suplementario que activa y desactiva el vuelo manual.

2.3 LUGAR DE ENSAYOS

El tamaño del avión requería la selección de un lugar de ensayos apropiado. Este debía ser al aire libre, muy abierto y preferiblemente conocido por ensayista. Por estos motivos se optó por un aeródromo que cuenta con varias pistas diseñadas para aviones de gran envergadura. El aeródromo que se utilizó se encuentra en Segovia, y es el utilizado por el 'Club Loreto de vuelo sin motor', consta con 3 pistas, dos de ellas de hierba y una de asfalto. Se utilizaron para no dañar el patín del avión las de hierba, utilizando aquella que resultase mejor según la dirección del viento del día del ensayo.



Ilustración 5: Campo de vuelo



Capítulo 3 MODELADO DEL AVIÓN

Existen diversos métodos de obtención del modelo físico de un vehículo, ya sea mediante la resolución de aquellas ecuaciones que expliquen el comportamiento de este con respecto a aquellos parámetros que puedan afectar, o la obtención empírica del comportamiento del vehículo mediante el registro de variables en ensayos controlados. En este proyecto se decidió el uso de este segundo método por ser más sencillo que la resolución matemática de las ecuaciones de un avión.

3.1 PROCEDIMIENTO

El método que se usó para realizar el modelado del avión consistió en registrar ciertas señales medidas por los sensores cuando se le aplicaba una determinada señal en cada uno de los mandos. Se realizaron dos ensayos diferentes, uno para cada uno de los mandos que se pretenden controlar, el roll y el pitch.

Para evitar trabajar en una sola frecuencia, se requería obtener la respuesta del avión para distintas frecuencias introducidas en el mando. La obtención de la respuesta para distintas frecuencias se podría haber obtenido de varias maneras, por un lado, mediante la repetición de ensayos en los que se variase la frecuencia de la señal introducida al mando, o mediante la generación de una señal con gran cantidad de armónicos no despreciables. Finalmente se optó por realizar un solo ensayo para cada mando utilizando para ello una señal PRBS (Pseudo Random Binary Sequence), se trata de una señal que varía entre su ancho de pulso de manera que el valor de la amplitud de los armónicos de la señal sean lo más parecida posible.

Las medidas que se registraron para cada uno de los ensayos fueron las siguientes:



-clk: mide el tiempo de registro, esta medida se incluyó para controlar por un lado que la memoria no se hubiese agotado durante el vuelo, y por otro, para la representación de correcta en función del tiempo.

-Altura: mide la altura del avión, esta medida se toma desde el barómetro que incluye el APM, se trata de uno de los parámetros que se pretende controlar por lo que la variación de este con respecto a los distintos mandos.

-Roll: ángulo de balanceo, este es otro parámetro que se pretende controlar, dado que se intentará que el avión esté en la medida de lo posible con las alas horizontales.

-Pitch: ángulo de cabeceo, este otro ángulo mide la inclinación del fuselaje hacia arriba y abajo, de nuevo se trata de un parámetro que se pretende controlar para mantener horizontal el avión.

-Pitot: mide la velocidad del avión relativa a la masa de aire en la que se encuentra. Esta se utilizó para poder controlar la velocidad del avión.

Las medidas del Roll y del Pitch se obtuvieron mediante las medidas del acelerómetro y giroscopios que incluye el APM, utilizando la función *Sensor_fusion*, que utiliza una serie de filtros de Kalman para estimar los ángulos de Euler del avión.

Además de las medidas obtenidas por sensores, también se registró durante los ensayos en la memoria flash las siguientes señales:

-ProfundidadEnsayos: se trata de la suma de la señal enviada manualmente de mando de profundidad y la señal PRBS creada para el ensayo.

-AlabeoEnsayos: esta señal es la suma de la señal enviada manualmente de mando de alabeo y la señal PRBS creada para el ensayo.

-EmisoraSelectProf: es la señal que indica cuando se comenzó el ensayo de profundidad, cuando empieza a actuar la señal PRBS en el mando de profundidad.

-EmisoraSelectAla: es la señal que indica cuando comenzó el ensayo de alabeo, cuando empieza a actuar la señal PRBS en el mando de alabeo.



-EmisoraMotor: es la señal recibida por el receptor que regula la velocidad de giro del motor.

Tanto la señal guardada de ProfundidadEnsayos como la de AlabeoEnsayos son la suma de las señales PRBS y los mandos enviados manualmente, con esto se pretendía poder ver el momento en que a la señal PRBS actúa sola para tomar únicamente los valores que resultan útiles.

3.2 PREPARACIÓN

La preparación para la realización de los ensayos se hizo en el entorno Simulink. Para ello se utilizó el conjunto de bloques para Ardupilot o se crearon los montajes necesarios para el tratamiento de las señales.

3.2.1 LECTURA Y ESCRITURA

Para realizar cualquier acción sobre el avión desde tierra se requería del conjunto emisora-receptor, a la hora de realizar los ensayos, las señales recibidas por el receptor deben pasar por el Ardupilot para tratarlas, es por ello que se necesita conectarlos. Dado que las señales con las que trabajan el receptor y los servos son pulsos de valor 5V y ancho variable entre 1000 y 2000 μ s, tratar las señales resulta complicado.

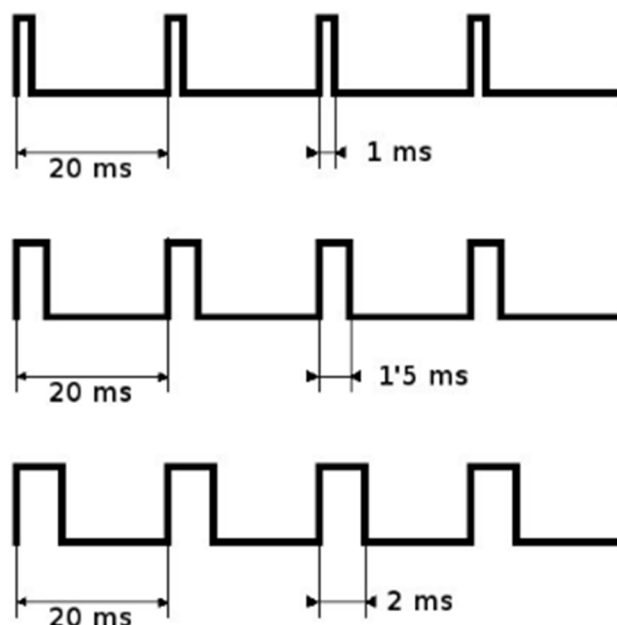


Ilustración 6: Señales PWM

La librería APM Simulink Blockset [3] proporciona un bloque que transforma la señal anterior en un valor continuo equivalente al ancho de pulso y otro que hace lo contrario para mandar las señales a los servos o motor.

A continuación, se muestra el montaje realizado en Simulink para la lectura de los valores que recibe el APM del receptor.

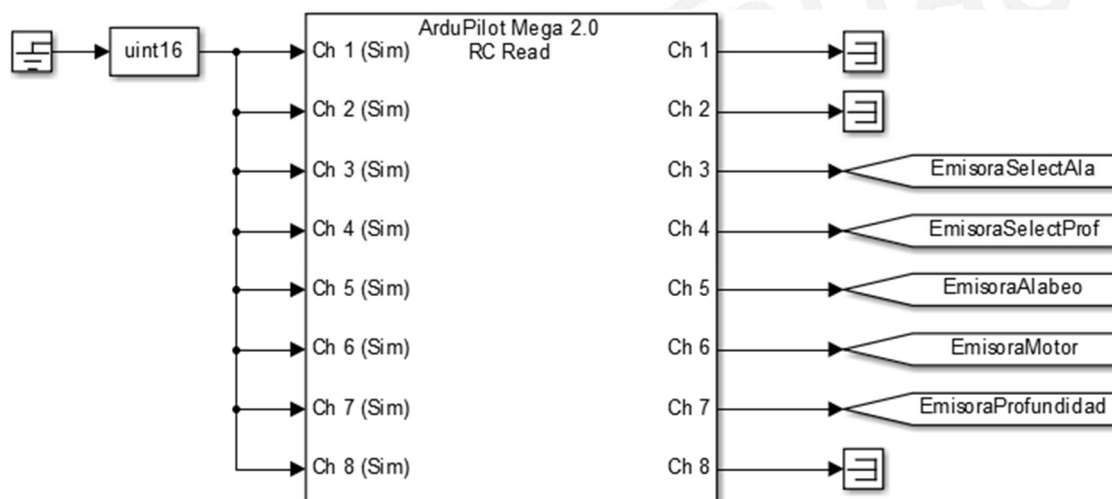


Ilustración 7: Bloque de lectura

Los canales utilizados para los ensayos fueron únicamente los tres necesarios para el vuelo manual (EmisoraProfundidad, EmisoraAlabeo y EmisoraMotor) y los dos que activan las señales PRBS para realizar los ensayos.

De la misma manera que se realizó el montaje para la lectura de los canales, se montó uno para escribir los canales, en este caso, solo fueron necesarios tres canales de escritura.

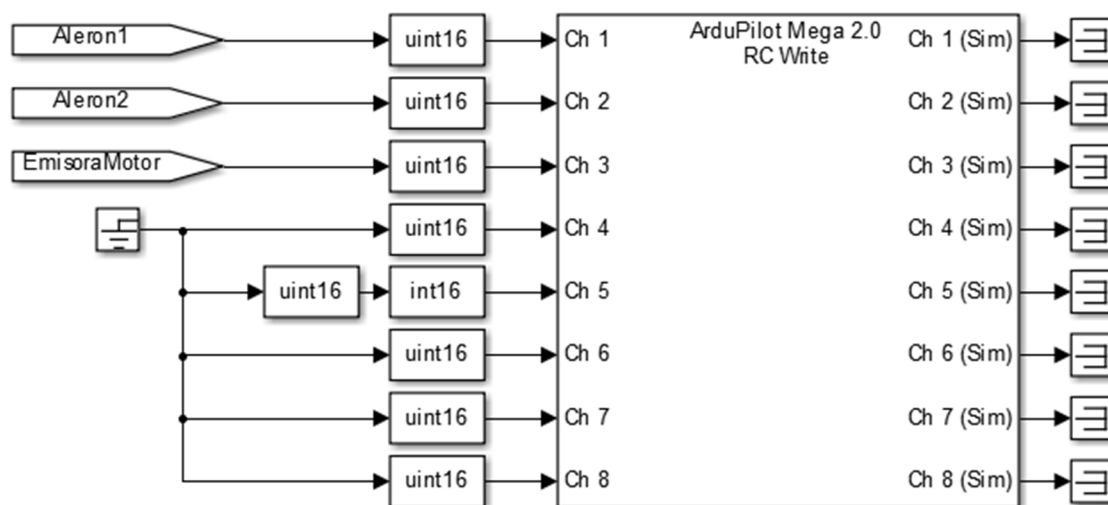


Ilustración 8: Bloque de escritura

Los canales utilizados de salida son las señales enviadas a cada uno de los alerones y la enviada al motor. La señal recibida desde el receptor con la señal del motor se manda sin tratarse al motor, ya que no es una señal que se quiera durante estos ensayos controlar. Las señales enviadas a los alerones son señales tratadas a partir de las señales de lectura del receptor y las señales PRBS creadas.

3.2.2 SEÑAL PRBS

La señal PRBS se crea para obtener la señal con mayor cantidad de frecuencias no despreciables posible, esto permite la obtención de una función de transferencia que sea correcta para un espectro lo más amplio posible. Esta señal se crea en Matlab mediante la siguiente línea de código.

$u = \text{idinput}(310, \text{'prbs'}, [0, 0.1], [-20, 20]);$

Esta variable u es una secuencia de 310 valores de 20 o -20, con ancho de banda entre 0 y 0.1. Se utilizó 310 dado que es una de las cantidades en las que Simulink es capaz de crear una señal PRBS. Se limitó el número de valores a 310 para no ocupar memoria excesiva, dado que un número mayor, implicaría una cantidad mayor de frecuencias no despreciables, pero también una ocupación mayor en cuanto a memoria. Una vez generada la señal, se introduce en Simulink como una señal periódica para que se repita durante todo el ensayo.

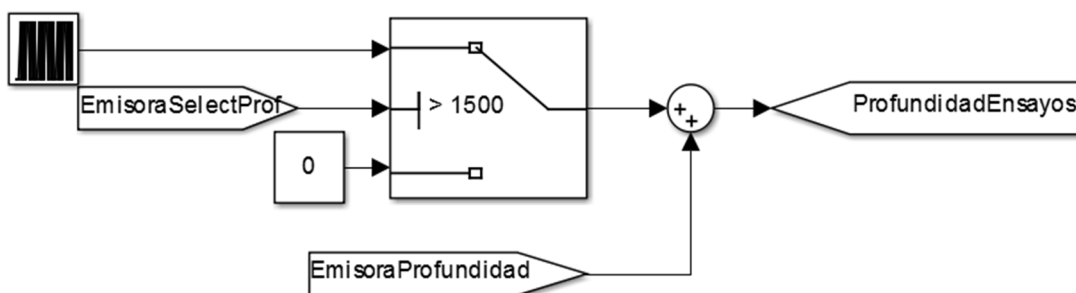


Ilustración 9: Señal de mando en ensayos

El montaje anterior toma el valor del control manual y le suma, si ha sido activado *EmisorSelectProf*, la señal PRBS, creando el mando de profundidad. Análogamente se realizó un montaje para el mando de alabeo.

Para comprobar que efectivamente la señal generada era lo que se pretendía, se realizó en Matlab el código que se muestra en Diagramas de Fourier, para comprobar el diagrama de Fourier de la señal.

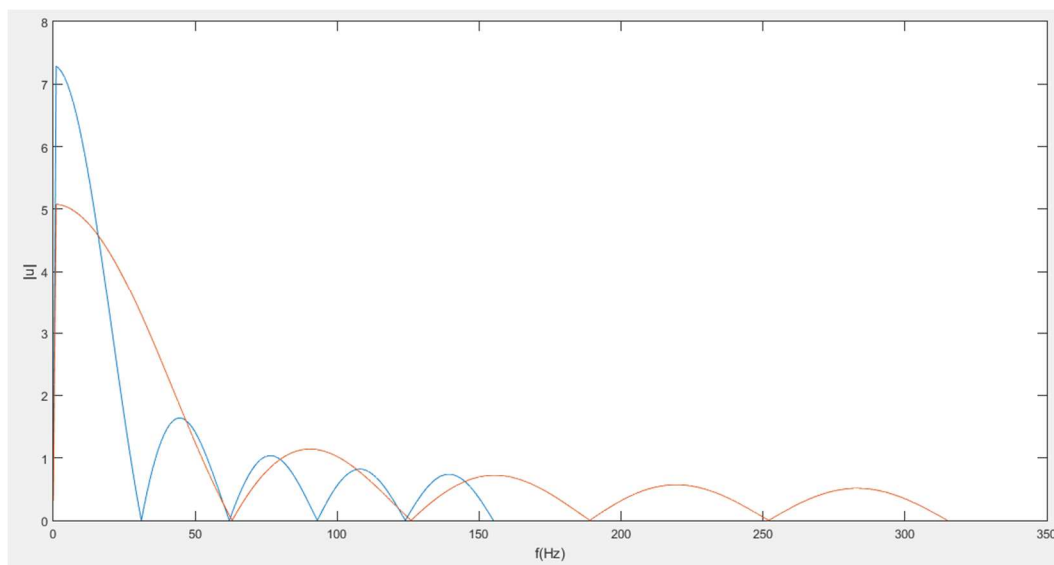


Ilustración 10: Diagrama de Fourier de la señal PRBS

Podemos comprobar en la Ilustración 10, donde la gráfica azul representa la señal utilizada en los ensayos y la roja una realizada con mayor número de valores (630 valores), que se obtienen buenos resultados para frecuencias por debajo de aproximadamente 20 Hz. Se consideró suficientemente buena la señal generada con 310 valores como para la realización de los ensayos.

3.2.3 MEZCLA ALERONES PARA LA COLA EN V

A pesar de que la emisora tiene la posibilidad de crear la mezcla necesaria para la configuración de cola en V del avión (usa las mismas superficies móviles para la profundidad y el alabeo), trabajar con las señales de cada uno de los alerones sería poco intuitivo, por lo que la mezcla para obtener las señales de cada uno de los alerones se realizó en Simulink para el microprocesador.

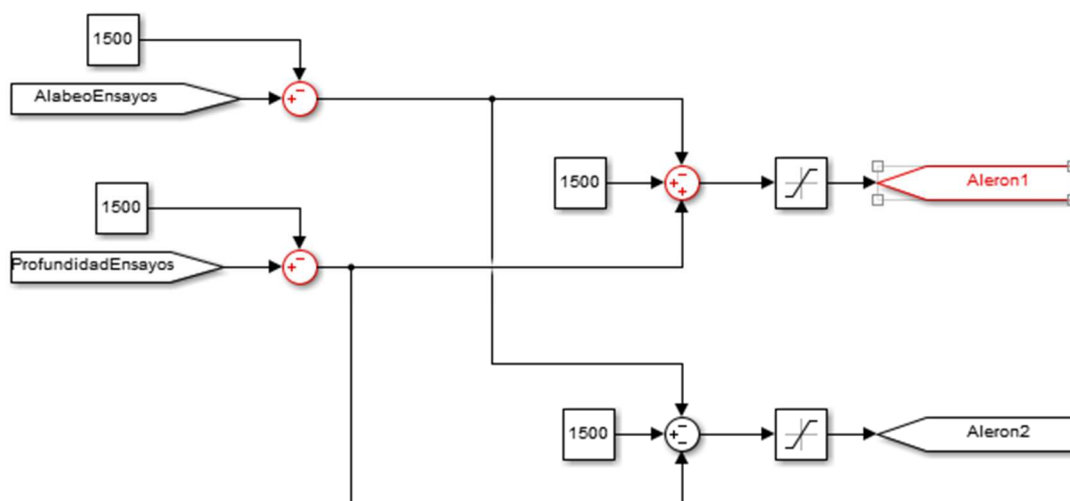


Ilustración 11: Mezcla alerones para cola en V

Debido a la colocación de los servos en el ala, la señal de profundidad entra con signo contrario en cada alerón mientras que la de alabeo entra con mismo signo. A la salida de la mezcla se incluyó una saturación que impidiese salirse a la señal de los límites de entre 1000 y 2000 a los que trabajan los servos.

3.2.4 SENSORES

Las medidas necesarias para conocer la situación del avión se realizaron usando diferentes sensores. Para reducir el ruido de los sensores se diseñó un filtro paso bajo que se utilizó en todas las medidas obtenidas de estos.

La orientación del avión se obtuvo usando el acelerómetro y giroscopios del APM, estos valores los proporciona el bloque IMU (Inertial Measurement Unit), que proporciona los ángulos de Euler del avión, usando para ello el filtro Kalman cuyo código se encuentra en Sensor_fusion. Se trata de un observador que, mediante una matriz de ganancias, estima las variables de estado que no puede medir. El filtro Kalman trabaja en dos etapas, una primera de predicción y una segunda de corrección, esto le permite hacer estimaciones muy fiables.

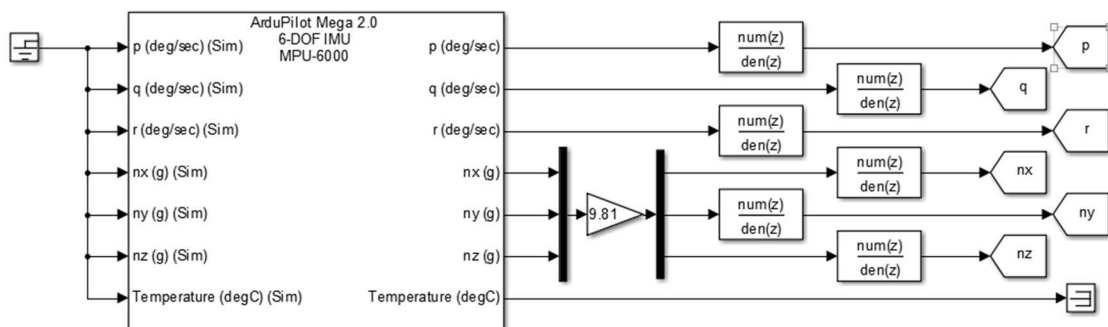


Ilustración 12: IMU acelerómetros y giroscopios

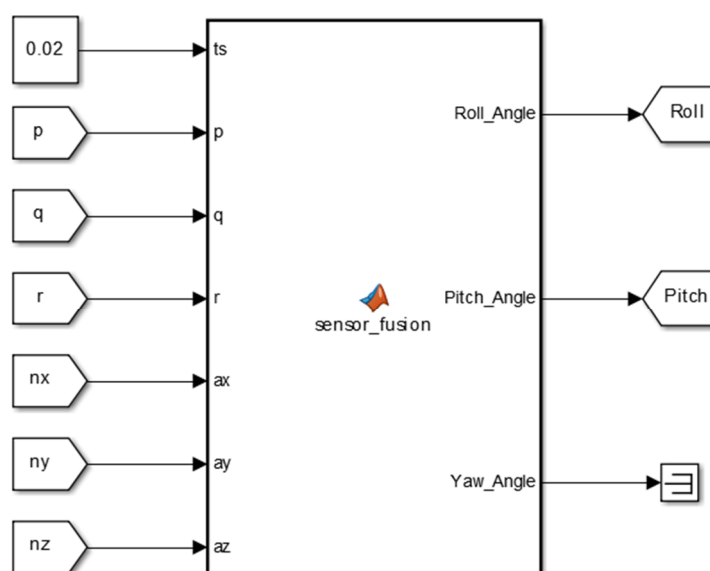


Ilustración 13: Función filtro de Kalman

El bloque de la Ilustración 13: Función filtro de Kalman, permite la obtención de los ángulos de pitch, roll y yaw del avión, pero dado que el control del ángulo de yaw no se realizará en este proyecto, no se utilizó.

La altura se obtuvo utilizando el barómetro del APM. La altura del avión se obtiene mediante esta fórmula

$$H = 7900 \times \ln\left(\frac{P}{P_0}\right)$$

Donde P es la presión medida en cada instante y P_0 la presión medida a la altura del suelo.

Dado que el valor que proporciona el barómetro se encuentra en inHg, este valor se multiplicó previamente por 25.4 para obtener el valor en mmHg. Para no tener que introducir a cada vez que se hiciese un vuelo el valor de la presión a la altura del suelo, se hizo que al iniciarse el microprocesador tomase el valor de inicio como esa presión. Para evitar el ruido de medida en este valor de presión se introdujo el filtro para tomar un valor más fiable de la presión y se tomó como presión P_0 la que hubiese pasado 3 segundos.

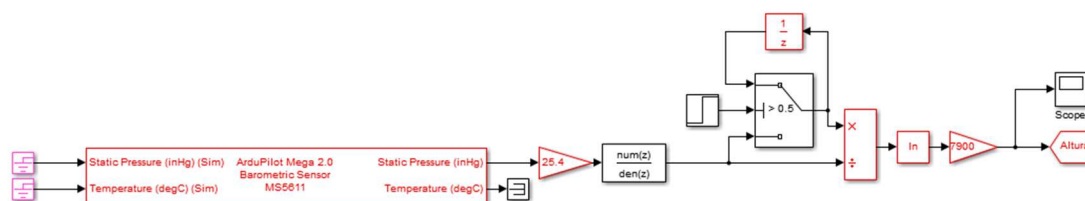


Ilustración 14: Altura, sensor barométrico

El bloque para el tubo pitot proporcionado por la librería de Arduino, estaba diseñado para otro modelo de placa de presión, por lo que fue necesario realizar unos ensayos para establecer la relación entre la velocidad real del avión y los valores leídos de manera analógica. Una vez realizados los ensayos que se explican en el apartado Ensayos, se utilizó el montaje siguiente para el resto de vuelos.

3.3 ENSAYOS

Para la realización de los ensayos se intentó tanto en el caso de los vuelos como en el del calibrado del tubo pitot hacerlos en un día en el que hubiese el mínimo número de perturbaciones. Se buscó un día con mínimo viento posible y



sin ráfagas, y no excesivo calor para evitar térmicas y bolsas de aire ascendente que afectasen el vuelo del avión.

3.3.1 TUBO DE PITOT

El primer ensayo que se realizó consistió en la toma de valores medidos con el tubo pitot que se usaría en el avión y de un tubo pitot que diese valores reales en un túnel de viento. El montaje del tubo que serviría para obtener los valores reales consistió en la conexión de los dos canales del tubo a un tubo en forma de U graduado verticales por los dos lados y relleno de agua hasta media escala.



Ilustración 15: Ensayo tubo de pitot

Este montaje permitía ver la diferencia de presión mediante un cálculo y así obtener la velocidad del aire alrededor del tubo pitot.



$$\Delta p = \rho w g h \cos(\alpha)$$

Donde:

Δp es la diferencia de presión medida entre los dos puntos de medida del tubo pitot.

ρw es la densidad del agua.

g es la aceleración de la gravedad.

h es la diferencia de altura del agua de cada lado del tubo en forma de U.

α es la inclinación del tubo en forma de U con respecto a la vertical.

Y la velocidad se obtuvo de la siguiente manera:

$$v = \sqrt{\frac{2 \times \Delta p}{\rho a}}$$

Donde:

v es la velocidad del fluido.

ρa es la densidad del aire.

Las medidas del tubo pitot que se montaría en el avión se obtuvieron viendo en Simulink los valores en directo, el montaje que se utilizó fue el siguiente:

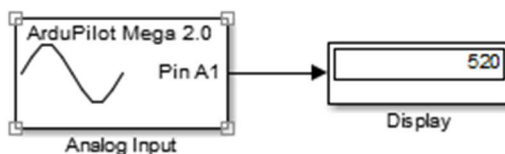


Ilustración 16: Lectura tubo de pitot

Lamentablemente, los valores obtenidos en el túnel de viento no resultaron fiables debido a una modificación que se había realizado en el túnel de viento unos años atrás que creaba realimentaciones del aire en el túnel de manera que resultaba imposible obtener valores satisfactorios.



Se realizó por lo tanto un segundo ensayo para obtener la relación entre velocidad del vehículo y el valor proporcionado por el pitot. Este ensayo consistió en el montaje del tubo pitot en un coche y obtener valores para velocidades constantes de este. Para la realización de este ensayo se procuró mantener lo más alejado posible el tubo pitot del vehículo para que la aerodinámica de este no afectase a las medidas.

A continuación, se muestra una tabla resumen con los valores experimentales obtenidos con el tubo pitot

Tabla 1: Resultados del ensayo del tubo de Pitot

Velocidad coche (km/h)	Valor medido por el tubo pitot
0	520
20	523
30	526
40	530
50	539
60	545
70	553
80	560
90	570
100	580

Una vez introducida la tabla anterior en el Excel se utilizó la herramienta de identificación de tendencias con los que se obtuvo la fórmula siguiente:

$$v = \frac{\sqrt{4 \times a \times (p - 520) + b^2} - b}{2 \times a}$$



Donde:

a y b son factores experimentales de valores respectivos 0.0049 y 0.1228

p es el valor medido por el tubo pitot

Se realizó una comprobación de que la curva se ajustase adecuadamente a los valores obtenidos y se obtuvo la siguiente gráfica en la que se muestran tanto los datos obtenidos como los calculados. Se puede comprobar que a bajas velocidades la precisión no es tan elevada como con valores de velocidad mayores.

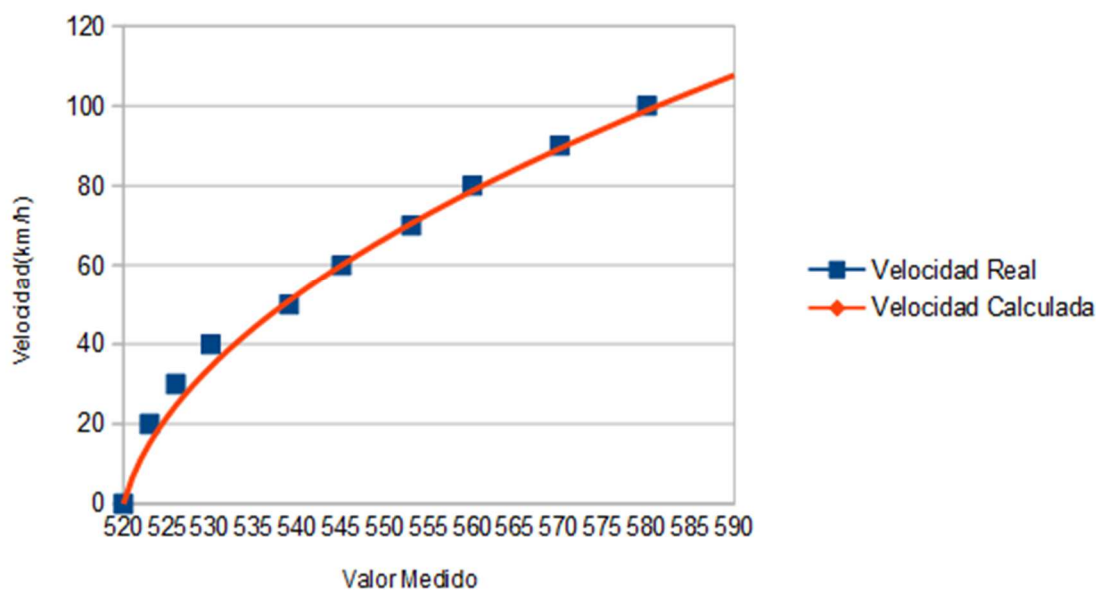


Ilustración 17: Gráfica de velocidad del pitot

Una vez finalizado este ensayo se montó la siguiente función en Simulink para el cálculo de la velocidad del avión con respecto a la masa de aire en la que se mueve.

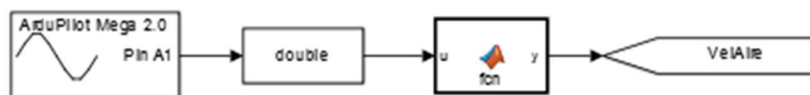


Ilustración 18: Velocidad del avión

Donde el bloque función es el mostrado en el Función Pitot de la parte IV.

3.3.2 MEMORIA FLASH Y VUELOS

Los ensayos en los que se realizaba un vuelo requerían el uso de la memoria flash del Ardupilot. Para evitar errores en el uso de esta, se necesita seguir los siguientes pasos.

- Montaje del Simulink como muestra la ilustración siguiente, tantas variables como se quiera conectadas a un bloque *mux* conectado al bloque *memoria flash*.
- Poner nombres a las variables en el bloque de la memoria. Estos nombres se ponen de la siguiente manera {'clk', 'Altura', 'Roll', 'Pitch', 'PM', 'AM', 'PS', 'PA', 'motor', 'pitot'}. Para evitar problemas a la hora de leer las variables, se debe poner tantos nombres como entradas al bloque.
- Una vez se ha usado *Deploy to hardware*, para comenzar a guardar datos se debe correr el programa *ByA*, para borrar los datos que se encontrasen previamente en la memoria y comenzar a guardar de nuevo.
- Una vez se ha comenzado a guardar datos, no debe cortarse la alimentación en toda la duración del ensayo, ya que esto pararía el proceso de guardado de datos. Por esta razón se debe, sin desconectar de la alimentación del ordenador, comenzar a alimentar con una batería externa. Una vez conectada esta batería se puede desconectar el ordenador y realizar el ensayo.

- Para finalizar el registro de datos, una vez acabado el ensayo, se debe desconectar la alimentación del microcontrolador.
- Para acceder a los datos guardados en la memoria, se debe conectar al ordenador y correr el programa *Ardudata*, y aparecerán los datos en la ventana de *Workspace* de Matlab con el nombre *data*. Esta variable *data* es una matriz que contiene todos los valores de cada una de las variables guardadas en el bloque *memoria flash*.

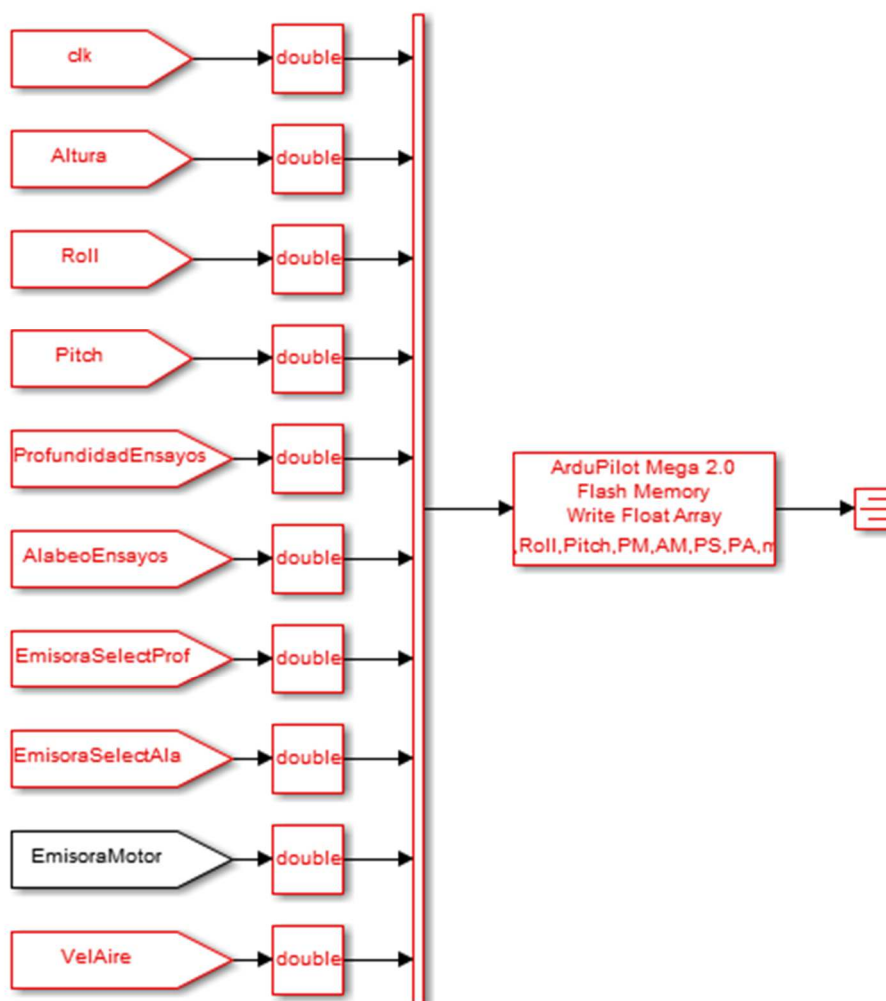


Ilustración 19: Memoria flash

Se realizaron dos ensayos distintos en cuanto a vuelos, el de profundidad y el de alabeo. Ambos se realizaron a velocidad constante, que se consiguió introduciendo un control PI con referencia de 35 km/h (velocidad que en vuelos anteriores había demostrado ser adecuada para el vuelo de este avión). Los ensayos

consistieron en la introducción en los mandos de profundidad y alabeo la señal PRBS (ver 3.2.2) sumada al mando enviado por la emisora. En ambos ensayos se trató de realizar el mínimo número de correcciones manualmente, únicamente se realizaron aquellas que evitasen que el avión alcanzase niveles en los ángulos de Pitch y Roll que pudiesen ser considerados como un riesgo.

El control PI que se utilizó para controlar la velocidad durante los ensayos fue diseñado sin tener previa planta, por lo que se diseñó utilizando valores que se consideraron aceptables. El control se implementó con el siguiente montaje de Simulink. Donde el selector izquierdo sirve para evitar que la acción integral tenga en cuenta los errores respecto a la velocidad de referencia durante el control manual del avión.

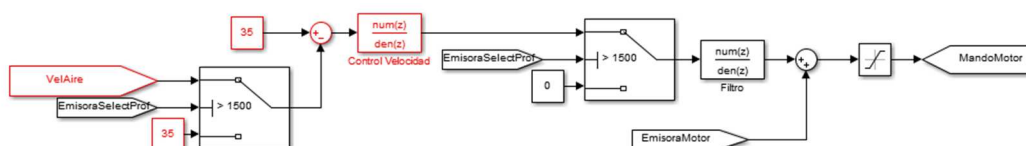


Ilustración 20: Montaje del control de velocidad

Tras los primeros ensayos realizados, se realizaron unos cambios en los valores introducidos inicialmente dado que se observó que oscilaba la velocidad en torno a la velocidad de referencia. Los valores que se usaron se encuentran en Control velocidad.

3.4 OBTENCIÓN DE LA PLANTA

La obtención de la planta se realiza mediante los valores registrados en los ensayos explicados en el apartado Memoria flash y vuelos. Las variables de mando y aquellas medidas por los sensores permiten encontrar la función de transferencia que aproxime el comportamiento de cada una de las variables que se quieren controlar del avión con respecto a aquellas que se pueden modificar de mando.



Las funciones de transferencia se obtuvieron mediante la herramienta de Matlab de identificación de funciones de transferencia. A continuación, se explica el proceso que se debe seguir para esto.

- Abrir la herramienta de identificación de funciones de transferencia de Matlab, para ello, en la ventana de comandos se debe escribir el siguiente comando *ident*.
- Una vez abierta la ventana, se deben introducir las variables que se quieren relacionar mediante una función de transferencia, se debe tener cuidado a la hora de introducirlas en la casilla correspondiente ya sea *input* u *output*.
- Para obtener una función correcta, se debe tratar previamente la señal, por un lado, se debe eliminar los valores medios de las señales y restringir el rango de los datos, ya que solo interesan aquellos en los que el avión está volando.
- Es recomendable que se hagan varias muestras, ya sea de dos ensayos distintos o dividir un ensayo en partes, de manera que la herramienta de identificación de funciones de transferencia sea capaz de, una vez obtenida la función comprobarla en otra muestra.
- A la hora de identificar la función de transferencia, se debe indicar el número de polos y ceros de la función, y se debe hacer la prueba con distintos números hasta obtener una función que se ajuste bastante al modelo real. Se sabe si estas se ajustan lo suficiente comparando los resultados que se obtuvieron en los sensores con los que se obtendrían de la función creada, este resultado se da a modo de porcentaje y se consideró suficientemente bueno aquel que superase el 50%, siempre eligiendo aquel que fuese de orden menor, pero con porcentajes mayores.

Una vez obtenidas las funciones proporcionadas por la herramienta de Matlab, se realizó unos montajes en Simulink que calculaban el error cuadrático al introducir a la planta calculada los datos de entrada y comparar la salida con los medidos por los sensores. Esto último permitió ajustar los valores de manera que mejorase en cierta medida las plantas proporcionadas.

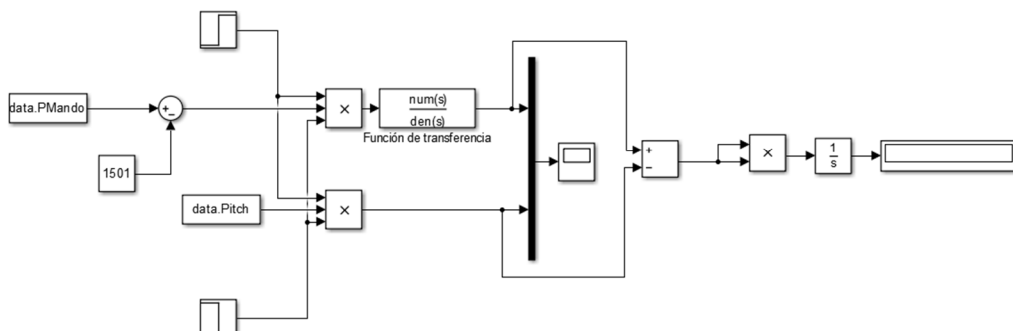


Ilustración 21: Simulink para calcular el error cuadrático de la función de transferencia

Las plantas que se obtuvieron fueron las que relacionan, ángulo pitch con altura, mando de profundidad con ángulo de pitch y mando de alabeo con ángulo de roll. A continuación, se muestran estas funciones de transferencia, que fueron las utilizadas para el diseño de los controles.

Planta Pitch-Altura:

$$\frac{6.966 \times s^3 - 42.82 \times s^2 + 2.951 \times s - 24.93}{s^5 + 20.41 \times s^4 + 71.57 \times s^3 + 21.47 \times s^2 + 53.58 \times s + 1.471e^{-10}}$$

Planta Mando de profundidad-Pitch:

$$\frac{0.1604 \times s + 0.005576}{s^2 + 0.008784 \times s + 0.0007731}$$

Planta Mando de alabeo-Roll:

$$\frac{0.5654 \times s^4 - 1.253 \times s^3 - 0.3665 \times s^2 - 0.109 \times s - 0.02788}{s^5 + 5.734 \times s^4 + 2.492 \times s^3 + 0.5961 \times s^2 + 0.1827 \times s + 0.007881}$$





Capítulo 4 DISEÑO TEÓRICO DE LOS CONTROLES

El diseño de los controles se realizó utilizando las plantas obtenidas en el apartado Obtención de la planta. Una vez realizados diseñados los controles se comprobó su funcionamiento realizando simulaciones en Simulink.

4.1 DISEÑO

Se decidió realizar controles distintos en función de las variables a controlar. Al haber únicamente dos variables de mando (mandos de profundidad y alabeo), tres variables a controlar (Pitch, Altura y Roll), y tener dos de estas variables a controlar relacionadas por una función de transferencia, se decidió diseñar un control en cascada para el Pitch y la Altura y un control PI para el Roll.

4.1.1 CONTROL EN CASCADA

El control en cascada consiste en el diseño de dos controles que afectan a variables que están relacionadas, en este caso los controles del ángulo de pitch y de la altura del avión.

4.1.1.1 Control interno: Pitch

El control interno que se realizó fue un control P, se trata del control típico utilizado en los controles internos de los controles en cascada, ya que al realizar un control exterior PI se obtiene uno equivalente a un PID para el control total, el control P del interior actúa como parte diferencial para el control externo. Para diseñar el interior, se introdujo la planta obtenida que relaciona el ángulo de pitch con el mando de profundidad en Matlab, y utilizando la aplicación de Matlab, PID

Turner, que permite modificar los parámetros de los controles y ver la respuesta a un escalón en la referencia. Esto permitió obtener un valor cercano al deseado y se realizó un script en Matlab para el diseño por margen de fase del control, según el código mostrado en Control de altura, con el que se ajustó completamente el control. El control P obtenido fue de una K de valor -0.3279 . Nótese que el valor obtenido de K es negativo, esto se debe a la ganancia estática negativa de la planta.

4.1.1.2 Control externo: Altura

El proceso que se siguió para el diseño del control externo del control en cascada fue similar equivalente al explicado en el apartado anterior, Control interno: Pitch, pero en este caso se eligió un control PI de manera que no existiese error en régimen permanente para la altura. En este caso la planta introducida en la aplicación de Matlab, PID Turner fue la función de transferencia equivalente a realizar la realimentación de la planta interna con control interno y añadir en serie la planta obtenida para la altura en función del ángulo de pitch. El control que se obtuvo no resultó ser demasiado bueno dado que, en la simulación, pese a finalizar en el valor deseado realizaba una bajada en la altura antes de alcanzar este. Para solucionar el problema que mostraba la simulación, se decidió añadir una ponderación a la referencia de manera que se obtuviese una respuesta simulada mejor.

El control final en cascada que se obtuvo fue el mostrado en la figura siguiente

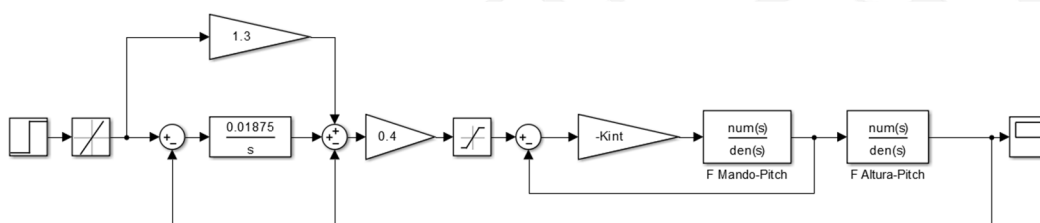


Ilustración 22: Control de altura en cascada



Donde K_{int} es el valor obtenido en para el control P interior, el signo menos se debe a que se diseñó en el script de Matlab de manera que diese un valor positivo, pese a que realmente el valor es negativo.

4.1.2 CONTROL ROLL

El diseño del control del ángulo de roll se diseñó de manera análoga al diseño de los controles anteriores, realizando primero una aproximación mediante la aplicación de Matlab y un posterior ajuste mediante las simulaciones en Simulink. El control que se decidió usar en este caso fue un PI dado que se buscó una máxima robustez frente a un control rápido, y considerando necesaria una acción integral para evitar error en régimen permanente. El control que se obtuvo fue un control PI con valores K (K_{roll} en la imagen) igual a -0.4511 y T_i (T_{iroll} en la imagen) igual a 10.4.

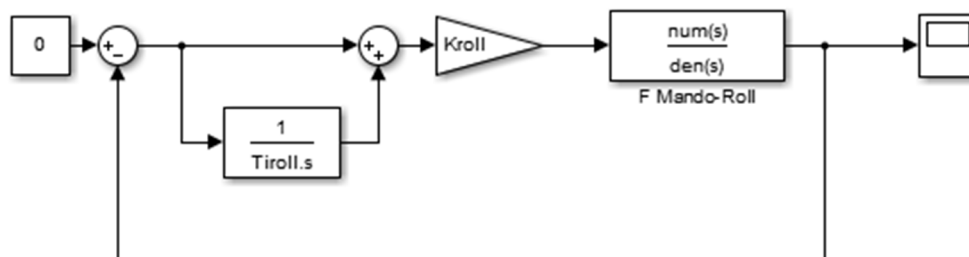


Ilustración 23: Control de Roll PI

Simultáneamente al diseño de los controles explicados, de altura en cascada y PI para el ángulo de roll, estos se fueron simulando para obtener los que diesen una simulación mejor frente a escalones en la referencia y perturbaciones para el control de altura y perturbaciones para el control de alabeo ya que se considera que se mantendrá siempre la referencia de cero grados en el ángulo de roll.



4.2 SIMULACIÓN

A la hora de realizar las simulaciones se introdujeron en los montajes de Simulink, Ilustración 22 los controles (Ilustración 22: Control de altura en cascada e Ilustración 23: Control de Roll PI) entradas que pudiesen simular o bien variaciones en la referencia para el control de altura o bien perturbaciones para ambos controles.

4.2.1 SIMULACIÓN CONTROL DE ALTURA

4.2.1.1 Escalón en la referencia

Para evitar que el salto en la referencia fuese excesivo se incluyó, tanto para la simulación como para la posterior implementación en el avión, un bloque, *Rate Limiter*, que limitaba la pendiente en la referencia a un metro por segundo, de esta manera, un salto en la referencia de 100 metros se realizaría mediante una pendiente constante durante 100 segundos. De la misma manera que se deseaba limitar los saltos en la referencia, también se quería limitar el ángulo pitch del avión por lo que la referencia del ángulo se limitó entre los valores -15° y 20° para evitar por un lado fuertes picados y por otro, subidas muy pronunciadas, esto se realizó añadiendo un bloque *Saturation* en la referencia de ángulo.

Para el ensayo de la respuesta a un escalón en la referencia se utilizó la Ilustración 22: Control de altura en cascada, a continuación, se muestran los resultados de la respuesta simulada a un escalón de 100 metros en la referencia de altura junto a la referencia después del *Rate Limiter*.

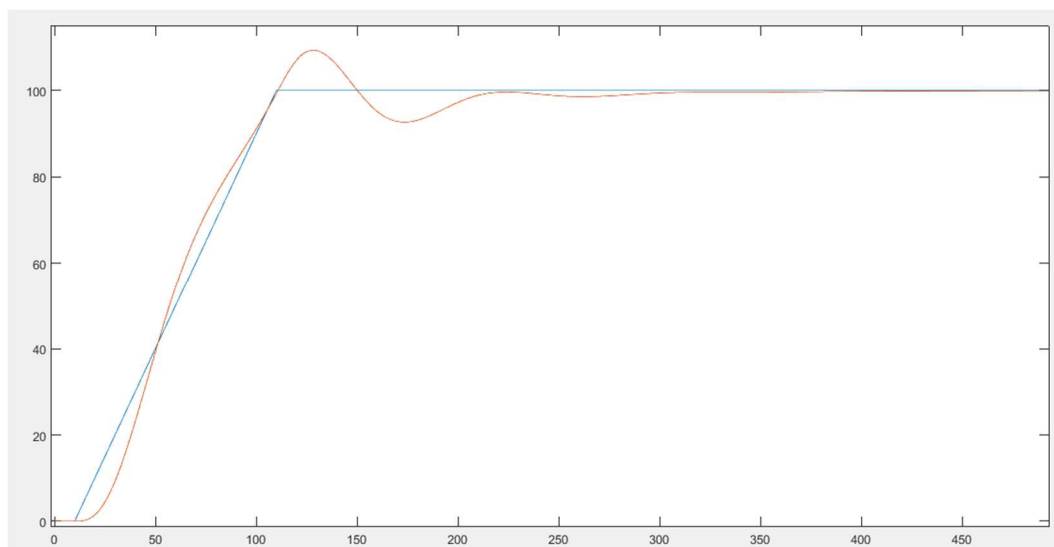


Ilustración 24: Respuesta a un escalón en la referencia del control de altura

El escalón simulado es de valor mucho mayor que el que se espera que pase en la realidad, dado que el control se activará una vez se encuentre el avión a una altura elevada, por lo que el sobrepaso del 10% sería mucho menos notable.

4.2.1.2 Tren de escalones como perturbación

Para comprobar la robustez del control se realizó una segunda simulación frente a perturbaciones. A continuación, se muestra el esquema utilizado para simular una perturbación en la medida del ángulo de pitch del avión, se utilizó como valores amplitudes de 5° .

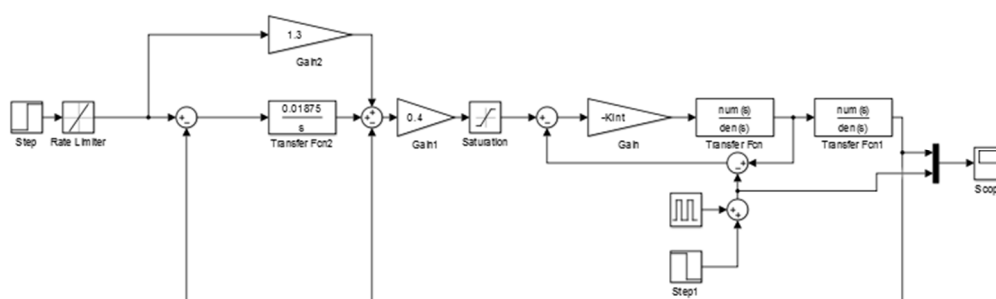


Ilustración 25: Esquema para una perturbación en el ángulo en el control de altura



En la siguiente gráfica se muestran los resultados de la simulación, tanto la gráfica de la perturbación como la de la respuesta a esta.

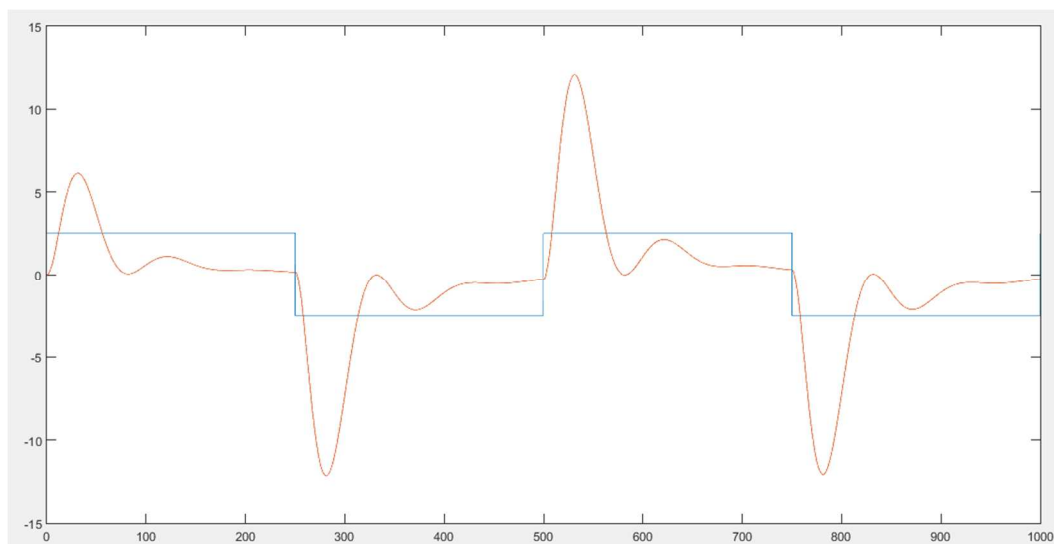


Ilustración 26: Respuesta a un tren de escalones en la perturbación del control de altura

Se puede observar que con una perturbación de 5° el avión tiene variaciones de 15 metros sobre la referencia, pero rápidamente recupera el valor de referencia.

4.2.2 SIMULACIÓN CONTROL DE ROLL

Dado que el roll iba a tener como referencia siempre cero, se pretende siempre mantener el avión equilibrado, se realizaron dos simulaciones en las que se introdujo un tren de escalones a modo de perturbación en el mando del control y en



la medida del ángulo. A continuación, se muestra, en la Ilustración 27, la gráfica de la respuesta a las perturbaciones junto a la de las perturbaciones en el mando.

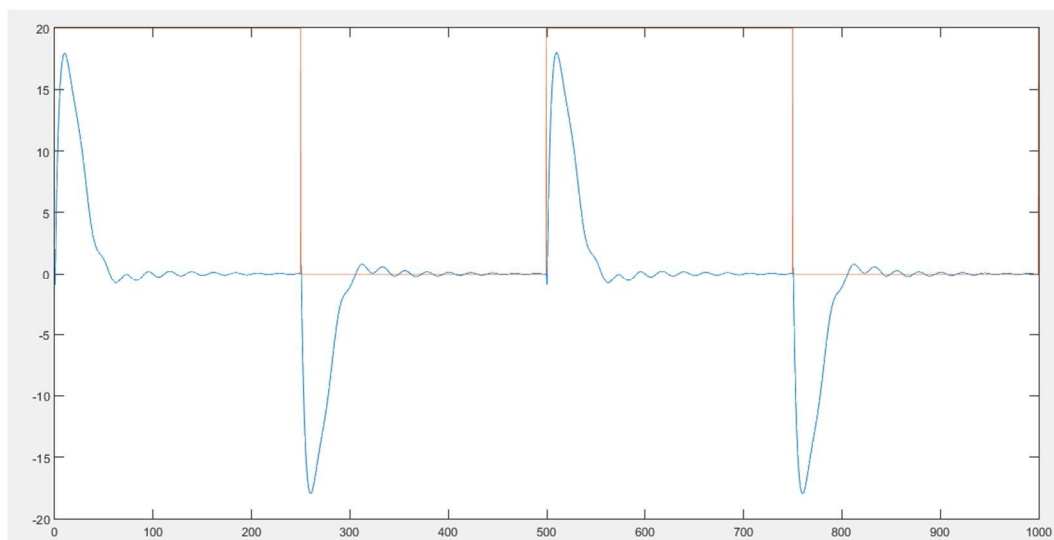


Ilustración 27: Respuesta a un tren de escalones a modo de perturbación en el control de roll

Una gran perturbación en el mando como la simulada, genera un ángulo bastante grande pero que es rápidamente eliminado y llevado a la referencia por el control.





Capítulo 5 COMPROBACIÓN DE LOS CONTROLES DISEÑADOS

Para comprobar la efectividad de cada uno de los controles diseñados, se realizó un ensayo para cada uno de los controles, implementando a cada vuelo uno solo de los controles.

5.1 IMPLEMENTACIÓN DE LOS CONTROLES

Se realizó un Simulink distinto para la comprobación de cada uno de los controles, de manera que se pudiese con un vuelo realizar la comprobación del buen funcionamiento del control de altura y con un segundo vuelo el del alabeo.

5.1.1 COMPROBACIÓN DE LOS CONTROLES

Los controles fueron implementados en base a los controles diseñados en el apartado Diseño, pero se realizaron pequeñas modificaciones en las ganancias en las que aparentemente los controles actuaban escasamente. Para ello se creó un nuevo Simulink con la mayor parte de los montajes explicados para realizar los ensayos. La única variante con respecto a los montajes realizados para los ensayos fue la sustitución de la señal PRBS de la Ilustración 9 por el control. El detalle de los sistemas de Simulink creados para la implementación de los controles se encuentra en Ficheros de Simulink.

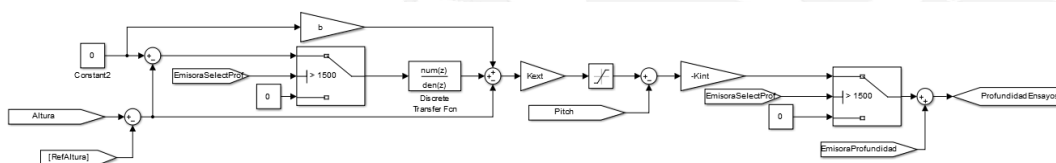


Ilustración 28: Implementación del control de altura

Para evitar poner una referencia fija, se decidió crear un subsistema en Simulink que tomase como referencia de altura la altura que tuviese el avión en el momento de la activación del control, y que se pudiese añadir a voluntad mediante otro canal un escalón de 10 metros en la referencia. De la misma manera, se creó un subsistema que permitiese cambiar a tres distintos valores de referencia de roll. Esto permitiría comprobar la eficacia de los controles tanto a escalones positivos como a escalones negativos. Estos subsistemas se muestran en las Ilustración 48 e Ilustración 49.

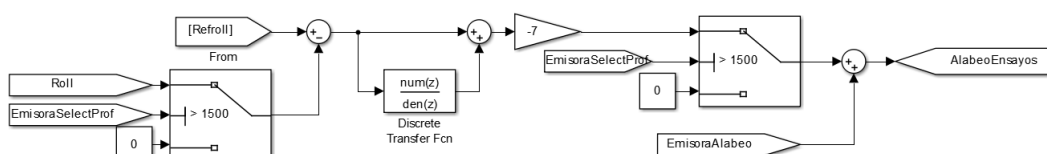


Ilustración 29: Implementación del control de alabeo

El selector izquierdo hubo que incluirlo en ambas implementaciones para evitar que la acción integral tuviese en cuenta valores anteriores al momento de activación de los controles.

Finalmente se realizó un vuelo en el que se implementaron ambos controles para así conseguir un avión que tuviese vuelo estable a altura de referencia.

5.2 RESULTADOS

Durante los vuelos de comprobación de los controles, se realizaron registros de datos en la memoria flash tal y como se explica en el apartado Memoria flash y vuelos, para comprobar la diferencia de estabilidad entre el vuelo controlado y el vuelo sin los controles activos.

Como ya se ha comentado, los controles se comprobaron primero por separado para identificar si alguno de ellos era erróneo o no tan bueno como se



deseaba. Todos los ensayos de los controles se hicieron utilizando el control que se diseñó para mantener la velocidad constante.

5.2.1 CONTROL DE ALTURA

Los primeros ensayos realizados con el control de altura devolvieron unos resultados que mostraban el control como muy lento, tal y como se había diseñado por la dificultad de la planta, es por ello que se decidió aumentar un poco el valor de la constante K del control interno en cascada. Una vez realizados nuevos ensayos se obtuvieron resultados más satisfactorios, pero aún demasiado lentos. Se optó por dar los resultados obtenidos como válidos dado que en el programa de diseño *PID Tuner*, el control se convertía rápidamente en inestable cuando se hacía más rápido.

La gráfica siguiente muestra los resultados del control una vez se utilizaron los valores finales.

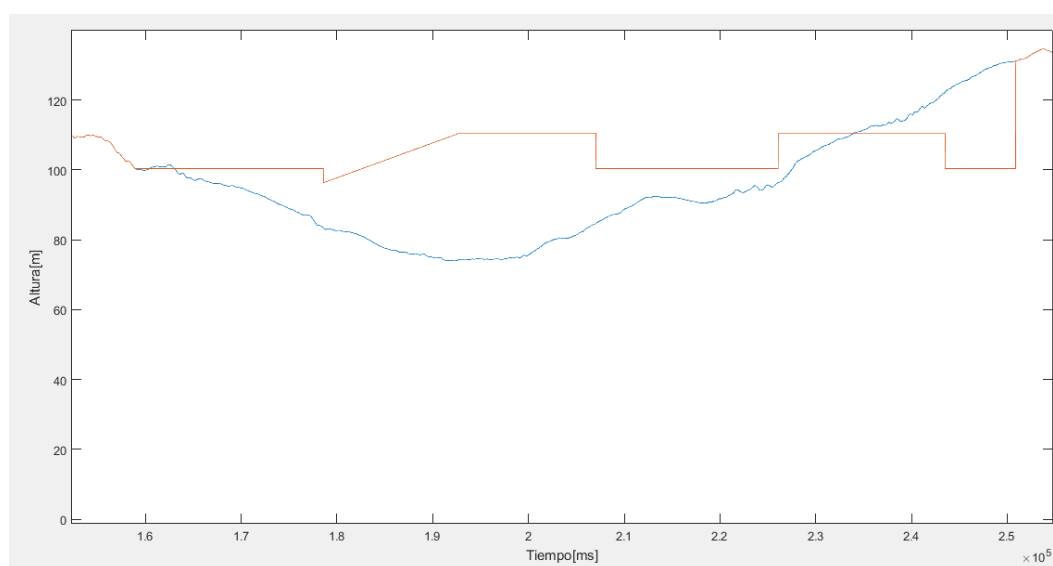


Ilustración 30: Resultado ensayo control de altura

A la vista de los resultados obtenidos se consideró que las plantas obtenidas para este control en cascada de la altura no eran suficientemente precisas como para obtener un control suficientemente rápido y estable. A pesar de todo, se consideró que la prioridad de este control residía en la estabilidad y no en la velocidad de este, por lo que comprobar que el avión recuperaba la altura perdida inicialmente se



consideró como válido. También se considera que este control debe ser mejorado si se decidiese continuar el proyecto en un futuro.

5.2.2 CONTROL DE ALABEO

Los primeros resultados obtenidos con el control de alabeo mostraron que el diseño había sido correcto pero que se podía mejorar, inicialmente el control resultaba lento y sin sobrepaso, a pesar de ello conseguía, en un día con viento ligero controlar el avión sin problema. Para mejorar el resultado y así obtener un control más eficaz, se decidió aumentar un poco el valor inicial de la constante K utilizada y de esta manera obtener un control mejor. Se realizaron varias variaciones hasta obtener uno que resultase satisfactorio. A continuación, se muestra el resultado obtenido con este último control implementado.

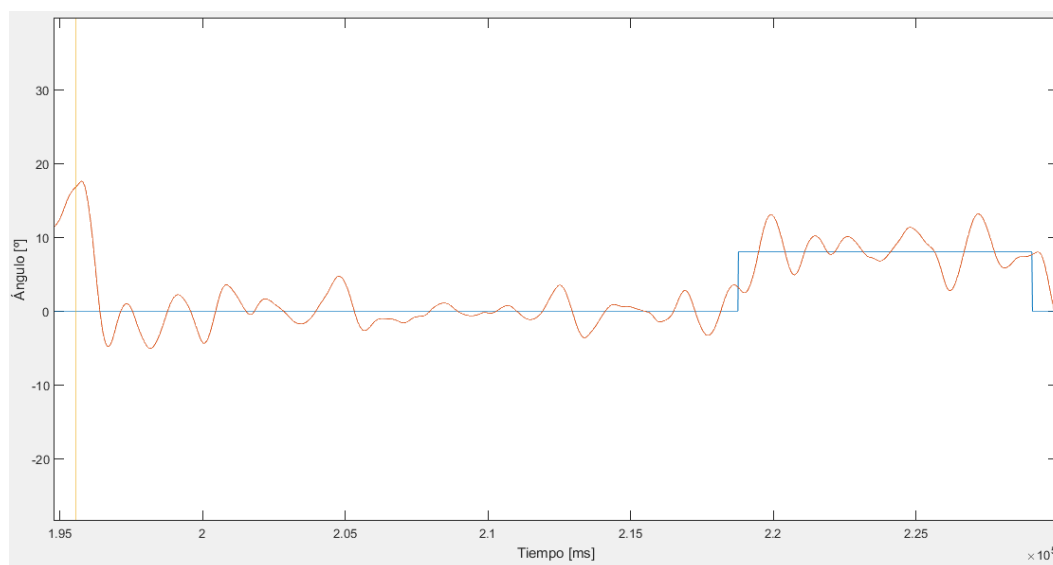


Ilustración 31: Resultado ensayo control de alabeo

La Ilustración 31: Resultado ensayo control de alabeo, muestra todo el tiempo que estuvo actuando el control, durante este ensayo, la referencia fue variada, se puede ver como el control actúa satisfactoriamente, variando en escasos grados en torno al valor de referencia.



5.2.3 CONTROL COMPLETO

Finalmente, pese a tener un control de altura que no resultaba completamente satisfactorio, se decidió que resultaba seguro hacer ensayos con ambos controles activados simultáneamente. Esta vez, como era de esperar, los resultados obtenidos fueron satisfactorios para el control de alabeo y demasiado lentos para el de altura, pero permitió mantener el avión en vuelo sin modificar más que la dirección en algunas ocasiones para no perderlo de vista durante un tiempo significativo. A continuación, se muestran los resultados que se obtuvieron con los dos controles simultáneos.

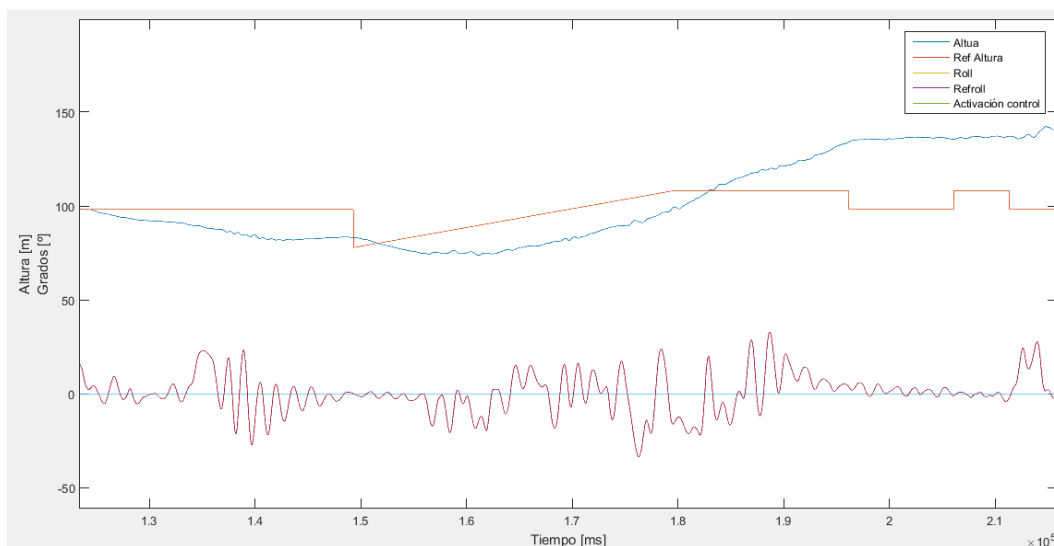


Ilustración 32: Resultado ensayo control completo

En este ensayo, se estuvo volando con el control completo un tiempo bastante extenso por lo que hubo que girar en ocasiones, esto hace que lo que parece un error del control realmente sea el mando enviado por el piloto para evitar que el avión se alejase. Se puede comprobar que cuando se dejaba solo el control actuar, este llevaba rápidamente el avión a una situación cercana a la referencia.



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL

Comprobación de los controles diseñados



Capítulo 6 CONCLUSIONES

Se ha recogido en este documento el desarrollo del proyecto “Navegación autónoma de una aeronave no tripulada de ala fija” realizado como Proyecto Fin de Grado (PFG) de los estudios de Ingeniería Electromecánica del ICAI.

Las conclusiones obtenidas del proyecto se resumen a continuación:

- El proyecto ha constado de dos actividades claramente diferenciadas:
 - Modelado y desarrollo de controles utilizando los programas Matlab y Simulink.
 - Realización de vuelos experimentales con el avión “RV Jet”, para comprobación y optimización de los controles realizados.
- Las plantas obtenidas relacionan los ángulos de “roll” y “pitch” con sus respectivos mandos de alabeo y profundidad, así como el ángulo de “pitch” con la altura. Cabe destacar que estas funciones dependen de las características del avión utilizado (aerodinámica, peso, potencia...), pero no deben depender de las condiciones meteorológicas en la que se esté volando. Es por ello que se trató de realizar los ensayos que permiten obtener las funciones de transferencia en un día con una intensidad de viento prácticamente despreciable y así minimizar sus efectos en la obtención de la planta.
- Se han diseñado los controles teóricos de altura y “roll” utilizando las plantas obtenidas, y se realizaron las simulaciones que permiten comprobar teóricamente la efectividad de los citados controles en las plantas obtenidas empíricamente. Los controles realizados fueron acordes a las plantas obtenidas de manera que, al obtenerse el ángulo de “pitch” tanto entrada como salida de una de las funciones de transferencia, se utilizó como variable intermedia de un control en cascada para la altura, mientras que para controlar el ángulo de “roll” se realizó un control PI.



- Los controles se introdujeron en el Ardupilot del avión una vez ajustados y se realizaron diversos vuelos experimentales para su comprobación, obteniéndose un comportamiento satisfactorio del avión para el control de “roll” y un comportamiento demasiado lento para el de altura. Considerándose que estos últimos resultados podrían ser debidos a que la obtención de la planta no fue suficientemente precisa, se ajustaron los parámetros de los controles de acuerdo con el comportamiento observado en los diferentes vuelos de prueba.
- Del análisis de los resultados finales obtenidos, se considera que el control diseñado para el “roll” ha sido razonablemente satisfactorio. Por su parte, el ensayo de altura no ha confirmado su efectividad, ya que la reacción del avión ha sido demasiado lenta en todos los casos.
- Como solución a los problemas detectados en el ensayo de altura se descartó el aumento de la constante de los controles sin realizar una obtención de planta nueva ya que se considera que un eventual sobrepaso excesivo en la altura sería un riesgo inaceptable. Se concluye como posible causa que debido a la obtención de una planta inexacta el diseño teórico del control no resultó efectivo. Por otro lado, los datos obtenidos de estos ensayos pueden utilizarse de nuevo para la obtención de una nueva planta de manera a, realizando un proceso iterativo, obtener una planta más precisa.
- Se considera que el proyecto realizado ofrece posibilidades de continuidad y desarrollo con el objetivo de confirmar los motivos por los que no ha sido adecuado el control diseñado de altura y para implantar nuevas capacidades para el vehículo.



Capítulo 7 FUTUROS DESARROLLOS

Existen bastantes posibles desarrollos para este proyecto, comenzando por aquellos objetivos que no pudieron cumplirse debido a diversos problemas que surgieron en el transcurso del proyecto. De los objetivos que se fijaron y no se pudieron realizar cabe destacar la introducción de un sistema GPS que permitiese la realización de rutas por el avión programadas previamente. Dado que el avión no consta de ninguna manera de controlar con un solo mando la dirección, los cambios de dirección se suelen obtener mediante la combinación de los mandos de alabeo y profundidad. La manera de realizar este control sería realizar el cambio del control PI de roll que se ha diseñado en este proyecto por un control P, que se convertiría en el control interno de un control en cascada para controlar la dirección del avión con un control externo PI que afectase igualmente al mando de profundidad. Este control sería similar al realizado en este proyecto con el control P interno de pitch y el externo PI de altura.

Como ya se comentó en Motivación del proyecto, el proyecto forma parte de un conjunto mayor que pretende realizar modificaciones en la estructura del avión para dotarlo de la capacidad de despegue vertical. Esto se realizaría tal y como se explica “*Desarrollo de un vehículo aéreo de ala fija con capacidad de despegue vertical para el transporte de pequeñas cargas*”[5]. Esto requeriría el diseño de controles para el vuelo transitorio, así como para el vuelo vertical, que se realizaría según el método utilizado en diversos proyectos de realización de vuelos estables de cuadricópteros. Una vez hechos los cambios estructurales para permitir el despegue vertical y hecha la comprobación de los controles diseñados para el vuelo transitorio, dado que el avión constaría de varios motores, se podría conseguir la guiñada con una excitación diferencial a los motores, a diferencia del control en cascada explicado en el párrafo anterior.

Otro posible desarrollo sería el realizar controles que permitiesen no solo el vuelo del avión, pero también un aterrizaje y despegues en su modelo de vuelo horizontal. Esto requeriría la modificación de los controles en los que se diese



menos importancia a la velocidad del control, pero asegurarse de que no tuviese sobrepaso, así como que el control de dirección permitiese un posicionamiento suficientemente certero para realizar el aterrizaje en el lugar programado.



BIBLIOGRAFÍA

- [1] A. D. M. L. Tadeo Espinoza, “*Linear and nonlinear controllers applied to fixed-wing UAV*”, Instituto Tecnológico de la Laguna, 2012
- [2] A. Noth, “*Dynamic Modeling of Fixed-Wing UAV*”, Swiss Federal Institute of Technology, Zurich, 2006.
- [3] F. Hugon, R. F. Hartley, B. DeRosa “*How To Guide: ArduPilot 2.0 Simulink Blockset*”, Embry-Riddle Aeronautical University Savannah, Georgia, 2012.
- [4] J. Martínez “*Control de un cuadricóptero para vuelos autónomos en interiores*”, Universidad Pontificia de Comillas, Madrid, 2015.
- [5] R. Álvarez “*Desarrollo de un vehículo aéreo de ala fija con capacidad de despegue vertical para el transporte de pequeñas cargas*”, Universidad Pontificia de Comillas, Madrid, 2016.
- [6] Eagletreesystem.com





Parte II ESTUDIO

ECONÓMICO



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



Capítulo 1 ESTUDIO ECONÓMICO

Para entender bien este proyecto hay que considerar que forma parte de un proyecto mayor cuyo objetivo es conseguir un vehículo volador no tripulado que permita por un lado una gran autonomía y por otro despegue y aterrizaje en un lugar muy limitado. Este proyecto se centra únicamente en el desarrollo del vuelo horizontal del avión, mientras que en otro proyecto paralelo se ha diseñado la modificación del avión para permitir el despegue y aterrizaje vertical. La coordinación entre ambos proyectos es esencial para obtener el resultado esperado.

Para considerar su utilidad hay que tener en cuenta el momento social en que nos encontramos en el que hay países que necesitan de una ayuda humanitaria, ya sea por conflictos bélicos o por catástrofes naturales. Este proyecto, pretende crear un UAV minimizando los gastos energéticos y los riesgos que puedan suponer una operación de envío de mercancías. El uso de vehículos de control remoto, supone, pese al gasto inicial necesario para su desarrollo, un ahorro energético en cuanto al peso de un UAV frente a un avión tripulado.

Tanto en el aspecto de viabilidad como en el de rentabilidad, debe considerarse el proyecto a largo plazo, ya que pretende sustituir funciones realizadas normalmente por humanos y hasta no conseguir obtener resultados completamente fiables, la utilización de este producto en ciertas situaciones supondría un riesgo no asumible.

La rentabilidad del proyecto ha de plantearse no solo en términos económicos sino en vidas humanas. El vehículo desarrollado en este proyecto podría utilizarse en misiones de rescate en lugares de difícil acceso donde un vehículo tripulado, ya sea por tamaño o por riesgo del piloto no se podría utilizar.

En cuanto a la viabilidad del proyecto, se ha comprobado que es posible obtener un UAV de vuelo horizontal autónomo, el vuelo vertical se realizaría mediante rotores colocados de manera que se comporte como un cuadricóptero, cuya viabilidad ha quedado demostrada en otros proyectos centrados en



cuadricópteros[4]. Falta demostrar la viabilidad de la fase transitoria de vuelo horizontal a vuelo vertical y viceversa.

Los aportes de un vehículo híbrido con capacidad de vuelo tanto horizontal como vertical hacen que se pueda considerar un proyecto muy versátil. Dado que se trataría de un vehículo con diversos usos, se considera que se trata de un proyecto rentable tanto económicamente como en recursos humanos.



Parte III MANUAL DE USUARIO





Capítulo 1 VUELO AUTÓNOMO

Para efectuar un vuelo autónomo se han de seguir los pasos siguientes:

- Ejecutar el fichero Fichero Maestro para generar las variables necesarias de los controles;
- Introducir en la memoria del Ardupilot el archivo de Simulink, Control completo;
- En caso de querer registrar datos de vuelo se deberá ejecutar, alimentando con una batería simultáneamente el Ardupilot, se ejecutará desde Matlab el código *ByA*, que se encuentra en Funciones memoria flash;
- El despegue y aterrizaje han de realizarse manualmente por lo que se recomienda activar los controles una vez se encuentre el avión a una altura prudente;
- Una vez finalizado el vuelo, se deberá, para finalizar el registro de datos desconectar toda alimentación al Ardupilot;
- En caso de haber registrado datos, se accederá a ellos ejecutando el código *Ardudata*, que se encuentra en Funciones memoria flash, en Matlab.





Parte IV CÓDIGO FUENTE





Capítulo 1 FICHEROS DE MATLAB

1.1 *SENSOR_FUSION*

A continuación, se muestra el código de la función que calcula los ángulos de Euler a partir de las medidas de los acelerómetros y giroscopios mediante filtros de Kalman. Esta función se utilizó en todos los ensayos y comprobaciones en las que requería el vuelo del avión.

```
function [Roll_Angle, Pitch_Angle, Yaw_Angle] = sensor_fusion (ts,
p, q, r, ax, ay, az)

%=====Constants=====
====

    delta_t = ts;
    degrees = 180 / pi;
    radians = pi / 180;

%Attitude Estimator
Q_gyroBias = 0.0000000087;           % Q_gyroBias: gyroscope noise
Q_angle = 0.000039;                 % Q_angle: accelerometer noise
R_angle = 20;                       % R_angle: high ==> gyroscope,
low ==> accelerometer

%=====Initialization=====
====

persistent initialize angle_roll bias1 P1 K1 phi angle_pitch bias2
P2 K2 theta angle_yaw;

if isempty(initialize)
    if isempty(angle_roll)
        angle_roll = 0;
        bias1 = 0;
        P1 = zeros (2);
        K1 = zeros ([2,1]);
```



```
phi = 0;
elseif(abs(ax)<0.01 && abs(ay)<0.01 && (abs(az-1)) <0.01 &&
p<1 && q<1 && r<1)
    angle_roll = 0;
end
if isempty(angle_pitch)
    angle_pitch = 0;
    bias2 = 0;
    P2 = zeros(2);
    K2 = zeros([2,1]);
    theta = 0;
elseif(abs(ax)<0.01 && abs(ay)<0.01 && (abs(az-1))<0.01 && p<1
&& q<1 && r<1)
    angle_pitch = 0;
end
if isempty(angle_yaw)
    angle_yaw = 0;
end
initialize = 1;
else
%=====Kalman
Filter=====

%-----Roll Angle-----
----

    Rate_Roll = p + (q * sin(phi) + r * cos(phi)) * tan(theta) -
bias1;
    angle_roll = angle_roll + Rate_Roll * delta_t;
%Projected State (integrated gyro data)
    earth_roll = -atan(ay / az) * degrees;
%Measurement Update (gravity vector)
    P1(1,1) = P1(1,1) + delta_t * (P1(2,2) * delta_t - P1(1,2) -
P1(2,1) + Q_angle); %Projected error covariance
    P1(1,2) = P1(1,2) - P1(2,2) * delta_t;
    P1(2,1) = P1(2,1) - P1(2,2) * delta_t;
    P1(2,2) = P1(2,2) + Q_gyroBias * delta_t;
    S1 = P1(1,1) + R_angle;
%Kalman Gain
    K1(1) = P1(1,1) / S1;
    K1(2) = P1(2,1) / S1;
    Y1 = earth_roll - angle_roll;
%Estimate Update
    angle_roll = angle_roll + K1(1) * Y1;
    bias1 = bias1 + K1(2) * Y1;
    P1(1,1) = P1(1,1) - K1(1) * P1(1,1);
%Update Error Covariance
```



```
P1(1,2) = P1(1,2) - K1(1) * P1(1,2);
P1(2,1) = P1(2,1) - K1(2) * P1(1,1);
P1(2,2) = P1(2,2) - K1(2) * P1(1,2);
phi = angle_roll * radians;

%-----Pitch Angle-----
%-----

Rate_Pitch = q * cos(phi) - r * sin(phi) - bias2;
angle_pitch = angle_pitch + Rate_Pitch * delta_t;
%Projected State (integrated gyro data)
earth_pitch = atan(ax / az) * degrees;
%Measurement Update (gravity vector)
P2(1,1) = P2(1,1) + delta_t * (P2(2,2) * delta_t - P2(1,2) -
P2(2,1) + Q_angle); %Projected error covariance
P2(1,2) = P2(1,2) - P2(2,2) * delta_t;
P2(2,1) = P2(2,1) - P2(2,2) * delta_t;
P2(2,2) = P2(2,2) + Q_gyroBias * delta_t;
S2 = P2(1,1) + R_angle;
%Kalman Gain
K2(1) = P2(1,1) / S2;
K2(2) = P2(2,1) / S2;
Y2 = earth_pitch - angle_pitch;
%Estimate Update
angle_pitch = angle_pitch + K2(1) * Y2;
bias2 = bias2 + K2(2) * Y2;
P2(1,1) = P2(1,1) - K2(1) * P2(1,1);
%Update Error Covariance
P2(1,2) = P2(1,2) - K2(1) * P2(1,2);
P2(2,1) = P2(2,1) - K2(2) * P2(1,1);
P2(2,2) = P2(2,2) - K2(2) * P2(1,2);
theta = angle_pitch * radians;

%-----Yaw Angle-----
%-----

angle_yaw = angle_yaw + (q * sin(phi) + r * cos(phi)) *
sec(theta) * delta_t; %Integrated gyro data

end

Roll_Angle = angle_roll;
Pitch_Angle = angle_pitch;
Yaw_Angle = angle_yaw;
```



1.2 FILTRO

Se realizaron dos filtros, uno de primer orden con el fin de no utilizar mucha potencia de cálculo, pero resultó no ser todo lo eficaz que se esperaba, por lo que se realizó uno de segundo orden. Este filtro tiene frecuencia de corte del orden del hercio.

```
%% Diseño filtro  
ts=0.02;  
s=tf('s');  
z=tf('z');  
wn=7*0.8;  
seta=.7;  
%P=1/(s/wn+1);  
P=1/((s/wn)^2+2*s/wn*seta+1);  
Pd=c2d(P, ts);
```

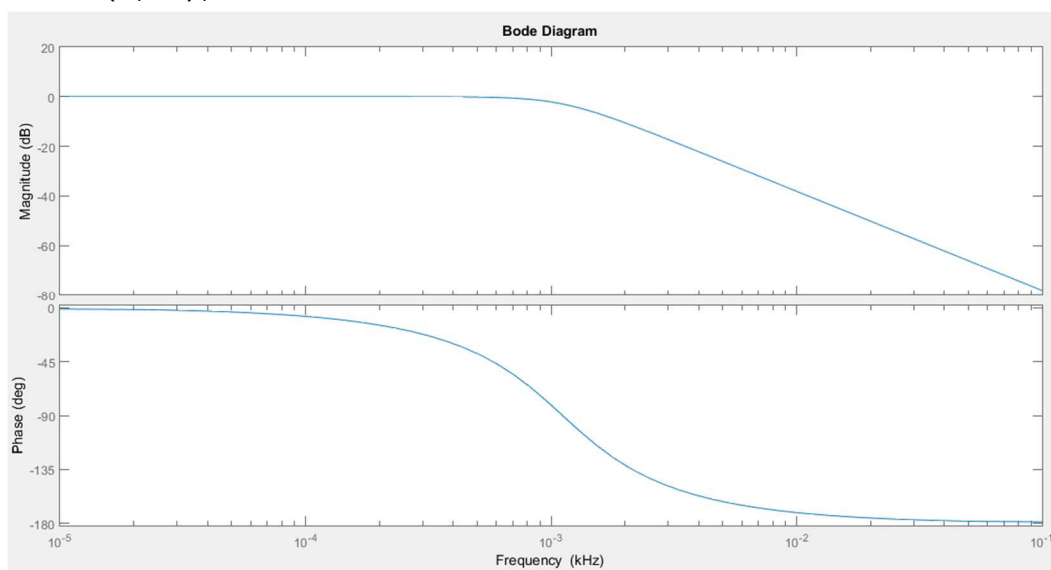


Ilustración 33: Diagrama de Bode del filtro

1.3 FUNCIONES MEMORIA FLASH

A continuación, se muestran las funciones para el registro de datos por la memoria flash. Estas funciones requieren introducir que puerto COM se está utilizando para poder comunicarse con el Ardupilot.



Función ByA, para borrar los datos y comenzar a guardar nuevos.

```
%% Borrar Datos
%Definir el puerto serie
sp=serial('COM4');

%Borrar memoria flash
fopen(sp);
set(sp, 'BaudRate', 115200, 'StopBits',1)
fprintf(sp, 'show attitude');

%Buscar la linea donde empiezan los datos
aux_int=[];
while isempty(aux_int)
    aux_str=fscanf(sp)
    if strfind(aux_str, '(p)')
        fprintf(sp, 'ey');
    end
    aux_int=strfind(aux_str, 'No data');
end

fclose(sp);

%% Armar Datos
%Definir el puerto serie
sp=serial('COM4');

%abrir el puerto serie
fopen(sp)

%Definir configuracion
set(sp, 'BaudRate', 115200, 'StopBits',1)

%Conexion con APM2
fprintf(sp, 'show attitude');
fscanf(sp)

%Petición de datos de la flash
fprintf(sp, 'a');

%Buscar la linea donde empiezan los datos
aux_int=[];
```



```
while isempty(aux_int)
    aux_str=fscanf(sp)
    aux_int=strfind(aux_str, 'Exit');
end
fclose(sp)

%Borrar variables intermedias
clear aux_str;
clear aux_int;
clear sp;
```

Función ArduData, para crear las variables con los datos guardados en Matlab.

```
%% ArduData

%Definir el puerto serie
sp=serial('COM4');

%abrir el puerto serie
fopen(sp);

%Definir configuracion
set(sp, 'BaudRate', 115200, 'StopBits',1)

%Conexion con APM2
fprintf(sp, 'show attitude');
fscanf(sp);

%Petición de datos de la flash
fprintf(sp, 'p');

%Buscar la linea donde empiezan los datos
aux_int=[];
while isempty(aux_int)
    aux_str=fscanf(sp);
    aux_int=strfind(aux_str, '(1) ');
end
fscanf(sp);

%Leer los nombres de las variables
read_var=fscanf(sp);
pos=regexp(read_var, ',');
```



```
num_var=length(pos)+1;
variables=cell(num_var,1);
variables{1}=read_var(1:pos(1)-1);
for nn=1:length(pos)-1
    variables{nn+1}=read_var(pos(nn)+1:pos(nn+1)-1);
end
variables{end}=read_var(pos(end)+1:end-2);

%Leer los datos. Tarda unos 20 segundos por cada 1000 muestras
aux_int=[];
cont=0;
while isempty(aux_int)
    aux_str=fscanf(sp);
    aux_int=strfind(aux_str, 'Y');
    if isempty(aux_int)
        cont=cont+1;
        num_var2=length(regexp(aux_str, ','))+1;
        %En las filas los instantes de tiempo y por columnas las
variables
        arduino(cont,:)=sscanf(aux_str,'%f,', num_var2)';
    end
    if mod(cont,100)==0
        cont
    end
end
% fclose(sp)
% plot(arduino);
% legend('d1','d2','l1','l2','h1','h2','bat')

%Generar tiempo
if exist('ts')==0
    ts=0.01;
end
t=[0:ts:ts*(cont-1)]';

%Asignar variables
if num_var2<=num_var
    for nn=1:num_var2
        aux_name=char(variables{nn});
        dat.(aux_name)=arduino(1:cont,nn);
        data.(aux_name)=[t,arduino(1:cont,nn)];
    end
end
```



```
%Cerrar puerto
fclose(sp);

%Borrar variables intermedias
if num_var==num_var2
    clear ardudata;
end
clear cont;
clear aux_str;
clear aux_int;
clear aux_name;
clear num_var;
clear nn;
clear sp;
clear pos;
clear num_var2;
clear variables;
```

1.4 FUNCIÓN PITOT

La función definida para calcular la velocidad del avión es la siguiente. Pese a que se considera improbable que el pitot mida valor negativo de velocidad, se incluyó en la función una condición en la que, si el valor medido era inferior al de velocidad 0, se considerase esta velocidad como nula.

```
%FuncionPitot
function y = fcn(u)
%Parametros experimentales
a=0.0049;
b=0.1228;
%Condición velocidad positiva
if u-520<0
    z=0;
else
    z=u-520;
end;
%Velocidad en km/h
y = (sqrt(4*a*z+b^2)-b)/(2*a);
```



1.5 CONTROL VELOCIDAD

En este apartado se muestra el código que generaba las variables para la implementación del control de velocidad. Los valores se ajustaron a base de realizar modificaciones sobre unos valores iniciales introducidos según se creía que funcionarían correctamente.

%% Control velocidad

```
Kvel=0.3;           %bajar K implica más amortiguamiento 0.5
Tivel=2;           %subir Ti más amortiguamiento 1
C=Kvel+Kvel/(Tivel*s);
Cd=c2d(C,ts,'zoh');
```

1.6 CONTROL DE ALTURA

A continuación, se muestra el fichero con que se diseñó el control de altura. En las últimas líneas de código se incluye los valores finales utilizados una vez realizados los ajustes realizado tras las comprobaciones.

%% Control Altura

% Diseño del control interno del control en cascada

```
s=tf('s');
Pint = -(6.966*s^3 - 42.82*s^2 + 2.951*s - 24.93)/(s^5 + 20.41*s^4
+ 71.57*s^3 + 21.47*s^2 + 53.58*s + 1.471e-10);
```

% Parametros del control

```
Fm=85.5;
```

```
wo = fsolve(@(w) 180 -Fm -180/pi*angle(freqresp(Pint,w)),1);
```

% Ganancia de la planta

```
Ap = abs(freqresp(Pint,wo));
Ac = 1/Ap;
Kint = Ac;
Kint =5;
```

% Diseño del control externo del control en cascada

```
Int=Pint*Kint/(1+Pint*Kint);
```



```
Pext=(0.1604*s+0.005576)/(s^2+0.008784*s+0.0007731);  
Ptot=Int*Pext;  
Kext=0.4;  
b=1.3;  
Kiext=0.01875;  
  
%versión 2 (mismo b)  
Kext=0.5;  
Kiext=0.015;  
  
Cext=Kiext/s;  
Cdext=c2d(Cext,ts);
```

1.7 CONTROL DE ROLL

A continuación, se muestra el código en el que se ejecutaban los valores de diseño del control de roll, que se obtuvieron mediante la aplicación PID Turner de Matlab.

```
%% Control Roll  
s=tf('s');  
Proll=(0.5654*s^4-1.253*s^3-0.3665*s^2-0.109*s-0.02788)  
/(s^5+5.734*s^4+2.492*s^3+0.5961*s^2+0.1827*s+0.007881);  
% valores del App  
Kroll=-0.4511;  
Tiroll=5;  
Croll=1/(Tiroll*s);  
Cdroll=c2d(Croll,ts);
```

1.8 FICHERO MAESTRO

Para evitar tener que ejecutar varios códigos de los anteriores se decidió crear un único fichero para los controles y funciones creadas por el proyectista. Este fichero cuenta con el diseño de todos los controles, así como el del filtro, la señal PRBS.

```
%% Diseño filtro  
ts=0.02;  
s=tf('s');
```



```
z=tf('z');
wn=7;
seta=.7;
P=1/((s/wn)^2+2*s/wn*seta+1);
%P=1/(s/wn+1);
Pd=c2d(P,ts);
bode(P);

%% señal para ensayos
u = idinput(310, 'prbs', [0,0.1], [-20,20]);

%% Control velocidad

Kvel=0.3;%bajar K implica mas amortiguamiento 0.5
Tivel=2;%subir Ti mas amortiguamiento 1
C=Kvel+Kvel/(Tivel*s);
Cd=c2d(C,ts, 'zoh');

%% Control Roll
s=tf('s');
Proll=(0.5654*s^4-1.253*s^3-0.3665*s^2-0.109*s-
0.02788)/(s^5+5.734*s^4+2.492*s^3+0.5961*s^2+0.1827*s+0.007881);
% valores del App
Kroll=-0.4511;
Tiroll=5;%10.4
Croll=1/(Tiroll*s);
Cdroll=c2d(Croll,ts);

%% Control Altura
% Diseño del control interno del control en cascada

s=tf('s');
Pint = -(6.966*s^3 - 42.82*s^2 + 2.951*s - 24.93)/(s^5 + 20.41*s^4
+ 71.57*s^3 + 21.47*s^2 + 53.58*s + 1.471e-10);

% Parametros del control
Fm=85.5;

wo = fsolve(@(w) 180 -Fm -180/pi*angle(freqresp(Pint,w)),1);

% Ganancia de la planta
Ap = abs(freqresp(Pint,wo));
Ac = 1/Ap;
Kint = Ac;
Kint =5;

% Diseño del control externo del control en cascada
Int=Pint*Kint/(1+Pint*Kint);
Pext=(0.1604*s+0.005576)/(s^2+0.008784*s+0.0007731);
Ptot=Int*Pext;
Kext=0.4;
b=1.3;
Kiext=0.01875;
```



```
%versión 2 (mismo b)  
Kext=0.5;  
Kiext=0.015;
```

```
Cext=Kiext/s;  
Cdext=c2d(Cext,ts);
```

1.9 DIAGRAMAS DE FOURIER

Para comprobar la validez de la señal PRBS creada se realizó un código para ver el diagrama de Fourier. El código fue el siguiente:

```
%% Diagrama de Fourier  
  
%Parámetros  
L=310;      %Longitud  
  
%Señal  
u = idinput(L, 'prbs',[0,0.1],[-20,20]);  
  
%transformada de Fourier  
Y=fft(u);  
P2=abs(Y/L);  
P1=P2(1:L/2+1);  
P1(2:end-1)=2*P1(2:end-1);  
f=L*(0:(L/2))/L;  
plot(f,P1)  
xlabel('f(Hz)')  
ylabel('|u|')
```



Capítulo 2 FICHEROS DE SIMULINK

Los ficheros de Simulink están organizados por subsistemas, en este primer apartado se muestran únicamente el nombre de los subsistemas utilizados para cada uno de los ficheros de manera que sea fácil identificar en cada fichero que subsistemas se utilizaron. Los subsistemas se muestran en el segundo apartado.

2.1 FICHEROS

2.1.1 ENSAYO PROFUNDIDAD

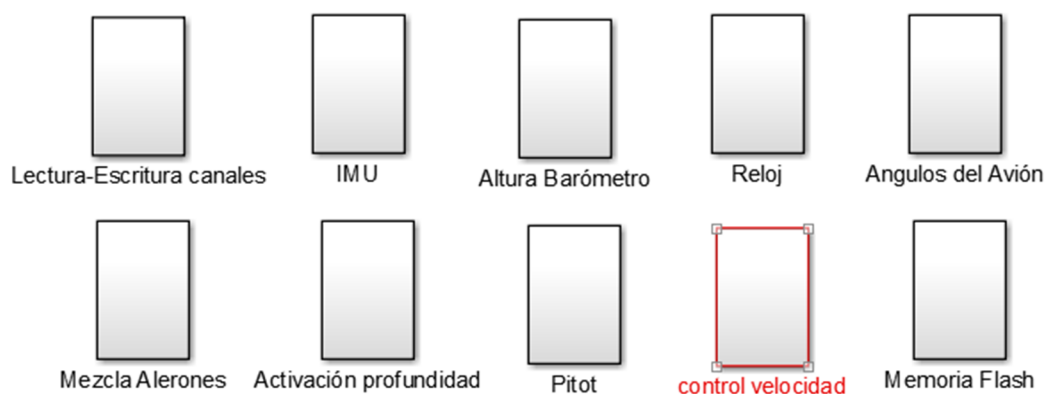


Ilustración 34: Simulink del ensayo de profundidad



2.1.2 ENSAYO ALABEO

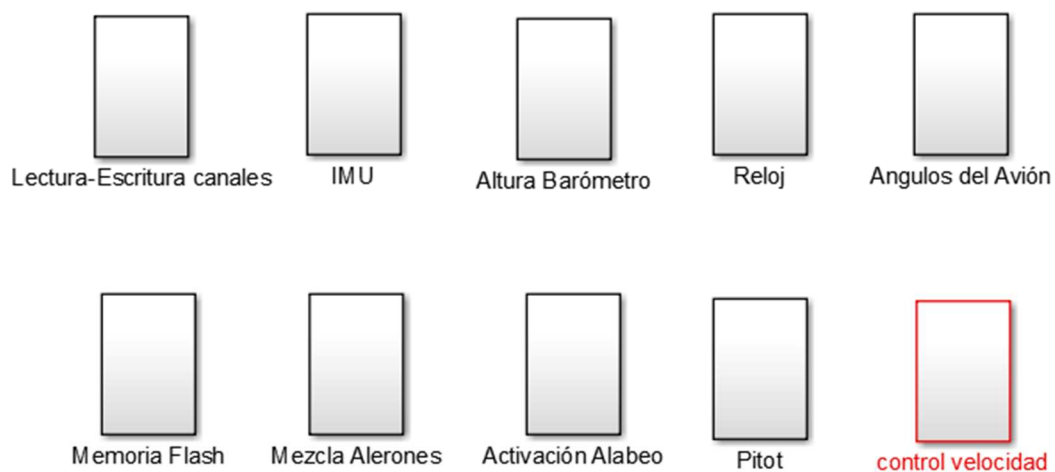


Ilustración 35: Simulink del ensayo de alabeo

2.1.3 CONTROL ALTURA

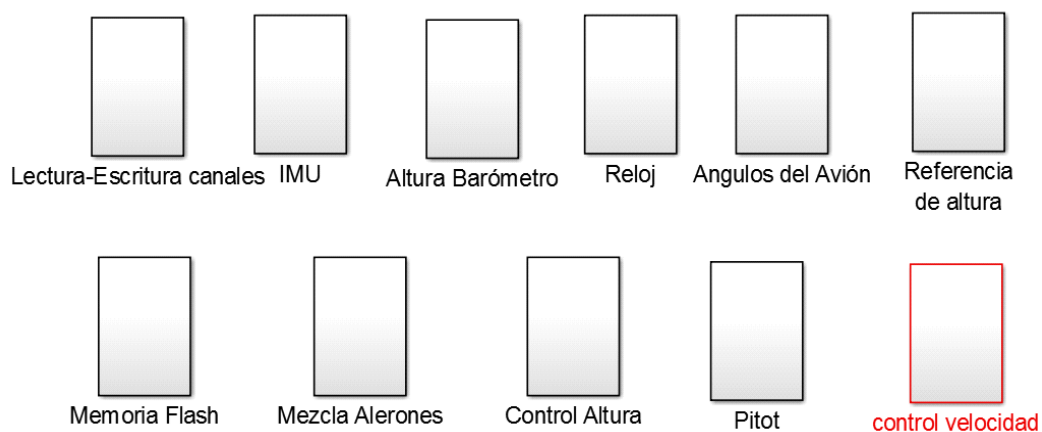


Ilustración 36: Simulink del control de altura



2.1.4 CONTROL ALABEO

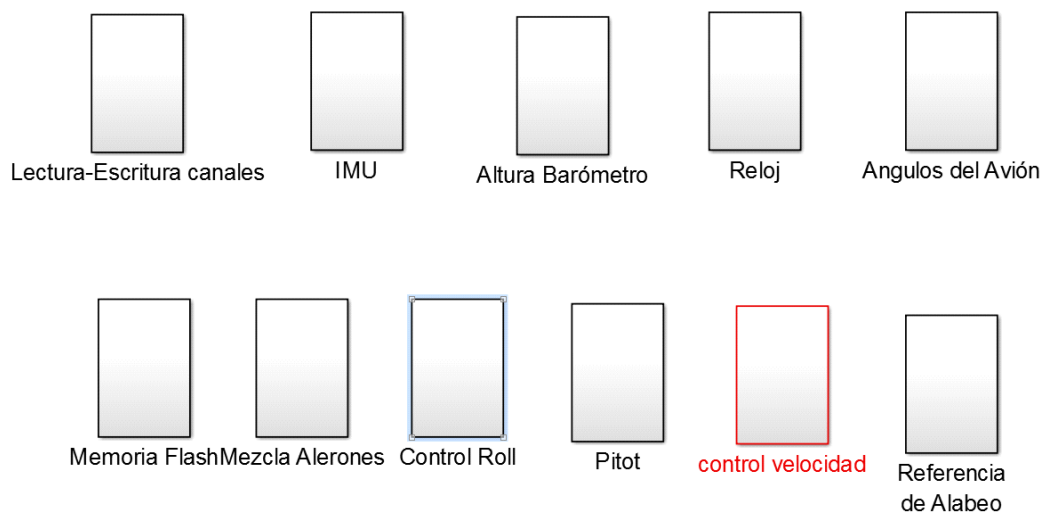


Ilustración 37: Simulink del control de alabeo

2.1.5 CONTROL COMPLETO

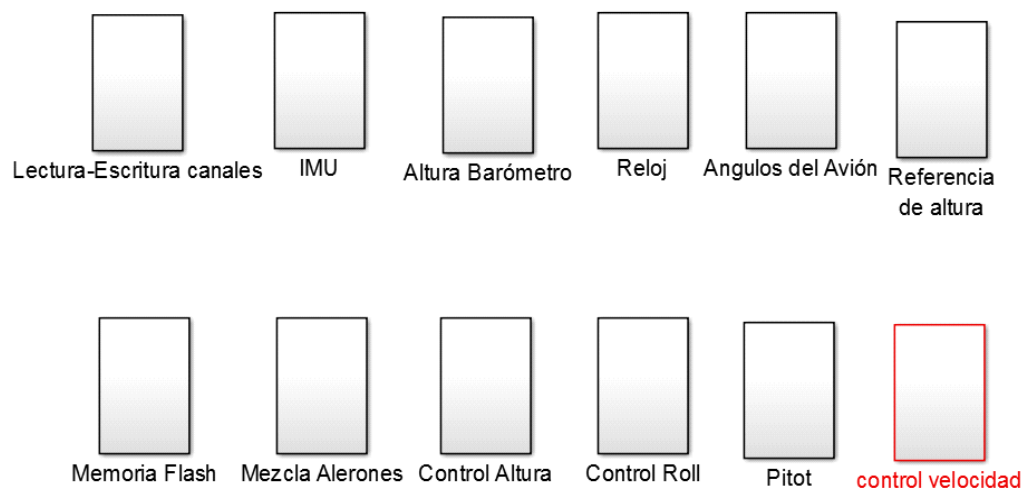


Ilustración 38: Simulink del control completo

2.2 SUBSISTEMAS

En este apartado se detallan cada uno de los subsistemas utilizados en los ficheros anteriores. Alguno de los ficheros de Simulink requirió pequeñas modificaciones en los subsistemas que se muestran a continuación. Todas las referencias que se hacen al filtro, se trata del filtro definido en Filtro.

2.2.1 LECTURA Y ESCRITURA DE CANALES

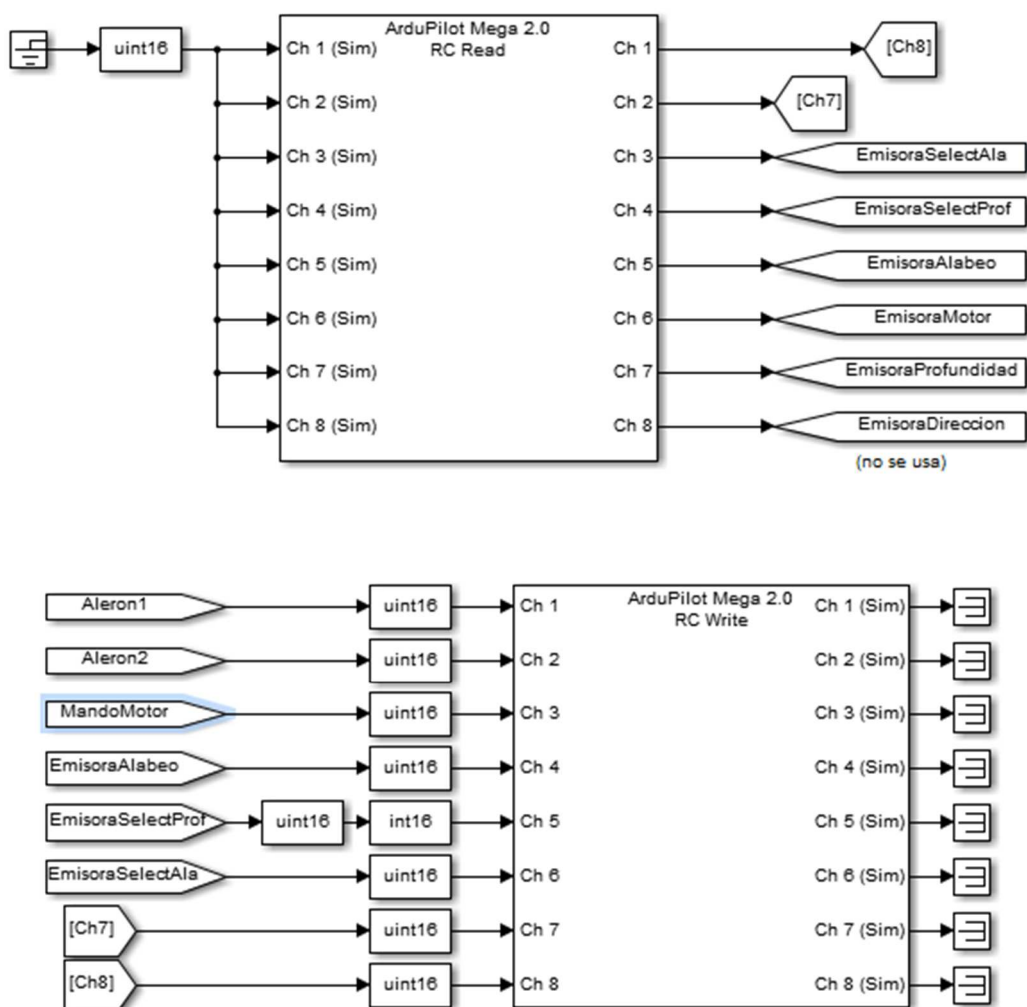


Ilustración 39: Subsistema lectura y escritura

2.2.2 IMU

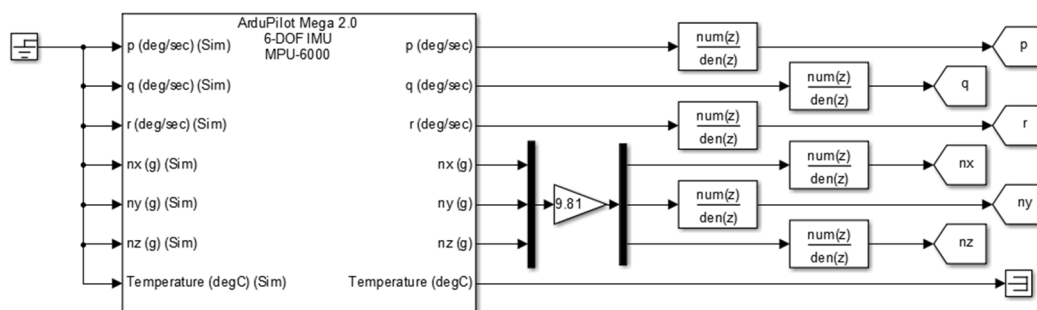


Ilustración 40: Subsistema IMU

2.2.3 ALTURA BARÓMETRO

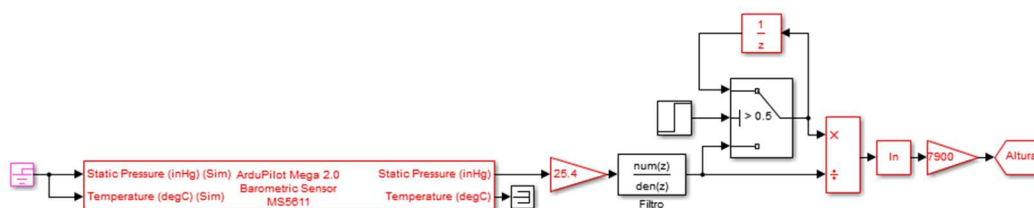


Ilustración 41: Subsistema altura barómetro

2.2.4 RELOJ

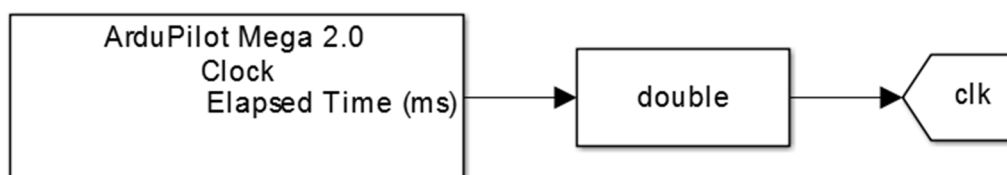


Ilustración 42: Subsistema reloj

2.2.5 ÁNGULOS AVIÓN

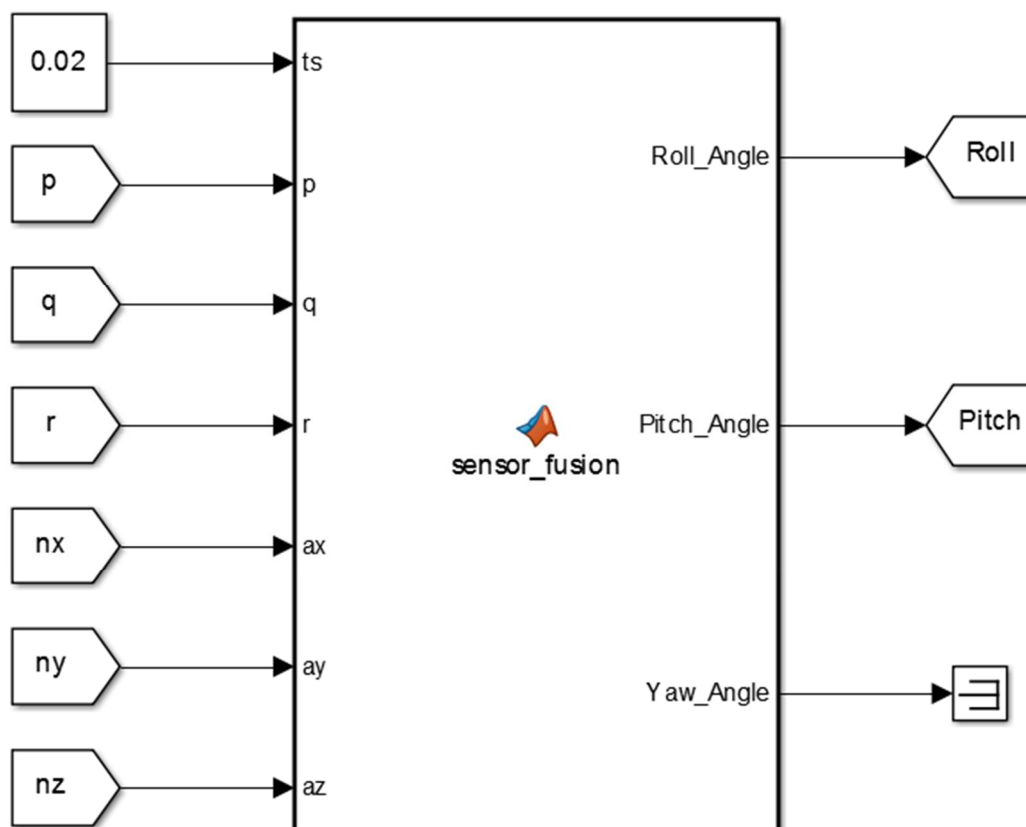


Ilustración 43: Subsistema ángulos avión

2.2.6 MEZCLA ALERONES

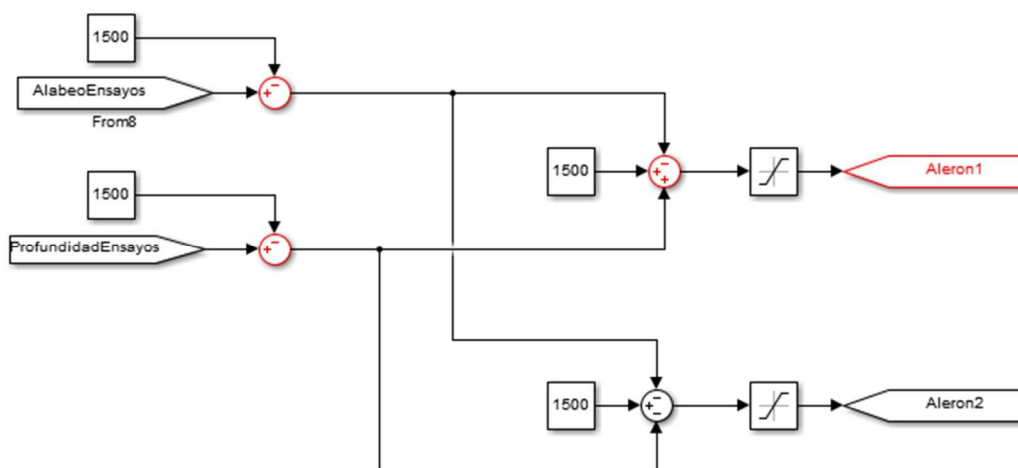


Ilustración 44: Subsistema mezcla alerones

Este subsistema tiene como función crear la mezcla de alerones que permite el vuelo del ala, dado que cada uno de los canales de profundidad afecta a los dos mismos servos.

2.2.7 PITOT

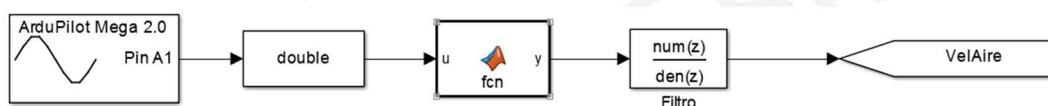


Ilustración 45: Subsistema pitot

Donde la función es la definida en Función Pitot.

2.2.8 ACTIVACIÓN PROFUNDIDAD/ALABEO

A continuación, se muestra el montaje de la activación de la señal PRBS en el canal de profundidad, para el de alabeo el montaje sería equivalente intercambiando la etiqueta *EmisoraProfundidad* por *EmisoraAlabeo*.

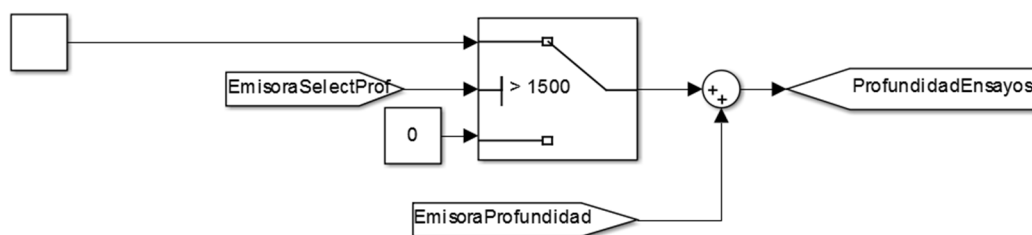


Ilustración 46: Subsistema activación profundidad

2.2.9 MEMORIA FLASH

La memoria flash se utilizó en todos los ensayos y en los vuelos de comprobación de controles, según cual fuese el tipo de vuelo realizado se guardaron aquellas variables que resultasen de interés en cada caso para evitar consumir la memoria de manera innecesaria.

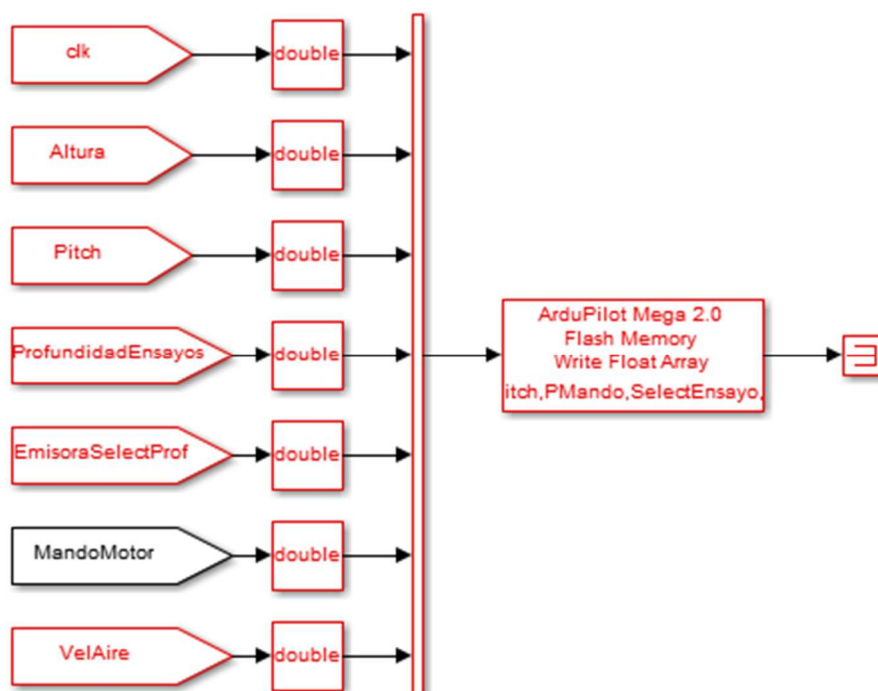


Ilustración 47: Subsistema memoria flash

2.2.10 REFERENCIA DE ALTURA

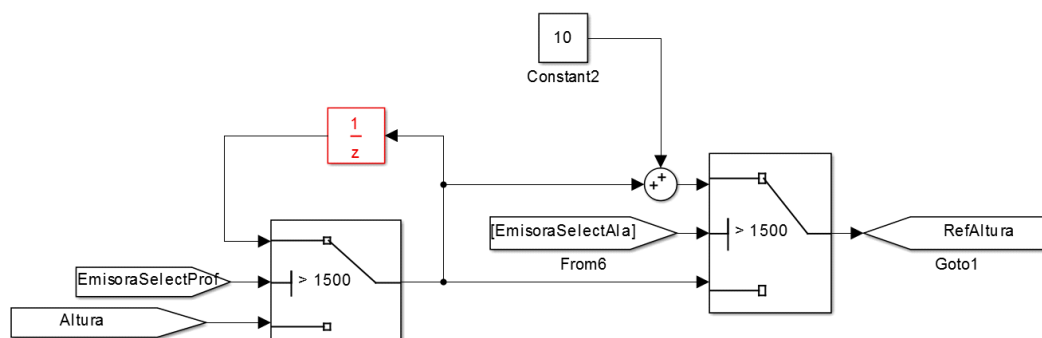


Ilustración 48: Subsistema referencia de altura

2.2.11 REFERENCIA DE ALABEO

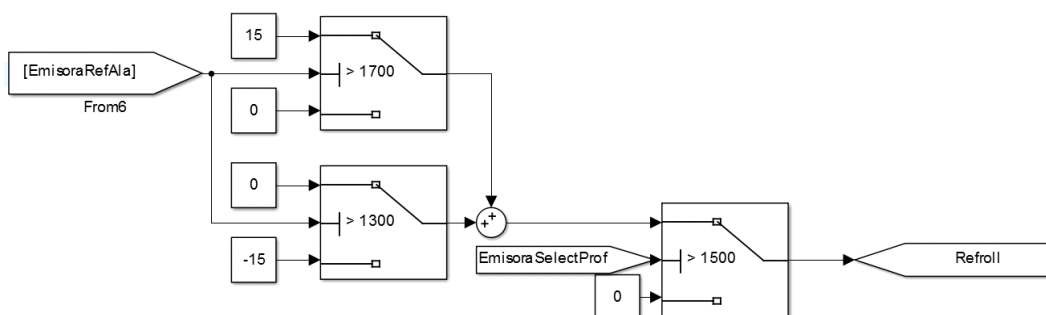


Ilustración 49: Subsistema referencia de alabeo

2.2.12 CONTROL VELOCIDAD

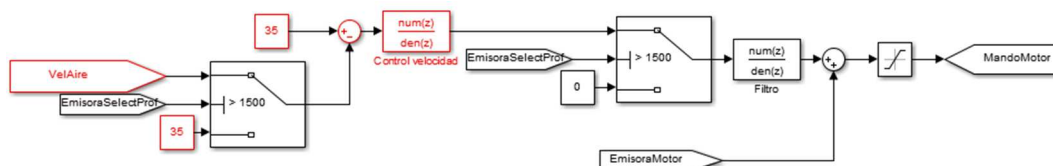


Ilustración 50: Subsistema control velocidad

Donde el bloque control velocidad es el definido en Control velocidad.

2.2.13 CONTROL ALTURA

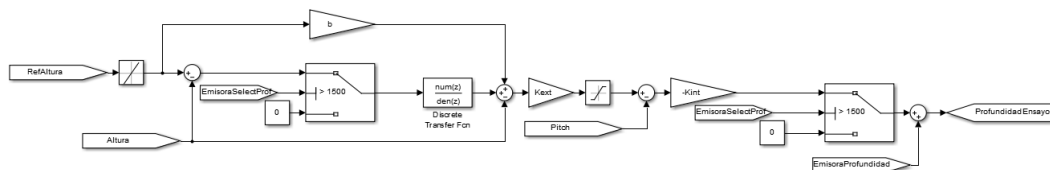


Ilustración 51: Subsistema control altura

Donde el bloque *Discrete Transfer Fcn* es Cdext y las ganancias definidas en Control de altura.

2.2.14 CONTROL ALABEO

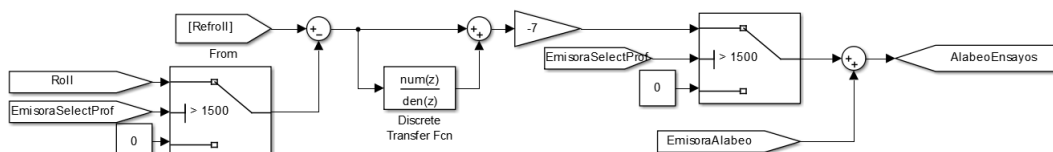


Ilustración 52: Subsistema control alabeo

Donde el bloque de nombre *Discrete Transfer Fcn* es la función definida como Cdroll en Control alabeo.



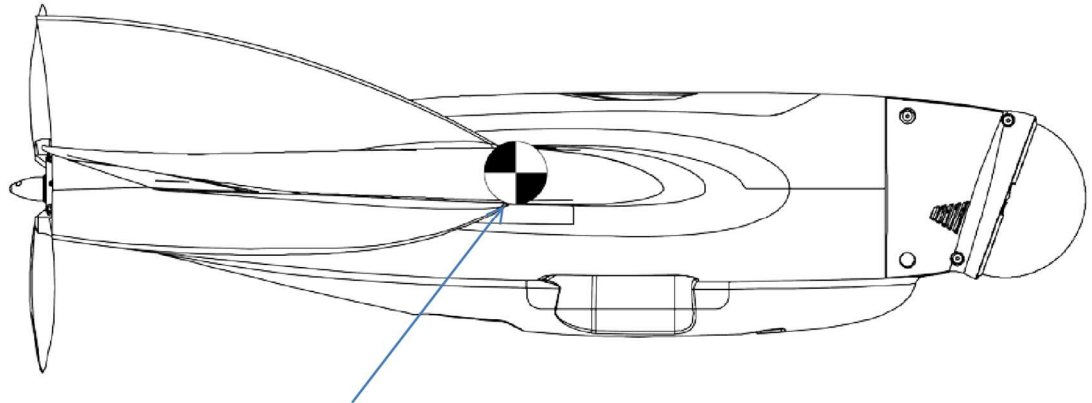
Parte V PLANOS



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



Capítulo 1 CENTRO DE GRAVEDAD



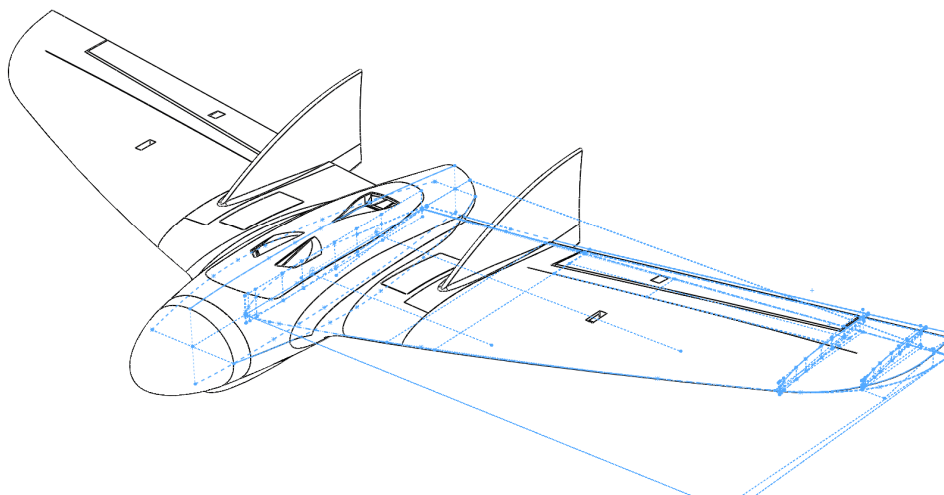
CG should be where the tip of the fins joins the wing.

Plano 1: Centro de gravedad del avión





Capítulo 2 VISTA TRIDIMENSIONAL

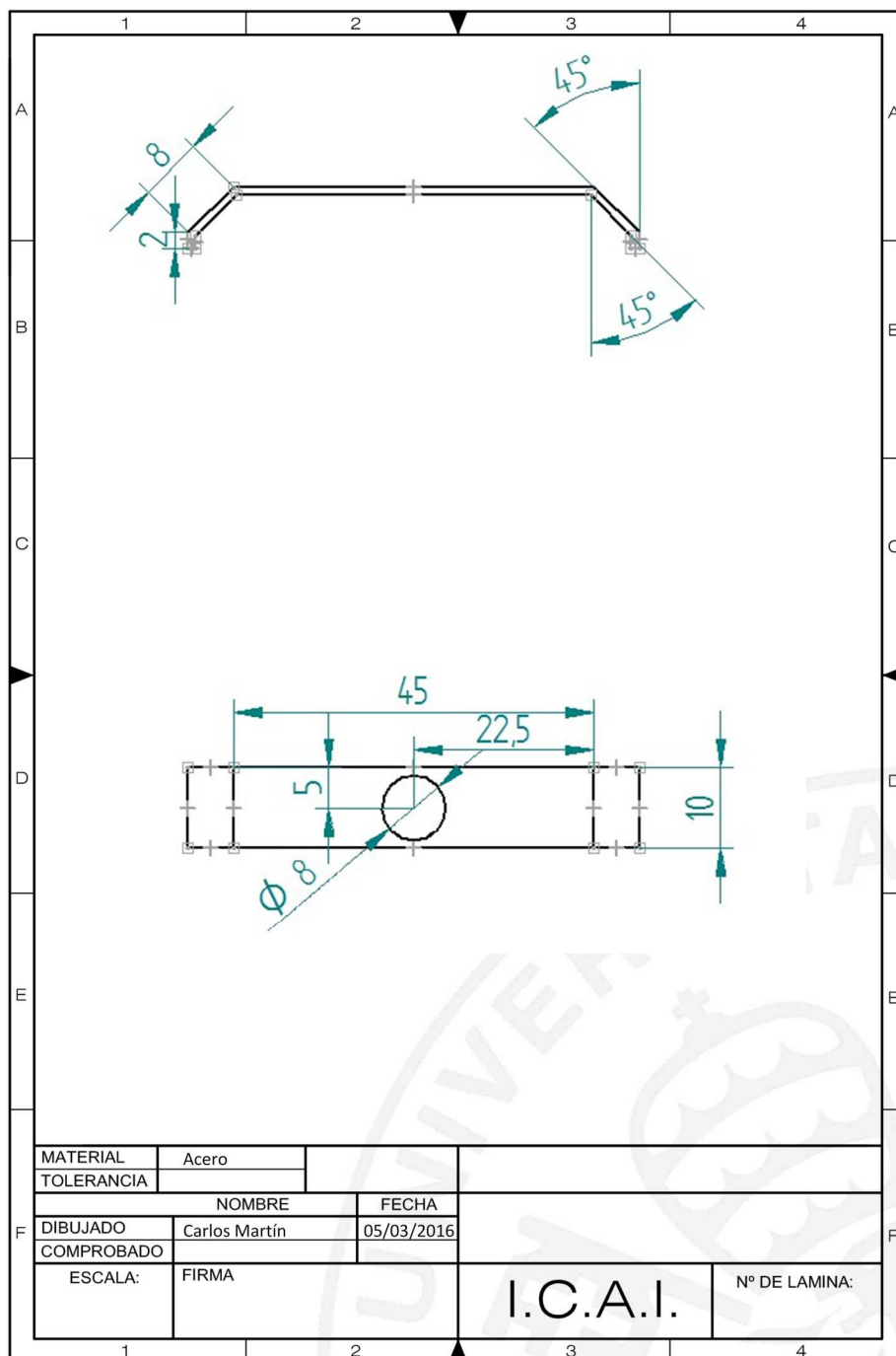


Plano 2: Vista tridimensional del avión





Capítulo 3 PIEZA HÉLICE



Plano 3: Pieza hélice





Parte VI PRESUPUESTO





Capítulo 1 COMPONENTES PRINCIPALES

Tabla 2: Presupuesto Componentes principales

Componente	Cantidad	Coste(€/ud)	Total (€)
Avión RC	1	150	150
Motor Brushless 900KV	1	20	20
Regulador electrónico de velocidad	1	15	15
Batería LiPO 8000 mAh	1	70	70
Hélice	2	5	10
Emisor+Receptor RC 2.4GHz	1	165	165
Placa reguladora de tensión	1	9	9
Buje	1	18	18
Ardupilot Mega 2.6	1	60	60
Subtotal		517	



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



Capítulo 2 EQUIPO Y HERRAMIENTAS

Tabla 3: Presupuesto Equipo y herramientas

Descripción	Cantidad	Coste(€/ud)	Total (€)
Ordenador	1	700	700
Cargador de baterías	1	75	75
Juego de herramientas, pegamento	1	50	50
Subtotal	825€		



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



Capítulo 3 SOFTWARE

Tabla 4: Presupuesto Software

Descripción	Cantidad	Coste(€/ud)	Total (€)
Matlab-Simulink	1	1.000	1.000
Putty*	1	0	0
Subtotal	1.000		

*El precio de este programa se debe a que se trata de un denominado *open source*, es decir que se distribuye y desarrolla libremente, es gratuito.



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



Capítulo 4 MANO DE OBRA Y DESPLAZAMIENTOS

Tabla 5: Presupuesto Mano de obra y desplazamientos

Descripción	Horas	Coste (€/h)	Total (€)
Diseño de los algoritmos de control	200	40	8.000
Montaje del avión	50	25	1.250
Documentación proyecto	50	20	1.000
Desplazamientos	20	1,50	30
Subtotal	10.280		





Capítulo 5 TOTAL

Capítulo 1. Componentes principales	517€
Capítulo 2. Equipo y herramientas	825€
Capítulo 3. Software	1.000€
Capítulo 4. Mano de obra y desplazamientos	10.280€
Presupuesto total:	12.622€



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



Parte VI DATASHEETS



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



Capítulo 1 AVIÓN

Modelo: RVJET

Marca: Range Video

Envergadura: 1.95 m

Fuselaje: 1.55 m

Manual:

<https://www.dropbox.com/s/mas62x5615aoa1g/RVJET%20Manual%20V2.pdf?dl>

=0



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



Capítulo 2 EMISORA

Modelo: Turnigy TGY-i10

Especificaciones:

Canales: 10

Rango de frecuencias: 2.4055-2.475GHz

Ancho de banda: 500kHz

Potencia de RF: menor de 20 dBm

Sistema de 2.4G: AFHDS 2A

Tipo de codificación: GFSK

Longitud de antena: 26 mm

Peso: 620 gramos

Capacidad de batería: 1700mAh

Dimensiones: 195 x 189 x 43 mm



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



Capítulo 3 RECEPTOR

Modelo: Turnigy iA10

Canales: 10

Frecuencia: 2.4GHz

Potencia RF: menor de 20 dBm

Alimentación: 4.0-6.5V

Intensidad de funcionamiento: 240 mA

Peso neto: 30 gramos

Dimensiones: 47 x 33 x 15 mm

Rango: más de 500 m



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



Capítulo 4 MOTOR

Modelo: D3548/5

Tipo: Brushless

Rpm/V=900kv

Voltaje: 11.1-18.5V

Peso: 159g

Vatios: 770W

Medidas: 35 x 48 mm

Amperaje máximo: 43 A

Eje: 5mm



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



Capítulo 5 ESC

Modelo: Gecko 65^a SBEC 8 A

Corriente de funcionamiento: 65 A

Corriente de pico: 85 A

Alimentación: 5-18 elementos NC

2-6 elementos Lipo

BEC output: 5-6-7.4-8.4V

Dimensiones: 33*69*11.5 mm

<http://www.rcgroups.com/forums/showatt.php?attachmentid=6652155>



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



Capítulo 6 PLACA DE PRESIÓN PARA PITOT

Modelo: V20 Pressure Board

Rango de presiones: -2 kPa a 2kPa

Alimentación: 4.75-5.25V

Menor que 10 mA

Offset: 2.5V

Máximo de escala: 4.5V

Máximo rango de escala: 4V

Precisión: $\pm 2.5\%$ del rango

Sensibilidad: 1V/kPa

Tiempo de respuesta: 1 ms

https://www.nxp.com/files/sensors/doc/data_sheet/MPXV7002.pdf



UNIVERSIDAD PONTIFICIA COMILLAS
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)
INGENIERO INDUSTRIAL



Capítulo 7 APM 2.6

Giróscopo de 3 ejes

Acelerómetro de 3 ejes

Barómetro de alta resolución

Módulo GPS de 5Hz

Dimensiones: 70.5 x 45 x 13.5 mm

Peso: 31 g