



Escola d'Enginyeria de Telecomunicació i
Aeroespacial de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

PROYECTO FINAL DE GRADO

TÍTULO DEL TFG: *Drone trajectory analysis*

TITULACIÓN: Grado en ingeniería de sistemas aeroespaciales

AUTOR: Pol González Marquès

DIRECTOR: Enric Pastor Llorens

FECHA: 23/11/2020

TÍTULO DEL TFG: *Drone trajectory analysis*

AUTOR: Pol González Marquès

DIRECTOR: Enric Pastor Llorens

FECHA: 23/11/2020

Compendio

En el marco de la aviación, determinar una ruta de vuelo óptima o disponer de información actualizada sobre el estado del sistema interno de la aeronave, son factores clave para mejorar la eficiencia y asegurar que los vuelos queden libres de incidencias.

Dado el contexto actual de los vehículos aéreos no tripulados (*UAV*), que se encuentra en pleno auge, se ha decidido dedicar este proyecto a desarrollar un algoritmo, a través de Excel-VBA, para monitorizar los parámetros de navegación y elementos internos del sistema. Concretamente se han querido monitorizar las *performances* de un hexacóptero y las fases de vuelo, para poder poner en contexto los demás parámetros que se analicen. La forma en que se ha querido proporcionar toda esta información es en forma de tablas y gráficos esencialmente.

Dicho estudio se ha llevado a cabo recopilando sets de datos de navegación para varios vuelos en un campo de aeromodelismo. Se ha tenido en cuenta que las condiciones para cada vuelo fueran prácticamente las mismas.

Finalmente, tras el proceso y monitorización de los datos, se ha podido comprobar que, el algoritmo identifica las fases y maniobras de forma adecuada.

TITLE: *Drone trajectory analysis*

AUTHOR: Pol González Marquès

DIRECTOR: Enric Pastor Llorens

DATE: 23/11/2020

Overview

Within the aviation framework, determining an optimal flight path or having up-to-date information on the status of the aircraft's internal system are key factors for improving efficiency and ensuring incident free flights.

Given the current rise of unmanned aerial vehicles, this project entails the development of an algorithm, by Excel-VBA, to monitor the navigation parameters and internal system elements. Specifically, we aim to monitor the performance of a hex-copter and the phases of flight, in order to be able to put into context the other parameters to be analyzed. This information is essentially provided with tables and graphics.

This study has been carried out by collecting sets of navigation data for several flights in a model airfield. It has been considered that the conditions for each flight were practically the same.

Finally, after processing and monitoring the data gathered, it is verified that the algorithm is capable to identify the phases and flight maneuvers adequately.

ÍNDICE

Introducción.....	1
Objetivos y metodología.....	2
Capítulo 1. Contexto de los drones y definiciones.....	4
1.1. Marco histórico.....	4
1.2. Marco actual.....	4
1.3. Concepto de drones.....	5
1.4. Sistemas de referencia.....	6
Capítulo 2. Sistema UAS utilizado para el estudio.....	8
2.1. Análisis del UAS empleado.....	8
2.2. Estación terrestre.....	8
2.3. Vehículo no tripulado.....	8
2.3.1. Esquema conceptual de los componentes dron.....	9
2.4. Enlace de comunicaciones.....	10
Capítulo 3. Protocolo de comunicaciones MavLink.....	12
3.1. Bases del protocolo.....	12
3.2. Formato del mensaje.....	12
3.3. Criterio para la elección de mensajes.....	13
Capítulo 4. Creación del algoritmo analizador de datos.....	15
4.1. Análisis de la estructura de datos origen.....	15
4.1.1. Reestructuración de los datos.....	16
4.2. Proceso del tiempo y <i>sampling rate</i>	16
4.2.1. Creación de un margen temporal del vuelo.....	17
4.2.2. Tiempo de muestreo.....	18
4.3. Automatización para la visualización gráfica de datos.....	19
4.3.1. Elección de gráficos.....	19
4.3.2. Aplicación de expresiones analíticas para la altitud.....	21
4.4. Analizador de fases de vuelo.....	24
4.4.1. Consideraciones para la identificación de fases de vuelo.....	24
4.4.2. Identificación de la transición de fases.....	25
4.5. Analizador de maniobras de vuelo.....	30
4.5.1. Asunciones previas para la identificación de maniobras de vuelo.....	31

4.5.2.	Identificación de maniobras de vuelo	32
4.6.	Formato para la presentación de datos en la identificación de fases y maniobras de vuelo.....	38
4.7.	Monitorización de datos adicionales de interés.....	39
Capítulo 5.	Resultados y discusiones	41
5.1.	Contexto de los dos planes de vuelo	41
5.1.1.	Vuelo 1	41
5.1.2.	Vuelo 2	42
5.2.	Resultados basados en los gráficos	43
5.2.1.	Tracking.....	43
5.2.2.	Accel. and attitude	47
5.2.3.	Troposphere behave	49
5.2.4.	Consumption analysis	52
5.3.	Resultados basados en identificación de fases y maniobras.....	55
5.3.1.	Sectorization and motion.....	55
Capítulo 6.	Desenlace del estudio.....	62
6.1.	Conclusiones.....	62
6.2.	Futuros estudios y mejoras.....	63
6.2.1.	Detectar deficiencia de dispositivos	63
6.2.2.	Detección de mayor consumo según maniobras	64
6.2.3.	Solución para mejorar la identificación de maniobras y fases de vuelo	64
Bibliografía.....		65

ANEXOS

ANEXO A. Elementos integrantes en la estación tierra.

ANEXO B. Elementos físicos en el vehículo no tripulado.

ANEXO C. Elección de mensajes *MavLink* de interés para las lecturas del vuelo.

ANEXO D. Definición de las performances que puede realizar un multicoptero.

ANEXO E. Representación de la opción 2 para la ruta de la figura 4.11.

ANEXO F. Instrucciones de las maniobras insertadas en MP para los vuelos 1 y 2.

ANEXO G. Grafico de posición del vuelo 1 en coordenadas geodesicas.

ANEXO H. Contenido de la pestaña *Accel. and attitude*.

ANEXO I. Contenido de la pestaña *Consumption Analysis*.

ANEXO J. Contenido de la pestaña *Troposphere behave*.

ANEXO K. Contenido de la pestaña *Tracking*.

ANEXO L. Contenido de la pestaña *Sectorization and motion*.

ANEXO M. Contenido de los scripts en *Visual Basic for Applications* para la realización del algoritmo procesador de datos.

ÍNDICE DE FIGURAS

Fig. 1.1 Clasificación de sistemas dron según su control.....	6
Fig. 1.2 Sistemas de referencia considerados para el estudio.....	7
Fig. 2.1 Conexiones entre <i>autopilot</i> y dispositivos que conforman nuestro UAV.....	9
Fig. 2.2 Esquema de la comunicación entre los componentes del sistema UAS	10
Fig. 3.1 Composición de un paquete MavLink.....	12
Fig. 4.1 Estructura de registros de datos (logs) en el archivo origen	16
Fig. 4.2 Ejemplo del envío de mensajes iniciados en tiempos distintos.....	17
Fig. 4.3 Ejemplo de columnas de tiempos según el mensaje #33 del GPS.....	18
Fig. 4.4 Marcas en rojo debidas al error de <i>SamplingTime</i> que sobrepasan el umbral.....	19
Fig. 4.5 Tendencia de la presión, temperatura y densidad del aire en la troposfera.	23
Fig. 4.6 Ejemplo de trayectoria vertical para el análisis de sectorización.	24
Fig. 4.7 Puntos dónde se capta movimiento ascendente por manejo del usuario.....	27
Fig. 4.8 Puntos en la fase de despegues, discrepancias para confirmar fin de la fase.	28
Fig. 4.9 Puntos en la fase de despegue (158s-167s) y aterrizaje (339s-370s).....	30
Fig. 4.10 Situación origen del radio inicial y disposición de puntos cardinales.....	32
Fig. 4.11 Trayectoria en 3D con las 3 fases principales dividido en 10 <i>waypoints</i>	33
Fig. 4.12 Limite en el eje Z a partir del cual se detecta un cambio de altura ya sea estático o dinámico. Los últimos instantes pueden acarrear un pequeño error debido a la desaceleración del movimiento.....	34
Fig. 4.13 Limite angular a partir del cual se detecta un giro ya sea estatico o dinámico... 36	36
Fig. 4.14 Limite radial a partir del cual se detecta un suceso movimiento en XY	38
Fig. 4.15 Ejemplo de sectorización de fases en forma de perfil vertical	38
Fig. 5.1 Diseño de la ruta del vuelo 1 mediante <i>waypoints</i>	41
Fig. 5.2 Diseño de la ruta del vuelo 2 mediante <i>waypoints</i>	42
Fig. 5.3 Tracking en el plano XY de la ruta del vuelo 1 y 2	45
Fig. 5.4 Perfil vertical para el vuelo 1 mediante distintos sensores.....	47
Fig. 5.5 Perfil vertical para el vuelo 2 mediante distintos sensores.....	47
Fig. 5.6 Cambios de aceleración y altitud en el eje Z, vuelo 1	48
Fig. 5.7 Cambios de aceleración y altitud en el eje Z, vqueo 2	49
Fig. 5.8 Variación de presión y altura vs el tiempo de los vuelos 1 y 2.....	49
Fig. 5.9 Variación de presión vs la altura de los vuelos 1 y 2.....	50
Fig. 5.10 Variación de temperatura y altura vs tiempo de los vuelos 1 y 2.....	51
Fig. 5.11 Variación de temperatura vs tiempo de los vuelos 1 y 2.....	52
Fig. 5.12 Variación de <i>battery remain</i> . Y voltaje vs tiempo de los vuelos 1 y 2	53
Fig. 5.13 Variación de corriente y voltaje vs tiempo de los vuelos 1 y 2.....	53
Fig. 5.14 Variación de aceleración y voltaje vs tiempo de los vuelos 1 y 2.....	54

Fig. 5.15 Correcta sectorización de fases.....	55
Fig. 5.16 Sectorización de fases, disminuyendo el valor Rang_Air.....	56
Fig. 5.17 Sectorización de fases, aumento del valor Vertical_tol_ground.....	57
Fig. 5.18 Ruta del vuelo 1 con la asignación de etiquetas de la tabla 5.5	59
Fig. 5.19 Ruta del vuelo 1 con la asignación de etiquetas de la tabla 5.6	60

ÍNDICE DE TABLAS

Tabla 3.1 Definición de las etiquetas de un paquete MavLink	13
Tabla 3.2 Relación entre sensores y mensajes de lectura correspondientes	13
Tabla 4.1 Variables ecuación troposfera estándar	22
Tabla 4.2 Variables ecuación hipsométrica	23
Tabla 4.3 Variables ecuación MRUA	34
Tabla 4.4 Segmentación de maniobras implicadas en el desplazamiento vertical	34
Tabla 4.5 Opción 1 para la segmentación de maniobras de rotación según la Fig. 4.11 ...	35
Tabla 4.6 Opción 1 para la segmentación de maniobras desplazamiento en XY	37
Tabla 4.7 Cabecera de los datos para la identificación de maniobras	39
Tabla 5.1 Exactitud en puntos clave de la ruta 1	43
Tabla 5.2 Exactitud en puntos clave de la ruta 1	44
Tabla 5.3 Exactitud en los segmentos de ascenso/descenso para el vuelo	46
Tabla 5.4 Exactitud en los segmentos de ascenso/descenso para el vuelo	46
Tabla 5.5 Resultados de la identificación de maniobras para el vuelo 1	59
Tabla 5.6 Resultados de la identificación de maniobras para el vuelo 2	60
Tabla 5.7 Resultados adicionales del vuelo 1	61
Tabla 5.8 Resultados adicionales del vuelo 2	61

Introducción

Cuando hablamos de navegación aérea, se suele hacer referencia a las técnicas y elementos que permiten la maniobrabilidad segura de una aeronave en relación con algunos parámetros del entorno exterior.

A día de hoy, el uso de la navegación ya no reside únicamente en dirigir un vuelo de forma segura, sino que también ha despertado interés en obtener datos de navegación para analizarlos y posteriormente hacer un mejor uso del espacio aéreo u optimizar las rutas de determinados vuelos [1] [2].

Una de las posibles formas de abordar el proceso de obtención de datos para la información de navegación, consiste en identificar las maniobras de vuelo que ha llevado a cabo un determinado vehículo. De esta forma se puede asegurar la eficiencia de las operaciones aéreas [3].

Por este motivo, en las últimas décadas, se ha invertido capital en el sector de la navegación [4], depositando más confianza en sistemas GNSS (*Global Navigation Satellite System*) o dispositivos INS (*Inertial Navigation System*), ya que el nivel de fiabilidad de las lecturas acaba siendo muy superior al de los órganos sensoriales de las personas u otros métodos de navegación.

Paralelamente, el uso de la navegación por instrumentos y satélites está arraigándose cada vez más a la nueva realidad de los UAV (*Unmanned aerial vehicle*) [5]. A pesar de que algunos aspectos de la navegación se han ido mejorando a lo largo de los años, otros puntos como el consumo de las baterías o la degradación de ciertos componentes ha pasado más desapercibido, pudiéndose mejorar notablemente la autonomía de un vuelo.

En este proyecto, se creará una herramienta capaz de analizar un set de parámetros relativos al vuelo de un UAV. Se hará mediante un algoritmo que tendrá como finalidad identificar las fases de vuelo y las maniobras empleadas en el mismo. Los resultados finales mostrarán ciertas tendencias a mejorar si se quiere optimizar la eficiencia del vuelo.

Objetivos y metodología

Se estudiará un modelo de *drone* propuesto por la escuela (hexacóptero) y se intentarán lograr los siguientes objetivos:

- Determinar si las maniobras de vuelo se efectúan adecuadamente según el *flight plan*, graficando los parámetros que caracterizan el vuelo.
- Ofrecer un programa capaz de automatizar el análisis de diferentes tipos de vuelos, haciendo distinción de las distintas fases de vuelo.
- Identificar y clasificar las diferentes maniobras que se empleen a lo largo del vuelo.

Estos objetivos tendrán la finalidad de predecir eventos de mantenimiento del propio *drone* y optimizar las maniobras de vuelo para llevar a cabo una determinada acción.

Una vez tengamos monitorizados y analizados los parámetros de varios planes de vuelo, haremos una valoración de los mismos para determinar si las maniobras que leemos se ajustan adecuadamente a la realidad y si la forma en que se efectúa es óptima en cuanto a navegación u otros parámetros.

Concretamente, para realizar los objetivos, realizaremos 3 procedimientos clave que nos permitirán lograr los objetivos propuestos.

→ Selección de mensajes MavLink

Si deseamos llevar a cabo el análisis del vuelo, primero deberemos tener claros qué parámetros de navegación analizar según el protocolo de mensajes MavLink.

Se pretende estandarizar la recepción de una serie mensajes, que por lo menos sean adecuados para la correcta monitorización del vehículo.

En el apartado 4.2. se comentarán los más destacados para el análisis.

→ Analizador de datos de vuelo (algoritmo VBA)

En este proyecto vamos a contemplar la posibilidad de crear un algoritmo, mediante VBA-Excel.

El algoritmo debe tener la capacidad de segmentar y procesar los datos aportados por los sensores del multicoptero . Para lograrlo hemos desarrollado dos partes:

- Proceso de datos y extracción de gráficos

Se hará un algoritmo capaz de interpretar la información desde un archivo origen. Se clasificarán y procesarán los datos que vayan a ser necesarios para el análisis del vuelo, como la altura y los cambios de presión entre otros y, finalmente, extraeremos los gráficos necesarios para realizar comparaciones.

- Segmentación de fases de vuelo e identificación de maniobras

Con la finalidad de saber en qué fase de vuelo se está ejecutando cada maniobra, crearemos una extensión del algoritmo que sea capaz de dividir 5 fases de vuelo según la lectura de la altitud. Esas fases deben ser fase tierra inicial, fase despegue, fase crucero, fase aterrizaje y fase tierra final.

Capítulo 1. Contexto de los drones y definiciones

1.1. Marco histórico

A lo largo de la historia de la aviación, han surgido diversas formas de navegación que han permitido asegurar con cierto margen de error la posición del vehículo en tiempo real.

No obstante, la aparición del GPS ha sido la más remarcable, que en conjunto con una unidad de medición inercial (navegación por estima) lograría la geolocalización de un usuario alrededor del mundo, con un error de 5 m aprox. para la población civil [6].

La nueva implementación de sistemas para *trackear* el recorrido de un vuelo despertó especial interés en el ámbito militar, donde en 1995 en EEUU, se construiría el RQ1-Predator (UAV) como uno de los primeros sistemas aéreos no tripulados usando la reciente tecnología GPS [7] .

Durante los años el concepto de “drone” fue tomando una nueva definición popular. Mientras que, hasta hace unas pocas décadas eran casi exclusivamente para el uso militar y con un *frame* de ala fija, ahora son más conocidos por ser de ala giratoria, mucho más compactos y al alcance de cualquier civil.

1.2. Marco actual

Actualmente, el interés por los drones se está extendiendo por todo el mundo gracias al amplio rango de posibilidades que ofrecen:

(Ventajas de un UAV de ala giratoria)

- Llegar a zonas de difícil acceso.
- No requiere la actuación de pilotos.
- Capacidad de vuelo estacionario.
- En caso de accidente no hay tripulación.
- Gran variedad de maniobras.
- Bajo coste de mantenimiento.

-Diseño pensado para la integración de la electrónica y mecanismos de propulsión.

-Versatilidad para incorporar dispositivos.

Gracias a estas ventajas son numerosos los sectores que pretenden aplicar esta tecnología en su ámbito. Entre otros encontramos empresas dedicadas al sector cartográfico, agrónomo, fotográfico, mensajería, vigilancia, I+D, etc.

A pesar de ello, y considerando que muchos drones se alimentan mediante baterías eléctricas, debemos ser conscientes de que actualmente las baterías representan el principal obstáculo en el sector [8]. Eso se debe al elevado consumo que requieren los propulsores para realizar maniobras tan sencillas como descensos.

Es por eso, que se intentará regular la agresividad de ciertas *performances* para disminuir el consumo eléctrico y optimizar tiempo de vuelo mediante el previo análisis del propio vuelo.

1.3. Concepto de drones

Cuando hablamos sobre *unmed vehicles*, podemos pensar en cualquier vehículo que tenga la capacidad de maniobrar sin necesidad de que haya nadie a bordo. En nuestro caso, el concepto de vehículo a estudiar es un *drone*, y por tanto podríamos hacer referencia a los términos UAS, UAV, RPAS y RPA.

El término UAV (*Unmanned aerial vehicle*) empezó a ser muy usado en temas militares, especialmente en Estados Unidos. En este caso, un UAV solo se refiere única y exclusivamente a la aeronave.

En Europa, en cambio, empezó a usarse el término UAS (*Unmanned Aerial System*), que también contempla una estación de tierra y un enlace de comunicaciones, y no únicamente al vehículo.

Por último, se puede diferenciar el hecho de que la aeronave la tripule un piloto a distancia, en vez de tener un piloto automático a bordo (como sería el caso del UAV). Pues en este caso, si lo pilota una persona a distancia nos referiremos al *drone* como RPA (Remoted Piloted Aircraft) o bien al sistema global RPAS (Remoted Piloted Aircraft System) si lo pilota un ordenador a bordo [9].

A modo esquema podemos representarlo del siguiente modo:

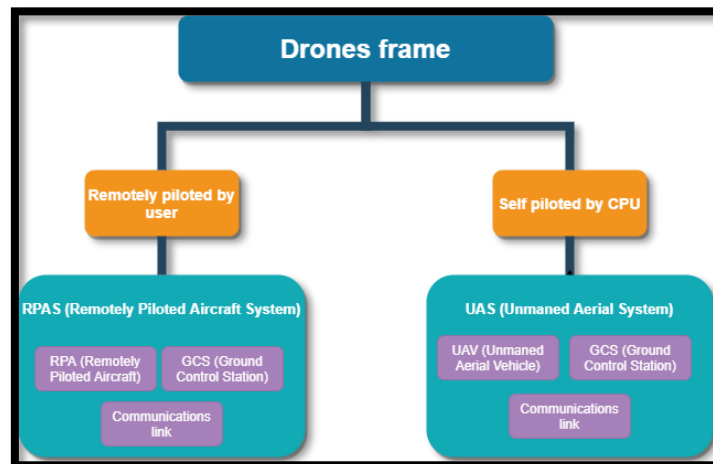


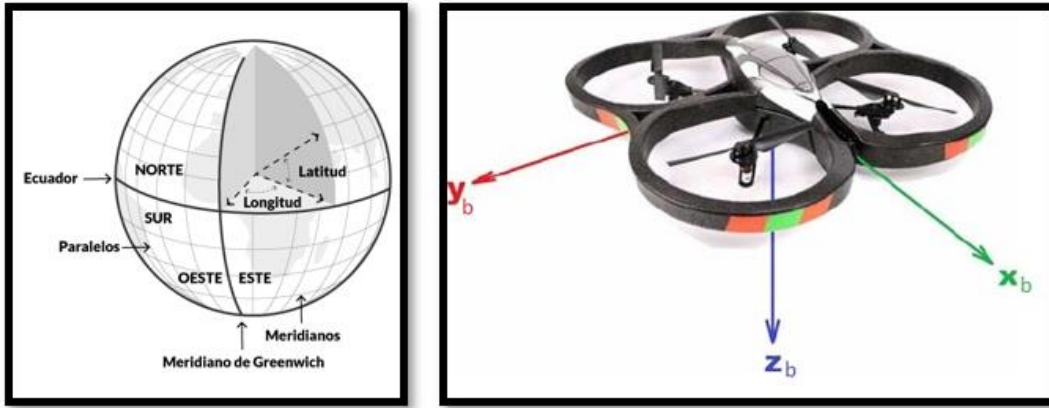
Fig. 1.1 Clasificación de sistemas dron según su control

1.4. Sistemas de referencia

Para los análisis que realizaremos en este proyecto vamos a tener que definir 2 tipos de sistemas de coordenadas que serán necesarios para referenciar correctamente la posición de nuestro UAV.

Por un lado, tenemos el sistema de coordenadas del GPS. Dicho sistema se basa en el modelo de sistema de referencia geodésico *WGS 84*, el cual tiene en cuenta la forma aproximadamente elipsoidal del planeta [10]. Este sistema no considera la orientación del cuerpo que quiere conocerse su ubicación, solamente obtenemos su posición en latitud, longitud y altura. Las componentes que definen el este sistema de coordenadas son la latitud, la longitud y altura (coordenadas esféricas) y tienen su origen en el plano del ecuador, el plano del meridiano de Greenwich y superficie terrestre, respectivamente. Véase en la *figura 2.2 (a)*

Por otro lado, tenemos el sistema de referencia cuerpo. Este sistema tiene como origen el centro de masas del cuerpo del *drone* en coordenadas cartesianas X , Y y Z . Su definición encaja, por ejemplo, con las lecturas de aceleración o velocidades angulares que nos ofrece la IMU (integrada en el interior del dron). Los ejes de este sistema son cartesianos y deben mirarse como si formaran parte de la estructura de la aeronave, independientemente de su movimiento. Finalmente, queremos añadir que los movimientos angulares *Roll*, *Pitch* y *Yaw* se dan alrededor de los ejes X_b , Y_b y Z_b respectivamente (*figura 2.2 (b)*).



(a) Sistema de coordenadas geográficas (b) Sistema de coordenadas cuerpo (*body frame*)

Fig. 1.2 Sistemas de referencia considerados para el estudio

Capítulo 2. Sistema UAS utilizado para el estudio

2.1. Análisis del UAS empleado

En este apartado tenemos como finalidad explicar el funcionamiento que tendrá cada componente del sistema UAS.

Este proyecto se ha llevado a cabo con un dron eléctrico con un *frame* de 6 hélices. También disponemos el software Mission Planer y un PC como estación de tierra, para monitorizar las acciones del vehículo e incluso cambiar el rumbo de la aeronave o redirigirla manualmente con un control a distancia si es necesario.

En la Fig. 2.2 podemos ver un diagrama el que aparecen los 3 componentes del sistema UAS están conectados.

2.2. Estación terrestre

La estación de tierra tiene como objetivo leer la telemetría proporcionada por el dron, así como hacer un seguimiento del *tracking* e incluso detectar errores y enviar instrucciones al vehículo. La estación de tierra, normalmente, también permite al usuario diseñar y crear una ruta (*flight plan*) que será llevado a cabo por el UAV.

Podemos ver, con cierto nivel de detalle, el conjunto de componentes que forman la estación terrestre de nuestro sistema UAS (ver [ANEXO A]).

2.3. Vehículo no tripulado

El UAV o dron será la plataforma que usaremos para emprender el vuelo y realizar todas las pruebas para el estudio. No se trata de ningún modelo comercial, más bien es un dron hecho a piezas para fines de investigación. En nuestro caso, muchos de los componentes ya han sido estudiados a nivel de compatibilidad en investigaciones anteriores realizadas en la escuela.

En este apartado comentaremos cada uno de los dispositivos que se integrarán en el *frame* y que, en conjunto, formarán un vehículo aéreo no tripulado capaz de comunicarse con nuestra estación tierra.

2.3.1. Esquema conceptual de los componentes dron

Es conveniente saber la arquitectura interna del UAV que empleemos para el análisis a fin de saber las limitaciones que uno tiene. Por este motivo, presentamos un mapa conceptual con la distribución de dispositivos electrónicos integrados en el vehículo.

El sistema gira entorno al piloto automático, que se encarga de recibir, procesar y/o enviar la información correspondiente a cada dispositivo que se encuentren conectados.

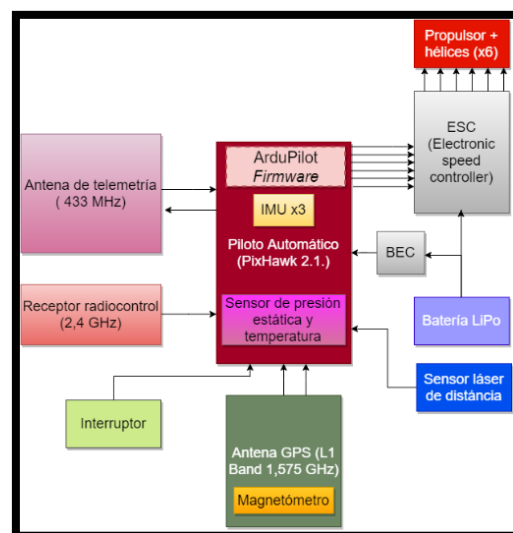


Fig. 2.1 Conexiones entre *autopilot* y dispositivos que conforman nuestro UAV

En el esquema, únicamente se representan las entradas y salidas de los flujos de datos. A excepción del caso de las baterías, donde se puede ver que la salida se bifurca a través de un ESC y un BEC (encargados de distribuir la cantidad de energía adecuada a cada receptor).

Se puede ver el listado de componentes explicados con detalle en el [ANEXO B].

2.4. Enlace de comunicaciones

En este apartado se mirarán el tipo de comunicaciones empleadas para que los sistemas Vehículo – Estación de Tierra – Constelación GPS, estén bien cohesionados. En primer lugar, es esencial mirar las frecuencias a las que trabaja cada sistema y evitar que se solapen.

Si tenemos en cuenta la disposición de nuestro sistema UAS vemos que hay comunicación **EMISOR – RECEPTOR** en los siguientes casos:

- **Satélites – UAV:** El satélite envía los datos de la posición del UAV al mismo mediante señales portadoras de la banda L1 (1,575 GHz).
- **Satélites – Estación de tierra** (no se aplica a nuestro caso): El satélite envía los datos de la posición de la estación de tierra a la misma mediante señales portadoras de la banda L1 (1,575 GHz).
- **Estación de tierra – UAV:** La estación de tierra puede mandar instrucciones al dron a través de la antena de telemetría a 433 MHz (dentro de Europa). También puede emplearse la emisora radiocontrol (2,4 GHz) para cambiar el modo de vuelo del dron y controlarlo manualmente.
- **UAV – Estación de tierra:** El dron envía los datos de telemetría a la estación de tierra a través de la antena de 433 MHz. Parte de los datos que envía son los proporcionados por la constelación de satélites.

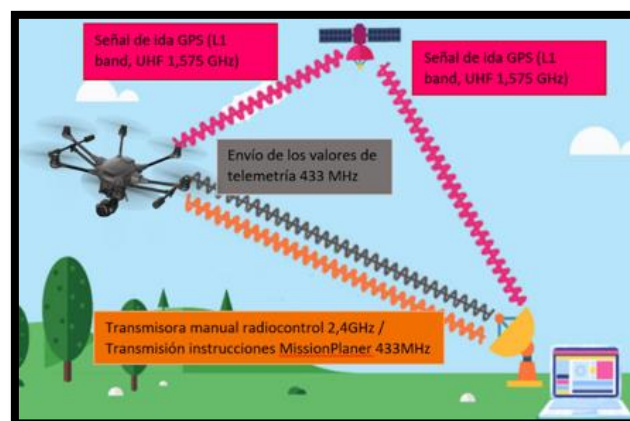


Fig. 2.2 Esquema de la comunicación entre los componentes del sistema UAS

Cabe remarcar la importancia de los radios de cobertura para la señal ya que actualmente, uno de los problemas que se están resolviendo es la posibilidad de hacer que el dron sea capaz de retornar a un punto de encuentro incluso cuando ha perdido la señal con la estación de tierra.

Si falla el enlace principal UAV-Estación de tierra/ “Estacion de Tierra – UAV” probablemente el dron no sepa qué hacer y deberíamos recurrir al modo manual. Por otro lado, si falla la comunicación GPS el sistema seguiría funcionando, pero maniobrando con errores cada vez más grandes debido a la deriva de medidas en la IMU y la falta de información en el filtro de Kalman.

En este apartado hemos planteado el GPS como un sistema de comunicaciones, aunque no es falso, es más correcto decir que es un sistema de posicionamiento ya que ese es el propósito final.

Capítulo 3. Protocolo de comunicaciones MavLink

3.1. Bases del protocolo

El envío de datos y comandos de cualquier componente dentro o fuera del dron, se llevará a cabo mediante el protocolo de comunicaciones MavLink (Micro Air Vehicle Link).

El protocolo define un rango de más de 200 mensajes. Los mensajes que vamos a usar pueden ser fácilmente identificables ya que disponen de un ID único, y debe ser el propio usuario quien defina los mensajes que vayan a leerse (si es que se permite el envío de dicho mensaje en nuestro sistema).

Cabe remarcar que cada lectura de mensaje tendrá como requisito disponer del *hardware* o periférico necesario para hacer la lectura que hemos demandado.

3.2. Formato del mensaje

La estructura genérica de los mensajes se compone de varios paquetes, cada paquete puede tener una extensión de entre 8 y 263 *bytes*. En cada uno de ellos hay 8 *labels* tal y como se ve en la figura 4.1 [11]:

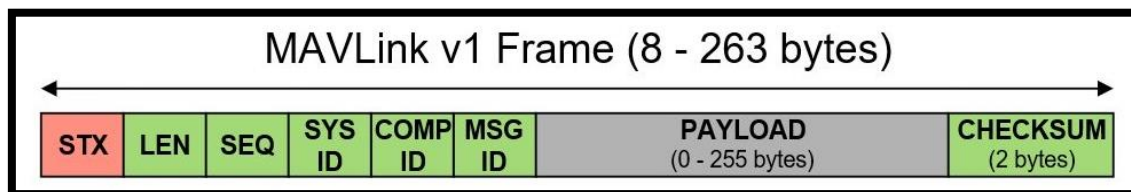


Fig. 3.1 Composición de un paquete MavLink

Tabla 3.1 Definición de las etiquetas de un paquete MavLink

Nombre etiqueta	Nombre completo	Definición de la etiqueta
STX	Start Tx	Marca identificativa inicio del paquete
LEN	Length	Peso del <i>payload</i> del paquete
SEQ	Sequence Number	Numero de secuencia del paquete, único para cada mensaje
SYS ID	System ID	ID del sistema UAS
COMP ID	Component ID	ID del componente periférico / sensor que envía los datos
MSG ID	Message ID	ID del mensaje del cual forma parte el paquete
PAYLOAD	Payload	Contenido de nuestro paquete
CHECKSUM	Byte error	Comprobación del correcto envío del paquete

3.3. Criterio para la elección de mensajes

Para escoger los mensajes con los que queremos trabajar, debemos ser conscientes de la metodología que vamos a aplicar en este proyecto para llevar a cabo los objetivos. Ya que, principalmente nos centramos en el análisis de la navegación, será interesante hacer uso de las lecturas que puedan ofrecernos todos los sensores y dispositivos integraos en nuestro dron. De este modo, tendremos diversidad de variables con las que hacer comparativas.

Por tanto, el factor limitante a la hora de escoger los mensajes será la cantidad de sensores y dispositivos que estén al equipados en nuestro UAV. El listado de dispositivos asociados al mensaje de sus lecturas se muestra en la tabla 4.2,

Tabla 3.2 Relación entre sensores y mensajes de lectura correspondientes

Disponibilidad de dispositivos	Mensaje asociado
Módulo GPS	#33 – Global position Int
Sensor 1 de presión estática	#29 – Scaled Pressure_1
Sensor 2 de presión estática	#137 – Scaled Pressure_2
Sensor 1 de temperatura	#29 – Scaled Pressure_1
Sensor 2 de temperatura	#137 – Scaled Pressure_2
IMU 1	#26 – Scaled IMU_2
IMU 2	#116 – Scaled IMU_2
IMU 3	#129 – Scaled IMU_3
Sensor de distancia laser	#132 – Distance sensor

Cabe destacar que, a parte de los mensajes de navegación, también resultará importante tener información el estado del dron y baterías. Puede resultar útil para asegurarnos de que las consideraciones que hace el piloto automático respecto al *frame* del vehículo, modo de vuelo y estatus del vehículo son los adecuados en todo momento. También nos serán de ayuda los mensajes de la actitud del vehículo que no es más que los datos del giróscopo procesados para obtener los ángulos Pitch, Roll y Yaw.

Las numerosas variables que contiene cada mensaje que recibimos quedan explicadas en el ANEXO C.

A parte de los mensajes de la tabla 3.2, también disponemos de algunos mensajes que hemos tenido en cuenta, pero que no han sido tan relevantes para monitorizar la navegación (ver en [ANEXO C]).

Capítulo 4. Creación del algoritmo analizador de datos

En este capítulo explicaremos las partes más destacables de nuestro código en *Visual Basic for Applications*, que tendrá como principal objetivo graficar de manera automática el comportamiento de los parámetros de navegación, e interpretar las fases de vuelo en función de las maniobras realizadas por el dron.

Para la construcción de este código, pondremos especial interés en hacer que el algoritmo sea “inteligente”. Es decir, tendrá que ser capaz identificar cada mensaje, saber cuántos mensajes de interés para la navegación hay disponibles y además saber qué *plots* nos puede proporcionar con las variables que dispone.

El programa también tendrá que ser capaz de proporcionar el *sampling ratio* de cada mensaje como parámetro adicional, hacer la transformación adecuada de un margen temporal (ya que las lecturas temporales se miden en UNIX) y , siendo menos importante, estructurar y mostrar todos los datos con un aspecto fácil de manejar, en forma de tablas y gráficos.

4.1. Análisis de la estructura de datos origen

Para preparar una forma de análisis automatizada, primero hay que extraer una muestra de *logs* de la placa controladora para ver de qué forma se presentan los datos.

Normalmente se registran de forma que puede parecer aleatoria debido al funcionamiento del procesador del *autopilot*, que en un instante de tiempo puede haber enviado varios mensajes “a la vez”.

A lo largo de todo el vuelo, por cada segundo, se envía un *set* de N mensajes de tipos distintos (por cada ciclo el orden de los mensajes puede ser distinto). Los tipos de mensajes representan la salida de información de los sensores con los cuales se haya establecido comunicación.

No obstante, siempre hay una cierta probabilidad de que los paquetes sufran un error y un mensaje no se envíe, en este caso se vuela a solicitar el envío de paquetes actualizado.

El archivo origen de datos es un fichero con una serie de filas de mensajes según vemos en la Fig. 4.1.

```

mHeartbeat date timestamp type autopilot base_mode custom_mode system_status mavlink_version
mGlobalPositionInt date timestamp latitude longitude altitude relativeAltitude vx vy vz heading
mGPSRawInt date timestamp fixtype latitude longitude altitude hdop vdop hspeed course satInView altitudeOnigs84 hAccuracy
mScaledIMU datetimesamp x_lin_acc y_lin_acc z_lin_acc x_ang_acc y_ang_acc z_ang_acc x_mag y_mag z_mag
mScaledPressure date timestamp pressure_abs pressure_diff temperature
mAltitude date timestamp altitude_mon altitude_amsl altitude_local altitude_relative altitude_terrain bottom_clearance
mRollPitchYaw date timestamp roll pitch yaw rollSpeed pitchSpeed yawSpeed
mHeartbeat 1,58047E+12 0 HEXAROTOR ARDUPILOTMEGA { CUSTOM_MODE_ENABLED GUIDED_ENABLED STABILIZE_ENABLED MANUAL_INPUT } COPTER_MODE_LOITER STANDBY 0
mAltitude 1,58047E+12 0 0,004804772 -0,001606021 0,04246006 0,001528462 6,39E-04 0,42E-04
mGlobalPositionInt 1,58047E+12 0 41,28767 1,3308624 248,75 248,75 0,05 -0,05 -0,01 2,43
mScaledIMU 1,58047E+12 0 -0,006 -0,009 -0,995 0,006 -0,003 0 0,249 -0,031 0,18
mScaledPressure 1,58047E+12 0 992,18585 0 28,44
mGPSRawInt 1,58047E+12 0 GPS_FIX_TYPE_ID_FIX 41,28767 1,3308594 248,82 73 85 0 0 14 0 0 0 0 0
mHeartbeat 1,58047E+12 1,58047E+12 HEXAROTOR ARDUPILOTMEGA { CUSTOM_MODE_ENABLED GUIDED_ENABLED STABILIZE_ENABLED MANUAL_INPUT } COPTER_MODE_LOITER STANDBY
mAltitude 1,58047E+12 1,58047E+12 0,005057183 -0,00111611 0,042560373 -4,12E-04 0,001297803 -7,52E-04
mGlobalPositionInt 1,58047E+12 1,58047E+12 41,28767 1,3308611 248,78 248,78 0,02 -0,04 -0,02 2,43
mScaledIMU 1,58047E+12 1,58047E+12 -0,005 0,001 -0,996 0,006 -0,002 0,001 0,251 -0,028 0,179

```

Fig. 4.1 Estructura de registros de datos (logs) en el archivo origen

Cuando se abre el documento contenedor de los datos del registro de vuelo, en este caso en formato tipo `.txt` obtenemos numerosas filas y columnas de distintos mensajes tomados en tiempos distintos.

Podemos interpretar estos datos como una serie de filas, en cada una de ellas hay un mensaje, y por cada mensaje encontramos su nombre y los parámetros que se aporten (tiempo, posición, temperatura, etc). Cada tipo de mensaje tiene una extensión propia.

4.1.1. Reestructuración de los datos

Ahora que ya sabemos cómo recibimos los datos, vamos a reordenarlos de manera más sencilla, con la finalidad de simplificar el código que se encargue de realizar las comparativas.

Crearemos un archivo Excel con tantas hojas distintas como tipos de mensajes tengamos. En cada hoja se imprime el listado de valores de un determinado mensaje.

4.2. Proceso del tiempo y *sampling rate*

Cuando obtenemos el tiempo de cada mensaje, lo recibimos en medidas UNIX. A fin de poder comparar cualquier parámetro con un margen temporal más común, lo ideal sería tener el instante 0 como referencia del inicio del vuelo.

Vemos que todos acaban las lecturas al final de vuelo, pero sin embargo cada uno empieza a leerse en distintos momentos. Por este motivo podemos decir que el inicio del vuelo empieza a considerarse en el instante que se recibe el primer mensaje, cualquier mensaje posterior se entiende como un offset temporal respecto del primer mensaje.

4.2.1. Creación de un margen temporal del vuelo

Viendo que cada mensaje empieza en instantes distintos, podríamos extraer 2 tiempos diferentes para cada mensaje:

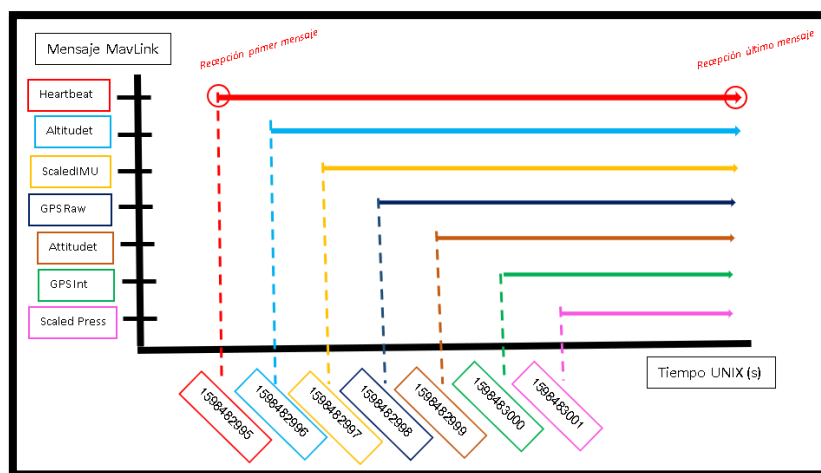


Fig. 4.2 Ejemplo del envío de mensajes iniciados en tiempos distintos

-TimeStamp: Recibiendo el tiempo en medidas UNIX, nos guardamos el primer valor del primer mensaje que recibamos y lo sustraemos a los N siguientes valores de todas las muestras de cada mensaje. Así conseguimos pasar el tiempo a segundos referenciando el inicio en 0.

Con el eje temporal de “TimeStamp” aseguramos un tiempo común para todas las muestras.

-Local Time: Siempre tendremos un primer instante igual a 0 para cada tipo de mensaje, ya que no hay *offset*. Por tanto, puede existir un desfase en la lectura de los mensajes.

-Date: Se trata del tiempo original, en segundos, que recibimos sin ser procesado. Los tiempos UNIX tienen como referencia el primer instante en las 0 horas del 1 de Enero de 1970.

Tiempos que contiene cada mensaje

date	timeStamp [s]	timeLocal [s]
1,60025E+12	0,982	0
1,60025E+12	1,98	0,998
1,60025E+12	2,98	1,998
1,60025E+12	3,98	2,998
1,60025E+12	4,981	3,999
1,60025E+12	5,981	4,999
1,60025E+12	6,981	5,999
1,60025E+12	7,98	6,998
1,60025E+12	8,98	7,998
1,60025E+12	9,981	8,999
1,60025E+12	10,981	9,999
1,60025E+12	11,98	10,998
1,60025E+12	12,982	12
1,60025E+12	13,984	13,002
1,60025E+12	14,982	14
1,60025E+12	15,982	15
1,60025E+12	16,987	16,005
1,60025E+12	17,982	17
1,60025E+12	18,982	18
1,60025E+12	19,981	18,999
1,60025E+12	20,981	19,999

Fig. 4.3 Ejemplo de columnas de tiempos según el mensaje #33 del GPS.

4.2.2. Tiempo de muestreo

El período de recepción entre 2 mensajes del mismo tipo será de 1 segundo aproximadamente. Pero dado que este factor puede variar en algunos momentos del vuelo, hemos declarado una tolerancia del 10%.

La idea es que la base de datos de Excel muestre en rojo si esa tolerancia está siendo vulnerada, y en caso de que lo esté, no tener en cuenta dicha muestra para futuros gráficos e inspecciones.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	mScaledIMU	datetimeStamp	timeLocal [s]	x_lin_acc	y_lin_acc	z_lin_acc	x_ang_acc	y_ang_acc	z_ang_acc	x_mag	y_mag	z_mag	SamplingTime [s]
110	mScaledIMU	108,455	107,967	1,58047E+12	-0,076	-0,041	-0,867	-0,056	-0,002	-0,007	-0,034	-0,259	0,931
111	mScaledIMU	109,427	108,939	1,58047E+12	-0,083	0,08	-0,717	0,011	0,018	-0,014	-0,039	-0,254	0,972
112	mScaledIMU	110,519	110,031	1,58047E+12	0,09	0,177	-1,119	-0,01	0,036	-0,009	-0,043	-0,246	1,092
113	mScaledIMU	111,451	110,963	1,58047E+12	0,004	0,203	-1,009	-0,05	0,024	0,064	-0,046	-0,24	0,932
114	mScaledIMU	112,515	112,027	1,58047E+12	-0,193	0,097	-0,934	0,121	0,026	-0,026	-0,033	-0,25	1,064
115	mScaledIMU	113,515	113,027	1,58047E+12	-0,116	-0,089	-0,895	-0,021	-0,016	0,045	-0,054	-0,231	1
116	mScaledIMU	114,449	113,961	1,58047E+12	0,093	0,076	-0,991	-0,062	0,123	0,017	-0,046	-0,24	0,934
117	mScaledIMU	115,497	115,009	1,58047E+12	-0,096	0,177	-0,981	-0,058	-0,028	0,034	-0,035	-0,249	1,048
118	mScaledIMU	116,523	116,035	1,58047E+12	-0,046	0,06	-0,972	-0,024	0,032	0,017	-0,039	-0,244	1,026
119	mScaledIMU	117,598	117,11	1,58047E+12	-0,094	0,038	-0,892	0,055	-0,007	0,065	-0,042	-0,249	1,075
120	mScaledIMU	118,495	118,007	1,58047E+12	-0,089	0,064	-1,014	-0,016	0,012	-0,007	-0,044	-0,235	0,897
121	mScaledIMU	119,547	119,059	1,58047E+12	-0,077	0,201	-1,539	-0,044	0,007	-0,009	-0,047	-0,239	1,052
122	mScaledIMU	120,395	119,907	1,58047E+12	-0,061	-0,003	-1,048	-0,031	0,027	0,008	-0,04	-0,232	0,848
123	mScaledIMU	121,522	121,034	1,58047E+12	-0,073	-0,192	-0,711	0,02	-0,051	0,015	-0,04	-0,233	1,127
124	mScaledIMU	122,452	121,964	1,58047E+12	-0,039	0,096	-1,02	-0,011	0,024	0,013	-0,037	-0,23	0,93
125	mScaledIMU	123,398	122,91	1,58047E+12	-0,085	0,06	-0,861	0,035	-0,022	0,005	-0,04	-0,244	0,946
126	mScaledIMU	124,528	124,04	1,58047E+12	-0,138	-0,025	-0,934	0,049	-0,029	0,001	-0,031	-0,242	1,13
127	mScaledIMU	125,456	124,968	1,58047E+12	-0,146	0,009	-0,806	0,085	-0,022	0,018	-0,03	-0,237	0,928
128	mScaledIMU	126,499	126,011	1,58047E+12	-0,19	-0,013	-0,943	0,033	-0,037	0,026	-0,052	-0,239	1,043
129	mScaledIMU	127,463	126,975	1,58047E+12	-0,169	0,086	-1,083	-0,019	0,012	0,021	-0,035	-0,228	0,964
130	mScaledIMU	128,419	127,931	1,58047E+12	-0,067	0,07	-1,054	-0,049	-0,02	0,001	-0,033	-0,229	0,956
131	mScaledIMU	129,559	129,071	1,58047E+12	-0,028	0,051	-1,077	-0,012	0,004	-0,018	-0,03	-0,249	1,14
132	mScaledIMU	130,461	129,973	1,58047E+12	-0,07	0,136	-0,954	-0,043	-0,031	0,027	-0,044	-0,247	0,902
133	mScaledIMU	131,479	130,991	1,58047E+12	-0,022	0,003	-1,159	-0,022	0,028	-0,039	-0,036	-0,242	1,018
134	mScaledIMU	132,52	132,032	1,58047E+12	-0,191	0,027	-0,782	0,016	0,014	0,059	-0,04	-0,248	1,041
135	mScaledIMU	133,481	132,993	1,58047E+12	-0,21	0,033	-0,977	-0,015	-0,085	-0,024	-0,048	-0,241	0,961

Fig. 4.4 Marcas en rojo debidas al error de *SamplingTime* que sobrepasan el umbral especificado [0,9s-1,1s]

En el caso de la figura 5.4 tenemos mensajes relacionados con las lecturas de la IMU_2, y en concreto, hay varias lecturas casi consecutivas que sufren una pequeña variación significativa del *sampling rate*.

4.3. Automatización para la visualización gráfica de datos

El siguiente paso para construir el programa que llevará a cabo el proceso de datos, es la creación de los distintos *plots*. Esos *plots* pueden interesarnos para hacer un análisis cualitativo de los parámetros de navegación.

4.3.1. Elección de gráficos

La idea es graficar los parámetros de principal interés, mostrados en la Tabla 3.2 y graficarlos de distintas maneras y comparándolos con otros parámetros para sacar la mayor cantidad de información.

Las series de gráficos se presentarán en las 5 últimas hojas del libro. Y concretamente, la última hoja contendrá una gráfica y una tabla como resultado

de los algoritmos que se encargan de identificar tanto las maniobras en el vuelo como las fases de vuelo. El nombre de esas familias de gráficos son *Tracking, Acceleration and Attitude, Troposphere behave, Consumption analysis y Sectorization of phases*.

4.3.1.1. Gráficos para el análisis de la posición

En primer lugar, vamos a mirar las variaciones de altitud a lo largo del tiempo. A diferencia de la posición en el plano transversal (plano del dron paralelo al suelo), tenemos múltiples dispositivos que nos permitirán ver, con más o menos exactitud, la altura en cada instante de tiempo. Los dispositivos que pueden llegar a ofrecernos la posición vertical son todos los que ya hemos visto en la *tabla 4.2*.

Cabe remarcar que, para determinar la altitud a partir de la presión y temperatura, se han utilizado las expresiones matemáticas (4.1), (4.2) que relacionan el termino correspondiente de presión y temperatura con la altitud. Podemos ver dichas expresiones en la sección 4.3.2.

Por otro lado, miraremos cómo ha sido el desplazamiento transversal en el tiempo. Se ofrecerá la posición tanto en unidades de longitud-latitud, como en X-Y (coordenadas cartesianas). Para este caso solamente determinaremos dichas variables con el dispositivo GPS.

Llamaremos *tracking* a la hoja que contenga los gráficos relacionados con la posición del *drone* durante el vuelo.

4.3.1.2. Gráficos para el análisis de la actitud y aceleración

Se mostrará el comportamiento de la aceleración en los 3 ejes espaciales y la actitud en los ángulos *Pitch* y *Roll*.

Puede ser interesante graficar los cambios de aceleración en el eje vertical, mientras también vemos la variación de altitud producida a lo largo de todo el vuelo, todo en el mismo gráfico. Para la creación de este gráfico solamente necesitaremos las lecturas del acelerómetro y la posición en altura proporcionada por el GPS.

Se plotearán los cambios de aceleración en el plano horizontal de nuestro sistema de coordenadas. En el mismo *plot* también quisiéramos mostrar las

variaciones de los ángulos *Pitch* y *Roll*, a lo largo del vuelo. Estos datos nos los proporciona el giróscopo integrado en la IMU.

4.3.1.3. *Gráficos para un análisis troposférico simple*

Ver la relación que hay entre la presión estática y temperatura en función de la altura, podría justificar una lectura inesperada de algún valor en los sensores, dados los diferentes fenómenos atmosféricos que pueden ocurrir en las primeras capas de la atmósfera [12].

Por ejemplo, podríamos ser capaces de percibir un fenómeno de inversión térmica al ver que las lecturas del sensor de temperatura son opuestas a las del modelo estándar de la troposfera, tal y como podemos ver en la Fig. 4.5.

Llamaremos *Troposphere behave* a la hoja que contenga los gráficos de presión y temperatura, relacionadas con la troposfera.

4.3.1.4. *Gráficos para el consumo de la batería*

A pesar de no disponer de una batería inteligente, se han conseguido leer los valores de voltaje y corriente en cada momento del vuelo. Por tanto, se podrán graficar dichos valores para tener una idea aproximada del ratio de consumo tanto en pequeños periodos de interés, como del vuelo en global.

Además, se podrá comparar ese consumo con las mismas lecturas de aceleración y comprobar que durante movimientos más bruscos el voltaje debería caer más rápido.

Consumption análisis será el nombre de la hoja que contenga los gráficos relacionados con el consumo de las baterías.

4.3.2. **Aplicación de expresiones analíticas para la altitud**

Será conveniente la conversión o el proceso de algunas variables para ofrecer ciertos resultados de interés.

Cabe remarcar que, los datos para las condiciones iniciales de las expresiones (4.2) y (4.3) se han tomado a partir del mensaje #33, que hace referencia a los valores filtrados del GPS.

→ Modelo Altitud por Presión (Modelo atmosfera estándar internacional) [13]

$$h_i = \frac{\theta}{\alpha} \cdot \left[1 - \left(\frac{p_i}{p_0} \right)^{\left(\frac{R \cdot \alpha}{g} \right)} \right] \quad (4.1)$$

Donde:

Tabla 4.1 Variables ecuación troposfera estándar

Variables	Definición
h_i	Altitud en el instante i (m)
α	Gradiente térmico (6,5 K/Km)
θ	Temperatura SL (288,15 K)
g	Gravedad (9,806 m/s ²)
p_i	Presión en el instante i (Pa)
p_0	Presión SL (Pa)
R	Constante gases del aire (287,053 J·K/Kg)

Esta ecuación nos va a permitir tener una aproximación de la altura únicamente con la lectura de los valores de presión y siempre y cuando el dron se encuentre dentro de la troposfera.

De todas formas, cabe remarcar que puede presentar más discrepancias con la realidad cuando estamos muy cerca de la superficie terrestre ya que pueden ocurrir fenómenos como la inversión térmica o la turbulencia que no están contemplados en esta ecuación [14].

→ Modelo Altitud por Temperatura y Presión (ecuación hipsométrica) [12]

$$h_i = h_{i-1} + \frac{R \cdot \bar{T}}{g} \cdot \ln \left(\frac{p_{i-1}}{p_i} \right) \quad (4.2)$$

Donde:

Tabla 4.2 Variables ecuación hipsométrica

Variable	Definición
h	Altitud en el instante i (m)
h_{i-1}	Altitud en el instante $i-1$ (m)
\bar{T}	Variación entre temperatura en los instantes i e $i-1$ (K)
g	Gravedad a nivel del mar ($9,806 \text{ m/s}^2$)
p_i	presión en el instante i (Pa)
p_{i-1}	Presión en el instante $i-1$ (Pa)
R	Constante gases del aire ($287,053 \text{ J}\cdot\text{K}/\text{Kg}$)

La Fig. 5.10 nos muestra cómo debería variar la presión, temperatura y densidad del aire a lo largo de los primeros 20 Km de troposfera y tropopausa [12].

Remarcamos que, en cuanto a estas dos últimas ecuaciones, no esperamos tener valores con un nivel muy alto de exactitud, ya que volamos a bajas alturas (entre 10 y 300 metros) y las lecturas pueden tender a un mayor error por fenómenos inesperados.

A pesar de ello, las muestras recogidas servirán para tener una referencia más en nuestras comparativas.

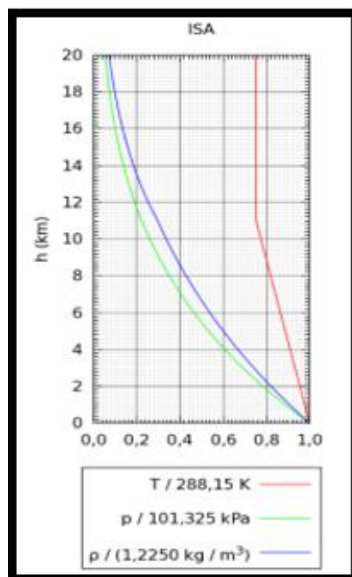


Fig. 4.5 Tendencia de la presión, temperatura y densidad del aire a lo largo de la troposfera.

4.4. Analizador de fases de vuelo

Parte de este programa, ha estado dedicado a crear un código capaz de analizar y segmentar nuestro vuelo en 5 partes distintas: fase tierra inicial (Initial Ground), despegue (Take Off), crucero (Cruise), aterrizaje (Land) y fase tierra final (Final Ground).

El estudio para lograr identificar cada fase se centrará en las observaciones del movimiento vertical, ya que, uno de los principales factores a tener en cuenta en el emprendimiento de un vuelo es el cambio de altura.

En concreto, los datos que vamos a analizar son los valores de altitud ofrecidos por el GPS, ya que es la forma de navegación más fiable, ya que, sus valores han sido resultado de una combinación de parámetros dentro del propio filtro de Kalman [15], ejecutado en el *autopilot*.

4.4.1. Consideraciones para la identificación de fases de vuelo.

¿cómo identificamos un despegue y un aterrizaje en un dron?

Hoy, el mundo de los UAV todavía está en desarrollo, especialmente por parte de leyes y normativas; no existe una forma estandarizada única de despegar o aterrizar en un vuelo.

Por eso, limitaremos la forma más simple de efectuar dichas acciones, que en un primer instante tienen que ver con el cambio de altura a lo largo del eje Z. Viendo cómo se efectúa esa variación podremos tener indicios sobre una transición de fase del vuelo.

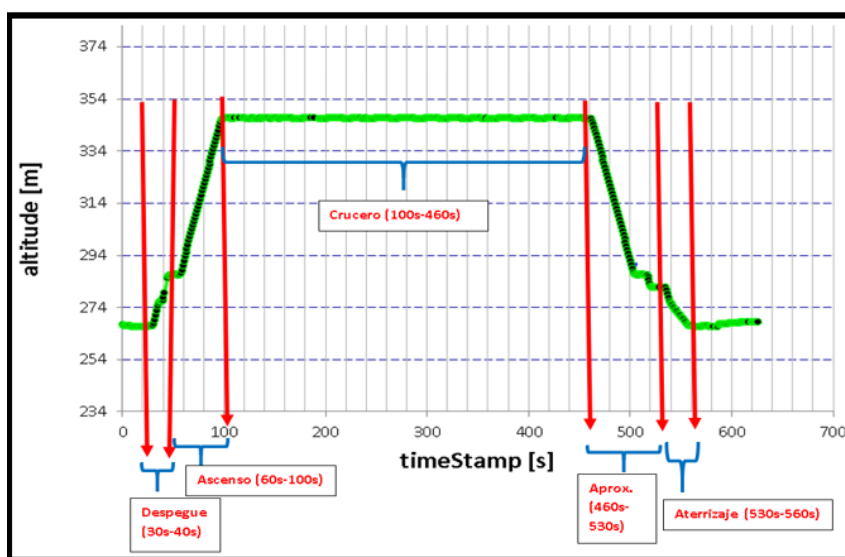


Fig. 4.6 Ejemplo de trayectoria vertical para el análisis de sectorización.

Tal y como podemos ver, las fases podrían quedar más o menos definidas a simple vista:

La primera subida en el instante 30s da indicio de que empieza el despegue. Y podría terminar al inicio del pico del vuelo (en el instante 100s), pero dado que entre el instante 40s y 60s tenemos variación nula en altura, sería más conveniente pensar que esa segunda subida es un *climb* para alcanzar la fase crucero.

Vemos que toda la parte central, que permanece en la misma altitud durante un tiempo prolongado, pertenecería a la fase crucero. Y, justo cuando empieza la primera bajada después de la fase crucero diríamos que se da inicio al aterrizaje o por lo menos a una fase de aproximación final.

Hablamos en condicional porque, según la visualización del gráfico esa es la impresión que tenemos sobre los sectores principales.

Habiendo hecho este pequeño análisis de forma superficial, tenemos que ser capaces de proporcionar datos más concretos a nuestro algoritmo y definir unas condiciones con las cuales podamos decir si un segmento está arraigado a una fase u otra.

Por tanto, vamos a definir una serie de condiciones que nos indiquen qué indicios tomar como referencia a la hora de determinar una fase específica.

4.4.2. Identificación de la transición de fases

4.4.2.1. De Initial Ground a Take Off phase

El primer sector del vuelo en realidad será ese período en el que el dron todavía está en el suelo a la espera de que el usuario dé la instrucción de despegar, lo llamaremos *ground phase* o fase tierra.

Pasada esta primera fase empezará a producirse la fase del *takeoff*. Para hallar el cambio a fase de despegue nos centraremos en el primer segmento de puntos donde encontremos una variación de altura considerable y continua.

Durante la fase tierra podemos encontrarnos con pequeñas variaciones de alturas que representen el manejo del dron por el usuario, por ejemplo, para ponerlo en el suelo (0,2 – 2m).

Teniendo este umbral de referencia podemos establecer un coeficiente de tolerancia respecto a esas perturbaciones a la vez que establecemos un margen medido en metros para el inicio del despegue.

A continuación, presentamos las variables de control de la parte del algoritmo que identifique el inicio del *Take Off*.

- *Ground_tolerance*:

Esto se define como una primera condición y un umbral que nos permite saber si pasamos de fase tierra a despegue, al mismo tiempo también se define como el grado de tolerancia de las fluctuaciones que afecten al dron.

Es decir, si ponemos un valor de *Ground_tolerance* muy bajo (0,5m) el programa podría pensar que, dada una variación de 0,5 de altura y continua, son indicios de ejecutar un despegue, cuando en realidad podría ser el simple manejo del dron para disponerlo en la plataforma de lanzamiento.

Por otro lado, si ponemos un valor de la variable demasiado alto (5m), podríamos incluso no llegar a percibir un despegue, dado que imponemos que para que se efectúe un despegue debe sobrepasarse obligatoriamente una variación de 5m.

- *Discontinuity*:

Nos referimos a una discontinuidad de puntos en base a que ya no forman parte de la misma fase. Precisamente esta variable, es una condición más para valorar si podemos estar cambiando de sector. Dado un rango de N puntos, podemos compararlos uno a uno con el *Ground_Tolerance* a fin de saber si se encuentran dentro del umbral.

Si en algún momento todos los N puntos sobrepasan el umbral de la variable de tolerancia (*ground tolerance*), pasaremos a fase despegue.

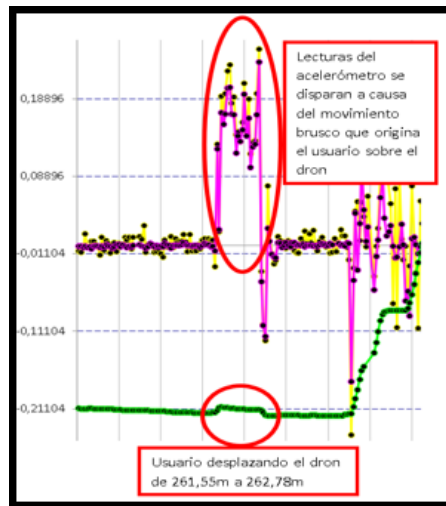


Fig. 4.7 Muestreo de puntos dónde se capta movimiento ascendente por manejo del usuario.

Representa el número máximo de valores continuos fuera de los márgenes de la tolerancia (pueden ser puntos que evolucionen hacia la fase *take off* o bien solo sean puntos singulares). Esto nos permitirá controlar el número de N valores que se encuentran después de una primera fluctuación.

Si por ejemplo, establecemos este valor a 5 y se da la casualidad que tenemos 5 puntos singulares seguidos fuera de los márgenes de la tolerancia, la lectura del takeoff será incorrecta. Si los 5 puntos se quedan fuera de la tolerancia y ascienden implicará que estamos en fase de despegue.

El valor de esta variable es muy aproximado a la duración en segundos de esa actuación aérea ya sea un despegue (movimiento deseado) o una prolongada fluctuación debida a una ráfaga de viento (movimiento indeseado).

4.4.2.2. *De Take Off a Cruise phase*

Una vez se está iniciando el despegue sabemos que en algún punto el vehículo dejará de ascender y se estabilizará. Tendríamos que hallar la forma de encontrar ese segmento de puntos, donde se incorpora a una altitud constante (o incluso hace un cambio de sentido en Z). Para ello volveremos a establecer algún umbral que nos ayude a identificar el final de un despegue.

A continuación, presentamos la variable de control para identificar el final del *Take Off*.

○ *Rang_air*:

Cuando el dron se considere dentro de la fase *Take Off* buscaremos los *N* siguientes valores según la variable *Rang_air*, y miraremos si se está produciendo un cambio continuo en los ascensos o se está estabilizando a una determinada altitud.

Si en algún momento los *N* valores de *Rang_Air* no superan variaciones de 0,25 metros (es decir, recorrer 0,25 metros a lo largo de *Z*, en *N* segundos) entonces entenderemos que se ha terminado la fase de despegue y pasamos a la de crucero.

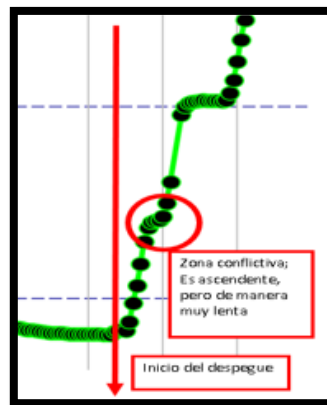


Fig. 4.8 Muestreo de puntos en la fase de despegue con posibles discrepancias para confirmar fin de la fase.

Cuanto mayor sea el valor de *Rang_air* más lento podrá ser nuestro despegue, pero menor será la precisión del punto que marque la transición de fase. Por otro lado, si situamos el valor demasiado bajo, podría ser que en una lectura errónea sucesiva de 2 o 3 puntos consecutivos durante un despegue, se determine que ya ha terminado.

La cualidad de estas lecturas dependerá del dispositivo con el que se tomen las muestras (módulo *Here2* en nuestro caso).

Por último, destacar que esos 0,25 metros mencionados anteriormente los hemos escogido a modo de prueba y error, pero en realidad podría ser una variable cuyo valor sea elegido por el usuario.

4.4.2.3. *De Final Ground a Landing phase*

Aunque parezca un tanto extraño, vamos a identificar la fase de *Landing* empezando por el último de los puntos que tengamos y recorriéndolos marcha

atrás. Dicho de otro modo, estamos analizando un aterrizaje marcha atrás aplicando la metodología del despegue (*Ground a Take Off*).

La metodología empleada será nuevamente el uso de los coeficientes de tolerancia, discontinuidad y rango aire (condicionados por las características propias de un aterrizaje). Por este motivo, si recorremos los puntos marcha atrás, creemos más conveniente empezar a mirar el cambio de la fase *Final Ground a Landing*.

A continuación, presentamos las variables de control para identificar el inicio del *Landing*, analizado en sentido opuesto.

- *Ground_tolerance*:

En este caso volveremos a tratar con fluctuaciones no deseadas o perturbaciones a causa del manejo del propio usuario. Podemos cambiar este parámetro siempre que queramos y distinguirlo de la parte inicial, es decir, el *Ground tolerance* de la parte inicial y final no tienen por qué compartir el mismo valor. También nos indica la altura a partir de la cual ha habido un cambio de fase; en este caso el cambio se estudiará como un *landing* invertido en el tiempo. Más o menos aproximado al estudio de un despegue.

- *Discontinuity*:

Esta variable contemplará las mismas condiciones que el parámetro *discontinuity* de la fase *Take Off*. Con él vamos a elegir de cuantos datos será el rango de valores que se estudien periódicamente para determinar si se trata de un tramo de transición de fase o no.

4.4.2.4. *De Landing a Cruise phase*

Este representará el último tramo a estudiar ya que la etapa de crucero estará asignada a todos aquellos puntos que no hayamos asociado a ninguna fase. Es decir, sabemos cuál es la fase de crucero por descarte de las otras fases.

Por tanto, para estudiar la transición entre la fase de aterrizaje marcha atrás y las fases de crucero usaremos también la variable *Rang_Air* pero teniendo en cuenta que un aterrizaje puede tener una mayor duración que un despegue, ya que un dron suele ir más lento en el descenso que en el ascenso según hemos visto en la mayoría de vuelos.

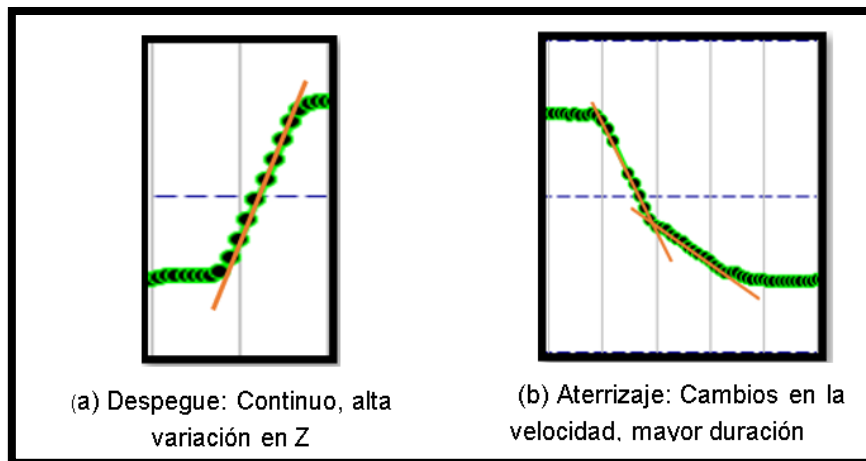


Fig. 4.9 Muestreo de puntos en la fase de despegue (158s-167s) y aterrizaje (339s-370s) entre 157 y 171 metros de altura

A continuación, presentamos las variables de control para identificar el final del *Landing*, en sentido opuesto.

- *Rang_air*:

Anteriormente en el apartado *De Take Off a Cruise phase*, habíamos usado este parámetro para valorar el ratio de ascenso del dron y la extensión del rango de muestras a valorar; si el ratio que presentan los valores de lectura era demasiado bajo, podíamos pensar que el dron se estaría estabilizando en la fase crucero, siendo demasiado sensible.

Según el ejemplo de la Fig. 4.9, asumimos que en el despegue la variación es más rápida que en el aterrizaje (aunque no siempre tiene por qué ser así). No obstante, se intentará encontrar una variable comuna tanto para el despegue como para el aterrizaje.

4.5. Analizador de maniobras de vuelo

Tal y como hemos podido ver en el apartado 4.4, la sectorización de fases no profundiza en los eventos que ocurren en la fase crucero. Precisamente, en esa fase es dónde se emplean la mayoría de las maniobras y también donde se prolonga la mayor parte del vuelo.

Así pues, volveremos a hacer un análisis cualitativo, pero esta vez centrándonos en el conjunto movimientos que puede efectuar el dron tanto en el plano transversal como a lo largo o alrededor del eje Z.

Antes de empezar el análisis, presentamos el *set* de maniobras convencionales que puede realizar cualquier *drone* y así tener un poco más claro en qué tipo de comportamientos deberemos centrarnos para identificarlos

4.5.1. Asunciones previas para la identificación de maniobras de vuelo

Explicaremos cómo procesar cada tipo de movimiento basándonos únicamente en el GPS, el elemento más fiable ya que contempla mediante el filtro de Kalman, las lecturas de las IMU [16].

También queremos comentar algunas tendencias que tiene el dron para realizar desplazamientos entre dos *waypoints*, cuando le asignamos un *flight plan* a través de *Mission Planner*:

- Si no se indica específicamente, el *heading* del vehículo suele estar orientado hacia la dirección del movimiento. Esto implica que raramente van a ocasionarse movimientos laterales (*Roll*) o movimientos marcha atrás o combinaciones de estos.
- Si no se indica específicamente, el dron no ejecuta *Hovers* para hacer un cambio de orientación (en *Yaw*). Más bien cambia la orientación mientras va avanzando, es decir, suele efectuar giros en movimiento.

A pesar de esas tendencias, es posible encontrarse con alguno de estos movimientos que acabamos de mencionar.

Finalmente, hay que mencionar que cualquier multicoptero tiene 6 grados de libertad; 3 en traslación (X, Y, Z) y 3 más en rotación (Pitch- θ , Roll- ϕ , Yaw- ψ).

No obstante, en nuestro caso, para la identificación de maniobras no pondremos atención ni en el Pitch ni en el Roll y las coordenadas de traslación las tendremos en consideración, pero en forma polar y no en cartesianas (aunque nos refiramos al plano transversal/radial como el plano XY).

Presentamos la forma polar de las coordenadas del *drone*:

$$r = \sqrt{X^2 + Y^2} \quad (4.4)$$

$$\rho = \textit{heading} \quad (4.5)$$

El origen del radio se situará en el punto donde se arme el dron, que suele ser en el punto de despegue. En cambio, el *heading* vendrá ya definido por el propio magnetómetro; donde el 0° se situaría enfocando al norte y 90° al este.

Podemos verlo representado en la Fig. 4.10.

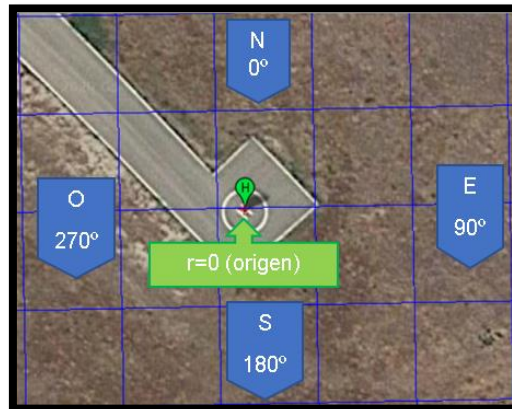


Fig. 4.10 Situación origen del radio inicial ($X=0$, $Y=0$) y disposición de los puntos cardinales en el escenario *Pla de vent*.

4.5.2. Identificación de maniobras de vuelo

Para explicar el tratamiento de datos, se hará a través de la 4.11, que consta de una ruta fragmentada en 10 *waypoints* distintos, juntamente con el despegue y el aterrizaje.

Las maniobras que puede realizar un multicoptero están explicadas en el ANEXO D.

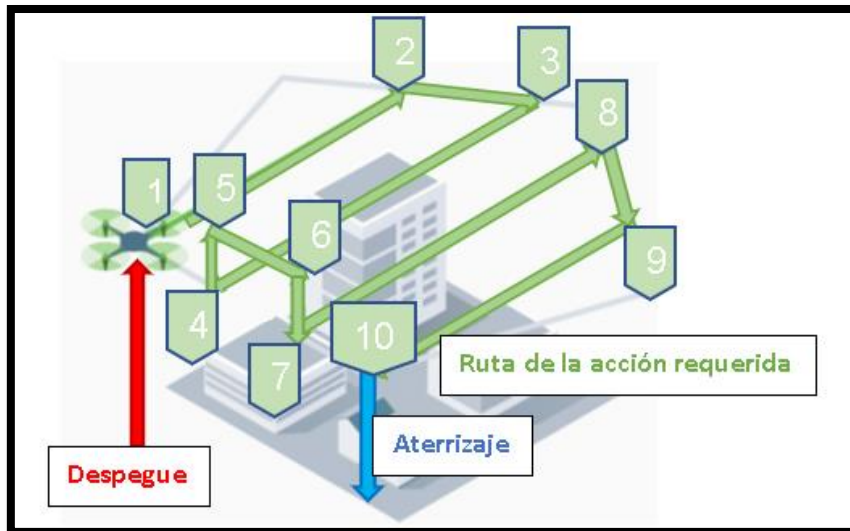


Fig. 4.11 Ejemplo de trayectoria en 3D con las 3 fases principales dividido en 10 waypoints

Empezaremos explicando que, ya que el *sampling time* es de 1s de media, elegiremos un valor de 3 muestras para conformar el rango de datos que será analizado de manera iterada en el tiempo. Con ello asumimos que una maniobra perceptible tendrá que durar 3 segundos o más.

Cada rango de datos se irá iterando y analizando para descifrar la *performance* que hay detrás de ese posible cambio de posición.

Estos rangos se analizan de 3 maneras según el eje o plano de movimiento. Se respetará un orden de prioridad.

4.5.2.1. Análisis según el eje vertical (variación en Z)

Teniendo un rango de 4 muestras, el algoritmo las compara analizando si hay variación de posición entre ellas a lo largo del eje Z y también la variación total. Si hay movimiento vertical, es porque hemos impuesto que la variación global no debe ser inferior a 1m (supuestamente superior a 0,25m entre muestra y muestra). Además, es condición necesaria que los 4 puntos del rango deban seguir un patrón ascendente o descendente en Z.

El rango de puntos se irá extendiendo hasta que el algoritmo pueda asegurar que la variación en Z ha finalizado. Una vez finalice pasará a estudiar los puntos 2 para esa misma extensión de puntos que han ido acumulándose en un rango dinámico.

Tabla 4.4 Segmentación de maniobras implicadas en el desplazamiento vertical según la Fig. 4.11

Rango en Z	From – To (Waypoints)	Variación en Z / Variación nula
1	1 - 4	Variación nula
2	4 - 5	Variación en Z
3	5 - 6	Variación nula
4	6 - 7	Variación en Z
5	7 - 10	Variación nula

La variable de control para el cambio de la altura es la siguiente:

- Alt_lim:

Para esta variable consideraremos que, para que se haya producido un cambio de altitud considerable, la altura tiene que haber sobrepasado 1 metro entre muestra y muestra. Si se diera el caso de que un ascenso se origina por debajo de 1 m/s es probable que el algoritmo no lo asuma como un ascenso, aunque sí se monitorizaría ese cambio de altitud, por pequeño que sea.

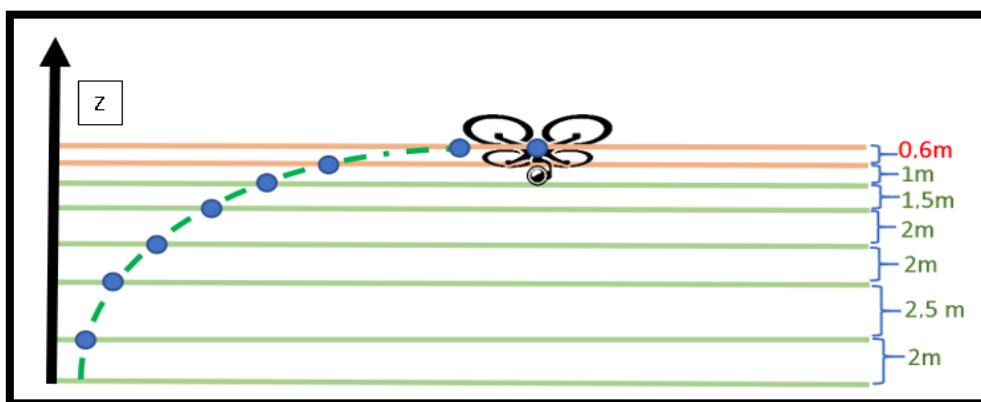


Fig. 4.12 Limite en el eje Z a partir del cual se detecta un cambio de altura ya sea estático o dinámico. Los últimos instantes pueden acarrear un pequeño error debido a la desaceleración del movimiento.

4.5.2.2. Análisis según el heading (variación en Yaw)

Teniendo definido un rango basado en un período de variación en Z, el algoritmo mirará si dentro de ese rango hay variaciones angulares.

Dado que las variaciones pueden haberse producido, por ejemplo, solamente en una parte del rango en Z, se crearía un sub-rango que contemple el inicio de variación en yaw hasta el final de dicha variación. Acto seguido, se procedería al análisis del 3º punto y luego volvería al punto 2 porque aún está pendiente ver la evolución de la variación de *heading* hasta el final del rango en Z.

Tabla 4.5 Opción 1 para la segmentación de maniobras de rotación según la Fig. 4.11

Rango en Z	Sub – Rango <i>heading</i>	From – To (Waypoints)	Variación en <i>heading</i> / Variación nula
1	1	1 – (1+N)	Variación en <i>heading</i>
	2	(1+N) - 2	Variación nula
	3	2 – (2+N)	Variación en <i>heading</i>
	4	(2+N) – 3	Variación nula
	5	3 – (3+N)	Variación en <i>heading</i>
	6	(3+N) - 4	Variación nula
	7	4 - (4+N)	Variación en <i>heading</i>
2	8	(4+N) – 5	Variación nula
3	9	5 – 6	Variación nula
	10	6 – (6+N)	Variación en <i>heading</i>
4	11	(6+N) – 7	Variación nula
5	12	7 – 8	Variación nula
	13	8 – (8+N)	Variación en <i>heading</i>
	14	(8+N) – 9	Variación nula
	15	9 – (9+N)	Variación en <i>heading</i>
	16	(9+N) – 10	Variación nula

Anotaciones:

- En los casos donde sumamos “±N”, suponemos que los valores definidos de *From-To* corresponden a un vector de datos X,Y, mientras que N solamente hace referencia al número de muestras anteriores o siguientes a ese mismo vector.
- Existe una segunda opción [ANEXO E] de cómo el dron podría realizar los movimientos en heading de la Fig. 4.11, ya que en la opción 1 el dron

realiza un *Hover* para ejecutar la rotación, mientras que en la opción 2, el dron sigue desplazándose en la medida que también ejecuta la rotación.

A continuación, presentamos la variable de control para el análisis angular:

- Ang_lim:

El ángulo de variación para poder identificar tanto giros dinámicos como rotaciones en *yaw* está establecido en $\pm 4,5^\circ$ respecto al eje de orientación (X_b *body frame*). Se ha buscado un valor angular que pueda percibir pequeñas variaciones de un giro en movimiento que presente un radio de curvatura considerable. Este valor supone la identificación de radios de curvatura de hasta 400m.

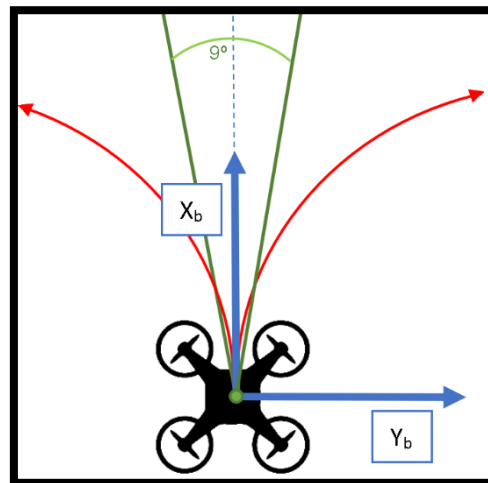


Fig. 4.13 Limite angular a partir del cual se detecta un giro ya sea estatico o dinámico.

4.5.2.3. Análisis según la posición en el plano XY (variación del radio)

Cuando el algoritmo llega al tercer punto asumimos que ya tenemos un primer rango en Z y un sub-rango en *heading*. Entonces se recorren todos los puntos del sub-rango de *heading* más reciente, mirando la variación de posición en el plano transversal (plano XY) e imponiendo que el cambio de posición implique una distancia mayor a 1,2 m.

Una vez finalice este 3r análisis fragmentado volveríamos al punto 2 para que sigan haciéndose evaluaciones y creando los sub-rangos necesarios a fin de haber estudiado todos los puntos del rango en Z según las 3 metodologías enumeradas.

Tabla 4.6 Opción 1 para la segmentación de maniobras desplazamiento en XY según la Fig. 4.11

Sub – Rango <i>heading</i>	Sub – Rango <i>para los puntos de XY</i>	<i>From –To</i> (Waypoints)	Variación en XY / Variación nula
1	1	1 – (1+N)	Variación nula
2	2	(1+N) – 2	Variación en XY
3	3	2 – (2+N)	Variación nula
4	4	(2+N) – 3	Variación en XY
5	5	3 – (3+N)	Variación nula
6	6	(3+N) – 4	Variación en XY
7	7	4 – (4+N)	Variación nula
8	8	(4+N) - 5	Variación nula
9	9	5 – 6	Variación en XY
10	10	6 – (6+N)	Variación nula
11	11	(6+N) – 7	Variación nula
12	12	7 – 8	Variación en XY
13	13	8 – (8+N)	Variación nula
14	14	(8+N) – 9	Variación en XY
15	15	9 – (9+N)	Variación nula
16	16	(9+N) – 10	Variación en XY

Anotaciones:

- Aunque no hemos mencionado el “sub-rango para los puntos XY” en este 3r punto, sí que es cierto que lo usamos a nivel auxiliar para distinguir los puntos que se recorren en los movimientos del *heading* y los que se recorren en XY. La extensión del sub-rango para puntos en XY va a ser la misma que en el sub-rango para heading.

- Rad_lim:

Para esta variable de control tendremos en cuenta un valor de 1,2m; un valor que, a prueba y error, ha captado mejor la diferencia entre movimientos por vibraciones / posibles errores del GPS (no intencionadas), y movimientos por maniobras de avance (intencionadas).

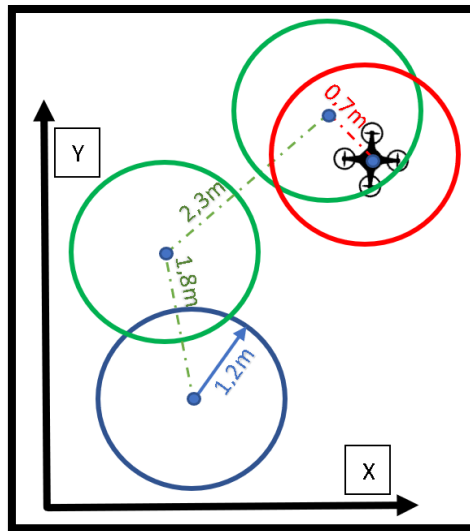


Fig. 4.14 Limite radial a partir del cual se detecta un suceso movimiento o un suceso estático en el plano XY

4.6. Formato para la presentación de datos en la identificación de fases y maniobras de vuelo

En último lugar, queremos destacar que los datos de los apartados 4.5.2 y 4.6.2 se mostrarán en forma de gráfico y tabla respectivamente.

En la parte de la sectorización de fases se monitorizará un gráfico con el perfil vertical del vuelo, pero enfatizando cada sector con un color distinto tal y como se muestra en la Fig. 4.15.

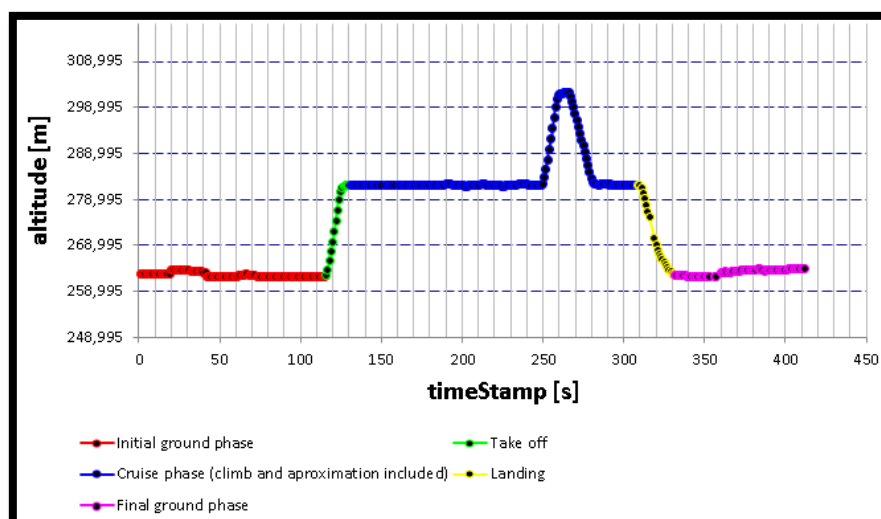


Fig. 4.15 Ejemplo de sectorización de fases en forma de perfil vertical

En cambio, la identificación de las *performances* se hará a través de una tabla, que contendrá un *set* de información de 8 parámetros que va a permitirnos tener una idea más clara de cómo se ha llevado a cabo cada maniobra. En la Tabla 4.7 podemos ver una tabla resumen.

Tabla 4.7 Cabecera de los datos para la identificación de maniobras

Time (duration) [s]	Performance along Z axis	Performance in XY plane
Change of distance in altitude [m]	Change of distance in XY plane [m]	Angular change in XY plane [°]
Mean speed in Z axis [m/s]	Mean speed module in XY plane [m/s]	

4.7. Monitorización de datos adicionales de interés

Una vez finalizadas las identificaciones de maniobras, ya disponemos de información suficiente como para ofrecer un último *set* de 7 datos globales que pueden ser de interés para el análisis de vuelo.

Básicamente se quieren mostrar máximos y mínimos de ciertos parámetros en el vuelo global a parte del tiempo total del vuelo y el *landing path error*:

- **Total flight time:** nos indica el tiempo transcurrido entre el momento en que el dron deja de tocar el suelo en el despegue y el momento en que lo vuelve a tocar en el aterrizaje.
- **Landing path error:** suponiendo que se quiere despegar y aterrizar en el mismo punto, obtendremos la desviación, en metros, entre los puntos (X,Y) en el despegue y (X,Y) en el aterrizaje.
- **Max. Altitude:** nos dará el valor del punto máximo en Z que se ha alcanzado a lo largo del vuelo.
- **Ground altitude:** obtención de la altitud del suelo dependiendo del escenario donde ejecutemos el vuelo.

- ***Max speed in ascence and descence:*** con estos dos parámetros se quiere identificar la velocidad máxima en subida y en bajada, que no tienen por qué darse en el despegue y aterrizaje. Normalmente será más rápida la velocidad de subida que de bajada.
- ***Max module speed in XY plane:*** se identificará el pico máximo de la velocidad en el plano transversal dado un instante de tiempo

Capítulo 5. Resultados y discusiones

En este capítulo se mostrarán los resultados obtenidos con relación a la información que nos aporten los gráficos y también a la identificación de maniobras y fases de vuelo.

El análisis contemplará la comparación entre dos vuelos tomados el mismo día con condiciones meteorológicas prácticamente idénticas.

Antes de ir directamente a los resultados, presentamos los 2 vuelos de los cuales se han extraído los resultados.

5.1. Contexto de los dos planes de vuelo

5.1.1. Vuelo 1

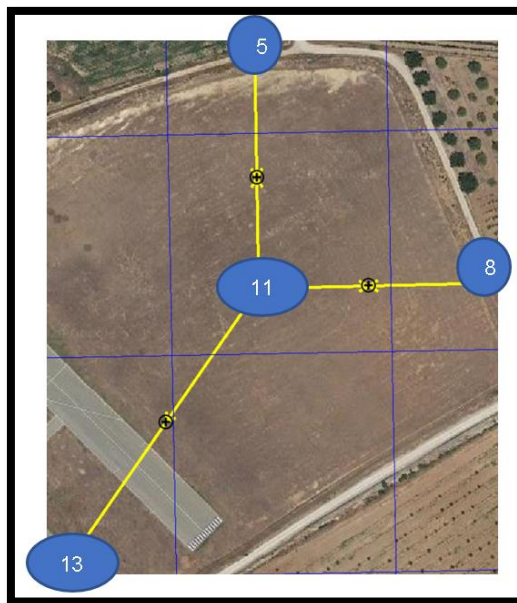


Fig. 5.1 Diseño de la ruta del vuelo 1 mediante *waypoints* [velocidad de crucero 10 m/s, radio del WP 1m]

Este vuelo fue diseñado con la finalidad de hacer 3 comprobaciones:

- Tomar muestras de los sensores y estados del dron para su futuro análisis.
- Verificar que el set de *waypoints* y maniobras insertadas en MP se efectúan correctamente [ANEXO F]
- Comprobar los sistemas de referencia de algunos sensores y dispositivos (por ejemplo, GPS y IMU no tienen por qué compartir el mismo marco de referencia).

5.1.2. Vuelo 2

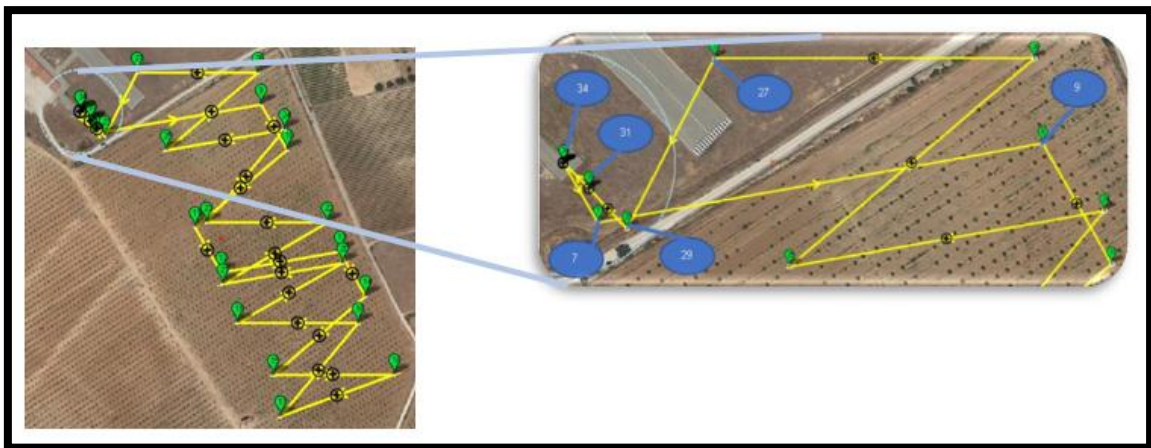


Fig. 5.2 Diseño de la ruta del vuelo 2 mediante *waypoints* [velocidad de crucero 10 m/s, radio del WP 1m]

Este vuelo fue diseñado con la finalidad de hacer 2 comprobaciones:

- Corroborar la curvatura de giros considerablemente cerrados y ver la actitud del dron en realizar esa maniobra.
- Verificar que el set de *waypoints* y maniobras insertados en MP se sigue de manera adecuada durante el vuelo [ANEXO F]
- Tomar muestras de los sensores y estados del dron para su futuro análisis.

5.2. Resultados basados en los gráficos

En los siguientes apartados mostraremos los *plots* efectuados por el algoritmo que hemos creado y haremos una valoración de los resultados a la vez que, cuando sea conveniente, realizar ciertas comparativas entre los vuelos.

Mostraremos 4 familias de gráficos comentadas en el apartado 4.3.1.

5.2.1. Tracking

5.2.1.1. Tracking según el plano XY

A primera vista, según vemos en la Fig. 5.3, tanto la ruta 1 como la 2 cumplen con la trayectoria del recorrido diseñado en MP. Aunque solo mostramos la trayectoria en unidades métricas, también está graficada en unidades geodésicas [ANEXO G].

El nivel de exactitud se puede comprobar en la tabla 5.1 comparando la latitud-longitud de los *waypoints* de MP (Fig. 5.1 y Fig.2) con los valores latitud-longitud de los vértices de las fases principales y finales en el vuelo 2 y la fase crucero del vuelo 1 (Fig. 5.3)

Tabla 5.1 Exactitud en puntos clave de la ruta 1

<i>Waypoints</i>	Lat-Lon del WP [°]	Lat-Lon Real [°]	Error [m]
11 (primera vez)	(41,2885391-1,3321245)	(41,288536-1.3321255)	0,3543
11 (segunda vez)	(41,2885391-1,3321245)	(41,288540-1,3321258)	0,1478
11 (tercera vez)	(41,2885391-1,3321245)	(41,28854-1,3321245)	0,0993
5	(41,2894398-1,3321013)	(41,28944-1,3321003)	0,0866
8	(41,2885567-1,3333179)	(41,288555-1,3333176)	0,1904

Tabla 5.2 Exactitud en puntos clave de la ruta 1

<i>Waypoints</i>	Lat-Lon del WP [°]	Lat-Lon Real [°]	Error [m]
7	[41,28722763 - 1,33127367]	[41,287235- 1,3312863]	1,3374
9	[41,28751373 - 1,33342576]	[41,287502- 1,3333921]	2,1055
27	[41,28781891 -1,33183718]	[41,28781- 1,3318428]	1,0931
29	[41,28720474 , 1,33141303]	[41,287205- 1,3314122-]	0,0752
31	[41,2873497- 1,33123064]	[41,287346- 1,3312339]	0,4933

Según las tablas 6.1 y 6.2 podemos decir de que el nivel de exactitud de nuestro sistema de vuelo se adecua bastante al plan de vuelo diseñado en MP. En la ruta 1 se puede apreciar que los errores oscilan entre los 0,0866 y 0,3543 metros, mientras que en la ruta 2 se presenta un error considerable en la zona de las curvas (0,0752 – 2,1055) debido a la tendencia natural del *drone* para efectuar la combinación de giros, en vez de efectuar un *hover* con un cambio de *yaw* y luego avanzar (como pasa en el caso de la ruta 1).

En cuanto a las velocidades, los 2 vuelos están programados para moverse a 10 m/s y según vemos en el parámetro de velocidad máxima (tablas 5.7 y 5.8), son los 8,71 y 8,79 m/s respectivamente.

Sería posible que el *drone* buscara el equilibrio entre satisfacer esa velocidad y evitar el sobrepaso del radio máximo que se le ha asignado a los WPs (1m).

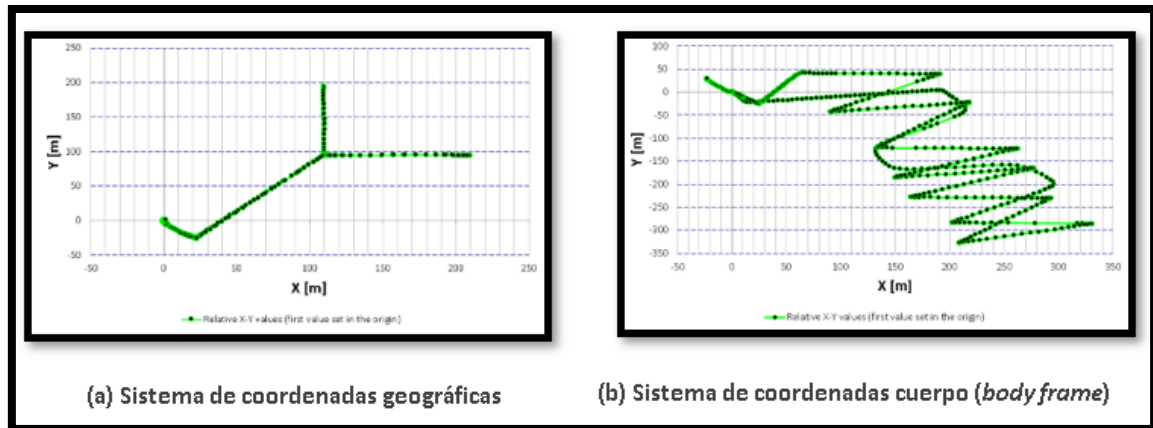


Fig. 5.3 Tracking en el plano XY de la ruta del vuelo 1 y 2

5.2.1.2. Tracking según el eje Z

Lo primero que salta a la vista es que la altura proporcionada por el GPS queda completamente eclipsada por las alturas de la ecuación hipsométrica y la que proporciona el sensor de distancia láser.

Las otras dos, que quedan más abajo, son también teóricas pero procesadas por la ecuación de atmosfera estándar. El hecho de que queden separadas por un *offset* de altitud se debe a que no sabemos exactamente la presión a nivel del mar en *Pla de Vent*, y en vez de ello se estableció el valor teórico estándar a nivel del mar (101300 Pa).

No obstante, si quitáramos ese *offset* veríamos como el resultado de esas alturas también se superponen con las lecturas del GPS. Por otro lado, decir que la altura de la ecuación hipsométrica no sufre ese desplazamiento ya que la condición inicial implica una altura (sacada del GPS en el instante 0) y no una presión a nivel del mar.

También tenemos las lecturas del sensor de altura láser, que son muy buenas en todo momento, excepto en la fase crucero donde podemos ver perturbaciones por irregularidades en el terreno de vuelo. Este dispositivo podría ser utilizado en el despegue y aterrizaje.

El GPS (juntamente con el EKF) nos da la mayor aproximación posible para la altura de entre todos los dispositivos. En la tabla 5.3 podemos apreciar los

errores de la altura GPS para los segmentos donde se realiza un ascenso o un descenso.

Tabla 5.3 Exactitud en los segmentos de ascenso/descenso para el vuelo

Rango temporal del segmento [s]	Cambio de altitud teórico [m]	Cambio de altitud real [m]	Error [m]
[115,981 - 130,981]	+20m	+19,74	0,26
[249,984 - 258,982]	+20m	+18,49	1,51
[266,984- 283,987]	-20m	-17,93	-2,07
[308,981- 331,981]	-20m	-19,70	-0,3

Tabla 5.4 Exactitud en los segmentos de ascenso/descenso para el vuelo

Rango temporal del segmento [s]	Cambio de altitud teórico [m]	Cambio de altitud real [m]	Error [m]
[29,986-50,989]	+20	+19,67	0,33
[57,988 - 97,988]	+60	+58,74	1,26
[462,987- 503,987]	-60	-57,73	-2,27
[517,987- 521,988]	-5	-3,82	-1,18
533,989	-15	-14,87	-0,13

No es ninguna sorpresa ver que los errores en el eje vertical son mayores que en XY, ya que, el GPS suele presentar un mayor error en el eje vertical que en el plano horizontal [17]. Aun así solamente podemos remarcar que en ambos vuelos el error es mucho más pequeño en el despegue y el aterrizaje, que en la fase crucero.

Un hecho negativo de la altitud marcada por el GPS es que, sabiendo que la altitud de la plataforma de lanzamiento está ubicada a 262m, vemos que para el vuelo 2 se establece dicha altura a 266 metros. Con lo cual tenemos un error de 4 metros, posiblemente por la posición de los satélites en ese instante.

Por último, hay que destacar que, según la tabla 5.7 y tabla 5.8, para ambos vuelos las velocidades de ascenso son mayores que las velocidades en el

descenso. Esto se debe a la física y el consumo energético detrás de los descensos para que se realicen de forma estable, a base de pequeños acelerones continuos[18].

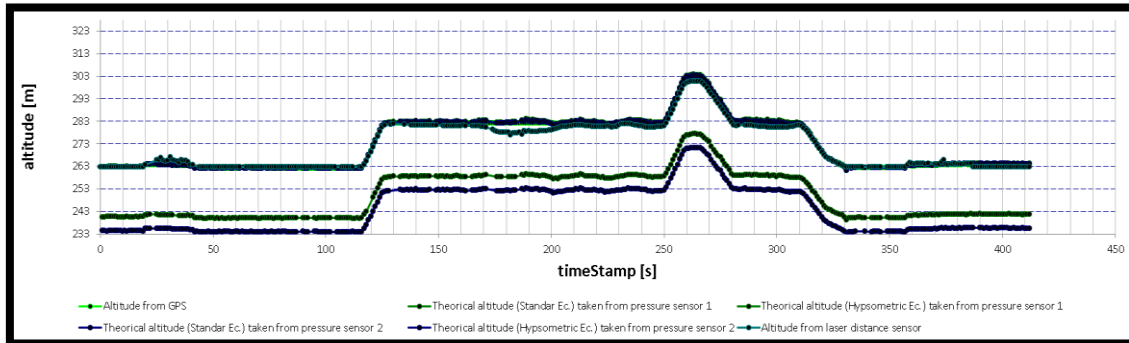


Fig. 5.4 Perfil vertical para el vuelo 1 mediante distintos sensores

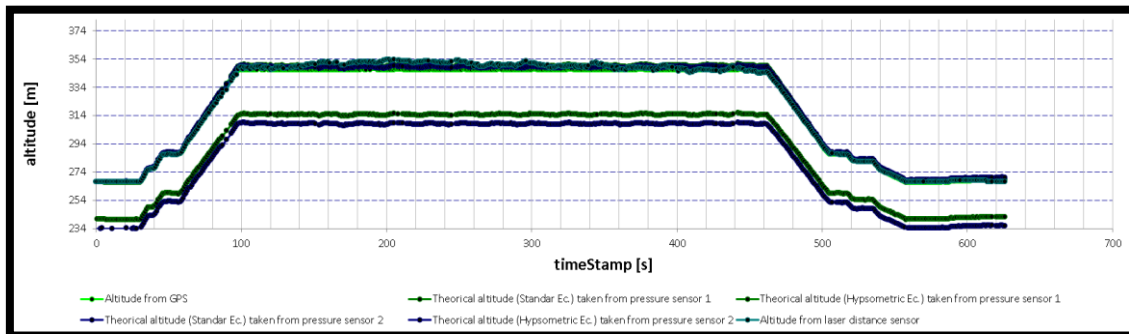


Fig. 5.5 Perfil vertical para el vuelo 2 mediante distintos sensores

5.2.2. Accel. and attitude

5.2.2.1. Aceleración vertical

En la figura 5.6 podemos ver la variación en altura y aceleración compartiendo el mismo eje temporal, de este modo será más intuitivo relacionar un suceso con esas dos variables graficadas.

Se pueden ver los cambios de aceleración por las IMU 2 y 3 en el eje Z. En la grafica están representados algunos de los picos de aceleración (0.21, -0.09 m/s²) que marcan un cambio de estado, ya sea para acelerar como para desacelerar en puntos clave del vuelo.

Cuando el dron se encuentra estático sobre el suelo, en la fase tierra, la IMU que parece responder mejor al estado en reposo es la IMU 3. En el resto de los movimientos tiene una tendencia más suave e incluso parece que con menor ruido.

En la fase tierra inicial i final podemos notar las perturbaciones generadas por el usuario al llevar el *drone* hacia la plataforma de lanzamiento tanto en ambas IMUs.

Los sucesos que acabamos de mencionar los comparten ambos vuelos (ver vuelo 2 [ANEXO H])

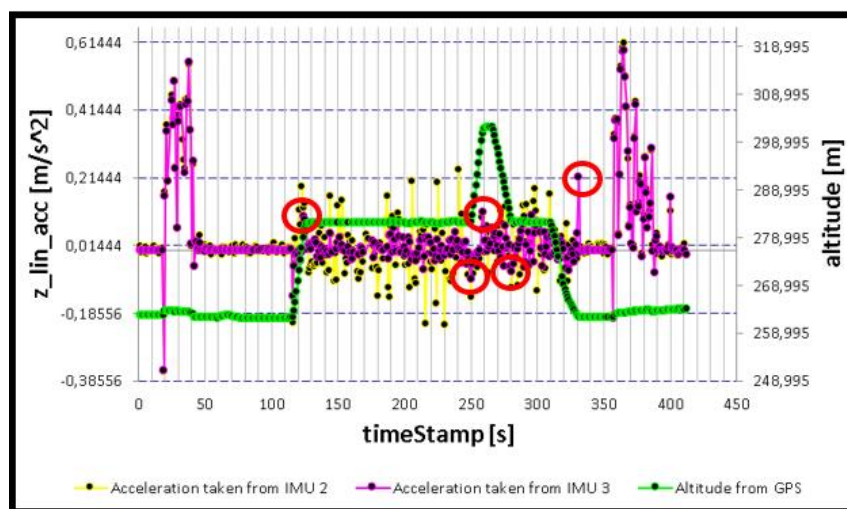


Fig. 5.6 Cambios de aceleración y altitud en el eje Z con señalización de picos de la IMU 3, en el vuelo 1

5.2.2.2. Aceleración en el plano XY

Por una parte, tenemos el módulo de los cambios de aceleración a través del plano XY [ANEXO H]. Y después tenemos la descomposición de la aceleración según los ejes cuerpo del dron, podemos verlos en la figura 5.7.

Mientras que la aceleración en X se grafica con el movimiento en Pitch, la aceleración en Y se grafica con el movimiento en roll.

La ruta del vuelo 2 tiene multitud de giros, con lo cual podemos apreciar repetidos picos (m/s^2) de aceleración y variación de ángulo en ambos gráficos al igual que sucede con el eje vertical.

Mientras que los picos del ángulo pitch oscila entre 0,11 y -0,3 radianes, los del ángulo roll oscilan entre 0,2 y -0,18 radianes.

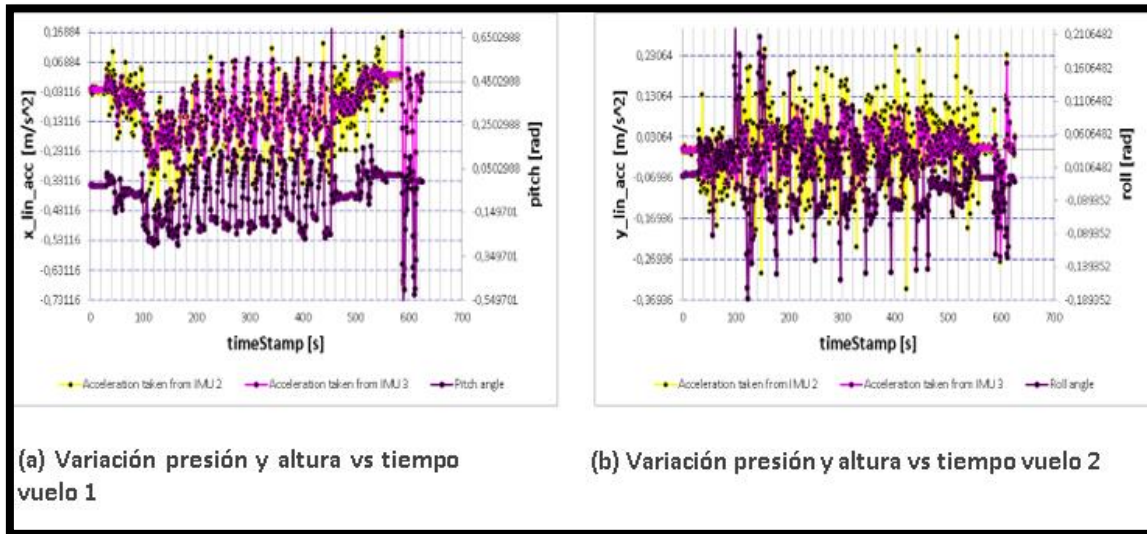


Fig. 5.7 Cambios de aceleración y altitud en el eje Z con señalización de picos de de la IMU 3, en el vuelo 1

5.2.3. Troposphere behave

5.2.3.1. Presión

Por un lado, se ha ploteado en la Fig. 5.8 el cambio de presión con la altura del GPS compartiendo el mismo eje temporal. La proporcionalidad es inversa y se ve de forma muy clara. En ambos vuelos se comparte esa relación entre presión y altura.

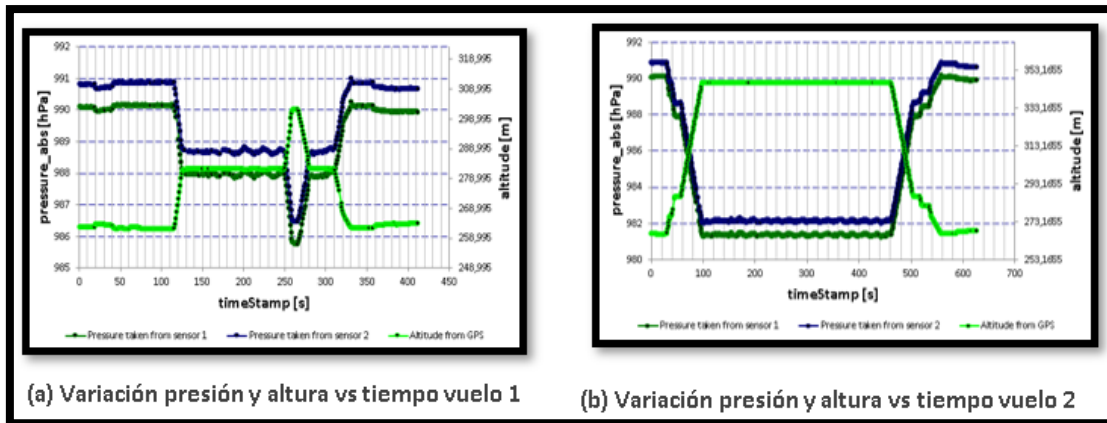


Fig. 5.8 Variación de presión y altura vs el tiempo de los vuelos 1 y 2

En la Fig. 5.9 vemos que el gradiente de presión a lo largo de la altura cambia de la forma en que nos lo indica la figura la Fig. 4.5 para el vuelo 2, pero no para el vuelo 1. Pues, remarcando que el cambio de altura es considerable en el vuelo 2 en comparación con el 1, podríamos deducir que para tener lecturas fiables sobre el cambio de presión en bajas regiones de la troposfera, debemos hacer cambios de altura considerables, de lo contrario obtenemos distintos valores de presión para la misma altura y viceversa.

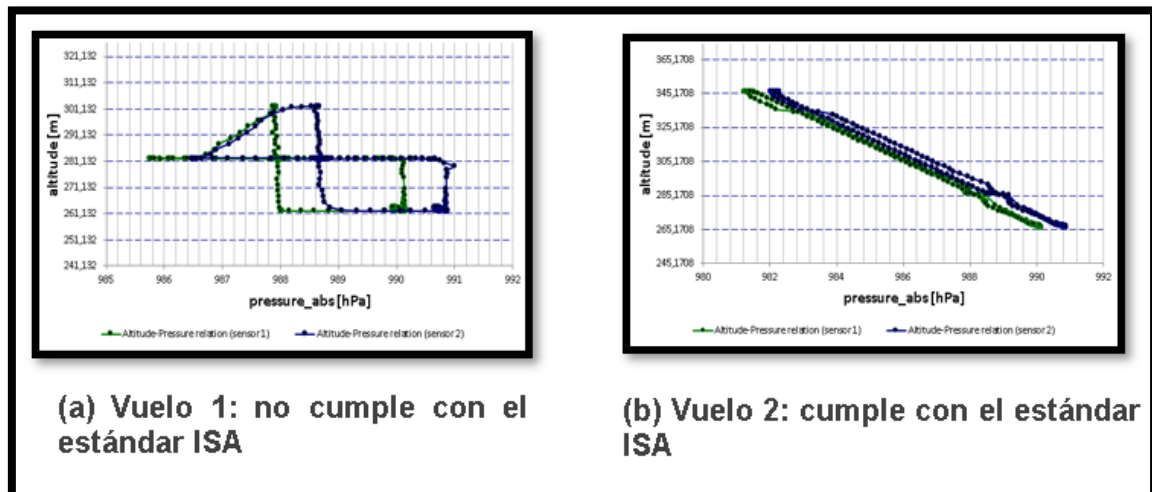


Fig. 5.9 Variación de presión vs la altura de los vuelos 1 y 2

5.2.3.2. Temperatura

La idea de los gráficos para la temperatura es exactamente la misma que para la presión. En la figura 5.10 vemos que hay una tendencia que no se identifica con el estándar ISA (International Standard Atmosphere).

Esto se debe a que la temperatura medida proviene de la resistencia de que se encarga de dar una buena temperatura de trabajo para las IMU. En principio, las debe mantener a una temperatura constante, por tanto, en función de la temperatura exterior y la temperatura global del sistema, el resistor tendrá que emitir más o menos calor.

Además, hay una variación significativa entre las lecturas de temperatura 1 y 2 en ambos vuelos. Mientras que el sensor 2 detecta más oscilaciones de temperatura, el sensor 1 permanece prácticamente estable.

Podría ser que un sensor estuviera más cerca del resistor térmico.

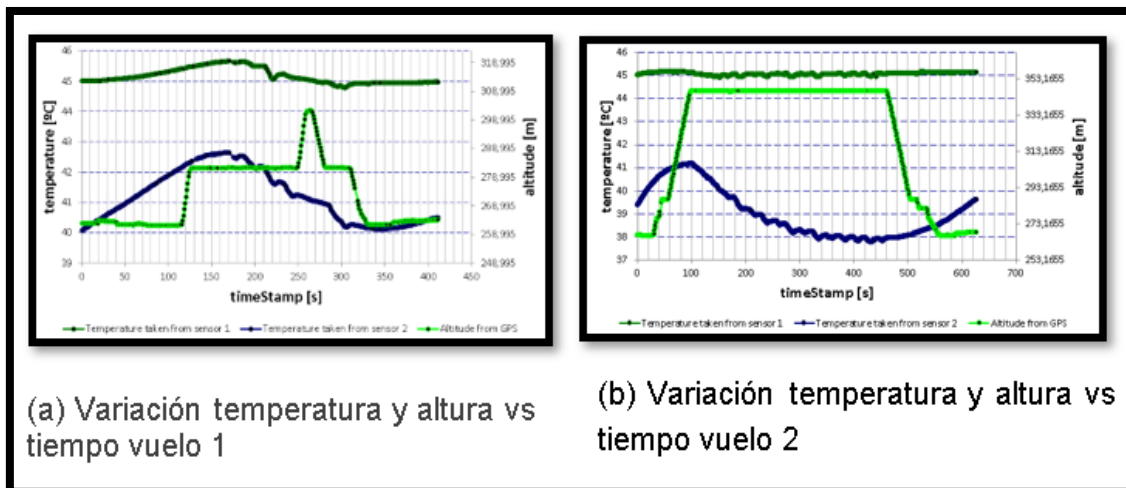


Fig. 5.10 Variación de temperatura y altura vs tiempo de los vuelos 1 y 2

Por otro lado, tenemos los valores de temperatura tomados en distintas alturas. Podemos verlo en la Fig. 5.11. Y es que tanto para el vuelo 1 como para el vuelo 2 tenemos varias temperaturas para un mismo valor de altura y viceversa.

En este caso, ni si quiera haciendo grandes cambios de altitud obtenemos una curva descendente a lo largo de la altura, como debería salirnos según el modelo ISA.

En principio esto nos indica el comportamiento del resistor térmico a lo largo de la altura de los vuelos, pero no parece que guarde una correlación directa con el comportamiento teórico que debería tener la atmosfera.

Finalmente es destacable que, aun haber obtenido valores de temperatura anómalos, no han afectado a los resultados de la ecuación hipsométrica, ya que esta es más sensible a los cambios de presión que de temperatura.

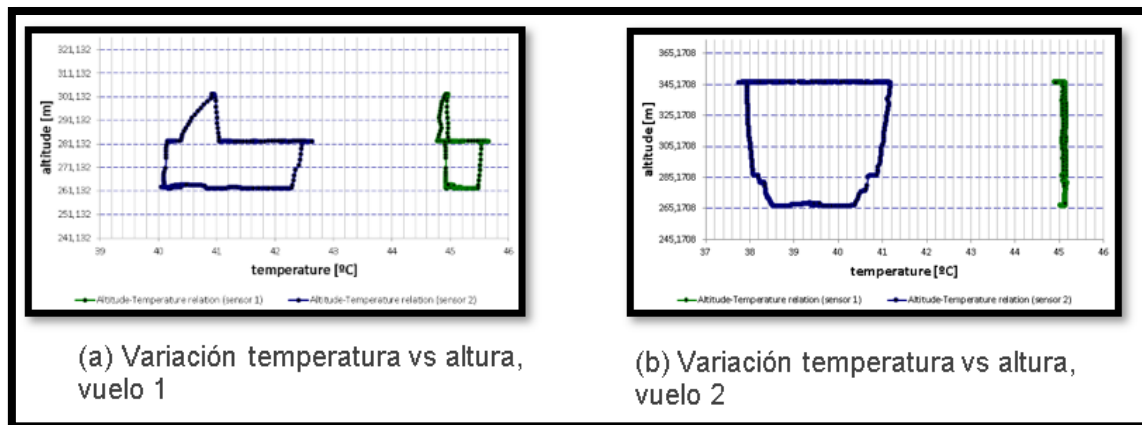


Fig. 5.11 Variación de temperatura vs tiempo de los vuelos 1 y 2

5.2.4. Consumption analysis

5.2.4.1. Battery remaining y corriente

En la Fig. 5.12 se puede ver la caída de la batería restante al mismo tiempo que vemos la caída de tensión por cada instante.

En la parte inferior del grafico puede verse la pendiente (*slope*/ratio de consumo) con la que cae la batería restante. Si comparamos la diferencia de batería restante entre el vuelo 1 y 2, entre el inicio y el final de cada vuelo, vemos en el vuelo 1 ha consumido el 21% de la batería (120s-330s) , mientras que en el vuelo 2 el 49% (20s – 550s).

El *slope* es más negativo para el vuelo 2, ya que ha realizado más maniobras de giros, enfatizando cambios de intensidad comentado en el apartado 6.2.4.2.

Aun viendo las caídas de tensión, con este modelo no podemos analizar la causa de la disminución global del voltaje.

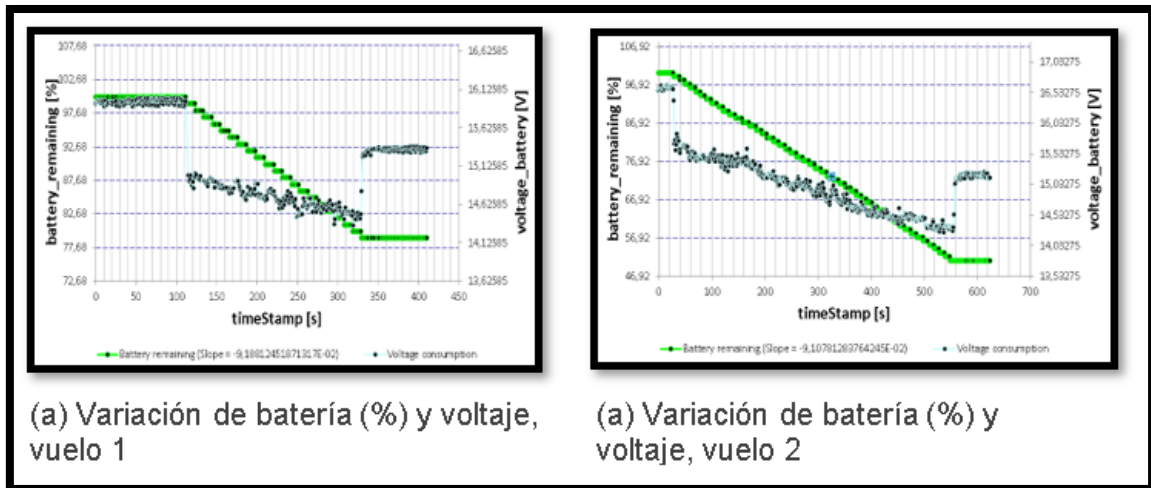


Fig. 5.12 Variación de *battery remain*. Y voltaje vs tiempo de los vuelos 1 y 2

Por otro lado, tenemos los datos de la corriente, los cuales vemos que oscilan entre picos (9A – 16A) con mayor frecuencia en el vuelo 2; también en el vuelo 1 también oscila con la misma intensidad, aunque podríamos decir que la frecuencia de oscilación es menor. Los cambios de aceleración en el vuelo 1 son menos continuos, por tanto, se asume que los cambios de aceleración son proporcionales a los cambios de corriente.

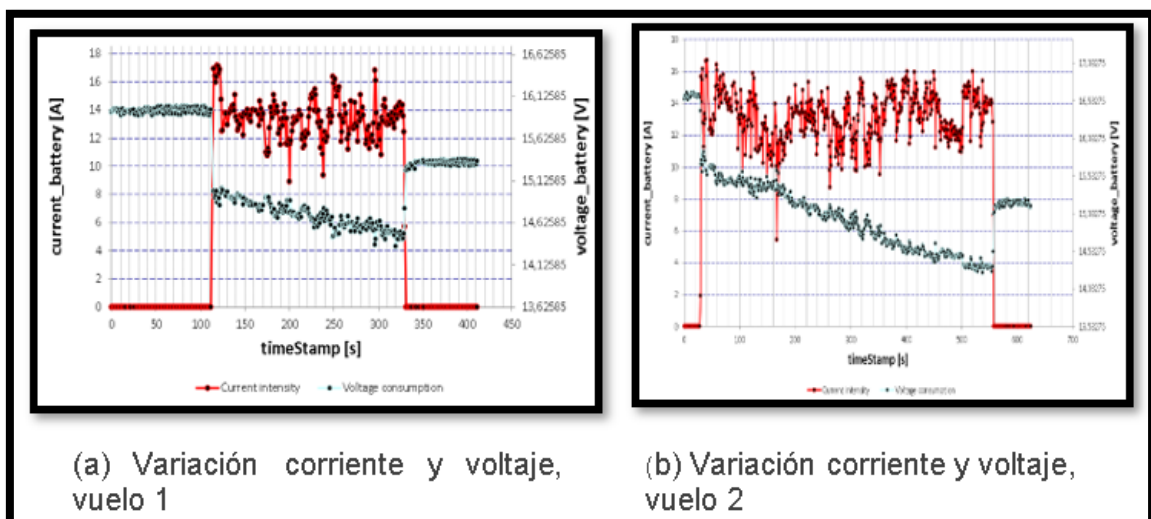


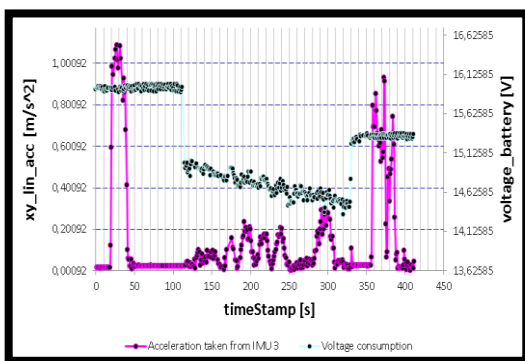
Fig. 5.13 Variación de corriente y voltaje vs tiempo de los vuelos 1 y 2

5.2.4.2. Aceleración

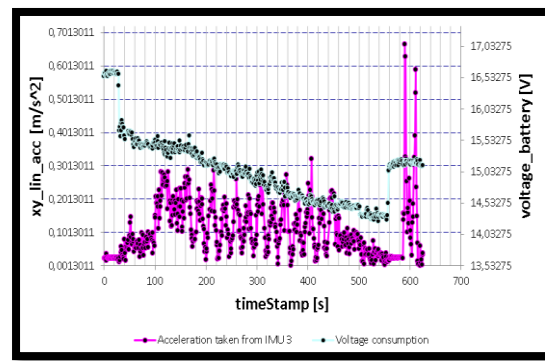
Finalmente vamos a ver si las aceleraciones marcan algún patrón que pueda asociarse a la reducción de la batería.

Según los gráficos de la Fig. 5.14 podemos asociar la forma de sierra que presentan los picos de aceleración con otra forma de sierra en desescalada para las muestras de voltaje.

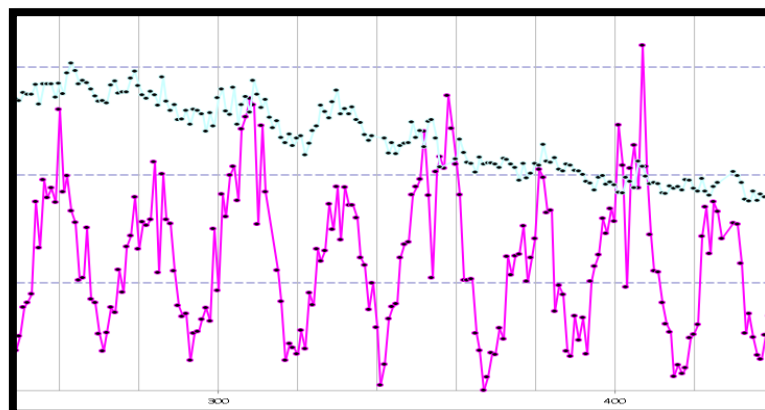
Aunque, esa forma de sierra desescalada en el voltaje se nota más en el vuelo 1, si hacemos zoom vemos que también ocurre de forma acentuada en el vuelo 2 (ver en Fig. 5.14 (c)). Lo notamos más en el vuelo 1 porque la prolongación de un cambio de aceleración es mayor, aunque en realidad sabemos que el ratio de consumo es mayor en el caso vuelo 2.



(a) Variación corriente y aceleración, vuelo 1



(b) Var. corriente y aceleración, vuelo 2



(c) Variación corriente y aceleración, vuelo 2 (zoom in)

Fig. 5.14 Variación de aceleración y voltaje vs tiempo de los vuelos 1 y 2

5.3. Resultados basados en identificación de fases y maniobras

5.3.1. Sectorization and motion

5.3.1.1. Resultados para la sectorización de fases de vuelo

Los resultados de este apartado pueden variar en función de los valores de las variables de control definidas en el apartado 4.4.2.

Aunque, a prueba y error hemos logrado obtener resultados satisfactorios, contrastaremos otros casos en los que pueda notarse el papel que juegan algunas de las variables.

En principio los valores que más se adaptan a la solución que queremos obtener son los siguientes:

- Vertical_tol_ground = 2
- Rang_air = 5
- Discont = 5

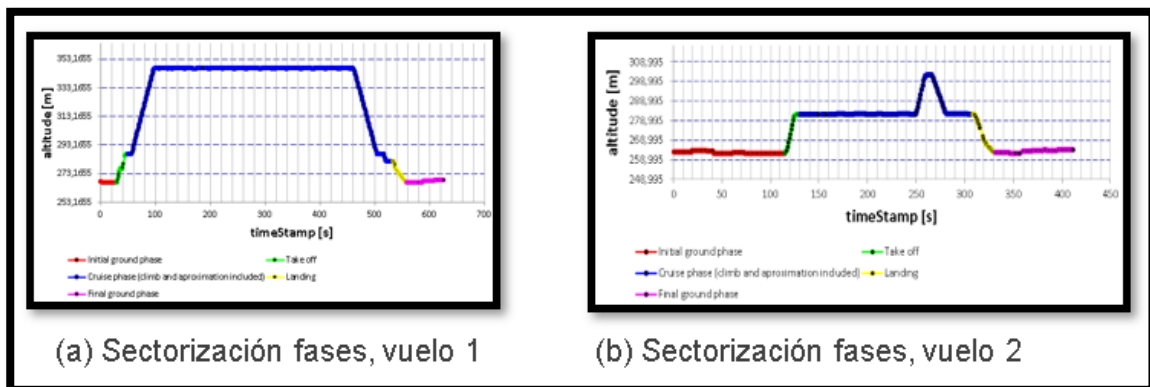


Fig. 5.15 Correcta sectorización de fases

Creemos que la variable, que puede acarrear mayor error es Rang_Air, ya que ha resultado ser más difícil determinar el final de un despegue que el inicio.

→ Disminuimos el valor Rang_Air en 2 (lo hacemos más restrictivo)

- Vertical_tol_ground = 2
- Rang_air = 5
- Discont = 2

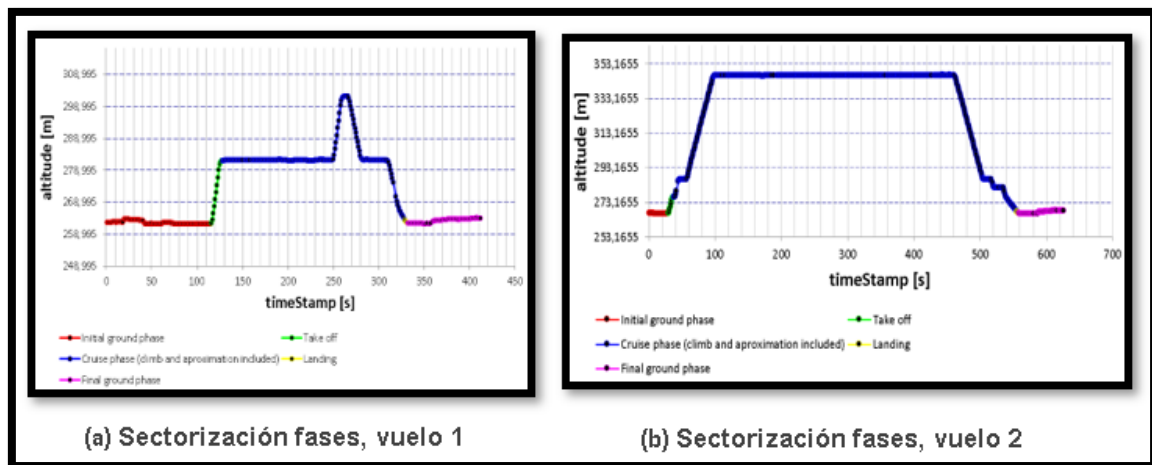


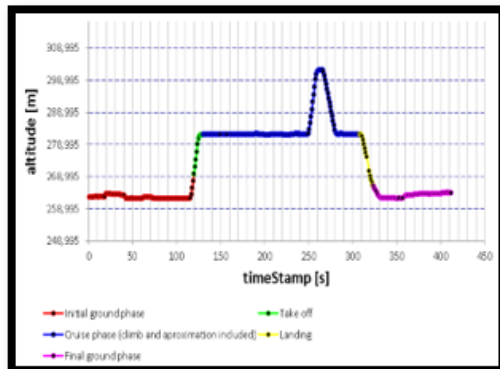
Fig. 5.16 Sectorización de fases, disminuyendo el valor Rang_Air

Para el vuelo 1 vemos que la identificación del despegue es bastante buena dado que el movimiento está muy bien definido y las variaciones parecen ser apreciables por distintos valores de Rang_Air. En cambio, en el vuelo 2, la fase de despegue se queda a la mitad ya que encontramos una zona conflictiva donde el dron desacelera como consecuencia de la sensibilidad del nuevo valor de Rang_Air.

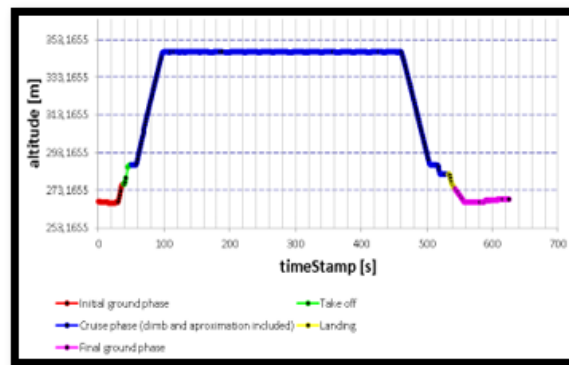
En cambio, para la identificación del aterrizaje sí que afecta gravemente a los dos vuelos. El hecho es que con cambios demasiado lentos o puntos singulares que entren en conflicto hacen que el programa piense que ya ha terminado la etapa, llegando a una fase de estabilización. Pero vemos claramente, que no es así y apenas se han identificado escasos puntos (casi imperceptible) en la región de *Landing*.

→ Aumentamos el valor `Vertical_tol_ground` en 2 (lo hacemos menos restrictivo)

- `Vertical_tol_ground` = 10
- `Rang_air` = 5
- `Discont` = 5



(a) Si Sectorización fases, vuelo 1



(b) Sectorización fases, vuelo 2

Fig. 5.17 Sectorización de fases, aumento del valor `Vertical_tol_ground`

En este segundo caso puede verse muy claramente como el análisis de la fase *landing* se estudia como un despegue invertido, ya que nos dice con una buena aproximación cuándo termina el despegue y cuándo empieza el aterrizaje.

De lo contrario, el valor de la variable `Vertical_tol_ground` nos asegura con buen margen el fin e inicio del despegue y aterrizaje respectivamente, pero pierde en precisión ya que, por ejemplo, en el caso del despegue no nos asegura con mucha exactitud el momento en que empieza (lo mismo sucede con la finalización del *Landing*).

5.3.1.2. Resultados para la identificación de maniobras de vuelo

Para este apartado no nos centraremos en los resultados según los valores de los parámetros de control. Más bien analizaremos los resultados ya asociados a los valores de las variables de control que hemos considerado adecuados en el apartado 4.5.2.

En las tablas 5.5 y 5.6 vemos todas las maniobras que se han llevado a cabo desde que se emprende el vuelo hasta que el *drone* vuelve a tocar el suelo. No consideraremos ni el despegue ni el aterrizaje para esta sección, aunque pueden verse representados en las tablas de resultados.

→ Tipología de errores

En general las maniobras que se describen son muy parecidas a las que esperábamos que el dron hiciera según el diseño del *flight plan* en MP (ver Fig. 5.1 y 5.2)

No obstante, ciertas descripciones no son demasiado acuradas o carecen de información ya que para ciertos eventos no se sabe si el movimiento es más de un tipo o de otro (debido al solapamiento de maniobras entre eventos).

Teniendo como referencia la tabla del vuelo 1, que contiene más tipos de maniobras.

Por ejemplo, en el evento 3 (162,981-164,98), se indica que el movimiento es un giro longitudinal de 356° a 359° habiendo recorrido 1,37 m de distancia. En teoría, son demasiado bajas ambas magnitudes como para que el algoritmo lo califique como tal.

En todo caso esos 3° tendrían que formar parte del *Yaw* del evento anterior, y esos 1,37 metros tendrían que formar parte del siguiente evento, que es un avance continuo. Entonces, podríamos decir que el evento 3, surge del solapamiento del final de una maniobra y el inicio de otra. Nos indica un cambio de transición

A pesar de ello es un evento que aporta muy poca información y lo consideraríamos como elemento sobrante de la tabla.

La idea de estos resultados es que nos indiquen hechos destacables.

Otra tipología de eventos superfluos, son aquellos que nos indican una misma maniobra con múltiples eventos. Es el ejemplo del descenso al final de la tabla (entre los instantes 266,984s – 283,987s). En realidad, solo hay un solo descenso en esos 17 segundos de diferencia. El problema está en que nuestro algoritmo es sensible a ciertas variaciones angulares, y parece que, durante la bajada, el dron iba oscilando entre 261° y 274°, una diferencia probablemente causada por vibraciones y ruido que no deberíamos tener en cuenta, pero que superan el umbral especificado.

→ Representación del vuelo 1

Tabla 5.5 Resultados de la identificación de maniobras para el vuelo 1 sin cabecera; enumeración de conjunto de eventos según la parte del vuelo

115,981 - 130,981 (15)	Ascent motion (Take Off)	Forward motion	262,95 - 282,09 (19,74)	3,1	133,99 --> 32,14 (-101,85)	1,32	0,21	1
131,986 - 157,98 (25,994)	Steady motion (282,12)	Forward motion	-	109,09	32,99 --> 34,44 (1,51)	-	4,23	
157,98 - 162,981 (5,001)	Steady motion (282,14)	Yaw motion	-	0,88	34,44 --> 356,81 (322,37)	-	0,58	
162,981 - 164,98 (1,999)	Steady motion (282,1)	Turn (Clockwise)	-	1,37	356,81 --> 359,86 (3,05)	-	0,69	
164,98 - 166,981 (2,001)	Steady motion (282,09)	Forward motion	-	6,69	359,86 --> 359,21 (-0,65)	-	3,34	
166,981 - 170,981 (4)	Steady motion (282,12)	Forward motion	-	20,96	359,21 --> 0 (-359,21)	-	5,24	
170,981 - 172,987 (2,006)	Steady motion (282,22)	Forward motion	-	16,84	0 --> 0,01 (0,01)	-	8,39	
172,987 - 177,983 (4,996)	Steady motion (282,23)	Forward motion	-	35,32	0,01 --> 359,8 (359,79)	-	7,07	
177,983 - 183,982 (5,999)	Steady motion (282,08)	Forward motion	-	14,73	359,8 --> 359,48 (-0,32)	-	2,46	
183,982 - 186,981 (2,999)	Steady motion (282,11)	Turn (clockwise)	-	3,92	359,48 --> 43,7 (-315,78)	-	1,31	2
186,981 - 190,981 (4)	Steady motion (282,18)	Turn (Clockwise)	-	19,16	43,7 --> 177,86 (134,16)	-	4,79	
190,981 - 205,983 (15,002)	Steady motion (282,28)	Forward motion	-	60,44	177,86 --> 188,85 (10,99)	-	4,03	
205,983 - 209,984 (4,001)	Steady motion (282,03)	Turn (Clockwise)	-	5,62	188,85 --> 359,99 (171,14)	-	1,4	3
209,984 - 210,981 (0,997)	Steady motion (282,19)	Lateral movement	-	6,01	359,99 --> 359,98 (-0,01)	-	6,03	
210,981 - 220,982 (10,00)	Steady motion (282,2)	Lateral movement	-	64,13	359,98 --> 359,32 (-0,66)	-	6,41	
220,982 - 228,989 (8,007)	Steady motion (282,06)	Lateral movement	-	20,9	359,32 --> 1,12 (-358,2)	-	2,61	4
228,989 - 232,981 (3,992)	Steady motion (282,03)	Turn (Clockwise)	-	15,07	1,12 --> 268,63 (267,51)	-	3,78	
232,981 - 249,984 (17,003)	Steady motion (282,17)	Forward motion	-	73,18	268,63 --> 273,74 (5,11)	-	4,3	5
249,984 - 258,982 (8,998)	Straight ascent motion	Steady motion	282,41 - 300,9 (18,49)	-	273,74 --> 272,78 (-0,96)	2,05	-	
258,982 - 266,984 (8,002)	Steady motion (300,9)	Steady motion	-	-	272,78 --> 273,18 (0,4)	-	-	6
266,984 - 269,981 (2,997)	Straight descent motion	Steady motion	301,48 - 297,66 (3,82)	-	273,18 --> 261,42 (-11,76)	1,27	-	
269,981 - 273,989 (4,008)	Straight descent motion	Yaw motion	297,66 - 292,02 (5,64)	-	261,42 --> 271,77 (10,35)	1,41	-	7
273,989 - 274,98 (0,991)	Straight descent motion	Yaw motion	292,02 - 290,62 (1,4)	-	271,77 --> 275,8 (4,03)	1,41	-	
274,98 - 278,981 (4,001)	Straight descent motion	Steady motion	290,62 - 284,91 (5,71)	-	275,8 --> 266,98 (-8,82)	1,43	-	8
278,981 - 279,98 (0,999)	Straight descent motion	Steady motion	284,91 - 283,55 (1,36)	-	266,98 --> 274,88 (7,9)	1,36	-	
279,98 - 283,987 (4,007)	Straight descent motion	Forward motion	-	1,39	274,88 --> 271,93 (-2,95)	-	0,35	9
283,987 - 287,985 (3,998)	Steady motion (282,21)	Turn (Counter-clock)	-	16,22	271,93 --> 214,68 (-57,31)	-	4,06	
287,985 - 308,981 (20,996)	Steady motion (282,27)	Forward motion	-	98,55	214,68 --> 210,89 (-3,69)	-	4,69	
308,981 - 331,981 (23)	Descent motion (Landing)	Forward motion	282,07 - 262,37 (19,7)	1,52	210,89 --> 209,07 (-1,92)	0,86	0,07	

Mientras que en la tabla 5.5 vemos los sets de maniobras realizados durante el vuelo, en la Fig. 5.18 vemos asignadas dichas maniobras según la parte del vuelo que corresponda.

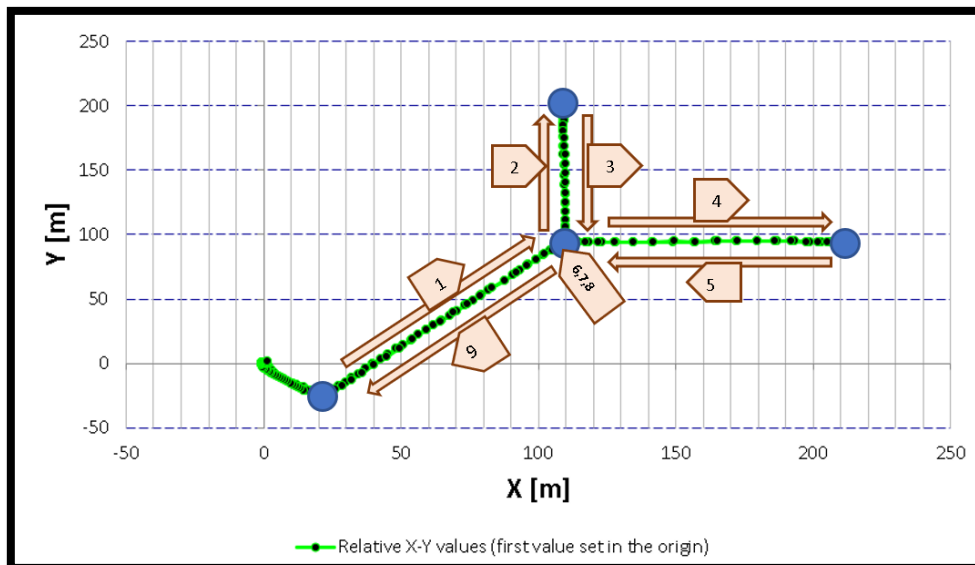


Fig. 5.18 Ruta del vuelo 1 con la asignación de etiquetas de la tabla 5.5

→ Representación del vuelo 2

Tabla 5.6 Resultados de la identificación de maniobras para el vuelo 2 sin cabecera; enumeración de conjunto de eventos según la parte del vuelo

29,986 - 50,989 (21,003)	Ascent motion (Take Off)	Forward motion	266,85 - 286,52 (19,67)	11,09	130,51 --> 153,23 (22,72)	0,94	0,53	
51,986 - 55,987 (3,999)	Steady motion (286,47)	Forward motion	-	13,78	153,39 --> 151,08 (-2,31)	-	3,45	1
55,987 - 57,988 (2,001)	Steady motion (286,42)	Turn (Counter-clockwise)	-	6,31	151,08 --> 89,07 (-62,01)	-	-	
57,986 - 96,991 (39,003)	Ascent motion	Forward motion	287,1 - 344,74 (57,64)	161,27	89,07 --> 80,76 (-8,31)	1,48	4,13	2
96,991 - 97,988 (0,997)	Ascent motion	Turn (Clockwise)	344,74 - 345,84 (1,1)	7,85	80,76 --> 89,8 (9,04)	1,1	7,87	
97,986 - 101,993 (4,005)	Steady motion (345,84)	Turn (Clockwise)	-	15,3	89,8 --> 148,04 (58,24)	-	3,82	3
101,993 - 105,987 (3,994)	Steady motion (346,53)	Forward motion	-	17	148,04 --> 153,66 (5,62)	-	4,26	
105,987 - 109,988 (4,001)	Steady motion (346,5)	Turn (Clockwise)	-	7,75	153,66 --> 219,73 (66,07)	-	1,94	
109,988 - 110,987 (0,999)	Steady motion (346,47)	Turn (Clockwise)	-	13,89	219,73 --> 223,42 (3,69)	-	13,9	
110,987 - 120,987 (10)	Steady motion (346,49)	Forward motion	-	33,31	223,42 --> 227,09 (3,67)	-	3,33	
120,987 - 124,988 (4,001)	Steady motion (346,51)	Turn (Counter-clockwise)	-	12,32	227,09 --> 172,56 (-54,53)	-	3,08	
124,988 - 127,987 (2,999)	Steady motion (346,4)	Forward motion	-	25,41	172,56 --> 156,96 (-15,6)	-	8,47	
127,987 - 131,987 (4)	Steady motion (346,51)	Turn (Counter-clockwise)	-	27,98	156,96 --> 99,58 (-57,38)	-	7	
390,988 - 394,993 (4,005)	Steady motion (346,38)	Turn (Clockwise)	-	16,3	41,06 --> 260,08 (219,02)	-	4,07	
394,993 - 414,989 (19,995)	Steady motion (346,52)	Forward motion	-	107,5	260,08 --> 258,16 (-1,92)	-	5,38	
414,989 - 418,987 (3,998)	Steady motion (346,38)	Turn (Counter-clockwise)	-	7,97	258,16 --> 53,09 (-205,07)	-	1,99	4
418,987 - 438,989 (20,002)	Steady motion (346,52)	Forward motion	-	90,61	53,09 --> 51,12 (-1,97)	-	4,53	
438,989 - 442,988 (3,999)	Steady motion (346,46)	Turn (Clockwise)	-	17,35	51,12 --> 270,91 (219,79)	-	4,34	
442,988 - 460,986 (17,998)	Steady motion (346,67)	Forward motion	-	105,67	270,91 --> 270,33 (-0,58)	-	5,87	
460,986 - 462,987 (2,001)	Steady motion (346,44)	Turn (Counter-clockwise)	-	6,76	270,33 --> 231,97 (-38,36)	-	-	
462,987 - 466,988 (4,001)	Descent motion	Turn (Counter-clockwise)	345,19 - 340,08 (-5,11)	8,13	231,97 --> 209,84 (-22,13)	-1,28	2,03	
466,988 - 503,988 (37)	Descent motion	Forward motion	340,08 - 287,46 (-52,62)	21,62	209,84 --> 203,89 (-5,95)	-1,42	0,58	
503,988 - 506,988 (3)	Steady motion (287,46)	Forward motion	-	2,08	203,89 --> 264,26 (60,37)	-	0,69	
506,988 - 510,987 (3,999)	Steady motion (286,45)	Turn (Clockwise)	-	15,1	264,26 --> 315,91 (51,65)	-	3,78	
510,987 - 517,987 (7)	Steady motion (286,49)	Forward motion	-	8,26	315,91 --> 324,2 (8,29)	-	1,18	
517,987 - 521,988 (4,001)	Straight descent motion	Steady motion	285,51 - 281,69 (-3,82)	-	324,2 --> 322,81 (-1,39)	-	-	6
521,988 - 533,989 (12,001)	Steady motion (281,69)	Forward motion	-	10,33	322,81 --> 322,5 (-0,31)	-	0,86	
533,989 - 558,986 (24,997)	Descent motion (Landir)	Forward motion	281,66 - 266,79 (14,87)	0,86	322,5 --> 321,42 (-1,08)	0,59	0,03	7

Se han dejado de mostrar el rango de puntos que va de 131,98s a 390,98s (ver tabla entera en [ANEXO L]).

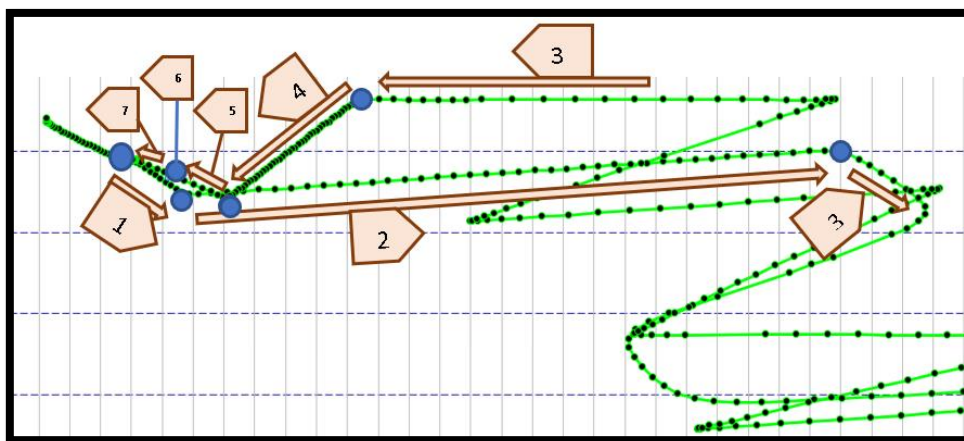


Fig. 5.19 Ruta del vuelo 1 con la asignación de etiquetas de la tabla 5.6

4.1. Resultados adicionales

A continuación, presentamos las dos tablas que contienen el último conjunto de resultados numéricos. Muchos de estos resultados ya se han comentado en el apartado 6.2.1.

Tabla 5.7 Resultados adicionales del vuelo 1

Total Flight time [s]	Landing Path Error [m]	Altitude. max. [m]	Ground Altitude [m]	Max speed in ascence [m/s]	Max speed in descence [m/s]	Max module speed in XY plane [m/s]
210	0,83	302,15	262,1	2,41	1,53	8,71

Tabla 5.8 Resultados adicionales del vuelo 2

Total Flight time [s]	Landing Path Error [m]	Altitude. max. [m]	Ground Altitude [m]	Max speed in ascence [m/s]	Max speed in descence [m/s]	Max module speed in XY plane [m/s]
510	0,01	346,67	266,49	2,12	1,52	8,79

Capítulo 6. Desenlace del estudio

6.1. Conclusiones

En primer lugar, hay que comentar que las trayectorias analizadas en el *tracking* se adecuan considerablemente bien según *flight plan* correspondiente. En la ruta 1 los errores son más pequeños, la trayectoria del dron sigue mejor el plan de vuelo porque tiene la instrucción de detenerse en un punto durante unos instantes, en cambio en el vuelo 2, el *drone* puede mantener una cierta velocidad con la condición de no sobrepasar 1m del WP.

Aunque esto le hace un poco más inexacto en los movimientos, la tendencia natural del hexacóptero está en aprovechar mejor las rutas efectuando pequeños recortes. No obstante, en el último tramo de aproximación final los valores de error han disminuido mucho y eso nos hace creer que en las fases iniciales y finales tiene una tendencia para actuar de forma más segura (con más exactitud).

La elección favorable entre exactitud en la ruta u optimización en el trazado de la ruta depende del campo donde se quisiera aplicar el uso de los UAV.

En cuanto a los cambios verticales, el dispositivo que mejor ha permitido monitorizar la altitud es el GPS (con EKF), con errores submétricos en *take off* y *landing*, pero con un mayor error en los cambios de altitud de las demás fases.

No obstante, otro dispositivo que también puede hacer frente a los valores de la exactitud del GPS es el sensor de presión 1 haciendo la conversión *presión-altura* con la ecuación hipsométrica.

En siguiente lugar, la relación entre movimientos y consumo de batería es clara, especialmente para el vuelo 2, que ha efectuado una serie de giros cerrados de forma continua durante un tiempo mucho más prolongado que el vuelo 1. En efecto, parece que el factor determinante para que el ratio de consumo sea más elevado es la continuidad de giros, los cuales hacen que los picos de intensidad también sean más continuos, haciendo que la rapidez con la que se consume la batería sea un poco más elevada que en la ruta 1. Por tanto, si se quiere aumentar la eficiencia energética, probablemente deberíamos emplear movimientos más suaves y lentos.

Creemos que los resultados de la identificación de fases de vuelo han sido muy favorables, ya que, con un nivel de precisión adecuado, el dron, es capaz de identificar las fases más comunes en cualquier tipo de vuelo. Nos han ayudado

a clasificar cada tipo de maniobra en un sector u otro y ver las tendencias en función de donde se encuentre.

Por otra parte, aunque se han conseguido descifrar las maniobras empleadas por el dron en todos los segmentos del vuelo, creemos que todavía se presentan ambigüedades que hacen que el nivel de precisión de la identificación sea bajo.

En cambio, sí que hemos podido comprobar que la tendencia del dron para efectuar los giros reside en técnicas como el *fly-by* en donde no es necesario pasar por el *waypoint* específico, lo cual es eficiente ya que supone llegar antes al destino. Aunque creemos que también supone un recorte en el gasto energético no tenemos datos para comprobarlo, en ese caso miraremos de proponer métodos para poder corroborarlo en futuros estudios.

6.2. Futuros estudios y mejoras

6.2.1. Detectar deficiencia de dispositivos

Las aceleraciones y cambios angulares nos han permitido ver grosso modo el comportamiento de la aeronave. Se han podido asociar giros y cambios de velocidad con simples picos de aceleraciones.

Estas lecturas se pueden aprovechar para determinar cuan deteriorado está el sistema propulsor o la IMU; basta con realizar la ruta de un *flight plan* específico de forma periódica y, eligiendo ciertos puntos de interés donde haya picos considerables de aceleración, comparar dicha región de cada vuelo observando que, a lo largo del tiempo ese pico pierde intensidad, lo cual se traduce como una pérdida de eficacia en el movimiento o en las lecturas de la IMU.

El estudio del deterioro de las baterías podría seguir un proceso similar. Realizando varios vuelos idénticos de forma periódica con las mismas condiciones e ir viendo que, o bien el vuelo se realiza de forma más lenta teniendo un ratio de consumo más o menos igual en cada vuelo, o bien realizando el vuelo con la misma velocidad pero teniendo un ratio de consumo más elevado a lo largo del tiempo.

Una de las formas que tendríamos para evaluar la eficacia de la transmisión de mensajes entre los sensores y la placa controladora es mirando la cantidad de mensajes fallidos según el *sampling time*, monitorizado en cada una de las tablas que contienen las muestras de los mensajes. Lo ideal sería, nuevamente, realizar

el mismo vuelo de forma periódica e ir viendo que en el tiempo cada vez hay más porcentaje de mensajes fallidos.

6.2.2. Detección de mayor consumo según maniobras

Una de las formas de comprobar con mucha más seguridad el consumo de baterías en función de la agresividad de los movimientos, es creando un plan de vuelo que suponga varias paradas y aceleraciones, y otro con una trayectoria similar, pero con movimientos más fluidos en vez de tener que parar para efectuar giros.

Podría ser el caso de un plan de vuelo con una trayectoria cuadrada y otro con una trayectoria circular. El cuadrado y el círculo deberían tener el mismo perímetro y tendríamos dos opciones de efectuar esos vuelos.

Una sería efectuar los dos vuelos durante un periodo de tiempo determinado y después ver la distancia total recorrida, el consumo global de la batería y ratio de consumo.

Y la otra implica los mismos planes de vuelo, pero en vez de imponer la finalización del vuelo dado un tiempo de vuelo, imponer un límite en la distancia recorrida y que entonces finalice el vuelo.

Haciendo estas pruebas podríamos ver realmente la eficiencia de las maniobras en tiempo y distancia en función del consumo de baterías.

6.2.3. Solución para mejorar la identificación de maniobras y fases de vuelo

En último lugar, se podría considerar la inclusión de otras variables, especialmente las de aceleración, para detectar cuándo realmente empiezan a producirse cambios de movimiento como giros y frenazos. Esto reduciría el error en cuanto a solapamiento de maniobras, ya que tendríamos más información a la hora de saber cuándo se está produciendo una transición de movimiento

Por otro lado, podríamos extender el algoritmo con nuevas variables de control para identificar la fase de ascenso posterior al despegue y la fase de aproximación previa al aterrizaje. Esto aumentaría en 2 el número de fases a identificar y aun así no tendrían por qué darse esos casos como sucede en el vuelo 1, donde nada más despegar ya empieza la fase crucero.

Bibliografía

- [1] M. Conceptual and D. Performance, “Sistema Mundial de Navegación Aérea ~ Elementos ~,” no. May, pp. 4–8, 2009.
- [2] O. De Aviación and C. Internacional, “PROYECTO Plan mundial de navegación aérea 2016 – 2030,” pp. 1–136, 2016.
- [3] K. Krajčec, D. Nikolić, and A. Domitrović, “Praćenje stvarnih performansi zrakoplova prema podacima iz leta,” *Teh. Vjesn.*, vol. 22, no. 5, pp. 1337–1344, 2015, doi: 10.17559/TV-20131220145918.
- [4] “GMV _ Sistemas de navegación aérea basados en GNSS.” GMV Innovating solutions, Madrid, 2016.
- [5] “GMV _ estandarización de drones con los sistemas GNSS.” Solutions, GMV Innovating, Madrid.
- [6] “Official U.S. government information about the Global Positioning System (GPS).” U.S. GNSS public services, California, 2020.
- [7] “RQ-1 Predator Medium Altitude Endurance (MAE) UAV.” [Online]. Available: <http://www.fas.org/irp/program/collect/predator.htm>.
- [8] A. Energ, “Optimización del ciclo de vida de las baterías de litio,” pp. 38–39.
- [9] Agencia Estatal de Seguridad Aérea, “Marco regulatorio para operaciones con Drones,” 2017.
- [10] M. Frassia *et al.*, “Sistema Gauss-Krüger,” *file:///C:/Users/Pol-Pavilion (HP)/Desktop/Drone docs Res. data. Artic. fotos y más/Actuaciones en vuelo drones _ Not. la Cienc. y la Tecnol. (Amazings® _ NCYT®).html*, pp. 1–20.
- [11] MAVLink Developer Team, “Messages (common) · MAVLink Developer Guide.” [Online]. Available: <https://mavlink.io/en/messages/common.html>.
- [12] J. M. Wallace and P. V. Hobbs, *Atmospheric Science: An Introductory Survey: Second Edition*. 2006.
- [13] “ISO - ISO 2533_1975 - Standard Atmosphere.” .
- [14] M. L. Salby, *Fundamentals of Atmospheric Physics*, Renata Dmo., vol. 61, no. 9. AP, 2011.
- [15] V. Sazdovski, T. Kolemishvska-Gugulovska, and M. Stankovski, *Kalman filter implementation for unmanned aerial vehicles navigation developed within a graduate course*, vol. 16, no. 1. IFAC, 2005.
- [16] A. Pascual, “EKF y UKF : dos extensiones del filtro de Kalman para sistemas no lineales aplicadas al control de un péndulo invertido,” pp. 1–35, 2006.

- [17] R. B. Langley, "Dilution of Precision," *GPS World*, vol. 10, no. May, pp. 52–59, 1999, [Online]. Available: http://www.ceri.memphis.edu/people/rsmalley/ESCI7355/gpsworld_may99.pdf.
- [18] "Actuaciones en vuelo de drones — Noticias de la Ciencia y la Tecnología (Amazings® / NCYT®)." [Online]. Available: <http://noticiasdelaciencia.com/not/12949/actuaciones-en-vuelo-de-drones/>.
- [19] P. R. Chic, "UAS_Components_MindMap," *UPC*, no. "UAS_Components," 2017.
- [20] J. P. Carro, "TFG: Analisis y comparativa de autopilotos de código abierto para sistemas no tripulados," *UPC*, vol. 1, p. 55, 2017.
- [21] PIXHAWK, "Pixhawk v2 Feature Overview," vol. 2, [Online]. Available: http://www.hex.aero/wp-content/uploads/2016/07/DRS_Pixhawk-2-17th-march-2016.pdf.
- [22] "Erle-copter components." GitBook, 2019, [Online]. Available: <https://erlerobotics.gitbooks.io/erle-robotics-erle-copter/content/es/components/gps.html>.
- [23] Batteryuniversity, "BU-601: How does a Smart Battery Work? – Battery University." pp. 1–10, 2019, [Online]. Available: http://batteryuniversity.com/learn/article/inner_workings_of_a_smart_battery.

ANEXOS

ANEXO A. Elementos integrantes en la estación tierra del sistema UAS

A.1. Softwares y firmwares

El principal punto a tener en cuenta para que la estación de tierra pueda comunicarse con el dron, es que el software que usemos en nuestro PC (Estación de tierra) sea compatible con el *firmware* de la placa controladora del dron. Para ello hace falta que se comparta el mismo protocolo de comunicaciones, MavLink.

El *firmware* es, el software específico dentro del *autopilot*, que tiene un código de bajo nivel capaz de interactuar con los dispositivos de control del dron.

Haremos uso de Mission Planer, un GUI (*Graphic User Interface*) que nos ayudará a controlar los parámetros más básicos del *firmware* de una forma mucho más simple e intuitiva, así como llevar a cabo el *flight plan*

A.1.1. Mission Planner

Se trata del software definido como elemento de control principal en la estación de tierra, y que podremos manejar fácilmente como usuarios. Dentro de sus múltiples opciones se tendrá especial interés en:

- Diseñar un plan de vuelo mediante el posicionamiento de *waypoints*
- Calibrar los elementos de principal interés que constituyen el UAV, tales como la IMU y otros sensores.
- Monitorizar en tiempo real las variables navegación y estado del sistema.

Aunque también podríamos hacer una simulación previa tipo SITL (*Software in the loop*) antes de llevar el ejercicio práctico y/o configurar los valores de PID para variar el comportamiento del sistema de control del dron.

A continuación, mostramos la interfaz del menú principal de Mission Planner.



Fig. A.1 Interfaz gráfica de *Mission Planner*, ventana de monitorización.

A.2. Hardware y componentes físicos

A.2.1. Antena de telemetría (433 MHz)

El uso principal de esta antena consiste en el envío de datos telemétricos a la estación tierra, es decir, la recepción de la situación actual de nuestro *Unmanned Vehicle* (en nuestro PC), de datos como pueden ser: estado de las baterías, localización espacial, velocidad, valores del acelerómetro o giróscopo, presiones y temperaturas, modo de vuelo actual, etc.



Fig. A.2 Conexión entre módulo de telemetría y estación tierra

Todos los parámetros que acabamos de comentar se enviarían en tiempo real actualizándose en pequeños intervalos de tiempo. En principio cuanto menor sea el intervalo mejor, ya que se recibirá más información por unidad de tiempo, pero también hay que tener en cuenta que aumentará la posibilidad de que se pierdan paquetes de esos datos.

En nuestro caso tenemos asignado un *sampling time* de 1 segundo, admitiendo una tolerancia del 10%, entre los paquetes del mismo tipo de información.

El modelo de antena usado es el *3DR Radio v2* tanto para la estación tierra como para el UA. Es importante remarcar que, aunque la antena permite trabajar entre 433 y 915 MHz, en Europa está estipulado por ley que las frecuencias a usarse deberán ser las correspondientes a 433 MHz.

Tabla A.1 Especificaciones técnicas de la antena de telemetría (433 MHz)

SiK Telemetry Radio	
Sensibilidad del receptor	-121 dBm
Transmisión max. potencia	20dBm (100mW)
Velocidades de datos aéreos	250 kbps
Encuadre del protocolo <i>MAVLink</i>	Si
Ciclo de trabajo	Configurable (código abierto)
Alcance máximo	500m – 1,5Km (depende del escenario)
Módulos de radio	HM-TRP / Si4432
Microcontrolador	Si1000 8051

En esta antena disponemos de una entrada micro-USB para poder conectar al vehículo/ordenador. Pero en caso de que tengamos complicaciones o simplemente queramos ajustar otra configuración de conexión, podemos acceder a los 6 pines de debajo del módulo de la figura A.3.

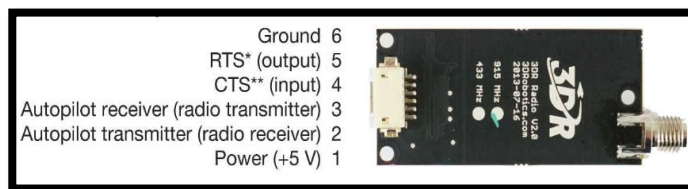


Fig. A.3 Puertos de conexión del módulo telemetría.

A.2.2. Emisora radiocontrol

Este elemento resultará de vital importancia en caso de que haya alguna incidencia o emergencia en la que se tenga que tomar acción inmediata. En caso de tener una incidencia

pasaríamos al modo manual, dando el control por completo al usuario que controle la emisora y aquí tomar las acciones pertinentes para evitar un desastre.

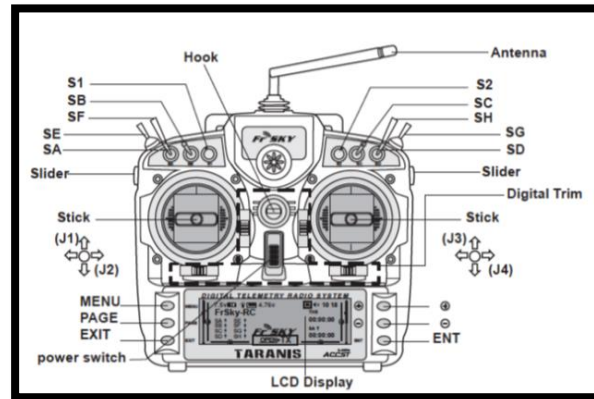


Fig. A.4 Controles de la emisora *TARANIS* (FrSKY)

Esta emisora forma parte de un plan secundario al de una estación terrena automatizada. De hecho, la estación de tierra principal (conjunto Mission Planer y antena de telemetría) no está vinculada directamente al control remoto. En todo caso, la emisora de radiocontrol está conectada por un enlace de 2,4GHz de frecuencia al receptor que llevará el dron.

El camino es solo de ida. Cuando se selecciona el modo manual, ya sea al inicio o en medio del vuelo, el dron deja de seguir el plan de vuelo y se pone a disposición de las instrucciones que pasen por el emisor de radiocontrol.

Tabla A.2 Especificaciones técnicas de la emisora radiocontrol

Emisora por radio TARANIS (FrSKY)	
Alimentación	5-15 V
Canales	16
Frecuencia	2,4 GHz
Alcance	3 Km (dependiendo del escenario)
Corriente máxima	0,26 A
Rango de trabajo temperaturas	-10 – 60 °C
Memoria máxima	60 MB
Transmisor	FrSky X9D, telemetría

ANEXO B. Elementos físicos en el vehículo no tripulado del sistema UAS

B.1. Motores y hélices

Disponemos de 6 propulsores eléctricos sin escobillas. El hecho de no tener escobillas hace que el rendimiento del rotor mejore, ya que disminuye el consumo al no haber un elemento de fricción, que a su vez implica menos desprendimiento de calor. Además, resulta en la reducción de ruido de los motores. Un aspecto destacable si consideramos la inminente implementación de UAS dentro de las ciudades.



Fig. B.1 Conjunto de 6 rotores eléctricos DJI con las hélices incorporadas

A continuación, presentamos la tabla *B.1*, donde podemos ver la características y recomendaciones por el fabricante:

Tabla B.1 Especificaciones de los rotores *DJI 2312E*

Rotores DJI 2312E 960KV	
Tipo de batería recomendada	3-4S Lipo Battery
ESC Recomendado	420 LITE
Helice recomendada	9450
Tow Recomendado	350g/rotor (SL)
Thrust máx.	710g/rotor (11.1V, SL)
Tamaño	23x12mm

Las hélices representan el elemento más primordial en la aerodinámica, indispensable para producir la sustentación del hexacóptero. Hay de diversos tipos y formas, sus principales características son:

- **Material (fibra de carbono, plásticos, madera...)**

Por un lado, tenemos el material, un balance común suele ser que la tenacidad y resistencia del material sea inversamente proporcional al peso, aunque también es verdad que cada vez la ciencia de los materiales avanza más rápido y es posible encontrar materiales *composite* de bajo peso y con muy buenas prestaciones, eso sí, adquirirlo podría ser realmente caro.

Actualmente son muy comunes las hélices de plástico dadas sus buenas propiedades físicas con relación a su peso.

- **Orientación del *pitch* y curvatura de la hélice**

Por otro lado, tenemos de las hojas, cuanto menor sea la curvatura a lo largo de la hoja más rápido podrá reaccionar el dron en cuanto a aceleraciones y movimientos (ideal para un modo acrobático), por lo contrario una inclinación mayor hace que los niveles de sustentación sean mayores ya que posibilita mover mayores cantidades de aire, por tanto, logran menos velocidad punta que las aspas más llanas (con menor ángulo de *pitch*) y también requiere de mucho más consumo para conseguir la misma velocidad angular que una aspa llana.

- **Número de hélices**

El efecto del número de hélices de la hoja también acaba siendo muy parecido al del ángulo *pitch*, Principalmente, añadiendo un aspa más se añade más superficie alar, lo que se traduce con más empuje a costa de un mayor arrastre y por tanto un mayor consumo de energía.

- **Longitud de la hélice**

Alargar la longitud del aspa permite mover más aire y llegar a más altas velocidades, pero implica un coste de consumo algo mayor también, por no mencionar el pequeño incremento en el peso.

Al final, la selección de cada característica vendrá condicionada por la utilidad que se le quiera dar al UAV y el presupuesto monetario.



Fig. B.2 Hélices de sentido horario y antihorario

B.2. ESC (Electronic speed controller)

Como ya se ha comentado, la placa controladora se encarga de sincronizar el funcionamiento de cada componente que integre el dron. Una parte fundamental de ese funcionamiento es la administración adecuada de voltaje para componentes como los rotores, y aquí es donde interviene el ESC. Entonces, lo que realmente realiza un ESC es variar la velocidad de giro de un rotor en función de las maniobras que quieran realizarse. Por este motivo también se le llama *variador* al ESC.

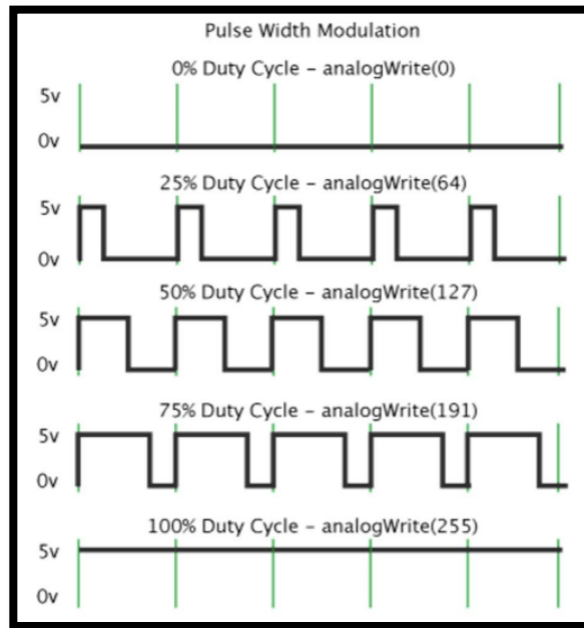


Fig. B.3 Ejemplos de señales digitales PWM

Esa información se transmite mediante señales *Pulse Width Modulation* [19] .

En la figura B.3 podemos apreciar distintos tipos de mensajes mediante modulación de pulsos digitales. El parámetro de mayor interés es el *duty cycle* de una señal digital, también llamado ciclo de trabajo. Cada ciclo de trabajo estará asociado a una velocidad angular concreta; la que corresponda a la maniobra global que se quiera realizar.

En la arquitectura de un sistema UAV puede contemplarse la disposición de un ESC para cada rotor. Si disponemos de un octacóptero, probablemente se vaya a hacer uso de 8 ESC. Cada rotor será controlado de forma independiente acorde con las instrucciones del *autopilot*, que será quien envíe los pulsos PWM.

En la siguiente imagen podemos ver los ESC que lleva nuestro dron para el control de rotores.



Fig. B.4 Sistema ESC utilizado en nuestro dron.

B.3. Módulo GPS (Here 2)

Los datos GPS, aunque no son esenciales para poder volar, hoy en día sí que lo son para navegar, puesto que el error de la posición y otros parámetros quedan significativamente reducidos.

Tener una o varias constelaciones de satélites a nuestra disposición, a pesar de que se encuentran decenas de miles de kilómetros, es el motivo por el cual podemos tener tanta exactitud a la hora de volar con un dron. En principio, la antena que se integra en un dron es el módulo externo, alejado de los demás elementos para evitar posibles interferencias o ruido.

En nuestro caso hemos usado el módulo *Here2*, que se conecta por un puerto serie I2C. También incluye un magnetómetro muy útil para determinar la dirección del dron y así poder aplicar las correcciones pertinentes.

Tabla B.2 Especificaciones del receptor GNSS

Modulo GPS Here2 (Chip Ublox Neo-M8N)	
Sensibilidad navegación	-167 dBm
Puertos	UART/I2C/CAN
Error	±2.5 m
Temperaturas de operación	-140 – 85 °C
Tiempo de arranque inicial	25 s

Cabe destacar que la antena GPS en si sola no determina la cualidad de la información recibida. La interoperabilidad de datos es primordial para una correcta determinación de la posición, y por tanto es estrictamente necesario que los datos que nos ofrecen los satélites sean posteriormente procesados y comparados con sistemas inerciales, sensores de temperatura y presión (mediante filtros de Kalman) e incluso sensores de posición para que realmente nos ofrezcan una buena resolución de nuestra posición.



Fig. B.5 Antena GPS receptora Here 2

Finalmente, hay que mencionar que el GPS usa la banda de trabajo L1 (1575.42 Mhz). Aporta información sobre el estado del GPS y otros datos en los cuales no entraremos en detalle, pero que permiten saber la ubicación del dron en un instante determinado.

B.4. Receptor radiocontrol (2,4 GHz)

En el apartado de *Estación Tierra* ya hemos hablado sobre la emisora radiocontrol, que funciona a una frecuencia de 2,4 GHz. Tiene el principal objetivo de permitir al usuario controlar la aeronave de forma manual. En este apartado hablaremos desde el punto de vista del receptor, ubicado dentro del vehículo no tripulado.

Un receptor, recibe la señal enviada por la emisora y la traduce para que los variadores, servos, y demás elementos hagan la función que tenga que llevarse a cabo según las acciones del usuario. Para que el receptor pueda transmitir la señal a cada dispositivo tendremos 16 canales distintos, en cada uno de ellos se asocia un elemento o dispositivo distinto del UAV. Por tanto, teniendo 16 canales podremos controlar 16 elementos de control.



Fig. B.6 Receptor de señal 2,4 GHz de la emisora radiocontrol

Para nuestro proyecto no es necesario usar el receptor/emisor radiocontrol, aun así, tenemos especial interés en prevenir cualquier tipo de incidente. Por este motivo, es razonable instalar un controlador externo capaz de dejar el dron directamente en manos del usuario.

B.5. Antena de telemetría (433 MHz)

El sistema de telemetría usa una frecuencia de 433MHz y es bidireccional; por un lado, el dron envía todos los datos de telemetría a la estación tierra, y la estación de tierra tiene la posibilidad de cambiar el modo de vuelo o enviar ciertas instrucciones al UAV.

Por tanto, por parte del dron solamente se envían los datos de telemetría, que son procesados por el piloto automático, y luego se envían a la estación de tierra donde podrán ser monitorizadas.

La antena usada para el dron es exactamente la misma que para la estación de tierra.



Fig. B.7 Sensor laser de distancia SF11/C

B.6. *Sensor de distancia laser*

En este proyecto, incorporaremos un sensor de distancia láser por primera vez en nuestro hexacóptero. Se trata de un modelo capaz de medir hasta 120m con un error de 10 cm en el mejor de los casos.

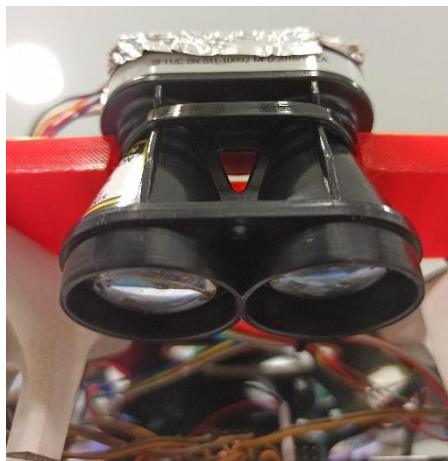


Fig. B.8 Sensor laser de distancia SF11/C

Actualmente, en el sector aeronáutico, estos dispositivos y los sensores ultrasonidos se usan para las mediciones de altura en las fases de despegue y aterrizaje, ya que en alturas cercanas al suelo el modelo de cálculo de la altura mediante variables como la presión y temperatura no se adaptan demasiado bien a la realidad hasta no sobrepasar un umbral considerable.

Tabla B.3 Especificaciones sensor laser SF11/C

Sensor laser de distancia SF11/C	
Rango	0,2 – 120 m // 2 – 40 m (sobre el agua)
Resolución	1cm
Velocidad de lectura	20 Hz
Exactitud	±10cm
Outputs	Serial I2C (400KHz)
Potencia del laser	14 mW – 20 mW
Voltaje de entrada	5 ± 0,5 V
Corriente de entrada	200 mA (Max.)
Temperatura de operación	0 – 40°C

B.6. Piloto automático (*PixHawk 2.1 Cube*)

Las placas controladoras, también denominadas pilotos automáticos, son dispositivos electrónicos con múltiples entradas y salidas, que interactúan con la mayoría de los sensores y elementos de control del UAV. Su finalidad es sincronizar adecuadamente el funcionamiento de cada elemento en función de la acción que se deba hacer.



Fig B.9 Placa controladora *Pixhawk 2.1 Cube*

Cuando hablamos de acciones, nos referimos tanto al cumplimiento de un plan de vuelo que haya elaborado un usuario, como en las acciones intrínsecas del dron para permanecer estable durante todo su vuelo. Estas últimas instrucciones las realiza el propio piloto automático con mediciones inerciales. De hecho, uno de los aspectos principales en cualquier sistema de control, es mantener la estabilidad de la planta en todo momento, entre otros.

Aunque este modelo ya nos venía integrado en el modelo de dron como propuesta de la escuela, cabe decir que en anteriores estudios, se han hecho comparativas entre la PixHawk 2.1 y la PixHawk 4, la cual resulta ser más potente a cambio de no tener un cubo contra IEM (Interferencias Electromagnéticas). Pero el caso es que, la diferencia de absorción de ruido entre las dos placas parece poco significativa [20].

En la tabla 3.1 podemos observar los parámetros técnicos de la placa [21] y ver que, efectivamente, cumple con los requisitos mínimos para navegar con seguridad. Nos referimos a requisitos mínimos el hecho de que disponga de los dispositivos más básicos para la navegación, IMU y GPS a parte de los puertos para las conexiones con los demás componentes.

Tabla B.1 Especificaciones técnicas de la PixHawk 2.1 Cube

Pixhawk 2.1 Cube	
Versión	FMU v3 (<i>Flight management unit</i>)
Memoria	2 MB
Procesador FMU	32bit STM32F765 Cortex-M7 Clock: 168 MHz RAM 256 KB
Procesador IO	32bit STM32F103 Cortex-M3 Clock: 24 MHz RAM 8 KB
IMUs	3
Magnetometro	3 (uno por IMU)
Sensor de presión	2 (MS5611)
Periferico de comunicación GNSS	1 (Here 2)
Compatibilidad GNSS	GPS / GLONASS /GALILEO /BeiDou
Outputs PWM/Servos	14 (8FMU, 6 Failsafe)
Puertos I2c	2
Puerto SPI	1
Puerto CAN	2
Fabricante	ProfiCNC

B.6. Baterías (Turnigy High Capacity 6600mAh)

Las baterías representan la principal fuente de energía que alimenta a todo el sistema del dron. Hay que tener en cuenta que hoy en día representan el mayor cuello de botella, no solo en sistemas UAS. Eso se debe al alto consumo que requiere el dron para realizar según qué maniobras (en general, hasta un descenso consume bastantes recursos).

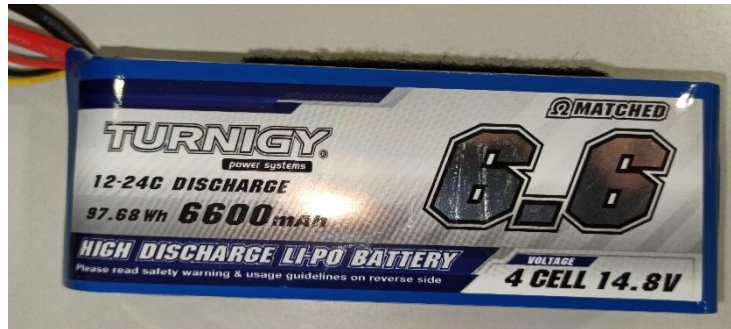


Fig B.10 Batería LiPo 6600 mAh utilizada en nuestro dron

Es importante saber interpretar la información que nos ofrecen las baterías, ya que algunos datos no son fácilmente deducibles si no hemos visto ninguna antes [22]:

-**Capacidad:** Es un valor numérico que se suele dar en *mAh*. Nos indica la cantidad de corriente que es capaz de proporcionar durante una hora (hasta que la batería quede sin recursos).

- **Las “S” (celdas):** actualmente una batería tipo LiPo está estructurada en diversas celdas, usando un método de carga y descarga proporcional en cada celda, que permite un mejor rendimiento de la batería global. Por tanto, el número que acompañe la S, corresponderá al número de celdas (y cada una con un mismo valor de voltaje).

- **Las “C” (tasa de descarga):** Es un índice que, multiplicado por la capacidad, nos da el valor de corriente máxima que la batería podría suministrar a uno o varios elementos del sistema electrónico, normalmente superior al de su capacidad (si no es 1C). El *tradeoff* de este parámetro está en la proporcionalidad inversa entre tiempo de vuelo y corriente suministrada.

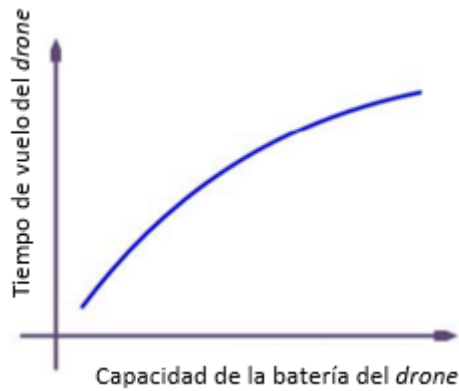


Fig B.11 Relación entre tiempo de vuelo y capacidad de una batería proporcional a su peso.

Aunque disponer de baterías con más capacidad puede parecer una solución para poder extender los tiempos de vuelo, es un hecho erróneo ya que, aumentar la capacidad también implica aumentar el peso de la propia batería, y por tanto el peso del dron que la lleve consigo. Dicho peso puede jugar en nuestra contra dado que el tiempo de vuelo es inversamente proporcional al peso del dron [19].

Dado un determinado dron y escenario, podemos representar el siguiente gráfico en función de distintos tipos de baterías que pueda llevar equipado, donde la capacidad de una batería es proporcional a su propio peso.

Tabla B.2 Especificaciones de la *Turnigy High Capacity 6600mAh*

<i>Turnigy High Capacity battery</i>	
Capacidad	6600 mAh
Numero de celdas	4 / 14,8V
Descarga constante	79,2 A (12C)
Descarga máxima (10 seg.)	158,4 A (24C)
Peso	598 g
Volumen	144 x 42 x 51

➔ Generación de baterías inteligentes

Recientemente están saliendo al mercado las primeras baterías inteligentes para drones, capaces de medir su estado y auto regularse progresivamente. Esto soluciona los problemas de optimización de recursos de las baterías más comunes actualmente. También incorporan sistemas de protección frente a posibles cortocircuitos o sobrecargas.



Fig. B.12 *Smart battery del DJI Matrice 600 Pro (batería LiPo + BMS)*

El concepto de BMS (Battery management system), es el nuevo elemento aplicado a las baterías *LiPo* convencionales, que en su globalidad está permitiendo optimizar los recursos energéticos a través de un software que opera un sistema electrónico acoplado a la misma batería [23].

Para este proyecto sería realmente interesante hacer un estudio más profundo sobre las baterías, teniendo una de tipo inteligente que pueda proporcionarnos datos sobre su eficacia en función de los tiempos de carga, etc.

De momento tendremos que usar una *LiPo* convencional, obteniendo los valores de voltaje, intensidad y almacenamiento. Consideramos que estos parámetros son suficientes para nuestro estudio.

ANEXO C. Elección de mensajes MavLink de interés para las lecturas del vuelo

MavLink puede proporcionar más de 200 tipos de mensajes. Para elegir los mensajes que nos pueden interesar hemos tenido que hacer un repaso uno por uno en la página web de los desarrolladores

Antes de entrar a mirar los parámetros puramente de la navegación, es primordial poder monitorizar el estado del sistema del dron. Se trataría de ver el estado de las funciones vitales como el estado de las baterías, modo de vuelo, reconocimiento del *frame* así como del *firmware*. Si el estado de uno de estos mensajes se reconoce como crítico durante vuelo, el dron podría sufrir problemas para sustentarse en el aire. O bien si todavía está en el suelo probablemente no llegue a despegar.

A continuación, vamos a mencionar algunos de los mensajes que nos ofrece MavLink que puedan ser de ayuda para el análisis de estado y navegación

C.1. Mensajes de estado del sistema

Lo que buscamos en estos mensajes es poder verificar que el funcionamiento del UAV es correcto mediante señales de estado del sistema. Los datos que pueden venirnos bien para hacer dicha valoración son los siguientes:

Heartbeat (#0):

- **Autopilot**, nos dice la versión del firmware con la que el piloto automático va a funcionar. Verificando este estado en *ARDUPILOTMEGA* sabremos que la lectura del mensaje es buena.
- **Base Mode**, principalmente nos dice si el vehículo está armado o no. Las etapas de vuelo tendrán que corresponder al modo *CUSTOM_MODE_ENABLED* *GUIDED_ENABLED* *STABILIZE_ENABLED* *MANUAL_INPUT* *SAFETY_ARMED*
- **Custom mode**, proporciona información del modo de vuelo. Normalmente obtendremos la lectura del modo *COPTER_MODE_AUTO* o *MANUAL* dependiendo si el dron está siguiendo un *flight plan* a través de una antena de telemetría o dirigido por una emisora radiocontrol. En fases iniciales es posible que esté en el estado *LOITER* por defecto.
- **System status**, deberá permanecer en *ACTIVE* durante todo el vuelo, excepto en fases iniciales que puede que aparezca como *STANDBY*
- **MavLink Version**, nos dice la versión del protocolo *mavlink* que se está usando para el envío de paquetes de datos.

Sys status (#1):

- **Onboard control_sensors enabled** y **Onboard control sensors health**, nos indicaran el estado de los sensores marcando 1 si están habilitado y fuera de errores.
- **Battery voltage**. obtenemos el valor de voltaje de la batería por cada unidad de tiempo. Podremos apreciar el consumo en unidades de potencia, voltaje, intensidad o en tanto por ciento respecto a la batería total.
- **Battery_remaining**, nos da la cantidad de energía que queda en las cargas. Se deberá tener en cuenta un umbral a partir del cual , el dron empiece una maniobra RTL (*Return To Launch*) para prevenir, en pleno vuelo, una posible interrupción de cualquier elemento esencial.

GPS Raw Int(#24):

- **Satellites visible**. El número de satélites para que el sistema de navegación GPS funcione de forma fiable es de 16. Aunque puede ser que nos encontremos rodeados de elementos interferentes y el número de satélites visibles disminuya. En ese caso tendríamos en cuenta que la exactitud del sistema se vería reducida.

C.2. Mensajes para la navegación

Estos mensajes nos muestran principalmente el valor de lectura de los sensores y dispositivos por cada unidad de tiempo UNIX. La frecuencia del *data sampling* está configurada a 1 muestra por segundo. En algunos casos el valor ya estará corregido.

Scaled IMU(#26):

- **XYZ lin acc**. Vemos los valores de la aceleración lineal leídos por la IMU 1 (acelerómetro) en los 3 ejes espaciales centrados en el dron. Podremos usar este parámetro como variable en una ecuación de movimiento y ver la diferencia de la posición y velocidad con el GPS. Esta unidad la obtenemos en *mG*.
- **XYZ angular acc**. Obtenemos valores de la aceleración angular en *mrad/s* leídos por la IMU 1 (acelerómetro) alrededor de los 3 ejes espaciales.
- **XYZ mag**. Obtenemos valores del campo magnético en leídos por la IMU 1 (magnetómetro) en de los 3 ejes espaciales.

Scaled Pressure(#29):

- **Pres abs.** Obtenemos los valores de presión estática en *hPa* del sensor de presión 1. Este parámetro nos permitirá calcular los cambios de altura a lo largo del vuelo si los metemos en la ecuación del modelo troposférico.
- **Temperature.** Más que la temperatura exterior, lo que se mide es la temperatura del *heating resistor*, que calienta ligeramente las IMUs para que realicen un trabajo óptimo. Este parámetro podría permitirnos calcular los cambios de altura a través de la ecuación hipsométrica.

Attitude(#30) :

- **Roll, Pitch y Yaw.** Nos indican el grado en radianes que se desplaza el UAV en función de los movimientos que ejecute el dron. Estos parámetros están corregidos por el EKF (*Extended kalman filter*), con lo cual son variables corregidas.
- **Rollspeed, Pitchspeed y Yawspeed.** Nos indican la velocidad en *rad/s* de los parámetros anteriormente indicados. Estos parámetros están corregidos por el EKF (*Extended kalman filter*), con lo cual son variables corregidas.

Global position int (#33):

- **Lat, lon, alt.** Estos mensajes nos indican la localización de nuestro dron en grados, para la latitud y longitud, o en metros, para la altitud. El GPS se encarga de transmitir esa información.
- **Vx, Vy, Vz.** Obtenemos la velocidad (*Ground speed*) en *cm/s* a lo largo de los ejes X, Y y Z, procesada ya por el filtro de *Kalman*.
- **Hdg.** El *heading* se encarga de indicarnos la orientación de nuestro dron. Son datos ya procesados por el filtro de *kalman* y el input del magnetómetro.

Scaled IMU 2 (#116):

- **XYZ lin acc.** Vemos los valores de la aceleración lineal leídos por la IMU 2 (acelerómetro) en los 3 ejes espaciales centrados en el dron. Podremos usar este parámetro como variable en una ecuación de movimiento y ver la diferencia de la posición y velocidad con el GPS. Esta unidad la obtenemos en *mG*.

- **XYZ angular acc.** Obtenemos valores de la aceleración angular en *mrad/s* leídos por la IMU 2 (acelerómetro) alrededor de los 3 ejes espaciales.
- **XYZ mag.** Obtenemos valores del campo magnético en leídos por la IMU 2 (magnetómetro) en de los 3 ejes espaciales.

Scaled IMU 3(#129):

- **XYZ lin acc.** Vemos los valores de la aceleración lineal leídos por la IMU 3 (acelerómetro) en los 3 ejes espaciales centrados en el dron. Podremos usar este parámetro como variable en una ecuación de movimiento y ver la diferencia de la posición y velocidad con el GPS. Esta unidad la obtenemos en *mG*.
- **XYZ angular acc.** Obtenemos valores de la aceleración angular en *mrad/s* leídos por la IMU 3 (acelerómetro) alrededor de los 3 ejes espaciales.
- **XYZ mag.** Obtenemos valores del campo magnético en leídos por la IMU 3 (magnetómetro) en de los 3 ejes espaciales.

Distance sensor (# 132):

- **Min distance, Max distance.** Estas medidas son dos valores fijos que nos va a indicar que rango de distancias es capaz de medir nuestro sensor en centímetros. Miraremos de no sobrepasar el rango máximo y establecer un límite de altura a partir del cual podríamos hacer uso de este dispositivo.
- **Current distance.** Es probablemente el parámetro en el cual depositaremos más interés a fin de saber la distancia al suelo en centímetros. Habrá que tener en cuenta la inclinación del dron y probablemente aplicar algún algoritmo de corrección.
- **Orientation.** Nos dice la orientación, no del UAV sino del propio sensor, a fin de saber los ángulos *pitch*, *roll* y *yaw* y poder aplicar las correcciones convenientes si es necesario. Las unidades están expresadas en grados.

Scaled pressure 2 (#137):

- **Pres abs.** Obtenemos los valores de presión estática en *hPa* del sensor de presión 2. Este parámetro nos permitirá calcular los cambios de altura a lo largo del vuelo si los metemos en la ecuación del modelo troposférico.
- **Temperature.** Obtenemos la temperatura del *heating resistor*, que calienta ligeramente las IMUs para que realicen un trabajo óptimo a una

temperatura adecuada. Este parámetro podría permitirnos calcular los cambios de altura a través de la ecuación hipsométrica.

Local Position NED Cov (#64):

- **XYZ position.** Vemos los valores de la aceleración lineal leídos por el conjunto acelerómetros-GPS en los 3 ejes espaciales centrados en el dron (NED). Podremos usar este parámetro como variable en una ecuación de movimiento y ver la diferencia de la posición y velocidad con el GPS. Esta unidad la obtenemos en *mG*.
- **Vx Vy Vz** Obtenemos valores de la velocidad respecto al suelo proporcionado por el conjunto acelerómetros-GPS
- **XYZ linear acc.** Obtenemos valores de la aceleración en *m/s* leídos por el conjunto acelerómetro-GPS. Valores ya procesados.

ANEXO D. Definición de las performances que puede realizar un multicoptero

El objetivo del siguiente apartado consiste en determinar cómo deben ser interpretados los datos, de forma generica, si queremos determinar las maniobras aéreas que se han llevado a cabo en el vuelo. Saber qué acciones aéreas se han empleado puede ayudarnos a corregir ciertas tendencias del dron que favorezcan el alto consumo de las baterías o que no optimizan el objetivo del vuelo.

Ya que las explicaciones que vamos a dar son válidas para prácticamente cualquier tipo de multicoptero, vamos a simplificar los ejemplos con un modelo *quadcopter* (*Parrot – AR.Drone 2.0*).

Antes de empezar a definir maniobras es importante remarcar que, para la correcta estabilidad del dron, los rotores opuestos en diagonal giren de igual manera mientras que los dos restantes en sentido contrario. A priori, la mayoría de los drones realizan maniobras empleando el mismo principio de coordinación y balanceo entre sus rotores.



Fig. D.1: Sentido de rotación de los rotores de un dron

En función del movimiento que deba hacerse ciertos rotores girarán con más o menos intensidad.

➔ *Vertical Take Off/ Landing (Movimiento ascendente/descendente)*

Es la primera/ultima maniobra que cualquier aeronave debe realizar para emprender/finalizar un vuelo.

A medida que la velocidad de giro aumenta, la fuerza debido al peso de la aeronave es superada por la fuerza resultante de sustentación generada por las aspas. Por otro lado, en el aterrizaje, la velocidad de giro debe disminuir paulatinamente hasta tocar suelo.



Fig. D.2 Despegue debido a la acción de aceleración ascendente

Uno puede basar la suposición de que un dron está ascendiendo o descendiendo debido a las siguientes lecturas dentro de un período de tiempo T:

- 1- Detección de variación descendente/ascendente en la presión por el sensor de presión
- 2- Detección de variación ascendente en la corriente empleada para dar potencia a los motores
- 3- Detección de fuerza ascendente/descendente que marca el acelerómetro
- 4- Detección de variación de posición vertical GNSS significativa
- 5- Detección de variación angular del giróscopo con tendencia nula
- 6- Detección de variación de posición horizontal GNSS con tendencia nula

Observaciones:

Mientras que las condiciones 1, 2, 3 y 4 garantizan la detección del cambio de altitud del dron, la 5 y 6 nos informan sobre la verticalidad del movimiento.

→ *Hover (Movimiento estático en el aire)*

Un *Hover* suele emplearse para mantener el dron en espera. Mantener su posición implica que no haya desplazamientos tanto verticales como horizontales. En cambio, sí que podría moverse según un movimiento en *Yaw (rotación sobre su eje vertical)*.

En dicha maniobra los rotores giran por igual y de manera uniforme manteniendo la posición del vehículo incluso si aparecen ráfagas de viento u otros fenómenos no deseados. En principio, en condiciones normales, el único efecto a contrarrestar es el del propio peso de la aeronave y el de su *Payload*.



Fig. D.3 Mantener la posición del dron (Hover)

Uno puede basar la suposición de que un dron se está manteniendo en *Hover* debido a las siguientes lecturas dentro de un período de tiempo T:

- 1- Detección de poca variación en la presión
- 2- Detección de poca variación en la corriente empleada para dar potencia a los motores (aunque podrían aparecer fuerzas a compensar)
- 3- Detección de poca fuerza que marca el acelerómetro
- 4- Detección de poca variación angular del giróscopo
- 5- Detección de poca variación de posición GNSS

➔ *Movimiento en Roll (Desplazamiento lateral)*

Básicamente es un movimiento rectilíneo del mismo modo que podríamos ir hacia adelante o hacia atrás. La diferencia está en que la antena GNSS tiene un *pointer* el cual indica a nuestro sistema de referencia cual es el *heading* natural del dron, pues un roll siempre se efectúa en una dirección transversal contraria a la dirección del *heading*.

Para efectuar un movimiento lateral debemos aplicar más potencia a los motores del lado opuesto al que queremos desplazarnos. En la figura se muestra gráficamente:



Fig. D.4 Disposición de fuerzas para ejecutar un movimiento lateral (hacia nuestra izquierda)

En el caso de la figura anterior debemos tener en cuenta que el dron giraría hacia su derecha, que es nuestra izquierda si consideramos que su *heading* apunta directamente hacia nosotros.

Uno puede basar la suposición de que un dron está realizando un movimiento lateral debido a las siguientes lecturas dentro de un período de tiempo T:

- 1- Detección de poca variación en la presión
- 2- Detección de corriente en dos motores laterales es igual entre sí, y mayor/menor que sus opuestos
- 3- Detección de la fuerza significativa que marca el acelerómetro en un instante y en la dirección del desplazamiento
- 4- Detección de variación angular significativa del giróscopo y en dirección del desplazamiento
- 5- Detección de variación de posición GNSS en la dirección del desplazamiento

Observaciones:

Para garantizar que estamos tratando con un movimiento lateral deben cumplirse, como mínimo, las condiciones 1 y 5 simultáneamente.

→ *Pitch (Desplazamiento frontal y trasero)*

Este movimiento está asociado de igual forma que el desplazamiento lateral, solo que esta vez el movimiento sí que es en la dirección del *heading*. Es por eso, que esta vez si el movimiento es hacia adelante los motores traseros impulsaran con mayor fuerza para realizar el movimiento frontal, de lo contrario se realizaría un desplazamiento trasero.

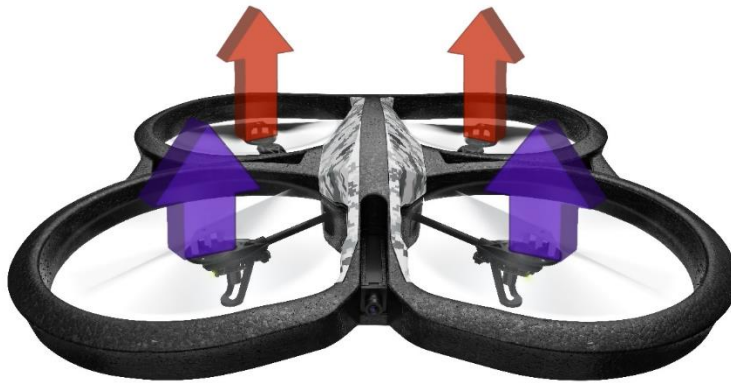


Fig. D.6 Disposición de fuerzas para ejecutar un movimiento hacia adelante (hacia nosotros)

Tal y como podemos ver los motores traseros actúan con la misma intensidad, mayor que la fuerza de los dos motores delanteros que también es la misma. Así el dron puede ejecutar un cabeceo (pitch) hacia adelante y hacer el debido desplazamiento frontal.

Uno puede basar la suposición de que un dron está realizando un movimiento lateral debido a las siguientes lecturas dentro de un período de tiempo T:

- 1- Detección de poca variación en la presión
- 2- Detección de corriente en dos motores frontales/traseros es igual entre sí, y mayor/menor que sus opuestos.
- 3- Detección de la fuerza significativa que marca el acelerómetro y en la dirección del desplazamiento.
- 4- Detección de variación angular significativa del giróscopo y en dirección del desplazamiento.
- 5- Detección de variación de posición GNSS en la dirección del desplazamiento.

→ *Yaw (Rotación sobre el eje perpendicular al dron)*

Un *yaw* puede usarse simplemente para hacer que el dron apunte hacia otra orientación (y así cambiamos su *heading* manteniendo un *hover*), o bien puede ser un movimiento que en combinación con un *roll* conseguimos un *Bank*, más bien utilizado en aviación de ala fija.

Para hacer un movimiento *yaw* en dirección horaria, el dron aumentará equitativamente la potencia en los dos motores opuestos en diagonal y disminuyendo la potencia de los otros dos restantes de igual forma. En principio, la fuerza neta resultante de los 4 motores será la misma que en un *Hover* (igualando el peso) haciendo que el dron ejecute el *Yaw* sin aumentar o disminuir su altitud.

Si la fuerza de los motores que giran horariamente es la que aumenta, el dron hará un giro de forma antihoraria. De lo contrario haría un giro horario. Esto se debe al momento angular que generan los rotores por el simple hecho de girar y estar dispuestos de forma simétrica alrededor del centro de masas.

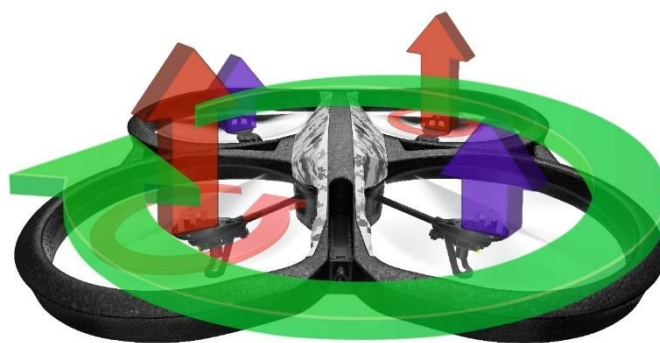


Fig. D.7 Giro horario debido al aumento de potencia a los rotores que giran antihorariamente

Uno puede basar la suposición de que un dron está realizando un movimiento lateral debido a las siguientes lecturas dentro de un período de tiempo T:

- 1- Detección de poca variación en la presión

- 2- Detección de voltajes en dos motores opuestos en diagonal igual entre si, y mayor/menor que sus opuestos.
- 3- Detección de poca fuerza vertical nula en el acelerómetro.
- 4- Detección de variación angular del giróscopo en azimut no nula y nula en ángulo cenital.
- 5- Detección de variación de dirección del *heading* GNSS y desplazamiento nulo.

ANEXO E. Representación de la opción 2 para la ruta de la figura 4.11

Tabla E.1 Opción 2 para la segmentación de maniobras de rotación

Rango en Z	Sub – Rango <i>heading</i>	From – To (Waypoints)	Variación en <i>heading</i> / Variación nula
1	1	1 – (2-N)	Variación nula
	2	(2-N) – (3+N)	Variación en <i>heading</i>
	3	(3+N) – (4-N)	Variación nula
2	4	(4-N) – (5+N)	Variación en <i>heading</i>
3	5	(5+N) – (6+N)	Variación en <i>heading</i>
4	6	(6+N) – (7-N)	Variación en <i>heading</i>
5	7	(7+N) – (8-N)	Variación nula
	8	(8-N) – (9+N)	Variación en <i>heading</i>
	9	(9+N) – (10+N)	Variación nula

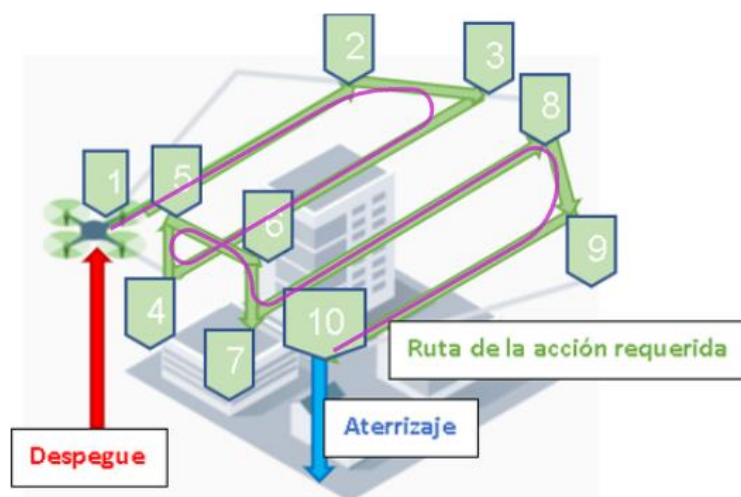


Fig E.1 Opción 2 para configuración de maniobras de rotación de la figura 4.11. La línea púrpura indica una trayectoria más curvilínea, sin que el *drone* tenga que pasar por encima de los WPs.

ANEXO F. Instrucciones de las maniobras insertadas en MP para los vuelos 1 y 2

Radio WP 1 Radio Perder Tiempo por Defecto 20 Relative Verify Height Agregar Abajo Alt Warn 0 Spline															
	Comandos	Delay			Lat	Long	Alt	Frame	Borrar	Arrib	Abaj	Grad %	Angle	Dist	AZ
1	TAKEOFF	0	0	0	41,2885379	1,3320048	20	Relative	X	🏠	🏠	0	0	0	0
2	WAYPOINT	2	0	0	41,2885391	1,3321245	20	Relative	X	🏠	🏠	13,5	7,7	149,9	35
3	CONDITION_YAW	0	0	1	0	0	0	Relative	X	🏠	🏠	0	0	0	0
4	DO_CHANGE_SPEED	1	10	0	0	0	0	Relative	X	🏠	🏠	0	0	0	0
5	WAYPOINT	2	0	0	41,2894398	1,3321013	20	Relative	X	🏠	🏠	0,0	0,0	100,2	359
6	WAYPOINT	2	0	0	41,2885391	1,3321245	20	Relative	X	🏠	🏠	0,0	0,0	100,2	179
7	CONDITION_YAW	0	0	-1	0	0	0	Relative	X	🏠	🏠	0	0	0	0
8	WAYPOINT	2	0	0	41,2885567	1,3333179	20	Relative	X	🏠	🏠	0,0	0,0	99,7	89
9	WAYPOINT	2	0	0	41,2885391	1,3321245	20	Relative	X	🏠	🏠	0,0	0,0	99,7	269
10	WAYPOINT	7	0	0	41,2885391	1,3321245	40	Relative	X	🏠	🏠	..	90,0	20,0	180
11	WAYPOINT	2	0	0	41,2885391	1,3321245	20	Relative	X	🏠	🏠	..	-90,0	20,0	180
12	WAYPOINT	2	0	0	41,2874406	1,3311106	20	Relative	X	🏠	🏠	0,0	0,0	148,6	215
13	WAYPOINT	0	0	0	41,2874406	1,3311106	5	Relative	X	🏠	🏠	..	-90,0	15,0	180
14	LAND	0	0	0	0	0	0	Relative	X	🏠	🏠	0	0	0	0

Fig. F.1 Tabla de comandos desde MP para el Vuelo 1

Radio WP 1														Radio Perder Tiempo 45		por Defecto 20		Relative		Verify Height		Agrega Abajo		Alt Warn 0		Spline	
	Comandos	Delay				Lat	Long	Alt	Frame	Borrar	Amb	Abaj	Grad %	Angle	Dist	AZ											
1	TAKEOFF	0	0	0	0	41,287441...	1,33110893	5	Relative	X			0	0	0	0											
2	DO_CHANGE_SPEED	1	2	0	0	41,287441...	1,33110893	5	Relative	X			0	0	0	0											
3	WAYPOINT	1	0	0	0	41,287441...	1,33110893	10	Relative	X			934...	89,4	10,0	309											
4	LOITER_TIME	3	0	0	0	41,287441...	1,33110893	10	Relative	X			NaN	NaN	0,0	180											
5	WAYPOINT	1	0	0	0	41,287441...	1,33110893	20	Relative	X			**	90,0	10,0	180											
6	DO_CHANGE_SPEED	1	3	0	0	41,287441...	1,33110893	20	Relative	X			0	0	0	0											
7	WAYPOINT	0	0	0	0	41,287227...	1,33127367	20	Relative	X			0,0	0,0	27,5	150											
8	DO_CHANGE_SPEED	1	10	0	0	41,287227...	1,33127367	80	Relative	X			0	0	0	0											
9	WAYPOINT	0	0	0	0	41,287403...	1,33258009	80	Relative	X			54,1	28,4	126,1	80											
10	WAYPOINT	0	0	0	0	41,286968...	1,33291173	80	Relative	X			0,0	0,0	55,7	150											
11	WAYPOINT	0	0	0	0	41,286537...	1,33223367	80	Relative	X			0,0	0,0	74,2	230											
12	WAYPOINT	0	0	0	0	41,285785...	1,33376884	80	Relative	X			0,0	0,0	153,1	123											
13	WAYPOINT	0	0	0	0	41,285354...	1,33309186	80	Relative	X			0,0	0,0	74,1	230											
14	WAYPOINT	0	0	0	0	41,285358...	1,33397627	80	Relative	X			0,0	0,0	73,9	90											
15	WAYPOINT	0	0	0	0	41,285129...	1,33314562	80	Relative	X			0,0	0,0	73,9	250											
16	WAYPOINT	0	0	0	0	41,285640...	1,33283031	80	Relative	X			0,0	0,0	62,6	335											
17	WAYPOINT	0	0	0	0	41,285858...	1,33272517	80	Relative	X			0,0	0,0	25,7	340											
18	WAYPOINT	0	0	0	0	41,286193...	1,33349109	80	Relative	X			0,0	0,0	74,1	60											
19	WAYPOINT	0	0	0	0	41,286193...	1,33260667	80	Relative	X			0,0	0,0	73,9	270											
20	WAYPOINT	0	0	0	0	41,286705...	1,33317482	80	Relative	X			0,0	0,0	74,1	40											
21	WAYPOINT	0	0	0	0	41,286586...	1,33230388	80	Relative	X			0,0	0,0	73,9	260											
22	WAYPOINT	0	0	0	0	41,287017...	1,33298087	80	Relative	X			0,0	0,0	74,1	50											
23	WAYPOINT	0	0	0	0	41,287017...	1,33209658	80	Relative	X			0,0	0,0	73,9	270											
24	DO_CHANGE_SPEED	1	4	0	0	41,287017...	1,33209658	20	Relative	X			0	0	0	0											
25	WAYPOINT	0	0	0	0	41,287204...	1,33141303	20	Relative	X			-98,7	-44,6	85,4	290											
26	WAYPOINT	3	0	0	0	41,2873497	1,33123064	20	Relative	X			0,0	0,0	22,2	317											
27	WAYPOINT	2	0	0	0	41,2873497	1,33123064	15	Relative	X			--	-90,0	5,0	180											
28	RETURN_TO_LAUNCH	5	0	0	0	41,287445...	1,33110917	15	Relative	X			0	0	0	0											
29	LOITER_TIME	4	0	0	0	41,287445...	1,33110917	15	Relative	X			0,0	0,0	14,7	316											
30	LAND	5	0	0	0	41,287445...	1,33110917	0	Relative	X			--	-90,0	15,0	180											

Fig. F.2 Tabla de comandos desde MP para el Vuelo 2

ANEXO G. Gráfico de posición del vuelo 1 en coordenadas geodesicas

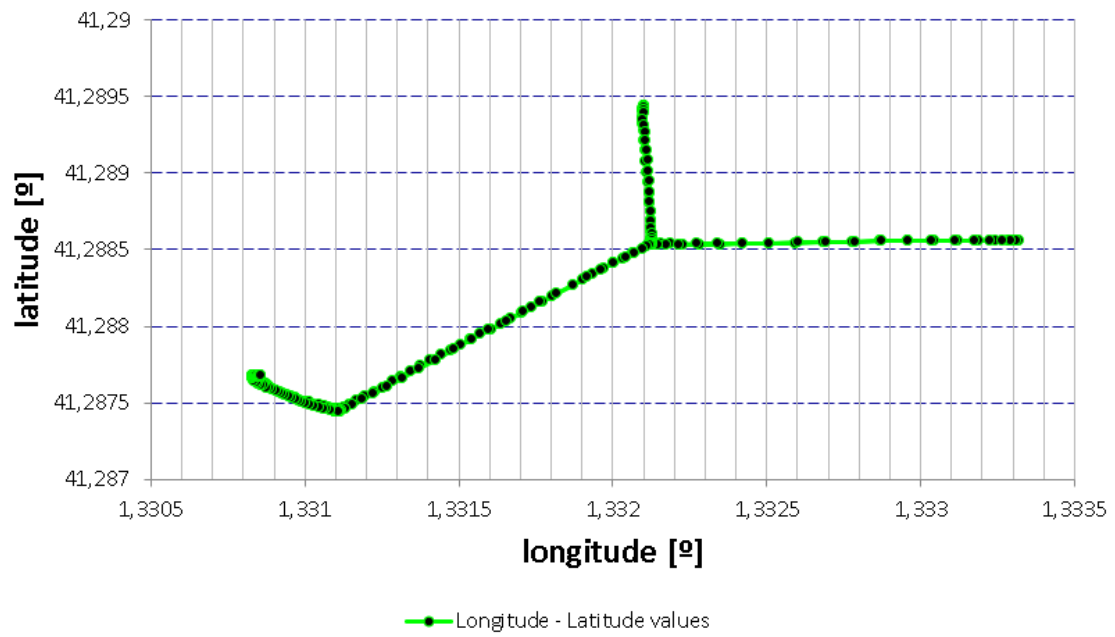


Fig. G.1 Representación del vuelo 1 en el plano XY en coordenadas lat-lon

ANEXO H. Contenido de la pestaña *Accel and attitude*: Graficos para las lecturas de la IMU de los vuelos 1 y 2

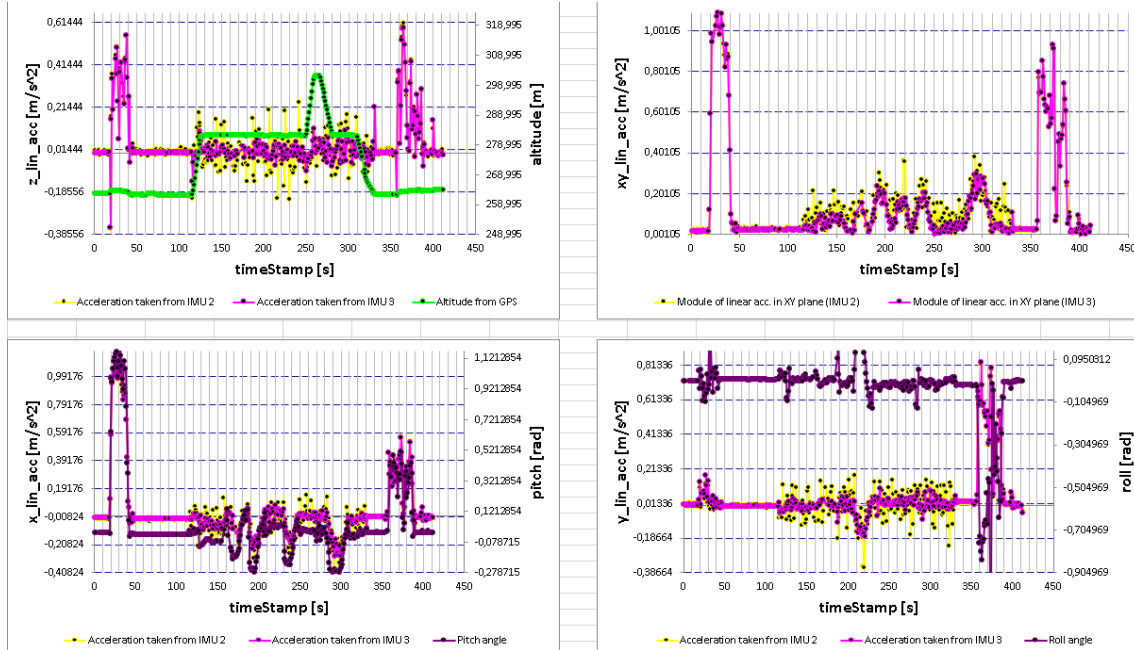


Fig. H.1 Representación del vuelo 1 con las medidas de aceleración e angulares a lo largo del tiempo (y de la altura).

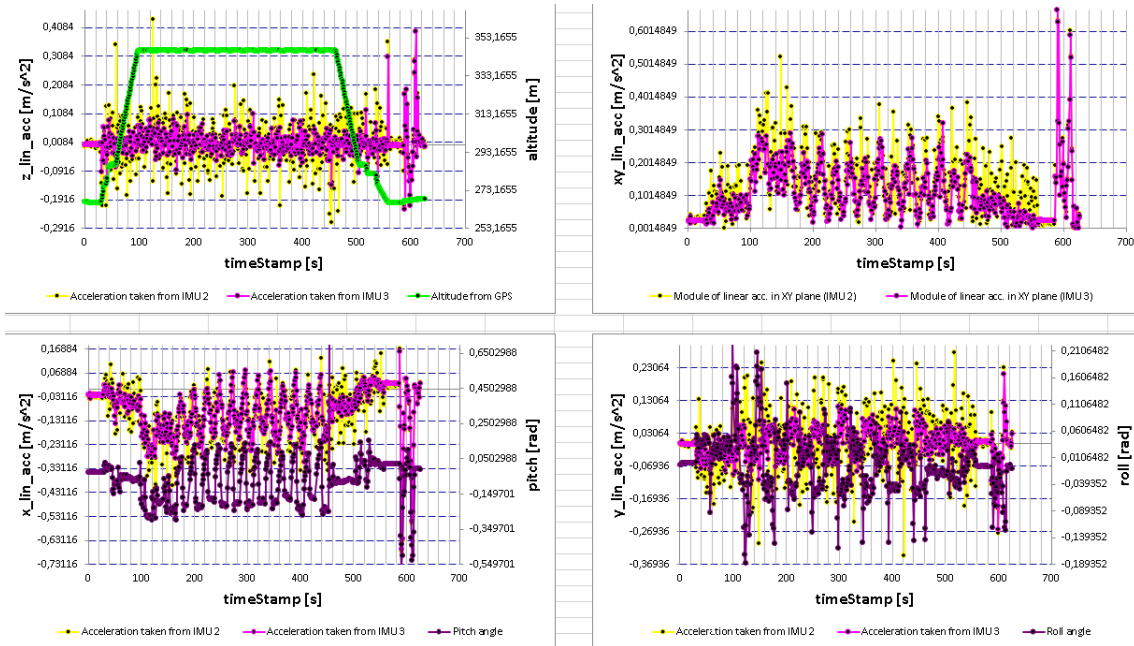


Fig. H.2 Representación del vuelo 2 con las medidas de aceleración e angulares a lo largo del tiempo (y de la altura).

ANEXO I. Contenido de la pestaña *Consumption analysis*: Graficos para las lecturas del voltaje de los vuelos 1 y 2

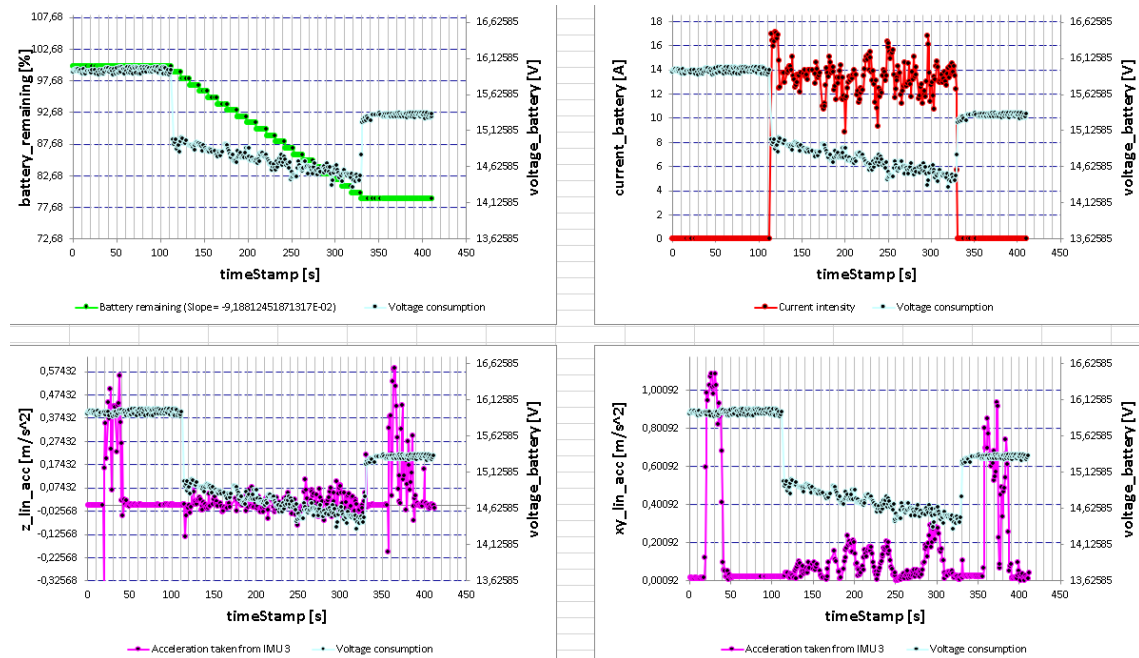


Fig. I.1 Representación del vuelo 1 con las medidas de la diferencia de voltaje en función del tiempo, corriente, aceleraciones y la batería restante (y de la corriente y aceleraciones).

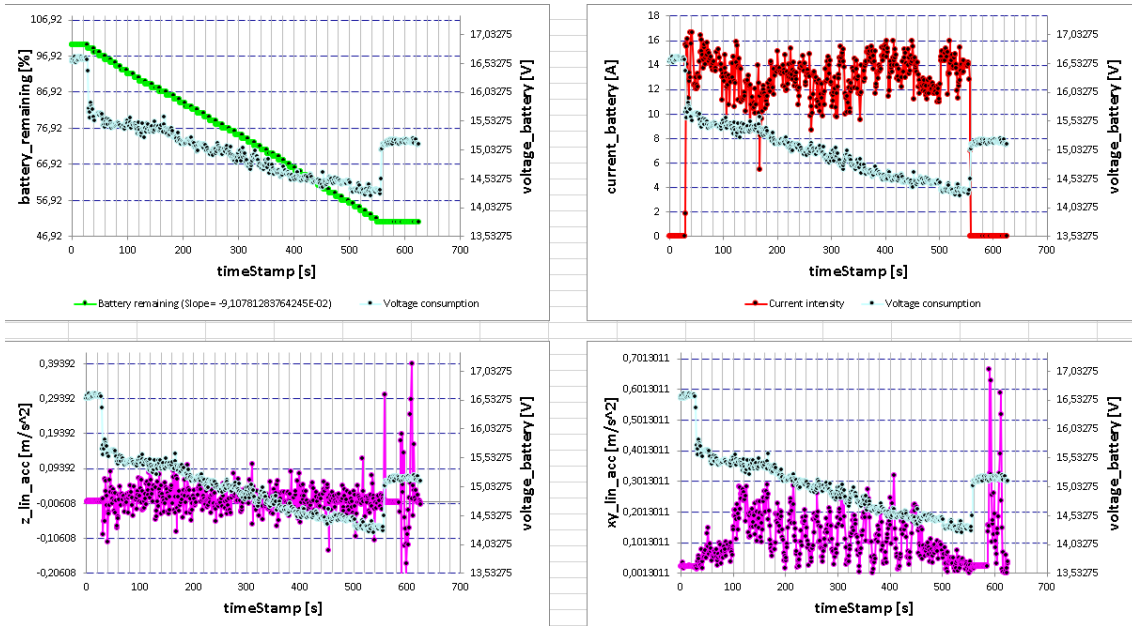


Fig. I.2 Representación del vuelo 2 con las medidas de la diferencia de voltaje en función del tiempo, corriente, aceleraciones y la batería restante (y de la corriente y aceleraciones).

ANEXO J. Contenido de la pestaña *Troposphere behave*: Graficos para las lecturas de la presión y temperatura de los vuelos 1 y 2

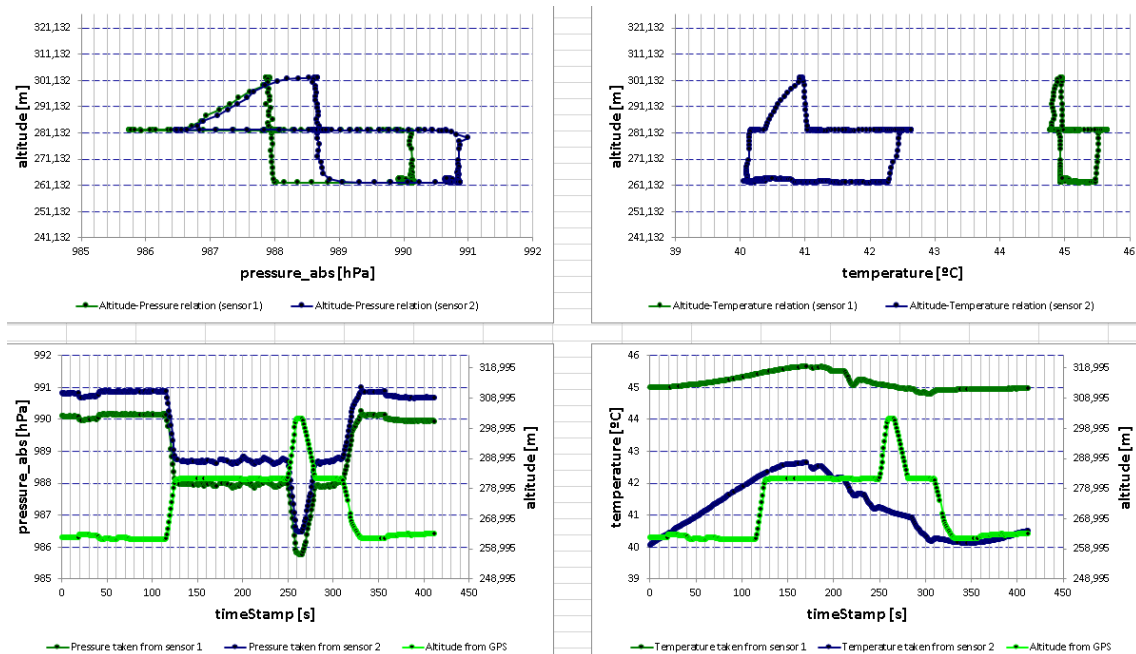


Fig. J.1 Representación del vuelo 1 con las lecturas de temperatura y presión en función del tiempo.

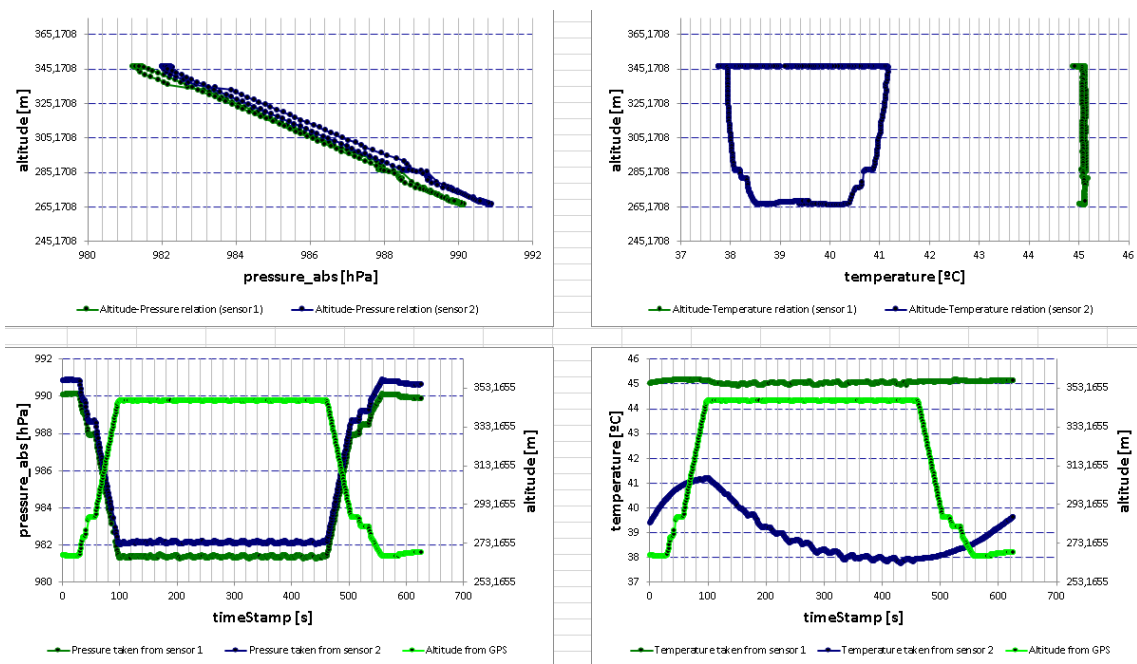


Fig. J.2 Representación del vuelo 2 con las lecturas de temperatura y presión en función del tiempo.

ANEXO K. Contenido de la pestaña *Tracking*: Graficos para las lecturas del movimiento en XYZ de los vuelos 1 y 2

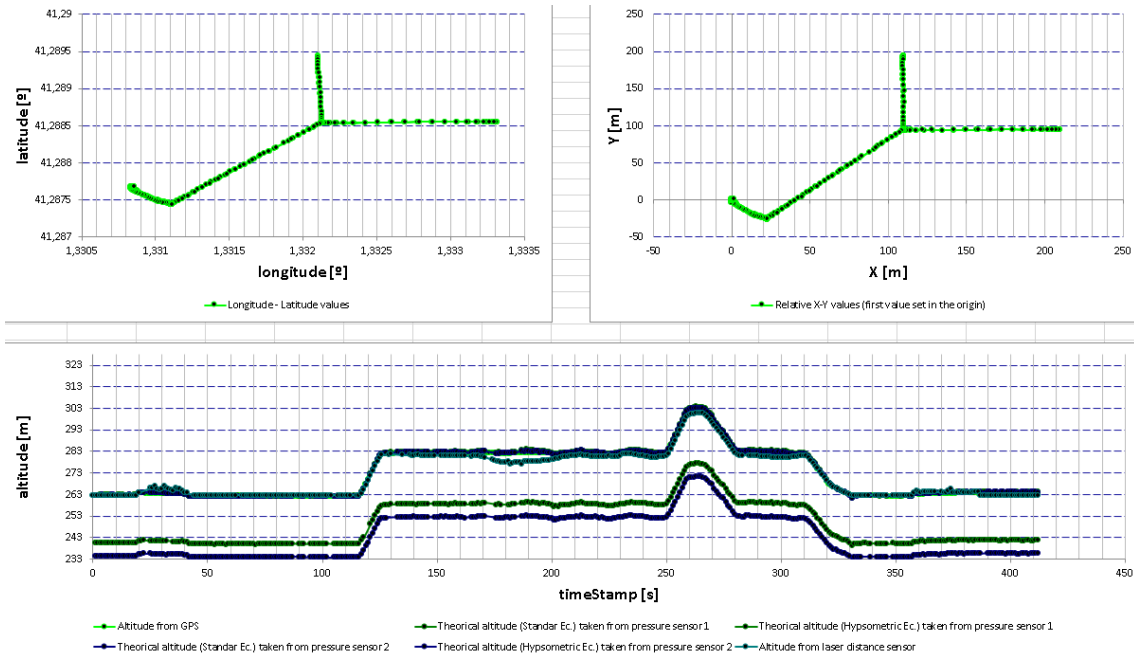


Fig. K.1 Representación del vuelo 1 con las lecturas del *tracking* función del tiempo.

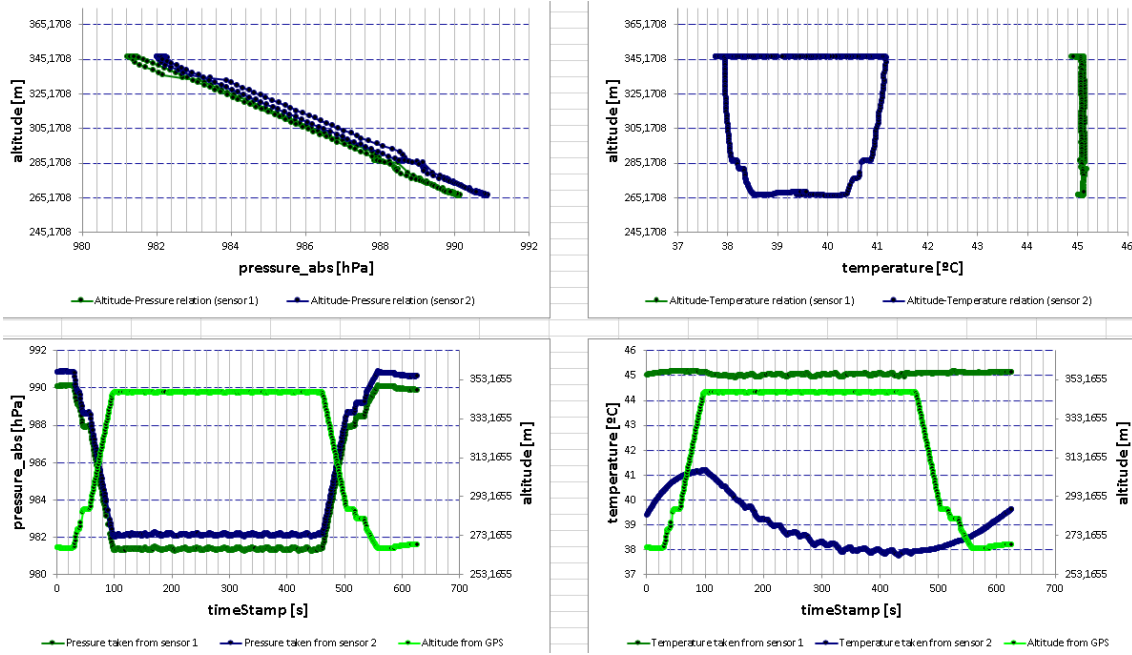
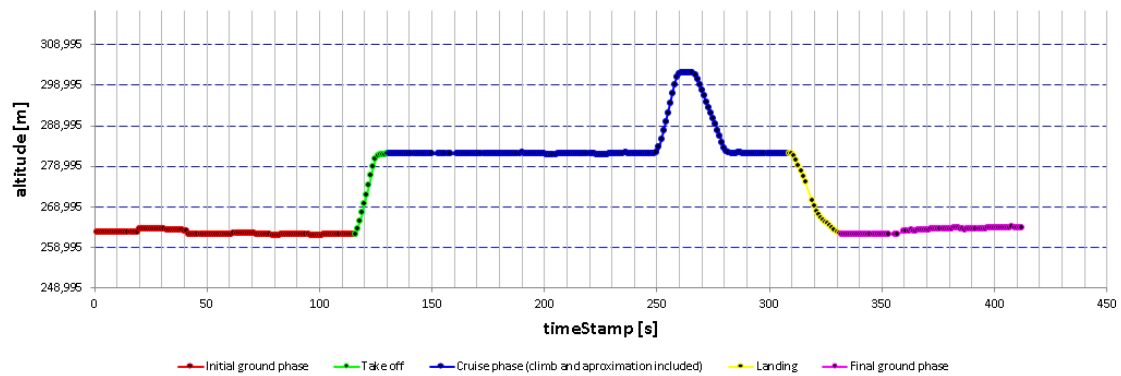


Fig. K.2 Representación del vuelo 2 con las lecturas del *tracking* en función del tiempo.

ANEXO L. Contenido de la pestaña *Sectorization and motion*: Gráficos y tablas para las resoluciones de las fases de vuelo y los movimientos de vuelo empleados, para los vuelos 1 y 2

Time (duration) [s]	Performance along Z axis	Performance in XY plane	Change of distance in altitude [m]	Change of distance in XY plane [m]	Angular change in XY plane [°]	Mean speed in Z axis [m/s]	Mean speed module in XY plane [m/s]
115,981 - 150,981 (15)	Ascent motion (Take Off)	Forward motion	262,95 - 282,09 (19,74)	3,1	153,99 → 32,14 (-101,85)	1,32	0,21
151,986 - 157,98 (25,994)	Steady motion (282,12)	Forward motion	-	109,89	32,99 → 34,44 (1,51)	-	4,23
157,98 - 162,981 (5,001)	Steady motion (282,14)	Forward motion	-	7,88	34,44 → 356,81 (322,37)	-	1,58
162,981 - 164,98 (1,999)	Steady motion (282,1)	Turn (Clockwise)	-	1,37	356,81 → 359,86 (3,05)	-	0,69
164,98 - 166,981 (2,001)	Steady motion (282,09)	Turn (Counter-clockwise)	-	6,69	359,86 → 359,21 (-0,65)	-	3,34
166,981 - 170,981 (4)	Steady motion (282,12)	Forward motion	-	20,96	359,21 → 0 (-359,21)	-	5,24
170,981 - 172,987 (2,006)	Steady motion (282,22)	Turn (Clockwise)	-	16,84	0 → 0,01 (0,01)	-	8,39
172,987 - 177,983 (4,996)	Steady motion (282,23)	Forward motion	-	35,32	0,01 → 359,8 (359,79)	-	7,07
177,983 - 183,982 (5,999)	Steady motion (282,08)	Forward motion	-	14,73	359,8 → 359,48 (-0,32)	-	2,46
183,982 - 186,981 (2,999)	Steady motion (282,11)	Forward motion	-	3,92	359,48 → 43,7 (-315,78)	-	1,31
186,981 - 190,981 (4)	Steady motion (282,18)	Turn (Clockwise)	-	19,16	43,7 → 177,86 (134,16)	-	4,79
190,981 - 205,983 (15,002)	Steady motion (282,28)	Forward motion	-	60,44	177,86 → 188,85 (10,99)	-	4,03
205,983 - 209,984 (4,001)	Steady motion (282,03)	Turn (Clockwise)	-	5,62	188,85 → 359,99 (171,14)	-	1,4
209,984 - 210,981 (0,997)	Steady motion (282,19)	Turn (Counter-clockwise)	-	6,01	359,99 → 359,98 (-0,01)	-	6,03
210,981 - 220,982 (10,001)	Steady motion (282,2)	Forward motion	-	64,13	359,98 → 359,32 (-0,66)	-	6,41
220,982 - 228,989 (8,007)	Steady motion (282,06)	Forward motion	-	20,9	359,32 → 1,12 (-358,2)	-	2,61
228,989 - 232,981 (3,992)	Steady motion (282,03)	Turn (Clockwise)	-	15,07	1,12 → 268,63 (267,51)	-	3,78
232,981 - 249,984 (17,003)	Steady motion (282,17)	Forward motion	-	73,18	268,63 → 273,74 (5,11)	-	4,3
249,984 - 258,982 (8,998)	Straight ascent motion	Steady motion	282,41 - 300,9 (18,49)	-	273,74 → 272,78 (-0,96)	2,05	-
258,982 - 266,984 (8,002)	Steady motion (300,9)	Steady motion	-	-	272,78 → 273,18 (0,4)	-	-
266,984 - 269,981 (2,997)	Straight ascent motion	Steady motion	301,48 - 297,66 (3,82)	-	273,18 → 261,42 (-11,76)	1,27	-
269,981 - 273,989 (4,008)	Straight ascent motion	Yaw motion	297,66 - 292,02 (5,64)	-	261,42 → 271,77 (10,35)	1,41	-
273,989 - 274,98 (0,991)	Straight ascent motion	Yaw motion	292,02 - 290,62 (1,4)	-	271,77 → 275,8 (4,03)	1,41	-
274,98 - 279,981 (4,001)	Straight ascent motion	Steady motion	290,62 - 284,91 (5,71)	-	275,8 → 266,98 (-8,82)	1,43	-
279,981 - 279,98 (0,999)	Straight ascent motion	Steady motion	284,91 - 283,55 (1,36)	-	266,98 → 274,88 (7,9)	1,36	-
279,98 - 283,987 (4,007)	Steady motion (283,55)	Forward motion	-	1,39	274,88 → 271,99 (-2,89)	-	0,35
283,987 - 287,985 (3,998)	Steady motion (282,21)	Turn (Counter-clockwise)	-	16,22	271,99 → 214,68 (-57,31)	-	4,06
287,985 - 308,981 (20,996)	Steady motion (282,27)	Forward motion	-	98,55	214,68 → 210,99 (-3,69)	-	4,69
308,981 - 331,981 (23)	Descent motion (Landing)	Forward motion	282,07 - 262,37 (19,7)	1,52	210,99 → 209,07 (-1,92)	0,86	0,07

Fig. L.1 Representación del conjunto de maniobras empleados para el vuelo 1



Total Flight time [s]	Landing Path Error [m]	Altitude. max. [m]	Ground Altitude [m]	Max speed in ascence [m/s]	Max speed in descence [m/s]	Max module speed in XY plane [m/s]
210	0,83	302,15	262,1	2,41	1,53	8,71

Fig. L.1 Representación de las fases del vuelo 1 juntamente con los valores de los datos adicionales

Time [duration] [s]	Performance along	Performance in XY	Change of distance in alt	Change of distance in XY	Angular change in XY p	Mean speed in Z axis	Mean speed module in XY plane
23.936 - 50.989 (27.053)	Ascent motion (Take off)	Forward motion	266,95 - 285,52 (18,57)	11,03	130,51 → 153,23 (22,72)	0,94	0,93
50.989 - 55.987 (4,998)	Steady motion (285,47)	Forward motion	-	13,78	153,39 → 151,08 (-2,31)	-	3,45
55.987 - 57.988 (2,001)	Steady motion (285,42)	Turn (Counter-clockwise)	-	6,31	151,08 → 89,07 (-62,01)	-	-
57.988 - 59.991 (2,003)	Ascent motion	Forward motion	287,1 - 344,74 (57,64)	16,22	89,07 → 80,76 (-8,31)	1,48	4,13
59.991 - 57.988 (0,997)	Ascent motion	Turn (Clockwise)	344,74 - 345,84 (1,1)	7,85	80,76 → 83,8 (3,04)	1,1	7,87
57.988 - 101.993 (44,005)	Steady motion (345,04)	Turn (Clockwise)	-	16,3	83,8 → 143,14 (59,34)	-	3,62
101.993 - 105.987 (3,994)	Steady motion (346,53)	Forward motion	-	17	143,04 → 153,68 (10,62)	-	4,26
105.987 - 109.988 (4,001)	Steady motion (346,5)	Turn (Clockwise)	-	17,5	153,68 → 215,1 (61,42)	-	13,4
109.988 - 110.987 (0,999)	Steady motion (346,47)	Turn (Clockwise)	-	13,89	215,73 → 223,42 (7,69)	-	13,9
110.987 - 120.987 (10)	Steady motion (345,43)	Forward motion	-	33,31	223,42 → 237,63 (14,21)	-	3,33
120.987 - 124.988 (4,001)	Steady motion (346,51)	Turn (Counter-clockwise)	-	12,32	227,09 → 172,58 (-54,53)	-	3,08
124.988 - 127.987 (2,999)	Steady motion (346,4)	Forward motion	-	25,41	172,58 → 156,36 (-16,2)	-	3,47
127.987 - 131.987 (4)	Steady motion (346,51)	Turn (Counter-clockwise)	-	27,98	156,36 → 89,58 (-67,38)	-	7
131.987 - 144,001 (12,014)	Steady motion (346,39)	Forward motion	-	73,26	93,58 → 84,33 (-9,25)	-	6,1
144,001 - 147,988 (3,987)	Steady motion (346,48)	Turn (Clockwise)	-	34,66	84,39 → 134,14 (49,75)	-	8,69
147,988 - 151,991 (4,002)	Steady motion (346,51)	Forward motion	-	32,61	134,14 → 146,6 (12,46)	-	8,15
151,991 - 155,987 (3,997)	Steady motion (346,5)	Turn (Clockwise)	-	8,78	146,6 → 221,84 (75,24)	-	2,2
155,987 - 175,987 (20)	Steady motion (346,44)	Forward motion	-	2,61	221,84 → 230,2 (8,36)	-	0,13
175,987 - 179,989 (4,002)	Steady motion (346,39)	Turn (Counter-clockwise)	-	8,08	230,2 → 89,1 (-141,1)	-	2,02
179,989 - 183,997 (20,008)	Steady motion (346,5)	Forward motion	-	81,81	89,1 → 88,4 (-0,7)	-	4,09
183,997 - 203,988 (19,991)	Steady motion (346,35)	Turn (Clockwise)	-	6,23	88,4 → 262,3 (163,9)	-	1,56
203,988 - 223,987 (19,999)	Steady motion (346,52)	Forward motion	-	40,44	262,3 → 253,29 (0,99)	-	2,02
223,987 - 227,986 (3,999)	Steady motion (346,36)	Turn (Counter-clockwise)	-	2,73	253,29 → 41,56 (-211,73)	-	0,68
227,986 - 247,988 (20,002)	Steady motion (346,52)	Forward motion	-	8,86	41,56 → 44,23 (2,67)	-	0,44
247,988 - 251,986 (3,998)	Steady motion (346,43)	Turn (Clockwise)	-	13,58	44,23 → 270,24 (226,01)	-	3,4
251,986 - 262,989 (11,003)	Steady motion (346,54)	Forward motion	-	66,19	270,24 → 270,77 (0,53)	-	6,02
262,989 - 271,986 (8,997)	Steady motion (346,5)	Forward motion	-	20,4	270,77 → 274,41 (3,64)	-	2,27
271,986 - 275,99 (4,004)	Steady motion (346,44)	Yaw motion	-	-	274,41 → 58,07 (-216,34)	-	-
275,99 - 295,988 (19,998)	Steady motion (346,52)	Forward motion	-	37,95	58,07 → 65,5 (7,43)	-	1,9
295,988 - 299,987 (3,999)	Steady motion (346,4)	Turn (Clockwise)	-	14,22	65,5 → 260,71 (195,21)	-	3,56
299,987 - 319,99 (20,003)	Steady motion (346,56)	Forward motion	-	72,1	260,71 → 264,66 (3,95)	-	3,6
319,99 - 323,987 (3,997)	Steady motion (346,4)	Turn (Counter-clockwise)	-	12,4	264,66 → 99,07 (-165,59)	-	0,21
323,987 - 342,988 (19,001)	Steady motion (346,55)	Forward motion	-	48,93	99,07 → 61,22 (-37,85)	-	2,56
342,988 - 346,987 (3,999)	Steady motion (346,39)	Turn (Clockwise)	-	14,7	61,22 → 261,54 (200,32)	-	3,83
346,987 - 366,991 (20,004)	Steady motion (346,52)	Forward motion	-	99,82	261,54 → 270,32 (8,78)	-	4,99
366,991 - 370,988 (3,997)	Steady motion (346,35)	Yaw motion	-	-	270,32 → 41,04 (-229,28)	-	-
370,988 - 390,988 (20,001)	Steady motion (346,5)	Forward motion	-	38,93	41,04 → 41,06 (0,02)	-	1,95
390,988 - 394,993 (4,005)	Steady motion (346,33)	Turn (Clockwise)	-	16,3	41,06 → 390,18 (249,12)	-	4,07
394,993 - 414,989 (19,996)	Steady motion (346,52)	Forward motion	-	107,5	390,18 → 298,16 (-92,02)	-	5,38
414,989 - 418,987 (3,998)	Steady motion (346,33)	Turn (Counter-clockwise)	-	7,97	298,16 → 53,09 (-245,07)	-	1,99
418,987 - 438,989 (20,002)	Steady motion (346,52)	Forward motion	-	90,61	53,09 → 51,12 (-1,97)	-	4,53
438,989 - 442,988 (3,999)	Steady motion (346,48)	Turn (Clockwise)	-	17,35	51,12 → 270,91 (219,79)	-	4,34
442,988 - 460,986 (17,998)	Steady motion (346,67)	Forward motion	-	105,67	270,91 → 270,33 (-0,58)	-	5,97
460,986 - 462,987 (2,001)	Steady motion (346,44)	Turn (Counter-clockwise)	-	6,76	270,33 → 231,97 (-38,36)	-	-
462,987 - 466,988 (4,001)	Descent motion	Turn (Counter-clockwise)	345,19 - 340,08 (-5,11)	8,13	231,97 → 209,84 (-22,13)	-1,28	2,03
466,988 - 503,988 (37)	Descent motion	Forward motion	340,08 - 287,46 (-52,62)	21,62	209,84 → 203,89 (-5,95)	-1,42	0,88
503,988 - 506,988 (3)	Steady motion (287,46)	Forward motion	-	2,08	203,89 → 264,26 (60,37)	-	0,69
506,988 - 510,987 (3,999)	Steady motion (286,45)	Turn (Clockwise)	-	15,1	264,26 → 316,31 (52,05)	-	3,78
510,987 - 517,987 (7)	Steady motion (286,49)	Forward motion	-	8,26	316,31 → 324,2 (7,89)	-	1,18
517,987 - 521,988 (4,001)	Straight descent motion	Steady motion	285,51 - 281,63 (-3,82)	-	324,2 → 322,81 (-1,39)	-	-
521,988 - 533,989 (12,001)	Steady motion (281,63)	Forward motion	-	10,33	322,81 → 322,5 (-0,31)	-	0,86
533,989 - 558,986 (24,997)	Descent motion (Land)	Forward motion	281,66 - 266,79 (14,87)	0,86	322,5 → 321,42 (-1,08)	0,59	0,03

Tabla L.2 Representación del conjunto de maniobras empleados para el vuelo 2

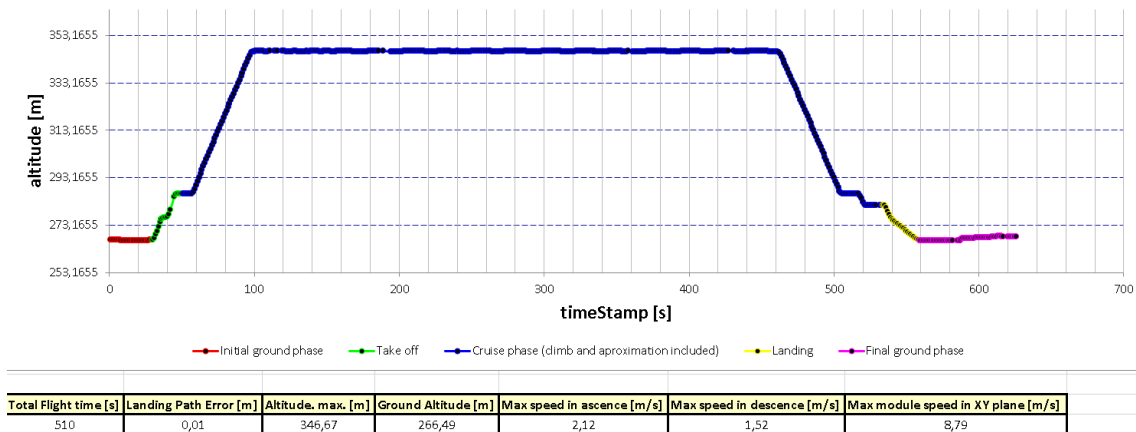


Fig. L.2 Representación de las fases del vuelo 2 juntamente con los valores de los datos adicionales

ANEXO M. Contenido de los *scripts* en *Visual Basic for Applications* para la realización del algoritmo procesador de datos.

ANEXO M.1 Código del primer módulo: Interacción archivo origen – Identificación de cada mensaje recibido – Reestructuración de datos – Escritura del sampling time y márgenes temporales.

'We create as many sheets as messages readen in the original File. The original File is the input document. We will know how many labels we must read since we will

'be comparing each message name (first position of every raw). The first time that 2 names are compared and both names are the same, it will mean that the definition

'of messages has ended and it has started the providing of values of every message.

```
Dim mensajes() As String 'This vector is containing the different kinds of messages that we can read
```

```
ReDim mensajes(1 To 1)
```

```
msg_len = 1 'starting conditions to make the comparisons and create the "mensajes" vector
```

```
fila = 2
```

```
msg_repited = 0
```

```
mensajes(1) = Trim(ThisWorkbook.Worksheets(1).Cells(1, 1)) 'Here appears the name of the 1st message (should be m1Heartbeat)
```

```
Do While msg_repited = 0 'Now we want to know how many messages do we have in the original file, it starts comparing the initial raws first column
```

```
    fila_actual = Trim(ThisWorkbook.Worksheets(1).Cells(fila, 1))
```

```
    For a = 1 To UBound(mensajes)
```

```
        mensajes_actuales = Trim(mensajes(a))
```

```
        If mensajes_actuales = fila_actual Then
```

```
            msg_repited = 1
```

```
            Exit Do
```

```
        End If
```

```
    Next a
```

```
    ReDim Preserve mensajes(1 To fila)
```

```

    mensajes(fila) = Trim(ThisWorkbook.Worksheets(1).Cells(fila, 1))
    fila = fila + 1
    msg_len = msg_len + 1
Loop

'We create as many sheets as messages we have
For a = 1 To msg_len
    Sheets.Add After:=Sheets(Sheets.Count)
    ThisWorkbook.Worksheets(a + 1).Name = mensajes(a)
Next a

'Now we must create a "copy paste" code that takes every row of the original
document and set in the correct sheet according to the message's name

'But first lets create a counter vector to be set at every message's row that
is printed. This way we will now in what new blank row (nº of that row) does
the new
'message go.

Dim row_counter() As Integer
ReDim row_counter(1 To msg_len)

For a = 1 To msg_len
    row_counter(a) = 1 'We create a vector of ones as large as msg_len (thats
the initial counter of every row's message)
Next a

'The original file is currently in a sheet. We move that sheet to the last
place.
Worksheets(1).Move _
    After:=Worksheets(msg_len + 1)

row_original = 1

'Now we can go through every row in the original document and see if that
information goes in one sheet or another according to the name of the message
Do While ThisWorkbook.Worksheets(msg_len + 1).Cells(row_original, 1) <> ""
    x = Trim(ThisWorkbook.Worksheets(msg_len + 1).Cells(row_original, 1))
    ThisWorkbook.Worksheets(msg_len + 1).Cells(row_original, 1) = x

```

```

    For a = 1 To msg_len
        If mensajes(a) = x Then
            ThisWorkbook.Worksheets(msg_len + 1).Cells(row_original,
1).EntireRow.Copy
            ThisWorkbook.Worksheets(a).Cells(row_counter(a),
1).EntireRow.PasteSpecial
            row_counter(a) = row_counter(a) + 1
            Exit For
        End If
    Next a
    row_original = row_original + 1
Loop

```

'Eliminamos los espacios en blanco que puedan contener algunos parámetros de las cabeceras

```

For a = 1 To msg_len
    numh = ThisWorkbook.Worksheets(a).UsedRange.Columns.Count
    For b = 1 To numh
        x = Trim(ThisWorkbook.Worksheets(a).Cells(1, b))
        ThisWorkbook.Worksheets(a).Cells(1, b) = x
    Next b
Next a

```

```

'Creamos un LocalTime (boot_time) en cada mensaje (lo creamos al lado derecho
de timeStamp asumiendo que timeStamp esta en la misma columna en todos mensajes)
numh = ThisWorkbook.Worksheets(1).UsedRange.Columns.Count 'columns length of a
message
For a = 1 To numh 'Primero buscamos la posicion de ese "timeStamp"
    x = ThisWorkbook.Worksheets(1).Cells(1, a)
    If x = "timeStamp" Then
        timestamp_pos = a
        Exit For
    End If
Next a

```

```
For a = 1 To msg_len 'con esto ponemos unidades a timeStamp y creamos la nueva
columna de timeLocal
```

```
    Sheets(a).Select
```

```
    ThisWorkbook.Worksheets(a).Cells(1, timestamp_pos).Select
```

```
    ThisWorkbook.Worksheets(a).Cells(1, timestamp_pos) = "timeStamp [s]"
```

```
    ThisWorkbook.Worksheets(a).Cells(1, timestamp_pos + 1).Select
```

```
    ActiveCell.EntireColumn.Select
```

```
    Selection.Insert Shift:=xlToRight
```

```
    ThisWorkbook.Worksheets(a).Cells(1, timestamp_pos + 1) = "timeLocal [s]"
```

```
Next a
```

```
'Ponemos unidades a las variables que vamos a considerar de interés
```

```
For a = 1 To msg_len
```

```
    numh = ThisWorkbook.Worksheets(a).UsedRange.Columns.Count 'columns length
of a message
```

```
    For b = 1 To numh 'Primero buscamos la posicion de ese "timeStamp"
```

```
        x = ThisWorkbook.Worksheets(a).Cells(1, b)
```

```
        If x = "latitude" Then
```

```
            ThisWorkbook.Worksheets(a).Cells(1, b) = "latitude [°]"
```

```
        ElseIf x = "longitude" Then
```

```
            ThisWorkbook.Worksheets(a).Cells(1, b) = "longitude [°]"
```

```
        ElseIf x = "altitude" Then
```

```
            ThisWorkbook.Worksheets(a).Cells(1, b) = "altitude [m]"
```

```
        ElseIf x = "altitude" Then
```

```
            ThisWorkbook.Worksheets(a).Cells(1, b) = "altitude [m]"
```

```
        ElseIf x = "x_lin_acc" Then
```

```
            ThisWorkbook.Worksheets(a).Cells(1, b) = "x_lin_acc [m/s^2]"
```

```
        ElseIf x = "y_lin_acc" Then
```

```
            ThisWorkbook.Worksheets(a).Cells(1, b) = "y_lin_acc [m/s^2]"
```



```
ElseIf x = "z_lin_acc" Then
    ThisWorkbook.Worksheets(a).Cells(1, b) = "z_lin_acc [m/s^2]"

'
'
    ElseIf x = "x_lin_acc" Then
        ThisWorkbook.Worksheets(a).Cells(1, b) = "x_lin_acc [m/s]"

ElseIf x = "pressure_abs" Then
    ThisWorkbook.Worksheets(a).Cells(1, b) = "pressure_abs [hPa]"

ElseIf x = "temperature" Then
    ThisWorkbook.Worksheets(a).Cells(1, b) = "temperature [°C]"

ElseIf x = "roll" Then
    ThisWorkbook.Worksheets(a).Cells(1, b) = "roll [rad]"

ElseIf x = "pitch" Then
    ThisWorkbook.Worksheets(a).Cells(1, b) = "pitch [rad]"

ElseIf x = "yaw" Then
    ThisWorkbook.Worksheets(a).Cells(1, b) = "yaw [rad]"

ElseIf x = "distance" Then
    ThisWorkbook.Worksheets(a).Cells(1, b) = "distance [m]"

ElseIf x = "voltage_battery" Then
    ThisWorkbook.Worksheets(a).Cells(1, b) = "voltage_battery [V]"

ElseIf x = "current_battery" Then
    ThisWorkbook.Worksheets(a).Cells(1, b) = "current_battery [A]"

ElseIf x = "battery_remaining" Then
    ThisWorkbook.Worksheets(a).Cells(1, b) = "battery_remaining [%]"

ElseIf x = "heading" Then
    ThisWorkbook.Worksheets(a).Cells(1, b) = "heading [°]"
```

```

        End If
    Next b
Next a

'Con esto ponemos el "sampling time" en la columna de cada mensaje y de paso
recojemos al extension en columnas y filas de cada msg

'Tambien queremos asignar un color con cada tipo de mensaje y congelar las

'----- Para pruebas y rapida ejecucion (nos ahorra unos 3
minutos)
'timestamp_pos = 3
'msg_len = 15
'Dim mensajes() As String
'ReDim mensajes(1 To 15)
'mensajes(1) = "mlHeartbeat"
'mensajes(2) = "mlGlobalPositionInt"
'mensajes(3) = "mlGPSRawInt"
'mensajes(4) = "mlScaledIMU"
'mensajes(5) = "mlHighresIMU"
'mensajes(6) = "mlVibration"
'mensajes(7) = "mlScaledPressure"
'mensajes(8) = "mlAltitude"
'mensajes(9) = "mlAttitude"
'mensajes(10) = "mlRangeFinder"
'mensajes(11) = "mlDistanceSensor"
'mensajes(12) = "mlHomePosition"
'mensajes(13) = "mlNavControllerOut"
'mensajes(14) = "mlVfrHud"
'mensajes(15) = "mlEkfStatusReport"

'-----

Dim len_h() As Integer
Dim len_v() As Integer

ReDim len_v(1 To msg_len)

```

```

ReDim len_h(1 To msg_len)

'Add one last column in the last position of every sheet as Sampling Time.
Coloring sheets and freezing panels

For a = 1 To msg_len
    numv = ThisWorkbook.Worksheets(a).UsedRange.Rows.Count 'row length of a
message
    len_v(a) = numv

    numh = ThisWorkbook.Worksheets(a).UsedRange.Columns.Count 'columns length
of a message
    len_h(a) = numh + 1

    ThisWorkbook.Worksheets(a).Cells(1, len_h(a)) = "Sampling Time [s]"
'seting "sampling time" at the end of every colum

    Sheets(a).Select

    Set mat = ThisWorkbook.Worksheets(a).Range("A:N") 'we assure that all
values are correctly sorted
    Set col = ThisWorkbook.Worksheets(a).Range("B:B")
    mat.Sort key1:=col, Order1:=xlAscending, Header:=xlYes

    ActiveSheet.Range("B2").Select 'this comand freeze all rows and columns
out of the "virtual square" that have as an up-lefted vertix the location we
set
    ActiveWindow.FreezePanes = True

    ThisWorkbook.Worksheets(a).Columns("A:Z").AutoFit

Next a
'
'Eliminamos todos los "0" que encontremos al inicio de cada mensaje (por UNIX)
'
For a = 1 To msg_len
    x = ThisWorkbook.Worksheets(a).Cells(2, timestamp_pos)
    Worksheets(a).Select
    Do While x = 0

```

```

ThisWorkbook.Worksheets(a).Cells(2, timestamp_pos).EntireRow.Delete
x = ThisWorkbook.Worksheets(a).Cells(2, timestamp_pos)
len_v(a) = len_v(a) - 1
If IsEmpty(x) Then
    Exit Do
End If
Loop
Next a
'----- !! !! !! !!
'----- !! !! !! !!
'Ahora ponemos los datos de las lecturas de los distintos sensores en hojas
separadas
'Damos por asumido que los sensores de la IMU van a ser el 2 y el 3, los
sensores de la presión el 1 y el 2, y que la IMU siempre aparecerá antes que
la presión.
For a = 1 To msg_len
    If mensajes(a) = "m1ScaledIMU" Then
        ThisWorkbook.Worksheets(a).Name = "m1ScaledIMU2"
        sh_IMU = a
    ElseIf mensajes(a) = "m1ScaledPressure" Then
        ThisWorkbook.Worksheets(a).Name = "m1ScaledPressure1"
        sh_Pres = a
    End If
Next a

Worksheets(sh_IMU).Copy After:=Worksheets(sh_IMU)
ThisWorkbook.Worksheets(sh_IMU + 1).Name = "m1ScaledIMU3"
Worksheets(sh_Pres + 1).Copy After:=Worksheets(sh_Pres + 1)
ThisWorkbook.Worksheets(sh_Pres + 2).Name = "m1ScaledPressure2"

Worksheets(sh_IMU + 1).Range("A2:AA2000").Delete
Worksheets(sh_Pres + 2).Range("A2:AA2000").Delete
'----- !! !! !! !!
'----- !! !! !! !!

sheets_len = Sheets.Count - 1

```

```
Dim messages() As String 'This vector is containing the different kinds of
messages that we can read
```

```
ReDim messages(1 To sheets_len)
```

```
Dim v_len() As Integer
```

```
ReDim v_len(1 To sheets_len)
```

```
Dim h_len() As Integer
```

```
ReDim h_len(1 To sheets_len)
```

```
For a = 1 To sheets_len
```

```
    Sheets(a).Select
```

```
    x = ThisWorkbook.Worksheets(a).UsedRange.Columns.Count
```

```
    h_len(a) = x
```

```
Next a
```

```
For a = 1 To sheets_len
```

```
    Sheets(a).Select
```

```
    x = ThisWorkbook.Worksheets(a).UsedRange.Rows.Count
```

```
    v_len(a) = x
```

```
Next a
```

```
For a = 1 To sheets_len
```

```
    messages(a) = ThisWorkbook.Worksheets(a).Name
```

```
    Sheets(a).Select
```

```
    Sheets(a).Tab.ColorIndex = a + 2 'this is how to fill spaces with colours
(numered coulours by default)
```

```
    Sheets(a).Range("A1").Select
```

```
    For q = 2 To v_len(a) ' with this we paint all the first column
```

```
        Sheets(a).Select
```

```
        ThisWorkbook.Worksheets(a).Cells(q, 1).Interior.ColorIndex = a + 2
```

```
    Next q
```

```

For k = 1 To h_len(a) 'with this we pint all the first row
    Sheets(a).Select
    ThisWorkbook.Worksheets(a).Cells(1, k).Interior.ColorIndex = a + 2
    ThisWorkbook.Worksheets(a).Cells(1, k).Borders.LineStyle =
xlContinuous
    ThisWorkbook.Worksheets(a).Cells(1, k).Borders.Weight = xlThick
Next k
Next a

count_IMU = 1
count_pres = 1

For a = 1 To sheets_len
    x = ThisWorkbook.Worksheets(a).Name
    If x = "m1ScaledIMU2" Then
        For b = 1 To h_len(a)
            y = ThisWorkbook.Worksheets(a).Cells(1, b)
            If y = "sensor" Then
                For c = 2 To v_len(a)
                    curow = ThisWorkbook.Worksheets(a).Cells(c, b)
                    If curow = 3 Then
                        count_IMU = count_IMU + 1
                        ThisWorkbook.Worksheets(a).Cells(c, b).EntireRow.Copy
                        ThisWorkbook.Worksheets(a + 1).Cells(count_IMU,
1).PasteSpecial
                        ThisWorkbook.Worksheets(a + 1).Cells(count_IMU,
1).Interior.ColorIndex = (a + 1 + 2)
                        ThisWorkbook.Worksheets(a).Cells(c, 1).EntireRow.Delete

                        c = c - 1
                        v_len(a) = v_len(a) - 1
                        v_len(a + 1) = v_len(a + 1) + 1
                    End If
                Next c
            End If
        Next b
    ElseIf x = "m1ScaledPressure1" Then

```

```

For b = 1 To h_len(a)
    y = ThisWorkbook.Worksheets(a).Cells(1, b)
    If y = "sensor" Then
        For c = 2 To v_len(a)
            curow = ThisWorkbook.Worksheets(a).Cells(c, b)
            If curow = 2 Then
                count_pres = count_pres + 1
                ThisWorkbook.Worksheets(a).Cells(c, b).EntireRow.Copy
                ThisWorkbook.Worksheets(a + 1).Cells(count_pres,
1).PasteSpecial
                ThisWorkbook.Worksheets(a + 1).Cells(count_pres,
1).Interior.ColorIndex = (a + 1 + 2)
                ThisWorkbook.Worksheets(a).Cells(c, 1).EntireRow.Delete

                c = c - 1
                v_len(a) = v_len(a) - 1
                v_len(a + 1) = v_len(a + 1) + 1
            End If
        Next c
    End If
Next b
End If
Next a

```

'Creación de la variable xy_acc

'We create a vector of all desired messages, and futurely see if it's able to plot all graphs we wanted

'We define all the variables that we are interested in, and that are going to be plotted

Dim desired_messages(1 To 8, 1 To 8) As Variant ' This will be the matrix where we control if we have all the variables we want, and then the corresponding position of information of interest

Dim current_messages(1 To 8, 1 To 8) As Variant

```
desired_messages(1, 1) = "m1GlobalPositionInt"  
desired_messages(1, 2) = "timeStamp [s]"  
desired_messages(1, 3) = "latitude [°]"  
desired_messages(1, 4) = "longitude [°]"  
desired_messages(1, 5) = "altitude [m]"  
desired_messages(1, 6) = "X [m]"  
desired_messages(1, 7) = "Y [m]"  
desired_messages(1, 8) = "heading [°]"
```

```
desired_messages(2, 1) = "m1ScaledIMU2"  
desired_messages(2, 2) = "timeStamp [s]"  
desired_messages(2, 3) = "x_lin_acc [m/s^2]"  
desired_messages(2, 4) = "y_lin_acc [m/s^2]"  
desired_messages(2, 5) = "z_lin_acc [m/s^2]"  
desired_messages(2, 6) = "altitude [m]"  
desired_messages(2, 7) = "xy_lin_acc [m/s^2]"
```

```
desired_messages(3, 1) = "m1ScaledIMU3"  
desired_messages(3, 2) = "timeStamp [s]"  
desired_messages(3, 3) = "x_lin_acc [m/s^2]"  
desired_messages(3, 4) = "y_lin_acc [m/s^2]"  
desired_messages(3, 5) = "z_lin_acc [m/s^2]"  
desired_messages(3, 6) = "altitude [m]"  
desired_messages(3, 7) = "xy_lin_acc [m/s^2]"
```

```
desired_messages(4, 1) = "m1ScaledPressure1"  
desired_messages(4, 2) = "timeStamp [s]"  
desired_messages(4, 3) = "pressure_abs [hPa]"  
desired_messages(4, 4) = "temperature [°C]"  
desired_messages(4, 5) = "altitude [m]"  
desired_messages(4, 6) = "altitude_hipso [m]"
```

```
desired_messages(5, 1) = "m1ScaledPressure2"  
desired_messages(5, 2) = "timeStamp [s]"
```



```
desired_messages(5, 3) = "pressure_abs [hPa]"
desired_messages(5, 4) = "temperature [°C]"
desired_messages(5, 5) = "altitude [m]"
desired_messages(5, 6) = "altitude_hipso [m]"
```

```
desired_messages(6, 1) = "mlAttitude"
desired_messages(6, 2) = "timeStamp [s]"
desired_messages(6, 3) = "roll [rad]"
desired_messages(6, 4) = "pitch [rad]"
desired_messages(6, 5) = "yaw [rad]"
```

```
desired_messages(7, 1) = "mlRangeFinder"
desired_messages(7, 2) = "timeStamp [s]"
desired_messages(7, 3) = "distance [m]"
desired_messages(7, 4) = "altitude [m]"
```

```
desired_messages(8, 1) = "mlSysStatus"
desired_messages(8, 2) = "timeStamp [s]"
desired_messages(8, 3) = "voltage_battery [V]"
desired_messages(8, 4) = "current_battery [A]"
desired_messages(8, 5) = "battery_remaining [%]"
```

'Ponemos las últimas variables que nos falta en cada label para poder graficar todos los plots que hemos propuesto

```
For a = 1 To sheets_len
    x = ThisWorkbook.Worksheets(a).Name

    If x = "mlGlobalPositionInt" Then
        Sheets(a).Select
        ThisWorkbook.Worksheets(a).Cells(1, h_len(a)).Select
        ActiveCell.EntireColumn.Select
        Selection.Insert Shift:=xlToLeft
        Selection.Insert Shift:=xlToLeft
        ThisWorkbook.Worksheets(a).Cells(1, h_len(a)) = "X [m]"
        ThisWorkbook.Worksheets(a).Cells(1, h_len(a) + 1) = "Y [m]"
        h_len(a) = h_len(a) + 2
    End If
Next a
```

```
        ThisWorkbook.Worksheets(a).Cells(1, 1) =  
ThisWorkbook.Worksheets(a).Name
```

```
ElseIf x = "m1ScaledIMU2" Then
```

```
    Sheets(a).Select
```

```
    ThisWorkbook.Worksheets(a).Cells(1, h_len(a)).Select
```

```
    ActiveCell.EntireColumn.Select
```

```
    Selection.Insert Shift:=xlToLeft
```

```
    ThisWorkbook.Worksheets(a).Cells(1, h_len(a)) = "altitude [m]"
```

```
    h_len(a) = h_len(a) + 1
```

```
    ThisWorkbook.Worksheets(a).Cells(1, 1) =  
ThisWorkbook.Worksheets(a).Name
```

```
    ThisWorkbook.Worksheets(a).Cells(1, h_len(a)).Select
```

```
    ActiveCell.EntireColumn.Select
```

```
    Selection.Insert Shift:=xlToLeft
```

```
    ThisWorkbook.Worksheets(a).Cells(1, h_len(a)) = "xy_lin_acc [m/s^2]"
```

```
    h_len(a) = h_len(a) + 1
```

```
    ThisWorkbook.Worksheets(a).Cells(1, 1) =  
ThisWorkbook.Worksheets(a).Name
```

```
ElseIf x = "m1ScaledIMU3" Then
```

```
    Sheets(a).Select
```

```
    ThisWorkbook.Worksheets(a).Cells(1, h_len(a)).Select
```

```
    ActiveCell.EntireColumn.Select
```

```
    Selection.Insert Shift:=xlToLeft
```

```
    ThisWorkbook.Worksheets(a).Cells(1, h_len(a)) = "altitude [m]"
```

```
    h_len(a) = h_len(a) + 1
```

```
    ThisWorkbook.Worksheets(a).Cells(1, 1) =  
ThisWorkbook.Worksheets(a).Name
```

```
    ThisWorkbook.Worksheets(a).Cells(1, h_len(a)).Select
```

```
    ActiveCell.EntireColumn.Select
```

```
    Selection.Insert Shift:=xlToLeft
```

```
    ThisWorkbook.Worksheets(a).Cells(1, h_len(a)) = "xy_lin_acc [m/s^2]"
```

```
    h_len(a) = h_len(a) + 1
```

```
        ThisWorkbook.Worksheets(a).Cells(1, 1) =  
ThisWorkbook.Worksheets(a).Name
```

```
ElseIf x = "mlScaledPressure1" Then
```

```
    Sheets(a).Select
```

```
    ThisWorkbook.Worksheets(a).Cells(1, h_len(a)).Select
```

```
    ActiveCell.EntireColumn.Select
```

```
    Selection.Insert Shift:=xlToLeft
```

```
    ThisWorkbook.Worksheets(a).Cells(1, h_len(a)) = "altitude [m]"
```

```
    h_len(a) = h_len(a) + 1
```

```
    ThisWorkbook.Worksheets(a).Cells(1, 1) =  
ThisWorkbook.Worksheets(a).Name
```

```
    ThisWorkbook.Worksheets(a).Cells(1, h_len(a)).Select
```

```
    ActiveCell.EntireColumn.Select
```

```
    Selection.Insert Shift:=xlToLeft
```

```
    ThisWorkbook.Worksheets(a).Cells(1, h_len(a)) = "altitude_hipso [m]"
```

```
    h_len(a) = h_len(a) + 1
```

```
    ThisWorkbook.Worksheets(a).Cells(1, 1) =  
ThisWorkbook.Worksheets(a).Name
```

```
ElseIf x = "mlScaledPressure2" Then
```

```
    Sheets(a).Select
```

```
    ThisWorkbook.Worksheets(a).Cells(1, h_len(a)).Select
```

```
    ActiveCell.EntireColumn.Select
```

```
    Selection.Insert Shift:=xlToLeft
```

```
    ThisWorkbook.Worksheets(a).Cells(1, h_len(a)) = "altitude [m]"
```

```
    h_len(a) = h_len(a) + 1
```

```
    ThisWorkbook.Worksheets(a).Cells(1, 1) =  
ThisWorkbook.Worksheets(a).Name
```

```
    ThisWorkbook.Worksheets(a).Cells(1, h_len(a)).Select
```

```
    ActiveCell.EntireColumn.Select
```

```
    Selection.Insert Shift:=xlToLeft
```

```
    ThisWorkbook.Worksheets(a).Cells(1, h_len(a)) = "altitude_hipso [m]"
```

```
    h_len(a) = h_len(a) + 1
```

```

        ThisWorkbook.Worksheets(a).Cells(1, 1) =
ThisWorkbook.Worksheets(a).Name
    ElseIf x = "mlRangeFinder" Then
        Sheets(a).Select
        ThisWorkbook.Worksheets(a).Cells(1, h_len(a)).Select
        ActiveCell.EntireColumn.Select
        Selection.Insert Shift:=xlToLeft
        ThisWorkbook.Worksheets(a).Cells(1, h_len(a)) = "altitude [m]"
        h_len(a) = h_len(a) + 1
        ThisWorkbook.Worksheets(a).Cells(1, 1) =
ThisWorkbook.Worksheets(a).Name

```

```

    End If

```

```

Next a

```

'Ahora vamos a chequear si tenemos todas las variables que nos interesan.

```

For a = 1 To 8

```

```

    found = 0

```

```

        For b = 1 To 8

```

```

            found = 0

```

```

                For c = 1 To sheets_len 'fila desired message

```

```

                    If found = 1 Then

```

```

                        Exit For

```

```

                    End If

```

```

                For d = 1 To h_len(c) 'columna desired message

```

```

                    x_label = ThisWorkbook.Worksheets(c).Cells(1, 1)

```

```

                    y_label = desired_messages(a, 1)

```

```

                    x = ThisWorkbook.Worksheets(c).Cells(1, d)

```

```

                    y = desired_messages(a, b)

```

```

                    If x = y And x_label = y_label And d <> 1 Then

```

```

                        current_messages(a, b) = d

```

```

                        found = 1

```

```

                        Exit For

```

```

                    ElseIf x = y And x_label = y_label And d = 1 Then

```

```

                        current_messages(a, b) = c

```

```

        found = 1
        Exit For
    ElseIf d = h_len(sheets_len) And c = sheets_len Then
        current_messages(a, b) = 0
    End If
Next d
Next c

Next b
Next a

'Buscamos el tiempo minimo de cada hoja y nos quedamos con el más pequeño
min_time = ThisWorkbook.Worksheets(1).Cells(2, timestamp_pos) 'initial
condition
For a = 1 To sheets_len
    x = ThisWorkbook.Worksheets(a).Cells(2, timestamp_pos)
    y = ThisWorkbook.Worksheets(a).Cells(2, timestamp_pos + 1)
    If x < min_time And Not IsEmpty(x) Then
        min_time = x
    End If
Next a

'Pasamos el tiempo a segundos de la columna timeStamp

For a = 1 To sheets_len
    ini = ThisWorkbook.Worksheets(a).Cells(2, timestamp_pos)
    For b = 2 To v_len(a)
        ThisWorkbook.Worksheets(a).Cells(b, timestamp_pos).Copy
        y = ThisWorkbook.Worksheets(a).Cells(b, timestamp_pos)
        ThisWorkbook.Worksheets(a).Cells(b, timestamp_pos + 1).PasteSpecial

        ThisWorkbook.Worksheets(a).Select
        x = ThisWorkbook.Worksheets(a).Cells(b, timestamp_pos)

```

```

        ThisWorkbook.Worksheets(a).Cells(b, timestamp_pos) = (x - min_time) /
1000
        ThisWorkbook.Worksheets(a).Cells(b, timestamp_pos + 1) = (y - ini) /
1000
    Next b
Next a

```

'Ponemos los tiempos correspondientes de cada Sampling Time

'Debemos tener en cuenta que algunas labels van a tener el parámetro de "sensor" cuando haya varias lecturas de distintos dispositivos para un mismo parámetro

' como puede ser el caso de la presión y la temperatura.

```
tol = 0.1
```

```
For a = 1 To sheets_len
```

```
    act_label = ThisWorkbook.Worksheets(a).Cells(1, 1)
```

```
    For b = 2 To v_len(a) - 1
```

```
        x = ThisWorkbook.Worksheets(a).Cells(b, timestamp_pos)
```

```
        y = ThisWorkbook.Worksheets(a).Cells(b + 1, timestamp_pos)
```

```
        Z = y - x
```

```
        ThisWorkbook.Worksheets(a).Cells(b + 1, h_len(a)) = Z
```

```
        If Z > (1 + tol) Or Z < (1 - tol) Then
```

```
            For c = 1 To h_len(a)
```

```
                ThisWorkbook.Worksheets(a).Cells(b + 1,
c).Interior.ColorIndex = 38
```

```
            Next c
```

```
        End If
```

```
    Next b
```

```
Next a
```

'Separamos las muestras que sobrepasen la telorancia del sampling_time

```
For a = 1 To sheets_len
```

```
    count_y = 3
```

```
    Sheets(a).Select
```

```
    For b = 2 To v_len(a)
```

```
        ThisWorkbook.Worksheets(a).Cells(b, 1).Select
```

```

x = ThisWorkbook.Worksheets(a).Cells(b, 1).Interior.ColorIndex
Z = ThisWorkbook.Worksheets(a).Cells(b, 1)
If x = 38 Then 'color de los sampling times
    ThisWorkbook.Worksheets(a).Cells(b, 1).EntireRow.Copy
    ThisWorkbook.Worksheets(a).Cells(v_len(a) + count_y,
1).PasteSpecial
    ThisWorkbook.Worksheets(a).Cells(b, 1).EntireRow.Delete
    v_len(a) = v_len(a) - 1
    b = b - 1
    count_y = count_y + 1
ElseIf IsEmpty(Z) Then
    Exit For
End If
Next b

ThisWorkbook.Worksheets(a).Cells(v_len(a) + 2, 1) = "Sampling Time ERROR list"
ThisWorkbook.Worksheets(a).Cells(v_len(a) + 2, 1).Interior.ColorIndex = 38
Next a

```

ANEXO M.2 Código del segundo módulo: Preprocesador de las variables de interés para los gráficos – Realización de los gráficos

'Transformamos las variables de presión, temperatura y aceleración a altura, y la latitud y longitud GPS a unidades métricas

```
Pi = WorksheetFunction.Pi()
a = 6378137
b = 6356752.31424518
c = 6399593.62575849
exc = 8.18191908426203E-02
exc2 = 8.20944379496945E-02
e2 = 6.73949674227624E-03

For a = 1 To 8

    If a = 1 Then
        For b = 2 To v_len(current_messages(a, 1))
            lat = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 3))
            lon = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 4))

            radlat = (Pi / 180) * lat
            radlon = (Pi / 180) * lon
            hus = Fix((lon / 6) + 31)
            merid = (6 * hus) - 183
            delt_lamb = radlon - ((merid * Pi) / 180)
            aa = Cos(radlat) * Sin(delt_lamb)
            xi = 0.5 * Log((1 + aa) / (1 - aa))
            eta = Atn((Tan(radlat)) / Cos(delt_lamb)) - radlat
            ni = (c / (1 + e2 * (Cos(radlat)) ^ 2) ^ (1 / 2)) * 0.9996
            Z = (e2 / 2) * (xi * xi) * (Cos(radlat)) ^ 2
            a1 = Sin(2 * radlat)
            a2 = a1 * (Cos(radlat)) ^ 2
            j2 = radlat + (a1 / 2)
            j4 = ((3 * j2) + a2) / 4
            j6 = (5 * j4 + a2 * (Cos(radlat)) ^ 2) / 3

            alfa = (3 / 4) * e2
            beta = (5 / 3) * alfa * alfa
            gama = (35 / 27) * alfa * alfa * alfa
            fi = 0.9996 * c * (radlat - (alfa * j2) + (beta * j4) - (gama *
j6))

            este = xi * ni * (1 + zeta / 3) + 500000
            norte2 = eta * ni * (1 + zeta) + fi

            ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 7)) = norte2 'shan dactualitzar les "len_h"
            ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 6)) = este
        Next b

        'Esta parte nos va a servir para dar las unidades en terminos relativos
        (punto despegue --> (0,0) )
        Sheets(current_messages(a, 1)).Select
```



```

'           col_x = Range(Cells(2, current_messages(a, 6)),
Cells(v_len(current_messages(a, 1)), current_messages(a, 6)))
'           col_y = Range(Cells(2, current_messages(a, 7)),
Cells(v_len(current_messages(a, 1)), current_messages(a, 7)))
'           Range(Cells(2, current_messages(a, 6)), Cells(v_len(current_messages(a,
1)), current_messages(a, 6))).Select
'           col_x=Application.WorksheetFunction.Min(col_x)

           first_x = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(2,
current_messages(a, 6))
           first_y = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(2,
current_messages(a, 7))

           For b = 2 To v_len(current_messages(a, 1))

               x = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 6))
               y = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 7))
               ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 6)) = x - first_x
               ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 7)) = y - first_y
           Next b

ElseIf a = 2 Then
           For b = 2 To v_len(current_messages(a, 1))

               Sheets(current_messages(a, 1)).Select
               ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 5)).Select
               x = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 5))
               ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 5)) = x + 1
           Next b

           alt_ini = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(2,
current_messages(1, 5)) 'valor inicial del gps
           ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(2,
current_messages(a, 6)) = alt_ini

           For b = 3 To v_len(current_messages(a, 1))

               If b = 3 Then

                   velim1 = 0

                   ti = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
timestamp_pos)
                   tim1 = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b -
1, timestamp_pos)
                   dif_t = ti - tim1

                   accelim1 = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b
- 1, current_messages(a, 5)) * 9.81 / 1000

                   posi = alt_ini + (velim1 * dif_t) + 0.5 * accelim1 * (dif_t) ^ 2
'We calculate the position knowing the accel

```

```

        ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 6)) = posi
        velim2 = velim1
        posim1 = alt_ini

    Else

        ti = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
timestamp_pos)
        tim1 = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b
- 1, timestamp_pos)
        tim2 = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b
- 2, timestamp_pos)
        accelim1 = ThisWorkbook.Worksheets(current_messages(a,
1)).Cells(b - 1, current_messages(a, 5)) * 9.81 / 1000
        accelim2 = ThisWorkbook.Worksheets(current_messages(a,
1)).Cells(b - 2, current_messages(a, 5)) * 9.81 / 1000
        dif_t1 = tim1 - tim2

        velim1 = (velim2 + accelim2 * dif_t1)

        dif_t = ti - tim1
        posi = posim1 + (velim1 * dif_t) + 0.5 * accelim1 * (dif_t) ^
2 'We calculate the position knowing the accel
        ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 6)) = posi
        velim2 = velim1

    End If
Next b

'
For b = 2 To v_len(current_messages(a, 1))
    x = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 3))
    y = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 4))
    ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 7)) = Sqr(x * x + y * y)
Next b

ElseIf a = 3 Then

    For b = 2 To v_len(current_messages(a, 1))

        x = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 5))
        ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 5)) = x + 1

    Next b
    ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(2,
current_messages(a, 6)) = alt_ini

    For b = 3 To v_len(current_messages(a, 1))

        If b = 3 Then

```

```

        velim1 = 0

        ti = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
timestamp_pos)
        tim1 = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b -
1, timestamp_pos)
        dif_t = ti - tim1

        accelim1 = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b
- 1, current_messages(a, 5)) * 9.81 / 1000

        posi = alt_ini + (velim1 * dif_t) + 0.5 * accelim1 * (dif_t) ^ 2
'We calculate the position knowing the accel
        ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 6)) = posi
        velim2 = velim1
        posim1 = alt_ini
        Else

        ti = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
timestamp_pos)
        tim1 = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b -
1, timestamp_pos)
        tim2 = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b -
2, timestamp_pos)
        accelim1 = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b
- 1, current_messages(a, 5)) * 9.81 / 1000
        accelim2 = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b
- 2, current_messages(a, 5)) * 9.81 / 1000
        dif_t1 = tim1 - tim2

        velim1 = (velim2 + accelim2 * dif_t1)

        dif_t = ti - tim1
        posi = posim1 + (velim1 * dif_t) + 0.5 * accelim1 * (dif_t) ^ 2
'We calculate the position knowing the accel
        ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 6)) = posi
        velim2 = velim1
        End If
    Next b

    For c = 2 To v_len(current_messages(a, 1))
        x = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(c,
current_messages(a, 3))
        y = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(c,
current_messages(a, 4))
        ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(c,
current_messages(a, 7)) = Sqr(x * x + y * y)
    Next c

ElseIf a = 4 Then ' presión_1
'initial conditions
    Sheets(current_messages(a, 1)).Select

    For b = 2 To v_len(current_messages(a, 1))

```

```

        x = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 3)) * 100 'c_m(a,3) es columna de valores de presion en
hPa
        local_p = 101880 'Sea level pressure of the flight in hectopascals
'The following equation is determined by the standard of
aeronautics in the throposphere. The local pressure might be
        ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 5)) = (((288.15 - ((288.15) / ((x / local_p) ^ (-6.5 /
34.1632)))) * (1 / 6.5)) * 1000
        Next b

        hm1 = alt_ini
        ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(2,
current_messages(a, 6)) = alt_ini
        y = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(2,
current_messages(a, 3)) * 100
        tempm1 = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(2,
current_messages(a, 4))

        For b = 3 To v_len(current_messages(a, 1))
            x = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 3)) * 100 'c_m(a,3) es columna de valores de presion en
hPa

            temp = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 4))
            av = (tempm1 + temp + 2 * 273.15) / 2

            alth = hm1 + (287.053 * av) / (9.81) * Log(y / x)
            ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 6)) = alth

            tempm1 = temp
            y = x
            hm1 = alth
        Next b

    ElseIf a = 5 Then
        Sheets(current_messages(a, 1)).Select
        For b = 2 To v_len(current_messages(a, 1))
            x = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 3)) * 100 'c_m(a,3) es columna de valores de presion en
hPa

            local_p = 101880 'Sea level pressure of the flight in hectopascals
'The following equation is determined by the standard of
aeronautics in the throposphere. The local pressure might be
            ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 5)) = (((288.15 - ((288.15) / ((x / local_p) ^ (-6.5 /
34.1632)))) * (1 / 6.5)) * 1000
            Next b

            hm1 = alt_ini
            ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(2,
current_messages(a, 6)) = alt_ini
            y = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(2,
current_messages(a, 3)) * 100
            tempm1 = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(2,
current_messages(a, 4))

```

```

        For b = 3 To v_len(current_messages(a, 1))
            x = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 3)) * 100 'c_m(a,3) es columna de valores de presion en
hPa

            temp = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 4))
            av = (tempm1 + temp + 2 * 273.15) / 2

            alth = hm1 + (287.053 * av) / (9.81) * Log(y / x)
            ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 6)) = alth

            tempm1 = temp
            y = x
            hm1 = alth
        Next b

    ElseIf a = 7 Then
        alt_ini = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(2,
current_messages(1, 5)) 'primer valor de la altura, dada por el GPS
        For b = 2 To v_len(current_messages(a, 1))
            x = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 3))

            ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 4)) = x + alt_ini
        Next b

    ElseIf a = 8 Then
        For b = 2 To v_len(current_messages(a, 1))
            x = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 3))
            ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 3)) = x / 1000
            y = ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 4))
            ThisWorkbook.Worksheets(current_messages(a, 1)).Cells(b,
current_messages(a, 4)) = y / 100
        Next b
    End If
Next a

```

'1- Grafico de posición vertical diferentes sensores

'Plot 1.1.

```

Dim plots_11() As Integer
ReDim plots_11(1 To 6)

```

```

Dim column_11() As Integer
ReDim columns_11(1 To 6)

```

```

plots_11(1) = 1
plots_11(2) = 4 'aplicaremos un for de 1 a 2 (para recorrer la ecuacion de
presion e hipsometrica)
plots_11(3) = 4
plots_11(4) = 5
plots_11(5) = 5
plots_11(6) = 7

```

```

columns_11(1) = 5
columns_11(2) = 5
columns_11(3) = 6
columns_11(4) = 5
columns_11(5) = 6
columns_11(6) = 4

```

Count = 2 ' cada vez que cambiemos de hoja tambien sumaremos un contador para automatizar el proceso de cambio de hoja, y crear la nueva serie de graficos

```

Set graf = Sheets(sheets_len + Count).ChartObjects.Add(Left:=62.8, Top:=(45.6 / 3) * 21, Width:=1000, Height:=285.7) ' Top y Left se toman a 'partir del margen superior e izquierdo de la Hoja, respectivamente. el ancho de una columna corresponde aprox a 58 unidades en "Left"

```

```

ibase = ThisWorkbook.Worksheets(current_messages(plots_11(1), 1)).Cells(2, current_messages(plots_11(1), columns_11(1)))
ttop = ThisWorkbook.Worksheets(current_messages(plots_11(1), 1)).Cells(2, current_messages(plots_11(1), columns_11(1)))

```

```

For a = 1 To 6 '(gps,p1,p2,t1,t2,laser)

```

```

'_'
    'Buscamos maximos y minimos
    Sheets(current_messages(plots_11(a), 1)).Select
    rango = ThisWorkbook.Worksheets(current_messages(plots_11(a), 1)).Range(Cells(2, current_messages(plots_11(a), columns_11(a))), Cells(v_len(current_messages(plots_11(a), 1)), current_messages(plots_11(a), columns_11(a))))

```

```

    val_max = Application.WorksheetFunction.Max(rango)
    val_min = Application.WorksheetFunction.Min(rango)

```

```

    'we will consider the maximum and the minimum of the function for the plots
    If val_max > ttop Then
        ttop = val_max
    End If
    If val_min < ibase Then
        ibase = val_min
    End If

```

```

    ThisWorkbook.Sheets(current_messages(plots_11(a), 1)).Cells(1, 1).Select
    'we have to consider the same color as the labels
    colorserie = Selection.Interior.Color
    '_,

```

```

    ThisWorkbook.Worksheets(current_messages(plots_11(a), 1)).Cells(2, timestamp_pos).Select
    ThisWorkbook.Worksheets(current_messages(plots_11(a), 1)).Cells(v_len(current_messages(plots_11(a), 1)), timestamp_pos).Select
    ThisWorkbook.Worksheets(current_messages(plots_11(a), 1)).Cells(2, current_messages(plots_11(a), columns_11(a))).Select
    ThisWorkbook.Worksheets(current_messages(plots_11(a), 1)).Cells(v_len(current_messages(plots_11(a), 1)), current_messages(plots_11(a), columns_11(a))).Select
    With graf.Chart
        .SeriesCollection.NewSeries
        .ChartType = xlXYScatterLines
    End With

```

```

        .SeriesCollection(a).XValues =
ThisWorkbook.Worksheets(current_messages(plots_11(a), 1)).Range(Cells(2,
timestamp_pos), Cells(v_len(current_messages(plots_11(a), 1)), timestamp_pos))
'x values
        .SeriesCollection(a).Values =
ThisWorkbook.Worksheets(current_messages(plots_11(a), 1)).Range(Cells(2,
current_messages(plots_11(a), columns_11(a))),
Cells(v_len(current_messages(plots_11(a), 1)), current_messages(plots_11(a),
columns_11(a)))) 'y values

    For j = 1 To v_len(current_messages(plots_11(a), 1)) - 1
        .SeriesCollection(a).Points(j).Select
        .SeriesCollection(a).Points(j).Format.Fill.ForeColor.RGB = 1
        .SeriesCollection(a).Points(j).Format.Line.ForeColor.RGB = colorserie
        .SeriesCollection(a).MarkerStyle = xlMarkerStyleCircle
    Next j

        .SeriesCollection(a).Format.Fill.ForeColor.RGB = 1
        .SeriesCollection(a).Format.Line.ForeColor.RGB = colorserie

    If a = 1 Then
        .SeriesCollection(a).Name = "Altitude from GPS "
    ElseIf a = 2 Then
        .SeriesCollection(a).Name = "Theoretical altitude (Standar Ec.)
taken from pressure sensor 1"
    ElseIf a = 3 Then
        .SeriesCollection(a).Name = "Theoretical altitude (Hypsometric Ec.)
taken from pressure sensor 1"
    ElseIf a = 4 Then
        .SeriesCollection(a).Name = "Theoretical altitude (Standar Ec.)
taken from pressure sensor 2"
    ElseIf a = 5 Then
        .SeriesCollection(a).Name = "Theoretical altitude (Hypsometric Ec.)
taken from pressure sensor 2"
    ElseIf a = 6 Then
        .SeriesCollection(a).Name = "Altitude from laser distance sensor"
    End If

        .SetElement (msoElementPrimaryCategoryGridLinesMinor)

'TITULO EN EL EJE X
'TITLE IN THE X AXIS
        .SetElement msoElementPrimaryCategoryAxisTitleAdjacentToAxis
        .Axes(xlCategory, xlPrimary).AxisTitle.Text =
ThisWorkbook.Worksheets(current_messages(plots_11(a), 1)).Cells(1,
timestamp_pos)
        .Axes(xlCategory,
xlPrimary).AxisTitle.Format.TextFrame2.TextRange.Font.Size = 15
'TITULO EN EL EJE Y
'TITLE IN THE Y AXIS
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Text =
ThisWorkbook.Worksheets(current_messages(plots_11(a), 1)).Cells(1,
current_messages(plots_11(a), columns_11(a)))
        .Axes(xlValue,
xlPrimary).AxisTitle.Format.TextFrame2.TextRange.Font.Size = 15

        .Axes(xlValue, xlPrimary).MinimumScale = Round(bbase, 0)
        .Axes(xlValue, xlPrimary).MaximumScale = ttop + (ttop * 0.08)

```

```
'MODIFICAR LINEAS DE LOS EJES
'WE MODIFY THE LINES OF THE AXIS
    .Axes(xlValue, xlPrimary).HasMajorGridlines = True
    '.Axes(xlValue, xlPrimary).HasMinorGridlines = True
    .Axes(xlValue, xlPrimary).MajorGridlines.Border.Color = RGB(20, 19,
156)
    .Axes(xlValue, xlPrimary).MajorGridlines.Border.LineStyle = xlDash
End With
Sheets(sheets_len + Count).Select
Next a
graf.Chart.Legend.Position = xlLegendPositionBottom
```

(Se ha omitido la inclusión de las demás gráficas, ya que representan una estructura de código muy similar a la de este primer plot)

ANEXO M.3 Código del tercer módulo: Identificador de fases de vuelo – Gráfico para la identificación de fases.

```
Dim fila_GPS() As Integer
ReDim fila_GPS(1 To 1)
Dim columna_altura() As Integer
ReDim columna_altura(1 To 1)

'Ubicación de la altura del GPS
fila_GPS(1) = 1
columna_altura(1) = 5

'Vectores en los cuales segmentaremos los puntos de las distintas fases
Dim groundi_samp As Integer
Dim tkoff_samp As Integer
Dim land_samp As Integer
Dim groundf_samp As Integer

vertical_tol_ground = 2 'un ascenso que supere los 2 metros en relación a la
altura tomada en el instante 0, sera motivo para pensar que alomejor se está
produciendo
discont = 5 'una vez se superen los 2 metros, si en los 5 siguientes puntos
(aprox. en los 5 siguientes segundos)
rang_air = 5 'cuando nos encontramos en pleno despegue (ya se han superado 5
puntos fuera de la tolerancia), vamos mirando los 5 siguientes puntos para ver
'si todavía se está ejecutando el despegue. Hay un requisito para que se
considere que se esta realizando el despegue y es que en esos 5 puntos se hayan
sobrepasado los 2 metros
'(0,5 m/s = (2,5m)/5s)
vel_min = 0.5
dist_min = vel_min * discont

altura_inicial = ThisWorkbook.Worksheets(current_messages(fila_GPS(1),
1)).Cells(2, current_messages(fila_GPS(1), columna_altura(1)))
altura_tol = altura_inicial + vertical_tol_ground
Z = altura_inicial

For a = 1 To 4 'v_len(current_messages(fila_GPS(1), 1))
cambio = 0 'variable que, cuando sea 1, nos va a permitir salir del bucle
"while" a fin de poder rellenar el vector de la siguiente fase con los puntos
que correspondan (segun lo calcule nuestro algoritmo)

    If a = 1 Then
        b = 2
        Do While cambio = 0

            Sheets(current_messages(fila_GPS(1), 1)).Select
            ThisWorkbook.Worksheets(current_messages(fila_GPS(1),
1)).Cells(b, current_messages(fila_GPS(1), columna_altura(1))).Select

            x = ThisWorkbook.Worksheets(current_messages(fila_GPS(1),
1)).Cells(b, current_messages(fila_GPS(1), columna_altura(1)))

            If x >= altura_tol Then
                c = b
                For d = c To c + discont
                    ThisWorkbook.Worksheets(current_messages(fila_GPS(1),
1)).Cells(d, current_messages(fila_GPS(1), columna_altura(1))).Select
```

```

                z1
ThisWorkbook.Worksheets(current_messages(fila_GPS(1),          1)).Cells(d,
current_messages(fila_GPS(1), columna_altura(1)))
                If z1 < altura_tol Then
                    b = d
                    Exit For

                ElseIf z1 >= altura_tol And d = c + discont Then

                    groundi_samp = b - 2 'ESTE "2" ES UN FACTOR DE
CORRECCIÓN PARA AJUSTAR MEJOR EL PUNTO INICIAL DEL DESPEGUE

                    cambio = 1
                    End If

                Next d
            End If

            b = b + 1
        Loop
        b = b - 2

    ElseIf a = 2 Then 'miramos el despegue

        Do While cambio = 0

            ThisWorkbook.Worksheets(current_messages(fila_GPS(1),
1)).Cells(b, current_messages(fila_GPS(1), columna_altura(1))).Select
            y = ThisWorkbook.Worksheets(current_messages(fila_GPS(1),
1)).Cells(b - 1, current_messages(fila_GPS(1), columna_altura(1)))
            x = ThisWorkbook.Worksheets(current_messages(fila_GPS(1),
1)).Cells(b, current_messages(fila_GPS(1), columna_altura(1)))
            c = b
            z1 = 0

            For d = c To c + rang_air
                y1 = ThisWorkbook.Worksheets(current_messages(fila_GPS(1),
1)).Cells(d - 1, current_messages(fila_GPS(1), columna_altura(1)))
                x1 = ThisWorkbook.Worksheets(current_messages(fila_GPS(1),
1)).Cells(d, current_messages(fila_GPS(1), columna_altura(1)))
                ThisWorkbook.Worksheets(current_messages(fila_GPS(1),
1)).Cells(d, current_messages(fila_GPS(1), columna_altura(1))).Select
                z1 = z1 + (x1 - y1)

                If z1 < dist_min And d = c + rang_air Then 'cuando ya nos
estamos estabilizando
                    For e = c To c + rang_air
                        y1
ThisWorkbook.Worksheets(current_messages(fila_GPS(1), 1)).Cells(e - 1,
current_messages(fila_GPS(1), columna_altura(1)))
                        x1
ThisWorkbook.Worksheets(current_messages(fila_GPS(1), 1)).Cells(e,
current_messages(fila_GPS(1), columna_altura(1)))

                        If x1 < y1 Then
                            tkoff_samp = e - 1
                            cambio = 1
                            Exit For
                    Next e
                Next d
            Next b
        Loop
    End If
End Sub

```

```

        ElseIf e = c + rang_air Then
            tkoff_samp = e
            cambio = 1
            Exit For
        End If

    Next e

    End If
Next d
    b = b + 1
Loop
    b = b - 2

    ElseIf a = 3 Then ' etapa de la fase "final ground" una vez se ha realizado
el despegue, que se analizará de forma inversa (hablando en terminos de tiempo)
        b = v_len(current_messages(fila_GPS(1), 1))
        altura_inicial = ThisWorkbook.Worksheets(current_messages(fila_GPS(1),
1)).Cells(b, current_messages(fila_GPS(1), columna_altura(1)))
        altura_tol = altura_inicial + vertical_tol_ground

        Do While cambio = 0
            ThisWorkbook.Worksheets(current_messages(fila_GPS(1),
1)).Cells(b, current_messages(fila_GPS(1), columna_altura(1))).Select

            x = ThisWorkbook.Worksheets(current_messages(fila_GPS(1),
1)).Cells(b, current_messages(fila_GPS(1), columna_altura(1)))

            If x >= altura_tol Then
                c = b
                For d = c To c - discont Step -1
                    ThisWorkbook.Worksheets(current_messages(fila_GPS(1),
1)).Cells(d, current_messages(fila_GPS(1), columna_altura(1))).Select
                    z1 = ThisWorkbook.Worksheets(current_messages(fila_GPS(1),
1)).Cells(d,
current_messages(fila_GPS(1), columna_altura(1)))
                    If z1 < altura_tol Then
                        b = d
                        Exit For
                    End If
                Next d
            ElseIf z1 >= altura_tol And d = c - discont Then

                groundf_samp = b + 6 'ESTE "7" ES UN FACTOR DE
CORRECCIÓN PARA AJUSTAR MEJOR EL PUNTO FINAL DEL ATERRIZAJE

                cambio = 1
            End If
        Next d
    End If

    b = b - 1
Loop
    b = b + 6
ElseIf a = 4 Then
    dist_min = dist_min - 0.65 'factor de corrección para dar un poco más
de margen a los descensos, que se efectuan más lentamente que los ascensos
    Do While cambio = 0

```

```

        ThisWorkbook.Worksheets(current_messages(fila_GPS(1),
1)).Cells(b, current_messages(fila_GPS(1), columna_altura(1))).Select
        y = ThisWorkbook.Worksheets(current_messages(fila_GPS(1),
1)).Cells(b + 1, current_messages(fila_GPS(1), columna_altura(1)))
        x = ThisWorkbook.Worksheets(current_messages(fila_GPS(1),
1)).Cells(b, current_messages(fila_GPS(1), columna_altura(1)))
        c = b
        z1 = 0

        For d = c To c - rang_air Step -1
            y1 = ThisWorkbook.Worksheets(current_messages(fila_GPS(1),
1)).Cells(d + 1, current_messages(fila_GPS(1), columna_altura(1)))
            x1 = ThisWorkbook.Worksheets(current_messages(fila_GPS(1),
1)).Cells(d, current_messages(fila_GPS(1), columna_altura(1)))
            ThisWorkbook.Worksheets(current_messages(fila_GPS(1),
1)).Cells(d, current_messages(fila_GPS(1), columna_altura(1))).Select
            z1 = z1 + (x1 - y1)

            If z1 < dist_min And d = c - rang_air Then 'cuando ya nos
estamos estabilizando
                For e = c To c - rang_air Step -1
                    y1 =
ThisWorkbook.Worksheets(current_messages(fila_GPS(1), 1)).Cells(e + 1,
current_messages(fila_GPS(1), columna_altura(1)))
                    x1 =
ThisWorkbook.Worksheets(current_messages(fila_GPS(1), 1)).Cells(e,
current_messages(fila_GPS(1), columna_altura(1)))

                    If x1 < y1 Then
                        land_samp = e + 1
                        cambio = 1
                        Exit For
                    ElseIf e = c - rang_air Then
                        land_samp = e
                        cambio = 1
                        Exit For
                    End If

                Next e

            End If
        Next d
        b = b - 1
    Loop
    b = b - 2
End If

Next a

'Creamos un vector con los puntos de interés de los sectores
Dim puntos_fases() As Integer
ReDim puntos_fases(1 To 6)
puntos_fases(1) = 2
puntos_fases(2) = groundi_samp
puntos_fases(3) = tkoff_samp
puntos_fases(4) = land_samp
puntos_fases(5) = groundf_samp
last = v_len(current_messages(1, 1))
puntos_fases(6) = last

```

```

Dim colors() As Integer
ReDim colors(1 To 6)

colors(1) = 3
colors(2) = 4
colors(3) = 5
colors(4) = 6
colors(5) = 7

'5- Grafico de fases

'Plot 5.1.
Dim plots_51() As Integer
ReDim plots_51(1 To 5)

Dim column_51() As Integer
ReDim columns_51(1 To 5)

plots_51(1) = 1
plots_51(2) = 1
plots_51(3) = 1
plots_51(4) = 1
plots_51(5) = 1

columns_51(1) = 5
columns_51(2) = 5
columns_51(3) = 5
columns_51(4) = 5
columns_51(5) = 5
Count = 6
Set graf = Sheets(sheets_len + Count).ChartObjects.Add(Left:=62.8, Top:=(45.6 / 3), Width:=500, Height:=285.7) ' Top y Left se toman a 'partir del margen superior e izquierdo de la Hoja, respectivamente. el ancho de una columna corresponde aprox a 58 unidades en "Left"

For a = 1 To 5 '(gi,to,cr,la,gf)

    Sheets(a).Select
    ThisWorkbook.Sheets(a).Cells(1, 1).Select 'we have to consider the same color as the labels
    colorserie = Selection.Interior.Color
'
    'Buscamos maximos y minimos

    Sheets(current_messages(plots_51(a), 1)).Select
    rango = ThisWorkbook.Worksheets(current_messages(plots_51(a), 1)).Range(Cells(2, current_messages(plots_51(a), columns_51(a))), Cells(v_len(current_messages(plots_51(a), 1)), current_messages(plots_51(a), columns_51(a))))

    val_max = Application.WorksheetFunction.Max(rango)
    val_min = Application.WorksheetFunction.Min(rango)
'
    ThisWorkbook.Worksheets(current_messages(plots_11(a), 1)).Cells(2, timestamp_pos).Select
    ThisWorkbook.Worksheets(current_messages(plots_11(a), 1)).Cells(v_len(current_messages(plots_11(a), 1)), timestamp_pos).Select

```

```

'          ThisWorkbook.Worksheets(current_messages(plots_11(a), 1)).Cells(2,
current_messages(plots_11(a), columns_11(a))).Select
'          ThisWorkbook.Worksheets(current_messages(plots_11(a),
1)).Cells(v_len(current_messages(plots_11(a), 1)),
current_messages(plots_11(a), columns_11(a))).Select
    With graf.Chart
        .SeriesCollection.NewSeries
        .ChartType = xlXYScatterLines
        .SeriesCollection(a).XValues =
ThisWorkbook.Worksheets(current_messages(plots_51(a),
1)).Range(Cells(puntos_fases(a), timestamp_pos), Cells(puntos_fases(a + 1),
timestamp_pos)) 'x values
        .SeriesCollection(a).Values =
ThisWorkbook.Worksheets(current_messages(plots_51(a),
1)).Range(Cells(puntos_fases(a), current_messages(plots_51(a),
columns_51(a))), Cells(puntos_fases(a + 1), current_messages(plots_51(a),
columns_51(a)))) 'y values

        For j = 1 To (puntos_fases(a + 1) - puntos_fases(a)) - 1
            .SeriesCollection(a).Points(j).Select
            .SeriesCollection(a).Points(j).Format.Fill.ForeColor.RGB = 1
            .SeriesCollection(a).Points(j).Format.Line.ForeColor.RGB = colorserie
            .SeriesCollection(a).MarkerStyle = xlMarkerStyleCircle
        Next j

        .SeriesCollection(a).Format.Fill.ForeColor.RGB = 1
        .SeriesCollection(a).Format.Line.ForeColor.RGB = colorserie

        If a = 1 Then
            .SeriesCollection(a).Name = "Initial ground phase"
        ElseIf a = 2 Then
            .SeriesCollection(a).Name = "Take off"
        ElseIf a = 3 Then
            .SeriesCollection(a).Name = "Cruise phase (climb and aproximation
included)"
        ElseIf a = 4 Then
            .SeriesCollection(a).Name = "Landing"
        ElseIf a = 5 Then
            .SeriesCollection(a).Name = "Final ground phase"
        End If

        .SetElement (msoElementPrimaryCategoryGridLinesMinor)

        'TITULO EN EL EJE X
        'TITLE IN THE X AXIS
        .SetElement msoElementPrimaryCategoryAxisTitleAdjacentToAxis
        .Axes(xlCategory, xlPrimary).AxisTitle.Text =
ThisWorkbook.Worksheets(current_messages(plots_51(a), 1)).Cells(1,
timestamp_pos)
        .Axes(xlCategory,
xlPrimary).AxisTitle.Format.TextFrame2.TextRange.Font.Size = 15
        'TITULO EN EL EJE Y
        'TITLE IN THE Y AXIS
        .Axes(xlValue, xlPrimary).HasTitle = True
        .Axes(xlValue, xlPrimary).AxisTitle.Text =
ThisWorkbook.Worksheets(current_messages(plots_51(a), 1)).Cells(1,
current_messages(plots_51(a), columns_51(a)))
        .Axes(xlValue,
xlPrimary).AxisTitle.Format.TextFrame2.TextRange.Font.Size = 15

```

```

.Axes(xlValue, xlPrimary).MinimumScale = val_min - val_min * 0.05
.Axes(xlValue, xlPrimary).MaximumScale = val_max + (val_max * 0.05)

'MODIFICAR LINEAS DE LOS EJES
'WE MODIFY THE LINES OF THE AXIS
    .Axes(xlValue, xlPrimary).HasMajorGridlines = True
    .Axes(xlValue, xlPrimary).HasMinorGridlines = True
    .Axes(xlValue, xlPrimary).MajorGridlines.Border.Color = RGB(20, 19,
156)
    .Axes(xlValue, xlPrimary).MajorGridlines.Border.LineStyle = xlDash
End With
Sheets(sheets_len + Count).Select
Next a
graf.Chart.Legend.Position = xlLegendPositionBottom

'---- Identificación de performances
'Esta será la última parte del código donde distinguiremos entre movimientos:
'maniobra frontal, trasera, lateral o diagonal (en el plano horizontal a una
misma altitud)
'maniobra ascendiente o descendiente en todo el plano vertical
'maniobra hover y yaw en hover

Dim vec_perf() As String
ReDim vec_perf(1 To 8)
vec_perf(1) = "Time (duration) [s]"
vec_perf(2) = "Performance along Z axis"
vec_perf(3) = "Performance in XY plane"
vec_perf(4) = "Change of distance in altitude [m]"
vec_perf(5) = "Change of distance in XY plane [m]"
vec_perf(6) = "Angular change in XY plane [°]"
vec_perf(8) = "Mean speed module in XY plane [m/s]"
vec_perf(7) = "Mean speed in Z axis [m/s]"
i = 10

For a = 1 To 8
    ThisWorkbook.Worksheets(sheets_len + Count).Cells(2, i + a) = vec_perf(a)
    ThisWorkbook.Worksheets(sheets_len + Count).Cells(2, i +
a).Interior.ColorIndex = 19
    ThisWorkbook.Worksheets(sheets_len + Count).Cells(2, i +
a).Borders.LineStyle = xlContinuous
    ThisWorkbook.Worksheets(sheets_len + Count).Cells(2, i + a).Borders.Weight
= xlThick
Next a

Dim vec_inf() As String
ReDim vec_inf(1 To 7)
vec_inf(1) = "Total Flight time [s]"
vec_inf(2) = "Landing Path Error [m]"
vec_inf(3) = "Altitude. max. [m]"
vec_inf(4) = "Ground Altitude [m]"
vec_inf(5) = "Max speed in ascence [m/s]"
vec_inf(6) = "Max speed in descence [m/s]"
vec_inf(7) = "Max module speed in XY plane [m/s]"
i = 1

For a = 1 To 7
    ThisWorkbook.Worksheets(sheets_len + Count).Cells(23, i + a) = vec_inf(a)

```

```

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(23, i +
a).Interior.ColorIndex = 19
        ThisWorkbook.Worksheets(sheets_len + Count).Cells(23, i +
a).Borders.LineStyle = xlContinuous
        ThisWorkbook.Worksheets(sheets_len + Count).Cells(23, i +
a).Borders.Weight = xlThick
Next a

```

```

Worksheets(sheets_len + Count).Range("K2:R2").Font.Bold = True
Worksheets(sheets_len + Count).Range("B23:H23").Font.Bold = True

```

```

'Maximo y minimo de altitud que printearemos como dato
Sheets(current_messages(1, 1)).Select
rango = ThisWorkbook.Worksheets(current_messages(1, 1)).Range(Cells(2,
current_messages(1, 5)), Cells(v_len(current_messages(1, 1)),
current_messages(1, 5)))

```

```

alt_max = Application.WorksheetFunction.Max(rango)
alt_max = Round(alt_max, 2)
alt_min = Application.WorksheetFunction.Min(rango)
alt_min = Round(alt_min, 2)

```

```

'Tiempo total de vuelo
diftime = groundf_samp - groundi_samp
diftime = Round(diftime, 3)

```

```

'Error entre posicion inicial y final en la lanzadera (H del helipuerto)
x1 = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(groundi_samp,
current_messages(1, 6))
y1 = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(groundi_samp,
current_messages(1, 7))
x2 = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(groundf_samp,
current_messages(1, 6))
y2 = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(groundf_samp,
current_messages(1, 7))
rad1 = Sqr(x1 * x1 + y1 * y1)
rad2 = Sqr(x2 * x2 + y2 * y2)
landpad_error = Abs(rad1 - rad2)
landpad_error = Round(landpad_error, 2)

```

```

'Velocidad máxima en Z
rango = ThisWorkbook.Worksheets(current_messages(1, 1)).Range(Cells(2,
current_messages(1, 8) - 1), Cells(v_len(current_messages(1, 1)),
current_messages(1, 8) - 1))
velz_min = Application.WorksheetFunction.Max(rango) 'recordemos que la
velocidad con signo negativo es ascendente mientras que signo negativo implica
ir hacia abajo

```

```

velz_max = Application.WorksheetFunction.Min(rango)
velz_max = velz_max * (-1)

```

```

'Velocidad maxima en XY
d = v_len(current_messages(1, 1))
hh = ThisWorkbook.Worksheets(current_messages(1, 1)).UsedRange.Columns.Count
For a = 2 To d
    vx = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(a,
current_messages(1, 8) - 3)
    vy = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(a,
current_messages(1, 8) - 2)

```



```

    vxy = Sqr(vx * vx + vy * vy)
    ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(a, hh + 1) = vxy
Next a

rango = ThisWorkbook.Worksheets(current_messages(1, 1)).Range(Cells(2, hh +
1), Cells(v_len(current_messages(1, 1)), hh + 1))
vxy_max = Application.WorksheetFunction.Max(rango)
vxy_max = Round(vxy_max, 2)

ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(a, hh + 1).Select
ActiveCell.EntireColumn.Delete

'Printeo de valores

ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 2) = diftime
ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 2).Borders.LineStyle =
xlContinuous
ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 2).Borders.Weight = xlThick
ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 2).HorizontalAlignment =
xlCenter

ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 3) = landpad_error
ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 3).Borders.LineStyle =
xlContinuous
ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 3).Borders.Weight = xlThick
ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 3).HorizontalAlignment =
xlCenter

ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 4) = alt_max
ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 4).Borders.LineStyle =
xlContinuous
ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 4).Borders.Weight = xlThick
ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 4).HorizontalAlignment =
xlCenter

ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 5) = alt_min
ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 5).Borders.LineStyle =
xlContinuous
ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 5).Borders.Weight = xlThick
ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 5).HorizontalAlignment =
xlCenter

ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 6) = velz_max
ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 6).Borders.LineStyle =
xlContinuous
ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 6).Borders.Weight = xlThick
ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 6).HorizontalAlignment =
xlCenter

ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 7) = velz_min
ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 7).Borders.LineStyle =
xlContinuous
ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 7).Borders.Weight = xlThick
ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 7).HorizontalAlignment =
xlCenter

ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 8) = vxy_max
ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 8).Borders.LineStyle =
xlContinuous

```

```
ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 8).Borders.Weight = xlThick
ThisWorkbook.Worksheets(Sheets.Count).Cells(24, 8).HorizontalAlignment =
xlCenter
```

ANEXO M.4 Código del tercer módulo: Identificador de maniobras de vuelo – Tablas para la identificación de maniobras – Inclusión de resultados adicionales en forma de tabla.

'Empezamos el código que identificará y clasificará cada performance

'----- Quinta versión

'Variables de printeo

print_fila = 3

'Límites y factores de corrección

rad_lim = 1.2 'este radio nos va a servir para marcar el límite entre avanzar en XY o no avanzar

ang_lim = 4.5 'el máximo de grados para apreciar un cambio de dirección ya sea por un movimiento yaw o un giro

comp = 2 'factor de corrección de 2 unidades para los estados de comprobación/confirmación de cualquier movimiento

'Variables binarias para saber en qué estado nos encontramos y qué hay que ir a analizar

alt_mov = 0 '1 si hay movimiento en altura y 0 si no hay movimiento en altura

ang_mov = 0

steady_mov = 0

up = 0

down = 0

steady_ang = 0

avance = 0 'para la etapa de despegue y aterrizaje

steady_alt = 0 '1 si confirmamos que no hay variación de altitud y se lleva a cabo una operación en la misma altitud continua

'1 si se confirma que hay movimiento en el plano XY, 0 si no hay movimiento en XY

i = groundi_samp 'condicions inicials, aqui li diem on ha de començar a fer l'anàlisi (en l'enlairament)

' saldremos de este while cuando hayamos analizado la ultima performance identificable

Sheets(current_messages(1, 1)).Select

'PRIMER CASO: toda la fase de despegue

If i = groundi_samp Then 'PRIMER CASO DE TODOS; sabemos que se trata de un ascenso (despegue), por tanto solo miraremos el desplazamiento y el heading dentro de esa duración

curx = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(groundi_samp, current_messages(1, 6))

ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(groundi_samp, current_messages(1, 6)).Select

cury = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(groundi_samp, current_messages(1, 7))

ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(groundi_samp, current_messages(1, 7)).Select

xyin = Sqr(curx * curx + cury * cury) 'esto es la posicion inicial de referencia en XY para el despegue

For k = i To tkoff_samp 'recorremos todos los puntos del despegue para mirar cómo ha sido su movimiento en el plano XY

curx = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(k, current_messages(1, 6))

ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(k, current_messages(1, 6)).Select

cury = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(k, current_messages(1, 7))

```

        ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(k,
current_messages(1, 7)).Select

        x = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(k + 1,
current_messages(1, 6))

        ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(k + 1,
current_messages(1, 6)).Select

        y = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(k + 1,
current_messages(1, 7))

        ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(k + 1,
current_messages(1, 7)).Select

        rad1 = Sqr(x * x + y * y)

        rad0 = Sqr(curx * curx + cury * cury)

        difrad = Abs(rad1 - xyin) 'comparamos el valor de posicion de
referencia con la posición radial del instante k+1

        difradav1 = Abs(rad1 - rad0)

        If difrad > radio_lim And avance = 0 Then 'm de radio (si esto se
produce implica que cambiamos de estado de hover a movimiento en el plano XY,
pero tambien puede tratarse de un punto singular; lo comprobamos)

            k_comp = k + 1 '<----- hay que cambiar las
variables "a" ya que al final las hemos utilizado en el for principal

            relxyin = difrad 'en caso de que si qué se vulneren los
"comp" puntos en XY, tenemos un sumador para saber qué distancia hemos
recorrido

            For j = k_comp To k_comp + comp 'vamos a comprobar que los
"comp" siguientes puntos (2 siguientes puntos en este caso) sigan superando el
limite radial (ergo el dron se estaria moviendo el plano XY si cumple los 2
puntos)

                curxx = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(j, current_messages(1, 6))

                curyy = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(j, current_messages(1, 7))

                ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(j,
current_messages(1, 6)).Select

                ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(j,
current_messages(1, 7)).Select

                xx = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(j + 1, current_messages(1, 6))

```

```

        ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(j +
1, current_messages(1, 6)).Select

        yy = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(j + 1, current_messages(1, 7))

        ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(j +
1, current_messages(1, 7)).Select

        rad00 = Sqr(curxx * curxx + curyy * curyy)
        rad11 = Sqr(xx * xx + yy * yy)
        difrad_comp = Abs(rad11 - xyin)
        difrad_loc = Abs(rad11 - rad00)
        relxyin = relxyin + difrad_loc

        If difrad_comp < radio_lim Then

            i = j
            relxyin = 0
            Exit For

        ElseIf difrad_comp > radio_lim And j = k_comp + comp Then

            k = k_comp + comp
            avance = 1

        End If

    Next j

    ElseIf avance = 1 Then

        relxyin = relxyin + difradav1

    End If

Next k

i = k

End If

```

'Preparamos los parametros para los prints de la performance de despegue

```

'Prints del heading

    headingi = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(groundi_samp, current_messages(1, 8)) 'tenemos que meter el heading
en current messages

    headingf = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(tkoff_samp, current_messages(1, 8)) 'tenemos que meter el heading
en current messages

    dif_heading = headingf - headingi

    dif_heading = Round(dif_heading, 2)

    ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 16) =
headingi & " --> " & headingf & " (" & dif_heading & ")"

    ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
16).Interior.ColorIndex = 4

    ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
16).Borders.LineStyle = xlContinuous

    ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
16).Borders.Weight = xlThick

'Prints del tiempo

    t0 = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(groundi_samp, current_messages(1, 2))

    t1 = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(tkoff_samp, current_messages(1, 2))

    dif_t = t1 - t0

    dif_t = Round(dif_t, 3)

    ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 11) = t0 & "
- " & t1 & " (" & dif_t & ")"

    ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
11).Interior.ColorIndex = 4

    ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
11).Borders.LineStyle = xlContinuous

    ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
11).Borders.Weight = xlThick

'Diferencia altitud

```

```

        h0 = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(groundi_samp, current_messages(1, 5))

        h1 = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(tkoff_samp, current_messages(1, 5))

        dif_h = Abs(h1 - h0)
        dif_h = Round(dif_h, 2)

        relxyin = Round(relxyin, 2)

        If avance = 1 Then

            ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 14) = h0
& " - " & h1 & " (" & dif_h & ")"

            ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,
15).HorizontalAlignment = xlCenter

            ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
15).Interior.ColorIndex = 4

            ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
15).Borders.LineStyle = xlContinuous

            ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
15).Borders.Weight = xlThick

            ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,
14).HorizontalAlignment = xlCenter

            ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
14).Interior.ColorIndex = 4

            ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
14).Borders.LineStyle = xlContinuous

            ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
14).Borders.Weight = xlThick

            ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 15) =
relxyin

            ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 12) =
"Ascent motion (Take Off)"

```

```

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
12).Interior.ColorIndex = 4

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
12).Borders.LineStyle = xlContinuous

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
12).Borders.Weight = xlThick

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 13) =
"Forward motion"

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
13).Interior.ColorIndex = 4

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
13).Borders.LineStyle = xlContinuous

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
13).Borders.Weight = xlThick

    ElseIf avance = 0 Then

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 14) = h0
& " - " & h1 & " (" & dif_h & ")"

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
14).Interior.ColorIndex = 4

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
14).Borders.LineStyle = xlContinuous

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
14).Borders.Weight = xlThick

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 15) = " -
"

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
15).Interior.ColorIndex = 4

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
15).Borders.LineStyle = xlContinuous

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
15).Borders.Weight = xlThick

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,
15).HorizontalAlignment = xlCenter

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 12) =
"Straight ascent motion (Take Off)"

```



```

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
12).Interior.ColorIndex = 4

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
12).Borders.LineStyle = xlContinuous

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
12).Borders.Weight = xlThick

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 13) =
"Steady motion"

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
13).Interior.ColorIndex = 4

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
13).Borders.LineStyle = xlContinuous

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
13).Borders.Weight = xlThick

    End If

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
17).Interior.ColorIndex = 4

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
17).Borders.LineStyle = xlContinuous

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
17).Borders.Weight = xlThick

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
18).Interior.ColorIndex = 4

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
18).Borders.LineStyle = xlContinuous

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(3,
18).Borders.Weight = xlThick

c15 = ThisWorkbook.Worksheets(Sheets.Count).Cells(3, 15)
If c15 <> " - " Then
    var1 = relxyin / dif_t
    var1 = Round(var1, 2)

    ThisWorkbook.Worksheets(Sheets.Count).Cells(3, 18) = var1

```

```

        ThisWorkbook.Worksheets(Sheets.Count).Cells(3,
18).HorizontalAlignment = xlCenter

    Else

        ThisWorkbook.Worksheets(Sheets.Count).Cells(3, 18) = " - "

        ThisWorkbook.Worksheets(Sheets.Count).Cells(3,
18).HorizontalAlignment = xlCenter

    End If

var1 = dif_h / dif_t
var1 = Round(var1, 2)
ThisWorkbook.Worksheets(Sheets.Count).Cells(3, 17) = var1
ThisWorkbook.Worksheets(Sheets.Count).Cells(3,
17).HorizontalAlignment = xlCenter

avance = 0
print_fila = print_fila + 1

```

'SEGUNDO CASO: toda la fase de crucero (incluidos climb y approximation)

```
If i >= tkoff_samp And i <= land_samp Then
```

```
    i_ini = i
```

'con esto miramos punto por punto si se supera o no el limite en altitud

```

For i = i_ini + 1 To land_samp

    alt_ini = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(i
- 1, current_messages(1, 5))

    x = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(i,
current_messages(1, 5))

    dif_alt_abs = Abs(alt_ini - x) 'para determinar si cambiamos de
altitud (pasando el limite de 1 metro)

    dif_alt = x - alt_ini 'para determinar subida o bajada

'- - -'SI HAY CAMBIOS DE ALTITUD (SIN CONFIRMAR)...

    If dif_alt_abs > 1 Then ' si la variación de altura entre la
altura referencia (despues de una operación como el despegue) es de más de
1m, miramos los 2 siguientes puntos a ver si se sigue cumpliendo

        'COMPROBAMOS (recorriendo el rango de los 3 siguientes
puntos)

        For k = i To i + 3

            x_comp = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(k, current_messages(1, 5))

            dif_alt_abs_comp = Abs(alt_ini - x_comp)

            If i >= land_samp Then

                xy_mov = 1

                i = k

                alt_mov = 1

                Exit For

            End If

            'FALSO POSITIVO

            If dif_alt_abs_comp < 1 Then

                i = k

                alt_mov = 1

                Exit For

```

```

'CONFIRMAMOS UN CAMBIO DE ALTITUD
ElseIf dif_alt_abs_comp > 1 And k = i + 3 Then
    If dif_alt > 0 Then
        up = 1
        i = k
        alt_mov = 1
        steady_alt = 0
        xy_mov = 1
    ElseIf dif_alt < 0 Then
        down = 1
        i = k
        alt_mov = 1
        steady_alt = 0
        xy_mov = 1
    End If
End If
Next k

End If

'- - -'SI NO HAY CAMBIOS EN ALTITUD (SIN CONFIRMAR)
If dif_alt_abs < 1 And alt_mov = 0 Then ' condició hover

    'COMPROBAMOS (recorriendo el rango de los 3 siguientes
puntos)

    For k = i To i + 3

        x_comp = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(k, current_messages(1, 5))

        dif_alt_abs_comp = Abs(alt_ini - x_comp)

```

```

If i >= land_samp Then
    xy_mov = 1
    steady_alt = 1
    i = k
    Exit For
End If

'FALSO POSITIVO
If dif_alt_abs_comp > 1 Then
    i = k
    steady_alt = 1
    Exit For

'CONFIRMAMOS QUE NO HAY UN CAMBIO DE ALTITUD
ElseIf dif_alt_abs_comp < 1 And k = i + 3 Then
    steady_alt = 1
    alt_mov = 0
    up = 0
    down = 0
    i = k
    xy_mov = 1
End If

Next k

End If

Do While alt_mov = 1 'ASCENSO CONTINUO
    x = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(i,
current_messages(1, 5))
    y = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(i +
1, current_messages(1, 5))
    dif_alt_abs = Abs(y - x)

    If dif_alt_abs < 1 Then 'TERMINA EL ASCENSO

```

```

        alt_mov = 2
        i_fin = i 'en principio sumamos 2 como factor de
corrección

        xy_mov = 1

    End If

    If i >= land_samp Then
        steady_alt = 2
        i_fin = i 'en principio sumamos 2 como factor de
corrección

        xy_mov = 1
        Exit Do
    End If

    If alt_mov = 1 Then
        i = i + 1
    End If

Loop

    Do While steady_alt = 1 'STEADY ALTITUDE CONTINUO
        x = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(i,
current_messages(1, 5))
        y = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(i +
1, current_messages(1, 5))
        dif_alt_abs = Abs(y - x)

        If dif_alt_abs > 1 Then 'TERMINA EL STEADY ALTITUDE
            steady_alt = 2
            i_fin = i 'en principio sumamos 2 como factor de
corrección

            xy_mov = 1

```

```

End If

If i >= land_samp Then
    steady_alt = 2
    i_fin = i 'en principio sumamos 2 como factor de
corrección

    xy_mov = 1
    Exit Do
End If

If steady_alt = 1 Then
    i = i + 1

End If

Loop

'---

'tendremos que poner un while ya que no sabemos cuantas acciones
se producen en toda la fase crucero (es una opcion, alomejor no hace falta)

'CAMBIOS DE HEADING

'ahora que ya tenemos el inicio y final del ascenso o movimiento
en hover, vamos a analizar los cambios del heading

If xy_mov = 1 Then

'primero miramos el HEADING y luego miraremos el movimiento en XY
p_ini = i_ini

```

```

        For p = p_ini + 1 To i_fin ' i_fin es el ultimo punto el primer
filtro de cambio de altura

        x = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(p,
current_messages(1, 8))

        y = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(p + 1,
current_messages(1, 8))

        dif_ang_abs = Abs(y - x) 'para determinar si cambiamos de altitud
(pasando el limite de 1 metro)

'- - -'SI HAY CAMBIOS DE HEADING (SIN CONFIRMAR)...

If dif_ang_abs > ang_lim Then '

        ang_ini = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(p - 1, current_messages(1, 8))

        'COMPROBAMOS (recorriendo el rango de los 3 siguientes
puntos)

        For q = p To p + 3

        If q >= i_fin Then

                p_fin = i_fin

                ang_mov = 1

                Exit For

        End If

        x_comp = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(q, current_messages(1, 8))

        dif_ang_abs_comp = Abs(ang_ini - x_comp)

        'FALSO POSITIVO

        If dif_ang_abs_comp < ang_lim Then

                p = q

                p_fin = p

                ang_mov = 1

                Exit For

        'CONFIRMAMOS UN CAMBIO DE ANGULO

```



```

ElseIf dif_ang_abs_comp > ang_lim And q = p + 3 Then
    p = q
    ang_mov = 1

End If

Next q

End If

'- - -'SI NO HAY CAMBIOS ANGULARES (SIN CONFIRMAR)
If dif_ang_abs < ang_lim And ang_mov = 0 Then ' condició hover

    'COMPROBAMOS (recorriendo el rango de los 3 siguientes
puntos)

    ang_ini = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(p, current_messages(1, 8))

    For q = p + 1 To p + 3
        If q >= i_fin Then
            p_fin = i_fin
            steady_ang = 1
            Exit For
        End If

        x_comp = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(q, current_messages(1, 8))

        dif_ang_abs_comp = Abs(ang_ini - x_comp)

        'FALSO POSITIVO

        If dif_ang_abs_comp > ang_lim Then

            p = q

```

```

        p_fin = p
        steady_ang = 1
        Exit For

'CONFIRMAMOS QUE NO HAY UN CAMBIO DE ANGULO
ElseIf dif_ang_abs_comp < ang_lim And q = p + 3 Then
        steady_ang = 1
        p = q
    End If

Next q
End If

Do While ang_mov = 1 'CAMBIO DE ANGULO CONTINUO
    If p_fin >= i_fin Then
        Exit Do
    End If

    If p >= i_fin Then
        p_fin = p
        Exit Do
    End If

    x = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(p,
current_messages(1, 8))

    y = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(p +
1, current_messages(1, 8))

    dif_ang_abs = Abs(y - x)

    If dif_alt_abs < ang_lim Then 'TERMINA EL CAMBIO DE ANGULO
        ang_mov = 2
        p_fin = p 'no añadimos factores de corrección
        ang_fin = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(p_fin, current_messages(1, 8))

```

```

End If

If ang_mov = 1 Then
    p = p + 1
End If

Loop

Do While steady_ang = 1 'STEADY ANGULO CONTINUO
    If p_fin >= i_fin Then
        Exit Do
    End If
    If p >= i_fin Then
        p_fin = p
        Exit Do
    End If

    x = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(p,
current_messages(1, 8))
    y = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(p +
1, current_messages(1, 8))
    dif_ang_abs = Abs(y - x)

    If dif_ang_abs >= ang_lim Then 'TERMINA EL STEADY angle
        steady_ang = 2
        p_fin = p
        ang_fin = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(p_fin, current_messages(1, 8))

End If

```

```
If steady_ang = 1 Then
```

```
    p = p + 1
```

```
End If
```

```
Loop
```

'en principio aqui termina la parte del heading, pero ahora, con las mismas variables recorreremos los puntos de p_ini a p_fin viendo el recorrido en el plano XY

```
'-----  
- - - - - EMPEZAMOS LA PARTE XY
```

'JUSTO ACABAMOS DE ENCONTRAR LOS MOVIMIENTOS DE HEADING ENTRE LOS PUNTOS P_INI Y P_FINAL

'CASO EN QUE NO HAY MOV ANGULAR (enfocando al mismo sitio en todo el periodo "p_ini-p_fin")

```
If steady_ang = 1 Or steady_ang = 2 Then
```

'MOVIMIENTO EN XY MIRAMOS PUNTOS XY QUE COMPRENDEN DESDE EL HEADING P_INI HASTA P_FINAL

```
xy_loc = 0
```

```
For k = p_ini To p_fin 'aqui podría sumarse "+1" en "p_ini"
```

```
    x1 = ThisWorkbook.Worksheets(current_messages(1,  
1)).Cells(k, current_messages(1, 6))
```

```
    y1 = ThisWorkbook.Worksheets(current_messages(1,  
1)).Cells(k, current_messages(1, 7))
```

```
    x2 = ThisWorkbook.Worksheets(current_messages(1,  
1)).Cells(k + 1, current_messages(1, 6))
```

```
    y2 = ThisWorkbook.Worksheets(current_messages(1,  
1)).Cells(k + 1, current_messages(1, 7))
```

```

rad1 = Sqr(x1 * x1 + y1 * y1)
rad2 = Sqr(x2 * x2 + y2 * y2)
difrad_abs = Abs(rad1 - rad2) 'comparamos el valor de
posicion de referencia con la posición radial del instante k+1

'HAY AVANCE (SIN COMPROBAR)
If difrad_abs > rad_lim Then 'm de rad (si esto se
produce implica que cambiamos de estado de hover a movimiento en el plano XY,
pero tambien puede tratarse de un punto singular; lo comprobamos)
    xy_loc = xy_loc + difrad_abs 'en caso de que si qué
se vulneren los "comp" puntos en XY, tenemos un sumador para saber qué
distancia hemos recorrido
    av_mov = 1

'NO HAY AVANCE (SIN COMPROBAR)
ElseIf difrad_abs < rad_lim And av_mov = 0 Then
    xy_loc = 0
    steady_mov = 1

    'en caso de que si qué se vulneren los "comp" puntos
en XY, tenemos un sumador para saber qué distancia hemos recorrido
End If

Next k
End If

'CASO EN EL QUE HA HABIDO UN CAMBIO ANGULAR CONTINUO
If ang_mov = 1 Or ang_mov = 2 Then

'MOVIMIENTO EN XY MIRAMOS PUNTOS QUE COMPRENDEN EL HEADING
P_INI,P_FINAL

```

```

xy_loc = 0

For k = p_ini To p_fin

    x1 = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(k, current_messages(1, 6))

    y1 = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(k, current_messages(1, 7))

    x2 = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(k + 1, current_messages(1, 6))

    y2 = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(k + 1, current_messages(1, 7))

    rad1 = Sqr(x1 * x1 + y1 * y1)
    rad2 = Sqr(x2 * x2 + y2 * y2)

    difrad_abs = Abs(rad1 - rad2) 'comparamos el valor de
posicion de referencia con la posición radial del instante k+1

    'HAY AVANCE (SIN COMPROBAR)

    If difrad_abs > rad_lim Then 'm de rad (si esto se
produce implica que cambiamos de estado de hover a movimiento en el plano XY,
pero tambien puede tratarse de un punto singular; lo comprobamos)

        xy_loc = xy_loc + difrad_abs 'en caso de que si qué
se vulneren los "comp" puntos en XY, tenemos un sumador para saber qué
distancia hemos recorrido

        av_mov = 1

    'NO HAY AVANCE (SIN COMPROBAR)

    ElseIf difrad_abs <= rad_lim And av_mov = 0 Then

        xy_loc = 0

        steady_mov = 1

        'en caso de que si qué se vulneren los "comp" puntos
en XY, tenemos un sumador para saber qué distancia hemos recorrido

    End If

```

```
Next k
```

```
End If
```

```
'PRINTEAMOS LOS VALORES DE LA ÚLTIMA PERFORMANCE
```

```
'Preparamos los parametros para los prints de la performance  
de despegue
```

```
'Prints del heading
```

```
headingi = ThisWorkbook.Worksheets(current_messages(1,  
1)).Cells(p_ini, current_messages(1, 8)) 'tenemos que meter el heading en  
current messages
```

```
headingf = ThisWorkbook.Worksheets(current_messages(1,  
1)).Cells(p_fin, current_messages(1, 8)) 'tenemos que meter el heading en  
current messages
```

```
dif_heading = headingf - headingi
```

```
dif_heading = Round(dif_heading, 2)
```

```
ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 16) =  
headingi & " --> " & headingf & " (" & dif_heading & ")"
```

```
'Prints del tiempo
```

```
t0 = ThisWorkbook.Worksheets(current_messages(1,  
1)).Cells(p_ini, current_messages(1, 2))
```

```
t1 = ThisWorkbook.Worksheets(current_messages(1,  
1)).Cells(p_fin, current_messages(1, 2))
```

```
dif_t = t1 - t0
```

```
dif_t = Round(dif_t, 3)
```

```
ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 11) =  
t0 & " - " & t1 & " (" & dif_t & ")"
```

```
'Diferencia altitud
```

```
h0 = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(p_ini, current_messages(1, 5))
```

```
h1 = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(p_fin, current_messages(1, 5))
```

```
dif_h = Abs(h1 - h0)
```

```
dif_h = Round(dif_h, 2)
```

```
'Performance en vertical (Z)
```

```
    If alt_mov = 1 Or alt_mov = 2 Then
```

```
        If up = 1 And av_mov = 1 Then
```

```
ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 14) = h0 & " - " & h1  
& " (" & dif_h & ")"
```

```
ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,  
14).HorizontalAlignment = xlCenter
```

```
ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 12) = "Ascent motion"
```

```
    ElseIf down = 1 And av_mov = 1 Then
```

```
        dif_h = dif_h * (-1)
```

```
ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 14) = h0 & " - " & h1  
& " (" & dif_h & ")"
```

```
ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,  
14).HorizontalAlignment = xlCenter
```

```
ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 12) = "Descent  
motion"
```

```
    ElseIf up = 1 And steady_mov = 1 Then
```

```
ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 14) = h0 & " - " & h1  
& " (" & dif_h & ")"
```

```
ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,  
14).HorizontalAlignment = xlCenter
```



```
ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 12) = "Straight  
ascent motion"
```

```
ElseIf down = 1 And steady_mov = 1 Then
```

```
    dif_h = dif_h * (-1)
```

```
ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 14) = h0 & " - " & h1  
& " (" & dif_h & ")"
```

```
ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,  
14).HorizontalAlignment = xlCenter
```

```
ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 12) = "Straight  
descent motion"
```

```
End If
```

```
ElseIf steady_alt = 1 Or steady_alt = 2 Then
```

```
    ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,  
14) = " - "
```

```
    ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,  
14).HorizontalAlignment = xlCenter
```

```
    ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,  
12) = "Steady motion " & "(" & h0 & ")"
```

```
End If
```

```
'Performance en horizontal (XY)
```

```
If ang_mov = 1 Or ang_mov = 2 Then
```

```
    If av_mov = 1 Then
```

```
        xy_loc = Round(xy_loc, 2)
```

```
ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 15) = xy_loc
```

```

ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,
15).HorizontalAlignment = xlCenter

        If dif_heading > 0 Then

ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 13) = "Turn
(Clockwise)"

        ElseIf dif_heading < 0 Then

ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 13) = "Turn (Counter-
clockwise)"

        End If

        ElseIf steady_mov = 1 Then

ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 15) = " - "

ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,
15).HorizontalAlignment = xlCenter

ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 13) = "Yaw motion"

        End If

        ElseIf steady_ang = 1 Or steady_ang = 2 Then

        If av_mov = 1 Then

                xy_loc = Round(xy_loc, 2)

ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 15) = xy_loc

ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,
15).HorizontalAlignment = xlCenter

ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 13) = "Forward
motion"

        ElseIf steady_mov = 1 Then

```

```

ThisWorkbook.Worksheets(Sheets.Count).Cells(print_filas, 15) = " - "

ThisWorkbook.Worksheets(Sheets.Count).Cells(print_filas,
15).HorizontalAlignment = xlCenter

ThisWorkbook.Worksheets(Sheets.Count).Cells(print_filas, 13) = "Steady motion"
        End If
    End If

    'Valores de mean velocity
    c14 = ThisWorkbook.Worksheets(Sheets.Count).Cells(print_filas,
14) 'Z
    c15 = ThisWorkbook.Worksheets(Sheets.Count).Cells(print_filas,
15) 'XY

    If dif_t = 0 Then

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_filas,
1).EntireRow.Delete
        print_filas = print_filas - 1
        dif_t = 1
    End If

    'Ponemos los valores de "mean speed" en XY y Z
    If c14 <> " - " And dif_t = 0 Then
        var2 = dif_h / dif_t
        var2 = Round(var2, 2)

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_filas,
17) = var2

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_filas,
17).HorizontalAlignment = xlCenter
    Else

```

```

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,
17) = " - "

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,
17).HorizontalAlignment = xlCenter

    End If

    If c15 <> " - " And dift = 0 Then
        var2 = xy_loc / dif_t
        var2 = Round(var2, 2)

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,
18) = var2

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,
18).HorizontalAlignment = xlCenter

    Else

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,
18) = " - "

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,
18).HorizontalAlignment = xlCenter

    End If

    ' hemos borrado alt_mov = 0 y lo vamos a poner en los fors de
    "i" al inicio de la segunda fase. Los pondremos acorde se active o desactive

    ang_mov = 0
    steady_mov = 0
    'up = 0
    av_mov = 0
    'down = 0
    steady_ang = 0
    dift = 0

    '1 si confirmamos que no hay variacion de altitude y se lleva
    a cabo una operacion en la misma altitud continua

    xy_mov = 0
    p_ini = p_fin
    print_fila = print_fila + 1

```

'Quan arribem a un descens que implica varies maniobres, no sabem per que no marca aquestes maniobres amb el descens de 451 a 488 es on identifiquem aquest descens

```
'-----  
- - - - - ACABAMOS LA PARTE XY
```

```
'---- reiniciamos variables
```

```
Next p
```

```
End If
```

```
i_ini = i
```

```
alt_mov = 0
```

```
Next i 'corresponde al gran FOR de la fase 2, después del gran IF
```

```
End If 'corresponde al gran IF de la fase 2
```

```
'-----
```

```
'-----
```

```

'TERCER CASO: toda la fase de aterrizaje

If i >= land_samp Then

    curx = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(land_samp, current_messages(1, 6))

    ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(land_samp,
current_messages(1, 6)).Select

    cury = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(land_samp, current_messages(1, 7))

    ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(land_samp,
current_messages(1, 7)).Select

    xyin = Sqr(curx * curx + cury * cury) 'esto es la posicion inicial de
referencia en XY para el despegue

    For k = i To groundf_samp 'recorremos todos los puntos del despegue
para mirar cómo ha sido su movimiento en el plano XY

        curx = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(k,
current_messages(1, 6))

        ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(k,
current_messages(1, 6)).Select

        cury = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(k,
current_messages(1, 7))

        ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(k,
current_messages(1, 7)).Select

        x = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(k + 1,
current_messages(1, 6))

        ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(k + 1,
current_messages(1, 6)).Select

        y = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(k + 1,
current_messages(1, 7))

        ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(k + 1,
current_messages(1, 7)).Select

        rad1 = Sqr(x * x + y * y)

        rad0 = Sqr(curx * curx + cury * cury)

        difrad = Abs(rad1 - xyin) 'comparamos el valor de posicion de
referencia con la posición radial del instante k+1

```

```
difradav1 = Abs(rad1 - rad0)
```

If difrad > radio_lim And avance = 0 Then 'm de radio (si esto se produce implica que cambiamos de estado de hover a movimiento en el plano XY, pero tambien puede tratarse de un punto singular; lo comprobamos)

```
k_comp = k + 1 '<----- hay que cambiar las variables "a" ya que al final las hemos utilizado en el for principal
```

```
relxyin = difrad 'en caso de que si qué se vulneren los "comp" puntos en XY, tenemos un sumador para saber qué distancia hemos recorrido
```

For j = k_comp To k_comp + comp 'vamos a comprobar que los "comp" siguientes puntos (2 siguientes puntos en este caso) sigan superando el limite radial (ergo el dron se estaria moviendo el plano XY si cumple los 2 puntos)

```
curxx = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(j, current_messages(1, 6))
```

```
curyy = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(j, current_messages(1, 7))
```

```
ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(j, current_messages(1, 6)).Select
```

```
ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(j, current_messages(1, 7)).Select
```

```
xx = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(j + 1, current_messages(1, 6))
```

```
ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(j + 1, current_messages(1, 6)).Select
```

```
yy = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(j + 1, current_messages(1, 7))
```

```
ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(j + 1, current_messages(1, 7)).Select
```

```
rad00 = Sqr(curxx * curxx + curyy * curyy)
```

```
rad11 = Sqr(xx * xx + yy * yy)
```

```
difrad_comp = Abs(rad11 - xyin)
```

```
difrad_loc = Abs(rad11 - rad00)
```

```
relxyin = relxyin + difrad_loc
```

```
If difrad_comp < radio_lim Then
```

```

        i = j
        relxyin = 0
        Exit For
    ElseIf difrad_comp > radio_lim And j = k_comp + comp Then
        k = k_comp + comp
        avance = 1
    End If
Next j

ElseIf avance = 1 Then
    relxyin = relxyin + difradav1
End If

Next k

i = k
End If

'Preparamos los parametros para los prints de la performance de
despegue

'Prints del heading

headingi = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(land_samp, current_messages(1, 8)) 'tenemos que meter el heading en
current messages

headingf = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(groundf_samp, current_messages(1, 8)) 'tenemos que meter el
heading en current messages

dif_heading = headingf - headingi

dif_heading = Round(dif_heading, 2)

ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 16) =
headingi & " --> " & headingf & " (" & dif_heading & ")"

ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
16).Interior.ColorIndex = 6

ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
16).Borders.LineStyle = xlContinuous

```



```

ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
16).Borders.Weight = xlThick

'Prints del tiempo

t0 = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(land_samp,
current_messages(1, 2))

t1 = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(groundf_samp, current_messages(1, 2))

dif_t = t1 - t0

dif_t = Round(dif_t, 3)

ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 11) = t0 & "
- " & t1 & " (" & dif_t & ")"

ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
11).Interior.ColorIndex = 6

ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
11).Borders.LineStyle = xlContinuous

ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
11).Borders.Weight = xlThick

'Diferencia altitud

h0 = ThisWorkbook.Worksheets(current_messages(1, 1)).Cells(land_samp,
current_messages(1, 5))

h1 = ThisWorkbook.Worksheets(current_messages(1,
1)).Cells(groundf_samp, current_messages(1, 5))

dif_h = Abs(h1 - h0)

dif_h = Round(dif_h, 2)

relxyin = Round(relxyin, 2)

If avance = 1 Then

ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 14) = h0
& " - " & h1 & " (" & dif_h & ")"

ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
14).Interior.ColorIndex = 6

```

```

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
14).Borders.LineStyle = xlContinuous

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
14).Borders.Weight = xlThick

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 15) =
relxyin

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,
15).HorizontalAlignment = xlCenter

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
15).Interior.ColorIndex = 6

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
15).Borders.LineStyle = xlContinuous

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
15).Borders.Weight = xlThick

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 12) =
"Descent motion (Landing)"

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
12).Interior.ColorIndex = 6

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
12).Borders.LineStyle = xlContinuous

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
12).Borders.Weight = xlThick

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 13) =
"Forward motion"

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
13).Interior.ColorIndex = 6

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
13).Borders.LineStyle = xlContinuous

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
13).Borders.Weight = xlThick

    ElseIf avance = 0 Then

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 14) = h0
& " - " & h1 & " (" & dif_h & ")"

```

```

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
14).Interior.ColorIndex = 6

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
14).Borders.LineStyle = xlContinuous

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
14).Borders.Weight = xlThick

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 15) = " -
"

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,
15).HorizontalAlignment = xlCenter

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
15).Interior.ColorIndex = 6

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
15).Borders.LineStyle = xlContinuous

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
15).Borders.Weight = xlThick

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 12) =
"Steady descent motion (Landing)"

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
12).Interior.ColorIndex = 6

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
12).Borders.LineStyle = xlContinuous

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
12).Borders.Weight = xlThick

        ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 13) =
"Steady motion"

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
13).Interior.ColorIndex = 6

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
13).Borders.LineStyle = xlContinuous

        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fila,
13).Borders.Weight = xlThick

        End If

```

```
        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fil  
17).Interior.ColorIndex = 6
```

```
        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fil  
17).Borders.LineStyle = xlContinuous
```

```
        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fil  
17).Borders.Weight = xlThick
```

```
        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fil  
18).Interior.ColorIndex = 6
```

```
        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fil  
18).Borders.LineStyle = xlContinuous
```

```
        ThisWorkbook.Worksheets(sheets_len + Count).Cells(print_fil  
18).Borders.Weight = xlThick
```

```
'color columnas
```

```
For a = 4 To print_fil - 1
```

```
    If (a Mod 2) = 0 Then
```

```
        For b = 11 To 18
```

```
            ThisWorkbook.Worksheets(sheets_len + Count).Cells(a,  
b).Interior.ColorIndex = 23
```

```
            ThisWorkbook.Worksheets(sheets_len + Count).Cells(a,  
b).Borders.LineStyle = xlContinuous
```

```
            ThisWorkbook.Worksheets(sheets_len + Count).Cells(a,  
b).Borders.Weight = xlThick
```

```
        Next b
```

```
    Else
```

```
        For b = 11 To 18
```

```
            ThisWorkbook.Worksheets(sheets_len + Count).Cells(a,  
b).Borders.LineStyle = xlContinuous
```

```
            ThisWorkbook.Worksheets(sheets_len + Count).Cells(a,  
b).Borders.Weight = xlThick
```

```
        Next b
```

```
    End If
```

Next a

'Valores mean velocity en XY y Z

If c15 <> " - " Then

var3 = relxyin / dif_t

var3 = Round(var3, 2)

ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 18) =
var3

ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,
18).HorizontalAlignment = xlCenter

Else

ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 18) = " -
"

ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,
18).HorizontalAlignment = xlCenter

End If

var3 = dif_h / dif_t

var3 = Round(var3, 2)

ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila, 17) = var3

ThisWorkbook.Worksheets(Sheets.Count).Cells(print_fila,
17).HorizontalAlignment = xlCenter

ThisWorkbook.Worksheets(Sheets.Count).Columns("K:R").AutoFit

ThisWorkbook.Worksheets(Sheets.Count).Columns("B:H").AutoFit