

UNIVERSIDAD DE SEVILLA
ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA AEROESPACIAL Y MECÁNICA
DE FLUIDOS

INGENIERÍA AERONÁUTICA (PLAN 2002)

PROYECTO FIN DE CARRERA

**Análisis de trayectorias de aeronaves
obtenidas mediante ADS-B**

Autor:

Francisco CANTOS GALÁN

Tutor:

Alfonso VALENZUELA ROMERO

26 de septiembre de 2013

Índice general

1. Introducción	5
1.1. Motivación	5
1.2. Objetivo	5
1.3. Estructura del documento	6
2. Sistema ADS-B	7
2.1. Descripción	7
2.2. Transceptores	9
2.3. Aeronaves con ADS-B	10
2.4. Descripción del equipo utilizado	11
3. Procesamiento de los datos	13
3.1. Captura de datos	13
3.2. Partición de los archivos de datos	15
3.3. Lectura de los datos	16
3.4. Separación por aeronaves	17
3.5. Unión de las particiones	17
3.6. Búsqueda de modelos y aerolíneas	17
3.7. Separación de vuelos	19
3.8. Asignación de un CallSign a cada vuelo	19
3.9. Clasificación de los vuelos en cruceros, ascensos o descensos	20

3.10. Salidas y llegadas en el aeropuerto de Sevilla	20
3.11. Determinación de los tramos de ascenso/descenso	22
3.12. Cabeceras de pista	23
3.13. Calidad de la señal	24
3.14. Determinación de la distancia recorrida	25
4. Análisis de los datos	27
4.1. Perfil horizontal	27
4.1.1. Salidas	29
4.1.2. Llegadas	33
4.2. Perfil vertical	40
4.2.1. Salidas	41
4.2.1.1. Comparación entre modelos de avión	41
4.2.1.2. Distribución por destinos	46
4.2.1.3. Comparación entre modelos, para un mismo destino (Barcelona)	48
4.2.1.4. Comparación entre destinos, para un mismo modelo (Airbus A320-2)	52
4.2.2. Llegadas	56
4.2.2.1. Comparación entre modelos de avión	58
4.2.2.2. Distribución por destinos	61
4.2.2.3. Comparación entre modelos, para un mismo origen (Barcelona)	61
4.2.2.4. Comparación entre destinos, para un mismo modelo (Airbus A320-2)	65
4.3. Otros resultados	69
4.3.1. Obstáculos en la recepción de la señal	69
4.3.2. Circuitos de espera	70
4.3.3. Aterrizajes frustrados	70
4.3.4. Máximas altitudes en llegadas y salidas	71

4.3.5. Aeronaves particulares	71
4.3.5.1. Airbus A400M	71
4.3.5.2. Airbus A300-6 (Beluga)	73
4.3.6. Otros aeropuertos	73
4.3.6.1. Aeropuerto de Málaga	73
4.3.6.2. Aeropuerto de Jerez	74
5. Conclusiones	75
6. Trabajo futuro	77
Bibliografía	79
Apéndices	80
A. Código Visual Basic	83
B. Códigos de Matlab utilizados	85
B.1. gen_lectura.m	85
B.2. func_matriculas.m	87
B.3. gen_union.m	87
B.4. gen_model_AL.m	90
B.5. gen_sep.m	91
B.6. gen_CS.m	93
B.7. gen_estado.m	94
B.8. func_aeropuertos.m	96
B.9. func_estad.m	97
B.10. func_cabecera.m	99
B.11. gen_retraso.m	100
B.12. func_filtro_retraso.m	101

B.13.gen_repr.m	103
B.14.func_repr.m	106
C. Formato de las estructuras de datos	109
D. Proyección de Lambert	111

1

Introducción

1.1. Motivación

Antiguamente solo las aerolíneas y el control de tráfico aéreo podían acceder a información relativa a las trayectorias seguidas por las aeronaves. Sin embargo hoy en día, con tecnologías como el ADS-B (*Automatic Dependent Surveillance - Broadcast*), se pueden obtener una gran cantidad de datos interesantes transmitidos por las aeronaves, sin codificación alguna y que permiten analizar bastantes aspectos relacionados con los vuelos realizados en la zona de cobertura.

El procesado de estos datos y su posterior análisis darán lugar a conclusiones sobre toda la actividad aérea de la zona en cuestión.

En el presente proyecto se desea conocer cómo operan las aeronaves que llegan o salen del aeropuerto de Sevilla. De esta manera, se podrán estudiar tanto los perfiles horizontales como los verticales de las trayectorias descritas por las aeronaves identificadas, buscando las diferencias entre lo que se encuentra publicado en las normativas y cómo operan realmente las aeronaves.

1.2. Objetivo

El objetivo del presente proyecto es analizar las trayectorias de ascenso y descenso de aviones comerciales con destino u origen en el aeropuerto de Sevilla/San Pablo¹, las cuales se obtendrán mediante ADS-B. En concreto, se estudiarán:

- el cumplimiento de los perfiles horizontales con los procedimientos estándar de salida y llegada (SID y STAR, respectivamente) publicados por AENA (Aeropuertos Españoles y Navegación Aérea); y
- las variaciones en los perfiles verticales de las trayectorias, atendiendo a:

¹Código OACI (Organización de Aviación Civil Internacional) LEZL.

- ⇒ si son salidas o llegadas;
- ⇒ el modelo de avión; y
- ⇒ el origen o destino del vuelo.

1.3. Estructura del documento

El documento en el que se presenta el proyecto realizado consta de 5 capítulos principales en los que se describe, en primer lugar, cómo funciona el sistema ADS-B; a continuación, se explica la metodología seguida en lo que se refiere a la captura de datos y el procesamiento informático realizado; posteriormente aparece un capítulo para mostrar, analizar y extraer conclusiones de los resultados obtenidos y finalmente, se tienen dos capítulos dedicados a realizar unas conclusiones globales del proyecto y a definir posibles trabajos futuros para complementar el actual.

Por otro lado, se incluyen varios anexos que se consideran de interés, en los que se muestra parte de la programación (VBS (*Visual Basic Script*) y MATLAB) realizada para visualizar con más detalle todos los aspectos considerados y no detallados a lo largo del Capítulo 3, así como aspectos relacionados con la propia programación, como pueden ser la organización de los datos y resultados, y la proyección de Lambert utilizada para la representación de los perfiles horizontales.

2

Sistema ADS-B

En este capítulo se presenta una descripción del sistema utilizado, ADS-B, mediante el cual se reciben los datos enviados por las aeronaves para su posterior análisis.

En primer lugar se describirá el sistema así como los datos que transmite y los que puede llegar a transmitir. A continuación se explicarán los tipos de sistemas ADS que existen y sus principales diferencias. Posteriormente se nombrarán los tres posibles transceptores que pueden utilizarse para el uso del ADS-B, así como sus principales características y las frecuencias en las que operan. Se enumerarán algunas de las aeronaves que hoy en día utilizan ADS-B y finalmente, se describe el equipo utilizado para la realización de este proyecto, así como su localización geográfica.

2.1. Descripción

De acuerdo a OACI [1], el ADS (*Automatic Dependent Surveillance*) es una técnica de vigilancia que permite a las aeronaves proporcionar automáticamente, mediante enlace de datos, aquellos datos extraídos de sus sistemas de navegación y determinación de la posición instalados a bordo.

Como su propio nombre indica, este sistema es:

- **Automático:** Está siempre activo y no necesita intervención de ningún operador.
- **Dependiente:** La operación y su calidad dependen del comportamiento del sistema de navegación de la aeronave.
- **Vigilante:** Proporciona datos de vigilancia análogos a los del radar para los controladores de tierra y otros aviones.

La información que transmite el sistema ADS incluye la identificación de la aeronave y su posición en cuatro dimensiones. Adicionalmente, suelen aportarse más datos como la velocidad y el rumbo, los puntos de paso (*waypoints*) obtenidos durante el vuelo en el sistema de navegación

y datos meteorológicos (temperatura absoluta del aire y velocidad del viento aguas arriba). Los datos de ruta, incluidos los *waypoints* siguientes correspondientes a las alturas previstas, permiten a los controladores comprobar si se cumplen los planes de vuelo del avión, pudiendo ser controlados por el controlador responsable del vuelo los intervalos entre los que ADS transmite información .

Dentro del sistema ADS, pueden utilizarse dos subsistemas distintos: ADS-B (*ADS-Broadcast*) y ADS-C (*ADS-Contract*).

El ADS-B es una radiodifusión, dirigida a todos los posibles receptores dentro de su esfera de influencia ya que no se encuentra encriptado. De esta manera, un obstáculo geográfico puede enviar mensajes ADS-B, complementando los mapas que pueda tener el sistema de navegación.

El máximo alcance del sistema es, en línea de mira, de menos de 200 NM (370 km) si se emplea sobre VHF. Sin embargo, si se utiliza sobre un canal SatComm, su alcance es global. En zonas continentales puede usarse enlace de datos VHF o L-band directamente desde el avión, aunque en las operaciones transoceánicas estas comunicaciones se encuentran faltas de cobertura, por lo que es usual usar un satélite de comunicaciones geoestacionario como el mostrado en la Figura 2.1 como medio fiable de enlace de datos.

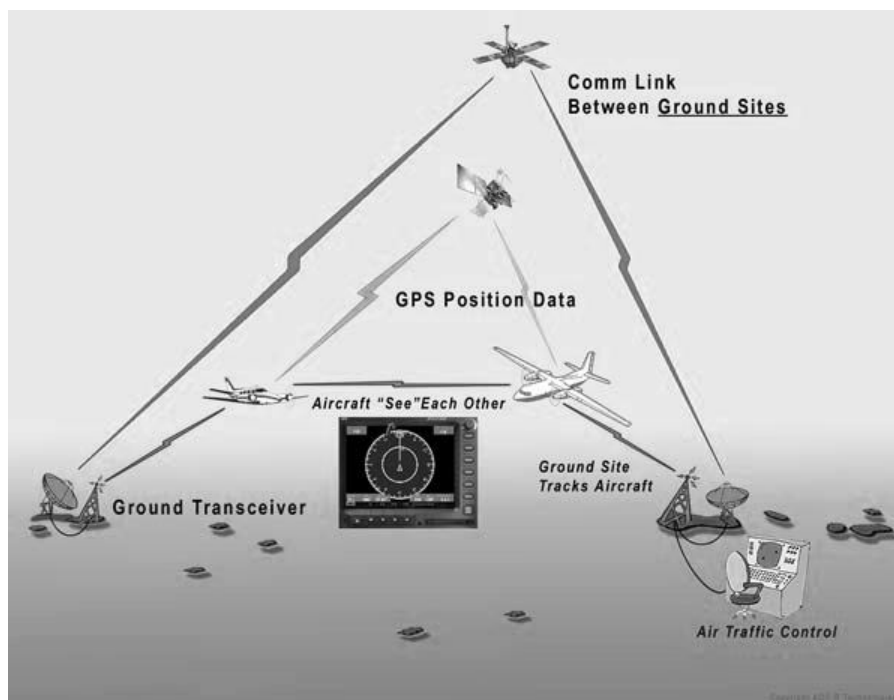


Figura 2.1: Esquema del funcionamiento del ADS-B. ¹

Una de las principales ventajas del sistema ADS-B frente a las redes de radares existentes es que la precisión de los datos no depende de la distancia a la que se encuentre la aeronave, o de la frecuencia de barrido del radar, puesto que los datos de posición se obtienen a partir de GNSS (*Global Navigation Satellite System*). El aumento de la precisión significa mayor seguridad, múltiples aplicaciones y un incremento de la capacidad del espacio aéreo y del aeropuerto. Como también trabaja a bajas alturas e incluso en el suelo, puede emplearse para monitorizar

¹<http://www.aero-news.net/images/content/genav/2005/capstone-ads-b-gfx-0405-1a.jpg> (21/09/2013).

el tráfico de tierra sobre las calles de rodaje, mejorando mucho la resolución del radar ASDE (*Airport Surface Detection Equipment*).

Por otro lado, no todos son ventajas, presentando el inconveniente de que es dependiente de la fiabilidad de los sensores del avión, necesitándose una medida de respaldo por parte del ATC (*Air Traffic Control*), ya que si el sistema de posicionamiento GNSS no posicionara correctamente a la aeronave (tanto por fallo en las comunicaciones, fallo en la interpretación de datos o por fallo en la obtención de la posición), el ATC no la situaría correctamente y no tendría manera independiente de la aeronave de intentar posicionarla.

Se encuentran dos modos de utilización del ADS-B: ADS-B Out para vigilancia desde el suelo, el cual solo requiere capacidad de la aeronave para transmitir mensajes ADS y ADS-B In para intercambio de información, que además, requiere capacidad de la aeronave para recibir mensajes ADS y procesarlos.

Por su parte, el sistema ADS-C proporciona un informe de datos de vigilancia a una o varias direcciones específicas, en lugar de a todos los receptores en su esfera de influencia, siendo la estación de tierra la que decide la frecuencia de las transmisiones y los parámetros de las mismas.

2.2. Transceptores

El sistema ADS-B puede utilizar tres tipos de transceptores:

- **Modo S extendido a un canal de 1090 MHz:** El enlace de datos se realiza a través del sistema SSR (*Secondary Surveillance Radar*), consistiendo en una extensión de este. Así, la aeronave transmite regularmente mensajes (conocidos como *extended squitters*) conteniendo información tal como la posición o la identificación. Los *extended squitters* son transmitidos en la misma frecuencia de respuesta que el SSR, 1090 MHz, y pueden ser recibidos por cualquier aeronave o estación de tierra convenientemente equipada.

Uno de los principales problemas de este método es que las transmisiones pueden confundirse con otras del modo S como la vigilancia elemental o mejorada, que operan con los mismos protocolos y formatos de mensaje en las mismas frecuencias. Para evitar esto, los *extended squitters* son transmitidos a intervalos aleatorios con ciertas tasas medias de transmisión. La transmisión de *extended squitters* no está sincronizada con las transmisiones del resto de los usuarios.

- **Universal Access Transceiver (UAT):** Está definido específicamente para el ADS-B. Se trata de un canal de 1 Mbps sobre portadora de 978 MHz (en la banda L, de 960 a 1215 MHz [2]) para transmitir informes ADS-B y para difusión de información aeronáutica desde tierra.

UAT emplea reutilización de frecuencias por reparto en células al igual que la telefonía móvil. El acceso al medio se resuelve por acceso aleatorio: elección pseudoaleatoria de uno de entre 3200 oportunidades de comienzo de mensaje (*Message Start Opportunities, MSO*) [3].

Utiliza una modalidad de TDMA (*Time Division Multiplexing Access*) convencional en la que hay una estación sincronizadora que marca los turnos de emisión y el resto de estaciones se ajustan a esos turnos (en este caso, la señal UTC del GNSS), disponiendo de 3200 *slots* temporales posibles, eligiendo uno de forma aleatoria con la esperanza de que los demás no colisionen. Si se produjera la colisión, simplemente se ignora el mensaje y se espera al siguiente *slot* [3].

En la Figura 2.2 se muestra un esquema básico del funcionamiento del TDMA.

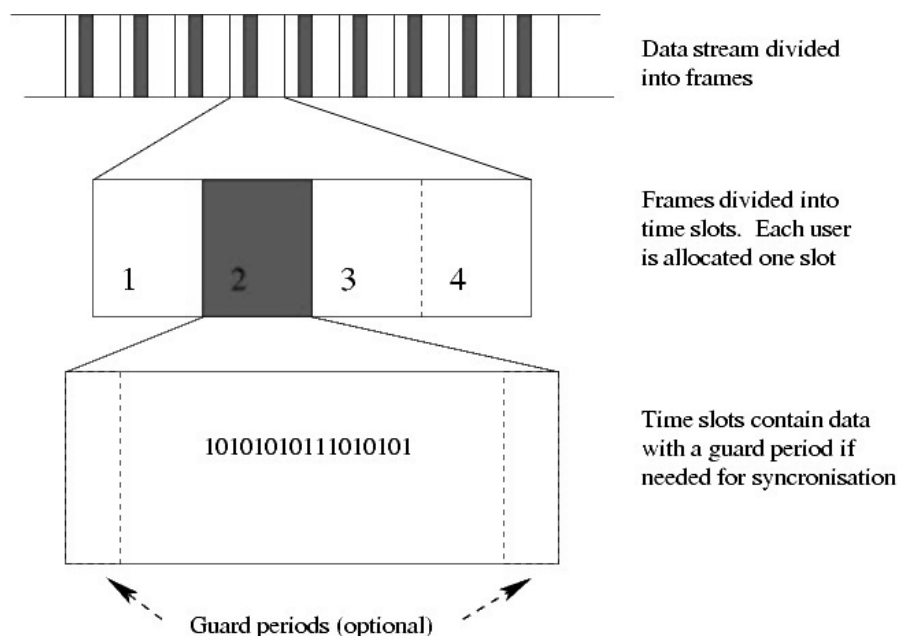


Figura 2.2: Esquema básico del funcionamiento del TDMA²

- **VDL Mode 4:** Se trata de un enlace de datos VHF estandarizado por OACI y ETSI (*European Telecommunications Standards Institute*) que provee comunicación digital entre dos estaciones móviles (aeronave y vehículos en la superficie del aeropuerto) o entre unidades móviles y estaciones de tierra.

Puede usar una o más de las frecuencias VHF de la radio actual como capa física para las transmisiones ADS-B. VDL Mode 4 emplea un protocolo STDMA (*Space Time Division Multiplexing Access*) que le permite autoorganizarse, sin necesidad de un control maestro desde tierra. El medio es óptimo para la transmisión de mensajes cortos para un gran número de usuarios.

2.3. Aeronaves con ADS-B

Ejemplos de aeronaves que por lo general tienen implementado el sistema ADS-B y por tanto pueden ser visibles, son:³

²<http://upload.wikimedia.org/wikipedia/en/f/f9/Tdma-frame-structure.png> (15/09/2013).

³<http://www.flightradar24.com/how-it-works> (15/09/2013).

- todos los modelos de Airbus: A300, A310, A318, A319, A320, A321, A330, A340, A380;
- los modelos de Boeing: 737, 747, 757, 767, 777, 787; y
- los modelos de McDonnell Douglas: MD-10, MD-11.

2.4. Descripción del equipo utilizado

Como receptor ADS-B se ha utilizado el dispositivo *SBS-3* de KINETIC, mostrado en la Figura 2.3.

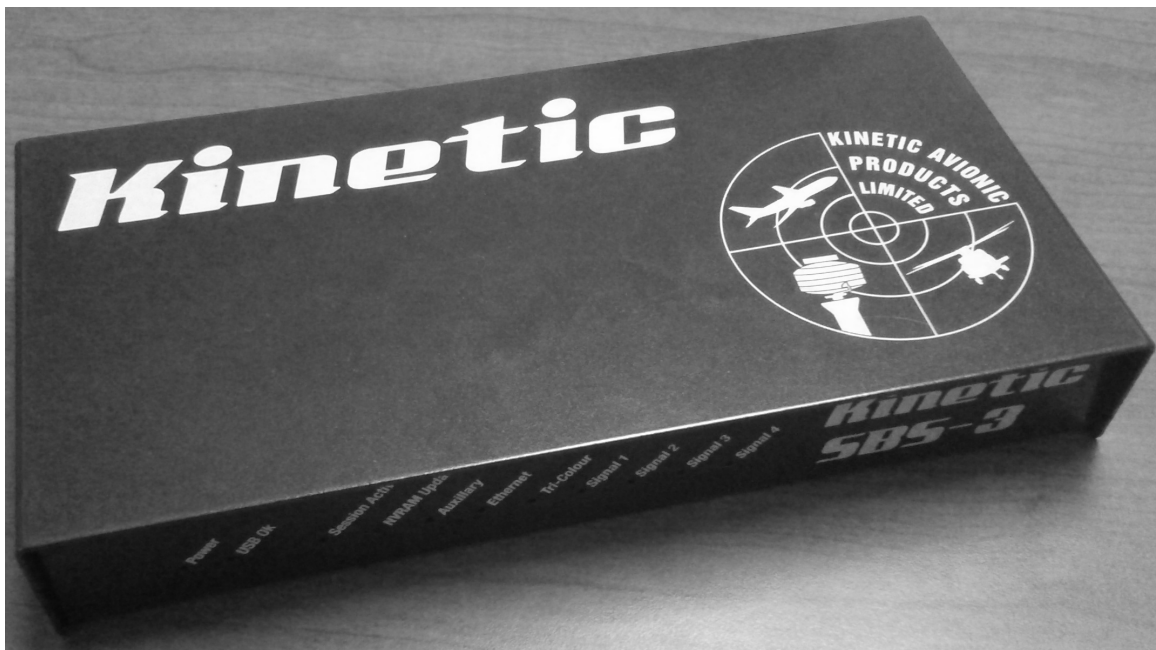


Figura 2.3: Dispositivo receptor de las señales ADS-B.

El *SBS-3* es un receptor *stand-alone* (no necesita del sistema operativo para funcionar), el cual es capaz de ejecutar operaciones de demodulación y decodificación internamente. Este diseño implica que puede monitorizar numerosas transmisiones de audio y datos simultáneamente y de manera remota.

El SBS-3 puede trabajar o bien como cliente, o como servidor, de tal manera que permite conectarse a él o puede conectarse a un determinado servidor a través de una red local o incluso internet, para su uso *stand-alone*. La alimentación eléctrica proviene de una entrada USB (*Universal Serial Bus*), la cual puede tener su origen en un ordenador, o un simple cargador USB. Se trata de un diseño flexible con una completa FPGA (*Field Programmable Gate Array*), la cual permite que en un futuro pudieran añadirse decodificadores para formatos como ACARS (*Aircraft Communications Addressing and Reporting System*) y POCSAG (*Post Office Code Standardisation Advisory Group*).

Además, trae implementadas nuevas características de receptor SDR (*Software-Defined Radio*) AM/FM, cubriendo un amplio rango de frecuencias, incluyendo las bandas VHF y UHF.

Algunas de las características del SBS-3 son:

- receptor ADS-B en Modo S de 1090 MHz;
- sintonizadores SDR duales e independientes desde 26 MHz hasta 980 MHz, con 8 MHz de ancho de banda por sintonizador;
- interfaz Ethernet de 100 Mbps de máxima velocidad con múltiples receptores;
- salida de audio a través de Ethernet; y
- procesado y demodulación incorporados, sin necesidad de ordenador.⁴

A fin de mejorar la cobertura que se conseguía con la antena que se suministraba con el receptor, se ha instalado una antena Kinetic Avionics BS-1105. La localización actual de la antena es el tejado de los laboratorios de la Escuela Técnica Superior de Ingenieros (ETSI) de la Universidad de Sevilla. Las coordenadas en las que está situada son 37.412122°, -6.002474°. La calidad de la señal conseguida con esta antena y la cobertura se tratan en las Secciones 3.13 y 4.3.1.

⁴<http://www.flightstore.co.uk/pilot-supplies-c1/kinetic-avionics-sbs-3-virtual-radar-p761> (15/09/2013).

3

Procesamiento de los datos

En este capítulo se presenta la metodología seguida para el procesamiento de los datos captados por el receptor ADS-B. El fin de este procesamiento es ordenar y filtrar la información recibida para, posteriormente, poder analizarlo.

Inicialmente, los datos emitidos mediante ADS-B por las aeronaves que se encuentran dentro del alcance de la antena son captados por el receptor y almacenados en archivos. A continuación, MATLAB realiza la lectura de estos datos y almacena únicamente aquellos que son de utilidad.

Posteriormente, se ordena la información atendiendo a la aeronave que la ha emitido y a los diferentes vuelos que esta ha realizado. Cada uno de los vuelos se clasifica en crucero, ascenso o descenso y se determina cuáles de estos dos últimos tienen como origen o destino el aeropuerto de Sevilla.

Seguidamente, se extraen los tramos de trayectoria en los que la aeronave se encuentra realmente ascendiendo o descendiendo y se les asigna una cabecera de pista. A estas trayectorias se les analizará el perfil horizontal, tal y como se muestra en el Capítulo 4.

Finalmente, para analizar el perfil vertical aun será necesario calcular la distancia recorrida y eliminar todos aquellos vuelos que no hayan transmitido datos con regularidad y para los que, por tanto, no se tenga una evolución suave de la altitud.

En las siguientes secciones se describen en detalle cada uno de estos pasos.

3.1. Captura de datos

El receptor ADS-B se encuentra conectado a un ordenador el cual, mediante el software comercial BASESTATION de KINETIC graba los datos obtenidos en archivos cuya extensión es *.bst. Estos archivos contienen 17 columnas, que corresponden a los siguientes datos¹:

¹Parte de la descripción de estos datos ha sido obtenida del foro de KINETIC (<http://www.kinetic.co.uk/forums/viewtopic.php?f=1&t=331> (15/09/2013)), mientras que la otra parte está basada en la experiencia de la realización de este proyecto.

- **Fecha:** Indica la fecha en la que se recibe el dato. Se encuentra en el formato “yyyy/m-m/dd”. El orden en el que aparecen los datos en el archivo de origen es cronológico, siendo la fecha la del propio ordenador.
- **Hora:** Indica la hora en la que se recibe el dato. Se encuentra en el formato “hh:mm:ss.mmm”, estando la hora en formato de 24 horas. Al igual que ocurría con la fecha, la hora es establecida por el propio ordenador. Además, se tiene la información cuantizada, es decir, los datos enviados por las aeronaves dentro de un mismo rango de tiempo, aparecen en el archivo de dato con la misma hora (aunque esta ventana de tiempo es pequeña, 0.015 segundos aproximadamente, los datos no son del todo exactos), que es la marcada por el ordenador en el momento en el que recibe el paquete de datos correspondiente.
- **Dirección del Modo S (formato *raw*):** Se trata de un código en formato *raw*, que identifica a la aeronave. Consiste en un número en formato decimal de 7 dígitos.
- **Dirección del Modo S (formato hexadecimal):** Se trata de la misma información que en el caso anterior, solo que en formato hexadecimal. De esta manera son necesarios únicamente 6 dígitos.
- **CallSign:** Identifica el vuelo realizado por una aeronave. Según OACI [4], el call sign asignado a un vuelo debe formarse atendiendo a uno de los siguientes métodos:
 - ➡ los caracteres correspondientes al código de registro de la aeronave;
 - ➡ el designador telefónico de la agencia operadora de la aeronave, seguido de los últimos cuatro caracteres del código de registro de la aeronave; o
 - ➡ el designador telefónico de la agencia operadora de la aeronave, seguido del código de identificación del vuelo.

Aunque se desconoce qué método utiliza cada aerolínea, según se ha observado durante la realización del presente proyecto, el tercer método parece ser el más común. Esto permitirá asociar call sign y orígenes o destinos.

El call sign es un dato que puede llegar a dar problemas a la hora de realizar el procesado de los datos, puesto que no siempre se transmite correctamente, no obteniéndose entonces ningún valor. Por este motivo tienen que tomarse ciertas precauciones como las que se comentan en la Sección 3.8.

- **País:** Indica el país de matriculación de la aeronave.
- **IsOnGround:** Se trata de un valor binario, el cual puede tomar los valores -1 ó 0, dependiendo si la aeronave se encuentra o no en tierra. Si el IsOnGround está marcado como -1, el valor de la altitud toma el valor 0 automáticamente.
- **Altitud:** Indica la altitud barométrica a la que se encuentra la aeronave en pies (ft). Este campo tiene una precisión de 25 ft y está expresado en formato decimal.
- **Altitud mostrada:** Se trata de un campo creado por si en el futuro se decide transmitir algún otro parámetro y que mientras tanto, se ocupa con una copia de la altitud mostrada en el campo anterior.

- **Latitud:** Indica la latitud en la que se encuentra la aeronave en cada momento. Se transmite en formato decimal, las unidades utilizadas son grados y se incluyen hasta cinco decimales.
- **Longitud:** Indica la longitud en la que se encuentra la aeronave. Utiliza el mismo formato, precisión y unidades que la latitud.
- **ROC (*Rate Of Climb*):** Indica la velocidad de ascenso de la aeronave en pies por minutos (ft/min). En caso de que la aeronave se encuentre descendiendo, simplemente se mostrará la velocidad de descenso con un valor negativo. Se expresa en formato decimal con precisión de 64 ft/min.
- **ROC calculado:** Se trata de un parámetro calculado para aquellos vuelos que transmiten la altitud en todo momento, pero no el ROC. Actualmente, se calcula para todas las aeronaves aunque solo se usa para aquellas que no transmiten el ROC directamente. Las unidades y el formato son las mismas que en el caso del ROC.
- **GroundSpeed:** Se trata de la velocidad de la aeronave respecto a tierra. Las unidades son kt, con precisión de décimas y formato decimal.
- **Rumbo:** Indica el rumbo seguido por la aeronave en grados, con precisión de décimas y formato decimal.
- **Squawk (decimal):** Se trata de un código decimal de cuatro dígitos asignado por el ATC a cada transpondedor que lo solicita para así mantenerse identificado. Cuando aun no se tiene identificación, toma un valor nulo.
- **Squawk (octal):** Es el parámetro anterior en codificación octal. Si el valor del *squawk* en decimal es nulo, el campo correspondiente a la codificación octal se mantiene en blanco.

Las columnas se encuentran separadas por comas y cada dato obtenido se encuentra entre comillas. En la Figura 3.1 se muestra un extracto de uno de los archivos, en el que se puede apreciar el formato en el que se obtienen los datos.

```
cto5Zk.bst
"2013/03/15","12:24:11.234","131309","0200ED","RAM662F","Morocco","0","36000","36000","38.33711","-5.41141","0","0","458.9","17.4","4183","1057"
"2013/03/15","12:24:11.234","4687504","478690","SAS7801","Norway","0","37000","37000","37.92206","-7.75798","0","0","426.0","193.0","1875","0753"
"2013/03/15","12:24:11.234","3417490","342592","VLG3046","Spain","0","3975","3975","37.40163","-5.80487","1728","1728","180.1","173.6","0",""
"2013/03/15","12:24:11.234","3950741","3C4895","BER808A","Germany","0","35025","35025","38.53105","-8.57111","64","64","423.2","205.2","9553","2551"
"2013/03/15","12:24:11.234","3412241","341111","IBS3910","Spain","0","35000","35000","38.78737","-7.78077","0","0","425.2","227.0","17712","4530"
"2013/03/15","12:24:11.234","3957037","3C612D","GMI3819","Germany","0","35975","35975","38.99797","-9.03955","0","25","459.0","45.8","21585","5451"
"2013/03/15","12:24:11.343","4882875","4A81BB","","Sweden","0","35000","35000","39.10918","-7.40514","0","0","448.2","193.4","1847","0737"
"2013/03/15","12:24:11.343","4565638","45AA86","JTG236","Denmark","0","32000","32000","38.73935","-7.35827","0","0","428.6","12.1","18247","4747"
```

Figura 3.1: Sección de uno de los archivos de datos recibidos.

Para no hacer que cada uno de estos archivos ocupe mucha memoria, se para la grabación cada varios días (de 4 a 8 días aproximadamente) y se reanuda a la mayor brevedad posible, de manera que se pierdan el menor número de datos posibles durante el cambio.

3.2. Partición de los archivos de datos

Durante la realización del proyecto se ha encontrado que cuando MATLAB realiza la lectura de los archivos *.bst, como se explica en la Sección 3.3, este requiere una cantidad de memoria

RAM bastante superior al tamaño del archivo, pudiendo dar lugar a falta de memoria. Para posibilitar entonces la lectura de los archivos ha sido necesario dividirlos en particiones más pequeñas.

La partición de los archivos se realiza mediante un script de VISUAL BASIC (puede consultarse en el Apéndice A), de tal manera que dicho script no necesite abrir los archivos completos, sino que accede a cada línea de ellos directamente y las copia en un nuevo archivo, deteniéndose una vez que haya alcanzado el número de líneas exigido y continuando con la creación de un nuevo archivo hasta copiar por completo el archivo original.

Este script necesita como entradas el nombre del archivo que se desea partir (deben encontrarse en la misma carpeta) y el número de líneas deseado en los nuevos archivos, creando una carpeta con el mismo nombre que el archivo original y dentro de esta, las nuevas particiones cuyo nombre serán el del archivo original seguido del orden que ocupa dicho archivo en el conjunto correspondiente.

En el caso de este proyecto, se ha decidido dividir los archivos originales de tal manera que los nuevos tengan 100000 líneas. De esta manera, el ordenador puede leerlos sin que aparezcan problemas de memoria RAM² en un tiempo aceptable y por otro lado, no necesita mucho tiempo para la posterior unión y clasificación de las aeronaves.

3.3. Lectura de los datos

Conocido el archivo de datos que se desea analizar, para realizar la lectura de cada archivo, MATLAB accede a la carpeta donde este se encuentra y lee todas las particiones que se hayan realizado, utilizando la función *textread* e indicándole en la opción correspondiente al formato en el que se encuentran los datos, que se trata de datos separados por comas y entrecomillados.

Para cada archivo que esté leyendo, se obtienen 17 vectores columnas correspondientes a cada uno de los datos recibidos, cuyo número de filas es igual al número de líneas en los que se eligiera partir los archivos de datos originales.

Comprobar la coincidencia del número de líneas del archivo original, de la suma de los números de líneas de las particiones y del número de líneas leídas se ha considerado como un método aceptable para asegurar la lectura de todas y cada una de las líneas del archivo original.

El formato utilizado para guardar los datos es en primera instancia *cell*. De estos 17 vectores columnas, se eliminan los que no vayan a utilizarse³ para así ir liberando parte de la memoria RAM utilizada. El script de MATLAB creado para desarrollar esta tarea se llama *gen_lectura.m* y se puede consultar en la Sección B.1.

²Con 3 GB, en una máquina corriendo con WINDOWS 7, los problemas aparecen cuando se aproxima a las 350000 líneas

³Los datos que se eliminarán serán los correspondientes al modo hexadecimal en formato raw, el país, la copia de la altitud, la ROC calculada, la velocidad respecto al suelo, el squawk en ambos formatos y el rumbo.

3.4. Separación por aeronaves

Una vez leídos los datos de una partición, se agrupan las líneas de los vectores según la coincidencia del código hexadecimal correspondiente a la aeronave que los ha enviado. Para esto, se escoge un código hexadecimal y, si no ha sido escogido anteriormente, se buscan todas las líneas en las que aparezca, guardando en un vector (para cada código hexadecimal distinto) las líneas en las que se repetía. La función de MATLAB desarrollada para realizar esta tarea se ha denominado *func_matriculas* y se puede consultar en la Sección B.2.

Una vez conocido el número de códigos hexadecimales distintos que han aparecido y las posiciones en las que se encuentra cada uno, se recorrerán todos estos códigos para almacenar en una estructura (cuya longitud será igual al número de códigos distintos encontrados) los vectores procedentes de la lectura considerados interesantes: fecha, hora, altitud, latitud, longitud, IsOnGround, ROC y CallSign. El formato de las estructuras creadas se encuentra en el Apéndice C.

De nuevo, para evitar problemas de falta de memoria RAM, se eliminan todos aquellos vectores que no sean de más utilidad.

3.5. Unión de las particiones

A continuación será necesario unir todos los datos guardados anteriormente correspondientes a cada una de las particiones realizadas de los datos originales. El script de MATLAB creado para desarrollar esta tarea se llama *gen_union.m* y se puede consultar en la Sección B.3.

Se podría plantear también realizar la unión de todas las partes leídas de todos los archivos de datos originales, pero de nuevo aparecen problemas de memoria RAM, aunque en este caso sí que sería posible abarcar una mayor ventana de tiempo, pero seguiría siendo insuficiente para unir las nueve semanas que se buscan en este proyecto.

Para realizar la unión de los datos leídos, se van cargando las estructuras guardadas anteriormente una a una, uniéndolas todas en una misma estructura y borrando el dato cargado una vez que este ya ha sido incluido en la estructura global con el fin de no ocupar más memoria RAM de la necesaria.

Cabe recordar que en cada estructura leída con anterioridad, los datos se encuentran agrupados por aeronaves. Existe la posibilidad de que la misma aeronave aparezca en varias estructuras, por lo que una vez unidas será necesario reagrupar las aeronaves en una única componente de la estructura.

3.6. Búsqueda de modelos y aerolíneas

Una vez se han unido todos los datos que se tienen de una misma aeronave, se pasará a buscar de qué modelo de avión se trata y cuál es la aerolínea que se encarga de operarlo.

Como esta búsqueda se realizará para todos los archivos obtenidos, no solo se buscarán los modelos y las aerolíneas correspondientes, sino que además, se creará una base de datos para que no haya que realizar la búsqueda cada vez que aparezca una aeronave repetida en los distintos archivos a analizar y puedan obtenerse algunas estadísticas globales relativas a estos datos. El script de MATLAB creado para desarrollar esta tarea se llama *gen_model_AL.m* y se puede consultar en la Sección B.4.

Conocidos el número de aeronaves distintas que se han encontrado, se recorren todas y cada una de ellas, comprobando en primer lugar si su código hexadecimal se corresponde con alguno de los ya almacenados en la base de datos. Si es así, se le asigna a la estructura el modelo y la aerolínea correspondiente. De lo contrario, se realizará una búsqueda automática en internet: utilizando como motor de búsqueda *Yahoo*⁴ y como base de datos *Flightradar24*⁵ y se construirá una *url* correspondiente a dicho motor de búsqueda y base de datos, de tal manera que se obtenga como respuesta un código *html* a partir del cual se pueden obtener el modelo de la aeronave y su aerolínea operadora correspondiente con un determinado criterio de búsqueda. El siguiente código es un ejemplo de *url* utilizada para buscar el modelo de una aeronave y la aerolínea a la que pertenece. Únicamente habría que sustituir el código hexadecimal de ejemplo “4CA2DB” (el cual se encuentra subrayado en el siguiente código) por el código en concreto que se quisiera buscar.

```
http://es.search.yahoo.com/search;_ylt=A7x9QbzyFXdRlx4AHheT.Qt.?p=site%3Ahttp%3A%2F%2Fwww.flightradar24.com%2Fdata%2FAirplanes%2F%20ModeS%3A%204CA2DB&fr2=sb-top&fr=sfp&rd=r1
```

Como se puede deducir de lo comentado anteriormente, es necesario tener conexión a internet con una cierta calidad para acceder sin problemas a la *url* buscada. Es más, por posibles problemas que pudieran aparecer en un determinado momento, se intenta hasta cuatro veces acceder a la *url* en caso de que falle cada una de las búsquedas anteriores, haciendo una pausa en la búsqueda cada vez que se obtenga un error para dar tiempo al restablecimiento de la conexión. La función de MATLAB utilizada para la búsqueda de esta información es *urlread*.

A la hora de realizar la búsqueda de los modelos se han encontrado dos tipos de errores en la base de datos consultada. Por un lado, a veces falta un “;” en el código *html*, lo que no delimita correctamente el modelo de la aeronave. Por otro lado, también se han detectado errores en los nombres de los modelos de las aeronaves o de las propias compañías, como por ejemplo dobles espacios, letras mayúsculas, guiones, etcétera. Es por esto que una vez conocidos tanto el modelo como la aerolínea, se les pasa a ambos un filtro para la corrección de estos errores y así tener un criterio unificado. Sin embargo, aunque se han detectado estos errores, en las estructuras propias de cada aeronave se guardan dos campos correspondientes al dato obtenido directamente de internet y al dato una vez ha pasado por los filtros (tanto para los modelos como para las aerolíneas).

Una vez se han obtenido de internet el modelo de aeronave y la aerolínea, se guardan estos resultados en la base de datos para su posible uso en casos posteriores.

⁴Se ha escogido *Yahoo* porque *Google* y *Bing* dieron problemas a la hora de automatizar el proceso, debido a que el código recibido de la *url* de *Google* no aporta ninguna información y la eficiencia de *Yahoo* a la hora de encontrar modelos es mayor que la de *Bing*.

⁵www.flightradar24.com(15/09/2013).

Debido a que por ejemplo, hay aeronaves fabricadas últimamente y no han podido ser incluidas aun en las base de datos consultada, o aviones especiales de algún gobierno o fuerza armada, se ha creado un pequeño programa en MATLAB, el cual recorre todos los campos (tanto los relativos a los modelos como a las aerolíneas) en busca de aquellos que estén en blanco. El programa devuelve el código hexadecimal de la aeronave en cuestión y permite introducir los datos que faltan, los cuales ha sido necesario encontrarlos manualmente en otras fuentes (como por ejemplo *airframes*⁶). Si aun así no se consigue encontrar la información buscada, se han completado los campos desconocidos con el término “unknown”.

El filtro de los modelos de las aeronaves termina con una agrupación de aeronaves similares. Por ejemplo los Airbus A320-2 operados por Vueling o por Iberia se consideran como el mismo modelo de aeronave, aunque tengan diferencias como el número de asientos o la motorización. Las aeronaves se han agrupado a nivel de series puesto que esta sí se ha considerado una diferencia sustancial como para diferenciar claramente dos aeronaves. En aeronaves que no son tan comunes y en las que no se tiene un criterio claro para identificarlas como el mismo modelo o distintos, se ha utilizado la lista empleada por EUROCONTROL, la cual se puede consultar en su base de datos BADA (*Base of Aircraft Data*) [5].

3.7. Separación de vuelos

Debido a que se tienen todos los datos agrupados por aeronave, es necesario separar estos según el vuelo al que pertenezcan. Esta separación se realizará buscando dentro de los datos de cada aeronave diferencias de tiempo entre dos componentes mayores que un determinado valor, en este caso 300 segundos. Se ha elegido este valor porque se ha considerado lo suficientemente pequeño como para diferenciar el aterrizaje y despegue de una misma aeronave que pasa el menor tiempo posible en el aeropuerto, y lo suficientemente grande como para que esta diferencia de tiempos no se alcance por cuestión de una señal mal recibida u otros motivos que pudieran aparecer.

En el caso de que aparecieran vuelos con una única componente, estos serán descartados debido a la poca utilidad que aportan. Este hecho puede deberse a que la aeronave se encuentre suficientemente lejos como para haberla detectado una vez, o que las diferencias de tiempo entre algunos de sus puntos sean mayores de 300 s y por tanto, se hayan considerado como vuelos distintos.

El script de MATLAB creado para desarrollar la separación de los vuelos se llama *gen_sep.m* y se puede consultar en la Sección B.5.

3.8. Asignación de un CallSign a cada vuelo

Cada paquete de datos recibido por la estación ADS-B contiene un campo relativo al Call-Sign. Si este dato se envía correctamente en todo momento, no presentaría ninguna variación

⁶<http://www.airframes.org> (15/09/2013).

para un mismo vuelo y asignándole el primer CallSign recibido sería suficiente para caracterizarlo, pudiendo eliminar el resto y así liberar algo de memoria.

Sin embargo, la columna de datos relativa al Callsign puede presentar algunos errores, como puede ser por ejemplo estar en blanco. Por este motivo, no se le puede asignar a un determinado vuelo el primer Callsign que contenga (o uno aleatorio), siendo necesario que este dato pase un filtro. Al tratarse de un dato que se envía repetidamente al mismo tiempo que el resto, de todos los enlaces de datos enviados solo unos pocos estarán en blanco, obteniéndose la mayoría en buenas condiciones. El script de MATLAB creado para desarrollar este filtro se llama *gen_CS.m* y se puede consultar en la Sección B.6. Este filtro, para cada vuelo, va leyendo los Callsign por el orden en el que aparecen y se queda con el primer Callsign cuya longitud sea mayor que 3, asignándole de esta manera un único Callsign a cada vuelo y eliminando el vector completo que se obtuvo originalmente. Aunque únicamente se han detectado campos nulos, se ha elegido mayor que 3 por si apareciera algún otro CallSign defectuoso con una longitud mayor. No se ha elegido un valor mayor, porque ya eliminaría algunos correctos que pudieran simplemente ser cortos.

3.9. Clasificación de los vuelos en cruceros, ascensos o descensos

Encontrándose ya todos los vuelos separados, se pasará a agruparlos según hayan ascendido, descendido o se encuentren en crucero. Una vez caracterizados los vuelos, se distinguirá si los ascensos y descensos corresponden a despegues o aterrizajes respectivamente, realizados en el aeropuerto de Sevilla. El script de MATLAB creado para desarrollar esta tarea se llama *gen_estado.m* y se puede consultar en la Sección B.7.

La determinación del estado de la aeronave según esté ascendiendo, en crucero o descendiendo se realiza de una manera sencilla: si la diferencia de altura entre la última componente del vuelo y la primera es mayor que un determinado valor será un ascenso, si es menor que el negativo de dicho valor, será un descenso y si no se da ninguno de los casos anteriores, se considerará que está en condición de crucero.

El valor de la diferencia de altura necesaria para distinguir entre un caso u otro se ha elegido de 3000 ft. Este valor se ha considerado suficientemente grande como para que un vuelo en crucero no se vea afectado por un posible cambio de nivel que pudiera experimentar y lo suficientemente pequeño como para detectar aeronaves con un despegue o un aterrizaje ya iniciado.

3.10. Salidas y llegadas en el aeropuerto de Sevilla

Una vez se conoce que el vuelo se trata de un ascenso o un descenso, sus datos se introducen en una función (*func_aeropuertos*) que se puede consultar en la Sección B.8, la cual determinará si la aeronave tiene como origen o destino el aeropuerto de Sevilla. Esta función obtiene una latitud y una longitud en las que se encuentra la aeronave en un determinado momento. Si esta

localización se encuentra dentro del rectángulo de tolerancia definido en torno a la pista del aeropuerto de Sevilla, se considerará que la aeronave ha salido o llegado de dicho aeropuerto.

Las coordenadas de referencia de cada vuelo son las correspondientes al primer paquete de datos enviado por la aeronave una vez que está en el aire en el caso de estar en ascenso o el paquete anterior al primer paquete de datos enviado una vez que la aeronave ha tocado tierra en el caso de un descenso.

El método que se ha utilizado para detectar estos instantes se basa en el uso del parámetro *IsOnGround*. En el caso de los ascensos, se utiliza la componente en la que el valor del *IsOnGround* es 0 por primera vez (significa que no está tocando tierra). Por otro lado, en el caso de los descensos, la componente buscada es la anterior a la primera en la que el valor del *IsOnGround* es -1 por primera vez (la aeronave se encuentra en contacto con el suelo), es decir, la componente que indica el último punto en el que la aeronave se encuentra en el aire.

En la Figura 3.2 se muestra el rectángulo de tolerancia definido alrededor de la pista del aeropuerto de Sevilla, habiendo utilizado para representarlo la proyección de Lambert descrita en el Apéndice D.

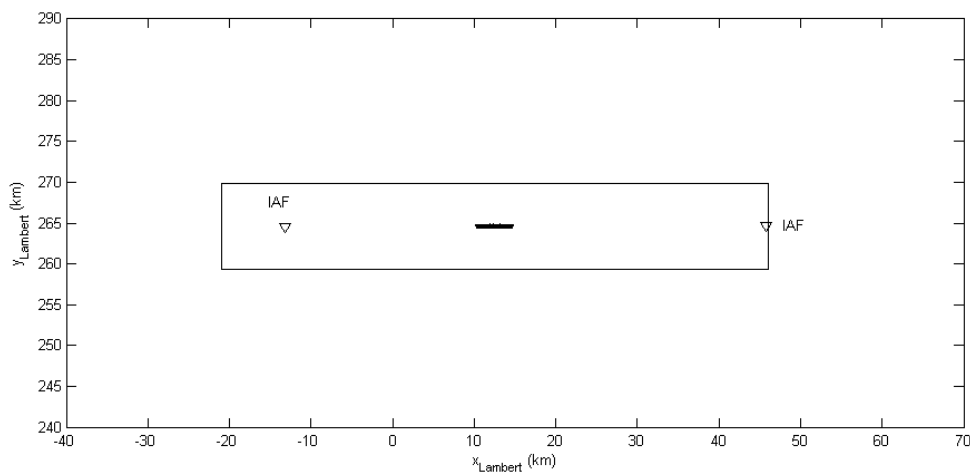


Figura 3.2: Representación del rectángulo de tolerancias.

Los requisitos para determinar el tamaño del rectángulo de tolerancias son:

- No puede ser demasiado pequeño porque es habitual perder la señal de la aeronave durante el despegue y el aterrizaje por debajo de 1000 ft.
- No puede ser demasiado grande para no confundir los despegues y aterrizajes de Sevilla con los de otros aeródromos.

Atendiendo a estos requisitos, se ha decidido escoger un rectángulo centrado en la pista del aeropuerto de Sevilla con un ancho que abarca los dos IAF (*Initial Approach Fix*), una relación de aspecto de 6 a 1 y orientación la de la pista. Se ha escogido esta relación de aspecto atendiendo al movimiento de las aeronaves, que suelen desplazarse en el sentido longitudinal de la pista y no tanto en el transversal. Debido a la orientación particular de la pista del aeropuerto de Sevilla,

09-27 (Este-Oeste), resulta muy sencilla la aplicación de este rectángulo, pues no es necesario girarlo. El IAF que se encuentra más alejado de la pista es el situado al Este y su distancia al centro de la pista en coordenadas geodésicas es de $\Delta\lambda = 16'24.85'' = 0.27357^\circ$ (*longitud*), por lo que el ancho del rectángulo será el doble de este valor.

Además, por tratarse de una proyección esférica, ha sido necesario aplicar un factor de corrección según la latitud a la que se encuentra el aeropuerto, por lo que finalmente la semialtura del rectángulo es $\Delta\phi = \Delta\lambda \cos(37.418)/6 = 0.03621^\circ$.

3.11. Determinación de los tramos de ascenso/descenso

A continuación, se aplica una función (*func_estad*), la cual puede consultarse en la Sección B.9 que obtiene los tramos de ascenso y descenso de los vuelos que han sido categorizados con destino u origen en Sevilla. A continuación se describe cómo se determinan estos tramos:

- **Salidas:** El inicio del tramo de ascenso viene marcado por el primer punto en el que el valor del parámetro *IsOnGround* es nulo (aeronave en el aire), mientras que el final es el primer punto en el que la aeronave se encuentra por encima de los 24500 ft⁷, en un nivel de vuelo concreto (la altitud es múltiplo de 1000 ft, con una tolerancia de 100 ft) y el valor absoluto del ROC es menor que 150 ft/min⁸. Si no se encontrara ningún punto que cumpliera la condición de fin de ascenso, se utilizaría el último punto que se tenga de dicho vuelo, pues se considerará que la aeronave ha alcanzado su tramo de crucero, o se ha perdido la señal (por cualquier motivo, como puede ser simplemente haber cortado la grabación) y así, se cortará lo más tarde posible.
- **Llegadas:** El final del tramo de descenso queda definido con el punto anterior al primer punto en el que el *IsOnGround* es -1 (aeronave en contacto con tierra), es decir, en el último punto que se tiene con la aeronave en el aire justo antes de tocar tierra, mientras que el inicio del descenso es el último punto en el que la aeronave se encuentra por encima de los 24500 ft, en un nivel de vuelo concreto (la altitud es múltiplo de 1000 ft, con una tolerancia de 100 ft) y el valor absoluto del ROC es menor que 150 ft/min. Si no se encontrara ningún punto que cumpliera la condición de inicio de aterrizaje, se utilizaría el primer punto que se tuviera de dicho vuelo por el mismo motivo alegado en los despegues.

Puede darse el caso que la componente inicial encontrada sea posterior a la componente final. El caso encontrado se muestra en la figura 3.3. Este hecho se debe a que, aunque la calidad de la señal sea buena en términos de frecuencia de datos recibidos, los datos recibidos no lo son, apareciendo, como puede apreciarse, valores nulos puntualmente en la altitud cuando la aeronave se encuentra aun lejos del suelo. Lo que se ha decidido hacer con este tipo de casos es descartarlos directamente, pues no suponen un gran volumen de datos y pueden llegar a ser problemáticos a la hora de analizar resultados.

⁷24500 ft es la separación entre espacio aéreo superior e inferior en España.

⁸Tanto la tolerancia para determinar un nivel de vuelo, como el límite establecido para el ROC se han elegido de tal manera que el valor resulte suficientemente pequeño como para no afectar a otro nivel de vuelo por un lado, y poder mantener la condición de crucero por el otro.

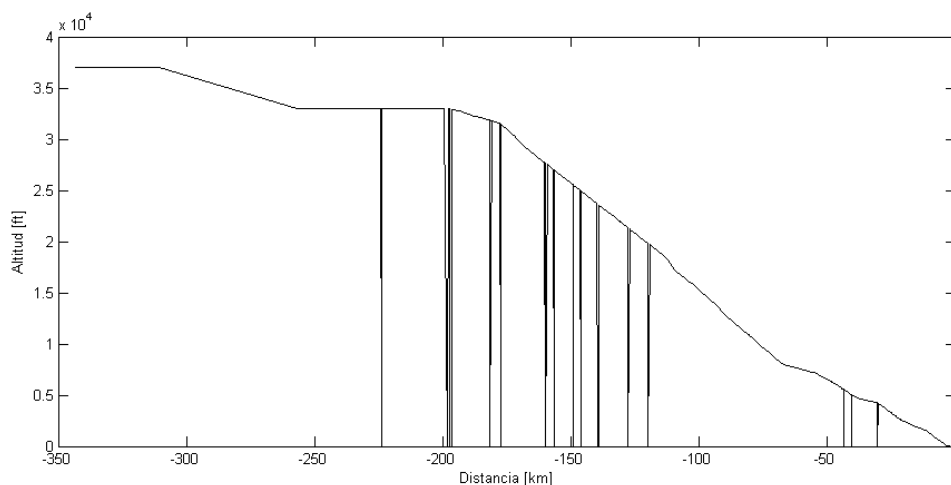


Figura 3.3: Vuelo en el que la componente final encontrada es menor que la inicial.

3.12. Cabeceras de pista

Se determinará la cabecera de la pista del aeropuerto de Sevilla por la que cada aeronave realiza el correspondiente despegue/aterrizaje. Para la realización de estos cálculos se ha usado la función *func_lambert.m*, la cual se puede consultar en la Sección B.10. Consiste simplemente en determinar si una diferencia entre dos valores de la longitud resulta positiva o negativa, conociendo de esta manera la orientación que sigue la aeronave. Estos dos valores se eligen dependiendo si se trata de un despegue o un aterrizaje:

- **Despegues:** La diferencia en longitud será la existente entre el cuarto punto en el que la aeronave se encuentra en el aire y el último en el que permaneció en tierra, si es que se ha llegado a encontrar en ambos estados. Si se diera el caso en el que la aeronave se encontrara ascendiendo aunque no se hubiera registrado ningún punto en tierra, será la diferencia entre el cuarto punto que superó los 1000 ft de altitud y el punto anterior a este momento. El hecho de elegir la cuarta componente es simplemente por dar una cierta diferencia entre ambas componentes elegidas, para así evitar casos en los que los valores de longitud sean muy próximos o incluso el mismo. No se ha elegido mayor para evitar casos en los que la aeronave realice algún viraje poco después de haber abandonado la pista⁹.
- **Aterrizajes:** En este caso, la diferencia será la existente entre el primer punto en el que la aeronave ha tocado tierra y la cuarta componente anterior si es que la aeronave se ha encontrado en ambas situaciones. De no encontrar ningún punto en tierra, será la diferencia entre el último punto registrado en dicho vuelo y el cuarto punto anterior. La razón de usar estas diferencias es similar al del caso de los despegues.

⁹Puede darse el caso de salidas de aeronaves en las que no se haya registrado el despegue como tal. Esto puede deberse a falta de cobertura o al caso concreto en el que la aeronave se encontrara despegando en el momento en el que se interrumpió la grabación de un archivo de datos y se inició la del siguiente archivo.

3.13. Calidad de la señal

Una vez se tienen los tramos de descenso/ascenso de los vuelos con destino/origen el aeropuerto de Sevilla, se realizan dos operaciones relativas a la calidad de la señal, en términos de tiempo transcurrido entre la recepción de dos paquetes de datos consecutivos, a fin de obtener perfiles verticales con una evolución suave de la altitud.

En primer lugar, se identifica el rango de altitudes en el que se suele tener una buena recepción en todos los vuelos. Este rango de altitudes será el que se utilice en el análisis del perfil vertical llevado a cabo en el Capítulo 4.

En segundo lugar, se descartan los vuelos en los que, en el rango de altitudes seleccionado, no se hayan recibido datos con una suficiente cadencia.

En las Figuras 3.4 y 3.5 se muestran los tiempos entre paquetes de datos como función de la altitud para las salidas y las llegadas desde Sevilla respectivamente, para los mismos datos analizados en el Capítulo 4. Estas gráficas muestran dentro de las cajas azules los valores que se encuentran entre el percentil 0 y el 95, mostrando la línea roja el valor de la media y los puntos rojos los valores que se encuentran fuera del percentil 95. Para realizarlas, se ha modificado la función *boxplot* de MATLAB.

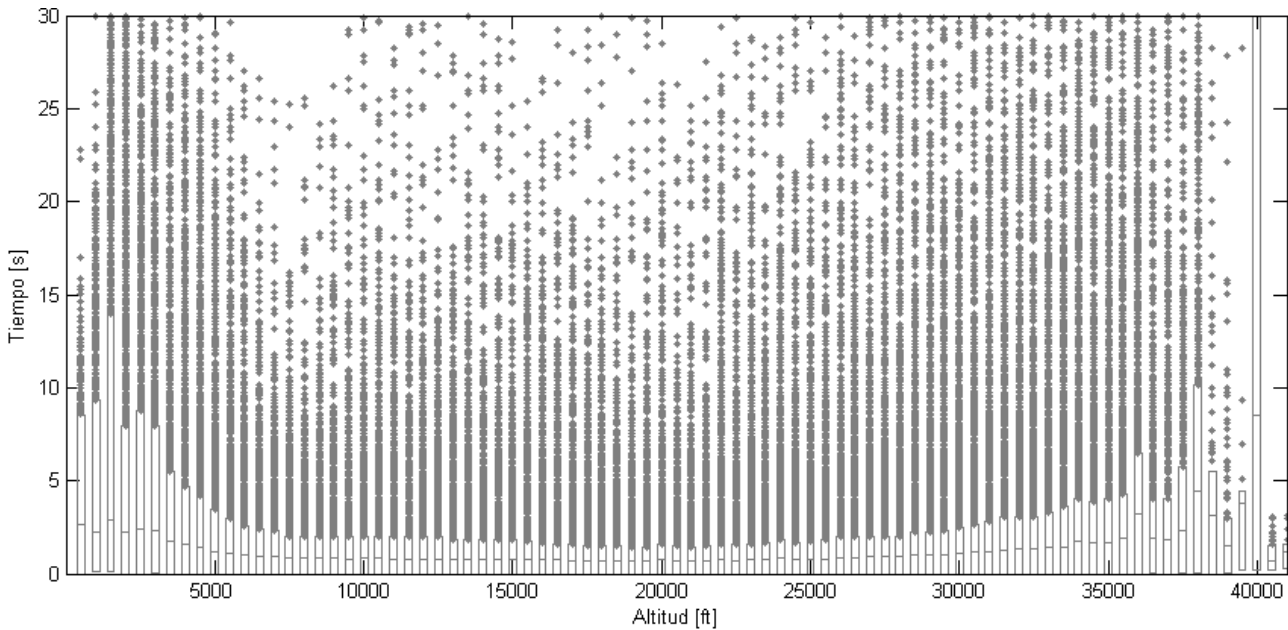


Figura 3.4: Calidad de la señal para el caso de las salidas.

Como se puede apreciar, en ambos casos la calidad de la señal a bajas altitudes no es todo lo buena que se podía esperar. Esto se debe principalmente a las interferencias que presenta la señal a la hora de ser recibida por parte de la antena. Por otro lado, la calidad de la señal también disminuye a altitudes bastante elevadas, debido principalmente a la distancia a la que se encuentran las aeronaves de la antena.

En el caso de las salidas, se ha escogido el rango entre 4000 y 35500 ft. Así, en este rango, para todas las altitudes, el 95% de los tiempos están por debajo de los 5 segundos. Con este mismo criterio, en el caso de las llegadas, se ha escogido el rango entre 3000 y 32000 ft.

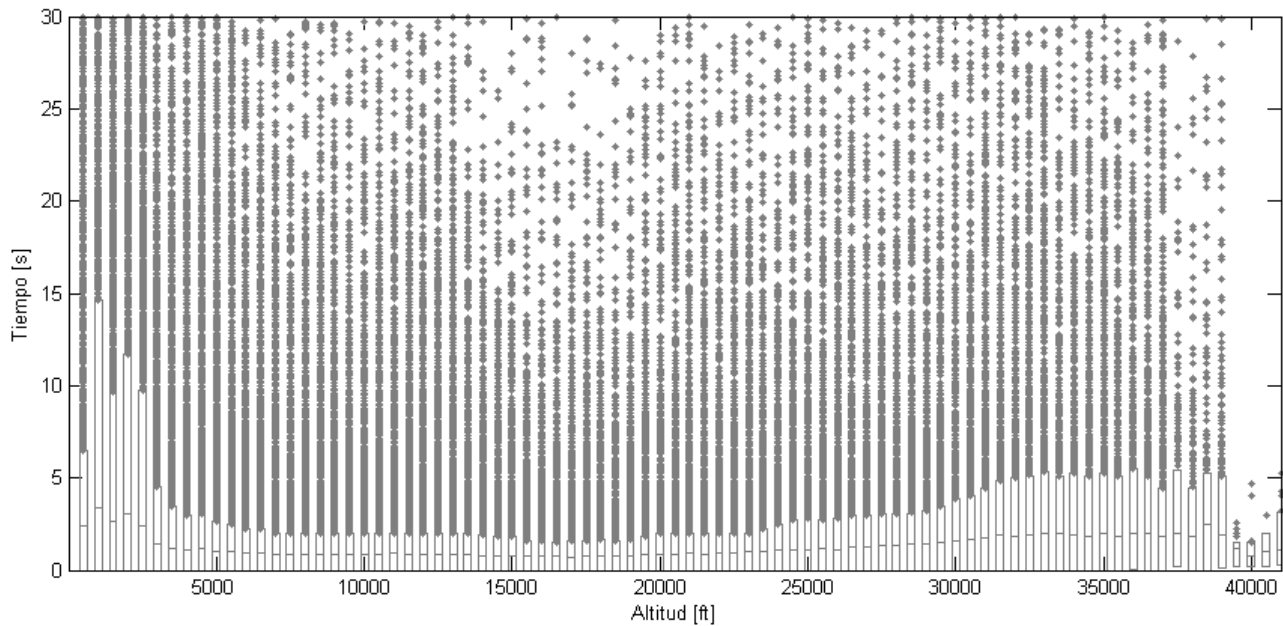


Figura 3.5: Calidad de la señal para el caso de las llegadas.

Finalmente, no se consideran todos aquellos vuelos cuya diferencia máxima entre dos datos consecutivos en los rangos seleccionados sea mayor que 30 segundos y la media sea mayor que 10 segundos.

El script de MATLAB creado para desarrollar este filtro doble, se llama *gen_retraso.m* y se puede consultar en la Sección B.11, así como la función principal utilizada, denominada *func_filtro_retraso.m*, que se puede consultar en la Sección B.12.

3.14. Determinación de la distancia recorrida

El cálculo de la distancia recorrida por la aeronave se realiza a través del script *gen_repr*, mostrado en la Sección B.13, el cual contiene la función *func_repr* que se puede consultar en la Sección B.14. Esta función calcula en primer lugar el radio de la Tierra en Sevilla y su excentricidad atendiendo al modelo WGS84¹⁰ (*World Geodetic System*). Posteriormente, pasará a calcular la distancia en todos los vuelos introducidos en la función excepto para aquellos vuelos que contengan una sola componente, puesto que no es posible calcular distancias recorridas en ese caso.

Para calcular la distancia entre cada pareja de puntos, se ha utilizado la función *distance* de MATLAB. En esta función se introducen las coordenadas de los puntos, el radio de la Tierra y su excentricidad, obteniendo directamente un vector con la distancia entre cada pareja de puntos. Esta función de MATLAB permite además una opción que indique el modo en el que se ha recorrido dicha distancia, habiéndose elegido en este caso rutas loxodrómicas. Otra opción podría haber sido calcular la distancia como si la aeronave hubiera seguido rutas ortodrómicas, sin embargo se descartó esta opción pensando que las rutas loxodrómicas resultarían más

¹⁰http://en.wikipedia.org/wiki/World_Geodetic_System (15/09/20139).

realistas.

Calculada la distancia entre cada dos puntos consecutivos, es inmediato el cálculo de la distancia acumulada desde el inicio del vuelo.

4

Análisis de los datos

Para la realización de este proyecto se han recogido datos de la estación ADS-B durante nueve semanas comprendidas entre el 15 de Mayo y el 17 de Julio de 2013. Durante este tiempo, se han registrado un total de 95760¹ vuelos de 82 modelos de aeronaves, pertenecientes a 313 compañías aéreas distintas, recogidos del receptor ADS-B en 14 archivos *.bst. El procesado de cada archivo viene descrito en el Capítulo 3. La agregación de todos los resultados obtenidos es la que se muestra en este capítulo.

En la Figura 4.1 se muestran los perfiles horizontales de todos los vuelos captados en este periodo. Todos los perfiles horizontales utilizados para el análisis de los datos, se han realizado con la proyección cónica de Lambert descrita en el Apéndice D. Como se puede comprobar, la mayor distancia a la que se ha llegado a capturar datos ronda los 500 km, llegando a captar aeronaves operando en los aeropuertos de Faro, Málaga, Jerez e incluso Barajas.

Los 20 modelos de aeronaves, así como las aerolíneas que más veces han transitado por el espacio aéreo captado por la estación ADS-B se muestran en la Tabla 4.1. Con gran diferencia destacan las aeronaves Boeing 737-8 y Airbus A320-2, así como la aerolínea Ryanair.

A continuación, en la Sección 4.1 se analizan los perfiles horizontales de los vuelos con origen o destino el aeropuerto de Sevilla. En la Sección 4.2 se analizan los perfiles verticales tras aplicar el filtro de calidad de la señal. Por último, en la Sección 4.3 se presentan otros resultados obtenidos, como aeronaves especiales, circuitos de espera, etc.

4.1. Perfil horizontal

En esta sección se analizarán por separado los perfiles horizontales de las salidas y llegadas, separándolos además por cabeceras para así poder analizar el comportamiento de los vuelos según el caso y la precisión con la que se cumplen las rutas marcadas en las cartas.

En cuanto a las aeronaves que operan desde el aeropuerto de Sevilla, se puede decir que

¹Se han registrado un total de 98479 vuelos, de los cuales 2719 (2%) tienen una única componente, por lo que no son de utilidad.

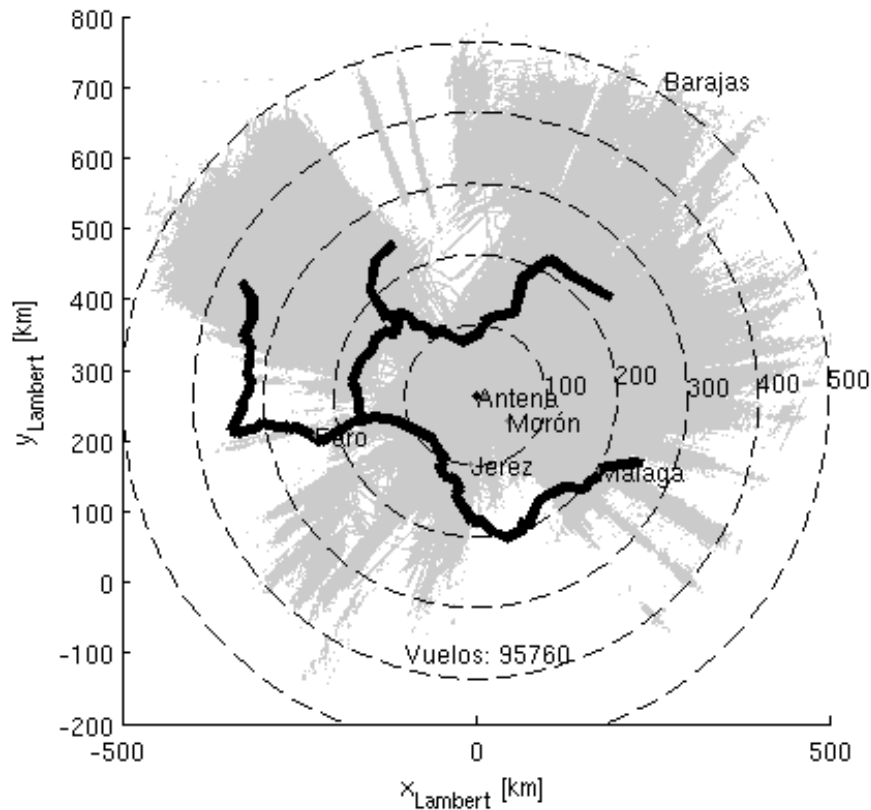


Figura 4.1: Perfiles horizontales de todos los vuelos.

Orden	Modelo	Nº de vuelos	Aerolínea	Nº de vuelos
1	Boeing 737-8	37375	Ryanair	17158
2	Airbus A320-2	21245	EasyJet	8430
3	Airbus A319-1	12552	Iberia	6009
4	Airbus A321-2	4380	Vueling	5765
5	Airbus A330-2	3345	Royal Air Maroc	5208
6	Boeing 737-7	2691	TAP Portugal	5168
7	Boeing 757-2	2064	Air Europa	3777
8	Airbus A340-3	1598	Transavia	3473
9	Boeing 737-3	1326	Monarch Airlines	2552
10	Airbus A340-6	1165	Jetairfly	2311
11	Boeing 767-3	1056	Air Berlin	2215
12	ATR 72-6	876	Air France	2151
13	Airbus A330-3	744	Thomson Airways	1917
14	Boeing 777-3	729	TUIfly	1839
15	Boeing 757-3	678	Jet2	1805
16	Fokker 100	605	Condor	1644
17	Boeing 777-2	449	Norwegian	1466
18	Airbus A321-1	444	Aer Lingus	1341
19	Boeing 737-4	346	Air Arabia Maroc	1168
20	Boeing 747-4	331	Lufthansa	1150

Tabla 4.1: Modelos de aeronaves y aerolíneas que más veces han sido registradas.

estas suponen un 4.5 % de los vuelos totales captados, habiéndose localizado 2177 despegues y 2176 aterrizajes. Estos valores tienen en cuenta todas aquellas aeronaves cuyas coordenadas al inicio del despegue o de la toma de tierra tienen sus coordenadas dentro del rectángulo de tolerancia explicado en la Sección 3.9, independientemente de la calidad de su señal.

En la Tabla 4.2 se muestra la distribución por cabeceras. Se observa que la cabecera 27 ha sido más utilizada que la 09, aproximadamente, el doble de operaciones, tanto en salidas como en llegadas.

	Cabecera 09	Cabecera 27	Total
Salidas	745 (34 %)	1432 (66 %)	2177
Llegadas	653 (30 %)	1523 (70 %)	2176

Tabla 4.2: Resumen de la distribución por cabeceras de las llegadas y salidas del aeropuerto de Sevilla.

4.1.1. Salidas

En total, se han identificado en el aeropuerto de Sevilla durante las 9 semanas de análisis, un total de 2177 salidas. El perfil horizontal de estos despegues se muestra en la Figura 4.2, observándose que existen 4 flujos principales, los cuales se dirigen hacia el Noroeste, Noreste, Este y Suroeste.

Los 5 modelos de aeronaves, así como las aerolíneas que más veces han salido del aeropuerto de Sevilla se muestran en la Tabla 4.3. Con gran diferencia destacan dos aeronaves: Boeing 737-8 (operados principalmente por Ryanair) y Airbus A320-2 (operados principalmente por Vueling), que totalizan 1974 despegues de los 2177 identificados (lo que supone un 90.6 %).

Orden	Modelo	Nº de salidas	Aerolínea	Nº de llegadas
1	Boeing 737-8	1027	Ryanair	821
2	Airbus A320-2	947	Vueling	785
3	Airbus A319-1	96	Iberia	147
4	Boeing 757-2	45	Transavia	100
5	Airbus A400M	9	Air Europa	73

Tabla 4.3: Modelos de aeronaves y aerolíneas que más veces han salido del aeropuerto de Sevilla.

En la Figura 4.3 se muestran los perfiles horizontales de las salidas por la cabecera 09 del aeropuerto de Sevilla, habiéndose superpuesto las rutas marcadas en la carta SID (incluida la de navegación aérea de precisión, P-RNAV, cuya salida finaliza en BUDIT) relativa a esta cabecera. Recordando los datos mostrados en la Tabla 4.2, el número total de salidas que utilizan esta cabecera es de 745, suponiendo un 34 % del total de las salidas registradas.

Como se puede apreciar, la salida por HIJ y la salida P-RNAV dirigida hacia BUDIT, son las más utilizadas, puesto que los vuelos más frecuentes se dirigen hacia el Norte de España y centro y Norte de Europa (salida de HIJ), además de a los países mediterráneos (salida de

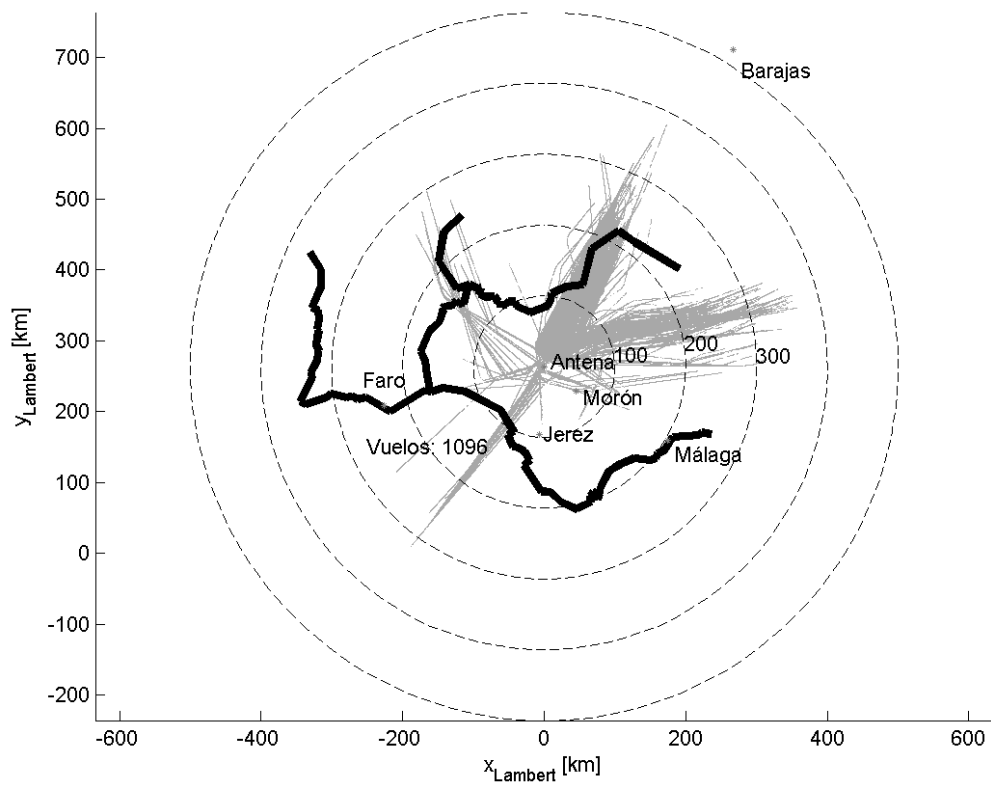


Figura 4.2: Perfil horizontal de las aeronaves que han salido del aeropuerto de Sevilla.

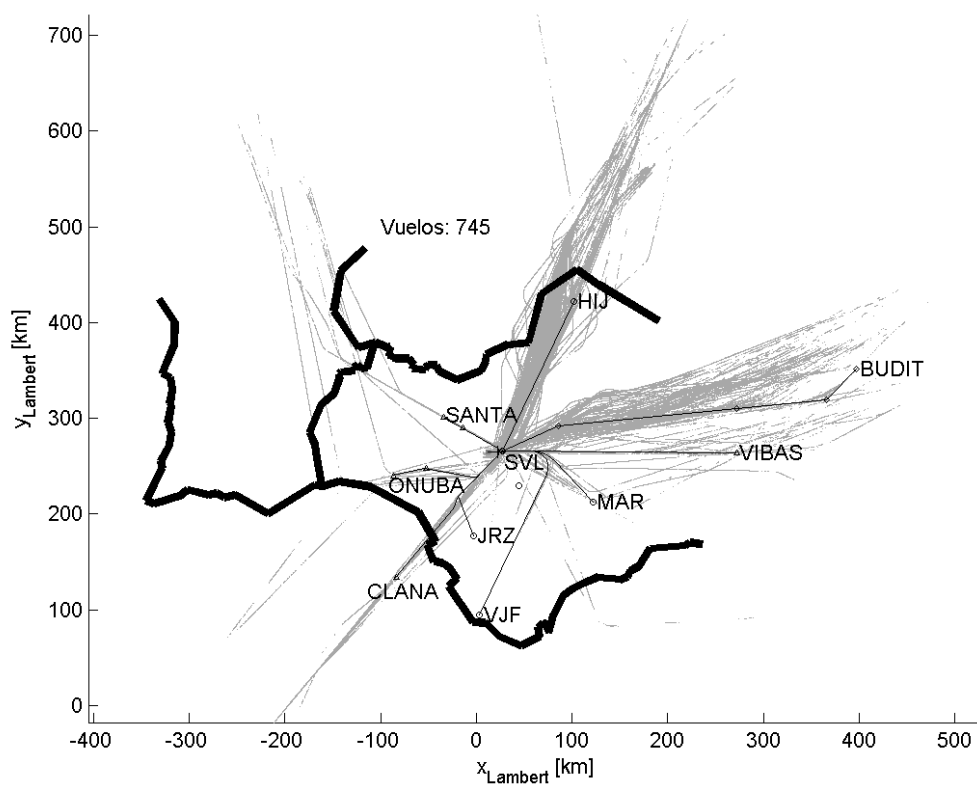


Figura 4.3: Perfil horizontal de las salidas por la cabecera 09 del aeropuerto de Sevilla.

BUDIT), como se aprecia en la Figura 4.17. En estas salidas, y en especial en la de BUDIT, se aprecia cómo muchos vuelos recortan la salida marcada por las cartas, llegando a alejarse de esta aproximadamente unos 50 km las aeronaves que más lo hacen.

En la Figura 4.4 se muestran las salidas por la cabecera 09, pero ampliando la representación en la zona cercana a la pista. En esta figura, se pueden apreciar proyecciones horizontales de

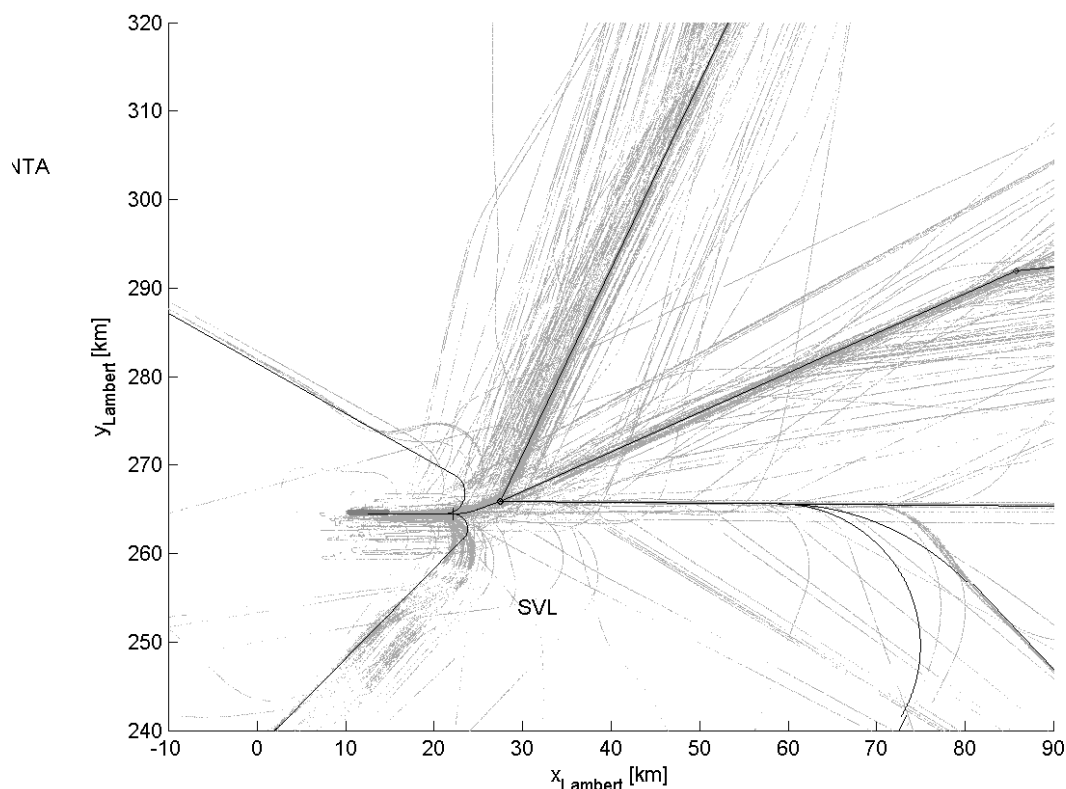


Figura 4.4: Perfil horizontal ampliado de las salidas por la cabecera 09 del aeropuerto de Sevilla.

despegues hasta 5 km al Sur de la pista (además de algunas al Norte) debido a errores cuyas causas se desconocen. El perfil horizontal de estas aeronaves sigue perfectamente la dirección de las calles de rodaje y la pista aunque se encuentre desplazada con respecto a estas. En la Figura 4.5 se muestra un ejemplo de este error.

Aunque muchos de los vuelos no siguen fielmente las rutas marcadas en las cartas, sí que la mayoría mantiene la dirección, aunque aparezcan algunas excepciones, como los vuelos que directamente alcanzan una dirección Sureste una vez que han finalizado el despegue.

En cuanto a la cabecera 27, en la Figura 4.6 se muestran los perfiles horizontales de las salidas por esta, habiéndose superpuesto las rutas marcadas en la carta SID (incluida la P-RNAV, la cual finaliza en BUDIT). En este caso, el número de vuelos que utilizan esta salida es de 1432, siendo prácticamente el doble de vuelos que utilizaron la salida 09. Al igual que ocurría con las salidas por la cabecera 09, las rutas más utilizadas en este caso son las de HIJ y BUDIT, apreciándose claramente recortes realizados en las trayectorias seguidas por los vuelos que utilizan las salidas de BUDIT, CLANA y VIBAS, destacando los recortes hacia esta última salida. En cuanto al grado de cumplimiento de las rutas marcadas en las cartas, se observa que

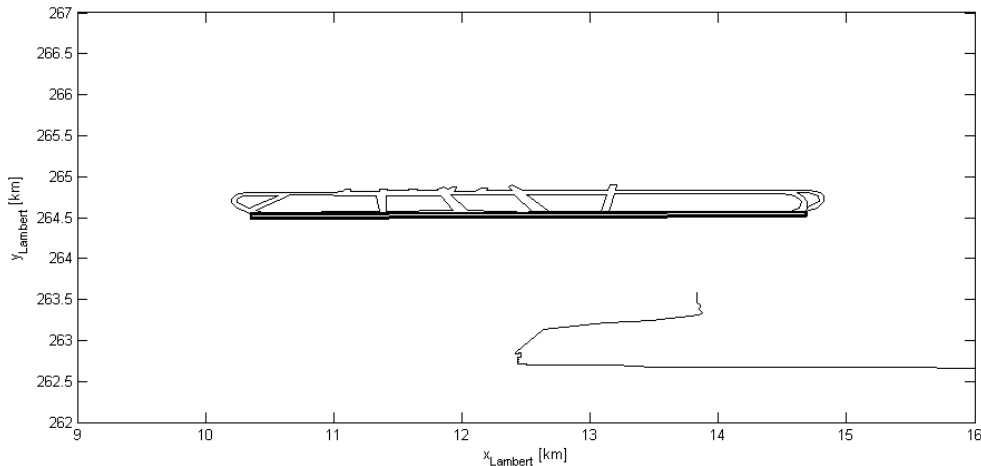


Figura 4.5: Ejemplo de desviación en la señal del GNSS durante un despegue.

las trayectorias voladas por las aeronaves presentan bastante dispersión respecto a las mismas. Caben destacar las salidas por BUDIT, que al ser P-RNAV, no deberían desviarse más de 1 NM (en el sentido perpendicular a la ruta) durante el 95 % del tiempo. Desviaciones mayores a esta cantidad podrían atribuirse a autorizaciones por parte de los controladores aéreos para modificar la ruta y así, por ejemplo, disminuir la distancia recorrida.

En la Figura 4.7 se muestran las salidas por la cabecera 27, ampliando la representación en la zona cercana a la pista.

En esta figura se puede observar que en todas las salidas hay aeronaves que recortan las rutas indicadas en las salidas estándar a fin de recorrer menos distancia. Destacan las salidas por HIJ, en las que las aeronaves abandonan el radial R-302.SVL mucho antes de alcanzar ARROS, y las salidas por BUDIT, que son P-RNAV, y también deberían llegar hasta ARROS.

4.1.2. Llegadas

Después de las 9 semanas de análisis, han terminado registrándose un total de 2176 llegadas. Como era de esperar, este valor es muy próximo al número registrado de despegues (2177). En la Figura 4.8 se muestran los perfiles horizontales de las aeronaves que han aterrizado en el aeropuerto de Sevilla durante estas 9 semanas. Se puede observar cómo aparecen 4 flujos parecidos a los observados en las salidas, con direcciones aproximadas hacia el Suroeste, Noroeste, Noreste y Este.

Los 5 modelos de aeronaves, así como las aerolíneas que más veces han aterrizado en el aeropuerto de Sevilla se muestran en la Tabla 4.4. Como cabría esperar, las aeronaves que más aterrizan son las que más despegan: B737-8 y A320-2, de Ryanair y Vueling respectivamente.

Al igual que se hizo con las salidas, también se han separado las llegadas según la cabecera utilizada. En la Figura 4.9 se muestran los perfiles horizontales de las llegadas por la cabecera 09, habiéndose superpuesto las rutas marcadas en la carta STAR (incluidas las de P-RNAV con origen en MAMIS y BAILÉN dirección Córdoba) relativa a esta cabecera.

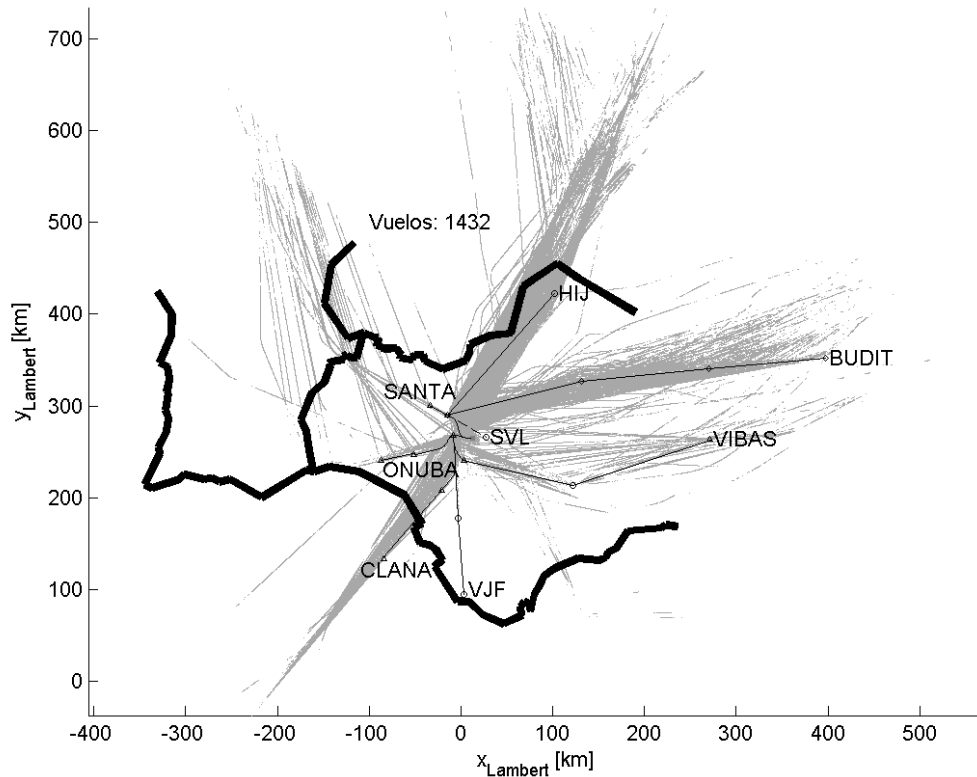


Figura 4.6: Perfil horizontal de las salidas por la cabecera 27 del aeropuerto de Sevilla.

Orden	Modelo	Nº de llegadas	Aerolínea	Nº de llegadas
1	Boeing 737-8	1033	Ryanair	824
2	Airbus A320-2	942	Vueling	782
3	Airbus A319-1	96	Iberia	145
4	Boeing 757-2	44	Transavia	101
5	Airbus A400M	9	Air Europa	75

Tabla 4.4: Modelos de aeronaves y aerolíneas que más veces han aterrizado en el aeropuerto de Sevilla.

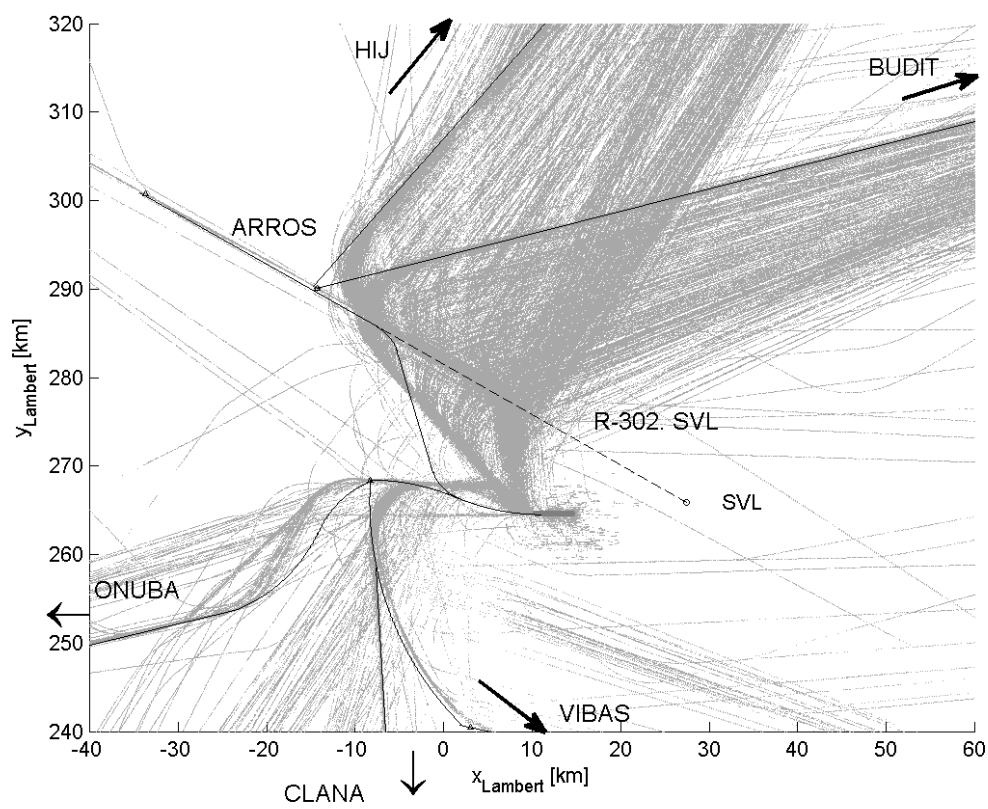


Figura 4.7: Perfil horizontal ampliado de las salidas por la cabecera 27 del aeropuerto de Sevilla.

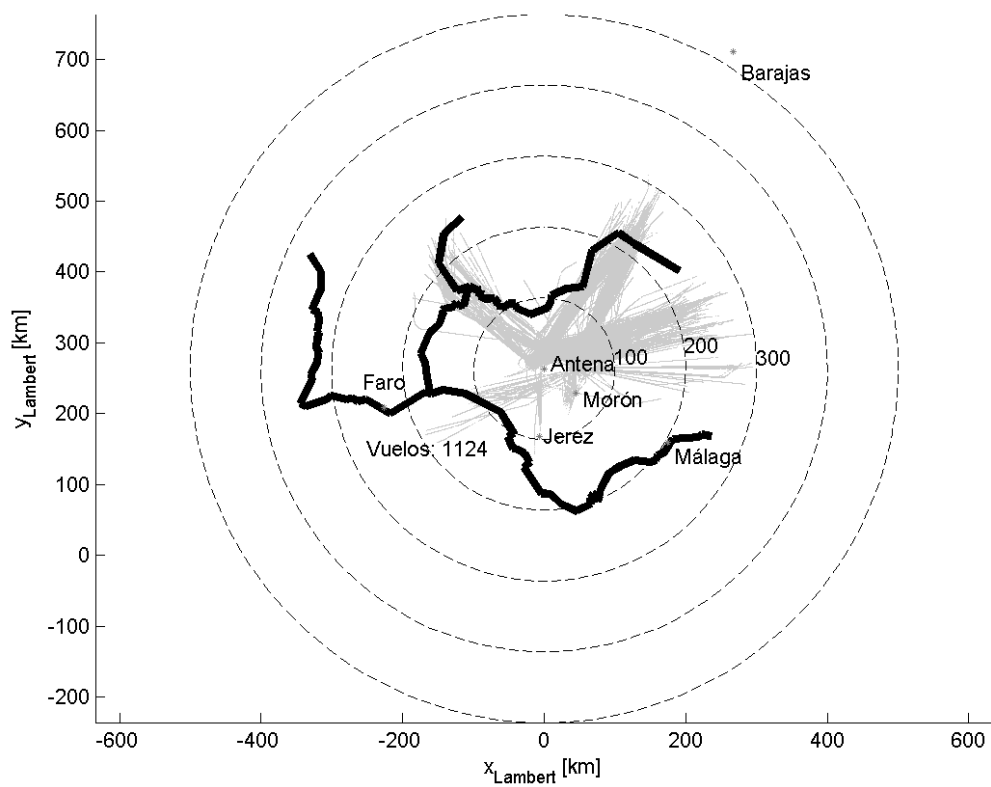


Figura 4.8: Perfil horizontal de las aeronaves que han llegado al aeropuerto de Sevilla.

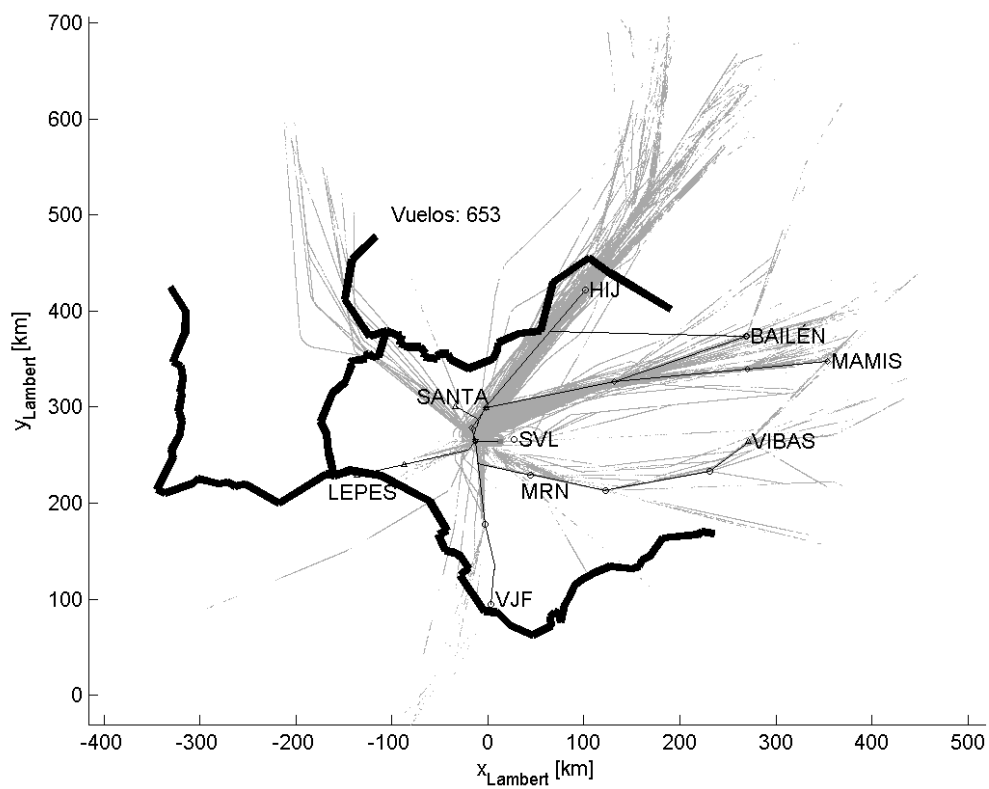


Figura 4.9: Perfil horizontal de las llegadas por la cabecera 09 del aeropuerto de Sevilla.

El número de vuelos que hicieron uso de esta cabecera fue 653, los cuales suponen un 30 % de todas las llegadas al aeropuerto de Sevilla. Como se puede observar, algunas de las llegadas procedentes de VIBAS recortan su trayectoria directamente hacia la pista del aeropuerto, sin seguir en ningún momento la llegada marcada por las cartas.

En la Figura 4.10 se muestran las llegadas por la cabecera 09, ampliando la representación en la zona cercana a la pista para así apreciar el grado de acercamiento al IAF TENDU. En

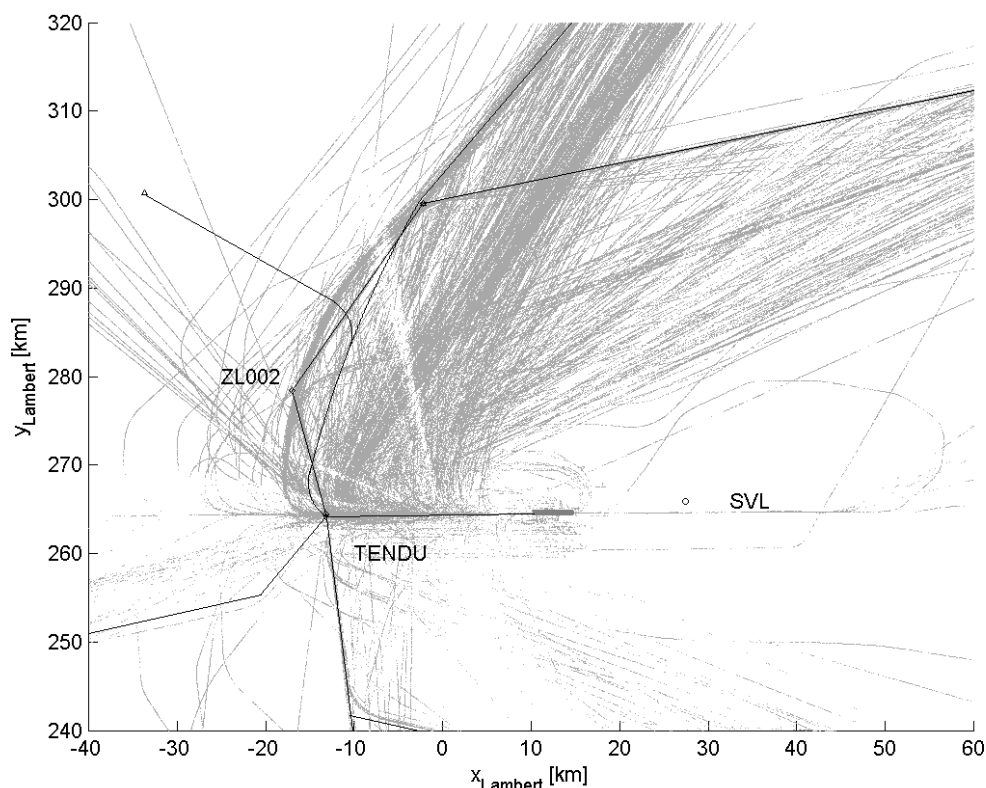


Figura 4.10: Perfil horizontal ampliado de las llegadas por la cabecera 09 del aeropuerto de Sevilla.

primer lugar, puede apreciarse cómo los vuelos que provienen del Norte siguiendo la ruta P-RNAV realizan un viraje de tal manera que sobrevuelan los dos puntos de paso que se encuentran definidos (ZL002 en la P-RNAV y el IAF TENDU) con una cierta precisión mientras que los vuelos que vienen siguiendo la ruta normal se alejan claramente de esta, encarando la pista antes del IAF recorriendo así menor distancia.

Con respecto a la cabecera 27, en la Figura 4.11 se muestran los perfiles horizontales de las llegadas a esta cabecera, habiéndose superpuesto las rutas marcadas en la carta STAR (incluida la de P-RNAV con origen en MAMIS) relativa a esta cabecera.

El número de vuelos que hicieron uso de esta cabecera fue 1523, los cuales suponen un 70 % de todas las llegadas al aeropuerto de Sevilla, apreciándose en este caso 5 llegadas principales (VIBAS, BAILÉN Y MAMIS, HIJ, SANTA Y LEPES).

En la Figura 4.12 se muestran las llegadas por la cabecera 27, ampliando la representación

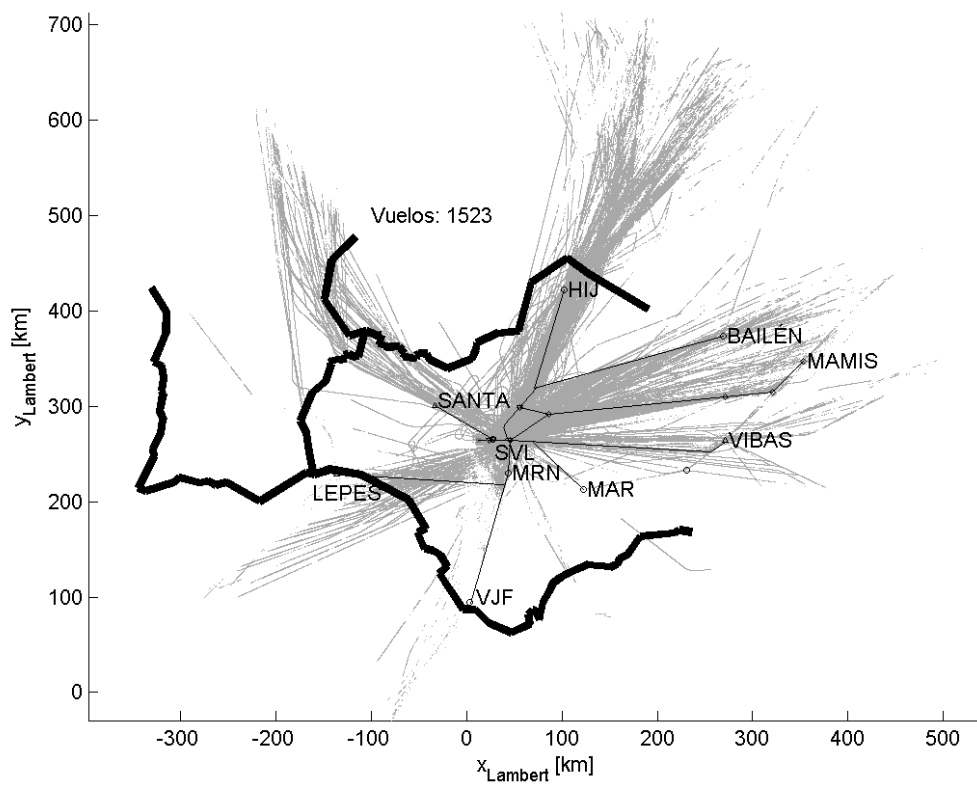


Figura 4.11: Perfil horizontal de las llegadas por la cabecera 27 del aeropuerto de Sevilla.

en la zona cercana a la pista. Como se puede apreciar en la figura, aparecen muchas llegadas

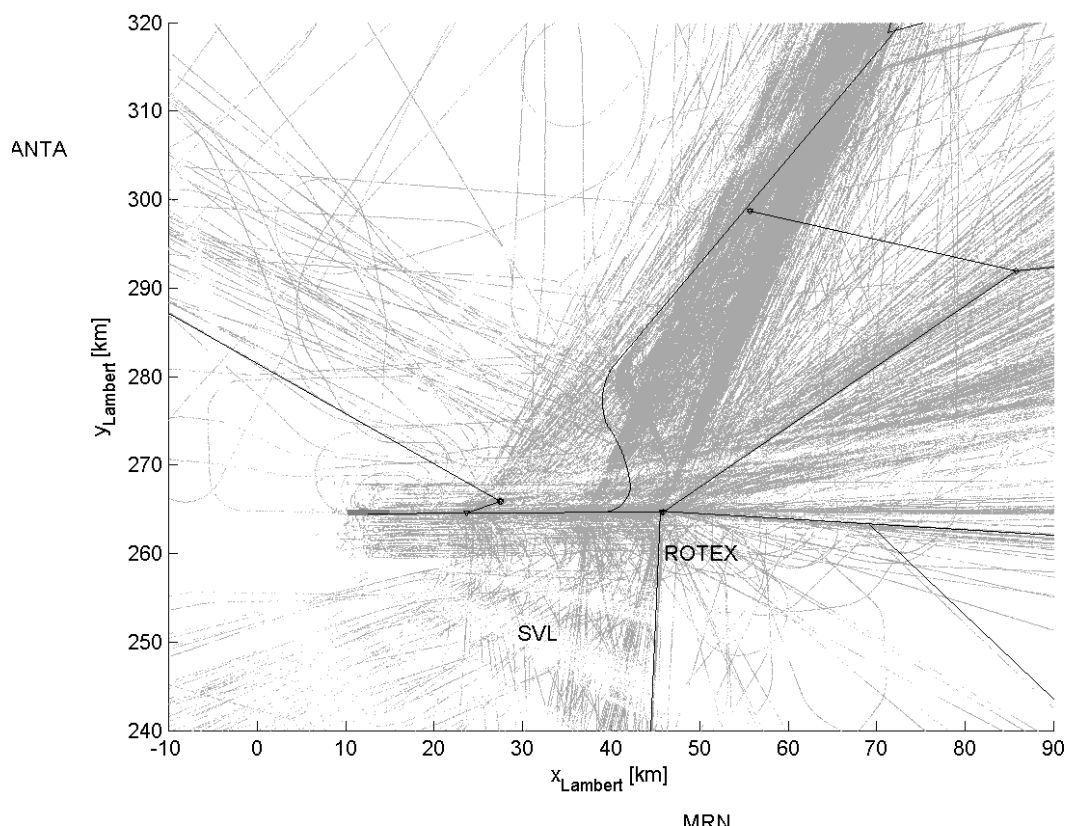


Figura 4.12: Perfil horizontal ampliado de las llegadas por la cabecera 27 del aeropuerto de Sevilla.

paralelas a la pista, presentando algunas una diferencia de hasta 5 km con esta debido a errores en el sistema de localización. Por otro lado, se observa cómo la mayoría de los vuelos que provienen del Norte buscando el IAF ROTEX, recortan directamente, evitando el doble viraje definido para alcanzar la orientación de la pista.

4.2. Perfil vertical

A continuación se representarán los perfiles verticales de las salidas y llegadas realizados en el aeropuerto de Sevilla cuya calidad de señal cumpla los mínimos requeridos en este proyecto. De esta manera, partiendo de las salidas y llegadas anteriores, se extraen los tramos entre 4000 y 35500 ft en el caso de las salidas y los tramos entre 3000 y 32000 ft en el caso de las llegadas, y se descartan aquellos con una diferencia máxima entre dos datos de 30 segundos, o una media entre datos mayor que 10 segundos, por las razones descritas en la Sección 3.13.

Tanto en salidas como en llegadas, los modelos más utilizados son el Boeing 737-8 y el Airbus A320-2. Estos dos modelos resultan ser similares entre sí, pues comparando dos de las características más representativas, se tiene que el MTOW (*Maximum Take-off Weight*) y el

empuje de ambos es 79010^2 kg y 78000^3 kg, y 240.2 kN y 240 kN (usan el prácticamente el mismo motor), respectivamente.

En esta sección se realizarán análisis atendiendo a los modelos utilizados, los orígenes/destinos de los vuelos y casos particulares como pueden ser dos modelos para un mismo destino o de tres destinos para un mismo modelo.

4.2.1. Salidas

A partir de las salidas anteriores, se obtuvieron las que mejor señal presentaban con el fin de analizarlas más detalladamente. En este caso se han obtenido 1096 salidas, lo que supone un 50 % de las salidas totales. Los 5 modelos de aeronaves, así como las aerolíneas que más veces han salido del aeropuerto de Sevilla con buena calidad de señal, se muestran en la Tabla 4.5.

Orden	Modelo	Nº de salidas	Aerolínea	Nº de salidas
1	Boeing 737-8	507	Ryanair	443
2	Airbus A320-2	475	Vueling	353
3	Airbus A319-1	57	Iberia	120
4	Boeing 757-2	25	Transavia	41
5	Airbus A300-6	4	Air France	29

Tabla 4.5: Modelos de aeronaves y aerolíneas que más veces han salido del aeropuerto de Sevilla con buena calidad de señal.

Los Boeing 737-8 han realizado un total de 507 salidas del aeropuerto de Sevilla, siendo este número cercano al de salidas realizadas por los Airbus A320-2. En cuanto a las aerolíneas, destacan Ryanair y Vueling.

En la Figura 4.13 se muestran los perfiles verticales de todas las salidas registradas en el aeropuerto de Sevilla, cuya señal se ha considerado lo suficientemente buena. Salvo alguna que otra excepción, el grueso de los vuelos que terminan el tramo de ascenso por encima de los 35500 ft de altitud, necesitan recorrer una distancia entre 156 y 267 km. A continuación se analizarán estos perfiles más detalladamente.

4.2.1.1. Comparación entre modelos de avión

En este caso y como se recoge en la Tabla 4.5, los modelos más utilizados son el Boeing 737-8, el cual aparece en 507 ocasiones y el Airbus A320-2, el cual aparece en 475 ocasiones.

En la Figura 4.14 se muestran los perfiles verticales de todos los despegues y resaltados en azul, los perfiles verticales de todos los Boeing 737-8. Por su parte, en la Figura 4.15 se muestran los perfiles verticales de todos los despegues y resaltados en azul, los perfiles verticales de todos los Airbus A320-2. Por lo general, el Boeing 737-8 alcanza los 35500 ft habiendo recorrido una

²http://www.boeing.com/boeing/commercial/737family/pf/pf_800tech.page (17/09/2013).

³<http://www.airbus.com/aircraftfamilies/passengeraircraft/a320family/a320/specifications/> (17/09/2013).

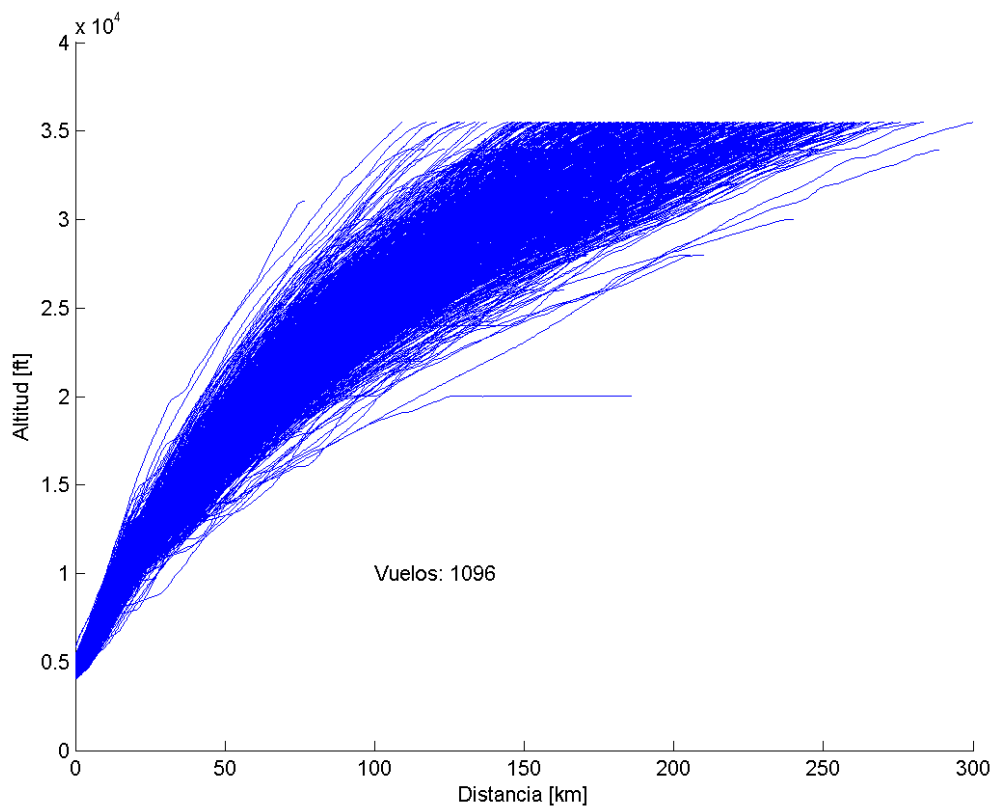


Figura 4.13: Perfiles verticales de todas las salidas del aeropuerto de Sevilla con buena calidad de señal.

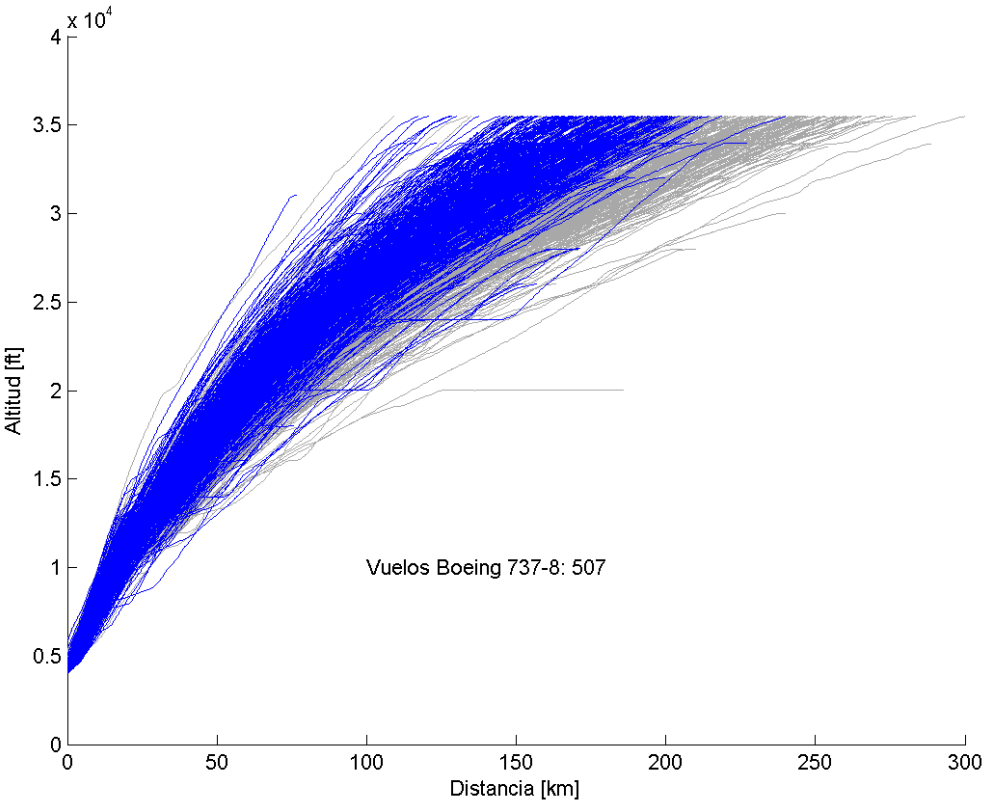


Figura 4.14: Perfiles verticales de las salidas, resaltando el modelo más común.

distancia entre los 145 y los 210 km, mientras que el Airbus A320-2 necesita una distancia mayor, comprendida entre 185 y 265 km.

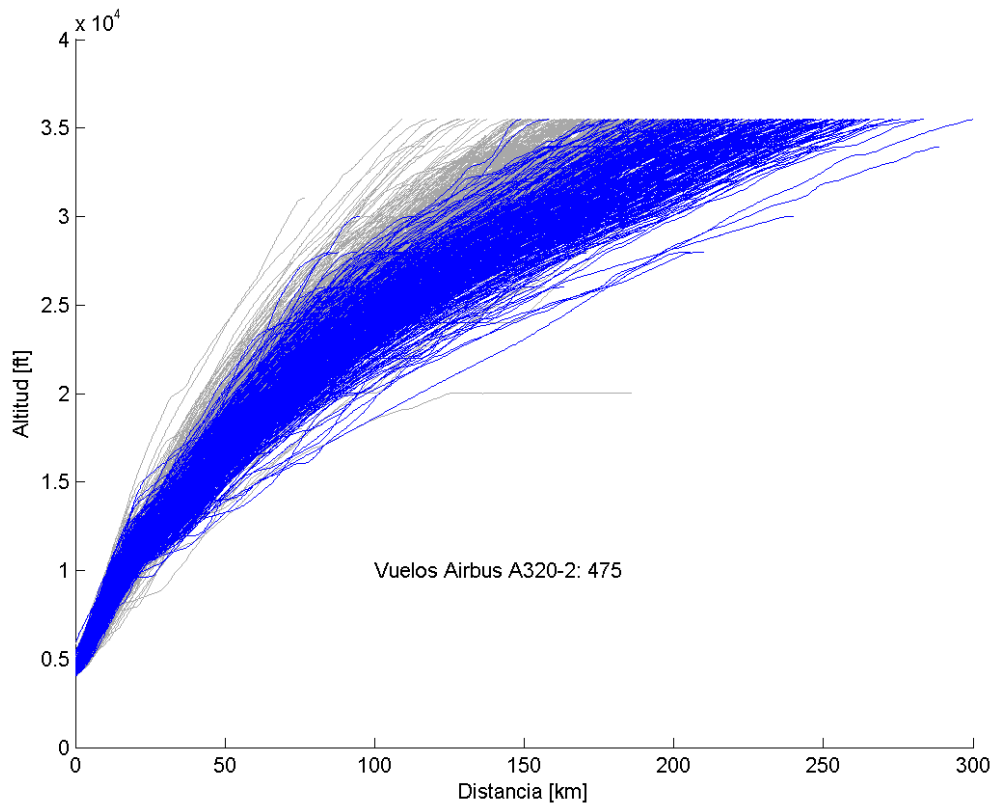


Figura 4.15: Perfiles verticales de las salidas, resaltando el segundo modelo más común.

A simple vista puede apreciarse cómo el Boeing 737-8 asciende con un gradiente mayor que el Airbus A320-2. De todas formas, resulta interesante compararlos en la misma representación. Para esto se ha decidido representar los percentiles 2.5 y 97.5 de la altitud de todos los vuelos del mismo modelo, calculados cada 1 km. Para el cálculo de los percentiles se ha utilizado la función *prctl* de MATLAB.

En la Figura 4.16 se muestran los perfiles verticales de todos los despegues y las curvas correspondientes a los percentiles 2.5 y 97.5 para los dos modelos de aeronaves más comunes. Al buscar un resultado más cualitativo que cuantitativo, se ha decidido no calcular los percentiles en todo el tramo de subida, puesto que al final de este, la escasez de puntos haría obtener un resultado poco relevante. De esta manera, se ha establecido como límite 200 km. Se representan en rojo los percentiles correspondientes a los Boeing 737-8 y en azul a los Airbus A320-2. Es importante remarcar que estos resultados corresponden a las salidas realizadas desde Sevilla, por lo que dependen directamente de los vuelos que se ofertan en este aeropuerto. Este mismo análisis realizado en otro aeropuerto podría arrojar resultados diferentes.

Se puede apreciar cómo el perfil vertical de ambos es muy parecido en los primeros 25 km, ascendiendo en general con un gradiente mayor el Boeing 737-8 una vez que han recorrido esta distancia. Por compararlos cuantitativamente, los percentiles 2.5 y 97.5 del Boeing 737-8 alcanzan un valor de 23550 y 30160 ft respectivamente, una vez se han recorrido 100 km,

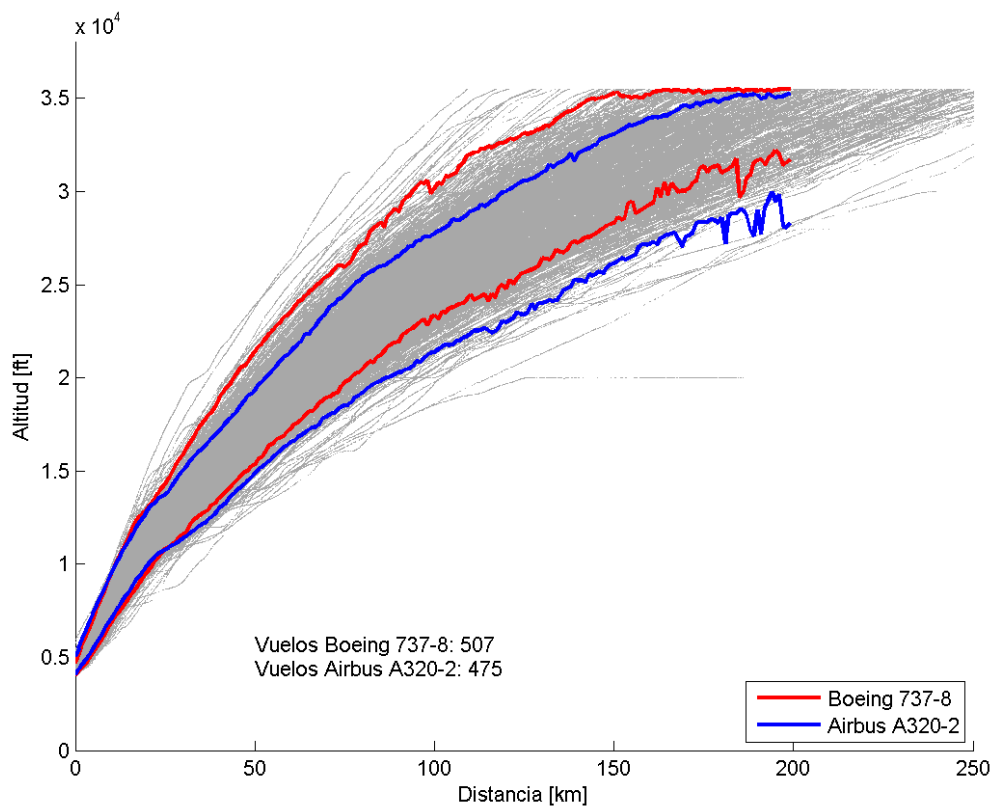


Figura 4.16: Perfiles verticales de las salidas, comparando los dos modelos más comunes.

mientras que por su parte, los mismos percentiles del Airbus A320-2 se encuentran a 21380 y 27730 ft.

4.2.1.2. Distribución por destinos

Uno de los datos enviados por las aeronaves mediante ADS-B es el CallSign, el cual indica la identificación asignada al vuelo que se encuentra realizando la aeronave y que, según se ha observado durante la realización del proyecto, en la mayoría de los casos no cambia para un mismo origen y destino.

Para realizar el análisis según los orígenes o destinos, se han clasificado todos los vuelos (llegadas y salidas, previos al filtro de calidad de señal) realizados hacia/desde el aeropuerto de Sevilla según el CallSign, obteniendo 161 call sign distintos en salidas y 148 call sign distintos en llegadas.

Para realizar la asignación de origen/destino según el call sign, se ha creado una pequeña base de datos con los datos obtenidos de *planefinder*⁴ y *Flightradar24*⁵, relacionando con su correspondiente origen/destino aquellos call sign en los que:

1. se ha comprobado que durante el periodo de captura de datos solo fueron utilizados para caracterizar un mismo vuelo; y
2. aquellos que aun no cubriendo todo el periodo de captura de datos, sí que cubren parte de este, no están asociados a otro vuelo y la fuente consultada lo tiene directamente asociado a un vuelo. Por este motivo, hay algunos CallSign con un volumen de tráfico considerable que por no tener la certeza del vuelo al que pertenecen, no se han tenido en cuenta en este apartado.

Las posibles estadísticas extraídas no son concluyentes en cuanto al reparto de tráfico en el aeropuerto de Sevilla, aunque sí que resultan útiles para dilucidar cuáles son aquellos orígenes/destinos más comunes y así poder realizar análisis comparativos según el origen/destino. Uno de los vuelos más afectados por esta falta de certeza se cree que es el vuelo Sevilla-Londres (y Londres-Sevilla), el cual parece estar asociado con varios CallSign con un cierto volumen de tráfico, aunque esta afirmación no se ha podido verificar.

En la Figura 4.17 se muestran los aeropuertos con los que el de Sevilla tiene conexión, representando el volumen del tráfico obtenido en los despegues con el espesor y el color de las líneas de conexión, y el diámetro de los círculos que posicionan cada aeropuerto. Se recuerda que no se corresponde con el total de despegues (pues no se han conseguido asociar todos los call sign), aunque sí con un gran porcentaje (84.4 %). Se aprecia cómo es Barcelona el destino al que más vuelos se dirigen, además de otros con cierta importancia como pueden ser Canarias, Baleares y París. Los destinos más alejados se encuentran aproximadamente en una distancia parecida, siendo estos los aeropuertos italianos, británicos o centroeuropeos.

⁴<http://planefinder.net/> (14/09/2013).

⁵www.flightradar24.com (14/09/2013).

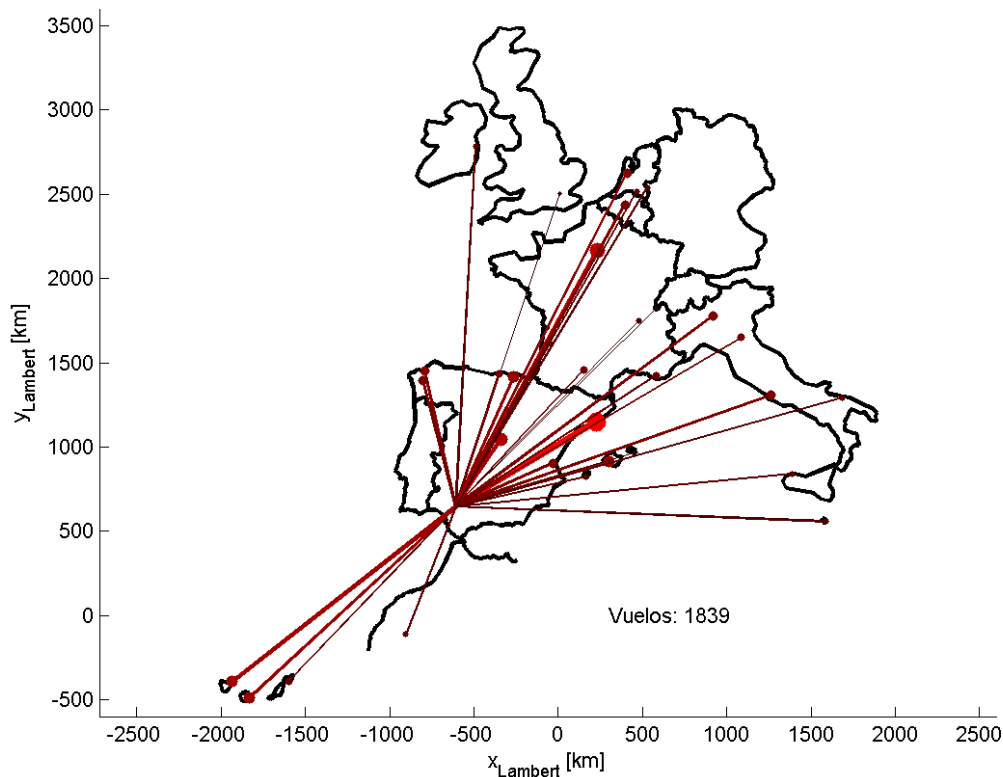


Figura 4.17: Mapa de Europa en el que se muestran los destinos del aeropuerto de Sevilla según el volumen de tráfico.

Aunque las estadísticas mostradas en cuanto a número de vuelos a cada destino se refiera al total de salidas/llegadas, las estadísticas presentadas por los vuelos que han pasado el filtro de calidad de señal no son muy distintas, salvo la parte proporcional que les corresponde en cuanto a número de vuelos. Los 10 destinos más frecuentes del aeropuerto de Sevilla y la distancia que debe recorrer el vuelo, medida a lo largo de la ruta ortodrómica que une ambos aeropuertos, se muestran en la Tabla 4.6.

Con diferencia, el destino más común es Barcelona, siendo París el único destino internacional destacado entre los destinos más habituales. Observando también la relación de distancias a la que se encuentran los destinos habituales, se pueden clasificar los destinos en tres grupos:

- **Distancias pequeñas:** se trata de aquellos vuelos en los que la distancia a recorrer es suficientemente pequeña como para que los vuelos alcancen un nivel de crucero a una altitud relativamente baja. En este grupo se incluiría únicamente a Madrid (214 NM) entre los vuelos registrados.
- **Distancias medias:** esta denominación agruparía al resto de destinos nacionales exceptuando los aeropuertos de Canarias, como pueden ser Barcelona o Santiago. Se encuentran aproximadamente a 400 NM y para este tipo de vuelos sí que resulta necesario que los vuelos alcancen altitudes mayores a los 30000 ft, lo que ocurre en la mayoría de los casos.
- **Distancias grandes:** agrupa al resto de vuelos, en su mayoría internacionales, aunque también se podría situar aquí los aeropuertos canarios. Las distancias van desde las

Orden	Destino	Repeticiones	Distancia [NM]
1	Barcelona	481	437
2	París	189	783
3	Madrid	138	214
4	Tenerife	111	750
5	Palma de Mallorca	95	426
6	Gran Canaria	92	743
7	Bilbao	83	379
8	Amsterdam	63	1001
9	A Coruña	59	371
10	Milán	50	858

Tabla 4.6: Destinos más frecuentes en el aeropuerto de Sevilla.

743 NM de Gran Canaria o las 783 NM de París hasta las 1001 NM de Amsterdam o las 1077 NM de Bari.

4.2.1.3. Comparación entre modelos, para un mismo destino (Barcelona)

El principal objetivo de este apartado es diferenciar de la manera más exacta posible los ascensos de dos modelos de aeronaves distintos. Para esto, uno de los factores a tener en cuenta es el destino, pues este marcará en gran medida el peso de la aeronave al despegue y los niveles de vuelo que buscará alcanzar. En este caso se ha elegido como destino Barcelona por ser el destino que mayor volumen de tráfico presenta y así poder contar con un número de vuelos significativos, además de que aparecen vuelos de los dos modelos más utilizados en las salidas del aeropuerto de Sevilla, el Boeing 737-8 y el Airbus A320-2.

En la Figura 4.18 se muestran los perfiles verticales de todas las salidas con destino Barcelona, y superpuesto en color azul, aquellas salidas realizadas con el Airbus A320-2. En el caso concreto del aeropuerto de Sevilla, se realizan más salidas hacia Barcelona con el Airbus A320-2, suponiendo las salidas de este un 72 % del total de vuelos con este destino. Como se puede observar, los perfiles verticales del Airbus A320-2 ocupan la mitad baja de la sombra gris que representa el total de los vuelos con destino Barcelona. Además, puede apreciarse también cómo es habitual que los ascensos realizados con esta aeronave alcanzan niveles de crucero por encima de los 35500 ft. El grueso de los ascensos de esta aeronave alcanza los 35500 ft una vez han recorrido una distancia comprendida entre los 200 y los 270 km.

Por su parte, los perfiles verticales de los Boeing 737-8 con destino a Barcelona se presentan en la Figura 4.19, encontrándose resaltados en azul y superpuestos al total de las salidas hacia Barcelona. En este caso, los perfiles verticales ocupan claramente la mitad superior de la sombra que representa el total de los vuelos con destino a Barcelona. Al igual que se diferenciaba en el caso anterior, se puede distinguir un nivel de vuelo de crucero preferente para los despegues de esta aeronave, situándose en 34000 ft y alcanzándose al recorrer una distancia comprendida entre los 146 y los 170 km.

Con el fin de cuantificar de alguna manera la diferencia entre los ascensos de ambas aeronaves

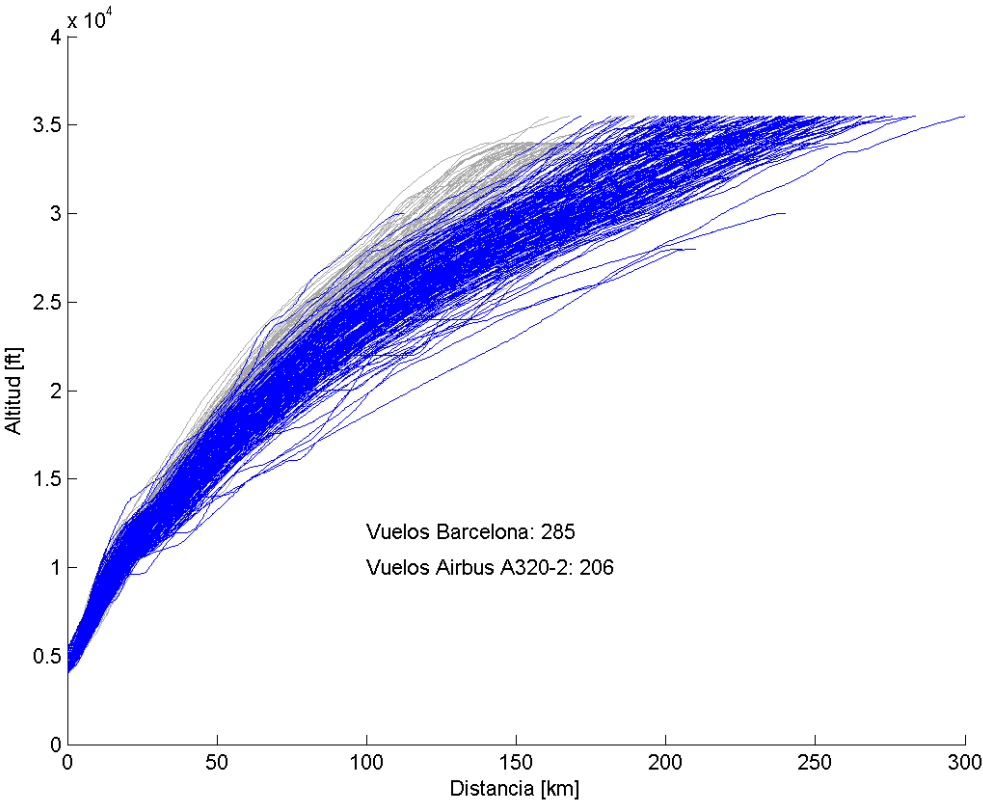


Figura 4.18: Perfiles verticales de los Airbus A320-2 con destino a Barcelona.

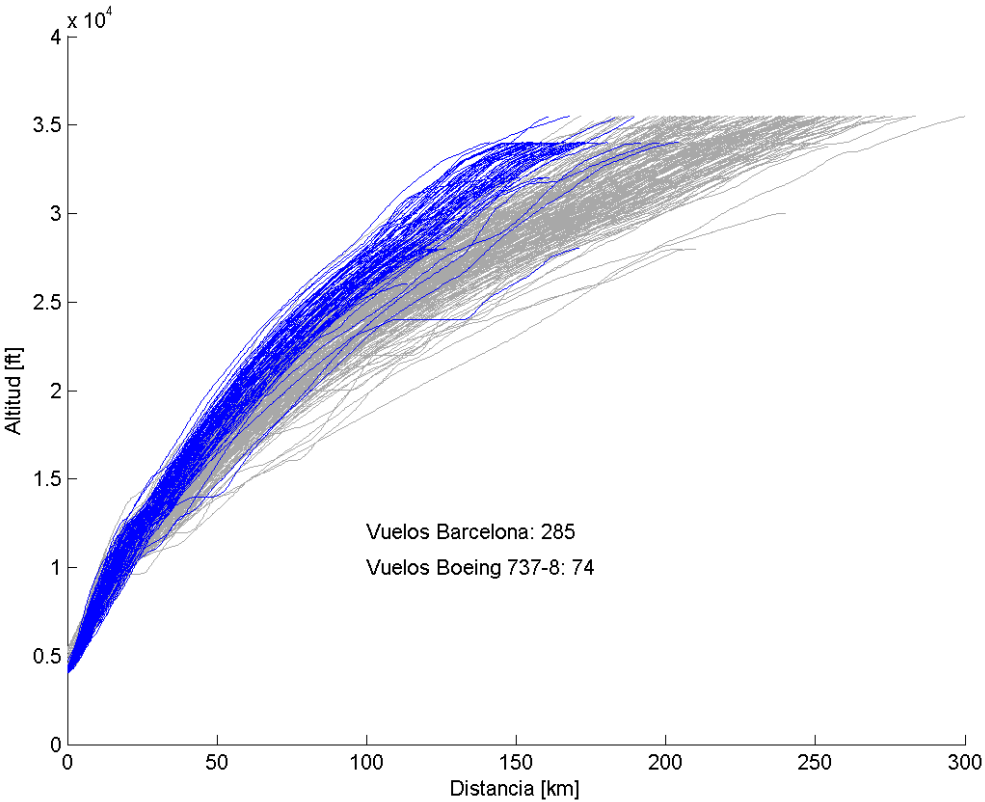


Figura 4.19: Perfiles verticales de los Boeing 737-8 con destino a Barcelona.

siendo Barcelona su destino, se ha decidido representar los percentiles 2.5 y 97.5 del ascenso de ambas aeronaves, utilizando en este caso rangos de distancia de 2 km debido a que el volumen de tráfico de los Boeing 737-8 no es suficiente para evitar un rizado excesivo en la parte final del ascenso. La representación de estos percentiles se muestra en la Figura 4.20. Se

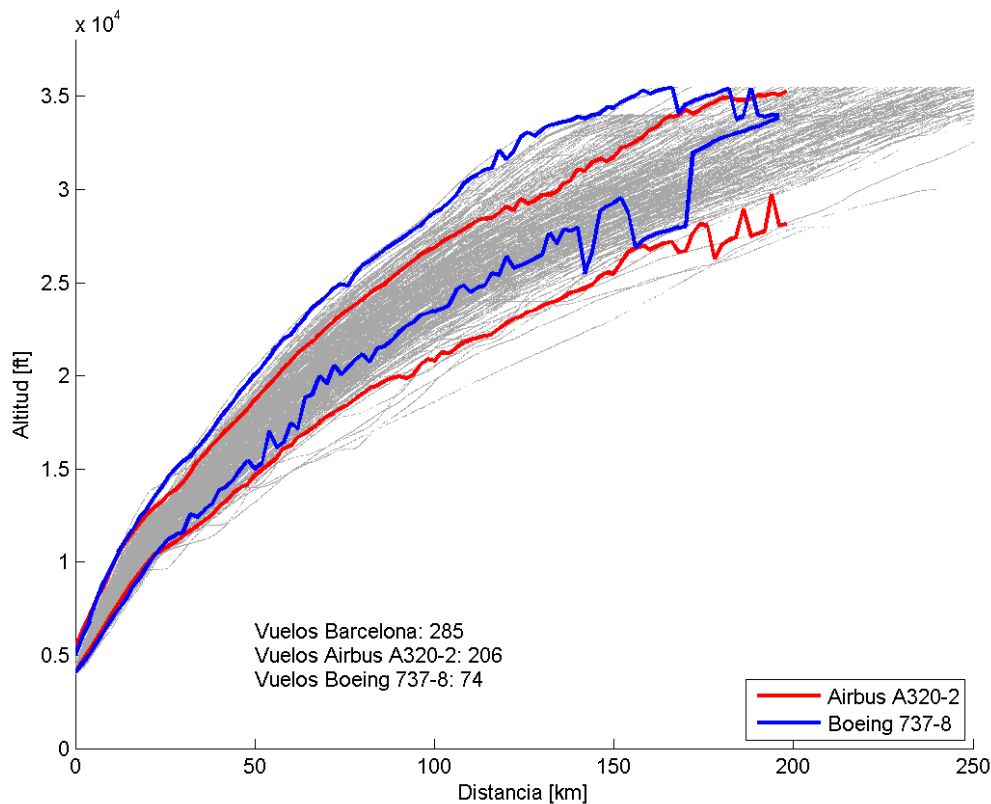


Figura 4.20: Perfiles verticales de los vuelos con destino a Barcelona y los percentiles 2.5 y 97.5 correspondientes a los dos modelos más comunes.

aprecia claramente los mayores gradientes de ascenso utilizados por los Boeing 737-8, aunque se mantengan iguales en los primeros 25 km de ascenso aproximadamente. El rizado mostrado por la curva correspondiente al Boeing 737-8 al final del ascenso se debe a que la mayoría de los vuelos han alcanzado ya el nivel de crucero y por tanto desaparecen de la representación, pudiéndose apreciar cómo el mayor escalón que presenta se sitúa cercano a los 160 km, como se comentó anteriormente. En cuanto a las altitudes alcanzadas por los percentiles 2.5 y 97.5 al haber recorrido 100 km, los correspondientes al Airbus A320-2 se encuentran en 20780 y 26880 ft respectivamente. Por su parte, los correspondientes al Boeing 738-8 alcanzan unas altitudes de 23640 y 28770 ft.

De esta manera, puede afirmarse que las salidas con destino a Barcelona realizadas por el Boeing 737-8 alcanzan el nivel de crucero aproximadamente 75 km antes que las salidas de los Airbus A320-2 debido a dos razones principales: los Boeing 737-8 tienen establecido el nivel de vuelo de crucero (34000 ft) más bajo que los Airbus A320-2 (>35500 ft) y además, los Boeing 737-8 alcanzan este nivel de vuelo con un gradiente de ascenso mayor.

4.2.1.4. Comparación entre destinos, para un mismo modelo (Airbus A320-2)

A continuación se estudiarán los perfiles verticales seguidos por el Airbus A320-2 para distintos destinos. Separando las salidas a cada destino por la aeronave que las realiza, permitirá sacar conclusiones sobre cómo afecta el destino elegido a la hora de realizar un ascenso. Los destinos elegidos son los 3 más comunes en las salidas, en este caso, Barcelona, Madrid y París, los cuales cuentan con 206, 114 y 49 salidas respectivamente de un total de 475 despegues realizados por el Airbus A320-2, suponiendo un 43 %, un 24 % y un 10 % respectivamente. Además, estos tres destinos representan los tipos de destinos definidos anteriormente según la distancia (pequeña, mediana o grande) a la que se encuentren del aeropuerto de Sevilla.

En la Figura 4.21 se muestran los perfiles verticales de todos los vuelos realizados por el Airbus A320-2 y resaltados en azul, los correspondientes a Barcelona. El gran volumen de

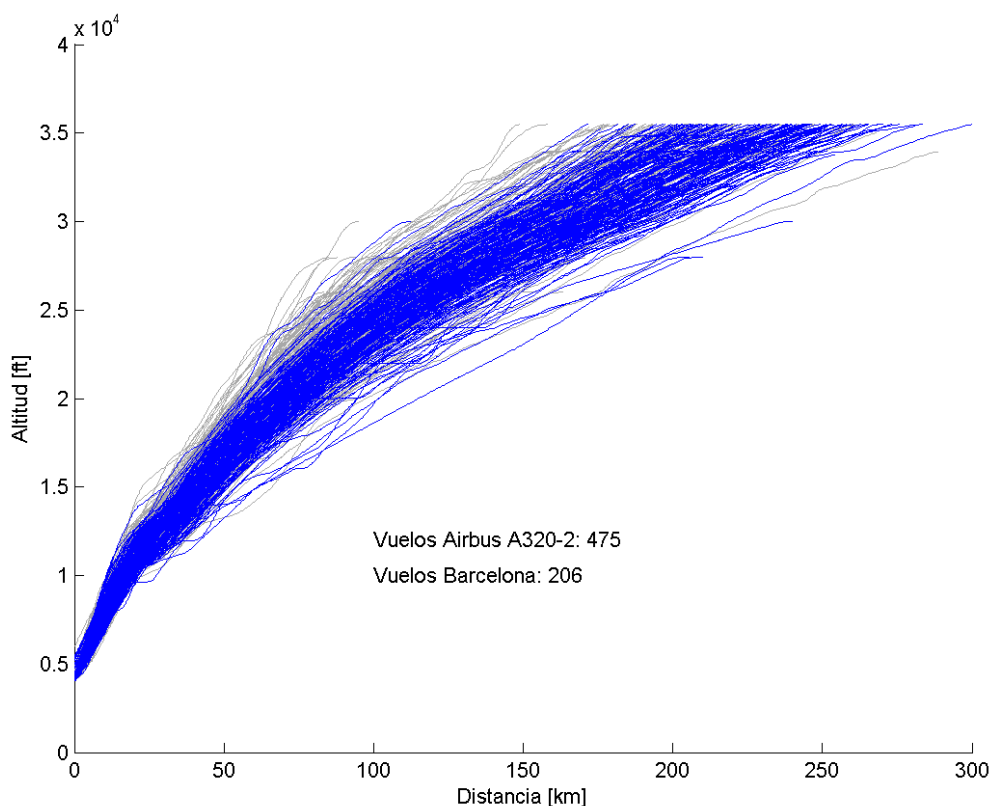


Figura 4.21: Perfiles verticales de los vuelos realizados por el Airbus A320-2 con destino a Barcelona.

tráfico que supone este destino hace que los perfiles verticales con destino a Barcelona ocupen casi por completo la sombra representativa de todos los vuelos del Airbus A320-2. Aun así, puede apreciarse cómo las salidas con este destino están en su mayoría por la parte baja de la sombra. Por lo general, los ascensos que alcanzan los 35500 ft necesitan una distancia comprendida entre los 197 y los 267 km.

En la Figura 4.22 se muestran los perfiles verticales de todos los vuelos realizados por el Airbus A320-2 y resaltados en azul, los correspondientes a Madrid. Los perfiles verticales

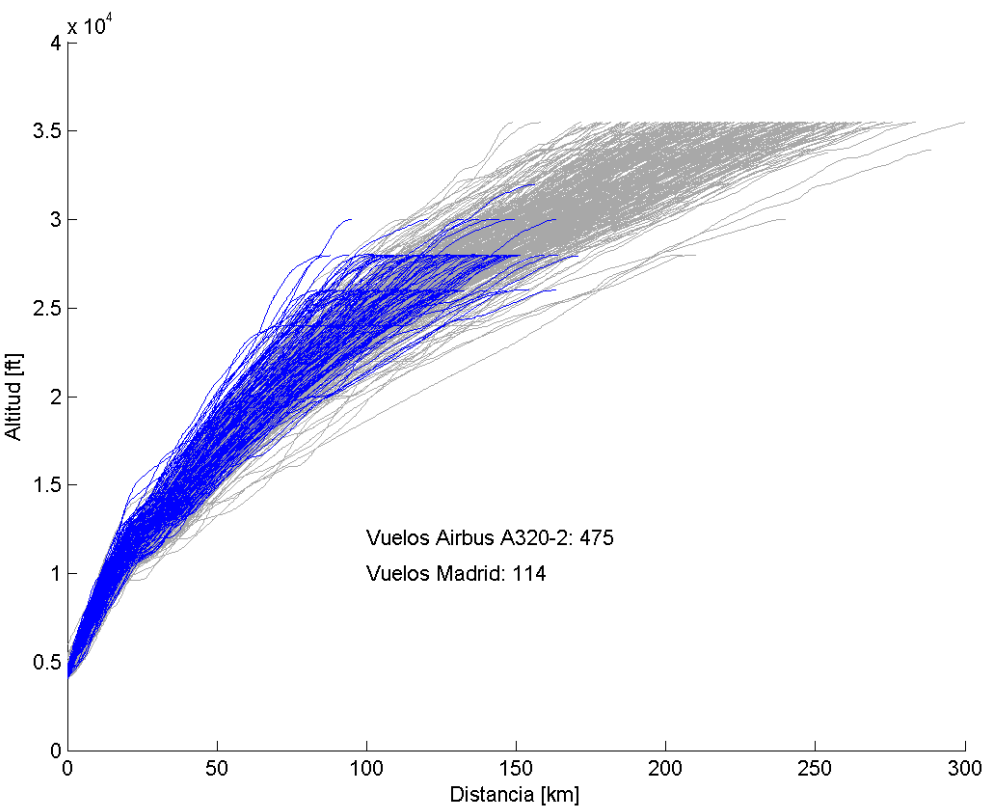


Figura 4.22: Perfiles verticales de los vuelos realizados por el Airbus A320-2 con destino a Madrid.

relativos a Madrid resultan muy característicos. En primer lugar, es el segundo destino más común (114 vuelos que hayan pasado el filtro de señal), y en segundo lugar, se trata del destino más cercano con diferencia (245 NM), y como se puede apreciar, los vuelos alcanzan el nivel de vuelo de crucero muy pronto, encontrándose este a 24000, 26000 ó 28000 ft en la mayoría de los casos. En cuanto al gradiente de ascenso utilizado, no aparece ningún rasgo característico cuantificable en esta representación. El grueso de los ascensos que alcanzan el nivel de vuelo situado a 28000 ft necesitan recorrer una distancia entre 97 y 148 km.

En la Figura 4.23 se muestran los perfiles verticales de todos los vuelos realizados por el Airbus A320-2 y resaltados en azul, aquellos cuyo destino es París. En este caso, el número de

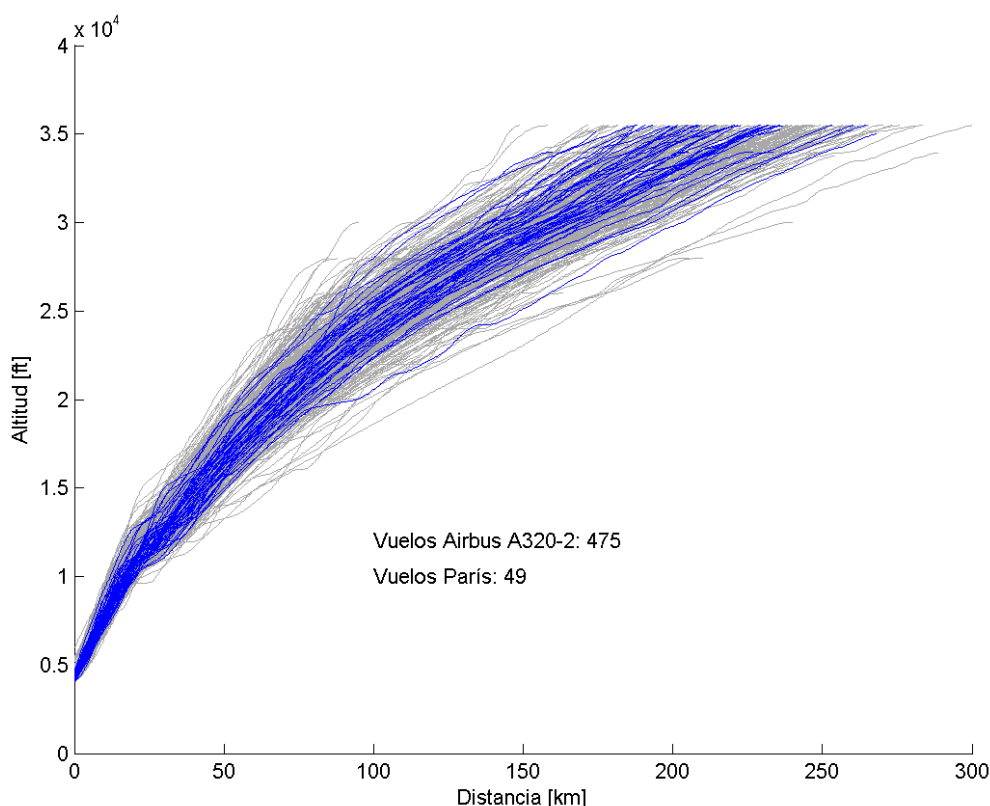


Figura 4.23: Perfiles verticales de los vuelos realizados por el Airbus A320-2 con destino a París.

salidas es considerablemente menor que en los dos anteriores, de ahí que se aprecie con claridad la sombra gris representativa del resto de salidas del Airbus A320-2. El bajo volumen de vuelos junto con la gran dispersión que presenta hacen que no tenga mucho sentido establecer un rango de distancia en el que este tipo de vuelo suele alcanzar los 35500 ft.

Al igual que se realizó con anterioridad, se representarán los percentiles 2.5 y 97.5 correspondientes a cada destino para las salidas realizadas por el Airbus A320-2, para así cuantificar de alguna manera los gradientes de ascenso seguidos por este modelo según el destino elegido. Esta representación se muestra en la Figura 4.24. Para la realización de esta representación se han utilizado de nuevo rangos de distancia de 2 km para evitar un rizado excesivo. Al igual que ocurrió en otras representaciones de percentiles con anterioridad, los perfiles verticales correspondientes a los tres destinos son prácticamente idénticos en los primeros 20 km de recorrido

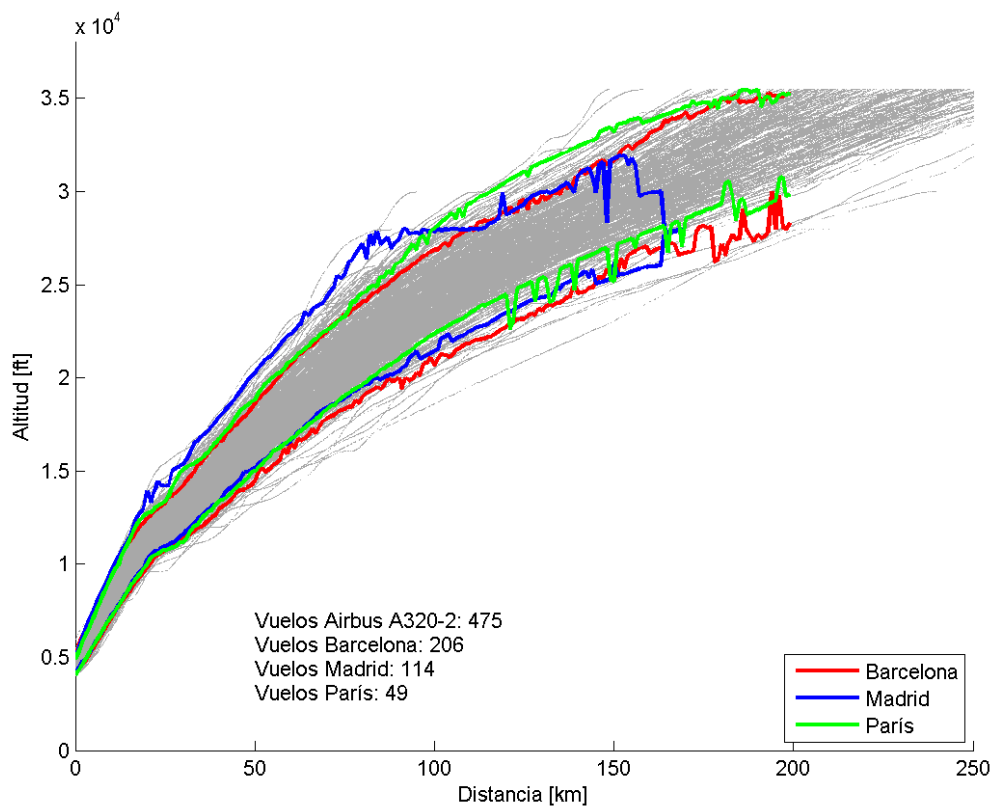


Figura 4.24: Perfiles verticales del Airbus A320-2 y los percentiles 2.5 y 97.5 correspondientes a cada destino.

aproximadamente, separándose a continuación. Una vez los vuelos han recorrido esta distancia, se ve cómo los vuelos con destino a Madrid ascienden claramente con un gradiente mayor, aunque su dispersión es tal que en algunos tramos el percentil 2.5 se encuentra por debajo del correspondiente a París. Además, se aprecia cómo antes de alcanzar los 100 km, el percentil 97.5 correspondiente a Madrid ya se ha estabilizado en el nivel de vuelo de crucero, lo que indica que algunos vuelos ya habrán alcanzado este nivel habiendo recorrido una distancia tan corta. Por otro lado, comparando los vuelos con destino Barcelona y París, se puede apreciar cómo describen prácticamente el mismo perfil vertical, presentando un poco más de variación en el percentil 2.5, aunque esta diferencia no sea muy significativa. En concreto, el percentil 2.5 de los vuelos con destino Barcelona se encuentra en 22360 una vez se han recorrido 100 km, mientras que el correspondiente a París alcanza un valor de 20650.

4.2.2. Llegadas

Partiendo de las llegadas con destino Sevilla, se extraen los tramos entre 3000 y 32000 ft y se descartan aquellos con una diferencia máxima entre dos datos mayor que 30 segundos, o una media entre datos mayor que 10 segundos, por las razones descritas en la Sección 3.13, obteniendo 1125 llegadas que suponen un 52 % del total de las llegadas. En este caso, la diferencia entre llegadas y salidas, una vez se ha pasado el filtro, no tiene por qué ser nula o al menos pequeña, puesto que depende de la calidad de la señal de cada vuelo el hecho de que se hayan eliminado más o menos. De hecho, se puede apreciar cómo de ser prácticamente el mismo valor antes del filtro (la diferencia era de una salida), se ha acabado teniendo más llegadas (29 más) que salidas. En la Tabla 4.7 se muestran los modelos de aeronave que más veces se han registrado en las llegadas con buena calidad de señal, así como las aerolíneas.

Orden	Modelo	Nº de llegadas	Aerolínea	Nº de llegadas
1	Airbus A320-2	539	Vueling	422
2	Boeing 737-8	483	Ryanair	376
3	Airbus A319-1	50	Iberia	107
4	Boeing 757-2	32	Transavia	55
5	Canadair CRJ-9	3	Air Europa	33

Tabla 4.7: Modelos de aeronaves y aerolíneas que más veces han llegado al aeropuerto de Sevilla con buena calidad de señal.

Los modelos más utilizados son el Airbus A320-2, el cual aparece en 539 ocasiones y el Boeing 737-8, el cual aparece en 483 ocasiones, siendo Vueling y Ryanair con 422 y 376 llegadas respectivamente las aerolíneas que más vuelos presentan.

En las llegadas, la distancia que se representará en los perfiles verticales representa la distancia que aun tienen que recorrer los vuelos para alcanzar los 3000 ft, que es la altitud a la que se aplica el filtro de calidad de la señal. Es por este motivo que se representa con valores negativos.

En la Figura 4.25 se muestran los perfiles verticales de todas las llegadas registradas en el aeropuerto de Sevilla. La mayoría de las llegadas registradas que vienen de un crucero cuya altitud es superior a los 32000 ft, inician su descenso cuando les falta una distancia comprendida

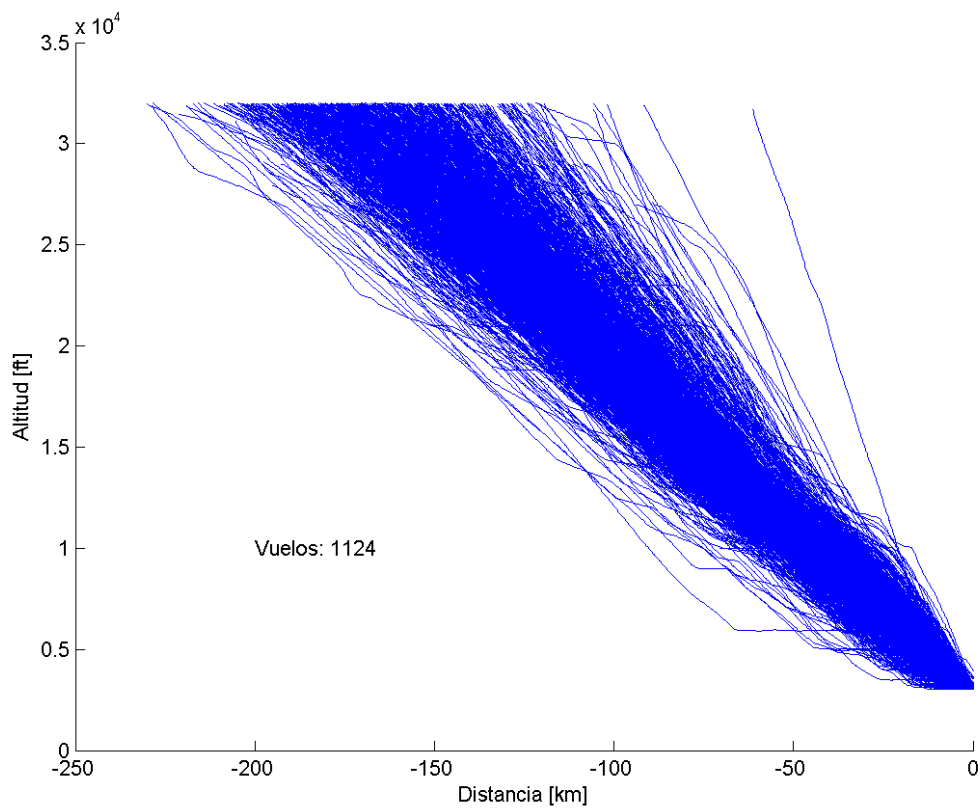


Figura 4.25: Perfiles verticales de todas las llegadas en el aeropuerto de Sevilla que presentan buena calidad de señal.

entre los 206 y los 135 km. Con respecto al global de las aeronaves, hay un descenso que destaca sobre el resto, situándose en la parte derecha de la representación con una pendiente bastante más acusada que la de los perfiles verticales del resto de llegadas. Este vuelo fue realizado por el A400M y por tanto se puede concluir que seguramente estuviera realizando ensayos en vuelo, pues ya se ha visto que el descenso realizado se sale por completo de lo esperado.

4.2.2.1. Comparación entre modelos de avión

En la Figura 4.26 se muestran los perfiles verticales de todas las llegadas y resaltados en azul, los perfiles verticales de todos los Airbus A320-2. El gran volumen de tráfico que presenta

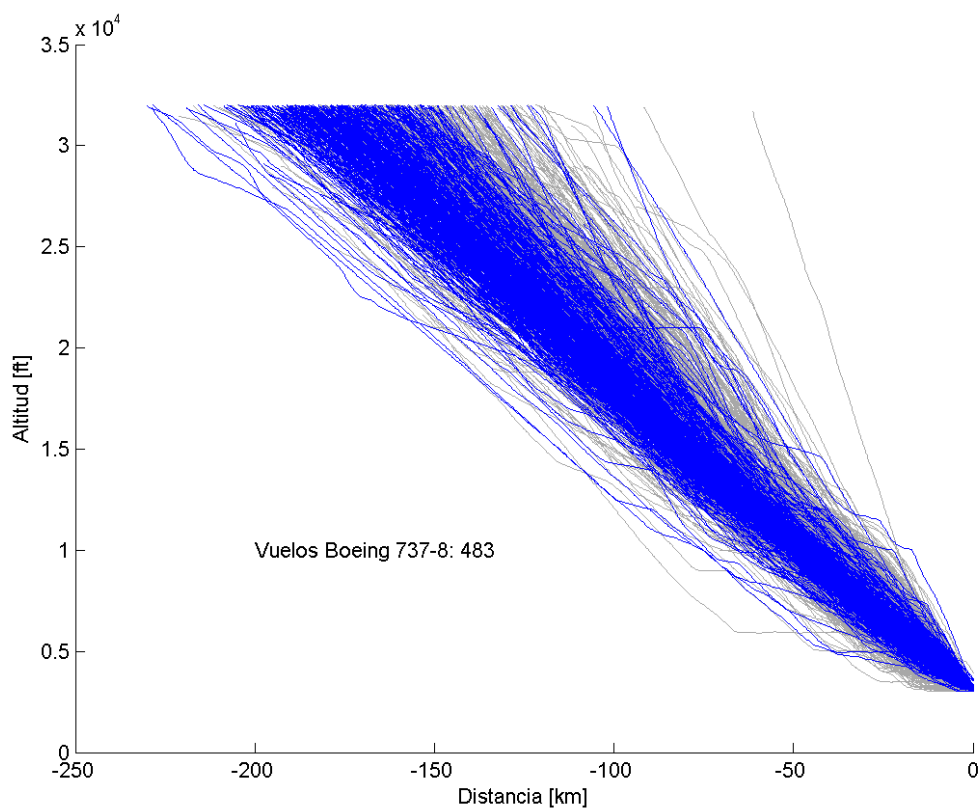


Figura 4.26: Perfiles verticales de las llegadas, resaltando el modelo más común.

este modelo, junto con la dispersión que muestran sus descensos ocupan casi el total de la sombra gris representativa del total de las llegadas, encontrándose el inicio de estos cuando las aeronaves se encuentran a una distancia comprendida entre los 193 y los 135 km de alcanzar los 3000 ft de altitud.

Por su parte, en la Figura 4.27 se muestran los perfiles verticales de todos los descensos y resaltados en azul, los perfiles verticales de todos los Boeing 737-8. En este caso, se observa cómo los perfiles verticales de los descensos de los Boeing 737-8 ocupan ligeramente la zona inferior de la sombra, aunque tampoco puede afirmarse esto con rotundidad. La dispersión mostrada sí que es algo menor, mientras que la distancia a la que comienzan el descenso aquellos vuelos

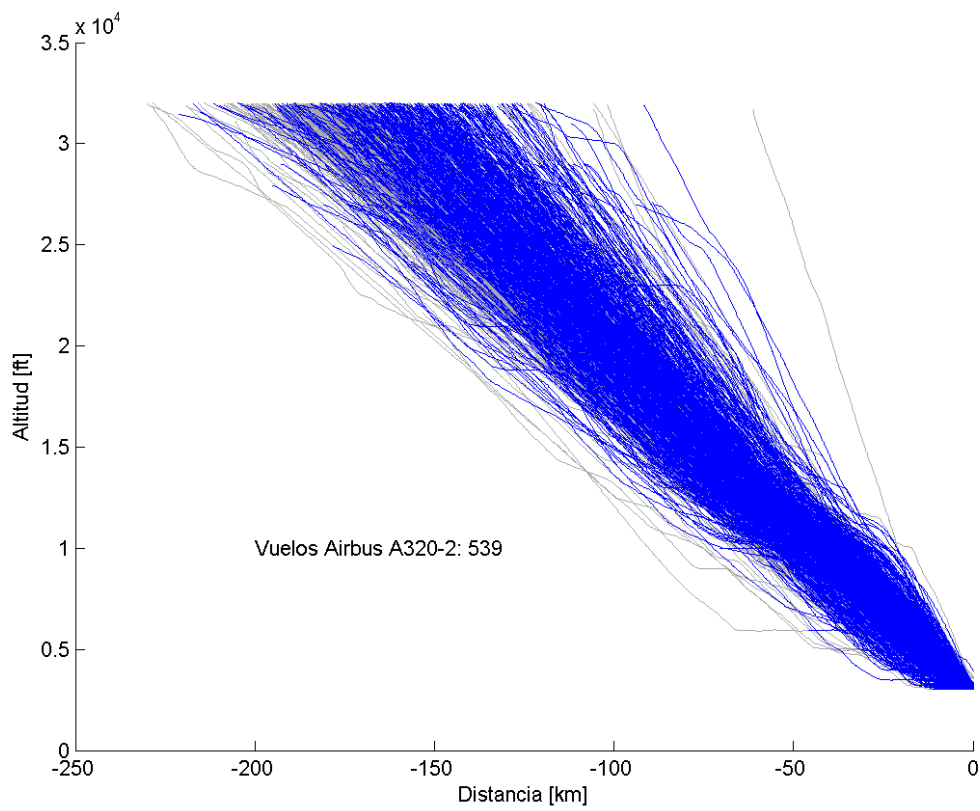


Figura 4.27: Perfiles verticales de los descensos, resaltando el segundo modelo más común.

cuya altitud de crucero se encontraba por encima de los 32000 ft, se encuentra entre los 204 y los 150 km.

A diferencia del caso de las salidas, en los que sí que se podía apreciar en los perfiles verticales por separado cuál de ellos utilizaba un mayor gradiente de ascenso, en este caso no resulta tan sencillo. Por esto, al igual que se hizo con las salidas, se compararán los perfiles verticales de los descensos de los dos modelos más comunes utilizando los perfiles verticales de todos los descensos con buena calidad de señal y las curvas de los percentiles 2.5 y 97.5 de los Boeing 737-8 y Airbus 320-2. Esta representación se muestra en la Figura 4.28, en la que se han agrupado las altitudes en tramos de distancia recorrida de 1 km y se han calculado los percentiles desde que les quedaban a las aeronaves 175 km para aterrizar. Se representan en azul los percentiles correspondientes a los Airbus A320-2 y en rojo a los Boeing 737-8. Al igual que ocurría con las

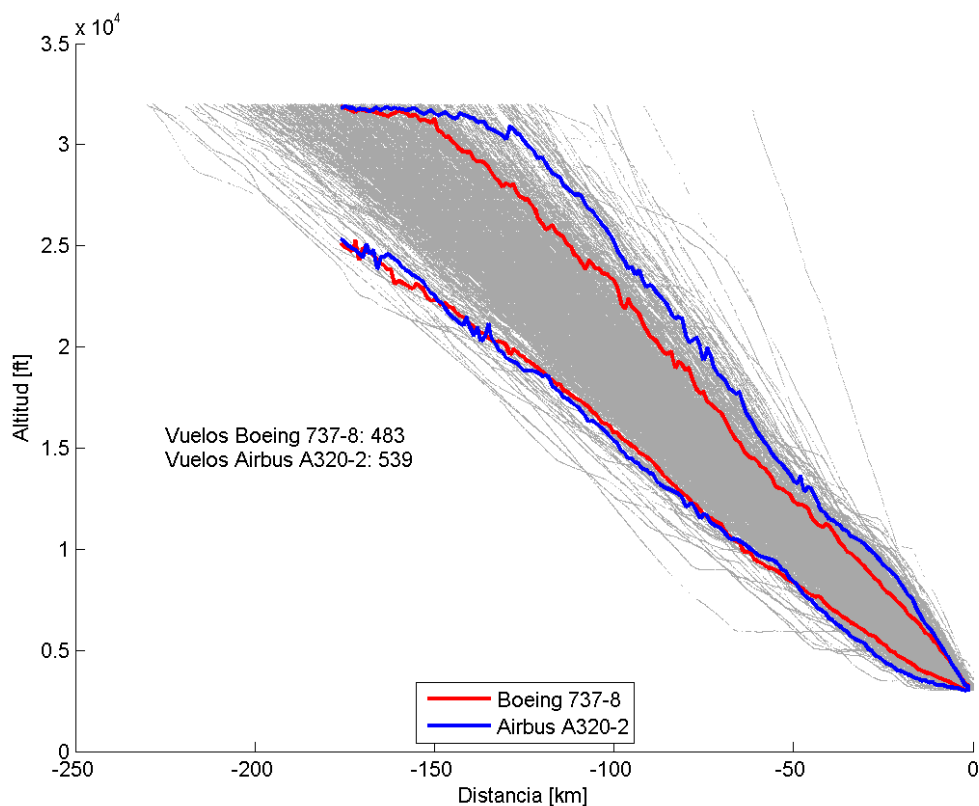


Figura 4.28: Perfiles verticales de los aterrizajes, comparando los dos modelos más comunes.

salidas, en los 50 km anteriores al aterrizaje, los perfiles verticales de ambos modelos son muy parecidos. Además, cuando aun la distancia por recorrer es mayor, se puede apreciar cómo los Airbus A320-2 utilizan un gradiente de descenso ligeramente mayor. En concreto, cuando los descensos distan 100 km del aeropuerto, los percentiles 2.5 y 97.5 de las altitudes de los Airbus A320-2, se encuentran a 15380 y 25280 ft, mientras que los correspondientes a los Boeing 737-8 están a 15780 y 23300 ft.

4.2.2.2. Distribución por destinos

Los 10 orígenes más frecuentes de vuelos que llegan al aeropuerto de Sevilla, así como la distancia que debe recorrer el vuelo se muestran en la Tabla 4.8. Las estadísticas obtenidas se refieren a los aterrizajes antes de pasar el filtro de calidad de señal, el cual se utilizará para la representación de los perfiles verticales.

Orden	Origen	Nº de llegadas	Distancia [NM]
1	Barcelona	420	437
2	París	144	783
3	Madrid	138	214
4	Tenerife	113	750
5	Gran Canaria	98	743
6	Palma de Mallorca	97	426
7	Londres	86	881
8	Bilbao	83	379
9	A Coruña	59	371
10	Amsterdam	58	1001

Tabla 4.8: Orígenes más frecuentes en el aeropuerto de Sevilla.

Al igual que en el caso de las salidas, se mantiene una clara diferencia entre el número de vuelos cuyo origen es Barcelona, y el resto de orígenes. Además, se puede comprobar que el número de vuelos en cada sentido para un mismo aeropuerto, están cercanos unos de otros, como cabía esperar, aunque puedan aparecer pequeñas diferencias como es el caso de Gran Canaria.

4.2.2.3. Comparación entre modelos, para un mismo origen (Barcelona)

Al igual que se realizó anteriormente con las salidas, ahora se analizarán las llegadas según el modelo de aeronave que lo realice para un mismo origen. De esta manera se evitará en la medida de lo posible la dependencia que pudiera tener el perfil vertical del aterrizaje con la distancia que lleve recorrida la aeronave o incluso el peso al despegue, pues al ser el mismo origen, estos serán similares para las dos aeronaves que se analizarán, que serán el Airbus A320-2 y el Boeing 737-8.

En la Figura 4.29 se muestra el perfil vertical de todas las llegadas con origen en Barcelona y superpuestas en azul, las llegadas realizadas por los Airbus A320-2, que resulta ser la aeronave que más veces se ha registrado que cubra este trayecto.

En este caso, el número de llegadas (que han pasado el filtro de calidad de señal) al aeropuerto de Sevilla con origen en Barcelona es de 258, habiendo sido 224 de ellos realizados por el Airbus A320-2, lo que supone un 87 %. Este porcentaje tan elevado provoca que la representación no aporte mucha información. Además, los perfiles verticales de este tipo de vuelo son tan dispersos que no se puede establecer con claridad una distancia a partir de la cual se pueda considerar que comienza el descenso para aquellas aeronaves que se encontraran por encima de

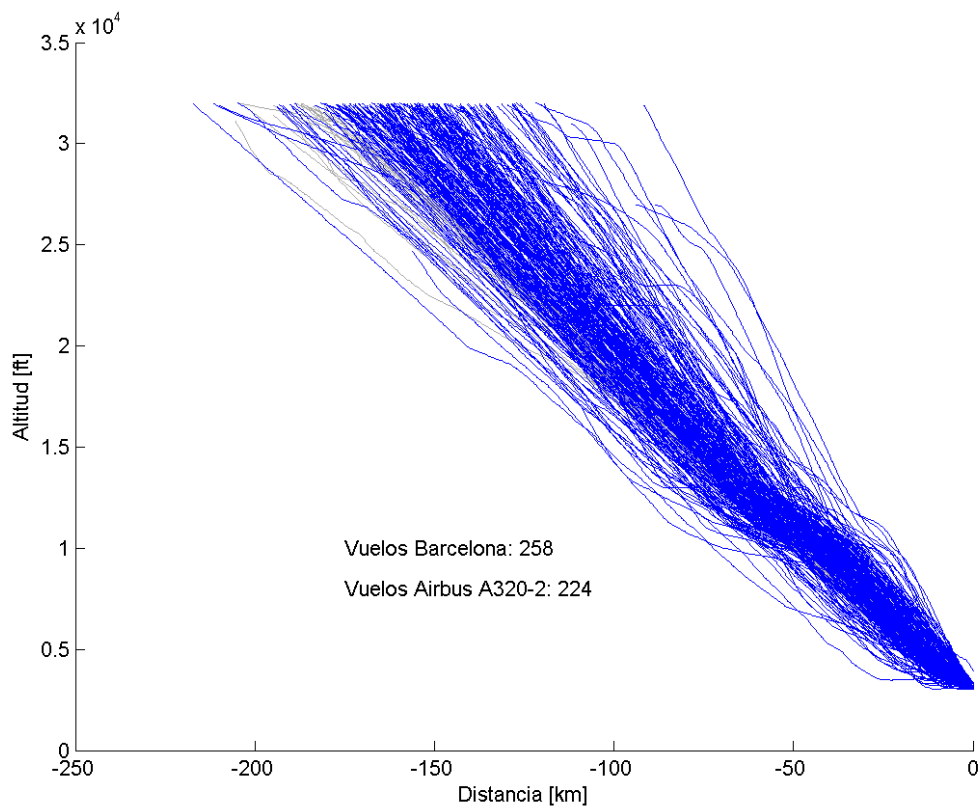


Figura 4.29: Perfiles verticales de los descensos del Airbus A320-2 con origen en Barcelona.

los 32000 ft.

Por otro lado, en la Figura 4.30 se muestran los perfiles verticales de las llegadas correspondientes al Boeing 737-8.

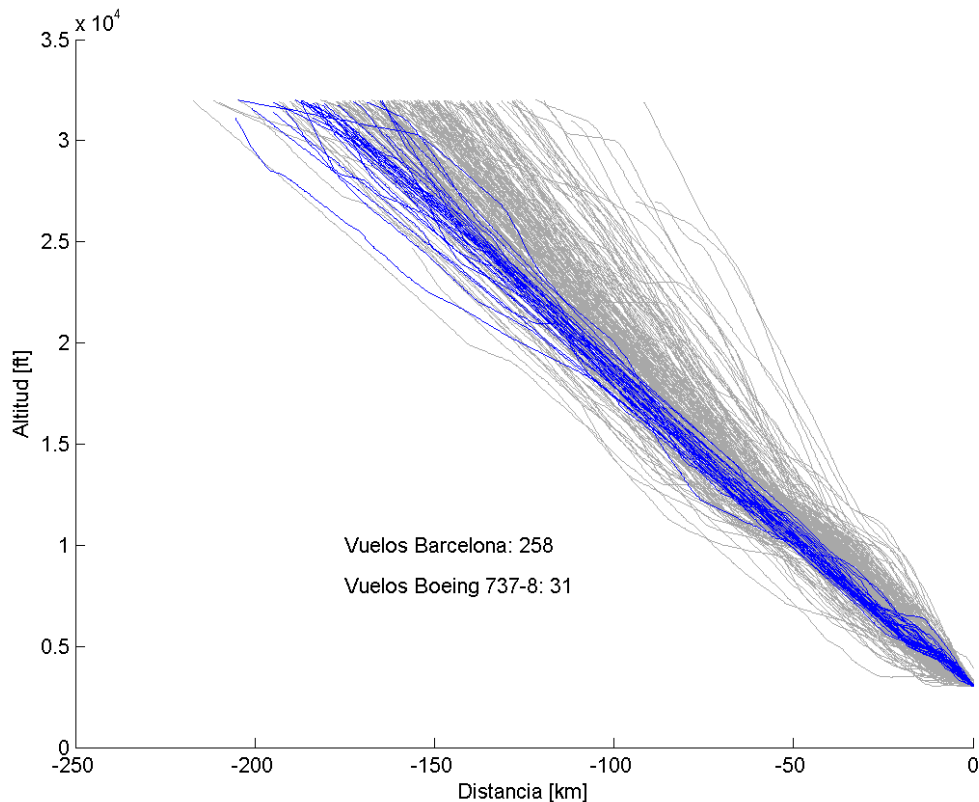


Figura 4.30: Perfiles verticales de los descensos del Boeing 737-8 con origen en Barcelona.

Únicamente se han registrado 31 llegadas, aunque sí que muestran una clara tendencia sobre el tipo de descenso que realizan, resultando a priori más tendido que el realizado por el Airbus A320-2. Estos descensos se inician cuando aun distan las aeronaves entre 196 y 177 km del aeropuerto de Sevilla.

De todas formas, atendiendo a la estructura seguida con anterioridad, se representarán los percentiles 2.5 y 97.5 realizados cada tramo de 2 km recorridos para así cuantificar de alguna forma el resultado obtenido. Esta representación se muestra en la Figura 4.31.

Se observa que la dispersión presentada por el Airbus A320-2 es mucho mayor que la que muestran los percentiles correspondientes al Boeing 737-8. Este hecho se aprecia claramente en la separación existente entre las curvas correspondientes a los percentiles, resultando mucho más cerradas las del Boeing 737-8. Las altitudes a las que se encuentran los percentiles 2.5 y 97.5 correspondientes a los descensos (una vez que faltan 100 km para llegar al aeropuerto de Sevilla) del Airbus A320-2 son 15300 y 25170 ft, mientras que por su parte, los correspondientes a los descensos de los Boeing 737-8 se sitúan a 16740 y 19080 ft respectivamente.

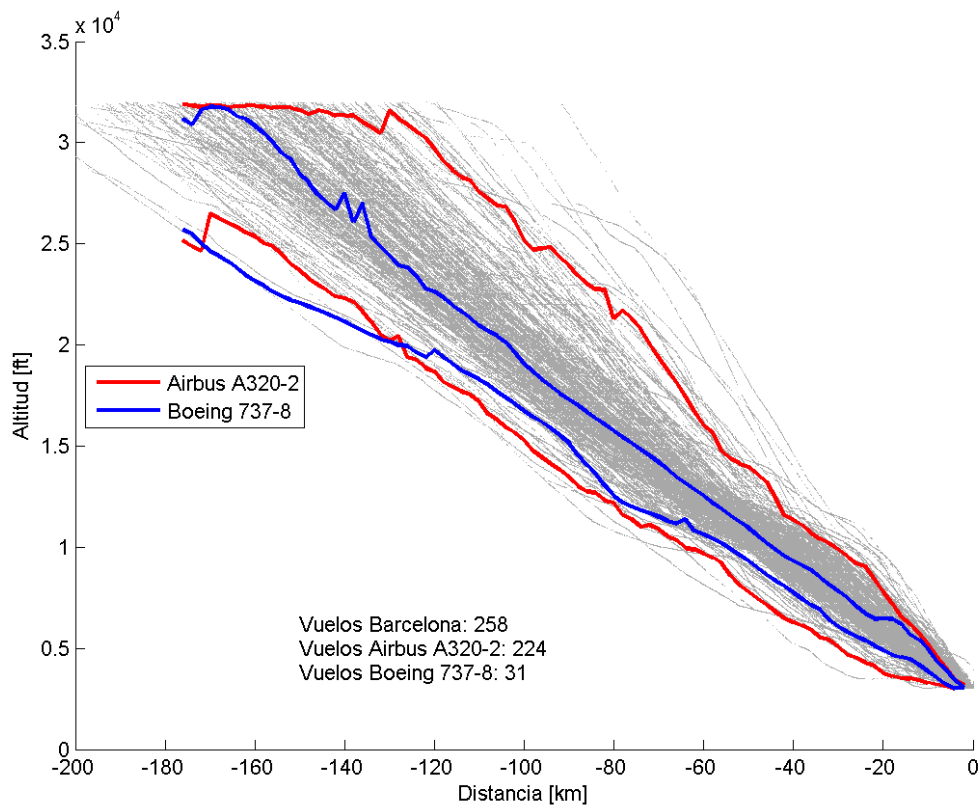


Figura 4.31: Perfiles verticales de los vuelos con origen en Barcelona y los percentiles 2.5 y 97.5 correspondientes a los dos modelos más comunes.

4.2.2.4. Comparación entre destinos, para un mismo modelo (Airbus A320-2)

Para finalizar con el estudio de los perfiles verticales, se analizarán las llegadas del Airbus A320-2 al aeropuerto de Sevilla, diferenciando tres orígenes distintos que serán Barcelona, Madrid y París. Para estos orígenes se registraron 224 llegadas desde Barcelona, 103 desde Madrid y 58 desde París entre un total de 539 llegadas realizadas por Airbus A320-2, lo que supone un 41 %, un 19 % y un 10 % respectivamente. Como se comentó en la sección análoga en el caso de las salidas, estos destinos representan tres tipos de orígenes distintos atendiendo a la distancia a la que se encuentran del aeropuerto de Sevilla.

En la Figura 4.32 se muestra el perfil vertical de todas las llegadas realizadas por los Airbus A320-2 y superpuestas en azul, las llegadas con origen en Barcelona. Al igual que ha ocurrido en

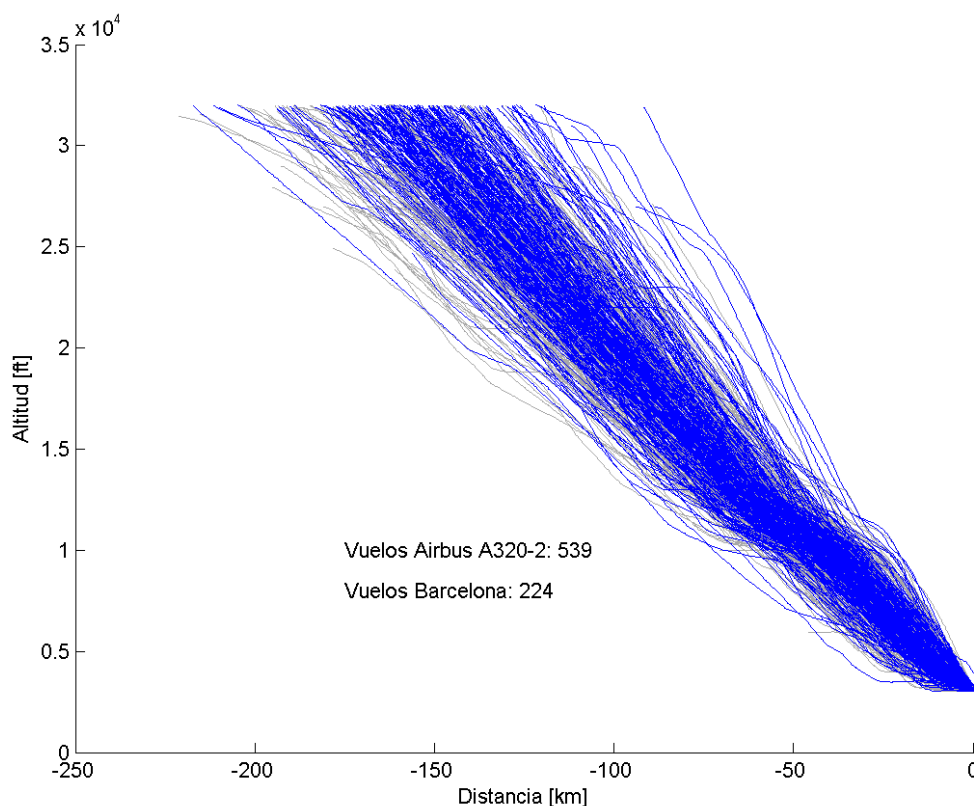


Figura 4.32: Perfiles verticales de los descensos del Airbus A320-2 con origen en Barcelona.

apartados anteriores, el gran volumen de tráfico correspondiente a este origen ocupa la mayor parte de la sombra representativa de todas las llegadas realizadas por el Airbus A320-2.

Por otro lado, en la Figura 4.33 se muestran los perfiles verticales de los descensos de los Airbus A320-2 con origen en Madrid. Los perfiles verticales correspondientes a este tipo de vuelo también presentan una dispersión que no permite encasillarlos con un tipo de descenso concreto (atendiendo al gradiente de descenso utilizado), encontrándose el grueso de los descensos iniciados a 26000 ft entre los 174 y los 120 km. Así, se puede concluir únicamente que comienzan el aterrizaje partiendo de un nivel de vuelo inferior al del resto de orígenes.

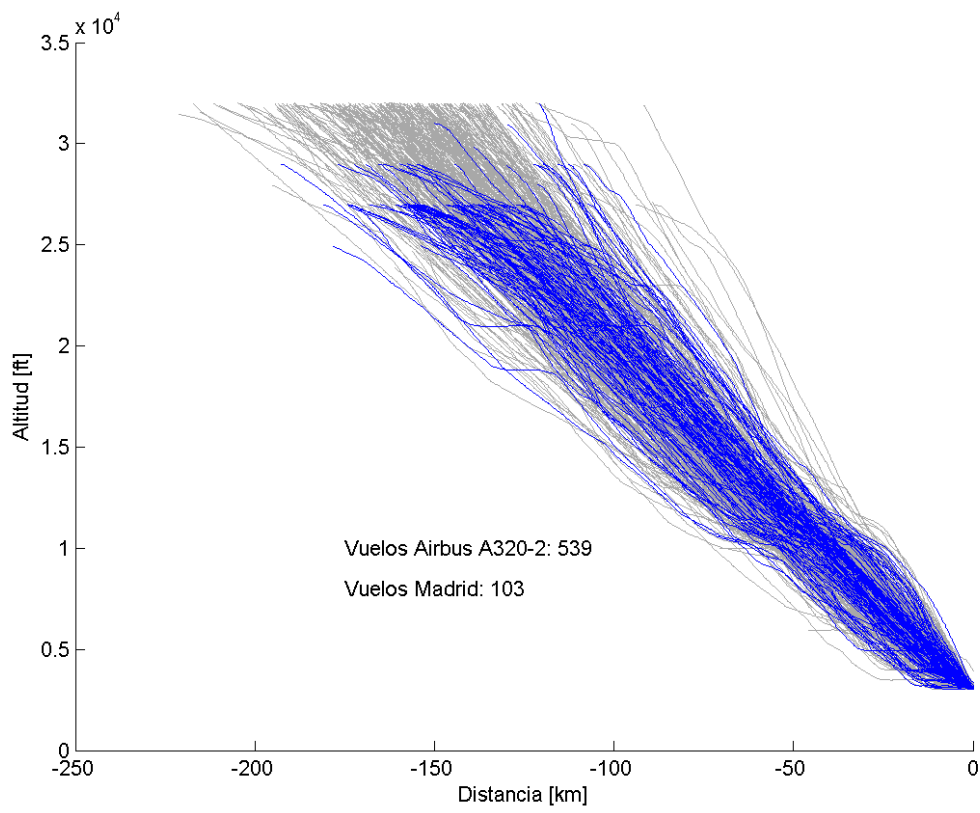


Figura 4.33: Perfiles verticales de los descensos del Airbus A320-2 con origen en Madrid.

Además, se ha decidido representar un origen internacional, cuya distancia al aeropuerto de Sevilla fuera claramente mayor a la que lo separa del aeropuerto de Barcelona. Es por este motivo por el que se ha elegido estudiar las llegadas de los Airbus A320-2 con origen en París. La representación de los perfiles verticales correspondientes a este tipo de vuelo se muestran en la Figura 4.34. Estos aterrizajes presentan al igual que los dos anteriores una gran dispersión

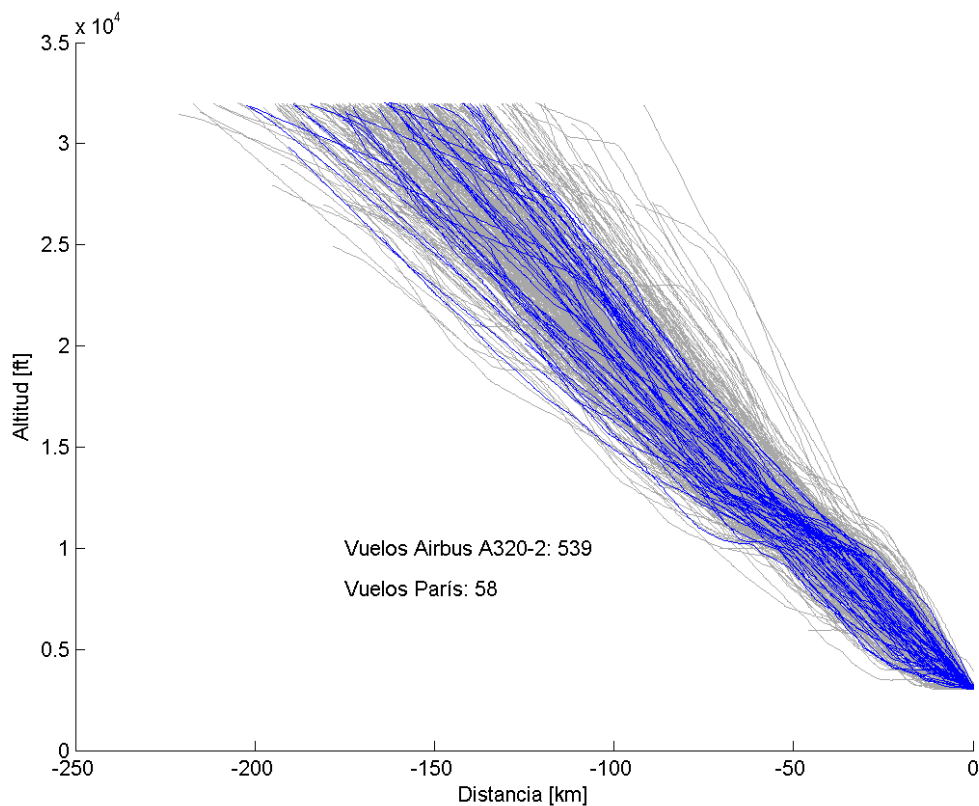


Figura 4.34: Perfiles verticales de los descensos del Airbus A320-2 con origen en París.

que no los diferencia del resto de vuelos con distinto origen.

De nuevo se representarán los percentiles 2.5 y 97.5 calculados en rangos de 2 km, asociados a cada una de las llegadas comentadas con el fin de poder sacar alguna conclusión cuantitativa sobre el efecto que puede presentar el origen para los aterrizajes de una misma aeronave. Esta representación se muestra en la Figura 4.35. Como se estaba observando levemente en las representaciones anteriores por separado, no se puede obtener una conclusión significativa acerca del efecto de la distancia entre los aeropuertos de origen y llegada para una misma aeronave sobre el perfil vertical mostrado en el descenso. En el caso mostrado, los percentiles 2.5 se mantienen prácticamente en los mismos valores durante 120 km aproximadamente, debido a que antes de esta distancia, muchos de los vuelos procedentes de Madrid aun se encuentran en el tramo de crucero. Sobre los percentiles 97.5 sí que se puede apreciar una mayor diferencia, siendo los aterrizajes que menor gradiente de descenso presentan aquellos procedentes de París.

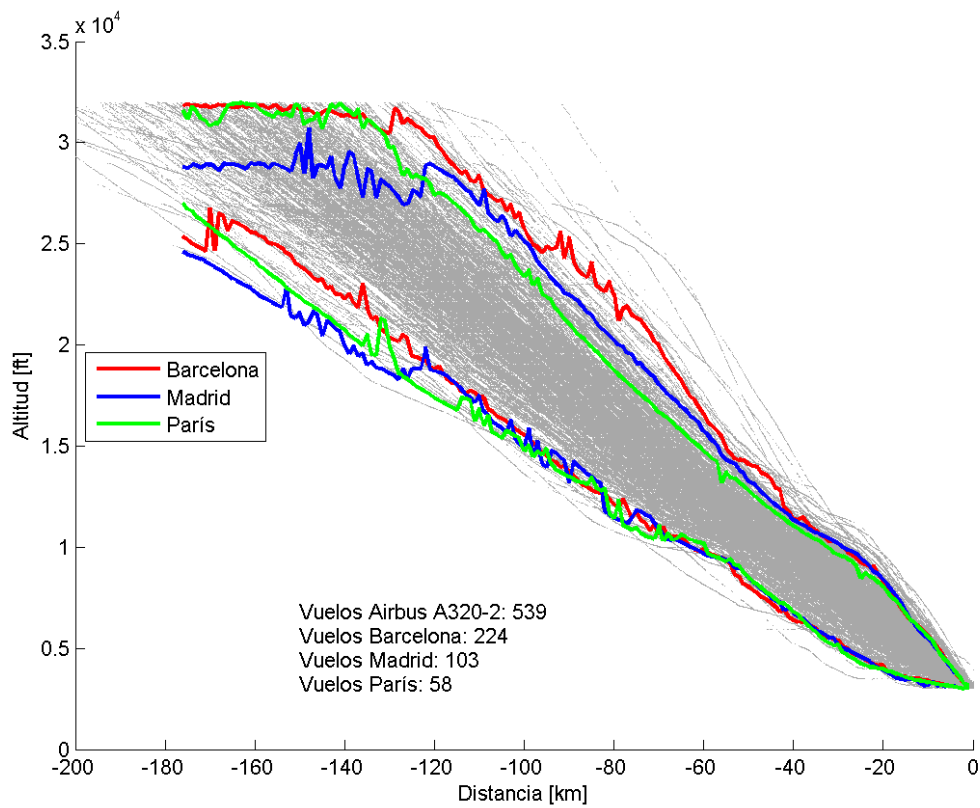


Figura 4.35: Perfiles verticales del Airbus A320-2 y los percentiles 2.5 y 97.5 correspondientes a cada origen.

4.3. Otros resultados

En el siguiente apartado se mostrarán otros resultados obtenidos, los cuales no se encuentran entre los objetivos principales del proyecto, pero sí que resultan interesantes de analizar.

4.3.1. Obstáculos en la recepción de la señal

Las señales de ADS-B enviadas por las aeronaves se reciben gracias a una antena localizada en el tejado de uno de los laboratorios de la ETSI. Aunque la localización puede considerarse bastante aceptable, sí que presenta algunos inconvenientes en cuanto a la cobertura, puesto que, aunque Sevilla es una ciudad sin apenas accidentes orográficos, sí que presenta edificios de una altura considerable, y más si se considera la altura del edificio en el que se encuentra la antena.

En la Figura 4.36 se muestran los perfiles horizontales de todos los despegues y aterrizajes registrados en el aeropuerto de Sevilla, así como la pista, el punto de localización de la antena y unas rectas (representadas por puntos) radiales con centro en la antena indicadoras de aquellas zonas más faltas de cobertura.

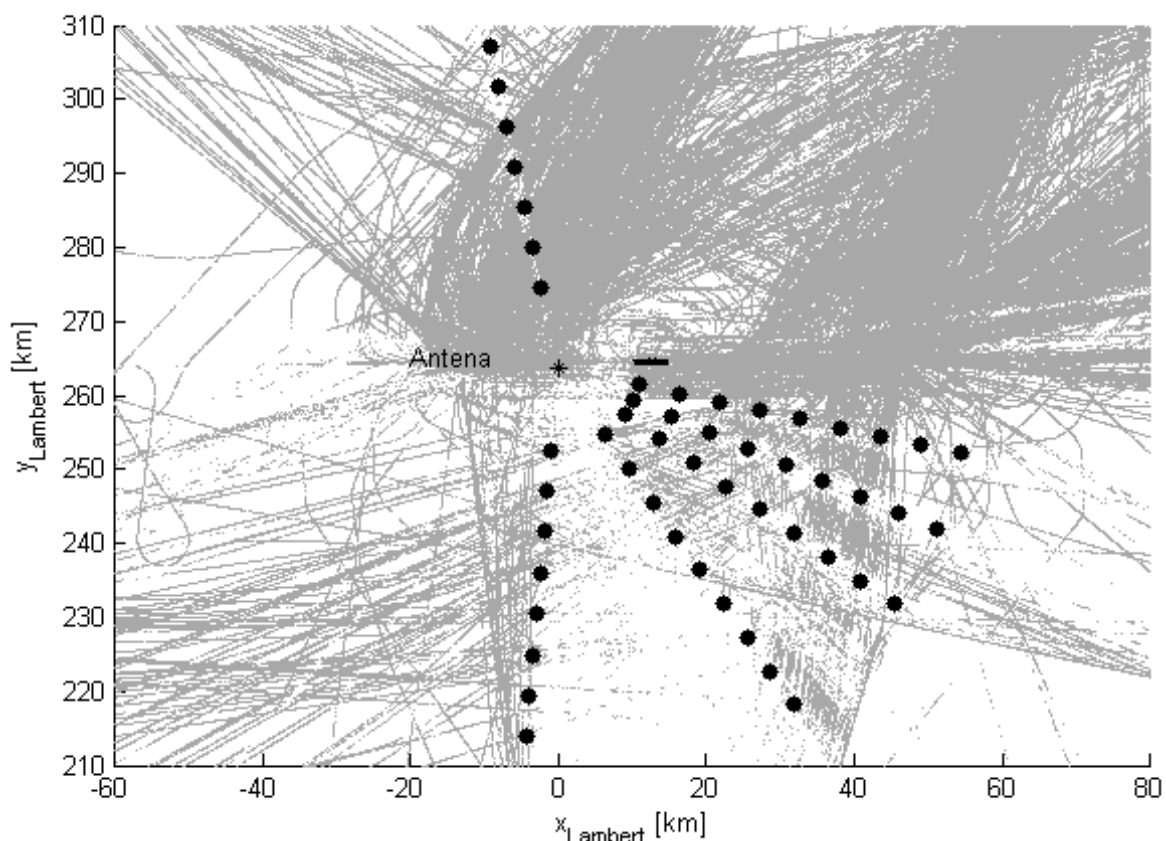


Figura 4.36: Perfil horizontal de todos los vuelos y posibles zonas faltas de cobertura debido a los obstáculos encontrados.

4.3.2. Circuitos de espera

Entre todos los aterrizajes realizados en Sevilla en el periodo de nueve semanas analizado, se han encontrado dos vuelos que necesitaron dar alguna vuelta a uno de los circuitos de espera definidos en las cartas STAR 27, en concreto el circuito de espera que se encuentra en el IAF de RUVEN.

En la Figura 4.37 se muestran los perfiles horizontales de ambos vuelos. Uno de los vuelos

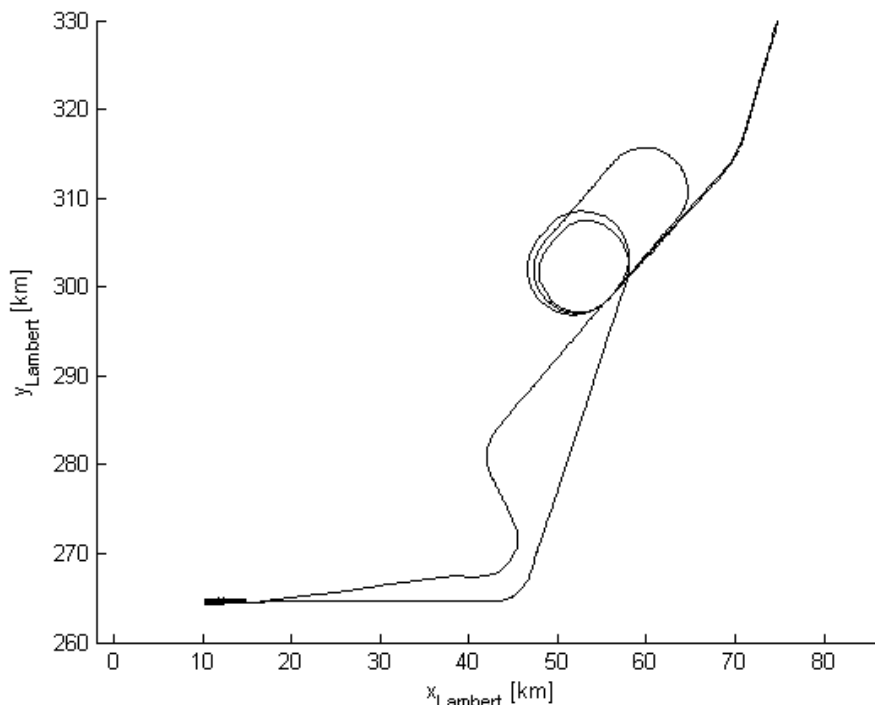


Figura 4.37: Perfil horizontal de los circuitos de espera recorridos.

lo recorrió continuando el descenso del que venía, dando una única vuelta, mientras que el otro, dio dos vueltas al circuito manteniendo la altura constante en 3000 ft. Ambos utilizaron el circuito de espera definido en RUVEN, realizando, uno de ellos, después de dar la vuelta al circuito, los dos virajes definidos en las cartas para alcanzar la pista, mientras que el otro se ahorró estos virajes, utilizando una trayectoria directa para encarar la pista.

4.3.3. Aterrizajes frustrados

Se han detectado entre los aterrizajes realizados en el aeropuerto de Sevilla, 10 vuelos que realizaron aterrizajes frustrados. Estos ocurrieron indiferentemente en ambas cabeceras de pista, volviendo las aeronaves a alcanzar una altitud entre 1000 y 3000 ft para volver a intentar el aterrizaje, como se observa en la Figura 4.38, la cual muestra los perfiles verticales de dos de estos vuelos, un Boeing 747 de la Fuerza Aérea Japonesa y el único Airbus A340 que ha aterrizado en el aeropuerto de Sevilla en las nueve semanas de análisis. En la Figura 4.38, el perfil que realiza hasta 7 aterrizajes frustrados, es el correspondiente al Airbus A340.

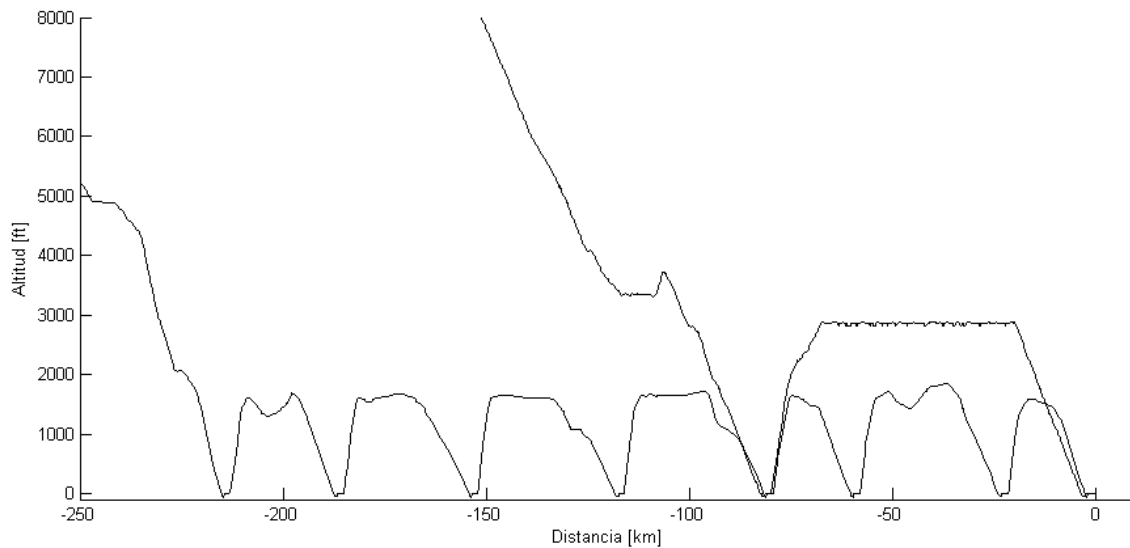


Figura 4.38: Perfil vertical de los aterrizajes frustrados.

4.3.4. Máximas altitudes en llegadas y salidas

Un resultado que se ha considerado interesante es la altitud máxima alcanzada por aeronaves que despegan y aterrizan en el aeropuerto de Sevilla. En ambos casos, se han encontrado aeronaves que han alcanzado los 43000 ft, tratándose de un Embraer EMB de la aerolínea Wells Fargo Bank Trustee en el caso de los despegues y un Cessna 525 de la aerolínea MACH Airlines en los aterrizajes.

4.3.5. Aeronaves particulares

Aunque lejos del aeropuerto de Sevilla, se han registrado cuatro vuelos correspondientes a Airbus A380-8 y tres correspondientes a C130 Hercules. Además, también se registraron vuelos correspondientes al Airbus A400M y Airbus A300-6 (Beluga), los cuales se comentan a continuación.

4.3.5.1. Airbus A400M

Debido a la localización de la FAL (*Final Assembly Line*) de esta aeronave, es relativamente frecuente encontrar algunos vuelos (79 en total) que ha realizado.

En la Figura 4.39 se pueden observar las zonas en las que se han llevado a cabo los vuelos de esta aeronave, correspondiéndose algunos de estos a ensayos en vuelos.

Como se puede deducir del carácter caótico del perfil horizontal, muchos de estos vuelos no siguen en ningún momento las rutas marcadas por las cartas.

Aunque se ha obtenido un vuelo con una distancia recorrida mayor que 1000 km en el área de

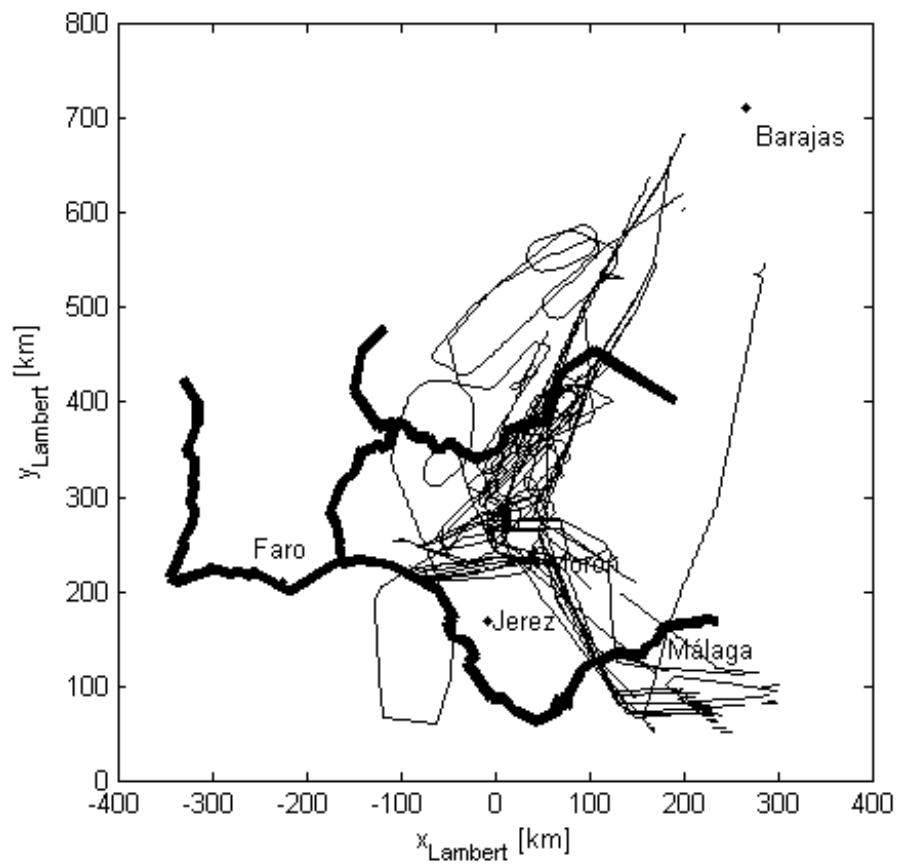


Figura 4.39: Perfil horizontal de los vuelos realizados por el A400M.

cobertura de la antena del receptor ADS-B, se muestran en la Figura 4.40 alguno de los perfiles verticales encontrados más representativos. Se observan tres perfiles verticales, presentando dos

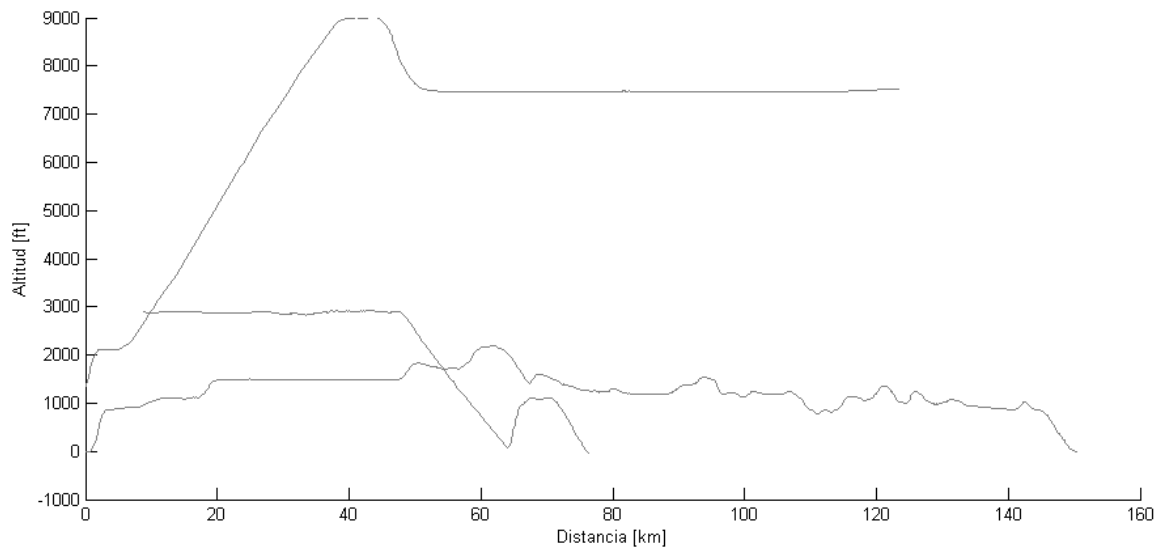


Figura 4.40: Perfil vertical de los vuelos realizados por el A400M.

de ellos una evolución de las altitudes muy suave, siendo el perfil vertical del tercero rugoso.

4.3.5.2. Airbus A300-6 (Beluga)

Además de la presencia del Airbus A400M, por la FAL de San Pablo también aparece el avión de transporte Airbus A300-6 (Beluga), el cual ha sido detectado en 6 ocasiones (3 despegues y 3 aterrizajes) en las nueve semanas de análisis.

4.3.6. Otros aeropuertos

Como se comentó anteriormente, se llegan a captar datos de aeronaves operando sobre los aeropuertos de Jerez, Málaga y por pocos kilómetros, Barajas. De hecho, tanto Jerez como Málaga tienen una cobertura relativamente buena considerando la distancia a la que se encuentran.

4.3.6.1. Aeropuerto de Málaga

En el aeropuerto de Málaga se han detectado 4120 despegues (casi el doble que en Sevilla) y 1775 aterrizajes. La diferencia tan grande que aparece entre ambos valores no tiene mayor importancia, pues la calidad de la señal recibida no es muy buena y en ningún caso se detecta a ninguna aeronave en tierra debido a la orografía existente entre la antena receptora y la pista, lo que provoca que a la hora de determinar si una aeronave se encuentra aterrizando o no, los filtros utilizados no sean muy eficientes. Tampoco se ha pensado en mejorarlos (cambiando las condiciones del IsOnGround), puesto que el análisis de estos vuelos no es objeto de este

proyecto. Sobre el número de vuelos que pasan los filtros de calidad de señal, no supera la decena, por lo que tampoco tiene sentido analizar perfiles verticales.

En cuanto a las aeronaves que operan, destacan al igual que en el aeropuerto de Sevilla, el Boeing 737-8 y el Airbus A320-2, siendo las aerolíneas que cuentan con más operaciones Ryanair y EasyJet.

4.3.6.2. Aeropuerto de Jerez

En cuanto al aeropuerto de Jerez, cabe comentar que se han detectado 356 despegues y 481 aterrizajes, siendo en este caso considerablemente mayor el número de vuelos que superan los filtros de calidad de señal en comparación con los detectados, aunque insuficiente para un análisis.

Las aeronaves que más operan en este aeropuerto son de nuevo el Boeing 737-8 y el Airbus A320-2. Sin embargo, las aerolíneas que más operan son Ryanair y Royal Air Maroc, esta segunda con un número de vuelos parecido a los de Ryanair.

Conclusiones

En el presente proyecto se han obtenido y analizado las trayectorias de los vuelos de aeronaves captados por el receptor ADS-B, considerando cubiertos los objetivos fijados en un principio.

Para la realización de este análisis, ha sido necesario particionar los archivos de datos capturados para evitar problemas de memoria RAM. Estas particiones se leen, se separan según las aeronaves que han aparecido y se unen las estructuras correspondientes a cada partición. De esta manera, se pasa a lo que es el procesamiento de los datos en sí. Este consiste en la búsqueda de los modelos y aerolíneas asociados a cada código hexadecimal, así como a la creación de una base de datos que recoja todos estos datos, su correspondiente clasificación según se trate cada vuelo de un crucero, un ascenso o un descenso. Seguidamente se determina cuáles de estos ascensos o descensos se corresponden con despegues y aterrizajes del aeropuerto de Sevilla, así como los tramos del ascenso o descenso de cada vuelo. A partir de estos tramos se realiza una estadística con la que analizar qué tramos de dichos ascensos o descensos cumplen con unos ciertos requisitos. A continuación se determina la cabecera de pista que cada despegue o aterrizaje utilizó y se les pasa a estos un filtro de calidad de señal atendiendo a las altitudes escogidas anteriormente. Los vuelos que pasan este filtro son los utilizados a la hora de realizar los perfiles verticales. Finalmente se determina la distancia recorrida por cada aeronave y se calculan las coordenadas correspondientes a la proyección cónica de Lambert utilizada.

En cuanto al análisis de los resultados obtenidos, se han representado perfiles horizontales y verticales de hasta 95760 vuelos.

- **Perfiles horizontales:** Se han clasificado las llegadas y salidas según la cabecera que utiliza cada una, apreciándose la preferencia por la cabecera 27. Además, se ha analizado el seguimiento de los procedimientos marcados en las correspondientes cartas aeronáuticas, se ha observado que las aeronaves no siempre cumplen con los procedimientos publicados y se ha comprobado que puede llegar a haber vuelos con desviaciones con respecto a la pista de hasta 5 km, cuyas causas se desconocen.
- **Perfiles verticales:** Se han analizado las salidas y llegadas al aeropuerto de Sevilla por separado, utilizando para estas representaciones únicamente aquellos vuelos que pasaron el filtro relativo a la calidad de la señal. Además, se han analizado los perfiles verticales

según el modelo de la aeronave y según el destino o el origen del que provienen. A partir de estas gráficas se ha observado que el Boeing 737-8 asciende con un gradiente mayor que el Airbus A320-2 (se trata de las dos aeronaves más comunes en el aeropuerto de Sevilla), y en cuanto a los destinos, cabe destacar los ascensos de aquellos vuelos con destino Madrid, los cuales alcanzan niveles de vuelo relativamente bajos.

En cuanto a las llegadas, no se han apreciado diferencias significativas en los gradientes de descenso independientemente de si el análisis es para una misma aeronave y distintos orígenes, como si es para distintas aeronaves con un mismo origen. Cabría destacar el caso particular de Madrid, en el cual los descensos se inician a altitudes menores que el resto de orígenes debido al nivel de vuelo en el que realizan el crucero.

6

Trabajo futuro

En el presente proyecto se han dedicado muchos esfuerzos a resolver los problemas ocasionados por la falta de memoria RAM. Una posible línea futura de mejora sería poder seleccionar qué datos recibir directamente del dispositivo SBS-3. Para esto quizás habría que prescindir del software *Kinetic* y acceder directamente al receptor.

Tal y como se ha comentado a lo largo de las memorias de este proyecto, asegurar una asociación entre call sign y aeropuertos de origen/destino de los vuelos puede resultar en la obtención de estadísticas y resultados bastante interesantes. Es por esto que de cara a la continuación del presente proyecto, se propone la investigación en este tema para dotar de una mayor veracidad a los resultados obtenidos y no tener que discriminar de alguna manera a ciertos aeropuertos por no contar con información suficiente.

A la hora de analizar los perfiles horizontales tanto de las salidas como de las llegadas, resultaría interesante estudiar, además del grado de seguimiento cualitativo de las rutas, un análisis cuantitativo de la desviación con respecto a estas, así como un estudio estadístico de las más usadas.

En este proyecto se han analizado principalmente dos aeronaves de fuselaje estrecho por ser las que mayor volumen de tráfico presentan con diferencia, apreciándose muy pocos vuelos realizados por aeronaves de fuselaje ancho. Podría resultar interesante captar con mejor calidad de señal los vuelos que operan en el aeropuerto de Málaga y Jerez, y así estudiar las actuaciones de aeronaves distintas, en especial los perfiles verticales. También se podrían obtener de alguna manera archivos *.bst captados en otras zonas de la geografía española, como podría ser a través de la colaboración con otras universidades o entidades interesadas en el tema.

En el presente proyecto se han analizado los perfiles horizontales de las aeronaves, comparándolos con las rutas marcadas por las diferentes cartas aeronáuticas (así como los perfiles verticales utilizados para el ascenso/descenso). Un posible estudio futuro podría ser comprobar si la trayectoria descrita por cada vuelo cumple con las restricciones en altitud.

Finalmente, Eurocontrol realiza una publicación en la que estudia las actuaciones realizadas por los modelos más comunes de aeronaves. Estas actuaciones podrían compararse con las obtenidas con el receptor ADS-B.

Bibliografía

1. International Civil Aviation Organization. *Doc 9713. International Civil Aviation Vocabulary*. Third edition, 2007.
2. Rob Strain. Implementation of data link and satcom communications. Federal Aviation Administration, November 2003.
3. Francisco Rogelio Palomo Pinto. *Apuntes de clase de Aviónica y Sistemas de Navegación*. ETSI Sevilla.
4. International Civil Aviation Organization. *Annex 10 to the convention on international civil aviation. Aeronautical telecommunications*. November 1996.
5. EUROCONTROL. *User Manual for the Base of Aircraft Data (BADA)*, 3.11 edition, May 2013.
6. John Parr Snyder. *Map projections—A working manual*. Number 1395. USGPO, 1987.
7. Rameshwar Prasad Misra and Attur Ramesh. *Fundamentals of cartography*. www.conceptpub.com, 1989.
8. Rafael Vázquez Valenzuela. *Apuntes de clase de Navegación Aérea*. ETSI Sevilla.

Apéndice

Apéndice A

Código Visual Basic

```
1 ' Programado por F. Cantos partiendo de una base de Neil Kilbride (http://
   neilkilbride.blogspot.com.es/2008/02/split-text-file-with-vbscript.html)
2 ' Fecha 24-03-2013
3 ' Versión 1.1
4
5 'Programa VBS que divide un archivo fuente en otros archivos menores que
   contienen cada uno las líneas por archivo exigidas y su nombre de salida es
   el mismo que el del
6 'archivo fuente solo que se le añade una "_" y el número ordinario que le
   corresponde.
7
8Nombre_fuente = INPUTBOX ("Insertar nombre del archivo fuente:", "Nombre del
   archivo fuente", "extracto10k")
9lineas_por_archivo = INPUTBOX ("Insertar el número de líneas por archivo:", "
   Líneas por archivo", "1000")
10
11max_lineas = lineas_por_archivo+0
12
13sourceFile = Nombre_fuente & ".bst"
14WScript.Echo ("Archivo fuente: " & sourceFile)
15Nombre_dir_datos = "D:\Archivos Paco\Matlab\PFC\datos\"
16
17basename = Nombre_fuente
18targetDir = Nombre_dir_datos & Nombre_fuente
19set objFSO = CreateObject("Scripting.FileSystemObject")
20k = 1
21
22if objFSO.FileExists( sourceFile ) then
23  WScript.Echo ("El archivo fuente existe")
24
25  DIM fso
26  Set fso = CreateObject("Scripting.FileSystemObject")
27  If fso.FolderExists(Nombre_dir_datos & Nombre_fuente) Then
28    conti=MsgBox("¿Continuar y sobrescribir los datos?",4,"La carpeta
       especificada ya existe")
29    if conti = 7 then
30      WScript.Echo ("Fin del script")
31      WScript.Quit()
32    else
33    end if
```

```

34 Else
35     Set crea_carpeta = CreateObject("Scripting.FileSystemObject")
36     crea_carpeta.createfolder Nombre_dir_datos & Nombre_fuente
37 End If
38
39 new1 = targetDir & "\" & basename & "_" & Int(k) & "." & objFSO.
    GetExtensionName(sourceFile)
40 'WScript.Echo ("new1: " & new1)
41
42 set fs = CreateObject("Scripting.FileSystemObject")
43 set w1 = objFSO.CreateTextFile(new1, True)
44
45 isNotSplit = True
46 with fs.OpenTextFile(sourceFile)
47     while Not .AtEndOfStream
48         if .Line <= max_lineas then
49             w1.WriteLine .ReadLine
50         else
51             k = k +1
52             max_lineas = lineas_por_archivo * k
53             new1 = targetDir & "\" & basename & "_" & Int(k) & "." & objFSO.
                GetExtensionName(sourceFile)
54             set fs = CreateObject("Scripting.FileSystemObject")
55             set w1 = objFSO.CreateTextFile(new1, True)
56             if( isNotSplit ) then
57                 isNotSplit = False
58                 WScript.Echo ("Dividiendo el archivo fuente")
59             end if
60         end if
61     end with
62 wend
63 numlines= .Line
64 WScript.Echo ("Número de líneas del archivo original: " & numlines)
65 WScript.Echo ("Se han creado: " & k & " archivos de " & lineas_por_archivo &
    " líneas.")
66 .Close
67 end with
68 w1.Close
69 if( isNotSplit ) then
70     WScript.Echo ("El archivo fuente no será dividido")
71 else
72     WScript.Echo ("El archivo fuente ha sido dividido")
73 end if
74 else
75     WScript.Echo ("El archivo de entrada no existe")
76 end if
77
78 MsgBox "¡Fin!"

```

Apéndice B

Códigos de Matlab utilizados

B.1. gen_lectura.m

```
1 % Este script se encarga de leer los archivos de datos y generar las estructuras
2 % necesarias para el procesado posterior.
3
4 % %%Descomentar cuando se analice un único archivo.
5 % clear all; clc; close all; % Se limpian valores, pantalla y cerramos figuras.
6 %
7 % nombre_fuente='20130603-172327'; % Nombre fuente del archivo a analizar.
8 %
9 % %%Log file
10 % fecha=date;
11 % nom_diario=['Logs/' fecha '_' nombre_fuente];
12 % diary(nom_diario)
13 %
14 % disp('XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX')
15 % disp('Comienzo de clasificación.m individual')
16 % disp('XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX')
17 % disp(datestr(now))
18 %
19 % clear nom_diario;
20
21 %% Obtención y clasificación de datos.
22
23 % 20130315-122405 350 Mb
24 % 20130318-154852 240 Mb
25 % 20130322-191010 43 Mb
26 % 20130326-102822 1.249 Mb
27
28 % 20130515-165440 481 Mb
29 % 20130521-180946 418 Mb
30 % 20130524-193631 720 Mb
31 % 20130529-155254 770 Mb
32 % 20130603-172327 560 Mb
33 % 20130607-162342 485 Mb
34 % 20130610-170542 967 Mb
35 % 20130617-114252 434 Mb
36 % 20130620-125108 841 Mb
```

```

37 % 20130626-105034 805 Mb
38 % 20130701-171200 584 Mb
39 % 20130705-200728 584 Mb
40
41 tic
42
43 % Ruta en la que se encuentra el archivo fuente.
44 ruta=['datos/' nombre_fuente '/'];
45 disp(['Inicio de obtención y clasificación de datos de ' nombre_fuente '.bst'])
46 dir_datos=dir(ruta);
47 N1=size(dir_datos,1);
48
49 Estadistica.Archivo=nombre_fuente;
50
51 % Bucle para calcular el número de directorios que hay en la carpeta.
52 for j1=1:N1,
53     a(j1)=dir_datos(j1).isdir;
54 end
55 N2=N1-length(find(a)); % Número de archivos a analizar.
56 clear dir_datos N1 a j1;
57 Estadistica.Particiones=N2;
58
59 % Bucle que recorrerá todos los archivos que han de analizarse.
60 for j1=1:N2,
61     % Nombre de los archivos que van a analizarse.
62     archivo=sprintf([ruta nombre_fuente '%.bst'],j1);
63     [fecha, hora, ModeS, ModeShex, Callsign, pais, IsOnGround, altitud,...
64         altitud2, latitud, longitud, ROC, CROC, GroundSpeed, Track,SquawkD,...
65         SquawkH]=textread(archivo, '%q%q%q%q%q%q%q%q%q%q%q%q%q%q%q%q',...
66         'delimiter',' ','');
67     % Lectura del archivo de datos. Contiene 17 parámetros distintos que se
68     % leerán como cell.
69
70     clear ModeS pais altitud2 CROC SquawkD Track SquawkH GroundSpeed;
71
72     [Vuelo]=func_matriculas(ModeShex); % Separación por matrículas.
73     m=length(Vuelo); % Número de aviones procesados.
74
75     % Bucle para recorrer todos los aviones y asignar sus parámetros
76     % correspondientes y guardarlos en forma de estructuras.
77     for j2=1:m,
78         % Fecha y hora del vuelo en formato string. El tiempo del vuelo es desde
79         % el comienzo del año.
80         % La altitud (ft), la latitud (rad), la longitud (rad), ROC (ft/s),
81         % GroundSpeed (kt) e IsOnGround (-) aparecen en formato string y se
82         % convierten en formato numérico.
83         Vuelo(j2).Fecha = datenum(fecha(Vuelo(j2).Posicion), 'yyyy/mm/dd');
84         Vuelo(j2).Tiempo = datenum(hora(Vuelo(j2).Posicion), 'HH:MM:SS.FFF');
85         Vuelo(j2).Altitud = str2double(altitud(Vuelo(j2).Posicion));
86         Vuelo(j2).Latitud = str2double(latitud(Vuelo(j2).Posicion));
87         Vuelo(j2).Longitud = str2double(longitud(Vuelo(j2).Posicion));
88         Vuelo(j2).ROC = str2double(ROC(Vuelo(j2).Posicion));
89 %         Vuelo(j2).GroundSpeed = str2double(GroundSpeed(Vuelo(j2).Posicion));
90         Vuelo(j2).IsOnGround=str2double(IsOnGround(Vuelo(j2).Posicion));
91         Vuelo(j2).Callsign2 = Callsign(Vuelo(j2).Posicion);
92     end
93
94     % Se limpian las variables utilizadas.

```

```

95  clear fecha hora Callsign latitud longitud posicion altitud Vueloaux ROC
96  clear ModeShex m IsOnGround;
97  Vuelo=rmfield(Vuelo,'Posicion');
98
99  %Se guardan los datos obtenidos en un archivo independiente del resto.
100 save_name=sprintf(['provi/Archivo_' nombre_fuente '/Parte' '_%i'],j1);
101 save(save_name,'Vuelo')
102 clear Vuelo save_name;
103
104 disp(sprintf(['Computado ' archivo ' (%i de %i).'],j1,N2))
105 end
106
107 clear kini kfin archivo ruta dir_datos;
108
109 Estadistica.Tiempos(1)=toc;
110 Nombre_est=['Resultados/Estadisticas/Estadistica_' nombre_fuente];
111 save(Nombre_est,'Estadistica');
112 disp(sprintf(['Fin de Obtención y clasificación de datos. Tiempo: %i s'],...
113  Estadistica.Tiempos(1)))

```

B.2. func_matriculas.m

```

1 function [Avion]=func_matriculas(dato)
2 %Dado un vector de datos en formato cell (formato en el que Matlab lee los
3 %datos), devuelve la posición en la que se encuentran datos idénticos en el
4 %archivo original y los datos en sí, según vayan apareciendo.
5
6 [m,n]=size(dato); %Dimensiones del dato a tratar.
7 k1=1; %Contador del número distintos de datos que aparecen.
8 for j1=1:m,
9  dato_act=dato{j1,1}; %Dato actual. Se utilizará para las comprobaciones.
10 %Condición para empezar a partir de la segunda componente. Para la primera no
11 %tiene sentido realizar la comprobación de si es igual o no a las componentes
12 %ya analizadas.
13 if j1~=1,
14  %¿Son el dato actual y alguno anterior iguales? Si la matrícula coincide,
15  %se pasa al siguiente avión.
16  if any(strcmp(dato_act,{Avion(:).Matricula})),
17   continue;
18  end
19 end
20 a=strcmp(dato_act,dato); %En qué posiciones se encuentra la matrícula.
21 %Se guardan las posiciones en las que ocurre esa matrícula y el dato.
22 Avion(k1).Posicion=find(a);
23 Avion(k1).Matricula=dato{Avion(k1).Posicion(1)};
24 k1=k1+1;%Se actualiza el contador, puesto que ha aparecido un dato nuevo.
25 end %Se pasa a la siguiente componente.

```

B.3. gen_union.m

```

1 %% Unión de los datos guardados.
2
3 disp('Inicio de la unión de los datos guardados')
4 tic
5
6 load(['Resultados/Estadisticas/Estadistica_' nombre_fuente(9:end) '.mat'])
7
8 direc=dir(['provi/' nombre_fuente]);
9 N4=length(direc)-2;
10
11 Vuelos=[];
12 if N4>0,
13     % Bucle que irá cargando el resto de archivos guardados.
14     for j1=1:N4
15         load_name=sprintf(['provi/' nombre_fuente '/Parte' '_%a'],j1);
16         load(load_name);
17         Vuelos=[Vuelos Vuelo]; % Lo asigna en la nueva estructura.
18         clear Vuelo load_name;
19     end
20 end
21
22 % delete_name=sprintf(['provi/fVuelo_' nombre_fuente '_*.mat']);
23 % delete(delete_name); % Se borran todos los archivos creados.
24 % clear N2 delete_name;
25 Estadistica.Tiempos(2)=toc;
26 disp(sprintf(['Fin de la unión de los datos guardados. Tiempo: %a s'],...
27     Estadistica.Tiempos(2)))
28
29
30 ea=Vuelos(1).Fecha(1);
31 eb=Vuelos(1).Tiempo(1);
32 ec=Vuelos(end).Fecha(end);
33 ed=Vuelos(end).Tiempo(end);
34 Estadistica.Inicio_archivo=[datestr(ea) ' ' datestr(eb,'HH:MM:SS')];
35 Estadistica.Final_archivo=[datestr(ec) ' ' datestr(ed,'HH:MM:SS')];
36 ee=ec-ea+ed-eb;
37 Estadistica.Intervalo=sprintf(['%a días'],ee);
38
39 clear ea eb ec ed ee;
40
41 %% Unión de las aeronaves con la misma matrícula.
42 % En esta celda se unen todos los aviones extraídos anteriormente que cuenten
43 % con la misma matrícula.
44
45 tic
46 disp('Inicio de Unión de las aeronaves con la misma matrícula.')
47
48 % Valor numérico que Matlab define como segundos que hay en un día.
49 diaseg=(datenum(2001,12,19,18,0,1)-datenum(2001,12,19,18,0,0))^-1;
50
51 k1=1; % Contador del avión a partir del cual se están buscando otros similares.
52
53 while true, % Bucle infinito. Recorre todos los aviones que se han encontrado.
54     matric=Vuelos(k1).Matricula; % Matrícula que se va a buscar.
55     % Búsqueda de coincidencias de matrículas, posición en la que se encuentran y
56     % número de aviones encontrados con la misma.
57     coin_matri=strcmp(matric,{Vuelos(:).Matricula});

```

```

58 pos_matri=find(coin_matri);
59 num_misma=length(pos_matri);
60
61 if num_misma>1, % Condición para que empiece por el segundo avión encontrado.
62     % Bucle dentro del cual se aumenta el tamaño del primer vuelo que aparece
63     % con una determinada matrícula, añadiéndole los vectores correspondientes
64     % a los otros aviones encontrados.
65     for j1=2:num_misma,
66         Vuelos(k1).Tiempo=[Vuelos(k1).Tiempo; Vuelos(pos_matri(j1)).Tiempo];
67         Vuelos(k1).Fecha=[Vuelos(k1).Fecha; Vuelos(pos_matri(j1)).Fecha];
68         Vuelos(k1).Altitud=[Vuelos(k1).Altitud; Vuelos(pos_matri(j1)).Altitud];
69         Vuelos(k1).Latitud=[Vuelos(k1).Latitud; Vuelos(pos_matri(j1)).Latitud];
70         Vuelos(k1).Longitud=[Vuelos(k1).Longitud;...
71             Vuelos(pos_matri(j1)).Longitud];
72         Vuelos(k1).ROC=[Vuelos(k1).ROC; Vuelos(pos_matri(j1)).ROC];
73 %         Vuelos(k1).GroundSpeed=[Vuelos(k1).GroundSpeed;...
74 %             Vuelos(pos_matri(j1)).GroundSpeed];
75         Vuelos(k1).IsOnGround=[Vuelos(k1).IsOnGround;...
76             Vuelos(pos_matri(j1)).IsOnGround];
77         Vuelos(k1).Callsign2=[Vuelos(k1).Callsign2;...
78             Vuelos(pos_matri(j1)).Callsign2];
79     end
80
81     % Bucle en el que se eliminarán los aviones que ya hayan sido guardados en
82     % su primera aparición.
83     for j1=num_misma:-1:2,
84         Vuelos(pos_matri(j1))=[];
85     end
86 end
87
88 % Inicialización de la fecha una vez se han unido todos los vuelos.
89
90 Vuelos(k1).Fecha2=(Vuelos(k1).Fecha-Vuelos(k1).Fecha(1));
91 Vuelos(k1).Tiempo2=Vuelos(k1).Tiempo;
92 Vuelos(k1).Tiempo=(Vuelos(k1).Tiempo-Vuelos(k1).Tiempo(1)+...
93     Vuelos(k1).Fecha2)*diasseg; % Inicialización del tiempo.
94
95 m=length(Vuelos); % Actualización del número de aviones.
96 k1=k1+1; % Actualización del contador del avión que ha de analizarse.
97 % Condición de salida del bucle infinito. Se saldrá si el valor del contador
98 % es mayor que el del número de aviones.
99 if k1>m,
100     break;
101 end
102 disp(sprintf(['Vuelos unidos: %a de %a.'],k1,m))
103 end
104
105 Vuelos=rmfield(Vuelos,'Fecha2');
106
107 Estadistica.Aviones=m;
108 clear diasseg k1 matric pos_matri coin_matri num_misma;
109
110 Estadistica.Tiempos(3)=toc;
111 disp(sprintf(['...
112     'Fin unión de las aeronaves con la misma matrícula. Tiempo: %a s.'],...
113     Estadistica.Tiempos(3)))
114
115 % Se guardan los datos obtenidos en un archivo independiente del resto.

```

```

116 save_name2=sprintf([ 'Leidos/Vuelos_ ' nombre_fuente(9:end) ] );
117 save(save_name2, 'Vuelos')
118 Nombre_est=[ 'Resultados/Estadisticas/Estadistica_ ' nombre_fuente(9:end) ];
119 save(Nombre_est, 'Estadistica');
120 clear save_name2;

```

B.4. gen_model_AL.m

```

1 %% Determinación del modelo de avión y compañía aérea.
2
3 tic
4 disp('Inicio de la determinación del modelo de avión y aerolínea.')
5
6 load([ 'Resultados/Estadisticas/Estadistica_ ' nombre_fuente(8:end) ])
7
8 for j1=1:m,
9     matricula=Vuelos(j1).Matricula;
10    a=strcmp(matricula,BD_modelos(:,2));
11    b=find(a);
12    if any(a),
13        Vuelos(j1).Modelo=BD_modelos(b,3);
14        modelo=Vuelos(j1).Modelo;
15        filtro_mod;
16        Vuelos(j1).Modelo2=modelo;
17        Vuelos(j1).Aerolinea=BD_modelos(b,4);
18        aerolinea=Vuelos(j1).Aerolinea;
19        filtro_AL;
20        Vuelos(j1).Aerolinea2=aerolinea;
21    else
22        busqueda=[ 'http://es.search.yahoo.com/search;_ylt=A7x9QbzyFXdRlx4AHhe' ...
23            'T.Qt.?p=site%3Ahttp%3A%2F%2Fwww.flightradar24.com%2Fdata%2Fairpl' ...
24            'anes%2F%20ModeS%3A%20' matricula '&fr2=sb-top&fr=sfp&rd=r1' ];
25        try
26            s=urlread(busqueda);
27        catch
28            try
29                pause(1.5)
30                disp('Segundo intento')
31                s=urlread(busqueda);
32            catch
33                try
34                    pause(4)
35                    disp('Tercer intento')
36                    s=urlread(busqueda);
37                catch
38                    pause(8)
39                    disp('Cuarto intento')
40                    s=urlread(busqueda);
41                end
42            end
43        end
44
45        pos1=strfind(s,'Type: ');
46        if isempty(pos1)==0,

```

```

47     pos1=pos1(1);
48     end
49     pos2=strfind(s(pos1:pos1+50),'Serial');
50     if isempty(pos2)==0,
51         pos2=pos2(1);
52         modelo=s(pos1+6:pos1+pos2-2);
53     else
54         pos2=strfind(s(pos1:pos1+50),'Serial');
55         modelo=s(pos1+6:pos1+pos2-3);
56     end
57     modelo={modelo};
58     Vuelos(j1).Modelo=modelo;
59     filtro_mod;
60     Vuelos(j1).Modelo_cor=modelo;
61     [m1,n1]=size(BD_modelos);
62     BD_modelos(m1+1,1)={m1};
63     BD_modelos(m1+1,2)={matricula};
64     BD_modelos(m1+1,3)=Vuelos(j1).Modelo;
65
66     pos1=strfind(s,'Airline: ');
67     if isempty(pos1)==0,
68         pos1=pos1(1);
69     end
70     pos2=strfind(s(pos1:pos1+60),'Images');
71     if isempty(pos2)==0,
72         pos2=pos2(1);
73         aerolinea=s(pos1+9:pos1+pos2-2);
74     else
75         pos2=strfind(s(pos1:pos1+50),'Images');
76         aerolinea=s(pos1+9:pos1+pos2-3);
77     end
78     aerolinea={aerolinea};
79     filtro_AL
80     Vuelos(j1).Aerolinea=aerolinea;
81     BD_modelos(m1+1,4)=Vuelos(j1).Aerolinea;
82 end
83 %     disp(sprintf(['Vuelo caracterizado número: %a de %a.'],j1,m))
84
85     save('datos/BD_modelos','BD_modelos');
86 end
87
88 disp('Fin de la determinación del modelo de avión y aerolínea.')
89 Estadistica.Tiempos(4)=toc;
90
91 clear BD_modelos a b matricula pos1 pos2 s busqueda m1 n1 a b aerolinea;
92 clear aerolinea_num matricula modelo modelo_num;

```

B.5. gen_sep.m

```

1 %% Separación de las aeronaves por vuelos.
2 % En esta celda, se separan las aeronaves con la misma matrícula según los
3 % distintos vuelos que hayan realizado. Para esto, se buscará que entre un vuelo
4 % y otro haya pasado un cierto tiempo sin que el avión haya emitido señal.
5

```

```

6 tic
7 disp(('Inicio de la separación de las aeronaves por vuelos.'))
8
9 dia_inicio=datetime('00:00:00.000','HH:MM:SS.FFF');
10
11 % Contador que aumentará cada vez que se detecte un vuelo nuevo extraído de un
12 % avión con varios vuelos.
13 k1=0;
14 j1=0; % Contador que recorrerá todos los Vuelos.
15
16 % Diferencia de tiempo que debe haber para considerar dos vuelos distintos
17 % (segundos).
18 diftemp=300;
19 while true, % Bucle infinito hasta que se cumpla la condición de parada.
20     j1=j1+1; % Actualización del contador que recorre todos los Vuelos.
21     % Condición de parada del bucle infinito. Cuando el número del contador de
22     % vuelos sea mayor que el número de vuelos detectados, sale del bucle.
23     if j1>m+k1,
24         break;
25     end
26
27     tiempo_inicio=Vuelos(j1).Tiempo2(1)-dia_inicio;
28     fecha_inicio=Vuelos(j1).Fecha(1)+tiempo_inicio;
29     Vuelos(j1).Inicio=datestr(fecha_inicio);
30
31     n=length(Vuelos(j1).Tiempo); % Número de datos del avión j1.
32
33     if n<2,
34         Vuelos(j1)=[];
35         continue;
36     end
37
38     for j2=2:n, % Bucle que recorre todas las componentes de un mismo avión.
39         % Condición: si el tiempo entre dos valores consecutivos es mayor que
40         % diftemp, se considera un vuelo distinto.
41
42         if Vuelos(j1).Tiempo(j2)-Vuelos(j1).Tiempo(j2-1)>diftemp,
43             k1=k1+1; % Se aumenta el contador de vuelos nuevos.
44             % Se asignan las subestructuras al vuelo nuevo.
45             Vuelos(m+k1).Matricula=Vuelos(j1).Matricula;
46             Vuelos(m+k1).Tiempo=Vuelos(j1).Tiempo(j2:n)-Vuelos(j1).Tiempo(j2);
47             Vuelos(m+k1).Tiempo2=Vuelos(j1).Tiempo2(j2:n);
48             Vuelos(m+k1).Fecha=Vuelos(j1).Fecha(j2:n);
49             Vuelos(m+k1).Altitud=Vuelos(j1).Altitud(j2:n);
50             Vuelos(m+k1).Latitud=Vuelos(j1).Latitud(j2:n);
51             Vuelos(m+k1).Longitud=Vuelos(j1).Longitud(j2:n);
52             Vuelos(m+k1).ROC=Vuelos(j1).ROC(j2:n);
53             Vuelos(m+k1).IsOnGround=Vuelos(j1).IsOnGround(j2:n);
54 %             Vuelos(m+k1).GroundSpeed=Vuelos(j1).GroundSpeed(j2:n);
55             Vuelos(m+k1).Callsign2=Vuelos(j1).Callsign2(j2:n);
56             Vuelos(m+k1).Modelo=Vuelos(j1).Modelo;
57             Vuelos(m+k1).Aerolinea=Vuelos(j1).Aerolinea;
58             Vuelos(m+k1).Modelo2=Vuelos(j1).Modelo2;
59             Vuelos(m+k1).Aerolinea2=Vuelos(j1).Aerolinea2;
60
61             % Se eliminan del original una vez han sido copiados.
62             Vuelos(j1).Fecha(j2:end)=[];
63             Vuelos(j1).Tiempo(j2:end)=[];

```

```

64      Vuelos(j1).Tiempo2(j2:end)=[];
65      Vuelos(j1).Altitud(j2:end)=[];
66      Vuelos(j1).Latitud(j2:end)=[];
67      Vuelos(j1).Longitud(j2:end)=[];
68      Vuelos(j1).ROC(j2:end)=[];
69 %      Vuelos(j1).GroundSpeed(j2:end)=[];
70      Vuelos(j1).Callsign2(j2:end)=[];
71      Vuelos(j1).IsOnGround(j2:end)=[];
72 %      disp(sprintf(['Vuelos separados: %. Vuelos totales: %.'],j1,m+k1))
73 %      % Se sale del bucle for que recorría las componentes del avión j1,
74 %      % puesto que ya ha separado el primer vuelo que este contenía.
75      break;
76  end % Acaba el if planteado.
77 end % Acaba el bucle que recorría todas las posiciones del avión.
78 end % Acaba el bucle que recorría todos los Vuelos.
79
80 Vuelos=rmfield(Vuelos,'Fecha');
81 Vuelos=rmfield(Vuelos,'Tiempo2');
82
83 Estadistica.Vuelos=m+k1; % Número de vuelos encontrados.
84
85 clear diftemp n tiempo_inicio fecha_inicio dia_inicio;
86 disp(['Fin de la separación de las aeronaves por vuelos.'])
87
88 Estadistica.Tiempos(5)=toc;
89 Nombre_est=['Resultados/Estadisticas/Estadistica_' nombre_fuente(8:end)];
90 save(Nombre_est,'Estadistica');

```

B.6. gen_CS.m

```

1 %% Callsign
2 % Obtiene un CS válido para cada vuelo. Es necesario debido a que aparecen
3 % algunos CS en blanco.
4 tic
5 disp(['Inicio de la determinación del Callsign.'])
6
7 m1=length(Vuelos);
8 for j=1:m1,
9     n=length(Vuelos(j).Callsign2);
10    for j2=1:n,
11        provi=cell2mat(Vuelos(j).Callsign2(j2));
12        if length(provi)>3,
13            Vuelos(j).Callsign=provi;
14            Vuelos(j).Destino=func_filtro_destino(Vuelos(j).Callsign);
15            break;
16        end
17    end
18 end
19
20 % Vuelos=rmfield(Vuelos,'Callsign2');
21
22 disp(['Fin de la determinación del Callsign.'])
23 Estadistica.Tiempos(6)=toc;

```

B.7. gen_estado.m

```

1 %% Determinación del estado del avión.
2 % Esta celda se encarga básicamente de aplicar un filtro para separar aquellos
3 % vuelos que despegan y aterrizan del aeropuerto de Sevilla.
4 % Se llama a la función func_aeropuertos, la cual determina si está
5 % saliendo/llegando de Sevilla o no.
6
7 tic
8 disp('Inicio de la determinación del estado de los vuelos.')
```

```

9
10 numvuelos=m+k1; % Número de vuelos detectados.
11 % Altura a partir de la cual se definirá el estado de una aeronave.
12 difaltura=3e3;
13 % Inicialización de los contadores que generarán los vectores de las coordenadas
14 % en las que se encuentra cada vuelo según su estado.
15 k2=0; k3=0; k4=0; k5=0; k6=0;
16 k7=0; k8=0; k9=0; k10=0;
17
18 % Tolerancias permitidas para considerar que un avión sale/proviene de Sevilla.
19 TOLdiflambda=0.27357;
20 TOLdifphi=0.03641;
21
22 alt_intervalo=500; % Rango de altitud según el que se agruparán los datos.
23
24 for j1=1:numvuelos, % Bucle que recorrerá todos los vuelos detectados.
25
26     Vuelos(j1).Estadistica.Medias=[];
27     Vuelos(j1).Estadistica.Des_Tip=[];
28     Vuelos(j1).Estadistica.Max=[];
29     Vuelos(j1).Estadistica.Min=[];
30     Vuelos(j1).Estadistica.Intervalo=[];
31     Vuelos(j1).Estadistica.Inicio=[];
32     Vuelos(j1).Estadistica.Fin=[];
33
34     Altura=Vuelos(j1).Altitud(:); % Se pasan las alturas del avión a un vector.
35     % Condición de altura: Si la la diferencia de altura es mayor que difaltura,
36     % el avión se encuentra en ascenso. Si es menor que -difaltura, se encuentra
37     % en descenso y si no se cumple ninguna, está en crucero.
38     if Altura(end)-Altura(1)>difaltura,
39         y=func_aeropuertos(Vuelos(j1),1,TOLdiflambda,TOLdifphi);
40         if y==1,
41             k2=k2+1;
42             % Los vectores possubida, posdescenso y poscrucero guardan las
43             % posiciones en las que se encuentran los vuelos según su estado.
44             posdespsev(k2)=j1;
45             Vuelos(j1).Estado='Despegue';
46             Vuelos(j1)=func_estad(Vuelos(j1),1,alt_intervalo);
47         elseif y==2,
48             k7=k7+1;
49             posdespmal(k7)=j1;
50             Vuelos(j1).Estado='Despegue_Malaga';
51             Vuelos(j1)=func_estad(Vuelos(j1),1,alt_intervalo);
52         elseif y==3,
53             k9=k9+1;
54             posdespjer(k9)=j1;
55             Vuelos(j1).Estado='Despegue_Jerez';
```

```

56     Vuelos(j1)=func_estad(Vuelos(j1),1,alt_intervalo);
57     else
58         k3=k3+1;
59         possub(k3)=j1;
60         Vuelos(j1).Estado='Subida';
61     end
62 elseif Altura(end)-Altura(1)<-difaltura,
63     y=func_aeropuertos(Vuelos(j1),2,TOLdiflambda,TOLdifphi);
64     if y==1,
65         k4=k4+1;
66         posatesev(k4)=j1;
67         Vuelos(j1).Estado='Aterrizaje';
68         Vuelos(j1)=func_estad(Vuelos(j1),2,alt_intervalo);
69     elseif y==2,
70         k8=k8+1;
71         posatemal(k8)=j1;
72         Vuelos(j1).Estado='Aterrizaje_Malaga';
73         Vuelos(j1)=func_estad(Vuelos(j1),2,alt_intervalo);
74     elseif y==3,
75         k10=k10+1;
76         posatejer(k10)=j1;
77         Vuelos(j1).Estado='Aterrizaje_Jerez';
78         Vuelos(j1)=func_estad(Vuelos(j1),2,alt_intervalo);
79     else
80         k5=k5+1;
81         posdesc(k5)=j1;
82         Vuelos(j1).Estado='Descenso';
83     end
84 else
85     k6=k6+1;
86     poscrucero(k6)=j1;
87     Vuelos(j1).Estado='Crucero';
88 end
89
90 %% Comprobación número de líneas.
91
92 num_comp(j1)=length(Vuelos(j1).Altitud);
93 end
94
95 Estadistica.Vuelos_desp_sev=k2;
96 Estadistica.Vuelos_asc=k3;
97 Estadistica.Vuelos_ate_sev=k4;
98 Estadistica.Vuelos_desc=k5;
99 Estadistica.Vuelos_Crucero=k6;
100
101 Estadistica.Vuelos_desp_mal=k7;
102 Estadistica.Vuelos_ate_mal=k8;
103 Estadistica.Vuelos_desp_jer=k9;
104 Estadistica.Vuelos_ate_jer=k10;
105
106 if k7>0,
107     Vuelos_desp_mal=Vuelos(posdespmal(:));
108 end
109 if k8>0,
110     Vuelos_ate_mal=Vuelos(posatemal(:));
111 end
112 if k9>0,
113     Vuelos_desp_jer=Vuelos(posdespjer(:));

```

```

114 end
115 if k10>0,
116     Vuelos_ate_jer=Vuelos(posatejer(:));
117 end
118
119 Estadistica.Num_lineas=sum(num_comp);
120 Vuelos_desp_sev=Vuelos(posdespsev(:));
121 Vuelos_sub=Vuelos(possusb(:));
122 Vuelos_ate_sev=Vuelos(posatesev(:));
123 Vuelos_desc=Vuelos(posdesc(:));
124 Vuelos_pos_cru=Vuelos(poscrucero(:));
125
126 Estadistica.Densidad_desp=func_densidad(Vuelos_desp_sev,alt_intervalo);
127 Estadistica.Densidad_ate=func_densidad(Vuelos_ate_sev,alt_intervalo);
128
129 [Vuelos_desp_sev, k09, k27]=func_cabecera(Vuelos_desp_sev,1);
130 Estadistica.Cab_Desp_09=k09;
131 Estadistica.Cab_Desp_27=k27;
132 [Vuelos_ate_sev, k09, k27]=func_cabecera(Vuelos_ate_sev,2);
133 Estadistica.Cab_Ate_09=k09;
134 Estadistica.Cab_Ate_27=k27;
135
136 clear m Altura TOLdiflambda TOLdifphi difaltura numvuelos poscrucero clear y k1;
137 clear k2 k3 k4 k5 k6 j1 j2 posdespsev possusb posatesev posdesc poscrucero;
138 clear Nombre_tot num_comp nTOL phi_ancho_pista alt_intervalo posdespmal;
139 clear posdespjjer posatemal posatejer Vuelos_pos_cru Vuelos_sub Vuelos_desc;
140 clear j k10 k7 k8 k9 k09 k27 m;
141
142 disp('Fin de la determinación del estado de los vuelos.')
143 Estadistica.Tiempos(7)=toc;
144 Nombre_est=['Resultados/Estadisticas/Estadistica_' nombre_fuente(8:end)];
145 save(Nombre_est,'Estadistica');

```

B.8. func_aeropuertos.m

```

1 function y=func_aeropuerto(Vuelo,estado,TOLdiflambda,TOLdifphi)
2
3 % Filtro de aviones que despegan/aterrizan en el aeropuerto de Sevilla.
4 % Adaptado también para Málaga y Jerez. Las tolerancias de estos últimos son
5 % inventadas.
6 % La salida será 1 si el aeropuerto es de Sevilla y 0 si no lo es.
7 % 2 si se trata del aeropuerto de Málaga y 3 si es el de Jerez.
8 % estado=1, implica que está subiendo. estado=2 implica que está bajando.
9
10 TOLdiflambdaAGP=0.3;
11 TOLdifphiAGP=0.3;
12 TOLdiflambdaXRY=0.3;
13 TOLdifphiXRY=0.3;
14
15 % Longitud y latitud del aeropuerto de Sevilla.
16 lambdaSVQ=-5.893106; phiSVQ=37.418;
17 % Longitud y latitud del aeropuerto de Málaga.
18 lambdaAGP=-(4+29/60+52.40/3600); phiAGP=(36+40/60+32/3600);
19 % Longitud y latitud del aeropuerto de Málaga.

```

```

20 lambdaXRY=-(6+3/60+36.40/3600); phiXRY=(36+44/60+40.64/3600);
21 if estado==1,
22     if any(Vuelo.IsOnGround==0)
23         a=find(Vuelo.IsOnGround==0);
24         b=a(1);
25         phiref=Vuelo.Latitud(b);
26         lambdaref=Vuelo.Longitud(b);
27     else
28         phiref=0;
29         lambdaref=0;
30     end
31 elseif estado==2,
32     if any(Vuelo.IsOnGround==-1)
33         a=find(Vuelo.IsOnGround==-1);
34         b=a(1)-1;
35         phiref=Vuelo.Latitud(b);
36         lambdaref=Vuelo.Longitud(b);
37     elseif any(Vuelo.IsOnGround==0)
38         a=find(Vuelo.IsOnGround==0);
39         b=a(end);
40         phiref=Vuelo.Latitud(b);
41         lambdaref=Vuelo.Longitud(b);
42     else
43         phiref=0;
44         lambdaref=0;
45     end
46 else
47     phiref=0; lambdaref=0;
48 end
49 % Cálculo de las diferencias con las coordenadas de referencia.
50 diflambdaSVQ=abs(lambdaref-lambdaSVQ); difphiSVQ=abs(phiref-phiSVQ);
51 diflambdaAGP=abs(lambdaref-lambdaAGP); difphiAGP=abs(phiref-phiAGP);
52 diflambdaXRY=abs(lambdaref-lambdaXRY); difphiXRY=abs(phiref-phiXRY);
53 if diflambdaSVQ<TOLdiflambda && difphiSVQ<TOLdifphi,
54     y=1;
55 elseif diflambdaAGP<TOLdiflambdaAGP && difphiAGP<TOLdifphiAGP,
56     y=2;
57 elseif diflambdaXRY<TOLdiflambdaXRY && difphiXRY<TOLdifphiXRY,
58     y=3;
59 else
60     y=0;
61 end

```

B.9. func_estad.m

```

1 function Vuelos=func_estad(Vuelos,estado,alt_intervalo)
2 % Cálculo de las estadísticas referentes a la calidad de la señal de una
3 % cierta aeronave.
4 % estado=1 --> Ascenso; estado=2 --> Descenso
5
6 Vuelos.Estadistica.Altitud_max=max(Vuelos.Altitud);
7
8 if estado==1,
9     a0=find(Vuelos.IsOnGround==0);

```

```

10  Vuelos.Estadistica.Inicio=a0(1);
11  try
12      b=find(abs(Vuelos.ROC)<150 & Vuelos.Altitud>24500 & ...
13          abs(Vuelos.Altitud-round(Vuelos.Altitud/1e3)*1e3)<100);
14      Vuelos.Estadistica.Fin=b(1);
15  catch
16      Vuelos.Estadistica.Fin=length(Vuelos.ROC);
17  end
18
19  ini=Vuelos.Estadistica.Inicio;
20  fin=Vuelos.Estadistica.Fin;
21
22  alt=alt_intervalo;
23  j=1;
24  while alt<Vuelos.Estadistica.Altitud_max+alt_intervalo ,
25      try
26          Vuelos.Estadistica.Altitud(j).Posiciones=find(Vuelos.Altitud(ini:fin)>
27              alt-alt_intervalo & Vuelos.Altitud(ini:fin)<=alt);
28          Vuelos.Estadistica.Altitud(j).Cantidad=length(Vuelos.Estadistica.
29              Altitud(j).Posiciones);
30          Vuelos.Estadistica.Altitud(j).Altitud=[alt-alt_intervalo alt];
31      catch
32      end
33      alt=alt+alt_intervalo;
34      j=j+1;
35  end
36  Vuelos.Estadistica.Altitud(j).Posiciones=find(Vuelos.Altitud(ini:fin)>alt-
37      alt_intervalo & Vuelos.Altitud(ini:fin)<=alt);
38  Vuelos.Estadistica.Altitud(j).Cantidad=length(Vuelos.Estadistica.Altitud(j).
39      Posiciones);
40  Vuelos.Estadistica.Altitud(j).Altitud=[alt-alt_intervalo alt];
41
42  elseif estado==2,
43  if any(Vuelos.IsOnGround==-1),
44      a0=find(Vuelos.IsOnGround==-1);
45      Vuelos.Estadistica.Fin=a0(1)-1;
46  else
47      Vuelos.Estadistica.Fin=length(Vuelos.Latitud);
48  end
49  try
50      b=find(abs(Vuelos.ROC)<150 & Vuelos.Altitud>24500 & abs(Vuelos.Altitud-
51          round(Vuelos.Altitud/1e3)*1e3)<100);
52      Vuelos.Estadistica.Inicio=b(end);
53  catch
54      Vuelos.Estadistica.Inicio=1;
55  end
56
57  ini=Vuelos.Estadistica.Inicio;
58  fin=Vuelos.Estadistica.Fin;
59
60  alt_intervalo=500;
61  alt=500;
62  j=1;
63  while alt<Vuelos.Estadistica.Altitud_max+alt_intervalo ,
64      try
65          Vuelos.Estadistica.Altitud(j).Posiciones=find(Vuelos.Altitud(ini:fin)>
66              alt-alt_intervalo & Vuelos.Altitud(ini:fin)<=alt);

```

```

61      Vuelos.Estadistica.Altitud(j).Cantidad=length(Vuelos.Estadistica.
        Altitud(j).Posiciones);
62      Vuelos.Estadistica.Altitud(j).Altitud=[alt-alt_intervalo alt];
63      catch
64      end
65      alt=alt+alt_intervalo;
66      j=j+1;
67  end
68  Vuelos.Estadistica.Altitud(j).Posiciones=find(Vuelos.Altitud(ini:fin)>alt-
        alt_intervalo & Vuelos.Altitud(ini:fin)<=alt);
69  Vuelos.Estadistica.Altitud(j).Cantidad=length(Vuelos.Estadistica.Altitud(j).
        Posiciones);
70  Vuelos.Estadistica.Altitud(j).Altitud=[alt-alt_intervalo alt];
71
72 end
73
74 Dif_tiempo=Vuelos.Tiempo(ini+1:fin)-Vuelos.Tiempo(ini:fin-1);
75 Vuelos.Estadistica.Dif_tiempo=Vuelos.Tiempo(ini+1:fin)-Vuelos.Tiempo(ini:fin-1);
76 Vuelos.Estadistica.Media=mean(Dif_tiempo);
77 Vuelos.Estadistica.Des_Tip=std(Dif_tiempo);
78 Vuelos.Estadistica.Max=max(Dif_tiempo);
79 Vuelos.Estadistica.Min=min(Dif_tiempo);
80 Vuelos.Estadistica.Intervalo=Vuelos.Tiempo(fin)-Vuelos.Tiempo(ini);

```

B.10. func_cabecera.m

```

1 function [Vuelos, k09, k27]=func_cabecera(Vuelos,estado)
2
3 % Esta función determina la cabecera que utiliza cada vuelo y extrae un valor
4 % estadístico del uso de cada una.
5 % estado==1 si asciende y estado==2 si desciende.
6 m=length(Vuelos);
7 k09=0; k27=0;
8 for j1=1:m,
9     if estado==1, % Ascensos
10         if any(Vuelos(j1).IsOnGround==0) & any(Vuelos(j1).IsOnGround==-1)
11             a=find(Vuelos(j1).IsOnGround==0);
12             b=a(1);
13             diflambda=Vuelos(j1).Longitud(b+3)-Vuelos(j1).Longitud(b-1);
14         else
15             a=find(Vuelos(j1).Altitud>1e3);
16             b=a(1);
17             diflambda=Vuelos(j1).Longitud(b+4)-Vuelos(j1).Longitud(b);
18         end
19         if diflambda>0,
20             Vuelos(j1).Cabecera='09';
21             k09=k09+1;
22         else
23             Vuelos(j1).Cabecera='27';
24             k27=k27+1;
25         end
26     elseif estado==2, % Descensos
27         if any(Vuelos(j1).IsOnGround==0) & any(Vuelos(j1).IsOnGround==-1)
28             a=find(Vuelos(j1).IsOnGround==0);

```

```

29         b=a(1);
30         diflambda=Vuelos(j1).Longitud(b)-Vuelos(j1).Longitud(b-4);
31     else
32         diflambda=Vuelos(j1).Longitud(end)-Vuelos(j1).Longitud(end-4);
33     end
34     if diflambda<0,
35         Vuelos(j1).Cabecera='27';
36         k27=k27+1;
37     else
38         Vuelos(j1).Cabecera='09';
39         k09=k09+1;
40     end
41 else
42     %No hacer nada.
43 end
44 end

```

B.11. gen_retraso.m

```

1 %% Filtro por retrasos
2
3 tic
4 disp('Inicio del filtro de los retrasos.')
5
6 % Destinos y orígenes
7 m=length(Vuelos_desp_sev);
8 for j=1:m,
9     Vuelos_desp_sev(j).Destino=func_filtro_destino(Vuelos_desp_sev(j).Callsign);
10 end
11 m=length(Vuelos_ate_sev);
12 for j=1:m,
13     Vuelos_ate_sev(j).Destino=func_filtro_destino(Vuelos_ate_sev(j).Callsign);
14 end
15
16
17 % Sevilla
18 [Vuelos_desp_sev_filtro,eliminados1]=func_filtro_retraso(Vuelos_desp_sev,1);
19 [Vuelos_ate_sev_filtro,eliminados2]=func_filtro_retraso(Vuelos_ate_sev,2);
20
21 Estadistica.Eliminados_desp_sev=eliminados1;
22 Estadistica.Eliminados_ate_sev=eliminados2;
23 Estadistica.Vuelos_desp_filtro_sev=length(Vuelos_desp_sev_filtro);
24 Estadistica.Vuelos_ate_filtro_sev=length(Vuelos_ate_sev_filtro);
25
26 % Málaga
27 [Vuelos_desp_mal_filtro,eliminados1]=func_filtro_retraso(Vuelos_desp_mal,1);
28 [Vuelos_ate_mal_filtro,eliminados2]=func_filtro_retraso(Vuelos_ate_mal,2);
29
30 Estadistica.Eliminados_desp_mal=eliminados1;
31 Estadistica.Eliminados_ate_mal=eliminados2;
32
33 try
34 Estadistica.Vuelos_desp_filtro_mal=length(Vuelos_desp_mal_filtro);
35 Estadistica.Vuelos_ate_filtro_mal=length(Vuelos_ate_mal_filtro);

```

```

36 end
37
38 % Jerez
39 [Vuelos_desp_jer_filtro, eliminados1]=func_filtro_retraso(Vuelos_desp_jer,1);
40 [Vuelos_ate_jer_filtro, eliminados2]=func_filtro_retraso(Vuelos_ate_jer,2);
41
42 Estadistica.Eliminados_desp_jer=eliminados1;
43 Estadistica.Eliminados_ate_jer=eliminados2;
44
45 try
46 Estadistica.Vuelos_desp_filtro_jer=length(Vuelos_desp_jer_filtro);
47 Estadistica.Vuelos_ate_filtro_jer=length(Vuelos_ate_jer_filtro);
48 end
49
50 clear eliminados1 eliminados2 j m;
51
52 disp('Fin del filtro de los retrasos.')
53 Estadistica.Tiempos(8)=toc;

```

B.12. func_filtro_retraso.m

```

1 function [Vuelos, vec_eliminados]=func_filtro_retraso(Vuelos, estado)
2
3 % Esta función determina los puntos iniciales y finales de cada ascenso/descenso
4 % estado==1 --> Ascenso; estado==2 --> Descenso;
5 eliminados=0;
6 eliminados2=0;
7 eliminados3=0;
8 eliminados4=0;
9
10 alt_intervalo=500;
11
12 m=length(Vuelos);
13 for j1=m:-1:1,
14     if estado==1,
15         if any(Vuelos(j1).Altitud>35500),
16             b=find(Vuelos(j1).Altitud>35500);
17             pos_fin=min(b(1), Vuelos(j1).Estadistica.Fin);
18         else
19             pos_fin=Vuelos(j1).Estadistica.Fin;
20             if length(Vuelos(j1).Tiempo)>pos_fin,
21                 pos_fin=pos_fin+1;
22             end
23         end
24         if any(Vuelos(j1).Altitud<=4000), %& Vuelos(j1).Altitud>500,
25             a=find(Vuelos(j1).Altitud<=4000); %& Vuelos(j1).Altitud>500);
26             pos_ini=a(end);
27         else
28 %             pos_ini=Vuelos(j1).Estadistica.Inicio;
29             Vuelos(j1)=[];
30             eliminados=eliminados+1;
31             continue;
32         end
33         if pos_fin<=pos_ini+1,

```

```

34     Vuelos(j1) = [];
35     eliminados = eliminados + 1;
36     continue;
37 end
38 if Vuelos(j1).Altitud(pos_ini+1) > 8e3,
39     Vuelos(j1) = [];
40     eliminados = eliminados + 1;
41     continue;
42 end
43 Vuelos(j1).Tiempo(pos_fin:end) = [];
44 Vuelos(j1).Altitud(pos_fin:end) = [];
45 Vuelos(j1).Latitud(pos_fin:end) = [];
46 Vuelos(j1).Longitud(pos_fin:end) = [];
47 Vuelos(j1).ROC(pos_fin:end) = [];
48 %     Vuelos(j1).GroundSpeed(pos_fin:end) = [];
49 Vuelos(j1).IsOnGround(pos_fin:end) = [];
50 Vuelos(j1).Tiempo(1:pos_ini) = [];
51 Vuelos(j1).Altitud(1:pos_ini) = [];
52 Vuelos(j1).Latitud(1:pos_ini) = [];
53 Vuelos(j1).Longitud(1:pos_ini) = [];
54 Vuelos(j1).ROC(1:pos_ini) = [];
55 %     Vuelos(j1).GroundSpeed(1:pos_ini) = [];
56 Vuelos(j1).IsOnGround(1:pos_ini) = [];
57 elseif estado == 2,
58     if any(Vuelos(j1).Altitud <= 3000),
59         a = find(Vuelos(j1).Altitud <= 3000);
60         pos_fin = a(1);
61     else
62         pos_fin = Vuelos(j1).Estadistica.Fin;
63         if length(Vuelos(j1).Tiempo) > pos_fin,
64             pos_fin = pos_fin + 1;
65         end
66     end
67     if any(Vuelos(j1).Altitud > 32000),
68         b = find(Vuelos(j1).Altitud > 32000);
69         pos_ini = max(b(end), Vuelos(j1).Estadistica.Inicio);
70     else
71         pos_ini = Vuelos(j1).Estadistica.Inicio;
72     end
73     if Vuelos(j1).Altitud(pos_fin-1) > 4.5e3,
74         Vuelos(j1) = [];
75         eliminados = eliminados + 1;
76         continue;
77     end
78     if pos_fin <= pos_ini + 1,
79         Vuelos(j1) = [];
80         eliminados = eliminados + 1;
81         continue;
82     end
83     Vuelos(j1).Tiempo(pos_fin:end) = [];
84     Vuelos(j1).Altitud(pos_fin:end) = [];
85     Vuelos(j1).Latitud(pos_fin:end) = [];
86     Vuelos(j1).Longitud(pos_fin:end) = [];
87     Vuelos(j1).ROC(pos_fin:end) = [];
88 %     Vuelos(j1).GroundSpeed(pos_fin:end) = [];
89     Vuelos(j1).IsOnGround(pos_fin:end) = [];
90     Vuelos(j1).Tiempo(1:pos_ini) = [];
91     Vuelos(j1).Altitud(1:pos_ini) = [];

```

```

92     Vuelos(j1).Latitud(1:pos_ini)=[];
93     Vuelos(j1).Longitud(1:pos_ini)=[];
94     Vuelos(j1).ROC(1:pos_ini)=[];
95 %     Vuelos(j1).GroundSpeed(1:pos_ini)=[];
96     Vuelos(j1).IsOnGround(1:pos_ini)=[];
97 end
98     Vuelos(j1).Estadistica=[];
99     Vuelos(j1)=func_estad2(Vuelos(j1),estado,alt_intervalo);
100    if Vuelos(j1).Estadistica.Max>30,
101        eliminados2=eliminados2+1;
102    end
103    if Vuelos(j1).Estadistica.Media>10,
104        eliminados3=eliminados3+1;
105    end
106    if Vuelos(j1).Estadistica.Max>30 | Vuelos(j1).Estadistica.Media>10,
107        eliminados4=eliminados4+1;
108    end
109    if Vuelos(j1).Estadistica.Max>30 | Vuelos(j1).Estadistica.Media>10, % Filtro
        del retraso máximo de 30 segundos o media de 10 segundos.
110        Vuelos(j1)=[];
111    else
112 %        Vuelos(j1).Estadistica=[];
113    end
114 end
115 vec_eliminados=[eliminados eliminados2 eliminados3 eliminados4];

```

B.13. gen_repr.m

```

1 %% Cálculo de las subestructuras necesarias para la representación de los datos
2 % (x_lambert, y_lambert, dist,...)
3
4 disp('Inicio de los cálculos para la representación de trayectorias.')
5
6 tic
7
8 [Vuelos, kas]=func_repr(Vuelos);
9 disp(kas)
10 Nombre_tot=['Resultados/Vuelos/Vuelos_' nombre_fuente(8:end)];
11 save(Nombre_tot,'Vuelos');
12 clear Vuelos;
13
14 Vuelos_desp_sev=func_repr(Vuelos_desp_sev);
15 Vuelos_desp_sev_filtro=func_repr(Vuelos_desp_sev_filtro);
16 Vuelos_ate_sev=func_repr(Vuelos_ate_sev);
17 Vuelos_ate_sev_filtro=func_repr(Vuelos_ate_sev_filtro);
18
19 Vuelos_desp_mal=func_repr(Vuelos_desp_mal);
20 Vuelos_desp_mal_filtro=func_repr(Vuelos_desp_mal_filtro);
21 Vuelos_ate_mal=func_repr(Vuelos_ate_mal);
22 Vuelos_ate_mal_filtro=func_repr(Vuelos_ate_mal_filtro);
23
24 Vuelos_desp_jer=func_repr(Vuelos_desp_jer);
25 Vuelos_desp_jer_filtro=func_repr(Vuelos_desp_jer_filtro);
26 Vuelos_ate_jer=func_repr(Vuelos_ate_jer);

```

```

27 Vuelos_ate_jer_filtro=func_repr(Vuelos_ate_jer_filtro);
28
29 Estadistica.Fecha_analisis=datestr(now);
30
31 clear eliminados1 eliminados2 k09f k27f;
32
33 %% Fijación de los puntos iniciales y finales de los despegues y aterriajes
34
35 m1=length(Vuelos_desp_sev);
36 m2=length(Vuelos_ate_sev);
37 m3=length(Vuelos_desp_sev_filtro);
38 m4=length(Vuelos_ate_sev_filtro);
39
40 for j1=1:m1,
41     Vuelos_desp_sev(j1).d_acum=Vuelos_desp_sev(j1).d_acum-...
42         Vuelos_desp_sev(j1).d_acum(1);
43 end
44 for j2=1:m2,
45     Vuelos_ate_sev(j2).d_acum=Vuelos_ate_sev(j2).d_acum-...
46         Vuelos_ate_sev(j2).d_acum(end);
47 end
48 for j3=1:m3,
49     Vuelos_desp_sev_filtro(j3).d_acum=Vuelos_desp_sev_filtro(j3).d_acum-...
50         Vuelos_desp_sev_filtro(j3).d_acum(1);
51 end
52 for j4=1:m4,
53     Vuelos_ate_sev_filtro(j4).d_acum=Vuelos_ate_sev_filtro(j4).d_acum-...
54         Vuelos_ate_sev_filtro(j4).d_acum(end);
55 end
56
57
58 m1=length(Vuelos_desp_mal);
59 m2=length(Vuelos_ate_mal);
60 m3=length(Vuelos_desp_mal_filtro);
61 m4=length(Vuelos_ate_mal_filtro);
62
63 for j1=1:m1,
64     Vuelos_desp_mal(j1).d_acum=Vuelos_desp_mal(j1).d_acum-...
65         Vuelos_desp_mal(j1).d_acum(1);
66 end
67 for j2=1:m2,
68     Vuelos_ate_mal(j2).d_acum=Vuelos_ate_mal(j2).d_acum-...
69         Vuelos_ate_mal(j2).d_acum(end);
70 end
71 for j3=1:m3,
72     Vuelos_desp_mal_filtro(j3).d_acum=Vuelos_desp_mal_filtro(j3).d_acum-...
73         Vuelos_desp_mal_filtro(j3).d_acum(1);
74 end
75 for j4=1:m4,
76     Vuelos_ate_mal_filtro(j4).d_acum=Vuelos_ate_mal_filtro(j4).d_acum-...
77         Vuelos_ate_mal_filtro(j4).d_acum(end);
78 end
79
80
81 m1=length(Vuelos_desp_jer);
82 m2=length(Vuelos_ate_jer);
83 m3=length(Vuelos_desp_jer_filtro);
84 m4=length(Vuelos_ate_jer_filtro);

```

```

85
86 for j1=1:m1,
87     Vuelos_desp_jer(j1).d_acum=Vuelos_desp_jer(j1).d_acum-...
88         Vuelos_desp_jer(j1).d_acum(1);
89 end
90 for j2=1:m2,
91     Vuelos_ate_jer(j2).d_acum=Vuelos_ate_jer(j2).d_acum-...
92         Vuelos_ate_jer(j2).d_acum(end);
93 end
94 for j3=1:m3,
95     Vuelos_desp_jer_filtro(j3).d_acum=Vuelos_desp_jer_filtro(j3).d_acum-...
96         Vuelos_desp_jer_filtro(j3).d_acum(1);
97 end
98 for j4=1:m4,
99     Vuelos_ate_jer_filtro(j4).d_acum=Vuelos_ate_jer_filtro(j4).d_acum-...
100         Vuelos_ate_jer_filtro(j4).d_acum(end);
101 end
102
103
104 clear m1 m2 m3 m4 j1 j2 j3 j4;
105
106 Estadistica.Tiempos(9)=toc;
107
108 disp(sprintf(['Fin de los cálculos para la representación de trayectorias.'...
109     'Tiempo: %f s.'],Estadistica.Tiempos(9)))
110
111 %% Se guardan los datos correspondientes a los despegues y aterrizajes de
112 % Sevilla.
113
114 tic
115 disp('Inicio del guardado de los resultados.')
116
117 Nombre_desp_sev=['Resultados/Despegues_sev/Vuelos_desp_sev' ...
118     nombre_fuente(8:end)];
119 Nombre_ate_sev=['Resultados/Aterrizajes_sev/Vuelos_ate_sev' ...
120     nombre_fuente(8:end)];
121 Nombre_desp_sev_filtro=['Resultados/Despegues_sev_filtro/Vuelos_desp' ...
122     'sev_filtro' nombre_fuente(8:end)];
123 Nombre_ate_sev_filtro=['Resultados/Aterrizajes_sev_filtro/Vuelos_ate' ...
124     'sev_filtro' nombre_fuente(8:end)];
125 Nombre_desp_mal=['Resultados/Despegues_mal/Vuelos_desp_mal' ...
126     nombre_fuente(8:end)];
127 Nombre_ate_mal=['Resultados/Aterrizajes_mal/Vuelos_ate_mal' ...
128     nombre_fuente(8:end)];
129 Nombre_desp_mal_filtro=['Resultados/Despegues_mal_filtro/Vuelos_desp' ...
130     '_mal_filtro' nombre_fuente(8:end)];
131 Nombre_ate_mal_filtro=['Resultados/Aterrizajes_mal_filtro/Vuelos_ate' ...
132     '_mal_filtro' nombre_fuente(8:end)];
133 Nombre_desp_jer=['Resultados/Despegues_jer/Vuelos_desp_jer' ...
134     nombre_fuente(8:end)];
135 Nombre_ate_jer=['Resultados/Aterrizajes_jer/Vuelos_ate_jer' ...
136     nombre_fuente(8:end)];
137 Nombre_desp_jer_filtro=['Resultados/Despegues_jer_filtro/Vuelos_desp_jer' ...
138     '_filtro' nombre_fuente(8:end)];
139 Nombre_ate_jer_filtro=['Resultados/Aterrizajes_jer_filtro/Vuelos_ate_jer' ...
140     '_filtro' nombre_fuente(8:end)];
141 save(Nombre_desp_sev,'Vuelos_desp_sev');
142 save(Nombre_ate_sev,'Vuelos_ate_sev');

```

```

143 save(Nombre_desp_sev_filtro, 'Vuelos_desp_sev_filtro');
144 save(Nombre_ate_sev_filtro, 'Vuelos_ate_sev_filtro');
145 save(Nombre_desp_mal, 'Vuelos_desp_mal');
146 save(Nombre_ate_mal, 'Vuelos_ate_mal');
147 save(Nombre_desp_mal_filtro, 'Vuelos_desp_mal_filtro');
148 save(Nombre_ate_mal_filtro, 'Vuelos_ate_mal_filtro');
149 save(Nombre_desp_jer, 'Vuelos_desp_jer');
150 save(Nombre_ate_jer, 'Vuelos_ate_jer');
151 save(Nombre_desp_jer_filtro, 'Vuelos_desp_jer_filtro');
152 save(Nombre_ate_jer_filtro, 'Vuelos_ate_jer_filtro');
153 clear Vuelos_desp_sev Vuelos_ate_sev Vuelos Nombre_desp Nombre_ate Nombre_tot;
154 Estadistica.Tiempos(10)=toc;
155 ttot=sum(Estadistica.Tiempos(:));
156 disp('Fin del guardado de los resultados.')
```

```

157 disp(sprintf([nombre_fuente '.bst procesado. Tiempo total: %a s.'], ttot))
158 Nombre_est=[ 'Resultados/Estadisticas/Estadistica_' nombre_fuente(8:end) ];
159 save(Nombre_est, 'Estadistica');
```

```

160 clear Estadistica Nombre_est ttot Vuelos_desp_sev Vuelos_ate_sev nombre_fuente;
161 clear Mensaje k09 k27 Vuelos_desp_sev_filtro;
162 clear Vuelos_ate_sev_filtro Nombre_ate_filtro Nombre_desp_filtro;
```

B.14. func_repr.m

```

1 function [Vuelos,k]=func_repr(Vuelos)
2
3 % Cálculo de la distancia recorrida por cada vuelo medida en Km, así como las
4 % coordenadas de correspondientes a la proyección cónica de Lambert.
5
6 m=length(Vuelos); % Número de vuelos.
7 k=0;
8 % Cálculo del radio de la Tierra en Sevilla según GSW84:
9 % Coordenadas esféricas de Sevilla:
10 fisvq=37.412144*pi/180; lambdasvq=-6.002517*pi/180;
11 f=1/298.257224; % Aplanamiento.
12 R=6378.137*1e3; % Radio ecuatorial de la Tierra.
13 % Coordenadas x, y y z de Sevilla:
14 x0=R/sqrt(1-f*(2-f)*sin(fisvq)^2)*cos(fisvq)*cos(lambdasvq);
15 y0=R/sqrt(1-f*(2-f)*sin(fisvq)^2)*cos(fisvq)*sin(lambdasvq);
16 z0=R/sqrt(1-f*(2-f)*sin(fisvq)^2)*sin(fisvq);
17 Rsvq=sqrt(x0^2+y0^2+z0^2); % Radio de la Tierra en Sevilla.
18 e=sqrt(2*f-f^2); % Excentricidad
19
20 for j1=m:-1:1,
21     n1=length(Vuelos(j1).Latitud);
22     if n1<2,
23         Vuelos(j1)=[];
24         k=k+1;
25         continue;
26     end
27     PT1=[Vuelos(j1).Latitud(1:end-1) Vuelos(j1).Longitud(1:end-1)];
28     PT2=[Vuelos(j1).Latitud(2:end) Vuelos(j1).Longitud(2:end)];
29     d_parc=distance('rh',PT1,PT2,[R/1e3 e]);
30     n2=length(d_parc);
31     for j2=1:n2,
```

```
32      Vuelos(j1).d_acum(j2)=(sum(d_parc(1:j2)))';
33 %      Vuelos(j1).alt_parc(j2)=(Vuelos(j1).Altitud(j2+1))';
34 end
35 [Vuelos(j1).xlambert,Vuelos(j1).ylambert]=...
36     func_lambert(Vuelos(j1).Latitud,Vuelos(j1).Longitud);
37 clear d_parc;
38 end
```

Apéndice C

Formato de las estructuras de datos

A la hora de agrupar y mantener ordenados los datos y resultados correspondientes a cada aeronave, se ha decidido usar *estructuras* en MATLAB. Estas estructuras tienen todas la misma organización, siendo cada componente de la estructura principal correspondiente a cada vuelo detectado.

El nombre de la estructura principal es *Vuelos* y la organización interna que contiene es como sigue:

- **Vuelos**

- ⇒ **Matricula:** Contiene el código hexadecimal correspondiente a la aeronave en cuestión.
- ⇒ **Tiempo:** Se trata de un vector que contiene el tiempo transcurrido desde el inicio del vuelo.
- ⇒ **Altitud:** Se trata de un vector que contiene los valores de la altitud (en ft) en cada momento.
- ⇒ **Latitud:** Se trata de un vector que contiene los valores de la latitud (en grados) en cada momento.
- ⇒ **Longitud:** Es un vector similar al de la latitud.
- ⇒ **ROC:** Se trata de un vector que contiene los valores del Rate Of Climb (en ft/min).
- ⇒ **IsOnGround:** Se trata de un vector que contiene los valores del parámetro IsOnGround.
- ⇒ **Modelo:** Contiene el modelo de la aeronave tal y como se obtuvo a partir de su código hexadecimal de una base de datos de internet.
- ⇒ **Modelo2:** Contiene el modelo de la aeronave corregido con los posibles errores que pudiera presentar la base de datos consultada, además de recortado lo justo para presentar la familia a la que pertenece el modelo.
- ⇒ **Aerolinea:** Contiene el nombre de la aerolínea propietaria de la aeronave tal y como se obtuvo a partir de su código hexadecimal.

-
- ➡ **Aerolinea2:** Contiene el nombre corregido de la aerolínea correspondiente, debido a posibles errores que pudieran aparecer en la base de datos y omitiendo los posibles “livery” que pudiera presentar.
 - ➡ **Inicio:** Se trata de una cadena de caracteres que contiene la fecha y la hora de inicio del vuelo.
 - ➡ **Callsign:** Se trata de una cadena de caracteres que contiene el CallSign correspondiente al vuelo del que se trate.
 - ➡ **Estado:** Se trata de una cadena de caracteres que indica si el avión se encuentra despegando, aterrizando o en crucero.
 - ➡ **d_acum:** Se trata de un vector cuyas componentes indican la distancia que ha recorrido la aeronave hasta dicha componente. Contiene una componente menos que otros vectores como por ejemplo *Tiempo*, *Altitud* o *Latitud* entre otros.
 - ➡ **xlambert:** Se trata de un vector que contiene las coordenadas x (en metros) de la proyección de Lambert realizada.
 - ➡ **ylambert:** Similar al vector *xlambert*.
 - ➡ **Cabecera:** Indica la cabecera por la que despegó/aterrizó la aeronave.
 - ➡ **Destino:** Aparece únicamente en aquellos despegues o aterrizajes cuyo Callsign haya sido encontrado en las bases de datos consultadas.
 - ➡ **Estadística:** Se trata de una subestructura que contiene datos relativos a datos generales del vuelo, así como a la calidad de la señal.
 - **Media:** Es la media (en segundos) de la diferencia de tiempos de la señal del vuelo.
 - **Des_Tip:** Es la desviación típica (en segundos) de la diferencia de tiempos de la señal del vuelo.
 - **Max:** Es el máximo (en segundos) de la diferencia de tiempos.
 - **Min:** Es el mínimo (en segundos) de la diferencia de tiempos.
 - **Intervalo:** Se trata de un valor (en segundos) que indica el tiempo durante el que se ha estado recibiendo la señal de dicho vuelo.
 - **Inicio:** Se trata de la componente en la que se considera que empieza el despegue/aterrizaje según el caso.
 - **Fin:** Equivalente a *Inicio*.
 - **Altitud_max:** Altitud máxima alcanzada por la aeronave.
 - **Altitud:** Se trata de una subestructura que divide las componentes de la aeronave por rangos de altitud, en este caso cada 500 ft.
 - ◊ **Posiciones:** Se trata de una subestructura que en cada una de sus componentes contiene las componentes en las que la aeronave se encuentra entre un cierto rango de altitudes, indicados en la subestructura que se encuentra en este mismo nivel denominada *Altitud*.
 - ◊ **Cantidad:** Se trata de un vector que indica el número de componentes encontradas en cada rango de altitudes.
 - ◊ **Altitud:** Se trata de una subestructura que contiene los rangos de altitudes entre los que se han dividido los vuelos.

Apéndice D

Proyección de Lambert

La proyección de Lambert se trata de una proyección cónica y conforme en la que los paralelos son arcos concéntricos no equiespaciados, más cercanos cuanto más se acercan al centro del mapa. Los meridianos por su parte se encuentran equiespaciados, siendo los radios del mismo círculo [6].

Deben considerarse dos paralelos estándar, denominados automecoicos, los cuales se seleccionarán de tal manera que abarquen dos tercios del rango de latitudes que se pretenda representar, como se recomienda en el capítulo 8 de [7].

En la Figura D.1 se muestra un esquema de la proyección de Lambert, en la que aparecen también los dos paralelos comentados anteriormente. En esta proyección, las líneas rectas apro-

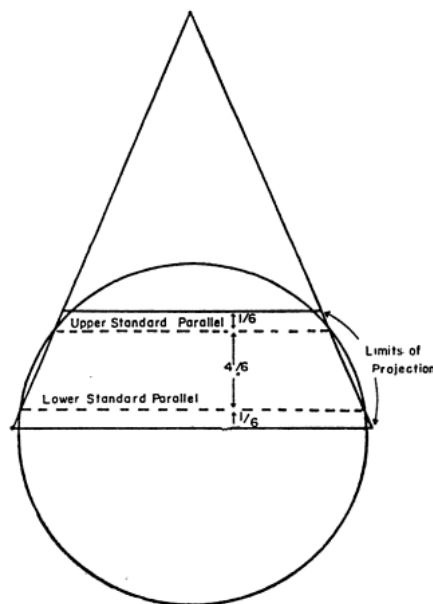


Figura D.1: Selección de los paralelos para la proyección de Lambert.

ximan rutas ortodrómicas, siendo una proyección bastante realista en la zona localizada cerca de los automecoicos, perdiendo validez para representaciones globales [8].

Las coordenadas x-y de la proyección vienen dadas por:

$$x = \rho \sin(n(\lambda - \lambda_0)); y = \rho_0 - \rho \cos(n(\lambda - \lambda_0)) \quad (D.1)$$

donde:

$$\begin{aligned} n &= \frac{\ln(\cos\phi_1 \sec\phi_2)}{\ln(\tan(\pi/4 + \phi_2/2) \cot(\pi/4 + \phi_1/2))} \\ \rho &= F \cot^n(\pi/4 + \phi/2) \\ \rho_0 &= F \cot^n(\pi/4 + \phi_0/2) \\ F &= 1/n \cos\phi_1 \tan^n(\pi/4 + \phi_1/2) \end{aligned} \quad (D.2)$$

siendo ϕ_1 y ϕ_2 las latitudes a las que se encuentran los automecoicos, ϕ_0 y λ_0 la latitud y longitud para el origen de las coordenadas.

La utilización de este tipo de representación se debe a que es la usada a la hora de realizar las cartas aeronáuticas que se seguirán a la hora de visualizar los perfiles horizontales de los vuelos capturados [6].

En el caso de este proyecto, un ejemplo de la proyección con los correspondientes automecoicos elegidos se muestra en la figura D.2.

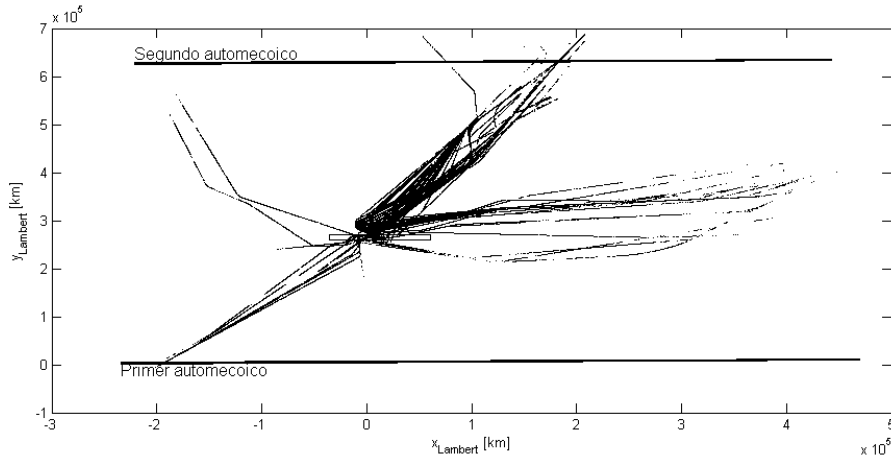


Figura D.2: Ejemplo de la proyección de Lambert y de los automecoicos elegidos.

Para la elección de los automecoicos, se ha decidido en primer lugar hallar los valores máximos y mínimos de latitud encontrados (entre todos los vuelos obtenidos), los cuales han resultado ser $\phi_{max} = 41^\circ$ y $\phi_{min} = 34.5^\circ$. De esta manera y buscando que los automecoicos abarquen 2/3 del rango total de latitudes, se obtiene que sus valores deben ser $\phi_1 = 35.58^\circ$ y $\phi_2 = 39.91^\circ$.