



**Centro Universitario de la Defensa
en la Escuela Naval Militar**

TRABAJO FIN DE GRADO

*Sistema de medición de diagramas de
radiación de antenas embarcadas
basado en dron*

Grado en Ingeniería Mecánica

ALUMNO: Julián García Domínguez

DIRECTORES: José María Núñez Ortuño

Carlos Casqueiro Placer

CURSO ACADÉMICO: 2016-2017

Universida_{de}Vigo



Centro Universitario de la Defensa en la Escuela Naval Militar

TRABAJO FIN DE GRADO

*Sistema de medición de diagramas de
radiación de antenas embarcadas
basado en dron*

Grado en Ingeniería Mecánica
Intensificación en Tecnología Naval
Cuerpo General

Universida_deVigo

RESUMEN

El presente Trabajo de Fin de Grado consiste en el desarrollo de un sistema de medida de diagramas de radiación de antenas embarcadas basado en un UAV. Este sistema está basado en un hexacóptero con un receptor SDR embarcado que realiza un vuelo circular alrededor de la antena bajo prueba tomando medidas con una determinada tasa y que son procesadas en un ordenador personal para la obtención del diagrama de radiación de la antena.

PALABRAS CLAVE

Diagrama de radiación, Dron, UAV, Antena Embarcada, Diagrama de cobertura

AGRADECIMIENTOS

A mis padres, hermanas y resto de familia por su apoyo y consejo.

A mi tutor el Doctor Ingeniero de Telecomunicación D. José María Núñez Ortuño por su trabajo, ayuda y aporte de conocimientos durante todo el proceso de elaboración de este proyecto.

A mi tutor el Doctor Ingeniero Industrial D. Carlos Casqueiro Placer por su trabajo y explicaciones en el área del aeromodelismo y, especialmente, los conocimientos aportados sobre baterías.

Al personal del CEMEDM por la ayuda facilitada en los primeros compases del trabajo.

CONTENIDO

Contenido	1
Índice de Figuras	5
Índice de Tablas.....	7
1 Introducción y objetivos	8
1.1 Contextualización.....	8
1.2 La antena	9
1.2.1 Funcionamiento de una antena	9
1.2.2 Tipos de antenas	10
1.2.3 Caracterización eléctrica de la antena.....	11
1.2.4 Caracterización de la antena como elemento radiante.....	12
1.2.5 Diagrama de radiación	13
1.3 Motivación y objetivos.....	15
1.4 Descripción de la memoria	15
2 Estado del arte	17
2.1 Introducción	17
2.2 La obtención del diagrama de radiación	17
2.2.1 Caracterización de antenas aisladas.....	17
2.2.2 Caracterización de antenas sobre el terreno.....	19
2.2.3 Caracterización de antenas por el CEMEDDEM	19
2.2.4 Técnicas para la medida de potencia	20
2.3 Los vehículos aéreos no tripulados	22
2.3.1 Concepto de UAS	22
2.3.2 Los UAS hoy en día.....	23
2.4 La radio definida por <i>software</i>	25
2.4.1 Concepto de SDR	25
2.4.2 Presente y futuro de la SDR.....	26
3 Diseño del sistema.....	28
3.1 Visión general	28
3.2 Diseño global del sistema	29
3.3 La medida de la potencia	30
3.3.1 El problema de la toma de medidas	30
3.3.2 Solución propuesta.....	33
3.3.3 Equipo de medida y software utilizado	34
3.4 Plataforma de vuelo	36

3.4.1 El problema del vuelo	36
3.4.2 Solución propuesta.....	36
3.4.3 Plataforma aérea	38
4 Desarrollo del sistema	40
4.1 Introducción	40
4.2 Diseño del <i>software</i> del sistema.....	40
4.3 La medida de la potencia	41
4.3.1 El programa de GNURadio	41
4.3.2 Arquitectura cliente-servidor	43
4.4 La plataforma de vuelo	44
4.4.1 Hilo servidor receptor de medidas	44
4.4.2 Hilo de control de vuelo	44
4.4.3 Hilo principal	45
4.5 Interpretación de los resultados.....	45
5 Pruebas y validación del sistema.....	46
5.1 Prueba de vuelo	46
5.1.1 Desarrollo de las pruebas.....	46
5.1.2 Conclusiones.....	49
5.1.3 Estudio de autonomía	49
5.2 Primera prueba de medición sin vuelo	51
5.2.1 Montaje de la prueba	51
5.2.2 Desarrollo de las pruebas.....	52
5.2.3 Conclusiones.....	54
5.3 Segunda medición sin vuelo	55
5.3.1 Desarrollo de las pruebas.....	55
5.3.2 Conclusiones.....	57
5.4 Prueba final	58
6 Conclusiones y líneas futuras	59
6.1 Conclusiones	59
6.2 Mejoras.....	60
6.3 Líneas futuras	60
7 Bibliografía.....	62
Anexo I: Siglas y acrónimos	67
Anexo II: Manual de usuario.....	69
AII.1 Planeamiento de la medida.....	69

AII.1.1 Radio de vuelo.....	69
AII.1.2 Altura de vuelo.....	70
AII.1.3 Ejemplo: Lancha de Instrucción.....	70
AII.2 Ejecución de la medida.....	71
AII.2.1 Pasos a seguir.....	71
AII.3 Instalación del sistema por primera vez.....	75
AII.3.1 Pasos a seguir en la instalación.....	75
Anexo III: Código fuente.....	77
AIII.1 Módulo medida de potencia.....	77
AIII.1.1 medida.py.....	77
AIII.1.2 radioDRON.py.....	78
AIII.2 Módulo plataforma de vuelo.....	80
AIII.2.1 vuelo.py.....	80
AIII.2.2 servidor.py.....	82
AIII.2.3 multicoptero.py.....	83
AIII.2.4 auxiliar.py.....	86

ÍNDICE DE FIGURAS

Figura 1-1 Las radiocomunicaciones en una operación de salvamento [3].....	8
Figura 1-2 Esquema de onda electromagnética emitida por una antena en un instante dado [6].....	9
Figura 1-3 Regiones del campo electromagnético [9].....	10
Figura 1-4 Antena parabólica [10]	11
Figura 1-5 El radar SPY de Lockheed Martin está formado por una agrupación de bocinas [11] ..	11
Figura 1-6 Modelo de sistema transmisor con antena como circuito eléctrico	12
Figura 1-7 Diagrama de radiación tridimensional de una antena Yagi usada en televisión [13].....	13
Figura 1-8 Diagramas de radiación polar y cartesiano [14]	13
Figura 1-9 Distintas antenas en el mástil de una fragata clase Álvaro de Bazán [16]	14
Figura 2-1 Esquema de una cámara anecoica para la obtención del diagrama [21].....	18
Figura 2-2 Cámara anecoica de la USAF en la Base Aérea Edwards (California) [22]	18
Figura 2-3 Esquema caracterización de antena sobre el terreno [23].....	19
Figura 2-4 Esquema típico de sensor térmico de potencia de RF [19].....	21
Figura 2-5 Detector de envolvente sencillo.....	21
Figura 2-6 Esquema de bloques de un analizador de espectro superheterodino de una etapa [26] ..	22
Figura 2-7 Analizador de espectro digital portátil de la casa Rohde&Schwarz [27]	22
Figura 2-8 Esquema global de funcionamiento de un UAS [30]	23
Figura 2-9 Demostración de Tesla en el Madison Square Garden [31]	23
Figura 2-10 Presupuesto dedicado a UAVs militares americanos desde 1988 a 2013 [32].....	24
Figura 2-11 Prototipo de taxi dron [37].....	24
Figura 2-12 Representación de componentes I y Q de la señal [42]	26
Figura 2-13 Esquema de bloques típico de una SDR [43]	26
Figura 2-14 Concepto nacional de empleo de la SDR en el ámbito militar [46]	27
Figura 2-15 Receptor RTL USB conectado a antena [48]	27
Figura 3-1 Visión general del sistema diseñado.....	28
Figura 3-2 Arquitectura modular del sistema diseñado.....	29
Figura 3-3 Esquema del primer elipsoide de Fresnel [49]	30
Figura 3-4 Modelo de propagación de tierra plana [12].....	31
Figura 3-5 Esquema sistema de medida de potencia.....	33
Figura 3-6 HackRF One utilizada para transmitir la señal piloto [54]	34
Figura 3-7 Receptor SDR NooElec utilizado [55]	35
Figura 3-8 Simple aplicación desarrollada con GNURadioCompanion [56].....	36
Figura 3-9 Detalle de aproximación de círculo (verde) por polígono de 36 lados (rojo) en Matlab	37
Figura 3-10 Esquema sistema de vuelo	38

Figura 3-11 Resultado final plataforma de vuelo	39
Figura 4-1 Arquitectura del <i>software</i> desarrollado.....	41
Figura 4-2 Programa en GNURadio.....	41
Figura 4-3 Bloque fuente.....	42
Figura 4-4 Primera parte de los bloques de proceso	42
Figura 4-5 Segunda parte de los bloques de proceso	43
Figura 4-6 Bloque de salida.....	43
Figura 5-1 Trayectoria primer vuelo	47
Figura 5-2 Trayectoria segundo vuelo.....	47
Figura 5-3 Trayectoria tercer vuelo.....	48
Figura 5-4 Trayectoria del cuarto vuelo hasta el momento que se manda RTL.....	48
Figura 5-5 Curva de descarga baterías LiPo fabricante Storm Batteries [57].....	50
Figura 5-6 Log de telemetría de consumo de batería (tiempo-voltaje)	51
Figura 5-7 Montaje de antena para realización de pruebas	52
Figura 5-8 Detalle de colocación de la antena junto a HackRF One.....	52
Figura 5-9 Diagrama de la primera medición 16/02/17	53
Figura 5-10 Diagrama de la segunda medición 16/02/17.....	54
Figura 5-11 Simulación del diagrama de radiación de la antena Yagi de 11 elementos utilizada...54	54
Figura 5-12 Diagrama de la primera medición 20/02/17	56
Figura 5-13 Diagrama de la segunda medición 20/02/17.....	56
Figura 5-14 Superposición de diagrama obtenido con diagrama simulado	57
Figura 5-15 Trayectoria seguida en la segunda medición 20/02/17.....	58
Figura 6-1 Firma IR de un buque obtenida de noche a 1 km de distancia por [61]	61
Figura AII-0-1 Simulación de efecto multitrayecto [62].....	71
Figura AII-0-2 Conectores XT60 de batería y módulo de potencia.....	71
Figura AII-0-3 Red inalámbrica del punto de acceso <i>wifi</i>	72
Figura AII-0-4 Conexión mediante Putty a la RPi	72
Figura AII-0-5 Botón <i>connect</i> de Mission Planner	73
Figura AII-0-6 Selección de puerto UDP.....	73
Figura AII-0-7 Interfaz del programa.....	74
Figura AII-0-8 Mensaje de servidor en escucha.....	74
Figura AII-0-9 Asignación IP a ordenador portátil. Figura 1.....	76
Figura AII-0-10 Asignación IP a ordenador portátil. Figura 2.....	76

ÍNDICE DE TABLAS

Tabla 5-1 Datos de entorno de la prueba de vuelo	46
Tabla 5-2 Primer vuelo	46
Tabla 5-3 Segundo vuelo	47
Tabla 5-4 Tercer vuelo	47
Tabla 5-5 Cuarto vuelo	48
Tabla 5-6 Quinto vuelo	49
Tabla 5-7 Datos de autonomía de diseño del UAV [17]	50
Tabla 5-8 Datos del entorno en la primera prueba de medición sin vuelo	51
Tabla 5-9 Primera medición 16/02/17	53
Tabla 5-10 Segunda medición 16/02/17	53
Tabla 5-11 Datos del entorno en la segunda prueba de medición sin vuelo	55
Tabla 5-12 Primera medición 20/02/17	55
Tabla 5-13 Segunda medición 20/02/17	56

1 INTRODUCCIÓN Y OBJETIVOS

1.1 Contextualización

El hecho de que en 2015 hubiera más de 7.100 millones de líneas de teléfono móvil contratadas a nivel mundial [1]; que el 47% de la población mundial esté conectada a internet [2], muchas de estas conexiones realizadas desde puntos de acceso inalámbricos; o la mera existencia de satélites destinados única y exclusivamente a la transmisión de mensajes entre unidades militares da una idea de la importancia que las radiocomunicaciones tienen en la sociedad actual.

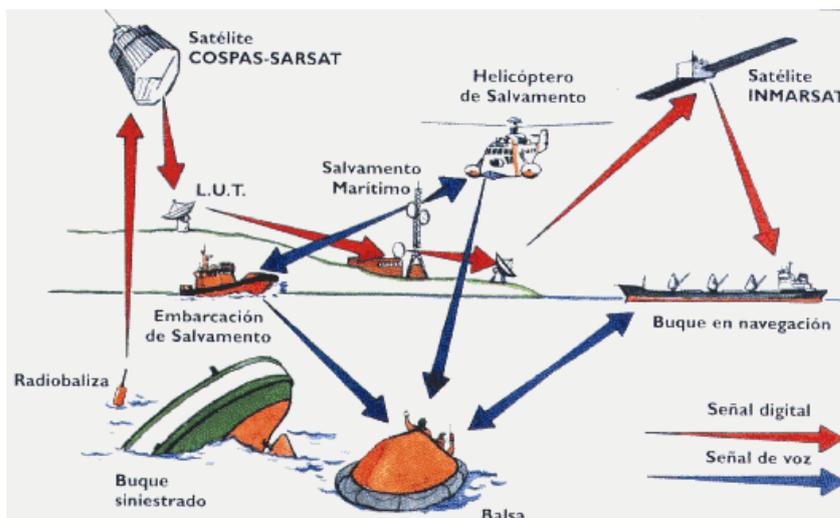


Figura 1-1 Las radiocomunicaciones en una operación de salvamento [3]

La Unión Internacional de Telecomunicaciones (UIT) define en [4] la radiocomunicación como: «Toda telecomunicación transmitida por ondas radioeléctricas». Entendiendo por ondas radioeléctricas: «Aquellas ondas electromagnéticas cuya frecuencia se fija convencionalmente por debajo de 3.000 GHz y que se propagan por el espacio sin guía artificial».

No obstante, las ondas radioeléctricas no solo tienen aplicación en las telecomunicaciones, también son la base de los radares, los hornos microondas, sistemas de radionavegación y sistemas de posicionamiento por satélite, además de otras utilidades. Por variada que sea su aplicación, todos los dispositivos que hacen uso de las ondas radioeléctricas tienen un elemento en común: la antena.

1.2 La antena

La antena es el «elemento del sistema transmisor o receptor diseñado para radiar o recibir ondas electromagnéticas» [5]. Por lo tanto, la antena es la interfaz entre el dispositivo y el medio por el que se propaga la energía en forma de ondas.

1.2.1 Funcionamiento de una antena

El principio de funcionamiento de una antena se basa en los principios del electromagnetismo sintetizados por el físico escocés James Clerk Maxwell en sus cuatro famosas ecuaciones.

El caso más sencillo de una antena, cuyo funcionamiento es fácilmente entendible, es un simple hilo radiante, por ejemplo, un filamento de cobre.

Cuando este filamento es excitado por una corriente variable en el tiempo, se genera un campo magnético variable el cual a su vez induce un campo eléctrico también variable, la acción recíproca de estos dos campos de componentes perpendiculares entre sí genera una onda electromagnética de frecuencia igual a la de la corriente variable a la que fue sometida y que se propaga por el espacio en todas direcciones cumpliendo las mismas propiedades que las ondas mecánicas como son la reflexión, refracción, difracción, dispersión y absorción. Este hilo de cobre se convierte así en una antena transmisora.

Cuando las ondas electromagnéticas lleguen a la antena receptora, otro hilo de cobre, la acción del campo magnético variable inducirá una corriente variable con la misma frecuencia de la onda.

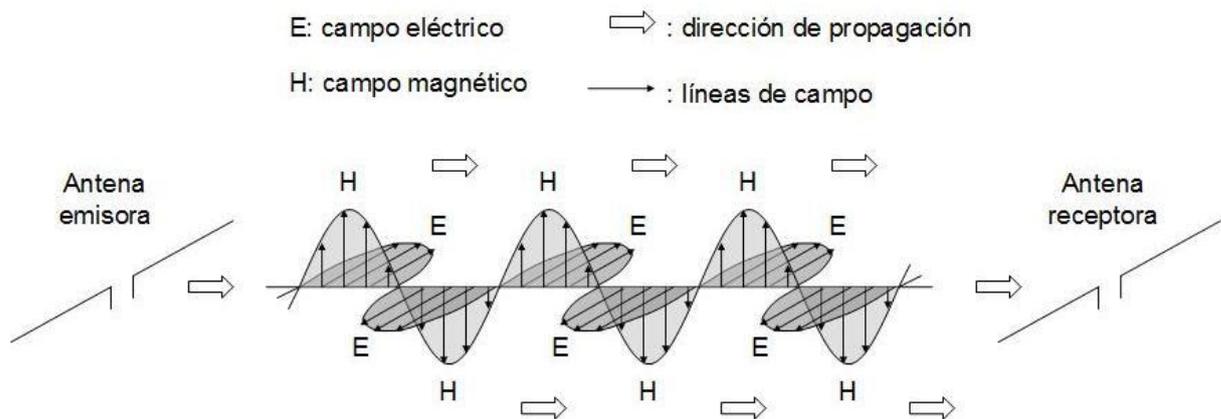


Figura 1-2 Esquema de onda electromagnética emitida por una antena en un instante dado [6]

1.2.1.1 Regiones del campo radiado

Las características del campo electromagnético radiado varían con la distancia a la fuente [7]. A pesar de que estos cambios son graduales, convencionalmente se distinguen tres regiones del campo radiado: la de campo cercano reactivo, la de campo cercano radiante y la de campo lejano radiante.

La **región de campo cercano reactivo** existe en el área más próxima a la antena. Como su propio nombre indica, se caracteriza porque el campo electromagnético es fundamentalmente reactivo y es provocado por las cargas en la propia antena. Según el tipo de antena, este campo será inductivo o capacitivo, traduciéndose en un predominio del campo eléctrico o magnético, lo cual es utilizado en la tecnología *Near Field Communication* (NFC) utilizada para pagos con tarjetas inalámbricas [8]. Este campo no es radiado y su energía decae rápidamente, considerándose despreciable por convención a una distancia desde la fuente de:

$$r = \frac{\lambda}{2\pi}$$

La **región de campo cercano radiante**, también denominada región de Fresnel por analogía con la terminología óptica, es la zona más próxima a la antena en la que el campo es radiado. Sin embargo, el campo en esta zona de transición depende del ángulo de observación y la distancia a la que se mida, produciéndose un desfase y una diferencia de amplitud relativa entre el campo eléctrico y el magnético. Esto es debido fundamentalmente a la geometría de la antena, lo que explica que esta región no se encuentre presente en todos los tipos de antenas.

A partir de cierta distancia, la diferencia de amplitud relativa y el desfase entre el campo electromagnético es prácticamente inexistente, lo que marca el comienzo de la región de campo lejano radiante. Aunque existen diversos criterios para la selección de esta distancia según la aplicación, una buena aproximación para antenas cuya dimensión principal D es grande comparada con la longitud de onda λ es:

$$r = \frac{2D^2}{\lambda}$$

En la **región de campo lejano radiante**, o región de Fraunhofer, el frente de onda se puede considerar plano. La mayoría de las aplicaciones de las ondas radioeléctricas trabajan en esta región, además de ser la más sencilla de medir.

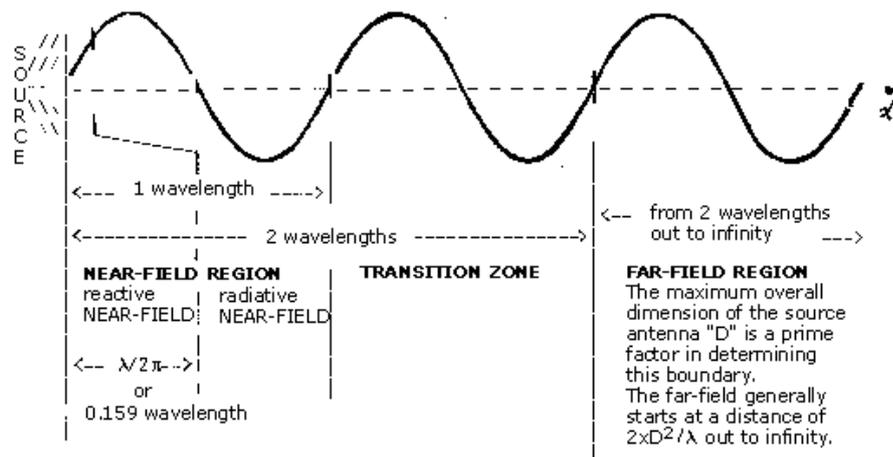


Figura 1-3 Regiones del campo electromagnético [9]

1.2.2 Tipos de antenas

Existen en la literatura técnica múltiples criterios de clasificación distintos para las antenas.

En función de cómo distribuyen espacialmente la energía radiada, las antenas pueden distinguirse entre directivas y omnidireccionales. Es fácil clasificar una antena en base a este criterio cuando se conoce el diagrama de radiación de la misma, tal y como será explicado en el apartado 1.2.5.

Otro criterio de clasificación es su modo de radiación, el cual a su vez está relacionado con la frecuencia en la que trabajan. Según [7], atendiendo a este criterio, se distinguen cuatro tipos de antenas: los elementos de corriente, las de onda progresiva, las de apertura y las agrupaciones o *arrays* de antenas.

Los **elementos de corriente** son hilos conductores que funcionan como se ha explicado en el apartado anterior. Dentro de este tipo de antenas se encuentran los monopolos y los dipolos.

En las **antenas de onda progresiva**, la onda es guiada a través de una estructura formada por hilo conductor. Ejemplos de estas son las antenas log periódicas y las helicoidales.

En las **antenas de apertura**, se considera que el campo electromagnético radiado emana de una apertura o fisura física. Para aumentar su directividad y conseguir la forma de onda necesaria, se emplean distintos tipos de aperturas o incluso reflectores. El ejemplo más sencillo es una guía de onda abierta por su extremo, pero también se encuentran dentro de este grupo las bocinas o las antenas parabólicas.



Figura 1-4 Antena parabólica [10]

Las **agrupaciones o arrays de antenas** consisten en un conjunto de antenas radiando individualmente que, con unas diferencias de fase y amplitud determinadas, permiten obtener la forma de onda deseada. Su gran ventaja es la flexibilidad que aportan ya que mediante las técnicas de control adecuado, la misma distribución física de antenas permite obtener muchísimas formas de onda diferentes.

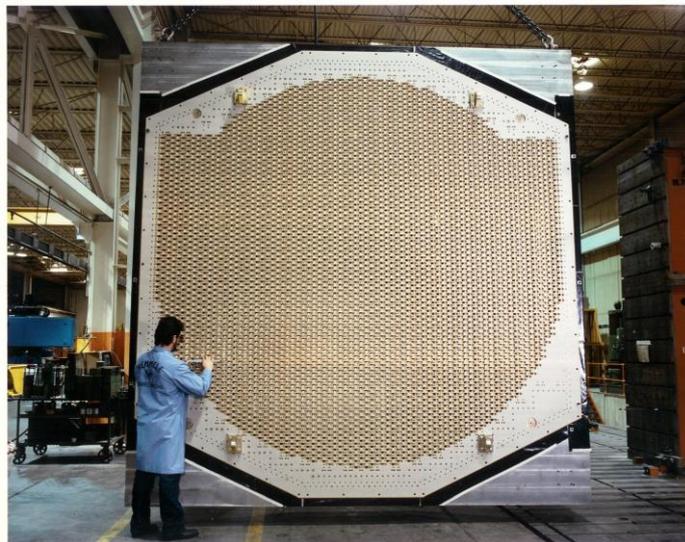


Figura 1-5 El radar SPY de Lockheed Martin está formado por una agrupación de bocinas [11]

1.2.3 Caracterización eléctrica de la antena

Como se aprecia en el circuito de la Figura 1-6, un dispositivo transmisor puede modelarse como una fuente de tensión variable en el tiempo V_g con una impedancia Z_g .

La antena, como dispositivo que entrega potencia, se caracteriza en este mismo circuito como una impedancia:

$$Z_a = R_a + jX_a$$

La parte reactiva de la impedancia X_a representa el comportamiento inductivo o capacitivo de la antena. Se dice que una antena es resonante para una frecuencia determinada cuando su reactancia es nula al transmitir ondas de esa misma frecuencia. Por lo tanto, la impedancia de una antena depende de la frecuencia de trabajo.

Por su parte, la resistividad de la antena, no está afectada por la frecuencia de trabajo, y, como se explica en [12], la parte resistiva de la impedancia es, a su vez, la suma de dos componentes:

$$R_a = R_p + R_r$$

Siendo R_p la resistencia de pérdidas y R_r una resistencia ficticia, denominada resistencia de radiación.

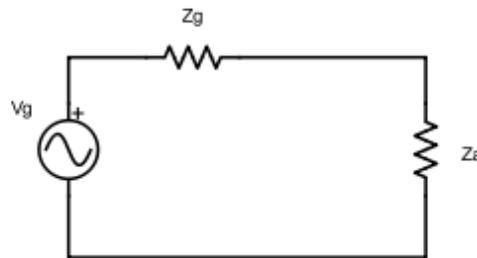


Figura 1-6 Modelo de sistema transmisor con antena como circuito eléctrico

Definiendo la eficiencia como la relación entre la potencia radiada por la antena y la potencia entregada a la misma, se tiene entonces:

$$\eta = \frac{P_r}{P_{in}} = \frac{R_r}{R_r + R_p}$$

1.2.4 Caracterización de la antena como elemento radiante

La caracterización de la antena como elemento radiante supone el conocimiento de una serie de parámetros que determinan su comportamiento y que están definidos en la norma [5], algunos de los más importantes son la directividad, la ganancia, la polarización y el ancho de banda.

Exceptuando el caso de la definida idealmente como antena isotrópica, la energía radiada por las antenas no es uniforme en todas direcciones. Para cuantificar este hecho, se define la **intensidad de radiación** como la potencia radiada p en una determinada dirección por unidad de ángulo sólido Ω .

$$i(\theta, \varphi) = \frac{dp}{d\Omega}$$

La **directividad** de la antena se define como la relación entre la intensidad de radiación en una determinada dirección y la intensidad de radiación media en todas las direcciones.

La **ganancia**, por su parte, se define como la relación entre la intensidad de radiación en una determinada dirección y la potencia total entregada a la antena si esta fuera radiada isotrópicamente. La ganancia, por lo tanto tiene en cuenta las pérdidas de la antena: siempre es igual o menor a la directividad y está relacionada con ella por la eficiencia de la antena:

$$\eta = \frac{G}{D}$$

La **polarización** de la antena es igual a la polarización de la onda que esta radia. La polarización de una onda describe la orientación del vector del campo eléctrico de una onda en función del tiempo. Existen tres tipos de polarizaciones, lineal, circular y elíptica. El conocimiento de este parámetro es

crítico, ya que un desacoplo en las polarizaciones de la antena transmisora y receptora produce una disminución en la energía recibida, llegándose incluso a producir una pérdida total en algunos casos como el de dos antenas con polarización lineal perpendicular entre ellas.

Por último, el **ancho de banda** de una antena se define como el rango de frecuencias en el cual la antena se comporta conforme a sus especificaciones de diseño. Como normal general, el tamaño de la antena está relacionado con la banda de frecuencias de trabajo, siendo mayor este cuanto menores son estas.

1.2.4.1 Principio de reciprocidad

El principio de reciprocidad establece que una antena se comporta igual en lo relativo a sus parámetros (ganancia, polarización, impedancia, etc.) tanto cuando trabaja en transmisión como cuando lo hace en recepción.

1.2.5 Diagrama de radiación

Para facilitar la comprensión del comportamiento de la antena, se utiliza el diagrama de radiación. Este diagrama no es más que una representación gráfica de la distribución espacial de la ganancia de la antena. Ya que la antena emite potencia en todas direcciones (Figura 1-7), se hacen cortes en planos representativos, generalmente el plano azimutal, y el plano vertical cortando al lóbulo principal.

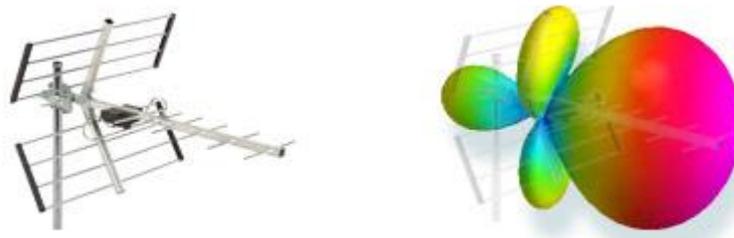


Figura 1-7 Diagrama de radiación tridimensional de una antena Yagi usada en televisión [13]

Existen multitud de tipos de representación, pero los más usados son el polar y el cartesiano (Figura 1-8), que expresan los valores de ganancia relativos al máximo de la antena, el cual se representa con 0 dB, respecto a la dirección en el plano de corte.

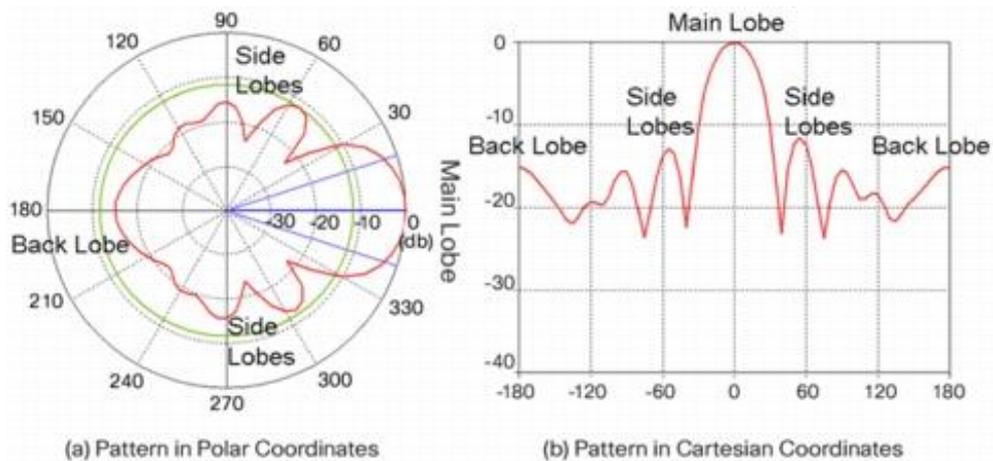


Figura 1-8 Diagramas de radiación polar y cartesiano [14]

La interpretación del diagrama de radiación de una antena aporta información muy valiosa.

El **ancho de haz** es el intervalo angular que ocupa la zona de máxima radiación. Este se puede definir como el ancho de haz de media potencia, que cubriría hasta las direcciones en las que la

potencia máxima se reduce a la mitad (-3 dB), o el ancho de haz del primer nulo, que ocuparía todo el lóbulo principal hasta el primer nulo.

El **nivel del lóbulo secundario** o relación de directividad se define como la relación entre la máxima potencia radiada en el lóbulo principal y la máxima potencia radiada en el mayor de los lóbulos secundarios.

La **relación delante atrás** es la relación entre la máxima potencia radiada en el lóbulo principal y la potencia radiada en la dirección diametralmente opuesta.

Todos estos parámetros son fundamentales para conocer, por ejemplo, la discriminación angular en una antena de radar, determinar si una antena de radiodifusión colocada en una situación geográfica determinada sobrepasará los límites de su zona de servicio asignada, o incluso determinar la orientación ideal de una antena receptora de televisión.

En vista de su utilidad, es sencillo inferir que para la Armada Española también es conveniente conocer los diagramas de radiación de sus antenas, tanto las embarcadas como las que se encuentran en tierra en estaciones radio. Si bien es cierto que los diagramas de radiación de las antenas utilizadas en los buques son conocidos de manera aislada, las antenas no se comportan exactamente igual cuando están embarcadas: la superestructura y otros elementos del mástil generan diferencias en el patrón de radiación esperado, cuya representación gráfica es lo que se conoce como diagrama de cobertura [15]. La obtención de estos diagramas, permite conocer, por ejemplo, cuál es el aspecto que debe presentar el buque para optimizar sus comunicaciones con tierra, o con unidades alejadas, o podría incluso servir para determinar una mejor disposición de las antenas en el mástil, al detectar anomalías o puntos ciegos no esperados.

El Centro de Medidas Electromagnéticas de la Armada Española (CEMEDEM) es el encargado, entre otras muchas funciones, de obtener los diagramas de cobertura de las distintas antenas de comunicaciones, guerra electrónica, direcciones de tiro y radáricas, embarcadas. El método utilizado actualmente consiste en hacer al buque navegar describiendo un círculo de radio lo más reducido posible en una posición en la mar a la vez que transmite con la antena en cuestión, de tal manera que presente todos sus aspectos a una antena receptora en tierra que, conectada al sistema de medida, se utiliza para obtener el diagrama en cuestión.



Figura 1-9 Distintas antenas en el mástil de una fragata clase Álvaro de Bazán [16]

1.3 Motivación y objetivos

Existen dos motivaciones fundamentales para la realización de este proyecto: en primer lugar, el dar una solución alternativa, y que pueda resultar más ventajosa, a la utilizada por el CEMEDEM actualmente para caracterizar las antenas embarcadas; y, en segundo lugar, continuar profundizando en la línea de trabajo abierta por el Centro Universitario de la Defensa (CUD) con el Trabajo de Fin de Grado del A.N. D. José María Liaño Cuquerella durante el curso 2015-16: los vehículos aéreos no tripulados (UAV) [17]; a la vez que se exploran algunas de las posibilidades de la radio definida por *software* (SDR), una tecnología muy utilizada actualmente y con un futuro muy prometedor, como ya introdujo el Tte. D. Diego Isaac Hernández con su TFG durante el curso 2014-15 [18].

En base a estas dos motivaciones, se establecen los objetivos del presente proyecto.

El principal objetivo, y razón de ser del proyecto, es el de crear un sistema que presente una alternativa útil y fiable al actual método de obtención de diagramas de cobertura del CEMEDEM. El sistema propuesto consiste en un UAV, controlado desde un ordenador personal, que describa un vuelo circular alrededor de la antena bajo prueba (ABP) y que, con una antena conectada a un receptor SDR, tome medidas de la potencia recibida de tal manera que a partir de las cuales se pueda obtener el diagrama de cobertura de la antena.

Un sistema de estas características presentaría múltiples ventajas:

1. Reduciría considerablemente la duración de las campañas de medida y el personal involucrado en ellas.
2. Permitiría la obtención de diagramas fuera de los polígonos de medida utilizados habitualmente por el CEMEDEM (Ría de Pontevedra y Torregorda –Cádiz–) pudiéndose realizar medidas de «oportunidad» e incluso en puerto.
3. Es totalmente portátil y de pequeño tamaño, lo que permitiría su envío como herramienta de diagnóstico ante un eventual fallo de un sistema de comunicaciones a un buque de la Flota desplegado en una misión.

El sistema desarrollado se considerará útil si permite su empleo de manera intuitiva sin la necesidad de conocer sus detalles de implementación. De la misma manera, se considerará fiable si el sistema obtiene unos resultados dentro de unos márgenes de incertidumbre aceptables de 2,54 dB al igual que el CEMEDEM (apartado 2.2.3), siendo comprobado esto último con una antena cuyo diagrama sea bien conocido.

En la línea de este objetivo principal, los objetivos específicos que se espera obtener son:

1. La creación de una interfaz en línea de comandos que permita al usuario una sencilla interacción con el sistema modificando parámetros como radio del vuelo descrito, altura o posición geográfica de la antena, entre otros.
2. La obtención del diagrama de radiación de una antena en tierra para su comparación con el diagrama ya conocido de la misma.
3. Como colofón, la obtención del diagrama de radiación de una antena embarcada en una lancha de instrucción con las que cuenta la Escuela Naval Militar (ENM).

1.4 Descripción de la memoria

En esta introducción se han expuesto los fundamentos de la teoría de antenas sobre los que gira este proyecto, así como una definición clara de los objetivos perseguidos por el mismo. A continuación se describirán uno por uno los contenidos de los capítulos subsiguientes con el objetivo de dar al lector una clara visión global de la memoria del trabajo realizado.

En el **segundo capítulo**, se presenta el **estado del arte**. En primer lugar se hará un resumen de las tecnologías actualmente utilizadas con el mismo propósito que este proyecto, poniendo especial énfasis en trabajos de carácter similar al presente, y en las técnicas aplicadas actualmente por el

CEMEDEM. A continuación, se expondrán las dos tecnologías que tienen aplicación en este proyecto: los UAV y la SDR. Proporcionando en ambos casos una breve explicación conceptual, junto con su implantación actual y sus perspectivas de futuro.

En el **tercer capítulo**, se presenta el **diseño teórico del sistema** realizado. En este capítulo, se hará una exposición de cada uno de los dos módulos que componen el sistema desarrollado. Para cada uno de los módulos, se presentará en primer lugar la problemática relacionada, para continuar explicando la solución propuesta, y finalizar con una descripción de los elementos seleccionados para formar parte del sistema desarrollado.

En el **cuarto capítulo**, se describe el **desarrollo del proyecto**. Al igual que el capítulo anterior, se dedicará un apartado a cada uno de los módulos exponiendo cómo fue el proceso de desarrollo, explicando la arquitectura del *software* desarrollado, así como los problemas surgidos y la solución proporcionada.

En el **quinto capítulo**, se hace un resumen de las **pruebas realizadas** al sistema. Para cada una de las pruebas se expondrán las condiciones en las que fueron realizadas, los resultados obtenidos y las conclusiones extraídas.

En el **sexto capítulo de conclusiones y líneas futuras**, en primer lugar, se tratará de hacer un análisis objetivo del sistema obtenido estudiando si se ha conseguido o no la consecución de los objetivos propuestos; en segundo lugar, se definirán las mejoras, o cambios que deberían hacerse en este proyecto, así como líneas futuras o nuevas vías de investigación que se abren.

Por último, hay **tres anexos**. El primero es un compendio de las **siglas y acrónimos** utilizados a lo largo de la memoria. El segundo anexo, es un **breve manual** o guía de uso del sistema creado. El tercer anexo contiene el **código desarrollado** para este proyecto.

2 ESTADO DEL ARTE

2.1 Introducción

La obtención de diagramas de radiación no es algo novedoso, sino que existen multitud de técnicas y herramientas profesionales para su ejecución [19]. De la misma manera, existen multitud de universidades y centros de investigación que han experimentado previamente con el uso de drones con fines muy similares cuando no idénticos a los del presente proyecto.

En este capítulo se pretende describir al lector las técnicas utilizadas en la actualidad para caracterizar antenas, así como las usadas por el CEMEDEM para hacer lo propio con las antenas embarcadas en los buques de la Armada. A continuación, se procederá a explicar y detallar las posibilidades de las dos tecnologías utilizadas para la consecución de este proyecto, los vehículos aéreos no tripulados y la radio definida por *software*.

2.2 La obtención del diagrama de radiación

Para la obtención del diagrama de radiación de una antena aislada, o del diagrama de cobertura cuando se encuentre en el lugar en el que va a desempeñar su función, basta con tomar medidas de potencia transmitida o recibida (principio de reciprocidad) por la antena bajo prueba en todas las direcciones dentro del plano para el cual se está obteniendo el diagrama (azimutal, vertical, etc.).

Generalmente, debido a que es la condición en la que la mayoría de antenas serán empleadas, las mediciones se realizan en la región de campo lejano radiante. Por lo tanto las técnicas descritas en este apartado se refieren a la obtención de diagramas de radiación y cobertura de dicha región.

2.2.1 Caracterización de antenas aisladas

A efectos de hacer un estudio de su comportamiento real, obtener el diagrama de radiación de la antena aislada no es tan útil como hacerlo sobre el terreno en el que va a actuar dicha antena. Sin embargo, sí es una necesidad de los ingenieros que diseñan antenas pues les permite trabajar eliminando el efecto de factores externos.

Existen principalmente dos técnicas utilizadas por estos expertos: la simulación mediante *software* y la caracterización en cámaras anecoicas.

2.2.1.1 Simulación mediante *software*

La simulación es una herramienta indispensable para el diseño de antenas pues permite hacer pruebas y ayuda a efectuar cambios de diseño sin el malgasto que supone realizar físicamente cada prototipo de la antena hasta alcanzar el diseño final.

Debido a las distintas técnicas numéricas empleadas, como el método de momentos o de elementos finitos; y a la gran variedad de antenas distintas que existen, hay en el mercado muchas soluciones de *software* distintas como Momentum, CST, EMPro o el módulo de antenas de Matlab.

2.2.1.2 Caracterización en cámaras anecoicas

De acuerdo con [20], una cámara anecoica es una cámara aislada utilizada para realizar mediciones de ganancia y diagramas de radiación de una antena. Aunque existen otros como el abocinado, el tipo más habitual es la cámara rectangular que consta de un dispositivo transmisor colocado en una de las paredes y la ABP colocada en el extremo opuesto sobre un soporte que permite la rotación alrededor de los tres ejes espaciales pudiendo obtener así diagramas de radiación en los planos azimutal y vertical, además de realizar estudios de la polarización de la antena.

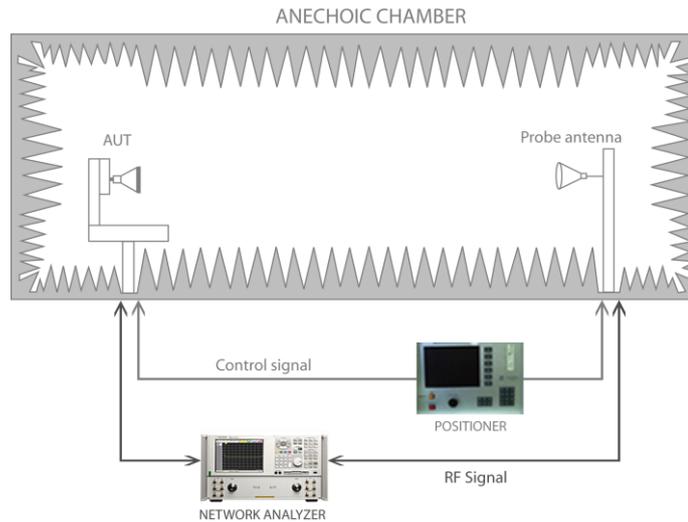


Figura 2-1 Esquema de una cámara anecoica para la obtención del diagrama [21]

Para simular condiciones de propagación en espacio libre se deben evitar las reflexiones, por lo tanto, las paredes, techo y suelo de las cámaras anecoicas se recubren de un material radioabsorbente, como la fibra de vidrio, con una geometría específicamente diseñada: típicamente formas piramidales o de cuña.

Como curiosidad, las cámaras anecoicas tienen muchas otras utilidades como, por ejemplo, medir la sección equivalente radar (RCS) de sistemas militares como se ilustra en la Figura 2-2.



Figura 2-2 Cámara anecoica de la USAF en la Base Aérea Edwards (California) [22]

2.2.2 Caracterización de antenas sobre el terreno

La obtención del diagrama de radiación de la antena sobre el terreno se realiza bien debido a que su caracterización en una cámara anecoica resulta impracticable por el tamaño que se necesitaría para conseguir condiciones de campo lejano, o bien porque se quiere comprobar el comportamiento de la antena *in situ*.

En el primer caso, se busca minimizar los efectos negativos de las reflexiones en el terreno, para ello se colocan ambas antenas en una disposición que provoque una interferencia constructiva entre el rayo directo y el reflejado, y se coloca la ABP sobre un soporte que permita orientarla como el esquema de la Figura 2-3.

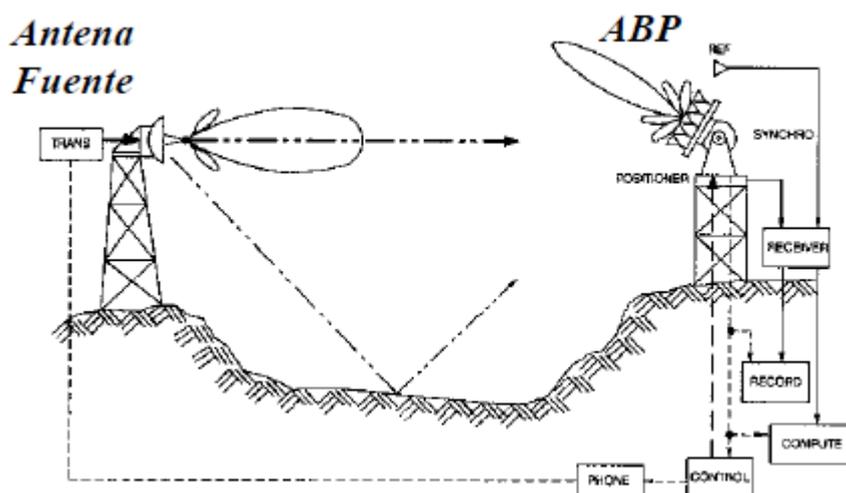


Figura 2-3 Esquema caracterización de antena sobre el terreno [23]

En el caso de que se busque caracterizar una antena en la posición en la que va a estar instalada para determinar su diagrama de cobertura, la ABP no debe rotar pues no se mediría el efecto del terreno y elementos circundantes, por lo tanto se han ideado otros métodos, generalmente basados en aeronaves pilotadas o UAS, que permiten tomar medidas de la potencia emitida en todas las direcciones alrededor de la antena.

La UIT ha emitido el informe [24] en el que se describen los procedimientos y el material recomendado así como los factores a tener en cuenta para realizar la caracterización con una aeronave. Dicho informe es una buena referencia tanto para el uso de aeronaves pilotadas como para su aplicación con UAS.

Por otra parte, en muchas universidades e instituciones se está investigando con el uso de drones para la consecución de esta tarea [25]. Las ventajas que aporta el uso de los UAS son claras:

1. Su coste es mucho menor que el que supone usar aeronaves pilotadas.
2. Su uso permite la obtención de los diagramas tridimensionales de una manera bastante sencilla con solo cambiar la programación de la trayectoria de vuelo.
3. Un buen sistema de posicionamiento y autopiloto proporcionan una altísima exactitud que redundará en diagramas más precisos.

2.2.3 Caracterización de antenas por el CEMEDM

Ya que uno de los objetivos de este trabajo es crear un sistema que presente una alternativa real al método utilizado actualmente por el CEMEDM, se ha creído conveniente dedicar un apartado específico a dicho método.

Tal y como recoge el procedimiento del propio Centro [15]: «El proceso de levantamiento de diagramas requiere que el buque navegue en círculos de radio reducido mientras emite o recibe con sus antenas. El CEMEDEM, desplegado en tierra o abordo, recibe la señal emitida midiendo la variación de la potencia recibida conforme el buque evoluciona obteniendo los correspondientes diagramas. Para ello cuenta con equipos receptores (analizadores de espectro) y con un sistema de control de medida (ordenador y programa de medidas)».

Para cada banda de frecuencias, el CEMEDEM ha seleccionado una frecuencia central representativa de la banda sobre la que transmite la señal piloto para la obtención del diagrama. Generalmente, esta señal piloto es transmitida por el buque y es recibida por una antena omnidireccional sintonizada en la banda correspondiente. En este proceso de medida, el CEMEDEM asume una incertidumbre de 2,54 dB.

Este procedimiento de medida presenta tres grandes inconvenientes.

1. Solo permite obtener el diagrama de radiación en el plano azimutal, útil de conocer para antenas de comunicaciones entre buques o con tierra, pero insuficiente para antenas de radar, direcciones de tiro y comunicaciones con aeronaves.
2. Requiere la colocación de la antena y los aparatos de medida en una zona despejada en tierra desde la que se tenga buena línea de visión con el barco evitando lo máximo posible las reflexiones en obstáculos.
3. Las medidas se realizan de forma manual: cuando el personal del barco confirma el paso por cada una de las demoras por las éste evoluciona, las anuncia por un canal radio auxiliar, son recibidas por el personal en tierra cada una y estos dan la orden de grabación de la potencia recibida en este instante. Este procedimiento añade cierto retardo en la toma de medidas que se traduce en un registro del valor de potencia para una demora que no es la correcta. Además introduce el factor humano en un procedimiento totalmente mecánico: cualquier error de interpretación de los comandos de voz o un descuido, puede inutilizar una campaña de medidas.

2.2.4 Técnicas para la medida de potencia

De acuerdo a [12], la potencia de una onda electromagnética en un punto dado en coordenadas esféricas respecto a la fuente transmisora es proporcional al cuadrado del valor eficaz e del campo eléctrico:

$$p_{tr}(\theta, \varphi, d) = \frac{e^2(\theta, \varphi)}{30d^2}$$

Para la medición de la potencia propiamente dicha, existe una serie de tecnologías utilizadas en distintos aparatos de medida; tal y como se explica en [19], las más importantes son los sensores térmicos, los detectores de envolvente, los receptores superheterodinos para medida (analizadores de espectro, etc.) y el cálculo basado en la digitalización de la señal.

2.2.4.1 Sensor térmico

Los sensores térmicos generalmente se basan en el uso de resistencias variables con la temperatura, también denominados termistores. Estos termistores actúan como carga de las ondas de radiofrecuencia recibidas normalmente a través de una guía de onda, aumentando de temperatura gradualmente en función de la potencia de la onda recibida y variando su resistividad.

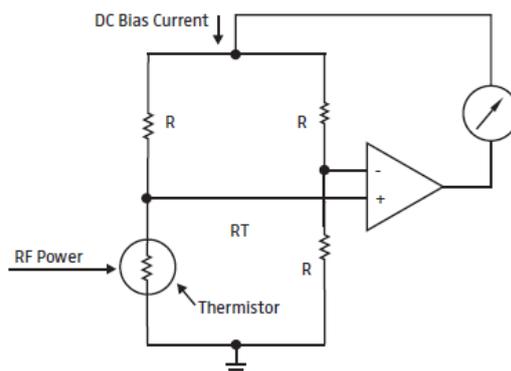


Figura 2-4 Esquema típico de sensor térmico de potencia de RF [19]

Debido a que los termistores son elementos altamente no lineales, normalmente se introducen en un circuito similar al de la Figura 2-4 en el cual se utiliza un puente con una corriente continua cuya función es calentar el termistor hasta que alcance el mismo valor que el resto de resistencias del circuito equilibrándolo y consiguiendo una respuesta uniforme por parte del sensor. La medida de la potencia se obtiene en función del valor de la corriente continua.

2.2.4.2 Detector de envolvente

Los detectores de envolvente, ampliamente utilizados para demodular señales de AM, también demuestran mucha eficacia a la hora de medir la potencia contenida en una onda electromagnética como función directa de la amplitud de la misma.

El detector de envolvente más sencillo (Figura 2-5) basa su funcionamiento en un diodo utilizado para rectificar la señal y un condensador que se carga rápidamente cuando el valor de la señal aumenta, y se descarga lentamente cuando este disminuye. Finalmente, el voltaje medido en la carga a la salida es la amplitud de la envolvente de la señal que es función de la potencia de la onda recibida.

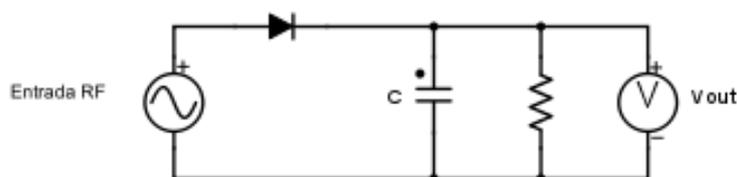


Figura 2-5 Detector de envolvente sencillo

2.2.4.3 Receptores superheterodinos para medida

Los analizadores de espectro, analizadores de redes y otros aparatos analógicos de medida similares funcionan de la misma manera que un receptor superheterodino añadiendo un filtro paso-banda en frecuencia intermedia (IF) cuyo ancho de banda marca la resolución en frecuencia del aparato y un detector de envolvente que sirve para determinar la amplitud de cada componente espectral. Para medir la potencia, estos sistemas barren el ancho de banda seleccionado con la resolución de ancho de banda (RBW) determinada por el usuario y calculan la media de todas las componentes espectrales.

En la Figura 2-6 se puede ver el esquema de bloques de un sencillo analizador de espectro analógico. Se aprecia cómo el oscilador local aumenta gradualmente de frecuencia, para el barrido de todo el ancho de banda deseado, a la vez que el filtro paso-banda ajusta su frecuencia central.

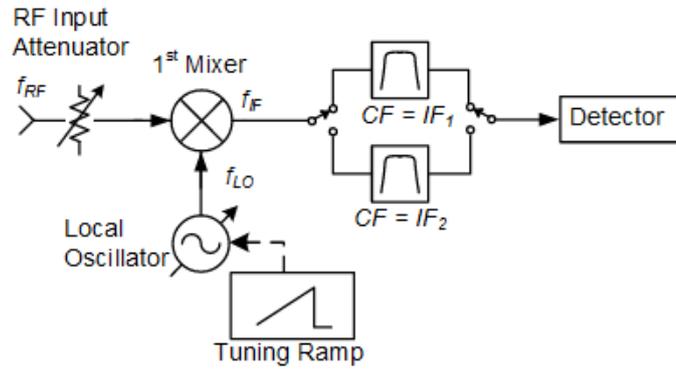


Figura 2-6 Esquema de bloques de un analizador de espectro superheterodino de una etapa [26]

2.2.4.4 Digitalización de la señal

Mediante un conversor analógico-digital, se puede digitalizar directamente la señal en RF o en IF si la anterior fuera demasiado alta. Una vez obtenidas las muestras, mediante el procesado digital de la señal se puede calcular la amplitud de la señal en el dominio del tiempo, o realizar una transformada de Fourier rápida (FFT) y trabajar en el dominio de la frecuencia. Esta técnica es empleada por los modernos analizadores de espectro digitales como el de la Figura 2-7.



Figura 2-7 Analizador de espectro digital portátil de la casa Rohde&Schwarz [27]

2.3 Los vehículos aéreos no tripulados

En las últimas décadas, los vehículos aéreos no tripulados, también conocidos como drones, han dejado de ser meros proyectos militares de presupuesto multimillonario, para convertirse en una tecnología cada vez más presente en nuestra vida diaria, no solo utilizados en operaciones militares, sino también por organismos estatales, empresas y particulares.

2.3.1 Concepto de UAS

En el contexto de esta tecnología, existe una serie de conceptos de uso habitual que, si bien parecen el mismo, tienen connotaciones diferenciadoras. El UAV es entendido como: «Aeronave capaz de operar en ausencia de piloto ubicado físicamente en ella» [28], por lo tanto, tal y como indica [29], esta definición se refiere simplemente a la plataforma, mientras que el sistema aéreo no tripulado (UAS) incluye a su vez todos los elementos que requiere la plataforma para su funcionamiento: como mínimo, una estación de control en tierra (GCS por sus siglas en inglés).



Figura 2-8 Esquema global de funcionamiento de un UAS [30]

Respecto a su modo de funcionamiento, estas aeronaves pueden ser totalmente autónomas realizando una misión programada antes del vuelo, o pueden ser pilotadas remotamente, las conocidas como RPA, cuyo salto cualitativo respecto a las clásicas aeronaves de los aficionados al aeromodelismo consiste en que permiten el pilotaje por fuera de la línea de visión del operador en tierra, ya que cuentan con sofisticados enlaces de datos que permiten el control y la monitorización de la aeronave en tiempo real.

Por último, aunque puede dar lugar a confusión, nunca debe perderse la perspectiva de que el rasgo diferenciador de esta tecnología frente a los misiles o las bombas guiadas, es que los drones están diseñados para ser reutilizables [28].

2.3.2 Los UAS hoy en día

Desde que en 1898 el inventor de origen serbio Nikola Tesla patentara el uso de ondas de radiofrecuencia para el control remoto de un bote, las grandes potencias mundiales han tratado de implementar esta tecnología en sus desarrollos armamentísticos.

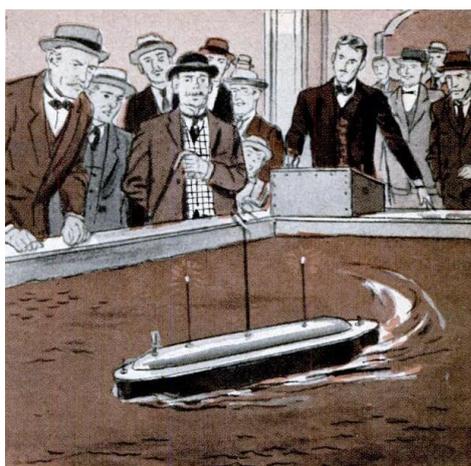


Figura 2-9 Demostración de Tesla en el Madison Square Garden [31]

Tras algunos desarrollos más o menos afortunados durante las dos Guerras Mundiales del siglo XX, los UAS experimentaron su gran desarrollo a partir de los años 70 con la incorporación de las modernas tecnologías de computación, control y geolocalización; siendo utilizados sistemáticamente en operaciones militares desde su primer empleo en combate con éxito por parte de Israel en 1982 [31].

A día de hoy, los drones son una herramienta fundamental en las operaciones militares, hecho constatado con el exponencial aumento que ha tenido la partida del presupuesto militar estadounidense dedicada a ellos (Figura 2-10).

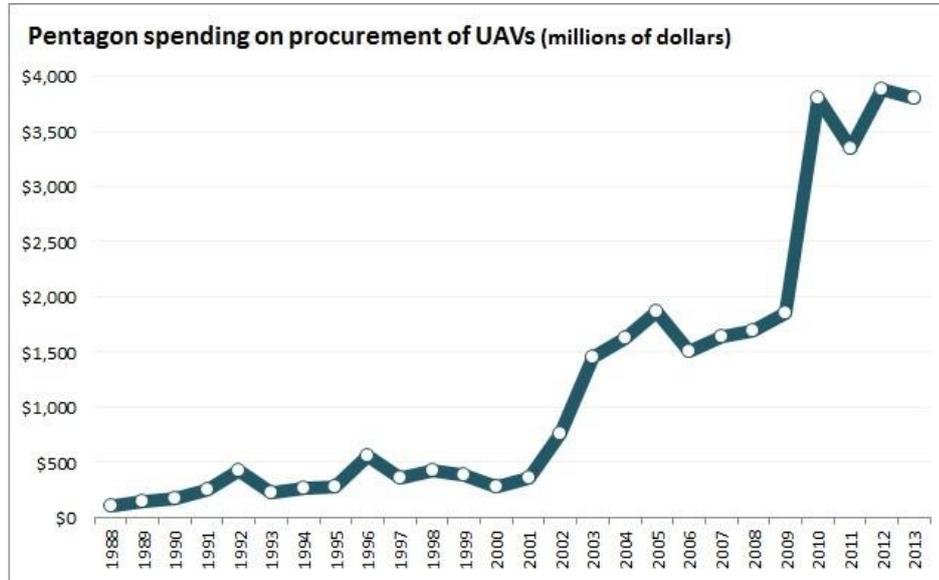


Figura 2-10 Presupuesto dedicado a UAVs militares americanos desde 1988 a 2013 [32]

En operaciones militares, los UAS se equipan con modernos sensores y cámaras digitales haciéndolos ideales para misiones de vigilancia y reconocimiento sin arriesgar ninguna vida. A su vez, las más de 1.300 bajas confirmadas oficialmente en más de 1.000 ataques selectivos en Afganistán en 2016 [33] demuestran que son un arma muy precisa y letal pero cuestionable desde el punto de vista ético [34].

En el ámbito civil, la explosión ha sido mucho más tardía y está comenzando prácticamente ahora mismo. Entre las aplicaciones que tienen están la de vigilancia forestal [35], de fronteras o de líneas de tendido eléctrico por parte de organismos estatales; la entrega de paquetes [36] y muchas otras que están por venir como la mostrada en la Figura 2-11.



Figura 2-11 Prototipo de taxi dron [37]

Con el desarrollo de las baterías LiPo (*Lithium Polymer*), los drones de propulsión eléctrica se han convertido en una realidad abaratando sus costes y permitiendo su llegada al gran público. En torno a estos drones eléctricos han surgido comunidades de aficionados y empresas que conjuntamente

desarrollan *hardware* y *software* de control de vuelo. Uno de los más importantes es Dronecode [38] que cuenta con numerosos proyectos como la GCS Mission Planner, controladoras de vuelo de *hardware* abierto como la Pixhawk y *software* de control de vuelo basado en el protocolo MAVLink como PX4.

2.4 La radio definida por *software*

La radio definida por *software* (SDR) es una tecnología emergente que, si bien no es demasiado conocida fuera de los círculos profesionales, representa un gran salto cualitativo respecto a la radio tradicional con una importancia cada vez mayor en el ámbito civil y militar.

2.4.1 Concepto de SDR

El Wireless Innovation Forum es una organización internacional encargada de coordinar y dirigir la investigación y el desarrollo de nuevas tecnologías inalámbricas como la SDR y la radio cognitiva. Este organismo, en el que participan numerosas empresas e instituciones de renombre como AT&T, Indra o la NASA [39]; define en la SDR como: «Aquella radio en la que todas o parte de las funciones de su capa física están definidas en *software*» [40]. Tradicionalmente, las funciones de la capa física de la radio han sido la amplificación, filtrado, demodulación, etc. Sin embargo, esta nueva tecnología permite implementar todas estas funciones utilizando *software*, presentando frente a la radio clásica algunas ventajas que, de acuerdo a [41], son:

1. Facilidad de diseño y fabricación: Pues es más sencillo el desarrollo de *software* que se ejecute en procesadores cada vez más baratos y potentes que el de *hardware* con circuitos específicos para cada una de las funciones de la radio.
2. Flexibilidad: Un mismo *hardware*, permite la ejecución de distinto *software* diseñado para soportar distintos protocolos de comunicaciones.
3. Menor número de componentes electrónicos, que reduce la probabilidad de fallos y el tamaño de los equipos.
4. Uso de técnicas avanzadas de procesamiento de señal: Al realizar el procesado de señales digitales, se reduce la inexactitud de los componentes analógicos y abre la puerta a nuevas técnicas y protocolos de comunicación.

La SDR ideal consistiría en una antena que actúe como interfaz de transmisión/recepción de las ondas de RF, un convertor analógico-digital (ADC) o digital-analógico (DAC) que transforme la señal a transmitir o recibir y un procesador digital de señales (DSP) que ejecute el *software* implementado de modulación, demodulación, filtrado, etc.

Uno de los problemas que existen a día de hoy, es que los ADC y DAC no son capaces de trabajar a las altas frecuencias requeridas en determinadas aplicaciones, por lo tanto, en la práctica se incluyen componentes, conocidos como *RF Front-end*, que filtran la señal sintonizada y convierten analógicamente las ondas de RF a frecuencia intermedia, trabajando de la misma manera que en un receptor superheterodino.

Para la definición completa de la señal, en el paso a IF de la señal recibida, se obtiene una réplica de la misma pero desfasada 90° , obteniendo las componentes en fase (I) y en cuadratura (Q) [42]. Estas componentes son digitalizadas en dos ADC, por lo que la señal discreta obtenida consiste en un flujo continuo de muestras complejas de la forma $I + jQ$, lo cual es una abstracción matemática que representa cualquier señal de manera tridimensional permitiendo obtener la amplitud ($\sqrt{I^2 + Q^2}$) y la fase ($\tan^{-1}(\frac{Q}{I})$) para cada muestra tomada en un instante de tiempo.

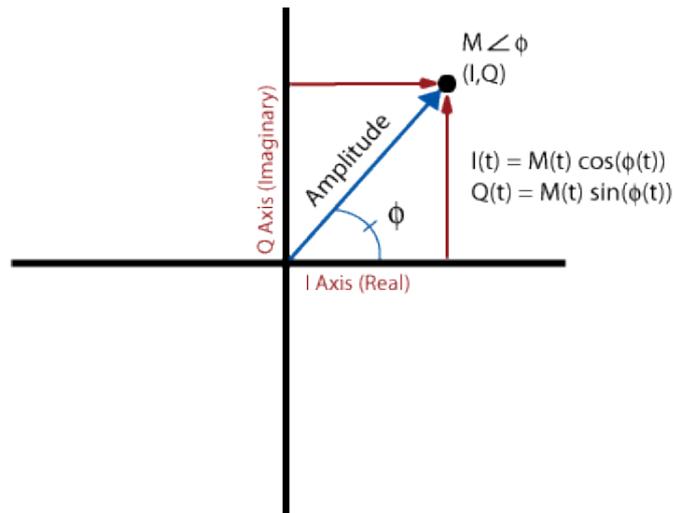


Figura 2-12 Representación de componentes I y Q de la señal [42]

Esta señal digitalizada en IF es convertida a banda base mediante los dispositivos DDC y DUC, siendo procesada finalmente con DSP o CPU de ordenador, quedando un esquema como el de la Figura 2-13.

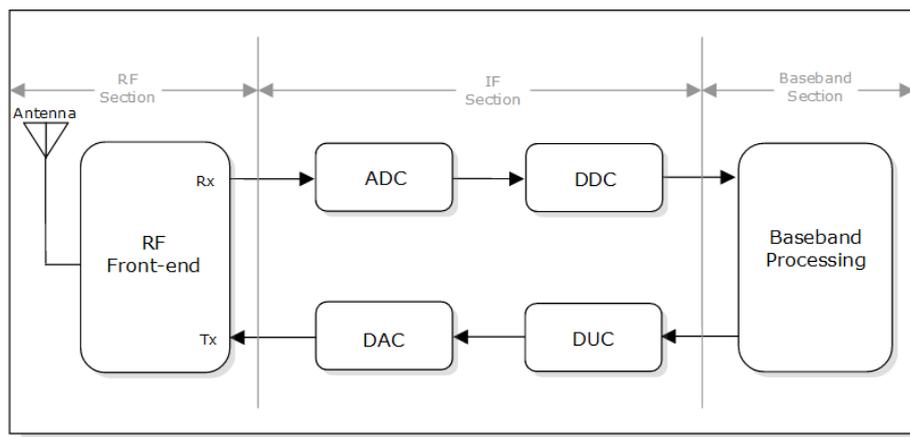


Figura 2-13 Esquema de bloques típico de una SDR [43]

2.4.2 Presente y futuro de la SDR

Los primeros desarrollos de SDR se produjeron en el ámbito militar con el proyecto SpeakEasy de la DARPA del Departamento de Defensa de los EEUU de la última década del siglo XX [44], por lo tanto aún es una tecnología en su fase inicial de implementación, sin embargo, tiene un futuro muy prometedor.

En el ámbito militar, además de las distintas fases del proyecto SpeakEasy en Estados Unidos, en Europa se está desarrollando el programa ESSOR (*European Security Software defined Radio*) que define una arquitectura común para radios tácticas basadas en SDR y que ya ha sido probado con éxito en 2015 [45]. Este proyecto es un esfuerzo conjunto de 6 naciones entre las que se incluye España a través de la empresa Indra Sistemas. De cara al futuro, la DGAM considera que esta tecnología será imprescindible para la consecución de la capacidad nacional de contar con «radiocomunicaciones militares propias para el entorno táctico, que sean seguras y con unas prestaciones suficientes para dar soporte a sus necesidades C4ISR (*Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance*)» [46].

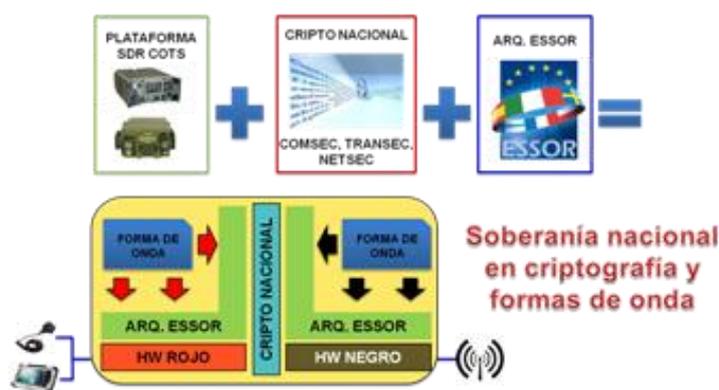


Figura 2-14 Concepto nacional de empleo de la SDR en el ámbito militar [46]

En el ámbito civil, esta tecnología aún no tiene una implantación grande, sin embargo es destacable la comunidad de radioaficionados que está surgiendo en torno a ella debido al uso de los baratos chips receptores RTL (Figura 2-15), utilizados originalmente como sintonizadores de DVB-T para ordenador [47]. Estos dispositivos cuentan un sintonizador y el chip receptor que digitaliza la señal recibida, y utilizan el microprocesador de un ordenador a modo de DSP para realizar las operaciones de *software*. Para estas operaciones, se han desarrollado algunas bibliotecas de código abierto, la más importante y que cuenta con una activa comunidad de desarrolladores es GNURadio, la cual está implementada en C++ y Python, e incluye un entorno gráfico que se basa en la programación mediante bloques para la obtención de un código funcional. Esta tecnología permite realizar funciones como la demodulación de la radio FM comercial, el seguimiento de aviones o incluso realizar llamadas a través de la red GSM [48].



Figura 2-15 Receptor RTL USB conectado a antena [48]

El futuro de la SDR es lo que se conoce como radio cognitiva, definida en [40] como: «Aquella radio cuyos sistemas de comunicación son “conscientes” de su entorno y estado interno, permitiendo la toma de decisiones sobre su modo de operación en base a esta información y unos objetivos predefinidos». Por ejemplo, la radio cognitiva será capaz de cambiar de canal de frecuencias en función de lo ocupada que esté la banda sobre la que actúa. Ya se han comenzado a dar algunos pasos en esta dirección con la red móvil de quinta generación (5G) que está aún en fase de diseño.

3 DISEÑO DEL SISTEMA

3.1 Visión general

Para la consecución del objetivo principal del proyecto, que es la obtención del diagrama de cobertura de una antena embarcada utilizando el dron, se ha diseñado el sistema propuesto en la Figura 3-1 formado físicamente por el segmento aéreo y el segmento terrestre.

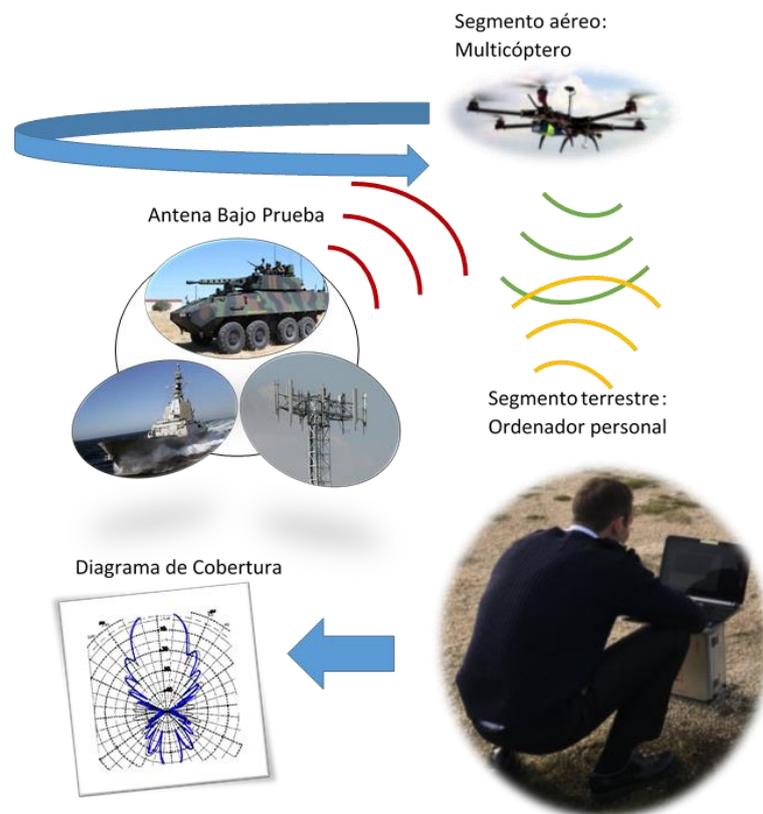


Figura 3-1 Visión general del sistema diseñado

El segmento aéreo está compuesto por un UAV que vuela describiendo un círculo alrededor de la ABP y, a la vez, recibe la señal transmitida por la ABP mediante una SDR. El segmento terrestre está basado en un ordenador personal que controla el vuelo del dron, a la vez que procesa las muestras recibidas por la SDR para obtener medidas de la potencia que darán lugar al diagrama de cobertura.

A su vez, se han definido unos criterios de diseño para el funcionamiento del sistema. Estos criterios son:

1. El alcance de este proyecto se restringe a las bandas de comunicaciones de VHF y UHF debido a limitaciones en las dimensiones de la antena que puede embarcar el UAV.
2. Para la obtención del diagrama, se considera suficiente con obtener una precisión de 3 medidas de potencia por cada grado cubierto alrededor de la antena.

3.2 Diseño global del sistema

El sistema diseñado (Figura 3-2) cuenta con dos módulos bien diferenciados e independientes el uno del otro pero que no coinciden con la separación física pues los dos contienen elementos en ambos segmentos: por un lado la plataforma de vuelo que consta del UAV en el segmento aéreo y las rutinas de control de vuelo en el terrestre, y, por otro lado, la medida de potencia que consta de la SDR embarcada en el segmento aéreo y las rutinas de procesamiento de muestras y obtención de potencia del segmento terrestre.

La integración de los dos módulos en uno final se realiza mediante *software*, creando una rutina que inicie ambos subsistemas y los combine obteniendo los datos correlacionados de cada medida de potencia realizada con una posición del dron relativa a la antena.

La representación final de los datos obtenidos en un diagrama se ha realizado mediante Matlab, aunque puede adaptarse fácilmente para su uso con el *software* con el que cuenta el CEMEDM.

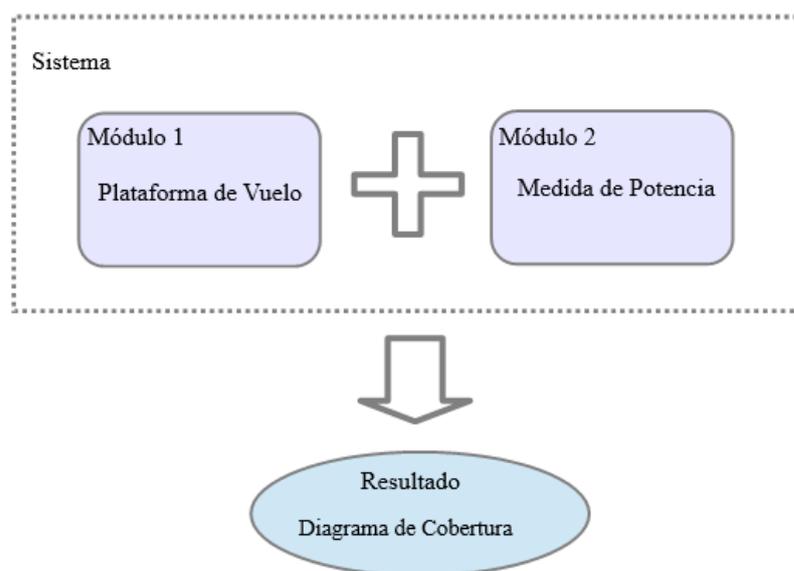


Figura 3-2 Arquitectura modular del sistema diseñado

La ventaja de dividir el sistema en dos módulos totalmente independientes es clara:

1. Permite abordar cada problema por separado pudiendo enfocar todo el esfuerzo en un objetivo menor cada vez.
2. Aporta flexibilidad: Al ser totalmente independientes, un cambio en uno no afecta al otro, si bien esto es cierto totalmente en el *software*, en el *hardware* no lo es ya que el espacio en la plataforma es limitado, y si se necesitara cambiar los elementos utilizados para la medida, por ejemplo, quizá habría que hacer una reconfiguración de la disposición del resto de elementos.
3. Facilita la realización de pruebas ya que no se necesita necesariamente que ambos sistemas estén funcionando para probar uno de ellos; esto resultó muy beneficioso al realizar las

pruebas de medida debido a que la disponibilidad para las pruebas de vuelo fue muy reducida.

A continuación, se abordará el planteamiento de ambos subsistemas de manera independiente, exponiendo el planteamiento de cada uno de los problemas, la solución propuesta, y el equipo utilizado en cada uno de los casos.

3.3 La medida de la potencia

3.3.1 El problema de la toma de medidas

Para ejecutar una toma de medidas adecuada hay que tener en cuenta una serie de aspectos. Algunos de estos aspectos son determinados por el operador en el planeamiento pues dependen de la antena que se va a caracterizar, este es el caso de la distancia a la que debe volar el dron respecto a la antena y de la altura del vuelo. Otros aspectos, son considerados aspectos de diseño y están muy relacionados con la señal transmitida, la antena bajo prueba trabajando en transmisión o recepción y con el método que utiliza este sistema para realizar las medidas.

3.3.1.1 Distancia de vuelo respecto a la antena

Debido a las bandas de frecuencia para las que está diseñado este proyecto, la propagación se va a producir por onda espacial entre la antena bajo prueba y la antena receptora en el dron siempre con visibilidad directa entre ambos, por lo tanto, para la elección de la distancia a la que debe volar el dron respecto a la antena, el operador debe tener en cuenta los siguientes factores:

1. Eslora del buque.
2. Radio del primer elipsoide de Fresnel.
3. Distancia del campo lejano.
4. Efecto del multitrayecto.

Cuando se caractericen antenas embarcadas es altamente recomendable volar por fuera del buque en todo momento por seguridad del personal transitando la cubierta y seguridad de la propia plataforma ya que no cuenta con sensores de detección de obstáculos. Por lo tanto, la medida de la **eslora** proporciona un margen de seguridad, sea cual sea la posición de la antena, y es una buena distancia para comenzar un estudio más detallado de la distancia óptima.

El segundo factor a tener en cuenta es el **radio del primer elipsoide de Fresnel**. Para evitar la difracción, el rayo directo entre las dos antenas debe pasar por encima de cualquier obstáculo con un margen mayor que el radio de la primera zona de Fresnel. En la mar, debido a la ausencia de obstáculos, y suponiendo que el rayo directo es paralelo a la superficie, debe ajustarse la distancia de vuelo para que la altura del mismo sea suficiente para salvar este primer elipsoide (Figura 3-3) cuyo radio viene dado por la ecuación:

$$r = \sqrt{\frac{\lambda d_1 d_2}{d}}$$

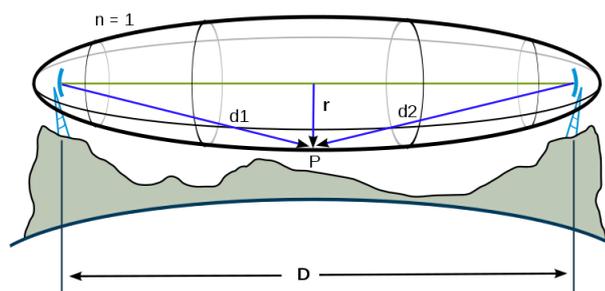


Figura 3-3 Esquema del primer elipsoide de Fresnel [49]

El tercer factor a tener en cuenta es el **campo lejano**. Tal y como se ha descrito en el apartado 1.2.1.1, la distancia mínima de medida r debe ser tal que la recepción se produzca en la región del campo lejano, dada por la ecuación:

$$r = \frac{2D^2}{\lambda}$$

El último factor a tener en cuenta es la **atenuación por multitrayecto**. El multitrayecto es un fenómeno que se produce por la interferencia entre el rayo directo y los rayos reflejados que han seguido trayectorias distintas de propagación en un tiempo distinto, por lo tanto alcanzan la antena receptora con fases distintas. Es posible que esta interferencia sea constructiva, amplificando la potencia recibida, sin embargo también es posible que esta interferencia sea destructiva llegando a provocar incluso un nulo en recepción para esa distancia entre ambas antenas.

Las causas que provocan la existencia de estos rayos reflejados son muy variadas y de carácter aleatorio especialmente en entornos urbanos, reflexiones en el suelo u obstáculos, movimientos de vehículos o personas, precipitaciones, difracción... Este carácter aleatorio requiere el uso de modelos estadísticos para tratar de predecir su efecto y que siguen la distribución de Rayleigh o la distribución de Rician [50].

Estos modelos son útiles para determinar los márgenes y los umbrales de potencia necesarios cuando se está dimensionando un sistema de radiofrecuencia, sin embargo, cuando se caracteriza una antena, se trata de reducir al máximo estas variables aleatorias ejecutando las medidas en un entorno controlado, quedando siempre presente el efecto del rayo reflejado en el suelo (o superficie del mar). Tal y como explica [12], el modelo de propagación de tierra plana (Figura 3-4), tiene en cuenta el efecto de este rayo reflejado, e introduce un término de las pérdidas en la ecuación de Friis que depende de la distancia entre las dos antenas, la altura de las mismas, el tipo de suelo y la polarización de la onda.

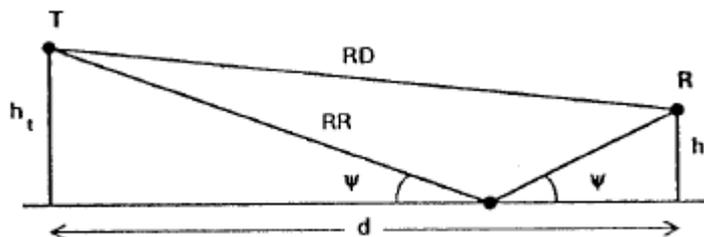


Figura 3-4 Modelo de propagación de tierra plana [12]

3.3.1.2 Altura de vuelo

Para la elección de la altura de vuelo, el factor más importante es la altura de la propia antena bajo prueba. Idealmente, la altura de vuelo debería coincidir con la altura de la máxima potencia emitida por la antena, que en la mayor parte de los casos coincide con la altura física a la que está la antena.

Por lo tanto, para determinar la altura del vuelo, habrá que conocer, al menos aproximadamente, el diagrama de radiación vertical de la ABP, teniendo en cuenta que si se vuela por fuera de la zona de máxima potencia radiada (demasiado alto o demasiado bajo), la medida realizada no va a ser adecuada por la escasa relación señal a ruido (SNR) que se va a obtener.

También habrá que tener en cuenta el efecto del multitrayecto aquí. Sin embargo, se recomienda fijar la altura de vuelo en función del criterio anterior, y variar la distancia respecto a la ABP para evitar el efecto negativo de este efecto.

3.3.1.3 Ejecución de la medida

Una vez definidas las condiciones en las que debe ser realizada la medida (altura y distancia), el problema reside en realizar la medida en sí con el elemento embarcado en el dron: la SDR.

En primer lugar, se debe seleccionar si la ABP trabajará en transmisión o en recepción. Ya que se cumple el principio de reciprocidad descrito en 1.2.4.1, los factores a tener en cuenta dependen del diseño del sistema en sí, por ejemplo la facilidad de embarcar el equipo transmisor en el dron o no.

En segundo lugar, es necesario conocer perfectamente el ancho de banda de la señal piloto transmitida para así ajustar adecuadamente los filtros y la frecuencia de muestreo.

Por último, hay que decidir cómo se van a realizar las medidas: en el dominio de la frecuencia tal y como trabajan los analizadores de espectro o en el dominio del tiempo.

Como se ha explicado en el apartado 2.4.1, el convertor analógico-digital (ADC) proporciona muestras complejas con las componentes en fase (I) y en cuadratura (Q) a la frecuencia de muestro seleccionada. Estas muestras no son otra cosa que la discretización de la señal en el dominio del tiempo.

Para calcular la potencia recibida en el dominio de la frecuencia, hay que realizar una transformada de Fourier con una ventana, o resolución de ancho de banda, determinada y a su vez calcular la media de las potencias de estas ventanas [51]. Sin embargo, este método requiere mucho tiempo de cálculo del procesador, sin presentar ventajas reales frente al cálculo en el dominio del tiempo.

Para el cálculo trabajando en el dominio del tiempo basta con operar con las muestras complejas obtenidas directamente del ADC. Para obtener la amplitud instantánea, bastará con calcular el módulo del vector definido por las componentes I y Q .

$$E = \sqrt{I^2 + Q^2}$$

Ya que la potencia es función de la amplitud al cuadrado, y que lo que se busca representar en el diagrama son potencias relativas en decibelios, para cada medida P bastará con calcular:

$$P = 10 \log \frac{I^2 + Q^2}{P_{max}}$$

Sin embargo, como se ha dicho antes, esta medida es la de la potencia de la señal para un instante dt representado por una muestra, haciendo analogía con el cálculo en el dominio de la frecuencia, equivaldría a coger un df del espectro y calcular su potencia, olvidando todo el ancho de banda que ocupa la señal realmente. Por lo tanto, se debe calcular la media de la potencia de un grupo de muestras de tal manera que se calcule la potencia de la señal para un $\Delta t = n \cdot T_s$ siendo n el número de muestras y T_s el período de cada muestra.

Aunque la lógica dicta que existe un n mínimo con el que se calcula la potencia de la señal sin error apreciable, para conocerlo hace falta conocer perfectamente la señal que se va a medir y calcular el Δt mínimo que se necesita para cubrir todo el período cuando es una señal periódica. Cuando la señal es desconocida o no periódica, el número de muestras que se seleccione para realizar la media es una solución de compromiso entre la precisión de la medida y cada cuánto Δt se quiere obtener una medida. Por lo tanto, cada medida P se calcula como:

$$P = 10 \log \frac{\sum_n (I_n^2 + Q_n^2)}{n} - 10 \log P_{max}$$

Por facilidad, el cálculo de P_{max} se realizará en el postprocesado de las medidas obtenidas.

3.3.2 Solución propuesta

Tras explicar el planteamiento y las variables que afectan a la ejecución de la medida, se pasará a exponer la solución propuesta comenzando con las decisiones tomadas:

En primer lugar, se decidió que la antena bajo prueba trabaje en transmisión. Esta decisión se tomó para evitar que fuera el dron el que transmitiera fundamentalmente porque el equipo transmisor del CEMEDEM no se podía embarcar en el UAV, y en caso de usar una SDR con capacidad para transmitir, esta es muy cara y no es aconsejable embarcarla en el dron donde está sometida a un riesgo mayor. Además, en la práctica, el CEMEDEM suele caracterizar las antenas embarcadas en transmisión, poniendo estas a transmitir y registrando en tierra los valores de potencia recibida a medida que el barco evoluciona.

En segundo lugar, se definió perfectamente la señal piloto transmitida por la ABP. Esta señal consiste en una portadora de 868 MHz modulada en FM de banda estrecha por un tono de 1 KHz. La banda de frecuencias se escogió porque es utilizada para dispositivos genéricos de corto alcance y de baja potencia [52], resultando ser una banda generalmente libre de interferencias.

Por otra parte, esta señal ocupa según la Regla de Carson 10 KHz [53]. Puesto que el receptor SDR traslada dicho espectro a banda base y luego lo muestrea, la frecuencia de muestreo mínima que garantiza la recuperación de la señal original sin *aliasing* [53] es 20 KHz. Este valor también permite ajustar la frecuencia de corte del filtro de paso bajo para limitar en banda la señal a medir.

Por último, se decidió realizar las medidas en el dominio del tiempo. Para cumplir el criterio de 3 medidas por cada grado (apartado 3.1), y tras realizar una serie de pruebas, se determinó como criterio de diseño que la toma de medidas 10 veces por segundo era adecuada. Por lo tanto, resolviendo la ecuación expuesta en el apartado anterior $\Delta t = n \cdot T_s$, teniendo en cuenta que $\Delta t = 0,1 s$ y la frecuencia de muestreo es 20 KHz, para cada medida se hará un promedio con 2.000 muestras.

Una vez tomadas las decisiones anteriormente expuestas, se procede a explicar el diseño del módulo de medida de potencia representado en la Figura 3-5:

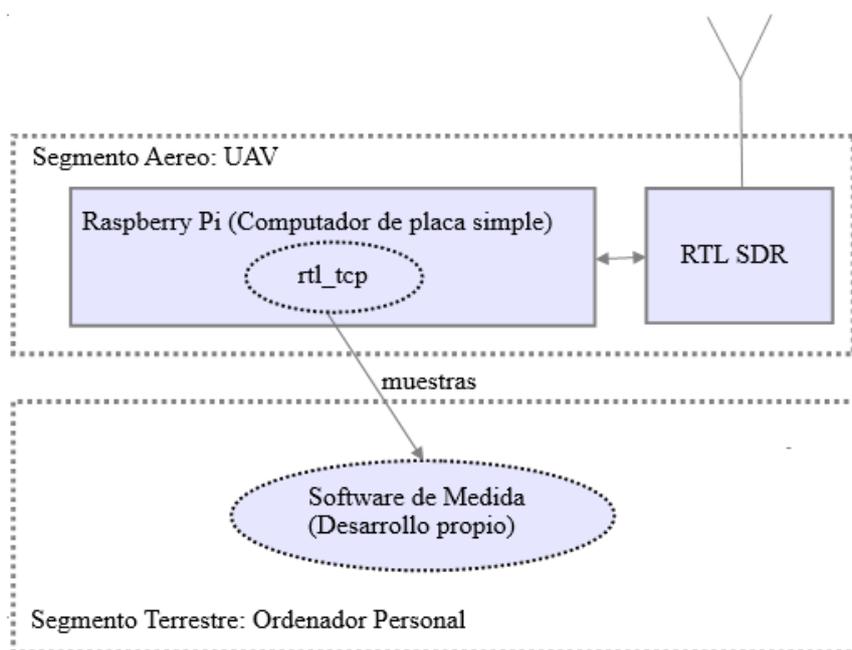


Figura 3-5 Esquema sistema de medida de potencia

En el segmento aéreo, embarcado en el UAV, se encuentra el receptor SDR conectado al ordenador de a bordo que es compartido por el módulo de vuelo, la Raspberry Pi. En este ordenador de a bordo se ejecuta la utilidad `rtl_tcp`, servidor de código abierto escrito en C que funciona tomando las muestras recibidas por una antena omnidireccional digitalizadas por la SDR y enviándolas a través de un *socket* TCP a la dirección IP especificada. En este caso, las muestras se envían directamente al ordenador personal en el segmento terrestre. En este ordenador se ejecuta el *software* desarrollado con GNURadio que obtiene la potencia de la señal recibida a partir de las muestras proporcionadas.

3.3.3 Equipo de medida y software utilizado

En este apartado se va a proceder a describir el equipo seleccionado para este módulo de medida de potencia del sistema: equipo transmisor instalado con la ABP, equipo receptor y *software* utilizado.

3.3.3.1 Equipo transmisor

A pesar de que el equipo de transmisión utilizado no forma parte del sistema en sí, se ha creído conveniente describir el utilizado para las pruebas del sistema. Sin embargo, no debe perderse de vista que para la medida de antenas embarcadas, se utilizaría el propio transmisor del buque, con una forma de onda generada por material del CEMEDM.

El equipo utilizado ha sido la SDR de *hardware* abierto HackRF One (Figura 3-6) de Great Scott Gadgets. La razón fundamental de la elección de este dispositivo es que permite tanto la recepción como la transmisión. Algunas de sus características según [54] son:

- Banda de trabajo entre 1 MHz y 6 GHz.
- Máxima frecuencia de muestreo 20 MHz.
- Muestras complejas de 16 *bits* en total.
- Potencia máxima transmisión: 15 dBm, recepción: -5 dBm.
- Filtro de banda base y ganancia RF, IF y BB configurables mediante *software*.
- Alimentada por USB.
- Compatible con GNURadio.
- Conector de antena SMA hembra de 50 mA a 3,3 V.
- Conexión para sincronización de relojes cuando se combine con otras SDR.

Todas estas características demuestran que es una SDR muy potente que puede ser utilizada para muchísimas otras aplicaciones.



Figura 3-6 HackRF One utilizada para transmitir la señal piloto [54]

3.3.3.2 Equipo receptor

El dispositivo receptor utilizado es la SDR NESDR Mini+ de NooElec. Este receptor es un dispositivo de bajo coste basado en el chip RTL2832 y el sintonizador R820T. A pesar de solo permitir la recepción, esta combinación de chip y sintonizador es ampliamente utilizada entre los aficionados a

la SDR por ser muy barata, entre 20 y 40 €, y tener buenas características para su uso *amateur* descritas en [47]:

- Banda de trabajo entre 24 MHz y 1766 MHz.
- Máxima frecuencia de muestreo 2,4 MHz.
- Muestras complejas de 16 *bits* en total.
- Ganancia configurable mediante *software*.
- Alimentada por USB.
- Compatible con GNURadio.
- Conector de antena hembra MCX.

La elección de este dispositivo receptor (Figura 3-7) ha estado motivada por sus características adecuadas para su aplicación dentro de su bajo precio y reducido tamaño, ideales para un UAV en pruebas, además de por contar con una carcasa de aluminio que actúa como una jaula de Faraday, necesario en un entorno con tanta radiación en las proximidades (señales recibida, GPS y *wifi*).



Figura 3-7 Receptor SDR NooElec utilizado [55]

3.3.3.3 Software utilizado

El *software* desarrollado se ha basado, tanto para generar la señal transmitida como para calcular la potencia de la señal recibida, en la biblioteca de código abierto GNURadio.

Según su página web [56]: «GNURadio es un entorno de trabajo que permite a los usuarios diseñar, simular y desarrollar sistemas radio con alta capacidad para el mundo real. GNURadio es una biblioteca orientada a flujos altamente modular que incluye una gran variedad de bloques de procesado que pueden ser combinados para crear aplicaciones complejas de procesado de señal».

Por lo tanto, GNURadio incluye todas las funciones que se ejecutan en la CPU de un ordenador personal y se encargan del procesado en banda base de la señal muestreada por los dispositivos anteriormente mencionados.

GNURadio ha sido desarrollada en C++ por ser el lenguaje más adecuado para desarrollar aplicaciones cuyo rendimiento es crítico como este caso. Sin embargo, para facilitar su uso, cuenta con un *front-end* escrito en Python.

La metodología de trabajo con esta biblioteca se ve muy simplificada por su entorno de trabajo GNURadio Companion, el cual genera el código necesario para ejecutar la aplicación desarrollada a partir de conectar gráficamente distintos bloques. Estos bloques son de tres tipos: fuente, de proceso y salida, siendo el esquema básico de funcionamiento de cualquier aplicación el mostrado en la Figura 3-8, que conecta una fuente a distintos bloques de proceso para volcar finalmente el resultado en la salida que puede ser de transmisión, audio, gráfica o una variable.

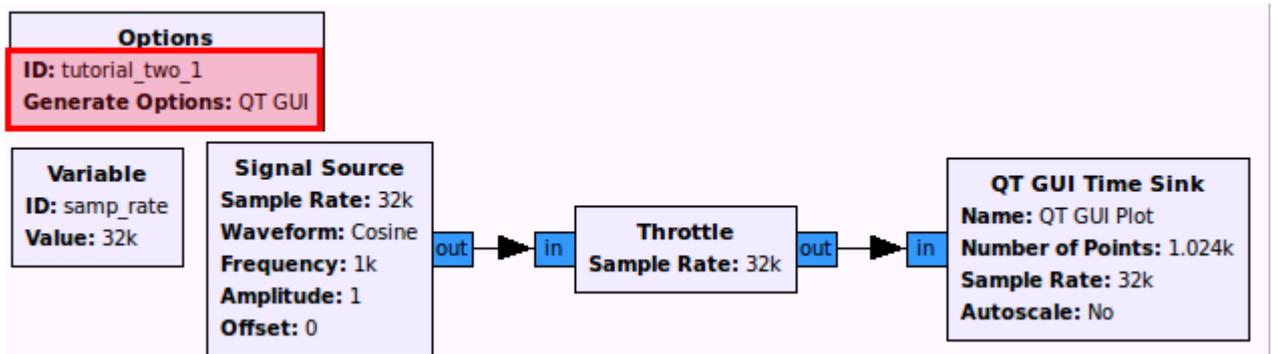


Figura 3-8 Simple aplicación desarrollada con GNURadioCompanion [56]

Se ha escogido esta biblioteca por ser prácticamente la única opción disponible para un usuario no profesional, además de ser realmente potente y contar con una comunidad de desarrolladores muy activa.

3.4 Plataforma de vuelo

3.4.1 El problema del vuelo

El objetivo del módulo de vuelo del sistema desarrollado es realizar un vuelo circular alrededor de la antena bajo prueba. Este módulo debe permitir al operador seleccionar de una manera sencilla la posición de la antena, así como la altura y el radio del círculo descrito. Para la realización de un vuelo exitoso existe un parámetro crítico: la autonomía del dron. Por otro lado, existen una serie de parámetros considerados de diseño que tienen que ver con el algoritmo de vuelo utilizado. Algunos de estos son el modo de vuelo y la velocidad a la que se ejecuta el vuelo.

3.4.2 Solución propuesta

Teniendo en cuenta las condiciones que afectan a la resolución de este problema descrito anteriormente, se procederá a exponer las decisiones tomadas relativas al diseño de este módulo del sistema:

Según el trabajo de fin de grado de diseño de este UAV [17]. La autonomía teórica son $T_{\min} = 5 \text{ min}$ a una $V_{\max} = 44 \text{ km/h}$ lo cual proporciona una autonomía de 3.600 m. Sin embargo, debido al desgaste sufrido por las baterías desde que se realizó dicho cálculo teórico, a las variaciones en la carga del dron, y al mayor consumo debido al equipo de medida, estos valores no se corresponden con la realidad. Se realizó una prueba de autonomía con un vuelo de radio de 40 m de una duración estimada de 3 minutos y medio. La autonomía estimada en esta prueba es de unos 5 minutos, lo que dejaría un margen de un minuto y medio para realizar el desplazamiento del UAV al buque.

Respecto al algoritmo de medida utilizado, se decidió aproximar la trayectoria circular por un polígono de 36 lados (Figura 3-9). Este número de lados se comprobó experimentalmente como idóneo, ya que menos lados creaban una trayectoria demasiado poligonal, y un número mayor de lados hacía que los *waypoints* de la trayectoria estuvieran demasiado pegados los unos a los otros lo que hacía que el UAV realizara un vuelo con un perfil de velocidad poco uniforme, acelerando y parando repentinamente cuando alcanzaba el siguiente punto.

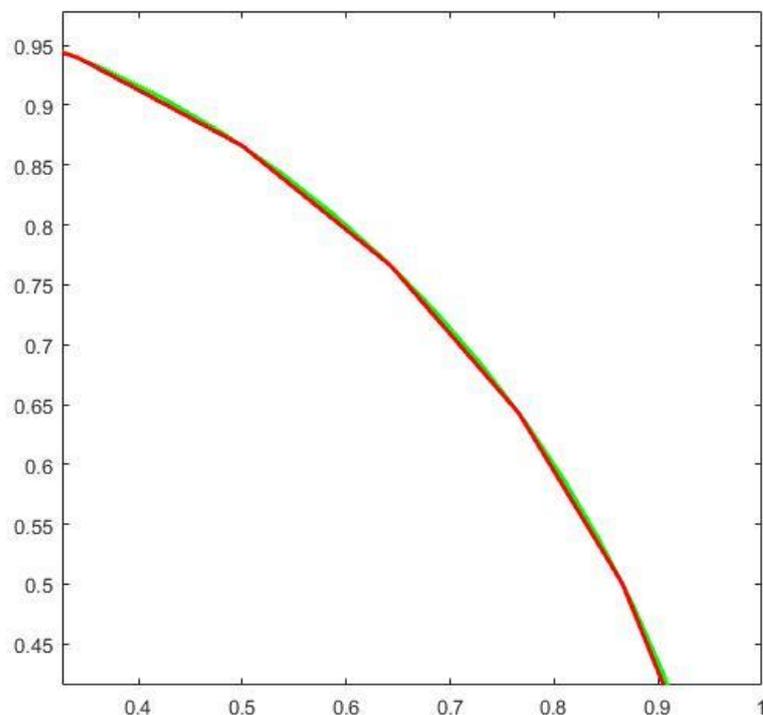


Figura 3-9 Detalle de aproximación de círculo (verde) por polígono de 36 lados (rojo) en Matlab

Para realizar esta trayectoria, hay que escoger entre hacerla en el modo automático (AUTO) o guiado (GUIDED). El primero requiere cargar todos los puntos de la trayectoria de antemano en la memoria del dron y después se da la orden de recorrerlos. Por el contrario, el segundo modo es mucho más flexible y permite dar una orden de ir al punto siguiente una vez se ha alcanzado el punto anterior.

El modo escogido es el guiado. Si bien ambos habrían resultado válidos para este proyecto, se escogió este de cara al futuro. Ya que al permitir cambiar los *waypoints* ordenados a los que tiene que dirigirse el dron, la trayectoria de vuelo se puede adaptar a una antena móvil, por ejemplo la de un barco en alta mar que siempre se ve afectado por el viento y las olas aunque tenga las máquinas paradas.

Por último, en la orden enviada al dron de dirigirse a un punto, se especifica la velocidad requerida. En esta trayectoria realizada, ya que los puntos se encuentran bastante próximos, esta velocidad nunca se alcanza, sin embargo es una manera de medir la aceleración que se le pide a los motores del UAV. Para evitar un cabeceo excesivo, que provocaría una inclinación de la antena de recepción que redundaría en pérdidas por desacoplo de la polarización con la ABP, la velocidad ordenada debe ser baja, en este caso, el valor de 2 m/s obtuvo buenos resultados experimentales.

Una vez explicadas las decisiones tomadas, se procede a explicar el diseño del módulo realizado representado en la Figura 3-10:

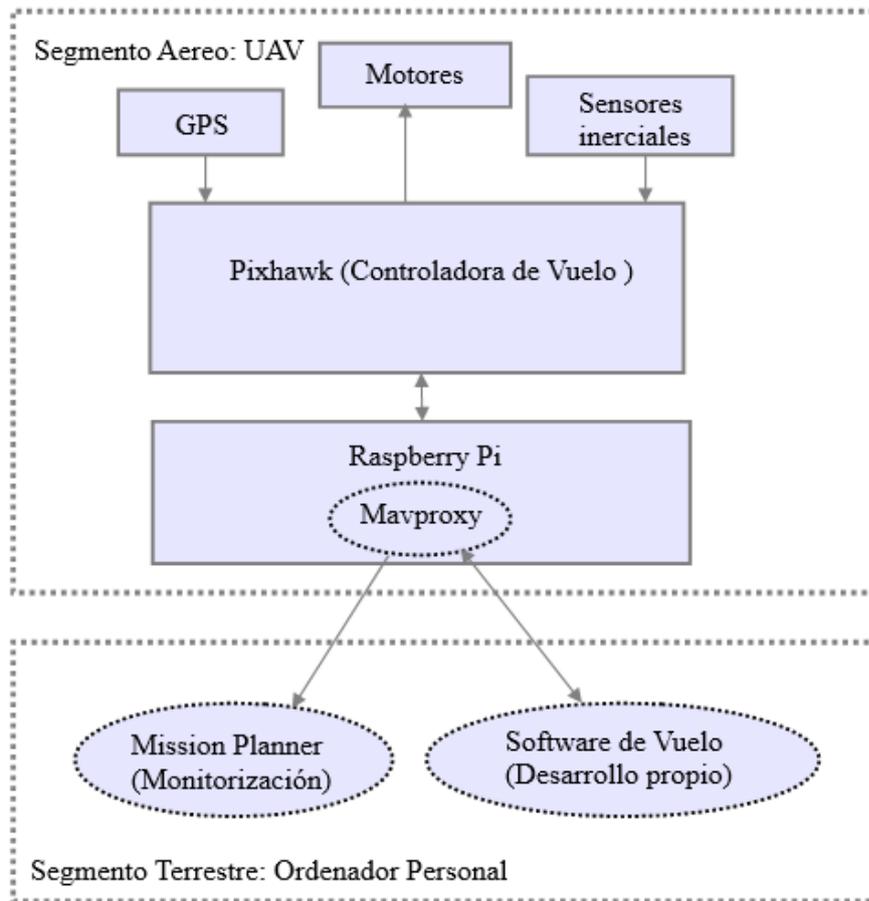


Figura 3-10 Esquema sistema de vuelo

El elemento clave del segmento aéreo es la controladora de vuelo. Este dispositivo es el encargado de recibir las órdenes de vuelo y convertirlas en órdenes para los motores en función de los datos que recibe a través del GPS y los sensores inerciales de que dispone. Las órdenes de vuelo y los datos de telemetría (posición, carga de la batería, estado general, etc.) constituyen un flujo bidireccional de mensajes del protocolo de comunicación MAVLink entre la controladora y el *software* en el segmento terrestre. Estos mensajes pasan a través de MAVProxy que es una pasarela ejecutada en la Raspberry Pi y que se encarga de encaminar los mensajes provenientes de diversas fuentes desde y hacia la controladora. El uso de esta pasarela es imprescindible pues permite realizar conexiones entre la controladora y más de un *software* de manera simultánea.

En el segmento terrestre, se ejecutan dos aplicaciones. Mission Planner y el *software* de vuelo desarrollado.

La GCS Mission Planner realmente no es imprescindible pues se utiliza únicamente para la monitorización y representación gráfica de la posición del UAV.

El *software* de vuelo desarrollado utiliza los datos seleccionados por el usuario de radio de vuelo, posición de la antena y altura de vuelo; para calcular los puntos de la trayectoria que debe seguir el dron y enviárselos como órdenes MAVLink.

3.4.3 Plataforma aérea

La plataforma aérea utilizada se basa íntegramente en el TFG de la referencia [17]. La plataforma es de tipo hexacóptero y consta de:

1. Chasis prediseñado DJI F550.
2. Controladora de vuelo Pixhawk.

3. Módulo compás+GPS para Pixhawk.
4. Receptor 8ch EzUHF.
5. Punto de acceso Wifi Ubiquity Picostation M2 HP.
6. Batería Desire Power V8.
7. 6 motores 2312E de 960 KWW
8. 6 controladores electrónicos de velocidad (ESC).
9. Raspberry Pi B 2

Para este proyecto se añadieron los elementos reseñados a continuación (Figura 3-11):

1. Antena omnidireccional receptora
2. Receptor RTL SDR NooElec



Figura 3-11 Resultado final plataforma de vuelo .

4 DESARROLLO DEL SISTEMA

4.1 Introducción

Una vez planteado el diseño teórico del sistema, los problemas que se plantean en cada uno de los módulos y la solución propuesta; solo resta implementar esta solución.

En este capítulo se expondrá el proceso de desarrollo del sistema, haciendo especial hincapié en el *software* desarrollado para cada módulo y en los problemas surgidos junto con su solución.

4.2 Diseño del *software* del sistema

El *software* del sistema desarrollado ha sido escrito en Python debido a que este es el lenguaje en el que está desarrollada la biblioteca DroneKit utilizada para transmitir los mensajes MAVLink al UAV, y también es el lenguaje que genera el programa GNURadio Companion.

De acuerdo al diseño modular de este sistema, el *software* también está diseñado de esa manera como se representa en la Figura 4-1.

Como parte del módulo de medida de potencia, el programa `medida.py` es el encargado de recibir las muestras que envía la RPi a través de `rtl_tcp` y se encarga de procesarlas para calcular la potencia medida con la tasa seleccionada como parámetro de diseño. A su vez, este programa también actúa como un cliente de TCP/IP que se conecta a un servidor, explicado a continuación, en el mismo ordenador y le envía constantemente la potencia calculada.

Como parte del módulo de vuelo, el programa `vuelo.py` es el encargado de ejecutar el vuelo además de actuar como elemento integrador de los dos módulos pues a su vez actúa como servidor recibiendo las medidas del cliente y escribiéndolas en un fichero junto a la demora en la que fueron tomadas. Este programa se ha diseñado para funcionar en tres hilos. El primer hilo es simplemente el servidor encargado de recibir las medidas. El segundo hilo es el encargado de dirigir al UAV en su vuelo enviando y recibiendo los mensajes MAVLink. El hilo principal actúa como integrador recibiendo las medidas del primer hilo y correlacionándolas con la posición GPS en la que fueron tomadas a partir de la cual se calcula la demora. Estos pares son escritos en un fichero de resultados. El diseño basado en hilos permite mantener la modularidad del sistema, pudiendo ejecutar el vuelo incluso aunque el servidor no reciba la petición de conexión de ningún cliente.

Finalmente, el fichero de resultados, es interpretado por un código de Matlab para obtener la representación del diagrama de cobertura.

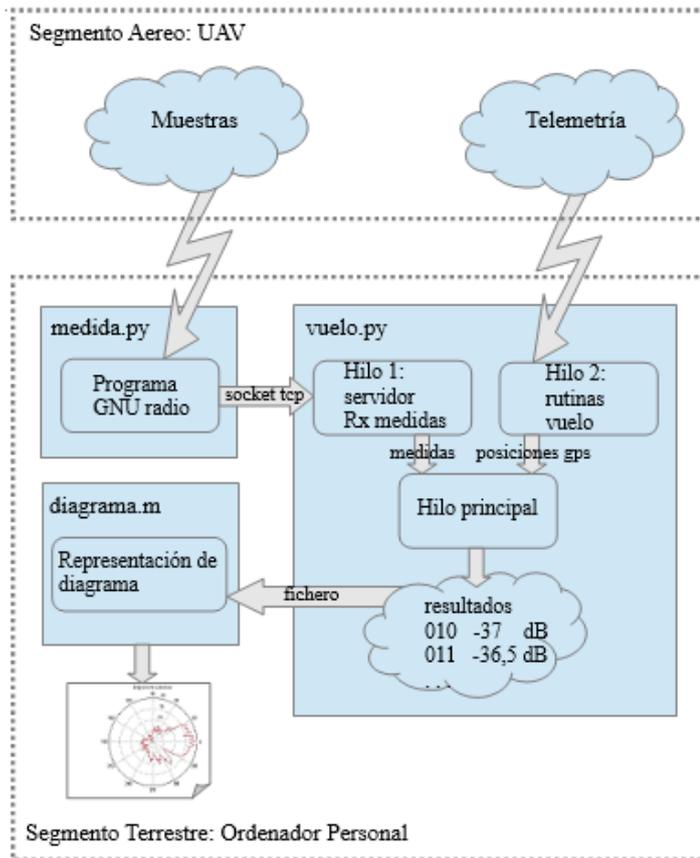


Figura 4-1 Arquitectura del software desarrollado

4.3 La medida de la potencia

Como se ha explicado, el desarrollo de esta parte consistió en un programa desarrollado con GNURadio Companion y un cliente que envía los datos obtenidos al servidor que se ejecuta en la parte de vuelo.

4.3.1 El programa de GNURadio

Tras un proceso de aprendizaje sobre procesamiento digital de señales básico, representación I y Q de la señal y funcionamiento básico de GNURadio; se tomaron las decisiones de diseño explicadas en el apartado 3.3.2 y se creó el programa de la Figura 4-2 en GNURadio.

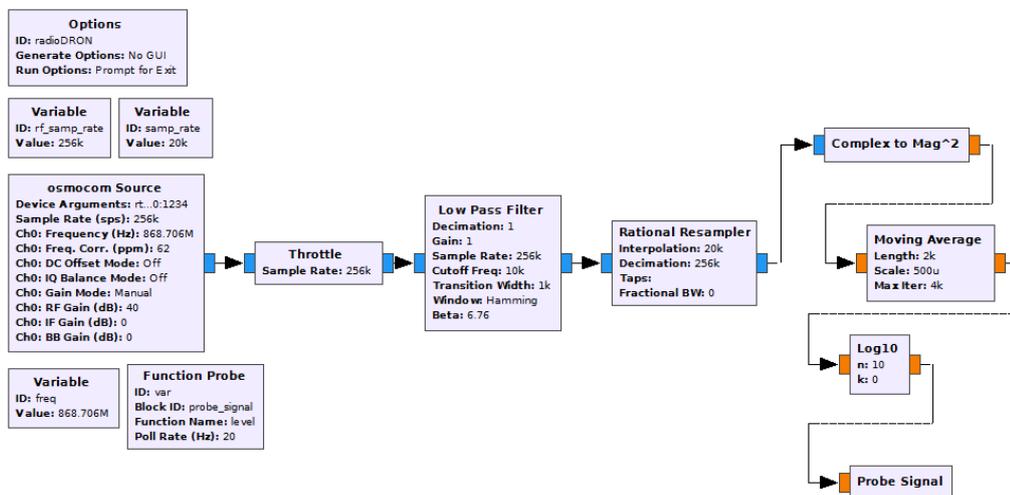


Figura 4-2 Programa en GNURadio

El diagrama de bloques, como todo programa GNURadio consta de tres partes principales: la fuente, bloques de proceso y la salida.

La fuente es una fuente diseñada para trabajar con chips RTL a la que se le especifica una dirección IP en la que el programa `rtl_tcp` está enviando las muestras. A su vez se ajusta la frecuencia central de escucha y la frecuencia de muestreo inicial, fijada en 256 KHz por ser la mínima que admiten estos chips. También es importante ajustar la corrección en frecuencia, medida en partes por millón que depende de cada receptor SDR individual y representa lo que se desvía cada aparato cuando sintoniza una frecuencia central. Una manera de calcularla es con un *software* como SDR#, sintonizando una emisora de FM, por ejemplo, cuya frecuencia central sea conocida y con el *zoom* ajustar la corrección para que se sintonice correctamente. Por último, es reseñable el color azul de la salida que indica que es compleja ($I + jQ$).

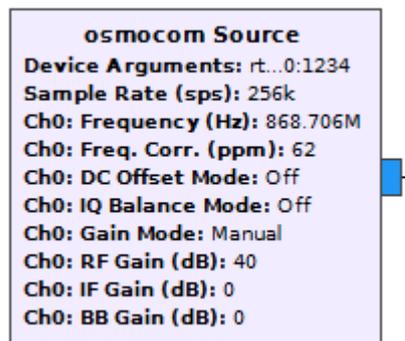


Figura 4-3 Bloque fuente.

Los bloques de proceso conforman la parte más importante de este programa. La primera parte de estos bloques (Figura 4-4) consiste en un bloque `throttle` que es una medida de seguridad ya que limita la frecuencia de muestreo procesada en caso de que el programa esté consumiendo demasiados recursos de la CPU. A continuación, se ejecuta un filtro de paso bajo en banda base que solo permite pasar las frecuencias menores a 10 KHz, el ancho de banda de la señal, pero con un margen de transición de 1 KHz para incluir posibles desviaciones en el ancho de banda de la señal. Por último, un bloque remuestreador que consiste en un diezmador junto con un interpolador para cambiar la frecuencia de muestreo a 20 KHz que, como se ha visto en el apartado 3.3.2 es la mínima necesaria.



Figura 4-4 Primera parte de los bloques de proceso

La segunda parte de los bloques de proceso (Figura 4-5) es la que ejecuta la medida propiamente dicha. Esta parte consta de un primer bloque que calcula $I^2 + Q^2$ obteniendo una salida de número real de color naranja. El segundo bloque hace una media móvil de las 2.000 muestras anteriores: este bloque obtiene resultados con la misma tasa que la frecuencia de muestreo seleccionada, sin embargo en cada muestra, devuelve la media móvil del número especificado de muestras anteriores, debe ser en otra parte del programa en la que se especifique que se quiere obtener un valor con la tasa determinada

(10 medidas/s). El tercer bloque simplemente obtiene el logaritmo en base 10 del valor medido y lo multiplica por 10.

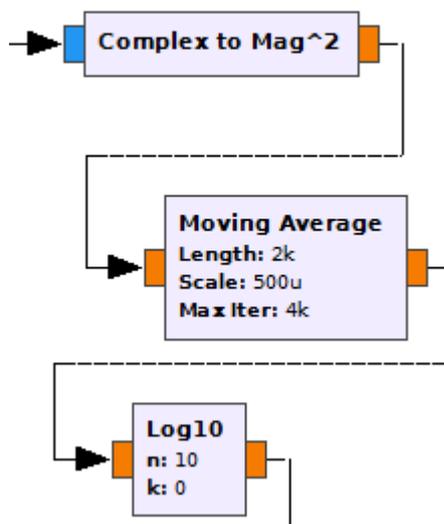


Figura 4-5 Segunda parte de los bloques de proceso

La salida (Figura 4-6) es lo que en GNURadio se denomina `probe`, que no es más que una salida que se guarda en una variable miembro de la clase generada por el código del programa. Esta variable se actualiza con la tasa que se le determine, siendo aquí donde se especifica que se actualice 10 veces/s.



Figura 4-6 Bloque de salida

Finalmente, el código obtenido se traduce a una clase que hereda de la clase de GNURadio `top_block` y que cuenta con una serie de métodos como `run()` y `stop()` necesarios para controlar la ejecución de este programa.

4.3.2 Arquitectura cliente-servidor

Una vez creado un programa en GNURadio que hace la función requerida de medir la potencia, el problema es acceder a ese valor para combinarlo con la posición del UAV en cada momento.

Como se ha explicado antes, el código generado consiste en una clase que guarda el valor de la medida en una variable miembro.

Lo más lógico sería que dentro del programa principal en el que se ejecuta el vuelo, en un hilo aparte se creara un objeto de esta clase y se ejecutara su método `run()`, accediendo a la variable de una manera muy sencilla. Sin embargo esto no fue posible porque las librerías en las que se apoya GNURadio son para arquitectura de 64 *bits* y las de DroneKit para arquitectura de 32 *bits*, no pudiendo ser combinadas en un mismo ejecutable.

La solución seleccionada finalmente fue crear un programa de Python que instancie un objeto de la clase creada por GNURadio y acceda a su variable enviándola a través de un cliente de TCP/IP a un servidor en el *localhost* cuyo puerto de escucha es conocido.

El paquete de instalación de GNURadio incluye su propio intérprete de Python de 64 *bits* así como todas las bibliotecas necesarias, por lo tanto, para ejecutar programas escritos en Python que utilicen partes de GNURadio, deben ejecutarse en lo que podríamos llamar el entorno GNURadio. Dicho entorno consiste en una consola normal de Windows cuyas variables de entorno dirigen hacia el intérprete antes mencionado y las bibliotecas necesarias. Esta consola se abre con el ejecutable GNURadio Command Prompt.

Una vez abierto este entorno, solamente hay que acceder a la carpeta en la que se encuentra el código y ejecutarlo con el siguiente comando:

```
> python medida.py
```

4.4 La plataforma de vuelo

Como se ha explicado, el programa `vuelo.py`, ejecuta tres hilos: un servidor que recibe las medidas, el hilo que controla el vuelo y el hilo principal que escribe en un fichero de resultados los pares medida-demora.

4.4.1 Hilo servidor receptor de medidas

En esta parte, se ha creado la clase `Servidor` que hereda de la clase de la biblioteca básica de Python `threading.Thread`. El método más importante de esta clase es el método `run()` que es un método de la clase madre y que está diseñado para ser reescrito por las clases hijas para implementar su funcionalidad específica. En este caso, cuando un objeto de la clase `Servidor` es instanciado y se ejecuta su método `run()`, se inicia un servidor TCP/IP en un hilo aparte del hilo principal que escucha en una dirección IP y puerto indicados y que recibe en un bucle la medida de potencia por parte del cliente, guardándola en una variable que será accedida por el propio hilo principal para obtener sus resultados.

4.4.2 Hilo de control de vuelo

En esta parte, se ha creado la clase `Multicoptero` que hereda de la clase de `DroneKit Vehicle`. Dicha clase contiene todas las funciones para ejecutar la conexión con la controladora de vuelo, y funciones de alto nivel para transmitir órdenes como `simple_goto()` que, a un nivel más bajo, envía mensajes MAVLink con órdenes, así como funciones de alto nivel para leer los datos recibidos por telemetría. En la clase creada, se han implementado métodos que ejecutan rutinas de vuelo efectuando comprobaciones diversas, estos métodos son:

1. `despegue(altura)`, que efectúa comprobaciones previas al armado del vehículo, lo arma y despega hasta la altura pasada como argumento.
2. `ir_hacia(posición)`, que recibe como parámetro la posición a la que debe dirigirse y se dirige en modo guiado hacia ella, realizando comprobaciones cada cierto tiempo de que llega. La precisión seleccionada para decidir que el UAV está en el punto es de 2 m, ya que esta es la precisión del módulo GPS instalado [17] y se ha comprobado en la práctica que es una buena medida, ya que si se intenta ser más preciso muchas veces el UAV se queda volando estático prácticamente en la misma posición hasta que su señal de GPS determina que ha llegado a la posición. Dentro de este método es reseñable que una vez se envía la orden de MAVLink, se comprueba que el dron envía el mensaje de vuelta «COMMAND_ACK». En caso de que este mensaje no llegue de vuelta, el código del programa vuelve a enviar la orden, si el dron vuelve a no responder, el programa termina enviándole a la Pixhawk la orden de cambio de modo a *Return to Launch* (RTL), y cerrando la conexión. Esta comprobación fue necesario incluirla pues en el simulador (SITL) utilizado para el desarrollo, había veces que el UAV no respondía a las órdenes y se comprobó que esto era debido a que la orden en sí no llegaba, con el método ofrecido se solucionó el problema y la gran mayoría de las veces al enviar la orden la segunda vez el

vehículo respondía adecuadamente. Este problema nunca ocurrió en las pruebas de vuelo con el UAV real.

3. `vuelo_aproximacion(pos_antena, radio, altura)`, este método está diseñado para que el dron se dirija hacia la posición más favorable para iniciar el vuelo circular. Cuando este método se ejecuta, el dron se dirige directamente hacia la antena, parándose a la distancia de radio seleccionado, que a la postre será el primer punto del vuelo circular.
4. `vuelo_circular(centro, radio, altura, precision, velocidad, demora_inicial, sentido)`, este método ejecuta un bucle de sucesivos `ir_hacia(posicion)`. Los parámetros que se envían son las coordenadas geográficas de la posición de la antena (`centro`), el radio y la altura del vuelo; la precisión que indica el número de lados del polígono por el que se aproxima el círculo (precisión=10° equivale a un polígono de 36 lados); la velocidad a la que se realiza el vuelo, que como indica el apartado 3.4.2 determina la aceleración que la controladora le imprime al UAV; la demora inicial respecto al centro de la antena en la que se iniciará el vuelo; y el sentido de giro, horario (`sentido=1`) o antihorario (`sentido=-1`).
5. Otros métodos auxiliares autoexplicados en su nombre como `cambiar_altura(altura, velocidad)` y `fin()` que cierra las conexiones.

Además de esta clase, se ha creado un módulo Python `auxiliar.py` que contiene funciones auxiliares para cálculos de demoras entre dos posiciones dadas en coordenadas geográficas, u obtención de las coordenadas de un punto dada demora y distancia desde otro. Estas funciones son fundamentales para la ejecución de los métodos de la clase `Multicoptero` y, en su mayor parte, han sido adaptadas de código de ejemplo del proyecto DroneKit.

4.4.3 Hilo principal

Finalmente, el hilo principal que se ejecuta al lanzar la aplicación `vuelo.py`, tiene varias funciones. En primer lugar determina los parámetros de vuelo como radio de vuelo, centro de la antena y altura. Estos parámetros pueden ser pasados como argumentos al lanzar la aplicación, o si no se hace así, son preguntados directamente. También, este programa pregunta al usuario si quiere ejecutarlo en el UAV real o el simulador SITL utilizado, lo que facilita mucho el proceso de realizar pruebas y desarrollar el *software*. A continuación, este hilo crea una instancia de la clase `Multicoptero` e inicia las conexiones. Después, inicia el servidor en un hilo aparte. Una vez todo esto está iniciado, se envía al UAV al primer punto donde iniciará la trayectoria circular. Al llegar a ese punto, el vuelo circular se ejecuta en un hilo aparte, y el hilo principal se encarga de escribir en un fichero la variable medida recibida por el servidor junto a la demora en la que se mide calculada a partir de la posición GPS recibida,

Finalizado el vuelo circular, se cierra la conexión del servidor, la conexión del UAV y se cierra el fichero, dando por finalizado el programa.

4.5 Interpretación de los resultados

El fichero obtenido es un fichero cuyo contenido en texto plano consiste en líneas de la forma `<<'medida' 'demora'\n>>`.

Se ha creado un *script* de Matlab que interpreta el fichero y representa el diagrama en forma de diagrama polar.

Como se ha dicho anteriormente, este fichero y el código del programa se puede adaptar fácilmente para crear ficheros que puedan ser interpretados por el *software* utilizado en el CEMEDM, que, tras unas cabeceras con datos, continen líneas de la forma `<<'hora' 'medida' 'demora'\n>>`.

5 PRUEBAS Y VALIDACIÓN DEL SISTEMA

5.1 Prueba de vuelo

La primera prueba realizada del sistema fue para comprobar el funcionamiento del módulo de vuelo, por lo que no se realizó ningún tipo de medida. En esta prueba se realizó una serie de vuelos circulares.

Entorno de la prueba	
Fecha	08 Febrero 2017
Lugar	Pista de vuelo al norte de Pastoriza. 42°20'30'' N, 008°43'27''W
Condiciones meteorológicas	No precipitaciones. Viento flojito ($< 11 \frac{km}{h}$)

Tabla 5-1 Datos de entorno de la prueba de vuelo

5.1.1 Desarrollo de las pruebas

Se realizaron cinco vuelos variando el número de lados del polígono descrito así como el radio de vuelo, el centro y el sentido de giro. Todos los vuelos se realizaron a una altura de 10 m sobre el suelo y a una velocidad ordenada de 8 m/s. Las imágenes de las trayectorias son tomadas de la GCS Mission Planner utilizada para monitorización.

5.1.1.1 Primer vuelo

Características	
Radio	30 m
Número de lados polígono	18
Sentido de giro	Horario

Tabla 5-2 Primer vuelo

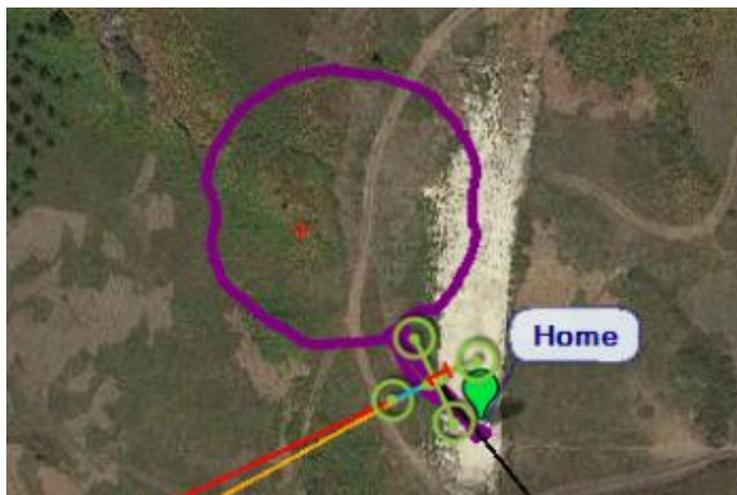


Figura 5-1 Trayectoria primer vuelo

5.1.1.2 Segundo vuelo

Características	
Radio	30 m
Número de lados polígono	36
Sentido de giro	Antihorario

Tabla 5-3 Segundo vuelo



Figura 5-2 Trayectoria segundo vuelo

5.1.1.3 Tercer vuelo

Características	
Radio	30 m
Número de lados polígono	72
Sentido de giro	Horario

Tabla 5-4 Tercer vuelo



Figura 5-3 Trayectoria tercer vuelo

5.1.1.4 Cuarto vuelo

Características	
Radio	40 m
Número de lados polígono	72
Sentido de giro	Horario
Observaciones	<p>Este vuelo fue cronometrado.</p> <p>En el <i>waypoint</i> 26 se cambió al modo RTL desde Mission Planner por batería baja finalizándose aquí la prueba.</p> <p>Después de este vuelo se hizo un estudio del consumo de la batería revisando los <i>logs</i>.</p> <p>En ese punto el vuelo había durado 2 min y 30 s.</p>

Tabla 5-5 Cuarto vuelo

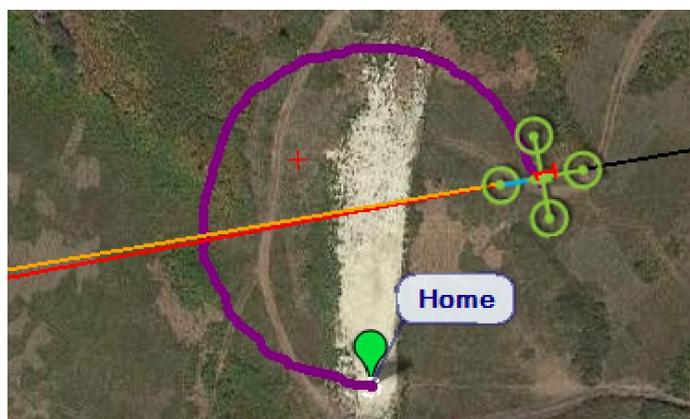


Figura 5-4 Trayectoria del cuarto vuelo hasta el momento que se manda RTL

5.1.1.5 Quinto vuelo

Características	
Radio	70 m
Número de lados polígono	36
Sentido de giro	Horario
Observaciones	En este vuelo surgió el problema de que había un árbol en la trayectoria que no estaba previsto. Se intentó tomar el control del UAV por medio de la emisora pero no se pudo resultando en un golpe contra el árbol y una caída que resultó en una rotura de una pata.

Tabla 5-6 Quinto vuelo

No se grabó imagen de la trayectoria de vuelo.

5.1.2 Conclusiones

La primera conclusión que se obtuvo de esta prueba es que las rutinas de vuelo que habían sido previamente probadas en el simulador SITL también funcionaron con el UAV real.

La segunda conclusión que se obtuvo es que con la velocidad marcada de 8 m/s el UAV cabeceaba demasiado porque al ser ordenado ir a un punto aceleraba mucho y al llegar frenaba bruscamente, al estar tan pegados los puntos unos de otros el efecto no era bueno. En esta prueba no se cambió la velocidad en ningún momento, sin embargo, se decidió que esto no sería bueno a la hora de realizar medidas pues se produciría desacoplo en la polarización y en futuras pruebas se bajaría esa velocidad a 2 m/s.

La tercera conclusión obtenida fue el tiempo de vuelo en la tercera prueba. Con esta medida, tomando 40 m como un radio posible de medida para buques tipo patrullero por ejemplo, se estimó un tiempo de vuelo total en dar una vuelta de 3 minutos y medio. Para cumplir el criterio de diseño de 3 medidas por cada grado (1.080 en total), hay que realizar algo más de 5 mediciones por segundo. Es por esto por lo que se decidió hacer 10, para asegurar el cumplimiento de este criterio incluso con vuelos de menor radio.

La última conclusión fue que había algo que no funcionaba del todo correctamente. Cuando en el quinto vuelo se produjo la caída, momentos antes se intentó tomar el control de vuelo manual desde la emisora que según [17] debería ser prioritario frente a las órdenes recibidas a través de MAVProxy, sin embargo el UAV no respondió como debería. El fallo podría estar bien porque se intentó hacer demasiado tarde y el golpe ya se había producido; o bien por alguna razón no llegaron las órdenes desde la emisora, a pesar de que antes de los vuelos en círculo había sido probada y funcionaba correctamente; o que realmente las órdenes de la emisora no son prioritarias, y aunque lleguen, la siguiente orden enviada por la rutina de vuelo provocó un comportamiento errático del UAV.

5.1.3 Estudio de autonomía

Una de las conclusiones más importantes de esta prueba fue un estudio realizado de la posible autonomía del UAV realizado en el cuarto vuelo.

Los datos de autonomía calculados de manera teórica en el trabajo de fin de grado en el que se diseñó este UAV [17] son:

Datos de autonomía	
Tiempo mínimo de vuelo	5 minutos
Tiempo de vuelo mixto	10 minutos
Tiempo de vuelo estático	13 minutos
Velocidad máxima	44 km/h

Tabla 5-7 Datos de autonomía de diseño del UAV [17]

Estos datos nunca fueron probados experimentalmente, además están calculados sin tener en cuenta el estado real de las baterías un año después.

Por lo tanto, esta prueba se aprovechó para hacer una pequeña aproximación de lo que podría ser la autonomía del dron.

La curva de funcionamiento de este tipo de baterías es bien conocida (Figura 5-5). Tras una primera caída más pronunciada, el voltaje de cada celda baja de forma prácticamente lineal hasta llegar aproximadamente a los 3,3 V, punto a partir del cual el voltaje desciende en picado.

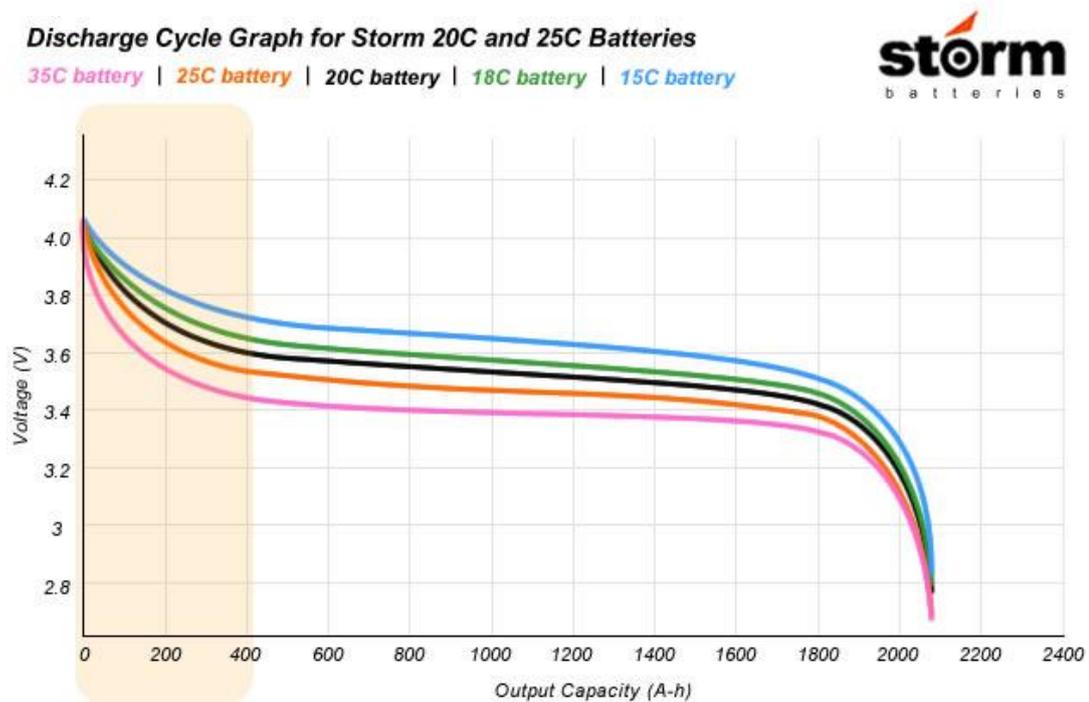


Figura 5-5 Curva de descarga baterías LiPo fabricante Storm Batteries [57]

La batería montada es de 3 celdas (12,6 V a máxima carga), 6000 mAh y una tasa de descarga 30C. Las baterías LiPo no deben bajar de un voltaje de 3 V por celda (9 V en este caso) pues quedan inutilizadas irreversiblemente, por razones de seguridad, en este proyecto se decidió que no bajen de 3,3 V (9,9 V en total).

En la prueba realizada, en la revisión del log de telemetría posterior, se puede comprobar cómo en un primer momento la batería tenía un voltaje de 12 V (no completamente cargada) que se reduce a 10,7 V en el momento del despegue debido a la resistencia interna de la batería.

A partir de ese momento, la diferencia de potencial desciende de una manera prácticamente lineal como cabía esperar, hasta llegar a 10 V en el momento que se termina el vuelo de 2 minutos y medio de duración.

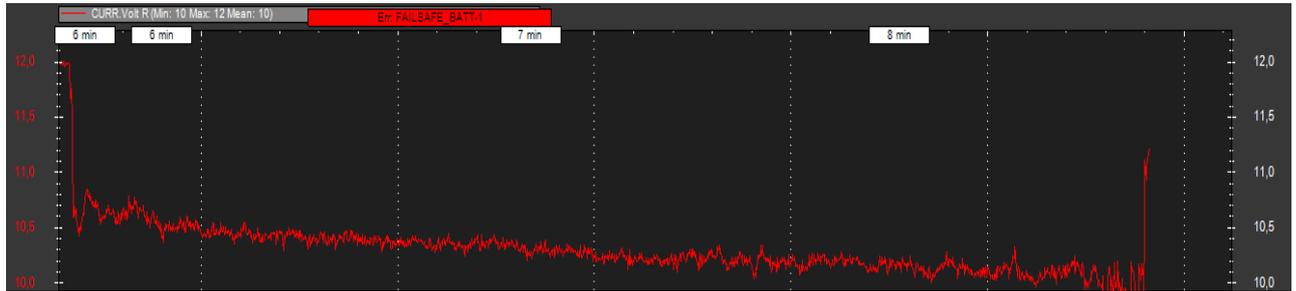


Figura 5-6 Log de telemetría de consumo de batería (tiempo-voltaje)

El estudio tuvo lugar en el cuarto vuelo en el cual realizó un círculo de 40 metros de radio y se alcanzaron 26 *waypoints* de los 36 del círculo completo. En base a estos datos, el tiempo esperado de vuelo habría sido 3 minutos y medio si hubiera realizado todo el círculo. Con la tasa de descarga medida de 0,7 V en 2 minutos y medio, se habría reducido el voltaje aproximadamente 1 V en 3 minutos y medio. Si la batería hubiera estado cargada completamente (12,6 V al inicio) y con la misma caída al despegar, que por otra parte es esperable por ser las mismas corriente consumida y resistencia interna de la batería, la batería habría mostrado una diferencia de potencial de 10,3 V. Hasta llegar a 9,9 V, el tiempo de vuelo sería de unos 5 minutos.

La conclusión de este estudio, que si bien necesita de muchas más pruebas, es que la autonomía del UAV es muy limitada con la batería utilizada actualmente. A falta de hacer cambios que puedan mejorar este aspecto (apartado 6.2), la obtención de diagramas de radiación está limitada a buques de pequeño porte que se sitúen a una distancia cercana a tierra.

5.2 Primera prueba de medición sin vuelo

La primera prueba de medición sin vuelo se realizó tan pronto como se consiguió terminar todo el *software* de medida e integrarlo con la parte de vuelo. Sin embargo para esta prueba se modificó para que el UAV no volara, pero si se utilizaron las posiciones GPS del mismo que fue llevado caminando a mano alrededor de la antena instalada en el centro de la pista de helicópteros en la pista militar de la ENM.

Entorno de la prueba	
Fecha	16 Febrero 2017
Lugar	Pista de helicóptero ENM. 42°23'48''N, 008°42'30''W
Condiciones meteorológicas	No precipitaciones.

Tabla 5-8 Datos del entorno en la primera prueba de medición sin vuelo

5.2.1 Montaje de la prueba

En el centro de la pista militar se colocó una antena Yagi de 11 elementos y una antena omnidireccional encima como se ve en la Figura 5-7:



Figura 5-7 Montaje de antena para realización de pruebas

En la Figura 5-8 se ve el detalle de cómo se insertó la antena en un agujero existente en la pista, además de la HackRF One utilizada para transmitir abajo a la derecha de la imagen.



Figura 5-8 Detalle de colocación de la antena junto a HackRF One

5.2.2 Desarrollo de las pruebas

A pesar del montaje de la antena omnidireccional, finalmente solo dio tiempo a realizar dos mediciones de la antena Yagi.

En esta prueba aún no se había decidido el criterio de diseño de 3 medidas por cada grado, por lo tanto se varió el número de muestras tomadas por segundo.

Se realizaron dos mediciones.

5.2.2.1 Primera medición

Características	
Radio	10 m
Medidas por segundo	2

Tabla 5-9 Primera medición 16/02/17

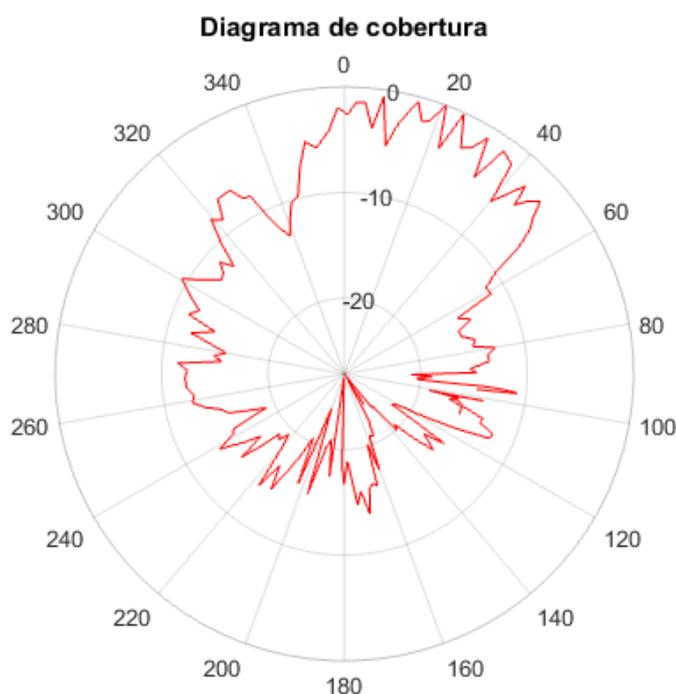


Figura 5-9 Diagrama de la primera medición 16/02/17

5.2.2.2 Segunda medición

Características	
Radio	10 m
Medidas por segundo	5

Tabla 5-10 Segunda medición 16/02/17

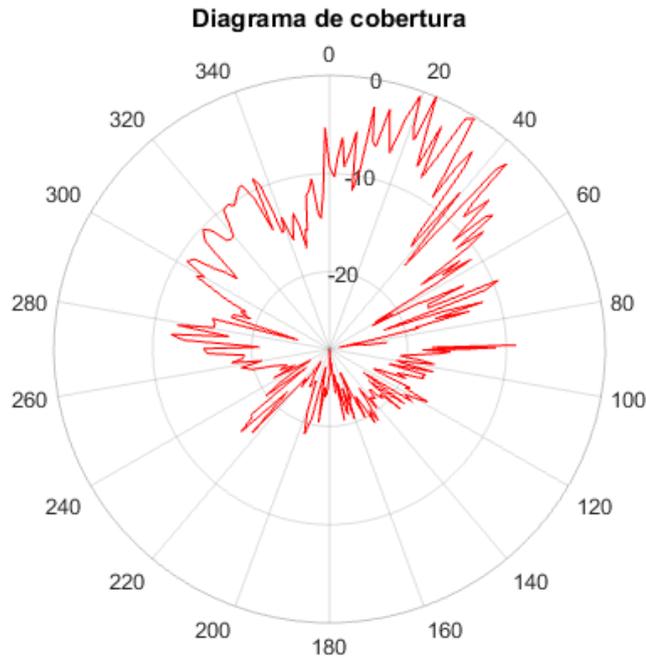


Figura 5-10 Diagrama de la segunda medición 16/02/17

5.2.3 Conclusiones

Para valorar esta prueba, se realizó una simulación del diagrama de radiación de esta antena con el software 4nec2 [58]. Este software es un entorno gráfico para la ejecución de los algoritmos NEC-2 que fue desarrollado en 1981 en Estados Unidos por el Lawrence Livermore National Laboratory, con el patrocinio del Naval Ocean Systems Center y el Air Force Weapons Laboratory [59] y está basado en el método de los momentos [60]. Los algoritmos de Nec2 (*Numeric Electromagnetic Code*) permiten estudiar las propiedades electromagnéticas de las antenas y otras estructuras metálicas.

De la simulación realizada para la antena Yagi de 11 elementos descrita, se obtiene el diagrama de radiación en acimut mostrado en la Figura 5-11.

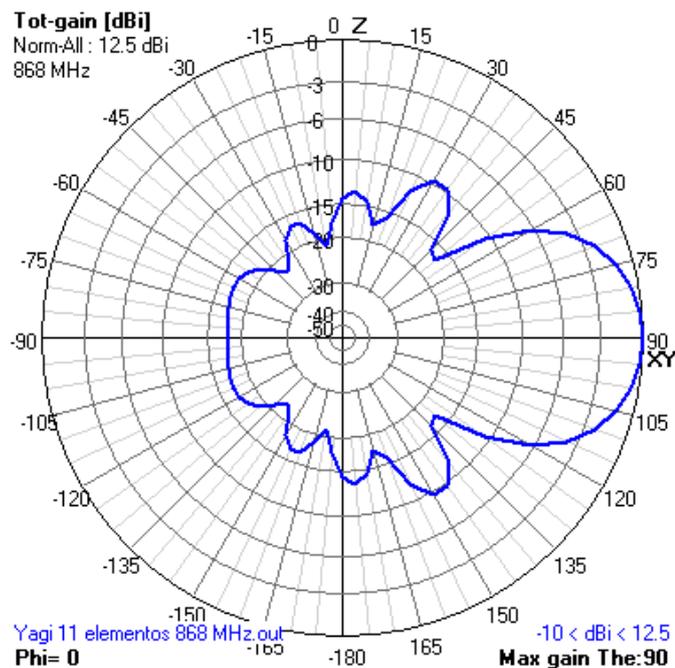


Figura 5-11 Simulación del diagrama de radiación de la antena Yagi de 11 elementos utilizada

Teniendo en cuenta las imprecisiones cometidas en la trayectoria al ir caminando, las interferencias causadas por la persona que tiene el UAV en la mano y que no se hizo ningún tipo de estudio del efecto del multitrayecto, esta primera prueba de funcionamiento del módulo de medidas fue bastante satisfactoria pues los resultados obtenidos se asemejan razonablemente a lo esperado.

Tomando la segunda medida como referencia, ya que se distinguen bastante mejor los lóbulos que en la primera, el ancho de haz del primer nulo es aproximadamente 70°, siendo este de 80° en la simulación. A su vez, la relación delante atrás de la medición es de 20 dB frente a los 18 dB de la simulación. Por su parte, la relación con el lóbulo secundario en la medición es de aproximadamente 10 dB igual a los 10 dB de la simulación.

Una conclusión obtenida es que aumentando la tasa de medidas se obtenían resultados más nítidos, lo que conllevó a tomar la decisión de establecer como requisito de diseño el tomar tres medidas por cada grado cubierto.

5.3 Segunda medición sin vuelo

Unos días más tarde de la primera medición, se realizó la segunda prueba aplicando los cambios en el algoritmo de medida para realizar 10 mediciones/s. El montaje realizado para esta medida fue exactamente el mismo que en la anterior prueba.

En este caso, se estudió el efecto del multitrayecto, determinando que con el modelo de tierra plana existe un nulo alrededor de 17 m, por lo que se realizaron las medidas más lejos, describiendo un círculo de 20 m de radio aproximadamente.

Entorno de la prueba	
Fecha	20 Febrero 2017
Lugar	Pista de helicóptero ENM. 42°23'48''N, 008°42'30''W
Condiciones meteorológicas	No precipitaciones.

Tabla 5-11 Datos del entorno en la segunda prueba de medición sin vuelo

5.3.1 Desarrollo de las pruebas

Se realizaron dos mediciones.

5.3.1.1 Primera medición

Características	
Radio	20 m
Medidas por segundo	10

Tabla 5-12 Primera medición 20/02/17

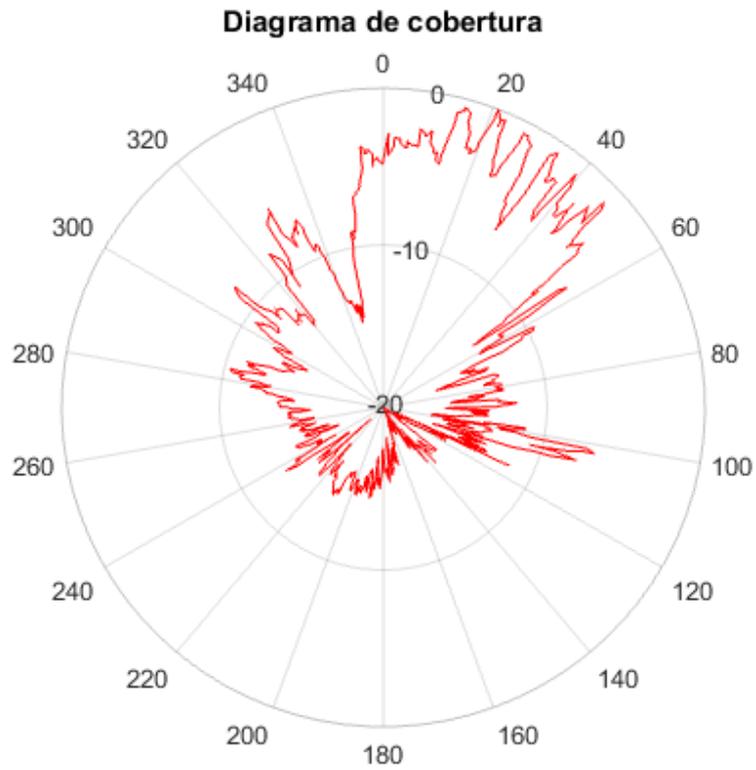


Figura 5-12 Diagrama de la primera medición 20/02/17

5.3.1.2 Segunda medición

Características	
Radio	20 m
Medidas por segundo	10

Tabla 5-13 Segunda medición 20/02/17

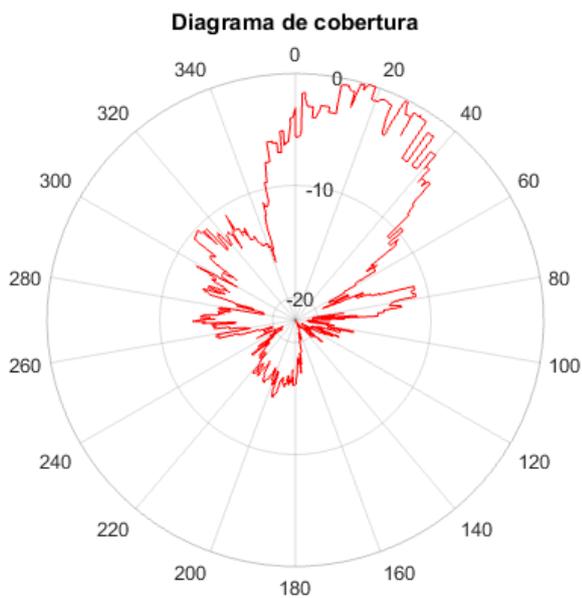


Figura 5-13 Diagrama de la segunda medición 20/02/17



Figura 5-15 Trayectoria seguida en la segunda medición 20/02/17

5.4 Prueba final

No se llegó a realizar una prueba combinada de vuelo y medición por diversas razones:

La disponibilidad para los vuelos fue muy mala porque no se pueden realizar en las instalaciones de la ENM y hay que buscar un horario que coincida al alumno y los dos tutores para ir a realizar el vuelo a la pista de vuelo en Pastoriza (Marín).

Las últimas semanas de desarrollo del trabajo el tiempo atmosférico fue en su mayor parte lluvioso lo que impidió el realizar cualquier tipo de prueba ya que el UAV no se encuentra estancado.

6 CONCLUSIONES Y LÍNEAS FUTURAS

6.1 Conclusiones

Partiendo del objetivo del TFG definido en el apartado 1.3 como: «Crear un sistema que presente una alternativa útil y fiable al actual método de obtención de diagramas de cobertura del CEMEDEM», teniendo en cuenta que: «El sistema desarrollado se considerará útil si permite su empleo de manera intuitiva sin la necesidad de conocer sus detalles de implementación. De la misma manera, se considerará fiable si el sistema obtiene unos resultados dentro de unos márgenes de incertidumbre aceptables de 2,54 dB al igual que el CEMEDEM (apartado 2.2.3), siendo comprobado esto último con una antena cuyo diagrama sea bien conocido» (mismo apartado), y contando con que si bien no se ha podido realizar una prueba completa del sistema (apartado 5.4), se han realizado pruebas de los dos módulos por separado, obteniéndose en ambas resultados positivos. Se obtienen las conclusiones siguientes:

Respecto al módulo de vuelo se obtiene la conclusión de que las rutinas desarrolladas funcionan adecuadamente pues se consiguió realizar varios vuelos circulares con éxito. Sin embargo, la estimación que se tiene de la autonomía del UAV determina que esta es un factor limitante y debería ser convenientemente mejorada (apartado 6.2) para ampliar la gama de buques y distancias a las que se puede realizar la medida.

Respecto al módulo de medida, la caracterización de la antena en tierra realizada obtuvo muy buenos resultados dentro de la incertidumbre considerada de 2,54 dB. Aun así, es necesario realizar más pruebas situando la antena en ubicaciones distintas para tratar de descubrir el origen de los nulos que aparecen en los diagramas obtenidos tal y como se indica en el apartado 5.3.2.

El criterio de utilidad del sistema se considera cumplido pues el sistema cuenta con una interfaz de línea de comandos intuitiva complementada con el manual de usuario del anexo II.

El criterio de fiabilidad no se considera probado completamente, pero las pruebas realizadas auguran resultados positivos sin una prueba completa hubiera tenido lugar.

En definitiva, se ha avanzado considerablemente en la consecución del principal objetivo del trabajo, aun cuando las dificultades relativas a la realización de vuelos de prueba, ya indicadas, han dejado pendiente parcialmente el segundo objetivo específico de obtener un diagrama en tierra y no han permitido podido abordar el tercero de hacer lo propio con una antena embarcada. Sin embargo, las pruebas parciales hacen razonable deducir que estos objetivos serían perfectamente alcanzables con el sistema propuesto.

6.2 Mejoras

Sobre este mismo proyecto pueden realizarse multitud de mejoras. A continuación se detallan algunas de ellas por orden de prioridad:

Estudiar la **mejora de la autonomía** del UAV. Existiendo para ello varias posibilidades:

1. Colocar dos baterías en paralelo lo que doblaría la capacidad de las mismas con el inconveniente del peso añadido.
2. Utilizar una batería de 4 celdas en lugar de tres. Esto aumenta el voltaje de pico de la batería, lo que provoca un aumento de la velocidad de rotación de los motores que habría que acompañar de unas hélices de paso menor para notar un aumento en la autonomía.

En línea con esto, se debería **mejorar la seguridad en el código**. Una vez conocido el comportamiento real de las baterías instaladas, el código puede mejorarse para avisar al usuario, o incluso actuar de manera autónoma, cuando no se considere que vaya a poder realizarse la misión asignada en base a autonomía y distancia restantes.

Adaptar el formato del fichero de mediciones al utilizado por el CEMEDM en su *software* de medidas. Esta mejora es muy sencilla de implementar y permitiría a los usuarios a los que va destinados el proyecto su utilización con herramientas a las que ya están acostumbrados.

Por último, teniendo en cuenta que la mar es un medio en el que el concepto estático queda en entredicho, el **conseguir que el vuelo del UAV se adapte a los movimientos del buque** debidos al viento y las olas, especialmente a los cambios de rumbo, mejoraría la precisión de las mediciones obtenidas. El algoritmo de vuelo está preparado para esto como se explicó en el apartado 3.4.2, por lo tanto, habría que implementar un sistema que se conectase en tiempo real con el buque y recibiera su posición GPS y su rumbo.

6.3 Líneas futuras

Además de resultar en un sistema útil en sí mismo, este proyecto ofrece un punto de partida para otras muchas aplicaciones útiles para la Armada Española.

El sistema desarrollado puede emplearse para el estudio y caracterización del multitrayecto para su aplicación en comunicaciones radio, radar, iluminadores para guiado de misiles rozaolas, etc. En ese mismo sentido, resultaría interesante el uso de un dron para la caracterización de conductos troposféricos en VHF y UHF.

Los algoritmos aplicados en el vuelo, e incluso la plataforma utilizada en algunos casos, pueden ser de gran utilidad en multitud de aplicaciones dirigidas al estudio de parámetros variables en los 360° alrededor del buque con solo cambiar el sensor utilizado. Por ejemplo, con un sensor infrarrojo convenientemente seleccionado, se podría obtener diagramas de la firma infrarroja del buque para distintas posiciones relativas al sol como el de la Figura 6-1. Incluso con un radar embarcado en el dron, podría llegar a calcularse la superficie equivalente radar del buque (RCS por sus siglas en inglés) para cada dirección, siendo esta otra de las tareas que realiza el CEMEDM, sin embargo, muy probablemente la plataforma utilizada para el presente proyecto no esté convenientemente dimensionada para esta misión.

Fig. 7. Contrasts radiation intensity conditions: Lw-010M (night, 1 Km).

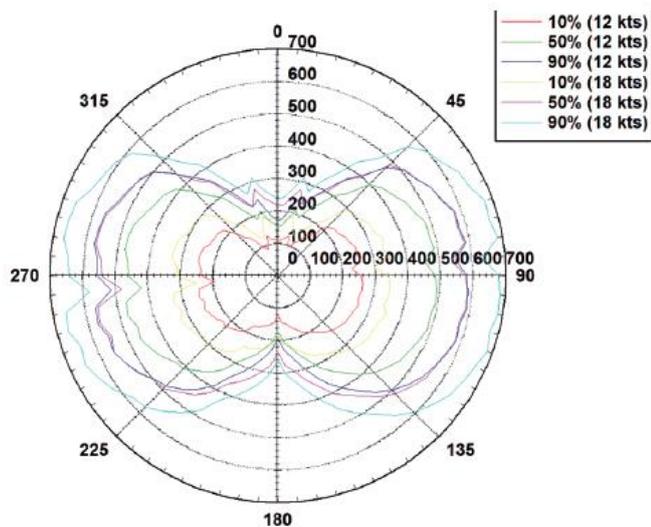


Figura 6-1 Firma IR de un buque obtenida de noche a 1 km de distancia por [61]

Si bien el dron utilizado y el código escrito en este proyecto no tendrían una aplicación directa, conceptualmente, el presente proyecto puede servir como referencia para el estudio de la utilización de sistemas aéreos no tripulados para otro tipo de tareas como la guerra electrónica, siendo muy útiles para ayudar a la triangulación de un emisor concreto, con solo incorporarles un radiogoniómetro y un enlace de datos robusto que permita analizar la señal desde el buque.

7 BIBLIOGRAFÍA

- [1] Unión Internacional de Telecomunicaciones, «Measuring the Information Society Report» 2015.
- [2] Unión Internacional de Telecomunicaciones, «ICT Facts and Figures» 2016.
- [3] Protección Civil, «Vademecum REMER. Telecomunicaciones.» Protección Civil, 2012. [En línea]. Available: <http://www.proteccioncivil.es/catalogo/carpeta02/carpeta24/vademecum12/vdm036.htm>. [Último acceso: 22 Febrero 2017].
- [4] Unión Internacional de Telecomunicaciones, «Convenio de la Unión Internacional de Telecomunicaciones» Ginebra, 1992.
- [5] Institute of Electrical and Electronics Engineers, «IEEE Standard definitions of terms for antennas» Std. 145-1993.
- [6] S. Manrique, «CQ Radio Amateur» 2004. [En línea]. Available: <http://www.qsl.net/ea3du/2004/01/Antenas.html>. [Último acceso: 12 Febrero 2017].
- [7] A. Rudge, K. Milne y others, *The Handbook of Antenna Design. Volume I*, Londres: Peter Peregrinus Ltd, 1982.
- [8] NFC Forum, «NFC Forum» 2017. [En línea]. Available: <http://nfc-forum.org/>. [Último acceso: 25 Febrero 2017].
- [9] Cincinnati Technical Center, «Electromagnetic radiation and how it affects your instruments.» Cincinnati, 1990.
- [10] A. Ruíz de Elvira, «¿Por qué tienen las antenas parabólicas forma de curva cónica?» *El Mundo*, 10 Noviembre 2013.
- [11] Military Review, «Military Review» 05 Julio 2012. [En línea]. Available: <http://penabad.blogspot.com.es/2012/07/spy-radar-arrays-arrive-in-adelaide.html>. [Último acceso: 25 Febrero 2017].
- [12] J. M. Hernando Rábanos, J. M. Riera y L. Mendo, *Transmisión por radio. 7ª Edición*, Madrid: Editorial Universitaria Ramón Areces, 2013.

- [13] «Promax» [En línea]. Available: <http://www.promax.es/esp/noticias/206/Analisis-de-ecos-dinamicos>. [Último acceso: 12 Febrero 2017].
- [14] Cisco, «Antenna Patterns and Their Meanings» 7 Agosto 2007. [En línea]. Available: http://www.cisco.com/c/en/us/products/collateral/wireless/aironet-antennas-accessories/prod_white_paper0900aec806a1a3e.html. [Último acceso: 2 Marzo 2017].
- [15] CEMEDEM, «Procedimiento de Medida de Diagramas» 2016.
- [16] Armada Española, «Armada Española» Ministerio de Defensa, [En línea]. Available: http://www.armada.mde.es/ArmadaPortal/page/Portal/ArmadaEspañola/buques_superficie/prefLang_es/04_Fragatas-F100-F80--01_fragatas-clase-alvaro-de-baza-f-100--03_fragatas-de-lezo-f-103. [Último acceso: 16 Febrero 2017].
- [17] J. M. Liaño, «Desarrollo de un sistema de control para UAV con capacidad ATOL entre lanchas de instrucción de la Escuela Naval Militar» Centro Universitario de la Defensa en la Escuela Naval Militar, Marín, 2016.
- [18] D. Isaac, «Diseño y puesta en marcha de un receptor SDR de bajo coste para experimentación en la banda de HF» Centro Universitario de la Defensa en la Escuela Naval Militar, Marín, 2015.
- [19] Wireless Telecom Group Inc., «Principles of Power Measurement» Parsippany, 2011.
- [20] S. Palmer y D. Walters, «RF Anechoic Chamber: Automated antenna radiation and gain measurement system with user interface» Rochester Institute of Technology, Rochester, 2012.
- [21] CATECHOM, «Introduction antenna measurement» Universidad de Alcalá de Henares, 2017. [En línea]. Available: http://www3.uah.es/catechom/index.php?option=com_content&view=article&id=46&Itemid=73&lang=en. [Último acceso: 2 Marzo 2017].
- [22] K. Mille, «What is an anechoic chamber?» Intellectual Ventures Laboratory, 20 Noviembre 2012. [En línea]. Available: <http://www.intellectualventureslab.com/invent/what-the-is-an-anechoic-chamber>. [Último acceso: 2 Marzo 2017].
- [23] J. L. Besada, «Introducción a la medida de antenas» Universidad Politécnica de Madrid, Madrid.
- [24] Unión Internacional de Telecomunicaciones, «UIT-R SM.2056-1. Verificación con aeronave de los diagramas de antena de las estaciones de radiodifusión» UIT, 2014.
- [25] T. Ridder, «Antenna radiation pattern measurement using an unmanned aerial vehicle» The Pennsylvania State University, State College, 2016.
- [26] National Instruments, «Super-Heterodyne Signal Analyzers. Description and applications».
- [27] Rohde & Schwarz, «R&S FSH. Handheld Spectrum Analyzer» [En línea]. Available: https://www.rohde-schwarz.com/es/producto/fsh-pagina-de-inicio-producto_63493-8180.html. [Último acceso: 1 Marzo 2017].
- [28] Unmanned Aerial Vehicle Systems Association, «UAVS» 2017. [En línea]. Available: https://www.uavs.org/index.php?page=what_is. [Último acceso: 21 Febrero 2017].

- [29] Instituto Tecnológico de Galicia, «Instituto Tecnológico de Galicia» Fundación ITG, 2013. [En línea]. Available: http://www.itg.es/?page_id=14762. [Último acceso: 21 Febrero 2017].
- [30] Uconsystem Co, «Real Time Remote Control» [En línea]. Available: <http://www.uconsystem.com/english/business/business04.asp>. [Último acceso: 2 Marzo 2017].
- [31] I. Shaw, «The Rise of the Predator Empire: Tracing the History of U.S. Drones» *Understanding Empire*, 2014.
- [32] P. Waldman, «Game of Drones» *The American Prospect*, 2013.
- [33] The Bureau of Investigative Journalism, «Afghanistan: Reported US air and drone strikes 2016» 2016. [En línea]. Available: <https://www.thebureauinvestigates.com/drone-war/data/get-the-data-a-list-of-us-air-and-drone-strikes-afghanistan-2016>. [Último acceso: 2 Marzo 2017].
- [34] T. Mockaitis, «Drones and the Ethics of War» 12 Enero 2016. [En línea]. Available: http://www.huffingtonpost.com/tom-mockaitis/drones-and-the-ethics-of_b_8961510.html. [Último acceso: 3 Marzo 2017].
- [35] C. Yuste, «EFE Verde. Drones: de la vigilancia de incendios a la restauración forestal» 28 Octubre 2015. [En línea]. Available: <http://www.efeverde.com/noticias/drones-restauracion-forestal/>. [Último acceso: 3 Marzo 2017].
- [36] V. Losada, «Los drones Amazon Prime Air han entregado su primer paquete» 14 Diciembre 2016. [En línea]. Available: <https://www.tuexperto.com/2016/12/14/los-drones-amazon-prime-air-han-entregado-su-primer-paquete/>. [Último acceso: 3 Marzo 2017].
- [37] BBC News, «Human 'drone taxi' to be tested in Nevada» 8 Junio 2016. [En línea]. Available: <http://www.bbc.com/news/technology-36478614>. [Último acceso: 3 Marzo 2017].
- [38] Dronecode Project, «Dronecode» 2017. [En línea]. Available: <https://www.dronecode.org/>. [Último acceso: 3 Marzo 2017].
- [39] The Wireless Innovation Forum, «The Wireless Innovation Forum. Main Page» 2017. [En línea]. Available: <http://www.wirelessinnovation.org/>. [Último acceso: 3 Marzo 2017].
- [40] The Wireless Innovation Forum, «Cognitive Radio Definitions» 2007.
- [41] J. Reed, *Software Radio. A modern approach to radio engineering*, Upper Saddle River: Prentice Hall, 2002.
- [42] The National Instruments Corporation, «What is I/Q Data?» 30 Marzo 2016. [En línea]. Available: <http://www.ni.com/tutorial/4805/en/>. [Último acceso: 3 Marzo 2017].
- [43] Wipro Technologies, «Software defined radio. A technology overview» 2002.
- [44] Nutaq Innovations, «A short history of software-defined radio (SDR) technology» [En línea]. Available: <https://www.nutaq.com/blog/short-history-software-defined-radio-sdr-technology>. [Último acceso: 3 Marzo 2017].
- [45] Portal de tecnología e innovación. Ministerio de Defensa, «Programa multinacional ESSOR (European Security Software defined Radio)» 2012. [En línea]. Available: <http://www.tecnologiaeinnovacion.defensa.gob.es/es->

- es/Contenido/Paginas/detallenoticia.aspx?noticiaID=215. [Último acceso: 3 Marzo 2017].
- [46] Portal de tecnología e innovación. Ministerio de Defensa, «La radio definida por software (SDR), tecnología habilitadora para el logro de soberanía nacional en las radiocomunicaciones militares» 19 Julio 2013. [En línea]. Available: <http://www.tecnologiaeinovacion.defensa.gob.es/es-es/Contenido/Paginas/detallereferencia.aspx?referenciaID=43>. [Último acceso: 3 Marzo 2017].
- [47] RTL-SDR, «RTL-SDR Main Page» 2017. [En línea]. Available: <http://www.rtl-sdr.com/>. [Último acceso: 3 Marzo 2017].
- [48] J. M. Alonso, «Hacking de comunicaciones móviles GSM con RTL-SDR» 12 Enero 2014. [En línea]. Available: <http://www.elladodelmal.com/2014/01/aumenta-la-vulnerabilidad-de-la-red-gsm.html>. [Último acceso: 3 Marzo 2017].
- [49] «Blog. Mundo Teleco» 4 Octubre 2014. [En línea]. Available: <http://mundotelecomunicaciones1.blogspot.com.es/2014/10/zona-de-fresnel.html>. [Último acceso: 23 Febrero 2017].
- [50] R. Valenzuela, O. Landron y D. Jacobs, «Estimating Local Mean Signal Strength of Indoor Multipath Propagation» *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, vol. 46, n° 1, pp. 203-212, 1997.
- [51] S. Scholl, «Exact signal measurements using FFT analysis» Microelectronic Systems Design Research Group, Kaiserslautern.
- [52] *Cuadro Nacional de Atribución de Frecuencias*, 2013.
- [53] W. Tomasi, *Sistemas de Comunicaciones Electrónicas*, Prentice Hall, 2001.
- [54] Great Scott Gadgets, «GSG. HackRF One» 2016. [En línea]. Available: <https://greatscottgadgets.com/hackrf/>. [Último acceso: 4 Marzo 2017].
- [55] NooElec, «RTL-SDR NooElec» 2017. [En línea]. Available: <http://www.nooelec.com/store/sdr/sdr-receivers/nesdr-mini-plus.html>. [Último acceso: 4 Marzo 2017].
- [56] GNURadio, «GNURadioCompanion Tutorial» 2017. [En línea]. Available: http://gnuradio.org/redmine/projects/gnuradio/wiki/Guided_Tutorial_GRC. [Último acceso: 4 Marzo 2017].
- [57] HELIPAL, «Storm 3.7V 680mAh 30C LiPo Battery Pack (for NE 318A)» [En línea]. Available: <http://www.helipal.com/storm-3-7v-600mah-20c-lipo-battery-pack-for-ne-318a.html>. [Último acceso: 7 Marzo 2017].
- [58] 4nec2, «4nec2 Main Page» [En línea]. Available: <http://www.qsl.net/4nec2/>. [Último acceso: 6 Marzo 2017].
- [59] J. Burke y A. Poggio, «Numerical Electromagnetics Code (NEC). Method of Moments. Part II: Program Description Code» Lawrence Livermore Laboratory, 1981.
- [60] R. Haviland, «Programs for Antenna Analysis by the Method of Moments» ARRL, 1995.
- [61] S. Marrugo, «Infrared Signature Analysis of Surface Ships» *SHIP. Science and Technology*, vol. VIII, n° 17, 2015.

- [62] Circuit Design Inc, «RF Technical Resources, Calculation Tools» [En línea]. Available: http://www.cdt21.com/resources/siryo1_02.asp. [Último acceso: 23 Febrero 2017].
- [63] GNURadio, «GNURadio Windows Install Guide» [En línea]. Available: <http://gnuradio.org/redmine/projects/gnuradio/wiki/WindowsInstall>. [Último acceso: 6 Marzo 2017].
- [64] Python, «Python Main Page» [En línea]. Available: <https://www.python.org/>. [Último acceso: 6 Marzo 2017].
- [65] ArduPilot, «Mission Planner Installation Guide» [En línea]. Available: <http://ardupilot.org/planner/docs/common-install-mission-planner.html>. [Último acceso: 6 Marzo 2017].
- [66] Putty Project, «Putty Main Page» [En línea]. Available: <http://www.putty.org/>. [Último acceso: 6 Marzo 2017].

ANEXO I: SIGLAS Y ACRÓNIMOS

ABP	Antena Bajo Prueba
ADC	<i>Analog-to-Digital Converter</i>
AM	<i>Amplitude Modulation</i>
C4ISR <i>Reconnaissance</i>	<i>Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance</i>
CEMEDEM	Centro de Medidas Electromagnéticas
CUD	Centro Universitario de la Defensa
CPU	<i>Central Processing Unit</i>
DAC	<i>Digital-to-Analog Converter</i>
DARPA	<i>Defense Advanced Research Projects Agency</i>
DDC	<i>Digital DownConverter</i>
DGAM	Dirección General de Armamento y Material
DSP	<i>Digital Signal Processor</i>
DUC	<i>Digital UpConverter</i>
DVB-T	<i>Digital Video Broadcasting. Terrestrial.</i>
ENM	Escuela Naval Militar
ESSOR	<i>European Security Software defined Radio</i>
FFT	<i>Fast Fourier Transform</i>
GCS	<i>Ground Control Station</i>
GPS	<i>Global Positioning System</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IF	<i>Intermediate Frequency</i>
IP	<i>Internet Protocol</i>
LiPo	<i>Lithium Polymer</i>
NFC	<i>Near Field Communication</i>
RBW	<i>Resolution BandWidth</i>
RCS	<i>Radar Cross-Section</i>
RF	Radio Frecuencia
RPAS	<i>Remotely Piloted Aircraft System</i>
RTL	<i>Register-Transfer Level</i>
RX	Recepción
SDR	<i>Software Defined Radio</i>
SMA	<i>SubMiniature versión A</i>

SNR	<i>Signal-to-Noise Ratio</i>
SSH	<i>Secure SHell</i>
TCP	<i>Transport Control Protocol</i>
TFG:	Trabajo Fin de Grado
TX:	Transmisión
UAS	<i>Unmanned Aerial System</i>
UAV	<i>Unmanned Aerial Vehicle</i>
UDP	<i>User Datagram Protocol</i>
UHF	<i>Ultra High Frequency</i>
UIT	Unión Internacional de Telecomunicaciones
USAF	<i>United States Air Force</i>
VHF	<i>Very High Frequency</i>

ANEXO II: MANUAL DE USUARIO

Este es el manual de usuario del sistema desarrollado.

AII.1 Planeamiento de la medida

AII.1.1 Radio de vuelo

Para seleccionar el radio de vuelo (distancia a la antena) hay que tener en cuenta los siguientes factores:

1. Eslora del buque.
2. Radio de Fresnel.
3. Campo lejano.
4. Efecto del multitrayecto.

All.1.1.1 Eslora del buque

Esta debe ser la medida mínima del radio de vuelo. Se usará como punto de partida para el resto de cálculos.

All.1.1.2 Radio de Fresnel

Se debe aplicar la siguiente fórmula:

$$r = \sqrt{\frac{\lambda d_1 d_2}{d}}$$

λ : Longitud de onda.

d : Radio de vuelo

d_1 y d_2 : La mitad del radio de vuelo

Al aplicar la fórmula, r debe ser menor que la altura a la que se sitúa la antena.

All.1.1.3 Campo lejano

Como mínimo, la el radio de vuelo debe ser mayor que:

$$r = \frac{2D^2}{\lambda}$$

D : Dimensión característica de la antena (longitud en el caso de un monopolo).

λ : Longitud de onda.

All.1.1.4 Efecto del multitrayecto

Se recomienda usar el recurso de Internet [62].

En este recurso se debe introducir la altura de la antena transmisora (embarcada) y la receptora (UAV) así como la frecuencia de trabajo y un número mayor pero del orden de la distancia a la que se está estimando realizar la medida. El resultado es una gráfica que indica dónde están los nulos y los máximos debidos al efecto del multitrayecto en el modelo de tierra plana. Este debe ser el último paso para realizar el ajuste fino de selección del radio de vuelo.

AII.1.2 Altura de vuelo

La altura de vuelo debe ser la altura a la que está la antena embarcada. Sin embargo, al introducirla en el programa hay que tener en cuenta que el UAV fija como altura 0 m el punto en el es armado, por lo tanto hay que tener en cuenta la diferencia entre este punto y la altura del nivel del mar.

AII.1.3 Ejemplo: Lancha de Instrucción

El siguiente ejemplo sería con una lancha de instrucción. Suponiendo que se va a caracterizar una antena VHF de 50 cm de largo situada a 4 m sobre el nivel del mar, se va a caracterizar a 150 MHz que es la frecuencia que contempla el CEMEDM en su procedimiento cuando se va a caracterizar una antena de VHF.

AII.1.3.1 Eslora de la lancha

La eslora de la lancha es de 20,5 m, se partirá de este valor.

AII.1.3.2 Radio de Fresnel

Aplicando la fórmula con:

$$\lambda = 2 \text{ m}$$

$$d = 20,5 \text{ m}$$

$$d_1 = d_2 = 10,25 \text{ m}$$

El resultado es:

$$r = 3,2 \text{ m}$$

Ya que el resultado es menor que los 4 m a los que se encuentra la antena sobre el mar, la distancia es buena, en caso contrario habría que aumentarla basándose en el principio universal de la ciencia empírica de «prueba y error».

AII.1.3.3 Campo lejano

Aplicando la fórmula con:

$$D = 0,5 \text{ m}$$

$$\lambda = 2 \text{ m}$$

El resultado es:

$$R = 0,25 \text{ m}$$

Por lo tanto, a más de 25 cm ya se está en el campo lejano, se puede continuar con 20,5 m.

AII.1.3.4 Efecto del multitrayecto

Utilizando el *applet* recomendado se comprueba que no existe ningún nulo a la distancia seleccionada (Figura AII-0-1).

La conclusión es que 20,5 m es un buen valor como radio de vuelo.

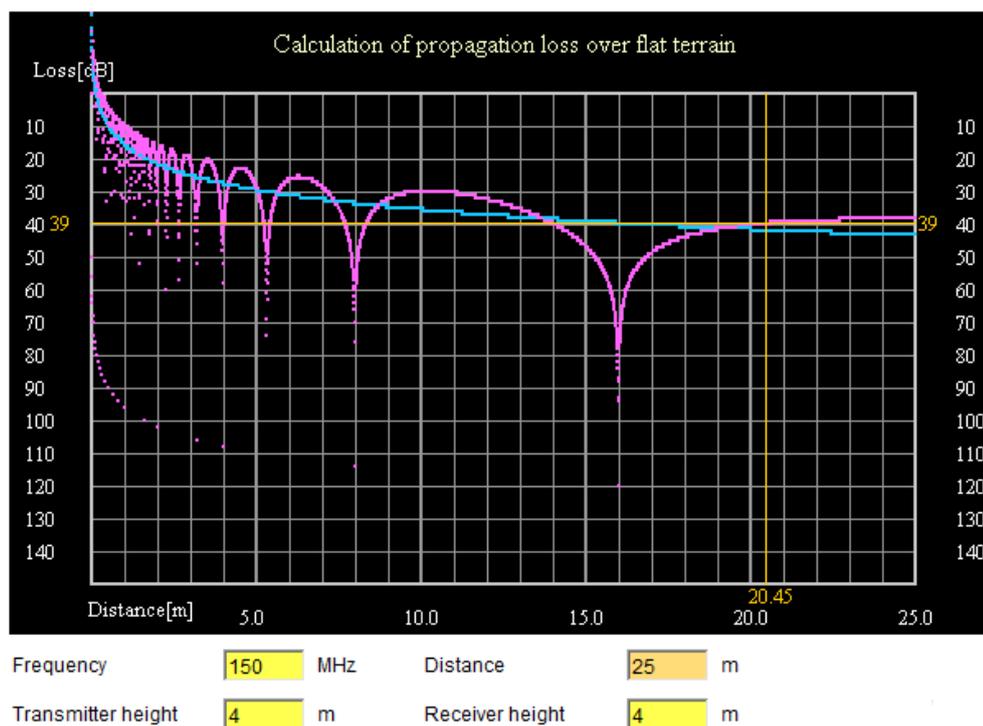


Figura AII-0-1 Simulación de efecto multitrayecto [62]

AII.2 Ejecución de la medida

AII.2.1 Pasos a seguir

Para ejecutar la medida hay que seguir los siguientes pasos:

1. Encender el UAV.
2. Conectarse al punto de acceso *wifi*.
3. Conectarse a la Raspberry Pi mediante SSH.
4. Iniciar `rtl_tcp` en la Raspberry Pi.
5. Iniciar MAVProxy en la Raspberry Pi.
6. Conectar Mission Planner al UAV para su monitorización.
7. Ejecutar `vuelo.py`.
8. Ejecutar `medida.py` cuando el servidor de `vuelo.py` esté en escucha.

AII.2.1.1 Encender el UAV

Para encender el UAV hay que conectar el conector XT60 de la batería con el del módulo de potencia (conectores mostrados en la Figura AII-0-2).



Figura AII-0-2 Conectores XT60 de batería y módulo de potencia

AII.2.1.2 Conectarse al punto de acceso *wifi*

Para conectarse al punto de acceso *wifi*, una vez esté iniciado (puede tardar algunos minutos desde que se enciende el UAV), hay que hacerlo con la utilidad de Windows que permite conectarse a una red inalámbrica (Figura AII-0-3).

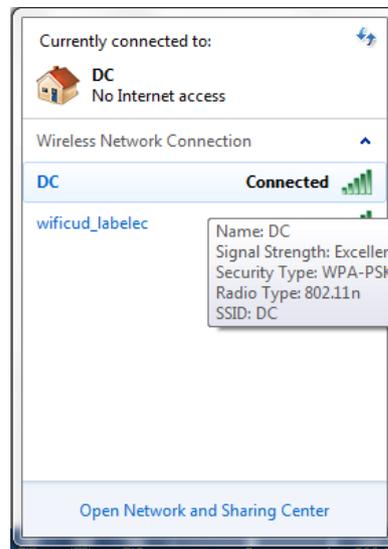


Figura AII-0-3 Red inalámbrica del punto de acceso *wifi*

AII.2.1.3 Conectarse a la Raspberry Pi mediante SSH

En Putty, es sencillo conectarse a la Raspberry Pi, simplemente hay que introducir la dirección IP de la misma que el punto de acceso *wifi* le haya asignado y el puerto 22 (Figura AII-0-4).

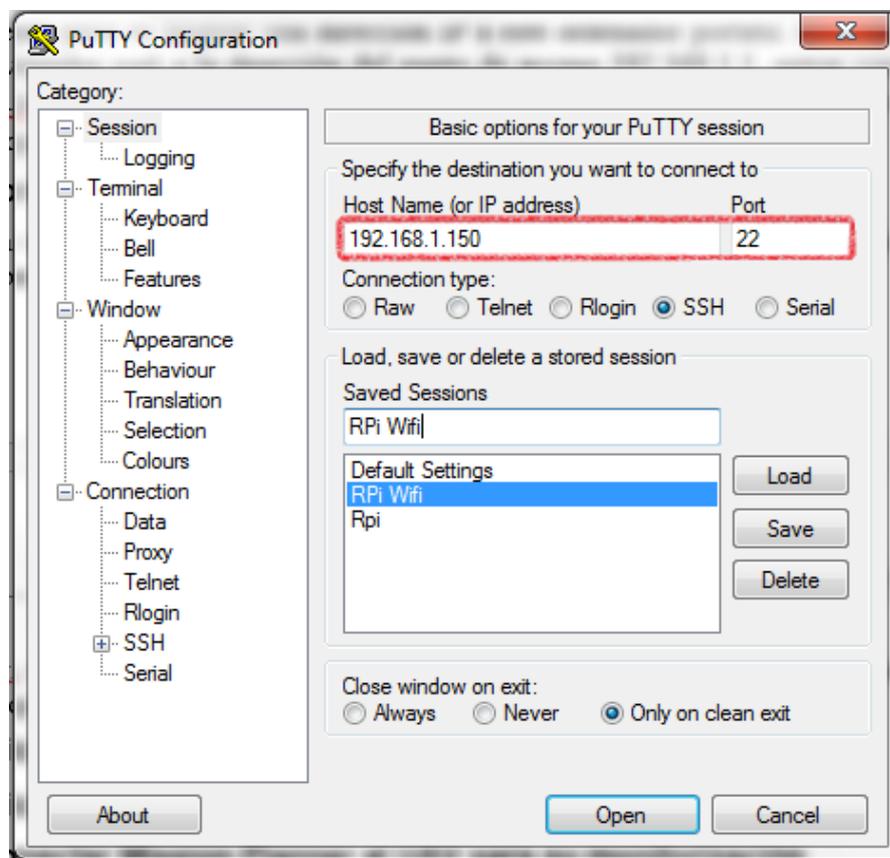


Figura AII-0-4 Conexión mediante Putty a la RPi

Además, para sistematizar el proceso, Putty da la opción de guardar esta configuración de dirección IP y puerto con solo pulsar el botón save.

Una vez iniciada la conexión, si ha sido exitosa se abrirá una ventana con una sesión de terminal remota en la que se debe iniciar sesión. El único usuario registrado en esta Raspberry es `pi` y su contraseña es `raspberrypi`.

AII.2.1.4 Iniciar `rtl_tcp` en la Raspberry Pi

Cuando se ha accedido a la Raspberry Pi, el siguiente paso es arrancar el servidor `rtl_tcp`. Para ello hay que ejecutar el siguiente comando en la terminal:

```
$ sudo ./inicio_radio
```

Que internamente ejecuta:

```
$ sudo rtl_tcp -a 192.168.1.150 &
```

AII.2.1.5 Iniciar MAVProxy en la Raspberry Pi

A continuación, debe iniciarse MAVProxy, para ello hay que ejecutar el siguiente comando en la terminal:

```
$ sudo ./inicio_mavproxy
```

Que internamente ejecuta:

```
$ sudo mavproxy.py --master=/dev/ttyAMA0 --baudrate 115200 --nowait --out
192.168.1.100:14550 --out 192.168.1.100:14551 --out 192.168.1.110:14552 --
out 192.168.1.110:14553
```

AII.2.1.6 Conectar Mission Planner al UAV para su monitorización

Una vez iniciado lo anterior, el siguiente paso es conectar Mission Planner al UAV para su monitorización. Para ello, hay que seleccionar el botón `Connect` (Figura AII-0-5) y conectarse al puerto UDP 14552 como se indica en la Figura AII-0-6:



Figura AII-0-5 Botón `connect` de Mission Planner

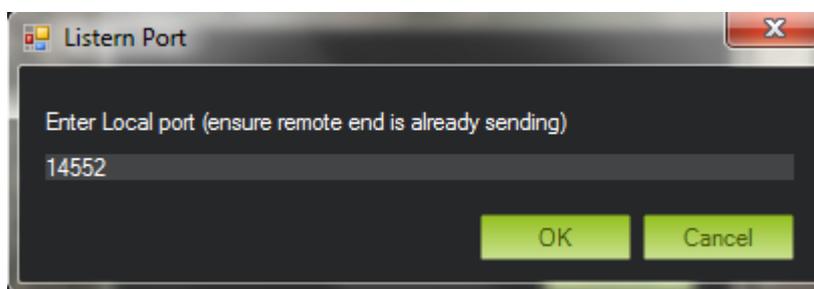


Figura AII-0-6 Selección de puerto UDP

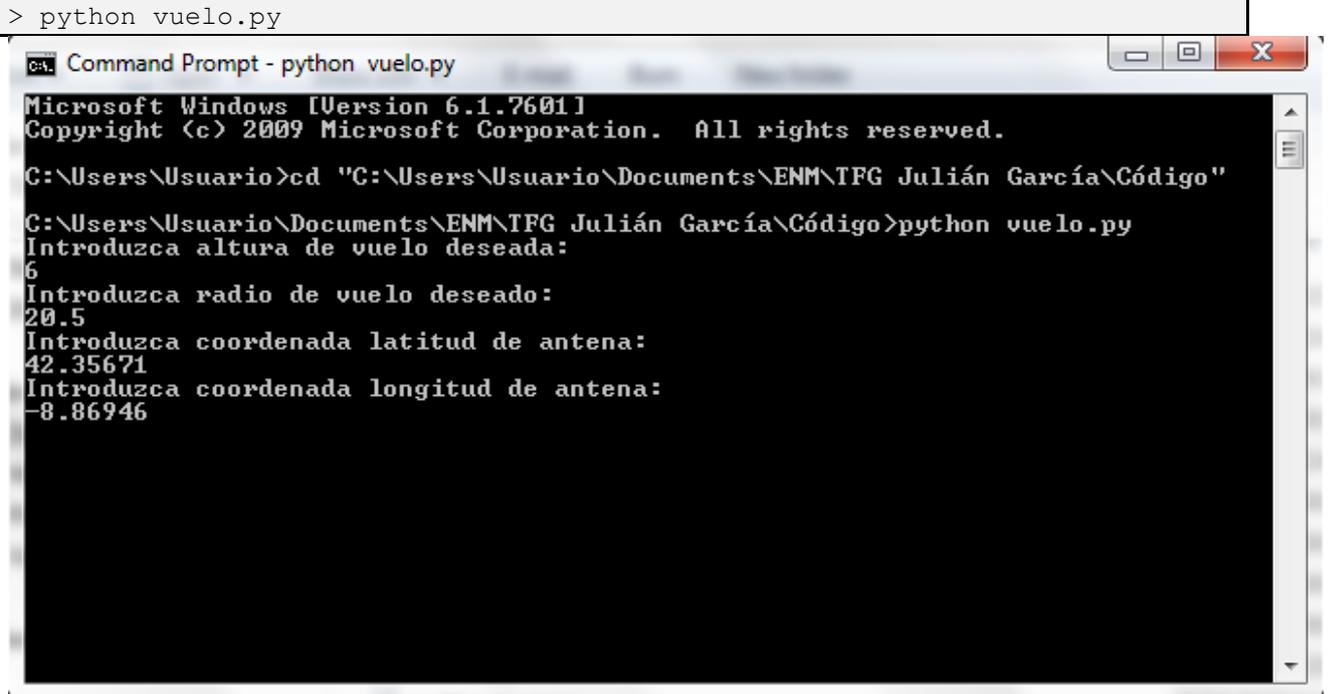
AII.2.1.7 Ejecutar `vuelo.py`

Para ejecutar el programa de vuelo, hay que acceder a una línea de comandos de Windows y cambiar de directorio de trabajo al directorio en el que estén grabados los códigos desarrollados.

En este punto, hay dos maneras de ejecutar el programa. Se puede iniciar directamente con los parámetros de radio y altura en metros, y posición de la antena introducidos como:

```
> python vuelo.py -h <altura> -r <radio> -l <latitud> -L <longitud>
EJEMPLO
> python vuelo.py -h 3 -r 25 -l 42.341444 -L -8.724360
```

O bien se puede iniciar el programa sin especificar ningún parámetro, en cuyo caso este irá pidiendo datos al usuario Figura AII-0-7:



```
> python vuelo.py

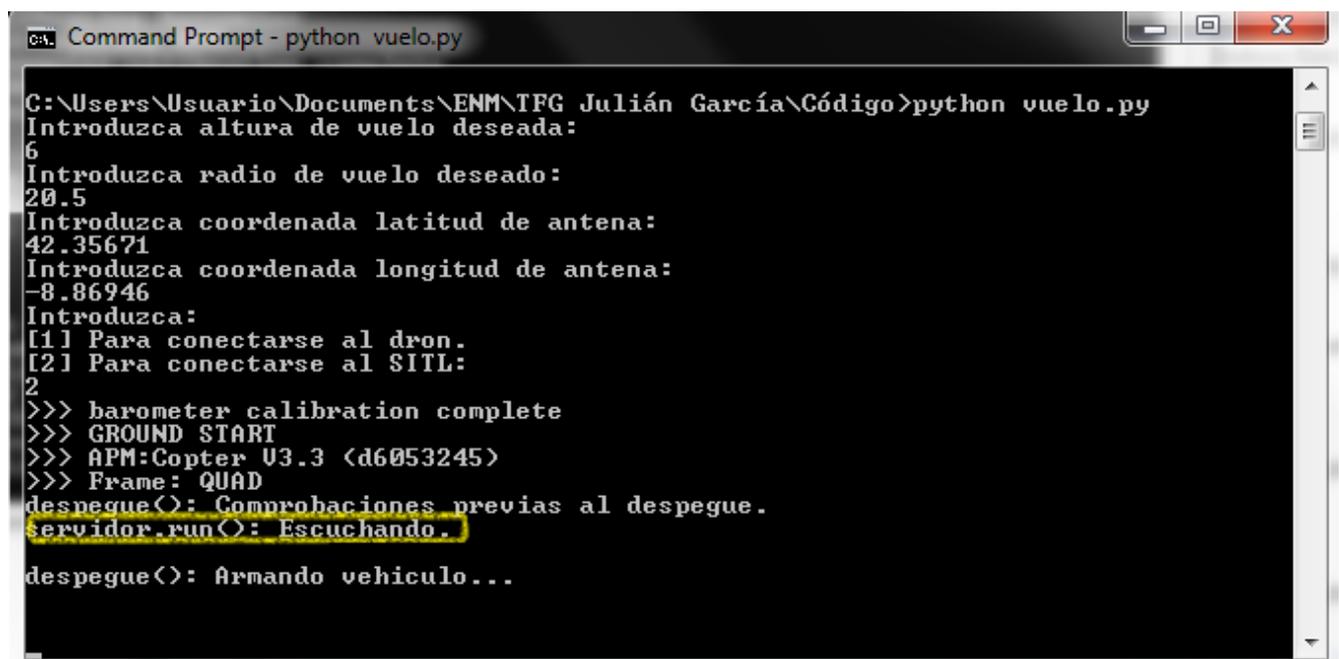
ca. Command Prompt - python vuelo.py
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\Usuario>cd "C:\Users\Usuario\Documents\ENM\TFG Julián García\Código"
C:\Users\Usuario\Documents\ENM\TFG Julián García\Código>python vuelo.py
Introduzca altura de vuelo deseada:
6
Introduzca radio de vuelo deseado:
20.5
Introduzca coordenada latitud de antena:
42.35671
Introduzca coordenada longitud de antena:
-8.86946
```

Figura AII-0-7 Interfaz del programa

A continuación el programa preguntará a dónde debe conectarse, al UAV o a SITL. Pulse 1 e <Intro> para conectarse al dron.

Si todas las conexiones están correctamente establecidas, el programa se conectará a la Pixhawk a través del MAVProxy instalado en la Raspberry Pi. Espere a que se conecte y se iniciará el servidor. Cuando reciba el mensaje de servidor en escucha (Figura AII-0-8), puede proceder al siguiente paso.



```
C:\Users\Usuario\Documents\ENM\TFG Julián García\Código>python vuelo.py
Introduzca altura de vuelo deseada:
6
Introduzca radio de vuelo deseado:
20.5
Introduzca coordenada latitud de antena:
42.35671
Introduzca coordenada longitud de antena:
-8.86946
Introduzca:
[1] Para conectarse al dron.
[2] Para conectarse al SITL:
2
>>> barometer calibration complete
>>> GROUND START
>>> APM:Copter U3.3 (d6053245)
>>> Frame: QUAD
despegue(): Comprobaciones previas al despegue.
servidor.run(): Escuchando.

despegue(): Armando vehiculo...
```

Figura AII-0-8 Mensaje de servidor en escucha

All.2.1.8 Ejecutar `medida.py` cuando el servidor de `vuelo.py` esté en escucha

Como se ha explicado en el apartado anterior, cuando el servidor de recepción de datos de medida esté en escucha, puede proceder a ejecutar el cliente `medida.py`.

Este cliente debe ejecutarse en el entorno GNURadio, para ello, abra la terminal GNURadio Command Prompt.

En esta terminal, que es una terminal normal de Windows con las variables de entorno adaptadas al entorno GNURadio, cambie al directorio de trabajo en el que tiene instalados los códigos.

En este directorio, debe ejecutar `medida.py`, de la misma manera que el programa `vuelo.py`, hay dos maneras de ejecutarlo, especificando la frecuencia central de escucha, o sin argumento, en cuyo caso el programa le pedirá introducir la frecuencia central.

```
ESPECIFICANDO LA FRECUENCIA
> python medida.py -f 868706250
SIN ESPECIFICAR LA FRECUENCIA
> python medida.py
```

All.3 Instalación del sistema por primera vez

All.3.1 Pasos a seguir en la instalación

Si es la primera vez que utiliza este sistema en su ordenador, debe instalar todo lo necesario. A continuación se detallan los pasos a seguir:

1. Instalación de GNURadio.
2. Instalación de Python y bibliotecas necesarias.
3. Instalación de Mission Planner.
4. Instalación de Putty y configuración de conexión a Raspberry Pi.

All.3.1.1 Instalación de GNURadio

GNURadio puede descargarse de su página *web* [63].

Para instalarlo solo tiene que lanzar el ejecutable y seguir los pasos indicados.

All.3.1.2 Instalación de Python y bibliotecas necesarias

Para instalar Python acceda a su página *web* [64]. Descargue la versión más reciente de la rama 2.7.*. No descargue las de la rama 3.* porque a día de redacción de la memoria las bibliotecas necesarias no funcionan en esta biblioteca.

Ejecute el instalador y siga los pasos indicados, cuando se le pregunte qué paquetes quiere instalar, hay una opción que habla de actualizar variables de entorno. Márquela de tal manera que al instalar las variables de entorno vayan a ser actualizadas.

Para instalar las bibliotecas necesarias, hay que usar el programa `pip` que viene por defecto con la instalación de Python. Acceda a una terminal de Windows y ejecute los siguientes comandos:

```
> pip install dronekit
> pip install numpy
```

Esto instalará las bibliotecas `dronekit`, `pymavlink` y `numpy`. Además de las dependencias necesarias.

All.3.1.3 Instalación de Mission Planner

Para instalar la GCS Mission Planner puede hacerlo desde su página *web* [65].

Ejecute el instalador y siga los pasos indicados.

All.3.1.4 Instalación de Putty y configuración de conexión a Raspberry Pi

Para conectarse a la Raspberry Pi se usará un cliente de terminal remoto basado en el protocolo SSH. Se recomienda utilizar Putty que puede descargarse desde [66].

Para conectarse al punto de acceso *wifi*, una vez esté iniciado (puede tardar algunos minutos desde que se enciende el UAV), hay que hacerlo con la utilidad de Windows que permite conectarse a una red inalámbrica (Figura AII-0-3). El nombre de la red es DC y la contraseña es lacasitos.

El siguiente paso es hacer que el punto de acceso *wifi* asigne la misma dirección IP siempre a este ordenador portátil. Para ello hay que acceder desde un navegador *web* a la dirección del punto de acceso 192.168.1.1, entrar con usuario *root* y contraseña *lacasitos*. En el interfaz *web*, se debe acceder a Network/"DHCP and DNS" (Figura AII-0-9):

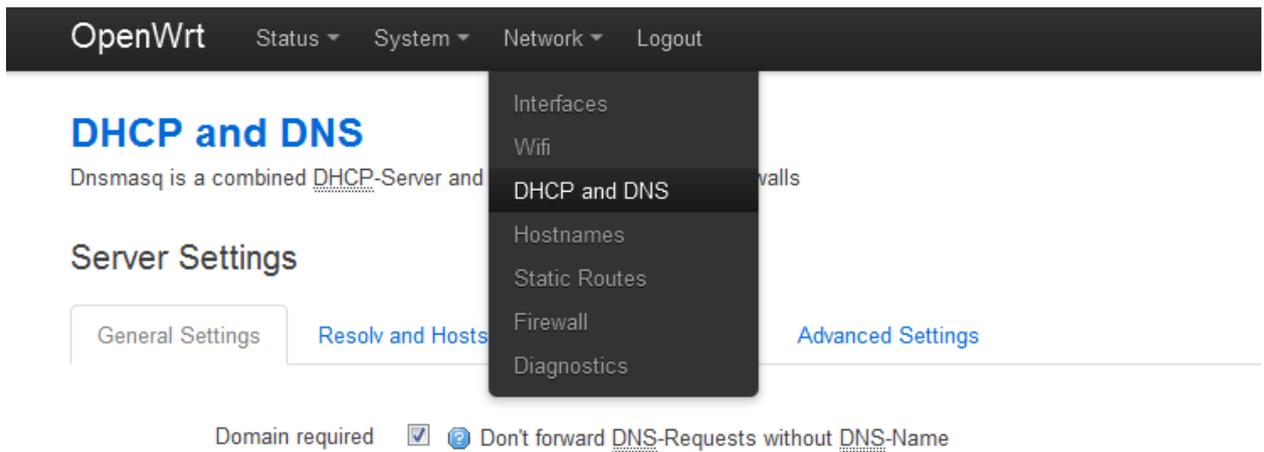


Figura AII-0-9 Asignación IP a ordenador portátil. Figura 1

En la pestaña *General Settings* (Figura AII-0-9), se desplaza la página hasta la parte más baja y en esta se encuentra la lista de direcciones IP que el punto de acceso asigna automáticamente a una serie de direcciones MAC señaladas. En este punto, debe añadir la dirección MAC del ordenador desde el que se va a ejecutar esta aplicación.

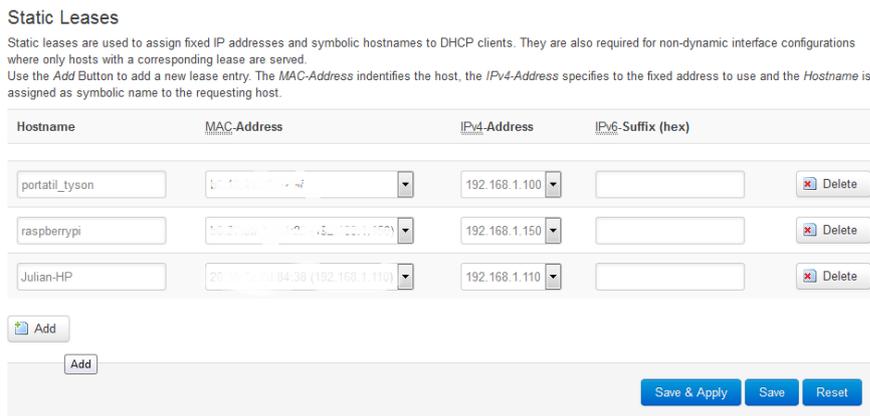


Figura AII-0-10 Asignación IP a ordenador portátil. Figura 2

ANEXO III: CÓDIGO FUENTE

AIII.1 Módulo medida de potencia

AIII.1.1 *medida.py*

Código fuente de *medida.py*.

```
#####
# medida.py
# Programa main del modulo de medida de potencia.
# En este programa se crea una instancia a la clase radioDRON
# de GNURadio y se ejecuta, a la vez que se conecta al servidor
# en el localhost y le envia las medidas de potencia tomadas.
#####

import time
import socket
import sys
import struct
import math
import radioSITL
import radioDRON
import getopt

opts, args= getopt.getopt(sys.argv[1:], "f:")
frecuencia = 868706250

for opt, arg in opts:
    if opt == "-f":
        frecuencia = float(arg)
    else:
        print "Ninguna opcion dada.\n"

if frecuencia == 868706250:
    print "Introduzca frecuencia central de escucha: "
    frecuencia = float(raw_input())

# radioSITL es el equivalente a radioDRON cuando se hicieron las pruebas con
# el ordenador tomando medidas y el SITL volando.
#tb = radioSITL.radioSITL()
tb = radioDRON.radioDRON(868706250)
tb.start()
time.sleep(2)

sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

dir_servidor = ("127.0.0.1", 6000)

sock.connect(dir_servidor)
print "Conectando\n"

while 1:
    msg = struct.pack('f', tb.get_var())
    sock.send(msg)
    time.sleep(0.1)

sock.close()
```

```

try:
    raw_input('Press Enter to quit: ')
except EOFError:
    pass

tb.stop()
tb.wait()

```

AIII.1.2 *radioDRON.py*

Código fuente de `radioDRON.py`, programa generado por GNURadio Companion y modificado ligeramente para poder variar la frecuencia central.

```

#!/usr/bin/env python2
# -*- coding: utf-8 -*-
#####
# GNU Radio Python Flow Graph
# Title: Radiodron
# Generated: Mon Feb 20 22:54:05 2017
#####

from gnuradio import blocks
from gnuradio import eng_notation
from gnuradio import filter
from gnuradio import gr
from gnuradio.eng_option import eng_option
from gnuradio.filter import firdec
from optparse import OptionParser
import osmosdr
import threading
import time

class radioDRON(gr.top_block):

    def __init__(self, frequency):
        gr.top_block.__init__(self, "Radiodron")

        #####
        # Variables
        #####
        self.var = var = 0
        self.samp_rate = samp_rate = 20e3
        self.rf_samp_rate = rf_samp_rate = 0.256e6
        self.freq = freq = frequency

        #####
        # Blocks
        #####
        self.probe_signal = blocks.probe_signal_f()

    def _var_probe():
        while True:
            val = self.probe_signal.level()
            try:
                self.set_var(val)
            except AttributeError:
                pass
            time.sleep(1.0 / (20))
        _var_thread = threading.Thread(target=_var_probe)
        _var_thread.daemon = True
        _var_thread.start()

```

```

self.rational_resampler_xxx_0 = filter.rational_resampler_ccc(
    interpolation=int(samp_rate),
    decimation=int(rf_samp_rate),
    taps=None,
    fractional_bw=None,
)
self.osmosdr_source_0 = osmosdr.source( args="numchan=" + str(1) + "
" + 'rtl_tcp=192.168.1.150:1234' )
self.osmosdr_source_0.set_sample_rate(rf_samp_rate)
self.osmosdr_source_0.set_center_freq(freq, 0)
self.osmosdr_source_0.set_freq_corr(62, 0)
self.osmosdr_source_0.set_dc_offset_mode(0, 0)
self.osmosdr_source_0.set_iq_balance_mode(0, 0)
self.osmosdr_source_0.set_gain_mode(False, 0)
self.osmosdr_source_0.set_gain(40, 0)
self.osmosdr_source_0.set_if_gain(0, 0)
self.osmosdr_source_0.set_bb_gain(0, 0)
self.osmosdr_source_0.set_antenna('', 0)
self.osmosdr_source_0.set_bandwidth(0, 0)

self.low_pass_filter_0 = filter.fir_filter_ccf(1, firdes.low_pass(
    1, rf_samp_rate, 10e3, 1e3, firdes.WIN_HAMMING, 6.76))
self.blocks_throttle_0 = blocks.throttle(gr.sizeof_gr_complex*1,
rf_samp_rate,True)
self.blocks_nlog10_ff_0 = blocks.nlog10_ff(10, 1, 0)
self.blocks_moving_average_xx_0 = blocks.moving_average_ff(2000,
0.0005, 4000)
self.blocks_complex_to_mag_squared_0 =
blocks.complex_to_mag_squared(1)

#####
# Connections
#####
self.connect((self.blocks_complex_to_mag_squared_0, 0),
(self.blocks_moving_average_xx_0, 0))
self.connect((self.blocks_moving_average_xx_0, 0),
(self.blocks_nlog10_ff_0, 0))
self.connect((self.blocks_nlog10_ff_0, 0), (self.probe_signal, 0))
self.connect((self.blocks_throttle_0, 0), (self.low_pass_filter_0,
0))
self.connect((self.low_pass_filter_0, 0),
(self.rational_resampler_xxx_0, 0))
self.connect((self.osmosdr_source_0, 0), (self.blocks_throttle_0,
0))
self.connect((self.rational_resampler_xxx_0, 0),
(self.blocks_complex_to_mag_squared_0, 0))

def get_var(self):
    return self.var

def set_var(self, var):
    self.var = var

def get_samp_rate(self):
    return self.samp_rate

def set_samp_rate(self, samp_rate):
    self.samp_rate = samp_rate

def get_rf_samp_rate(self):
    return self.rf_samp_rate

def set_rf_samp_rate(self, rf_samp_rate):

```

```

self.rf_samp_rate = rf_samp_rate
self.osmosdr_source_0.set_sample_rate(self.rf_samp_rate)
self.low_pass_filter_0.set_taps(firdes.low_pass(1,
self.rf_samp_rate, 10e3, 1e3, firdes.WIN_HAMMING, 6.76))
self.blocks_throttle_0.set_sample_rate(self.rf_samp_rate)

def get_freq(self):
    return self.freq

def set_freq(self, freq):
    self.freq = freq
    self.osmosdr_source_0.set_center_freq(self.freq, 0)

```

AIII.2 Módulo plataforma de vuelo

AIII.2.1 *vuelo.py*

Código fuente de *vuelo.py*.

```

#####
# vuelo.py
# Programa main del modulo plataforma de vuelo.
# Este programa solicita datos al usuario de altura, radio y
# posicion. Despues en un hilo aparte se inicia un objeto de la
# clase Servidor y se pone en escucha.
# Despues se crea una instancia de la clase
# Multicoptero, se arma y se le ordena hacer el vuelo circular.
# En un bucle se recogen las medidas del Servidor y se escriben
# en un fichero junto a la demora respecto a la antena.
#####

from dronekit import *
import multicoptero
import auxiliar
import servidor
import threading
import sys
import getopt

opts, args= getopt.getopt(sys.argv[1:], "h:r:l:L:")
altura = 0
radio = 0
lat = 0
long = 0

for opt, arg in opts:
    if opt == "-h":
        altura = float(arg)
    elif opt == "-r":
        radio = float(arg)
    elif opt == "-l":
        lat = float(arg)
    elif opt == "-L":
        long = float(arg)
    else:
        print "Ninguna opcion dada.\n"

if altura == 0:
    print "Introduzca altura de vuelo deseada: "
    altura = float(raw_input())
if radio == 0:
    print "Introduzca radio de vuelo deseado: "

```

```

    radio = float(raw_input())
if lat == 0:
    print "Introduzca coordenada latitud de antena: "
    lat = float(raw_input())
if long == 0:
    print "Introduzca coordenada longitud de antena: "
    long = float(raw_input())

print "Introduzca:\n[1] Para conectarse al dron.\n[2] Para conectarse al
SITL: "
opcion = raw_input()

if opcion == "1":
    #RaspberryPi
    dron = connect("udp:192.168.1.110:14553", wait_ready = True,
vehicle_class = multicoptero.Multicoptero)
elif opcion == "2":
    #MAVProxy
    dron = connect("udp:127.0.0.1:14561", wait_ready = True, vehicle_class =
multicoptero.Multicoptero)
else:
    print "Comando incorrecto, saliendo."
    exit()

#SITL
#dron = connect("tcp:127.0.0.1:5760", wait_ready = True, #vehicle_class =
multicoptero.Multicoptero)

serv = servidor.Servidor("", 6000)
serv.start()

dron.despegue(altura)

centro = LocationGlobalRelative(lat, long, 12)

demora = auxiliar.obtener_demora(centro,
dron.location.global_relative_frame)

dron.vuelo_aproximacion(centro, radio, altura)

#centro, radio, altura, precision, velocidad, demora_inicial, sentido
th_vuelo = threading.Thread(target=dron.vuelo_circular, args=(centro, radio,
altura, 10, 2, demora, 1))
th_vuelo.start()

file = open("medicion.txt", "w")

while dron.fin_vuelo == 0:
    potencia = serv.get_potencia()
    demora = auxiliar.obtener_demora(centro,
dron.location.global_relative_frame)
    #print "Potencia: "+ str("{0:.2f}".format(potencia)) + " Demora: " +
str("{0:.2f}".format(demora)) + "\n"
    file.write(str(time.strftime("%H%M%S")) + " " +
str("{0:.2f}".format(demora)) + " " + str("{0:.2f}".format(potencia)) +
str("\n"))
    time.sleep(0.1)

print "mision cumplida, volvemos.\n"
dron.mode = VehicleMode("RTL")

file.close()
serv.stop()

```

```
dron.fin()
```

AIII.2.2 servidor.py

Código fuente de la clase Servidor.

```
#####
# servidor.py
# Este es el modulo de la clase Servidor que hereda de Thread.
# Esta clase consiste en un servidor que se ejecuta en un hilo
# aparte y que lee la lectura de potencia obtenida por el
# programa implementado en GNURadio.
#####

import socket
import math
import sys
import time
import struct
import threading

class Servidor (threading.Thread):
    # Variable miembro de la clase
    # en la que se guarda la potencia leida.
    potencia = 0

    def __init__(self, dir_ip, puerto):
        threading.Thread.__init__(self)
        self.dir_servidor = (dir_ip, puerto)

    def recibe(self):
        # En este metodo se lee lo que llega del cliente,
        # y se guarda en la variable potencia como float.
        while 1:
            buf_rec = self.cliente.recv(4)
            fl = struct.unpack('f', buf_rec)
            self.potencia = fl[0]

    def run(self):
        # Metodo de inicio del servidor en un hilo aparte.
        # Este metodo es llamado al ejecutar el metodo start()
        self.sock_servidor = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)

        self.sock_servidor.bind(self.dir_servidor)

        self.sock_servidor.listen(1)
        print "servidor.run(): Escuchando.\n"

        self.cliente, addr = self.sock_servidor.accept()
        print "servidor.run(): Aceptado.\n"

        self.recibe()

    def stop(self):
        # Se cierran las conexiones
        # y se vuelve al hilo principal.
        self.cliente.close()
        self.sock_servidor.close()
        self.join()

    def get_potencia(self):
        return self.potencia
```

AIII.2.3 *multicoptero.py*

Código fuente de la clase Multicoptero.

```
#####
# multicoptero.py
# Este es el modulo de la clase Multicoptero
# que hereda de Vehicle.
# Esta clase implementa metodos a un mas alto nivel para
# despegue, cambio de altura, movimiento a un punto
# y vuelo circular.
#####

import sys, time
import dronekit
from dronekit import *
import auxiliar

class Multicoptero (dronekit.Vehicle):
    # Variable que determina si se ha recibido COMMAND_ACK o no
    cmd_ack = 0
    fin_vuelo=0

    def __init__(self, *args):
        # Se inicia desde la clase madre
        super(Multicoptero, self).__init__(*args)
        # Cuando se reciba el mensaje COMMAND_ACK se ejecuta el metodo ack
        self.add_message_listener("COMMAND_ACK", self.ack)

    def ack(self, name, msg, otr):
        #Cuando se recibe el mensaje COMMAND_ACK, la variable cmd_ack pasa
        a valer 1, en caso contrario deberia valer 0, esto se implementara en cada
        metodo segun sea necesario
        self.cmd_ack = 1

    def despegue(self, altura):
        #Este metodo ejecuta el armado y despegue del dron hasta la altura
        argumento

        #Comprobacion de si es armable o no
        print "despegue(): Comprobaciones previas al despegue.\n"
        while not self.is_armable == True:
            time.sleep(2)
            print "despegue(): Multicoptero no se puede armar.\n"

        #Armado de vehiculo
        print "despegue(): Armando vehiculo...\n"
        self.mode = VehicleMode("GUIDED")
        self.armed = True
        while not self.mode.name == "GUIDED" and not self.armed == True:
            time.sleep(2)
            print "despegue(): Me estoy armando...\n"
        print "despegue(): Vehiculo armado.\n"

        #Orden de despegue, a continuacion en un bucle se comprueba cada
        segundo si se ha alcanzado la altura ordenada
        print "despegue(): Despegando...\n"
        self.simple_takeoff(altura)
        while True:
            time.sleep(1)
            print "despegue(): Altura: "
            +str(self.location.global_relative_frame.alt)+ "\n"
            if self.location.global_relative_frame.alt>=altura*0.95:
```

```

        print "despegue(): Alcanzada altura de: " +
str(self.location.global_relative_frame.alt) + "\n"
        #Una vez alcanzada la altura, se devuelve 0 representando despegue
exitoso
        return 0

    def ir_hacia(self, destino, velocidad):
        #Metodo que ordena al dron dirigirse a un punto a una velocidad
determinada. El dron debe estar en modo GUIDED de antes.

        #self.mode = VehicleMode("GUIDED")
        #La variable de clase cmd_ack se pone a 0 para comprobar si las
ordenes llegan al dron
        self.cmd_ack = 0

        #Se ordena al dron dirigirse a la posicion destino, a continuacion
se comprueba si la orden ha llegado o no con cmd_ack, en caso de que no
llegue, se vuelve a enviar una vez, si tampoco llega, la funcion devuelve 1
indicando error
        print "ir_hacia(): Dirigiendome hacia: " + str(destino) + "\n"
        distoriginal =
auxiliar.obtener_distancia(self.location.global_relative_frame, destino)
        self.simple_goto(destino, airspeed=velocidad)
        time.sleep(2)
        if not self.cmd_ack == 1:
            print "ir_hacia(): No se envio la orden. Reenviando...\n"
            #Tiempo de espera antes de volver a enviar la orden
            time.sleep(5)
            self.simple_goto(destino, airspeed=velocidad)
            time.sleep(2)
            if not self.cmd_ack == 1:
                print "ir_hacia(): No es posible enviar la orden al
dron.\n"
                return 1

        #Una vez la orden ha llegado, en un bucle se comprueba cada 2
segundos si la aeronave ha llegado ya a su destino, con una precision de 0.5
metros. Una vez alcanzado se devuelve 0 indicando exito
        while self.mode.name == "GUIDED":
            dist =
auxiliar.obtener_distancia(self.location.global_relative_frame, destino)
            if dist <= 2:
                print "ir_hacia(): He llegado a mi destino.\n"
                return 0
            print "ir_hacia(): En vuelo, quedan " +
str(auxiliar.obtener_distancia(self.location.global_relative_frame,
destino)) + " m\n"
            time.sleep(2)

        #En caso de que no se llegue al destino, se devuelve 1 indicando
que hubo algun problema
        print "ir_hacia(): Hubo algun problema, no se ha alcanzado el
destino.\n"
        return 1

    def cambiar_altura(self, altura, velocidad):
        #Este metodo sirve para cambiar la altura en una posicion estatica.
Funciona de manera muy similar a ir_hacia() con comprobacion de
COMMAND_CHECK
        #self.mode = VehicleMode("GUIDED")
        posicion_nueva = self.location.global_relative_frame
        posicion_nueva.alt = altura
        self.cmd_ack = 0

```

```

        print "cambiar_altura(): Cambiando cota a "+str(altura)+" m a una
velocidad vertical de "+str(velocidad)+" m/s.\n"
        self.simple_goto(posicion_nueva, airspeed=velocidad)
        time.sleep(2)
        if not self.cmd_ack == 1:
            print "ir_hacia(): No se envio la orden. Reenviando...\n"
            #Tiempo de espera antes de volver a enviar la orden
            time.sleep(2)
            self.simple_goto(posicion_nueva, airspeed=velocidad)
            time.sleep(2)
            if not self.cmd_ack == 1:
                print "ir_hacia(): No es posible enviar la orden al
dron.\n"
                return 1

        while self.mode.name == "GUIDED":
            time.sleep(1)
            print "cambiar_altura(): Altura: "
+str(self.location.global_relative_frame.alt) + "\n"
            if math.fabs(self.location.global_relative_frame.alt-
altura)<=altura*0.05:
                print "cambiar_altura(): Alcanzada altura de: " +
str(self.location.global_relative_frame.alt) + "\n"
                #Una vez alcanzada la altura, se devuelve 0 representando
despegue exitoso
                return 0

        def vuelo_circular(self, centro, radio, altura, precision, velocidad,
demora_inicial, sentido):
            #Este metodo ordena al dron realizar un vuelo circular alrededor
del punto centro con un radio y a una velocidad dadas. La precision indica
los grados que abarcara cada lado que tendra el poligono asemejable a la
circunferencia. La demora inicial indica el angulo inicial respecto al norte
del primer punto del poligono, y el sentido 1 si es horario, -1 antihorario

            self.mode = VehicleMode("GUIDED")
            self.fin_vuelo = 0
            print "vuelo_circular(): Realizando vuelo circular.\n"
            for i in range((360/precision)+1):
                p1 = auxiliar.obtener_punto_demdist(centro, radio,
demora_inicial+sentido*i*precision, altura)
                print "vuelo_circular(): Waypoint: " + str(i) + "\n"
                if self.ir_hacia(p1, velocidad)==1:
                    print "vuelo_circular(): Hay un problema desconocido,
no se pudo alcanzar el siguiente waypoint.\n"
                    return 1
                print "vuelo_circular(): Realizado vuelo circular con exito.\n"
                self.fin_vuelo = 1
            return 0

        def vuelo_aproximacion(self, centro, dist_min, altura):
            posicion_original = self.location.global_relative_frame
            demora = auxiliar.obtener_demora(centro, posicion_original)
            punto_minimo = auxiliar.obtener_punto_demdist(centro, dist_min,
demora, altura)

            print "vuelo_aproximacion(): Iniciando vuelo de aproximacion hasta
punto situado a " +str(dist_min)+" del centro.\n"
            self.ir_hacia(punto_minimo, 10)
            return 0

        def fin(self):

```

```
self.close()
```

AIII.2.4 auxiliar.py

Código fuente del módulo auxiliar.py.

```
#####
# auxiliar.py
# En este modulo hay una serie de funciones auxiliares que
# permiten operaciones geometricas como obtener distancia
# u obtener demora entre dos puntos dados.
# Codigos obtenidos en su mayoria de ArduPilot
#####

import math
from dronekit import *

def obtener_distancia(punto1, punto2):
    """
    Returns the ground distance in metres between two `LocationGlobal` or
    `LocationGlobalRelative` objects.
    This method is an approximation, and will not be accurate over large
    distances and close to the earth's poles. It comes from the ArduPilot test
    code:
    https://github.com/diydrones/ardupilot/blob/master/Tools/autotest/common.py
    """
    dlat = punto2.lat - punto1.lat
    dlong = punto2.lon - punto1.lon
    return math.sqrt((dlat*dlat) + (dlong*dlong)) * 1.113195e5

def obtener_demora(punto1, punto2):
    """
    Returns the bearing between the two LocationGlobal objects passed as
    parameters.
    This method is an approximation, and may not be accurate over large
    distances and close to the earth's poles. It comes from the ArduPilot test
    code:
    https://github.com/diydrones/ardupilot/blob/master/Tools/autotest/common.py
    """
    off_x = (punto2.lon -
punto1.lon)*math.cos(math.pi*((punto1.lat+punto2.lat)/2)/180)
    off_y = punto2.lat - punto1.lat
    demora = math.atan2(off_x, off_y) * 57.2957795
    if demora < 0:
        demora += 360.00
    return demora

def obtener_punto_distancias(original_location, dNorth, dEast, altura):
    """
    Returns a LocationGlobal object containing the latitude/longitude
    `dNorth` and `dEast` metres from the specified `original_location`. The
    returned LocationGlobal has the same `alt` value as `original_location`.
    The function is useful when you want to move the vehicle around
    specifying locations relative to the current vehicle position.
    The algorithm is relatively accurate over small distances (10m within
    1km) except close to the poles.
    For more information see:
    http://gis.stackexchange.com/questions/2951/algorithm-for-offsetting-a-
    latitude-longitude-by-some-amount-of-meters
    """
    earth_radius = 6378137.0 #Radius of "spherical" earth
    #Coordinate offsets in radians
    dLat = dNorth/earth_radius
```

```
dLon = dEast/(earth_radius*math.cos(math.pi*original_location.lat/180))

#New position in decimal degrees
newlat = original_location.lat + (dLat * 180/math.pi)
newlon = original_location.lon + (dLon * 180/math.pi)
if type(original_location) is LocationGlobal:
    targetlocation=LocationGlobal(newlat, newlon, altura)
elif type(original_location) is LocationGlobalRelative:
    targetlocation=LocationGlobalRelative(newlat, newlon, altura)
return targetlocation

def obtener_punto_demdist(original_location, distancia, demora, altura):
    """
    Basado en la funcion obtener_punto_distancias() pero dando demora y
    distancia relativas a posicion central. Angulo de la demora empieza a contar
    el cero en el norte y aumenta en sentido horario igual que las demoras en
    los buques.
    """
    dNorth = distancia*math.cos((demora*math.pi)/180)
    dEast = distancia*math.sin((demora*math.pi)/180)
    earth_radius = 6378137.0 #Radius of "spherical" earth
    #Coordinate offsets in radians
    dLat = dNorth/earth_radius
    dLon = dEast/(earth_radius*math.cos(math.pi*original_location.lat/180))

    #New position in decimal degrees
    newlat = original_location.lat + (dLat * 180/math.pi)
    newlon = original_location.lon + (dLon * 180/math.pi)
    if type(original_location) is LocationGlobal:
        targetlocation=LocationGlobal(newlat, newlon, altura)
    elif type(original_location) is LocationGlobalRelative:
        targetlocation=LocationGlobalRelative(newlat, newlon, altura)
    return targetlocation
```