

Alex Lombao Chiuta

**DESARROLLO DE UN SISTEMA DE DECODIFICACIÓN
DE SEÑALES VOR PARA EL GUIADO Y LA
DETERMINACIÓN DE LA POSICIÓN DE UN AVIÓN EN
VUELO**

TRABAJO DE FIN DE GRADO

dirigido por la Dra. Elisabet Xifré Pérez

**Grado en Ingeniería de Sistemas y Servicios de
Telecomunicaciones**



UNIVERSITAT ROVIRA I VIRGILI

**Tarragona
2021**

Resumen

El objetivo principal de este proyecto es la recepción y decodificación de las señales emitidas por estaciones VOR y la creación, a partir de ellas, de un completo sistema para guiar a los pilotos de aeronaves durante el transcurso de su vuelo. Para tal fin, se utiliza una antena conectada a un RTL-SDR, cuya función es digitalizar la señal recibida para su posterior tratamiento y decodificación mediante software.

El resultado de la decodificación de la señal VOR proporciona información sobre la posición relativa del avión respecto a la estación VOR, es decir, los radiales de dicho VOR. Sin embargo, no proporciona la posición exacta de la aeronave, razón por la cual el segundo punto de la parte práctica del proyecto se basa en la creación de un algoritmo para el cálculo de la posición exacta de la aeronave, usando únicamente la señal recibida del VOR y sin la necesidad de utilización de sistemas como el GPS o el DME.

La decodificación y procesado de la señal VOR se ha realizado con el programa MATLAB mientras que la interfaz gráfica se ha desarrollado con el programa App Designer. Dicha interfaz muestra un conjunto de instrumentos, fáciles de interpretar para el piloto, con los resultados de la decodificación y del algoritmo de cálculo de la posición de la aeronave.

En definitiva, un piloto que use este software podrá reunir información del radial por el que vuela a tiempo real, el rumbo del vuelo del avión y la posición de la aeronave en el mapa. Estos tres factores reunidos en un único sistema, que además es barato de coste, hacen que sea una herramienta para el guiado por el aire que muchos pilotos agradecerán tener en la cabina.

Resum

L'objectiu principal d'aquest projecte és la recepció i descodificació dels senyals emesos per estacions VOR i la creació, a partir d'aquests, d'un complet sistema per guiar als pilots d'aeronaus durant el transcurs del seu vol. Per a tal finalitat, s'utilitza una antena connectada a un RTL-SDR, la funció de la qual és digitalitzar el senyal rebut per al seu posterior tractament i descodificació mitjançant software.

El resultat de la descodificació del senyal VOR proporciona informació sobre la posició relativa de l'avió respecte a l'estació VOR, és a dir, els radials d'aquest VOR. No obstant això, no proporciona la posició exacta de l'aeronau, raó per la qual el segon punt de la part pràctica del projecte es basa en la creació d'un algorisme per al càlcul de la posició exacta de l'aeronau, usant únicament el senyal rebut del VOR i sense la necessitat d'utilització de sistemes com el GPS o el DME.

La descodificació i processament del senyal VOR s'ha realitzat amb el programa MATLAB mentre que la interfície gràfica s'ha desenvolupat amb el programa App Designer. Aquesta interfície mostra un conjunt d'instruments, fàcils d'interpretar per al pilot, amb els resultats de la descodificació i de l'algorisme de càlcul de la posició de l'aeronau.

En definitiva, un pilot que faci servir aquest programari podrà reunir informació del radial pel que vola a temps real, el rumb de vol de l'avió i la posició de l'aeronau al mapa. Aquests tres factors units en un únic sistema, que a més és barat de cost, fan que sigui una eina per al guiat per l'aire que molts pilots agrairan tenir a la cabina.

Abstract

The aim of this project is the reception and decoding of the signals emitted by VOR stations and the creation, based on these signals, of a complete system to guide aircraft pilots during their flight. For this purpose, an antenna connected to an RTL-SDR is used, whose function is to digitize the received signal for later processing and decoding by software.

The result of the decoding of the VOR signal provides information about the relative position of the aircraft in relation with the VOR station, ergo the radials of that VOR. However, it does not provide the exact position of the aircraft, which is why the second point of the practical part of the project is based on the creation of an algorithm for calculating the exact position of the aircraft, using only the signal received from the VOR and without the need to use systems such as GPS or DME.

The decoding and processing of the VOR signal has been carried out with MATLAB while the graphical interface has been developed with App Designer. This interface shows a set of instruments, which are easy for the pilot to interpret, with the results of both the decoding and the algorithm for calculating the position of the aircraft.

In conclusion, a pilot using this software will be able to gather information of the radial through which he is flying in real time, the heading of the plane and the position of the aircraft on the map. These three factors combined in a single system, which in addition is cheap in cost, makes it a tool for air guidance that many pilots will appreciate having in the cockpit.

En especial agradecimiento a todas las personas que me han apoyado en la realización de este proyecto.

A mis padres, a la directora del trabajo Elisabet Xifré, a mi amigo y estudiante de aviación Edwin Cabrera, a mi amigo e ingeniero aeronáutico Alberto Modrego y a todo el personal del CESDA (Centro de Estudios Superiores de la Aviación).

Índice

1	Introducción	1
1.1	Motivación.....	1
1.2	Objetivos.....	2
<u>PARTE TEÓRICA</u>		
2	El Sistema VOR.....	4
2.1	Definición de VOR y Principales Características.....	4
2.2	Evolución e Historia del VOR	5
2.3	Línea de futuro de los VOR respecto a otros sistemas de navegación aérea	5
2.4	Clasificación de los VOR	6
2.5	Funcionamiento Básico de un VOR.....	7
3	Conceptos Básicos de las Señales Emitidas por un VOR.....	9
3.1	Introducción al Concepto de Modulación	9
3.2	Modulación de Amplitud en Cuadratura.....	10
3.3	Modulación en Amplitud AM	12
3.3.1	Análisis de una Señal AM en el Dominio Temporal	12
3.3.2	Análisis de una Señal AM en el Dominio Frecuencial	13
3.4	Modulación en Frecuencia FM	15
3.4.1	Análisis de una Señal FM en el Dominio Temporal	15
3.4.2	Las Funciones de Bessel	17
3.4.3	Análisis de una Señal FM en el Dominio Frecuencial.....	19
3.5	El Efecto Doppler	21
4	La Señal de los VOR	23
4.1	Estructura de la Señal Emitida por los VOR.....	23
4.2	El Espectro de la Señal de un VOR	27
4.3	Las antenas de los VOR	28
4.4	El Cono de Confusión.....	31
4.5	La Malla Metálica del VOR	31
5	Aplicación de los VOR en la Aviación.....	33
5.1	Guiado Mediante un VOR.....	33
5.2	Los VOR en las Cartas Aeronáuticas	34
5.3	Instrumentos VOR en la Cabina del Avión	35
5.4	El CDI.....	36
5.5	Las Banderas TO y FROM.....	36
5.6	Uso Operacional de un VOR	40
5.7	Maniobras con los VOR.....	41
5.7.1	Paso por Encima del VOR	41
5.7.2	Paso por el Lateral de un VOR	41
5.7.3	Intercepción de un Radial	41

6	<i>Introducción al Software Defined Radio</i>	42
6.1	Los Conversores	42
6.2	Fundamentos del RTL-SDR.....	43
6.3	Aplicaciones del SDR	46
6.4	RTL-SDR en Estado de Recepción	47
6.5	RTL-SDR en Estado de Transmisión	47
6.6	Software de Tratamiento Digital de Señal	47
6.7	Ejemplo de la Recepción y Visualización del Espectro de una Señal VOR Captada por el RTL-SDR.....	48

PARTE PRÁCTICA

7	<i>Diseño del Receptor de Señales de los VOR</i>	50
7.1	Hardware Utilizado	50
7.2	Análisis de las Antenas Apropriadas para la Recepción de Señales de los VOR.....	52
8	<i>Estructura del Código del Demodulador de las Señales del VOR</i>	54
8.1	Conexión entre el RTL-SDR y MATLAB.....	54
8.2	Declaración del Objeto SDR.....	55
8.3	Declaración del Objeto de los Filtros.....	55
8.4	Declaración del Objeto de Audio	56
8.5	Demodulación de la Señal AM	57
8.6	Demodulación de la Señal FM.....	58
8.7	Diezmado de las Señales Desmoduladas.....	60
8.8	Cálculo de la Diferencia de Fase	60
9	<i>Diseño de un Algoritmo para el Cálculo de la Posición de un Avión en Vuelo</i>	62
9.1	El Teorema del Seno.....	63
9.2	Variables y Lecturas Iniciales	63
9.3	Planteamiento del Algoritmo.....	64
9.4	Cálculo de los Ángulos Contiguos a la Línea de Desplazamiento del Avión.....	65
9.5	Cálculo de la Distancia entre el VOR y el Avión	67
9.6	Cálculo de las Variables de las Rectas que Definen los Radiales y la Trayectoria del Avión	67
9.7	Determinación del <i>Heading</i> del Avión	68
10	<i>Diseño de la Interfaz Gráfica del Panel con los Instrumentos del VOR</i>	70
10.1	App Designer.....	70
10.2	Estructura Visual de la Aplicación	71
10.2.1	Botón de Encendido y Apagado del Panel.....	71
10.2.2	Tabla de Selección del VOR y Sintonizador de Frecuencia	71
10.2.3	Cuadro de Información del VOR Seleccionado	72
10.2.4	Selector del Modo de Funcionamiento.....	73
10.2.5	Pantalla Principal.....	73
10.2.6	Instrumentos Tradicionales	74

10.3	Estructura Interna de la Aplicación	75
10.3.1	Funciones Globales	75
10.3.2	Callbacks	76
11	<i>Presupuesto</i>.....	78
	<i>Referencias</i>	79
	<i>Anexos</i>.....	82
	Anexo A. Código para la Demodulación de las Señales de los VOR	82
	Anexo B. Código para el Cálculo de la Posición del Avión.....	84
	Anexo C. Código para la Grabación de Señales captadas por un RTL-SDR	87
	Anexo D. Código de la Interfaz Gráfica	89

Índice de Figuras

Figura 1. Doppler VOR en la terminar de un aeropuerto.....	4
Figura 2. Símbolos de diferentes radioayudas en las cartas aeronáuticas [6].	6
Figura 3. Representación de los radiales de un VOR y el desfase entre la señal de referencia y la señal variable [7].	8
Figura 4. Esquema de modulación IQ [10].	10
Figura 5. Esquema de demodulación IQ [10].	11
Figura 6. Representación temporal de una señal modulada en AM.....	13
Figura 7. Representación frecuencial de una señal modulada en AM.	14
Figura 8. Esquema de generación de una señal AM.	14
Figura 9. Representación temporal de una señal modulada en FM.	17
Figura 10. Tabla de los coeficientes de Bessel en función del índice de modulación [18].	18
Figura 11. Representación gráfica de las funciones de Bessel [15].	19
Figura 12. Representación frecuencial de una señal modulada en FM.....	20
Figura 13. Esquema de generación de una señal FM.....	21
Figura 14. Representación del giro efectuado por la señal radiada por las antenas del VOR en sentido antihorario.....	24
Figura 15. Desviación de la frecuencia recibida por un avión en el radial 090 de la señal variable apuntando hacia el radial 270.	25
Figura 16. Desviación de la frecuencia recibida por un avión en el radial 090 de la señal variable apuntando hacia el radial 180.	25
Figura 17. Desviación de la frecuencia recibida por un avión en el radial 090 de la señal variable apuntando hacia el radial 090.	25
Figura 18. Desviación de la frecuencia recibida por un avión en el radial 090 de la señal variable apuntando hacia el radial 000 o 360.	26
Figura 19. Estructura del espectro radioeléctrico de las señales emitidas por un Doppler VOR.	27
Figura 20. Diseño original de la antena de Alford [24].	28
Figura 21. Distribución de corrientes por la antena de Alford [24].	29
Figura 22. Esquema del circuito que sujeta la antena [24].	30
Figura 23. Diagrama de la forma del radomo que protege la antena [24].	30
Figura 24. Diagrama representativo del cono de confusión del VOR [1].	31
Figura 25. Malla metálica de debajo de las antenas.....	32
Figura 26. Variación de las indicaciones de los diferentes instrumentos de la cabina del avión en función del rumbo [1].	34
Figura 27. Representación de un VOR en las cartas aeronáuticas [1].	35
Figura 28. Instrumentos de indicación de desviación OBI.	35
Figura 29. Esquema de la posición del avión e indicaciones en los instrumentos [28].	37
Figura 30. Esquema de la posición del avión e indicaciones en los instrumentos [28].	37
Figura 31. Esquema de la posición del avión e indicaciones en los instrumentos [28].	38
Figura 32. Esquema de la posición del avión e indicaciones en los instrumentos [28].	38
Figura 33. Esquema de la posición del avión e indicaciones en los instrumentos [28].	39

Figura 34. Esquema de la posición del avión e indicaciones en los instrumentos [28].	39
Figura 35. Desviación nula del CDI en el radial seleccionado y su opuesto [1]......	40
Figura 36. Esquema del montaje del receptor de señales VOR.	42
Figura 37. Proceso de digitalización de una señal continua en el tiempo.....	42
Figura 38. Estructura elemental de un RTL-SDR.	44
Figura 39. Proceso de traslado de la señal a banda base.	44
Figura 40. Estructura interna y detallada de un RTL-SDR [15].	45
Figura 41. Ejemplo de una captura de una señal de un VOR con el software Cubic SDR.	48
Figura 42. De izquierda a derecha, el amplificador de NooElec y el SDR con tecnología Bias Tee [33, 34].	50
Figura 43. Características del amplificador utilizado [35].	51
Figura 44. Esquema circuital del sistema Bias Tee.	51
Figura 45. Antenas VHF para la recepción de señales VOR en el fuselaje de un avión. ..	52
Figura 46. Diagrama de radiación de una antena monopolo.	53
Figura 47. Antena extensible utilizada en el proyecto.	53
Figura 48. Esquema de un demodulador de señales moduladas en AM.	57
Figura 49. Esquema del algoritmo de demodulación de señales moduladas en FM.	59
Figura 50. Diagrama de los radiales captados y la trayectoria seguida por el avión.	62
Figura 51. Lados y vértices de un triángulo cualquiera.	63
Figura 52. Representación de los radiales y la trayectoria seguida por el avión en el plano cartesiano.	64
Figura 53. Distancia entre el VOR y la trayectoria del avión para cada radial capturado.	65
Figura 54. Diseño de la interfaz del programa.	70
Figura 55. Interruptor general de encendido y apagado del programa.	71
Figura 56. Panel de selección del VOR y la frecuencia.	72
Figura 57. Cuadro de información general del VOR y el modo seleccionado.	73
Figura 58. Interruptores para la elección del modo de funcionamiento del programa.	73
Figura 59. Pantalla principal del programa. A la izquierda se muestra la imagen de un VOR seleccionado previamente en la lista predeterminada; a la derecha se muestra el modelo general para cualquier VOR.	74
Figura 60. Instrumentos tradicionales añadidos a la interfaz. De izquierda a derecha: brújula indicadora del heading del avión, el OBI y el velocímetro manual.	75

Índice de Tablas

Tabla 1. Señal en fase y cuadratura en función del tipo de modulación analógica.....	12
Tabla 2. Expresiones del efecto Doppler en función del movimiento del emisor y el receptor.	22
Tabla 3. Cálculo de los grados de rotación de la brújula del heading en función del desplazamiento.....	69
Tabla 4. Presupuesto del proyecto.....	78

Lista de Acrónimos

ADC, A/D	Conversor Analógico a Digital (<i>Analog to Digital Converter</i>)
AIS	Proveedor de Información Aérea (<i>Aeronautical Information Supplier</i>)
AM	Modulación en Amplitud (<i>Amplitude Modulation</i>)
CDI	Indicador de la Desviación del Rumbo (<i>Course Deviation Indicator</i>)
CVOR	VOR Convencional (<i>Conventional VOR</i>)
DAC, D/A	Conversor Digital a Analógico (<i>Digital to Analog Converter</i>)
DGAC	Dirección General de Aviación Civil
DME	Equipamiento de Medida de Distancia (<i>Distance Measurement Equipment</i>)
DSP	Procesado Digital de Señal (<i>Digital Signal Processing</i>)
DVOR	VOR Doppler (<i>Doppler VOR</i>)
FAA	Federal Aviation Administration
FFT	Transformada Rápida de Fourier (<i>Fast Fourier Transform</i>)
FIR	Respuesta de Impulso Finito (<i>Finite Impulse Response</i>)
FM	Modulación en Frecuencia (<i>Frequency Modulation</i>)
OACI/ICAO	Organización de Aviación Civil Internacional (<i>International Civil Aviation Organization</i>)
IIR	Respuesta de Impulso Infinito (<i>Infinite Impulse Response</i>)
IQ	Fase y Cuadratura o Amplitud en Cuadratura (<i>In-Phase and Quadrature</i>)
LoS	Línea Directa de Visión (<i>Line of Sight</i>)
OBI	Indicador Omnidireccional de Rumbo (<i>Omnidirectional Bearing Indicator</i>)
OBS	Setter Omnidireccional de Rumbo (<i>Omnidirectional Bearing Selector</i>)
PPL	Lazo de Seguimiento de Fase (<i>Phase Lock Loop</i>)
SDR	Radio Definida por Software (<i>Software Defined Radio</i>)
TACAN	Sistema Táctico de Navegación Aérea (<i>Tactical Air Navigation System</i>)
VCO	Oscilador Controlado por Tensión (<i>Voltage Controlled Oscillator</i>)
VHF	Muy Alta Frecuencia (<i>Very High Frequency</i>)
VOR	Rango de Radio Omnidireccional de Muy Alta Frecuencia (<i>Very High Frequency Omnidirectional Radio Range</i>)
VORTAC	VOR+TACAN

1 Introducción

La aviación es un mundo fascinante. A pesar de que el ser humano está hecho para caminar y correr, el afán de alzar el vuelo y separar los pies de tierra firme ha existido siempre. No fue hasta hace poco más de un siglo, el 17 de diciembre de 1903, cuando los hermanos Wright consiguieron realizar el primer vuelo controlado de la historia. A partir de dicho año, el mundo de la aviación no ha parado de crecer y evolucionar.

No obstante, lo que más me fascina del mundo aeronáutico es toda la ingeniería que hay detrás, desde la estructura del fuselaje del avión, las alas, el tren de aterrizaje y los motores, hasta los instrumentos de la cabina y todos los elementos de radiocomunicación de los que dispone una aeronave. De hecho, la relación entre las telecomunicaciones y la aviación es muy estrecha, ya que la única forma de la que dispone avión de interactuar con el exterior es mediante ondas de radio.

1.1 Motivación

Des de bien pequeño me han gustado los aviones. Cada día aprendo cosas nuevas sobre aviones, y cada día me fascina más el hecho de saber que un mundo tan exclusivo como el de la aviación tenga tantas cosas que ofrecer.

Entre pilotos existe una frase muy conocida, que además es la base de todas las acciones que realizan en la cabina de un avión: *Aviate, Navigate, Communicate*. Se trata de primeramente centrarse en volar el avión, para posteriormente guiarse en el aire y finalmente comunicarse con el exterior. Este trabajo se centrará en el segundo punto de esta norma mnemotécnica, la de navegar por el aire. Uno de los principales objetivos de este trabajo es relacionar las telecomunicaciones con la aviación, y dado que prácticamente todos los sistemas de navegación de una aeronave se basan en comunicaciones mediante ondas de radio, fue la combinación perfecta para comenzar a indagar en el tema y desarrollarlo más a fondo.

Los pilotos siempre han necesitado sistemas de navegación aérea que les permita conocer su posición, de donde vienen y hacia donde se dirigen, dado que en el aire no siempre disponen de una buena visibilidad del suelo para coger referencias. Un sistema muy utilizado actualmente, entre muchos otros, son los llamados VOR, siglas de *Very High Frequency Omnidirectional Radio Range*. Los VOR son unos radiofaros que todas las aeronaves utilizan en vuelo para su guiado por el aire, ya sean de aviación general, comercial o militar.

Para mostrarle la información al piloto de la aeronave, los aviones están provistos de antenas especialmente diseñadas para recibir las señales de los VOR y las cabinas de los aviones están provistas con varios instrumentos que muestran el rumbo del avión respecto a los radiales de un VOR, la desviación del avión respecto a uno de los radiales y muchas otras aplicaciones más.

Sin embargo, no existe ningún instrumento que los englobe a todos y que además permita la interacción con el piloto. Tampoco existen instrumentos que muestren a los pilotos realmente todos los detalles que se pueden extraer de la señal de un VOR, ya sean los radiales por los que vuela el avión, la posición del VOR, la frecuencia del VOR, etc. Por lo tanto, el afán de querer simplificar en un único sistema todo lo que el VOR puede aportar ha sido la motivación para comenzar a desarrollar este proyecto.

1.2 Objetivos

Como ya se ha mencionado anteriormente, el punto de partida de este trabajo ha sido el deseo que querer relacionar las telecomunicaciones con el mundo de la aviación. En el momento que conseguí definir el tema supe, des del primer momento, que los VOR serían un buen tema para desarrollar, ya que no es común encontrar artículos que hablen de aspectos técnicos de los VOR. De hecho, hay poca documentación sobre el tema, a diferencia de su hermano mayor, el GPS.

Tras muchas conversaciones con pilotos profesionales, ingenieros aeronáuticos, instructores de vuelo, el equipo de personal del Centro de Estudios Superiores de la Aviación (CESDA) de Reus y amigos míos estudiantes de aviación, todos estuvieron de acuerdo que la información que recibe un piloto en cabina cuando quiere navegar mediante las señales de un VOR son muy limitadas respecto a lo que realmente se podría conseguir hacer con ellas.

En consecuencia, mi objetivo principal des del primer momento fue el de construir un sistema de demodulación de señales de VOR, con el mínimo coste económico, que a través de una interfaz gráfica intuitiva pudiera ayudar a los pilotos a guiarse en el aire, siendo esto un sistema suplementario y redundante a los ya existentes instrumentos en la cabina de una aeronave.

Pero, además, quise llevar el concepto de la creación de un panel interactivo más allá, a algo que no ha existido jamás en la cabina de un avión como un instrumento propiamente dicho. Se trata del diseño de un algoritmo no muy exacto pero fiable del cálculo de la posición de una aeronave con tan solo conocer los radiales relativos de un solo VOR y la velocidad de la aeronave.

Por lo tanto, dado que en la actualidad todos los pilotos llevan siempre una tableta con ellos en cabina donde consultan sus *checklists*, las cartas aeronáuticas y realizan cálculos referentes al vuelo, entre muchas otras cosas, este programa lo pueden instalar en sus tabletas e interactuar con él siempre que lo deseen, sin necesidad de hacer modificaciones en el avión ni cursos de habilitación para nuevos instrumentos.

En definitiva, el objetivo es crear un sistema de navegación mediante VOR, redundante a los sistemas ya existentes en la cabina de un avión que, por un coste económico muy bajo y sin la necesidad de realizar modificaciones en el panel físico del avión, cualquier piloto de cualquier tipo de aeronave pueda utilizar y comprender.

PARTE TEÓRICA

2 El Sistema VOR

En este capítulo se trata de introducir el VOR como concepto general y se presentan algunas características básicas de este sistema.

2.1 Definición de VOR y Principales Características

El radiofaro omnidireccional de muy alta frecuencia, de sus siglas en inglés VOR (*Very High Frequency Omnidirectional Radio Range*), es una radioayuda usada en el ámbito de la aeronáutica que permite a una aeronave conocer su posición angular o su acimut con respecto a la estación VOR utilizando como referencia el norte magnético terrestre [1 – 3]. Estas estaciones se usan para el guiado de la aeronave por el aire de un punto a otro, constituyen los puntos de referencia para la constitución de las aerovías y son un gran sistema de ayuda para los pilotos, ya que facilita efectuar ciertas maniobras en el aire tales como aproximaciones a aeropuertos, órbitas o virajes.



Figura 1. Doppler VOR en la terminar de un aeropuerto.

Generalmente los VOR están formados por un *array* de 48 antenas dispuestas en forma de circunferencia con un radio de 13,4 metros aproximadamente (44 pies) y otra antena más en el centro de dicha circunferencia, todas ellas colocadas sobre una malla metálica, las cuales radian de forma omnidireccional en el plano paralelo al suelo, tal y como se puede observar en la Figura 1.

Ciertamente los VOR están estandarizados en todo el mundo, trabajando en el rango internacional de radiofrecuencias aeronáuticas de la banda VHF de muy alta frecuencia, concretamente entre los 108 MHz y 118 MHz [4]. En esta banda de frecuencias se consigue una calidad de señal muy robusta cuyo alcance de cobertura es elevado, evitando así degradaciones debido a la meteorología y a los efectos de la ionosfera terrestre.

Existen muchos tipos de radiofaros, tanto para la aviación general y comercial como para la aviación militar. Se clasifican por las prestaciones que pueden ofrecer, que van desde la potencia que emiten hasta la localización donde se encuentran instalados. Por ejemplo, la

señal de un VOR de largo alcance, dedicado a la aviación comercial y en condiciones de LoS (*Line of Sight*), puede viajar hasta 300 kilómetros a la redonda antes de atenuarse y hasta 45.000 pies de altitud, es decir, muy por encima del techo de altitud de los aviones comerciales de hoy en día.

La variante más popular y utilizada en la actualidad son los VOR-DME, acrónimo de *Distance Measurment Equipment*). Esta variante incluye las prestaciones de un DVOR, o Doppler VOR, más un sistema radar con respuesta activa que mide la distancia entre la aeronave (interrogador) y la estación VOR (transpondedor) basado en el principio de funcionamiento del efecto Doppler. Con la tecnología DME también se extraen los datos de la velocidad del avión respecto al suelo (*ground speed*) y del tiempo de vuelo hasta la estación de tierra. Ver apartado 2.4 para más información.

2.2 Evolución e Historia del VOR

Los radiofaros en la aviación comenzaron a aparecer en la década de 1930 en los Estados Unidos después de que a la FAA le empezara a atraer la idea de implementar radioayudas para la navegación aérea [5]. Uno de los primeros prototipos, llamados *Visual-Aural Range* o VAR, fueron las primeras estaciones terrenas en trabajar en la banda VHF, entre las frecuencias de 108 MHz y 112 MHz. Se comenzaron a probar en la aerovía que conecta la ciudad de Nueva York con Chicago sin conseguir ningún éxito, ya que las señales eran muy débiles y poco fiables.

Poco tiempo más tarde surgieron los VOR como un sistema de navegación revolucionario para la aviación de la época, debido a su versatilidad y fiabilidad. Los VOR se empezaron a desarrollar en el año 1937 y puestos en marcha en el 1943, sustituyendo todos los sistemas de navegación aérea usados hasta la fecha. Sin embargo, no fue hasta bien pasada la Segunda Guerra Mundial, en la década de 1950, cuando los VOR empezaron a extenderse por todo el territorio estadounidense.

La primera aerovía provista enteramente con estaciones VOR se inauguró en el año 1951, que hasta entonces se habrían implementado 271 estaciones por todo el país norteamericano. En 1952 ya se habría provisto 45.000 millas de aerovías con estaciones VOR.

A partir de entonces el nuevo concepto de radiofaro VOR empezó a internacionalizarse, llegando en 1959 a proclamarse por la ICAO el estándar de ayuda para la navegación aérea de la comunidad internacional.

A partir de la década de 1970, la precisión de los VOR se mejoró, introduciendo el nuevo concepto de los DVOR o Doppler VOR, cuyo funcionamiento se basa en el efecto Doppler y cuyo funcionamiento está detallado en el apartado 4 de este trabajo.

2.3 Línea de futuro de los VOR respecto a otros sistemas de navegación aérea

Actualmente, la utilización de estas estaciones terrenas en la aviación comercial moderna son un sistema de navegación que, en muchos casos, queda en la sombra de los sistemas satelitales de posición, como el GPS. A pesar de ello, los VOR siguen siendo muy útiles por la simplicidad de la señal emitida, ya que se trata de una potente señal analógica modulada

en AM y FM de largo alcance. Además, los VOR también son usados para determinar los puntos de referencia de las aerovías de la aviación comercial.

Sin embargo, en el mundo de la aviación general los VOR juegan un papel muy importante en cuanto a la navegación aérea se refiere, ya que, en vuelos instrumentales y juntamente con las cartas aeronáuticas pertinentes, son la única referencia de la que disponen los pilotos de aeronaves ligeras y de menor envergadura para guiarse por el aire.

En definitiva, el VOR es una radioayuda muy fiable y exacta, que requiere de relativamente poca instrumentación en cabina para interpretar la señal que emite. Además, es muy robusto a los cambios meteorológicos y, dado que la comunicación entre el VOR y la aeronave se hace en LoS o en línea directa entre emisor y receptor, no hay ninguna afectación negativa de la ionosfera sobre las señales (como podría pasar con señales satelitales en ciertas condiciones meteorológicas) ni debido a fenómenos meteorológicos, ni tampoco se reportan reflexiones con el suelo. Sin duda es un pilar clave en el mundo de la aviación por la inmensidad de usos que ofrece.

2.4 Clasificación de los VOR

Históricamente han existido dos tipos de VOR para la aviación civil [6]:

- **Conventional VOR (CVOR).** Estos tipos de VOR fueron el primer tipo de VOR que apareció. Estaban formados por cuatro antenas tipo *loop* de Alford y dos antenas tipo dipolo. Eran un tipo de radioayuda muy imprecisa y susceptible a fallos, la cual fue sustituida pocos años más tarde por los DVOR, o Doppler VOR.
- **Doppler VOR (DVOR).** Estos tipos de VOR son los que se usan en la actualidad, debido a su enorme precisión (error de $<1^\circ$). Usan el principio de funcionamiento del Efecto Doppler para su funcionamiento y tienen un coste relativamente barato de mantenimiento.

Además de estas clases de VOR, también existe una variante militar, llamada VORTAC, que fusiona las características de un VOR mucho más exacto y mejorado y las de un TACAN (sistema de navegación aérea táctica) y cuya representación se muestra en la Figura 2. Estos dispositivos funcionan en la banda de frecuencias de alrededor de 1 GHz y por lo general no son accesibles por las aeronaves de ámbito de la aviación civil.

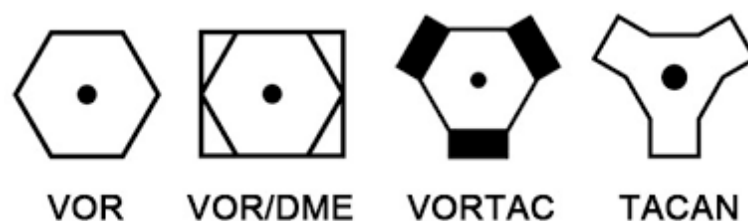


Figura 2. Símbolos de diferentes radioayudas en las cartas aeronáuticas [6].

Finalmente, a pesar de que no se trata en este trabajo debido a las estrictas normativas existentes respecto a la emisión de señales en la banda de frecuencias destinadas a la aviación, cabe destacar la versión más moderna de los VOR, el llamado VOR-DME, de sus

siglas en inglés *Distance Measurment Equipment*. Una aeronave, mediante el envío de pulsos hacia el VOR-DME y las respuestas de este, puede averiguar su posición actual, su velocidad y el tiempo de vuelo hacia el VOR-DME. En la actualidad es difícil encontrar un VOR sin el sistema DME, ya que estas dos tecnologías van prácticamente cogidas de la mano. Sin embargo, mientras que el DME depende de la señal VOR para determinar la posición del avión, el VOR no depende del DME para informar al avión de los radiales por los cuales vuela.

Otro tipo de clasificación se hace en función del rango de cobertura del VOR, que se pueden separar en diferentes grupos:

- **VOR Terminal (T).** Este tipo de VOR se puede localizar en las inmediaciones de un aeropuerto. Son VOR diseñados para dar cobertura a las aeronaves que maniobran alrededor del aeropuerto, ya que emiten poca potencia (hasta 50 W de potencia de transmisión según la OACI). Tienen un rango de hasta 12.000 pies de altitud y cubren una circunferencia de 24 millas náuticas de diámetro.
- **VOR de Baja Altitud (L).** Estos VOR se usan mayoritariamente en rutas, ya que proporcionan mayor rango de cobertura. Concretamente dan servicio hasta los 18.000 pies de altitud y cubren una circunferencia de 40 millas náuticas de diámetro.
- **VOR de Alta Altitud (H).** Estos VOR se usan mayoritariamente en rutas largas y vuelos en altitud de crucero (la altitud de crucero habitual se establece entre los 30.000 pies y los 40.000 pies). Llegan a tener un alcance de una circunferencia de 200 millas náuticas de diámetro y llegan hasta los 45.000 pies de altitud. Según la OACI, emiten una potencia de 200 W.

2.5 Funcionamiento Básico de un VOR

Los VOR proporcionan información a un piloto sobre la posición de su aeronave respecto a dicho VOR, por lo tanto, toda la información que proporciona un VOR es en referencia a él mismo. Dado el funcionamiento de los VOR, estos han de estar debidamente localizados y posicionados en los mapas y en las cartas aeronáuticas y, por consiguiente, todo piloto debe conocer la ubicación de los VOR que sobrevolará en el transcurso de su vuelo para poderse guiar en el aire de forma correcta [7].

Los VOR dividen el mapa donde se encuentran en 360 líneas rectas cuyo origen de todas ellas es el punto donde se localiza el VOR, formando cada una de estas rectas un ángulo de 1 grado respecto a las rectas adyacentes y dividiendo uniformemente los alrededores del VOR, obteniendo así 360 rectas en total. Cada una de estas líneas rectas son las direcciones del espacio en las que el VOR emite señal, y se las conoce como radiales del VOR. Se puede observar la representación de los radiales en la Figura 3.

Los VOR están orientados hacia el norte magnético de la Tierra, es decir, su punto de referencia es el norte magnético del planeta Tierra. Dado que el norte magnético de la Tierra va cambiando de posición con el paso de los años, los VOR deben someterse cada cierto tiempo a trabajos de mantenimiento y calibración para corregir esta desviación [1].

Asimismo, el radial que apunta hacia al norte magnético (también conocido como el radial de referencia) es el radial 0°. El resto de radiales se definen como el ángulo respecto al radial de referencia, es decir, del radial 0°. Los radiales se expresan con números de 3 cifras, así

pues, el rango de radiales se extiende desde el 000 hasta el 360, estos últimos siendo los equivalentes a 0° . Por lo tanto, la información que recibe un piloto que sintoniza el VOR siempre es el radial en el que se encuentra respecto al norte magnético del VOR.

Para conseguir lo anteriormente expuesto, el VOR emite, entre otras, dos señales que son las más características: la señal de referencia y la señal variable. La señal variable es una onda sinusoidal de 30 Hz, que se emite dando una vuelta completa en sentido antihorario por todos los radiales 30 veces por segundo. La idea es análoga a un faro, el cual va dando vueltas de manera uniforme e iluminando en diferentes direcciones para diferentes instantes de tiempo. Por lo tanto, cada vez que la señal variable completa una vuelta entera, habrá recorrido un ciclo o período de la onda de 30 Hz.

Por otro lado, la señal de referencia inicia una señal sinusoidal de fase 0 grados y se emite en todas direcciones, completando así un período de oscilación cada vez que la señal variable completa una vuelta entera, momento en el que vuelve al radial de referencia, el ángulo 0° , como se puede observar en la Figura 3.

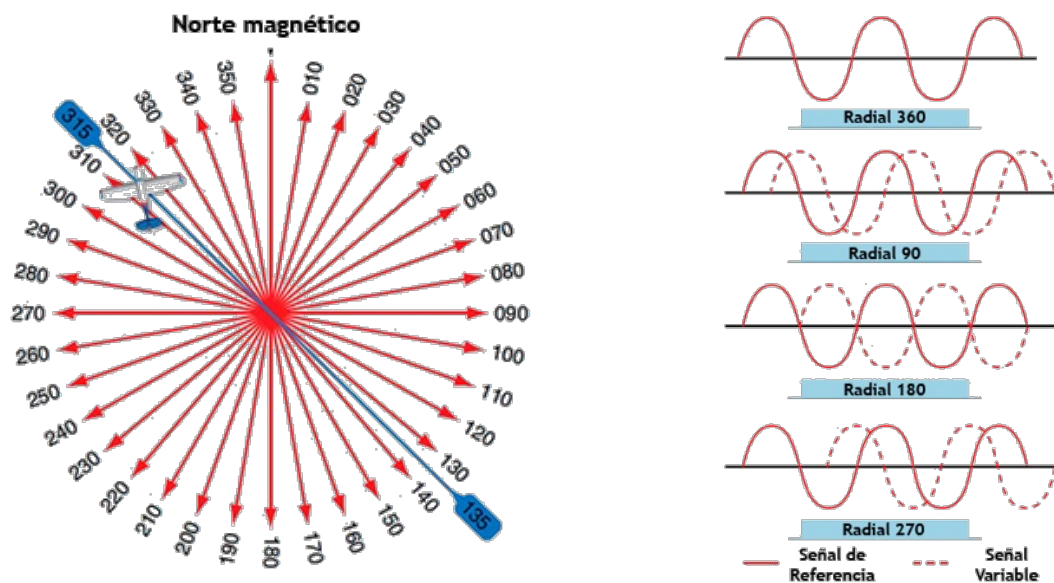


Figura 3. Representación de los radiales de un VOR y el desfase entre la señal de referencia y la señal variable [7].

Así pues, el receptor de un avión recibirá desfasadas la señal de referencia y la señal variable (excepto si se encuentra en el radial $0^\circ/360^\circ$, que entonces recibirá estas dos señales con exactamente la misma fase). Para posicionar el avión en uno de los radiales del VOR, simplemente se debe calcular la diferencia de fase entre las dos señales anteriores, la de referencia y la variable, siendo el máximo desfase entre ambas señales de 360° , es decir, 0° .

En este apartado se ha expuesto el tema de la estructura de la señal de forma muy básica. Para una explicación más técnica y detallada se recomienda leer el capítulo 4 de este trabajo.

3 Conceptos Básicos de las Señales Emitidas por un VOR

En este capítulo se introducen los conceptos básicos y necesarios para entender la estructura de la señal que emiten los VOR y para entender cuál es la forma más adecuada de desmodular la señal e interpretarla.

En concreto, debido que los VOR emiten señales moduladas en AM y FM, que también se pueden expresar como modulaciones de amplitud en cuadratura, en este capítulo se explica, de forma detalla, cada una de estas modulaciones.

Además, los VOR más actuales, los Doppler VOR, basan su funcionamiento en el efecto Doppler, que también se introduce en este capítulo.

3.1 Introducción al Concepto de Modulación

El espectro radioeléctrico es un recurso muy limitado para la cantidad de señales que conviven en el mismo medio, el aire. Para aprovechar al máximo la capacidad de este, la comunidad científica ha tenido que ingeniar técnicas para reducir el ancho de banda de las señales que se emiten, emitir las señales en múltiples rangos de frecuencias y evitar interferencias entre las múltiples señales que viajan simultáneamente. Una de las soluciones a este problema es la modulación.

Con la técnica de modulación se consigue tener bajo control la señal moduladora, que normalmente viaja encima de otra señal, llamada portadora. Esto permite transmitir los datos en bandas de frecuencia diferentes a la banda base, distribuyendo así el espectro radioeléctrico en diversos canales de anchos de banda controlados y generalmente más estrechos con respecto al ancho de banda que se usaría si se enviara las señales sin modular.

Así pues, con la técnica de modulación se consigue disminuir las dimensiones de las antenas, optimizar el ancho de banda utilizado, proteger la información de las degradaciones y definir la calidad de la información transmitida [8].

Por un lado, la modulación se puede llevar a cabo a partir de una señal analógica, es decir, cuando una señal de tiempo continuo queda impresa directamente en la forma de onda de la propia señal, ya sea en la amplitud, la frecuencia o la fase de esta. Por otro lado, también se puede modular una señal digital mediante una modulación digital, donde los símbolos (o bits) se transforman en señales compatibles con el canal de comunicaciones. En ambos casos, debido a que vivimos en un mundo analógico y de tiempo continuo, la señal emitida siempre tendrá una naturaleza analógica, pero se diferenciarán por el tipo de modulación utilizada.

Las tendencias actuales en el ámbito de la radiodifusión apuntan hacia las modulaciones digitales, ya que son más robustas, incluyen métodos de corrección de errores, los datos se pueden codificar o encriptar y se reciben exactamente igual a como se han enviado. No obstante, muchas aplicaciones hoy en día usan las modulaciones analógicas debido a su facilidad de tratamiento y por razones históricas, entre otras razones.

En cuanto al ámbito de la aviación, muchas de las radiocomunicaciones todavía se hacen sobre un ecosistema analógico. Esto es debido a la antagonista relación que hay en el mundo

aeronáutico de actualizar o modificar sistemas que funcionan correctamente y no acostumbran a dar problemas significantes.

Así pues, las radiocomunicaciones analógicas son uno de estos sistemas mencionados en el anterior párrafo. Actualizar el modo en el que una aeronave se comunica con otras aeronaves o con estaciones terrenas supondría un cambio extraordinario a nivel mundial que acarrearía un riesgo que, por el momento, las agencias de seguridad aérea de todo el mundo no quieren correr.

3.2 Modulación de Amplitud en Cuadratura

Una modulación tanto analógica como digital consiste en modificar la amplitud, la fase o la frecuencia de una señal portadora según el tipo de modulación [9]. De hecho, una señal modulada se puede expresar como:

$$\begin{aligned}
 y(t) &= A(t) \cdot \cos(2\pi f_c t + \theta(t)) = \\
 &= A(t) \cdot \cos(\theta(t)) \cdot \cos(2\pi f_c t) + A(t) \cdot \sin(\theta(t)) \cdot \sin(2\pi f_c t) = \\
 &= I(t) \cdot \cos(2\pi f_c t) + Q(t) \cdot \sin(2\pi f_c t) = \\
 &= \text{Re}\{(I(t) + jQ(t)) \cdot e^{j2\pi f_c t}\}
 \end{aligned}
 \tag{1}$$

Por lo tanto, esta señal se puede generar con un modulador IQ, es decir, multiplicando con un tono (coseno) la componente con fase I(t) y sumando la componente en cuadratura Q(t) multiplicada por el tono desfasado 90° (seno), tal y como se puede observar en el esquema de la Figura 4. La frecuencia central del canal la determina la portadora del oscilador del modulador y la multiplicación se realiza con un mezclador.

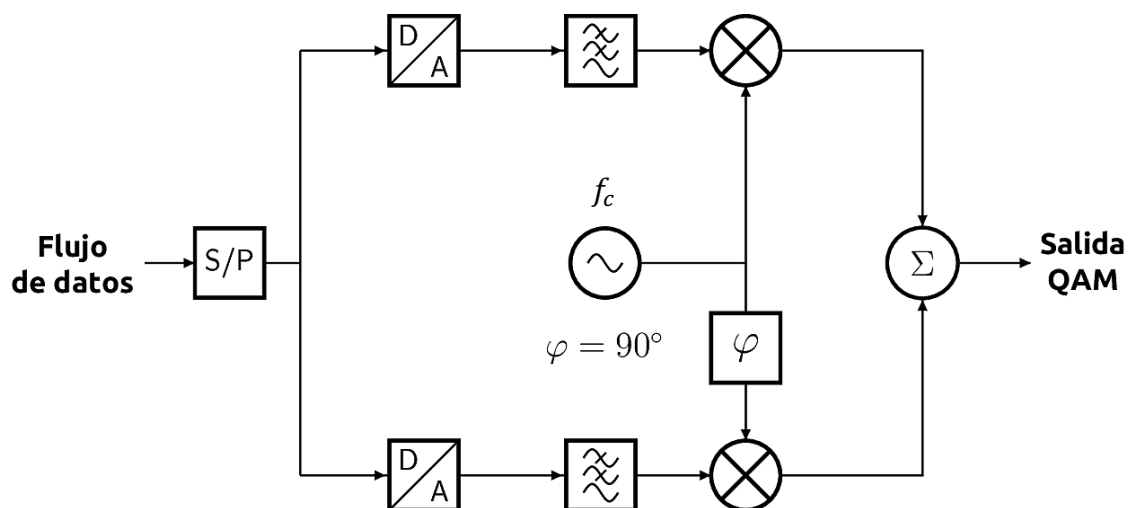


Figura 4. Esquema de modulación IQ [10].

Para desmodular se puede hacer el proceso inverso a la modulación, donde se obtienen las expresiones matemáticas 2 y 3, correspondientes a las dos componentes que constituyen este

tipo de modulaciones. Para eliminar las componentes de frecuencia doble es necesario el uso de filtros paso bajo, tal y como se puede observar en el esquema de la Figura 5.

Para la componente I(t):

$$\begin{aligned}
 I(t) &= 2 \cdot LPF\{y(t) \cdot \cos(2\pi f_c t)\} = \\
 &= 2 \cdot LPF\left\{\frac{1}{2} \cdot I(t) + \frac{1}{2} \cdot I(t) \cdot \cos(2 \cdot 2\pi f_c t) + Q(t) \cdot \cos(2\pi f_c t) \cdot \sin(2\pi f_c t)\right\} = \quad (2) \\
 &= 2 \cdot LPF\left\{\frac{1}{2} \cdot I(t) + \frac{1}{2} \cdot I(t) \cdot \cos(2 \cdot 2\pi f_c t) + \frac{1}{2} \cdot Q(t) \cdot \sin(2 \cdot 2\pi f_c t)\right\} = I(t)
 \end{aligned}$$

Para la componente Q(t):

$$\begin{aligned}
 Q(t) &= 2 \cdot LPF\{y(t) \cdot \sin(2\pi f_c t)\} = \\
 &= 2 \cdot LPF\left\{I(t) \cdot \cos(2\pi f_c t) \cdot \sin(2\pi f_c t) + \frac{1}{2} \cdot Q(t) - \frac{1}{2} \cdot Q(t) \cdot \sin(2 \cdot 2\pi f_c t)\right\} = \quad (3) \\
 &= 2 \cdot LPF\left\{\frac{1}{2} \cdot I(t) \cdot \sin(2 \cdot 2\pi f_c t) + \frac{1}{2} \cdot Q(t) - \frac{1}{2} \cdot Q(t) \cdot \sin(2 \cdot 2\pi f_c t)\right\} = Q(t)
 \end{aligned}$$

Que en definitiva se puede expresar de la siguiente manera:

$$x(t) = I(t) + jQ(t) = LPF\{y(t) \cdot e^{-j2\pi f_c t}\} \quad (4)$$

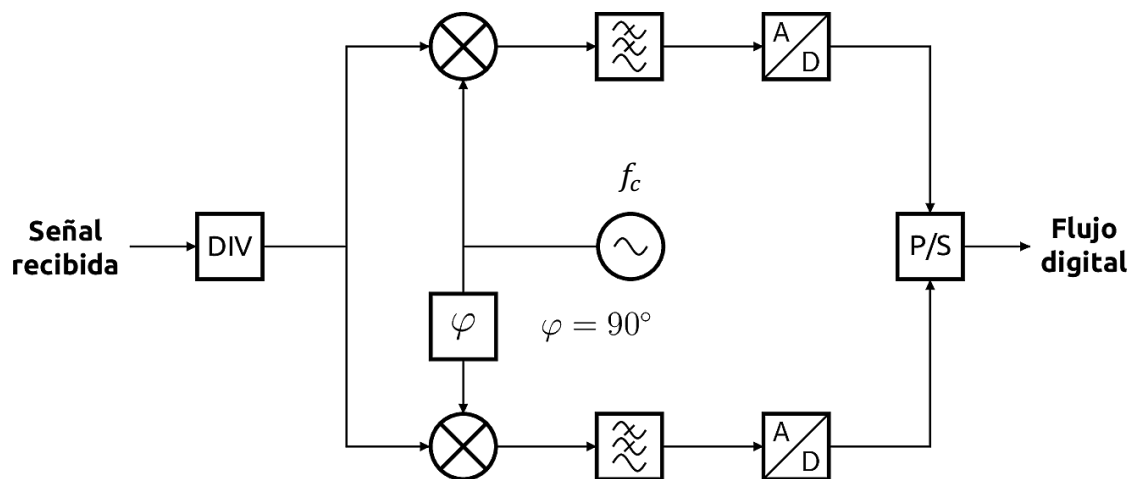


Figura 5. Esquema de demodulación IQ [10].

Para las dos modulaciones analógicas que se van a tratar en este trabajo (es decir, la modulación FM y AM que modulan la señal emitida por los VOR), se exponen las expresiones equivalentes de la señal de las componentes de fase y cuadratura, las cuales se definen en la Tabla 1:

Modulación	I(t)	Q(t)
AM	$A(1 + m \cdot x(t))$	0
FM	$A \cdot \cos\left(2\pi K_0 \int x(t) dt\right)$	$A \cdot \sin\left(2\pi K_0 \int x(t) dt\right)$

Tabla 1. Señal en fase y cuadratura en función del tipo de modulación analógica [9].

3.3 Modulación en Amplitud AM

La modulación en amplitud es una modulación analógica lineal, es decir, varia en las características de la amplitud de la señal manteniendo la fase constante [11, 12]. En su forma más básica está constituida por una mezcla superheterodina entre una señal con información y una señal portadora que tiene dos bandas laterales en el dominio de la frecuencia. Este proceso recibe el nombre de doble banda lateral, cuyo objetivo es subir la señal de banda base a la frecuencia de la señal portadora para que sea transmisible a través de un medio como el aire y sea menos vulnerable al ruido y a las interferencias.

Como en cualquier modulación, se tiene la señal que contiene la información (espectro paso bajo o en banda base) y la señal portadora, llamada *carrier*, que transporta la señal original.

3.3.1 Análisis de una Señal AM en el Dominio Temporal

En el dominio temporal, la señal AM presenta una forma característica mostrada en la Figura 6, donde se puede apreciar la fluctuación de la amplitud de la señal portadora [13].

Matemáticamente se expresa esta señal como en la expresión matemática numero 9, cuya demostración parte de una señal cualquiera cuya amplitud está acotada entre -1 y 1:

$$|x(t)| \leq 1 \forall t \quad (7)$$

Siendo la señal portadora:

$$A_c \cdot \cos(\omega_c t + \theta_c) \quad (8)$$

Por lo tanto, la señal modulada obtenida es:

$$\begin{aligned} s(t) &= A_c \cdot [1 + m \cdot x(t)] \cdot \cos(\omega_c t + \theta_c) = \\ &= A_c \cdot \cos(\omega_c t + \theta_c) + [m \cdot x(t)] \cdot A_c \cdot \cos(\omega_c t + \theta_c) \end{aligned} \quad (9)$$

Donde A_c es la amplitud de la portadora en V, ω_c es la frecuencia de la portadora en rad/s, θ_c es la fase de la portadora en rad y m es el índice de modulación que viene dado por:

$$m = \frac{E_{max} - E_{min}}{E_{max} + E_{min}} \forall 0 < m < 1 \quad (10)$$

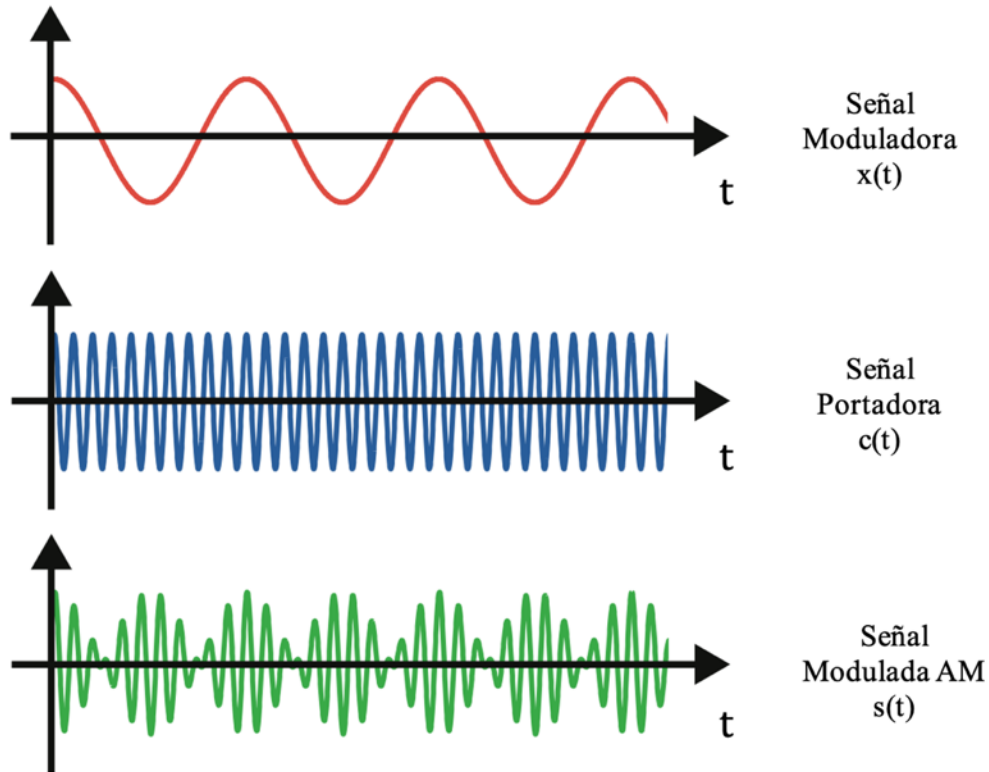


Figura 6. Representación temporal de una señal modulada en AM.

3.3.2 Análisis de una Señal AM en el Dominio Frecuencial

Se puede obtener su representación frecuencial haciendo la transformada de Fourier de la señal temporal anterior, cuya representación gráfica se muestra en la Figura 7. Así pues, partiendo de la transformada de Fourier de la expresión del coseno:

$$A_c \cdot \cos(\omega_c t + \theta_c) \leftrightarrow A_c \pi \cdot [\delta(\omega - \omega_c) + \delta(\omega + \omega_c)] \quad (11)$$

La expresión en el dominio frecuencial de la señal modulada es:

$$S(\omega) = A_c \pi [\delta(\omega - \omega_c) + \delta(\omega + \omega_c)] + m B A_c \pi [X(\omega - \omega_c) + X(\omega + \omega_c)] \quad (12)$$

Desarrollando la expresión anterior:

$$S(\omega) = A_c \pi [\delta(\omega - \omega_c) + \delta(\omega + \omega_c)] + m B A_c \pi \cdot \left\{ \begin{array}{l} \delta[\omega - (\omega_c + \omega_m)] + \delta[\omega + (\omega_c + \omega_m)] + \\ + \delta[\omega - (\omega_c - \omega_m)] + \delta[\omega + (\omega_c - \omega_m)] \end{array} \right\} \quad (13)$$

En definitiva, el espectro de la señal AM consiste en el espectro de la señal moduladora desplazada a la frecuencia de la portadora, más el término de la portadora. Por lo tanto, la señal AM es una señal pasa banda en la frecuencia de la portadora, cuyo ancho de banda es el doble de la señal moduladora.

$$\Delta B = 2 \cdot f_m \quad (14)$$

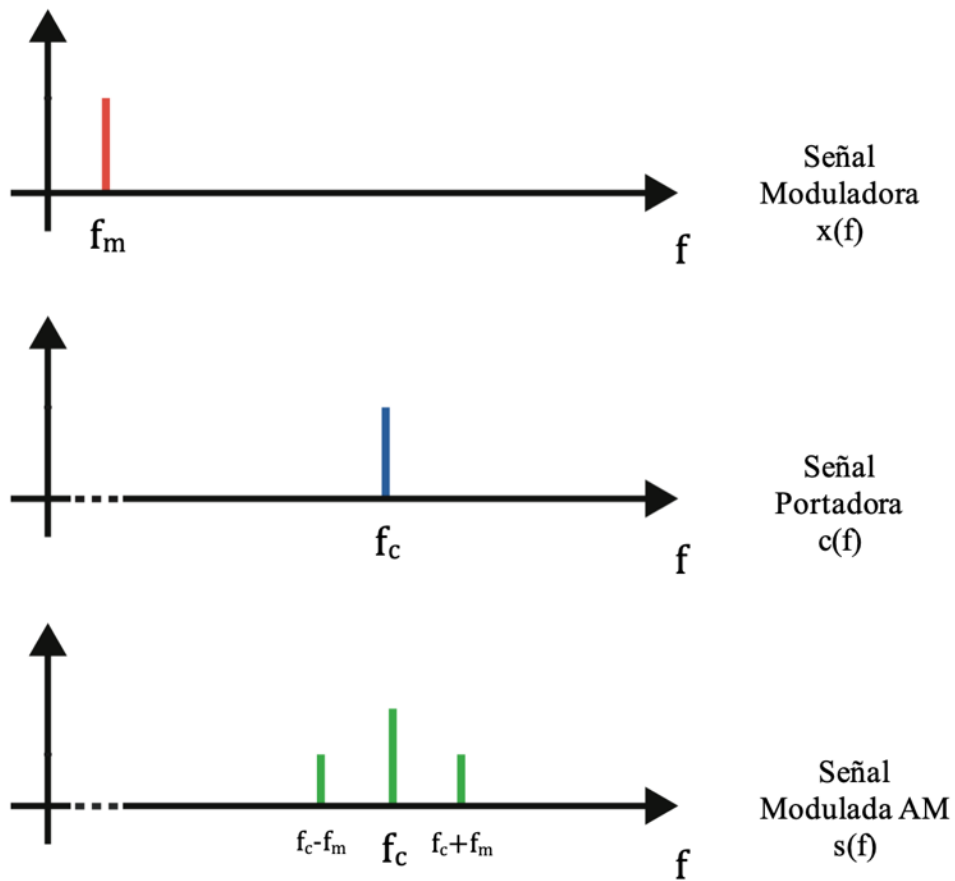


Figura 7. Representación frecuencial de una señal modulada en AM.

En definitiva, se puede expresar de forma esquemática el proceso de modular en amplitud una señal cualquiera de la manera en la que se muestra en la Figura 8. Es útil entender el proceso de generación de una señal AM para posteriormente conocer los pasos para desmodular esta señal.

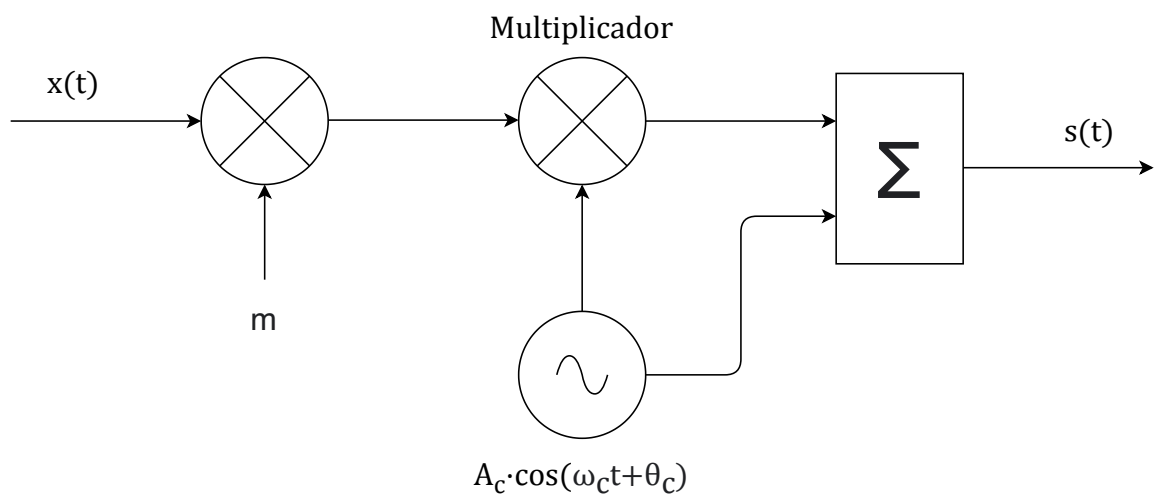


Figura 8. Esquema de generación de una señal AM.

3.4 Modulación en Frecuencia FM

La modulación en frecuencia es una modulación analógica angular, es decir, varía en las características de la frecuencia de la señal manteniendo la amplitud constante [12, 14]. Entre todas las modulaciones analógicas, la modulación en frecuencia es la más robusta al ruido, por lo que la hace una modulación fiable que se ha usado en muchas aplicaciones desde que apareció en el año 1933.

Como en cualquier modulación, se tiene la señal que contiene la información (espectro paso bajo o en banda base) y la señal portadora, llamada “*carrier*”, que transporta la señal original.

3.4.1 Análisis de una Señal FM en el Dominio Temporal

Una forma simple de analizar matemáticamente un modulador FM es basando la explicación en el funcionamiento de un VCO (*Voltage Controlled Oscillator*) [15]. Un VCO es un oscilador que usa un varactor, es decir, usa un diodo semiconductor cuya capacidad se ajusta variando la tensión inversa aplicada en la unión del diodo. Así pues, la señal FM puede expresarse de la siguiente forma:

$$x(t) = A \cdot \cos [\phi(t)] \quad (15)$$

Teniendo la siguiente señal portadora:

$$c(t) = A_c \cdot \cos (\omega_c t) \quad (16)$$

La señal de control del VCO se multiplica por una constante k , la cual representa la relación entre el voltaje y la frecuencia, medida en Hz/V. Así pues, el producto de la señal de entrada y la constante se integra de manera que se cambia la fase 90 grados.

$$\phi(t) = k \int_{-\infty}^t x(t) dt \quad (17)$$

La salida del VCO está diseñado para tener una frecuencia fija y una amplitud constante. La fase de la señal de salida viene determinada por el valor instantáneo de la expresión anterior. Por lo tanto, la señal obtenida a la salida del VCO ha de ser idéntica a la siguiente señal.

$$v(t) = A_0 \cdot \cos[2\pi f_0 t + \phi(t)] = A_0 \cdot \cos \left(\omega_0 t + K_0 \int_{-\infty}^t x(t) dt \right) \quad (18)$$

Sin embargo, cuando a la entrada se le inyecta una señal que transporta información, la constante k puede substituirse por la constante de desviación de frecuencia característica de la modulación FM, por lo tanto, la expresión anterior queda de la siguiente forma:

$$\phi(t) = 2\pi K_0 \int_{-\infty}^t x(t) dt \quad (19)$$

Y si finalmente se sustituyen los valores de amplitud y frecuencia por los valores de la señal portadora:

$$s(t) = A_c \cdot \cos \left[\omega_c t + 2\pi K_0 \int_{-\infty}^t x(t) dt \right] \quad (20)$$

Hasta aquí se ha presentado el modelo matemático general que representa una señal FM. La expresión anteriormente obtenida se puede generalizar para cualquier señal de entrada [16, 17]. Para demostrarlo se presenta debajo el siguiente caso, donde la señal de entrada es una señal cosenoidal.

$$x(t) = A_m \cdot \cos(\omega_m t) \quad (21)$$

Sabiendo que la señal portadora será la siguiente:

$$c(t) = A_c \cdot \cos(\omega_c t) \quad (22)$$

Y sabiendo que:

$$\phi(t) = \int \omega(t) dt = \omega_c t + \frac{2\pi \cdot K_0 \cdot A_m}{\omega_m} \cdot \sin(\omega_m t) \quad (23)$$

La señal modulada es:

$$s(t) = A_c \cdot \cos \left[\omega_c t + \frac{2\pi \cdot K_0 \cdot A_m}{\omega_m} \cdot \sin(\omega_m t) \right] \quad (24)$$

Donde A_c es la amplitud de la portadora en V, ω_c es la frecuencia de la portadora en rad/s, K_0 es la constante de desviación de frecuencia en Hz/V, A_m es la amplitud de la moduladora en V y ω_m es la frecuencia de la moduladora en rad/s.

Y suponiendo:

$$\beta = \frac{2\pi \cdot K_0 \cdot A_m}{\omega_m} \equiv \text{profundidad o índice de modulación} \quad (25)$$

Se puede arreglar la operación aplicando la siguiente ley trigonométrica:

$$\cos(a \pm b) = \cos(a) \cdot \cos(b) \mp \sin(a) \cdot \sin(b) \quad (26)$$

Y se obtiene la siguiente expresión:

$$s(t) = A \cdot \{ \cos(\omega_c t) \cdot \cos[\beta \cdot \sin(\omega_m t)] - \sin(\omega_c t) \cdot \sin[\beta \cdot \sin(\omega_m t)] \} \quad (27)$$

Notar en la Figura 9 que la señal resultante de una modulación FM se puede entender como la señal portadora centrada en una determinada frecuencia, donde las constantes variaciones de amplitud de la señal moduladora se transforman en variaciones en la frecuencia.

Por consiguiente, a partir de estas definiciones se puede extraer una variable llamada desviación en frecuencia, la cual mide el rango de frecuencias por el que la señal FM va fluctuando alrededor de la portadora. Este término sigue la siguiente expresión:

$$\Delta f = \beta \cdot f_m \quad (28)$$

Y, en ocasiones, se puede especificar el parámetro de porcentaje de modulación, que indica la relación entre los rangos de frecuencias máximos permitidos y los rangos de frecuencias realmente usados. Se especifica como:

$$\% \text{ modulación} = \frac{\Delta f_{real}}{\Delta f_{max}} \quad (29)$$



Figura 9. Representación temporal de una señal modulada en FM.

3.4.2 Las Funciones de Bessel

Se puede observar que en la definición matemática de la señal resultante de una modulación FM se han obtenido expresiones del tipo $\cos(\cos x)$, $\cos(\sin x)$, $\sin(\cos x)$ i $\sin(\sin x)$. La solución a las expresiones se puede expresar como una serie infinita basada en las funciones de *Bessel*.

En realidad, para entender el concepto de la modulación FM no es necesario entrar en el tratamiento de estas funciones, basta con mencionar cómo desarrollarlas [16]. Para los diferentes posibles casos se hacen las siguientes aproximaciones. Notar que todas ellas dependen del índice de modulación β .

$$\cos[\beta \cdot \sin(\omega_m t)] = \sum_{n=-\infty}^{\infty} J_n(\beta) \cdot \cos(n\omega_m t) \quad (30)$$

$$\sin[\beta \cdot \sin(\omega_m t)] = \sum_{n=-\infty}^{\infty} J_n(\beta) \cdot \cos(n\omega_m t) \quad (31)$$

$$\cos[\beta \cdot \cos(\omega_m t)] = \sum_{n=-\infty}^{\infty} J_n(\beta) \cdot \cos\left[n\left(\omega_m t + \frac{\pi}{2}\right)\right] \quad (32)$$

$$\sin[\beta \cdot \cos(\omega_m t)] = \sum_{n=-\infty}^{\infty} J_n(\beta) \cdot \sin\left[n\left(\omega_m t + \frac{\pi}{2}\right)\right] \quad (33)$$

La señal resultante tiene un número infinito de componentes, y cada componente impar de la banda lateral inferior estará desfasada 180° respecto a las demás componentes. Esto es siguiendo la expresión mostrada a continuación.

$$J_n(\beta) = (-1)^n \cdot J_{-n}(\beta) \quad (34)$$

Para determinar el valor de $J_n(\beta)$ se debe conocer el índice de modulación. Para cada valor del índice de modulación existe una función de Bessel diferente, por lo tanto, distintos valores de J. Estos valores pueden obtenerse fácilmente de la tabla de la Figura 9, y su representación gráfica se muestra en la Figura 10.

El número de coeficientes en la aproximación es infinito. Sin embargo, existe un límite en el cual los valores numéricos de las J son tan pequeños y próximos a cero que se pueden despreciar sin apenas afectar al resultado definitivo, que es la señal está por debajo de 0,01. Notar que al truncar los valores se está cometiendo un pequeño error que, aunque es despreciable, conlleva a que se considere una aproximación casi exacta.

β	ORDEN DE LA FUNCIÓN																
	Portadora	J0	J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14	J15
0	1,00	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
0,1	1,00	0,05	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
0,2	0,99	0,10	~	~	~	~	~	~	~	~	~	~	~	~	~	~	~
0,25	0,98	0,12	0,01	~	~	~	~	~	~	~	~	~	~	~	~	~	~
0,5	0,94	0,24	0,03	~	~	~	~	~	~	~	~	~	~	~	~	~	~
0,75	0,86	0,35	0,07	0,01	~	~	~	~	~	~	~	~	~	~	~	~	~
1	0,77	0,44	0,11	0,02	~	~	~	~	~	~	~	~	~	~	~	~	~
1,5	0,51	0,56	0,23	0,06	0,01	~	~	~	~	~	~	~	~	~	~	~	~
2	0,22	0,58	0,35	0,13	0,03	0,01	~	~	~	~	~	~	~	~	~	~	~
2,4	0,00	0,52	0,43	0,20	0,06	0,02	~	~	~	~	~	~	~	~	~	~	~
3	-0,26	0,34	0,49	0,31	0,13	0,04	0,01	~	~	~	~	~	~	~	~	~	~
4	-0,40	-0,07	0,36	0,43	0,28	0,13	0,05	0,02	~	~	~	~	~	~	~	~	~
5	-0,18	-0,33	0,05	0,36	0,39	0,26	0,13	0,05	0,02	0,01	~	~	~	~	~	~	~
6	0,15	-0,28	-0,24	0,11	0,36	0,36	0,25	0,13	0,06	0,02	0,01	~	~	~	~	~	~
7	0,30	0,00	-0,30	-0,17	0,16	0,35	0,34	0,23	0,13	0,06	0,02	0,01	~	~	~	~	~
8	0,17	0,23	-0,11	-0,29	-0,11	0,19	0,34	0,32	0,22	0,13	0,06	0,03	0,01	~	~	~	~
9	-0,09	0,25	0,14	-0,18	-0,27	-0,06	0,20	0,33	0,31	0,21	0,12	0,06	0,03	0,01	~	~	~
10	-0,25	0,04	0,25	0,06	-0,22	-0,23	-0,01	0,22	0,32	0,29	0,21	0,12	0,06	0,03	0,01	~	~
11	-0,17	-0,18	0,14	0,23	-0,02	-0,24	-0,20	0,02	0,22	0,31	0,28	0,20	0,12	0,06	0,03	0,01	0,01
12	0,05	-0,22	-0,08	0,20	0,18	-0,07	-0,24	-0,17	0,05	0,23	0,30	0,27	0,20	0,12	0,07	0,07	0,03
13	0,21	-0,07	-0,22	0,00	0,22	0,13	-0,12	-0,24	-0,14	0,07	0,23	0,29	0,26	0,19	0,12	0,07	0,07
14	0,17	0,13	-0,15	-0,18	0,08	0,22	0,08	-0,15	-0,23	-0,11	0,09	0,24	0,29	0,25	0,19	0,12	0,12
15	-0,01	0,21	0,04	-0,19	-0,12	0,13	0,21	0,03	-0,17	-0,22	-0,09	0,10	0,24	0,28	0,25	0,18	0,18

Figura 10. Tabla de los coeficientes de Bessel en función del índice de modulación [18].

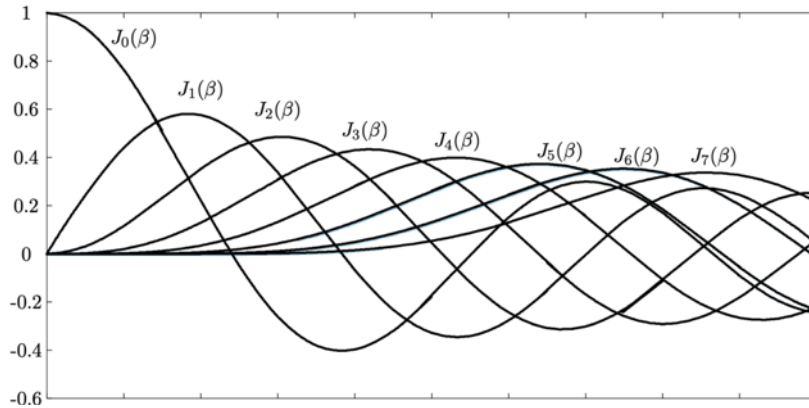


Figura 11. Representación gráfica de las funciones de Bessel [15].

Así pues, volviendo a la señal anterior, se sustituyen las funciones trigonométricas por las expresiones de las funciones de Bessel y se obtiene una señal como la que se presenta debajo.

$$s(t) = A_c \cdot \cos[\omega_c t + \beta \cdot \sin(\omega_m t)]$$

$$s(t) = A_c \cdot \sum_{n=-\infty}^{\infty} J_n(\beta) \cdot \cos(\omega_c t + n\omega_m t) \tag{35}$$

Si se desarrolla el sumatorio, se obtiene una aproximación casi exacta de la señal FM resultante:

$$s(t) = A_c \cdot J_0(\beta) \cdot \cos(\omega_c t) + J_1(\beta) \cdot [\cos(\omega_c + \omega_m) t - \cos(\omega_c - \omega_m) t] + J_2(\beta) \cdot [\cos(\omega_c + 2\omega_m) t + \cos(\omega_c - 2\omega_m) t] + \dots \tag{36}$$

3.4.3 Análisis de una Señal FM en el Dominio Frecuencial

Dado que la señal modulada en FM cambia constantemente su frecuencia, esto provocará que se generen infinitos armónicos. No obstante, los armónicos más grandes tienen una potencia muy baja, es por esto por lo que para el tratamiento de estas señales generalmente se cogen los armónicos de mayor potencia, es decir, los más cercanos a la fundamental. Se puede observar la representación gráfica de la señal FM en la Figura 12.

De hecho, existen múltiples expresiones para estimar el ancho de banda más adecuado para la demodulación, aunque generalmente se suele usar la regla de Carson:

$$B_T = 2 \cdot (f_{d_{max}} + f_{m_{max}}) \tag{37}$$

Otra forma popular de determinar el ancho de banda del espectro es representando todos los coeficientes J de las funciones de Bessel, donde se desprecian los coeficientes que son mas pequeños que un 1 % de $J_0(\beta)$ o menores a 0,01. Por lo tanto, es lógico que el número de coeficientes variará en función del índice de modulación y, en consecuencia, el ancho de banda se modificará siguiendo la siguiente expresión:

$$B_T = 2 \cdot f_m \cdot n_{coeficientes} \tag{38}$$

En definitiva, para obtener la representación frecuencial, se puede hacer mediante la transformada de Fourier de la señal temporal anteriormente planteada. Partiendo de la transformada de Fourier de la expresión del coseno:

$$A_c \cdot \cos(\omega_c t + \theta_c) \leftrightarrow A_c \pi \cdot [\delta(\omega - \omega_c) + \delta(\omega + \omega_c)] \quad (39)$$

La expresión en el dominio frecuencial de la señal modulada es, manteniendo la fase constante:

$$s(t) = \frac{A_c}{2} \cdot \{ J_0(\beta) \cdot [\delta(\omega + \omega_c) + \delta(\omega - \omega_c)] + J_1(\beta) \cdot [\delta(\omega + (\omega_c + \omega_f)) + \delta(\omega + (\omega_c - \omega_f)) + \delta(\omega - (\omega_c + \omega_f)) + \delta(\omega - (\omega_c - \omega_f))] + J_2(\beta) \cdot [\delta(\omega + (\omega_c + 2\omega_f)) + \delta(\omega + (\omega_c - 2\omega_f)) + \delta(\omega - (\omega_c + 2\omega_f)) + \delta(\omega - (\omega_c - 2\omega_f))] + \dots \} \quad (40)$$

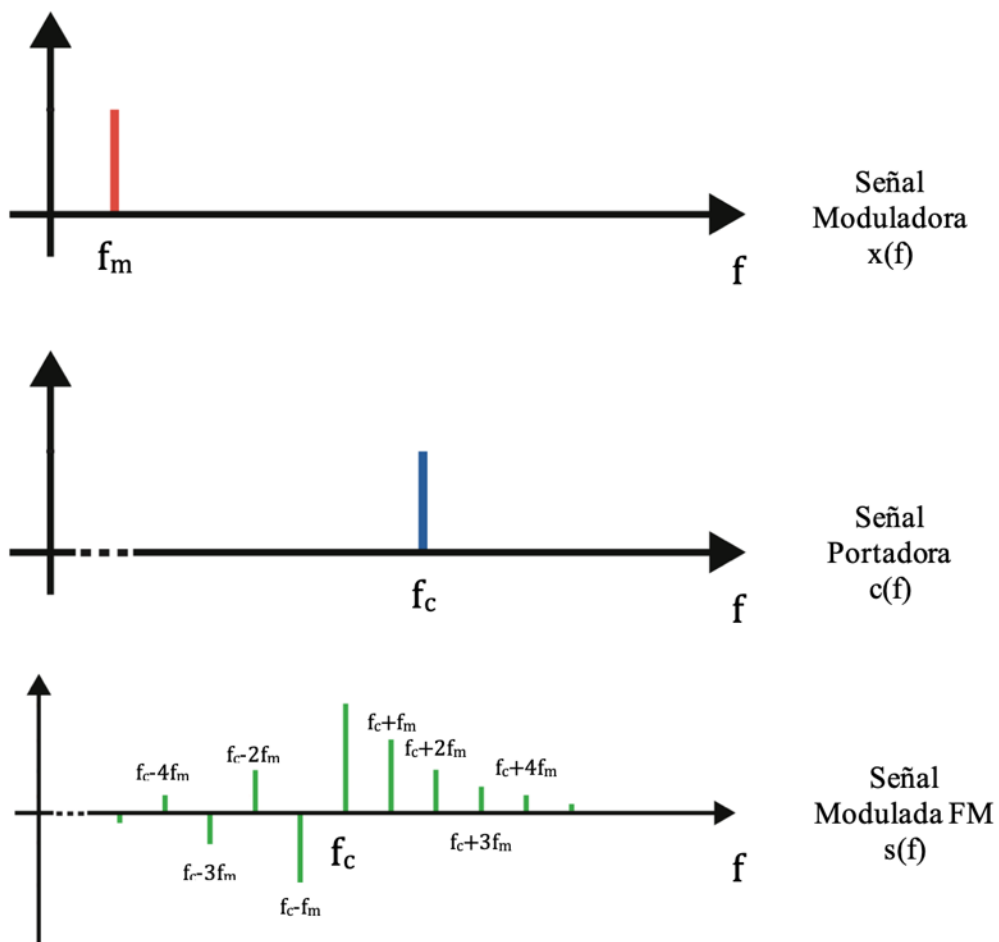


Figura 12. Representación frecuencial de una señal modulada en FM.

En definitiva, el espectro de la señal FM consiste en el espectro de la señal moduladora representada infinitas veces a intervalos constantes de frecuencia, cuyo valor del intervalo lo marca la frecuencia de la moduladora. Esta señal está presente a ambos lados de la señal portadora, por consiguiente, la señal FM es una señal pasa banda en la frecuencia de la portadora cuyo ancho de banda teórico es infinito. Sin embargo, dado que los coeficientes

más elevados son de muy baja potencia, se pueden despreciar según las reglas descritas en los apartados anteriores.

Además, se puede expresar de forma esquemática el proceso de modular en frecuencia una señal cualquiera de la manera en la que se muestra en la Figura 13. Es útil entender el proceso de generación de una señal FM para posteriormente conocer los pasos para desmodular esta señal.

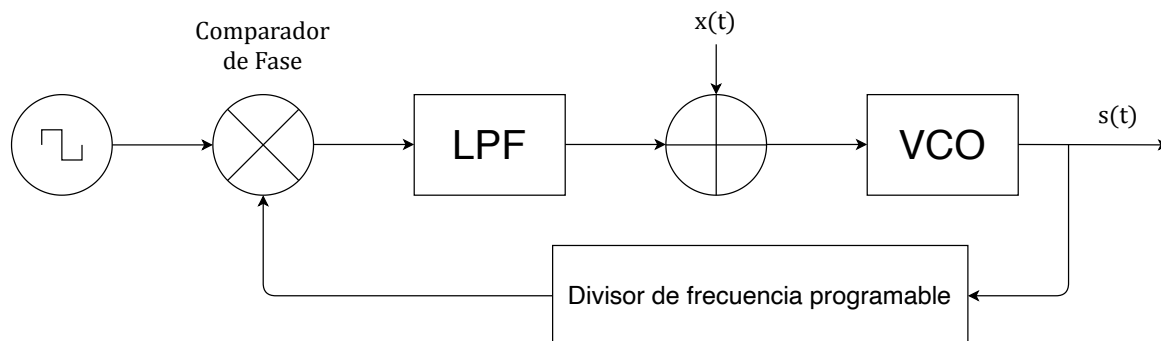


Figura 13. Esquema de generación de una señal FM.

3.5 El Efecto Doppler

El efecto Doppler, llamado así por el físico y matemático austriaco Christian Doppler, quien presentó este curioso efecto de las ondas a la comunidad científica el año 1842, surgió al querer explicar la razón por la que algunas estrellas tenían colores cálidos y otras estrellas colores más oscuros cuando el famoso científico las observaba.

Fue entonces cuando se descubrió que, si las estrellas se acercan a la Tierra (es decir, al observador) se aprecian de un color rojizo (se capta una frecuencia más grande), mientras que las estrellas que se alejan de la Tierra se aprecian con colores más azulados (se capta una frecuencia más pequeña).

Así pues, se daba a entender que existía una relación entre la variación de la velocidad de los emisores y receptores de radiación y entre la variación de la frecuencia, la cual Doppler consiguió demostrar.

Por lo tanto, el efecto Doppler es una de las bases principales para entender muchas de las comunicaciones mediante ondas, ya que explica el comportamiento de la frecuencia de una onda en función del movimiento tanto del emisor como del receptor, y la velocidad relativa entre estos.

El típico ejemplo que se usa para referenciar este efecto es el de una ambulancia con la sirena encendida como fuente emisora pasando muy rápido por delante del observador o receptor. El efecto se puede percibir mientras esta ambulancia se acerca y se aleja del observador: al acercarse hacia el observador, este recibirá un sonido más agudo debido a que la longitud de onda se estrecha, mientras que, al alejarse del observador, este recibirá un sonido más grave debido a que la longitud de onda se ensancha. También se puede extrapolar este ejemplo para la radiación electromagnética.

De forma matemática, el efecto Doppler se puede expresar en función del movimiento del emisor y del receptor y la dirección en la que se mueven. Estas expresiones son las que se exponen en la Tabla 2.

Emisor	Receptor	Expresión matemática	Frecuencia recibida
Quieto	Quieto	$f' = f$	Sin cambios (41)
Quieto	Movimiento en dirección al emisor	$f' = f \cdot \left(\frac{c + v_r}{c}\right)$	Más alta (42)
Quieto	Movimiento en dirección opuesta al emisor	$f' = f \cdot \left(\frac{c - v_r}{c}\right)$	Más baja (43)
Movimiento en dirección al receptor	Quieto	$f' = f \cdot \left(\frac{c}{c - v_e}\right)$	Más alta (44)
Movimiento en dirección opuesta al receptor	Quieto	$f' = f \cdot \left(\frac{c}{c + v_e}\right)$	Más baja (45)
Movimiento en dirección al receptor	Movimiento en dirección al emisor	$f' = f \cdot \left(\frac{c + v_r}{c - v_e}\right)$	Muy alta (46)
Movimiento en dirección opuesta al receptor	Movimiento en dirección opuesta al emisor	$f' = f \cdot \left(\frac{c - v_r}{c + v_e}\right)$	Muy baja (47)
Movimiento en dirección al receptor	Movimiento en dirección opuesta al emisor	$f' = f \cdot \left(\frac{c + v_r}{c + v_e}\right)$	Si $v_r = v_e$, sin cambios (48)
Movimiento en dirección opuesta al receptor	Movimiento en dirección al emisor	$f' = f \cdot \left(\frac{c - v_r}{c - v_e}\right)$	Si $v_r = v_e$, sin cambios (49)

Tabla 2. Expresiones del efecto Doppler en función del movimiento del emisor y el receptor.

Donde f' es la frecuencia percibida en el receptor en Hz, f es la frecuencia emitida en el transmisor en Hz, c es la velocidad de la luz en m/s, v_e es la velocidad del emisor en m/s y v_r es la velocidad de receptor en m/s.

4 La Señal de los VOR

En este capítulo, a diferencia del capítulo 3, se introducen los conceptos técnicos que corresponden a la señal emitida por los VOR y se concreta cómo se construye la señal final.

Así pues, en este capítulo se exponen conceptos que son fundamentales para entender la forma en la que trabaja un VOR, que es necesario conocer para poder entender qué pasos serán los convenientes para su correcta demodulación y cómo sintetizar la información que estas señales proporcionan.

4.1 Estructura de la Señal Emitida por los VOR

La operación de un VOR (concretamente de un Doppler VOR, que son los utilizados actualmente) está basada en el cálculo de la diferencia de fase entre dos señales de 30 Hz, moduladas en AM para la señal de referencia y en FM para la señal variable [1, 2, 8, 20 – 23, 26].

Por lo que a la señal de referencia se refiere, esta se obtiene modulando en amplitud sobre la señal portadora una señal de 30 Hz, que se radia de forma omnidireccional a través de la antena situada en el centro del VOR. En tal efecto, un avión recibe la misma señal de referencia independientemente de su posición respecto al VOR, dado que la transmisión de esta señal es uniforme para cualquier azimut.

Por otro lado, la señal variable se obtiene modulando en frecuencia una señal subportadora centrada en ± 9960 Hz, que esta última, a su vez, modula en amplitud la señal portadora principal. Esta señal se emite a través del *array* de 48 antenas exteriores que forman una circunferencia de 13,4 metros de diámetro (44 pies), donde dos antenas diametralmente opuestas en un instante de tiempo dado radian simulando una señal rotatoria, girando en sentido contrario a las agujas del reloj 30 veces por segundo, es decir, con una frecuencia de 30 Hz. En este caso, una antena radiará el espectro de la banda lateral superior y la opuesta radiará el espectro de la banda lateral inferior, o viceversa.

Las antenas que forman *arrays* (menos comúnmente llamadas antenas de arreglo de fase) son un grupo de antenas que están conectadas entre ellas, funcionando como una sola antena, cuyo haz y dirección puede cambiarse electrónicamente sin tener que mover físicamente ninguna de las antenas de dicho *array*. Este apartado explicado en 4.2.

Por lo tanto, en función de la posición del avión con respecto al VOR, el avión recibirá la señal variable sinusoidal de 30 Hz con un desfase en grados respecto a la señal de referencia igual al número del radial en el que se encuentre.

Para la emisión y recepción de la señal variable, teniendo en cuenta su desfase, se usa el principio básico del efecto Doppler. Las antenas dispuestas en forma de circunferencia simulan que giran de forma análoga al movimiento de un faro. Dicho giro se efectúa secuencialmente a lo largo de las 48 antenas dispuestas en forma de circunferencia, es decir, la señal conmuta de una antena a otra.

Por consiguiente, un observador, que puede ser un avión, apreciará una naturaleza circular en la onda recibida, es decir, una variación en la frecuencia con un vaivén sincronizado con

el VOR cuando radia girando en dirección al avión (máxima frecuencia) o cuando radia girando en dirección opuesta al avión (mínima frecuencia). Este vaivén circular se puede interpretar como una señal sinusoidal de 30 Hz, y así es como se genera la señal variable. Un avión volando en el lado derecho del VOR observará cambios en la frecuencia a medida que las antenas giren, recibiendo más frecuencia a medida que las antenas giran hacia el avión y menos frecuencia cuando se alejan.

Si este giro se representa sobre un plano 2D para todos los puntos por los que pasa el giro de la antena, se obtiene una señal sinusoidal con una frecuencia de 30 Hz. Dado que el giro de la antena completa 30 vueltas en un segundo, la señal creada es una señal sinusoidal de 30 Hz, modulada en frecuencia con una desviación de 480 Hz. Es decir, la señal variable tiene que moverse alrededor del círculo de antenas con un diámetro de 13,4 metros para dar esta desviación en frecuencia debida al efecto Doppler.

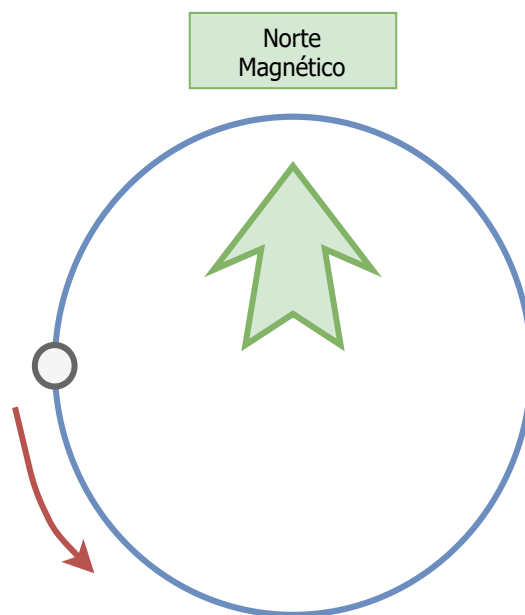


Figura 14. Representación del giro efectuado por la señal radiada por las antenas del VOR en sentido antihorario.

En el caso en que el avión se encuentre en otro punto respecto al VOR, la señal variable recibida tendrá exactamente la misma fase, ya que el efecto Doppler funciona de igual manera para todos los puntos. Sin embargo, la señal de referencia juega un papel clave en este proceso, el cual se introduce a continuación.

La señal de referencia será un seno cuyo valor es 0 cuando la señal de referencia apunta hacia el radial 0, es decir, pasa por su posición de reposo. El valor de la señal de referencia, a diferencia de la señal variable, es la misma para todos los radiales de alrededor del VOR. A medida que la señal variable inicia el giro, la señal de referencia irá avanzando su oscilación. Una oscilación completa en la señal de referencia será un giro completo de la circunferencia de la señal variable.

Para entender mejor el concepto de la señal, se presenta el ejemplo ilustrado en las Figuras 15 – 18. Dada la señal variable de un VOR que gira 30 veces por segundo en sentido contrario a las agujas del reloj:

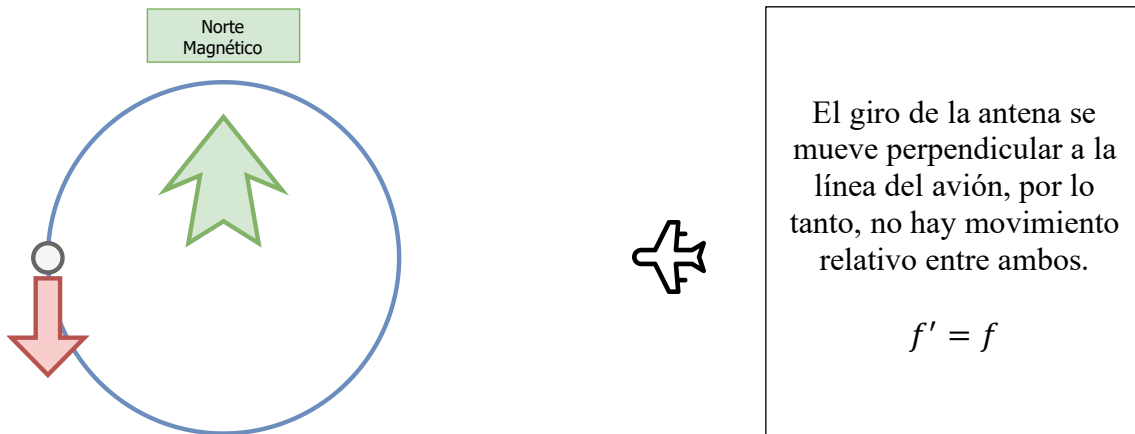


Figura 15. Desviación de la frecuencia recibida por un avión en el radial 090 de la señal variable apuntando hacia el radial 270.

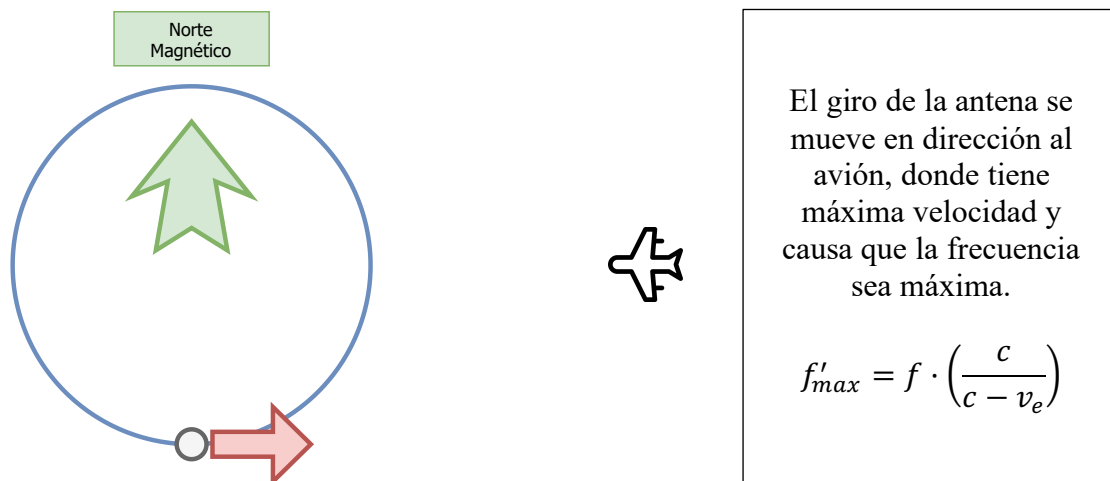


Figura 16. Desviación de la frecuencia recibida por un avión en el radial 090 de la señal variable apuntando hacia el radial 180.

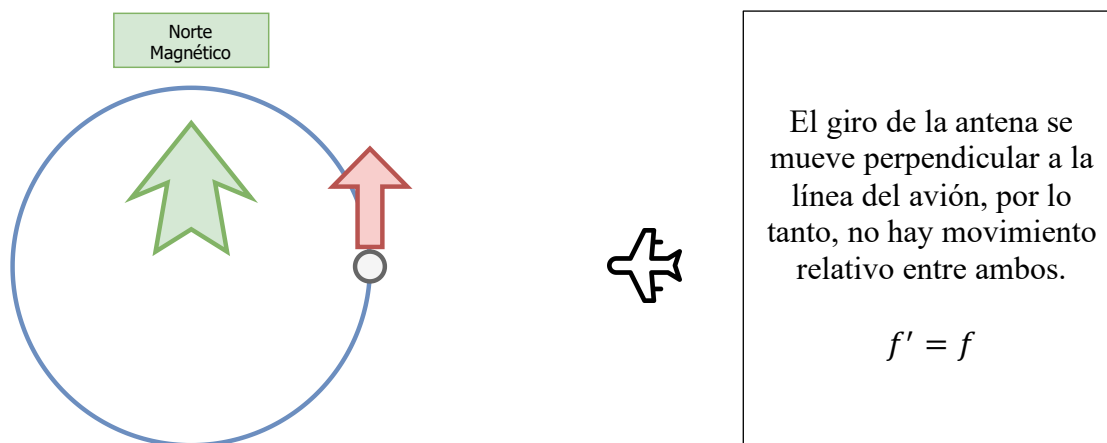


Figura 17. Desviación de la frecuencia recibida por un avión en el radial 090 de la señal variable apuntando hacia el radial 090.



Figura 18. Desviación de la frecuencia recibida por un avión en el radial 090 de la señal variable apuntando hacia el radial 000 o 360.

Se puede demostrar la desviación en frecuencia de la señal modulada en FM, teniendo en cuenta que la máxima desviación vendrá dada cuando la velocidad de la señal giratoria del VOR respecto a la posición del avión sea máxima o mínima.

Sabiendo que la señal gira 30 veces por segundo y que el diámetro de la circunferencia que forman las antenas que radian la señal variable es de 44 pies (13,4112 metros), se puede calcular la velocidad lineal en un punto dado mediante la siguiente expresión:

$$v = \omega \cdot r = 2 \cdot \pi \cdot 30 \cdot \frac{13,4112}{2} = 1263,98 \frac{m}{s} \equiv 4550 \frac{km}{h} \quad (50)$$

Aplicando la siguiente fórmula del efecto Doppler, donde la frecuencia de emisión es la frecuencia de la señal portadora (contenida en el rango de frecuencias entre 108 MHz y 118 MHz), se considera la media entre estas dos frecuencias, es decir, 113 MHz:

$$f'_{max} = f \cdot \left(\frac{c}{c - v_e} \right) = 113 \times 10^6 \cdot \left(\frac{3 \times 10^8}{3 \times 10^8 - 1263,98} \right) = 113,000476 \text{ MHz} \quad (51)$$

$$f'_{min} = f \cdot \left(\frac{c}{c + v_e} \right) = 113 \times 10^6 \cdot \left(\frac{3 \times 10^8}{3 \times 10^8 + 1263,98} \right) = 112,999524 \text{ MHz} \quad (52)$$

Se puede calcular la desviación en frecuencia de la modulación FM de la señal variable emitida por el VOR, entendiendo la desviación en frecuencia como los hercios que fluctúa la señal modulada en FM hacia arriba y hacia debajo de la portadora, y se puede expresar de la siguiente manera:

$$\Delta f = f_c \pm f_{max} \quad (53)$$

$$\Delta f = 113 \times 10^6 - 113,000476 \times 10^6 = -476 \text{ Hz} \approx -480 \text{ Hz} \quad (54)$$

$$\Delta f = 113 \times 10^6 + 112,999524 \times 10^6 = 476 \text{ Hz} \approx 480 \text{ Hz}$$

Por lo tanto, el índice de modulación se puede expresar como:

$$\beta = \frac{\Delta f}{f_m} = \frac{480}{30} = 16 \quad (55)$$

4.2 El Espectro de la Señal de un VOR

En el plano frecuencial se reparten todas las señales que emite un VOR a lo largo de un ancho de banda de 24 kHz entre las dos bandas. Así pues, la distancia mínima entre dos VOR que emiten en un mismo espacio aéreo ha de ser de mínimo 50 kHz entre sus respectivas portadoras, por normativa [24].

Las señales que se pueden identificar en el espectro del VOR son las que se muestran en la Figura 19.

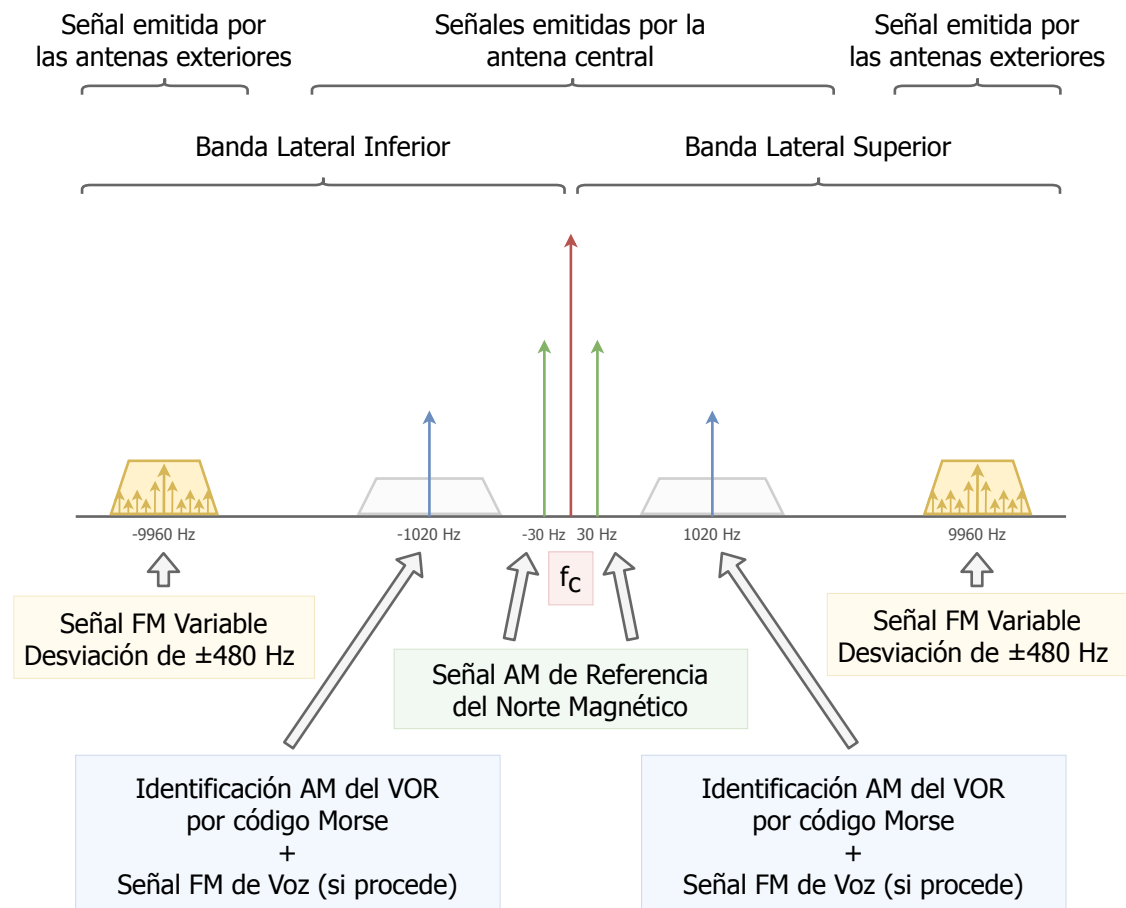


Figura 19. Estructura del espectro radioeléctrico de las señales emitidas por un Doppler VOR.

Estas señales son:

- **La señal portadora, f_c .** Es la frecuencia que aparece en todas las cartas aeronáuticas y que los pilotos de las aeronaves deben sintonizar en sus radios para recibir adecuadamente las señales emitidas por un VOR.

- **A ± 30 Hz de la frecuencia portadora.** Se emite la señal de referencia de naturaleza sinusoidal con una frecuencia de 30 Hz.
- **A ± 1020 Hz de la frecuencia portadora.** Se emite la señal de identificación del VOR en código morse. Además, algunos VOR proporcionan información del tiempo, información del aeropuerto e información generalmente útil para los pilotos, donde una voz lo repite en bucle. Esta señal ocupa un ancho de banda desde los 250 Hz hasta los 2500 Hz.
- **A ± 9960 Hz de la frecuencia portadora.** Se emite, modulada en FM, la señal variable con una desviación de ± 480 Hz.

4.3 Las antenas de los VOR

Las antenas internacionalmente utilizadas desde la introducción de los VOR en el ámbito de la aviación están basadas en el modelo de antena en forma de espira de Alford, llamadas así por su inventor y creador Andrew Alford (Samara, Rusia 1904 – California, Estados Unidos 1992). Alford trabajó e investigó diversos tipos de antena a lo largo de su vida profesional en la Universidad de California, donde también contribuyó en el diseño y estructura del sistema de aterrizaje instrumental o ILS (*Instrument Landing System*).

En la década de los cuarenta recibió la patente de las antenas en forma de espira cuadrada que se usan actualmente en las estaciones VOR. Estas antenas omnidireccionales emiten radiación polarizada horizontalmente y son capaces de emitir en una banda de frecuencias muy ancha.

Su diseño es muy simple y es el que se muestra en la Figura 20. Se parte de un dipolo de media longitud de onda, donde se doblan los dos brazos del dipolo hacia adentro, formando un ángulo de 90° entre ambos brazos. Posteriormente, se coloca una antena idéntica y de forma opuesta al dipolo doblado inicial, formando un eje de simetría entre ambas antenas.

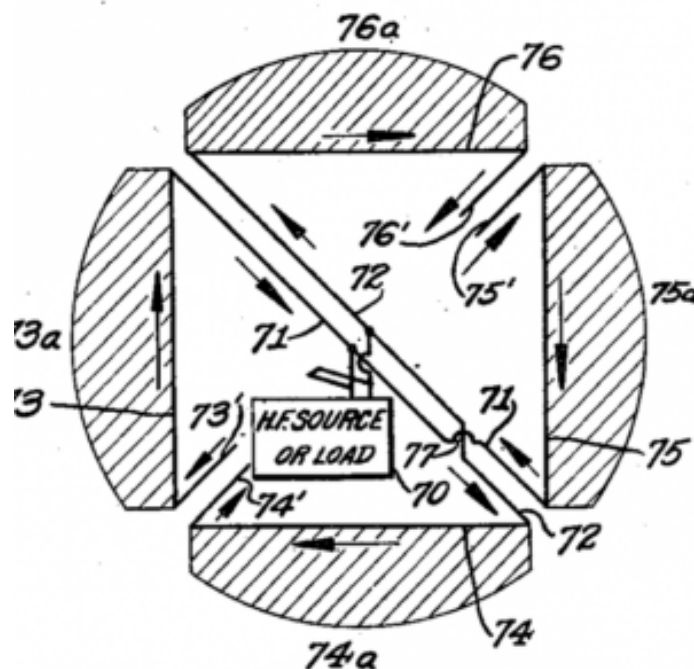


Figura 20. Diseño original de la antena de Alford [24].

Finalmente se unen las antenas cruzando las conexiones de estas, de manera que solo sea necesario un único punto de alimentación, y manteniendo al mismo tiempo la relación de fases. Esto último será una de las principales características que hacen que esta antena sea ideal para su uso en estaciones VOR.

La antena anteriormente descrita presentará una distribución de corrientes descompensada, ya que se puede medir un nodo de corrientes en el extremo de cada dipolo. Para compensar este efecto, Alford añadió a esta antena una carga capacitiva en los extremos de cada brazo, apuntando hacia el centro de la antena, de manera que el nodo de corrientes desapareciera y la distribución de corrientes se mantuviera más estable a lo largo del brazo del dipolo, haciendo que el promedio de la distribución de corrientes fuera mayor y resultando así en una antena cuya radiación aprovechable incrementara considerablemente. Se puede observar la distribución de corrientes de la antena en la Figura 21.

Estas antenas, como se ha mencionado anteriormente, son efectivas en un ancho de banda muy amplio de frecuencias. Sin embargo, se puede obtener la máxima ganancia para cada banda de frecuencias modificando el espacio que queda entre las cargas capacitivas de los extremos de los brazos de los dipolos. Para la banda VHF de los VOR, esta distancia es de aproximadamente 10 centímetros.

Dado que los VOR están formados por múltiples antenas como esta y la fase de las señales debe ser muy precisa, es importante que las antenas que emiten la señal variable lo hagan en el instante de tiempo oportuno. Para ello todas las antenas se deben alimentar con cables de la misma longitud eléctrica.

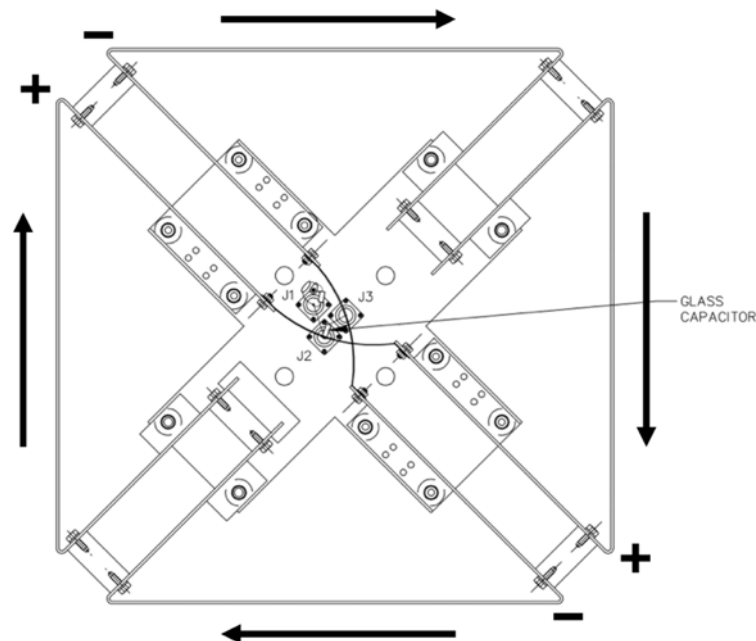


Figura 21. Distribución de corrientes por la antena de Alford [24].

Debajo de la antena se coloca un *balun*, que es una línea cuya sección hace una forma de 180 grados, usado para balancear la señal que sale del cable coaxial que alimenta la antena. La sujeción de esta antena se puede observar en la Figura 22.

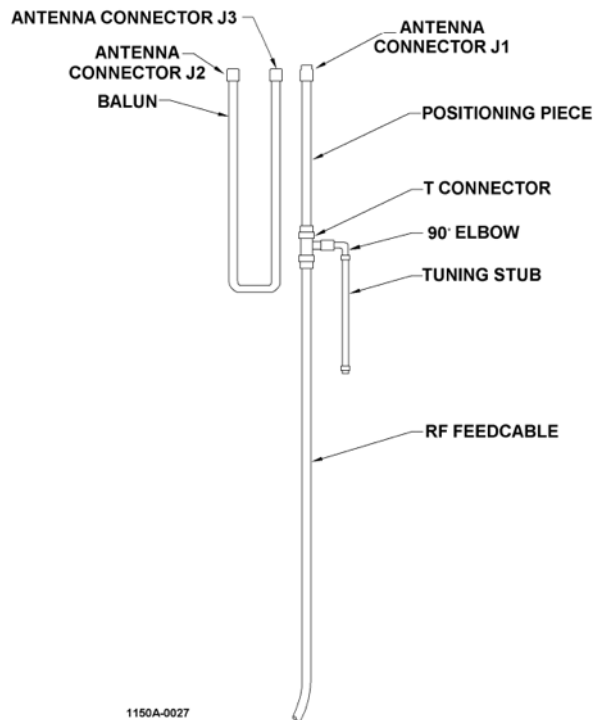


Figura 22. Esquema del circuito que sujeta la antena [24].

Es fácilmente observable que, debido a la transposición del nodo de alimentación de ambas antenas, en los vértices de la antena se obtiene una fase positiva y otra negativa en un mismo instante de tiempo, por lo que la distribución de corriente de los cuatro elementos fluye en la misma dirección. Por lo tanto, un receptor obtendrá de esta antena una señal con la misma fase que con la que se emitió, sin importar la distancia a la que se encuentre de esta.

Además, las antenas instaladas en los VOR están cubiertas por un radomo de fibra de vidrio que las protege de la intemperie y de los fenómenos meteorológicos adversos, tal y como se observa en la Figura 23, con una ligera forma de cono para evacuar rápidamente el agua de las precipitaciones y para evitar que se posen elementos sobre este que puedan afectar al correcto funcionamiento del VOR [24].

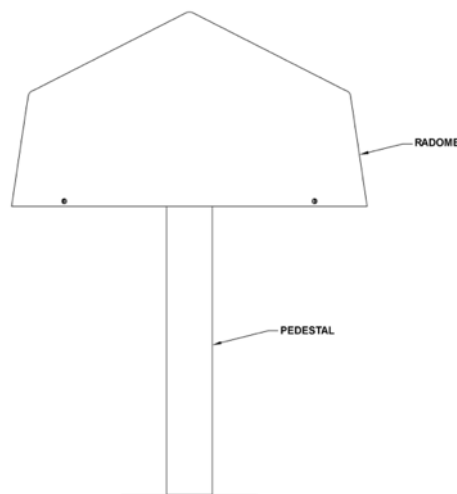


Figura 23. Diagrama de la forma del radomo que protege la antena [24].

4.4 El Cono de Confusión

La zona del espacio situada directamente encima del VOR se conoce como cono de confusión, que es un espacio en el que las señales emitidas por el VOR son débiles y pueden causar lecturas erróneas.

Sin embargo, una vez superada esta frontera, la transición se completa y las lecturas del VOR vuelven a ser fiables [1]. Esto es debido a la forma del diagrama de radiación de las antenas omnidireccionales del VOR, las cuales presentan una potencia de emisión muy baja directamente encima de ellas debido a la forma de rosquilla del diagrama de radiación característico de este tipo de antenas.

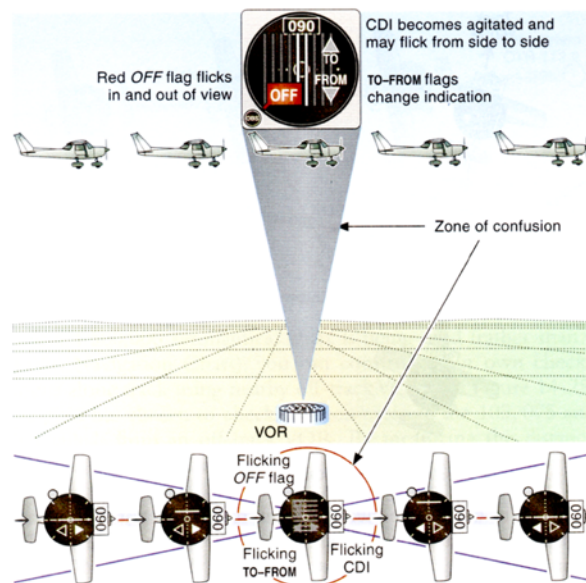


Figura 24. Diagrama representativo del cono de confusión del VOR [1].

4.5 La Malla Metálica del VOR

Debajo de todas las antenas que forman el VOR hay instalada una malla metálica de unos 30 a 45 metros de diámetro y 4 metros por encima del nivel del suelo, tal y como se puede observar en la Figura 25. Esta malla metálica tiene varios propósitos, entre ellos:

- Plano reflector de la señal para dirigir de forma más efectiva la señal hacia el aire.
- La posibilidad de predecir con más exactitud el desempeño de las antenas, ya que así se minimizan las reflexiones indeseadas contra el suelo.
- Separar el plano de mas de las antenas, haciendo que este no sea el suelo. En el hipotético caso en el que el plano de masa de todo el sistema de antenas fuera el suelo, se induciría mucho ruido en las señales debido a las interferencias de otros equipos eléctricos de alrededor el VOR.
- La conductividad del suelo depende de las condiciones meteorológicas. Esto podría cambiar el desempeño de las antenas en función del tiempo.

Por lo tanto, son razones suficientes como para optar a instalar la característica malla de metal debajo de las antenas. Además, debajo de esta maya se acostumbra a ver una caseta, que está climatizada, donde hay instalados todos los equipos eléctricos y electrónicos que controlan el VOR [25].



Figura 25. Malla metálica de debajo de las antenas.

5 Aplicación de los VOR en la Aviación

En este capítulo se introducen los usos que se le puede dar a los VOR en el ámbito de la aviación. Un piloto de una aeronave debe conocer muy bien el funcionamiento de estos radiofaros para poder hacer un correcto uso de ellos. Y no solo eso, sino que deben interiorizar las diferentes maniobras que se pueden realizar con la interacción de las señales de un VOR, ya que en pleno vuelo un piloto recibe muchos inputs y el tiempo de reacción se ve muy limitado [1, 2, 8, 26].

Adicionalmente, los pilotos deben conocer la posición de los VOR, cuya información se encuentra en las cartas aeronáuticas. Una carta aeronáutica es un mapa que muestra una porción de la Tierra, con sus relieves y construcciones. Estos mapas también reflejan las rutas que deben seguir las aeronaves, los diferentes elementos de navegación y comunicación y proporcionan información útil de dichos elementos.

En resumen, un piloto de una aeronave puede utilizar un VOR para:

- Cuando se desea conocer la posición del avión respecto al VOR y la trayectoria TO y FROM (más información en el apartado 5.5 de este trabajo).
- Para la realización de maniobras, virajes y órbitas (llamadas *holdings*). Este tipo de maniobras se usan en espacios aéreos muy concurridos, en los que el control de tráfico aéreo prioriza unos vuelos respecto a otros, ordenando a los aviones menos prioritarios a mantenerse en el aire haciendo órbitas alrededor de los VOR hasta que puedan ser atendidos.
- Aproximaciones instrumentales para aterrizar.

5.1 Guiado Mediante un VOR

Los VOR se pueden usar para indicarle a un piloto el rumbo deseado y la desviación angular del avión respecto a ese rumbo. Sin embargo, las señales originales del VOR no informan de la dirección de la fuerza de empuje de los motores (o la dirección a la que apunta el morro el avión, que se llamará *heading* a partir de ahora), sino que indica si el avión se encuentra en el rumbo correcto o si está desviado de este independientemente del *heading* que indique la brújula.

Esto es debido a que las aeronaves en pleno vuelo experimentan fuerzas externas que se deben tener en cuenta cuando se desea mantener un rumbo. Una de las fuerzas más importantes a tener presente es el viento, que debe corregirse constantemente para mantener el rumbo indicado.

En consecuencia, es común observar como un avión sigue un rumbo indicado con un *heading* diferente, fenómeno que recibe el nombre de *drift*. En este caso, la brújula indicará que el avión apunta hacia un punto diferente al rumbo indicado, mientras que las señales del VOR indicaran que el avión está siguiendo el rumbo adecuado.

En definitiva, las indicaciones del VOR dependen de la desviación angular del avión relativo al rumbo seleccionado, por lo tanto, son totalmente independientes del *heading* que tenga el avión en cualquier instante, como se puede observar en la Figura 26.

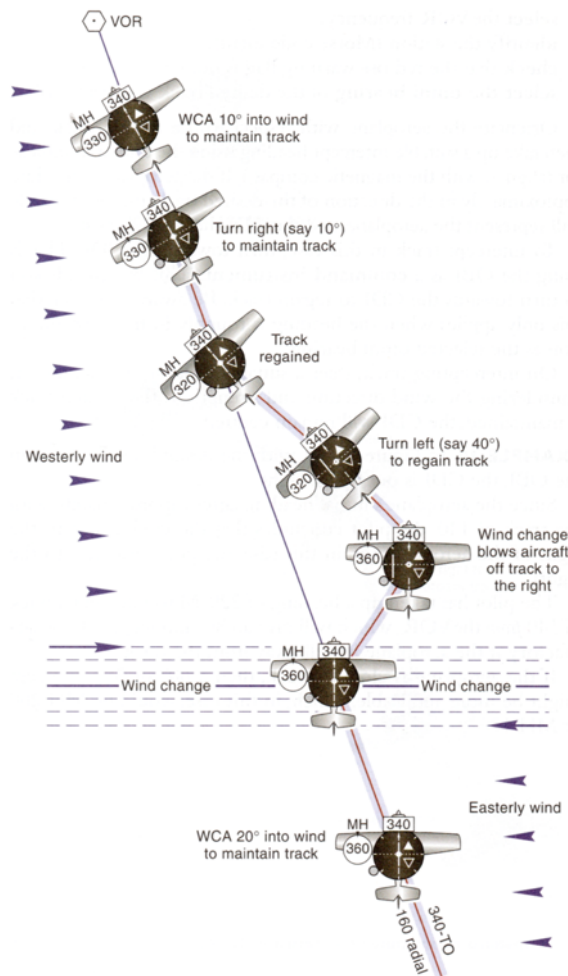


Figura 26. Variación de las indicaciones de los diferentes instrumentos de la cabina del avión en función del rumbo [1].

5.2 Los VOR en las Cartas Aeronáuticas

La mayoría de las cartas aeronáuticas informan de la posición, la frecuencia central y la identificación en código Morse de cada VOR. En caso de que un VOR sufra cualquier tipo de cambio o anomalía, se notificará a los pilotos mediante un NOTAM.

Los NOTAM son emitidos por ENAIRE, que es el único proveedor de servicios de información aeronáutica (AIS) de España. ENAIRE está autorizado por la Dirección General de Aviación Civil (DGAC), en cumplimiento de la normativa internacional de la Organización de Aviación Civil Internacional (OACI). ENAIRE define un NOTAM de la siguiente manera:

“Los NOTAM son avisos que informan sobre el establecimiento, condición o modificación de cualquier instalación aeronáutica, servicio, procedimiento o peligro. También informan sobre alertas meteorológicas y avisos sobre zonas reservadas que no pueden ser penetradas por aeronaves.” [27]

Un VOR se puede representar de múltiples maneras en una carta aeronáutica (ver Figura 27), sin embargo, las formas más comunes de representarlos son las que se muestran en la

Figura 2 del apartado 2.4 de este trabajo. Dado que el norte magnético de la Tierra es la referencia de los radiales de un VOR, normalmente los diagramas que representan los VOR indican el norte magnético mediante una flecha, y rodean el símbolo del VOR con una circunferencia donde están marcados los radiales en intervalos de 10 grados.

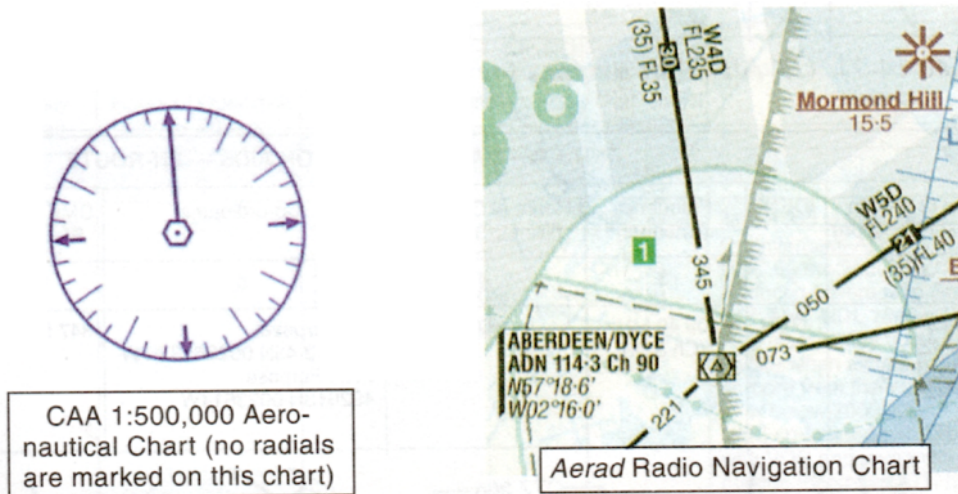


Figura 27. Representación de un VOR en las cartas aeronáuticas [1].

5.3 Instrumentos VOR en la Cabina del Avión

Existen varios tipos de instrumentos que muestran la información captada de las señales de los VOR, sin embargo, todos ellos son muy similares en términos de operación. Generalmente el instrumento que muestra dicha información recibe el nombre de OBI, el cual muestra el rumbo que sigue el avión y que el piloto previamente habrá indicado mediante el uso del OBS, que es una pequeña ruedecita que se encuentra justo debajo del instrumento que permite escoger el radial por el que se desea mantener el rumbo de vuelo.



Figura 28. Instrumentos de indicación de desviación OBI.

Para indicar que el rumbo es el correcto, estos instrumentos están provistos con una pequeña manecilla, llamada CDI, que indica el desvío en grados del avión con respecto al radial

seleccionado: si el avión se encuentra justo en el radial seleccionado por el piloto, la manecilla estará centrada; en caso contrario, la manecilla se moverá hacia la derecha o hacia la izquierda, indicando al piloto cuantos grados se ha desviado con respecto al radial seleccionado.

Además, estos instrumentos indican al piloto si se está dirigiendo en dirección al VOR (TO), o si se aleja de él (FROM). Esta información se muestra en el instrumento mediante una bandera con las letras TO o FR según corresponda. Para más información, referirse al apartado 5.5 de este trabajo.

Todo piloto debe saber que el OBI solamente se puede usar si no hay ningún indicador de fallo del instrumento y si se escucha claramente el identificador en código Morse del VOR sintonizado. Cumpliéndose estas dos condiciones, el margen de error de los radiales de un VOR no ha de ser superior a $\pm 1^\circ$.

5.4 El CDI

El CDI, como se ha mencionado anteriormente, es la manecilla que indica la desviación angular respecto al rumbo seleccionado. En todo momento, la posición de referencia será el centro, que es el radial que el piloto habrá seleccionado y por el cual se seguirá el rumbo deseado.

Para determinar la desviación respecto al rumbo establecido, el instrumento presenta unos puntos dispuestos de forma horizontal a ambos lados del punto central que indica el rumbo seleccionado por el OBS. Cada punto es equivalente a una desviación de 2° .

En definitiva:

- Si el avión se encuentra sobre el radial establecido, la manecilla permanecerá en el centro del instrumento.
- Si el avión se desvía dos grados hacia la derecha, la manecilla se moverá un punto hacia la izquierda, indicando al piloto que debe corregir su rumbo hacia la izquierda.
- Si el avión se desvía cuatro grados hacia la izquierda, la manecilla se moverá dos puntos hacia la derecha, indicando al piloto que debe corregir su rumbo hacia la derecha.
- Si el avión se desvía 10 grados o más, la manecilla llegará a su máximo recorrido.

Dado que el CDI indica la desviación angular, será importante determinar la distancia relativa que separa el avión del VOR, ya que a distancias más cortas la desviación respecto a la indicación del instrumento será más pequeña que a distancias más grandes.

5.5 Las Banderas TO y FROM

Las banderas TO y FROM indican si la selección del radial realizada con el OBS llevará al avión hacia el VOR o lo alejará del VOR. Se presentan diversos casos que se ilustran en las Figuras 29 - 34.

Caso 1.

Datos: El avión tiene seleccionado el radial 000 en el OBI como el radial objetivo (ver Figura 29). El avión está volando por el radial 180 con un *heading* de 0° respecto al norte magnético.

Consecuencias: El OBI indicará que no hay desviación respecto al rumbo establecido. El OBI indicará que se está volando en dirección “TO” del VOR (hacia el VOR).

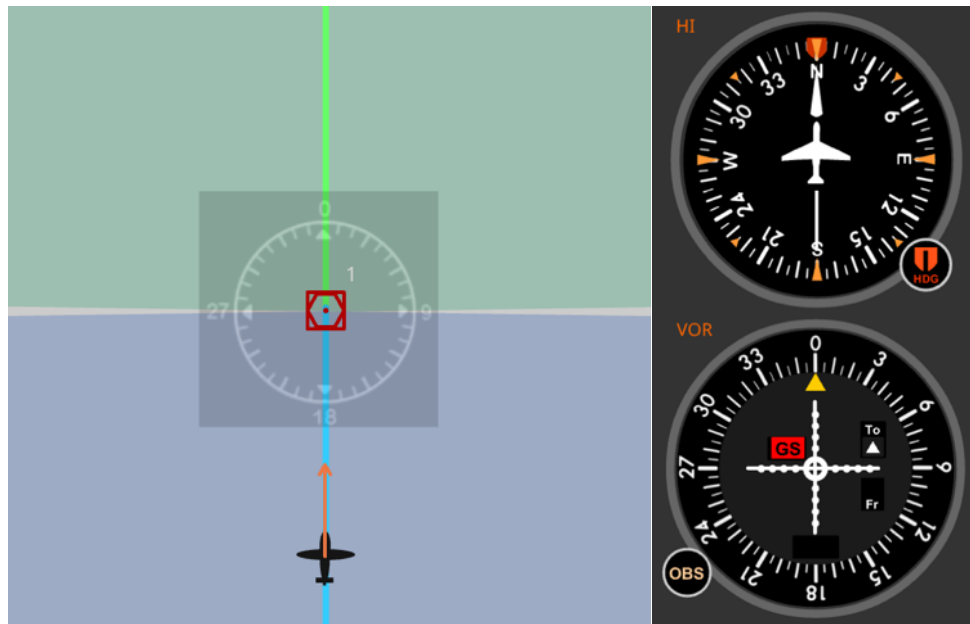


Figura 29. Esquema de la posición del avión e indicaciones en los instrumentos [28].

Caso 2.

Datos: El avión tiene seleccionado el radial 000 en el OBI como el radial objetivo (ver Figura 30). El avión está volando por el radial 000 con un *heading* de 0° respecto al norte magnético.

Consecuencias: El OBI indicará que no hay desviación respecto al rumbo establecido. El OBI indicará que se está volando en dirección “FROM” del VOR (alejándose del VOR).

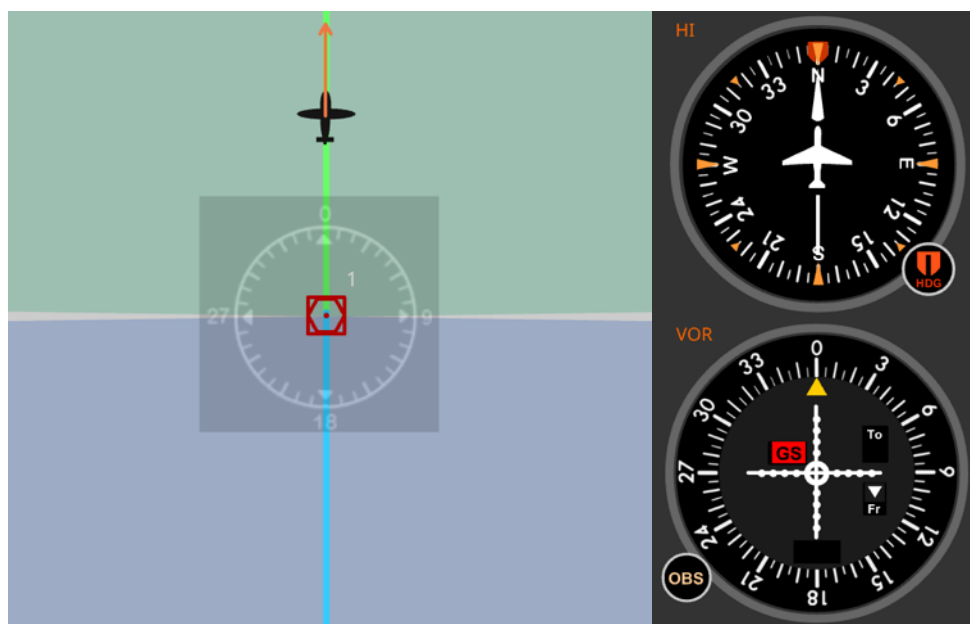


Figura 30. Esquema de la posición del avión e indicaciones en los instrumentos [28].

Caso 3.

Datos: El avión tiene seleccionado el radial 000 en el OBI como el radial objetivo. El avión está volando por el radial 172 con un *heading* de 340° respecto al norte magnético.

Consecuencias: El OBI indicará que hay una desviación de 8° hacia la derecha respecto al rumbo establecido. El OBI indicará que se está volando en dirección “TO” del VOR.

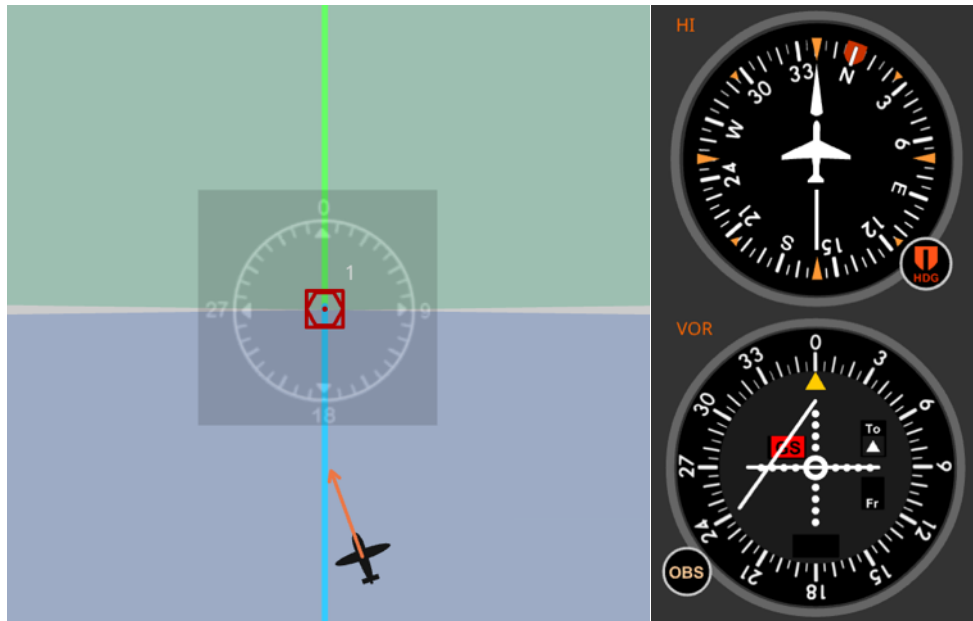


Figura 31. Esquema de la posición del avión e indicaciones en los instrumentos [28].

Caso 4.

Datos: El avión tiene seleccionado el radial 000 en el OBI como el radial objetivo. El avión está volando por el radial 352 con un *heading* de 20° respecto al norte magnético.

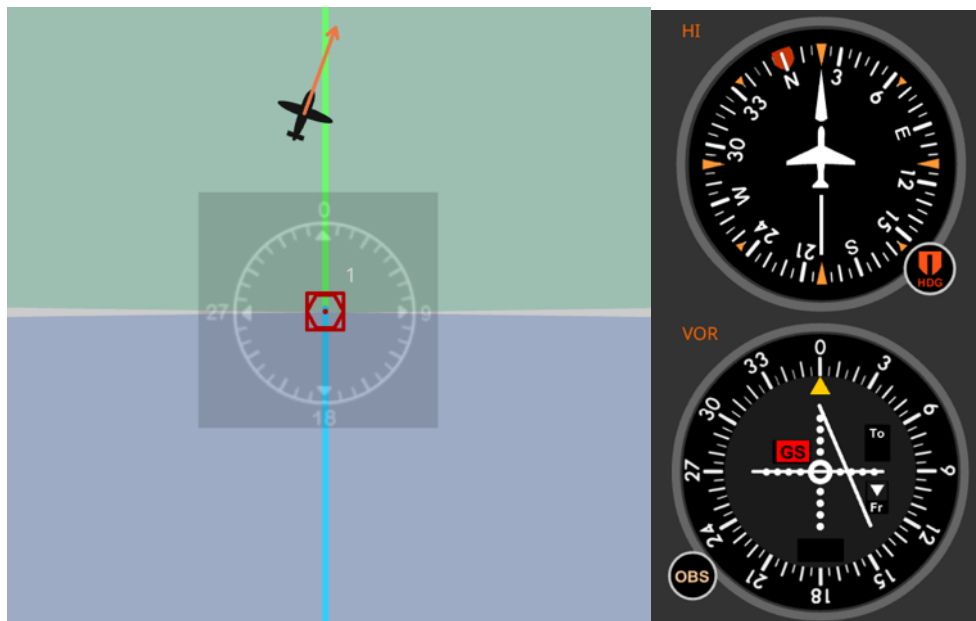


Figura 32. Esquema de la posición del avión e indicaciones en los instrumentos [28].

Consecuencias: El OBI indicará que hay una desviación de 8° hacia la izquierda respecto al rumbo establecido. El OBI indicará que se está volando en dirección “FROM” del VOR.

Caso 5.

Datos: El avión tiene seleccionado el radial 000 en el OBI como el radial objetivo. El avión está volando por el radial 30 con un *heading* de 180° respecto al norte magnético.

Consecuencias: El OBI indicará que hay una desviación de más de 10° hacia la izquierda respecto al rumbo establecido. El OBI indicará que se está volando en dirección “FROM” del VOR.

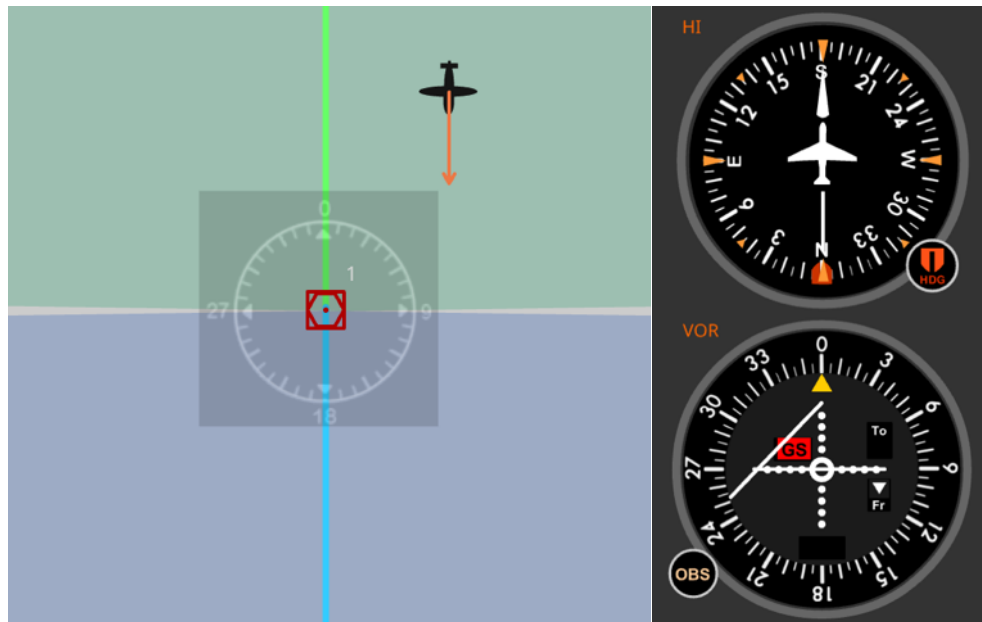


Figura 33. Esquema de la posición del avión e indicaciones en los instrumentos [28].

Caso 6.

Datos: El avión tiene seleccionado el radial 000 en el OBI como el radial objetivo. El avión está volando por el radial 150 con un *heading* de 180° respecto al norte magnético.

Consecuencias: El OBI indicará que hay una desviación de más de 10° hacia la izquierda respecto al rumbo establecido. El OBI indicará que se está volando en dirección “TO” del VOR.

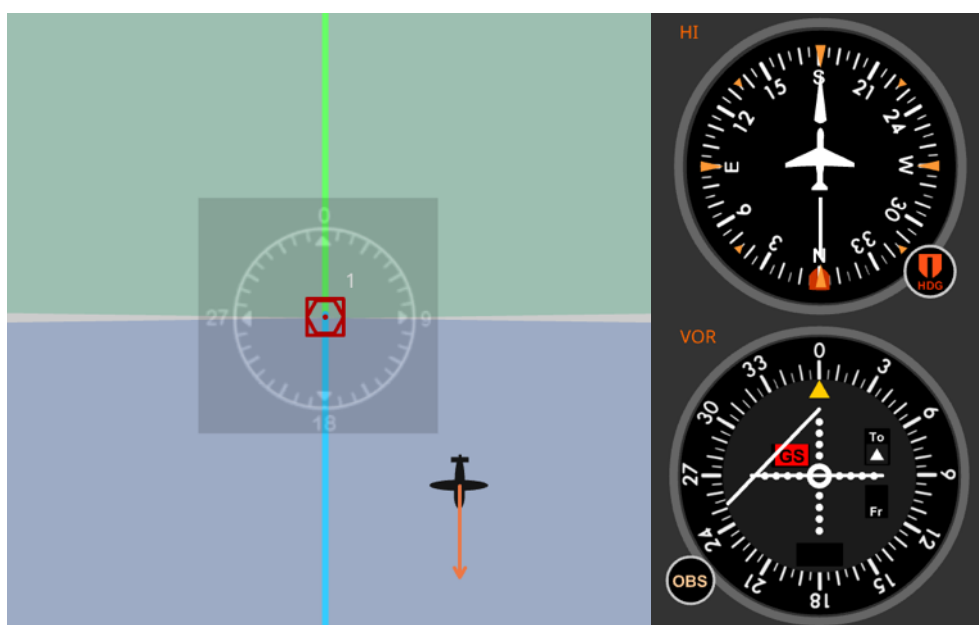


Figura 34. Esquema de la posición del avión e indicaciones en los instrumentos [28].

Es fácil confundir el mensaje que transmiten estas banderas con el actual rumbo del avión: el avión puede estar alejándose del VOR y el OBI puede marcar que el avión está en posición TO si el piloto ha seleccionado mediante el OBS la intercepción de un radial diametralmente opuesto a la posición del avión.

Se puede entender el funcionamiento de las banderas TO y FROM dividiendo el plano de los radiales en dos mitades. El piloto mediante el OBS indicará el radial que desea interceptar, que generalmente es un radial que se encuentra en el lado opuesto al que se encuentra el avión. Partiendo entonces de ese radial (que estará en la zona FROM), todos los radiales que se encuentren a 90 grados o menos por ambos lados de dicho radial se consideraran de la zona FROM, lo que lo hace un total de 180 radiales. Los 180 radiales opuestos a los radiales de FROM serán los radiales de la zona TO, donde se supone que se encuentra el avión en el momento de programar el OBI con el OBS.

En definitiva, notar que al seleccionar un radial con el OBS se estará indicando que el rumbo deseado será dicho radial y su opuesto. Esto último puede observarse con detalle en la Figura 35.

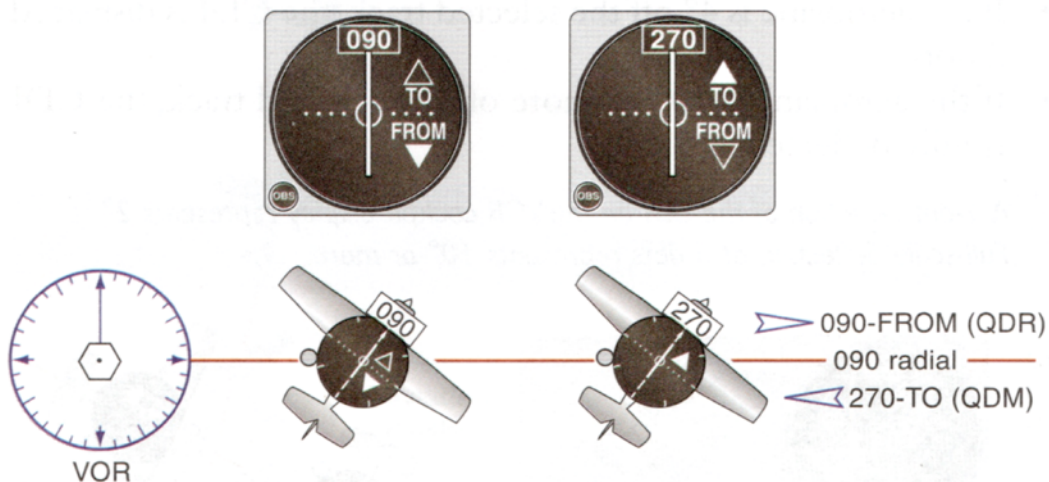


Figura 35. Desviación nula del CDI en el radial seleccionado y su opuesto [1].

5.6 Uso Operacional de un VOR

Para sintonizar un VOR en la cabina de un avión es necesario asegurar que la radio está correctamente alimentada al circuito eléctrico del avión. Posteriormente, se deberá comprobar que está en posición de recibir señales VOR, por lo tanto, mover el selector a VHF-NAV e indicar la frecuencia central del VOR, que estará provista por las cartas aeronáuticas.

Finalmente, es imprescindible asegurar que la señal recibida es potente; para ello se deberá escuchar claramente el código Morse que identifica el VOR. Una vez se han hecho estos pasos, simplemente será el mismo piloto quién deberá interpretar los datos y quién determinará el rumbo deseado moviendo la manecilla del CDI con el OBS.

5.7 Maniobras con los VOR

5.7.1 Paso por Encima del VOR

Este tipo de maniobra se realiza cuando se hacen travesías, que siguen referencias de las cartas aeronáuticas que en la mayoría de las ocasiones son VOR. Es decir, en estos vuelos la aeronave volará de un VOR a otro. A medida que el avión se vaya acercando al VOR, la manecilla del CDI empezará a agitarse cada vez más, ya que contra más pequeña sea la distancia, se reflejará una desviación angular mayor a pequeños cambios en el rumbo, incluso llegando a superar el margen de 10°. En cuanto el avión entra en la zona o cono de confusión, el CDI comenzará a dar lecturas erróneas del VOR, incluso llegando a activarse la bandera OFF, que es el indicio de señal insuficiente.

No será hasta que el avión salga del cono de confusión (aproximadamente un minuto de vuelo) por el lado contrario por el que entró para que el CDI se vuelva a estabilizar. En este caso, la bandera del TO cambiará a FROM, o viceversa.

5.7.2 Paso por el Lateral de un VOR

Esta es una forma común de comprobar el progreso del vuelo, donde el avión pasa por un costado del VOR. En este caso se mide el tiempo aproximado que tarda el avión en pasar por dos radiales que forman un ángulo concreto (generalmente de 90°). Hay normas establecidas para hacer cálculos mentales rápidos.

5.7.3 Intercepción de un Radial

Se trata de orientar el avión hacia el radial deseado y observar el CDI para corregir las desviaciones. En cuanto el CDI comienza a moverse, el piloto empezará a hacer las correcciones de rumbo pertinentes para recolocarse en dicho radial.

6 Introducción al *Software Defined Radio*

En este capítulo se introducen varias nociones básicas del funcionamiento de un RTL-SDR, ya que las señales de los VOR se captarán mediante una antena conectada a un RTL-SDR para su digitalización, tal y como se muestra en la Figura 36.



Figura 36. Esquema del montaje del receptor de señales VOR.

Los RTL-SDR están constituidos principalmente por convertidores analógico a digital, amplificadores, filtros, osciladores y demoduladores.

6.1 Los Conversores

En la era de la tecnología actual, la mayoría de los sistemas eléctricos y electrónicos están evolucionando hacia una naturaleza digital debido a la multitud de ventajas que aporta la digitalización. No obstante, el universo es un medio analógico, lo que significa que todas las señales que viajan a través de él son señales que varían de forma continua con el tiempo.

Algunas de las ventajas de digitalizar las señales son, por ejemplo, la robustez de las señales y las modulaciones ante interferencias, la poca afectación que tienen debido al ruido, presentan métodos de corrección de errores y permiten la transmisión de grandes cantidades de información utilizando un ancho de banda muy pequeño del espectro.

En consecuencia, teniendo en cuenta que la mayoría de las señales producidas por los transductores (los elementos que transforman unas determinadas magnitudes físicas en señales eléctricas o viceversa) son de tipo analógico, resulta necesaria la utilización de convertidores para adaptar las señales al medio por el que van a viajar, ya sean convertidores analógicos a digital (ADC o A/D) o viceversa, digital a analógico (DAC o D/A).

En los ADC se trata de traducir una magnitud analógica, ya sea de tensión o de corriente, a código binario, es decir, en bits. Se consigue mediante los tres pasos para la digitalización: muestreo, cuantificación y codificación [29].

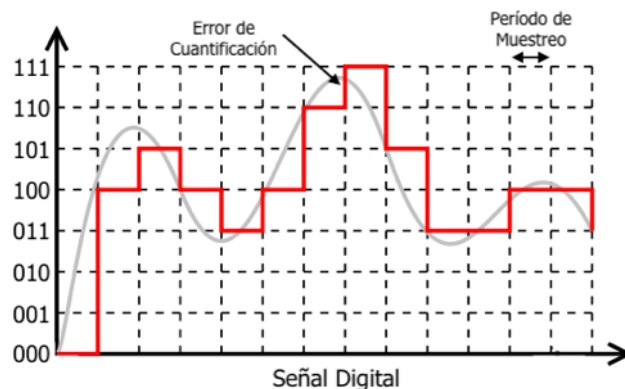


Figura 37. Proceso de digitalización de una señal continua en el tiempo.

Dado que se transforma una señal de tiempo continuo a tiempo discreto, se estará cometiendo un error (el error de cuantificación) que no permitirá recuperar la señal original a posteriori, pero sí una aproximación muy exacta a la realidad. Ver Figura 37.

No obstante, el error cometido se puede calcular y generalmente no afecta gravemente al correcto rendimiento de la señal y de los equipos. Por lo tanto, dado que el riesgo de digitalizar la señal es tan bajo y dadas las múltiples ventajas que aporta, se asume el inevitable error que se comete.

En el proceso de muestreo se determina el intervalo de tiempo que tarda el muestreo en determinar el valor de la señal en dicho instante; el inverso del tiempo de muestreo recibe el nombre de frecuencia de muestreo, que indica el número de muestras que se toman en un segundo.

Durante el proceso de cuantificación, que suele hacerse a la par con el proceso de codificación, se asignan los bits mediante la asignación de niveles. Notar que a mayor número de bits se le asigne a cada nivel, menor será el rango del error cometido.

Algunos de estos conversores son los conversores de aproximaciones sucesivas, los conversores de rampa y los conversores de doble rampa.

Del mismo modo, el hecho de digitalizar las señales requerirá de la necesidad de transformar las señales digitales a señales de naturaleza analógica, para que estas puedan viajar a través del medio de propagación [29].

Así pues, se trata de dividir la salida analógica en una serie de niveles o escalones, cuya cantidad depende del número de bits empleados en los datos a convertir. Por ejemplo, si los datos emplean 8 bits, se obtendrán los siguientes escalones:

$$2^8 = 256 \text{ escalones} \quad (56)$$

Por otra parte, la magnitud de cada escalón depende también del margen de tensión del convertidor. Por ejemplo, para un convertidor de 0 a 5 voltios, la magnitud de cada escalón sería:

$$\frac{5 \text{ V}}{256 \text{ escalones}} = 0,0195 \text{ V} \quad (57)$$

A efectos teóricos, es sencilla la introducción a este tipo de conversores, sin embargo, a efectos prácticos se presentan como elementos complejos de diseñar, especialmente cuando se desea realizar una conversión precisa. Además, debido a la forma de la señal de salida (escalonada), se obtienen señales analógicas que pueden contener información extraña e indeseada, por lo que será necesario filtrar la señal antes de transmitirla a través del medio.

6.2 Fundamentos del RTL-SDR

El SDR o *Software Defined Radio*, que traducido es Radio Definida por Software, en un sistema de radiocomunicación que consiste en la unión de un hardware radio de banda ancha y herramientas de procesamiento digital de señal. Su módulo externo, llamado RTL-SDR,

en su forma más simple y según se puede observar en la Figura 38, se compone de una antena conectada a un convertor analógico-digital seguido de un sistema de procesamiento de señal digital (DSP) para extraer las señales de interés [9, 30].

Existen módulos RTL-SDR que permiten recibir y transmitir señales al mismo tiempo, mientras que hay otros más sencillos que solo sirven para captar señales electromagnéticas. La estructura básica de un RTL-SDR es la siguiente:

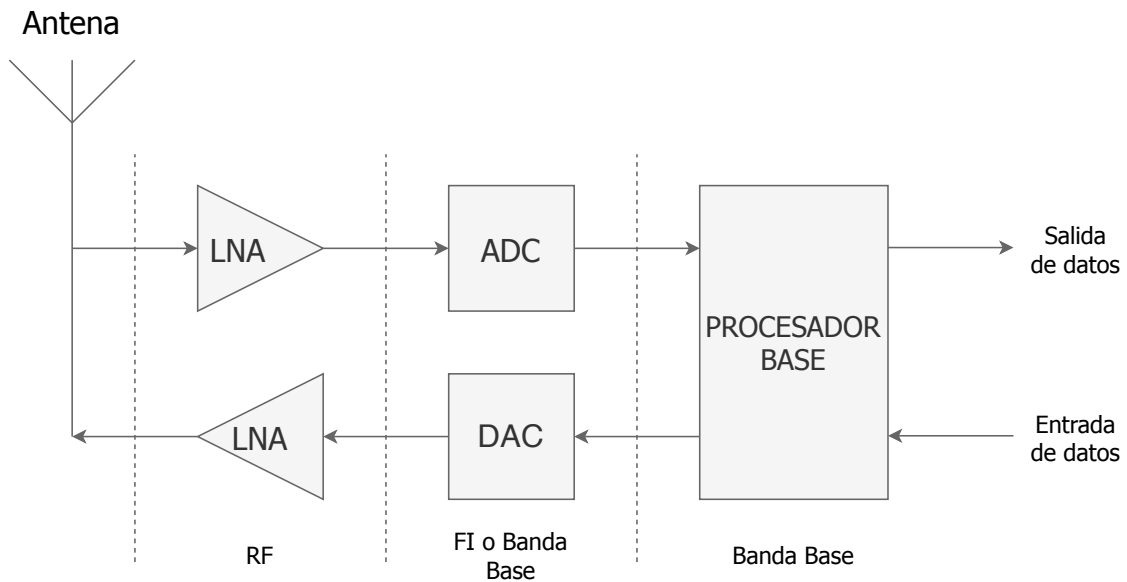


Figura 38. Estructura elemental de un *RTL-SDR*.

Dado que en la mayoría de las situaciones no se pueden realizar las conversiones analógico-digital a altas frecuencias, estas se realizan en banda base o en frecuencia intermedia trasladándolas con un PPL, como se observa en la Figura 39. Los componentes instalados en un RTL-SDR suelen ser mezcladores, moduladores/demoduladores, filtros o detectores.

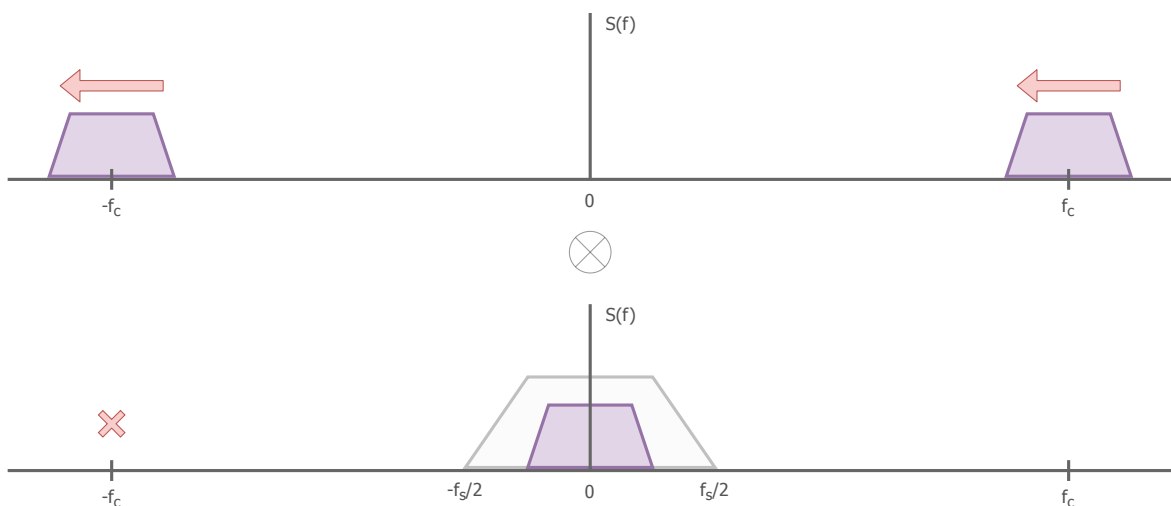


Figura 39. Proceso de traslado de la señal a banda base.

Los RTL-SDR desmodulan la señal mediante un demodulador IQ (referirse al apartado 3.2 de este trabajo para más información), cuya señal de salida se procesa normalmente de forma digital (observar la Figura 40). Por lo tanto, cada canal $I(t)$ y $Q(t)$ se muestrea a la frecuencia de muestreo indicada por software. Para evitar el efecto de *aliasing*, cada canal se filtra con un filtro paso bajo cuyo ancho de banda es la mitad de la frecuencia de muestreo, por lo tanto, pasarán las frecuencias que se encuentren en el rango de $\left[-\frac{f_s}{2}, +\frac{f_s}{2}\right]$.

En la práctica el ancho de banda de las señales que se pueden recibir es inferior, aproximadamente el 80% de todas ellas, ya que los filtros no son ideales y las transiciones entre la banda de paso y la banda atenuada del filtro no son totalmente abruptas.

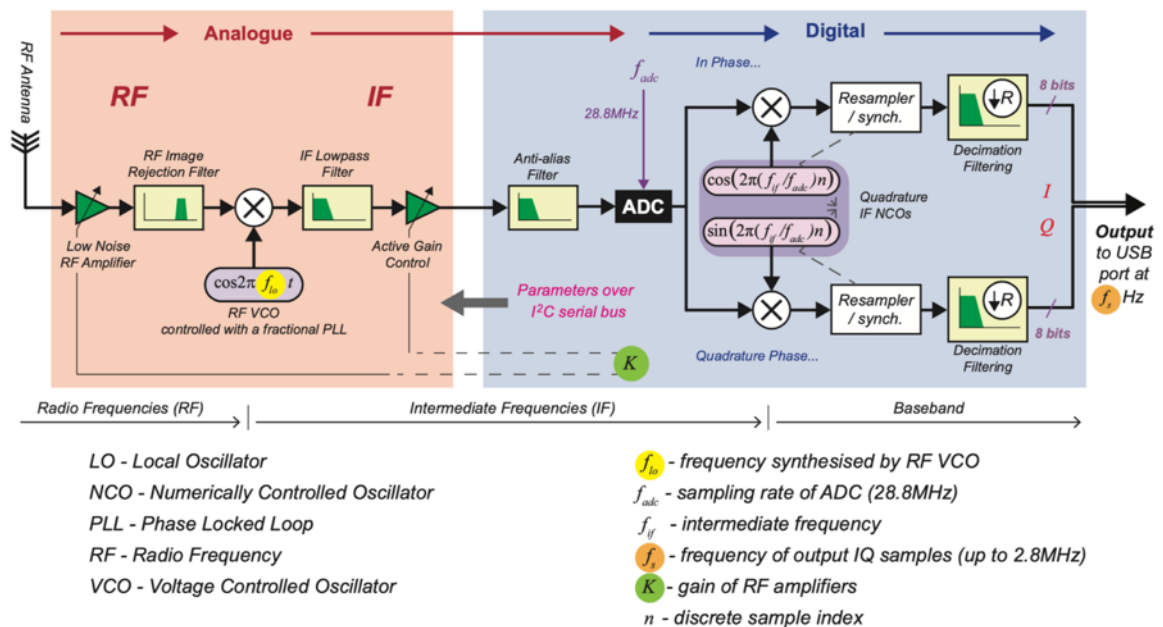


Figura 40. Estructura interna y detallada de un RTL-SDR [15].

El conversor AD del demodulador IQ del RTL-SDR implementa las dos funciones de muestrear y cuantificar la señal. El resultado de este proceso es una secuencia para cada canal cada n/f_s segundos, o una secuencia compleja si combinamos la señal en fase y en cuadratura:

$$r[n] = I[n] + jQ[n] \quad (58)$$

Dado que al cuantificador la secuencia está cuantificada (es decir, que toma un valor de los 2^n niveles entre $-R_{max}/2$ y $R_{max}/2$), la señal se ve afectada por el ruido de cuantificación y se puede expresar matemáticamente de la siguiente manera:

$$r[n] + e[n] = I[n] + jQ[n] + (e_I[n] + je_Q[n]) \quad (59)$$

Donde e_I y e_Q son variables aleatorias con distribución uniforme (en el caso que el nivel de señal se ajuste al máximo nivel del conversor). Por lo tanto, la relación señal a ruido se puede expresar como:

$$SNR = 6,02 \cdot B + 10,8 - 20 \cdot \log_{10} \left(\frac{R_{max}}{\sigma_r} \right) \approx 6B + 1,26 \text{ dB} \quad (60)$$

Donde $\pm R_{max}$ es el margen dinámico del cuantificador, σ_r^2 es la varianza de la señal $r[n]$ y B es el número de bits.

En el caso particular del RTL-SDR, el conversor AD es de 8 bits, no obstante, un bit es de signo. Suponiendo que $R_{max} = 3 \cdot \sigma_r$:

$$SNR = 6,02 \cdot 7 + 10,8 - 20 \cdot \log_{10} \left(\frac{3 \cdot \sigma_r}{\sigma_r} \right) = 43,4 \text{ dB} \quad (61)$$

El anterior valor puede parecer pobre, sin embargo, tras realizar la demodulación y utilizando técnicas de *downsamplig* se puede mejorar la relación señal a ruido.

Debido a que estos componentes son capaces de trabajar en una banda de frecuencias muy ancha (entre 20 MHz y 2-3 GHz, aunque depende del fabricante) y el tratamiento de señal se hace mediante software, hace que los RTL-SDR sean polivalentes y sirvan para la recepción de cualquier señal electromagnética con mucha comodidad. Además, se evita tener que montar circuitos con hardware, donde se evitan inconvenientes como las emisiones radiadas que estos pueden producir y que causan interferencias a la señal útil deseada, el ruido, interferencias, etc.

Por lo tanto, mediante software se pueden diseñar todo tipo de elementos usados en el procesamiento de señal, ya sean filtros, osciladores, mezcladores, moduladores y demoduladores, amplificadores, detectores y un largo etcétera. Esto hace que los RTL-SDR proporcionen una comodidad que no se consigue con ningún otro método de tratamiento de señal.

6.3 Aplicaciones del SDR

El SDR se acostumbra a utilizar para las siguientes aplicaciones que se presentan debajo, a pesar de que tiene infinidad de usos en las bandas de frecuencias en las que trabaja. Cabe añadir que como herramienta experimental es muy útil debido a sus prestaciones en relación con su coste, sin embargo, para aplicaciones más profesionales, es conveniente utilizar instrumentos más sofisticados.

- Recepción de señales de radio analógica y digital
- Radioaficionados
- Radioastronomía
- Recepción de señales satelitales (GPS, satélites de meteorología, etc.)
- Seguimiento del tráfico naval mediante las transmisiones AIS
- Seguimiento del tráfico aéreo mediante las transmisiones ADB-S (Modo S)
- Construcción de redes GSM
- Experimentación en la banda de frecuencias del LTE
- Experimentación y usos en la banda de frecuencias de Bluetooth, Wi-Fi, bandas de frecuencias públicas, etc.
- Control de máquinas teledirigidas

6.4 RTL-SDR en Estado de Recepción

Las señales electromagnéticas se captan a través de la antena, que puede ser cualquier tipo de antena (la más indicada para la banda de frecuencias donde se quiere trabajar) que conecta con los componentes del RTL-SDR mediante circuitos electrónicos especializados para garantizar la óptima transferencia. Seguidamente, la señal se amplifica y mediante un mezclador se convierte a la frecuencia intermedia o banda base. Si la frecuencia de trabajo es suficientemente baja, por lo general no será necesario hacer el traslado de frecuencias.

Tras el tratamiento anterior, se procede a convertir la señal analógica y continua a una señal discreta, la cual permitirá hacer el tratamiento digital por software. En este paso generalmente también se realizan filtrados, ya que el RTL-SDR capta un amplio espectro de frecuencias que son indeseadas a partir de este paso.

Finalmente se encuentra la sección de banda base donde un procesador se encarga de extraer la información deseada. A partir del procesador se envían los bits a través de un canal de comunicación, como puede ser un puerto serie USB.

6.5 RTL-SDR en Estado de Transmisión

No todos los RTL-SDR poseen la capacidad de transmitir, por lo tanto, antes de adquirir este producto es necesario asegurar que tiene dicha función. Para efectuar la transmisión, sencillamente se siguen los mismos pasos que si el RTL-SDR estuviera en estado de recepción, pero a la inversa.

Se empieza por el procesador, el cual transformará la señal para ser enviada a través de la antena. Cabe mencionar que los procesadores tienen capacidad de tratar diversas señales simultáneamente, así pues, permiten trabajar en diversas frecuencias a la vez si fuera necesario.

Seguidamente se filtra la señal y se hace una conversión digital-analógica para preparar la señal de manera que pueda ser emitida por la antena. Posteriormente se mezclan los datos con la señal del oscilador local para convertirla a la frecuencia a la que se va a transmitir. Finalmente se amplifica la señal resultante y se envía a través de la antena.

6.6 Software de Tratamiento Digital de Señal

Existen diversos programas de libre uso para visualizar las señales recibidas por un RTL-SDR, sin embargo, no todos sirven para procesar la señal. Algunos de los programas más populares de visualización de datos son:

- SDR#
- HDSDR
- SDR-RADIO
- SDR++
- CubicSDR (usado en este trabajo)
- Studio1

No obstante, los programas con los que se acostumbra a realizar el tratamiento digital son:

- GNU Radio
- MATLAB (usado en este trabajo)
- Simulink

6.7 Ejemplo de la Recepción y Visualización del Espectro de una Señal VOR Captada por el RTL-SDR

El siguiente ejemplo que se muestra en la Figura 41 ha sido captado con el programa Cubic SDR. Concretamente se capturó la señal emitida por el VOR del Aeropuerto Internacional de Reus, cuya frecuencia central se encuentra en los 114,2 MHz.

Notar la forma del espectro característica de un VOR: se observa un máximo a 30 Hz de distancia a ambos lados de la frecuencia central, un pequeño máximo a los 1030 Hz y finalmente la señal modulada en FM en los extremos de esta señal, cuyo ancho de banda (o, dicho de otra forma, su desviación en frecuencia) es de 480 Hz.

Se puede observar en la Figura 41 como el programa Cubic SDR ofrece la opción de visualizar el espectro de la señal, a la vez que también muestra un diagrama *waterfall*, que simplemente captura la señal en cada instante de tiempo y superpone los resultados, de manera que se pueden apreciar los cambios que sufre la señal con el paso del tiempo.

Además, el programa permite seleccionar la frecuencia central y tiene integrado varios demoduladores de varios tipos de modulación, que permiten escuchar las señales de audio. También permite definir la frecuencia que se desea sintonizar y el ancho de banda que se desea coger.

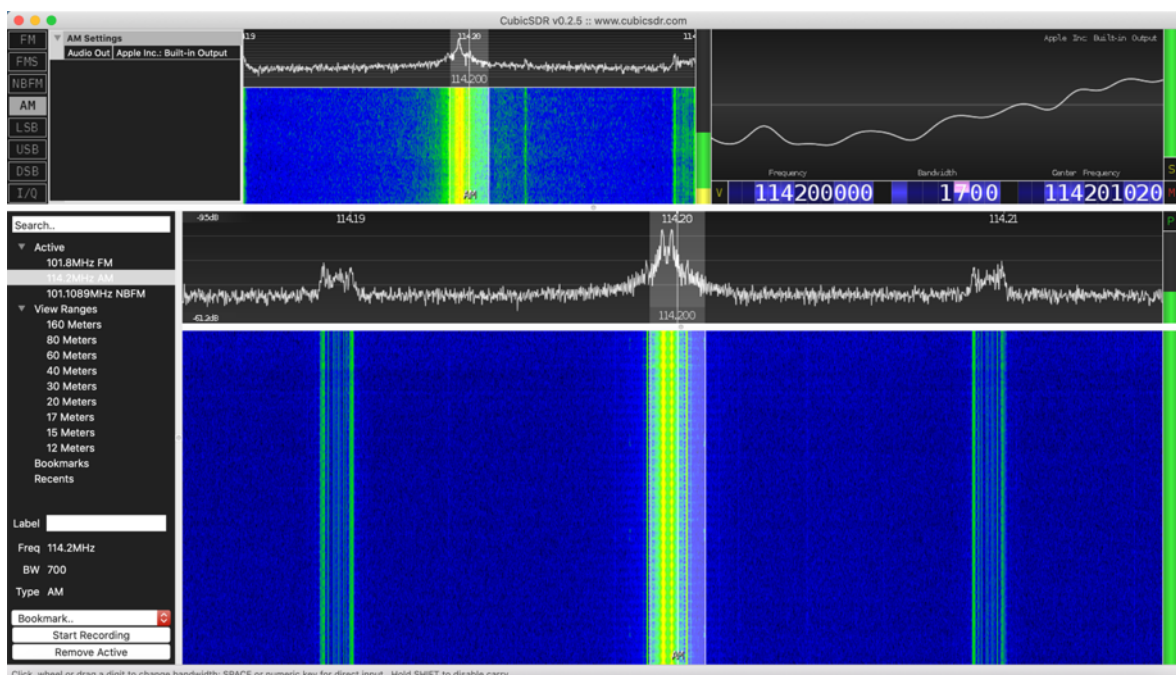


Figura 41. Ejemplo de una captura de una señal de un VOR con el software Cubic SDR.

PARTE PRÁCTICA

7 Diseño del Receptor de Señales de los VOR

El objetivo principal de este trabajo es captar señales de un VOR, desmodular estas señales y mostrar los resultados de la demodulación de forma que un piloto pueda interpretar la información de forma clara y eficaz [31, 32].

7.1 Hardware Utilizado

Para la captación de las señales de los VOR se ha utilizado un RTL-SDR de la marca NooElec, concretamente es el “*NooElec NESDR SMARTEE XTR SDR*” [33]. Este RTL-SDR tiene un rango de captación de frecuencias que va desde los 55 MHz hasta los 2,3 GHz, con una ganancia ajustable por software. La entrada de antena es un conector SMA hembra, el cual sobresale de la caja de blindaje de aluminio que evita el escape de las emisiones radiadas. La comunicación de datos se hace mediante un conector USB.

Además del RT-SDR, también se ha usado un amplificador de señal para obtener niveles de potencia más altos y para fijar el factor de ruido lo más bajo posible, sobretodo en zonas donde la cobertura de los VOR es débil. Este amplificador se colocará lo más cercano posible a la antena receptora para evitar que se induzca aún más ruido indeseado.

Dicho amplificador también es del fabricante NooElec, concretamente es el “*NooElec LaNA Barebones – Wideband Ultra Low-Noise Amplifier (LNA) Module 20 MHz-4GHz*” [34]. Dicho amplificador es compatible con la tecnología *bias tee* (explicada en el siguiente párrafo), además de que también se puede alimentar mediante un conector Micro USB o alimentado directamente a tensión continua de entre 3,3 V a 5 V. Por lo tanto, según los datos proporcionados por el fabricante, los cuales se muestran en la Figura 43, se trata de un amplificador de banda ancha, que aproximadamente amplifica unos 24 dB en la banda de VHF donde trabajan los VOR, induciendo menos de 1 dB de ruido al sistema.

Ambos dispositivos se deben conectar de la forma que se muestra en la Figura 42. Se conectarán mediante adaptadores SMA a SMA.



Figura 42. De izquierda a derecha, el amplificador de NooElec y el SDR con tecnología Bias Tee [33, 34].

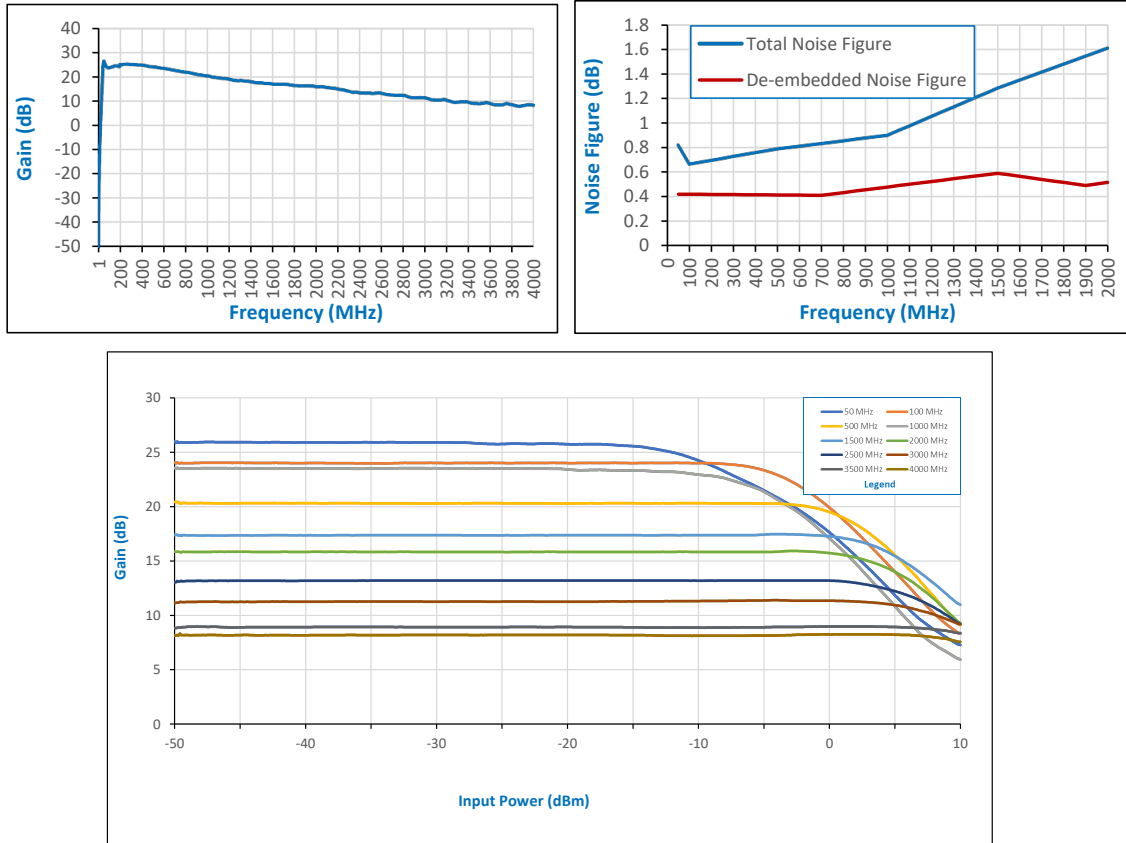


Figura 43. Características del amplificador utilizado [35].

Una de las características principales de este RTL-SDR en concreto es que posee la tecnología *bias tee*. El *bias tee* es un diplexor usado para separar las señales de RF de las señales en DC [36]. Esto permite colocar un amplificador compatible con esta tecnología justo antes de la entrada del RTL-SDR, cuya alimentación vendría suministrada por la señal DC enviada por el RTL-SDR en sentido contrario al flujo de RF, evitando así tener que alimentar el amplificador con una fuente externa de energía y evitar también la presencia de cables molestos en la disposición de todo el sistema de recepción. El esquema circuital de este sistema se muestra en la Figura 44.

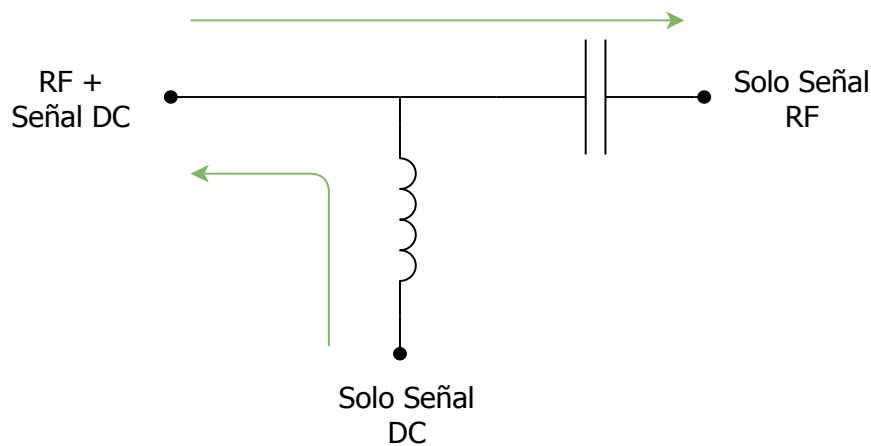


Figura 44. Esquema circuital del sistema Bias Tee.

El esquema de esta tecnología es el que se muestra en la Figura 44. Un condensador en serie solamente dejará pasar las señales de RF mientras que el inductor en paralelo bloqueará las señales de RF y solamente dejará pasar las señales en DC. Esto va a permitir alimentar perfectamente los amplificadores que sean compatibles con esta tecnología sin afectar al rendimiento de la señal.

7.2 Análisis de las Antenas Apropriadas para la Recepción de Señales de los VOR

Para la recepción de señales VHF se suelen usar antenas tipo dipolo o monopolo para señales poco directivas, y antenas Yagi o espiras para las señales más directivas [37]. Las características de estas antenas varia según la frecuencia de trabajo y las aplicaciones para las que se requieren.

Para el caso concreto de los VOR, dado que se trata de una señal omnidireccional y poco directiva, se suelen usar antenas tipo monopolo instaladas en el fuselaje de los aviones o antenas en forma de alerón colocadas debajo de las alas o en la parte inferior del fuselaje del avión. Se muestra un ejemplo de la instalación de antenas VHF en una aeronave en la Figura 45. En la ecuación 60 que la longitud óptima de esta antena es de un cuarto de longitud de onda, mientras que para las antenas tipo dipolo es la mitad de la longitud de onda.



Figura 45. Antenas VHF para la recepción de señales VOR en el fuselaje de un avión.

Para los VOR, dado que trabajan en la banda de 108 a 118 MHz (113 MHz de media), la longitud óptima de una antena monopolo será:

$$\lambda = \frac{c}{f} = \frac{3 \times 10^8}{113 \times 10^6} = 2,65 \text{ m} \quad (62)$$

$$l_{\text{monopolo óptima}} = \frac{\lambda}{4} = \frac{2,65}{4} \approx 0,66 \text{ m} \quad (63)$$

La aplicación de las antenas monopolo es muy extendida, ya que pueden funcionar para telefonía, radiodifusión y para comunicaciones HF, VHF y UHF.

Las propiedades de esta antena, cuyo diagrama de radiación se muestra en la Figura 46:

- Tienen polarización vertical.
- Presentan una ganancia de unos 4,76 dB.
- Trabaja con sus condiciones óptimas en un margen de frecuencias muy amplio, especialmente en bajas frecuencias.
- Presentan una impedancia de 37Ω .
- El ancho del haz a -3dB del diagrama de radiación es de aproximadamente 78° .

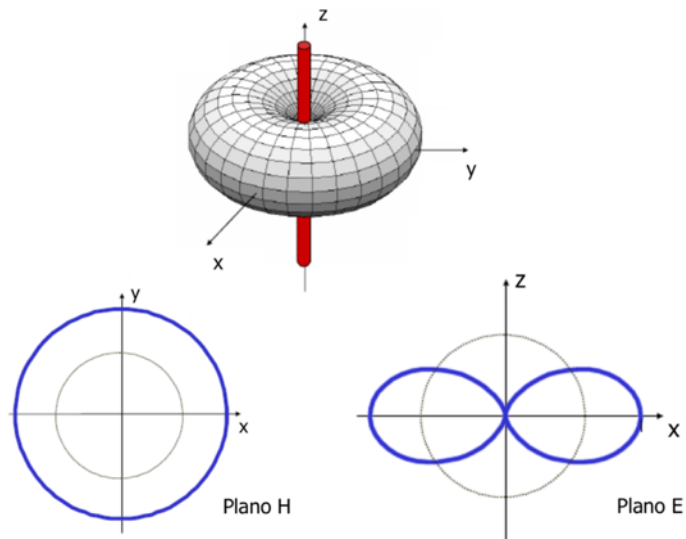


Figura 46. Diagrama de radiación de una antena monopolo.

Así pues, para la elaboración de este trabajo se ha utilizado la antena telescópica de la Figura 47, extensible hasta un máximo de 70 centímetros. Esta antena puede ser instalada en una base provista de un potente imán, el cual permite sujetar la antena en cualquier superficie metálica. Esta base de antena viene con un cable coaxial instalado que en su extremo opuesto tiene un conector SMA macho para poderlo conectar al RTL-SDR o a un amplificador.



Figura 47. Antena extensible utilizada en el proyecto.

8 Estructura del Código del Demodulador de las Señales del VOR

Todo el tratamiento digital, desde el filtrado de señal hasta la demodulación y la muestra de los resultados obtenidos, se ha realizado con MATLAB. MATLAB es un programa con mucha diversidad de funcionalidades y un catálogo de librerías muy amplio, llamadas Add-Ons. Para este trabajo, se han usado los siguientes Add-Ons:

- MathWorks DSP System Toolbox [38]
- MathWorks Communications System Toolbox [39]
- MathWorks Signal Processing Toolbox [40]
- MathWorks RTL-SDR Hardware Support Package [41]

8.1 Conexión entre el RTL-SDR y MATLAB

El funcionamiento del SDR es peculiar en cuanto a la captación de señales y el tratamiento digital se refiere, ya que la captura de señal y el tratamiento no se hacen simultáneamente, sino que se producen de forma secuencial.

Por lo tanto, en primer lugar, el RTL-SDR captará una porción de señal (de tiempo definido) que posteriormente enviará digitalizada por los buses USB para que el software, en este caso MATLAB, haga el tratamiento pertinente.

Así pues, se debe definir el tiempo de captura de cada porción de señal del SDR, que vendrá dado por la siguiente relación entre la frecuencia de muestreo de la señal digitalizada y el número de muestras que se deseen insertar en la captura de datos:

$$\text{tiempo captura} = \frac{\text{muestras captura}}{\text{frecuencia de muestreo}} \quad (64)$$

Para obtener una visualización continua de datos, como cabe esperar, se deberá repetir en bucle el proceso de captación de señal y tratamiento, colocando las porciones capturadas una detrás de la otra.

Por otro lado, es conveniente que la demodulación de la señal variable del VOR, modulada en FM, se realice en banda base. Para tratar esta señal en banda base existen dos opciones posibles: la primera, trasladar la frecuencia de la portadora de la FM a banda base mediante un multiplicador y, la segunda, centrar la portadora de la señal modulada en FM a la frecuencia central del SDR. La primera opción parece la más acertada, sin embargo, la segunda opción evitará tener que realizar el proceso de conversión en frecuencia. Asimismo, en este trabajo se ha optado por usar la segunda opción.

Es por este motivo, la frecuencia central del SDR será la frecuencia de la portadora AM del VOR menos 9960 Hz, de manera que se hace coincidir la frecuencia central del SDR con la frecuencia de la portadora FM de la banda lateral inferior (es indiferente la banda escogida, de hecho, también se podría haber hecho con la banda lateral superior). Se utilizará un filtro paso bajo para desmodular esta señal FM, mientras que para la demodulación de las señales AM se utilizará un filtro paso banda.

8.2 Declaración del Objeto SDR

Para declarar el RTL-SDR en MATLAB se debe usar un objeto llamado *SDRRTLReceiver*, que viene incluido en un paquete de los *Add-Ons* mencionados en el capítulo 8 de este trabajo. Este objeto permite inicializar algunos parámetros para que el RTL-SDR realice la digitalización de la señal de la forma más deseada.

En concreto permite:

- **Determinar el código de la dirección del RTL-SDR** (en caso de que hubiera varios RTL-SDR conectados a la vez). En este caso, dado que solamente se usa un único RTL-SDR, la dirección del RTL-SDR será la dirección “0”.
- **Determinar la frecuencia central.** En este caso, este parámetro debe estar comprendido entre 108 MHz y 118 MHz, ya que es la banda de frecuencias en la que funcionan los VOR.
- **Determinar una regulación de ganancia automática.** Este parámetro es cómodo de usar, pero puede dañar las señales más débiles. Por lo tanto, para este caso, se define en “false” para deshabilitar esta función.
- **Determinar un valor de ganancia fija.** Este valor, tras un estudio de prueba y error, se ha determinado que debe estar en 40 dB de ganancia sin el amplificador y a 20 dB con el amplificador conectado a la red receptora.
- **Determinar la frecuencia de muestreo de la señal digital a la salida del convertidor.** Esta frecuencia no debe ser muy baja ya que se perderían los detalles de la señal, pero tampoco puede ser un valor muy elevado, ya que podría afectar al rendimiento del programa. Tras un estudio de prueba y error, se ha determinado que el valor óptimo es de 240 kHz.
- **Numero de muestras por captura.** Es el numero de muestras que se desea obtener en cada captura. Tras un estudio de prueba y error, se ha determinado que el valor óptimo es de 375000 muestras por captura.

8.3 Declaración del Objeto de los Filtros

Los filtros también deben declararse al inicio del programa para ahorrar recursos a la hora de su ejecución. Se declararán filtros de respuesta finita al impulso, o FIR. Se hará con la función “fir1”, la cual permite inicializar el objeto de un filtro digital de respuesta finita. Para la inicialización del objeto del filtro, se debe especificar el tipo de filtro y la o las frecuencias de corte, además de especificar el orden de dichos filtros.

Los filtros FIR son un tipo de filtro digital que, si la entrada es una señal impulso, la salida tendrá un número finito de términos no nulos. Se caracterizan por ser sistemas no recursivos, es decir, no retroalimentados [42, 43]. Estos filtros pueden diseñarse para tener fase lineal y siempre son estables ya que todos los polos están en el origen.

Un filtro FIR de orden N se puede expresar mediante la siguiente expresión, donde $x(n)$ es la señal de entrada y donde los términos a_k son los coeficientes del filtro:

$$y_n = \sum_{k=0}^{N-1} a_k x_{n-k} = a_0 x_n + a_1 x_{n-1} + a_2 x_{n-2} + \dots + a_N x_{n-(N-1)} \quad (65)$$

La salida se puede expresar como la convolución de la señal de entrada $x(n)$ con la respuesta al impulso $h(n)$:

$$y_n = \sum_{k=0}^{N-1} h_k x_{n-k} \quad (66)$$

Aplicando la transformada Z a la expresión anterior:

$$H(z) = \sum_{k=0}^{N-1} h_k z^{-k} = h_0 + h_1 z^{-1} + \dots + h_{N-1} z^{-(N-1)} \quad (67)$$

La respuesta al impulso $h(n)$ es de duración finita. Esto significa que si la entrada se mantiene durante N períodos en cero la salida también será cero.

Para que la transición entre la banda de paso y la banda rechazada del filtro sea lo más precisa posible, se ha determinado el valor del orden de los filtros a 1000.

Dado que la señal está centrada 9960 Hz por debajo de la portadora de la señal modulada en AM (es decir, centrada en la portadora de la señal FM de la banda lateral inferior, que también se podría haber centrado en la portadora de FM de la banda lateral superior), el filtro paso bajo solamente admitirá la señal variable modulada en FM.

Para las señales de los datos del código morse y la señal de referencia modulada en AM se usarán dos filtros paso banda.

```
ordren_filtros = 1000;

% Filtro FIR paso banda (señal AM)
fn1 = (offset - 1030)/(fs/2);
fn2 = (offset + 1030)/(fs/2);
filtro_paso_banda_AM = fir1(ordren_filtros,[fn1 fn2], 'bandpass');

% Filtro FIR paso banda (codigo morse)
gn1 = (1015)/(fs/2);
gn2 = (1025)/(fs/2);
filtro_paso_banda_morse = fir1(ordren_filtros,[gn1 gn2], 'bandpass');

% Filtro FIR paso bajo (señal FM)
hn1 = (1e3)/(fs/2);
filtro_paso_bajo = fir1(ordren_filtros, hn1, 'low');
```

Código 1. Inicialización de los objetos de los filtros.

8.4 Declaración del Objeto de Audio

El objeto de audio se usará para la reproducción del código morse. Este se declara mediante el objeto llamado *Device Writer*. A este objeto se le debe especificar la frecuencia de muestreo de la señal que reproducirá.

8.5 Demodulación de la Señal AM

En la demodulación AM se debe averiguar la señal que queda impresa en la forma de la amplitud de la onda. A este proceso se le llama detección de la envolvente [13], cuyo esquema circuital se encuentra en la Figura 48.

Para desmodular una señal AM se requiere de un buen sincronismo, es decir, misma frecuencia y fase que la usada en el proceso de modulación. También se puede aplicar un rectificador de media onda u onda completa en la entrada del circuito del demodulador de la señal.

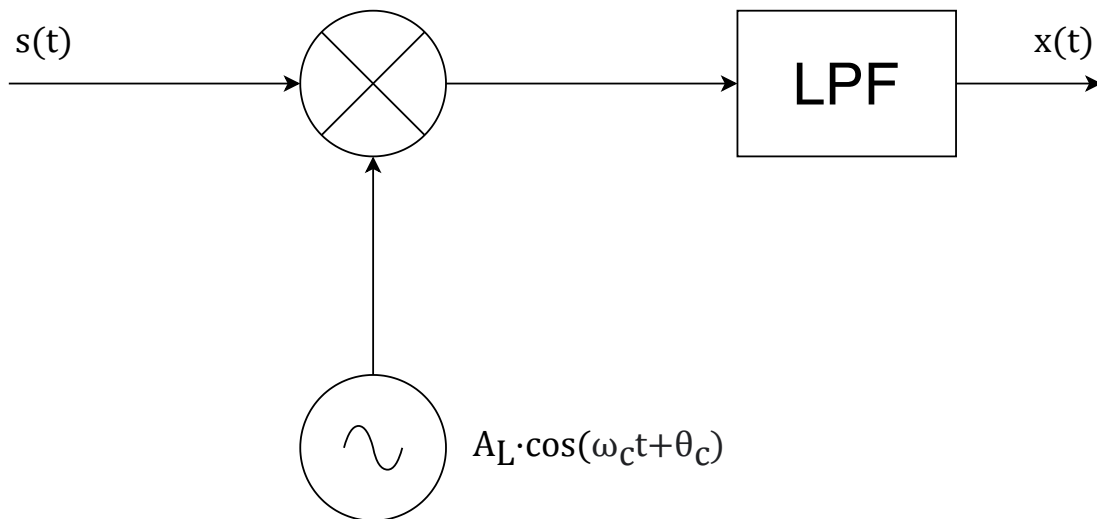


Figura 48. Esquema de un demodulador de señales moduladas en AM.

El modelo matemático se puede expresar de la siguiente forma. Partiendo de la señal $s(t)$ recibida en el demodulador, y de la señal del oscilador local:

$$s(t) = A_c \cdot [1 + m \cdot x(t)] \cdot \cos(\omega_c t + \theta_c) \quad (68)$$

$$A_L \cdot \cos(\omega_c t + \theta_c) \quad (69)$$

A la salida del multiplicador se obtiene la siguiente señal:

$$\begin{aligned} r(t) &= A_c \cdot A_L \cdot [1 + m \cdot x(t)] \cdot \cos^2(\omega_c t + \theta_c) = \\ &= A_c \cdot A_L \cdot \cos^2(\omega_c t + \theta_c) + m \cdot x(t) \cdot A_c \cdot A_L \cdot \cos^2(\omega_c t + \theta_c) \end{aligned} \quad (70)$$

Operando, se obtiene que:

$$x(t) = \frac{1}{2} \cdot A_L \cdot A_c \cdot [1 + m \cdot x(t)] \cdot [1 + \cos(2\omega_c t)] \quad (71)$$

Tras filtrar la señal por el filtro paso bajo, se separa la señal fundamental centrada en omega de los armónicos superiores que han aparecido debido a la demodulación.

$$x(t) = \frac{1}{2} \cdot A_L \cdot A_c \cdot [1 + m \cdot x(t)] \quad (72)$$

De esta forma, se ha eliminado la señal portadora y únicamente queda la señal moduladora con la información de interés.

La señal AM no queda centrada con la frecuencia central del RTL-SDR, por lo tanto, se debe filtrar mediante un filtro paso banda. Este proceso trasladará la señal AM a banda base, donde se podrá seguir el proceso de demodulación.

Como se ha explicado en el apartado 6.2 de este trabajo, las señales recibidas por el RTL-SDR están desmoduladas en IQ antes de ser digitalizadas, lo que quiere decir que los datos que recibe el software del tratamiento digital están divididos en los dos canales I y Q. En consecuencia, se trata de una señal compleja.

El objetivo principal es obtener la información contenida en la envolvente de la señal AM. Un detector de envolvente es un objeto no-lineal cuya representación matemática no es trivial. No obstante, se puede expresar mediante la siguiente convolución:

$$s_{demodulada}[n] = h_{LPF}[n] \otimes \{Re[s_{AM}(t)]\} \quad (73)$$

Para la demodulación de la señal AM se ha usado el siguiente bloque de código.

```
% Demodulación AM
senalAM = filter(filtro_paso_banda_AM, 1, datos_sdr);
senalAMdemodulado = abs(senalAM);

% Filtrado señal portadora del código morse
senalMorse = filter(filtro_paso_banda_morse, 1, senalAMdemodulado);
senalMorse = abs(senalMorse);
```

Código 2. Filtrado y demodulación de la señal modulada en AM.

También se puede usar la función *amdemod* de MATLAB, la cual realiza las operaciones anteriores de forma automática.

8.6 Demodulación de la Señal FM

En el apartado 3.2 se ha demostrado la expresión resultante de la modulación IQ de las señales AM y FM. Por lo tanto, partiendo de dicho concepto y de la señal genérica modulada en FM:

$$s(t) = A_c \cdot \cos \left[\omega_c t + 2\pi K_0 \int_{-\infty}^t x(t) dt \right] \quad (74)$$

Se puede expresar la señal de la siguiente forma:

$$\begin{aligned} s_{fm}(t) &= s(t) \cdot e^{-j\omega_c t} = \\ &= \frac{A_c}{2} \cdot \left[e^{j(\omega_c t + 2\pi K_0 \int_{-\infty}^t x(t) dt)} + e^{-j(\omega_c t + 2\pi K_0 \int_{-\infty}^t x(t) dt)} \right] \cdot e^{-j\omega_c t} \end{aligned} \quad (75)$$

Arreglando la ecuación anterior:

$$s_{fm}(t) = \frac{A_c}{2} \cdot \left[e^{j \cdot (2\pi K_0 \int_{-\infty}^t x(t) dt)} + e^{-j \cdot (2 \cdot \omega_c t - j 2\pi K_0 \int_{-\infty}^t x(t) dt)} \right] \quad (76)$$

Cuando la señal anterior pasa a través del filtro del modulador IQ, las frecuencias dobles se rechazan y solamente pasan las frecuencias en banda base.

$$s_{fm}(t) = \frac{A}{2} \cdot e^{j \cdot (2\pi K_0 \int_{-\infty}^t x(t) dt)} \quad (77)$$

Se puede simplificar de la siguiente forma:

$$\phi(t) = 2\pi K_0 \int_{-\infty}^t x(t) dt \quad (78)$$

Por lo tanto:

$$s_{fm}(t) = \frac{A}{2} \cdot e^{j \cdot \phi(t)} \quad (79)$$

Se puede utilizar un demodulador de retardo en banda base, que es el que se expresa en la Figura 49.

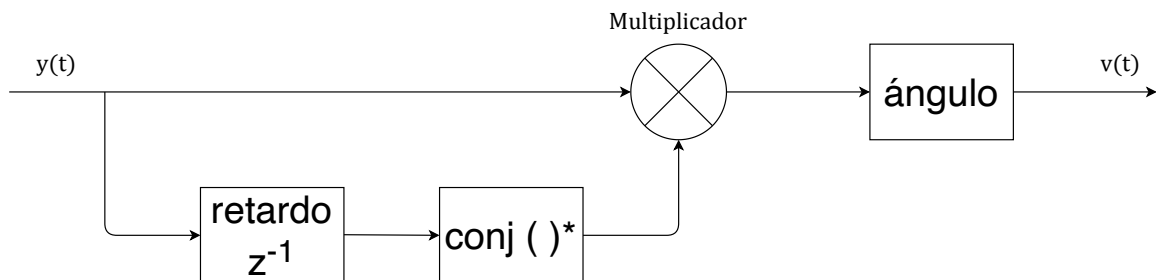


Figura 49. Esquema del algoritmo de demodulación de señales moduladas en FM.

La salida del multiplicador del modulador anterior es el resultado de la multiplicación de la señal original por el conjugado de la señal original retardada [44]. Matemáticamente se puede expresar de la siguiente manera, donde T es el tiempo entre muestras:

$$w(t) = \frac{A_c^2}{4} \cdot e^{j \cdot \phi(t)} \cdot e^{-j \cdot \phi(t-T)} = \frac{A_c^2}{4} \cdot e^{j \cdot \phi(t) - \phi(t-T)} \quad (80)$$

En la salida del bloque demodulador obtendremos la señal desmodulada, que corresponde al ángulo del número complejo resultante.

$$x(t) = \phi(t) - \phi(t - T) \quad (81)$$

```
% Demodulación FM
senalFM = filter(filtro_paso_bajo, 1, datos_sdr);
senalFMretardado = step(dsp.Delay, senalFM);
```

```
senalFMconjugado = conj(senalFM);  
multiplicacion = senalFMretardado .* senalFMconjugado;  
senalFMdemodulado = angle(multiplicacion);
```

Código 3. Filtrado y demodulación de la señal modulada en FM.

También se puede usar la función *fndemod* de MATLAB, la cual realiza las operaciones anteriores de forma automática.

8.7 Diezmado de las Señales Desmoduladas

Para reducir el número total de muestras de las señales anteriormente desmoduladas y para aumentar la relación señal a ruido, se utiliza la técnica de diezmar la señal, la cual consiste en eliminar algunas muestras que acostumbran a ser redundantes y que no afectan al resultado definitivo de la demodulación. También servirá para mejorar el rendimiento del programa.

En el demodulador de MATLAB se ha comprobado que se puede diezmar la señal en un factor 20 sin perder relativamente calidad de señal. Tras este valor hay algunas medidas que pueden fallar.

Para el diezmo de la señal se ha utilizado la función *downsample* de MATLAB.

```
% Diezmado por "factor_diezmado"  
senalAMdemodulada_diezmado = downsample(senalAMdemodulado,  
factor_diezmado);  
senalFMdemodulada_diezmado = downsample(senalFMdemodulado,  
factor_diezmado);  
senalMorse_diezmado = downsample(senalMorse, factor_diezmado);
```

Código 4. Diezmado de las señales demoduladas.

8.8 Cálculo de la Diferencia de Fase

Esta porción de código recoge dos variables, que son las dos señales desmoduladas (AM y FM) cuya fase se desea calcular, para posteriormente calcular el valor del desfase entre ambas señales [45].

En primer lugar, se reordena la lista de datos en una columna para que en el siguiente paso se pueda aplicar la función “mean” de MATLAB. Dicha función devuelve la media de los elementos de la columna de datos cuyo tamaño no es igual a 1. Se calculará la media para posteriormente restarla a la función original; con esto se conseguirá eliminar la componente DC de la señal, si existe.

Posteriormente se calcula la dimensión de los vectores para crear una ventana. Las ventanas sirven para acotar la señal y poder así calcular la transformada de Fourier de las señales en un ancho de banda definido, y no infinito como dicta la explicación teórica. Se calculará mediante el algoritmo de la transformada rápida de Fourier o FFT.

En último lugar se cogerá la parte real del resultado de la transformada y se calculará su valor máximo, es decir, su frecuencia fundamental. Por lo tanto, cuando se ha detectado la frecuencia que presenta un pico de potencia, se calculan los ángulos de ambas señales y se restan entre ellos, dando lugar a la diferencia de fase.

9 Diseño de un Algoritmo para el Cálculo de la Posición de un Avión en Vuelo

Este algoritmo se ha diseñado con el propósito de posicionar una aeronave en movimiento en un mapa con, simplemente, la recepción de las señales de un VOR, sin uso de ningún otro sistema de posicionamiento como el DME o el GPS. El diseño de este algoritmo se ha ingeniado desde cero. A pesar de que en algunas situaciones puede perder precisión, sobretodo cuando la señal del VOR es débil y comienzan a aparecer errores de un rango mayor a $\pm 1^\circ$, su funcionamiento es fiable y todos los cálculos se realizan en el software instalado en el avión.

Este sistema servirá de gran ayuda en situaciones donde se realicen aproximaciones a pista con el soporte de las señales VOR o simplemente como elemento redundante para conocer la posición de la aeronave. Dicho algoritmo no solamente determina la posición de la aeronave, sino que también es capaz de calcular el *heading* del avión para posteriormente indicárselo al piloto, tal y como se muestra en la Figura 50.

Sin embargo, su funcionamiento no da el resultado de forma inmediata y tampoco funciona en todas las situaciones de vuelo, ya que deben cumplirse las siguientes condiciones:

- La posición del VOR debe ser conocida (lo cual, como se ha mencionado en apartados anteriores de este trabajo, no es un problema ya que todos los VOR están correctamente localizados en las cartas aeronáuticas) y sirve como punto de partida para el cálculo de la posición final de la aeronave.
- El avión debe estar volando en línea recta (que prácticamente es la gran parte del tiempo que el avión pasa en el aire).
- La velocidad respecto al suelo (*ground speed*) debe ser constante durante todo el tiempo que se invierte en realizar las lecturas de los radiales, que son las variables de entrada del algoritmo.
- El intervalo de tiempo entre las lecturas de los radiales ha de ser constante.

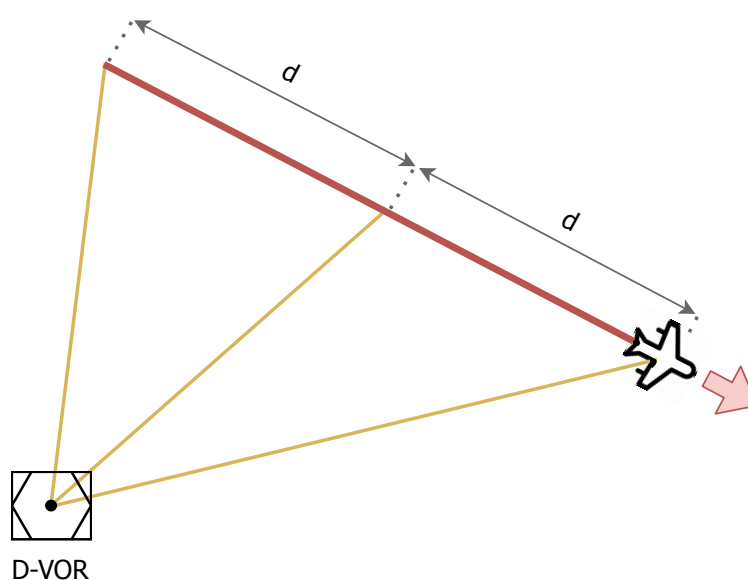


Figura 50. Diagrama de los radiales captados y la trayectoria seguida por el avión.

Cumpliendo estas condiciones se puede determinar la posición del avión y la dirección hacia la cual está volando mediante cálculos matemáticos que se basan en el teorema del seno. Notar que los resultados del algoritmo se obtendrán pasados varios intervalos de lectura de los radiales (mínimo 2 intervalos completos y la lectura de 3 radiales). Por lo tanto, los resultados no se obtendrán de forma inmediata y, además, dependerán de las lecturas de los radiales anteriores. Cada nueva lectura de un radial dependerá de las dos últimas lecturas de los radiales anteriores, pasa así siempre tener 3 radiales sobre los cuales poder hacer el cálculo.

9.1 El Teorema del Seno

El teorema del seno relaciona proporcionalmente los ángulos de un triángulo como el de la Figura 51 con el lado opuesto al ángulo de dicho triángulo. Matemáticamente se relacionan con la siguiente expresión:

$$\frac{a}{\sin(A)} = \frac{b}{\sin(B)} = \frac{c}{\sin(C)} \quad (82)$$

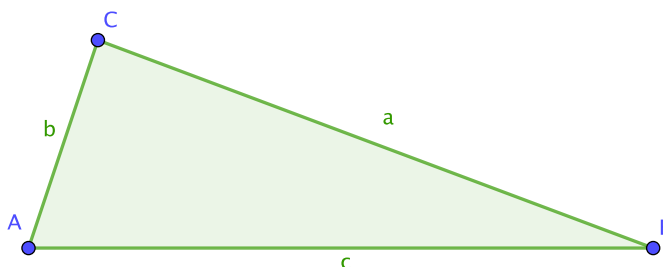


Figura 51. Lados y vértices de un triángulo cualquiera.

Este teorema es útil para determinar todos los ángulos del triángulo y la longitud de sus lados en caso de conocer las siguientes variables:

- Se conoce la longitud de dos lados del triángulo y el ángulo opuesto a uno de ellos.
- Se conocen dos ángulos y la longitud del lado opuesto de uno de ellos. Si el lado en cuestión no es el opuesto a ninguno de los dos ángulos conocidos, se puede determinar el tercer ángulo mediante la suma de todos los ángulos de un triángulo, que debe ser 180°.

9.2 Variables y Lecturas Iniciales

Antes de pasar por la parte puramente matemática del algoritmo, es de estricta necesidad preparar y adaptar los datos obtenidos de manera que se introduzcan con el formato correcto en los cálculos trigonométricos pertinentes. Para el funcionamiento básico del algoritmo se deberán tener en cuenta los últimos 3 radiales.

Primeramente, es preferible evitar que los radiales leídos sean 0, 90, 180 y 270 grados, ya que el resultado de hacer operaciones trigonométricas con estos números puede dar lugar a ceros o infinitos en el numerado o denominador y causar errores de ejecución del programa.

Por este motivo, los radiales entrantes que sean igual a los ángulos anteriormente mencionados se les sumará un pequeño offset de 0,01 grados que evitará que las operaciones con los senos y cosenos den lugar a números diferentes de cero e infinito.

Posteriormente se procederá a realizar el cálculo de la distancia recorrida por el avión, la cual se hará con el valor del tiempo que separa la lectura de dos radiales y la velocidad constante del avión.

$$\Delta x = v \cdot \Delta t \quad (83)$$

9.3 Planteamiento del Algoritmo

El objetivo principal es estimar la distancia que separa el avión del VOR. Dado que la velocidad del avión es constante y los intervalos de tiempo en los que se toma la lectura de los radiales son iguales, la distancia recorrida entre la lectura de radiales también será igual.

El punto de partida son los siguientes datos:

- Se conocen los radiales de cada lectura, por lo tanto, el ángulo entre los radiales de cada lectura es conocido (rectas de color amarillo de la Figura 51).
- Se conoce la distancia que el avión ha recorrido en cada intervalo de tiempo, independientemente de la dirección de desplazamiento (recta de color rojo de la Figura 52).

Con estos datos, solamente existe una combinación posible que cumpla las condiciones expuestas. Se trata de dos segmentos concatenados de longitud igual a la distancia recorrida por el avión entre las lecturas de dos radiales, los cuales forman una línea recta y cuyos extremos de ambos segmentos coinciden con dos de las rectas que definen los radiales capturados.

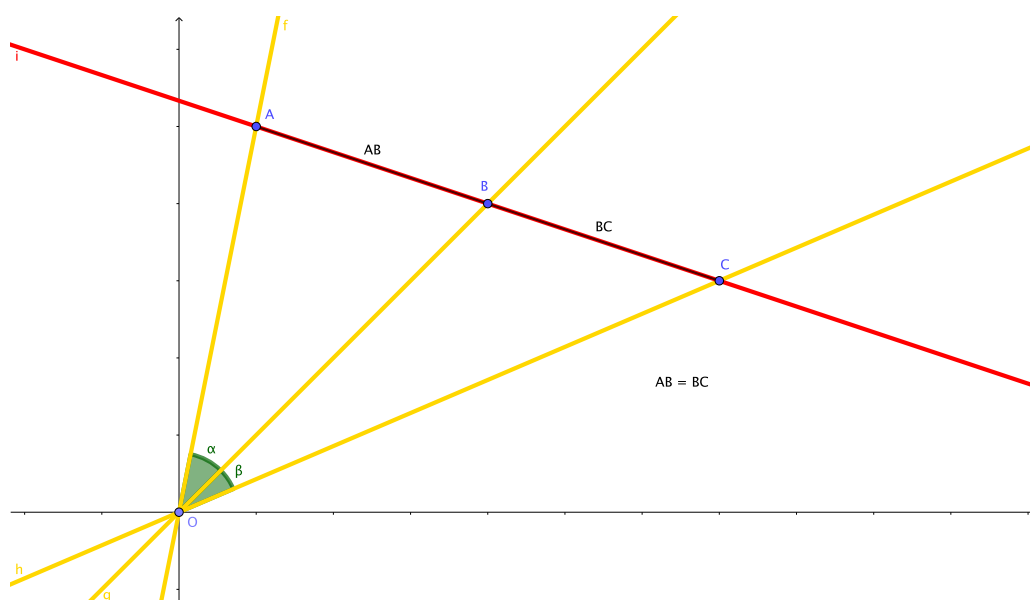


Figura 52. Representación de los radiales y la trayectoria seguida por el avión en el plano cartesiano.

9.4 Cálculo de los Ángulos Contiguos a la Línea de Desplazamiento del Avión

Con el fin de poder aplicar el teorema del seno para determinar la distancia entre el VOR y el paso del avión por cada radial capturado se necesita saber un dato más, ya que inicialmente se conoce un solo lado del triángulo y su ángulo opuesto. Sin embargo, el tercer dato que falta se puede obtener mediante la resolución de un sistema de ecuaciones y el teorema del seno aplicado a los dos triángulos más pequeños, que se definen en la Figura 53.

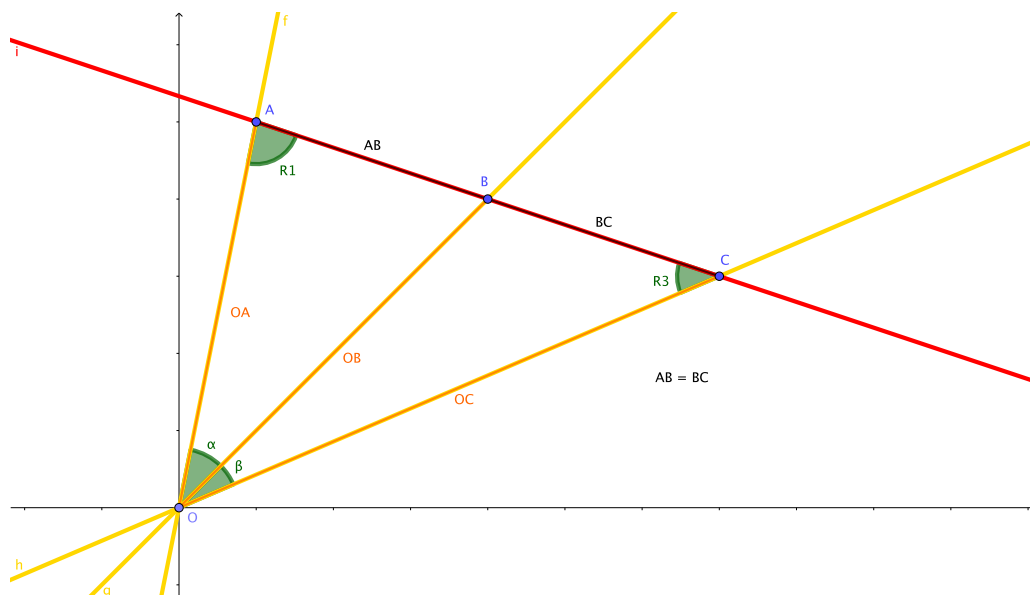


Figura 53. Distancia entre el VOR y la trayectoria del avión para cada radial capturado.

Notar que ambos triángulos comparten un lado (el radial del centro), por lo tanto, el tercer dato que falta podría ser la longitud de dicho lado o su ángulo opuesto, que es el ángulo contiguo a la línea de desplazamiento del avión. A partir de esta posible resolución, se plantean las siguientes ecuaciones:

$$\frac{AB}{\sin(\alpha)} = \frac{OB}{\sin(R_1)} = \frac{OA}{\sin(180 - \alpha - R_1)}$$

$$\frac{BC}{\sin(\beta)} = \frac{OB}{\sin(R_3)} = \frac{OC}{\sin(180 - \alpha - R_1)}$$
(84)

Donde los lados AB y BC son variables conocidas (y de igual valor) y los ángulos alfa y beta también. La longitud del lado OB es el mismo valor para ambas ecuaciones ya que comparten dicho lado.

Las longitudes OA y OC de ambos triángulos no aportan información relevante a la resolución, ya que no se conoce su longitud ni su ángulo opuesto y tampoco son un dato compartido entre ambos triángulos.

Notar que, despreciando los términos de las igualdades de OA y OC, se tienen 3 variables desconocidas y dos ecuaciones. Se necesitará una tercera ecuación, que será la que relaciona la suma de los ángulos alfa y beta, R₁ y R₃. El total de esta suma ha de ser 180.

$$180 = \alpha + \beta + R_1 + R_3 \quad (85)$$

Por lo tanto, se puede plantear el siguiente sistema de 3 ecuaciones y 3 incógnitas.

$$\begin{cases} \frac{AB}{\sin(\alpha)} = \frac{OB}{\sin(R_1)} \\ \frac{BC}{\sin(\beta)} = \frac{OB}{\sin(R_3)} \\ \alpha + \beta + R_1 + R_3 = 180 \end{cases} \quad (86)$$

Donde:

$$AB = BC = d \quad (87)$$

Para obtener el valor del ángulo R1, se puede empezar planteando la siguiente igualdad:

$$\begin{aligned} OB &= \frac{d}{\sin(\alpha)} \cdot \sin(R_1) \\ OB &= \frac{d}{\sin(\beta)} \cdot \sin(R_3) \end{aligned} \quad \rightarrow \quad \frac{d}{\sin(\alpha)} \cdot \sin(R_1) = \frac{d}{\sin(\beta)} \cdot \sin(R_3) \quad (88)$$

El valor de R3 viene condicionado por:

$$R_3 = (180 - \alpha - \beta) - R_1 \quad (89)$$

Dado que los valores de alfa y beta son conocidos, todo lo que se encuentra en el paréntesis de la expresión anterior también será un valor numérico conocido v.

$$v = 180 - \alpha - \beta \quad (90)$$

Por lo tanto:

$$\frac{d}{\sin(\alpha)} \cdot \sin(R_1) = \frac{d}{\sin(\beta)} \cdot \sin[v - R_1] \quad (91)$$

Aplicando la siguiente ley trigonométrica:

$$\sin(a - b) = \sin(a) \cdot \cos(b) - \cos(a) \cdot \sin(b) \quad (92)$$

Se obtiene que:

$$\frac{d}{\sin(\alpha)} \cdot \sin(R_1) = \frac{d}{\sin(\beta)} \cdot [\sin(v) \cdot \cos(R_1) - \cos(v) \cdot \sin(R_1)] \quad (93)$$

Aislado la R1:

$$\frac{\sin(v) \cdot \cos(R_1) - \cos(v) \cdot \sin(R_1)}{\sin(R_1)} = \frac{\frac{d}{\sin(\alpha)}}{\frac{d}{\sin(\beta)}} = \frac{\sin(\beta)}{\sin(\alpha)} \quad (94)$$

Arreglando la ecuación:

$$\sin(v) \cdot \frac{\cos(R_1)}{\sin(R_1)} - \cos(v) = \frac{\sin(\beta)}{\sin(\alpha)} \quad (95)$$

Y finalmente se obtiene:

$$R_1 = \arctan\left(\frac{\sin(v)}{\frac{\sin(\beta)}{\sin(\alpha)} + \cos(v)}\right) \quad (96)$$

9.5 Cálculo de la Distancia entre el VOR y el Avión

Una vez se conocen los ángulos R_1 y R_3 , se puede aplicar el mismo teorema del seno para averiguar la longitud de los segmentos OA , OB y OC .

$$OA = \frac{AB}{\sin(\alpha)} \cdot \sin(180 - \alpha - R_1) = \frac{OB}{\sin(R_1)} \cdot \sin(180 - \alpha - R_1) \quad (97)$$

$$OB = \frac{AB}{\sin(\alpha)} \cdot \sin(R_1) = \frac{BC}{\sin(\beta)} \cdot \sin(R_3) \quad (98)$$

$$OC = \frac{BC}{\sin(\beta)} \cdot \sin(180 - \alpha - R_1) = \frac{OB}{\sin(R_3)} \cdot \sin(180 - \alpha - R_1) \quad (99)$$

9.6 Cálculo de las Variables de las Rectas que Definen los Radiales y la Trayectoria del Avión

Para representar gráficamente la trayectoria del avión y los radiales capturados se hará mediante la representación gráfica de rectas en un plano de coordenadas cartesianas. Para ello, sabiendo el ángulo del radial y la distancia a la cual ha pasado el avión del VOR cuando se tomó la lectura del radial en cuestión, se pueden calcular los puntos pertenecientes a la recta de los radiales por lo cuales ha pasado el avión. Estos puntos servirán posteriormente para la representación de la recta que sigue la trayectoria del avión.

Primeramente, se deberá calcular la pendiente de los radiales. Dado que se desea representar un radial en un plano de coordenadas cartesianas, pero que viene expresado en coordenadas polares, será necesario hacer una conversión. Además, se debe corregir la referencia de 0° , ya que el radial de 90° corresponde al ángulo 0° de las coordenadas polares.

$$m = \frac{y}{x} = \tan(-90 - \text{radial}) \quad (100)$$

Seguidamente, se calculará el punto de cruce entre la recta de la trayectoria del avión y el radial en cuestión cuya pendiente ya es conocida. Para ello, se determinará los valores x e y del punto en cuestión. Sabiendo que el VOR se encuentra en el origen $(0, 0)$, y se conoce la distancia a la que se encuentra dicho punto y la pendiente de la recta, la distancia entre dos puntos se puede expresar como:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} = \{x_1 = 0, y_1 = 0\} = \sqrt{x_2^2 + y_2^2} \quad (101)$$

El punto del eje de las ordenadas puede substituirse mediante la ecuación de la pendiente de una recta, de manera que:

$$m = \frac{y}{x} \rightarrow y = mx \quad (102)$$

Por lo tanto, el valor del punto en el eje de las abscisas es:

$$d^2 = x^2 + (mx)^2 \rightarrow x = \sqrt{\frac{d^2}{1 + m^2}} \quad (103)$$

Finalmente, para determinar el valor de la y se puede hacer mediante la ecuación de la recta, sabiendo que la recta del radial corta por el origen:

$$y = m \cdot x + 0 \quad (104)$$

En definitiva, se podrá definir la recta en la forma continua que describe la trayectoria del avión mediante los tres puntos obtenidos de cada radial mediante la expresión:

$$\frac{x - x_1}{x_2 - x_1} = \frac{y - y_1}{y_2 - y_1} \quad (105)$$

9.7 Determinación del *Heading* del Avión

Notar que los radiales de cuadrantes opuestos (primero y tercero o segundo y cuarto) formaran parte de rectas con una pendiente del mismo signo. Esto es, por ejemplo, los radiales del primer cuadrante formarán rectas que tendrán el mismo signo positivo en la pendiente que los radiales del tercer cuadrante. Por lo tanto, para determinar el *heading* del avión va a ser imposible poder basarse solamente en la pendiente de las rectas, ya que no hay dependencia directa entre el *heading* y los radiales.

Por otro lado, se conocen seis puntos por lo cuales ha pasado la trayectoria del avión y la pendiente de la recta de dicha trayectoria. Simplemente con coordenadas de los últimos 2 puntos (del segundo y tercer radial) se puede determinar la pendiente de la recta de la trayectoria del avión y el sentido en el que vuela el avión.

$$m_h = \frac{y_3 - y_2}{x_3 - x_2} \quad (106)$$

En primer lugar, se debe hacer la diferencia entre las coordenadas del eje de las abscisas de ambos puntos para determinar si se obtiene un valor positivo, es decir, el avión vuela hacia la derecha del plano cartesiano; un valor negativo, es decir, el avión vuela hacia la izquierda del plano cartesiano; o cero, el avión no tiene desplazamiento horizontal.

Tras hacer esta comprobación, se debe hacer la misma diferencia en las coordenadas del eje de las ordenadas de ambos puntos para determinar si se obtiene un valor positivo, es decir, el avión vuela hacia arriba del plano cartesiano: un valor negativo, es decir, el avión vuela hacia abajo del plano; o cero, el avión no tiene desplazamiento vertical.

A partir de la obtención de esta información, se puede determinar el *heading* del avión mediante las siguientes expresiones de la Tabla 3. Los grados se pueden calcular mediante la transformación de coordenadas cartesianas a coordenadas polares, mediante la fórmula:

$$\theta = \arctan\left(\frac{y}{x}\right) = \arctan(m_h) \quad (107)$$

Desplazamiento Horizontal (eje X)	Desplazamiento Vertical (eje Y)	Desplazamiento	Grados de rotación de la brújula
Positivo	Positivo	Derecha - Arriba	$\theta = \arctan(m_h) - 90$ (108)
Positivo	Negativo	Derecha - Abajo	$\theta = \arctan(m_h) - 90$ (109)
Positivo	Cero	Derecha Horizontal	$\theta = 90^\circ$ (110)
Negativo	Positivo	Izquierda - Arriba	$\theta = \arctan(m_h) + 90$ (111)
Negativo	Negativo	Izquierda - Abajo	$\theta = \arctan(m_h) + 90$ (112)
Negativo	Cero	Izquierda Horizontal	$\theta = -90^\circ$ (113)
Cero	Positivo	Arriba Vertical	$\theta = 0^\circ$ (114)
Cero	Negativo	Abajo Vertical	$\theta = 180^\circ$ (115)

Tabla 3. Cálculo de los grados de rotación de la brújula del *heading* en función del desplazamiento horizontal y vertical del avión.

10 Diseño de la Interfaz Gráfica del Panel con los Instrumentos del VOR

Toda la parte experimental elaborada en este trabajo, tanto el demodulador de señales VOR como el algoritmo para determinar la posición la aeronave, debe poder mostrar los resultados a un piloto en la cabina de un avión de forma clara y concisa, ya que este es el objetivo principal de la existencia de los VOR.

Un hecho que ya se ha mencionado en varias ocasiones a lo largo de este trabajo es que los pilotos reciben muchos inputs a lo largo de un vuelo, tanto del propio pilotaje del avión, como de los instrumentos, de la radio de comunicaciones o de la tripulación entre muchos otros. Por lo tanto, es esencial que los pilotos puedan captar la mayor cantidad de información en el menor tiempo posible. Esto se consigue con los instrumentos adecuados que permitan que los pilotos sean capaces de tener una visión global de cómo el avión está desempeñando el vuelo y predecir las siguientes maniobras.

Es por esto por lo que, para completar los objetivos de este trabajo y cumplir con diseñar una herramienta útil para los pilotos aviadores, se ha creado una interfaz gráfica con la aplicación App Designer de MATLAB que se muestra en la Figura 54. Esta interfaz está pensada para usarla en un ordenador, a pesar de que también se puede extrapolar para ser usada en una Tablet o cualquier dispositivo portátil presente en la cabina de una aeronave.

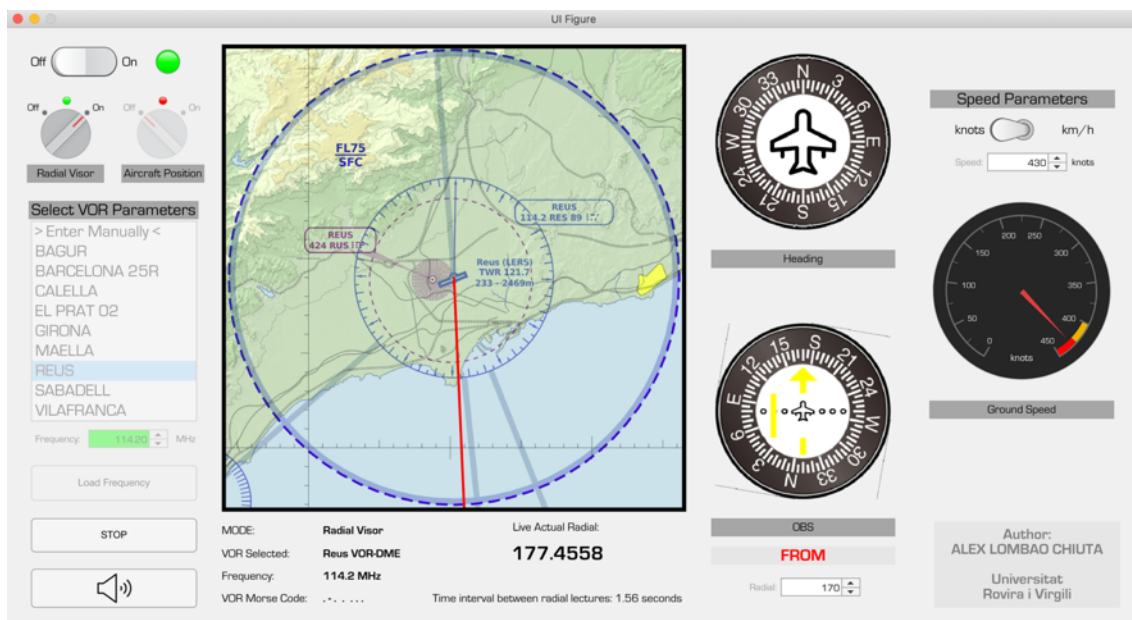


Figura 54. Diseño de la interfaz del programa.

10.1 App Designer

Según la web oficial de MathWorks:

“App Designer es un entorno de desarrollo interactivo para diseñar una aplicación y programar su comportamiento. Proporciona una versión totalmente integrada del editor de MATLAB y un gran conjunto de componentes interactivos de la IU. También ofrece un administrador de diseño de cuadrículas para organizar la

interfaz de usuario y opciones de reordenación automáticas para hacer que su app detecte y responda a los cambios del tamaño de pantalla. Permite distribuir apps empaquetándolas en los archivos del instalador directamente desde la barra de herramientas de App Designer, o bien creando una aplicación web o de escritorio independiente (se requiere MATLAB Compiler).” [46]

App Designer permite representar gráficos, añadir imágenes interactivas, tablas de datos, etc. Es decir, permite mostrar resultados experimentales mediante el uso de una aplicación interactiva con el usuario. El usuario puede controlar la aplicación mediante botones, cuadros de texto, ruedas y selectores y muchos más inputs. Es por esto por lo que esta aplicación cumple a la perfección las condiciones para el diseño de un panel de la cabina de un avión, en el que los pilotos (que serán los usuarios de esta aplicación) pueden interactuar mediante botones y otros inputs y, de esta forma, tener una visión clara de los datos recibidos de los VOR.

10.2 Estructura Visual de la Aplicación

En este apartado se mostrarán todos los elementos gráficos de la aplicación. Notar que el texto usado en la aplicación está en inglés; esto es debido a que el inglés es el idioma estándar del mundo aeronáutico y todos los pilotos del mundo deben hablarlo y entenderlo.

10.2.1 Botón de Encendido y Apagado del Panel

Es el botón general que, mientras está en posición de apagado (luz roja), no permite que el piloto interactúe con ningún otro botón del panel, mientras que al encenderlo (luz verde) permitirá al piloto sintonizar la frecuencia del VOR, colocar el CDI en la posición deseada y, si fuera necesario, indicar la velocidad de crucero que llevará el avión.

El interruptor recibe el nombre de *Switch*, mientras que la luz recibe el nombre de *System Lamp*. Su representación en la interfaz es la de la Figura 55.



Figura 55. Interruptor general de encendido y apagado del programa.

10.2.2 Tabla de Selección del VOR y Sintonizador de Frecuencia

Existen dos formas de sintonizar la frecuencia del VOR deseado:

- La primera forma de sintonizar la frecuencia de un VOR (y la más sencilla de todas) es escogiendo en la lista de VOR predeterminados el VOR por el cual se sobrevuela. Con esta opción se establecerá automáticamente la frecuencia de dicho VOR y en la pantalla principal aparecerá el VOR centrado en un mapa de los alrededores de la zona. La aplicación tiene la información de todos los VOR que hay en funcionamiento por todo el territorio catalán.

- El piloto decide la frecuencia del VOR que quiere sintonizar. En este caso no aparecerá ningún mapa, pero sí una brújula apuntando al norte magnético y los diferentes radiales del VOR.

En todos los casos, el piloto debe cargar la frecuencia apretando el botón de *Load Frequency*. Tras apretar este botón, se inicializará la radio y se habilitarán los botones para escoger el modo de funcionamiento de la radio: visualizador de radiales o posicionamiento del avión en el mapa. Para cualquier caso, se puede cancelar la ejecución apretando el botón “STOP”, donde la aplicación volverá al estado inicial (mismo estado que tras el encendido del panel).

Dado que los VOR se identifican mediante un código morse característico, este se podrá escuchar durante todo el transcurso de la ejecución del programa, a no ser que el piloto decida dejar de escucharlo, que entonces podrá apretar el botón con el altavoz dibujado.

La tabla para escoger el VOR recibe el nombre de *SelectVORParametersListBox*, el seleccionador de frecuencia *FrequencySpinner*, el botón para cargar la frecuencia *LoadFrequencyButton*, el botón de parar la ejecución *STOPButton* y el botón para silenciar o activar el audio *Sound Button*. Su representación en la interfaz es la de la Figura 56.

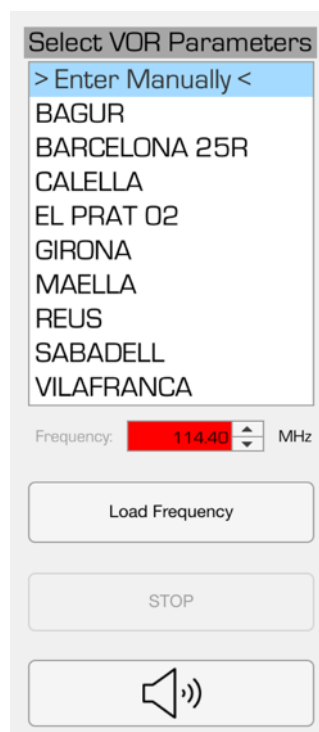


Figura 56. Panel de selección del VOR y la frecuencia.

10.2.3 Cuadro de Información del VOR Seleccionado

En este apartado se muestra la información del VOR después de que el piloto haya seleccionado un VOR de la lista. Durante el funcionamiento de la aplicación, se mostrará el modo escogido (visor de radiales o posicionamiento del avión), el nombre del VOR elegido, la frecuencia central de la portadora de la señal de dicho VOR, el código morse que identifica

al VOR y un cuadro de texto que muestra el radial actual leído. Su representación en la interfaz es la de la Figura 57.

MODE:	Radial Visor	Live Actual Radial:
VOR Selected:	Reus VOR-DME	177.4558
Frequency:	114.2 MHz	
VOR Morse Code:	.-	Time interval between radial lectures: 1.56 seconds

Figura 57. Cuadro de información general del VOR y el modo seleccionado.

10.2.4 Selector del Modo de Funcionamiento

Los dos modos de funcionamiento son el visor de radiales y la posición del avión en el mapa. Ambos botones están deshabilitados hasta que el piloto cargue una frecuencia, que será entonces cuando se habilitarán.

Durante el funcionamiento del programa, solamente estará habilitado el botón en uso, el otro se deshabilitará hasta que se cancele la ejecución en el botón *STOP*.

El botón para activar el programa en modo visor de radiales recibe el nombre de *RadialVisorKnob*, mientras que el botón para activar el programa en modo posición del avión recibe el nombre de *AircraftPositionKnob*. Su representación en la interfaz es la de la Figura 58.

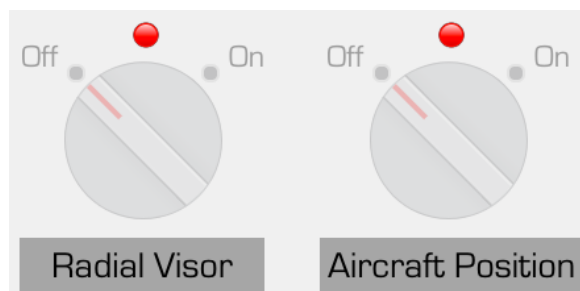


Figura 58. Interruptores para la elección del modo de funcionamiento del programa.

10.2.5 Pantalla Principal

En la pantalla principal aparecerá en todo momento el VOR seleccionado en el centro de la pantalla. Si el VOR seleccionado es uno de la lista predeterminada, aparecerá el mapa de la carta aeronáutica y las inmediaciones del VOR.

En caso contrario, aparecerá una brújula con todos los radiales escritos y el símbolo del un VOR en el centro, el cual solamente le indicará al piloto el radial donde se encuentra.

La pantalla principal recibirá el nombre de *MainScreenAxes*. Su representación en la interfaz es la de la Figura 59.

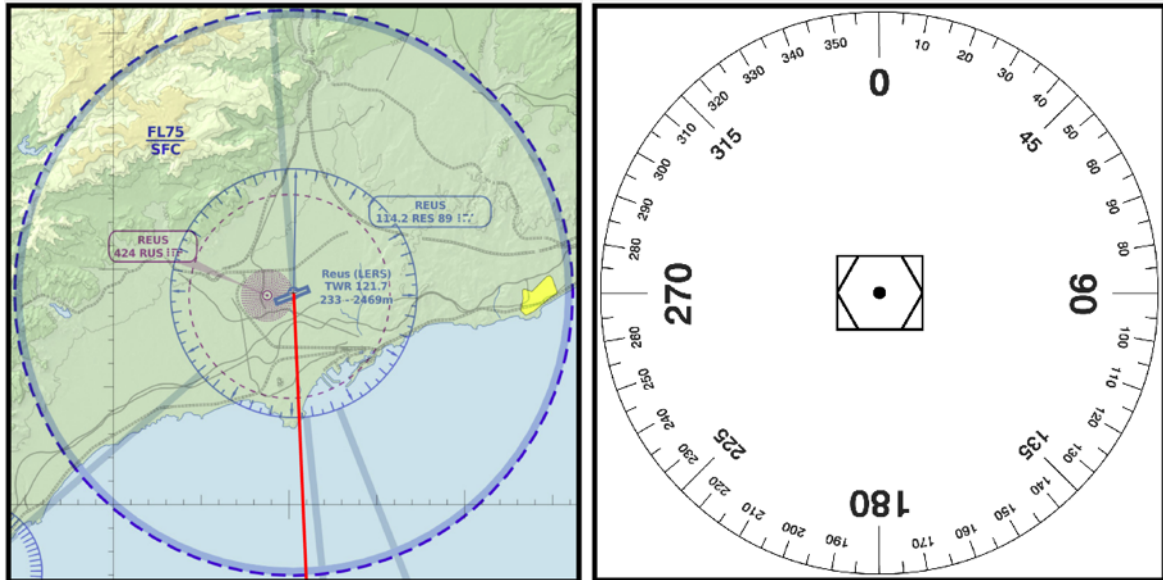


Figura 59. Pantalla principal del programa. A la izquierda se muestra la imagen de un VOR seleccionado previamente en la lista predeterminada; a la derecha se muestra el modelo general para cualquier VOR.

10.2.6 Instrumentos Tradicionales

Además de la pantalla principal, el panel de la interfaz trae instalados los instrumentos que indican al piloto del *heading*, un OBI y un indicador de velocidad que se debe ajustar la velocidad manualmente.

Uno de ellos es el **Heading indicator**. Este instrumento solamente estará activo cuando se ejecute el programa con el modo de la posición del avión. En caso contrario, no funcionará. Recibirá el nombre de *Heading Axes*.

Otro instrumento es el **OBI/OBS**. El OBI u OBS estará activo en todo momento. El piloto puede modificar el CDI mediante el selector de radiales. Esta caja de texto solamente admite número enteros del 0 a 360. Recibirá el nombre de *OBSAxes*, mientras que la caja para seleccionar el radial del CDI y el cuadro de texto de las banderas TO/FROM reciben el nombre de *RadialSpinner* y *TOFROMLabel* respectivamente.

Finalmente, el **indicador de velocidad**. La velocidad se debe introducir manualmente para que el algoritmo pueda ejecutarse correctamente. En un avión real la indicación de velocidad podría venir dada por otros sensores que en este trabajo no se han contemplado. Por lo tanto, el piloto introducirá manualmente la velocidad, tanto en nudos como en kilómetros por hora. Para la introducción del valor de la velocidad, se admitirán números enteros entre 0 y 450 nudos o aproximadamente 900 km/h. Recibirá el nombre de *GroundSpeedGauge*, mientras que la caja para seleccionar la velocidad y el interruptor para cambiar las unidades de medida de la velocidad reciben el nombre de *SpeedSpinner* y *SpeedParametersSwitch* respectivamente.

La representación en la interfaz de todos los elementos presentados en este apartado es la de la Figura 60.

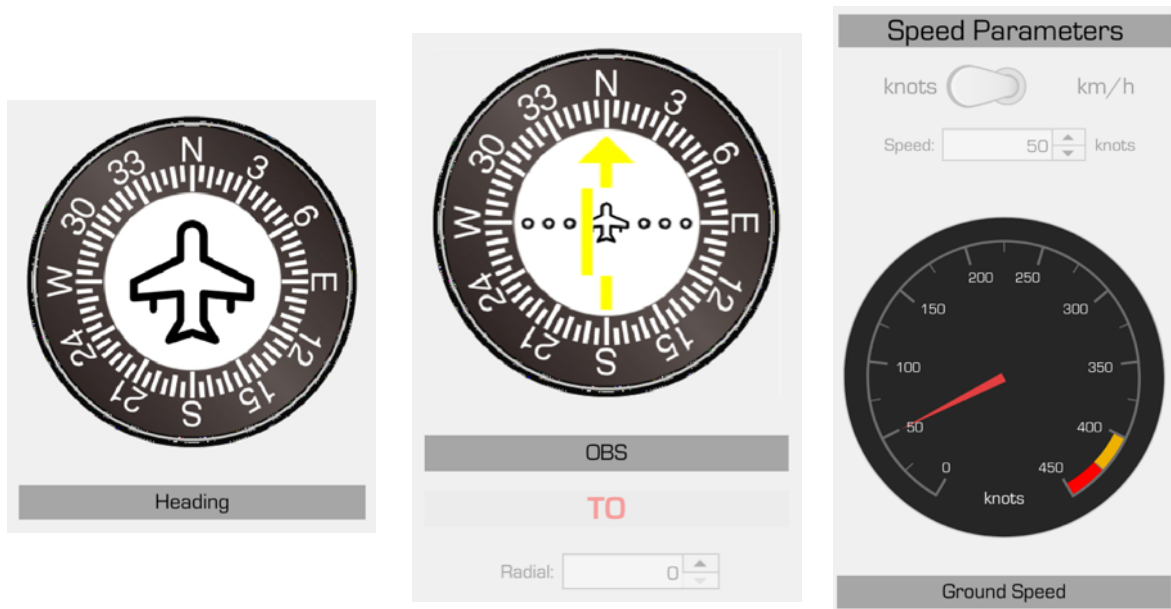


Figura 60. Instrumentos tradicionales añadidos a la interfaz. De izquierda a derecha: brújula indicadora del *heading* del avión, el OBI y el velocímetro manual.

10.3 Estructura Interna de la Aplicación

10.3.1 Funciones Globales

Las funciones son bloques de código que se ejecutan cada vez que el programador desee. Generalmente se usan para evitar tener que repetir bloques de código iguales o similares.

Las variables de toda la interfaz se han definido como variables globales. De esta manera, todas las funciones y *callbacks* de toda la interfaz pueden acceder a ellas y modificar su valor si fuera necesario.

En la interfaz se usan las siguientes funciones:

- **EstablecerEncendido.** Tal y como sugiere el nombre de la función, establece los parámetros de todos los elementos del interfaz a los correspondientes al estado de recién encendido el panel. Se utiliza al encender el panel y al presionar el botón STOP.
- **EstablecerApagado.** Esta función establece los parámetros de todos los elementos de la interfaz a los correspondientes al estado de apagado. En dicho caso, solamente estará habilitado en interruptor principal de On/Off, mientras que el resto de los elementos estarán deshabilitados.
- **TratamientoDigital.** En esta función se ejecuta todo el bloque de código correspondiente a la demodulación de la señal del VOR.
- **RepresentacionGraficaRadiales.** Esta función se llama para representar sobre el mapa los radiales que resultan de la demodulación de la señal del VOR.
- **ReprduccionCodigoMorse.** Esta función comprueba si el piloto ha habilitado o deshabilitado el audio, y en caso afirmativo reproducirá el código morse recibido del VOR.

- **ProcesarPosicion.** En esta función se realizan los cálculos de la posición del avión.
- **CalcularHeadingAvion.** Tal y como sugiere el nombre de la función, esta calcula los grados del *heading* del avión.
- **RepresentacionGraficaPosicionYHeading.** Esta función se llama para representar la trayectoria del avión sobre el mapa. Si los datos recibidos son incorrectos, se mostrará el siguiente mensaje: “NO DATA AVAILABLE”.
- **RepresentaciónGraficaOBS.** Esta función calcula la desviación del avión con respecto al radial indicado en el CDI. También representa la desviación en el OBI.

10.3.2 Callbacks

Los *callbacks* son bloques de código que se llaman en cualquier momento de la ejecución del programa tras un input del usuario, como podría ser apretar un botón o cambiar el valor de un cuadro de texto. En toda la interfaz se han programado los siguientes *callbacks*.

- **StartupFcn.** Este *callback* se llama automáticamente por el compilador del programa cada vez que se inicia una ejecución. En este *callback* se declaran los objetos de los filtros, el objeto del retardo para la demodulación de la señal FM y las imágenes de todo el programa.
- **SwitchValueChanged.** Este *callback* se llama cada vez que cambia el valor del interruptor de On/Off y comprueba el valor al cual se ha cambiado y llama a las funciones “EstablecerEncendido” o “EstablecerApagado” según corresponda.
- **LoadFrequencyButtonPushed.** Este *callback* se llama cuando el piloto presiona el botón de *Load Frequency*. En este *callback* se inicializa el objeto del SDR a la frecuencia central indicada en el *FrequencySpinner*.
- **RadialVisorKnobValueChanged.** Este *callback* se llama cuando el botón del visor de radiales cambia su estado. En este *callback* se ejecutará todo el código perteneciente a la demodulación de las señales del VOR y la representación de los radiales, el audio y el CDI del OBI.
- **STOPButtonPushed.** Este *callback* se llama cuando el piloto presiona el botón de *STOP* y ejecuta la función “EstablecerEncendido”.
- **FrequencySpinnerValueChanged.** Este *callback* se llama cuando cambia el valor de la frecuencia del *FrequencySpinner*. Se llamará para establecer los elementos de la interfaz de manera que el piloto deba volver a cargar la frecuencia que acaba de modificar.
- **SelectVORParametersListBoxValueChanged.** Este *callback* se llama cada vez que el piloto selecciona un VOR de la lista de VOR predeterminados, donde se comprobará el VOR seleccionado y se modificará el mapa del fondo de la pantalla principal.
- **SoundButtonPushed.** Este *callback* se llama cuando el piloto presiona el botón de activar o desactivar el audio.
- **SpeedParametersSwitchValueChanged.** Este *callback* se llama cada vez que se cambia el valor del interruptor del *SpeedParametersSwitch*. En este *callback* se realizan las conversiones de km/h a millas, o viceversa.
- **SpeedSpinnerValueChanged.** Este *callback* se llama cada vez que se modifica el valor del *SpeedSpinner* y modifica la indicación del *GroundSpeedGauge*.
- **AircraftPositionKnobValueChanged.** Este *callback* se llama cuando el botón de la posición del avión cambia su estado. En este *callback* se ejecutará todo el código

perteneciente al cálculo de la posición del avión y su *heading*, y modificará los valores de los instrumentos del OBI y el *heading*.

- **RadialSpinnerValueChanged.** Este *callback* se llama cada vez que se modifica el valor del *RadialSpinner*, donde se llamará a la función “RepresentaciónGraficaOBS” para cambiar la representación del CDI del OBI.

11 Presupuesto

Para la realización de este trabajo se ha requerido de cierto material, cuyo precio se especifica en la Tabla 4. Para una aplicación real, estos serían los costes fijos del diseño del proyecto.

Producto	Precio
Pack RTL-SDR y antenas monopolo de NooElec	54,95 €
Amplificador de NooElec	23,47 €
Licencia de MATLAB (proporcionada por la universidad)	119,00 €
Cable alargador de USB macho-hembra	5,47 €
Adaptadores SMA macho-macho	4,99 €
TOTAL €	207,88 €

Tabla 4. Presupuesto del proyecto.

Referencias

- [1] Air Pilot's Manual, "Radio Navigation & Instrument Flying", Air Pilot Publishing, 6th Edition 2009
- [2] Brian Kendal, "Manual of Avionics", Blackwell Science, 1999
- [3] Wikipedia, "Radiofaro Omnidireccional VHF", https://es.wikipedia.org/wiki/Radiofaro_omnidireccional_VHF [ONLINE], 28 de mayo de 2021
- [4] SigidWiki, "VHF Omnidirection Range (VOR)", [https://www.sigidwiki.com/wiki/VHF_Omnidirectional_Range_\(VOR\)](https://www.sigidwiki.com/wiki/VHF_Omnidirectional_Range_(VOR)) [ONLINE], 28 de mayo de 2021
- [5] Scott Thompson, "Flight Inspection History", https://www.faa.gov/air_traffic/flight_info/avn/media/documents/FlightInspectionHistory.pdf [ONLINE], 28 de mayo de 2021
- [6] Martin Latta, "VECTOR BASED HIGH AND LOW EN-ROUTE AERONAUTICAL CHARTS", <https://core.ac.uk/download/pdf/44404712.pdf> [PDF ONLINE], 28 de mayo de 2021
- [7] Aviator Blue Wings, "Navegación VOR. La guía definitiva.", <https://aviatorbluewings.com/navegacion-vor/> [ONLINE], 28 de mayo de 2021
- [8] Jesús Martínez Rueda, "Sistemas Eléctricos y Electrónicos de las Aeronaves", Thomson Pananinfo, 2007
- [9] Antonio Lázaro Guillen, "Introducció al SDR", Practica 7 de la asignatura de Laboratorio de Telecomunicaciones
- [10] Wikipedia, "Modulación en Amplitud en Cuadratura", https://es.wikipedia.org/wiki/Modulaci%C3%B3n_de_amplitud_en_cuadratura [ONLINE], 28 de mayo de 2021
- [11] Wikipedia, "Amplitud Modulada", https://es.wikipedia.org/wiki/Amplitud_modulada [ONLINE], 28 de mayo de 2021
- [12] Universidad Carlos III de Madrid, "Modulaciones Analógicas", http://ocw.uc3m.es/teoria-de-la-senal-y-comunicaciones/teoria-de-la-comunicacion/ficheroscurso/OCW_UC3M-TC-T3.pdf [PDF ONLINE], 28 de mayo de 2021
- [13] Ramon Villarino Villario, "Modulacions Lineals", Transparencias de la asignatura Fundamentos de Comunicaciones II, URV
- [14] Wikipedia, "Frecuencia Modulada", https://es.wikipedia.org/wiki/Frecuencia_modulada [ONLINE], 28 de mayo de 2021
- [15] Robert W. Stewart, Kenneth W. Barlee, Dale S. W. Atkinson, Louise H. Crockett, "Software Defined Radio using MATLAB & Simulink and the RTL-SDR", University of Strathclyde Glasgow, 1st Edition
- [16] Ramon Villarino Villario, "Modulacions Lineals", Transparencias de la asignatura Fundamentos de Comunicaciones II, URV
- [17] Unican, "Modulación Angular", https://personales.unican.es/perezvvr/pdf/CH6ST_Web.pdf [PDF ONLINE], 28 de mayo de 2021
- [18] "Funciones de Bessel", <https://slideplayer.es/slide/4132976/> [ONLINE], 28 de mayo de 2021
- [19] RTL-SDR.com, "Decoding Aviation VOR and ILS Signals with RTL-SDR", <https://www.rtl-sdr.com/decoding-aviation-vor-and-ils-signals-with-rtl-sdr/> [ONLINE], 28 de mayo de 2021
- [20] YouTube, "VOR", <https://www.youtube.com/watch?v=qEU2mGb0iak> [ONLINE], 28 de mayo de 2021
- [21] YouTube, "The Thechlical Wizardry of VORs – How they Work", <https://www.youtube.com/watch?v=tZ2gG1v9Xg8> [ONLINE], 28 de mayo de 2021
- [22] FAA, "Section 1. Navigation Aids", https://www.faa.gov/air_traffic/publications/atpubs/aim_html/chap1_section_1.html [ONLINE], 28 de mayo de 2021

- [23] Indra, “*Doppler VHR Omnidirectional Range*”, https://www.indracompany.com/sites/default/files/doppler_vhf_omnidirectional_range.pdf [PDF ONLINE], 28 de mayo de 2021
- [24] SELEX Sistemi Integrati Inc., “*VOR Specifications*”, <https://fccid.io/JPAVOR2/User-Manual/Users-Manual-1103052.pdf> [PDF ONLINE], 28 de mayo de 2021
- [25] Aviation, “*Why is there a grid below the antennas of a VOR?*”, <https://aviation.stackexchange.com/questions/43397/why-is-there-a-grid-below-the-antennas-of-an-vor> [ONLINE], 28 de mayo de 2021
- [26] José C. Meizoso Fernández, Carlos Meizoso Muñoz, “*Compendio de Aviónica Digital: Sistemas de Control y Guiado (Volumen 1)*”, BELLISCO Ediciones Técnicas y Científicas, 1ª Edición 2000
- [27] ICAO, “*Notices to Airmen (NOTAM)*”, <https://www.icao.int/safety/istars/pages/notams.aspx> [ONLINE], 28 de mayo de 2021
- [28] Luiz Monteiro, “*Online VOR Simulador*”, http://www.luizmonteiro.com/learning_vor_sim.aspx [ONLINE], 28 de mayo de 2021
- [29] “*Conversió a Digital*”, Transparencias de la asignatura Electrónica Digital, URV
- [30] Wikipedia, “*Radio Definida por Software*”, https://es.wikipedia.org/wiki/Radio_definida_por_software [ONLINE], 28 de mayo de 2021
- [31] RTL-SDR.com, “*Using an RTL-SDR to Decode VOR Aircraft Navigation Beacons in Real Time*”, <https://www.rtl-sdr.com/using-an-rtl-sdr-to-decode-vor-aircraft-navigation-beacons-in-real-time/> [ONLINE], 28 de mayo de 2021
- [32] F4GKR, “*In Depth Study of the VOR Signals (part 1)*”, <http://www.f4gkr.org/in-depth-study-of-the-vor-signals/> [ONLINE], 28 de mayo de 2021
- [33] NooElec, “*NooElec NESDR SMARTEE XTR SDR*”, <https://www.nooelec.com/store/sdr/nesdr-smartee-xtr-sdr.html> [ONLINE], 28 de mayo de 2021
- [34] NooElec, “*NooElec LaNA Barebones – Wideband Ultra Low-Noise Amplifier (LNA) Module*”, <https://www.nooelec.com/store/sdr/sdr-addons/lana-barebones.html> [ONLINE], 28 de mayo de 2021
- [35] NooElec, “*NooElec LaNA Barebones – Wideband Ultra Low-Noise Amplifier (LNA) Module Datasheet*”, https://www.nooelec.com/store/downloads/dl/file/id/93/product/321/lana_barebones_datasheet_revision_1.pdf [PDF ONLINE], 28 de mayo de 2021
- [36] Wikipedia, “*Bias Tee*”, https://en.wikipedia.org/wiki/Bias_tee [ONLINE], 28 de mayo de 2021
- [37] Revista Difital de ACTA, “*Antenas de Telecomunicaciones*”, https://www.acta.es/medios/articulos/ciencias_y_tecnologia/020001.pdf [PDF ONLINE], 28 de mayo de 2021
- [38] MathWorks, “*DSP System Toolbox*”, <https://es.mathworks.com/products/dsp-system.html> [ONLINE], 8 de junio de 2021
- [39] MathWorks, “*Communications Toolbox*”, <https://es.mathworks.com/products/communications.html> [ONLINE], 8 de junio de 2021
- [40] MathWorks, “*Signal Processing Toolbox*”, <https://es.mathworks.com/products/signal.html> [ONLINE], 8 de junio de 2021
- [41] MathWorks, “*RTL-SDR Hardware Support Package*”, <https://es.mathworks.com/hardware-support/rtl-sdr.html> [ONLINE], 8 de junio de 2021
- [42] MathWorks, “*fir1*”, <https://es.mathworks.com/help/signal/ref/fir1.html> [ONLINE], 28 de mayo de 2021
- [43] Wikipedia, “*FIR (Finite Impulse Response)*”, [https://es.wikipedia.org/wiki/FIR_\(Finite_Impulse_Response\)](https://es.wikipedia.org/wiki/FIR_(Finite_Impulse_Response)) [ONLINE], 28 de mayo de 2021
- [44] MathWorks, “*FM Demodulator Baseband*”, <https://es.mathworks.com/help/comm/ref/fmdemodulatorbaseband.html> [ONLINE], 28 de mayo de 2021

[45] MatWorks, Hristo Zhivomirov, “*Phase Difference Measurement with MATLAB*”, <https://es.mathworks.com/matlabcentral/fileexchange/48025-phase-difference-measurement-with-matlab> [ONLINE], 28 de mayo de 2021

[46] MathWorks, “*App Designer*”, <https://es.mathworks.com/help/matlab/app-designer.html#:~:text=App%20Designer%20es%20un%20entorno,componentes%20interactivos%20de%20la%20IU>. [ONLINE], 28 de mayo de 2021

Anexos

En este capítulo se adjunta todo el código utilizado para la creación de la interfaz gráfica del proyecto. Se ha separado por el código para la demodulación de las señales del VOR, el código para el cálculo de la posición del avión, el código para la grabación de señales para su posterior tratamiento (utilizado para hacer todas las pruebas posteriores a la captación de las señales) y finalmente todo el bloque de código de la interfaz gráfica llevada a cabo.

Anexo A. Código para la Demodulación de las Señales de los VOR

```

%% Demodulador de señales de VORs recibidas por un RTL-SDR
% Autor: Alex Lombao Chiuta
% Trabajo de fin de grado
% Grado en Ingeniería de Sistemas i Servicios de Telecomunicaciones
%
% Add-Ons necesarios para la correcta ejecución del programa:
% (probablemente no se necesario instalarlos a no ser que
%     el propio MATLAB lo pida (saltaría un error))
% - MathWorks DSP System Toolbox
% - MathWorks Communications System Toolbox
% - MathWorks Signal Processing Toolbox
% - MathWorks RTL-SDR Hardware Support Package

close all
clear

%% Frecuencia de funcionamiento del VOR del Aeropuerto de Reus
% (editable a cualquier otro VOR con frecuencia diferente)
f = 117.2e6; % Hz

%% Otros datos
offset = 9960; % distancia entre la portadora AM y la portadora FM [Hz]
fc = f - offset; % frecuencia central portadora señal FM [Hz]
ganancia = 50; % ganancia óptima [dB]
fs = 240e3; % frecuencia de muestreo de la señal digitalizada [Hz]
muestras_captura = 375e3; % numero de muestras por captura
tiempo_ejecucion = 40; % tiempo de la duración de la ejecución del
programa [s]
factor_diezmado = 2; % factor de compresión (diezmado) de los datos
tiempo_captura = muestras_captura/fs; % tiempo que dura una captura de
datos [s]

%% Receptor SDR (si se usa el tratamiento a tiempo real)
objeto_sdr = comm.SDRRTLReceiver('0', ...
    'CenterFrequency', fc,...
    'EnableTunerAGC', false,... % AUTOGAIN: por defecto es 'true' y no
recibe señales débiles
    'TunerGain', ganancia,...
    'SampleRate', fs, ...
    'OutputDataType', 'single', ...
    'SamplesPerFrame', muestras_captura);

%% Declaración de los objetos: elementos para el tratamiento de señal

orden_filtros = 1000;

% Filtro FIR paso banda (señal AM)
fn1 = (offset - 1030)/(fs/2);
fn2 = (offset + 1030)/(fs/2);

```

```

filtro_paso_banda_AM = firl(ordren_filtros,[fN1 fN2], 'bandpass');

% Filtro FIR paso banda (código morse)
gN1 = (1015)/(fs/2);
gN2 = (1025)/(fs/2);
filtro_paso_banda_morse = firl(ordren_filtros,[gN1 gN2], 'bandpass');

% Filtro FIR paso bajo (señal FM)
hN1 = (1e3)/(fs/2);
filtro_paso_bajo = firl(ordren_filtros, hN1, 'low');

% Retardo (se usa para la demodulación FM)
objeto_retardo = dsp.Delay;

% Objeto de audio
deviceWriter = audioDeviceWriter(fs/factor_diezmado);

%% Procesado
i = 1; % iteración columna de la matriz de datos (en caso de trabajar con
señal grabada)

for tiempo = 0:tiempo_captura:tiempo_ejecucion % control tiempo de
ejecución [s]

    % load('datos_vor_casa.mat');
    % datos_sdr = guardar(:,i); % señal grabado
    datos_sdr = step(objeto_sdr); % señal a tiempo real recibida por el
SDR (necesario SDR conectado)

    % Demodulación AM
    senyalAM = filter(filtro_paso_banda_AM, 1, datos_sdr);
    senyalAMdemodulado = abs(senyalAM);

    % Demodulación FM
    senyalFM = filter(filtro_paso_bajo, 1, datos_sdr);
    senyalFMretardado = step(dsp.Delay, senyalFM);
    senyalFMconjugado = conj(senyalFM);
    multiplicacion = senyalFMretardado .* senyalFMconjugado;
    senyalFMdemodulado = angle(multiplicacion);

    % Filtrado señal portadora del código morse
    senyalMorse = filter(filtro_paso_banda_morse, 1, senyalAMdemodulado);

    % Diezmado por "factor_diezmado"
    senyalAMdemodulada_diezmado = downsample(senyalAMdemodulado,
factor_diezmado);
    senyalFMdemodulada_diezmado = downsample(senyalFMdemodulado,
factor_diezmado);
    senyalMorse_diezmado = downsample(senyalMorse, factor_diezmado);

    % Cálculo de la diferencia de fase

    % Representar las señales como vectores-columna
    x = senyalAMdemodulada_diezmado(:);
    y = senyalFMdemodulada_diezmado(:);

    % Eliminar la componente DC de las señales
    x = x - mean(x);
    y = y - mean(y);

```

```

% Cálculo del tamaño de los vectores
xlen = length(x);
ylen = length(y);

% Generación de la ventana
xwin = hanning(xlen, 'periodic');
ywin = hanning(ylen, 'periodic');

% Aplicar la fft a las señales
X = fft(x.*xwin);
Y = fft(y.*ywin);

% Detección de la frecuencia fundamental
[~, indx] = max(abs(X));
[~, indy] = max(abs(Y));

% Estimación de la diferencia de fase
dif_fase = angle(Y(indy)) - angle(X(indx));
dif_fase_grados = dif_fase * 360 / (2*pi);
disp(dif_fase_grados);

% Reproducir la porción de señal
codigo_morse = senyalMorse_diezmado/max(abs(senyalMorse_diezmado));
deviceWriter(codigo_morse);

    i = i + 1; % en caso de señal grabado hay que iterar las columnas de
datos
end

release(deviceWriter); % cerrar el objeto de audio

```

Anexo B. Código para el Cálculo de la Posición del Avión

```

%% Algoritmo para el cálculo de la posición del avión respecto al VOR
% Autor: Alex Lombao Chiuta
% Trabajo de fin de grado
% Grado en Ingeniería de Sistemas i Servicios de Telecomunicaciones
%
% Suponiendo que el avión se mueve en línea recta y tiene una velocidad
respecto al suelo
% constante (ground speed constante), se determina la localización del
avión
% relativa a los últimos radiales interceptados de un solo VOR

close all
clear

%% Input primer radial
prompt = 'Entrar siguiente radial: ';
radial1 = input(prompt);
if radial1 == 0 || radial1 == 360
    radial1 = 0.01;
elseif radial1 == 180
    radial1 = 180.01;
end

%% Input segundo radial
prompt = 'Entrar siguiente radial: ';
radial2 = input(prompt);

```

```

if radial2 == 0 || radial2 == 360
    radial2 = 0.01;
elseif radial2 == 180
    radial2 = 180.01;
end

while true
    %% Input tercer radial
    prompt = 'Entrar siguiente radial: ';
    radial3 = input(prompt);
    if radial3 == 0 || radial3 == 360
        radial3 = 0.01;
    elseif radial3 == 180
        radial3 = 180.01;
    end

    %% Otros parámetros
    hold off
    figure(1);

    %% Cálculos previos
    groundSpeed = 120; % nudos
    intervalo = 5; % segundos

    velocidad = 1.852 * groundSpeed * 1000 / 3600; % m/s
    distancia = 2000; % velocidad * intervalo; % metros (410 es la
    distancia en m que recorre el avión a 80 nudos en 10 segundos)

    %% Cálculo de los ángulos entre la trayectoria del avión y los
    radiales
    %   alfamasbeta = 180 - ( abs(radial3-radial2) + abs(radial1-
    radial2) );
    alfamasbeta = 180 - ( radial3-radial2 + radial2-radial1 );

    %   anguloContiguoRadial1 = atand( 1 / ...
    %   ( ( (distancia/sind(abs(radial1-radial2))) /
    (distancia/sind(abs(radial3-radial2))) + cosd(alfamasbeta) ) /
    sind(alfamasbeta) ) );
    anguloContiguoRadial1 = atand( 1 / ...
    ( ( (distancia/sind(radial2-radial1)) / (distancia/sind(radial3-
    radial2)) + cosd(alfamasbeta) ) / sind(alfamasbeta) ) );

    anguloContiguoRadial3 = alfamasbeta - anguloContiguoRadial1;

    %% Cálculo de las distancias en el momento de la lectura de los
    radiales
    distanciaRadial2 = -distancia/sind(abs(radial1-radial2)) *
    sind(anguloContiguoRadial1);
    % distanciaRadial2 = distancia/sind(abs(radial3-radial2)) *
    sind(anguloContiguoRadial3);

    distanciaRadial1 = (2*distancia)/sind(abs(radial1-radial3)) *
    sind(anguloContiguoRadial3);

    distanciaRadial3 = -(2*distancia)/sind(abs(radial1-radial3)) *
    sind(anguloContiguoRadial1);

    %% Cálculo de los puntos de los radiales por donde ha pasado el avión
    mRadial1 = tand(-90 - radial1);
    mRadial2 = tand(-90 - radial2);
    mRadial3 = tand(-90 - radial3);

```

```

puntoXRadial1 = sqrt(distanciaRadial1^2 / (1 + mRadial1^2));
if (radial1 > 180)
    puntoXRadial1 = -puntoXRadial1;
end
puntoYRadial1 = mRadial1 * puntoXRadial1;

puntoXRadial2 = sqrt(distanciaRadial2^2 / (1 + mRadial2^2));
if (radial2 > 180)
    puntoXRadial2 = -puntoXRadial2;
end
puntoYRadial2 = mRadial2 * puntoXRadial2;

puntoXRadial3 = sqrt(distanciaRadial3^2 / (1 + mRadial3^2));
if (radial3 > 180)
    puntoXRadial3 = -puntoXRadial3;
end
puntoYRadial3 = mRadial3 * puntoXRadial3;

%% Comprobar que los datos cumplen las condiciones de la predicción
teórica
if (isnan(puntoXRadial1) || isnan(puntoXRadial2) ||
isnan(puntoXRadial3) || isnan(puntoYRadial1) || isnan(puntoYRadial2) ||
isnan(puntoYRadial3))
    disp('Los 3 últimos datos no cumplen las condiciones teóricas');
    txt = 'NO AVAILABLE DATA'; % mensaje en inglés porque lo tiene
que leer el piloto
    t = text(0,0,txt);
    t.FontSize = 20;
    t.HorizontalAlignment = 'center';
    t.FontWeight = 'bold';
    t.Color = [1, 0, 0];
    hold on
else
    %% Representación gráfica de los tres radiales
    plot([puntoXRadial1 0], [puntoYRadial1 0], '*')
    line([puntoXRadial1 0], [puntoYRadial1 0],
'Color','red','LineStyle','--')
    hold on

    plot([puntoXRadial2 0], [puntoYRadial2 0], '*')
    line([puntoXRadial2 0], [puntoYRadial2 0],
'Color','red','LineStyle','--')
    hold on

    plot([puntoXRadial3 0], [puntoYRadial3 0], '*')
    line([puntoXRadial3 0], [puntoYRadial3 0],
'Color','red','LineStyle','--')
    hold on

    %% Representación gráfica de la trayectoria del avión
    quiver(puntoXRadial1, puntoYRadial1, puntoXRadial2-puntoXRadial1,
puntoYRadial2-puntoYRadial1, 0, 'Color','magenta','LineWidth',1.75)
    hold on
    quiver(puntoXRadial2, puntoYRadial2, puntoXRadial3-puntoXRadial2,
puntoYRadial3-puntoYRadial2, 0, 'Color','magenta','LineWidth',1.75)
    hold on

    %% Determinar el heading del avión

    if (puntoXRadial3 - puntoXRadial2) > 0 % positivo

```

```

        if (puntoYRadial3 - puntoYRadial2) > 0 % positivo
            mHeading = (puntoYRadial3 - puntoYRadial2) /
(puntoXRadial3 - puntoXRadial2);
            rotate = atand(mHeading) - 90;
        elseif (puntoYRadial3 - puntoYRadial2) == 0 % cero
            rotate = 90;
        else % negativo
            mHeading = (puntoYRadial3 - puntoYRadial2) /
(puntoXRadial3 - puntoXRadial2);
            rotate = atand(mHeading) - 90;
        end
        elseif (puntoXRadial3 - puntoXRadial2) == 0 % cero
            if (puntoYRadial3 - puntoYRadial2) > 0 % positivo
                rotate = 0;
            else % negativo
                rotate = 180;
            end
        else % negativo
            if (puntoYRadial3 - puntoYRadial2) > 0 % positivo
                mHeading = (puntoYRadial3 - puntoYRadial2) /
(puntoXRadial3 - puntoXRadial2);
                rotate = atand(mHeading) + 90;
            elseif (puntoYRadial3 - puntoYRadial2) == 0 % cero
                rotate = -90;
            else % negativo
                mHeading = (puntoYRadial3 - puntoYRadial2) /
(puntoXRadial3 - puntoXRadial2);
                rotate = atand(mHeading) + 90;
            end
        end
        end

        d = imread('airplane.jpg');
        d = imrotate(d, rotate);
        image(flipud(d), 'XData', puntoXRadial3-200, 'YData',
puntoYRadial3-200); % valores escogidos por prueba y error

    end

    %% Determinar la imagen de fondo (mapa de los alrededores del
Aeropuerto de Reus)
    axis([-5750 5750 -5750 5750])

    I = imread('LERS.png');
    h = image(xlim,-ylim,I);
    uistack(h,'bottom')

    %% Preparación para la siguiente lectura del siguiente radial
    radial1 = radial2;
    radial2 = radial3;

end

```

Anexo C. Código para la Grabación de Señales captadas por un RTL-SDR

```

%% Grabador de señales de VORs recibidos por un RTL-SDR
% Autor: Alex Lombao Chiuta
% Trabajo de fin de grado
% Grado en Ingeniería de Sistemas i Servicios de Telecomunicaciones
%
% Add-Ons necesarios para la correcta ejecución del programa:

```

```

% (probablemente no es necesario instalarlos a no ser que
%   el propio MATLAB lo pida (saltaría un error))
% - MathWorks DSP System Toolbox
% - MathWorks Communications System Toolbox
% - MathWorks Signal Processing Toolbox
% - MathWorks RTL-SDR Hardware Support Package

close all
clear
clc

%% Frecuencia de funcionamiento del VOR del Aeropuerto de Reus
% (editable a cualquier otro VOR con frecuencia diferente)
f = 114.2e6; % Hz

%% Dades
offset = 9960; % distancia entre la portadora AM i la portadora FM [Hz]
fc = f - offset; % frecuencia central de la seÑal portadora FM [Hz]
ganancia = 40; % ganancia [dB]
fs = 240e3; % frecuencia de muestreo [Hz]
muestras_captura = 375e3; % numero de muestras por captura
tiempo_ejecucion = 120; % tiempo que queremos que dure la ejecuciÛn del
programa [s]
tiempo_captura = muestras_captura/fs; % tiempo que dura una captura de
datos [s]

%% Receptor SDR
objeto_sdr = comm.SDRRTLReceiver('0', ...
    'CenterFrequency', fc,...
    'EnableTunerAGC', false,...
    'TunerGain', ganancia,...
    'SampleRate', fs, ...
    'OutputDataType', 'single', ...
    'SamplesPerFrame', muestras_captura);

%% Procesado
espacio_memoria = 850 * tiempo_ejecucion / tiempo_captura / 1000;
fprintf('La grabaciÛn ocupar . %.2f MB en el disco.\n', espacio_memoria);
fprintf('Si no deseas continuar, cancela la ejecuciÛn del programa
haciendo ctrl + C, en caso contrario, espera 2 segundos.\n');
pause(2);

clc
fprintf('Inicio de la grabaciÛn...\n');
i = 1;
muestras = floor(tiempo_ejecucion/tiempo_captura) + 1;

for tiempo = 0:tiempo_captura:tiempo_ejecucion % tiempo real de
simulaciÛn [s]

    datos_sdr = step(objeto_sdr);

    guardar(:,i) = datos_sdr;

    fprintf(' - Muestra: %i de %i muestras.\n', i, muestras);
    i = i + 1;
end

clc
fprintf('La grabaciÛn ha sido guardada en la variable "guardar".\n');

```

Anexo D. Código de la Interfaz Gráfica

```

classdef prue < matlab.apps.AppBase

    % Properties that correspond to app components
    properties (Access = public)
        UIFigure                matlab.ui.Figure
        SelectVORParametersListBoxLabel  matlab.ui.control.Label
        SelectVORParametersListBox  matlab.ui.control.ListBox
        MainScreenAxes           matlab.ui.control.UIAxes
        FrequencySpinnerLabel     matlab.ui.control.Label
        FrequencySpinner          matlab.ui.control.Spinner
        MHzLabel                  matlab.ui.control.Label
        Switch                    matlab.ui.control.RockerSwitch
        RadialVisorKnobLabel      matlab.ui.control.Label
        RadialVisorKnob          matlab.ui.control.DiscreteKnob
        GroundSpeedGauge         matlab.ui.control.Gauge
        SystemLamp                matlab.ui.control.Lamp
        STOPButton                matlab.ui.control.Button
        LoadFrequencyButton       matlab.ui.control.Button
        SoundButton               matlab.ui.control.Button
        HeadingAxes               matlab.ui.control.UIAxes
        OBSAxes                   matlab.ui.control.UIAxes
        OBSLabel                  matlab.ui.control.Label
        HeadingLabel              matlab.ui.control.Label
        AircraftPositionKnobLabel matlab.ui.control.Label
        AircraftPositionKnob     matlab.ui.control.DiscreteKnob
        SpeedUnitsLabel           matlab.ui.control.Label
        SpeedLabel                matlab.ui.control.Label
        SpeedSpinner              matlab.ui.control.Spinner
        PositionLamp              matlab.ui.control.Lamp
        RadialLamp                matlab.ui.control.Lamp
        SpeedParametersSwitchLabel matlab.ui.control.Label
        SpeedParametersSwitch     matlab.ui.control.ToggleSwitch
        GroundSpeedLabel          matlab.ui.control.Label
        SpeedUnitsGaugeLabel      matlab.ui.control.Label
        RadialLabel               matlab.ui.control.Label
        RadialSpinner             matlab.ui.control.Spinner
        AuthorLabel               matlab.ui.control.Label
        TOFROMLabel               matlab.ui.control.Label
        VORSelectedLabel          matlab.ui.control.Label
        MODELabel                 matlab.ui.control.Label
        FrequencyLabel             matlab.ui.control.Label
        VORMorseCodeLabel         matlab.ui.control.Label
        LiveActualRadialLabel     matlab.ui.control.Label
        Timeintervalbetweenradiallectures156secondsLabel  matlab.ui.contr
ol.Label
        MODEInfoLabel             matlab.ui.control.Label
        VORSelectedInfoLabel      matlab.ui.control.Label
        FrequencyInfoLabel        matlab.ui.control.Label
        VORMorseCodeInfoLabel     matlab.ui.control.Label
        LiveRadialLabel           matlab.ui.control.Label
    end

    properties (Access = private)
        Property % Description

        %% Variables globales
        objeto_sdr;
        filtro_paso_banda_AM;
        filtro_paso_banda_morse;
    end
end

```

```

    filtro_paso_bajo;
    objeto_retardo;
    deviceWriter;
    i;
    offset = 9960; % distancia entre la portadora AM y la portadora FM
[Hz]
    ganancia = 50; % ganancia óptima [dB]
    fs = 240e3; % frecuencia de muestreo de la señal digitalizada [Hz]
    muestras_captura = 375e3; % numero de muestras por captura
    tiempo_ejecucion = 60; % tiempo de la duración de la ejecución del
programa [s]
    factor_diezmado = 2; % factor de compresión (diezmado) de los datos
    tiempo_captura;
    senyalMorse_diezmado;

    radial; % radial de la muestra actual
    velocidad; % velocidad en m/s
    intervalos = 14; % numero de muestras omitidas antes de leer el
siguiente radial
    numeroLecturaRadial = 0; % para indicar las primeras 3 lecturas de
radiales
    radial1;
    radial2;
    radial3;
    puntoXRadial1;
    puntoYRadial1;
    puntoXRadial2;
    puntoYRadial2;
    puntoXRadial3;
    puntoYRadial3;
    rotar; % angulo para rotar la imagen del avión sobre el mapa
    brujula; % angulo para rotar la brújula

    VORSelected = "manually";

    running = false; % variable en true mientras se recibe y se muestra
señal
    sound = true; % variable para escuchar o silenciar el código morse
del VOR
    cancel = false; % variable para cancelar la ejecución del programa
    inicio = true; % variable que indica que es la primera ejecución
del programa

    grabado;
    senal;
end

methods (Access = private)

function EstablecerEncendido(app)
    app.SystemLamp.Color = [0 1 0];
    app.SelectVORParametersListBox.Enable = true;
    app.FrequencySpinner.Enable = true;
    app.MHzLabel.Enable = true;
    app.SelectVORParametersListBox.Value = "> Enter Manually <";
    app.FrequencySpinner.BackgroundColor = [1 0 0];
    app.RadialVisorKnob.Enable = false;
    app.AircraftPositionKnob.Enable = false;
    app.LoadFrequencyButton.Enable = true;
    app.STOPButton.Enable = false;
    app.SoundButton.Enable = true;
    app.RadialVisorKnob.Value = "Off";
    app.AircraftPositionKnob.Value = "Off";

```

```

app.SoundButton.Icon = 'volumen-on.png';
app.SpeedUnitsLabel.Enable = false;
app.RadialLamp.Color = [1 0 0];
app.PositionLamp.Color = [1 0 0];
app.TOFROMLabel.Enable = true;
app.RadialSpinner.Enable = true;
app.SpeedParametersSwitch.Enable = true;
app.SpeedSpinner.Enable = true;
app.SpeedUnitsLabel.Enable = true;

I = imread('rose.png');
h = imresize(I, [4000 4000]);
image(app.MainScreenAxes,0, 0, h);
app.MainScreenAxes.YDir = 'normal';

app.running = false;
app.cancel = true;
app.sound = true;

app.grabado = true; % true si se usa señal grabado, false si
se usa SDR

app.MODEInfoLabel.Text = "-";
app.VORSelectedInfoLabel.Text = "-";
app.FrequencyInfoLabel.Text = "-";
app.VORMorseCodeInfoLabel.Text = "-";
app.LiveRadialLabel.Text = "-";

clear app.objeto_sdr;
end

function EstablecerApagado(app)

app.SystemLamp.Color = [1 0 0];
app.RadialVisorKnob.Enable = false;
app.LoadFrequencyButton.Enable = false;
app.SelectVORParametersListBox.Enable = false;
app.SelectVORParametersListBox.Value = "> Enter Manually <";
app.FrequencySpinner.Enable = false;
app.STOPButton.Enable = false;
app.MHzLabel.Enable = false;
app.RadialVisorKnob.Value = "Off";
app.AircraftPositionKnob.Enable = false;
app.FrequencySpinner.BackgroundColor = [1 0 0];
app.SoundButton.Enable = false;
app.SoundButton.Icon = 'volumen-on.png';
if app.running
    app.cancel = true;
end
app.TOFROMLabel.Enable = false;
app.RadialSpinner.Enable = false;
app.SpeedParametersSwitch.Enable = false;
app.SpeedSpinner.Enable = false;
app.SpeedUnitsLabel.Enable = false;

I = imread('roseOFF.png');
h = imresize(I, [4000 4000]);
image(app.MainScreenAxes,0, 0, h);
app.MainScreenAxes.YDir = 'normal';

app.running = false;
app.cancel = true;
app.inicio = true;

```

```

        app.sound = true;
    end

    function TratamientoDigital(app)

        if app.grabado
            % appsenal = load('datos_vor_casa.mat');
            appsenal = load('T11TGN.mat');
            app.grabado = false;
        end

        datos_sdr = appsenal.guardar(:,app.i); % señal grabada
        % datos_sdr = step(app.objeto_sdr); % señal a
        tiempo real recibida por el SDR (necesario SDR conectado)

        % Demodulación AM
        senyalAM = filter(app.filtro_paso_banda_AM, 1, datos_sdr);
        senyalAMdemodulada = abs(senyalAM);

        % Demodulación FM
        senyalFM = filter(app.filtro_paso_bajo, 1, datos_sdr);
        senyalFMretardada = step(app.objeto_retardo, senyalFM);
        senyalFMconjugada = conj(senyalFM);
        multiplicacion = senyalFMretardada .* senyalFMconjugada;
        senyalFMdemodulada = angle(multiplicacion);

        % Filtrado señal portadora del código morse
        senyalMorse = filter(app.filtro_paso_banda_morse, 1,
senyalAMdemodulada);

        % Diezmado pof "factor_diezmado"
        senyalAMdemodulada_diezmado = downsample(senyalAMdemodulada,
app.factor_diezmado);
        senyalFMdemodulada_diezmado = downsample(senyalFMdemodulada,
app.factor_diezmado);
        app.senyalMorse_diezmado = downsample(senyalMorse,
app.factor_diezmado);

        %% Cálculo de la diferencia de fase

        % Representar las señales como vectores-columna
        x = senyalAMdemodulada_diezmado(:);
        y = senyalFMdemodulada_diezmado(:);

        % Eliminar la componente DC de las señales
        x = x - mean(x);
        y = y - mean(y);

        % Cálculo del tamaño de los vectores
        xlen = length(x);
        ylen = length(y);

        % Generación de la ventana
        xwin = hanning(xlen, 'periodic');
        ywin = hanning(ylen, 'periodic');

        % Aplicar la fft a las señales
        X = fft(x.*xwin);
        Y = fft(y.*ywin);

        % Detección de la frecuencia fundamental
        [~, indx] = max(abs(X));
        [~, indy] = max(abs(Y));
    end

```

```

% Estimación de la diferencia de fase
dif_fase = angle(Y(indy)) - angle(X(indx));

% Acomodación del valor del radial

dif_fase_grados = dif_fase * 360 / (2*pi);
if dif_fase_grados < 0
    dif_fase_grados = dif_fase_grados + 360;
end

if dif_fase_grados == 0 || dif_fase_grados == 360
    dif_fase_grados = 0.01;
elseif dif_fase_grados == 180
    dif_fase_grados = 180.01;
end

app.radial = dif_fase_grados;
app.LiveRadialLabel.Text = num2str(dif_fase_grados);

end

function RepresentacionGraficaRadiales(app)

    %% Cálculo de la representación gráfica sobre coordenadas
cartesianas

mRadial = tand(-90 - app.radial);

hold(app.MainScreenAxes, 'off');

if app.SelectVORParametersListBox.Value == "BAGUR"
    I = imread('bagur.png');
    h = imresize(I, [4000 4000]);
    image(app.MainScreenAxes,0, 0, h);
    app.MainScreenAxes.YDir = 'normal';
elseif app.SelectVORParametersListBox.Value == "BARCELONA 25R"
    I = imread('barcelona25R.png');
    h = imresize(I, [4000 4000]);
    image(app.MainScreenAxes,0, 0, h);
    app.MainScreenAxes.YDir = 'normal';
elseif app.SelectVORParametersListBox.Value == "CALELLA"
    I = imread('calella.png');
    h = imresize(I, [4000 4000]);
    image(app.MainScreenAxes,0, 0, h);
    app.MainScreenAxes.YDir = 'normal';
elseif app.SelectVORParametersListBox.Value == "EL PRAT 02"
    I = imread('barcelona02.png');
    h = imresize(I, [4000 4000]);
    image(app.MainScreenAxes,0, 0, h);
    app.MainScreenAxes.YDir = 'normal';
elseif app.SelectVORParametersListBox.Value == "GIRONA"
    I = imread('girona.png');
    h = imresize(I, [4000 4000]);
    image(app.MainScreenAxes,0, 0, h);
    app.MainScreenAxes.YDir = 'normal';
elseif app.SelectVORParametersListBox.Value == "MAELLA"
    I = imread('maella.png');
    h = imresize(I, [4000 4000]);
    image(app.MainScreenAxes,0, 0, h);
    app.MainScreenAxes.YDir = 'normal';
elseif app.SelectVORParametersListBox.Value == "REUS"
    I = imread('reus.png');

```

```

        h = imresize(I, [4000 4000]);
        image(app.MainScreenAxes,0, 0, h);
        app.MainScreenAxes.YDir = 'normal';
    elseif app.SelectVORParametersListBox.Value == "SABADELL"
        I = imread('sabadell.png');
        h = imresize(I, [4000 4000]);
        image(app.MainScreenAxes,0, 0, h);
        app.MainScreenAxes.YDir = 'normal';
    elseif app.SelectVORParametersListBox.Value == "VILAFRANCA"
        I = imread('vilafranca.png');
        h = imresize(I, [4000 4000]);
        image(app.MainScreenAxes,0, 0, h);
        app.MainScreenAxes.YDir = 'normal';
    elseif app.SelectVORParametersListBox.Value == "> Enter"
Manually <"
        I = imread('rose.png');
        h = imresize(I, [4000 4000]);
        image(app.MainScreenAxes,0, 0, h);
        app.MainScreenAxes.YDir = 'normal';
    end

    hold(app.MainScreenAxes, 'on');

    if app.radial < 180
        line(app.MainScreenAxes,[4000+2000 2000],
[mRadial*4000+2000 2000], 'Color','red','LineStyle','-','LineWidth',3)
    else
        line(app.MainScreenAxes,[-4000+2000 2000], [-
mRadial*4000+2000 2000], 'Color','red','LineStyle','-','LineWidth',3)
    end

end

function ReproduccionCodigoMorse(app)
    % Reproducir el código morse de la porción de señal recibida
    if app.sound
        codigo_morse
        = app.senyalMorse_diezmado/max(abs(app.senyalMorse_diezmado));
        app.deviceWriter(codigo_morse);
    end

end

function ProcesarPosicion(app)

    if app.numeroLecturaRadial == 1
        app.radial1 = app.radial;
        if app.radial1 == 0 || app.radial1 == 360
            app.radial1 = 0.01;
        elseif app.radial1 == 180
            app.radial1 = 180.01;
        end
        app.numeroLecturaRadial = 2;
    elseif app.numeroLecturaRadial == 2
        app.radial2 = app.radial;
        if app.radial2 == 0 || app.radial2 == 360
            app.radial2 = 0.01;
        elseif app.radial2 == 180
            app.radial2 = 180.01;
        end
        app.numeroLecturaRadial = 3;
    else
        app.radial3 = app.radial;
    end
end

```

```

if app.radial3 == 0 || app.radial3 == 360
    app.radial3 = 0.01;
elseif app.radial3 == 180
    app.radial3 = 180.01;
end
app.numeroLecturaRadial = 0;

    distancia = app.velocidad * (app.intervalos *
app.tiempo_captura) * 4000/45000;

    %% Cálculo de los ángulos entre la trayectoria del avión
y los radiales
    %    alfamasbeta = 180 - ( abs(radial3-radial2) +
abs(radial1-radial2) );
    alfamasbeta = 180 - ( app.radial3-app.radial2 + app.radial2-
app.radial1 );

    %    anguloContiguoRadial1 = atand( 1 / ...
    %    ( ( distancia/sind(abs(radial1-radial2))) /
(distancia/sind(abs(radial3-radial2))) + cosd(alfamasbeta) ) /
sind(alfamasbeta) );
    anguloContiguoRadial1 = atand( 1 / ...
    ( ( distancia/sind(app.radial2-app.radial1)) /
(distancia/sind(app.radial3-app.radial2)) + cosd(alfamasbeta) ) /
sind(alfamasbeta) );

    anguloContiguoRadial3 = alfamasbeta - anguloContiguoRadial1;

    %% Cálculo de las distancias en el momento de la lectura
de los radiales
    distanciaRadial2 = -distancia/sind(abs(app.radial1-
app.radial2)) * sind(anguloContiguoRadial1);
    % distanciaRadial2 = distancia/sind(abs(radial3-radial2))
* sind(anguloContiguoRadial3);

    distanciaRadial1 = (2*distancia)/sind(abs(app.radial1-
app.radial3)) * sind(anguloContiguoRadial3);

    distanciaRadial3 = -(2*distancia)/sind(abs(app.radial1-
app.radial3)) * sind(anguloContiguoRadial1);

    %% Cálculo de los puntos de los radiales por donde ha
pasado el avión
mRadial1 = tand(-90 - app.radial1);
mRadial2 = tand(-90 - app.radial2);
mRadial3 = tand(-90 - app.radial3);

    app.puntoXRadial1 = sqrt(distanciaRadial1^2 / (1 +
mRadial1^2));
if (app.radial1 > 180)
    app.puntoXRadial1 = -app.puntoXRadial1;
end
app.puntoYRadial1 = mRadial1 * app.puntoXRadial1;

    app.puntoXRadial2 = sqrt(distanciaRadial2^2 / (1 +
mRadial2^2));
if (app.radial2 > 180)
    app.puntoXRadial2 = -app.puntoXRadial2;
end
app.puntoYRadial2 = mRadial2 * app.puntoXRadial2;

    app.puntoXRadial3 = sqrt(distanciaRadial3^2 / (1 +
mRadial3^2));

```

```

        if (app.radial3 > 180)
            app.puntoXRadial3 = -app.puntoXRadial3;
        end
        app.puntoYRadial3 = mRadial3 * app.puntoXRadial3;
    end

end

function CalcularHeadingAvion(app)
    %% Determinar el heading del avión

    if (app.puntoXRadial3 - app.puntoXRadial2) > 0 % positivo
        if (app.puntoYRadial3 - app.puntoYRadial2) > 0 % positivo
            mHeading = (app.puntoYRadial3 - app.puntoYRadial2) /
(app.puntoXRadial3 - app.puntoXRadial2);
            app.rotar = atand(mHeading) - 90;
            app.brujula = app.rotar;
        elseif (app.puntoYRadial3 - app.puntoYRadial2) == 0 % cero
            app.rotar = 90;
            app.brujula = -app.rotar;
        else % negativo
            mHeading = (app.puntoYRadial3 - app.puntoYRadial2) /
(app.puntoXRadial3 - app.puntoXRadial2);
            app.rotar = atand(mHeading) - 90;
            app.brujula = app.rotar;
        end
    elseif (app.puntoXRadial3 - app.puntoXRadial2) == 0 % cero
        if (app.puntoYRadial3 - app.puntoYRadial2) > 0 % positivo
            app.rotar = 0;
            app.brujula = -app.rotar;
        else % negativo
            app.rotar = 180;
            app.brujula = -app.rotar;
        end
    else % negativo
        if (app.puntoYRadial3 - app.puntoYRadial2) > 0 % positivo
            mHeading = (app.puntoYRadial3 - app.puntoYRadial2) /
(app.puntoXRadial3 - app.puntoXRadial2);
            app.rotar = atand(mHeading) + 90;
            app.brujula = app.rotar;
        elseif (app.puntoYRadial3 - app.puntoYRadial2) == 0 % cero
            app.rotar = -90;
            app.brujula = -app.rotar;
        else % negativo
            mHeading = (app.puntoYRadial3 - app.puntoYRadial2) /
(app.puntoXRadial3 - app.puntoXRadial2);
            app.rotar = atand(mHeading) + 90;
            app.brujula = app.rotar;
        end
    end
end

end

function RepresentacionGraficaPosicionYHeading(app)

    hold(app.MainScreenAxes, 'off');

    if app.SelectVORParametersListBox.Value == "BAGUR"
        I = imread('bagur.png');
        h = imresize(I, [4000 4000]);
        image(app.MainScreenAxes,0, 0, h);
        app.MainScreenAxes.YDir = 'normal';
    end
end

```

```

elseif app.SelectVORParametersListBox.Value == "BARCELONA 25R"
    I = imread('barcelona25R.png');
    h = imresize(I, [4000 4000]);
    image(app.MainScreenAxes,0, 0, h);
    app.MainScreenAxes.YDir = 'normal';
elseif app.SelectVORParametersListBox.Value == "CALELLA"
    I = imread('calella.png');
    h = imresize(I, [4000 4000]);
    image(app.MainScreenAxes,0, 0, h);
    app.MainScreenAxes.YDir = 'normal';
elseif app.SelectVORParametersListBox.Value == "EL PRAT 02"
    I = imread('barcelona02.png');
    h = imresize(I, [4000 4000]);
    image(app.MainScreenAxes,0, 0, h);
    app.MainScreenAxes.YDir = 'normal';
elseif app.SelectVORParametersListBox.Value == "GIRONA"
    I = imread('girona.png');
    h = imresize(I, [4000 4000]);
    image(app.MainScreenAxes,0, 0, h);
    app.MainScreenAxes.YDir = 'normal';
elseif app.SelectVORParametersListBox.Value == "MAELLA"
    I = imread('maella.png');
    h = imresize(I, [4000 4000]);
    image(app.MainScreenAxes,0, 0, h);
    app.MainScreenAxes.YDir = 'normal';
elseif app.SelectVORParametersListBox.Value == "REUS"
    I = imread('reus.png');
    h = imresize(I, [4000 4000]);
    image(app.MainScreenAxes,0, 0, h);
    app.MainScreenAxes.YDir = 'normal';
elseif app.SelectVORParametersListBox.Value == "SABADELL"
    I = imread('sabadell.png');
    h = imresize(I, [4000 4000]);
    image(app.MainScreenAxes,0, 0, h);
    app.MainScreenAxes.YDir = 'normal';
elseif app.SelectVORParametersListBox.Value == "VILAFRANCA"
    I = imread('vilafranca.png');
    h = imresize(I, [4000 4000]);
    image(app.MainScreenAxes,0, 0, h);
    app.MainScreenAxes.YDir = 'normal';
elseif app.SelectVORParametersListBox.Value == "> Enter
    Manually <"
    I = imread('rose.png');
    h = imresize(I, [4000 4000]);
    image(app.MainScreenAxes,0, 0, h);
    app.MainScreenAxes.YDir = 'normal';
end

hold(app.MainScreenAxes, 'on');

app.puntoXRadial1 = app.puntoXRadial1 + 2000;
app.puntoXRadial2 = app.puntoXRadial2 + 2000;
app.puntoXRadial3 = app.puntoXRadial3 + 2000;
app.puntoYRadial1 = app.puntoYRadial1 + 2000;
app.puntoYRadial2 = app.puntoYRadial2 + 2000;
app.puntoYRadial3 = app.puntoYRadial3 + 2000;

%% Representació gràfica dels tres radials
plot(app.MainScreenAxes, [app.puntoXRadial1 2000],
[app.puntoYRadial1 2000], '*')
line(app.MainScreenAxes, [app.puntoXRadial1 2000],
[app.puntoYRadial1 2000], 'Color','red','LineStyle','--')
hold(app.MainScreenAxes, 'on');

```

```

        plot(app.MainScreenAxes, [app.puntoXRadial2 2000],
[app.puntoYRadial2 2000], '*')
        line(app.MainScreenAxes, [app.puntoXRadial2 2000],
[app.puntoYRadial2 2000], 'Color','red','LineStyle','--')
        hold(app.MainScreenAxes, 'on');

        plot(app.MainScreenAxes, [app.puntoXRadial3 2000],
[app.puntoYRadial3 2000], '*')
        line(app.MainScreenAxes, [app.puntoXRadial3 2000],
[app.puntoYRadial3 2000], 'Color','red','LineStyle','--')
        hold(app.MainScreenAxes, 'on');

        %% Representación gráfica de la trayectoria del avión
        quiver(app.MainScreenAxes, app.puntoXRadial1,
app.puntoYRadial1, app.puntoXRadial2-app.puntoXRadial1,
app.puntoYRadial2-app.puntoYRadial1, 0,
'Color','magenta','LineWidth',1.75)
        hold(app.MainScreenAxes, 'on');
        quiver(app.MainScreenAxes, app.puntoXRadial2,
app.puntoYRadial2, app.puntoXRadial3-app.puntoXRadial2,
app.puntoYRadial3-app.puntoYRadial2, 0,
'Color','magenta','LineWidth',1.75)
        hold(app.MainScreenAxes, 'on');

        d = imread('airplane.jpg');
        d = imrotate(d, app.rotar);
        d = imresize(d, [150 150]);
        image(app.MainScreenAxes, flipud(d), 'XData',
app.puntoXRadial3-75, 'YData', app.puntoYRadial3-75); % valors escollits
a base de prueba i error

        I = imread('compass.jpg');
        I = imresize(I, [400 400]);
        I = imrotate(I, app.brujula, 'bilinear','crop');
        I(I == 0) = 240;
        image(app.HeadingAxes,0, 0, I);
        app.HeadingAxes.YDir = 'normal';

        hold(app.HeadingAxes, 'on');

        I = imread('plane.jpg');
        h = imresize(I, [150 150]);
        image(app.HeadingAxes,125, 125, h);
        app.HeadingAxes.YDir = 'normal';

end

function RepresentacionGraficaOBS(app)

    hold(app.OBSAxes, 'off');

    I = imread('compass.jpg');
    I = imresize(I, [400 400]);
    I = imrotate(I, -app.RadialSpinner.Value, 'bilinear','crop');
    I(I == 0) = 240;
    image(app.OBSAxes, 0, 0, I);
    app.OBSAxes.YDir = 'normal';

    hold(app.OBSAxes, 'on');

    I = imread('hsiplane.jpg');
    h = imresize(I, [50 200]);

```

```

image(app.OBSAxes,100, 175, h);
app.OBSAxes.YDir = 'normal';

hold(app.OBSAxes, 'on');

I = imread('flechaHSI.jpg');
h = imresize(I, [60 70]);
image(app.OBSAxes, 165, 240, h);
app.OBSAxes.YDir = 'normal';

hold(app.OBSAxes, 'on');

I = imread('baseHSI.jpg');
h = imresize(I, [40 15]);
image(app.OBSAxes, 192, 100, h);
app.OBSAxes.YDir = 'normal';

hold(app.OBSAxes, 'on');

if app.running

    diferencia = app.radial - app.RadialSpinner.Value;

    if abs(diferencia) > 180
        if diferencia > 0
            diferencia = abs(diferencia - 360);
        else
            diferencia = abs(diferencia + 360);
        end
    end

    if abs(diferencia) > 90 % TO

        app.TOFROMLabel.Text = "TO";

        radialTO = app.RadialSpinner.Value - 180;
        if radialTO < 0
            radialTO = radialTO + 360;
        end

        desviacion = app.radial - radialTO;

        if desviacion > 0
            if desviacion > 10
                desviacion = 10;
            end
            I = imread('paloHSI.jpg');
            h = imresize(I, [100 15]);
            image(app.OBSAxes, 192-(desviacion*9), 140, h); %
centrado, 192 - 140
            app.OBSAxes.YDir = 'normal';
        else
            if desviacion < -10
                desviacion = -10;
            end
            I = imread('paloHSI.jpg');
            h = imresize(I, [100 15]);
            image(app.OBSAxes, 192+(desviacion*9), 140, h); %
centrado, 192 - 140
            app.OBSAxes.YDir = 'normal';
        end

    else % FROM

```

```

        app.TOFROMLabel.Text = "FROM";

        radialFROM = app.RadialSpinner.Value;

        desviacion = app.radial - radialFROM;

        if desviacion > 0
            if desviacion > 10
                desviacion = 10;
            end
            I = imread('paloHSI.jpg');
            h = imresize(I, [100 15]);
            image(app.OBSAxes, 192-(desviacion*9), 140, h); %
centrado, 192 - 140
            app.OBSAxes.YDir = 'normal';
        else
            if desviacion < -10
                desviacion = -10;
            end
            I = imread('paloHSI.jpg');
            h = imresize(I, [100 15]);
            image(app.OBSAxes, 192+(desviacion*9), 140, h); %
centrado, 192 - 140
            app.OBSAxes.YDir = 'normal';
        end

        end
    else
        I = imread('paloHSI.jpg');
        h = imresize(I, [100 15]);
        image(app.OBSAxes, 170, 140, h); % centrado, 192 - 140
        app.OBSAxes.YDir = 'normal';
    end

end

end

methods (Access = private)

    % Code that executes after component creation
    function startupFcn(app)

        clc

        %% Declaración de los objetos: elementos para el tratamiento
de señal

        orden_filtros = 1000;

        % Filtro FIR paso banda (señal AM)
        fN1 = (app.offset - 1030)/(app.fs/2);
        fN2 = (app.offset + 1030)/(app.fs/2);
        app.filtro_paso_banda_AM = fir1(orden_filtros,[fN1 fN2],
'bandpass');

        % Filtro FIR paso banda (codigo morse)
        gN1 = (1015)/(app.fs/2);
        gN2 = (1025)/(app.fs/2);
        app.filtro_paso_banda_morse = fir1(orden_filtros,[gN1 gN2],
'bandpass');

```

```

% Filtro FIR paso bajo (señal FM)
hN1 = (1e3)/(app.fs/2);
app.filtro_paso_bajo = fir1(orden_filtros, hN1, 'low');

% Retardo (se usa para la demodulación FM)
app.objeto_retardo = dsp.Delay;

% Objeto de audio
audioDeviceWriter(app.fs/app.factor_diezmado);
app.deviceWriter =

%% Procesado
app.i = 1; % iteración columna de la matriz de datos (en caso
de trabajar con señal grabada)

app.SystemLamp.Color = [1 0 0];

% imagen de fondo pantalla principal
I = imread('roseOFF.png');
h = imresize(I, [4000 4000]);
image(app.MainScreenAxes,0, 0, h);
app.MainScreenAxes.YDir = 'normal';

% imagenes instrumento de heading
I = imread('compass.jpg');
I = imresize(I, [400 400]);
I(I == 0) = 240;
image(app.HeadingAxes,0, 0, I);
app.HeadingAxes.YDir = 'normal';

hold(app.HeadingAxes, 'on');

I = imread('plane.jpg');
I = imresize(I, [150 150]);
image(app.HeadingAxes,125, 125, I);
app.HeadingAxes.YDir = 'normal';

% imágenes instrumento OBS
hold(app.OBSAxes, 'on');
I = imread('compass.jpg');
I = imresize(I, [400 400]);
I(I == 0) = 240;
image(app.OBSAxes,0, 0, I);
app.OBSAxes.YDir = 'normal';

hold(app.OBSAxes, 'on');

I = imread('hsiplane.jpg');
h = imresize(I, [50 200]);
image(app.OBSAxes,100, 175, h);
app.OBSAxes.YDir = 'normal';

hold(app.OBSAxes, 'on');

I = imread('flechaHSI.jpg');
h = imresize(I, [60 70]);
image(app.OBSAxes,165, 240, h);
app.OBSAxes.YDir = 'normal';

hold(app.OBSAxes, 'on');

I = imread('baseHSI.jpg');

```

```

h = imresize(I, [40 15]);
image(app.OBSAxes,192, 100, h);
app.OBSAxes.YDir = 'normal';

hold(app.OBSAxes, 'on');

I = imread('paloHSI.jpg');
h = imresize(I, [100 15]);
image(app.OBSAxes,170, 140, h); % centrado, 192 - 140
app.OBSAxes.YDir = 'normal';

app.MainScreenAxes.BackgroundColor = 'black';
app.HeadingAxes.BackgroundColor = [0.94 0.94 0.94];

end

% Value changed function: Switch
function SwitchValueChanged(app, event)
    value = app.Switch.Value;
    if value == "On"
        EstablecerEncendido(app);
    else
        EstablecerEncendido(app);
        EstablecerApagado(app);
    end
end

% Button pushed function: LoadFrequencyButton
function LoadFrequencyButtonPushed(app, event)

    % Frecuencia del VOR (válida para cualquier VOR con frecuencias
diferentes)
    f = single(app.FrequencySpinner.Value) * 1e6; % Hz
    %         round(f, 1);
    fc = f - app.offset; % frecuencia central portadora señal FM
[Hz]

    fc = double(fc);
    %% Receptor SDR (si se utiliza tratamiento de señal a tiempo
real)
    if app.inicio
        app.objeto_sdr = comm.SDRRTLReceiver('0', ...
            'CenterFrequency', fc,...
            'EnableTunerAGC', false,... % AUTOGAIN: por defecto es
'true' y no recibe señales débiles
            'TunerGain', app.ganancia,...
            'SampleRate', app.fs, ...
            'OutputDataType', 'single', ...
            'SamplesPerFrame', app.muestras_captura);
        app.inicio = false;
    else
        app.objeto_sdr.CenterFrequency = fc;
    end

    app.FrequencySpinner.BackgroundColor = [0 1 0];
    app.LoadFrequencyButton.Enable = false;

    if app.Switch.Value == "On"
        app.RadialVisorKnob.Enable = true;
        app.AircraftPositionKnob.Enable = true;
    end
end

```

```

                                app.FrequencyInfoLabel.Text =
strcat(num2str(app.FrequencySpinner.Value), " MHz");

end

% Value changed function: RadialVisorKnob
function RadialVisorKnobValueChanged(app, event)
    value = app.RadialVisorKnob.Value;

    app.MODEInfoLabel.Text = "Radial Visor";

    app.SoundButton.Enable = true;

    if value == "On"

        app.running = true;
        app.cancel = false;

        app.STOPButton.Enable = true;
        app.FrequencySpinner.Enable = false;
        app.MHzLabel.Enable = false;
        app.SelectVORParametersListBox.Enable = false;
        app.AircraftPositionKnob.Enable = false;
        app.RadialLamp.Color = [0 1 0];

        %% Procesado de señal
        app.i = 1; % interacción columna de la matriz amb datos (en
caso de trabajar con señal grabado)
        app.tiempo_captura = app.muestras_captura/app.fs; % tiempo
que dura una captura de datos [s]

        for tiempo = 0:app.tiempo_captura:app.tiempo_ejecucion %
control tiempo d'execución [s]

            if app.cancel % si cancela la ejecución del programa
-> salir del bucle
                break
            end

            TratamientoDigital(app); % calcula el radial

            RepresentacionGraficaRadiales(app); % representa el
radial

            ReproduccionCodigoMorse(app); % reproduce codigo morse

            RepresentacionGraficaOBS(app); % cambia el estado del
OBS

            app.i = app.i + 1; % en caso de trabajar con señal
grabado, iteración de la columna de datos

            % crear interrupción y sincronizar los tiempos de
captación de señal y reproducción del código morse
            pause(app.tiempo_captura-0.01);

        end

        release(app.deviceWriter); % cerrar el objeto de audio

    else
        EstablecerEncendido(app);
    end
end

```

```

end

end

% Button pushed function: STOPButton
function STOPButtonPushed(app, event)
    EstablecerEncendido(app);
end

% Value changed function: FrequencySpinner
function FrequencySpinnerValueChanged(app, event)
    app.FrequencySpinner.BackgroundColor = [1 0 0];
    app.LoadFrequencyButton.Enable = true;
    app.RadialVisorKnob.Enable = false;
end

% Value changed function: SelectVORParametersListBox
function SelectVORParametersListBoxValueChanged(app, event)
    value = app.SelectVORParametersListBox.Value;

    app.FrequencySpinner.BackgroundColor = [1 0 0];
    app.LoadFrequencyButton.Enable = true;
    app.RadialVisorKnob.Enable = false;

    if value == "BAGUR"
        app.FrequencySpinner.Value = 112.2; % MHz

        I = imread('bagur.png');
        h = imresize(I, [4000 4000]);
        image(app.MainScreenAxes,0, 0, h);
        app.MainScreenAxes.YDir = 'normal';

        app.VORSelectedInfoLabel.Text = "Bagur VOR-DME";
        app.FrequencyInfoLabel.Text = "112.2 MHz";
        app.VORMorseCodeInfoLabel.Text = "- . . . - - . . - .";

    elseif value == "BARCELONA 25R"
        app.FrequencySpinner.Value = 116.7; % MHz

        I = imread('barcelona25R.png');
        h = imresize(I, [4000 4000]);
        image(app.MainScreenAxes,0, 0, h);
        app.MainScreenAxes.YDir = 'normal';

        app.VORSelectedInfoLabel.Text = "Barcelona VOR-DME";
        app.FrequencyInfoLabel.Text = "116.7 MHz";
        app.VORMorseCodeInfoLabel.Text = "- . . . - . - . - .";

    elseif value == "CALELLA"
        app.FrequencySpinner.Value = 115.35; % MHz

        I = imread('calella.png');
        h = imresize(I, [4000 4000]);
        image(app.MainScreenAxes,0, 0, h);
        app.MainScreenAxes.YDir = 'normal';

        app.VORSelectedInfoLabel.Text = "Calella VOR-DME";
        app.FrequencyInfoLabel.Text = "115.35 MHz";
        app.VORMorseCodeInfoLabel.Text = "- . - . . - . . . .";

    elseif value == "EL PRAT 02"
        app.FrequencySpinner.Value = 114.3; % MHz

```

```

I = imread('barcelona02.png');
h = imresize(I, [4000 4000]);
image(app.MainScreenAxes,0, 0, h);
app.MainScreenAxes.YDir = 'normal';

app.VORSelectedInfoLabel.Text = "El Prat VOR-DME";
app.FrequencyInfoLabel.Text = "114.3 MHz";
app.VORMorseCodeInfoLabel.Text = ". - - . . - . . -";

elseif value == "GIRONA"
    app.FrequencySpinner.Value = 114.1; % MHz

    I = imread('girona.png');
    h = imresize(I, [4000 4000]);
    image(app.MainScreenAxes,0, 0, h);
    app.MainScreenAxes.YDir = 'normal';

    app.VORSelectedInfoLabel.Text = "Gerona VOR-DME";
    app.FrequencyInfoLabel.Text = "114.1 MHz";
    app.VORMorseCodeInfoLabel.Text = "- - . . . . - .";

elseif value == "MAELLA"
    app.FrequencySpinner.Value = 112.1; % MHz

    I = imread('maella.png');
    h = imresize(I, [4000 4000]);
    image(app.MainScreenAxes,0, 0, h);
    app.MainScreenAxes.YDir = 'normal';

    app.VORSelectedInfoLabel.Text = "Maella VOR-DME";
    app.FrequencyInfoLabel.Text = "112.1 MHz";
    app.VORMorseCodeInfoLabel.Text = "- - . - . . . . -";

elseif value == "REUS"
    app.FrequencySpinner.Value = 114.2; % MHz

    I = imread('reus.png');
    h = imresize(I, [4000 4000]);
    image(app.MainScreenAxes,0, 0, h);
    app.MainScreenAxes.YDir = 'normal';

    app.VORSelectedInfoLabel.Text = "Reus VOR-DME";
    app.FrequencyInfoLabel.Text = "114.2 MHz";
    app.VORMorseCodeInfoLabel.Text = ". - . . . . .";

elseif value == "SABADELL"
    app.FrequencySpinner.Value = 112; % MHz

    I = imread('sabadell.png');
    h = imresize(I, [4000 4000]);
    image(app.MainScreenAxes,0, 0, h);
    app.MainScreenAxes.YDir = 'normal';

    app.VORSelectedInfoLabel.Text = "Sabadell VOR-DME";
    app.FrequencyInfoLabel.Text = "112 MHz";
    app.VORMorseCodeInfoLabel.Text = ". . . . - . . . . - .
.";

elseif value == "VILAFRANCA"
    app.FrequencySpinner.Value = 113.15; % MHz

    I = imread('vilafranca.png');
    h = imresize(I, [4000 4000]);

```

```

image(app.MainScreenAxes,0, 0, h);
app.MainScreenAxes.YDir = 'normal';

app.VORSelectedInfoLabel.Text = "Vilafranca VOR-DME";
app.FrequencyInfoLabel.Text = "113.15 MHz";
app.VORMorseCodeInfoLabel.Text = ". . . - . - . . -";

elseif value == "> Enter Manually <"

    I = imread('rose.png');
    h = imresize(I, [4000 4000]);
    image(app.MainScreenAxes,0, 0, h);
    app.MainScreenAxes.YDir = 'normal';

    app.VORSelectedInfoLabel.Text = "-";
    app.FrequencyInfoLabel.Text = "-";
    app.VORMorseCodeInfoLabel.Text = "-";

end
end

% Button pushed function: SoundButton
function SoundButtonPushed(app, event)
    if app.sound
        app.sound = false;
        app.SoundButton.Icon = 'volumen-off.png';
    else
        app.sound = true;
        app.SoundButton.Icon = 'volumen-on.png';
    end
end

end

% Value changed function: SpeedParametersSwitch
function SpeedParametersSwitchValueChanged(app, event)

    if app.SpeedParametersSwitch.Value == "knots"
        app.SpeedUnitsLabel.Text = "knots";
        % app.SpeedUnitsGaugeLabel.Text = "knots";
        speed = app.SpeedSpinner.Value * 1.852 * 1000 / 3600;
        app.SpeedSpinner.Value = speed;
        app.GroundSpeedGauge.Value = speed;
    else
        app.SpeedUnitsLabel.Text = "km/h";
        % app.SpeedUnitsGaugeLabel.Text = "km/h";
        speed = app.SpeedSpinner.Value / 1.852 / 1000 * 3600;
        app.SpeedSpinner.Value = speed;
        app.GroundSpeedGauge.Value = speed * 1.852 * 1000 / 3600;
    end
end

end

% Value changed function: SpeedSpinner
function SpeedSpinnerValueChanged(app, event)

    if app.SpeedParametersSwitch.Value == "knots"
        speed = app.SpeedSpinner.Value;
        app.GroundSpeedGauge.Value = speed;
    else
        speed = app.SpeedSpinner.Value * 1.852 * 1000 / 3600;
        app.GroundSpeedGauge.Value = speed;
    end
end

```

```

    app.velocidad = speed;

end

% Value changed function: AircraftPositionKnob
function AircraftPositionKnobValueChanged(app, event)
    value = app.AircraftPositionKnob.Value;

    app.MODEInfoLabel.Text = "Aircraft Position";

    app.SoundButton.Enable = true;

    if value == "On"

        app.running = true;
        app.cancel = false;

        app.STOPButton.Enable = true;
        app.FrequencySpinner.Enable = false;
        app.MHzLabel.Enable = false;
        app.SelectVORParametersListBox.Enable = false;
        app.RadialVisorKnob.Enable = false;
        app.PositionLamp.Color = [0 1 0];

        app.numeroLecturaRadial = 1;

        %% Procesado de señal
        app.i = 31; % interacción columna de la matriz amb datos
        (en caso de trabajar con señal grabado)
        j = 0; % interador del bucle
        app.tiempo_captura = app.muestras_captura/app.fs; % tiempo
        que dura una captura de datos [s]

        for tiempo = 0:app.tiempo_captura:app.tiempo_ejecucion %
        controls tiempo de ejecución [s]

            if app.cancel % si cancela la ejecución del programa
            -> salir del bucle
                break
            end

            TratamientoDigital(app); % calcula el radial

            if (mod(j,app.intervalos) == 0)
                ProcesarPosicion(app); % calcula la posicion

                if (app.numeroLecturaRadial ~= 0)
                    txt = 'NO AVAILABLE DATA'; % mensaje en inglés
                    porque lo tiene que leer el piloto
                    t = text(app.MainScreenAxes, 2000,1500,txt);
                    t.FontSize = 20;
                    t.HorizontalAlignment = 'center';
                    t.FontWeight = 'bold';
                    t.Color = [1, 0, 0];

                    elseif (isnan(app.puntoXRadial1) ||
                    isnan(app.puntoXRadial2) ||
                    isnan(app.puntoYRadial1) ||
                    isnan(app.puntoYRadial2) ||
                    isnan(app.puntoYRadial3))
                        txt = 'NO AVAILABLE DATA'; % mensaje en inglés
                        porque lo tiene que leer el piloto
                        t = text(app.MainScreenAxes, 2000,1500,txt);
                        t.FontSize = 20;

```

```

        t.HorizontalAlignment = 'center';
        t.FontWeight = 'bold';
        t.Color = [1, 0, 0];

        app.radial1 = app.radial2;
        app.radial2 = app.radial3;
    else
        CalcularHeadingAvion(app);
        RepresentacionGraficaPosicionYHeading(app);

        app.radial1 = app.radial2;
        app.radial2 = app.radial3;
    end

end

ReproduccionCodigoMorse(app); % reproduce codigo morse

RepresentacionGraficaOBS(app);

    app.i = app.i + 1; % en caso de trabajar con señal
grabado, iteración de la columna de datos
    j = j + 1;

    % crear interrupción y sincronizar los tiempos de
captación de señal y reproducción del código morse
    pause(app.tiempo_captura-0.01);

end

release(app.deviceWriter); % cerrar el objeto de audio

else
    EstablecerEncendido(app);
end
end

% Value changed function: RadialSpinner
function RadialSpinnerValueChanged(app, event)

    RepresentacionGraficaOBS(app);

end
end

% App initialization and construction
methods (Access = private)

    % Create UIFigure and components
function createComponents(app)

    % Create UIFigure
    app.UIFigure = uifigure;
    app.UIFigure.AutoResizeChildren = 'off';
    app.UIFigure.Color = [0.9412 0.9412 0.9412];
    app.UIFigure.Position = [100 100 1369 722];
    app.UIFigure.Name = 'UI Figure';
    app.UIFigure.Resize = 'off';

    % Create SelectVORParametersListBoxLabel
    app.SelectVORParametersListBoxLabel = uilabel(app.UIFigure);
    app.SelectVORParametersListBoxLabel.BackgroundColor = [0.651
0.651 0.651];

```

```

        app.SelectVORParametersListBoxLabel.HorizontalAlignment =
'center';
        app.SelectVORParametersListBoxLabel.FontName = 'Eurostile';
        app.SelectVORParametersListBoxLabel.FontSize = 20;
        app.SelectVORParametersListBoxLabel.Position = [28 492 205
22];
        app.SelectVORParametersListBoxLabel.Text = 'Select VOR
Parameters';

        % Create SelectVORParametersListBox
        app.SelectVORParametersListBox = uilistbox(app.UIFigure);
        app.SelectVORParametersListBox.Items = {'> Enter Manually <',
'BAGUR', 'BARCELONA 25R', 'CALELLA', 'EL PRAT 02', 'GIRONA', 'MAELLA',
'REUS', 'SABADELL', 'VILAFRANCA'};
        app.SelectVORParametersListBox.ValueChangedFcn =
createCallbackFcn(app, @SelectVORParametersListBoxValueChanged, true);
        app.SelectVORParametersListBox.Enable = 'off';
        app.SelectVORParametersListBox.FontName = 'Eurostile';
        app.SelectVORParametersListBox.FontSize = 20;
        app.SelectVORParametersListBox.Position = [31 248 200 242];
        app.SelectVORParametersListBox.Value = '> Enter Manually <';

        % Create MainScreenAxes
        app.MainScreenAxes = uiaxes(app.UIFigure);
        app.MainScreenAxes.XLim = [0 4000];
        app.MainScreenAxes.YLim = [0 4000];
        app.MainScreenAxes.Box = 'on';
        app.MainScreenAxes.XTick = [];
        app.MainScreenAxes.YTick = [];
        app.MainScreenAxes.Position = [261 139 560 564];

        % Create FrequencySpinnerLabel
        app.FrequencySpinnerLabel = uilabel(app.UIFigure);
        app.FrequencySpinnerLabel.HorizontalAlignment = 'right';
        app.FrequencySpinnerLabel.FontName = 'Eurostile';
        app.FrequencySpinnerLabel.Enable = 'off';
        app.FrequencySpinnerLabel.Position = [31 216 59 22];
        app.FrequencySpinnerLabel.Text = 'Frequency';

        % Create FrequencySpinner
        app.FrequencySpinner = uispinner(app.UIFigure);
        app.FrequencySpinner.Step = 0.05;
        app.FrequencySpinner.Limits = [108 118];
        app.FrequencySpinner.ValueDisplayFormat = '%.2f';
        app.FrequencySpinner.ValueChangedFcn = createCallbackFcn(app,
@FrequencySpinnerValueChanged, true);
        app.FrequencySpinner.FontName = 'Eurostile';
        app.FrequencySpinner.BackgroundColor = [1 0 0];
        app.FrequencySpinner.Enable = 'off';
        app.FrequencySpinner.Position = [100 216 96 22];
        app.FrequencySpinner.Value = 114.4;

        % Create MHzLabel
        app.MHzLabel = uilabel(app.UIFigure);
        app.MHzLabel.FontName = 'Eurostile';
        app.MHzLabel.Enable = 'off';
        app.MHzLabel.Position = [206 216 29 22];
        app.MHzLabel.Text = 'MHz';

        % Create Switch
        app.Switch = uiswitch(app.UIFigure, 'rocker');
        app.Switch.Orientation = 'horizontal';

```

```

        app.Switch.ValueChangedFcn = createCallbackFcn(app,
@SwitchValueChanged, true);
        app.Switch.BusyAction = 'cancel';
        app.Switch.FontName = 'Eurostile';
        app.Switch.FontSize = 16;
        app.Switch.Position = [55 664 80 36];

        % Create RadialVisorKnobLabel
        app.RadialVisorKnobLabel = uilabel(app.UIFigure);
        app.RadialVisorKnobLabel.BackgroundColor = [0.651 0.651 0.651];
        app.RadialVisorKnobLabel.HorizontalAlignment = 'center';
        app.RadialVisorKnobLabel.FontName = 'Eurostile';
        app.RadialVisorKnobLabel.FontSize = 14;
        app.RadialVisorKnobLabel.Position = [26 536 93 22];
        app.RadialVisorKnobLabel.Text = 'Radial Visor';

        % Create RadialVisorKnob
        app.RadialVisorKnob = uiknob(app.UIFigure, 'discrete');
        app.RadialVisorKnob.Items = {'Off', 'On'};
        app.RadialVisorKnob.ValueChangedFcn = createCallbackFcn(app,
@RadialVisorKnobValueChanged, true);
        app.RadialVisorKnob.Enable = 'off';
        app.RadialVisorKnob.FontName = 'Eurostile';
        app.RadialVisorKnob.Position = [43 565 60 60];

        % Create GroundSpeedGauge
        app.GroundSpeedGauge = uigauge(app.UIFigure, 'circular');
        app.GroundSpeedGauge.Limits = [0 450];
        app.GroundSpeedGauge.MajorTicks = [0 50 100 150 200 250 300
350 400 450];
        app.GroundSpeedGauge.ScaleColors = [1 0 0;0.929411764705882
0.690196078431373 0.129411764705882];
        app.GroundSpeedGauge.ScaleColorLimits = [425 450;400 425];
        app.GroundSpeedGauge.BackgroundColor = [0.149 0.149 0.149];
        app.GroundSpeedGauge.FontName = 'Eurostile';
        app.GroundSpeedGauge.FontSize = 9;
        app.GroundSpeedGauge.FontColor = [1 1 1];
        app.GroundSpeedGauge.Position = [1118 298 215 215];
        app.GroundSpeedGauge.Value = 50;

        % Create SystemLamp
        app.SystemLamp = uilamp(app.UIFigure);
        app.SystemLamp.Position = [181 667 30 30];
        app.SystemLamp.Color = [1 0 0];

        % Create STOPButton
        app.STOPButton = uibutton(app.UIFigure, 'push');
        app.STOPButton.ButtonPushedFcn = createCallbackFcn(app,
@STOPButtonPushed, true);
        app.STOPButton.Interruptible = 'off';
        app.STOPButton.Enable = 'off';
        app.STOPButton.Position = [31 90 200 41];
        app.STOPButton.Text = 'STOP';

        % Create LoadFrequencyButton
        app.LoadFrequencyButton = uibutton(app.UIFigure, 'push');
        app.LoadFrequencyButton.ButtonPushedFcn =
createCallbackFcn(app, @LoadFrequencyButtonPushed, true);
        app.LoadFrequencyButton.Enable = 'off';
        app.LoadFrequencyButton.Position = [31 152 200 43];
        app.LoadFrequencyButton.Text = 'Load Frequency';

        % Create SoundButton

```

```

app.SoundButton = uibutton(app.UIFigure, 'push');
    app.SoundButton.ButtonPushedFcn = createCallbackFcn(app,
@SoundButtonPushed, true);
app.SoundButton.Icon = 'volumen-on.png';
app.SoundButton.Enable = 'off';
app.SoundButton.Position = [31 23 200 47];
app.SoundButton.Text = '';

% Create HeadingAxes
app.HeadingAxes = uiaxes(app.UIFigure);
app.HeadingAxes.XTick = [];
app.HeadingAxes.XTickLabel = {'[ ]'};
app.HeadingAxes.YTick = [];
app.HeadingAxes.Color = [0.9412 0.9412 0.9412];
app.HeadingAxes.Position = [851 474 222 223];

% Create OBSAxes
app.OBSAxes = uiaxes(app.UIFigure);
app.OBSAxes.XTick = [];
app.OBSAxes.XTickLabel = {'[ ]'};
app.OBSAxes.YTick = [];
app.OBSAxes.Position = [851 149 222 223];

% Create OBSLabel
app.OBSLabel = uilabel(app.UIFigure);
app.OBSLabel.BackgroundColor = [0.651 0.651 0.651];
app.OBSLabel.HorizontalAlignment = 'center';
app.OBSLabel.FontName = 'Eurostile';
app.OBSLabel.FontSize = 14;
app.OBSLabel.Position = [851 109 222 22];
app.OBSLabel.Text = 'OBS';

% Create HeadingLabel
app.HeadingLabel = uilabel(app.UIFigure);
app.HeadingLabel.BackgroundColor = [0.651 0.651 0.651];
app.HeadingLabel.HorizontalAlignment = 'center';
app.HeadingLabel.FontName = 'Eurostile';
app.HeadingLabel.FontSize = 14;
app.HeadingLabel.Position = [851 433 222 22];
app.HeadingLabel.Text = 'Heading';

% Create AircraftPositionKnobLabel
app.AircraftPositionKnobLabel = uilabel(app.UIFigure);
    app.AircraftPositionKnobLabel.BackgroundColor = [0.651 0.651
0.651];
app.AircraftPositionKnobLabel.HorizontalAlignment = 'center';
app.AircraftPositionKnobLabel.FontName = 'Eurostile';
app.AircraftPositionKnobLabel.FontSize = 14;
app.AircraftPositionKnobLabel.Position = [140 536 100 22];
app.AircraftPositionKnobLabel.Text = 'Aircraft Position';

% Create AircraftPositionKnob
app.AircraftPositionKnob = uiknob(app.UIFigure, 'discrete');
app.AircraftPositionKnob.Items = {'Off', 'On'};
    app.AircraftPositionKnob.ValueChangedFcn =
createCallbackFcn(app, @AircraftPositionKnobValueChanged, true);
app.AircraftPositionKnob.Enable = 'off';
app.AircraftPositionKnob.FontName = 'Eurostile';
app.AircraftPositionKnob.Position = [159 565 60 60];

% Create SpeedUnitsLabel
app.SpeedUnitsLabel = uilabel(app.UIFigure);
app.SpeedUnitsLabel.FontName = 'Eurostile';

```

```

app.SpeedUnitsLabel.Enable = 'off';
app.SpeedUnitsLabel.Position = [1286 550 33 22];
app.SpeedUnitsLabel.Text = 'knots';

% Create SpeedLabel
app.SpeedLabel = uilabel(app.UIFigure);
app.SpeedLabel.BackgroundColor = [0.9412 0.9412 0.9412];
app.SpeedLabel.HorizontalAlignment = 'right';
app.SpeedLabel.FontName = 'Eurostile';
app.SpeedLabel.Enable = 'off';
app.SpeedLabel.Position = [1140 550 39 22];
app.SpeedLabel.Text = 'Speed: ';

% Create SpeedSpinner
app.SpeedSpinner = uispinner(app.UIFigure);
app.SpeedSpinner.Limits = [1 1000];
app.SpeedSpinner.ValueDisplayFormat = '%.0f';
app.SpeedSpinner.ValueChangedFcn = createCallbackFcn(app,
@SpeedSpinnerValueChanged, true);
app.SpeedSpinner.FontName = 'Eurostile';
app.SpeedSpinner.Enable = 'off';
app.SpeedSpinner.Position = [1184 550 96 22];
app.SpeedSpinner.Value = 50;

% Create PositionLamp
app.PositionLamp = uilamp(app.UIFigure);
app.PositionLamp.Position = [185 630 10 10];
app.PositionLamp.Color = [1 0 0];

% Create RadialLamp
app.RadialLamp = uilamp(app.UIFigure);
app.RadialLamp.Position = [69 630 10 10];
app.RadialLamp.Color = [1 0 0];

% Create SpeedParametersSwitchLabel
app.SpeedParametersSwitchLabel = uilabel(app.UIFigure);
app.SpeedParametersSwitchLabel.BackgroundColor = [0.651 0.651
0.651];
app.SpeedParametersSwitchLabel.HorizontalAlignment = 'center';
app.SpeedParametersSwitchLabel.FontName = 'Eurostile';
app.SpeedParametersSwitchLabel.FontSize = 20;
app.SpeedParametersSwitchLabel.Position = [1115 626 223 22];
app.SpeedParametersSwitchLabel.Text = 'Speed Parameters';

% Create SpeedParametersSwitch
app.SpeedParametersSwitch = uiswitch(app.UIFigure, 'toggle');
app.SpeedParametersSwitch.Items = {'knots', 'km/h'};
app.SpeedParametersSwitch.Orientation = 'horizontal';
app.SpeedParametersSwitch.ValueChangedFcn =
createCallbackFcn(app, @SpeedParametersSwitchValueChanged, true);
app.SpeedParametersSwitch.Enable = 'off';
app.SpeedParametersSwitch.FontName = 'Eurostile';
app.SpeedParametersSwitch.FontSize = 16;
app.SpeedParametersSwitch.Position = [1187 582 81 36];
app.SpeedParametersSwitch.Value = 'knots';

% Create GroundSpeedLabel
app.GroundSpeedLabel = uilabel(app.UIFigure);
app.GroundSpeedLabel.BackgroundColor = [0.651 0.651 0.651];
app.GroundSpeedLabel.HorizontalAlignment = 'center';
app.GroundSpeedLabel.FontName = 'Eurostile';
app.GroundSpeedLabel.FontSize = 14;
app.GroundSpeedLabel.Position = [1114 251 223 22];

```

```

app.GroundSpeedLabel.Text = 'Ground Speed';

% Create SpeedUnitsGaugeLabel
app.SpeedUnitsGaugeLabel = uilabel(app.UIFigure);
app.SpeedUnitsGaugeLabel.HorizontalAlignment = 'center';
app.SpeedUnitsGaugeLabel.FontName = 'Eurostile';
app.SpeedUnitsGaugeLabel.FontColor = [1 1 1];
app.SpeedUnitsGaugeLabel.Position = [1209 314 33 22];
app.SpeedUnitsGaugeLabel.Text = 'knots';

% Create RadialLabel
app.RadialLabel = uilabel(app.UIFigure);
app.RadialLabel.BackgroundColor = [0.9412 0.9412 0.9412];
app.RadialLabel.HorizontalAlignment = 'right';
app.RadialLabel.FontName = 'Eurostile';
app.RadialLabel.Enable = 'off';
app.RadialLabel.Position = [892 37 38 22];
app.RadialLabel.Text = 'Radial: ';

% Create RadialSpinner
app.RadialSpinner = uispinner(app.UIFigure);
app.RadialSpinner.Limits = [0 360];
app.RadialSpinner.ValueDisplayFormat = '%.0f';
    app.RadialSpinner.ValueChangedFcn = createCallbackFcn(app,
@RadialSpinnerValueChanged, true);
app.RadialSpinner.FontName = 'Eurostile';
app.RadialSpinner.Enable = 'off';
app.RadialSpinner.Position = [935 37 96 22];

% Create AuthorLabel
app.AuthorLabel = uilabel(app.UIFigure);
app.AuthorLabel.BackgroundColor = [0.902 0.902 0.902];
app.AuthorLabel.HorizontalAlignment = 'center';
app.AuthorLabel.FontName = 'Eurostile';
app.AuthorLabel.FontSize = 18;
app.AuthorLabel.FontWeight = 'bold';
app.AuthorLabel.FontColor = [0.502 0.502 0.502];
app.AuthorLabel.Position = [1120 23 224 104];
app.AuthorLabel.Text = {'Author: '; 'ALEX LOMBAO CHIUTA'; ''
'Universitat'; 'Rovira i Virgili'};

% Create TOFROMLabel
app.TOFROMLabel = uilabel(app.UIFigure);
app.TOFROMLabel.BackgroundColor = [0.902 0.902 0.902];
app.TOFROMLabel.HorizontalAlignment = 'center';
app.TOFROMLabel.FontName = 'Eurostile';
app.TOFROMLabel.FontSize = 20;
app.TOFROMLabel.FontWeight = 'bold';
app.TOFROMLabel.FontColor = [1 0 0];
app.TOFROMLabel.Enable = 'off';
app.TOFROMLabel.Position = [851 75 222 22];
app.TOFROMLabel.Text = 'TO';

% Create VORSelectedLabel
app.VORSelectedLabel = uilabel(app.UIFigure);
app.VORSelectedLabel.FontName = 'Eurostile';
app.VORSelectedLabel.FontSize = 14;
app.VORSelectedLabel.Position = [261 77 87 22];
app.VORSelectedLabel.Text = 'VOR Selected: ';

% Create MODELabel
app.MODELabel = uilabel(app.UIFigure);
app.MODELabel.FontName = 'Eurostile';

```

```

app.MODELabel.FontSize = 14;
app.MODELabel.Position = [261 105 45 22];
app.MODELabel.Text = 'MODE: ';

% Create FrequencyLabel
app.FrequencyLabel = uilabel(app.UIFigure);
app.FrequencyLabel.FontName = 'Eurostile';
app.FrequencyLabel.FontSize = 14;
app.FrequencyLabel.Position = [261 50 68 22];
app.FrequencyLabel.Text = 'Frequency: ';

% Create VORMorseCodeLabel
app.VORMorseCodeLabel = uilabel(app.UIFigure);
app.VORMorseCodeLabel.FontName = 'Eurostile';
app.VORMorseCodeLabel.FontSize = 14;
app.VORMorseCodeLabel.Position = [261 23 108 22];
app.VORMorseCodeLabel.Text = 'VOR Morse Code: ';

% Create LiveActualRadialLabel
app.LiveActualRadialLabel = uilabel(app.UIFigure);
app.LiveActualRadialLabel.FontName = 'Eurostile';
app.LiveActualRadialLabel.FontSize = 14;
app.LiveActualRadialLabel.Position = [612 109 109 22];
app.LiveActualRadialLabel.Text = 'Live Actual Radial: ';

% Create Timeintervalbetweenradiallectures156secondsLabel
app.Timeintervalbetweenradiallectures156secondsLabel =
uilabel(app.UIFigure);
app.Timeintervalbetweenradiallectures156secondsLabel.Horizont
alAlignment = 'center';
app.Timeintervalbetweenradiallectures156secondsLabel.FontName
= 'Eurostile';
app.Timeintervalbetweenradiallectures156secondsLabel.FontSize
= 14;
app.Timeintervalbetweenradiallectures156secondsLabel.Position
= [510 23 311 22];
app.Timeintervalbetweenradiallectures156secondsLabel.Text =
'Time interval between radial lectures: 1.56 seconds';

% Create MODEInfoLabel
app.MODEInfoLabel = uilabel(app.UIFigure);
app.MODEInfoLabel.FontName = 'Eurostile';
app.MODEInfoLabel.FontSize = 14;
app.MODEInfoLabel.FontWeight = 'bold';
app.MODEInfoLabel.Position = [383 105 117 22];
app.MODEInfoLabel.Text = '-';

% Create VORSelectedInfoLabel
app.VORSelectedInfoLabel = uilabel(app.UIFigure);
app.VORSelectedInfoLabel.FontName = 'Eurostile';
app.VORSelectedInfoLabel.FontSize = 14;
app.VORSelectedInfoLabel.FontWeight = 'bold';
app.VORSelectedInfoLabel.Position = [383 77 145 22];
app.VORSelectedInfoLabel.Text = '-';

% Create FrequencyInfoLabel
app.FrequencyInfoLabel = uilabel(app.UIFigure);
app.FrequencyInfoLabel.FontName = 'Eurostile';
app.FrequencyInfoLabel.FontSize = 14;
app.FrequencyInfoLabel.FontWeight = 'bold';
app.FrequencyInfoLabel.Position = [383 50 117 22];
app.FrequencyInfoLabel.Text = '-';

```

```
% Create VORMorseCodeInfoLabel
app.VORMorseCodeInfoLabel = uilabel(app.UIFigure);
app.VORMorseCodeInfoLabel.FontName = 'Eurostile';
app.VORMorseCodeInfoLabel.FontSize = 14;
app.VORMorseCodeInfoLabel.FontWeight = 'bold';
app.VORMorseCodeInfoLabel.Position = [383 23 117 22];
app.VORMorseCodeInfoLabel.Text = '-';

% Create LiveRadialLabel
app.LiveRadialLabel = uilabel(app.UIFigure);
app.LiveRadialLabel.HorizontalAlignment = 'center';
app.LiveRadialLabel.FontName = 'Eurostile';
app.LiveRadialLabel.FontSize = 24;
app.LiveRadialLabel.FontWeight = 'bold';
app.LiveRadialLabel.Position = [542 76 248 25];
app.LiveRadialLabel.Text = '-';
end
end

methods (Access = public)

% Construct app
function app = prue

    % Create and configure components
    createComponents(app)

    % Register the app with App Designer
    registerApp(app, app.UIFigure)

    % Execute the startup function
    runStartupFcn(app, @startupFcn)

    if nargin == 0
        clear app
    end
end

% Code that executes before app deletion
function delete(app)

    % Delete UIFigure when app is deleted
    delete(app.UIFigure)
end
end
end
```