



ULPGC

Universidad de
Las Palmas de
Gran Canaria

Escuela de Ingeniería
de Telecomunicación y Electrónica

eit@

ESCUELA DE INGENIERÍA DE TELECOMUNICACIÓN Y ELECTRÓNICA



TRABAJO FIN DE MÁSTER

DISEÑO DE UN TRANSCÉPTOR PARA COMUNICACIONES MARÍTIMAS
DIGITALES SOBRE UNA PLATAFORMA DE RADIO DEFINIDA POR SOFTWARE

**Titulación: Máster Universitario en Ingeniería de
Telecomunicación**

Autor: Dayron Yaret Romero Godoy

Tutores: D. Nicolás Molina Padrón

D. Francisco José Cabrera Almeida

D. Víctor Alexis Araña Pulido

Fecha: septiembre de 2022

Agradecimientos

En la vida profesional, el ingeniero no trabaja en solitario, sino que se incorpora en un equipo con compañeros que comparten profesión u objetivos. Es por ello, que me gustaría agradecer el apoyo incondicional de mis padres, Ana Rosa Godoy Déniz y Antonio Romero González, que me han educado y guiado en mis peores momentos. Es gracias a ellos por lo que soy la persona que soy hoy en día. También me gustaría agradecer el apoyo y ayuda prestada por mis amigos Sergio Celada, Antonio Galván, Jorge Monzón, Alberto Ramos y Javier Verona, ya que juntos conseguimos pasar asignatura por asignatura al compartir horas de estudio, resultados y apuntes. Me gustaría agradecer también a los compañeros que conocí en mi época por el IDeTIC (Instituto Universitario para el Desarrollo Tecnológico y la Innovación en Comunicaciones) mientras realizaba este trabajo, en especial a Víctor Alonso, Daniel Moreno y Jaime Ticay, que respondían mis dudas espontáneas. Como no, agradecer a mis tutores, Nicolás Molina Padrón, Francisco José Cabrera Almeida y Víctor Alexis Araña Pulido, por su guía y tutelaje y en especial a Nico, por su eterna paciencia y apoyo durante la realización de este proyecto. Por último, me gustaría agradecer a mi difunta mascota, Kiara, que falleció un 13 de junio con 17 años, por su lealtad, cariño y por enseñarme que es la resiliencia, al aguantar hasta su último aliento a mi lado.

Índice general

I Memoria	1
1. Introducción	3
1.1. Antecedentes	3
1.2. Objetivos	6
1.3. Estructura de la memoria	7
2. Comunicaciones marítimas AIS y VDES	9
2.1. Evolución de las comunicaciones marítimas digitales	9
2.2. Aspectos generales de los sistemas AIS y VDES	11
2.3. Capa física y de enlace del sistema AIS	16
2.3.1. Capa física del sistema AIS	17
2.3.2. Capa de enlace del sistema AIS	22
2.3.3. Tipos de mensajes AIS y capa de red	26
2.4. Capa física y de enlace del sistema VDES	30
2.4.1. Capa física del sistema VDES	30
2.4.2. Capa de enlace del sistema VDES	34
3. Fundamentos de sistemas de radio definida por software	37
3.1. Introducción a la radio definida por software	37

3.2. Bloques de un transceptor basado en radio definida por software	40
3.2.1. Etapa RF	41
3.2.2. Etapa de conversión de señales	43
3.2.3. Etapa de conversión digital	47
3.2.4. Procesado digital de señales	52
3.3. Arquitecturas basadas en radio definida por software	53
3.3.1. Arquitectura directa	54
3.3.2. Arquitectura FI-cero	55
3.3.3. Disposición del DDC/DUC	57
3.4. Plataformas comerciales de radio definida por software	58
3.4.1. RTL-SDR	59
3.4.2. HackRF One	60
3.4.3. BladeRF	61
3.4.4. ADALM-Pluto	62
3.4.5. Serie USRP	64
3.4.6. LimeSDR	65
3.4.7. Comparación de plataformas comerciales SDR	65
4. Implementación de algoritmos AIS y VDES para plataformas SDR	67
4.1. Parámetros de configuración del SDR	67
4.2. Algoritmo de transmisión AIS	69
4.2.1. Generación de mensajes NMEA0183 AIS	69
4.2.2. Generación de tramas AIS y transmisión	71
4.3. Algoritmo de recepción AIS	78
4.3.1. Rastreo de señales AIS	79
4.3.2. Demodulación y decodificación de paquetes AIS	85

4.3.3.	Algoritmo de transmisión VDES	94
4.4.	Selección de servicio y configuración de parámetros VDES	94
4.4.1.	Generación aleatoria del paquete binario	97
4.4.2.	Modulación y transmisión de paquetes	100
5.	Diseño del módulo hardware de filtrado y amplificación	103
5.1.	Introducción de conceptos sobre filtros y amplificadores	103
5.1.1.	Revisión de conceptos prácticos sobre filtros	104
5.1.2.	Revisión de conceptos prácticos sobre amplificadores	105
5.2.	Etapas de filtrado y amplificación en recepción	106
5.2.1.	Filtro TA0395A	106
5.2.2.	Amplificador TQP3M9009	107
5.2.3.	Análisis de diferentes configuraciones de la etapa	108
5.3.	Etapas de amplificación en transmisión	109
5.3.1.	Elección de un disipador de potencia	110
5.4.	Diseño e implementación del circuito	114
5.5.	Caracterización de antenas para recepción y transmisión	116
6.	Pruebas y resultados	119
6.1.	Caracterización de la plataforma ADALM-PLUTO	119
6.2.	Verificación y resultados de transmisión y recepción de mensajes AIS en entornos controlados y abierto	126
6.2.1.	Pruebas realizadas en banda base	126
6.2.2.	Pruebas realizadas en entorno controlado	130
6.2.3.	Pruebas realizadas en entorno real	135
6.3.	Verificación y resultados de transmisión de señales VDE-TER en entornos controlados	138

6.4. Verificación y resultados del módulo hardware	140
6.4.1. Verificación y resultados de la etapa receptora del módulo hardware	140
6.4.2. Verificación y resultados de la etapa transmisora módulo hardware	143
7. Conclusiones y líneas futuras	145
7.1. Análisis del diseño y conclusiones	145
7.2. Líneas futuras	146
II Bibliografía	147
1. Bibliografía	149
III Presupuesto	155
Presupuesto	157
P.1. Introducción	157
P.2. Recursos Materiales	157
P.2.1. Recursos Hardware	158
P.2.2. Recursos Software	159
P.3. Recursos Humanos	160
P.4. Redacción del documento	161
P.5. Derechos de visado del COIT	161
P.6. Gastos de tramitación y envío	162
P.7. Aplicación de impuestos y coste final	162
IV Anexos	165
A. Instalación de GNU Radio en Linux	167

Índice de figuras

1.1. Enlace de comunicación basado en AIS. Fuente [5].	4
1.2. Diagrama de bloques de la arquitectura radio tradicional y la arquitectura SDR. .	5
1.3. Diagrama de bloques del prototipo.	6
2.1. Plan de frecuencias para el sistema VDES. Fuente [8].	14
2.2. Comunicación entre segmentos del sistema VDES.	15
2.3. Arquitectura OSI del sistema AIS.	17
2.4. Ejemplo de codificación NRZI.	19
2.5. Espectro de las modulaciones MSK y GMSK.	20
2.6. Constelación de la modulación GMSK.	21
2.7. Respuesta al impulso de un filtro gaussiano con factor $BT = 0.3$ y $BT = 0.5$	22
2.8. Esquemas TDMA utilizados en AIS. Fuente [18].	24
2.9. Espectros de la modulación QPSK y $\pi/4$ -QPSK.	31
2.10. Constelaciones de las modulaciones $\pi/4$ -QPSK y QPSK.	31
2.11. Espectro y constelación de la modulación 8-PSK.	33
2.12. Espectro y constelación de la modulación 16-QAM.	33
3.1. Diagrama de bloques de la arquitectura radio tradicional y la arquitectura SDR. .	37
3.2. Primer equipo SDR Speakeasy.	38
3.3. Primera arquitectura de radio definida por software propuesta por Joseph Mitola III.	38

3.4. Arquitectura de un transceptor de radio definida por software.	39
3.5. Diagrama de bloques de un transceptor basado en radio definida por software genérico.	40
3.6. Arquitectura receptor heterodino.	41
3.7. Funcionamiento de un mezclador ideal.	42
3.8. Arquitectura transmisor homodino.	43
3.9. Diagrama de bloques de un transceptor basado en radio definida por software genérico.	44
3.10. Etapa de conversión discreto-continuo.	47
3.11. Etapa de interpolación.	48
3.12. Diagramas de bloques de un DDC/DUC.	48
3.13. Componentes IQ y señal suma.	49
3.14. Filtrado de términos de alta frecuencia.	50
3.15. Componentes IQ representados en el plano complejo.	51
3.16. Diagrama de bloques de un receptor SDR.	53
3.17. Diagrama de bloques de un transmisor SDR.	54
3.18. Diagrama de bloques de la arquitectura directa.	54
3.19. Diagrama de bloques de la arquitectura FI-cero.	56
3.20. Diagrama de bloque de un receptor SDR tradicional.	58
3.21. Nuevo diagrama de bloque propuesto de un receptor SDR.	58
3.22. Plataforma RTL-SDR.	59
3.23. Plataforma HackRF One.	60
3.24. Plataforma BladeRF xA9.	61
3.25. Plataforma ADALM-Pluto.	63
3.26. Plataforma USRP serie N200.	64
3.27. Plataforma LimeSDR.	65

4.1. Diagrama de flujo del algoritmo de transmisión de paquetes AIS.	69
4.2. Constelación de la señal AIS a transmitir.	77
4.3. Señal AIS en el dominio temporal.	78
4.4. Diagrama de flujo del algoritmo de recepción de paquetes AIS.	79
4.5. Simulador QK-AT011 y herramienta de configuración.	79
4.6. Señal AIS recibida.	81
4.7. Señal AIS recibida acortada.	82
4.8. Resultado de la compensación de frecuencia de la señal AIS.	84
4.9. Señal AIS recortada, con componente continua eliminada y compensada en frecuencia.	85
4.10. Señal AIS demodulada.	86
4.11. Decodificación y extracción de campos de la trama AIS.	90
4.12. Diagrama de flujo del algoritmo de recepción de paquetes VDES.	94
4.13. Constelación de las señales moduladas a transmitir.	100
5.1. Diagrama de bloques de los subsistemas del transceptor AIS y VDES.	103
5.2. Clasificación de filtros según su respuesta en frecuencia.	105
5.3. Simulación y caracterización del filtro TA0395A.	107
5.4. Simulación y caracterización del amplificador TQP3M9009.	108
5.5. Parámetros S de la topología con configuración filtro → amplificador → amplificador.	110
5.6. Modelo térmico equivalente del diseño planteado.	111
5.7. Modelo térmico equivalente del chip sin disipador.	112
5.8. Modelo térmico equivalente del chip con disipador.	113
5.9. Layout del módulo hardware diseñado para el ADALM-Pluto.	115
5.10. Módulo hardware de amplificación y filtrado.	115

5.11. Parámetro de fase de las antenas utilizadas.	116
5.12. Parámetro S11 de las antenas utilizadas.	117
5.13. ROE de las antenas utilizadas.	117
6.1. Fuga de OL de la plataforma SDR.	121
6.2. Diagrama de bloques del montaje realizado para las medidas de sensibilidad.	122
6.3. Configuración del generador de funciones sin utilizar el módulo amplificador.	122
6.4. Medida de potencia recibida en el ADALM-PLUTO sin módulo amplificador.	123
6.5. Potencia recibida al incorporar el módulo amplificador.	123
6.6. Configuración del generador de funciones con el módulo amplificador.	124
6.7. Potencia recibida con la nueva configuración del generador de funciones.	124
6.8. Señal recibida al emplear ambos amplificadores.	125
6.9. Potencia máxima de transmisión del ADALM-PLUTO.	125
6.10. Resultado del algoritmo de transmisión y recepción en banda base con una SNR de 0 dB o superior.	129
6.11. Resultado del algoritmo de transmisión y recepción en banda base con una SNR de -10 dB.	129
6.12. Resultado del algoritmo de transmisión y recepción en banda base con una SNR de -20 dB.	130
6.13. Traspondedor comercial SRT Marine Cobalt, junto a su kit de desarrollo.	131
6.14. Resultado al realizar 20 transmisiones con una ráfaga de 10 paquetes.	131
6.15. Resultado al realizar transmisiones durante 10 minutos.	132
6.16. Resultado al recibir transmisiones 100 transmisiones con el traspondedor Cobalt.	133
6.17. Recepción de tramas enviadas desde la plataforma ADALM-PLUTO.	134
6.18. Resultado de la ejecución de los algoritmos AIS implementados en dos ADALM- PLUTO.	135
6.19. Montaje del experimento en entorno real.	135

6.20. Distancia entre el IDeTIC y la costa.	136
6.21. Resultados tras 10 minutos de recepción de la plataforma con amplificador.	136
6.22. Resultados tras 10 minutos de recepción de la plataforma sin amplificador.	137
6.23. Datos del buque <i>Shoshinmaru</i> , recibido desde el IDeTIC.	137
6.24. Espectro de los servicios ASM y VDE-TER de 25 KHz y modulación $\pi/4$ -QPSK.	139
6.25. Espectro del servicio VDE-TER de 50 KHz y modulación 8-PSK.	139
6.26. Espectro del servicio VDE-TER de 100 KHz y modulación 16-QAM.	140
6.27. Espectro de la banda VHF antes de incorporar el módulo.	141
6.28. Espectro de la banda VHF tras incorporar el módulo.	141
6.29. Cálculo del punto 1 dB compresión y el punto de cruce de tercer orden.	142
6.30. Consumo de la rama receptora del módulo hardware.	143
6.31. Parámetro S11 y carta de Smith de la etapa transmisora del módulo.	143
A.1. Interfaz de GNU Radio.	169
A.2. Tipos de datos en GNU Radio.	170
A.3. Diagrama de bloques de un receptor radio FM.	171
A.4. Señal de radio FM recibida por el ADALM-PLUTO en GNU Radio.	172

Índice de tablas

2.1. Porcentajes de saturación de canal AIS en 2012.	13
2.2. Parámetros radioeléctricos del sistema AIS.	18
2.3. Parámetros especiales en transmisión.	18
2.4. Parámetros especiales en recepción.	18
2.5. Periodos de transmisiones de mensajes AIS según la velocidad y maniobras.	27
2.6. Listado de mensajes AIS y su función.	28
2.7. Ejemplo de mensaje AIS codificado con NMEA0183.	29
2.8. Resumen de parámetros radioeléctricos de los servicios VDES.	30
2.9. Requisitos del transmisor ASM.	32
2.10. Requisitos del receptor ASM.	32
2.11. Requisitos de un transmisor VDE-TER.	34
2.12. Campos de las tramas ASM.	35
2.13. Campos de las tramas VDE-TER.	35
3.1. Modelos en función de su chip y rango de frecuencia asociados.	60
3.2. Potencias máximas de transmisión según el rango de frecuencias para HackRF One.	61
3.3. Características de la FPGA según el modelo de la plataforma BladeRF.	62
3.4. Características de los integrados AD9363 y AD9364 del PlutoSDR.	63
3.5. Recopilación de las principales características de las plataformas analizadas.	66

4.1. Campos y tamaños asociados para los servicios ASM y VDE-TER.	97
6.1. Resultados de la medida de la sensibilidad del ADALM-PLUTO.	125
6.2. Resultados de la medida de la potencia de transmisión máxima del ADALM-PLUTO.	126
6.3. Resultados de las pruebas en banda base.	128
6.4. Resultados de las pruebas de recepción utilizando como transmisor el simulador QK-AT011.	132
6.5. Resultados de las pruebas de transmisión y recepción entre dos ADALM-PLUTO.	134
6.6. Resultados de las pruebas de recepción en entorno real.	138
6.7. Características medidas de la rama receptora del módulo hardware.	142
P.1. Coste de los recursos hardware fungibles.	158
P.2. Coste de los recursos hardware no fungibles.	159
P.3. Coste de los recursos software.	160
P.4. Coste de los recursos humanos.	160
P.5. Presupuesto parcial.	161
P.6. Presupuesto con el coste de redacción.	162
P.7. Presupuesto total.	163

Índice de secciones de código

4.1. Creación de tramas NMEA0183 con PyAIS.	71
4.2. Trama NMEA0183 resultante.	71
4.3. Conversión de UTF-8 a UTF-6.	72
4.4. Tablas utilizadas para cambiar de ASCII 8 bits a ASCII bits.	72
4.5. Conversión de UTF-6 a hexadecimal.	73
4.6. Definición de parámetros y objeto SDR en transmisión.	74
4.7. Generación y codificación de la trama.	75
4.8. Función <i>Bytes_Flipped</i>	76
4.9. Función <i>Bits_Stuffing</i>	76
4.10. Declaración y uso de objeto modulador y representación de constelaciones.	77
4.11. Declaración y uso de objeto modulador y representación de constelaciones.	80
4.12. Recepción y recorte de señal.	82
4.13. Establecimiento del umbral de potencia y eliminación de la componente continua.	83
4.14. Compensación de frecuencia de la señal recibida.	85
4.15. Establecimiento del umbral de potencia y eliminación de la componente continua.	87
4.16. Búsqueda del <i>flag</i> de inicio y cálculo de la longitud del campo de datos.	89
4.17. Decodificación de los datos de la trama AIS recibida.	90
4.18. Definición de servicios VDES.	95
4.19. Generación de paquetes VDES.	98

4.20. Modulaciones y transmisión de paquetes VDES. 100

6.1. Algoritmo del analizador de espectros. 121

6.2. Algoritmo de transmisión y recepción en banda base. 126

Resumen

Desde el accidente del Titanic en 1912, la seguridad en el ámbito marítimo se convirtió en un campo de especial interés para la navegación. Hoy en día, el AIS (*Automatic Identification System*) es el principal sistema de comunicaciones buque a buque o estación a buque. Su funcionamiento es enviar datos de velocidad, posición y rumbo entre los datos enviados para evitar colisiones entre embarcaciones y mejorar la seguridad en el mar. Con el paso de los años, el uso del AIS se ha extendido a niveles donde sus dos canales de comunicaciones pueden llegar a verse saturado en zonas de alta densidad de tráfico. Es por ello, que se ha propuesto el VDES (*VHF Data Exchange System*). Incorpora cuatro servicios diferentes, además del AIS, junto a nuevos canales, esquemas de modulación y anchos de banda de transmisión diferentes. Sin embargo, la transición entre estos sistemas no es tan sencilla. En primer lugar, se han de reemplazar todos los dispositivos AIS existentes y, en segundo lugar, el alto coste de adquisición que conlleva la implementación de este nuevo sistema. Por esta razón, la tecnología SDR (*Software-Defined Radio*) ha ganado relevancia, ya que es capaz de abaratar los costes de producción y adquisición, al definir la mayoría de la funcionalidad de sus bloques mediante software. El objetivo de este Trabajo Fin de Máster es desarrollar un transceptor acorde con los estándares AIS y VDES, utilizando una plataforma SDR como base del prototipo, junto a la programación de algoritmos de transmisión y recepción. Para mejorar el rendimiento de esta plataforma, también se diseña un módulo hardware de amplificación y filtrado.

Palabras clave: Automatic Identification System – Comunicaciones marítimas digitales – Software-Defined Radio – Transceptor – VHF Data Exchange System

Abstract

Since the Titanic accident in 1912, maritime safety has become a field of special interest to shipping. Today, AIS (Automatic Identification System) is the main ship-to-ship or station-to-ship communications system. It works by sending data on speed, position, and course among the data sent to avoid collisions between vessels and improve safety at sea. Over the years, the use of AIS has been extended to levels where its two communication channels can become saturated in areas of high traffic density. That is why the VDES (VHF Data Exchange System) has been proposed. It incorporates four different services, in addition to AIS, together with new channels, modulation schemes, and different transmission bandwidths. However, the transition between these systems is not so simple. Firstly, all existing AIS devices have to be replaced, and, secondly, the high acquisition cost involved in implementing this new system. For this reason, SDR (Software-Defined Radio) technology has gained relevance, since it can reduce production and acquisition costs by defining most of the functionality of its blocks through software. The objective of this Master Thesis is to develop a transceiver compliant with AIS and VDES standards, using an SDR platform as the basis of the prototype together with the programming of transmitting and receiving algorithms. To improve the performance of this platform, an amplification and filtering hardware module is also designed.

Keywords: Automatic Identification System – Maritime digital communication – Software-Defined Radio – Transceiver – VHF Data Exchange System

Acrónimos

Siglas	Descripción
ADC	Analog-to-Digital Conversor
AIS	Automatic Identification System
AM	Amplitude Modulation
ASM	Application Specific Message
AtoN	Aids-to-Navigation
BER	Bit Error Rate
CAG	Control Automático de ganancia
CQD	Come Quick, Distress
CRC	Código de Redundancia Cíclica
CSTDMA	Carrier-Sense Time Division Multiple Access
DAC	Digital-to-Analog Conversor
DDC	Digital Down Converter
DLS	Data Link Service
DLM	Data Link Management
DSP	Digital Signal Processor
DUC	Digital Up Converter
DVB-T	Digital Video Broadcasting - Terrestrial
ENOB	Enough Number Of Bits
FATDMA	Fixed-Access Time Division Multiple Access
FEC	Forward Error Correction
FI	Frecuencia Intermedia
FM	Frequency Modulation

FPGA	Field Programmable Gate Array
GMSK	Gaussian Minimum-Shift Keying
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
Gsps	Giga samples per symbol
HDL	Hardware Description Language
HF	High Frequency
IALA	International Association of Marine Aids to Navigation and Lighthouse Authorities
IDeTIC	Instituto para el Desarrollo Tecnológico y la Innovación en Comunicaciones
IEC	International Electrotechnical Commission
IMO	International Maritime Organization
IoV	Internet of Vessels
IQ	In-Phase and Quadrature
ISI	Inter-Symbol Interference
ITDMA	Incremental Time Division Multiple Access
ITU	International Telecommunication Union
LME	Link Management Entity
LNA	Low Noise Amplifier
LR-AIS	Long Range AIS
LSB	Least Significant Bit
MAC	Medium Access Control
MIMO	Multiple Input Multiple Access
MMSI	Maritime Mobile Service Identity
MN	Milla Náutica
MSK	Minimum-Shift Keying
Msps	Mega samples per symbol
NAVTEX	NAVigational TEXt messages
NCO	Numerically-Controlled Oscillator
NMEA	National Marine Electronics Association
NRZI	Non-Return to Zero Inverted
OL	Oscilador Local

OSI	Open System Interconnection
PLL	Phase-Locked Loop
PPA	Personal Package Archives
PSK	Phase-Shift Keying
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase-Shift Keying
RATDMA	Random Access Time Division Multiple Access
RF	Radio Frequency
RMS	Root Mean Square
ROE	Relación de Onda Estacionaria
SAR	Search And Rescue
SART	Search And Rescue Transponder
SAW	Surface Acoustic Wave
SCR	Software-Controlled Radio
SDK	Software Development Kit
SDR	Software-Defined Radio
SINAD	Signal-to-Noise-and-Distortion
SNR	Signal-to-Noise Ratio
SOLAS	Safety Of Life At Sea
SOTDMA	Self-Organized Time Division Multiple Access
TBB	Terrestrial Bulletin Board
TCP	Transport Control Protocol
TDMA	Time Division Multiple Access
TFM	Trabajo Fin de Máster
THD	Total Armonic Distortion
UDP	User Datagram Protocol
UHF	Ultra High Frequency
USB	Universal Serial Bus
USRP	Universal Software Radio Peripheral
UTC	Universal Time Coordinated
VDE-SAT	VHF Data Exchange for Satelital Links
VDE-TER	VHF Data Exchange for Terrestrial Links

VHF Very High Frequency

Parte I

Memoria

1 | Introducción

1.1. Antecedentes

Desde el accidente del Titanic en 1912 [1], las medidas de seguridad marítima se han visto reforzadas a través de una serie de normas internacionales. El convenio SOLAS (*Safety Of Life At Sea*), celebrado en 1914, fue un hito histórico donde se aprobaron las primeras normas internacionales para la seguridad en el mar, muchas de ellas vigentes en la actualidad. En los años siguientes, las propuestas y modificaciones de estas normas internacionales propició la necesidad de crear un organismo internacional para gestionar las enmiendas del convenio SOLAS. En 1948, tras una conferencia realizada en Ginebra (Suiza), se constituyó formalmente la IMO (*International Maritime Organization*). Esta entidad se encarga de proporcionar mecanismos de cooperación entre países en el ámbito de la reglamentación y tecnología marítimas, y su principal objetivo es aprobar o rechazar las enmiendas del convenio SOLAS.

En 1988 se adoptó el GMDSS (*Global Maritime Distress Safety System*) [2]. Esta red está constituida por las radiocomunicaciones marítimas terrestres (MRC, *Maritime Radio Communication*), como el DSC (*Digital Selective Calling*) en HF y VHF y por satélite (MSC, *Maritime Satellite Communication*), como pueden ser Inmarsat y COSPAS-SARSAT. La integración de estas redes debe asegurar que, independientemente del lugar en el que se encuentre un buque en situación de peligro, se pueda enviar una alerta y así movilizar las ayudas pertinentes. También garantiza la llegada de la información de seguridad marítima (MSI, *Maritime Safety Information*) a nivel global, como son la información meteorológica (típicamente aportada por el sistema NAVTEX, *NAVigational TEXt messages*) y la información de navegación.

Actualmente, las comunicaciones marítimas tienden a transmitir una mayor cantidad de información utilizando menores anchos de banda. Debido a que el espectro radioeléctrico es un recurso

finito y demandado, surge la necesidad de sustituir los sistemas analógicos por digitales, reduciendo así los requisitos de ancho de banda [3]. Ante este reto, y con el objetivo de reducir los errores humanos, se plantea una renovación del sistema GMDSS. Esta renovación ha dado paso a sistemas de comunicaciones digitales modernos, siendo el AIS (*Automatic Identification System*) el máximo exponente de esta transición.

El AIS [4] fue desarrollado en 1998 con la colaboración de varios organismos internacionales como IMO, IALA (*International Association of Marine Aids to Navigation and Lighthouse Authorities*), ITU (*International Telecommunication Union*) e IEC (*International Electrotechnical Commission*). Se trata de un sistema de comunicaciones ideado para embarcaciones e infraestructuras costeras que permite el intercambio de parámetros de navegación, como la posición, velocidad, rumbo y puerto de origen y destino. Con estas transmisiones, los buques son capaces de alertar su presencia y conocer las condiciones del tráfico marítimo. En la figura 1.1 se aporta un diagrama sobre los enlaces de comunicación entre distintas entidades AIS.

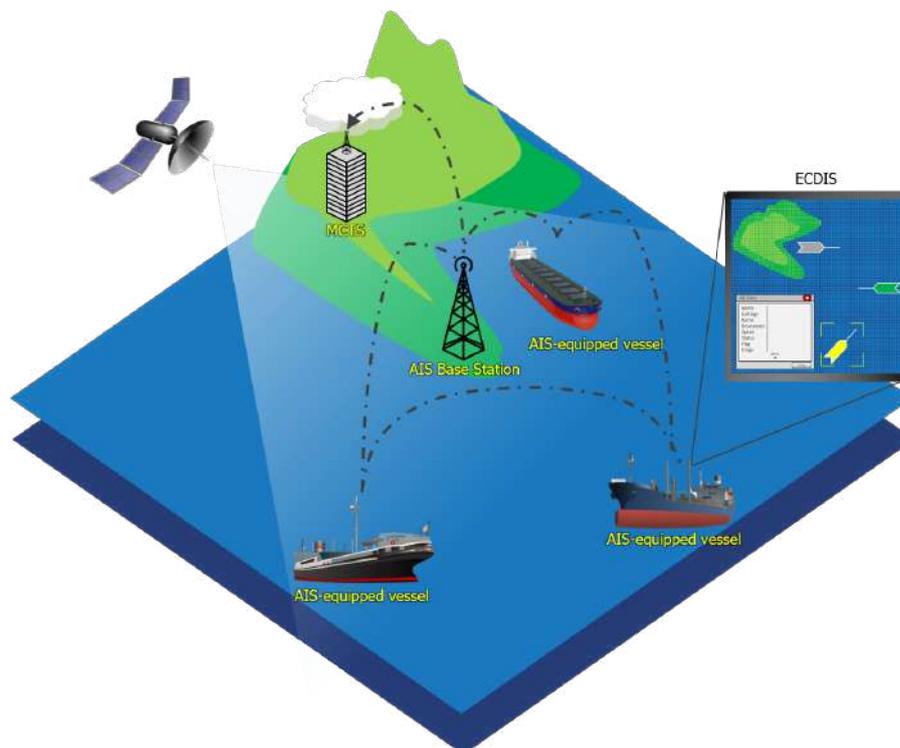


Figura 1.1: Enlace de comunicación basado en AIS. Fuente [5].

En un principio, AIS fue concebido únicamente como un sistema anticolidión de barcos, aunque en los últimos años han surgido aplicaciones en ámbitos tan diversos como el control de flotas y seguimientos de buques [6] y la monitorización medioambiental y de la biodiversidad [7], entre otras.

En 2013, con el objetivo de abordar la saturación de los canales AIS, la IALA junto con otras organizaciones propusieron una nueva versión del AIS denominada VDES (*VHF Data Exchange System*) [8]. Este sistema, aprobado en febrero de 2022, integra el AIS y añade nuevos servicios que emplean tasas de transmisión y canalizaciones mayores, así como nuevos esquemas de modulación y codificación.

Con el aumento de las tasas de transmisión en los servicios VDES respecto a las alcanzadas con el AIS, el VDES es considerado un buen candidato para muchos de los paradigmas de las comunicaciones y navegación marítimas que se pretenden implementar en el futuro, como puede ser la radio cognitiva marítima [9] o el IoV (*Internet of Vessels*) [10]. La clave se encuentra en su capacidad de interoperabilidad y su estandarización global [11].

A pesar de los aspectos positivos que ofrece el VDES, aún quedan algunas cuestiones técnicas por resolver. Los cambios en la capa física del VDES suponen modificaciones en el hardware de los equipos de comunicaciones utilizados para el AIS. Esto se debe a que deben ser capaces de soportar nuevos esquemas de modulación y codificación, además de anchos de banda mayores. Por lo tanto, es lógico suponer un incremento en el coste de los equipos respecto a los actuales equipos AIS [12], por lo que una alternativa a considerar sería la tecnología de radio definida por software o SDR (*Software-Defined Radio*).

Los dispositivos SDR son equipos de radiocomunicaciones que permiten implementar gran parte de las funciones radioeléctricas mediante software, con lo que se simplifican las necesidades hardware. La arquitectura SDR permite configurar, sobre una misma plataforma hardware, diferentes esquemas de modulación, codificación y entramados (Figura 1.2). No obstante, presenta un consumo de potencia mayor que las arquitecturas radio tradicionales, además de contar con menor sensibilidad y escasa protección frente a ruido e interferencias.

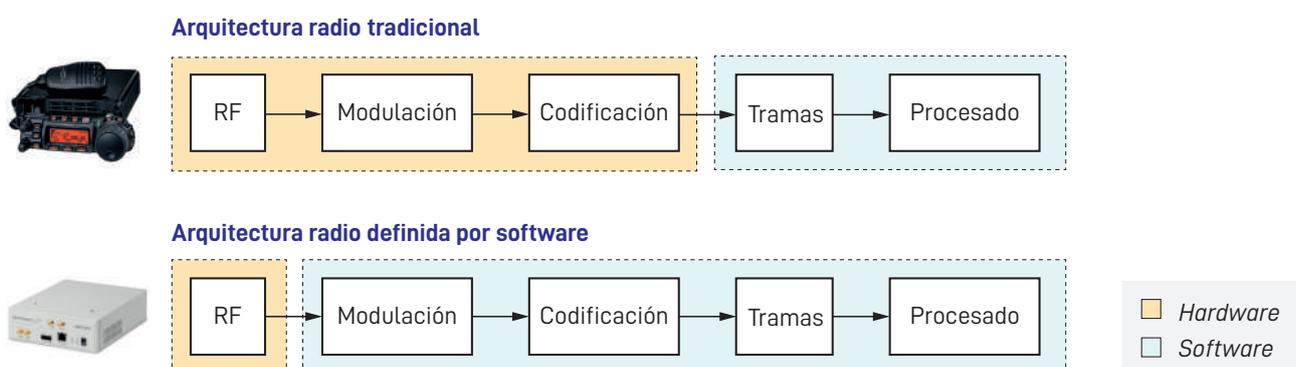


Figura 1.2: Diagrama de bloques de la arquitectura radio tradicional y la arquitectura SDR.

Un receptor SDR recibe las señales desde un cabezal RF clásico. Estas señales se trasladan a una banda de trabajo menor, comúnmente a banda base, para posteriormente ser muestreadas a través de un conversor analógico-digital. Las muestras se transforman en señales complejas en fase y cuadratura, que son tratadas por un procesador especializado mediante técnicas de procesamiento digital de señales. Un transmisor SDR genera una señal mediante software, que posteriormente es convertida de digital a analógica y, finalmente, es trasladada a una banda de trabajo superior.

1.2. Objetivos

Este TFM (*Trabajo Fin de Máster*) tiene como objetivo fundamental desarrollar un transceptor para el nuevo sistema de comunicaciones marítimas VDES sobre una plataforma SDR. Adicionalmente, se ha diseñado una etapa previa para mejorar las prestaciones del prototipo. Esta etapa consiste en un filtro y amplificadores en cascada para la rama de recepción y amplificadores de potencia para la rama de transmisión.

Como VDES implementa varios servicios, incluyendo el AIS se han programado una serie de *scripts* para la transmisión y recepción de paquetes del sistema AIS y para la transmisión del resto de servicios que engloba el sistema VDES.

Por último, el prototipo consiste en tres componentes principales:, que se representan en el diagrama de bloques de la figura 1.3

- Ordenador personal. Se encarga de configurar la plataforma SDR, generar los paquetes a transmitir y procesar las señales recibidas.
- Plataforma SDR. Su función es transmitir los paquetes previamente generados y de recibir señales.
- Etapa RF previa. Se ocupa de amplificar la señal en la etapa de transmisión y de amplificar y filtrar en la rama de recepción.

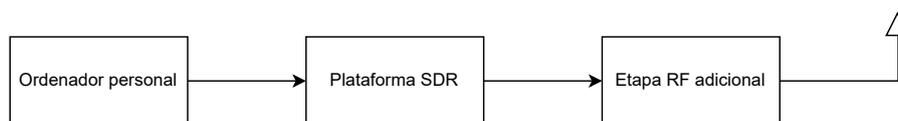


Figura 1.3: Diagrama de bloques del prototipo.

Para alcanzar el objetivo de este TFM se han seguido los siguientes pasos:

- Estudio en profundidad de los estándares AIS y VDES.
- Implementación de los algoritmos de transmisión y recepción para el estándar AIS y el algoritmo de transmisión del estándar VDES.
- Diseño de una etapa previa para mejorar las prestaciones del prototipo.

1.3. Estructura de la memoria

La memoria de este TFM se divide en siete capítulos, donde en el **capítulo 1** se realiza una introducción sobre la historia y evolución de las comunicaciones marítimas y los estándares AIS y VDES. Estos estándares son definidos detalladamente en el **capítulo 2**, explicando desde las características radioeléctricas como el funcionamiento de las capas que se recogen en sus recomendaciones. Para realizar la transición entre estándares, se propone el uso de plataformas SDR, concepto explicado detalladamente en el **capítulo 3**, desde su evolución y funcionamiento de sus bloques hasta las características de diferentes plataformas comerciales. En el **capítulo 4** se explica la metodología utilizada y los algoritmos implementados para la transmisión y recepción del estándar AIS y la transmisión de paquetes de los servicios de VDES. Con el objetivo de mejorar las prestaciones de la plataforma, en el **capítulo 5** se explica el proceso de diseño de un módulo hardware capaz de filtrar y amplificar señales comprendidas en la banda VHF marítima, explicando conceptos básicos de filtros y amplificadores y las medidas de sus parámetros más importantes. Seguidamente, en el **capítulo 6** se recoge un análisis de los resultados de las pruebas realizadas a la implementación de los algoritmos, el módulo hardware y una serie de comparaciones con un receptor y software comercial. Finalmente, en el **capítulo 7** se recogen las conclusiones y líneas futuras planteadas a partir de este trabajo.

Además de los capítulos definidos, se incluye la bibliografía utilizada y el presupuesto asociado a este TFM. Adicionalmente, en la sección de anexos se puede encontrar información adicional no detallada en la memoria.

2 | Comunicaciones marítimas AIS y VDES

2.1. Evolución de las comunicaciones marítimas digitales

La radio marítima fue la primera aplicación comercial de la tecnología radio, que permitía a los barcos mantenerse en contacto con la costa y otros barcos, así como enviar llamadas de socorro en situaciones de emergencia. Guglielmo Marconi inventó la comunicación por radio en la década de 1890, y su compañía Marconi instaló estaciones de telegrafía inalámbrica en los barcos a partir de 1900. Marconi construyó una serie de estaciones en tierra y en 1904 estableció la primera llamada de socorro en código Morse, las letras CQD (*Come Quick, Distress*), utilizadas hasta 1906 cuando se acordó el SOS. El primer rescate marítimo importante debido a la radio fue en 1909, con el hundimiento del transatlántico de lujo RMS Republic, en el que se salvaron 1.500 vidas. Éste y el rescate del RMS Titanic en 1912 hicieron que el campo de la radio marítima cobrara conciencia pública, y los operadores de radio marítima fueron considerados héroes. En 1920, Estados Unidos contaba con una cadena de 12 estaciones costeras que se extendían a lo largo de la costa atlántica desde Bar Harbor (Maine) hasta Cape May (Nueva Jersey).

La radio VHF marítima es un sistema mundial de transceptores de radio en barcos y embarcaciones que se utiliza para la comunicación de voz bidireccional de barco a barco, de barco a tierra (por ejemplo, con los guardacostas) y, en determinadas circunstancias, de barco a aeronave. Utiliza canales FM en la banda de VHF en el rango de frecuencias entre 156 y 174 MHz, inclusive, designada por la ITU como la banda móvil marítima VHF. En algunos países se utilizan canales adicionales, como los canales L y F para embarcaciones de recreo y de pesca en los países nórdicos (a 155,5 MHz – 155,825 MHz). La potencia de los transmisores está limitada a 25 watios, con un alcance sobre los 100 km o 54 millas náuticas ¹.

¹1 MN = 1,852 km

Las radios de VHF existen desde hace muchos años y siguen siendo el principal medio de comunicación para las embarcaciones. El uso de esta banda se justifica por el hecho de que las señales de baja frecuencia cubren mayores distancias en su propagación y son menos susceptibles a sufrir diversos tipos de interferencias. Al emplear frecuencias mayores que la banda HF, evita la propagación ionosférica, limitando el alcance. Otro aspecto positivo de esta banda de frecuencia es su robustez, ya que los efectos meteorológicos no afectan tanto a la señal, a diferencia de la banda UHF. Además, VHF funciona mejor en situaciones en las que el coste de los equipos y el consumo de energía son una preocupación.

Los equipos de radio VHF marinos están instalados en todos los buques grandes y en la mayoría de las embarcaciones pequeñas. También se utiliza, con una regulación ligeramente diferente, en ríos y lagos. Estos equipos se utilizan para una amplia variedad de propósitos, incluyendo la navegación marítima y el control del tráfico, llamadas a los servicios de rescate y la comunicación con los puertos, esclusas, puentes y puertos deportivos.

La recomendación ITU M. 1084-5 [13] establece una serie de pautas para mejorar la eficiencia del uso de esta banda de comunicaciones, y se especifican las características técnicas para el uso de canales de banda estrecha. Dentro de las recomendaciones se recoge que las administraciones que tengan problemas de congestión pueden utilizar modulaciones de frecuencia analógicas a 12,5 KHz, teniendo en cuenta las repercusiones en la navegación marítima, que cuenta con separación de canales de 25 KHz. Si se emplea esta solución, se deben utilizar los medios apropiados para evitar la interferencia con los canales de socorro y seguridad, así como los canales de navegación marítima internacional. El mecanismo que se emplea es el entrelazado de canales de banda estrecha con el objetivo de reducir el número de canales de 25 KHz. Dentro de las especificaciones del receptor, se indica que debe tener un rechazo de canal adyacente de al menos 70 dB para aquellos separados en 12,5 KHz. Además, también recoge la asignación de los números de canales que se encuentra en el Anexo 4 de la misma recomendación.

Las soluciones recogidas en la recomendación son utilizadas por el sistema AIS, detallado en la recomendación ITU-R M.1371 [4]. Siguiendo la arquitectura OSI (*Open System Interconnection*) de 7 capas, en este documento se especifican las características de la capa física hasta la capa de transporte.

2.2. Aspectos generales de los sistemas AIS y VDES

El AIS es un sistema de comunicación de difusión que opera en la banda VHF asignada a los servicios móviles marítimos (156,025 MHz - 162,025 MHz), capaz de realizar comunicaciones entre barcos comprendidas entre las 20 y 40 millas náuticas o MN. Esta tecnología permite el intercambio de información marítima entre barcos, así como entre estos y las estaciones de control ubicadas en la costa. Las transmisiones son codificadas mediante NRZI (*Non Return to Zero Inverted*) y moduladas utilizando la modulación en fase GMSK (*Gaussian Minimum Shifting Keying*). AIS utiliza un protocolo abierto para el intercambio de datos de navegación, siendo los sistemas de acceso a los medios diferentes formas de TDMA (*Time-Division Multiple Access*).

La información intercambiada entre diferentes entidades se transmite a intervalos de 26,66 ms, lo que corresponde a 256 bits (intervalo de guarda + *flags* + datos) a una velocidad de 9600 bits por segundo. Este valor suele ser suficiente sabiendo que el tipo de información que se envía se corresponde con parámetros básicos como velocidad, posición, identidad o rumbo.

Una de las principales ventajas del AIS sobre la tecnología radar es su capacidad para establecer contacto con barcos sin línea de visión directa. Además, los datos recibidos no dan lugar a falsos positivos debido a la orografía costera o el estado del mar. Sin embargo, AIS todavía se considera una tecnología complementaria, por lo que no sustituye anteriores sistemas de comunicaciones como el Canal 16 dedicado a las llamadas y tráfico de socorro y la llamada selectiva digital o DSC situado en el canal 70.

Con el objetivo de mejorar la seguridad en el transporte marítimo, la IMO ha establecido ciertos requisitos, relacionados con el tipo de arqueo y tonelaje, para que todos los buques que cumplan con los requisitos instalen AIS de forma obligatoria [1]. Estos requisitos son los siguientes:

- Buques con arqueo bruto superior a 500 toneladas.
- Buques con arqueo bruto superior a 300 toneladas en travesías internacionales.
- Todos los buques de pasajeros, independientemente de su tamaño.

Ya que el propio estándar del sistema menciona una clasificación entre AIS Clase A, con mayor rendimiento y costo, y AIS Clase B, con características y precios más modestos, los dispositivos de la Clase A son imprescindibles y se han de incorporar a naves que cumplan, al menos, uno de

los requisitos anteriores. Sin embargo, hay embarcaciones que no están sujetas a esta regulación, tales como barcos militares y barcos que prestan servicios no comerciales. Estos buques no están obligados a identificarse en el sistema, pero pueden implementarlo si lo ven necesario.

Por otro lado, el sistema AIS se puede integrar en diferentes dispositivos dependiendo de su aplicación. Por tanto, se establecen las siguientes clases de AIS:

- Estación base. Estaciones de control montadas en la costa que integran el sistema AIS para garantizar la cobertura de los barcos cercanos a puertos y operaciones de atraque.
- Ayuda a la navegación (AtoN, *Aid to Navigation*). Boyas de salvamento marítimo que prestan cobertura en zonas alejadas de la costa.
- Traspondedores de Búsqueda y Rescate (SART, *Search and Rescue Transponder*). Dispositivos AIS que se integran en una variedad de vehículos terrestres, marítimos y aéreos, con el fin de localizar embarcaciones de rescate y así poder servir como red de apoyo en las operaciones de salvamento.

A pesar de que el AIS surgió para mejorar la seguridad marítima, han surgido diversas aplicaciones que han contribuido a mejorar la eficiencia de la navegación. Dentro de estas aplicaciones se encuentran las siguientes [14]:

- *Minería de datos AIS*. La minería de datos es un proceso de extracción de información basado en datos brutos. Los datos AIS en bruto están compuestos por un grupo de puntos dispersos espacial y temporalmente, de los que sólo se puede obtener directamente una información limitada. Por lo tanto, la minería de datos es importante para los datos AIS y proporciona la base para la mayoría de los estudios basados en estos datos. Los métodos más comunes para procesar y extraer datos AIS incluyen la extracción, agrupación y la predicción de trayectorias.
 - *Seguridad en la navegación*. Se ha realizado un número considerable de estudios iniciales sobre cómo utilizar los datos del AIS para evitar accidentes marítimos. Estos estudios se han desarrollado principalmente en dos direcciones: evitar colisiones para los buques y mejorar la vigilancia del tráfico para las autoridades costeras. Los datos del AIS cuentan con dos principales focos de estudio en esta categoría, siendo la detección de anomalías y detección de tráfico marítimo.
-

- *Análisis del comportamiento de buques.* Las trayectorias de los buques incluyen información sobre su comportamiento. La siguiente revisión se lleva a cabo según tres categorías de comportamiento de los buques que se pueden resumir en actividades de pesca, navegación en aguas abiertas y el navegación en aguas restringidas.
- *Evaluación medioambiental.* El transporte marítimo emite unos 1000 millones de toneladas de CO₂ al año, según las estimaciones de la IMO, y es responsable de aproximadamente el 2,5 % de las emisiones mundiales de gases de efecto invernadero [15]. Aparte de estas emisiones, el transporte marítimo también contribuye a otras amenazas medioambientales, como los vertidos de petróleo, el ruido oceánico y los daños a la biodiversidad marina.

Sin embargo, el AIS es un sistema que se puede utilizar en actividades de piratería, terrorismo y pesca ilegal debido a que muchas embarcaciones están exentas de tener que identificarse en el sistema. En este sentido, las prestaciones y garantías que ofrecen otras tecnologías, como el radar, superan claramente al AIS en términos de seguridad y confidencialidad.

Otro de los principales problemas es la alta densidad de tráfico marítimo, como puede darse en los puertos más transitados. Un informe de valoración sobre la funcionalidad del AIS del 2013 [16] evalúa el desempeño del sistema en las zonas con mayor saturación. Los resultados del informe indican que los efectos de saturación del sistema dificulta a las estaciones poder encontrar *slots* disponibles por los que transmitir, reduciendo así la efectividad de las estaciones costeras AIS. Dentro de los lugares evaluados se encuentran cuatro zonas donde la densidad de mensajes AIS es elevada. Estos resultados se pueden ver en la tabla 2.1.

Tabla 2.1: Porcentajes de saturación de canal AIS en 2012.

Zona estudiada	Porcentaje de saturación de canal
Puerto de Shangai y Bohai Bay (China)	30 - 40 %
Norte del Golfo de México (EEUU)	64 %
Zona costera de Tokio (Japón)	27.4 - 38 %
Zona costera de Busán (Corea del Sur)	40.73 %

El informe considera una zona en peligro de saturación si su porcentaje excede el 50 % de la ocupación del canal, que en este caso, solo es el norte del Golfo de México. No obstante, el resto de zonas pueden llegar a padecer estos efectos. La solución a este problema reside en el aumento de canales comunicación para el sistema AIS y ese es el punto de partida del sistema VDES [8].

VDES es un sistema que integra AIS y cuyo objetivo es solventar los problemas de saturación

señalados previamente. Para ello, integra cuatro servicios adicionales que se caracterizan por su número de canales, esquemas de modulación y codificación y canalización. Cada nuevo servicio realiza una función diferente:

- LR-AIS (*Long Range AIS*). Servicio empleado para habilitar comunicaciones AIS satelitales sin modificar la operabilidad de los canales AIS terrestres.
- ASM (*Application Specific Messages*). Servicio destinado al envío de mensajes no prioritarios para informar de eventos o datos de interés, como accidentes, tormentas o estado del mar.
- VDE-TER (*VHF Data Exchange for Terrestrial Links*). Servicio utilizado para enviar mensajes no prioritarios con altas tasas de transmisión en sistemas terrestres como buques y estaciones costeras.
- VDE-SAT (*VHF Data Exchange for Satellite Links*). Similar a VDE-TER pero para sistemas satelitales.

La figura 2.1 muestra la asignación de canales para los diferentes servicios que componen este nuevo sistema:

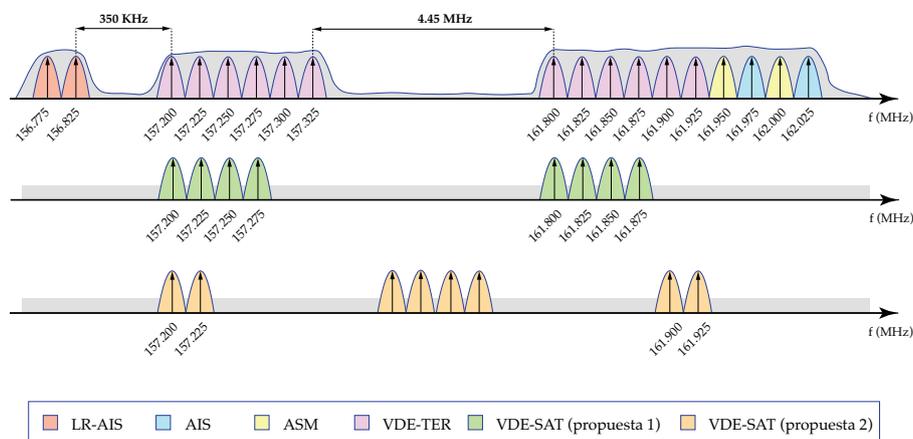


Figura 2.1: Plan de frecuencias para el sistema VDES. Fuente [8].

Por último, con la integración de comunicaciones por satélites, se pueden diferenciar tres segmentos: el *segmento marítimo* que está constituido por buques, dispositivos AtoN y aeronaves de salvamento; el *segmento costero* constituido por estaciones transmisoras y receptoras en la costa, y el *segmento espacial* constituido por satélites y estaciones que integren dispositivos VDES. Sin embargo, no todos los segmentos pueden interactuar entre sí, como indican las vías de comunicación entre segmentos en el diagrama de la figura 2.2.

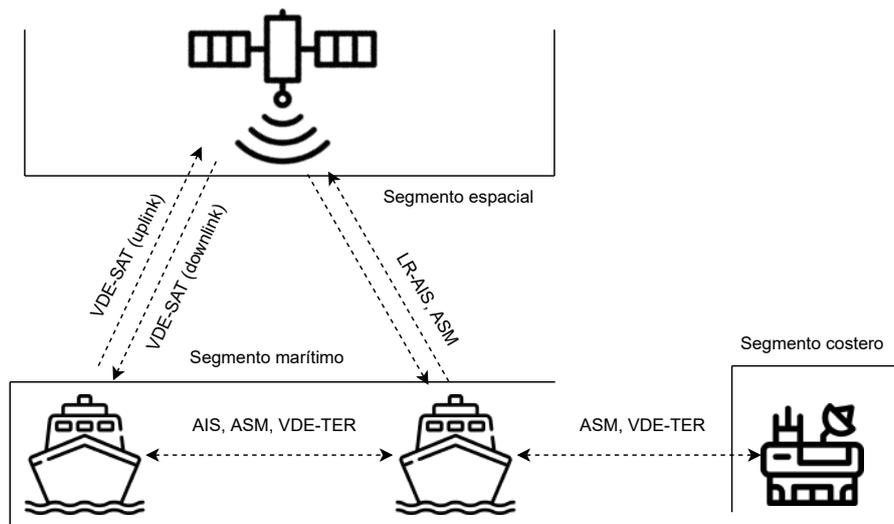


Figura 2.2: Comunicación entre segmentos del sistema VDES.

No obstante, el desarrollo y despliegue de dispositivos VDES no es un proceso inmediato, ya que requiere de tiempo para que los buques se adapten a este nuevo sistema. Además, otro de los motivos por lo que esta transición tomará tiempo es debido al hardware necesario para poder implementar este sistema. No solo el tiempo de prototipado de los dispositivos, sino su elevado coste al tener que implementar una capa física bastante más compleja.

A continuación, se presentan algunas particularidades de los nuevos servicios que incluye el VDES, con excepción del VDE-SAT.

LR-AIS

Este servicio formaba parte del sistema AIS como un tipo de mensaje definido en la norma, el 27. Consta de dos canales de 25 KHz en las frecuencias 156.775 MHz (LR-AIS-1) y en 156.825 MHz (LR-AIS-2.)

ASM

Este servicio formaba parte del sistema AIS como varios tipos de mensajes definidos en la norma, siendo los mensajes 6, 7 y 8. Cuenta con dos canales de 25 KHz ubicados en 161.950 MHz (ASM-1) y en 162 MHz (ASM-2). Debido a la cercanía de ambos canales con los canales AIS-1 y AIS-2, es necesario utilizar máscaras de filtrados muy restrictivas para no interferir, lo que puede complicar

el diseño hardware.

VDE-TER

Este servicio está compuesto por 12 canales de ancho de banda variables, pudiendo ser de 25, 50 o 100 KHz. La tasa binaria máxima de este servicio es de 307.2 kbps (32 veces superior a la tasa binaria del AIS). Además, se pueden emplear tres esquemas de modulación diferentes: $\pi/4$ -QPSK (*Quadrature Phase-Shift Keying*), 8-PSK (*Phase-Shift Keying*) y 16-QAM (*Quadrature Amplitude Modulation*).

El principal aspecto de desarrollo del VDE-TER es el uso de tabloneros de anuncio o TBB (*Terrestrial Bulletin Board*), que consiste en informes transmitidos por las estaciones costeras para planificar las transmisiones en zonas portuarias eficientemente y evitar la saturación de canales. Empleando estos tabloneros se pueden establecer los parámetros más importantes para prestar servicio a la zona de control de la estación, como frecuencias de uso y zona de cobertura. Cada embarcación es capaz de reconocer cuando se encuentra en estas zonas de control mediante los anuncios. En caso de que un buque se encuentre dentro de esta zona de control, cada transmisión deberá pasar por la estación costera para autenticar su veracidad, para luego ser retransmitida por esta al resto de embarcaciones de su zona, mientras que en el caso de estar fuera de la zona de control, las comunicaciones entre buques serían directas. De esta forma, se aumenta la seguridad en las zonas con mayor densidad de tráfico marítimo.

2.3. Capa física y de enlace del sistema AIS

A continuación, se detallarán los aspectos claves de las dos primeras capas que se implementan en el sistema AIS, siendo la capa física y la capa de enlace. Según la recomendación, la arquitectura OSI del sistema AIS tiene una estructura como se indican en la figura 2.3.

La primera hace referencia a la interfaz de comunicaciones y canal del propio sistema, y se tienen en cuenta los esquemas de codificación, modulación y estructura de la trama. La segunda capa se encarga de llevar a cabo la transferencia de datos hasta la capa de red, y está subdividida en tres subcapas: la subcapa MAC (*Medium Access Control*), la subcapa DLS (*Data Link Service*) y la subcapa LME (*Link Management Entity*). La capa de red, junto a la de transporte, se encargan de darle el formato marcado en la norma a los datos, así como los mecanismos de envío y recepción

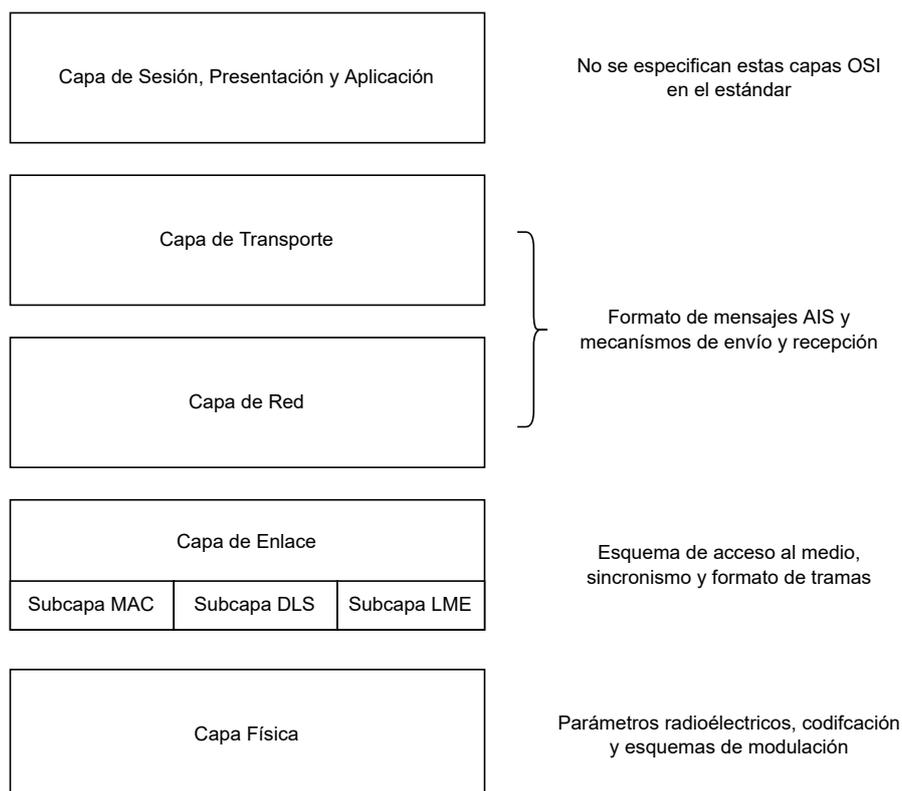


Figura 2.3: Arquitectura OSI del sistema AIS.

de mensajes.

2.3.1. Capa física del sistema AIS

El servicio AIS se estableció en la banda marina VHF, teniendo en esta banda dos posibles canales: AIS-1 (161.975 MHz) y AIS-2 (162.025 MHz), denominados canales 87 y 88. La canalización de la banda es de 25 KHz y está establecido en la recomendación ITU-R M.1084-5 [13].

En la recomendación ITU-R M.1371 [4] se especificaron los parámetros que han de cumplir las transmisiones de los canales AIS, recogidos en la tabla 2.2. Además, también se especifican diferentes parámetros especiales tanto en transmisión como en recepción, recogidos en las tablas 2.3 y 2.4.

Una vez definidos los parámetros que han de seguir los dispositivos transmisores y receptores, se explicarán los procesos de codificación y modulación que realiza esta capa.

Tabla 2.2: Parámetros radioeléctricos del sistema AIS.

Parámetros radioeléctricos	Valor
Banda de trabajo del servicio	156.025 – 162.025 MHz
Frecuencia del canal AIS-1	161.975 MHz
Frecuencia del canal AIS-2	162.025 MHz
Canalización de servicio	25 KHz
Tasa binaria	9600 baudios
Potencia transmitida en AIS clase A	12.5 W
Potencia transmitida en AIS clase B	1 W
Factor BT en transmisión	0.4
Factor BT en recepción	0.5
Índice de modulación	0.5

Tabla 2.3: Parámetros especiales en transmisión.

Parámetros radioeléctricos	Valor
Desviación de frecuencia de la portadora	± 500 Hz
Desviación de potencia de la portadora	± 1.5 dB
Respuesta de rechazo para emisiones espúreas	-36 dBm (9 KHz - 1 GHz) -30 dBm (1 GHz - 4 GHz)
Atenuación de intermodulación	>40 dB

Tabla 2.4: Parámetros especiales en recepción.

Parámetros radioeléctricos	Valor
Sensibilidad	-107 dBm
Tasa de error por saturación	1 %
Selectividad para canales adyacentes	70 dB
Selectividad cocanal	10 dB
Atenuación sobre emisiones espúreas	>70 dB
Atenuación de intermodulación	>74 dB

Esquema de codificación

El estándar AIS incluye un tipo específico de codificación diferencial, que es un conjunto de codificación ampliamente utilizado en comunicaciones radio y satelitales, usado para capturar bits con precisión cuando una señal se modula digitalmente.

Estas técnicas permiten que los datos se transmitan no solo en función del bit actual, sino también del bit anterior que se transmitió. De esta forma, se tiene un mejor control sobre el flujo de datos que se envía, aunque cabe señalar que es tan sensible a los llamados *errores de multiplicación* que la identificación errónea de un solo símbolo conduce a la generación de dos símbolos erróneos y, por lo tanto, duplica la tasa de error de bits (BER, *Bit Error Rate*).

La codificación diferencial utilizada por el estándar AIS se llama NRZI. Esta codificación mantiene constante el nivel de voltaje durante un bit, de modo que los datos se codifican por la presencia o ausencia de transiciones de señal al comienzo del período del bit. Cuando hay un 1 lógico, se codifica manteniendo el valor anterior, por lo que si anteriormente se ha tenido un 0, este valor se mantiene igual al comienzo del período de bit. Por el contrario, el 0 lógico se representa como una transición, por lo que se codificará con el valor opuesto al anterior. De este modo si anteriormente se ha codificado un 0, el valor actual pasará a 1. En la figura 2.4 se observa los cambios que presenta la señal original respecto a la codificada.

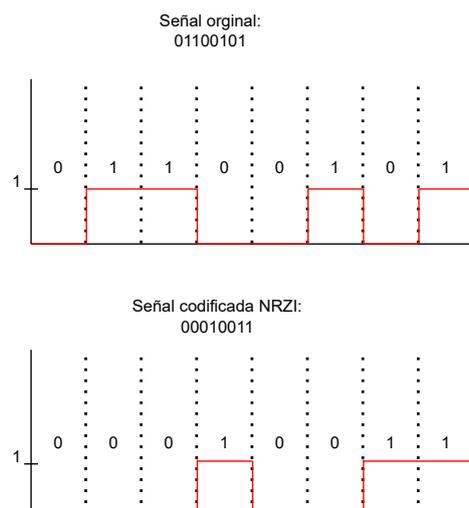


Figura 2.4: Ejemplo de codificación NRZI.

Las ventajas que ofrece esta codificación, y por lo que es elegida en el estándar AIS, son las siguientes:

- Precisión en la detección o ausencia de señal.
- Inmunidad frente a ruido e interferencias en el canal.
- Facilidad para detectar la transición entre símbolos.

Sin embargo, la principal desventaja que presenta esta codificación surge cuando existe una corriente continua en una larga secuencia de 0 lógicos, ya que perturba el sincronismo entre transmisor y receptor.

Esquema de modulación

El estándar AIS utiliza una modulación banda base denominada GMSK (*Gaussian Minimum-Shift Keying*). Esta modulación es un tipo particular de modulación MSK (*Minimum-Shift Keying*) en banda base que emplea un filtrado previo sobre los datos para reducir el ancho de banda de la señal. Así se consigue eliminar las componentes relativas a los lóbulos secundarios del espectro [17]. En la figura 2.5 se muestra la diferencia espectral de las modulaciones nombradas previamente. Se puede apreciar cómo la modulación GMSK consiste de un único lóbulo debido al filtrado que realiza.

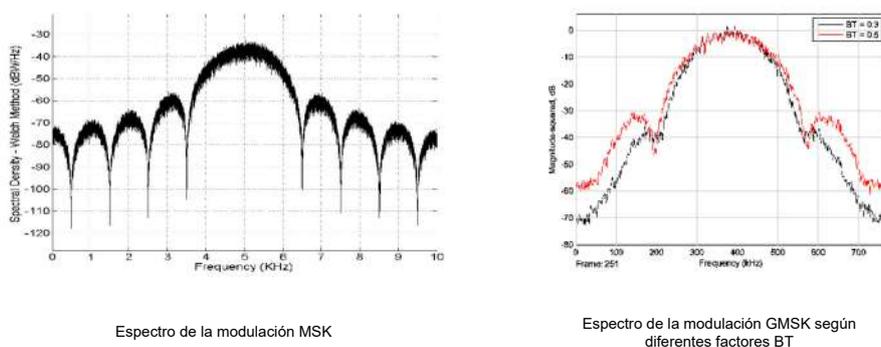


Figura 2.5: Espectro de las modulaciones MSK y GMSK.

La constelación de este esquema de modulación consiste en un símbolo que se desplaza trazando una circunferencia, como se puede ver en la figura 2.6.

Como anteriormente se comentó, esta modulación emplea un filtro a la señal de datos. Este filtro es un filtro tipo gaussiano, similar a un filtro paso-bajo con una respuesta al impulso que transforma una delta de Dirac a la entrada en una forma de onda similar a una curva de Gauss. La respuesta al impulso del filtro gaussiano se expresa en la ecuación 2.1

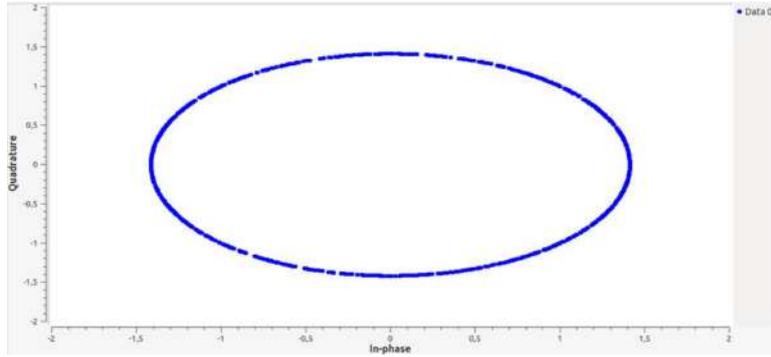


Figura 2.6: Constelación de la modulación GMSK.

$$g(t) = \frac{1}{2} \cdot \left[Q \cdot \left(-\frac{2\pi B}{\sqrt{2 \cdot \ln(2)}} \cdot t \right) - Q \left(-\frac{2\pi B}{\sqrt{2 \cdot \ln(2)}} \cdot (t - T_B) \right) \right] \quad (2.1)$$

donde B representa el ancho de banda a -3 dB de la señal filtrada, T_B el periodo de bit y Q es la función de distribución de Gauss, que se expresa en la ecuación 2.2:

$$Q(t) = \int_t \frac{1}{\sqrt{2}} \cdot e^{-\frac{x^2}{2}} dx \quad (2.2)$$

Además, cabe destacar que uno de los parámetros más importantes de esta modulación es el factor BT. Este factor relaciona la frecuencia de corte del filtro a -3 dB y la tasa binaria de la señal, representada en la ecuación 2.3.

$$BT = \frac{f_{-3dB}}{R_B} \quad (2.3)$$

Cuando se emplea esta modulación hay dos valores de factor BT que son muy comunes, $BT = 0.3$ y $BT = 0.5$. La diferencia de uso de estos valores se puede observar en la figura 2.7.

Como se puede observar en la figura 2.7, utilizando un factor $BT = 0.5$ la respuesta del filtro se extiende a 2 periodos de bits. Sin embargo, empleando un factor $BT = 0.3$ la respuesta se extiende a los 3 periodos de bits. Por lo tanto, la diferencia entre ambos factores BT se encuentra en un fenómeno conocido como interferencias entre símbolos o ISI (*Inter-Symbol Interference*). Para una misma frecuencia de corte, un factor BT menor permite una mayor tasa binaria a costa de mayor probabilidad de ISI. Con factores BT mayores sucede el caso inverso, debido a que la tasa binaria y el factor BT son inversamente proporcionales como se ve en la expresión 2.3.

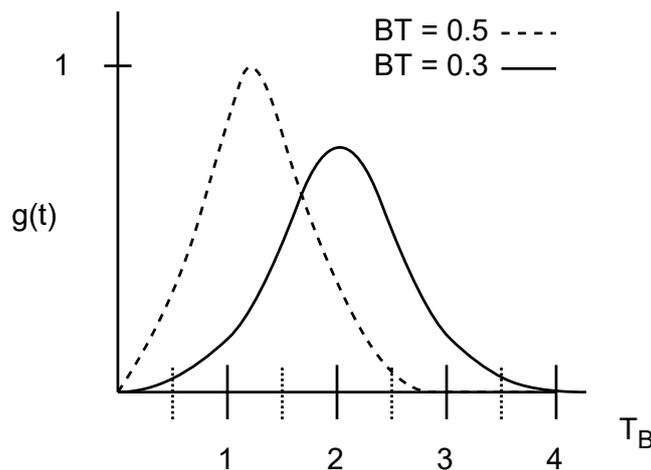


Figura 2.7: Respuesta al impulso de un filtro gaussiano con factor $BT = 0.3$ y $BT = 0.5$.

2.3.2. Capa de enlace del sistema AIS

La capa de enlace de cualquier sistema tiene como objetivo realizar la transferencia fiable de datos a la capa de red. Para ello, esta capa se encarga de la detección y corrección de errores en las tramas. Como se vio en la figura 2.3, la capa de enlace cuenta con tres subcapas, siendo estas la subcapa MAC, DLS y LME. Estas subcapas se explicarán a continuación.

Subcapa MAC

El objetivo de la subcapa MAC es garantizar que la comunicación entre las capas físicas de transmisor y receptor sea fiable. Como solución, AIS incluye referencias temporales comunes entre las estaciones del AIS que forman la red. Estas estaciones deben estar sincronizadas bajo la referencia temporal UTC (*Universal Time Coordinated*), recibida mediante satélites GNSS (*Global Navigation Satellite System*). Por parte de las estaciones, existen diversas maneras de acceder al sincronismo UTC, pero destacan las siguientes:

- *Sincronismo UTC directo.* El sincronismo deriva de la red GNSS, por lo que el transpondedor AIS tiene que incluir un receptor GNSS. De esta forma, es capaz de adquirir la referencia temporal y sincronizarse con el resto de estaciones.
- *Sincronismo UTC indirecto.* El transpondedor toma la referencia UTC de otras estaciones sincronizadas mediante la red GPS (*Global Positioning System*).

La sincronización entre estaciones AIS permite la actualización de los informe de posición. Esto significa que dependiendo de la condición del barco, y más específicamente de su velocidad de viaje, se envía un mayor número de información de posición cuanto mayor sea la velocidad del buque.

Subcapa DLS

La subcapa DLS se encarga de aplicar un formato específico a las tramas AIS. Cada trama esta compuesta por 32 bytes, repartidos por los siguientes campos:

- *Preámbulo*. Una cadena de 24 bits que corresponde a una secuencia de 1 y 0 lógicos alternados (010101010101010101010101) que en hexadecimal se representa como 3 bytes 0x55.
 - *Flag de inicio*. Una secuencia de 8 bits que se emplea como *flag* para alertar del inicio de los datos transmitidos. Se corresponde de forma binaria con la secuencia 01111110, que en hexadecimal se representa como un byte 0x7E.
 - *Datos*. También conocido como *payload*, es un campo de 168 bits donde se incluye la información que conforma cada paquete AIS. El contenido de este campo se define en la capa de red. En el caso de que un paquete tenga un campo de datos superior a 168 bits, se utilizan técnicas de segmentación de paquetes.
 - *Código de redundancia cíclica (CRC)*. Un polinomio generador de 16 bits que usa un código CRC definido en la norma ISO/CEI 3309:1993 con el que se realiza el control de errores de la trama.
 - *Flag de fin*. Similar al *flag* de inicio, se utiliza para señalar el final del campo de datos. Cuenta con la misma secuencia que el *flag* de inicio, 01111110 o 0x7E.
 - *Buffer temporal*. Una secuencia adicional de 24 bits que se utiliza para las siguientes tareas:
 - Relleno de bits (4 bits).
 - Retardo por distancia (12 bits)
 - Retardo del repetidor (4 bits)
 - Efectos por jitter (4 bits)
-

Subcapa LME

La subcapa LME se encarga de implementar el esquema de acceso al medio TDMA. Así, cada estación AIS tiene un intervalo de tiempo o *slot* para acceder al medio compartido con el resto de estaciones y realizar su transmisión. Cuando se agota el tiempo asignado, la estación desocupa el *slot*, permitiendo a otra estación acceder al medio.

Comúnmente, los esquemas TDMA requieren que las estaciones estén sincronizadas, y para ello, AIS utiliza el sincronismo UTC previamente explicado. En AIS, cada *slot* dura 26.67 milisegundos. Por lo que en un minuto, se puede calcular que el número de *slots* disponibles es de $N = 60 \cdot \Delta t_{slot} = 60 \cdot 26,67 \approx 2250 \text{ slots}$. AIS incorpora diferentes tipos de TDMA, siendo el SOTDMA (*Self Organized TDMA*) y CSTDMA (*Carrier Sense TDMA*) los más utilizados en AIS clase A y AIS clase B respectivamente. En la figura 2.8 se pueden ver los diferentes tipos de esquemas de acceso al medio que emplea AIS y que se explicarán brevemente.

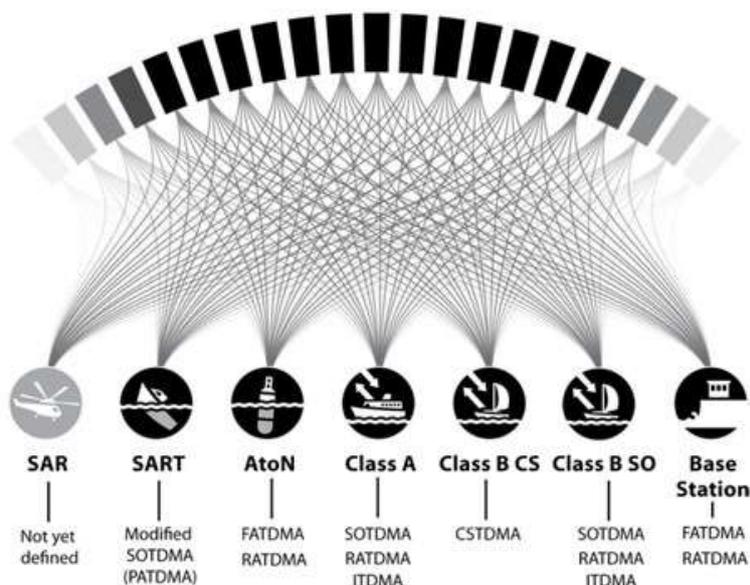


Figura 2.8: Esquemas TDMA utilizados en AIS. Fuente [18].

SOTDMA

Es el esquema que caracteriza a los sistemas AIS e integrado en los traspondedores de clase A. Este esquema cuenta con tres modos de operación:

- *Modo autónomo y continuo.* Modo que emplean las estaciones continuamente para realizar transmisiones y recepciones AIS.

- *Modo asignado.* Modo programado por una autoridad competente, configurado según los datos del área de operación.
- *Modo interrogación.* Modo que solo se ejecuta cuando una estación solicita una respuesta por parte de otra estación sobre un determinado evento.

El modo autónomo comienza con un periodo de escucha de 60 segundos para monitorizar ambos canales AIS y así estimar la actividad de los canales y las estaciones circundantes. Después de esta monitorización trata de elegir el primer *slot* disponible para realizar su primera transmisión. De esta forma, el resto de las estaciones serán capaces de conocer la incorporación de una nueva estación AIS y gestionar los *slots* de acceso. Tras transmitir su primer mensaje, la estación calcula el *slot* donde realizar sus futuras transmisiones continuamente. El modo autónomo se mantiene hasta que se apaga el dispositivo, pasa a modo asignado o modo interrogación o si cambia la tasa de actualización de informes.

CSTDMA

Este esquema se utiliza en los transpondedores AIS de clase B y ha de ser compatible con el esquema SOTDMA, priorizándolo sobre cualquier otro esquema. Emplea sincronismo UTC indirecto, por lo que es dependiente de otras estaciones cercanas con sincronismo UTC directo. Las estaciones con esquema CSTDMA reciben el ruido de fondo de los canales AIS continuamente y lo toman como referencia. De esta manera se puede medir el nivel de la señal al inicio de cada *slot*. En cada inicio de *slot* se mide el nivel de señal y según su nivel tomará una decisión:

- Si el nivel de señal es superior al ruido de referencia, significa que ese *slot* está en uso, por lo que no accede.
- Si el nivel de señal es inferior al nivel de ruido de referencia, significa que el *slot* está libre y trata de acceder.

La principal ventaja de este esquema es que su integración es más económica, pero cuenta con dos desventajas principales. La primera es la dependencia de otras estaciones con otro esquema TDMA y que de realizar una mala medida del nivel de ruido ocasionará interferencias y colisiones de paquetes, por lo que se suele implementar en receptores

RATDMA

El esquema RATDMA (*Random Access TDMA*) se utiliza para el anuncio de transmisiones aperiódica de *slots*. Las estaciones transmisoras utilizan un mapa de *slots* interno con el que es capaz de elegir un *slot* que no esta siendo ocupado. A diferencia de otros esquemas, no se anuncia el uso de ningun *slot*. Se suele utilizar por estaciones AIS clase A para anunciar su incorporación a la red. Sin embargo, no es aconsejable utilizarlo para transmisiones periódicas ya que no permite el reconocimiento de los *slots* usados a otros dispositivos al venir de un mapa interno, por lo que provocaría un gran número de colisiones de paquetes.

ITDMA

El esquema ITDMA (*Incremental TDMA*) se utiliza cuando una estación necesita asignar un *slot* pero no ha sido anunciada previamente. A diferencia de RATDMA, ITDMA anuncia que *slots* están siendo ocupados en todo momento. Así es posible que una estación anuncie un cambio temporal en las transmisiones periódicas, permitiendo el envío de mensajes aperiódicos. Se suele emplear como el RATDMA en la fase de entrada a la red, pero al requerir de otros esquemas no se usa como esquema independiente

FATDMA

El esquema FATDMA (*Fixed Access TDMA*) es gestionado manualmente para los dispositivos AIS configurados para utilizar los *slots* que se usan para el resto de transmisiones. Las estaciones con este esquema transmiten un mensaje DLM (*Data Link Management*) que informan al resto de estaciones que integran el mismo esquema sobre la asignación de *slots*, pudiendo bloquear los *slots* para que no sean ocupados por otras estaciones. Este esquema se utiliza exclusivamente en estaciones base AIS y en sistemas AtoN.

2.3.3. Tipos de mensajes AIS y capa de red

La información de los mensajes AIS depende de la tecnología del transpondedor que ofrece el servicio, pero se puede clasificar de forma general según:

- *Información estática.* Información referida a los parámetros propios de la embarcación que
-

permite identificarla de manera unívoca. Se engloba aquí el MMSI (*Maritime Mobile Service Identity*), el nombre, tipo, dimensiones y bandera de país de origen de la embarcación.

- *Información dinámica.* Se incluyen los aspectos variables de una travesía, crucial para el control de operaciones marítimas. Se incluye, así, la posición, velocidad, rumbo y estado de navegación de la embarcación.
- *Información sobre el viaje.* son aquellos datos que los usuarios pueden introducir manualmente acerca de la travesía, pero que no son monitorizados automáticamente por el sistema AIS.
- *Información de seguridad.* Mensajes cortos utilizados para garantizar la seguridad marítima o para informar de una situación de alerta.

El periodo de envío de mensajes depende de diversos factores. Cuando la embarcación está atracada o anclada envía información estática cada 3 minutos, independientemente de la clase AIS del dispositivo. Sin embargo, cuando la embarcación se desplaza durante la travesía o ejecuta maniobras, se actualizan sus datos dinámicos según su velocidad. En la tabla 2.5 se resumen la periodicidad de los mensajes según la velocidad tipo de movimiento y clase AIS.

Tabla 2.5: Periodos de transmisiones de mensajes AIS según la velocidad y maniobras.

Velocidad de la embarcación	Intervalo de mensajes
AIS Clase A	
Amarrado, anclado o inferior a 3 nudos ²	3 minutos
Amarrado o anclados pero con velocidad superior a 3 nudos	10 segundos
Entre 0 y 14 nudos	10 segundos
Entre 14 y 23 nudos	6 segundos
Entre 0 y 14 nudos con cambios de rumbo	3 segundos
Entre 14 y 23 nudos con cambios de rumbo	2 segundos
Superior a 23 nudos	2 segundos
Superior a 23 nudos con cambios de rumbo	2 segundos
AIS Clase B	
Inferior a 2 nudos	3 minutos
Superior a 2 nudos	30 segundos

Por último, AIS tiene definido 27 tipos de mensajes dependiendo de la operatividad del sistema y contiene los 3 posibles tipos de información definidos previamente. En la tabla 2.6 quedan recogido los diferentes mensajes establecidos.

²1 nudo = 0,514 m/s

Tabla 2.6: Listado de mensajes AIS y su función.

ID	Mensaje	Propósito
1-3	Informe de posición	Mensajes periódicos de posición de estaciones móviles Clase A
4	Informe de estación base	Mensajes periódicos de estaciones base que informa de posición y datos temporales
5	Datos estáticos y relativos al viaje	Mensajes empleados por equipos Clase A y aeronaves SAR para informar de datos estáticos o de viaje
6	Mensaje binario destinado	Mensaje dirigido a una estación en concreto para avisar de una determinada información
7	Asentimiento binario	Mensaje usado para asentir mensajes 6 y tienen que se enviados por el mismo canal por el que se recibió
8	Broadcast binario	Similar al 6 pero para transmisiones broadcast
9	Informe de aeronave SAR estándar	Mensaje usado por aeronaves en situaciones SAR, enviado cada 10 segundos
10	Petición de hora y fecha UTC	Mensaje usado cuando una estación solicita información temporal
11	Respuesta de hora y fecha UTC	Respuesta al mensaje 10 que comparte formato con el mensaje 4
12	Mensaje de seguridad destinado	Mensaje similar al 6, pero con envío de texto relativo a la seguridad
13	Asentimiento mensaje de seguridad	Mensaje similar al 7, responde a mensajes 12
14	Broadcast de seguridad	Similar al mensaje 12 pero para transmisiones broadcast
15	Interrogatorio	Mensaje para solicitar información distinta a la demandada por el mensaje 10. La respuesta se transmite por el mismo canal.
16	Comando para modo controlado	Mensaje para signar un patrón de transmisiones a otra estación AIS. El patrón se reserva previamente con el mensaje 20
17	Broadcast binario GNSS	Mensaje transmitido por estaciones base conectadas a una señal DGNS para un cálculo preciso de posición
18	Informe estándar de equipamiento Clase B	Mensaje similar a 1, 2 y 3, pero para dispositivos Clase B
19	Informe extendido de equipamiento Clase B	Mensaje de refuerzo para el mensaje 18, pero con un periodo mayor
20	Mensaje DLM	Mensaje utilizado por estaciones base que usan FATDMA
21	Informe AtoN	Mensaje usado por estaciones AtoN, informan de su funcionalidad
22	Administración de canal	Mensaje transmitido por una estación base para notificar los parámetros del enlace VHF asignados a una zona concreta
23	Comando de asignación grupal	Mensaje que permite que una estación base asigne un conjunto de parámetros de operación a una clase concreta de estaciones AIS
24	Informe de datos estáticos	Mensaje que permite asociar un MMSI con su nombre. Longitud máxima: 20 caracteres de 6 bits.
25	Mensaje binario un <i>slot</i>	Mensaje similar a 6 y 8, pero de longitud acotada a un <i>slot</i>
26	Mensaje de <i>slot</i> múltiple con estado de comunicación	Mensaje para planificar la transmisión de hasta 5 <i>slots</i> con datos binarios
27	Broadcast LR-AIS	Mensaje similar a 1, 2 y 3 pero para aplicaciones de largo alcance orientado a equipos Clase A

AIS emplea la codificación NMEA0183 [19]. Es un formato común para la transmisión de datos entre dispositivos electrónicos destinados a equipos marítimos que combina algunas especificaciones eléctricas con los datos recibidos por GNSS. Con esta codificación se permite la comunicación en tiempo real entre embarcaciones y estaciones marítimas. La comunicación se realiza a través del puerto serie de los equipos y se presentan mediante una ráfaga de datos AIVDM basados en ASCII de 6 bits y son codificados a través de los protocolos de red TCP (*Transmission Control Protocol*) o UDP (*User Datagram Protocol*) [20].

Un ejemplo de mensaje NMEA0183 y un análisis de sus diferentes campos se pueden ver en la tabla 2.7.

Tabla 2.7: Ejemplo de mensaje AIS codificado con NMEA0183.

Descripción del campo	Campo NMEA0183
Mensaje original:	
!AIVDM,1,1,,B,177KQJ5000G?tO'K>RA1wUbN0TKH,0*5C	
Tipo de mensaje	!AIVDM
Número de líneas del mensaje	1
Indicar de línea del mensaje	1
Espacio en blanco separador de mensajes multilinea	
Canal de transmisión (A = AIS-1, B= AIS-2)	B
Datos codificados	177KQJ5000G?tO'K>RA1wUbN0TKH
Fin de datos	0*
Cheksun NMEA	5C

2.4. Capa física y de enlace del sistema VDES

Las capas del modelo OSI del sistema VDES son muy similares respecto al AIS, ya que realizan las mismas funciones. La capa física es algo más compleja al tener que implementar diversos esquemas de modulación y tasas binarias. Las características técnicas de cada servicio, excepto el VDE-SAT se encuentran en la tabla 2.8. A continuación, se detallarán en la sección de capa física y de enlace de VDES las características técnicas de cada servicio reflejado en la tabla.

Tabla 2.8: Resumen de parámetros radioeléctricos de los servicios VDES.

Servicio	Número de canales	R_b (kbps)	BW_C (KHz)	Esquema de modulación
AIS	2	9.6	25	GMSK
LR-AIS	2	9.6	25	GMSK
ASM	2	19.2	25	$\pi/4$ -QPSK
VDE-TER	12	307.2	25, 50, 100	$\pi/4$ -QPSK, 8-PSK, 16-QAM

2.4.1. Capa física del sistema VDES

Debido a que VDES emplea diversos servicios, la capa física es bastante más compleja que en el caso del AIS, por lo que se dividirá la sección por cada servicio explicado previamente.

LR-AIS

Este servicio pertenecía al sistema AIS y estaba implementado como el mensaje 27, por lo que comparte gran parte de las características de la capa física de este sistema. Las frecuencias de los canales utilizados se ubican en 156.775 MHz (LR-AIS-1) y en 156.825 MHz (LR-AIS-2). La tasa binaria utilizada es de 9.6 kbps y emplea el esquema de modulación GMSK.

ASM

Aunque este servicio pertenecía al sistema AIS como el servicio anterior, éste presenta cambios significativos en su capa física. Las frecuencias de los canales se ubican en 162 MHz (ASM-1) y en 162.05 MHz (ASM-2). El primer cambio es su tasa binaria, siendo de 19.2 kbps, el doble que en el sistema AIS.

Su siguiente cambio significativo es el esquema de modulación empleado, ya que utiliza $\pi/4$ -QPSK. Similar a la modulación QPSK, se le añade un desfase de 45 grados o su equivalente, de $\pi/4$ radianes. Este esquema de modulación elimina los cambios de fase de $\pi/2$ y π radianes y están limitados a cambios de fase de $\pi/4$ radianes. Es capaz de mantener la propiedad de envolvente constante mejor que el esquema QPSK y se puede codificar diferencialmente, permitiendo el uso de técnicas de modulación tanto coherentes como no coherentes. Al utilizar un demodulador no coherente da a lugar un receptor más sencillo. En las figuras 2.9 y 2.10 se pueden ver la diferencia de espectro y constelación de los esquemas QPSK y $\pi/4$ -QPSK.

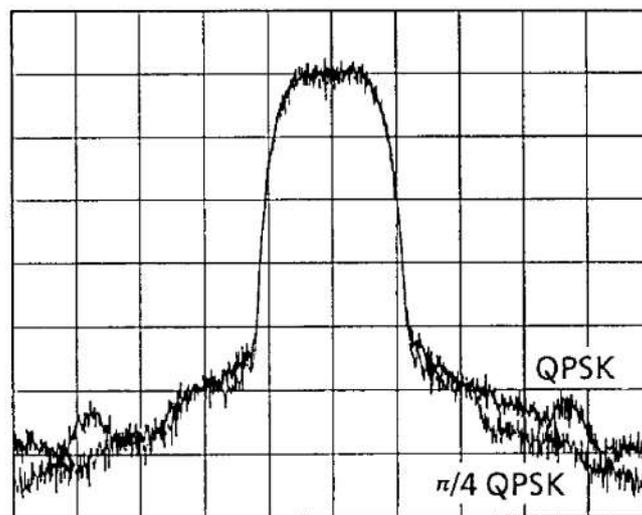


Figura 2.9: Espectros de la modulación QPSK y $\pi/4$ -QPSK.

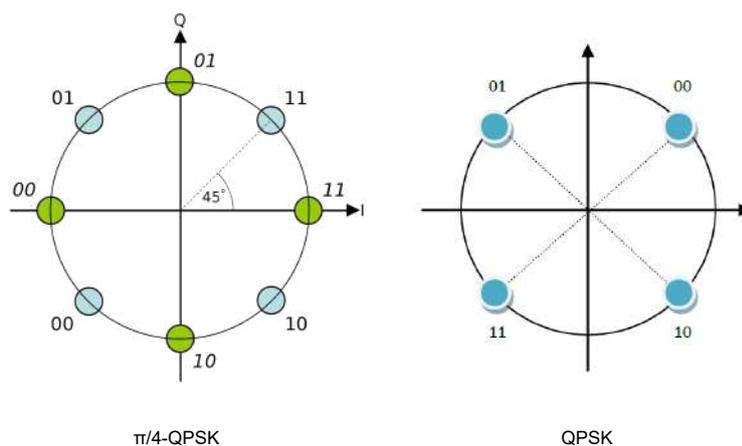


Figura 2.10: Constelaciones de las modulaciones $\pi/4$ -QPSK y QPSK.

En la figura 2.9 se puede observar que la modulación $\pi/4$ -QPSK presenta menores lóbulos secundarios a la modulación QPSK. En cuanto a las constelaciones, $\pi/4$ -QPSK emplea dos constelaciones en una, alternando cada símbolo entre la constelación de la modulación QPSK y la

desfasada $\pi/4$ radianes, siendo el orden verde \rightarrow azul \rightarrow verde o azul \rightarrow verde \rightarrow azul, dependiendo del momento que empiece. Las características que tienen que cumplir los equipos se recogen en las tablas 2.9 y 2.10.

Tabla 2.9: Requisitos del transmisor ASM.

Parámetros de transmisor	Requisitos
Error de potencia de portadora	+/- 1.5 dB
Error de frecuencia de portadora	+/- 500 Hz
Emisiones espúreas	-36 dBm: 9 KHz - 1 GHz -30 dBm: 1 GHz - 4 GHz

Tabla 2.10: Requisitos del receptor ASM.

Parámetros de receptor	Requisitos
Sensibilidad	-107 dBm
Selectividad de canal adyacente	70 dB
Rechazo de señales espúreas	70 dB
Rechazo de productos de intermodulación	74dB

VDE-TER

El servicio VDE-TER es el servicio con mayores cambios en la capa física, ya que implementa diferentes esquemas de modulación y tasas binarias. Cuenta con 12 canales de ancho de banda variables entre 25, 50, 100 KHz situados en las frecuencias de la figura 2.1. Su tasa binaria cambia según la modulación empleada, consiguiendo su máxima tasa al emplear 16-QAM en canales de 100 KHz, llegando a 307.2 kbps, 32 veces superior a la tasa del sistema AIS. Aparte de la modulación 16-QAM, VDE-TER emplea $\pi/4$ -QPSK y 8-PSK. Las características de la modulación $\pi/4$ -QPSK se explicaron en el servicio anterior, por lo que se analizaran las ventajas de las modulaciones 8-PSK y 16-QAM. La modulación 8-PSK emplea 8 símbolos, utilizando 3 bits por símbolo. Es capaz de transmitir datos más eficiente que otros tipos de modulación, es menos susceptible a errores y capaz de conseguir tasas binarias elevadas empleando modulaciones de orden mayores. Su espectro y constelación se puede ver en la figura 2.11.

La modulación 16-QAM emplea tanto variaciones de fase como de amplitud para la representación de datos. Esta modulación emplea 16 símbolos, con 4 bits por símbolo. La principal ventaja de esta modulación es su eficiente uso del ancho de banda al poder representar un mayor número de bits por portadora. Su espectro y constelación se puede ver en la figura 2.12

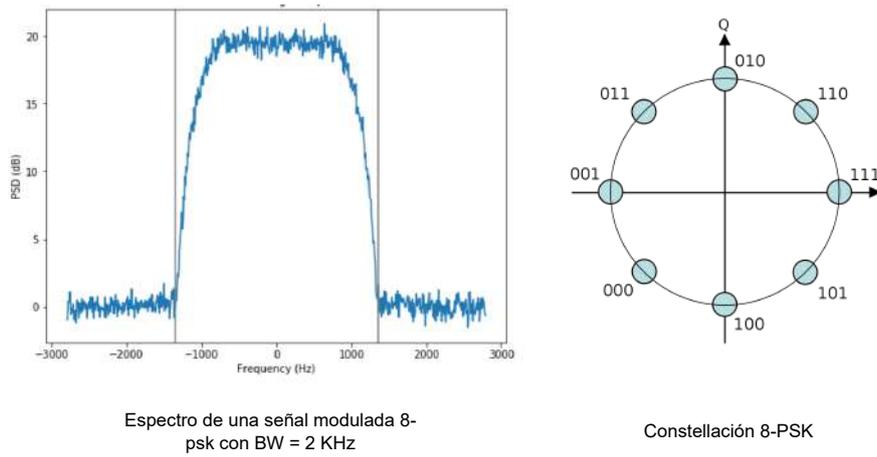


Figura 2.11: Espectro y constelación de la modulación 8-PSK.

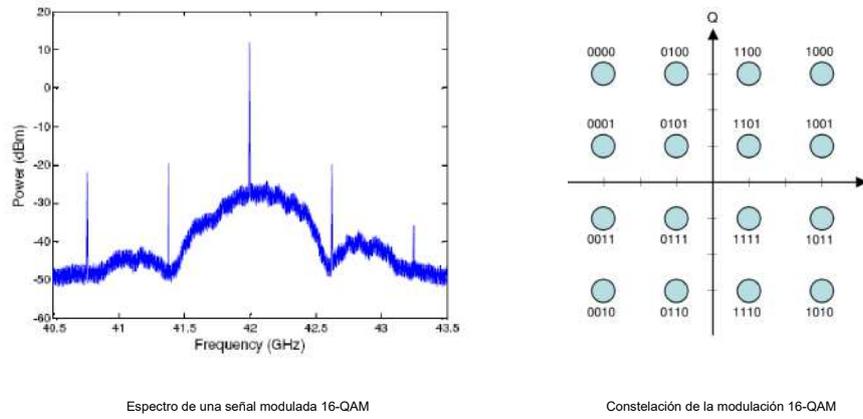


Figura 2.12: Espectro y constelación de la modulación 16-QAM.

Por último, las características de los transmisores del servicio VDE-TER se recogen en la tabla 2.11.

Tabla 2.11: Requisitos de un transmisor VDE-TER.

Parámetros de transmisión	Requisitos	Condición
Error de frecuencia	3 ppm	Normal
Potencia de transmisión	12.5 vatios (mínimo) No debe exceder los 50 vatios ± 1.5 dB normal +2 / -6 extremo	
Espectro modulado canal 25 KHz	0 dBc -25 dBc -60 dBc	$\Delta_{fc} < \pm 12,5$ KHz $\pm 12,5$ KHz $< \Delta_{fc} < \pm 25$ KHz ± 25 KHz $< \Delta_{fc} < \pm 75$ KHz
Espectro modulado canal 50 KHz	0 dBc -25 dBc -60 dBc	$\Delta_{fc} < \pm 25$ KHz ± 25 KHz $< \Delta_{fc} < \pm 50$ KHz ± 50 KHz $< \Delta_{fc} < \pm 100$ KHz
Espectro modulado canal 100 KHz	0 dBc -25 dBc -60 dBc	$\Delta_{fc} < \pm 50$ KHz ± 50 KHz $< \Delta_{fc} < \pm 100$ KHz ± 100 KHz $< \Delta_{fc} < \pm 150$ KHz
Emisiones espúreas	-36 dBm -30 dBm	9 KHz - 1 GHz 1 GHz - 4 GHz

2.4.2. Capa de enlace del sistema VDES

Para la explicación de la capa de enlace del sistema VDES se procederá de igual manera que el punto anterior.

ASM

Ya que este servicio proviene del AIS, sus subcapas son muy similares, siendo la capa MAC igual que en AIS. La capa DLS es modificada ya que los campos de las tramas ASM son diferentes. En la tabla 2.12 se puede ver los campos de la trama junto a la longitud en bits y los valores que toman. Por último, la capa LME utiliza el esquema ITDMA para poder sincronizarse con las estaciones en la primera transmisión. Después de sincronizarse con las estaciones se emplean los esquemas ITDMA, RATDMA y FATDMA.

VDE-TER

La capa MAC es igual que en el sistema AIS, ya que tiene que ser capaz de sincronizarse con las transmisiones de este sistema. La trama VDE-TER cuenta con campos diferente y además, al permitir diferentes tasas binarias, las tramas también tienen tamaños variables dependiendo de dicha tasa. En la tabla 2.13 se pueden ver los campos de la trama, su longitud en símbolos y los

Tabla 2.12: Campos de las tramas ASM.

Campo de la trama	Tamaño y valor del campo
Ramp up	16 bits
Secuencia de entreno	27 bits (111111001101010000011001010)
Información de señal FEC	Codificación Hamming (7,4) 0000 - no codificado 0001 - codificación 1/2 0010 - codificación 3/4 0011 - codificación 5/6
Longitud de datos	10 bits (0110011100 por defecto)
Datos	380 bits sin codificar, varía según la codificación
CRC	32 bits sin codificar, varía según la codificación Solo el campo de datos y longitud de datos se incluyen en el CRC
Buffer	40 bits
Total	512 bits

valores asignados. La capa LME emplea tres esquemas TDMA diferentes, siendo estos FATDMA, RATDMA e ITDMA.

Tabla 2.13: Campos de las tramas VDE-TER.

Campo de la trama	Tamaño y valor del campo
Ramp up	4 símbolos
Secuencia de entreno	27 símbolos (111111001101010000011001010)
Identificador de enlace	16 símbolos
Datos	1 <i>slot</i> : 176 símbolos 2 <i>slots</i> : 432 símbolos 3 <i>slots</i> : 688 símbolos
CRC	16 símbolos y solo incluye el campo de datos
Bits finales FEC	6 símbolos
Ramp Down	4 símbolos
Tiempo de guarda	7 símbolos
Total	1 <i>slot</i> : 256 símbolos 2 <i>slots</i> : 512 símbolos 3 <i>slots</i> : 768 símbolos

3 | Fundamentos de sistemas de radio definida por software

3.1. Introducción a la radio definida por software

Una radio definida por software o SDR es un equipo de radiocomunicaciones que permite implementar gran parte de los subsistemas de comunicación a través de software [21]. Frente a los equipos de radio tradicionales, los SDR reducen las necesidades hardware al permitir integrar varios sistemas de comunicación sobre una misma plataforma, que se puede configurar mediante software. De esta forma, una misma plataforma puede operar en distintas bandas de frecuencia, aplicar diferentes esquemas de modulación y codificación e implementar los protocolos radio correspondientes (figura 3.1). Este concepto no debe confundirse con la radio controlada por software o SCR (*Software-Controlled Radio*), que se trata de equipos radio convencionales que ajustan ciertas etapas por software, como el nivel de ganancia o la frecuencia de sintonía.

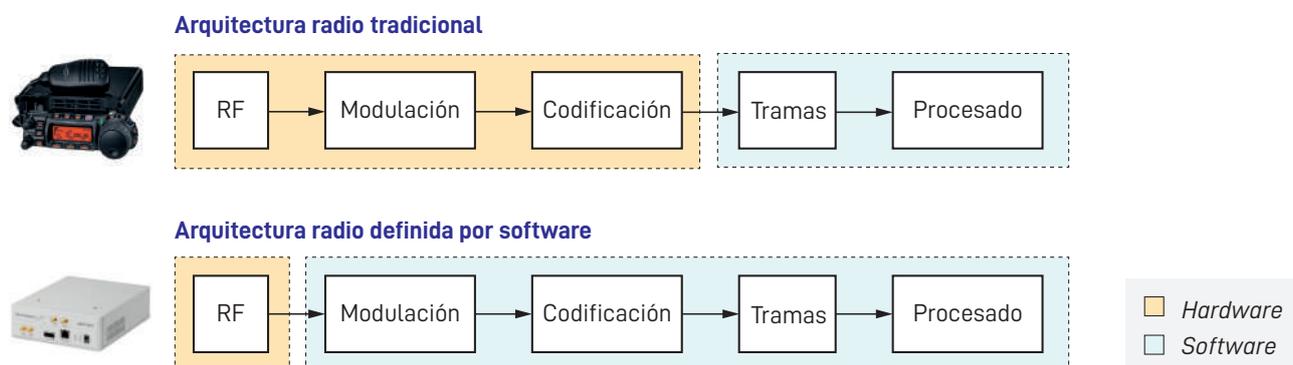


Figura 3.1: Diagrama de bloques de la arquitectura radio tradicional y la arquitectura SDR.

El origen de los SDR se remonta a 1991, cuando la Marina de Estados Unidos desarrolló Speakeasy [22]. Se trataba de un equipo programable que integraba hasta diez tipos de sistemas de comunicaciones, operando en un rango de frecuencias de 2 MHz hasta 200 MHz. Sin embar-

go, Speakeasy requería un hardware muy específico, y presentaba unas dimensiones elevadas que llegaban a ocupar el maletero de un coche, como se puede ver en la imagen 3.2 por lo que se apostó por mejorar el diseño con Speakeasy II. Con los avances de la tecnología electrónica y la programación modular, se consiguió reducir el tamaño del equipo y mejorar sus prestaciones radioeléctricas con respecto al primer modelo [23].



Figura 3.2: Primer equipo SDR Speakeasy.

Sin embargo, fue Joseph Mitola III quien propuso la primera arquitectura de receptor SDR en 1995 [24]. Su arquitectura se basaba en una antena conectada a una etapa de RF configurable, seguida de una etapa convertora analógica-digital o ADC (*Analog-to-Digital Converter*) y una unidad de procesamiento de las señales recibidas y convertidas en formato digital (figura 3.3). Cabe señalar que las arquitecturas SDR actuales no distan demasiado de esta primera aproximación, aunque haya aumentado en muchos casos su complejidad.

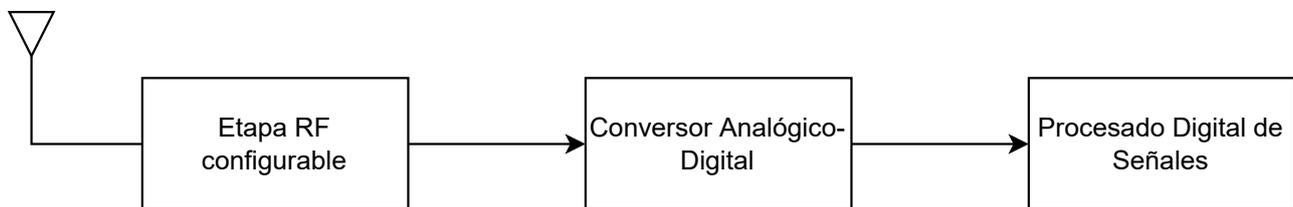


Figura 3.3: Primera arquitectura de radio definida por software propuesta por Joseph Mitola III.

De forma esquemática, y tomando como referencia la arquitectura SDR propuesta por Mitola III, un receptor SDR debe contener una etapa de RF que permita al equipo sintonizar las bandas de frecuencia que el usuario quiere analizar. Esta etapa suele seguir un esquema superheterodino con un número mayor o menor de etapas de conversión a frecuencia intermedia (FI). A esta etapa

le sigue una etapa ADC, que convierte las señales analógicas en un número mayor o menor de muestras, en función de las prestaciones del equipo y las necesidades de la aplicación. Por último, las señales muestreadas son tratadas mediante técnicas de procesamiento de la señal que se implementan como algoritmos en una unidad de procesamiento, ya sea un procesador de propósito general, una FPGA (*Field Programmable Gate Array*) o un DSP (*Digital Signal Processor*). En el caso de un transmisor SDR, el flujo de trabajo es el opuesto al del receptor. Inicialmente, una unidad de procesamiento genera las señales digitales que se desean transmitir utilizando técnicas de procesamiento de la señal. Esta señal digital, conformada por muestras, pasa por una etapa DAC (*Digital-to-Analog Converter*) que interpola las muestras generadas para convertirlas en señal analógica. Mediante una etapa de RF, la señal analógica se traslada a la frecuencia de sintonía donde el usuario pretende realizar la transmisión.

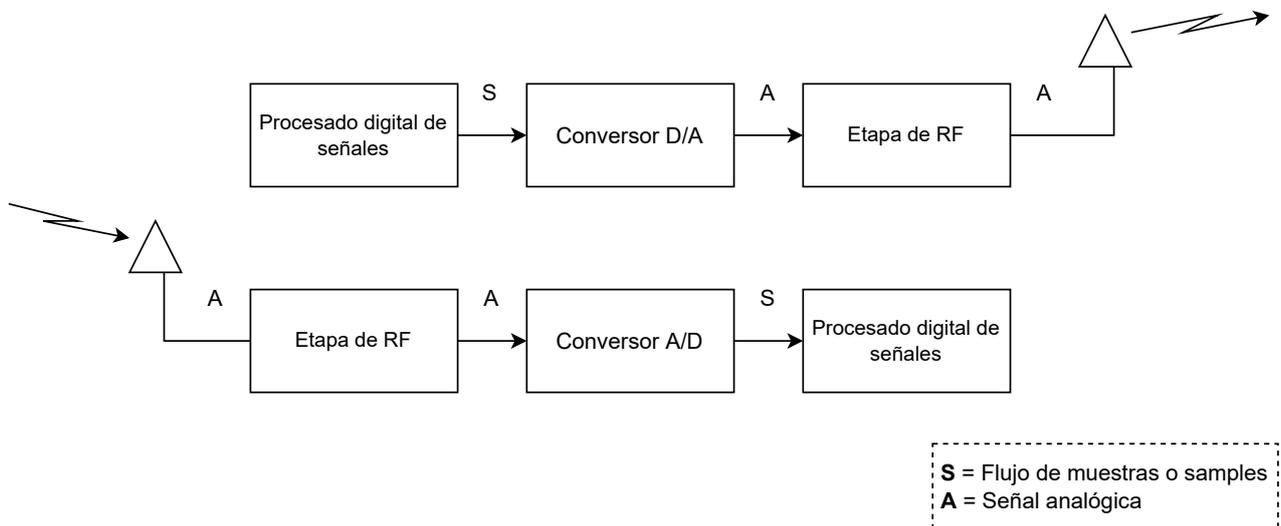


Figura 3.4: Arquitectura de un transceptor de radio definida por software.

Aunque los SDR son una alternativa que ofrece numerosas ventajas en términos de programabilidad, adaptabilidad y reducción de tiempos de desarrollo, cuentan con una serie de limitaciones. Una de ellas viene ligada a las etapas de conversión de señal, ya que los ADC y DAC son quienes fijan el máximo ancho de banda operable por el equipo. Otra es el consumo de potencia, que se incrementa notablemente con respecto a las arquitecturas radio tradicionales, principalmente por el uso de plataformas FPGA en muchos casos y al propio consumo de los ADC y DAC. No obstante, estas limitaciones no han mermado el interés por el desarrollo de sistemas de comunicaciones mediante SDR, y muchos equipos de radiocomunicaciones que se emplean en la actualidad han adoptado esta tecnología. Dentro de la División de Ingeniería de Comunicaciones del IDeTIC, se empezó con el uso de SDR [25] con la plataforma SDR-1000 en el año 2005.

En los siguientes puntos se tratará en detalle cada uno de los bloques que conforman un equipo transceptor SDR. También se llevará a cabo una revisión de algunas de las principales plataformas comerciales que existen en la actualidad.

3.2. Bloques de un transceptor basado en radio definida por software

En este apartado se explican los conceptos más importantes de los bloques que componen un transceptor SDR. Para un seguimiento gráfico de las explicaciones, se aporta en la figura 3.5 un diagrama de un transceptor SDR genérico, donde se diferencia una etapa receptora y otra transmisora con sus respectivos bloques fundamentales. Estas explicaciones permitirán definir en el siguiente punto las arquitecturas SDR más comunes.

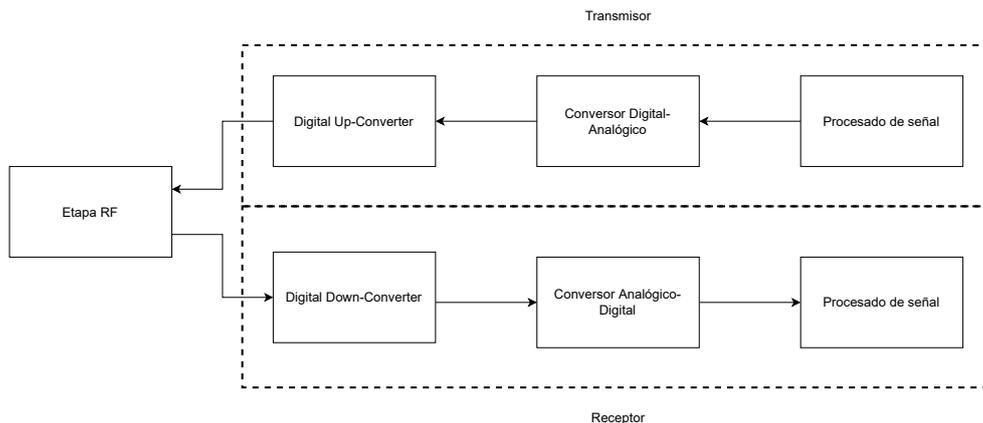


Figura 3.5: Diagrama de bloques de un transceptor basado en radio definida por software genérico.

3.2.1. Etapa RF

La etapa de RF se encarga de acondicionar las señales que son recibidas o transmitidas por el equipo. Es decir, en ella se adecuarán los niveles de señal a través de amplificadores, se eliminarán las componentes espectrales indeseadas mediante filtros y se ajustará la frecuencia de sintonía del equipo mediante osciladores locales y mezcladores. Para ello, existen diferentes topologías para implementar la etapa de RF de un receptor o un transmisor, siendo las más habituales la topología homodina y la superheterodina. Es común utilizar la topología homodina en transmisores y la superheterodina en receptores, por lo que este será el enfoque para explicar cada una de ellas en este apartado.

La topología superheterodina se caracteriza por realizar una conversión de frecuencia intermedia o FI en la cadena receptora. Tomando como referencia la figura 3.6, las señales recibidas por la antena pasan por un filtro paso-banda ajustado a la banda de sintonía de la aplicación a la que se destinan. Este filtro permite atenuar las componentes espectrales que se encuentran fuera de la banda de paso del filtro, mientras que aquellas componentes que se encuentran en la banda de paso son atenuadas levemente.

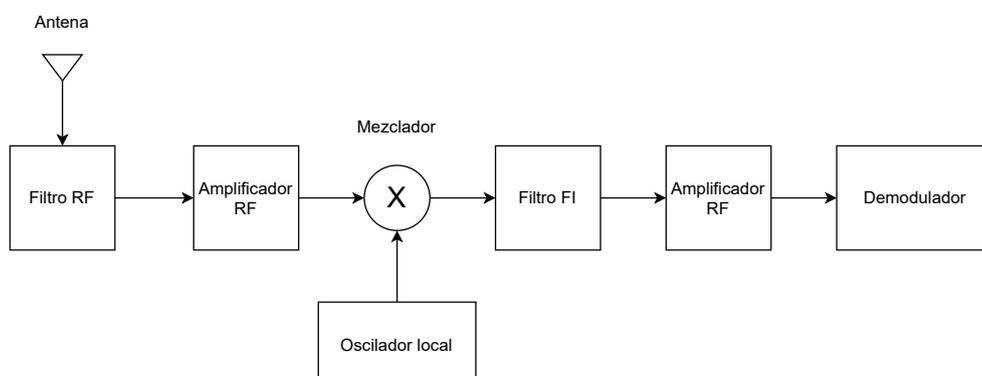


Figura 3.6: Arquitectura receptor heterodino.

Los amplificadores de RF son dispositivos capaces de incrementar a su salida el nivel de potencia de una señal recibida a su entrada. En el caso de los receptores, se suelen emplear LNA (*Low Noise Amplifier*), que por lo general incrementan entre 10 y 30 dB los niveles de señal de entrada y mantienen una figura de ruido lo suficientemente baja. En esta topología, pueden ubicarse tanto después del filtro como antenas, siendo esta última opción la que produciría una menor figura de ruido.

Cuando la señal recibida es filtrada y adecuada en ganancia, pasa por una etapa de mezcla. Esta

etapa está compuesta por un mezclador y un oscilador local. El oscilador genera un tono a una frecuencia F_{LO} que pasa por una de las entradas del mezclador, y al combinarse con la señal de entrada, con frecuencia F_{IN} , producirá a su salida una suma y resta de componentes espectrales, $F_{FI} = F_{IN} \pm F_{LO}$ (figura 3.7). Por ejemplo, si la frecuencia de la señal de entrada es de 250 MHz y el oscilador local se ajusta a una frecuencia de 180 MHz, a la salida del mezclador se obtendrá la señal de entrada en las frecuencias intermedias de 70 MHz y 430 MHz. En los receptores suele ser interesante reducir la frecuencia de las señales de entrada, por lo que la componente de interés (frecuencia intermedia) sería la de 70 MHz en este ejemplo.

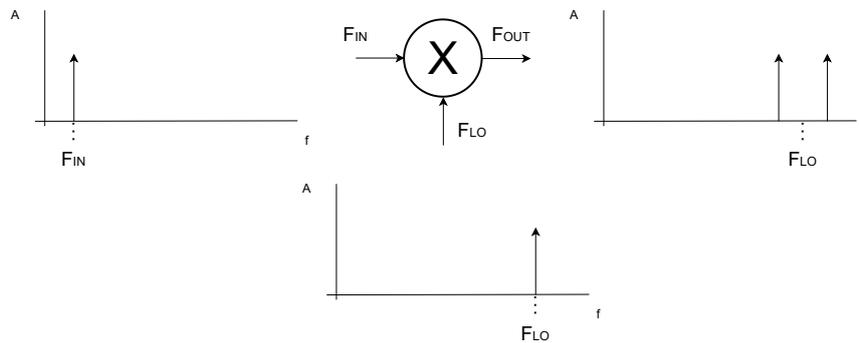


Figura 3.7: Funcionamiento de un mezclador ideal.

Durante el proceso de mezcla, además de la señal de entrada aparecen otras componentes no deseadas que se acoplan en esta etapa. Puede existir una componente a una frecuencia tal que, durante la mezcla, se ubique en la misma frecuencia intermedia que la señal de interés. A esta componente se le llama frecuencia imagen, y puede ser calculada mediante la ecuación 3.1. De hecho, una de las funciones del filtro de RF será garantizar una atenuación suficiente sobre la componente ubicada en la frecuencia imagen.

$$F_{IM} = |F_{IN} - 2 \cdot F_{OL}| \quad (3.1)$$

Por último, el filtro de FI debe garantizar el paso de los canales convertidos en frecuencia después del mezclador, que deben ser ajustados en ganancia mediante una nueva etapa de amplificación a la que le seguirá el proceso de demodulación.

La topología homodina, a diferencia de la heterodina, no realiza conversiones de frecuencia en transmisión. Siguiendo la figura 3.8 como referencia, la señal se genera a la frecuencia de transmisión con baja potencia para posteriormente modularla. Una vez modulada la señal, ésta se amplifica dependiendo de la modulación empleada (en el caso de modulación AM (*Amplitude*

Modulation) se amplifica de forma lineal, y en el caso de modulación FM (*Frequency Modulation*) de forma no lineal) y pasan por un filtro paso-banda ajustado a la frecuencia de transmisión con el objetivo de eliminar las componentes no deseadas como los productos de intermodulación. En el caso de recepción, esta topología se denomina conversión directa, debido a que traslada la señal a banda base en lugar de frecuencias intermedias.

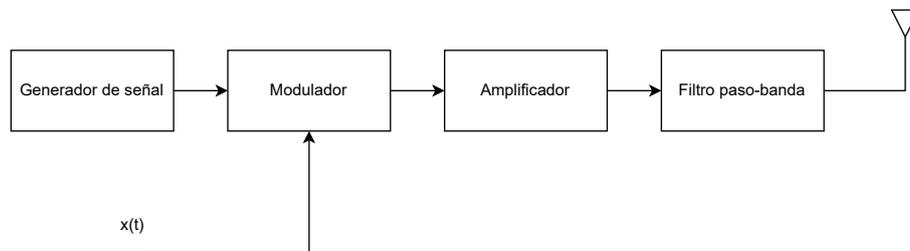


Figura 3.8: Arquitectura transmisor homodino.

Ya se ha comentado en el primer punto que los equipos SDR pueden implementar las funciones del demodulador mediante un bloque software, por lo que se puede considerar que la etapa de RF en un equipo receptor SDR se limita a los subsistemas que han sido explicados en este punto.

3.2.2. Etapa de conversión de señales

Los conversores de señal, tanto ADCs como DACs, son dispositivos críticos en los equipos SDR. En el caso de los ADC, el objetivo es convertir una señal analógica de entrada en una secuencia de muestras con las que puede trabajar el procesador digital de señales. En el caso de los DAC, el procesador digital de señales genera una secuencia de muestras que es convertida en una señal analógica para ser transmitida. En los equipos SDR, las prestaciones de los ADC y DAC suelen fijar la calidad de las plataformas, algo que se puede notar en las diferencias de precio entre las plataformas. En la figura 3.9 se representa un diagrama de bloques con las etapas que componen un ADC y un DAC.

En el caso de los ADC, se muestrea la señal analógica de entrada para convertirla en una secuencia de muestras. Teóricamente, esta secuencia de muestras sigue estando en el dominio temporal, por lo que se debería aplicar un conversor del dominio continuo al dominio discreto para mantener la consistencia del análisis matemático, aunque no será tratado en este trabajo. La frecuencia de muestreo debe ser elegida cuidadosamente para cumplir el teorema de Nyquist, que impone que la frecuencia de muestreo debe ser igual o superior al doble de la frecuencia máxima de la

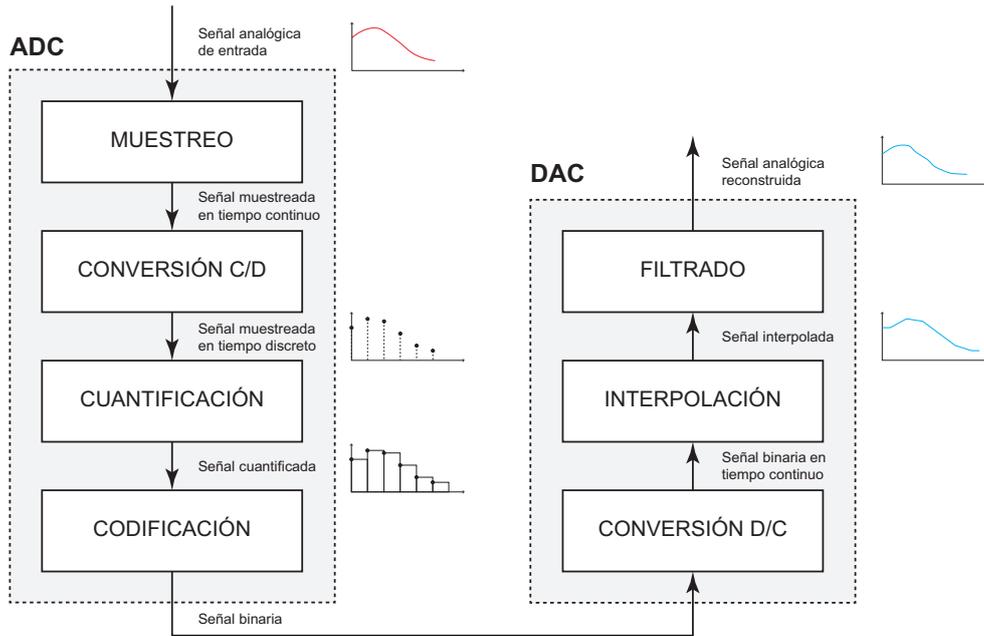


Figura 3.9: Diagrama de bloques de un transceptor basado en radio definida por software genérico.

señal de entrada (ecuación 3.2).

$$f_s \geq 2 \cdot f_{\text{señal}}, \quad (3.2)$$

Si no se cumple el teorema de Nyquist, se produce un fenómeno conocido como *aliasing*, donde los espectros de las señales muestreadas se solapan e impiden reconstruir la señal original, perdiendo la información de forma definitiva.

El siguiente subsistema lo forma un cuantificador, que asocia las muestras obtenidas con diferentes niveles de amplitud o de cuantificación. A su vez, estos niveles se asocian a un código binario único, que permite convertir niveles de cuantificación en bits. Los ADC pueden operar en un rango de valores de amplitud en la etapa de cuantificación. Por tanto, si V_{MAX} es el valor máximo de amplitud del rango de cuantificación y se emplean N bits, el nivel mínimo de señal con el que puede operar el conversor se puede obtener mediante la ecuación 3.3, lo que se conoce como *LSB (Least Significant Bit)*.

$$LSB = \frac{V_{ref}}{2^N}, \quad (3.3)$$

Además, la cuantificación es un proceso donde se pierde información debido a la interpolación, ya que no se pueden interpolar muestras y obtener exactamente la misma señal de origen. Por

ello, se habla de error de cuantificación, cuyo valor ideal no debe sobrepasar $\pm 1/2$ LSB. También es importante definir el rango dinámico de un ADC como la diferencia entre los niveles máximo y mínimo de señal que el dispositivo es capaz de medir, y que se relaciona con el número de bits de resolución y el LSB a través de la ecuación 3.4.

$$DR = 20\log_{10} \left(\frac{(2^N - 1) * LSB}{LSB} \right) = 6,02 \cdot N, \quad (3.4)$$

Esta expresión indica que si, por ejemplo, se tuviese un ADC de 12 bits, el rango dinámico del conversor sería de 72.24 dB.

También es necesario definir la relación señal a ruido o SNR (*Signal-to-Noise Rate*) como el nivel de señal útil recibida frente al ruido presente en el sistema, análogamente a su definición en otros ámbitos de las comunicaciones. La SNR en un ADC viene definida por la ecuación 3.5

$$SNR = 20\log_{10} \left(\frac{\text{rms} \cdot \sigma_S}{\text{rms} \cdot \sigma_Q} \right), \quad (3.5)$$

donde RMS (*Root Mean Square*) es el valor cuadrático medio, σ_S es la señal recibida en escala completa y σ_Q es el ruido de cuantificación. Asimismo, el valor cuadrático medio de la señal recibida se define mediante la ecuación 3.6

$$\sigma_S = \frac{LSB * 2^N}{2\sqrt{2}}, \quad (3.6)$$

donde N es el número de bits de resolución del ADC. También interesa conocer el valor cuadrático medio del ruido de cuantificación, que se define por la ecuación 3.7

$$\sigma_Q = \frac{LSB}{\sqrt{12}}, \quad (3.7)$$

Tomando las expresiones 3.5, 3.6 y 3.7, se obtiene una relación compacta de la SNR del ADC donde se muestra su dependencia con el número de bits de resolución, y queda expresada por la ecuación 3.8

$$SNR = 20\log_{10}(2^N) + 20\log_{10} \left(\sqrt{\frac{3}{2}} \right) = 6,02 * N + 1,76, \quad (3.8)$$

Esta expresión indicaría que si, por ejemplo, se tuviese un conversor ADC de 12 bits en un equipo SDR, la SNR sería de 74 dB frente al ruido.

Otro parámetro de interés en los ADC es el ENOB (*Effective Number Of Bits*), que se define como el número de bits que intervienen realmente en la conversión A/D cuando se presenta ruido de cuantificación y otras distorsiones. En la práctica, su valor es inferior al número de bits de resolución del ADC, por lo que un conversor de altas prestaciones presentará un ENOB con un valor elevado, próximo al número de bits de resolución indicados por el fabricante. Junto a este parámetro, la relación SINAD (*Signal-to-Noise-and-Distortion*) es un indicador de calidad de la conversión A/D, que se relaciona con la SNR mediante la ecuación 3.9

$$\text{SINAD} = -10\log(10^{-\text{SNR}/10} + 10^{\text{THD}/10}), \quad (3.9)$$

además del valor THD (*Total Harmonic Distortion*), que es la suma cuadrática desde el segundo al décimo armónico, y se expresa mediante la ecuación 3.10

$$\text{THD} = 10\log_{10} \left(\sum_{i=2}^{10} 10^{\frac{D_i}{10}} \right), \quad (3.10)$$

La relación entre el ENOB y el SINAD viene dada por la ecuación 3.11

$$\text{ENOB} = \frac{\text{SINAD} - 1,76}{6,02}. \quad (3.11)$$

En el caso del DAC, el primer subsistema consiste en una etapa de conversión de la señal codificada y cuantificada a niveles de tensión. Los bits de resolución del sistema define la precisión de estos valores y, por lo tanto, la exactitud de la señal reconstruida. Para conseguir una referencia temporal, el DAC requiere de una señal de reloj. Sin embargo, la señal en este punto aún es discreta, ya que solo se cuenta con valores de tensión, como se puede observar en la figura 3.10.

Para convertir la señal anterior en una señal continua se utilizan métodos de interpolación. Esta operación convolucionada dos señales en el tiempo, siendo estas la señal de valores de tensión discreta y un pulso rectangular de amplitud unidad. El resultado de este subsistema se representa en la figura 3.11. Sin embargo, para conseguir que la señal resultante de esta operación sea bastante parecida a una señal continua se necesita que el factor de interpolación sea bastante alto. De esta forma habrán más valores intermedios entre muestra y muestra y suaviza la señal resultante. Este

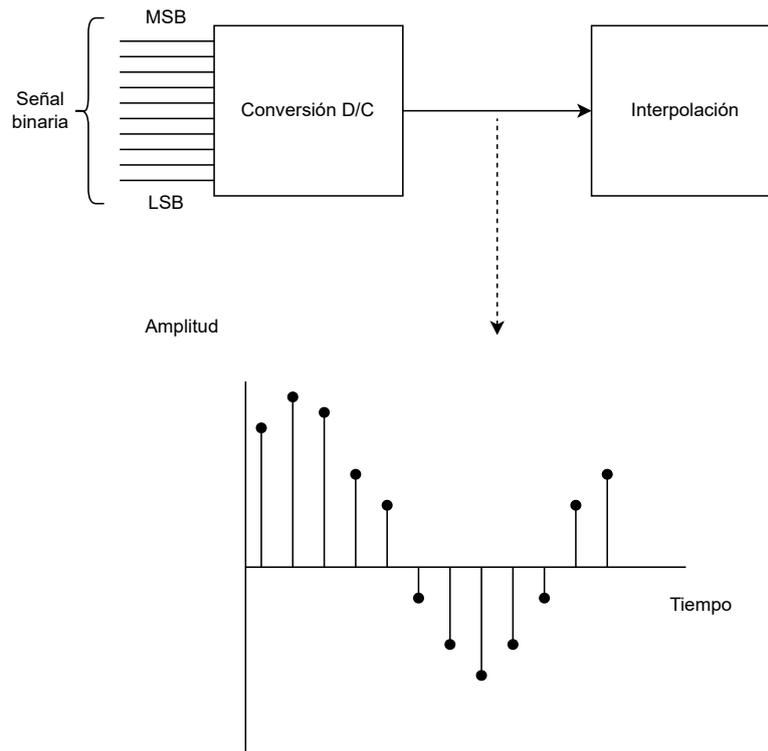


Figura 3.10: Etapa de conversión discreto-continuo.

proceso modificará la frecuencia de muestreo de forma directamente proporcional al índice de intermodulación utilizado quedando como $F_{S_o} = f_{S_i} * I$

Por último, el DAC realiza un filtrado de la señal para eliminar cualquier producto indeseado de las operaciones anteriores. Para ello, se emplea un filtro paso-bajo ya que todas las operaciones se realizan en banda base.

3.2.3. Etapa de conversión digital

En los SDR, tanto el DDC (*Digital Down Converter*) como el DUC (*Digital Up Converter*) son de los bloques más importantes en la cadena receptora y transmisora, respectivamente [26]. El DDC prepara la señal recibida para ser tratada digitalmente por el procesador digital de señales, reduciendo su tasa de muestreo mediante operaciones de diezmado y filtrado, mientras que el DUC aumenta la tasa de muestreo mediante operaciones de interpolación y filtrado, obteniendo una señal ajustada a la conversión D/A. En ambos casos, se emplean filtros paso-bajo que se denominan *multirate* [27]. En las figuras 3.12a y 3.12b se representan los diagramas de bloques correspondientes a un DDC y un DUC genéricos.

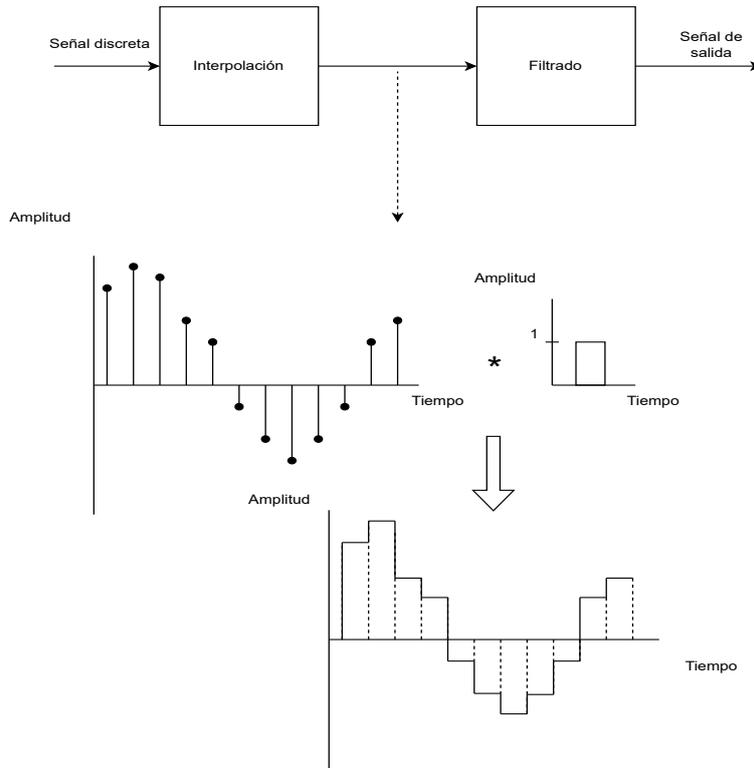
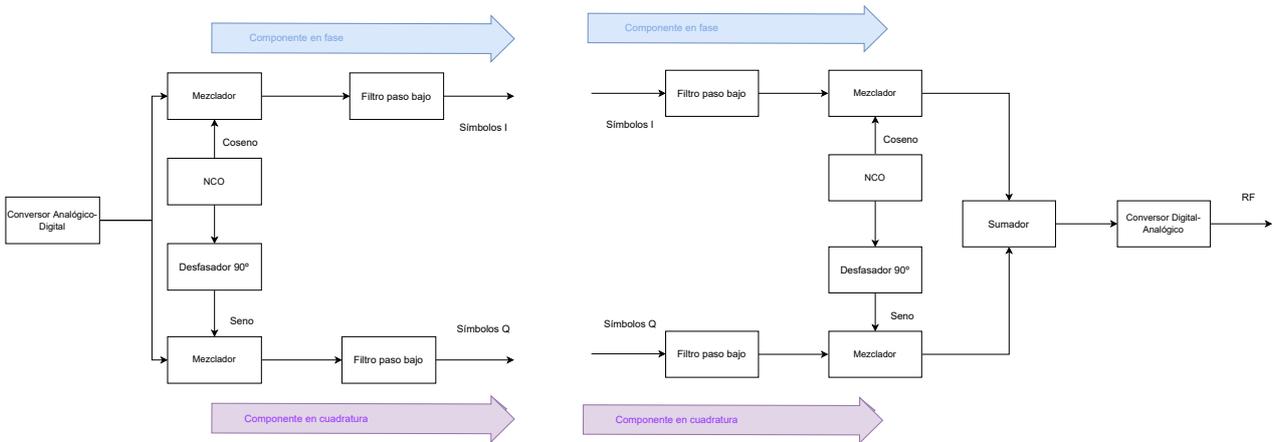


Figura 3.11: Etapa de interpolación.



(a) Diagrama de bloques de un DDC.

(b) Diagrama de bloques de un DUC.

Figura 3.12: Diagramas de bloques de un DDC/DUC.

Como se observa en la figura 3.7, los bloques DDC y DUC operan con señales analíticas que suelen recibir el nombre de señales IQ (*In-phase and Quadrature*). Estas señales resultan de la suma de una parte real llamada componente en fase, $x_I(t)$, y otra imaginaria llamada componente en cuadratura, $x_Q(t)$, de modo que la señal será compleja tendrá la forma $x(t) = x_I(t) + j x_Q(t)$. Ambas señales se encuentran desfasadas entre sí 90° , tal y como se representa en la figura 3.8.

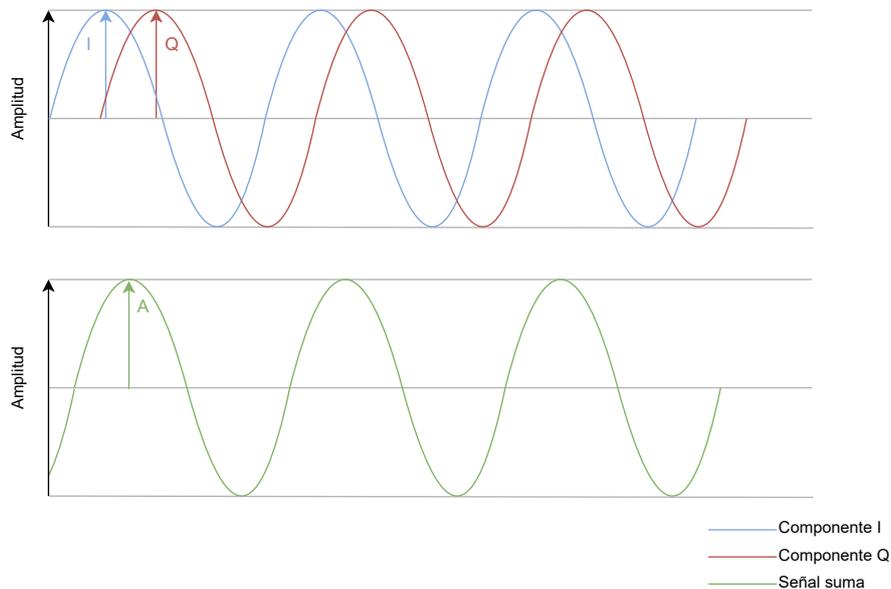


Figura 3.13: Componentes IQ y señal suma.

Tomando como referencia el diagrama de bloques del DDC de la figura 3.12a, se observa que la señal queda multiplicada, en cada rama, por un tono desfasado 90° . El NCO (*Numerically-Controlled Oscillator*) es un dispositivo que implementa un algoritmo optimizado para generar una señal digital controlada a su salida, que generalmente es un tono. No se entrará en detalles sobre sus características en este trabajo, por lo que, a efectos prácticos, se considerará que su funcionamiento es el mismo que el de un oscilador local. Esto hace que, a la salida del mezclador en cada rama, las señales en fase y cuadratura tomen la forma $x'_I(t) = x(t) \cdot \cos(\omega_0 t)$ y $x'_Q(t) = x(t) \cdot \sin(\omega_0 t)$, respectivamente, donde se asume que la señal recibida tiene forma senoidal, $x(t) = A \cdot \cos(\omega_0 t + \phi)$. Estas señales pasan por un filtro paso-bajo, por lo que las ecuaciones 3.12 y 3.13 representan la salida del DDC:

$$\begin{aligned}
x'_I(t) &= x(t) \cdot \cos(w_0t) = A \cdot \cos(w_0t + \phi) \cdot \cos(w_0t) = \\
&= \frac{A}{2} \cdot [\cos(w_0t + \phi + W_0t) + \cos(w_0t + \phi - w_0t)] = \\
&= \frac{A}{2} \cdot [\cos(2w_0t + \phi) + \cos(\phi)] \implies x_I(t) = LPF\{x'_I(t)\} = \cos(\phi)
\end{aligned} \tag{3.12}$$

$$\begin{aligned}
x'_Q(t) &= x(t) \cdot \sin(w_0t) = A \cdot \cos(w_0t + \phi) \cdot \sin(w_0t) = \\
&= \frac{A}{2} \cdot [\sin(w_0t + \phi + W_0t) + \sin(w_0t + \phi - w_0t)] = \\
&= \frac{A}{2} \cdot [\sin(2w_0t + \phi) + \sin(\phi)] \implies x_Q(t) = LPF\{x'_Q(t)\} = \sin(\phi)
\end{aligned} \tag{3.13}$$

Se observa que las señales en fase y cuadratura toman el valor de la fase de la señal recibida, y los términos en alta frecuencia son eliminados gracias a los filtros paso-bajo, como se representa en la figura 3.14. En el caso del DUC, el proceso es el inverso, pero se puede seguir una misma secuencia de pasos que con el DDC.

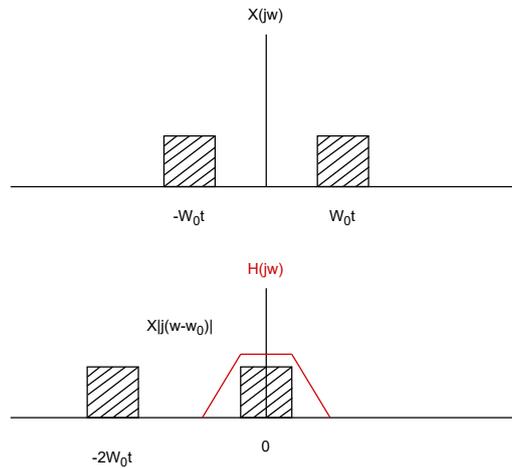


Figura 3.14: Filtrado de términos de alta frecuencia.

Otra forma de analizar las señales IQ es llevando a cabo una representación fasorial de las mismas. Al tratarse de un número complejo, una señal IQ puede describirse a partir de un módulo y una fase, como indica la ecuación 3.14, representándose como en la figura 3.15.

$$\begin{aligned} \text{Amplitud} &= \sqrt{I^2 + Q^2} \\ \text{Fase} &= \tan^{-1} \left(\frac{Q}{I} \right) \end{aligned} \quad x(t) = \sqrt{I^2 + Q^2} \cdot \cos \left(2\pi ft + \tan^{-1} \left(\frac{Q}{I} \right) \right) \quad (3.14)$$

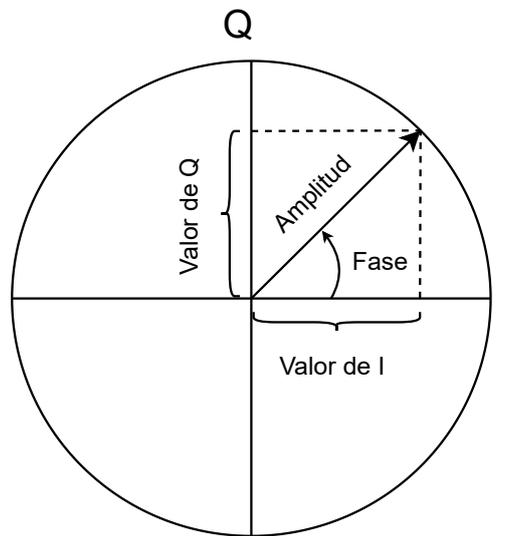


Figura 3.15: Componentes IQ representados en el plano complejo.

Tras este proceso, los SDR suelen aplicar filtros digitales tipo *multirate* para ajustar la tasa de muestreo óptima del sistema. Estos filtros dependen de la relación de interpolación o diezmado aplicada, llamada relación *multirate*, tal y como muestra la ecuación 3.15.

$$R_{\text{multirate}} = \frac{f_{\text{sistema}}}{f_{\text{simbolo}}} \quad (3.15)$$

En los SDR se emplean bancos de filtros *multirate* para la adaptación de la tasa de muestreo. Por ejemplo, para obtener una relación de diezmado de 200, se puede emplear un filtro con una relación *multirate* de 200 o utilizar cuatro filtros con relaciones de 50, u ocho filtros con relaciones de 25. El uso de relaciones *multirate* menores reduce el coste computacional del sistema, por lo que su uso está bastante extendido en la mayoría de las aplicaciones comerciales con SDR.

Por lo general, se diferencian tres tipos de filtros *multirate* en los DDC y DUC, que son:

- Filtro FIR (*Finite Impulse Response*): es un filtro con respuesta de fase lineal que utiliza métodos polifásicos para optimizar los tiempos de cómputo.
- Filtros CIC (*Cascade-Integrator-Comb*): es un filtro computacionalmente eficiente, ya que

requiere muy pocas operaciones aritméticas, aunque presenta una eficiencia espectral baja que obliga a corregir las atenuaciones en la banda de paso, generalmente aplicando un filtro FIR adicional. Suelen utilizarse para pasar de tasas de muestreo superiores a otras inferiores que pueden ser programables..

- Filtros RRC (*Root-Raised-Cosine*): es un filtro que se emplea para la conformación de pulsos, la modificación de tasas de muestreo o filtrado y la correlación de señales (filtro óptimo). Suelen aplicarse como última etapa de filtrado en topologías en cascada, ya que minimiza la interferencia entre símbolos.

3.2.4. Procesado digital de señales

La etapa de procesamiento digital de la señal cuenta con claras ventajas frente a las técnicas analógicas implementadas en arquitecturas radio tradicionales. Una de las principales motivaciones de estas técnicas es la adaptación de un mismo hardware a varias configuraciones software para diferentes aplicaciones. Esto produjo la aparición de los bloques de procesamiento digital de señales o DSP [28], aunque no son las únicas plataformas que se emplean como etapas de procesamiento digital de señales en arquitecturas SDR. Existen microprocesadores de propósito específico y FPGA que son capaces de tratar digitalmente las muestras recibidas, así como generar señales a partir de códigos en bajo nivel.

En esta etapa, el SDR realiza diferentes operaciones para comprimir o aumentar la tasa de datos generada por la aplicación, y en ella se integran los algoritmos de la aplicación. Es decir, todos aquellos aspectos que en las arquitecturas radio tradicionales eran tratados como bloques analógicos pasan a ser códigos que implementan esquemas de modulación y demodulación, codificación y decodificación, detección y corrección de errores e, incluso, protocolos radio para distintos estándares.

La configuración de este bloque en las arquitecturas SDR dependerá en gran medida de la aplicación que se pretenda desarrollar, por lo que debe ser estudiado de forma independiente. Por ejemplo, en aplicaciones biomédicas puede ser suficiente aplicar técnicas sobre frecuencias de muestreo de 1 Hz, mientras que para la transmisión y el procesamiento in-situ de imágenes, estas frecuencias de muestreo deben ser superiores a 10 MHz. En este último caso, la memoria necesaria para alojar las muestras adquiridas debe ser superior al del primer caso, por lo que se puede intuir

la importancia de la plataforma de procesamiento digital de señales que se debe usar en cada aplicación.

3.3. Arquitecturas basadas en radio definida por software

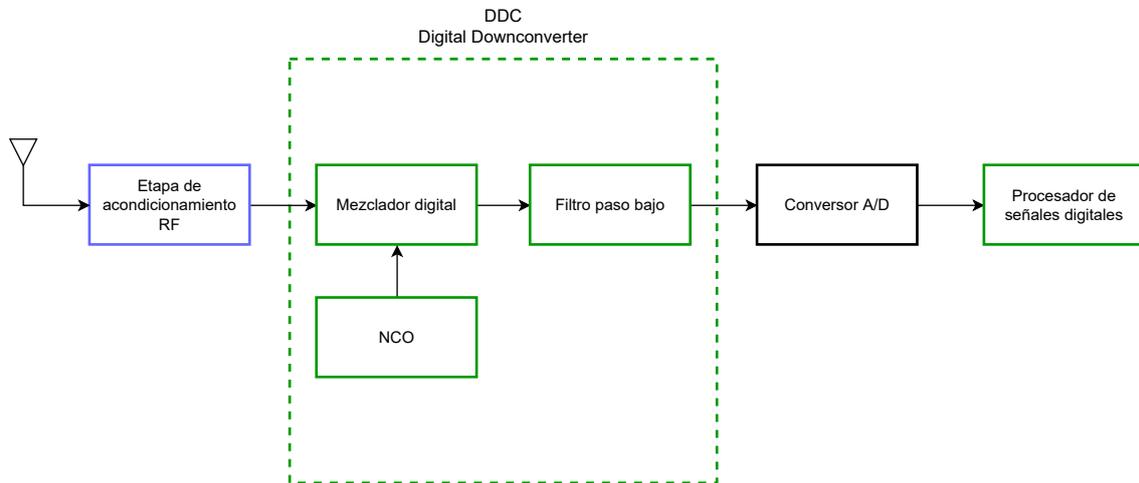


Figura 3.16: Diagrama de bloques de un receptor SDR.

En la figura 3.16, los bloques con borde azul son elementos analógicos y los bloques con borde verde son elementos digitales. El primer bloque tras la antena se considera opcional dependiendo de la aplicación a la que se destine el receptor. En capítulos posteriores se explicará la necesidad de este bloque y qué elementos analógicos lo conforman. Después de este bloque, se encuentra el ADC, necesario para muestrear la señal recibida y aplicarle tratamiento digital vía software. Finalmente, se ubica el segmento digital, que se define mediante un bloque software. Los bloques digitales son completamente análogos a sus contrapartes analógicas en cuanto a su funcionalidad.

Otros procedimientos comunes son la interpolación y el diezmado para modificar la tasa de muestreo o de símbolos según las necesidades de la aplicación. Estas operaciones ocurren en el DDC, que contiene el mezclador, NCO y filtro digital tipo FIR además de otros elementos como sumadores, multiplicadores y desplazadores. Por último, el bloque del procesador de señales digitales se encarga de realizar demodulaciones y decodificaciones entre otras funciones.

El transmisor de radio definida por software cuenta con una estructura similar al receptor, como se puede ver en la figura 3.17.

El procesador de señales digitales codifica y modula la señal según el esquema definido en la aplicación. El DUC interpola la señal para aumentar la tasa de bit y desplazar la señal banda base a la frecuencia deseada mediante el NCO y el mezclador. El convertor digital analógico convierte

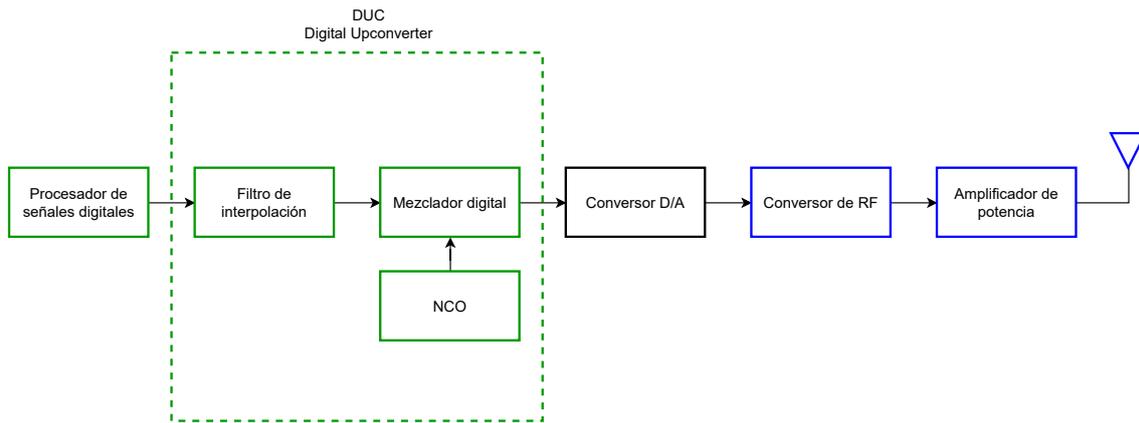


Figura 3.17: Diagrama de bloques de un transmisor SDR.

las muestras digitales al dominio analógico mientras que el convertor de RF desplaza la señal a transmitir a frecuencias superiores. Finalmente, el amplificador de potencia aumenta el nivel de amplitud de la señal para ser transmitida por la antena.

Actualmente, existen diferentes arquitecturas SDR, siendo las principales la arquitectura directa y la arquitectura FI-cero. Asimismo, es posible que la posición del bloque DDC también varíe, situándose antes o después del ADC.

3.3.1. Arquitectura directa

Gracias al avance de los convertidores analógico-digital, se ha acercado el ADC a la antena cada vez más [29]. Los únicos elementos que desaparecen serían los bloques encargados de realizar la conversión a FI o a banda base, para así conseguir la conversión directa. De esta forma, se evita el uso de osciladores locales y mezcladores permitiendo muestrear la señal en su frecuencia original (figura 3.18).

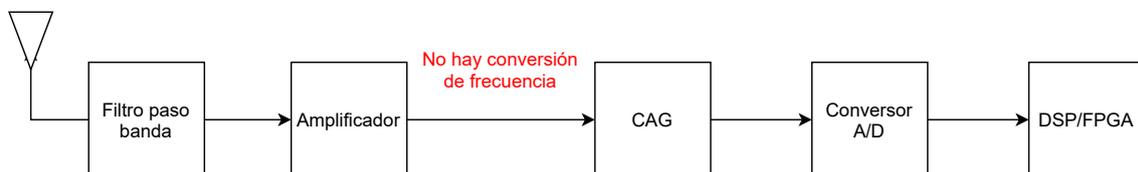


Figura 3.18: Diagrama de bloques de la arquitectura directa.

Esta arquitectura consiste principalmente en un filtro que selecciona la banda deseada, seguido de un amplificador de bajo ruido o LNA y el ADC. En algunas ocasiones, se incluye un control automático de ganancia o CAG para ajustar los niveles de señal de entrada y evitar la saturación del equipo. Se puede llegar a reemplazar la mayoría del filtrado analógico por digital, excepto el filtro

antialiasing y los filtros de reconstrucción de la señal. Ya que no se utiliza ninguna conversión de frecuencia, se simplifica en gran medida el diseño y el coste. Con la evolución de los ADC, se puede utilizar esta arquitectura en las banda L (1-2 GHz) y S (2-4 GHz). Además, existen conversores capaces de llegar a las giga muestras por símbolo (Gsp/s *Giga samples per symbol*) y se utilizan en las bandas C (4-8 GHz) y X (8-12 GHz) [30].

La principal ventaja de la arquitectura directa es la sencillez de la cadena de RF. Como consecuencia, al utilizar menos componentes se obtiene un menor consumo y una mayor eficiencia energética. Esta arquitectura es especialmente útil al diseñar arrays de radar activos en fase y conformar haces de señales desplazadas en fase emitidas por múltiples antenas. Además, también se reducen potenciales fuentes de ruido, frecuencias imagen y otros errores como las fugas de Oscilador Local (OL) y deterioros de cuadratura del propio dispositivo. Por último, también simplifica la sincronización. Para conseguir coherencia de fase, se ha de sincronizar el reloj interno del dispositivo como de los OL. Ya que no se requieren OL, solo se necesita sincronizar el reloj.

En [31] se muestra un ejemplo de esta arquitectura, donde se implementa tanto para una banda como para varias un receptor GPS. Para ello, se acoplan diferentes filtros paso-banda en paralelo. En el caso de muestreo multibanda, la frecuencia de muestreo mínima es el doble de la suma de los ancho de banda de interés.

3.3.2. Arquitectura FI-cero

La arquitectura FI-cero se puede encontrar hoy en día en la mayoría de los equipos de radio-comunicaciones comerciales que emplean SDR. La razón principal por la que se adopta principalmente esta arquitectura es debido a que reduce el precio, potencia y tamaño de los dispositivos [32]. Las arquitectura FI-cero actuales son capaces de ofrecer un alto rendimiento y consigue remediar los desperfectos asociados a este tipo de arquitecturas mediante la combinación de diseño, particionado, procesado y algorítmica. Esta arquitectura se representa en la figura 3.19.

La señal recibida es trasladada a banda base, eliminando la necesidad de utilizar tanto filtros de FI como de amplificadores para esta etapa. La selectividad se consigue introduciendo un par de filtros paso-banda en la rama de la señal IQ, pueden ser integrados como filtros paso-bajo activos en lugar de componentes fuera del integrado de FI, que introducen mayor ruido. La ventaja de los filtros activos frente a los fijos es que permiten ajustar su ancho de banda. De esta forma, permite

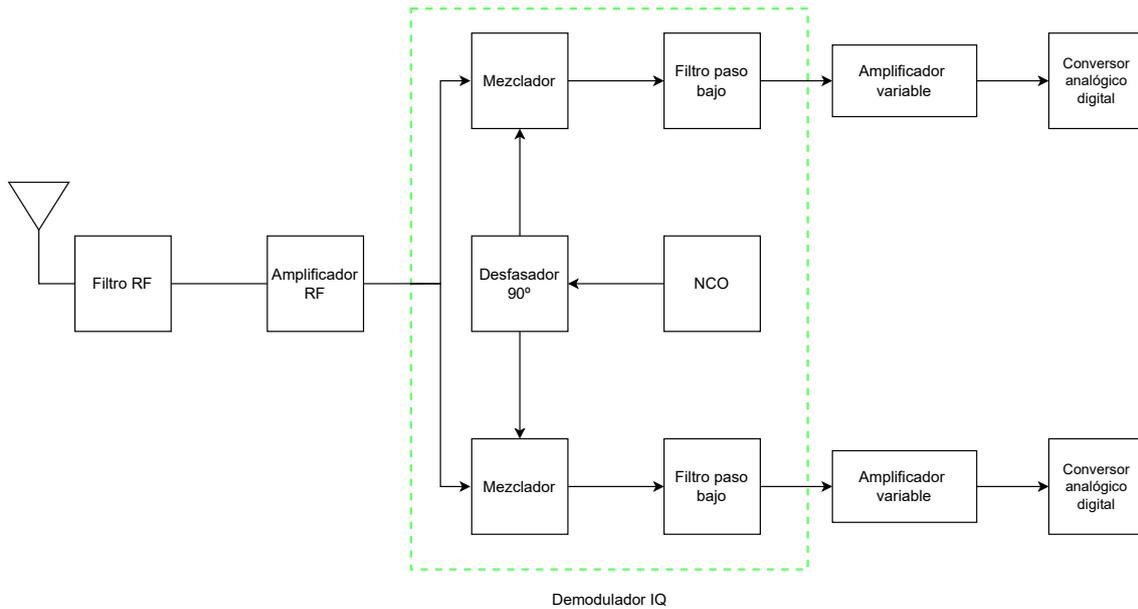


Figura 3.19: Diagrama de bloques de la arquitectura FI-cero.

al mismo dispositivo cubrir diferentes rangos de ancho de banda sin utilizar varios filtros fijos de FI.

Otra característica de esta arquitectura es que puede cubrir un amplio rango de frecuencias al cambiar los valores del NCO. El valor a establecer de este oscilador es igual a la frecuencia de interés, por lo que la señal recibida se convierte en señales IQ en banda base.

Sin el uso de filtros fijos, se pueden diseñar dispositivos flexibles, sin desarrollar variaciones para distintas bandas. Con el uso de los ADC y filtros banda base programables, la arquitectura FI-cero no solo se consigue un alto rendimiento y adaptabilidad, sino que el rendimiento se mantiene para cada configuración, sin necesidad de optimizar el circuito analógico. Asimismo, también reduce el tamaño requerido al eliminar los bancos de filtros necesarios para aplicaciones multibanda, simplificando a su vez el diseño. Además, los ADC y DAC operan con tasas de muestreo en banda base, que es menor que en otras arquitecturas y consigue un consumo de potencia menor. Por último, con la reducción de la circuitería analógica y otros componentes, esta arquitectura permite desarrollar transceptores con una reducción considerable de peso, tamaño y potencia, conocido en inglés como SWaP (*Size, Weight and Power*).

Sin embargo, esta arquitectura también presenta una serie de desventajas:

- Dificultad de mantener un offset de 90° entre la señal I y la señal Q. Estas imperfecciones resultan en una degradación en el rechazo de la imagen.

- Posibles fugas de OL.
- Tanto las fugas de OL como las imágenes degradan la sensibilidad del receptor produciendo componentes espectrales espúreas.

Para ello, se utilizan técnicas avanzadas de procesado de la señal para las señales IQ, como la compensación de fase IQ. El desfase IQ es un factor limitante en el diseño de receptores radio y sucede debido a las imperfecciones al desfasar la señal del oscilador 90° y generar así las señales en fase y cuadratura. Al dividir la señal en sus componentes I y Q, facilita la reconstrucción de la señal original, permitiendo controlar tanto la fase como la amplitud de la señal al modificar el nivel de las componentes, sin necesidad de operar con sus fases.

Existen diferentes técnicas para compensar estos desfases tanto para transmisores como receptores, siendo la más común el uso de patrones y modulaciones conocidos [33]. Además, se puede eliminar la componente continua de ambas ramas por separado para mejorar el rechazo de la portadora a la salida. De esta forma, se consigue mejorar los valores de los vectores de errores de la señal modulada, mejorando la decodificación y disminuyendo el BER.

3.3.3. Disposición del DDC/DUC

La disposición tanto del DDC como del DUC no son un asunto trivial [34]. Tradicionalmente, los receptores radio colocaban esta estructura antes del conversor, necesitando un conversor para la componente de fase y otro para la componente de cuadratura. Al situar el ADC detrás del DDC, se consigue simplificar los requisitos que el conversor debe cumplir. Esta estructura es la que comúnmente se utiliza en las plataformas SDR que se analizarán y se representa en la figura 3.20.

Sin embargo, existen tres factores limitantes con esta estructura:

- Se necesita que los subsistemas de ambas ramas tengan una respuesta muy similar, tanto en mezcladores, filtros y conversores.
 - Los filtros analógicos introducen distorsión de fase.
 - Los conversores introducen una componente continua que no es sencilla de eliminar. Además, se produce un *offset* en el conversor debido a sus bloques previos, como amplificadores y comparadores. Este *offset* influye en gran medida en sistemas que trabajan a muy bajas frecuencias.
-

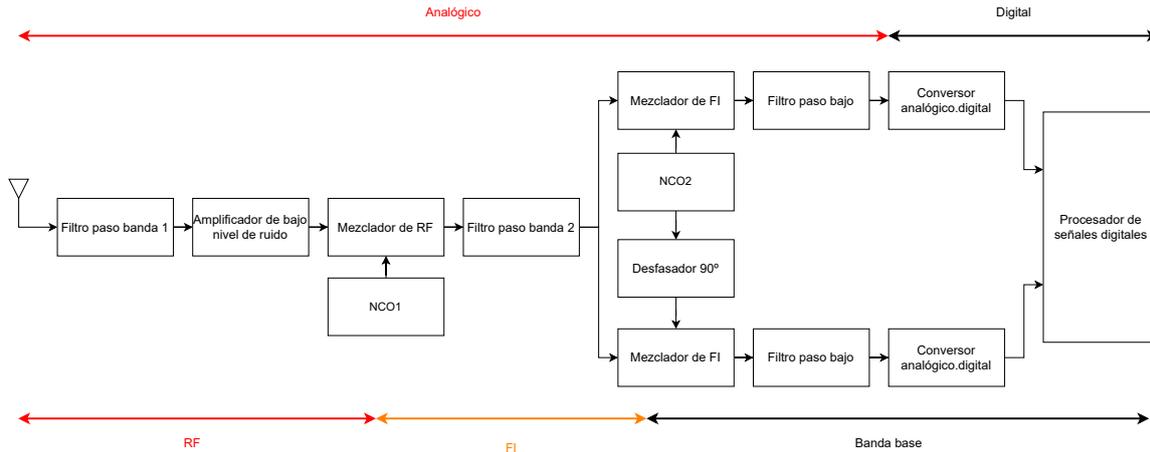


Figura 3.20: Diagrama de bloque de un receptor SDR tradicional.

Con el objetivo de evitar estos problemas, se ha considerado cambiar el orden de los bloques y situar el DDC después del convertor. Con este orden, se pueden implementar los componentes del DDC digitalmente, con lo que se eliminan las distorsiones de fase que generan los componentes. Asimismo, el diseño de filtros digitales de fase lineal es bastante sencillo y se puede eliminar la componente continua del convertor con mayor facilidad. El resultado de la nueva arquitectura se muestra en la figura 3.21.

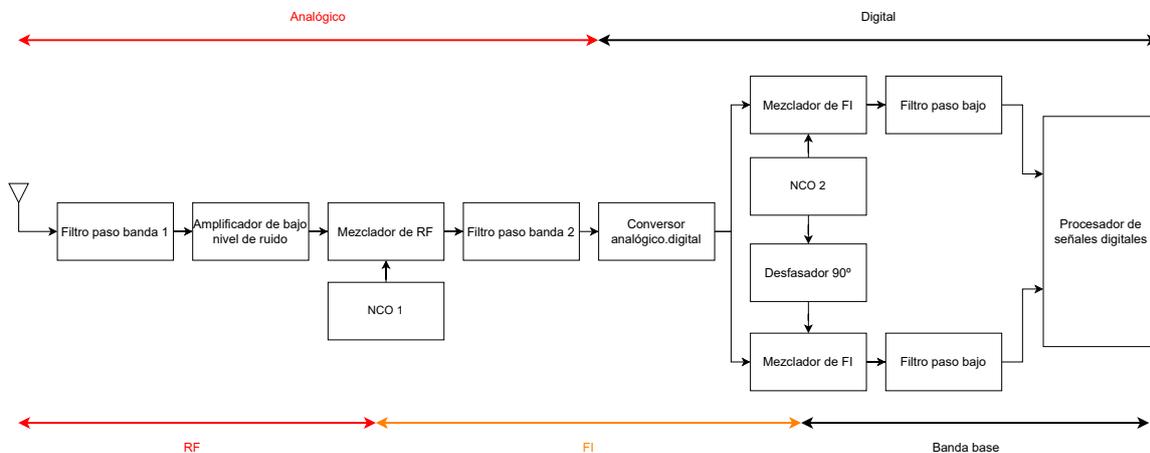


Figura 3.21: Nuevo diagrama de bloque propuesto de un receptor SDR.

3.4. Plataformas comerciales de radio definida por software

En esta sección se realizará una introducción a las diferentes plataformas de radio definida por software comerciales más conocidas. Además, se hará énfasis en sus características más importantes como precio, bandas de frecuencia de uso, ganancia, arquitectura y modos de funcionamiento.

Finalmente, se presentará, a modo de resumen, una tabla comparativa de las características de las plataformas analizadas.

3.4.1. RTL-SDR



Figura 3.22: Plataforma RTL-SDR.

RTL-SDR [35], creado por el grupo Osmocom, es un adaptador USB (*Universal Serial Bus*) que se puede conectar al ordenador y otorgar la capacidad de recibir señales radioeléctricas. El origen de esta plataforma viene de los adaptadores para sintonizar el estándar DVB-T (*Digital Video Broadcasting - Terrestrial*) para los televisores, basados en el integrado RTL2832U. Este integrado era capaz de acceder directamente a las muestras IQ sin procesar y, mediante un driver personalizado, se consiguió transformar el adaptador en una radio definida por software. Su principal característica es su bajo coste de adquisición, aproximadamente de 25 €, y la existencia de diferentes modelos que cubren diferentes bandas de frecuencia. La plataforma solo puede realizar la función de receptor, a diferencia de las demás plataformas comerciales analizadas en este capítulo. Además, se pueden encontrar diversas librerías y repositorios desarrollados por la comunidad amateur y son de libre acceso. En la tabla 3.1 se presentan los principales modelos y sus rangos de frecuencias asociados.

Los modelos que cubren un mayor rango de frecuencias son el Elonics E4000 y Rafael Micro R820T/2 siendo el segundo el más utilizado. El primero no dispone de la banda en su totalidad, sino que existe un *gap* que puede variar dentro de las frecuencias 1100 - 1250 MHz y no es utilizable. Respecto al segundo, es posible extender ligeramente el rango de frecuencias de trabajo mediante drivers experimentales, cubriendo la banda de 13 - 1864 MHz. La tasa de muestreo máxima de

Tabla 3.1: Modelos en función de su chip y rango de frecuencia asociados.

Chip	Rango de frecuencia
Elonics E4000	52 – 2200 MHz
Rafael Micro R820T/2	24 – 1766 MHz
Fitipower FC0013	22 – 1100 MHz
Fitipower FC0012	22 – 948,6 MHz
FCI FC 2580	146 – 308 MHz
	438 – 924 MHz

esta plataforma es de 3,2 Msps ¹, aunque ofrece un rendimiento inestable en muchas ocasiones. La tasa de muestreo máxima en el que el dispositivo es estable es de 2,56 Msps. La resolución del conversor analógico digital es de 8 bits, pero ENOB es de 7 bits. La potencia máxima de entrada que soporta el dispositivo es de +10 dBm. Por último, existen soluciones en diferentes lenguajes de programación, como Python, MATLAB/Simulink e incluso Android (Java), y tiene soporte para GNU Radio Companion [36].

3.4.2. HackRF One



Figura 3.23: Plataforma HackRF One.

HackRF One, creado por Great Scott Gadgets, es SDR capaz de transmitir y recibir señales en el rango de 1 MHz a 6 GHz. Es una plataforma de código abierto diseñada para el desarrollo y testeo de tecnologías radio de nueva generación. Cuenta con un encapsulado de plástico o de metal y un cable micro USB para su conexión como periférico, aunque también puede ser usado de forma independiente.

Esta plataforma es un transceptor que opera en modo half-duplex con una tasa de muestreo máxima de 20 Msps y una resolución de 8 bits. Cuenta con tres conectores SMA, uno para la

¹mega muestras por segundo

antena y dos para el sincronismo bajo una referencia de tiempo de reloj externa, tanto a la entrada como a la salida, posibilitando la conexión de varios HackRF One y reduciendo los problemas de jitter. Presenta un conector USB para la alimentación y configuración de datos. Es compatible con los sistemas operativos Windows, Linux y macOS, y programas como GNU Radio Companion, gqrx [37].

Mediante software, se pueden configurar los amplificadores, que tiene 3 etapas dedicadas para la recepción y 2 para la transmisión. En recepción, la etapa FI se puede configurar para ofrecer una ganancia de 0 a 40 dB en pasos de 8 dB, y en la etapa de banda base de 0 a 62 dB con pasos de 2 dB, mientras que en transmisión, la etapa FI permite ganancias de 0 a 47 dB con pasos de 1 dB. Tanto en recepción como transmisión, la etapa RF solo es posible activar el amplificador con ganancia de 14 dB o desactivarlo. A su vez, los filtros también son programables, con un máximo de 28 MHz y una caída de 3 dB hasta 30 MHz. Por último, la potencia máxima recibida recomendada es de -5dBm, aunque teóricamente sean 10 dBm debido a que por posibles errores del software, se active el amplificador de la cadena RF y dañe el dispositivo. Respecto a la potencia de transmisión, en la tabla 3.2 se puede ver los niveles según las bandas de frecuencias.

Tabla 3.2: Potencias máximas de transmisión según el rango de frecuencias para HackRF One.

Banda de frecuencia	Potencia máxima de TX
1 MHz – 10 MHz	+5 – +15 dBm aumenta al aumentar la frecuencia
10 MHz – 2150 MHz	+5 – +15 dBm disminuye al aumentar la frecuencia
2150 MHz – 2750 MHz	+13 – +15 dBm
2750 MHz – 4000 MHz	0 – +5 dBm disminuye al aumentar la frecuencia
4000 MHz – 6000 MHz	-10 – 0 dBm disminuye al aumentar la frecuencia

3.4.3. BladeRF



Figura 3.24: Plataforma BladeRF xA9.

BladeRF [38], creada por Nuand, es una plataforma para la siguiente generación de radios definida por software, con un rango de frecuencia de 47 MHz a 6 GHz, una tasa de muestreo de 61,44 MHz y capaz de emplear un MIMO (*Multiple Input Multiple Output*) con configuración 2x2. Diseñado para aplicaciones de alto rendimiento, es compatible para los sistemas operativos Windows, Linux y macOS, y tiene soporte para GNU Radio Companion, gqrx y MATLAB/Simulink entre otros. Las licencias de las bibliotecas, firmware y plataformas HDL (*Hardware Description Language*) son de código abierto y están disponibles en la red.

Tanto la FPGA como el controlador de USB 3.0 son programables utilizando las herramientas y SDK (*Software Development Kit*) que provee el vendedor de forma gratuita. Cuenta con control automático de ganancia y corrección automática de las componentes continuas e IQ, y una resolución de 12 bits en los ADC y DAC. Además, se puede programar el ancho de banda del filtro de RF de 0.2 a 56 MHz. El reloj interno está calibrado de fábrica a 38.4 MHz, pero se puede ajustar el valor de del oscilador mediante un PLL (*Phase-Locked Loop*) integrado o con el DAC. Por último, existen diferentes modelos que cambian la FPGA según las necesidades exigidas, siendo el modelo xA4 el de menores prestaciones y el modelo xA9 el de mejores prestaciones, con sus características recogidas en la tabla 3.3.

Tabla 3.3: Características de la FPGA según el modelo de la plataforma BladeRF.

	xA4	xA9
Elementos lógicos	49000 elementos	301000 elementos
Memoria	3383 kbits	13917 kbits
Multiplicadores empotrados 18x18	132 multiplicadores	684 multiplicadores

La plataforma se alimenta con 4.5 voltios mediante USB, y tiene una potencia máxima de recepción de +23 dBm y una potencia máxima de transmisión de +8 dBm. El fabricante dispone de diferentes accesorios para la plataforma, como una carcasa para proteger los componentes electrónicos y amplificadores tanto para transmisión (BT-100) como para recepción (BT-200).

3.4.4. ADALM-Pluto

El ADALM-PLUTO o PlutoSDR [39], creada por Analog Devices Inc, es una plataforma de radio definida por software destinada para el aprendizaje de las radiocomunicaciones y la tecnología SDR. Cuenta con dos integrados, el AD9363, configurado por defecto en la plataforma, y el AD9364. Ambos integrados son muy similares entre sí, donde sus principales diferencias son



Figura 3.25: Plataforma ADALM-Pluto.

el rango de frecuencia soportado, el ancho de banda del filtro y el número de canales disponibles tanto en transmisión como en recepción (tabla 3.4). Para efectuar este cambio se debe conectar la plataforma al ordenador mediante USB y utilizando una interfaz serie como PuTTY introducir una serie de comandos para modificar el firmware del dispositivo [40].

Tabla 3.4: Características de los integrados AD9363 y AD9364 del PlutoSDR.

	AD9363	AD9364
Banda de frecuencia	325 MHz – 3800 MHz	70 MHz – 6000 MHz
Ancho de banda máximo del filtro	20 MHz	56 MHz
Número de canales de transmisión	2 canales	1 canal
Número de canales de recepción	2 canales	1 canal
Figura de ruido del receptor	<3 dB	<2.5 dB

Cuenta con unos ADC y DAC de 12 bits de resolución, capaces de soportar una tasa de muestreo de 61.44 Msps. Tiene soporte para los sistemas operativos Windows, Linux y mac OS. A su vez, es programable mediante MATLAB/Simulink, GNU Radio Companion, C y Python, entre otros. Su potencia máxima de saturación en recepción es de +2.5 dBm, mientras que su potencia máxima transmitida es de +7 dBm. Para su uso, se requieren diferentes drivers que dependen del sistema operativo y bibliotecas que el fabricante provee de forma gratuita [41].

Se ha demostrado que esta plataforma SDR es versátil, ya que se ha implementado en diversos campos como comunicación marítima satelital [42], radares de onda continua modulada [43], clasificación de señales en tiempo real mediante *Deep Learning* [44], radio cognitiva [45], radio-astronomía [46] y como anteriormente se ha nombrado, enseñanza [47].

3.4.5. Serie USRP



Figura 3.26: Plataforma USRP serie N200.

La plataforma USRP (*Universal Software Radio Peripheral*) [48], creado por Ettus Researchs, es un transceptor ajustable pensado para el diseño, prototipado y despliegue de sistemas de radiocomunicación. Cuenta con una gran variedad de modelos que se ajustan a las necesidades o requisitos de los usuarios. De forma resumida, se analizarán los modelos USRP N200 y N210. Estos modelos ofrecen el hardware con mayor rendimiento dentro de la familia USRP y son utilizados en aplicaciones que requieren grandes anchos de banda y alto rendimiento, como prototipado de capas físicas, acceso dinámico y monitorización del espectro y despliegue de redes de sensores. Tiene soporte para los sistemas operativos de Windows, Linux, y macOS y programable mediante C, Python, MATLAB/Simulink, GNU Radio Companion y LabVIEW [49].

Opera en la banda DC - 6 GHz, con un ancho de banda de 50 MHz para muestras de 8 bits y de 25 MHz para muestras de 16 bits. La potencia máxima permitida por el dispositivo tanto de entrada como de salida es de + 15 dBm. El ADC utilizado en estas plataformas dispone de 14 bits de resolución y una tasa de muestreo de 100 Msps, mientras que el DAC cuenta con 16 bits de resolución y una tasa de muestreo de 400 Msps. Sin embargo, estas tasas son teóricas, ya que la interfaz serie USB3.0 limita este aspecto a 61.44 Msps. Además, son capaces de sincronizarse mediante una fuente de sincronismo externa a una frecuencia de 5 o 10 MHz. Ambas plataformas cuentan con dos canales tanto para transmisión como para recepción haciendo uso de la configuración MIMO. Por último, la precisión de frecuencia del dispositivo es de aproximadamente de 2.5 ppm ². La principal diferencia entre ambas plataformas reside en la FPGA y, en consecuencia, el coste de adquisición.

²Partes por millón

3.4.6. LimeSDR



Figura 3.27: Plataforma LimeSDR.

LimeSDR, creada por Lime Microsystems, es una plataforma de radio definida por software de bajo coste y código abierto capaz de soportar cualquier estándar de comunicación inalámbrica. Los fabricantes ofrecen tanto los esquemáticos del dispositivo como de su *firmware*, y están disponibles de forma abierta en la web. Tienen soporte para Python, C, GNU Radio Companion y LabVIEW, así como para los sistemas operativos de Windows, Linux y macOS.

La plataforma cuenta con un rango de frecuencia de trabajo comprendido desde los 100 KHz hasta los 3.8 GHz, y un ancho de banda de 61,44 MHz. Los ADC y DAC cuentan con 12 bits de resolución, soportan una tasa de muestreo máxima de 61.44 Msps, limitada por su puerto USB 3.0. Emplea MIMO 2x2 con canales full duplex y utiliza 10 conectores U.FL, 4 para transmisión y 6 para recepción. Por último, tiene una precisión de 4 ppm y una potencia máxima de transmisión que varía entre 0 y +10 dBm dependiendo de la frecuencia, y una potencia máxima de saturación en recepción de +12 dBm.

3.4.7. Comparación de plataformas comerciales SDR

En este punto, se recogen en la tabla 3.5 las características de todas las plataformas revisadas a modo de resumen, analizando el rango de frecuencia, ancho de banda de los filtros, bits de resolución de los convertidores, sistemas operativos soportados, frameworks y lenguajes de programación disponibles, así como su coste de mercado.

Tabla 3.5: Recopilación de las principales características de las plataformas analizadas.

	RTL-SDR (R820T/2)	HackRF One	BladeRF	PlutoSDR (AD9364)	LimeSDR	USRP N210	
Hardware							
Número de canales TX	-	1 canal	2 canales	1 canal	2 canales	2 canales	
Número de canales RX	1 canal	1 canal	2 canales	1 canal	2 canales	2 canales	
Rango de frecuencias TX	-	1 – 6000 MHz	70 – 6000 MHz	70 – 6000 MHz	0.1 – 3800 MHz	DC – 6000 MHz	
Rango de frecuencias RX	24 – 1766 MHz	1 – 6000 MHz	70 – 6000 MHz	70 – 6000 MHz	0.1 – 3800 MHz	DC – 6000 MHz	
Máximo ancho de banda del filtro	250 MHz	28 MHz	56 MHz	56 MHz	61.44 MHz	50 MHz (8 bits) 25 MHz (16 bits)	
Potencia máxima TX	-	Ver en la tabla 3.2	+8 dBm	+7 dBm	+10 dBm	+15 dBm	
Potencia máxima RX	+10 dBm	Ver en la tabla 3.2	+8 dBm	+7 dBm	+12 dBm	+15 dBm	
Tasa máxima TX	-	20 Msps	61.44 Msps	61.44 Msps	61.44 Msps	100 Msps	
Tasa máxima RX	2.56 Msps	20 Msps	61.44 Msps	61.44 Msps	61.44 Msps	400 Msps	
Número de bits de resolución ADC	8 bits	8 bits	12 bits	12 bits	12 bits	14 bits	
Número de bits de resolución DAC	-	8 bits	12 bits	12 bits	12 bits	16 bits	
Sincronización externa	-	✓	✓	-	-	✓	
Precisión en frecuencia	± 18 ppm	± 20 ppm	± 1 ppm	± 25 ppm	± 4 ppm	± 2.5 ppm	
Transceptor	R820T/2	Hardware personalizable	± 1 ppm	AD9364	LMS7002M	ET1011	
Software							
Sistemas operativos soportados	W, L y M	W, L y M	W, L y M	W, L y M	W, L y M	W, L y M	
Soporte en C/C++	✓	✓	✓	✓	✓	✓	
Soporte en Python	✓	✓	✓	✓	✓	✓	
Soporte en MATLAB/Simulink	✓	✗	✓	✓	✗	✓	
Soporte en GRC	✓	✓	✓	✓	✓	✓	
Esquemáticos de libre acceso	✓	✓	✓	✓	✓	✓	
Firmware de libre acceso	✓	✓	✓	✓	✓	✓	
Precio							
Precio de la plataforma	~ 25 €	~ 285 €	~ 460 € (xA4) ~ 637 € (xA9)	~ 202 €	~ 283 €	~ 2324 €	

Plataformas comerciales de

no definida por soft

ware

4 | Implementación de algoritmos AIS y VDES para plataformas SDR

En este capítulo se explicará en detalle los algoritmos de recepción y transmisión de paquetes AIS, y el algoritmo de transmisión de paquetes VDES sobre una plataforma transceptora SDR. Se empezará explicando los parámetros establecidos en la plataforma SDR para poder transmitir y recibir los paquetes según los estándares AIS y VDES. Tras definir estos parámetros se define el algoritmo de transmisión y recepción para AIS, y el algoritmo de transmisión para VDES, siguiendo el orden de sus diagramas de flujo.

4.1. Parámetros de configuración del SDR

El primer paso de ambos algoritmos es la definición de parámetros y la configuración de la plataforma SDR utilizada, el ADALM-PLUTO. Aunque el algoritmo de recepción cuente con más parámetros, se recogen todos en esta sección y se analiza su impacto en el funcionamiento de los algoritmos:

- *Frecuencia central.* Frecuencia de sintonía de la plataforma SDR. Se ha establecido la frecuencia 162.025 para todos los casos.
- *Muestras por símbolo.* Número de muestras que conforman un símbolo y con el que se puede cambiar la tasa de muestreo. Esta variable se define con valores arbitrarios, ya que afecta directamente a la tasa de muestreo, pero en los algoritmos AIS se le define el valor de 24, ya que aporta mejores resultados, mientras que en VDES es variable según el servicio.
- *Tasa de símbolo.* Número de pulsos por segundos definido por la recomendación ITU-M-1371 [4] con un valor de 9600 baudios o símbolos por segundo para AIS y variable en VDES

según el servicio.

- *Tasa de muestreo.* Frecuencia de adquisición de muestras por segundo, que depende de las variables *muestras por símbolo* y *tasa de símbolo* según la expresión $f_S = S_{symbol} \cdot R_S$. La frecuencia de muestreo que permite la plataforma ADALM-PLUTO está comprendida entre 65,105 KHz y 61,44 MHz. También se considera el ancho de banda de recepción.
- *Factor BT.* Producto del ancho de banda del filtro gaussiano y la tasa binaria de la señal. Según la recomendación ITU-M-1371 [4], para transmisión se emplea un factor de 0.4 y en recepción de 0.5.
- *Duración del pulso.* Tamaño del pulso medido en tiempo de bits explicado en la figura 2.7 y que varía según el *factor BT*.
- *Tamaño de ventana deslizante.* Tamaño definido para la ventana deslizante empleada para la búsqueda del segmento de señal con mayor potencia. Dependiendo de su valor el algoritmo requerirá de mayor coste computacional y toma valores en potencia de 2, siendo los más comunes 512 y 1024. Se ha utilizado el valor de 1024.
- *Tamaño máximo de paquete.* Tamaño definido de la trama recibida. Debe tener una longitud mínima igual al número de bits de la trama AIS, aunque por seguridad se ha decidido utilizar un valor definido como el producto $MP_S = 600 \text{ bits} \cdot sps$ ¹.
- *Tamaño del buffer de la plataforma SDR.* Número de muestras que la plataforma SDR capta entre transmisiones. Una vez que el buffer se llena, la plataforma envía al ordenador las muestras IQ recibidas. Esta variable modifica en gran medida la velocidad del algoritmo de recepción, ya que el tiempo de recepción se puede calcular como $T_{ejecucin} = \frac{B_S}{f_s}$ ².
- *Ganancia.* Ganancia interna de la plataforma SDR. Esta ganancia tiene diferentes rangos de valores; en transmisión se puede poner valores desde -89.75 dB hasta 0 dB con pasos de 0.25 dB, mientras que en recepción tiene un rango de -4 hasta 71 dB con pasos de 0.25 dB.

Para poder utilizar la plataforma SDR y programar los algoritmos, se han empleado las *toolboxes* oficiales de MATLAB, siendo estas:

¹ $MP_S = \text{Max Packet Size}$, $sps = \text{samples per symbol}$

² $B_S = \text{Buffer Size}$

- *Communications Toolbox Support Package for Analog Devices ADALM-Pluto Radio*. Necesaria para definir y utilizar la plataforma ADALM-PLUTO.
- *Signal Processing Toolbox*, *DSP System Toolbox* y *Communications Toolbox*. Necesaria para el uso de algunas funciones específicas que no incorpora MATLAB de base.

4.2. Algoritmo de transmisión AIS

El algoritmo de transmisión está dividido en dos secciones, la generación de mensajes NMEA0183 (*National Marine Electronics Association*) AIS, programado en Python 3.9, y la generación de tramas AIS y su transmisión, programado en MATLAB. Se empleó Python para la generación de los mensajes debido a la gran cantidad de bibliotecas de código libre disponibles, facilitando el proceso. Para la generación de tramas y transmisión AIS se ha utilizado MATLAB por su facilidad para prototipar algoritmos mediante este lenguaje. En la figura 4.1 se puede ver el diagrama de flujo de este algoritmo.

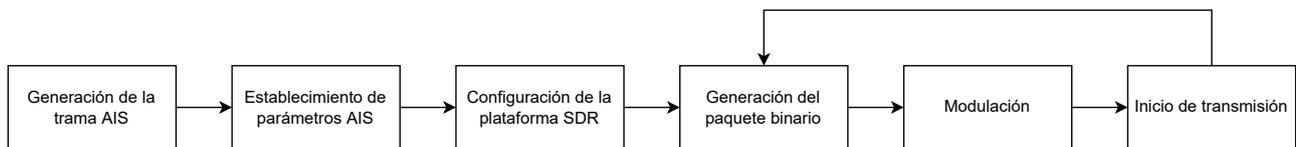


Figura 4.1: Diagrama de flujo del algoritmo de transmisión de paquetes AIS.

4.2.1. Generación de mensajes NMEA0183 AIS

Se ha utilizado la biblioteca PyAIS [50] para la generación de mensajes NMEA0183, debido a que incluye diversos métodos para generar cualquiera de los 27 tipos de mensajes AIS. Se ha elegido producir mensajes tipo 1, los informes de posición transmitidos por los dispositivos AIS Clase A. Los campos de cada mensaje se pueden ver en [51] y se explicarán los campos utilizados a la hora de producir el mensaje, junto al valor escogido y el mensaje final:

- *MMSI*. Es el número de identificación del servicio móvil marítimo que distingue inequívocamente cada dispositivo AIS y consiste en una serie de nueve dígitos. Se ha generado un número aleatorio del 0 al 999999999 que servirá como el MMSI del mensaje, siendo "249191000".

- *Repetición.* El indicador de repetición permite que los transpondedores AIS retransmitan el mensaje o no, permitiendo que llegue a embarcaciones lejanas. El valor máximo de retransmisiones es 3.
 - *Estado.* Campo que indica el estado de la embarcación con 16 posibilidades. Con un 0 se indica que la embarcación está en movimiento con el motor activo.
 - *Giro.* Indica el grado de giro que está ejecutando el buque. Se ha establecido 0 para indicar que no está girando.
 - *Velocidad.* Indicador de velocidad en nudos con una resolución a nivel decimal desde 0 hasta un máximo de 1022 o 102.2 nudos. En caso de no poder indicar la velocidad se establece el valor de 1023. Se estableció un valor de 50 para simular una velocidad de 5 nudos.
 - *Precisión.* Indica si la precisión del campo segundos es menor a 10 ms con un 1 o con un 0 si es mayor a 10 ms.
 - *Longitud.* Posición de la embarcación en el meridiano que cubre de 0 a 180 para indicar el este y de 0 a -180 para indicar el oeste. Utilizar el valor 181 cuando esta información no está disponible. Se ha utilizado un valor aleatorio comprendido entre -10.0 y 10.0, resultando en -6.
 - *Latitud.* Posición en el ecuador que cubre de 0 a 90 para indicar el norte y 0 a -90 para indicar el sur. Utilizando el valor 91 indica que esta información no está disponible. Al igual que con la longitud, se empleó un número aleatorio resultando -8.2.
 - *Dirección.* Indicador de la dirección que toma la embarcación, siendo las posibilidades de 0 al 359, con el 0 indicando el norte. Se ha elegido el valor 180 indicando que el buque se dirige hacia el sur.
 - *Segundos.* El campo de segundos que proviene del sincronismo UTC. Si se usa el valor 60 significa que no está disponible, 61 si el sistema de posicionamiento está en modo manual, 62 si está en modo aproximado y 63 si no está operativo.
 - *Maniobra.* Indica si la embarcación está ejecutando alguna maniobra especial como cruces regionales. Con 0 indica que esta información no está disponible, 1 que no realiza ninguna maniobra especial y 2 que si se está realizando maniobra.
-

En la sección de código 4.1 se muestra el código utilizado para generar nuevas tramas NMEA0183. Consiste en la creación de un objeto *MessageType1* al que se le asignan los parámetros explicados previamente. Tras crear este objeto, se utiliza el método *encode_msg* para que el objeto se codifique según el protocolo NMEA0183 y también se asigna el canal de transmisión. Finalmente, los métodos *print* permiten visualizar el resultado del código.

Código 4.1: Creación de tramas NMEA0183 con PyAIS.

```

1 mensaje1 = MessageType1.create(mmsi=249191000, repeat=0, status=0, turn=0,
2                               speed=50, accuracy=0, lat=-8.2, lon=-6,
3                               heading=180, second=3, maneuver=1)
4 print(mensaje1)
5
6 mensaje1_cod = encode_msg(mensaje1, radio_channel="A")
7 print(mensaje1_cod)

```

El resultado de la generación del mensaje se puede ver en la sección de código 4.2. Este mensaje se utilizará en el algoritmo de MATLAB que se explicará a continuación.

Código 4.2: Trama NMEA0183 resultante.

```

MessageType1(msg_type=1, repeat=0, mmsi=249191000,
status=<NavigationStatus.UnderWayUsingEngine: 0>, turn=0.0, speed=50.0,
accuracy=False, lon=-6.0, lat=-8.2, course=0, heading=180, second=3, maneuver=1,
spare_1=b'', raim=0, radio=0)
[ '!AIVDO,1,1,,A,13eaJF007l0TR<1sCe@005'6P000,0*27' ]

```

4.2.2. Generación de tramas AIS y transmisión

Una vez generado el mensaje NMEA, el primer paso es convertir el campo de datos en hexadecimal. Siendo el mensaje "AIVDO,1,1,,A,13eaJF007l0TR<1sCe@00g'6P000,0*27", el campo de datos se encuentra entre la penúltima y la última coma. Destacar que esta cadena se encuentra codificada en ASCII (*American Standard Code for Information Interchange*) de 8 caracteres, pero AIS emplea una codificación ASCII de 6 caracteres, por lo que es necesario esta conversión en primer lugar. Para ello, se han seguido los pasos descritos en [20], donde se explica que para realizar esta conversión se ha de restar 48 al valor ASCII del carácter. Adicionalmente, si el valor de esta resta es mayor a 40 se le ha de restar 8 a este nuevo valor. Tras realizar esta operación con todo el vector,

se pasa a binario y se agrupan estos bits en grupos de 6. En la sección de código 4.3 se muestra el código de MATLAB que realiza la conversión explicada.

Código 4.3: Conversión de UTF-8 a UTF-6.

```

1 mensaje_ais = '!AIVDO,1,1,,A,1Amsg=5B20jW9hp0FD>@1u6@6V00?,0*2B'
2 payload_ais = find(mensaje_ais == ',', 2, 'last');
3 data_ais = mensaje_ais(payload_ais(1)+1:payload_ais(2)-1);
4
5 % Convertir PAYLOAD en un vector de 6 bits:
6 mensaje_ais_binario = unicode2native(data_ais); % valores de 48 a 119
7 a = mensaje_ais_binario;
8 b = dec2hex(a);
9 mensaje_ais_binario_vector = [];
10 for i = 1:length(mensaje_ais_binario)
11     if (mensaje_ais_binario(i) > 47)
12         mensaje_ais_binario(i) = mensaje_ais_binario(i) - 48;
13         if(mensaje_ais_binario(i) > 39)
14             mensaje_ais_binario(i) = mensaje_ais_binario(i) - 8
15         end
16         mensaje_ais_binario_vector = [mensaje_ais_binario_vector, dec2bin(
17             mensaje_ais_binario(i), 6)];
18 end
19 mensaje_ais_binario = str2num(mensaje_ais_binario_vector)';

```

Utilizando el nuevo vector y con el uso de tablas, se transforman los grupos de 6 bits en la nueva asignación de chars y posteriormente en hexadecimal. En la sección de código 4.4 se puede observar las tablas utilizados para estas conversiones.

Código 4.4: Tablas utilizadas para cambiar de ASCII 8 bits a ASCII bits.

```

1 tabla_ascii6_bits = ['000000'; '000001'; ...
2     '000010'; '000011'; '000100'; '000101'; ...
3     '000110'; '000111'; '001000'; '001001'; ...
4     '001010'; '001011'; '001100'; '001101'; ...
5     '001110'; '001111'; '010000'; '010001'; ...
6     '010010'; '010011'; '010100'; '010101'; ...
7     '010110'; '010111'; '011000'; '011001'; ...
8     '011010'; '011011'; '011100'; '011101'; ...
9     '011110'; '011111'; '100000'; '100001'; ...
10    '100010'; '100011'; '100100'; '100101'; ...

```

```

11     '100110'; '100111'; '101000'; '101001'; ...
12     '101010'; '101011'; '101100'; '101101'; ...
13     '101110'; '101111'; '110000'; '110001'; ...
14     '110010'; '110011'; '110100'; '110101'; ...
15     '110110'; '110111'; '111000'; '111001'; ...
16     '111010'; '111011'; '111100'; '111101'; ...
17     '111110'; '111111'];
18
19 tabla_ascii6_char = ['0'; '1'; '2'; '3'; '4'; '5'; ...
20     '6'; '7'; '8'; '9'; ':'; ';' '<'; '=' '>'; '?'; '@'; ...
21     'A'; 'B'; 'C'; 'D'; 'E'; 'F'; 'G'; 'H'; 'I'; 'J'; 'K'; ...
22     'L'; 'M'; 'N'; 'O'; 'P'; 'Q'; 'R'; 'S'; 'T'; 'U'; 'V'; ...
23     'W'; '`'; 'a'; 'b'; 'c'; 'd'; 'e'; 'f'; 'g'; 'h'; 'i'; 'j'; ...
24     'k'; 'l'; 'm'; 'n'; 'o'; 'p'; 'q'; 'r'; 's'; 't'; 'u'; 'v'; ...
25     'w'];
26
27 tabla_ascii6_hex = ['00'; '01'; '02'; '03'; '04'; ...
28     '05'; '06'; '07'; '08'; '09'; '0A'; '0B'; '0C'; ...
29     '0D'; '0E'; '0F'; '10'; '11'; '12'; '13'; '14'; ...
30     '15'; '16'; '17'; '18'; '19'; '1A'; '1B'; '1C'; ...
31     '1D'; '1E'; '1F'; '20'; '21'; '22'; '23'; '24'; ...
32     '25'; '26'; '27'; '28'; '29'; '2A'; '2B'; '2C'; ...
33     '2D'; '2E'; '2F'; '30'; '31'; '32'; '33'; '34'; ...
34     '35'; '36'; '37'; '38'; '39'; '3A'; '3B'; '3C'; ...
35     '3D'; '3E'; '3F'];

```

Para realizar el cambio del vector de grupos binarios de 6 bits a hexadecimal se emplean bucles *for* para recoger el índice del grupo binario a transformar y se busca su equivalente en caracteres ASCII de 6 bits. Tras esta conversión, se realiza el mismo proceso para transformar los caracteres en sus valores hexadecimales. La sección de código 4.5 muestra como se realiza este proceso.

Código 4.5: Conversión de UTF-6 a hexadecimal.

```

1 %Conversión de paquetes de 6 bits en caracteres ASCII de 6 bits:
2 for k = 1:length(mensaje_ais_binario_6bits)
3     val = mensaje_ais_binario_6bits(k, 1:6);
4     for i = 1:length(tabla_ascii6_bits)
5         if (val == tabla_ascii6_bits(i, 1:6))
6             ind = i;
7         end

```

```

8     end
9     vector_caracter(k) = tabla_ascii6_char(ind);
10  end
11
12  % Conversión de paquetes de caracteres ASCII de 6 bits a hexadecimal:
13  for k = 1:length(mensaje_ais_binario_6bits)
14      val = mensaje_ais_binario_6bits(k, 1:6);
15      for i = 1:length(tabla_ascii6_hex)
16          if (val == tabla_ascii6_bits(i, 1:6))
17              ind = i;
18          end
19      end
20      vector_hexadecimal(k, 1:2) = tabla_ascii6_hex(ind, 1:2);
21  end

```

Tras este proceso se definen las variables necesarias, los campos de la trama y se configura la plataforma SDR con los valores establecidos en el estándar. En la sección de código 4.6 se observan los valores establecidos y la creación del objeto SDR.

Código 4.6: Definición de parámetros y objeto SDR en transmisión.

```

1  % Definición de variables:
2  samples_per_symbol = 24; % Valor arbitrario, con 24 muestras da mejores
   resultados
3  symbol_rate = 9600; % 9600 -> estandar
4
5  % Tasa de muestreo (ancho de banda de adquisicion de señales):
6  sample_rate = samples_per_symbol*symbol_rate;
7
8  bt_factor = 0.4; % Estándar -> 0.4
9  pulse_size = 3; % Otros valores da errores
10 ais_data_1 = '483B69696003FFFE488C07B4ED41E10FFF80081773CF6D'; % Mensaje
   generado
11
12 ais_training_sequence = logical([1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
   0]); % 0xAAAAAA
13 ais_start_flag = logical([0 1 1 1 1 1 1 0]); % 0x7E
14 ais_stop_flag = ais_start_flag;
15
16 tx_sdr = sdrtx('Pluto', ...

```

```

17         'RadioID', 'usb:0', ...
18         'CenterFrequency', 162.025e6, ... %Frecuencia del canal AIS B
19         'BasebandSampleRate', sample_rate, ...
20         'Gain', 0); %Valores entre -89.75 y 0

```

Una vez se tiene la cadena hexadecimal, se prepara el mensaje añadiendo el preámbulo, los *flags* de inicio y fin y dos cadenas de '0's lógicos al principio y al final de la trama para evitar las propias interferencias de la plataforma. Para crear la trama, se binariza la cadena hexadecimal y se invierte el orden de los bits, debido a que el bit más significativo en AIS se encuentra en la derecha. Tras la inversión, se realiza un proceso denominado *bit stuffing* que consiste en añadir un '0' lógico cada cinco '1's y viceversa. Luego se concatenan todos los vectores binarios de la forma *preámbulo + flag_i + datos + flag_f* junto a los conjuntos de '0's comentados anteriormente. Tras formar la trama, se codifica mediante NRZI. En la sección de código 4.7 se puede observar como se implementa esta parte en MATLAB y en las secciones 4.8 y 4.9 se recoge el código de los métodos *Bytes_Flipped* y *Bits_Stuffing*.

Código 4.7: Generación y codificación de la trama.

```

1 pre_tr = logical([0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0]);
2 post_stop = pre_tr;
3 msg = ais_data_1;
4 size_msg = size(msg);
5
6 %Declaración de la trama rellena con '0's
7 bit_stream = false(size_msg(1), length(start_flag) + ...
8 length(training_sequence) + length(end_flag) + ...
9 size_msg(2)*4 + 100);
10
11 max_size_msg = size_msg(2)*4;
12 for k = 1:size_msg(1)
13     flipped_bits = Bytes_Flipped(bit_msg(k,:));
14     stuffed_bits = Bits_Stuffing(flipped_bits);
15     full_msg = [pre_tr training_sequence start_flag stuffed_bits ' end_flag
16               post_stop];
17     size_msg = length(full_msg); %Trama con los campos incorporados
18     %Corrección del tamaño de trama tras añadir todos los campos
19     if size_msg > max_size_msg
20         max_size_msg = size_msg;

```

```

20     end
21     bit_stream(k, 1:length(full_msg)) = full_msg;
22     %NRZI
23     for h = 2:length(bit_stream(k, :))
24         if bit_stream(k, h) == 1
25             bit_stream(k, h) = bit_stream(k, h-1);
26         else
27             bit_stream(k, h) = ~ bit_stream(k, h-1);
28         end
29     end
30 end
31 bit_stream = bit_stream(:, 1:max_size_msg);

```

Código 4.8: Función *Bytes_Flipped*.

```

1 function flipped_bits = Bytes_Flipped(input_bits)
2     nbytes = length(input_bits)/8;
3     r1 = reshape(input_bits, 8, nbytes);
4     r2 = flipud(r1);
5     flipped_bits = r2(:);
6 end

```

Código 4.9: Función *Bits_Stuffing*.

```

1 function bits_out = Bits_Stuffing(bits_in)
2     ones_counter = 0;
3     bits_out = zeros(length(bits_in) + 100, 1);
4     ind_bits_out = 1;
5     for k = 1:length(bits_in)
6         if bits_in(k) == 1
7             ones_counter = ones_counter + 1;
8         else
9             ones_counter = 0;
10        end
11        if ones_counter == 5
12            bits_out(ind_bits_out) = bits_in(k);
13            bits_out(ind_bits_out + 1) = 0;
14            ind_bits_out = ind_bits_out + 2;
15            ones_counter = 0;
16        else

```

```

17         bits_out(ind_bits_out) = bits_in(k);
18         ind_bits_out = ind_bits_out + 1;
19     end
20 end
21 bits_out_end = max(length(bits_in), ind_bits_out-1);
22 bits_out = bits_out(1:bits_out_end);
23 end

```

Tras generar el paquete binario, se emplea el objeto `comm.GMSKModulator` y se pasan por parámetros el factor BT, las muestras por símbolo y la duración del pulso. Para comprobar que la modulación se realiza correctamente se utiliza el objeto `comm.ConstellationDiagram` para observar la constelación de la señal a transmitir. En la figura 4.2. Como se puede observar, la constelación resultante es idéntica a la figura 2.6. La definición de estos objetos y su uso se puede ver en la sección de código 4.10.

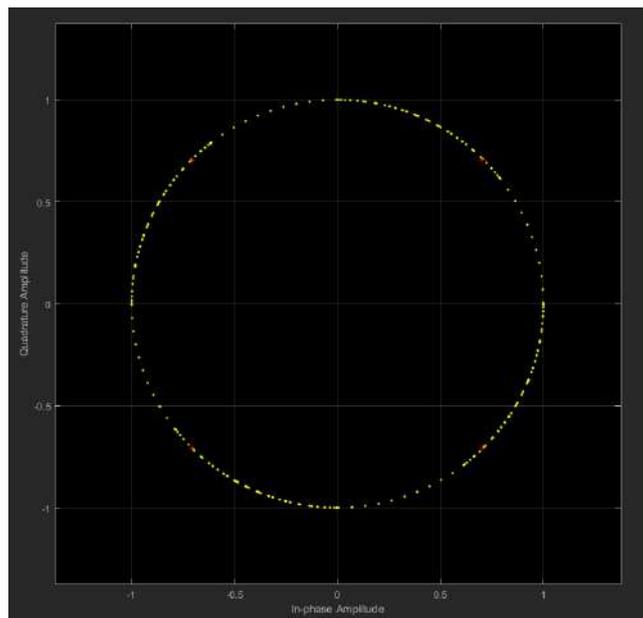


Figura 4.2: Constelación de la señal AIS a transmitir.

Código 4.10: Declaración y uso de objeto modulador y representación de constelaciones.

```

1 % Definición del objeto modulador
2 gmsk = comm.GMSKModulator('BandwidthTimeProduct', bt, ...
3     'SamplesPerSymbol', sps, ...
4     'BitInput', true, ...
5     'PulseLength', pl);
6
7 waveform_out = step(gmsk, bit_stream(1, :)).'; % Modulación de señal

```

```

8 ais_waveform = [zeros(1, sps*10) waveform_out zeros(1, sps*10)]; %Más '0's
9
10 %Definición del objeto para representar constelaciones
11 constellation_diagram = comm.ConstellationDiagram;
12 %Visualización de la constelación de la señal modulada
13 constellation_diagram(ais_waveform)

```

Otra comprobación a realizar es verificar la forma de onda de la señal generada donde se pueden ver los diferentes campos de la trama AIS. La figura 4.3 se puede ver la forma de onda de la señal. Por último, con el método *step* se ejecuta la transmite la señal generada pasando por parámetro el objeto SDR configurado y la señal modulada.

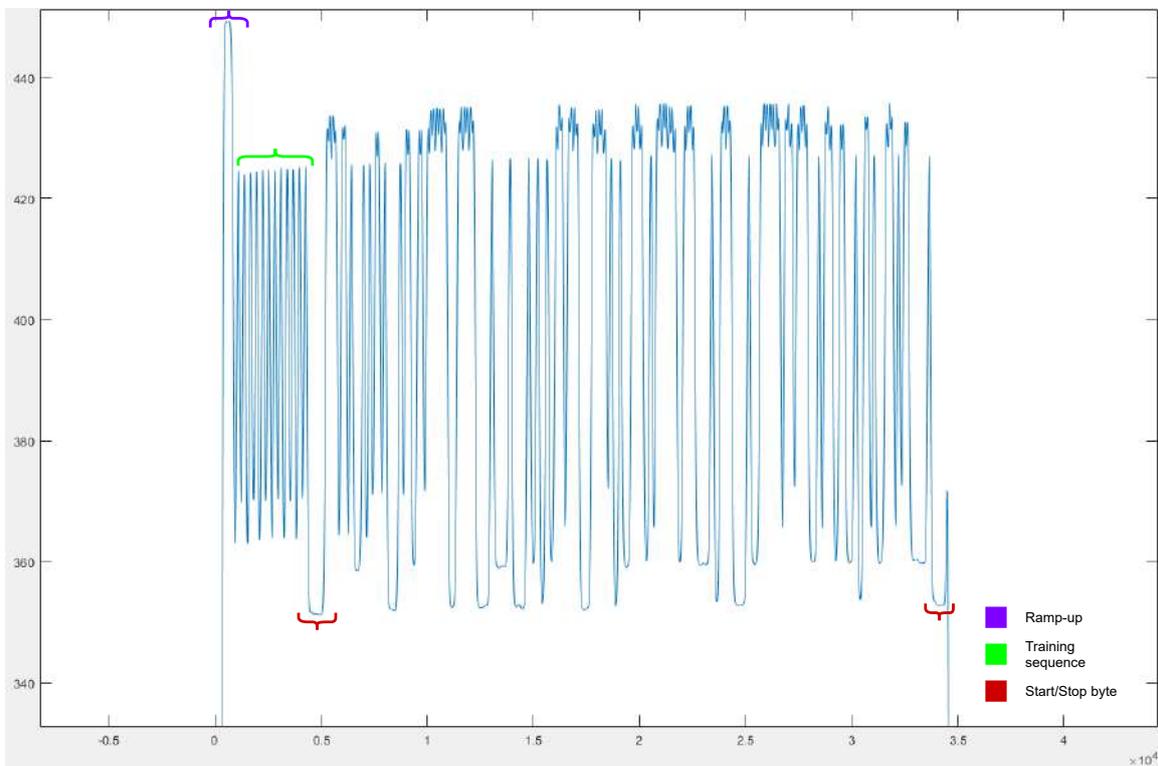


Figura 4.3: Señal AIS en el dominio temporal.

4.3. Algoritmo de recepción AIS

A diferencia del algoritmo de transmisión, toda la programación se realizó en MATLAB, dividiendo el algoritmo en pequeñas funciones. A continuación se explicarán dichas funciones y el procesamiento de la señal recibida, analizando los cambios que sufre hasta que se consigue decodificar el mensaje transmitido. Dentro de los 27 tipos de mensajes AIS, el algoritmo es capaz de procesar

los tipos 1, 2, 3, 4, 9, 11, 18, 19, 21 y 27. En la figura 4.4 se puede ver el diagrama de flujo asociado a este algoritmo.

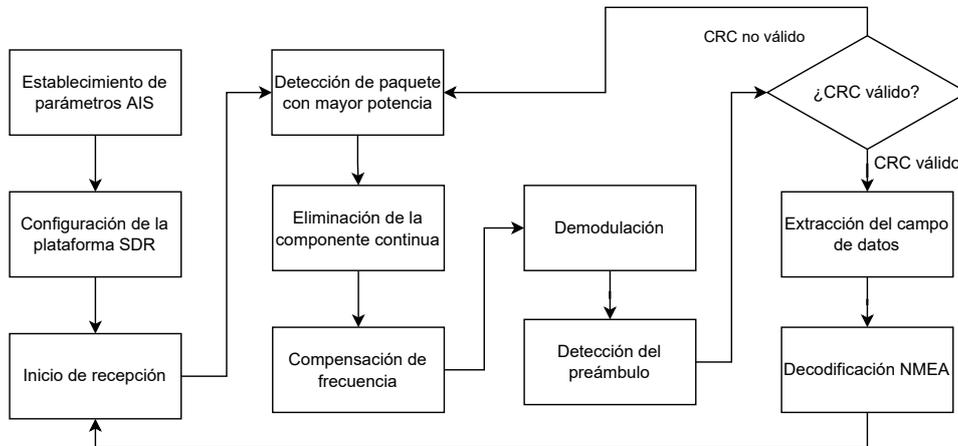


Figura 4.4: Diagrama de flujo del algoritmo de recepción de paquetes AIS.

4.3.1. Rastreo de señales AIS

Para la explicación de este algoritmo se ha utilizado un simulador de transmisiones AIS denominado QK-AT011. Este simulador permite realizar transmisiones de mensajes tipo 18 con los campos de latitud y longitud aleatorios entre los valores umbrales establecidos. Por último, el dispositivo utiliza 10 MMSI diferentes que rota entre transmisiones. Transmite con una potencia de -51 dBm con una frecuencia central de 162.025 MHz, que corresponde con el canal AIS-2. En la figura 4.5 se puede ver el dispositivo y el software de configuración del mismo.



(a) Simulador de transmisiones AIS QK-AT011.

(b) Software de configuración del dispositivo QK-AT011.

Figura 4.5: Simulador QK-AT011 y herramienta de configuración.

El primer paso de la recepción es establecer los parámetros del estándar, así como la configuración de la plataforma SDR. La configuración de la plataforma en este algoritmo utiliza más parámetros como el tamaño del buffer y el tipo de dato que tomarán las muestras IQ. Estos datos son de tipo *single*, variables de 32 de bits que se utilizan para mayor precisión. También se declaran tres variables extras, siendo estas el contador de paquetes, el contador de CRC correcto y la tasa de éxito. En la sección de código 4.11 se puede ver la declaración de variables y la configuración de la plataforma SDR en el algoritmo de recepción.

Código 4.11: Declaración y uso de objeto modulador y representación de constelaciones.

```

1 %% Definición de variables:
2 symbol_rate = 9600; % Definido en estandar
3 samples_per_symbol = 24; % Valor arbitrario , mejores resultados con 24 muestras
4
5 % Tasa de muestreo (ancho de banda de adquisicion de señales):
6 sample_rate = samples_per_symbol*symbol_rate;
7
8 % Definición de la ventana de señal útil
9 bits_size = 600;
10 max_packet_size = bits_size*samples_per_symbol;
11
12 % Definición de ventana deslizante para fft
13 window_size = 1024;
14
15 % Definición de factores del demodulador GMSK
16 bt_factor = 0.5; % Especificado en la itu 0.5
17 gaussian_span = 3; % Otro valor produce errores en el demodulador
18 pulse_length = gaussian_span;
19
20 % Definición de campos fijos en la trama
21 training_sequence = logical([1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0]);
    % 0xAAAAAA
22 start_flag = logical([0 1 1 1 1 1 1 0]); % 0x7E
23 end_flag = start_flag;
24
25 % Definición de objeto SDR en recepción:
26 rx_sdr = sdrxx('Pluto', 'RadioID', 'usb:0', ...
27             'CenterFrequency', 162e6, ... % Freq central entre canal AIS A y B
28             'GainSource', 'Manual', ...

```

```

29     'Gain', 70, ...           %-4 a 71 dB
30     'ChannelMapping', 1, ...
31     'BasebandSampleRate', sample_rate, ... %Min 65.105 KHz Max 61.44 MHz
        número bits muestreados al segundo
32     'OutputDataType', 'single', ...
33     'SamplesPerFrame', 20000, ... %Buffer de recepción. A mayor valor,
        mayor buffer, mayor probabilidad de éxito
34     'ShowAdvancedProperties', false);
35
36 %Definición de contadores
37 packet_counter = 0;
38 crc_counter = 0;
39 success_rate = 0;

```

Después de establecer las variables y configurar la plataforma SDR, ésta comienza con la recepción de señales. Una vez relleno el buffer, envía las muestras al ordenador para comenzar con el procesamiento de la señal. En la figura 4.6 se puede ver la forma de onda de la señal recibida.

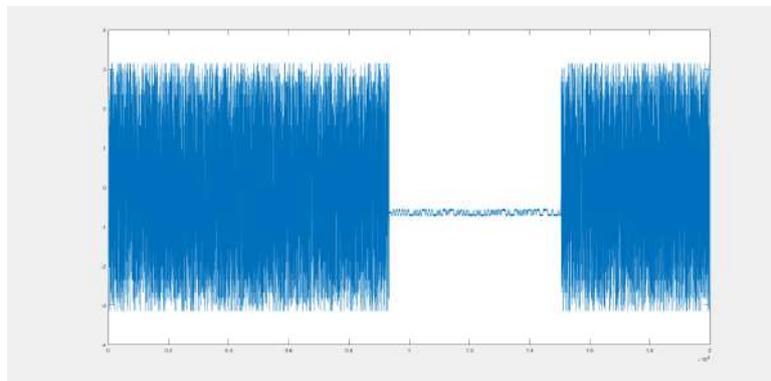


Figura 4.6: Señal AIS recibida.

Cabe destacar que esta señal está formada por 20000 muestras, que coincide con el tamaño del buffer del receptor establecido previamente, por lo que para agilizar el proceso de cómputo, se tratará de localizar el segmento de señal con mayor potencia, debido a que estadísticamente es donde hay mayor probabilidad de que exista un paquete. Para este segmento se emplea una ventana deslizante donde se calcula la media de potencia. Cuando se procesa toda la señal, se extraen los índices de inicio y fin de la ventana con mayor potencia y se recorta la señal con esos índices. Así se pasa de 20000 muestras a un tamaño bastante menor, agilizando el resto de operaciones. En la figura 4.7 se puede ver el resultado del recorte de la señal que ha pasado de 20.000 muestras a 10.875 muestras y en la sección de código 4.12 se recoge el código de la recepción de paquetes y

como se realiza el recorte de la señal.

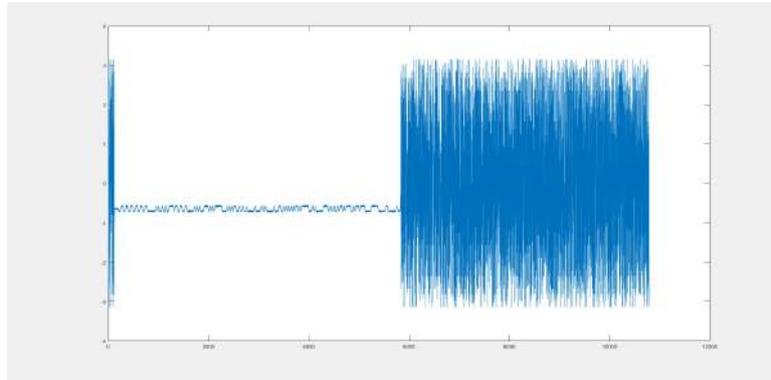


Figura 4.7: Señal AIS recibida acortada.

Código 4.12: Recepción y recorte de señal.

```

1 %% Recepcion de señales AIS:
2 ais_signal_rx = step(rx_sdr);
3
4 %% Deteccion de paquetes AIS en señales recibidas:
5 [ais_signal_rx_trim , ais_signal_rx_boundary , max_Mag] = AIS_RX_Packet_Search(
    ais_signal_rx , window_size , max_packet_size);
6
7 function [rxsignalTrim , rxsignalBoundary , maxMag] = AIS_RX_Packet_Search(
    rxsignal , windowLength , maxPacketLength)
8 %Función para buscar la ventana con mayor potencia de la señal recibida ,
9 %diviando la señal en diferentes contenedores.
10 %rxsignal: señal recibida por la plataforma
11 %windowLength: tamaño de ventana o bin , valores tipicos --> 512, 1024...
12 %maxPacketLength: tamaño máximo definido para los paquetes
13 %rxsignalTrim: devuelve la señal recortada , solo la ventana de mayor potencia.
14 %rxsignalBoundary: indices de inicio y final de la ventana.
15     %Paso 1: dividir las señales recibidas en varias ventanas
16     rx_windows = floor(length(rxsignal)/windowLength);
17     %Paso 2: buscar la ventana de mayor potencia
18     winMag = sum(reshape(abs(rxsignal(1:rx_windows*windowLength)) , windowLength ,
    rx_windows));
19     diffMag = diff(winMag);
20     [maxMag, indMax] = max(diffMag);
21     meanMag = mean(abs(diffMag));
22     indMax = max(2, indMax);
23     indMax = min(indMax, rx_windows-2);

```

```

24     if maxMag > meanMag
25         tmp1 = diffMag(indMax) - diffMag(indMax - 1);
26         tmp2 = diffMag(indMax) - diffMag(indMax + 1);
27         if tmp1 > tmp2
28             indStart = indMax*windowLength;
29         else
30             indStart = max(1, (indMax-1)*windowLength);
31         end
32     else
33         indStart = 1;
34     end
35     %Paso 3: tomar la ventana de mayor potencia en función del índice
36     %calculado previamente
37     indEnd = min(length(rxsignal), indStart + maxPacketLength);
38     rxsignalBoundary = [indStart indEnd];
39     rxsignalTrim = rxsignal(indStart:indEnd);
40     maxMag2 = typecast(maxMag, 'uint16');
41 end

```

En el caso de que la potencia media de la señal recortada no supere el umbral establecido, se descarta la señal y se procede a recibir más muestras. El umbral varía respecto a la ganancia que se le establece a la plataforma, ya que hay que considerar que cuanto mayor su ganancia, la potencia del ruido aumentará. Ya con la señal recortada, se le elimina la componente continua, debido a que las plataformas SDR trabajan en banda base y puede entorpecer el resto de operaciones. Para eliminar la componente continua se le resta a la señal recortada su media. En la sección de código 4.13 se recoge el código utilizado para establecer el umbral y la eliminación de la componente continua.

Código 4.13: Establecimiento del umbral de potencia y eliminación de la componente continua.

```

1  %Umbral de potencia, diferencia cuando hay señal o no
2  if max_Mag > 20.0
3  %% Filtro DC para eliminar la componente continua de la señal AIS:
4  ais_signal_rx_rdc = Remove_DC_Offset(ais_signal_rx_trim);
5
6  function rxsignal_removeDC = Remove_DC_Offset(rxsignal)
7  %Función para eliminar la componente de continua de la señal recibida
8  %rxsignal: señal recibida de la ventana de mayor potencia.
9  %rxsignal_removeDC: señal recibida de la ventana de mayor potencia sin

```

```

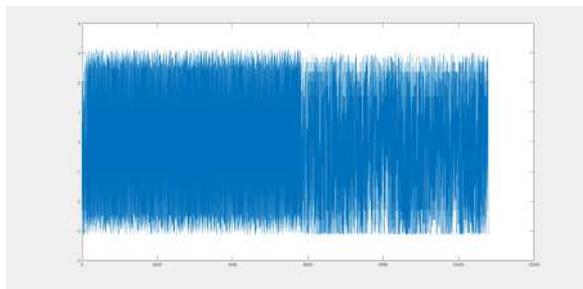
10     componente continua.
11     rxsignal_removeDC = rxsignal - mean(rxsignal);
12 end

```

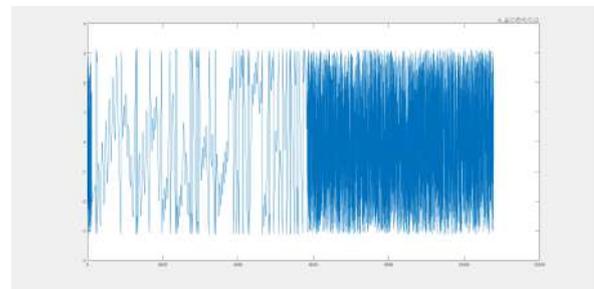
El último paso de esta sección es la compensación de frecuencia [52] a la señal recortada y con la componente continua eliminada. Para ello, se calcula el desfase que sufre la señal debido a los componentes internos de la plataforma (principalmente el desfasador para generar las muestras IQ), frecuencias Döppler y otros fenómenos. La solución es estimar el valor de compensación que restaure la periodicidad de la señal. Por lo tanto, se busca la muestra con mayor potencia espectral y se calcula el desfase con las muestras vecinas. Luego se aplica la fórmula 4.1 [52] para conseguir la señal compensada. En la figura 4.8 se puede ver como cambia la señal al realizar la compensación de frecuencia. En la sección de código 4.14 se recoge la implementación de la compensación de frecuencia.

$$z(k) = e^{-j2\pi f_0 k} \cdot x(k), \quad (4.1)$$

donde $x(k)$ es la señal recibida a la que se le realizará la compensación de frecuencia.



(a) Diagrama de fase previo a la compensación de frecuencia.



(b) Diagrama de fase posterior a la compensación de frecuencia.

Figura 4.8: Resultado de la compensación de frecuencia de la señal AIS.

La señal en el dominio temporal se ha modificado ligeramente, como se puede ver en la figura 4.9. En la siguiente sección se explica la demodulación, decodificación y extracción de datos de la señal AIS.

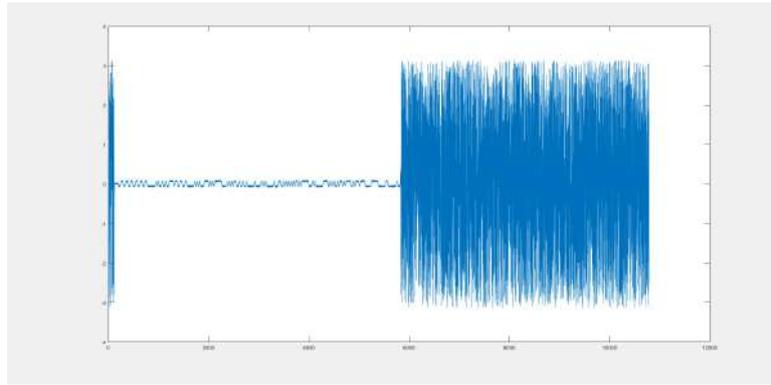


Figura 4.9: Señal AIS recortada, con componente continua eliminada y compensada en frecuencia.

Código 4.14: Compensación de frecuencia de la señal recibida.

```

1 %%Compensación de la desviación de frecuencia de la señal AIS:
2 ais_signal_rx_cfo = Frequency_Offset_Compensation(ais_signal_rx_rdc , sample_rate
   );
3
4 function rx_signal_cfo = Frequency_Offset_Compensation(rxsignal , sample_rate)
5 %Función para compensar la frecuencia de la señal recibida por el SDR
6 %rxsignal: señal recibida de la ventana con mayor potencia sin componente
   continua
7 %sample_rate: tasa de muestreo especificada samples_per_symbol*symbol_rate
8 %rx_signal_cfo: señal recibida de la ventana con mayor potencia sin componente
   continua y compensada en frecuencia.
9 %Estimar la desviación de frecuencia:
10 rx_signal_fft = abs(fftshift(fft(rxsignal)));
11 ind_max = find(rx_signal_fft == max(rx_signal_fft));
12 rx_freq_shift = floor(length(rxsignal)/2 - ind_max)*(sample_rate)/length(
   rx_signal_fft);
13 rx_signal_cfo = rxsignal.*exp(1j*2*pi*(rx_freq_shift/sample_rate)*(0:length(
   rxsignal) - 1)).';
14 end

```

4.3.2. Demodulación y decodificación de paquetes AIS

A diferencia del algoritmo de transmisión, la demodulación se ha programado sin el uso del objeto `comm.GMSKDemodulator`, ya que el bloque no contiene un decisor, importante a la hora de demodular la señal. El primer paso de este proceso es sincronizar las muestras y para ello, se

utiliza el preámbulo de la trama. El sincronismo de muestras se consigue modulando el preámbulo para extraer la fase de los símbolos modulados y la posición del campo de datos. Luego se calcula cuál es la mejor muestra, para posteriormente, diezmar la señal con un factor igual a las muestras por símbolos. El diezmo se realiza seleccionando la posición de la mejor muestra y guarda los valores del resto del vector dando saltos equivalentes al número de muestras por símbolo. Con esto se consigue el número original de bits de la trama enviada. Sin embargo, aún queda un proceso extra y es la decisión del valor binario de la muestra. Para ello se compara el valor de fase de las muestras de la señal diezmada y, si esta es mayor a $\pi/4$ radianes, se considera un '1' binario. En caso contrario se asigna un '0' binario. Con esto termina el proceso de demodulación. En la figura 4.10 se puede ver como la señal demodulada ha sido diezmada, pasando de 10.875 a 577 muestras. En la sección de código 4.15 se recoge la implementación completa de la demodulación de la señal, junto al sincronismo y diezmo.

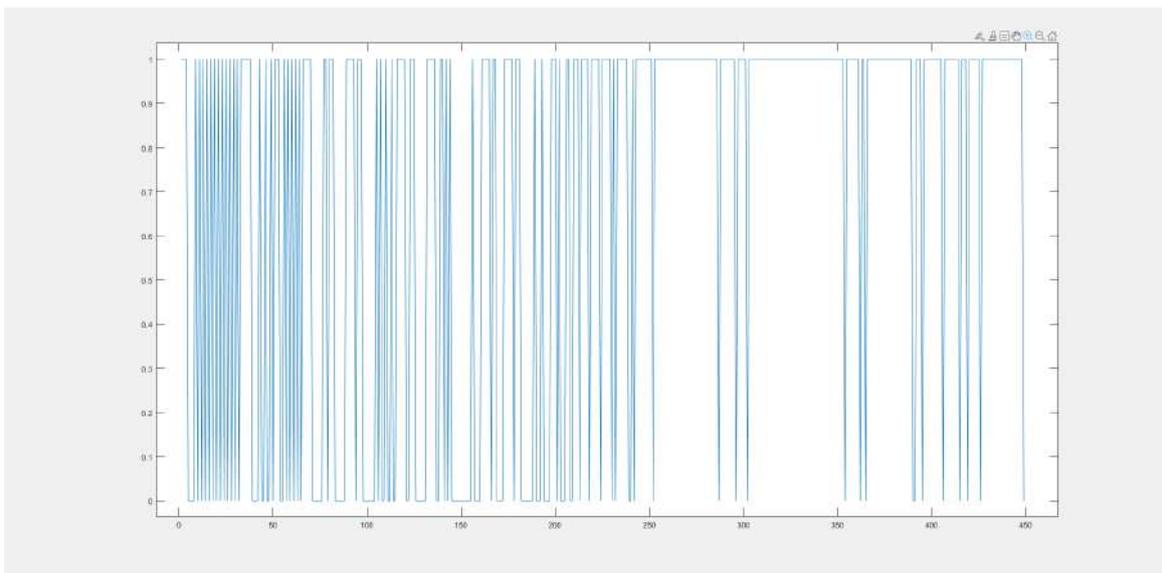


Figura 4.10: Señal AIS demodulada.

Código 4.15: Establecimiento del umbral de potencia y eliminación de la componente continua.

```

1 %% Demodulación GMSK y decodificación NRZI
2 [ais_dec_bits] = AIS_RX_Sync_Demodulation(ais_signal_rx_cfo , bt_factor ,
    samples_per_symbol , pulse_length , training_sequence);
3
4 function [demod_bits , rx_downsample , samp_phs] = AIS_RX_Sync_Demodulation(
    rxsignal , bt , sps , pl , tr_seq)
5 % Función para demodular la señal recibida con modulación GMSK, sincroniza la
    señal, detecta el preambulo y realiza un diezmado con el valor de
    samples_per_symbol (sps)
6 % rxsignal: señal de la ventana de mayor potencia, sin componente continua y
    compensada en frecuencia
7 % bt: factor BT del filtro gaussiano
8 % sps: samples_per_symbol, cuantas muestras se requieren para conformar un
    símbolo, factor de diezmado
9 % pl: pulse length, la duración del pulso, igual al ancho del filtro gaussiano
    especificado.
10 % tr_seq: training sequence, secuencia de '1's y '0's intercalados que forma el
11 % preambulo [1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0]
12 % demod_bits: señal demodulada y diezmada
13     % Sincronizar en base a los angulos:
14     [rx_downsample , samp_phs] = Sync_Phase(rxsignal , bt , sps , pl , tr_seq);
15     % Decision de bit:
16     demod_bits = zeros(size(rx_downsample));
17     ind = find(abs(diff(rx_downsample)) > pi/4); %pi/4
18     demod_bits(ind) = 1;
19     end
20
21 function [rx_downsample , sample_phase] = Sync_Phase(rxsignal , bt , sps , pl ,
    tr_seq)
22 % Función de sincronización de la señal mediante los angulos de la fase y
    diezmado.
23 % rxsignal: señal de la ventana de mayor potencia, sin componente continua y
    compensada en frecuencia
24 % bt: factor BT del filtro gaussiano
25 % sps: samples_per_symbol, cuantas muestras se requieren para conformar un
    símbolo, factor de diezmado
26 % pl: pulse length, la duración del pulso, igual al ancho del filtro gaussiano
    especificado.

```

```

27 %tr_seq: training sequence, secuencia de '1's y '0's intercalados que forma el
28 %preambulo [1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0]
29 %rx_downsample: señal demodulada y diezmada
30     %Angulos de la señal recibida:
31     rx_angles = unwrap(angle(rxsignal));
32     %Correlación para encontrar el indice inicial modulando el preámbulo:
33     gmsk = comm.GMSKModulator('BandwidthTimeProduct', bt, ...
34                               'SamplesPerSymbol', sps, ...
35                               'BitInput', true, ...
36                               'PulseLength', pl)
37     sync_symbols = gmsk(tr_seq');
38     sync_angles = unwrap(angle(sync_symbols));
39     correlation_window_size = 2*length(sync_angles);
40     %Correlacion respecto a un preambulo conocido:
41     if length(rx_angles) > correlation_window_size
42         [rx_acor, lag]=xcorr(sync_angles, rx_angles(1:correlation_window_size));
43         [~,I] = max(abs(rx_acor));
44         lag_diff = lag(I);
45         ind = lag_diff;
46         rx_ind = -ind + 1;
47     else
48         rx_ind = 1;
49     end
50     %Mejor fase para tomar decisiones de bits:
51     sample_phase = mod(rx_ind, sps) + floor(sps/2);
52     %Diezmado de la señal a 1 muestra por símbolo
53     rx_downsample = rx_angles(sample_phase:sps:end);
54 end

```

El último paso antes de extraer los datos de la señal es la decodificar, por lo que se ha de realizar los procesos inversos vistos en el algoritmo de transmisión. En primer lugar, se debe localizar donde comienza el campo de datos. Como se vio anteriormente, el campo de datos comienza justo después del *flag* de inicio, por lo que se busca esta secuencia dentro de la señal y se extrae la posición del último bit. Luego, se recorta la señal eliminando el *flag* de inicio, dejando datos y *flag* de fin. Con este nuevo vector se calcula su longitud, y de ser menor a 24 bits se descarta automáticamente. En la sección de código 4.16 se implementa la búsqueda del *flag* de inicio y el cálculo de la longitud del campo de datos. Seguidamente, se extrae el tipo de mensaje recibido.

Si el tipo de mensaje coincide con los tipos previamente mencionados, se calcula donde se sitúa el final del campo de datos y se retiran los bits de relleno para ejecutar el detector de CRC y comprobar que el paquete recibido no contiene errores. En el caso de que se detecte un error se descarta el paquete automáticamente. Por último, se decodifica la trama mediante NRZI para poder extraer los campos MMSI, longitud y latitud. Estos datos se encuentran en hexadecimal, por lo que es necesario convertirlos en caracteres para que sean legibles. En la figura 4.11 se puede ver el resultado de la decodificación y extracción de los campos de la trama AIS. En la sección de código 4.17 se recoge la implementación de la extracción de datos de un mensaje tipo 18.

Código 4.16: Búsqueda del *flag* de inicio y cálculo de la longitud del campo de datos.

```

1 %% Sincronización con start byte y decodificación de los datos del mensaje
2 [ais_packet_received, packet_counter, crc_counter, mmsi, lat, long] =
   AIS_RX_Bit_Parser(ais_dec_bits, packet_counter, crc_counter);
3 mmsi
4 long
5 lat
6 packet_counter
7 crc_counter
8 success_rate = crc_counter/packet_counter
9
10 function [payload_bits, packet_cnt, crc_cnt, mmsi, lat, lon] = AIS_RX_Bit_Parser
   (dec_bits, packet_cnt, crc_cnt)
11 % Función que decodifica la señal demodulada en NRZI y comprueba el campo CRC en
   busca de un paquete válido
12 % dec_bits: señal recibida demodulada y diezmada
13 % packet_cnt: variable donde se almacenan la cantidad de paquetes recibidos
14 % crc_cnt: variable donde se almacenan la cantidad de paquetes cuyo CRC es
   válido.
15 % packet_rx: paquete decodificado en ASCII
16 % packet_cnt: variable donde se almacenan la cantidad de paquetes recibidos
17 % crc_cnt: variable donde se almacenan la cantidad de paquetes cuyo CRC es
   válido.
18 max_message_bytes = 90; % Longitud máxima del mensaje en bytes
19 tmp.MessageBytes = char(zeros(1, max_message_bytes));
20 tmp.MMSI = 0;
21 tmp.Longitude = 0;
22 tmp.Latitude = 0;
23 % Búsqueda del flag de inicio (0x7E)

```

```

24 %start_flag = [0 1 1 1 1 1 1 0].'; %original
25 start_flag = [0 1 1 1 1 1 1 0].';
26 start_bit = 1;
27 for k = 1:length(dec_bits)-7 %7
28     rx_sf = dec_bits(k:k+7); %7
29     if isequal(rx_sf, start_flag)
30         start_bit = k+8; %8
31         break;
32     else
33         start_bit = 1;
34     end
35 end
36 %Descarte del flag de inicio y cálculo de la longitud del campo de datos
37 payload_bits = dec_bits(start_bit:end);
38 if(length(payload_bits) < 24) %16-bit CRC and 8-bit end flag
39     valid_crc = false;
40     ubits_trim = zeros(480, 1);
41     msg_id = 0;
42 else

```



```

Command Window
mmsi =
    316013198

long =
    -4

lat =
    6.0002

success_rate =
    1
fx

```

Figura 4.11: Decodificación y extracción de campos de la trama AIS.

Código 4.17: Decodificación de los datos de la trama AIS recibida.

```

1 %Decodificar el tipo de mensaje
2 msg_id = comm.internal.utilities.bi2deRightMSB(payload_bits(3:8).',2);
3 %Quitar relleno de bits
4 unstuff_bits = Bits_Unstuffing(payload_bits);
5 temp_id_set = [1, 2, 3, 4, 9, 11, 18];
6 %Definir el índice de final de payload

```

```

7  switch msg_id
8      case temp_id_set(temp_id_set == msg_id)
9          end_index = 184;
10     otherwise
11         end_index = length(unstuff_bits);
12     end
13     %Recortar el paquete hasta el final del payload
14     if(end_index > length(unstuff_bits))
15         end_index = length(unstuff_bits);
16     end
17     ubits_trim = unstuff_bits(1:end_index);
18     %Deteccion de CRC
19     crc_detection=comm.CRCDetector('Polynomial',[1 0 0 0 1 0 0 0 0 0 0 1 0 0 0 0
20         1],...
21         'InitialConditions', 1, ...
22         'DirectMethod', true, ...
23         'FinalXOR', 1);
24     try
25         [~, err] = crc_detection(ubits_trim(1:end_index));
26         valid_crc = ~err;
27     catch
28         err = 0;
29         valid_crc=0;
30     end
31     %Decodificación de los bits de datos y aumento del contador CRC
32     if (valid_crc)
33         %Desvoltar los bits de datos
34         if(rem(length(ubits_trim), 8) ~= 0)
35             dummy = zeros(1, 8-rem(length(ubits_trim), 8)).';
36             ubits_trim = [ubits_trim; dummy];
37         end
38     flipped_bits = Bytes_Flipped(ubits_trim);
39     tmp.MessageBytes = Bits_to_Hex(flipped_bits);
40     tmp.MMSI = comm.internal.utilities.bi2deLeftMSB(flipped_bits(9:38).',2);
41     pos_res_factor_1 = 10000; %Precision with a resolution of 1/10000 min of the
42         value
43     pos_res_factor_2 = 10; %Precision with a resolution of 1/10 min of the value
44     seconds_per_min = 60;

```

```

44 scale_factor_1 = pos_res_factor_1*seconds_per_min;
45 scale_factor_2 = pos_res_factor_2*seconds_per_min;
46 switch msg_id
47     case 18
48         tmp.Longitude = (-2^27*flipped_bits(58)+2.^(26:-1:0)*flipped_bits(59:85)
49             )/scale_factor_1;
50         tmp.Latitude = (-2^26*flipped_bits(86)+2.^(25:-1:0)*flipped_bits(87:112)
51             )/scale_factor_1;
52     otherwise
53         tmp.MMSI = 0;
54         tmp.Longitude = 0;
55         tmp.Latitude = 0;
56 end
57 crc_cnt = crc_cnt + 1;
58 %En caso de que el CRC sea incorrecto
59 else
60     tmp.MessageBytes = char(zeros(1, 90));
61     tmp.MMSI = 0;
62     tmp.Longitude = 0;
63     tmp.Latitude = 0;
64 end
65 if(valid_crc)
66     size_message = length(tmp.MessageBytes);
67     message_char = tmp.MessageBytes.';
68     info.MessageBytes(crc_cnt, 1:size_message) = message_char(1, 1:size_message)
69     ;
70     info.MMSI(crc_cnt) = tmp.MMSI;
71     info.Longitude(crc_cnt) = tmp.Longitude;
72     info.Latitude(crc_cnt) = tmp.Latitude;
73     packet_rx = message_char;
74 else
75     packet_rx = ' ';
76 end
77 %Incrementar el contador de paquetes
78 packet_cnt = packet_cnt + 1;
79 lon = tmp.Longitude;
80 lat = tmp.Latitude;
81 mmsi = tmp.MMSI;
82 end

```

```
80
81 %Funciones extras utilizadas
82
83 function [output_bits] = Bits_Unstuffing(input_bits)
84 %Función para descartar los bits de relleno existentes en la señal recibida
85 %input_bits: señal recibida
86     %Se descartan los bits de relleno:
87     cnt_ones = 0;
88     input_bits = double(input_bits);
89     output_bits = zeros(length(input_bits), 1);
90     ind_output_bits = 1;
91     for k = 1:length(input_bits)
92         if cnt_ones < 5
93             output_bits(ind_output_bits) = input_bits(k);
94             ind_output_bits = ind_output_bits + 1;
95         end
96         if input_bits(k) == 1
97             cnt_ones = cnt_ones + 1;
98         else
99             cnt_ones = 0;
100        end
101    end
102    ind_output_bits = min(length(input_bits), ind_output_bits);
103    output_bits = output_bits(1:ind_output_bits);
104 end
105
106 function data_hex = Bits_to_Hex(data_bit)
107     %Convertir bits en caracteres hexadecimales:
108     ndata = floor(length(data_bit)/4);
109     data_hex = char(zeros(ndata, 1));
110     for k = 1:ndata
111         bit = data_bit((k-1)*4 + 1:k*4);
112         hex = dec2hex(2.^(3:-1:0)*bit);
113         data_hex(k) = hex;
114     end
115 end
```

4.3.3. Algoritmo de transmisión VDES

Para el estándar VDES se ha programado exclusivamente el algoritmo de transmisión, debido a que no existen muchos dispositivos comerciales capaces de transmitir acorde a este sistema con los que verificar la eficacia del algoritmo de recepción. En el mismo algoritmo se diferencian los servicios descritos previamente, y cada uno cuenta con sus propios parámetros. Los servicios implementados son el AIS y LR-AIS (ya que cuentan con la misma estructura de trama y ya está definido completamente, y se agrupan en uno), ASM y VDE-TER con los diferentes anchos de banda de transmisión y modulaciones asociados. La explicación del algoritmo seguirá el orden descrito por el diagrama de flujo representado en la figura 4.12.

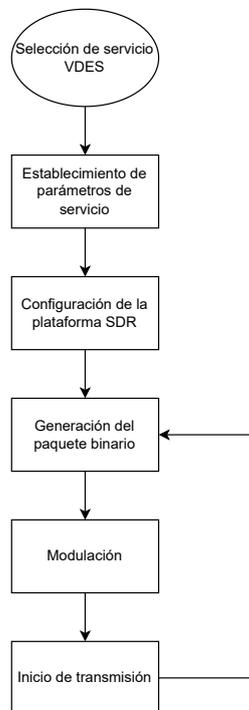


Figura 4.12: Diagrama de flujo del algoritmo de recepción de paquetes VDES.

4.4. Selección de servicio y configuración de parámetros VDES

El inicio del algoritmo consiste en una estructura switch-case que sirve para seleccionar el servicio que se pretende transmitir. Los parámetros de cada servicio son diferentes, por lo que se deben configurar tras el proceso de selección. Las muestras por símbolo dependen de la modulación empleada, siendo para $\pi/4$ -QPSK de 2 muestras por símbolo, 8-PSK de 3 muestras por símbolo y 16-QAM de 4 muestras por símbolo.

Las tasas de símbolos están establecidas en [11], siendo para canales de 25 KHz de 19,2 kbps, para 50 KHz de 38,4 kbps y para canales de 100 KHz de 76,8 kbps. Sin embargo, el producto entre las muestras por símbolo y la tasa de símbolos no siempre es mayor a 65 KHz (mínimo valor de tasa de transmisión que soporta el Pluto). Cuando esto sucede, se ajusta la tasa de transmisión para que cumpla con la norma. Para resumir la sección de código a mostrar 4.18, se recoge la implementación de los servicios VDE-TER con modulación y ancho de banda diferentes, además, el servicio ASM comparte las características del servicio VDE-TER con modulación $\pi/4$ -QPSK. Para el caso del AIS se llaman a los algoritmos previamente mostrados.

Código 4.18: Definición de servicios VDES.

```

1   training_sequence = logical([1 1 1 1 1 1 0 0 1 1 0 1 0 1 0 0 0 0 0 1 1 0 0 1
   0 1 0]);
2   signal_information_fec = logical([0 0 0 0 0 0 0]); %0000 = sin codificación ->
   Hamming (7,4) otros casos definen una tasa de código (0001 = 1/2, 000 = 3/4,
   0011 = 5/6)
3   crc = logical(randi([0 1], 32, 1)');
4   disp('A continuación, pulse una de las siguientes teclas para seleccionar el
   servicio a transmitir \n')
5   disp('VDE-TER 25 KHz con modulación pi/4 QPSK --> 1 \n')
6   disp('VDE-TER 25 KHz con modulación 8PSK --> 2 \n')
7   disp('VDE-TER 25 KHz con modulación 16QAM --> 3 \n')
8   disp('VDE-TER 50 KHz con modulación pi/4 QPSK --> 4 \n')
9   disp('VDE-TER 50 KHz con modulación 8PSK --> 5 \n')
10  disp('VDE-TER 50 KHz con modulación 16QAM --> 6 \n')
11  disp('VDE-TER 100 KHz con modulación pi/4 QPSK --> 7 \n')
12  disp('VDE-TER 100 KHz con modulación 8PSK --> 8 \n')
13  disp('VDE-TER 100 KHz con modulación 16QAM --> 9 \n')
14  disp('AIS --> 10 \n')
15  disp('ASM --> 11 \n')
16  key = input('Elija el modo de transmisión');
17  switch key
18      case 1
19          %25 KHz pi/4-QPSK
20          samples_per_symbol = 5;
21          symbol_rate = 19200;
22          sample_rate = 921600; %samples_per_symbol (48) * symbol_rate (19200),
   valor arbitrario para ajustar
23          tx_sdr = sdrtx('Pluto', ...

```

```

24         'RadioID', 'usb:0', ...
25         'CenterFrequency', 162e6, ...
26         'BasebandSampleRate', sample_rate, ...
27         'Gain', 0);
28     qpsk = comm.QPSKModulator('PhaseOffset', pi/4, ...
29         'BitInput', true, ...
30         'SymbolMapping', 'Binary', ...
31         'OutputDataType', 'double');
32     case 5
33         % 50 KHz 8-PSK
34         samples_per_symbol = 8;
35         symbol_rate = 38400;
36         sample_rate = 921600;
37         tx_sdr = sdrtx('Pluto', ...
38             'RadioID', 'usb:0', ...
39             'CenterFrequency', 162e6, ...
40             'BasebandSampleRate', sample_rate, ...
41             'Gain', 0);
42         psk = comm.PSKModulator('ModulationOrder', 8, ...
43             'PhaseOffset', 0, ...
44             'BitInput', true, ...
45             'SymbolMapping', 'Binary', ...
46             'OutputDataType', 'double');
47     case 9
48         % 100 KHz 16-QAM
49         samples_per_symbol = 4;
50         symbol_rate = 19200;
51         sample_rate = samples_per_symbol*symbol_rate;
52         tx_sdr = sdrtx('Pluto', ...
53             'RadioID', 'usb:0', ...
54             'CenterFrequency', 162e6, ...
55             'BasebandSampleRate', sample_rate*1.3, ...
56             'Gain', 0);
57     otherwise
58         disp('Tecla incorrecta, terminando el programa')
59
60 end

```

Como se puede ver en las secciones de los servicios menores a 100 KHz, el ancho de banda de

transmisión es un valor fijo y arbitrario, ya que la plataforma no es capaz de emplear anchos de banda menores a 65 KHz. Se han ajustado los valores de la variable *samples_per_symbol* para que las transmisiones de datos ocupen el ancho de banda establecido en el estándar [11]. El resultado de las transmisiones se muestran en el capítulo 7

4.4.1. Generación aleatoria del paquete binario

Antes de generar los campos necesarios para formar la trama de los servicios VDES, se debe tener en cuenta su longitud. Para ello, se acude a la norma ITU M.2092-1 [53] (versión actualizada y publicada en febrero de 2022) donde se definen los tamaños de los campos de cada trama. Los campos del servicio ASM se encuentran en la tabla 7 y los campos del servicio VDE-TER en la tabla 8 y son recogidos en la tabla 4.1. Conociendo los valores de los campos se generan vectores de binarios aleatorios con los tamaños estipulados, para así simular una trama ASM o VDE-TER.

Tabla 4.1: Campos y tamaños asociados para los servicios ASM y VDE-TER.

Campo del servicio VDES	Tamaño en símbolos
ASM	
Preámbulo	27 (111111001101010000011001010)
Datos + CRC	Variable
VDE-TER	
Ramp-up	4
Preámbulo	27 (111111001101010000011001010)
Identificador de enlace	16
Datos	1 slot: 176 2 slot: 432 3 slot: 688
CRC	16
Bits finales FEC	6
Intervalo de guardia	7
Total	1 slot: 256 2 slot: 512 3 slot: 768

En la sección de código 4.19 se muestra la generación de mensajes aleatorias por servicios en el mismo orden que se ha mostrado anteriormente. Para los casos de canales de 100 KHz no es necesario añadir redundancia ya que se ocupa todo el ancho de banda de transmisión para el envío de datos. La función *VDE_TX_Message_Generator* añade redundancia repitiendo el campo de datos tantas veces como muestras por símbolos se haya definido, para que al realizar la modulación se mantenga el mismo número de símbolos que bits declarados, cumpliendo así con el estándar.

Por último, se realiza un proceso previo a la modulación que añade tantos 0 como sean necesarios para que el número de bits totales a modular sea divisible entre el orden de la modulación.

Código 4.19: Generación de paquetes VDES.

```

1 %VDE-TER 25 KHz modulación pi/4-QPSK
2 ramp_up = logical(randi([0 1], 16, 1))'; %Para 25 KHz 8 símbolos, igual que
   ramp_down (16 para simplificar)
3 data_length = logical([0 1 1 0 0 1 1 1 0 0]);
4 data = logical(randi([0 1], 400, 1))'; %Para 25 KHz, 432 símbolos con crc,
   tamaño fijo por falta de codificación en FEC
5 buffer = logical(randi([0 1], 30, 1))'; %Para 25 KHz, 30 símbolos de buffer
6 data2 = [ramp_up data_length data crc buffer];
7 vdes_message = VDE_TX_Message_Generator(samples_per_symbol, training_sequence,
   signal_information_fec, data2, 'qpsk');
8
9 %VDE-TER 50 KHz modulación 8-PSK
10 ramp_up = randi([0 1], 32, 1)'; %Para 50 KHz 32 símbolos no se modula
11 data_length = [0 1 1 0 0 1 1 1 0 0];
12 data = randi([0 1], 896-32, 1)'; %Para 50 KHz, 896 símbolos con crc, tamaño
   fijo por falta de codificación en FEC
13 buffer = randi([0 1], 62, 1)'; %Para 50 KHz, 62 símbolos de buffer
14 data2 = [ramp_up data_length data crc buffer];
15 vdes_message = VDE_TX_Message_Generator(samples_per_symbol, training_sequence,
   signal_information_fec, data2, 'psk');
16
17 %VDE-TER 100 KHz modulación 16-QAM
18 ramp_up = randi([0 1], 64, 1)'; %Para 100 KHz 64 símbolos no se modula
19 data = randi([0 1], 1792-32, 1)'; %Para 100 KHz, 1792 símbolos con crc,
   tamaño fijo por falta de codificación en FEC
20 buffer = randi([0 1], 158, 1)'; %Para 100 KHz, 158 símbolos de buffer
21 vde_ter_message = [ramp_up training_sequence signal_information_fec data crc
   buffer];
22
23 %Función común utilizada
24 function vde_waveform = VDE_TX_Message_Generator(sps, trs, sf, msg, modulation)
25     training_sequence = trs;
26     training_sequence_inv = ~training_sequence;
27     start_flag = sf;
28     end_flag = start_flag;

```

```

29     full_msg = logical([training_sequence start_flag msg end_flag]);
30     msg2 = repmat(full_msg,1,sps); %Repetir el vector de datos para modificar el
        ancho del pulso de la señal a transmitir
31     full_msg = logical([zeros(1, 48*100) msg2 zeros(1, 48*100)]); %48 valor
        arbitrario con el que se hicieron pruebas
32     size_msg = length(full_msg);
33     max_size_msg = size_msg*4;
34     bit_msg = msg;
35     bit_stream = false(1, length(start_flag) + length(training_sequence) + ...
36         length(end_flag) + size_msg*4 + 100);
37 %Acomodar el mensaje para que pueda ser modulado y enviado
38     if size_msg < max_size_msg
39         max_size_msg = size_msg;
40         bit_stream(1, 1:length(full_msg)) = full_msg;
41         for h = 2:length(bit_stream(1, :))
42             if bit_stream(1, h) == 1
43                 bit_stream(1, h) = bit_stream(1, h-1);
44             else
45                 bit_stream(1, h) = ~ bit_stream(1, h-1);
46             end
47         end
48     end
49 %Añadir 0 si fuera necesario hasta que la longitud total del mensaje sea un
        numero divisible por la modulación deseada
50     switch modulation
51         case 'qpsk'
52             if(mod(size(bit_stream), 2) ~=0)
53                 bit_stream = [bit_stream 0];
54             end
55         case 'psk'
56             while(mod(size(bit_stream), 3) ~=0)
57                 bit_stream = [bit_stream 0];
58             end
59         case 'qam'
60             while(mod(size(bit_stream), 4) ~=0)
61                 bit_stream = [bit_stream 0];
62             end
63     end
64     vde_waveform = bit_stream;

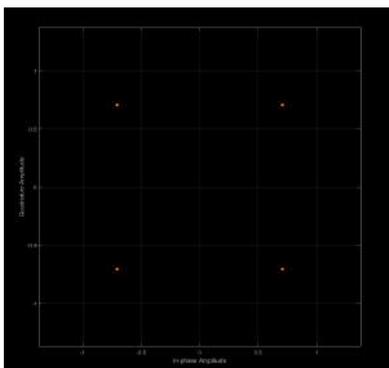
```

65 end

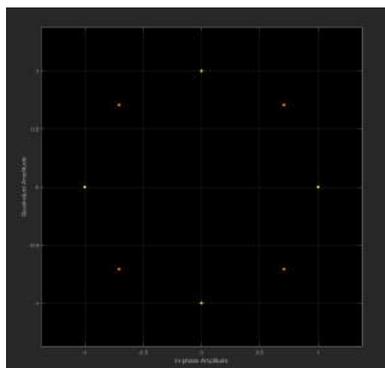
4.4.2. Modulación y transmisión de paquetes

Con las tramas ya generadas, se emplean los objetos moduladores `comm.QPSKModulator` y `comm.PSKModulator`, junto al método `qammod`. Para generar una modulación $\pi/4$ -QPSK, se pasa por parámetro el valor de desfase ($\pi/4$ radianes en este caso), para la modulación 8-PSK se pasa por parámetro el orden de modulación (8 en este caso) y un desfase de 0 radianes, y en el último caso, la modulación 16-QAM se pasa por parámetro la señal de datos, el orden de modulación (16 en este caso) y se especifica que los valores de entrada son binarios. Para que realice la modulación se ejecuta el comando `step` con el objeto modulador y la señal a modular para los casos de $\pi/4$ -QPSK y 8-PSK, y se guarda la señal resultante en una variable, mientras que el método de la modulación 16-QAM no es necesario, ya que devuelve la señal modulada sin la necesidad de crear objetos.

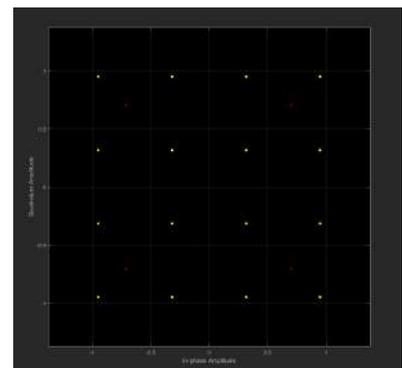
Para que la plataforma SDR transmita, se ejecuta el comando `step` también, como se ha visto en los algoritmos AIS, pasando por parámetro el objeto SDR y la señal modulada a transmitir. Para comprobar que las señales generadas se modulan correctamente, se observa la constelación de las señales moduladas, reflejándose en la figura 4.13. Por último, en la sección de código 4.20 se muestra la llamada de los objetos moduladores y la transmisión de los paquetes.



(a) Constelación de la señal modulada $\pi/4$ -QPSK.



(b) Constelación de la señal modulada 8-PSK.



(c) Constelación de la señal modulada 16-QAM.

Figura 4.13: Constelación de las señales moduladas a transmitir.

Código 4.20: Modulaciones y transmisión de paquetes VDES.

```

1 %VDE-TER 25 KHz modulación pi/4-QPSK
2 waveform_out = step(qpsk, vdes_message');
3

```

```
4 %VDE-TER 50 KHz modulación 8-PSK
5 waveform_out = psk(vdes_message ');
6
7 %VDE-TER 100 KHz modulación 16-QAM
8 waveform_out = qammod(vdes_message ', 16, 'InputType', 'bit', 'UnitAveragePower',
    true);
9 %Común para todos los servicios
10 while 1
11     step(tx_sdr, waveform_out);
12 end
```


5 | Diseño del módulo hardware de filtrado y amplificación

5.1. Introducción de conceptos sobre filtros y amplificadores

ADALM-Pluto es una plataforma transceptora SDR que incluye dos conectores SMA para la recepción y transmisión de señales. Con el objetivo de mejorar tanto la sensibilidad de la etapa receptora como alcanzar los niveles de potencia idóneos en la etapa transmisora, se plantea la incorporación de un circuito de filtrado y ajuste de ganancia en cada una de las etapas del SDR. Por un lado, la etapa receptora consta de un filtro ajustado a la banda de operación de los sistemas AIS y VDES y una red de amplificadores de bajo ruido. Por otro lado, la etapa transmisora consta de un amplificador de potencia para alcanzar los niveles de transmisión propios de equipos transmisores AIS clase B. En la figura 5.1 se representa un diagrama de bloques con los subsistemas que conforman el sistema diseñado en este trabajo.

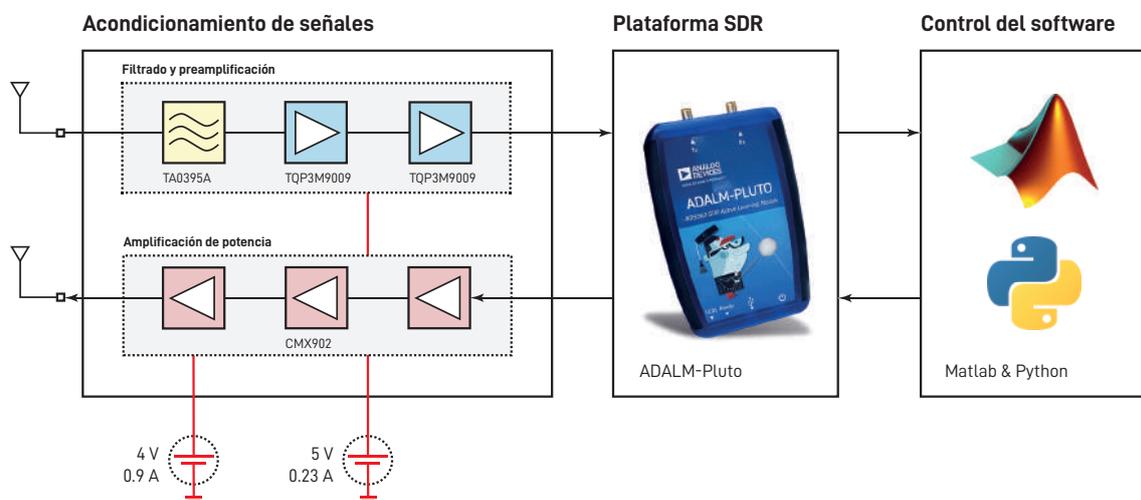


Figura 5.1: Diagrama de bloques de los subsistemas del transceptor AIS y VDES.

Antes de describir el diseño del circuito propuesto, en los siguientes puntos se detallan algunos conceptos prácticos sobre filtros de radiofrecuencia y amplificadores de bajo ruido y de potencia que serán utilizados a lo largo de las explicaciones aportadas en este capítulo.

5.1.1. Revisión de conceptos prácticos sobre filtros

Un filtro es un dispositivo electrónico que permite discriminar un conjunto de frecuencias de una señal de entrada, atenuando estas frecuencias frente a otras componentes espectrales. En esencia, estos dispositivos pueden ser redes de elementos lineales (condensadores y bobinas) o no lineales (transistores), distinguiéndose entre filtros *pasivos* y *activos*, respectivamente. A la banda de frecuencias que sufre la menor atenuación posible en un filtro se le llama *banda de paso*, mientras que a la banda de frecuencias más atenuadas se le llama *banda de atenuación*. Además, existe una frecuencia que divide la banda de paso de la atenuada, y se le llama *frecuencia de corte*. La imposibilidad física de implementar filtros ideales, que se caracterizan por una caída abrupta en las frecuencias de corte, exige considerar una región con una atenuación intermedia entre las bandas de paso y atenuada, a la que se le llama *banda de transición*.

Una de sus características más importantes de los filtros es su *respuesta en frecuencia*, $H(j\omega)$, que describe los cambios en la amplitud de una señal al pasar a través de ellos. En función de su respuesta en frecuencia, los filtros pueden clasificarse en *paso-bajo* (atenúa a aquellas frecuencias superiores a una frecuencia de corte dada), *paso-banda* (permite el paso de aquellas frecuencias que se encuentran comprendidas en una banda concreta), *elimina-banda* (atenúa a aquellas frecuencias que se encuentran comprendidas en una banda concreta) y *paso-alto* (atenúa a aquellas frecuencias inferiores a una frecuencia de corte dada). En la figura 5.2 se representan las respuestas en frecuencia de estos tipos de filtros, así como sus regiones.

Aunque generalmente se suele atender a la amplitud para el diseño de filtros, estos dispositivos pueden modificar la fase de las señales de entrada. De esta forma, es posible caracterizar un parámetro de interés como es el *retardo de grupo*, que da idea del retardo que sufre la envolvente de una señal cuando pasa a través de un filtro. Los parámetros S (o de dispersión) son una de las métricas más importantes en la caracterización de filtros. Como un filtro se puede considerar una red de dos puertas, se define un conjunto de cuatro parámetros S, siendo los parámetros S_{11} y S_{21} los de mayor interés. El parámetro S_{11} se relaciona con las pérdidas de retorno del filtro, $RL = -20 \cdot \log |S_{11}|$, que dan idea de la cantidad de potencia de entrada que se refleja durante

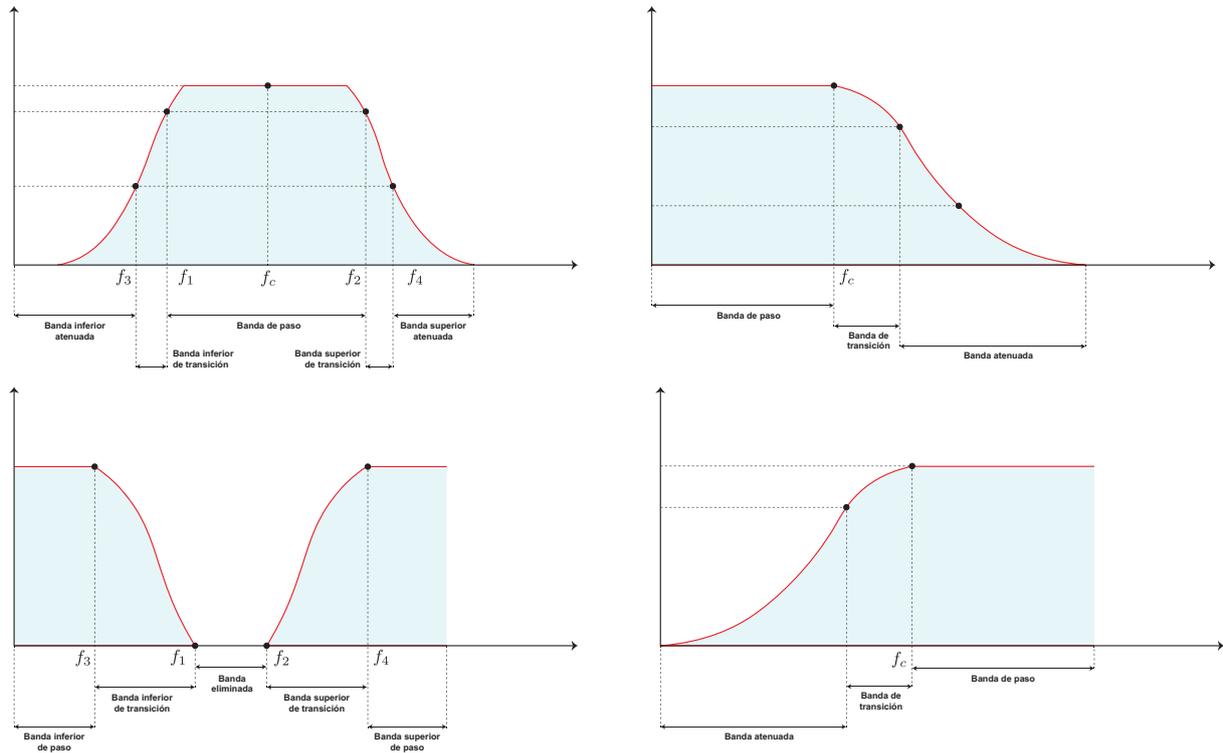


Figura 5.2: Clasificación de filtros según su respuesta en frecuencia.

su paso a través del filtro. El parámetro S_{21} se relaciona tanto con la respuesta en frecuencia del filtro, $|H(j\omega)|^2 = |S_{21}|$, como con las pérdidas de inserción del mismo, $IL = -20 \cdot \log |S_{21}|$, que dan idea de la atenuación que se produce en un circuito cuando se conecta el filtro. Todos los parámetros pueden relacionarse entre sí a través de la relación $S_{11} \cdot S_{21}^* + S_{12} \cdot S_{22}^* = 0$. A su vez, se puede decir que las pérdidas de inserción son máximas cuando se obtienen pérdidas de retorno mínimas, dado que $|S_{11}|^2 + |S_{21}|^2 = 1$.

5.1.2. Revisión de conceptos prácticos sobre amplificadores

Un amplificador de radiofrecuencia [54] es un dispositivo electrónico que aumenta la potencia de señales débiles. Dependiendo de la aplicación, existen diferentes tipos de amplificadores, siendo los amplificadores de potencia utilizados normalmente en transmisores y amplificadores de bajo nivel de ruido o LNA en receptores. A la hora de elegir un amplificador se deben tener en cuenta diferentes parámetros como la ganancia del dispositivo, el factor de amplificación que aplica el dispositivo a la señal entrante en dBs y que sigue la ecuación $P_{out}(dBm) = G(dB) + P_{in}(dBm)$, su frecuencia de trabajo, rango de frecuencias a la que el dispositivo es capaz de aplicar su ganancia, su figura de ruido, el factor de degeneración de señal expresado en dBs, su punto de 1 dB compresión,

potencia en la que difiere en 1 dB la zona lineal de la no lineal, su punto de cruce de tercer orden, relación de potencia de entrada y salida del amplificador, que de ser lineal, la potencia de la señal y de su tercer armónico son iguales, y su tensión de alimentación. Junto al amplificador se define una red de adaptación formado por componentes pasivos que aseguran el funcionamiento del dispositivo.

Por un lado, los amplificadores de potencia están ideados para obtener la máxima potencia a la salida y cuenta con diferentes clases. La clase A son amplificadores usados cuando se exige linealidad, mientras que la clase B se utiliza para amplificar señales con una polaridad determinada, y se encuentra al extremo de la zona lineal. La clase AB combina las especificaciones de las clases anteriores, trabajando en el extremo de la zona lineal como el B, pero eliminando su distorsión de cruce a costa de un mayor consumo de potencia, y por último, la clase C prioriza la máxima potencia de salida y rendimiento al trabajar en zonas no lineales. Se suelen utilizar los amplificadores de potencia clase C en comunicaciones moduladas en frecuencia. Por otro lado, los amplificadores de bajo ruido se caracterizan por su baja degradación de señal al tener una baja figura de ruido, aunque no aporta una ganancia tan alta como los amplificadores de potencia, por lo que se utilizan principalmente en receptores.

5.2. Etapa de filtrado y amplificación en recepción

La etapa de filtrado y ajuste de ganancia para recepción consta de los siguientes subsistemas: un filtro paso-banda basado en el chip TA0395A de Tai-Saw Technology y dos amplificadores LNA basados en el chip TQP3M9009 de Qorvo. El filtro se encargará de atenuar aquellas componentes espectrales fuera de la banda de interés de los sistemas AIS y VDES, mientras que los LNA se encargarán de aumentar la ganancia del sistema y mejorar tanto la sensibilidad como la figura de ruido.

5.2.1. Filtro TA0395A

El chip TA0395A es un filtro paso-banda fabricado con tecnología SAW (*Surface Acoustic Wave*) por la compañía Tai-Saw Technology. Este filtro está ajustado a una frecuencia central, f_c , de 159.1625 MHz. Su banda de paso está comprendida entre 156.3 MHz y 162.025 MHz, e introduce unas pérdidas de inserción comprendidas entre 2.3 – 3.8 dB. La banda de rechazo puede alcanzar

hasta 57 dB de caída (referenciada en 0 dB) entre $f_c - 50$ MHz y $f_c - 30$ MHz. El chip es capaz de soportar potencias de entrada máximas de +20 dBm, además de valores de tensión continua de hasta +5 V. El fabricante no aporta los ficheros S2P para caracterizar los parámetros S de este filtro. Por tanto, se ha desarrollado una PCB sencilla que consta de dos conectores SMA y un chip TA0395A. Mediante un analizador de redes, se han almacenado los parámetros S del filtro en formato S2P, importándose en ADS Keysight. Cabe señalar que, a partir de los parámetros S_{11} y S_{22} , es posible obtener las pérdidas de retorno, $RL = -20 \cdot \log |S_{11}|$, y de inserción, $IL = -20 \cdot \log |S_{21}|$, tal y como se observa en la figura 5.3.

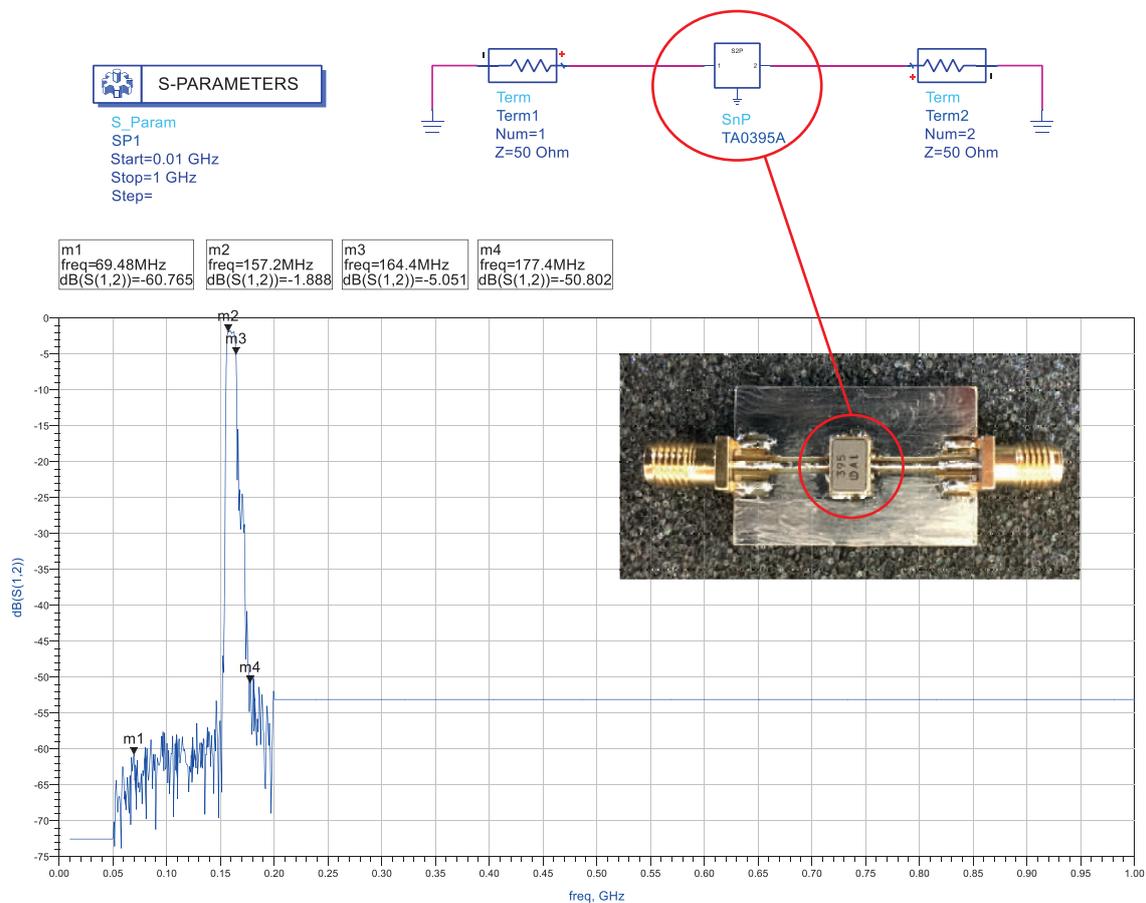


Figura 5.3: Simulación y caracterización del filtro TA0395A.

5.2.2. Amplificador TQP3M9009

El chip TQP3M9009 es un amplificador de bajo ruido o LNA que puede operar en frecuencias comprendidas entre 50 MHz hasta 4 GHz, con una ganancia entre 20 y 23 dB. Su punto a 1 dB compresión, P_{1dBc} , se encuentra en +22 dBm, su punto de intercepción de tercer orden, OIP_3 , y su figura de ruido es de 1.3 dB. Este amplificador debe contar con una red de adaptación externa

que dependerá del rango de frecuencias al que el usuario quiera destinarlo, diferenciando entre los rangos de 50 – 500 MHz y 500 – 4000 MHz. El chip puede soportar potencias de entrada máximas de +23 dBm, y debe ser alimentado por tensiones continuas entre +5 y +7 V. El fabricante aporta los ficheros S2P para caracterizar los parámetros S de este amplificador. Estos parámetros se han importado en ADS Keysight, obteniendo la representación que se muestra en la figura 5.4.

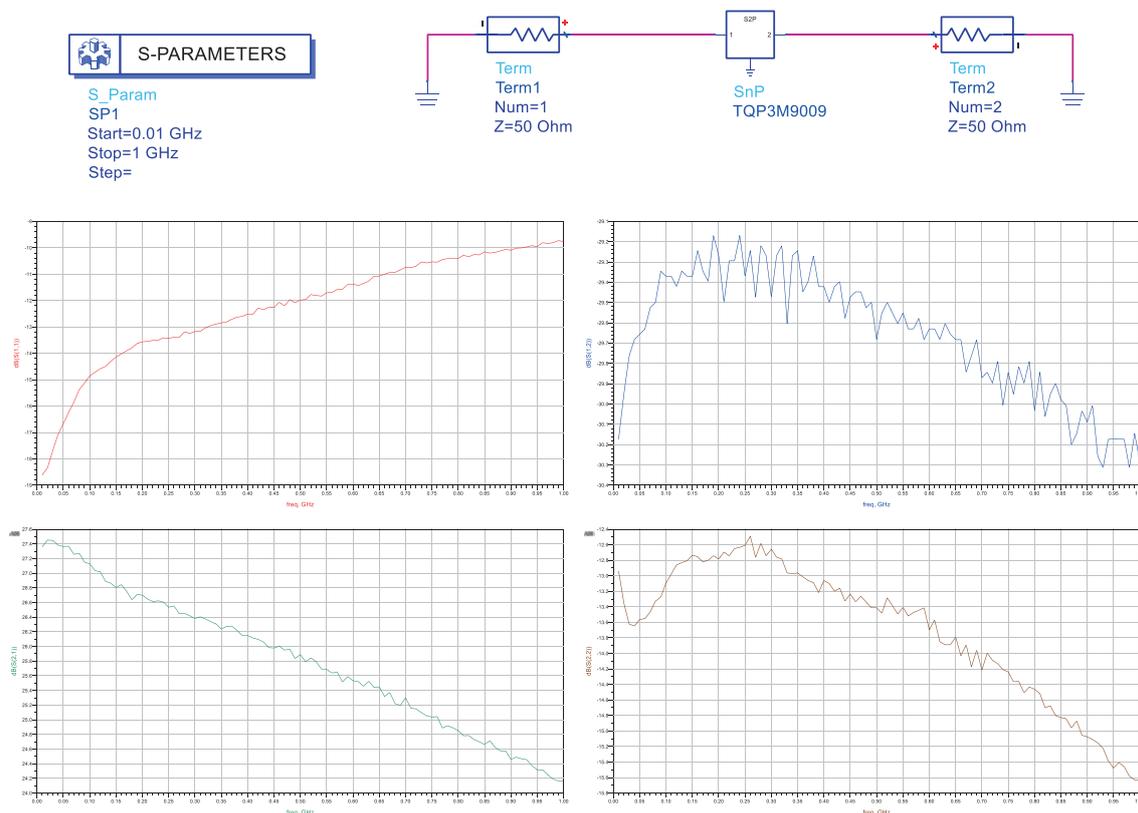


Figura 5.4: Simulación y caracterización del amplificador TQP3M9009.

5.2.3. Análisis de diferentes configuraciones de la etapa

Se plantean tres topologías o configuraciones en función de la disposición del filtro y los LNA: 1) conectar el filtro a la antena, seguido de ambos LNA mediante una conexión en cascada entre ambos; 2) conectar un LNA a la antena, seguido del filtro y, posteriormente, de un LNA, y 3) conectar uno de los LNA a la antena, que a su vez se conecta en cascada con otro LNA, y conectar el filtro. Para llevar a cabo el análisis de la configuración más propicia, se ha implementado en ADS Keysight una simulación de los tres casos, analizando sus parámetros S.

Se observa que la respuesta de los parámetros S11 y S21 en las tres configuraciones propuestas no difieren demasiado entre sí, por lo que se emplea como criterio de decisión el análisis de la

figura de ruido. Para ello, se aplica en cada caso la fórmula de Friis,

$$F = F_1 + \frac{F_2 - 1}{G_1} + \frac{F_3 - 1}{G_1 \cdot G_2} + \dots + \frac{F_n - 1}{G_1 \cdot G_2 \cdot \dots \cdot G_{n-1}} = F_1 + \sum_{i=2}^n \frac{F_i - 1}{\prod_{j=2}^{i-1} G_j} \quad (5.1)$$

Para aplicar esta expresión, se considera que las pérdidas de retorno del filtro son de 2.3 dB, y que los amplificadores de bajo ruido presentan una ganancia de 23 dB y una figura de ruido de 1.3 dB. Como cabría esperar, la configuración 1) es la que presenta una figura de ruido menor al constar de dos amplificadores conectados en cascada a la entrada, siendo de 1.36 dB. La configuración 2) presentaría la segunda figura de ruido menor, con 1.39 dB, mientras que la configuración 3) presentaría la figura de ruido más alta al ubicar a su entrada un filtro, siendo su figura de ruido de 3 dB. Sin embargo, la configuración 3) evitaría saturaciones en el receptor si se reciben señales intensas procedentes de sistemas y servicios externos a la banda de operación del SDR, como es el caso del servicio de radiodifusión comercial FM (88 MHz – 108 MHz) o los servicios de radioaficionados en VHF (144 MHz), entre otros. Por tanto, se ha escogido la configuración 3) para cubrir los propósitos de este trabajo. En la figura 5.5 se muestra la topología escogida y sus parámetros S asociados.

5.3. Etapa de amplificación en transmisión

A la salida de la etapa transmisora del ADALM-Pluto, se agrega un circuito basado en un amplificador de potencia. El amplificador de potencia utilizado es el modelo CMX902 de CML Microcircuits.

El chip CMX902 cubre la banda de 130 MHz – 700 MHz, y llevará asociada una red de adaptación externa basadas en componentes pasivos, permitiendo que el chip opere en las frecuencias más bajas (160 MHz como referencia) o más altas (435 MHz como referencia) de esta banda. En este trabajo, será de interés la banda más baja que cubre este chip, para la cual se obtiene una potencia de salida máxima de 2.8 W (34.5 dBm, aproximadamente) para una potencia de entrada mínima de -5 dBm.

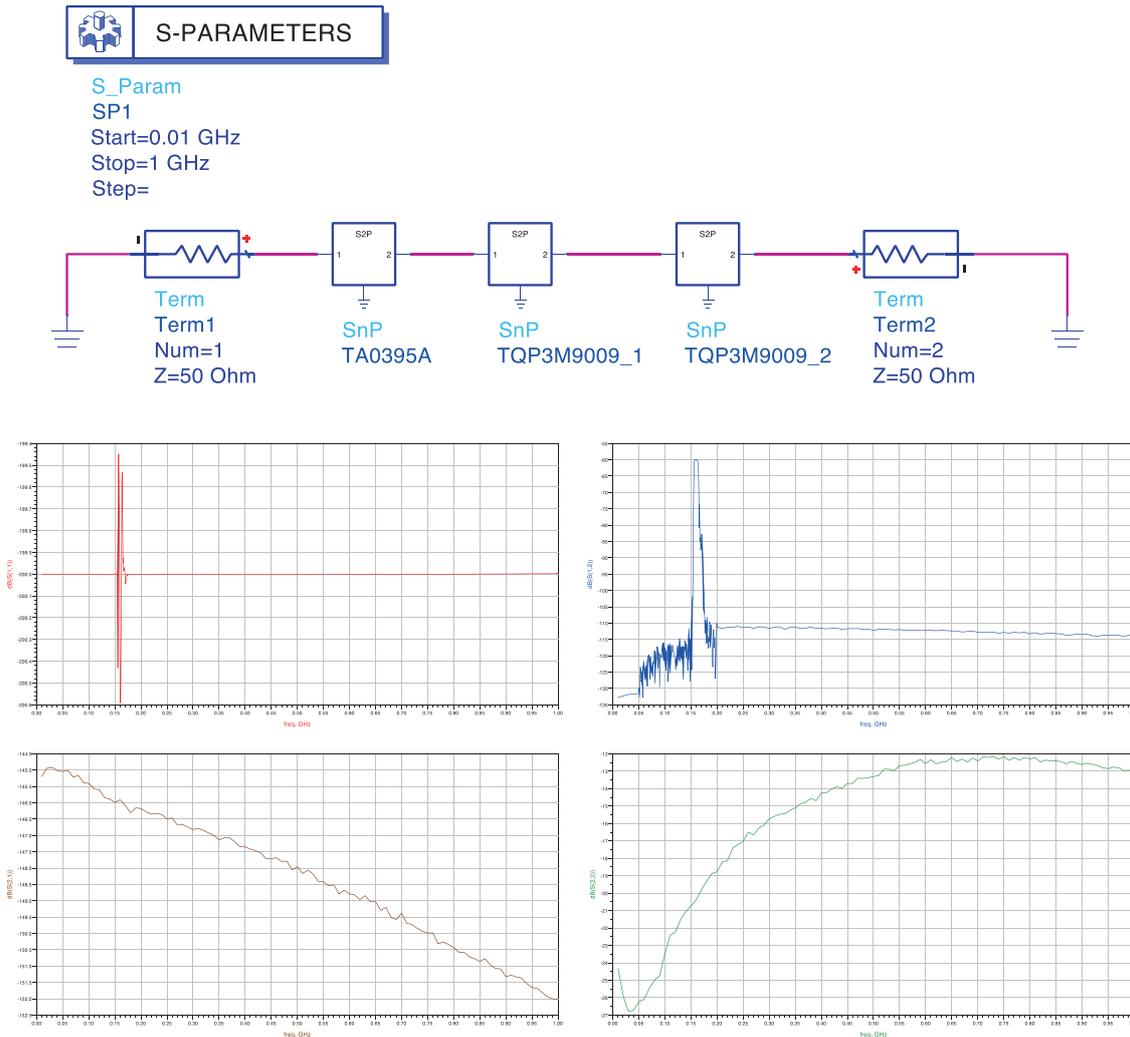


Figura 5.5: Parámetros S de la topología con configuración filtro → amplificador → amplificador.

5.3.1. Elección de un disipador de potencia

Un disipador de potencia es una estructura rígida metálica que facilita la evacuación del flujo del calor generado en un dispositivo electrónico cuando este disipa potencia debido al consumo de corriente [55], [56]. El disipador impide que el dispositivo sobrepase la temperatura máxima permitida que indica el fabricante, evitando una alteración de su comportamiento habitual o daños irreversibles. En términos generales, el disipador actúa como intermediario entre el dispositivo electrónico y el ambiente, recibiendo el exceso de calor del dispositivo y transfiriéndolo al ambiente, por lo que actuaría como sistema de refrigeración. Por tanto, se distinguen diferentes etapas en un montaje de un circuito con disipadores de potencia, como se representa en la figura en la figura 5.6.

El componente disipará una cantidad de potencia en forma de calor cuando se le aplica una cierta

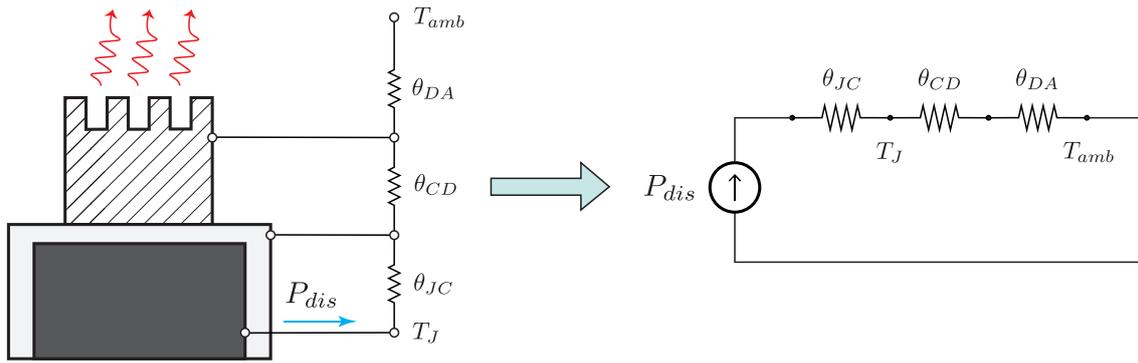


Figura 5.6: Modelo térmico equivalente del diseño planteado.

tensión V_{CC} y consume una corriente I_{CC} , $P_{dis} = V_{CC} \cdot I_{CC}$, lo que permite definir la temperatura de unión (*junction temperature*), T_J [57]. El calor generado por esta potencia disipada se transfiere al encapsulado del dispositivo, y llevará asociada una *resistencia térmica* que cuantificará la oposición que ofrece este cuerpo a la conducción térmica. Este hecho permite establecer una analogía con los circuitos eléctricos a través de la ley de Ohm, donde la temperatura equivale a la tensión y la potencia a la corriente. Por tanto, se puede establecer un *modelo térmico equivalente* como el de la figura 5.6, representando la dificultad que encontraría la temperatura para ser conducida a través de las distintas etapas del montaje.

La temperatura de la unión representará a la temperatura del componente, y su valor viene fijado por el fabricante. Como el calor se extiende a otras partes del componente, como su encapsulado, y se cuantifica su evolución a partir del valor de la resistencia térmica entre la unión y el encapsulado, θ_{JC} , expresada en $^{\circ}C/W$. En caso de incluirse un disipador térmico, debe considerarse una resistencia térmica entre el encapsulado y el disipador, θ_{CD} , y otra entre el disipador y el ambiente, θ_{DA} . Esto hará que en cada etapa del montaje exista una temperatura asociada, que siempre deberá ser mayor a la temperatura ambiente, T_{amb} . Por tanto, del modelo térmico equivalente de la figura 5.6 se puede formular la ecuación 5.2.

$$T_J = P_{dis} \cdot (\theta_{JC} + \theta_{CD} + \theta_{DA}) + T_{amb} \quad (5.2)$$

Para dimensionar correctamente el disipador de potencia requerido para un dispositivo electrónico, es necesario obtener el valor de la resistencia térmica asociada al disipador, θ_{DA} . Por tanto, se despeja este valor de la ecuación 5.2, quedando la ecuación 5.6.

$$\theta_{DA} = \frac{T_J - T_{amb}}{P_{dis}} - \theta_{JC} - \theta_{CD} \quad (5.3)$$

El chip CMX902 de CML Microcircuits es un amplificador de potencia de alta eficiencia que puede operar en un rango de frecuencias comprendido entre 130 MHz y 700 MHz. Los diseños que se pueden realizar con este amplificador requieren una red de adaptación externa basada en componentes pasivos. En el caso que se evalúa, se estudia un ajuste de referencia a 160 MHz. Las características térmicas a tener en cuenta para el dimensionamiento óptimo del disipador necesario para este chip son las siguientes:

- Operating air temperature, T_{amb} : $-40^{\circ}C$ (mín.), $+85^{\circ}C$ (máx.).
- Maximum allowable junction temperature, $T_J]_{max}$: $+125^{\circ}C$.
- Maximum continuous power dissipation, P_{dis} : 1.8 W.
- QT8 package 28-pin WQFN storage temperature, θ_{CD} : $-50^{\circ}C/W$ (mín.), $+125^{\circ}C/W$ (máx.).
- Thermal resistance junction to central heatsink ground pad, θ_{JC} : $2.5^{\circ}C/W$ (típ.), $3^{\circ}C/W$ (máx.).

Se pretende calcular la temperatura a la que se encontraría el chip en caso de no disponer de un disipador de potencia. Para ello, se plantea una nueva resistencia térmica entre el encapsulado y el ambiente, θ_{CA} , y su valor se puede obtener a partir de la ecuación 5.2 (eliminando los términos θ_{CD} y θ_{CD} debido a que no existe disipador). En la siguiente figura se representa el modelo térmico equivalente del caso analizado, y el valor de θ_{CA} se obtiene a partir de la ecuación 5.4.

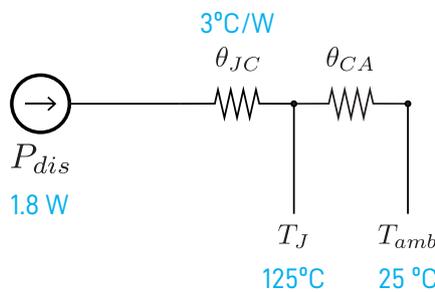


Figura 5.7: Modelo térmico equivalente del chip sin disipador.

$$\theta_{CA} = \frac{T_J - T_{amb}}{P_{dis}} - \theta_{JC} = \frac{125 - 25}{1,8} - 3 = 52,55^{\circ}C/W \quad (5.4)$$

El fabricante indica que si el chip es alimentado por 12 V (máxima tensión permitida), presentará un consumo de corriente de 2.2 A, por lo que la máxima potencia consumida en continua será $P_{DC}]_{max} = 12 \cdot 2,2 = 26,4 \text{ W}$. Ahora, se calcula la temperatura a la que se encontrará el chip bajo estas condiciones de alimentación, T_J , para lo que se aplica la ecuación 5.5.

$$T_J = \theta_{CA} \cdot P_{DC} = 52,55 \cdot 26,4 = 1387,32^\circ\text{C} \quad (5.5)$$

La temperatura que alcanzaría el chip sería muy superior a la máxima permitida indicada por el fabricante, $T_J \gg T_J]_{max} = 125^\circ\text{C}$, por lo que se evidencia la necesidad de incluir un disipador de potencia para evitar daños irreversibles en el chip. Si se añade un disipador de potencia, el modelo térmico equivalente será el representado en la siguiente figura, donde la resistencia térmica θ_{CA} desaparecerá, y se considerarán los términos θ_{JC} y θ_{CD} que aparecían en la ecuación 5.2.

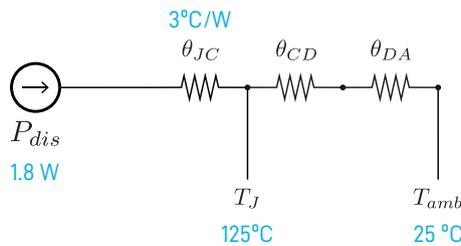


Figura 5.8: Modelo térmico equivalente del chip con disipador.

Sin embargo, algunos de los parámetros son difíciles de conocer, como es el caso de la resistencia térmica entre el encapsulado y el disipador, θ_{CD} , ya que en buena medida dependerá de si se usan tornillos para ajustar el contacto térmico o incluso de la calidad de la pasta térmica utilizada. Para este caso, se establece el siguiente criterio:

- Si el dispositivo se ajusta con tornillos, entonces θ_{CD} tendrá un valor entre 1°C y 2°C .
- Si sobre el dispositivo se ha aplicado pasta térmica, θ_{CD} tendrá un valor entre 0.2°C y 0.6°C .

En la ecuación 5.6 se ha empleado la ecuación 5.2 para obtener la resistencia térmica entre el disipador y el ambiente, θ_{DA} , donde se establece que el chip alcanza la temperatura máxima de 125° indicada por el fabricante.

$$\theta_{DA} = \frac{T_J - T_{amb}}{P_{dis}} - \theta_{JC} - \theta_{CD} = \frac{125 - 25}{26,4} - 3 - 0,5 = 0,287^\circ\text{C}/\text{W} \quad (5.6)$$

A partir del valor de θ_{DA} , se puede obtener la temperatura que alcanzaría el chip si se incluye un disipador de potencia mediante la ecuación 5.7, considerando que disipa la máxima potencia posible. El resultado obtenido indica que $T_J > T_J]_{max} = 125^\circ C$, pero no con tanta diferencia como cuando no se incluye el disipador de potencia.

$$T_J = (\theta_{DA} + \theta_{JC} + \theta_{CD}) \cdot P_{DC} + T_{amb} = (0,287 + 3 + 0,5) \cdot 26,4 + 25 = 124,98^\circ C \quad (5.7)$$

Si se escoge un disipador de potencia con una resistencia térmica entre disipador y ambiente inferior al valor calculado previamente, $\theta_{DA} < 0,287^\circ C/W$, se conseguiría que $T_J \leq T_J]_{max}$. Este sería el punto de partida para la elección del disipador de potencia, por lo que quedaría realizar una búsqueda de disipadores de potencia que presenten un valor de θ_{DA} apropiado y que sus dimensiones no afecten a la aplicación que se está desarrollando.

5.4. Diseño e implementación del circuito

Se ha implementado el circuito sobre una PCB con sustrato FR4 de dimensiones 7,9 x 5,8 cm. La PCB incluye cuatro conectores SMA, dos de ellos para la conexión con las antenas VHF en recepción y en transmisión, y otros dos para conectarse con los puertos de recepción y transmisión de la plataforma ADALM-Pluto. La rama de transmisión se encuentra en la mitad superior de la placa y cuenta con un amplificador de potencia CMX902 junto a su red de adaptación de componentes pasivos. Asimismo, la rama de recepción se encuentra en la mitad inferior y está estructurada según la topología filtro \rightarrow amplificador \rightarrow amplificador. Estos componentes son un filtro SAW TA0395A y dos amplificadores TQP3M9009 en cascada junto a su red de adaptación de componentes pasivos. Cada rama cuenta con su propias entradas de alimentación, ya que a la rama transmisora se le suministra 4 V y a la rama receptora 5 V.

Con el layout definido, se mandó a fabricar varias PCBs y se compraron los componentes necesarios para conformar el prototipo del módulo. Una vez con los materiales necesarios, se soldaron los componentes en el circuito impreso. En la figura 5.10 se puede observar como quedó el módulo hardware.

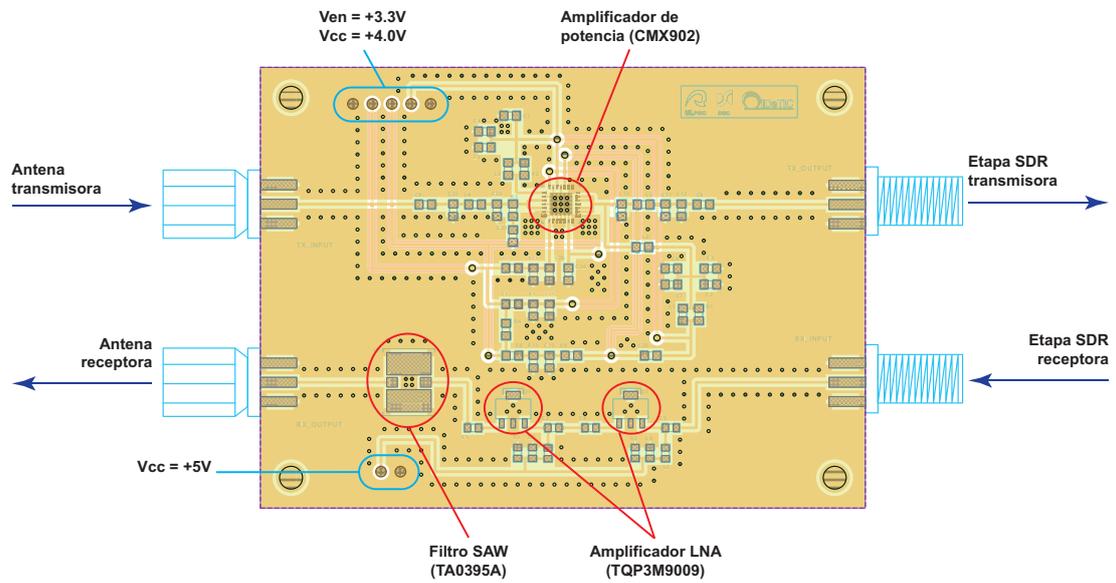


Figura 5.9: Layout del módulo hardware diseñado para el ADALM-Pluto.

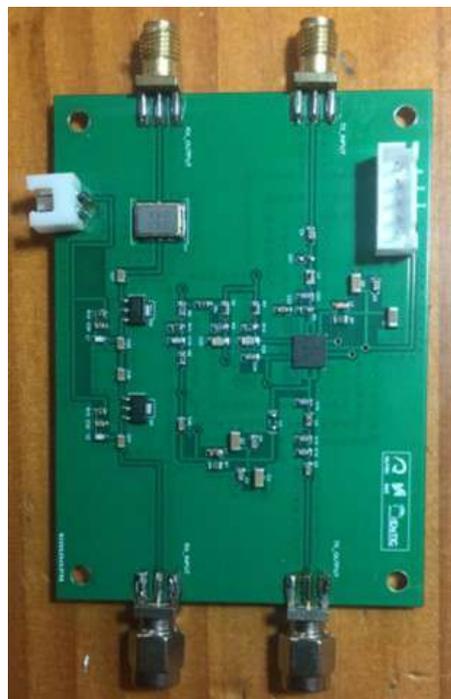


Figura 5.10: Módulo hardware de amplificación y filtrado.

5.5. Caracterización de antenas para recepción y transmisión

En este trabajo, se emplean dos antenas idénticas para transmisión y recepción de señales. Se ha escogido una antena corta helicoidal ajustada a la frecuencia de 162 MHz, basada en el modelo PI-2/SMAM de la compañía española Lambda Antenas. Para la caracterización de este modelo de antena, se ha utilizado un analizador de redes vectorial y se ha medido el parámetro de fase de la onda reflejada, el parámetro S11 o pérdidas de retorno y la relación de onda estacionaria. El parámetro de fase de la onda reflejada indica el retardo o desfase que sufre la onda al incidir en la antena. El tramo que presenta cambios de fase indica la frecuencia de sintonía de la antena utilizada. En la figura 5.11 se puede observar este parámetro de las antenas utilizadas y como los cambios de fase ocurren en la frecuencia de 162 MHz, frecuencia cercana a los canales AIS-1 y AIS-2.

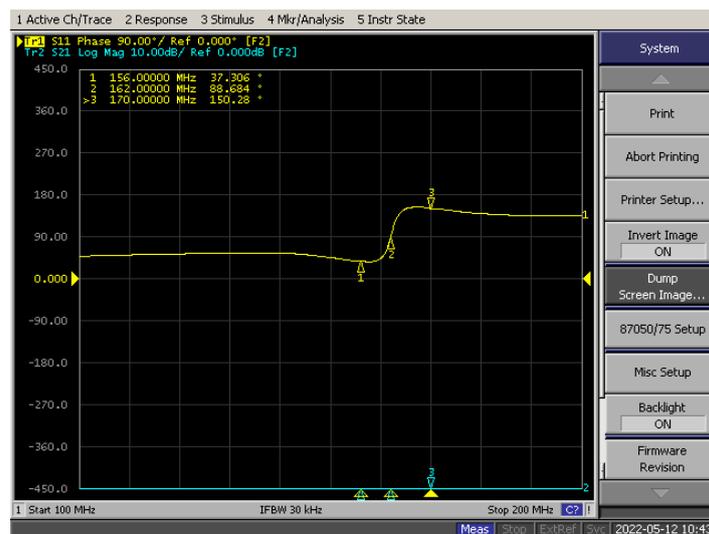


Figura 5.11: Parámetro de fase de las antenas utilizadas.

El parámetro S11 corresponde con las pérdidas de retorno, o en otras palabras, la pérdida de potencia de la onda reflejada y está relacionado con el parámetro de fase y es expresado matemáticamente como $S_{11} = \frac{b_1}{a_1} \Big|_{a_2=0} = \frac{Z_L - Z_0}{Z_L + Z_0}$, siendo Z_L la impedancias de carga y Z_0 la impedancia característica. Con la combinación de estos parámetros se puede conocer las características de la onda reflejada, ya que el parámetro S11 influye en la parte real y la fase en la parte imaginaria de la señal. Si el parámetro es de -20 dB significa que la onda reflejada es 20 dB menor que la onda incidente, dando a entender que parte de la onda incidente es reflejada por la antena. Al contrario, cuando el parámetro tiene valores cerca de 0 dB significa que gran parte de la onda o en su totalidad es reflejada. Por lo general, un valor de 20 dB en este parámetro es un buen resultado, ya

que se establece 10 dB como el mínimo aceptable. En la figura 5.12 se puede ver el el parámetro S11 de las antenas utilizadas.

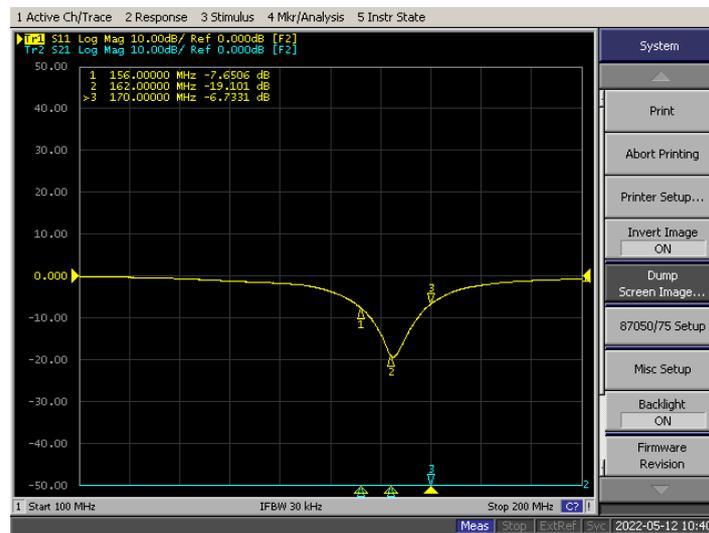


Figura 5.12: Parámetro S11 de las antenas utilizadas.

Por último, la relación de onda estacionaria o ROE es la relación entre valores de tensión máximos y mínimos de la onda reflejada que matemáticamente se expresa como $ROE = \frac{V_i + V_r}{V_i - V_r}$. Conceptualmente, la señal reflejada se suma a la señal incidente y se crea el fenómeno denominado onda estacionaria. El mínimo valor de esta relación es de 1 y significa que no refleja la onda. Este valor es teóricamente ideal, por lo que en la práctica no se consigue llegar a este valor exacto. En la figura 5.13 se puede ver la ROE de las antenas utilizadas y como en la frecuencia de interés se encuentra el mínimo valor de ROE.

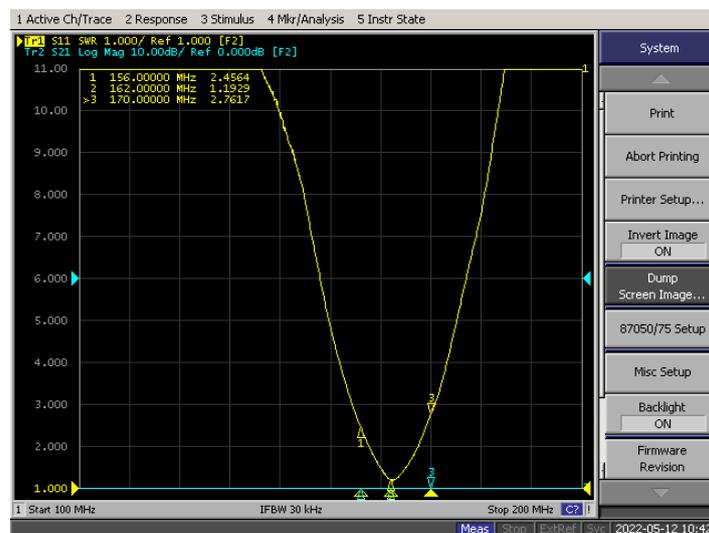


Figura 5.13: ROE de las antenas utilizadas.

6 | Pruebas y resultados

Tras la implementación de los diferentes algoritmos de transmisión y recepción del sistema AIS y el algoritmo de los servicios VDES, junto con la integración del módulo hardware de amplificación y filtrado, se han realizado diferentes pruebas y medidas para verificar el funcionamiento del prototipo. En primer lugar, se caracterizará la plataforma SDR elegida, midiendo su sensibilidad y la potencia de transmisión de la plataforma con el máximo de ganancia permitido. En segundo lugar, se evaluarán los algoritmos implementados del sistema AIS en tres entornos diferentes: banda base, donde no se utiliza la plataforma; entorno controlado, donde se prueba la plataforma con diferentes dispositivos comerciales y un segundo ADALM-PLUTO, y el entorno real, donde se prueba el algoritmo de recepción con transmisiones reales. En tercer lugar, se evalúa espectralmente las transmisiones de los diferentes servicios VDES para comprobar que se han implementado correctamente las modulaciones y se cumple con el estándar. Finalmente, se evalúa el comportamiento del módulo integrado mediante una prueba de recepción de señales AIS para estimar la mejora añadida y un análisis del espectro antes y después de aplicar el módulo. En cuanto a la etapa transmisora se analiza la potencia con la que se transmiten paquetes.

6.1. Caracterización de la plataforma ADALM-PLUTO

Para la caracterización de la plataforma ADALM-PLUTO, se tiene en consideración dos parámetros muy importantes. El primero es la sensibilidad, parámetro común en receptores, y el segundo es la potencia de transmisión cuando se programan los amplificadores al máximo permitido (0 dB).

La sensibilidad de un receptor, analógico o digital, se define como su capacidad para demodular y extraer información de señales con potencia débil. Por lo tanto, existe un valor de potencia recibida donde el receptor aún será capaz de demodular la señal. Este valor de potencia será la

sensibilidad del receptor, S_e . De modo que, se tienen las siguientes relaciones:

- Si $P_r < S_e \rightarrow$ no demodula correctamente.
- Si $P_r \geq S_e \rightarrow$ demodula correctamente.

Existen varios parámetros para estimar la sensibilidad de un receptor. En los receptores digitales se emplea la tasa de error de bit o BER. El BER mide el porcentaje de bits erróneos frente al número de bits recibidos, refiriéndose a la capacidad del receptor para demodular correctamente las señales recibidas. Otro método es relacionar la sensibilidad con el plato de ruido, N , la figura de ruido del receptor, N_f , y la relación portadora a ruido, C/N en la ecuación 6.1.

$$S_e = 10 \log(kTB) + N_f + C/N \quad (6.1)$$

Para el ADALM-PLUTO, el fabricante indica que con la ganancia ajustada en 0 dB la figura de ruido es de 2.5 dB. El término k se refiere al valor de la constante de Boltzmann, con valor $1,38 \cdot 10^{-23} JK^{-1}$. Además, se considera que la temperatura T es de 25° C (298 K) y un ancho de banda de 25 KHz, igual que en el sistema AIS. En el caso de añadir el módulo hardware desarrollado, también se obtiene un nuevo valor de sensibilidad. Por lo tanto, se realizaron diferentes medidas para estimar la sensibilidad de la plataforma. Para esta prueba se programó un algoritmo que permite al ADALM-PLUTO actuar como un analizador de espectros, y se recoge en la sección de código 6.1. Sin embargo, cabe destacar que este analizador no está calibrado. En primer lugar se realizaron las medidas sin amplificador y en segundo lugar con amplificador. También se realizó las mismas medida con el amplificador interno del ADALM-PLUTO al máximo valor, 70 dB, para observar su comportamiento. El generador de funciones se ha configurado para que transmita una señal con modulación GMSK y de ancho de banda de 25 KHz, simulando una señal AIS. En la figura 6.2 se puede ver el montaje de la medida, utilizando un generador de funciones y una fuente de alimentación, además del ordenador que controla la plataforma, y en las figuras 6.4, 6.7 y 6.8 se pueden ver las medidas para los casos sin módulo amplificador, con módulo, y aplicando el amplificador interno al valor máximo junto al módulo. Por último, en la figura 6.1 se puede apreciar cómo, en ausencia de señal existe el fenómeno de fuga de OL.

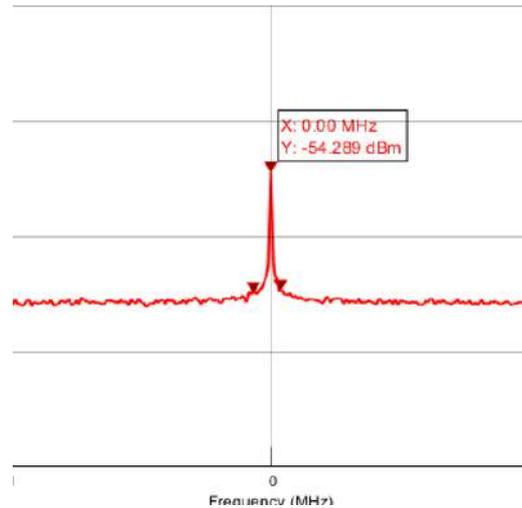


Figura 6.1: Fuga de OL de la plataforma SDR.

Código 6.1: Algoritmo del analizador de espectros.

```

1  %Creación de objeto "ADALM-Pluto Receiver" y ajuste de parámetros
2  fc = 161.525*10^6; %Definir frecuencia central (en Hz)
3  fs = 8*10^6;     %Definir tasa de muestreo (en Hz)
4  rx_sdr = sdrx('Pluto', 'RadioID', 'usb:0', ...
5               'CenterFrequency', fc, ...
6               'GainSource', 'Manual', ...
7               'Gain', 0, ...
8               'ChannelMapping', 1, ...
9               'BasebandSampleRate', fs, ...
10              'OutputDataType', 'single', ...
11              'ShowAdvancedProperties', false);
12 %Creación de objeto "Spectrum Analyzer" y ajuste de parámetros
13 spectrum_analyzer = dsp.SpectrumAnalyzer('SampleRate', fs, ...
14                                         'SpectralAverages', 1000, ...
15                                         'RBWSource', 'Auto', ...
16                                         'PlotNormalTrace', true, ...
17                                         'PlotMaxHoldTrace', false, ...
18                                         'PowerUnits', 'dBm');
19 %Ejecución del análisis espectral
20 while 1
21     rx = step(rx_sdr);
22     spectrum_analyzer(rx);
23 end

```

La potencia de la fuga de OL es de -54.3 dBm, por lo que para diferenciar la señal transmitida,

ésta debe tener una potencia mínima superior en 3 dB.

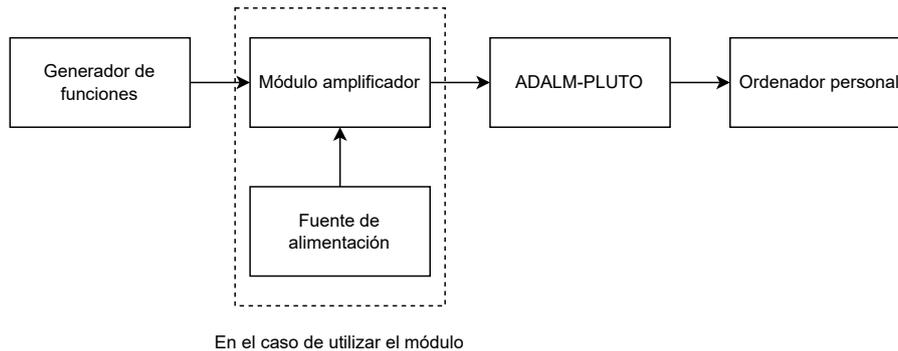


Figura 6.2: Diagrama de bloques del montaje realizado para las medidas de sensibilidad.



Figura 6.3: Configuración del generador de funciones sin utilizar el módulo amplificador.

Para esta medida, se ha configurado el generador de funciones para que emita una señal con una potencia de -78 dBm, que en el ADALM-PLUTO se recibe a -50.6 dBm, 3 dB por encima de la potencia de la fuga de OL. Por lo tanto, la sensibilidad de la plataforma sin el uso del módulo amplificador es de -78 dBm. Al incorporar el módulo, la potencia de la señal recibida aumenta hasta los -3 dBm, aumentando la potencia en 47.6 dB (figura 6.5). Por lo tanto, se debe reducir aún más la potencia de la señal transmitida en el generador de funciones, estimando así una nueva sensibilidad. Los resultados de esta nueva configuración se pueden ver en la figura 6.7.

Con este resultado se establece que el nuevo valor de sensibilidad es de -130 dBm, aumentando en 52 dB la capacidad de detectar señales débiles. Este valor se considera un margen de seguridad, siendo la diferencia entre la antigua sensibilidad y la nueva, expresándose como $MS = Se_o - Se_n$ siendo Se_o el valor de la antigua sensibilidad y Se_n el nuevo valor de sensibilidad. Al utilizar los amplificadores internos de la plataforma, aumenta la probabilidad de que el receptor se sature,

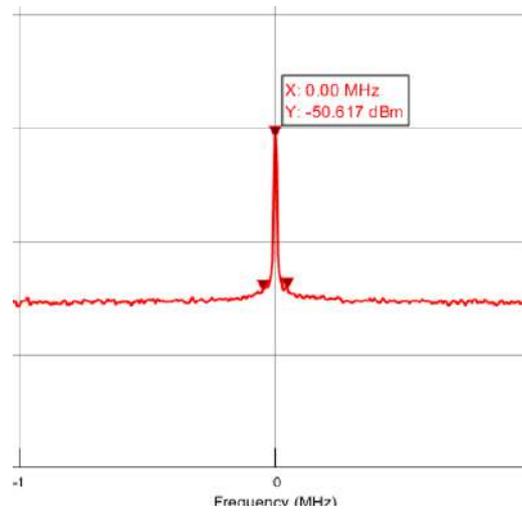


Figura 6.4: Medida de potencia recibida en el ADALM-PLUTO sin módulo amplificador.

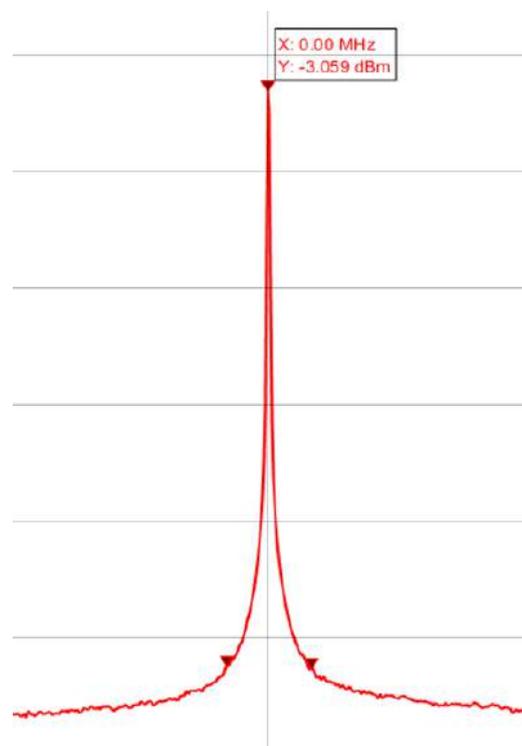


Figura 6.5: Potencia recibida al incorporar el módulo amplificador.



Figura 6.6: Configuración del generador de funciones con el módulo amplificador.

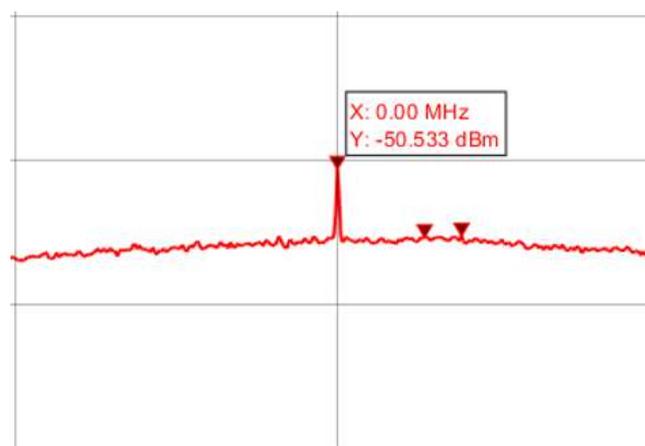


Figura 6.7: Potencia recibida con la nueva configuración del generador de funciones.

como pasa en la figura 6.8, por lo que no es recomendable utilizar valores elevados al programar la plataforma. Los resultados de la prueba se recogen en la tabla 6.1.

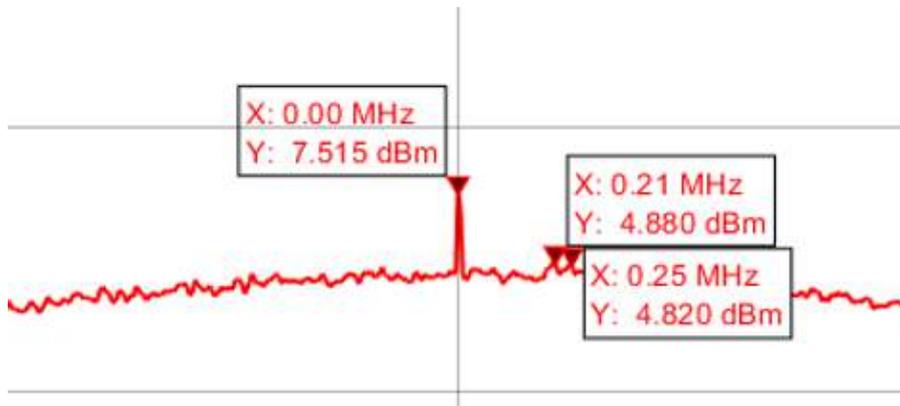


Figura 6.8: Señal recibida al emplear ambos amplificadores.

Tabla 6.1: Resultados de la medida de la sensibilidad del ADALM-PLUTO.

Sensibilidad pre-módulo	-78 dBm
Sensibilidad post-módulo	-130 dBm
Margen de seguridad	52 dB

La última medida de esta sección es la potencia máxima transmitida por la plataforma, por lo que se conectó a un analizador de espectros y se configuró su frecuencia central en 162.025 MHz. El analizador es capaz de soportar potencias hasta 20 dBm, y la ganancia máxima de la plataforma es de 0 en transmisión, por lo que no superará este valor. En la figura 6.9 se puede ver que la potencia máxima transmitida es de -6.77 dBm. En la tabla 6.2 se recogen el valor de ganancia utilizado y el valor de potencia de transmisión medido.

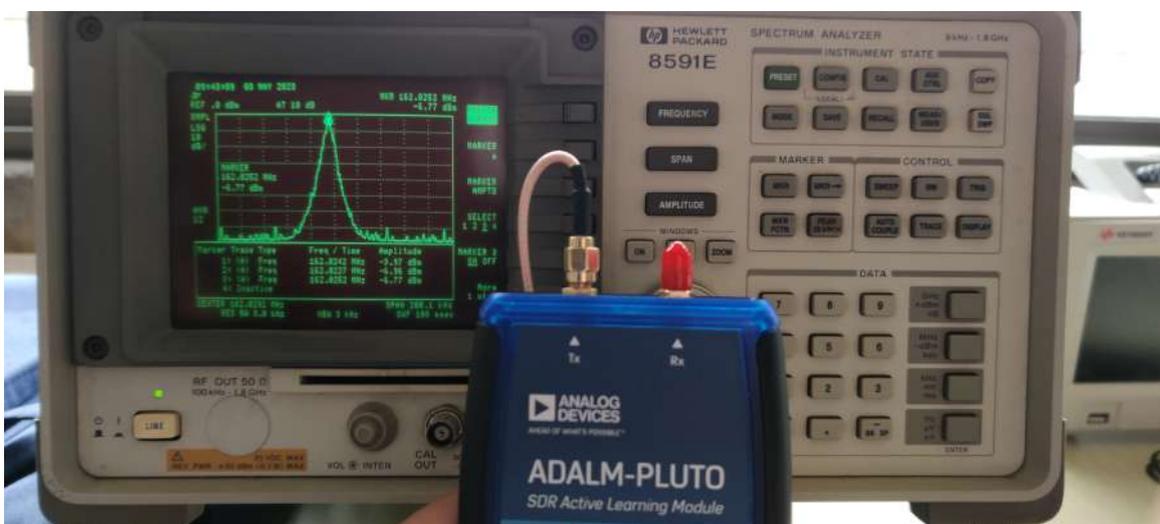


Figura 6.9: Potencia máxima de transmisión del ADALM-PLUTO.

Tabla 6.2: Resultados de la medida de la potencia de transmisión máxima del ADALM-PLUTO.

Valor máximo de ganancia en transmisión	0 dB
Potencia de transmisión medida	-6.77 dBm

6.2. Verificación y resultados de transmisión y recepción de mensajes AIS en entornos controlados y abierto

En esta sección se analizarán los resultados obtenidos de las pruebas realizadas a los algoritmos referentes al sistema AIS. Para ello, se realizaron diversos experimentos en diferentes entornos, como se comentó al inicio del capítulo.

6.2.1. Pruebas realizadas en banda base

La primera prueba realizada fue comprobar que el algoritmo es capaz de transmitir y recibir paquetes AIS, pero sin utilizar las plataformas SDR. Para ello, se combinaron los algoritmos de transmisión y recepción y ha sido ligeramente modificado, eliminando las operaciones de recorte, debido a que se está transmitiendo un mensaje que se sabe que es válido, agilizando el algoritmo. Uno de los cambios más significante es la incorporación de un simulador de canal, el *AWGN-Channel*, que introducirá ruido según la relación señal a ruido que se defina. Otra incorporación al *script* es el objeto *ErrorRate*, que comparará el mensaje a modular con el mensaje demodulado, indicando el número de bits erróneos como el porcentaje de error. En la sección de código 6.2 se encuentra la implementación de este nuevo algoritmo.

Código 6.2: Algoritmo de transmisión y recepción en banda base.

```

1  %VARIABLES:
2  symbol_rate = 9600;
3  samples_per_symbol = 24;
4  sample_rate = samples_per_symbol*symbol_rate;
5  bits_size = 300;
6  max_packet_size = bits_size*samples_per_symbol;
7  window_size = 1024;
8  bt_factor = 0.5;
9  gaussian_span = 3;
10 pulse_length = gaussian_span;
11 training_sequence = logical([1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0]);

```

```

12 start_flag = logical([0 1 1 1 1 1 1 0]);
13 end_flag = start_flag;
14 ais_data_1 = '483B69696003FFFE488C07B4ED41E10FFF80081773CF6D0000';
15 %Generar mensaje AIS:
16 [ais_message, stuffed_bits, msg_binario, full_msg2] = AIS_TX_Message_Generator(
    samples_per_symbol, training_sequence, start_flag, ...
17                                     ais_data_1, bt_factor, pulse_length);
18 size_ais_msg = size(ais_message);
19 waveform_size = size_ais_msg(2);
20 ais_tx_zeros = single(complex(zeros(waveform_size, 1)));
21 ais_tx_gmsk = single(ais_message. ');
22 awgnchan = comm.AWGNChannel('EbNo', 20);
23 outsignal = awgnchan(ais_tx_gmsk);
24 ais_signal_rx = outsignal;
25 %Compensación de componente continua
26 ais_signal_rx_rdc = Remove_DC_Offset(ais_signal_rx);
27 %Compensacion de la desviacion de frecuencia de la señal AIS:
28 ais_signal_rx_cfo = Frequency_Offset_Compensation(ais_signal_rx_rdc, ...
29                                                     sample_rate);
30 %Sincronizacion y demodulacion GMSK:
31 [ais_dec_bits, ais_prb_flag, rx_decimation, rx_sample_phase] =
    AIS_RX_Sync_Demodulation(ais_signal_rx_cfo, bt_factor, samples_per_symbol,
    pulse_length, training_sequence);
32 gmsk =comm.GMSKDemodulator('BandwidthTimeProduct', bt_factor, ...
33                             'BitOutput', true, ...
34                             'PulseLength', pulse_length, ...
35                             'SamplesPerSymbol', samples_per_symbol, ...
36                             'SymbolPrehistory', [-1 1]);
37 hError = comm.ErrorRate('ReceiveDelay', gmsk.TracebackDepth);
38 gmsk_demod = gmsk(ais_signal_rx);
39 errorStats = step(hError, double(msg_binario)', gmsk_demod)
40 gmsk_demod_decod = NRZI_Decod(gmsk_demod);
41 figure(1)
42 subplot(3,1,1)
43 plot(full_msg2)
44 title('msg binario');
45 subplot(3,1,2)
46 plot(gmsk_demod)
47 title('gmsk demod');

```

```

48 subplot(3,1,3)
49 plot(gmsk_demod_decod)
50 title('gmsk demod decod');
51 packet_counter = 0;
52 crc_counter = 0;
53 [payload, decoded_bits, crc_counter, mmsi, lat, lon] = AIS_RX_Bit_Parser(
    gmsk_demod_decod', packet_counter, crc_counter);
54 hexa_out = Bits_to_Hex(payload)
55 out = ['Paquete: ', hexa_out, ...
56       ' MMSI: ', num2str(mmsi), ...
57       ' Longitud: ', num2str(lon), ...
58       ' Latitud: ', num2str(lat)]

```

La primera prueba consiste en emplear un canal en el que la SNR sea positiva, es decir, que la potencia de la señal transmitida sea mayor al ruido. El resultado se puede ver en la figura 6.10 y como es de esperar, se modula y demodula correctamente, pudiendo extraer los campos de la trama AIS enviada. Este resultado se repite hasta llegar al valor de SNR de 0. Cabe destacar que al inicio de la señal demodulada existen 16 bits que no coinciden con la señal de datos enviada. Esto es debido a que la demodulación GMSK emplea el algoritmo de Viterbi, y requiere de 16 muestras o más para un funcionamiento óptimo.

La segunda prueba consiste en emplear un canal en el que la SNR tenga un valor de -10 dB, para observar cómo afecta el ruido a la modulación. El resultado de esta prueba se puede ver en la figura 6.11, y el objeto *ErrorRate* indica que existen 6 bits erróneos y un porcentaje de error del 3.28 %. Por lo tanto el ruido afecta considerablemente al proceso de demodulación.

Por último, se realizó una tercera prueba utilizando una SNR de -20 dB, para observar el efecto con una mayor potencia de ruido. Este resultado se puede ver en la figura 6.12 donde se comprueba que esta vez existen 103 bits erróneos junto a una tasa de error del 42.21 %. Este resultado confirma que cuanto mayor sea la potencia del ruido, más empeora el proceso de demodulación. En la tabla 6.3 se recogen los resultados de las tres pruebas realizadas en banda base

Tabla 6.3: Resultados de las pruebas en banda base.

	Bits eróneos	Porcentaje de error	¿Decodifica?
Canal con una SNR >0	0	0 %	✓
Canal con una SNR = -10 dB	8	3.28 %	✗
Canal con una SNR = -20 dB	103	42.21 %	✗

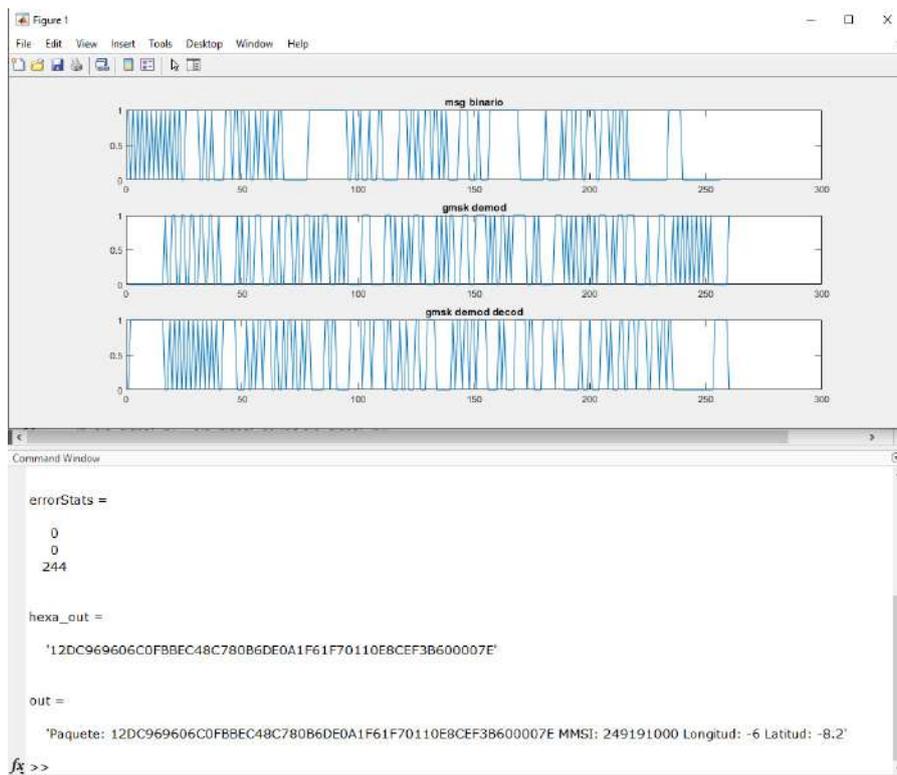


Figura 6.10: Resultado del algoritmo de transmisión y recepción en banda base con una SNR de 0 dB o superior.

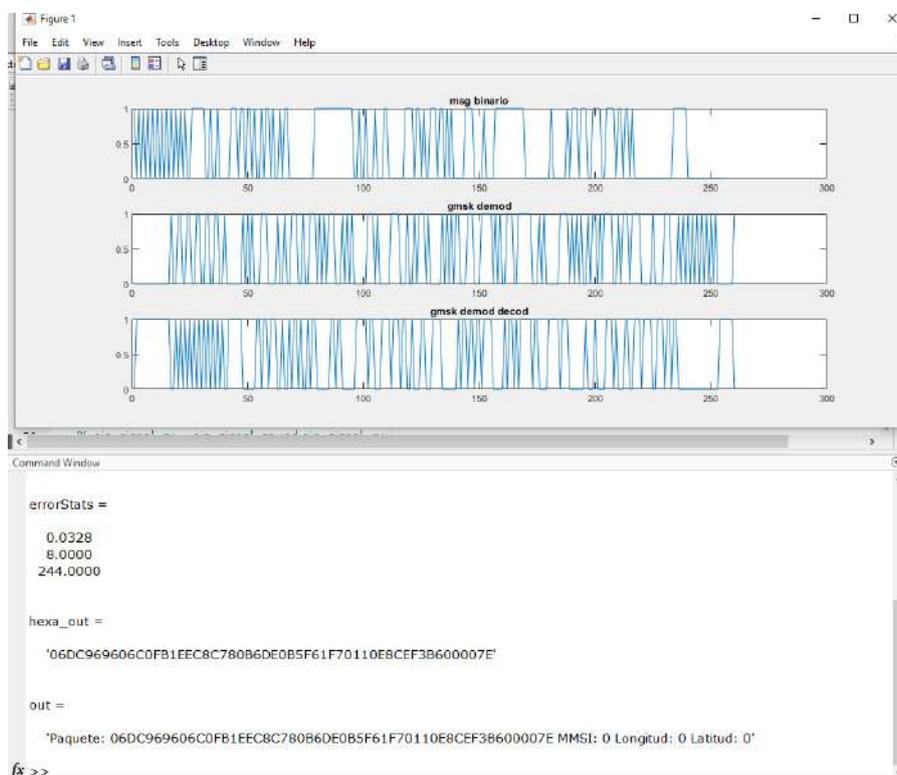


Figura 6.11: Resultado del algoritmo de transmisión y recepción en banda base con una SNR de -10 dB.

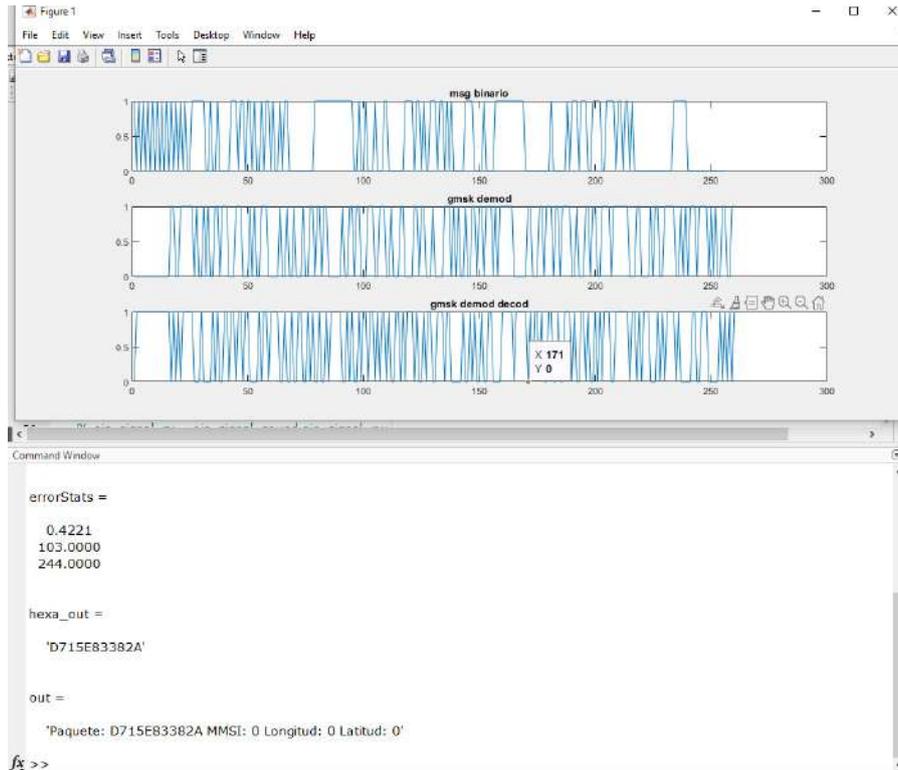


Figura 6.12: Resultado del algoritmo de transmisión y recepción en banda base con una SNR de -20 dB.

6.2.2. Pruebas realizadas en entorno controlado

Para las pruebas realizadas en entorno controlado se han utilizado el simulador de transmisiones AIS QK-AT011 mencionado en el capítulo 4 para evaluar los algoritmos de recepción y un transpondedor AIS clase B comercial denominado Cobalt de la empresa SRT Marine [58]. El módulo Cobalt es un transpondedor compacto que permite transmitir mensajes AIS de tipos 13, 18, 19 y 24, y capaz de recibir la mayoría de los tipos de mensajes AIS, además de decodificar los mensajes GPS. Para poder transmitir el Cobalt requiere de la conexión de una antena GPS, además de la antena VHF. Este transpondedor se utilizó con su kit de desarrollo, que se puede ver en la figura 6.13 junto a los conectores de alimentación y datos, y el software propio de la empresa, el *ProAIS2* para comprobar el algoritmo de transmisión.

Pruebas de recepción

Para las pruebas de recepción se configurado la plataforma con una amplificación interna de 30 dB, ya que la potencia de transmisión del simulador es de -51 dBm. Este valor no satura a la plataforma y permite amplificar la señal para diferenciarla del posible ruido. Se realizaron tres pruebas

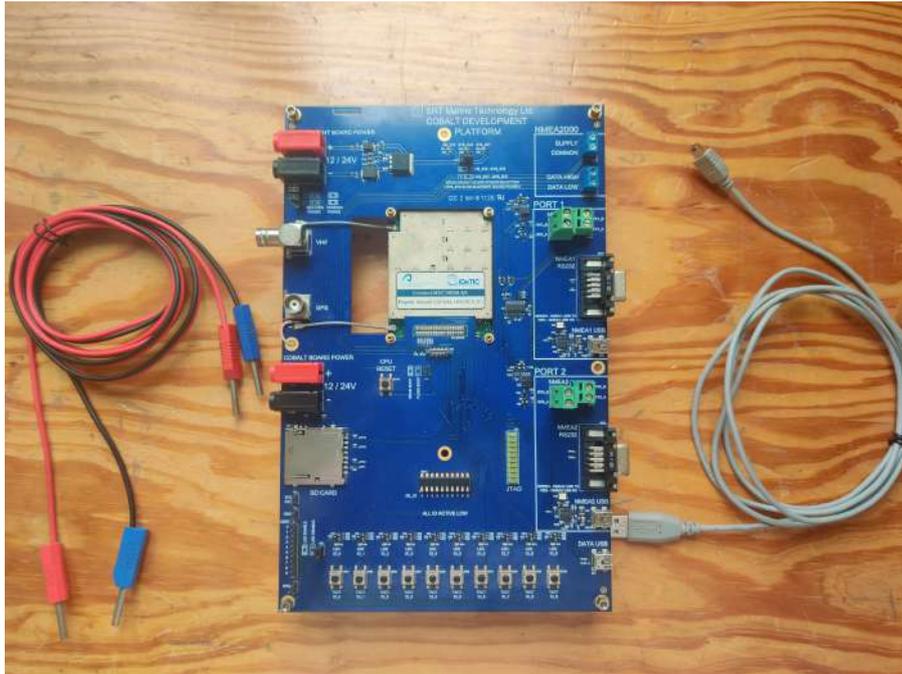


Figura 6.13: Traspondedor comercial SRT Marine Cobalt, junto a su kit de desarrollo.

en total, con envíos de tramas individuales, con 20 ráfagas de 10 paquetes y una transmisión de 10 minutos. Empezando por la prueba de transmisión de tramas individuales, se realizaron 50 transmisiones y la plataforma detecta las tramas, pero no es capaz de demodular correctamente. En la prueba de transmisión por ráfagas se observa que, debido a que el buffer de transmisión es del tamaño de la frecuencia de muestreo, la plataforma muestrea cada segundo 230400 muestras. Con este tamaño de buffer es posible recoger varias transmisiones del simulador, por lo que reduce el número de paquetes recibidos. Por último, el porcentaje de éxito de esta prueba es del 41.67%. Los resultados de esta prueba se pueden ver en la figura 6.14.

```

Command Window
packet_counter =
    36

crc_counter =
    15

success_rate =
    0.4167
  
```

Figura 6.14: Resultado al realizar 20 transmisiones con una ráfaga de 10 paquetes.

Por último, la prueba de transmisión durante 10 minutos revela que el porcentaje de éxito es del 45.26%, cercano a la mitad de tramas decodificadas. Este resultado puede deberse a que, al recortar la señal donde se encuentran los paquetes de mayor potencia, el recorte se realice en la mitad de una transmisión, por lo que en lugar de detectar el *flag* de inicio, se detecta el *flag* de fin.

Junto a este caso se ha de sumar posibles demodulaciones incorrectas. El resultado de esta prueba se puede ver en la figura 6.15. En la tabla 6.4 se recogen los datos de las tres pruebas realizadas.

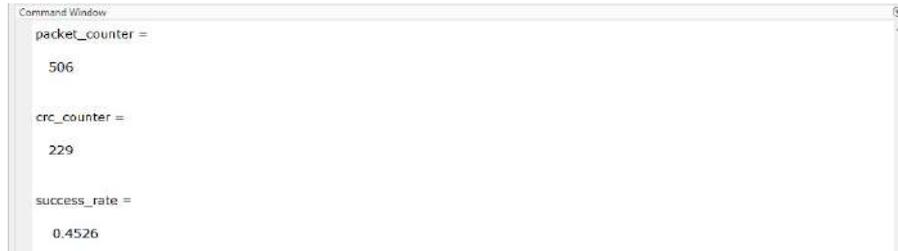


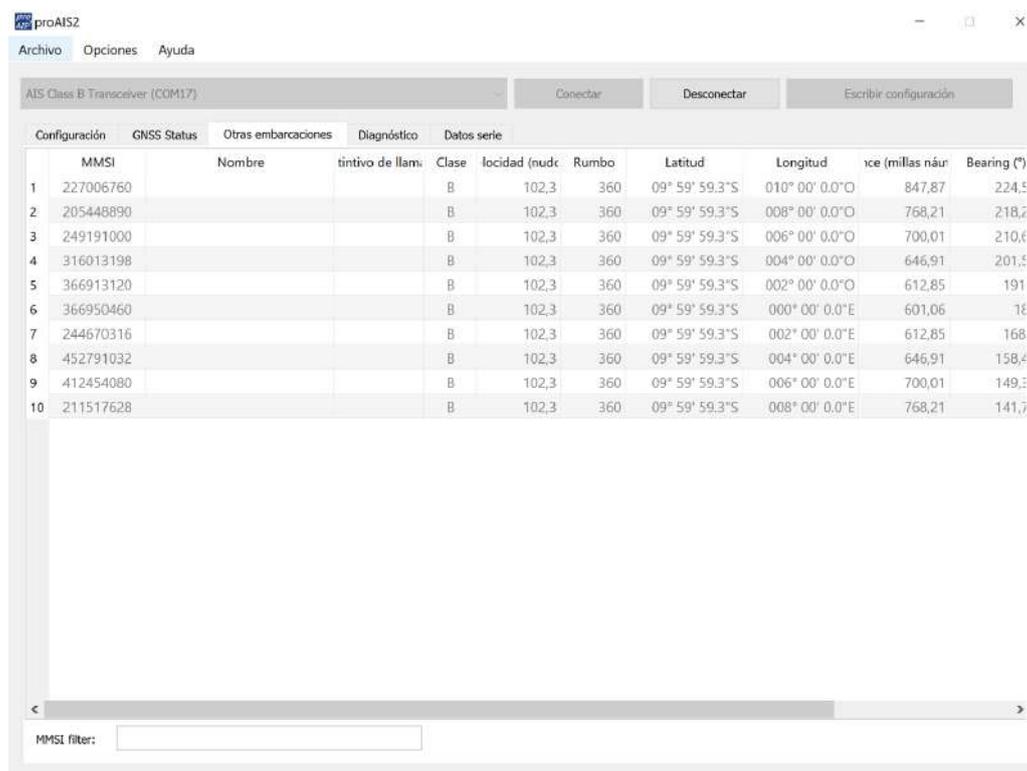
Figura 6.15: Resultado al realizar transmisiones durante 10 minutos.

Tabla 6.4: Resultados de las pruebas de recepción utilizando como transmisor el simulador QK-AT011.

	Paquetes recibidos	CRC válido	Porcentaje de éxito
Transmisiones individuales	50	0	0 %
Transmisiones a ráfagas	36	15	41.67 %
Transmision continua	506	229	45.26 %

Pruebas de transmisión

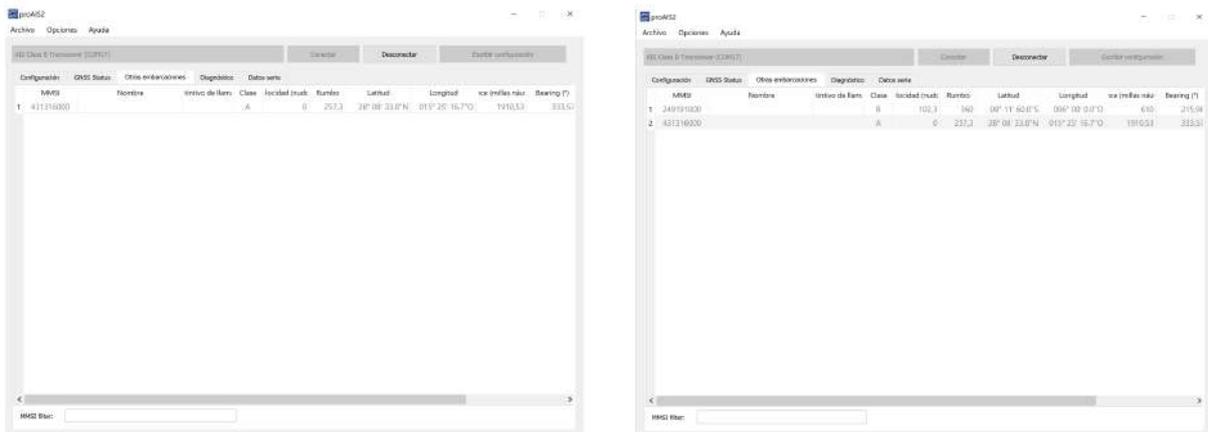
Para las pruebas de transmisión se han hecho tres experimentos. El primero es recibir las tramas generadas por el simulador QK-AT011 y en el transpondedor Cobalt para estimar la precisión de éste último. Tras 100 transmisiones, se puede observar que se han registrado 10 MMSI diferentes, pero que a medida que se transmitían cíclicamente, su valor de latitud y longitud se modificaba, por lo que es difícil establecer si se han recibido las 100 transmisiones. En la figura 6.16 se puede ver el resultado de las transmisiones y cómo con el software *ProAIS2* quedan recogidos los mensajes y MMSI recibidos.



Configuración	GNSS Status	Otras embarcaciones	Diagnóstico	Datos serie						
MMSI	Nombre	tintivo de llam	Clase	localidad (nude)	Rumbo	Latitud	Longitud	ica (millas náur)	Bearing (°)	
1	227006760		B	102,3	360	09° 59' 59.3"S	010° 00' 0.0"O	847,87	224,5	
2	205448890		B	102,3	360	09° 59' 59.3"S	008° 00' 0.0"O	768,21	218,2	
3	249191000		B	102,3	360	09° 59' 59.3"S	006° 00' 0.0"O	700,01	210,6	
4	316013198		B	102,3	360	09° 59' 59.3"S	004° 00' 0.0"O	646,91	201,5	
5	366913120		B	102,3	360	09° 59' 59.3"S	002° 00' 0.0"O	612,85	191	
6	366950460		B	102,3	360	09° 59' 59.3"S	000° 00' 0.0"E	601,06	18	
7	244670316		B	102,3	360	09° 59' 59.3"S	002° 00' 0.0"E	612,85	168	
8	452791032		B	102,3	360	09° 59' 59.3"S	004° 00' 0.0"E	646,91	158,4	
9	412454080		B	102,3	360	09° 59' 59.3"S	006° 00' 0.0"E	700,01	149,3	
10	211517628		B	102,3	360	09° 59' 59.3"S	008° 00' 0.0"E	768,21	141,7	

Figura 6.16: Resultado al recibir transmisiones 100 transmisiones con el transpondedor Cobalt.

La segunda prueba de esta sección consistía en transmitir una trama AIS desde la plataforma ADALM-PLUTO, mientras que el transpondedor Cobalt recibe. En la figura 6.17a se observa que el mensaje es transmitido correctamente, validando que el algoritmo de transmisión funciona correctamente. Sin embargo, se repitió la prueba pero modificando el código de transmisión para poder enviar dos tramas secuencialmente. En la figura 6.17b se observa que ambas tramas se han transmitido correctamente.



(a) Recepción de una trama enviada por el algoritmo de transmisión AIS implementado en el ADALM-PLUTO.

(b) Recepción de dos trama enviadas por el algoritmo de transmisión AIS implementado en el ADALM-PLUTO.

Figura 6.17: Recepción de tramas enviadas desde la plataforma ADALM-PLUTO.

Pruebas entre ADALM-PLUTOs

Este experimento consiste en probar el funcionamiento de los algoritmos de transmisión y recepción utilizando dos plataformas ADALM-PLUTO. El montaje consiste en configurar un Pluto como transmisor y conectar un cable SMA en rama transmisora. Asimismo, se configura la otra plataforma como receptora y se conecta este cable en la rama receptora. Se realizaron 3 tipos de transmisiones, al igual que en el apartado 6.2.2, empezando por transmisiones individuales, ráfagas de 10 tramas y transmisiones continuas durante 10 minutos. Para evitar problemas de saturación, se ha configurado el amplificador interno de la plataforma transmisora con una ganancia de -40 dB, dando lugar a una potencia de transmisión de -39 dBm al medirlo con el analizador de espectros. A la plataforma receptora se le ajustó una ganancia interna de -20 dB. Para el caso de las transmisiones individuales se han realizado 50 iteraciones, pero la plataforma receptora no consiguió decodificar ninguna trama. Se consiguió el mismo resultado para el caso de las ráfagas de transmisiones, tras 20 iteraciones no se llegó a decodificar ningún paquete. Sin embargo, para el caso de la transmisión continua se llegaron a decodificar 93 paquetes de 601 recibidos, dando un porcentaje de éxito del 15.47%. Este resultado se puede ver en la figura 6.18. En la tabla 6.5 se recogen los resultados de la prueba realizada.

Tabla 6.5: Resultados de las pruebas de transmisión y recepción entre dos ADALM-PLUTO.

Paquetes recibidos	CRC válido	Porcentaje de éxito
601	93	15.47 %



```
Command Window
-----
mmsi =
    431316000

long =
   -15.4213

lat =
    28.1427

packet_counter =
    601

crc_counter =
    93

success_rate =
    0.1547
```

Figura 6.18: Resultado de la ejecución de los algoritmos AIS implementados en dos ADALM-PLUTO.

6.2.3. Pruebas realizadas en entorno real

Para las pruebas realizadas en entorno real se han utilizado ambas plataformas ADALM-PLUTO para comprobar así el factor de mejora que introduce el módulo amplificador. En la figura 6.19a y 6.19b se puede ver el montaje del experimento. Cabe destacar que la prueba se realizó en el IDE-TIC, ubicado en la segunda planta del edificio Polivalente II del Parque Científico Tecnológico, a una distancia mayor de 3.5 km de la costa y sin visión directa (figura 6.20)



(a) Montaje de ambas plataformas, una con amplificador incorporado.



(b) Monitorización de los paquetes recibidos.

Figura 6.19: Montaje del experimento en entorno real.

Se configuraron las plataformas para que presenten la misma ganancia, teniendo en cuenta que



Figura 6.20: Distancia entre el IDETIC y la costa.

el amplificador presenta una ganancia cercana a 35 dB, se configuró el ADALM-PLUTO con el módulo incorporado una ganancia interna de 30 dB, mientras que la otra plataforma se le estableció el máximo de ganancia. Ambas plataformas cuentan con el mismo umbral de detección de paquetes, con el objetivo de igualar las condiciones entre los dos ADALM-PLUTO. El experimento se llevó a cabo durante 10 minutos y los resultados obtenidos se encuentran en la figuras 6.21 para el caso de la plataforma con el amplificador y 6.22 para el caso de la plataforma sin el módulo. La diferencia entre las tasas de éxito de ambas plataformas es de un factor de 5.28 veces mejor en el ADALM-PLUTO con el amplificador incorporado, ya que, al presentar teóricamente la misma ganancia o muy similar, éste se encontraba menos saturado.

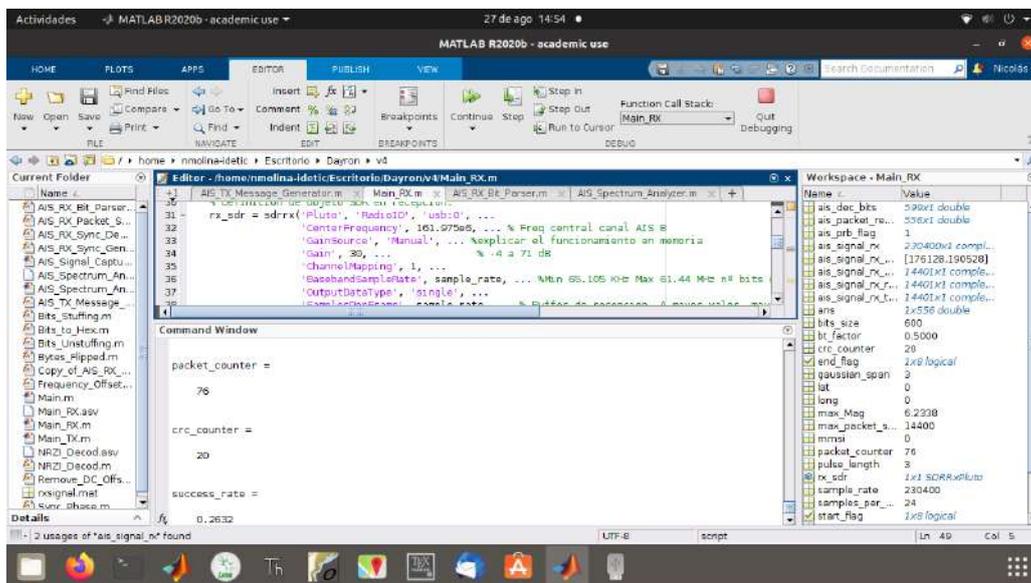


Figura 6.21: Resultados tras 10 minutos de recepción de la plataforma con amplificador.

Aparte del rendimiento del amplificador, también se puede observar que se han conseguido recibir mensajes reales de transpondedores AIS. Si se extrae el MMSI de alguna transmisión y se busca en la página de MarineTraffic [59], se puede conocer todos los detalles del buque que

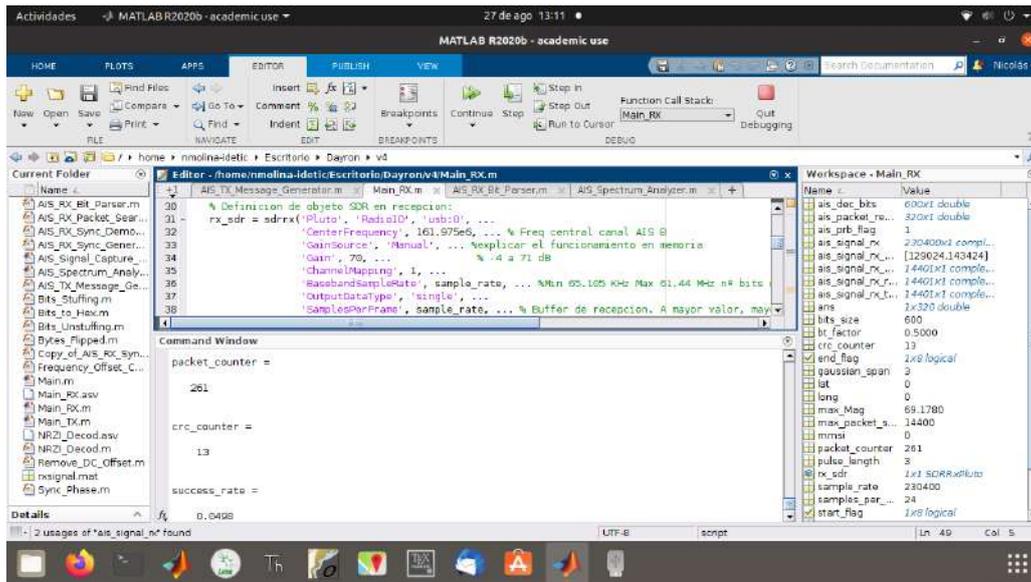


Figura 6.22: Resultados tras 10 minutos de recepción de la plataforma sin amplificador.

ha transmitido el mensaje. En la figura 6.23 se puede observar como el pesquero denominado *Shoshinmaru* transmitió el paquete el día 16 de junio a las 12:40. Los resultados de esta prueba se recogen en la tabla 6.6.

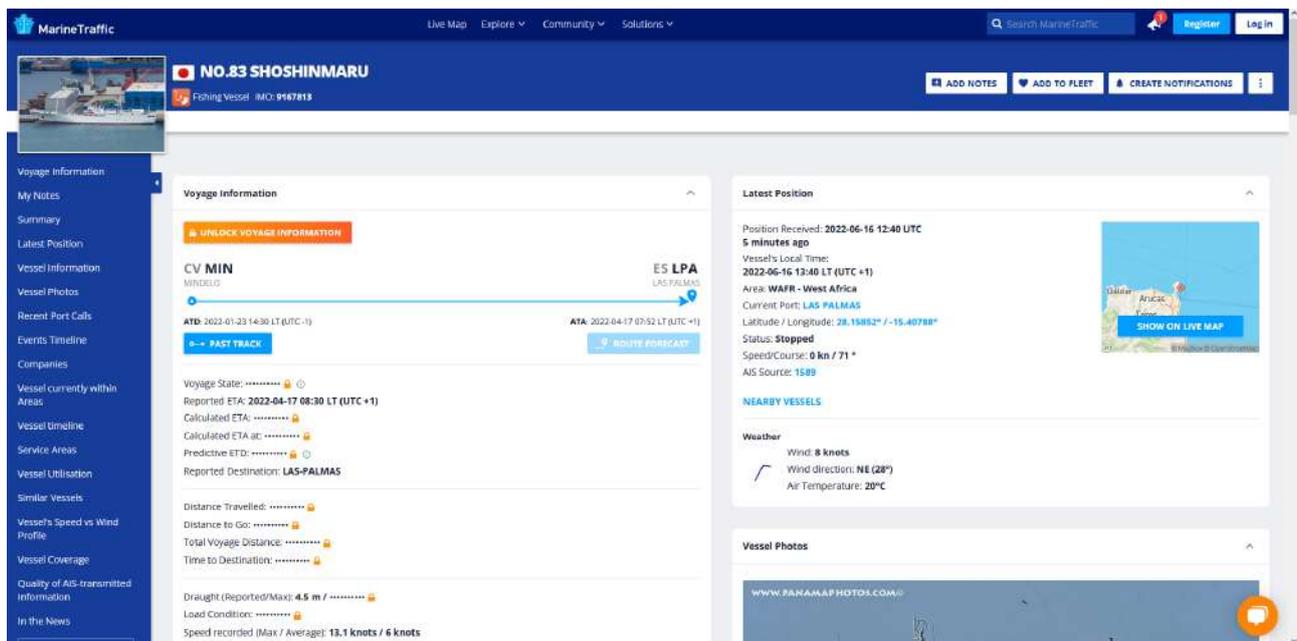


Figura 6.23: Datos del buque *Shoshinmaru*, recibido desde el IDeTIC.

Tabla 6.6: Resultados de las pruebas de recepción en entorno real.

	Paquetes recibidos	CRC válido	Porcentaje de éxito
SDR con amplificador	76	20	26.32 %
SDR sin amplificador	261	13	4.98 %

6.3. Verificación y resultados de transmisión de señales VDE-TER en entornos controlados

Debido a la reciente publicación del estándar definitivo de VDES y la escasez de equipos comerciales que implementan este estándar, las pruebas realizadas para verificar el correcto funcionamiento de los algoritmos de transmisión se han limitado a la comprobación de los aspectos espectrales de cada servicio. Las constelaciones de cada modulación implementada se han visto en la figura 4.13, por lo que en esta sección se centrará en los resultados obtenidos a través de un analizador de espectros. Para los anchos de banda menores a 100 KHz se puede diferenciar la señal de datos en el analizador, ya que se desplaza frecuencialmente en el ancho de banda de transmisión utilizado, cercano a 1 MHz. Esta señal de datos tiene un ancho de banda igual al que indica la norma [53] siendo para ASM y VDE-TER para canales de 25KHz una señal de datos de 25 KHz de ancho de banda, para VDE-TER para canales de 50 KHz una señal de datos de 49.9 KHz de ancho de banda y para VDE-TER de 100 KHz una señal de datos de 99.8 KHz. Los espectros mostrados en las figuras 6.24, 6.25 y 6.26 corresponden a los servicios ASM y VDE-TER de 25 KHz con modulación $\pi/4$ -QPSK, VDE-TER de 50 KHz con modulación 8-PSK y VDE-TER de 100 KHz con modulación 16-QAM respectivamente.

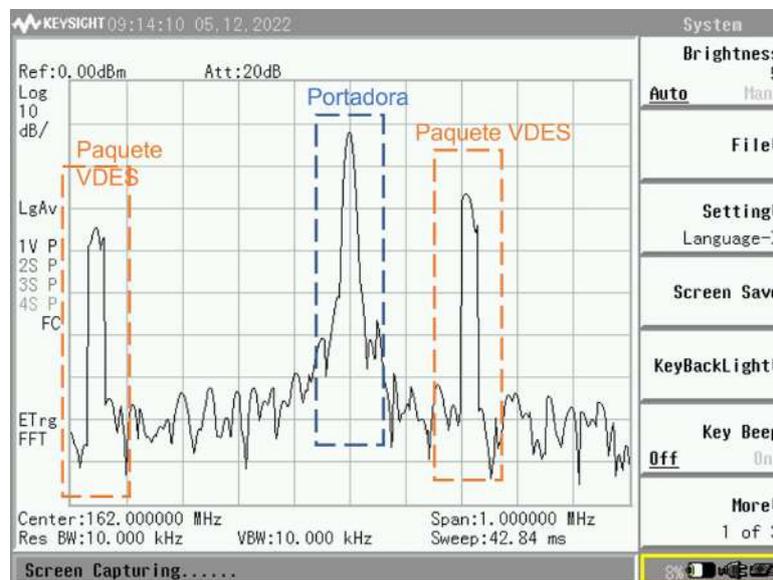


Figura 6.24: Espectro de los servicios ASM y VDE-TER de 25 KHz y modulación $\pi/4$ -QPSK.

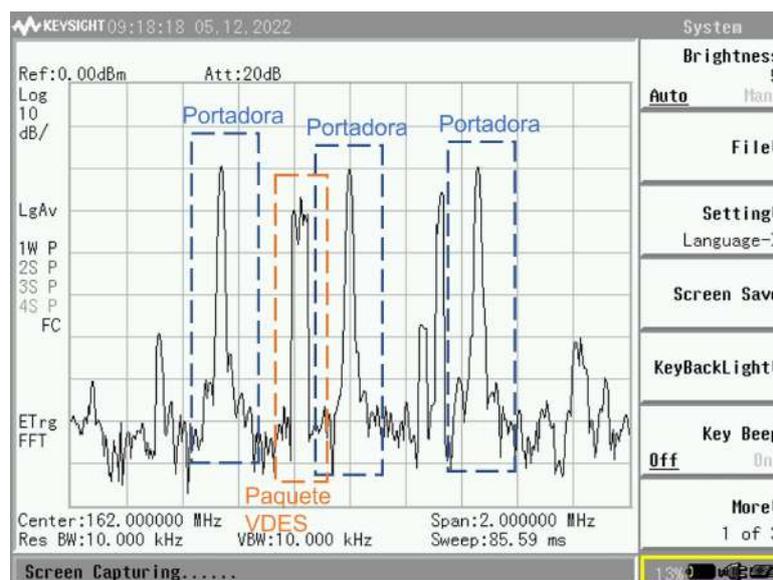


Figura 6.25: Espectro del servicio VDE-TER de 50 KHz y modulación 8-PSK.

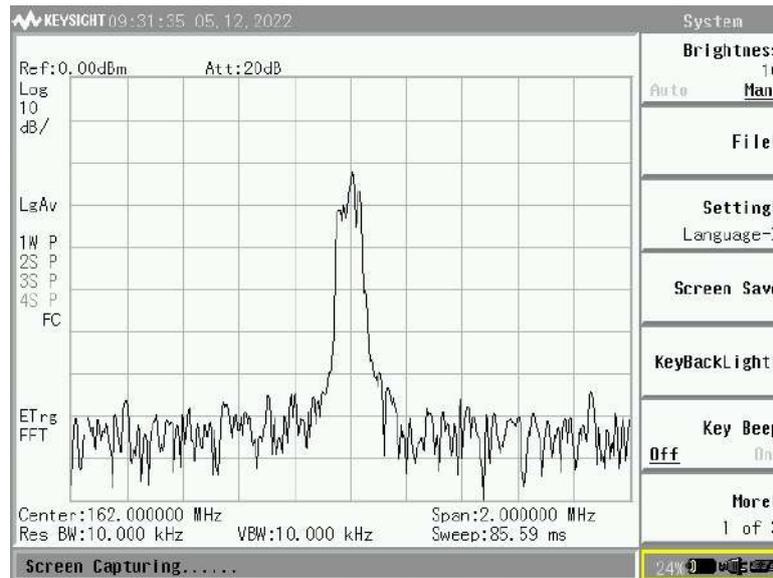


Figura 6.26: Espectro del servicio VDE-TER de 100 KHz y modulación 16-QAM.

6.4. Verificación y resultados del módulo hardware

En esta última sección se evaluará el comportamiento del módulo de amplificación y filtrado desarrollado en este trabajo, tanto la rama receptora como la rama transmisora.

6.4.1. Verificación y resultados de la etapa receptora del módulo hardware

Ya que previamente se comprobó el comportamiento de la etapa de amplificación de esta rama, en esta sección se evaluará el comportamiento del filtro en toda la banda de VHF. Para ello, se ha de analizar previamente el estado de la banda antes de emplear el módulo y se puede observar en la figura 6.27. La banda FM tiene una potencia muy superior a las transmisiones AIS, situadas en 162 MHz y podría llegar a causar interferencias o saturar la plataforma SDR, por lo que es necesario atenuar las bandas que no son de interés.

Tras aplicar el módulo de amplificación y filtrado, el espectro resultante se muestra en la figura 6.28. El módulo atenúa la banda FM 9.9 dB, pasando de -21.4 dBm a -31.3 dBm. Además, las señales AIS recibidas se han amplificado 40.8 dB, pasando de -61.3 dBm a -20.5 dBm.

Tras analizar estos resultados, se puede afirmar que el módulo de amplificación y filtrado mejora las prestaciones en recepción, no solo amplificando la banda deseada, sino filtrando el resto de frecuencias. Otra de las pruebas fue el cálculo del punto 1 dB compresión y el punto de cruce

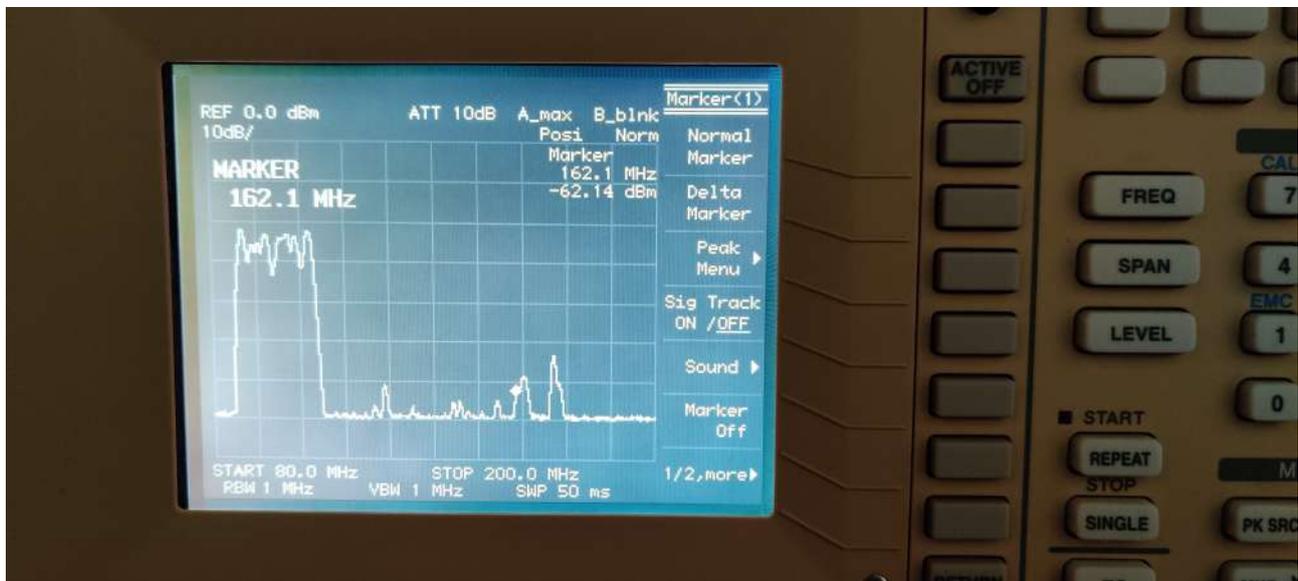


Figura 6.27: Espectro de la banda VHF antes de incorporar el módulo.



Figura 6.28: Espectro de la banda VHF tras incorporar el módulo.

de tercer orden. Para ello, con un generador de funciones se introduce una señal de frecuencia igual a la frecuencia de sintonía (162 MHz en este caso) de débil potencia y se anota el valor de la señal recibida y de su tercer armónico. Para el experimento, se tomaron las medidas empezando con una señal de -50 dBm hasta 0 dBm, incrementando la potencia 5 dBm hasta que el tercer armónico igualaba en potencia la señal recibida. Trasladando los datos a MATLAB, se calcula el punto de la recta en la que difiere 1 dB de la zona lineal, obteniendo el punto 1 dB compresión. Para el punto de cruce de tercer orden se calculan las rectas teóricas para la señal recibida y el tercer armónico, y donde se crucen estas rectas se halla este parámetro. En la figura 6.29 Se puede ver el valor de tanto el punto 1 dB compresión como el punto de cruce de tercer orden.

La última medida de la rama receptora es el consumo de total de la rama. Dado que en los *datasheets* de los amplificadores TA0395A indica que la alimentación típica de estos componentes es de 5 V, ésta es la tensión suministrada a la rama. Al suministrar 5 V la fuente de alimentación indica que la rama receptora consume 232 mA, como se puede ver en la figura 6.30. En la tabla 6.7 se recogen los resultados de las medidas realizadas a la rama receptora del módulo hardware.

Tabla 6.7: Características medidas de la rama receptora del módulo hardware.

Atenuación de FM	Ganancia AIS	Punto 1dB compresión	Punto de cruce de tercer orden	Alimentación	Consumo
9.9 dB	40.8 dB	-29.25 dBm	1 dBm	5 V	232 mA

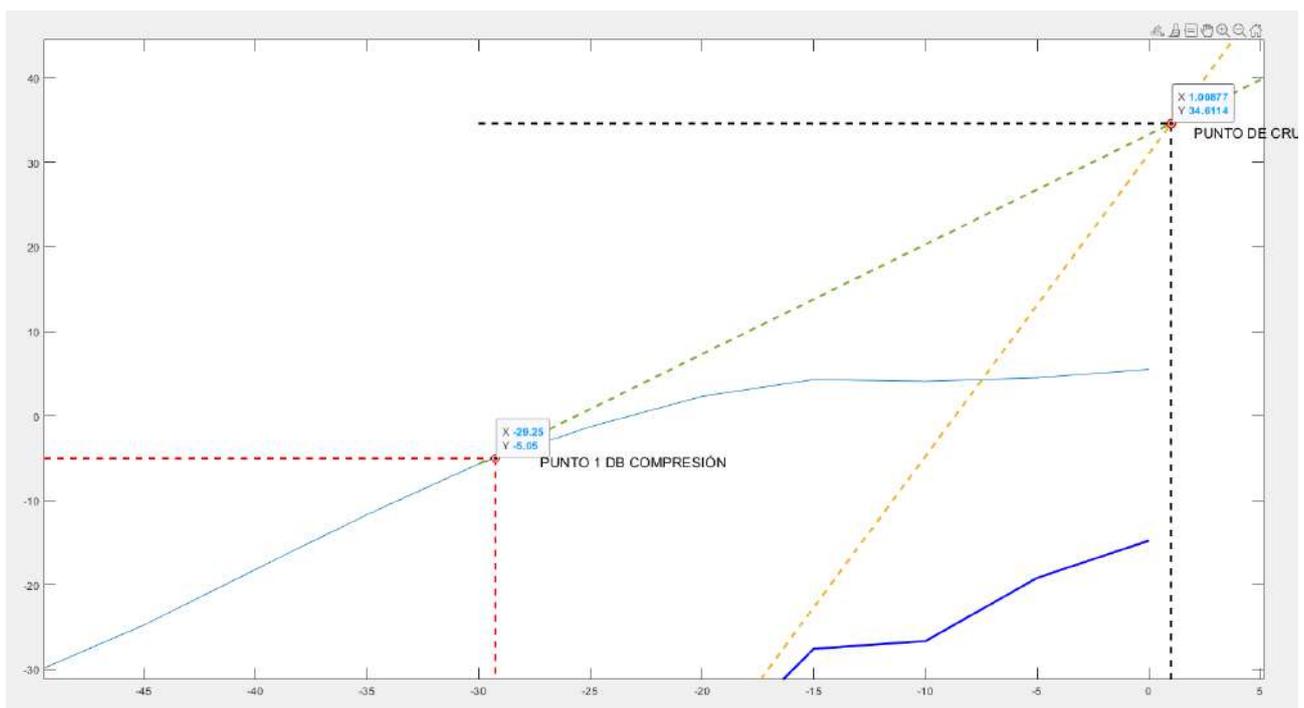


Figura 6.29: Cálculo del punto 1 dB compresión y el punto de cruce de tercer orden.



Figura 6.30: Consumo de la rama receptora del módulo hardware.

6.4.2. Verificación y resultados de la etapa transmisora módulo hardware

Por último, al implementar el módulo hardware en la rama transmisora, éste amplifica la señal de -6.77 dBm a 32.8 dBm, un factor de ganancia de 39.57 dB, quedando cercano a los 34 dBm o 2.5 W, que es la potencia que emplean los dispositivos AIS clase B para transmitir. En la figura 6.31 se representa tanto la carta de Smith como la curva del parámetro S11 del amplificador, que además requerirá un disipador con las dimensiones calculadas en el apartado 5.3.1.

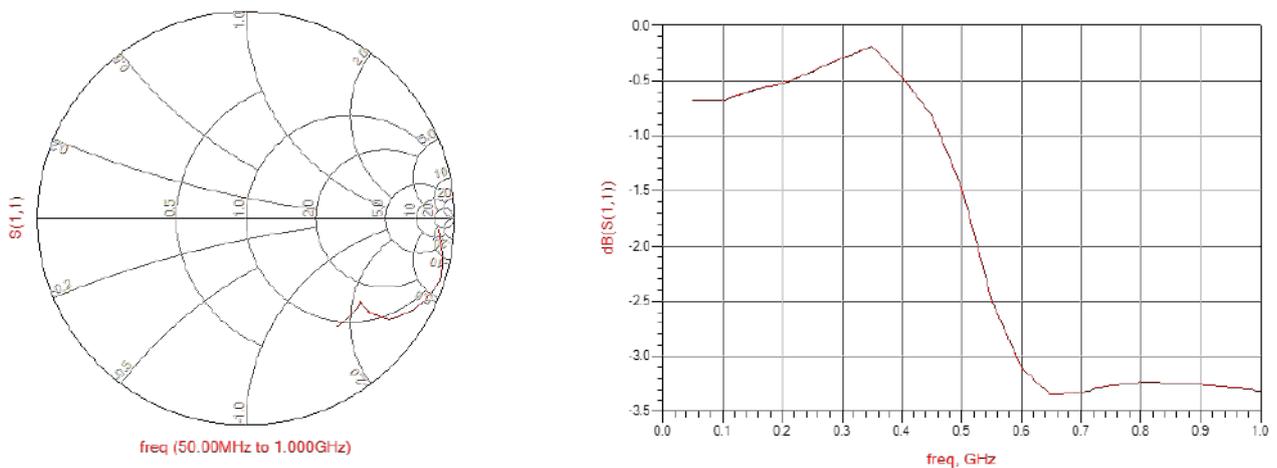


Figura 6.31: Parámetro S11 y carta de Smith de la etapa transmisora del módulo.

7 | Conclusiones y líneas futuras

7.1. Análisis del diseño y conclusiones

Tras el análisis de los resultados obtenidos del desarrollo del prototipo se concluye que, debido a la gran flexibilidad que aporta la tecnología SDR, ésta permite la implementación de diferentes estándares y protocolos, y la posibilidad de actualizar el firmware según las necesidades o cambios en los estándares. Uno de los principales problemas que presenta el prototipo es que tiende a saturar, por lo que hay que escoger los valores de ganancia adecuados para conseguir resultados favorables. Por otra parte, la escasez de transceptores VDES comerciales o su alto coste ha limitado la cantidad de experimentos realizados con los algoritmos de transmisión. Además, durante la realización de este trabajo se han actualizado los estándares del sistema VDES, donde se definen los tamaños de las tramas y sus campos con mayor exactitud.

Asimismo, se han cumplido los objetivos propuestos para este TFM, ya que se realizó un estudio de los estándares AIS y VDES, como de la tecnología SDR y su viabilidad para llevar a cabo la transición al nuevo estándar. Además, la integración del módulo hardware desarrollado mejora las prestaciones que ofrece una plataforma de bajo coste como es el ADALM-PLUTO, asemejando sus características y funcionamiento de los transpondedores de Clase B del AIS.

Después de verificar el comportamiento del prototipo con los diferentes experimentos realizados, los resultados obtenidos son favorables, ya que indican que el uso de tecnología SDR es completamente viable para la implementación de ambos estándares en una misma plataforma. Otro aspecto a destacar ha sido el uso de equipos comerciales como el transpondedor COBALT o el simulador QK-AT011, ya que confirma que el prototipo es completamente compatible con dispositivos diseñados expresamente para las comunicaciones marítimas.

7.2. Líneas futuras

Una vez finalizado este Trabajo Fin de Máster y con los resultados obtenidos, es posible plantear futuras líneas de trabajo a partir de este proyecto. Dentro de las posibles líneas se destacan las siguientes:

- Implementación de los servicios satelitales de VDES, ya que antes de la realización de este trabajo aún no se había publicado las especificaciones de estos servicios en la norma. Actualmente en la última versión del estándar se han detallado características suficientes para realizar su integración.
 - Desarrollo en un único lenguaje, debido a que se utilizan Python y MATLAB en conjunto, una mejora evidente sería la implementación de los algoritmos en un único lenguaje, reduciendo los tiempos de ejecución en los algoritmos de recepción.
 - Integración del resto de mensajes AIS, completando el algoritmo de recepción para que el prototipo pueda recibir cualquier tipo de mensaje. De esta forma no se limita la funcionalidad del prototipo.
 - Cambios en los algoritmos implementados, debido a que los mejores resultados obtenidos han sido del orden del 50 - 60 %, por lo que aun existe margen de mejora.
 - Análisis y mejora del consumo de los algoritmos, no solo energético, sino también el coste computacional asociado. Al optimizar y refactorizar el código de los algoritmos se pueden reducir los costes asociados e incluso mejorar los resultados obtenidos.
 - Mejora del módulo hardware, dando la oportunidad de programar la ganancia necesaria mediante software o manualmente, ya que la plataforma tiende a saturar al emplear sus amplificadores internos y el módulo desarrollado.
 - Integración y comparación con otras plataformas SDR. De esta forma se puede comprobar el rendimiento de los algoritmos en diferentes plataformas y comparar su efectividad en cada una de ellas.
-

Parte II

Bibliografía

1 | Bibliografía

- [1] W. Heijden e I. A. of Lighthouse Authorities, *AIS in a Historic Perspective: A History of the Identification of Ships*. IALA-AISM, 2020, ISBN: 9782910312046.
- [2] M. Ilcev, «New Aspects for Modernization Global Maritime Distress and Safety System (GMDSS),» *TransNav, the International Journal on Marine Navigation and Safety of Sea Transportation*, vol. 14, págs. 991-998, ene. de 2020. DOI: 10.12716/1001.14.04.26.
- [3] *La digitalización de las comunicaciones marítimas*, Spanish. Gradiant, nov. de 2019.
- [4] «Technical characteristics for an automatic identification system using time division multiple access in the VHF maritime mobile frequency band (ITU-R M.1371-5),» ITU, RECOMMENDATION ITU-R M.1371-5, 2014.
- [5] *Automatic Identification System (AIS) on Ships*, 2021, <https://e-navigation.canada.ca/topics/traffic/docs/ais/index-en>, Online; accessed 5 September 2022.
- [6] S. Fossen y T. I. Fossen, «Extended Kalman Filter Design and Motion Prediction of Ships Using Live Automatic Identification System (AIS) Data,» en *2018 2nd European Conference on Electrical Engineering and Computer Science (EECS)*, 2018, págs. 464-470. DOI: 10.1109/EECS.2018.00092.
- [7] M. Tichavska y B. Tovar, *Environmental cost and eco-efficiency from vessel emissions in Las Palmas Port*, 2015. DOI: 10.1016/j.tre.2015.09.002.
- [8] N. Molina, F. Cabrera, V. Araña y M. Tichavska, «An Overview About the Physical Layer of the VHF Data Exchange System (VDDES),» en *Computer Aided Systems Theory – EUROCAST 2019*, R. Moreno-Díaz, F. Pichler y A. Quesada-Arencibia, eds., Cham: Springer International Publishing, 2020, págs. 67-74, ISBN: 978-3-030-45093-9.

- [9] C. B. Ali Wael, N. Armi, T. Miftahushudur, Y. S. Amrullah y R. Sariningrum, «NC-OFDM Transceiver Design for Maritime Cognitive Radio,» en *2018 International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET)*, 2018, págs. 117-120. DOI: 10.1109/ICRAMET.2018.8683927.
- [10] L. Zhang, B. Lin, S. Zhang, T. Zhao e Y. Zhang, «An Improved MAC Protocol Design in VHF Data Exchange System (VDES) For Internet of Vessels,» *Procedia Computer Science*, vol. 129, págs. 45-51, 2018, 2017 INTERNATIONAL CONFERENCE ON IDENTIFICATION, INFORMATION AND KNOWLEDGE IN THE INTERNET OF THINGS, ISSN: 1877-0509. DOI: <https://doi.org/10.1016/j.procs.2018.03.042>. dirección: <https://www.sciencedirect.com/science/article/pii/S1877050918302655>.
- [11] «Technical characteristics for a VHF data exchange system in the VHF maritime mobile band (ITU-R M.2092-0),» ITU, RECOMMENDATION ITU-R M.2092-0, 2015.
- [12] F. Lazaro, R. Raulefs, W. Wang, F. Clazzer y S. Plass, «VHF Data Exchange System (VDES): An enabling technology for maritime communications,» *CEAS Space Journal*, vol. 11, págs. 55-63, mar. de 2019. DOI: 10.1007/s12567-018-0214-8.
- [13] «Interim solutions for improved efficiency in the use of the band 156-174 MHz by stations in the maritime mobile service (ITU-R M.1084-5),» ITU, RECOMMENDATION ITU-R M.1084-5, 2012.
- [14] D. Yang, L. Wu, S. Wang, H. Jia y K. X. Li, «How big data enriches maritime research a critical review of Automatic Identification System (AIS) data applications,» *Transport Reviews*, vol. 39, n.º 6, págs. 755-773, 2019. DOI: 10.1080/01441647.2019.1649315.
- [15] S. Gössling, C. Meyer-Habighorst y A. Humpe, «A global review of marine air pollution policies, their scope and effectiveness,» *Ocean & Coastal Management*, vol. 212, pág. 105 824, 2021, ISSN: 0964-5691. DOI: <https://doi.org/10.1016/j.ocecoaman.2021.105824>. dirección: <https://www.sciencedirect.com/science/article/pii/S0964569121003070>.
- [16] «Automatic identification system VHF data link loading (ITU-R M.2287-0(1),» ITU, REPORT ITU-R M.2287-0, 2013.
- [17] A. Linz y A. Hendrickson, «Efficient implementation of an I-Q GMSK modulator,» *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 43, n.º 1, págs. 14-23, 1996. DOI: 10.1109/82.481470.
-

-
- [18] N. M. Padrón, *Diseño e implementación de un prototipo hardware para un banco de pruebas del estándar AIS*, 2015.
- [19] R. B. Langley, «NMEA 0183 : A GPS Receiver Interface Standard,» 2004.
- [20] *AIVDM/AIVDO protocol decoding*, 2021, <https://gpsd.gitlab.io/gpsd/AIVDM.html>, Online; accessed 5 September 2022.
- [21] T. F. Collins, R. Getz, D. Pu y A. M. Wyglinski, *Software-defined radio for engineers*. Analog Devices, 2018, ISBN: 978-1-63081-457-1.
- [22] R. Lackey y D. Upmal, «Speakeasy: the military software radio,» *IEEE Communications Magazine*, vol. 33, n.º 5, págs. 56-61, 1995. DOI: 10.1109/35.392998.
- [23] P. Cook y W. Bonser, «Architectural overview of the SPEAKeasy system,» *IEEE Journal on Selected Areas in Communications*, vol. 17, n.º 4, págs. 650-661, 1999. DOI: 10.1109/49.761042.
- [24] J. Mitola, «The software radio architecture,» *IEEE Communications Magazine*, vol. 33, n.º 5, págs. 26-38, 1995. DOI: 10.1109/35.393001.
- [25] D. S. S. Rodríguez, *Desarrollo de las librerías de control e interfaz gráfica en un entorno linux para el SDR-1000*, 2005.
- [26] B. Stewart, K. Barlee, D. Atkinson y L. Crockett, *Software-defined radio using MATLAB & Simulink and the RTL-SDR*. Strathclyde Academic Media, 2015.
- [27] L. Mili, T. Saramaki y R. Breeovic, «Multirate Filters: An Overview,» ene. de 2007, págs. 912-915. DOI: 10.1109/APCCAS.2006.342190.
- [28] R. Woods, J. McAllister, G. Lightbody e Y. Yi, *FPGA-based Implementation of Signal Processing Systems*, English. John Wiley, 2017.
- [29] A. Blais, «Feasibility of a Direct Sampling Dual-Frequency SDR Galileo Receiver for Civil Aviation,» sep. de 2014.
- [30] R. Reeder, «Radically Extending Bandwidth to Crush the X-Band Frequencies Using a Track-and-Hold Sampling Amplifier and RF ADC,» *Analog Dialogue*, vol. 51, n.º 09, 2017. dirección: <https://www.analog.com/media/en/analog-dialogue/volume-51/number-4/articles/radically-extending-bandwidth-to-crush-the-x-band-frequencies.pdf>.
-

- [31] J. Thor y D. Akos, «A direct RF sampling multifrequency GPS receiver,» en *2002 IEEE Position Location and Navigation Symposium (IEEE Cat. No.02CH37284)*, 2002, págs. 44-51. DOI: 10.1109/PLANS.2002.998887.
- [32] B. Brannon, «Where Zero-IF Wins: 50 % Smaller PCB Footprint at the Cost,» *Analog Dialogue*, vol. 50, n.º 09, 2016. dirección: <https://www.analog.com/media/en/analog-dialogue/volume-50/number-3/articles/where-zero-if-wins.pdf>.
- [33] M. M. R. R y J. H.-J, «IQ-imbalance and its compensation for non-ideal analog receivers comprising frequency-selective components,» *Advances in Radio Science - Kleinheubacher Berichte*, vol. 4, sep. de 2006. DOI: 10.5194/ars-4-189-2006.
- [34] S. Arar, «Basics of Digital Down-Conversion in DSP,» *All About Circuits*, Article, 2018.
- [35] *About RTL-SDR*, 2013, <https://www.rtl-sdr.com/about-rtl-sdr/>, Online; accessed 5 September 2022.
- [36] *About GNU Radio*, 2014, <https://www.gnuradio.org/about/>, Online; accessed 5 September 2022.
- [37] *Gqrx SDR*, 2018, <https://gqrx.dk/>, Online; accessed 5 September 2022.
- [38] *bladeRF 2.0 micro*, 2018, <https://www.nuand.com/bladerf-2-0-micro/>, Online; accessed 5 September 2022.
- [39] *ADALM-PLUTO Overview*, 2021, <https://wiki.analog.com/university/tools/pluto>, Online; accessed 5 September 2022.
- [40] *Customizing the Pluto configuration*, 2021, <https://wiki.analog.com/university/tools/pluto/users/customizing>, Online; accessed 5 September 2022.
- [41] *ADALM-PLUTO for End Users*, 2021, <https://wiki.analog.com/university/tools/pluto/users>, Online; accessed 5 September 2022.
- [42] S.-H. Kim, C.-H. Lee, G.-W. Kim y D.-H. Cho, «Utilizing Software-Defined Radio, Reception Test of AIS Payload Used in a Cube-Satellite,» *Journal of Space Technology and Applications*, vol. 2, n.º 2, págs. 121-136, 2022. DOI: 10.52912/jsta.2022.2.2.121. dirección: <https://doi.org/10.52912/jsta.2022.2.2.121>.
- [43] S. Kaya, A. Ç. Yapc, B. Tbkolu y. O. Yazc, «Implementation of FMCW Radar by Using SDR,» en *2021 13th International Conference on Electrical and Electronics Engineering (ELECO)*, 2021, págs. 234-238. DOI: 10.23919/ELECO54474.2021.9677650.
-

-
- [44] C. Gravelle y R. Zhou, «SDR Demonstration of Signal Classification in Real-Time Using Deep Learning,» en *2019 IEEE Globecom Workshops (GC Wkshps)*, 2019, págs. 1-5. DOI: 10.1109/GCWkshps45667.2019.9024661.
- [45] E. O. Kandaurova y D. Chirov, «Algorithm and Software for Intelligent Analysis of the Frequency Spectrum for Cognitive Radio Systems,» en *2020 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO)*, 2020, págs. 1-5. DOI: 10.1109/SYNCHROINFO49631.2020.9166031.
- [46] W. Liu, Q. Meng, C. Wang, C. Zhou, S. Yao e I. Tariq, «An efficient channelization architecture and its implementation for radio astronomy,» *Journal of Instrumentation*, vol. 16, n.º 08, P08047, ago. de 2021. DOI: 10.1088/1748-0221/16/08/p08047. dirección: <https://doi.org/10.1088/1748-0221/16/08/p08047>.
- [47] M. Rice y M. McLernon, «Teaching Communications with SDRs: Making It Real for Students,» *IEEE Communications Magazine*, vol. 57, n.º 11, págs. 14-19, 2019. DOI: 10.1109/MCOM.001.1900185.
- [48] *N200/N210 - Ettus Knowledge Base*, 2014, <https://kb.ettus.com/N200/N210>, Online; accessed 5 September 2022.
- [49] *¿Qué es LabVIEW?* 2021, <https://www.ni.com/es-es/shop/labview.html>, Online; accessed 5 September 2022.
- [50] *Documentación sobre PyAIS*, 2022, <https://pypi.org/project/pyais/>, Online; accessed 5 September 2022.
- [51] *Class A position report type 1, 2 and 3*, 2022, https://gpsd.gitlab.io/gpsd/AIVDM.html#_types_1_2_and_3_position_report_class_a, Online; accessed 5 September 2022.
- [52] A. G. Orozco-Lugo, M. M. Lara, E. Alameda-Hernandez, S. Moosvi y D. C. McLernon, «Frequency Offset Estimation and Compensation Using Superimposed Training,» en *2007 4th International Conference on Electrical and Electronics Engineering*, 2007, págs. 118-121. DOI: 10.1109/ICEEE.2007.4344988.
- [53] «Technical characteristics for a VHF data exchange system in the VHF maritime mobile band (ITU-R M.2092-1),» ITU, RECOMMENDATION ITU-R M.2092-1, 2022.
-

- [54] J. M. R. S. Manuel Sierra Pérez Juan de la Calle García y F. G. Muñiz, *Electrónica de comunicaciones*, Spanish. 1994.
- [55] *DISIPADORES TÉRMICOS PARA DISPOSITIVOS ELECTRÓNICOS*, 2016, http://ieb-srv1.upc.es/gieb/tecnicas/pdf/disipadores_termicos.pdf, Online; accessed 5 September 2022.
- [56] *MOSFET de potencia: Cálculo de disipador y temperaturas*, 2020, <https://www.laboratoriogluon.com/mosfet-de-potencia-calculo-de-disipador-y-temperaturas/>, Online; accessed 5 September 2022.
- [57] *Understanding Thermal Dissipation and Design of a Heatsink*, 2011, <https://www.ti.com/lit/an/slva462/slva462.pdf?ts=1653633453446>, Online; accessed 5 September 2022.
- [58] *Cobalt AIS Class B*, 2016, <https://srt-marine.com/product/ais-transceiver-modules/cobalt-3/>, Online; accessed 5 September 2022.
- [59] *MarineTraffic*, 2007, <https://www.marinetraffic.com/en/ais/home/centerx:-15.5/centery:28.0/zoom:11>, Online; accessed 5 September 2022.
- [60] ULPGC, *Boletín Oficial de la Universidad de Las Palmas de Gran Canaria*, 2021, https://www.ulpgc.es/sites/default/files/ArchivosULPGC/boulpgc/BOULPGC/boulpgc_1_julio_2021.pdf, Online; accessed 5 September 2022.
- [61] COIT, *Derechos de visado 2021*, 2021, https://www.coit.es/sites/default/files/20210204_tarifas_2021_visado.pdf, Online; accessed 5 September 2022.
- [62] *PyBOMBS*, 2016, <https://github.com/gnuradio/pybombs>, Online; accessed 5 September 2022.
- [63] *GNU Radio and IIO Devices: gr-iio*, 21, <https://wiki.analog.com/resources/tools-software/linux-software/gnuradio>, Online; accessed 5 September 2022.
- [64] M. Á. F. G. de Aranda, *Introducción a SDR con GNU Radio*, Spanish. 2020.
- [65] D. Romero-Godoy, N. Molina-Padrón, F. Cabrera, V. Araña y E. Jiménez, «Design and implementation of a prototype with a low-cost SDR platform for the next generation of maritime communications,» en *2022 3rd URSI Atlantic and Asia Pacific Radio Science Meeting (AT-AP-RASC)*, 2022, págs. 1-4. DOI: 10.23919/AT-AP-RASC54737.2022.9814184.
-

Parte III

Presupuesto

| Presupuesto

P.1. Introducción

Las siguientes secciones desarrollan el coste para las diversas categorías presupuestarias consideradas en este trabajo.

- Recursos materiales.
 - Recursos hardware.
 - Recursos software.
- Recursos humanos.
- Redacción del documento.
- Derechos de visado del COIT (Colegio Oficial Ingenieros de Telecomunicaciones).
- Gastos de tramitación y envío.

Después de analizar estos puntos, se aplicarán los impuestos y se calculará el costo total.

P.2. Recursos Materiales

Con este concepto se agrupan los gastos que vienen dados por el uso de los recursos hardware y software. La forma de modelar el coste final será mediante la amortización lineal (ecuación P.1), teniendo en cuenta que no se tomará en consideración el valor de mantenimiento y que el valor residual es nulo. El coste final solo tendrá en cuenta el periodo de uso de cada uno de los equipos durante el transcurso del TFM.

$$C_f = \frac{V_{ad} - V_{res}}{A_u} \quad (\text{P.1})$$

donde C_f es el coste final, V_{ad} el valor de adquisición, V_{res} el valor residual y A_u los años de vida útil.

P.2.1. Recursos Hardware

Para calcular los costes se tiene en cuenta que los recursos fueron adquiridos antes de la realización de este TFM. A su vez, se tienen en cuenta los componentes fungibles utilizados como la PCB manufacturada, sus componentes integrados, cables y conectores. El coste de estos recursos se recoge en la tabla P.1. Además, se tienen en cuenta los recursos hardware no fungibles, como el ordenador personal, instrumentación de medidas y la plataforma SDR utilizada. Se considera que estos recursos materiales tienen una vida útil de 6 años y los equipos de medida utilizados se estima su vida útil de 10 años. El coste asociado de todos los recursos se muestra en la tabla P.2.

Tabla P.1: Coste de los recursos hardware fungibles.

Recursos Hardware Fungibles			
Descripción	Unidades	Valor de adquisición	Coste final
PCB + envío	1	50 €	50 €
Componentes PCB	1	288.66 €	288.66 €
Cables USB-microUSB	2	2.89 €	5.78 €
Cables SMA macho - SMA macho 50cm	2	1.83 €	3.66 €
Par de cables banana-cocodrilo	2	0.86 €	1.72 €
Transición BNC-SMA	2	0.43 €	0.86 €
Transición tipo N-SMA	1	2.04 €	2.04 €
Antenas VHF	2	24.5 €	49 €
Antenas GPS	1	15.85 €	15.85 €
Antena AIS	1	122.00 €	122 €
Total:			539.57 €

Tabla P.2: Coste de los recursos hardware no fungibles.

Recursos Hardware No Fungibles					
Descripción	Tiempo	Valor de adquisición	Amortización anual	Coste final	
Portátil Lenovo Ideapad 330	4 meses	615.24 €	102.54 €	34.18 €	
ADALM-PLUTO	4 meses	202.00 €	33.66 €	11.22 €	
SRT Marine Cobalt	4 meses	250.00 €	41.67 €	13.89 €	
QK-AT011 Simulador AIS	4 meses	250.00 €	41.67 €	4.74 €	
Analizador de espectro Advantest R3131A	4 meses	3.970,00 €	397.00 €	132.33 €	
Fuente de alimentación Agilent E3620A	4 meses	610.30 €	31.03 €	20.34 €	
Generador de funciones SIGLENT SSG3032X	4 meses	3.500,00 €	350.00 €	116.67 €	
Total:		9.397,54 €	4.147,57 €	333.37 €	

Por lo tanto, el coste final de los recursos hardware es de OCHOCIENTOS SETENTA Y DOS EUROS Y NOVENTA Y CUATRO CÉNTIMOS.

P.2.2. Recursos Software

Para la realización de este TFM se utilizaron los recursos software mostrados en la tabla P.3. Como se hizo en la parte de recursos hardware, el coste final depende del tiempo de utilización de cada uno de ellos durante el período de trabajo correspondiente, considerándose en este caso una vida útil de 3 años para cada uno de ellos.

Por lo tanto, el coste final para recursos software es de QUINIENTOS SETENTA Y TRES EUROS Y DOCE CÉNTIMOS.

Tabla P.3: Coste de los recursos software.

Recursos Software				
Descripción	Tiempo	Valor de adquisición	Amortización anual	Coste final
MATLAB R2020b	4 meses	2.000,00 €	666.67 €	166.67 €
Pack ADS y GENESYS	4 meses	4.732,45 €	1577.48 €	394.37 €
SO Windows 10	4 meses	145.00 €	48.33 €	12.08 €
GNU-Radio Companion	4 meses	0.00 €	0.00 €	0.00 €
Pro-AIS	4 meses	0.00 €	0.00 €	0.00 €
QK-AT011 Software	4 meses	0.00 €	0.00 €	0.00 €
Overleaf editor LaTeX online	4 meses	0.00 €	0.00 €	0.00 €
Total:		6.877,45 €	2.292,48 €	573.12 €

P.3. Recursos Humanos

Este es el coste asociado al estudiante, que ha empleado 4 meses a media jornada en las tareas de estudio previo, desarrollo de algoritmos y hardware, implementación e integración del prototipo y documentación. Para el cálculo se ha tenido en cuenta que el proyecto se desarrolla dentro del entorno de investigación de la Universidad de Las Palmas de Gran Canaria, el tipo de tarifa aplicada corresponde al de personal investigador técnico (máster) según la tabla de contrataciones de personal con cargo de proyectos, programas, convenios y contratos, correspondiente al BOULPGC de julio de 2021 [60]. Así, el coste a tiempo completo (7.5 horas diarias) es el que se muestra en la tabla P.4.

Tabla P.4: Coste de los recursos humanos.

Recursos Humanos			
Personal	Coste total mensual	Tiempo	Total
Ingeniero	1.801,02 €	4 meses	7.204,08 €

El coste total de recursos humanos es de SIETE MIL DOSCIENTOS CUATRO EUROS Y OCHO CÉNTIMOS.

P.4. Redacción del documento

El importe de la redacción del proyecto se calcula de acuerdo con la siguiente expresión:

$$R = 0.07 \cdot P \cdot C_n \quad (\text{P.2})$$

donde P es el presupuesto del proyecto y C_n es el coeficiente de ponderación en función del presupuesto. En este caso el coeficiente C1 está fijado a 1.00 para presupuestos menores a 30.050 €.

En la tabla P.5 se muestra el presupuesto calculado hasta el momento. Por lo que los costes derivados de la redacción del documento se obtienen sustituyendo los valores en la ecuación P.2.

Tabla P.5: Presupuesto parcial.

Recurso	Coste
Recursos Materiales Hardware	872,94 €
Recursos Materiales Software	573,12 €
Recursos Humanos	7.204,08 €
Total:	8.650,14 €

$$R = 0.07 \times 8.650,14 \times 1 = 605,51 \text{ €}$$

El coste de redacción asciende a un total de SEISCIENTOS CINCO EUROS Y CINCUENTA Y UN CÉNTIMOS.

P.5. Derechos de visado del COIT

Para proyectos de carácter general [61], los derechos de visado se calculan de acuerdo con la siguiente fórmula:

$$0.006 \cdot P_1 \cdot C_1 + 0.003 \cdot P_2 \cdot C_2 \quad (\text{P.3})$$

donde P_1 = Presupuesto general para este proyecto, P_2 = Presupuesto de ejecución material que corresponde a la obra civil, C_1 = Coeficientes Reductores (C) correspondientes a P1 y C_2 = Coeficientes Reductores (C) correspondientes a P2.

De la sección anterior se tiene que el coeficiente C_1 está fijado a 1.00. En este caso, para el desarrollo de este TFM no se ha requerido obra civil por lo que el valor de P_2 es 0.00 €. Asimismo, el valor del presupuesto P_1 se calcula sumando los costes de las secciones anteriores. Esto viene incluido en la tabla P.6.

Tabla P.6: Presupuesto con el coste de redacción.

Recurso	Coste
Recursos Materiales Hardware	872,94 €
Recursos Materiales Software	573,12 €
Recursos Humanos	7.204,08 €
Redacción del documento	605,51 €
Total:	9.255,65 €

Sustituyendo los valores en la ecuación P.3.

$$R = 0.006 \times 9.255,65 \times 1 = 55,53 \text{ €}$$

El coste por derecho de visado es de CINCUENTA Y CINCO EUROS Y CINCUENTA Y TRES CÉNTIMOS.

P.6. Gastos de tramitación y envío

Cada documento visado por vía telemática tiene un coste de 6.00 € según [61].

P.7. Aplicación de impuestos y coste final

El resultado final con todos los apartados desglosados en la tabla P.7, estableciendo el coste total del trabajo presentado.

Tabla P.7: Presupuesto total.

Presupuesto Final	
Partidas	Totales
Recursos Materiales	
- Recursos Hardware	872,94 €
- Recursos Software	573,12 €
Recursos Humanos	7.204,08 €
Total parcial:	8.650,14 €
Redacción del documento	605,51 €
Visado del COIT	55,53 €
Gastos de tramitación y envío	6,0 €
Total:	9.317,18 €
IGIC (7%):	652,20 €
Total después de impuestos:	9.969,38 €

El presupuesto total teniendo en cuenta los impuestos asciende a NUEVE MIL NOVECIENTOS SESENTA Y NUEVE Y TREINTA Y OCHO CÉNTIMOS.

Las Palmas de Gran Canaria a 5 de septiembre de 2022

Fdo.:

Parte IV

Anexos

A | Instalación de GNU Radio en Linux

GNU Radio [36] es una herramienta de desarrollo libre y abierta que provee bloques de procesamiento de señal para implementar sistemas SDR. Es ampliamente utilizada en ambientes académicos, aficionados y de investigación, aunque también existen soluciones comerciales implementadas con este software. Su funcionamiento es similar a Simulink, ya que las aplicaciones de GNU Radio se programan mediante un entorno gráfico utilizando diferentes bloques. Los bloques se pueden programar utilizando Python y C++, aunque este último caso es más común para crear nuevas bibliotecas.

En este anexo se explicará el proceso de instalación de esta herramienta para sistemas Linux. Para ello, se utilizó una máquina virtual con Ubuntu 20.04 recién instalado. Aparte de la instalación, se explicará brevemente el flujo de trabajo que sigue esta herramienta junto a un ejemplo de receptor radio FM.

Existen diferentes métodos de instalación, como el compilado de código fuente, instalación mediante PyBOMBS [62] o en este caso, el uso de repositorios personales o PPA (*Personal Package Archives*). Toda la instalación se realizará mediante consola o terminal, por lo que se introducen los siguientes comandos:

- `sudo add-apt-repository ppa:gnuradio/gnuradio-releases-3.7`
- `sudo apt update`
- `sudo apt install gnuradio`

Los repositorios PPA son repositorios software diseñados para los usuarios de Ubuntu y son más fáciles de instalar que otros repositorios de terceros. Los PPAs se utilizan a menudo para distribuir software de pre-lanzamiento para que pueda ser probado, por lo que a la hora de utilizarlos, es importante conocer si son de confianza. Tras la descarga e instalación de los paquetes de GNU Radio,

la herramienta está lista para su uso escribiendo en la terminal el comando *gnuradio-companion*. Sin embargo, para poder utilizar la plataforma ADALM-PLUTO, es necesario instalar los paquetes y controladores necesarios. Analog Devices, La empresa creadora del ADALM-PLUTO, explica en [63] como añadir los drivers y paquetes en GNU Radio. En primer lugar, se instalan las bibliotecas necesarias para los bloques. Luego, se descargarán los drivers y bloques para GNU Radio desde los repositorios oficiales de Analog Devices.

- `sudo apt install liborc-dev libxml2 libxml2-dev bison flex cmake git libaio-dev libboost-all-dev doxygen libusb-1.0-0-dev libavahi-common-dev libavahi-client-dev bison flex cmake git libgmp-dev swig -y`
- Descarga e instalación de los drivers del ADALM-PLUTO
 - `git clone https://github.com/analogdevicesinc/libiio.git`
 - `cd libiio`
 - `cmake .`
 - `make`
 - `sudo make install`
 - `cd ..`
- Descarga e instalación de los bloques del ADALM-PLUTO para GNU Radio
 - `git clone -b upgrade-3.8 https://github.com/analogdevicesinc/gr-iio.git`
 - `cd gr-iio`
 - `cmake .`
 - `make`
 - `sudo make install`
 - `cd ..`
 - `sudo ldconfig`

Tras este paso, al ejecutar GNU Radio, saldrá un mensaje de alerta antes de mostrar la interfaz de la herramienta donde indica que los paquetes `gr-iio` (bloques del ADALM-PLUTO) se encuentran bloqueadas. Para corregir este aviso, hay que establecer algunas variables del sistema, como la ruta de Python y las rutas de las bibliotecas.

- `export PYTHONPATH=/usr/lib/python3/dist-packages:/usr/lib/python3/site-packages:$PYTHONPATH`
- `export LS_LIBRARY_PATH=/usr/lib:$LD_LIBRARY_PATH`
- `export PYTHONPATH=/usr/local/lib/python3/dist-packages:/usr/local/lib/python3.6/dist-packages:$PYTHONPATH`
- `export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH`

En este punto, GNU Radio está preparado para ejecutar tanto los bloques por defecto, como los bloques del ADALM-PLUTO. En la figura A.1 se puede ver la interfaz al ejecutar GNU Radio.

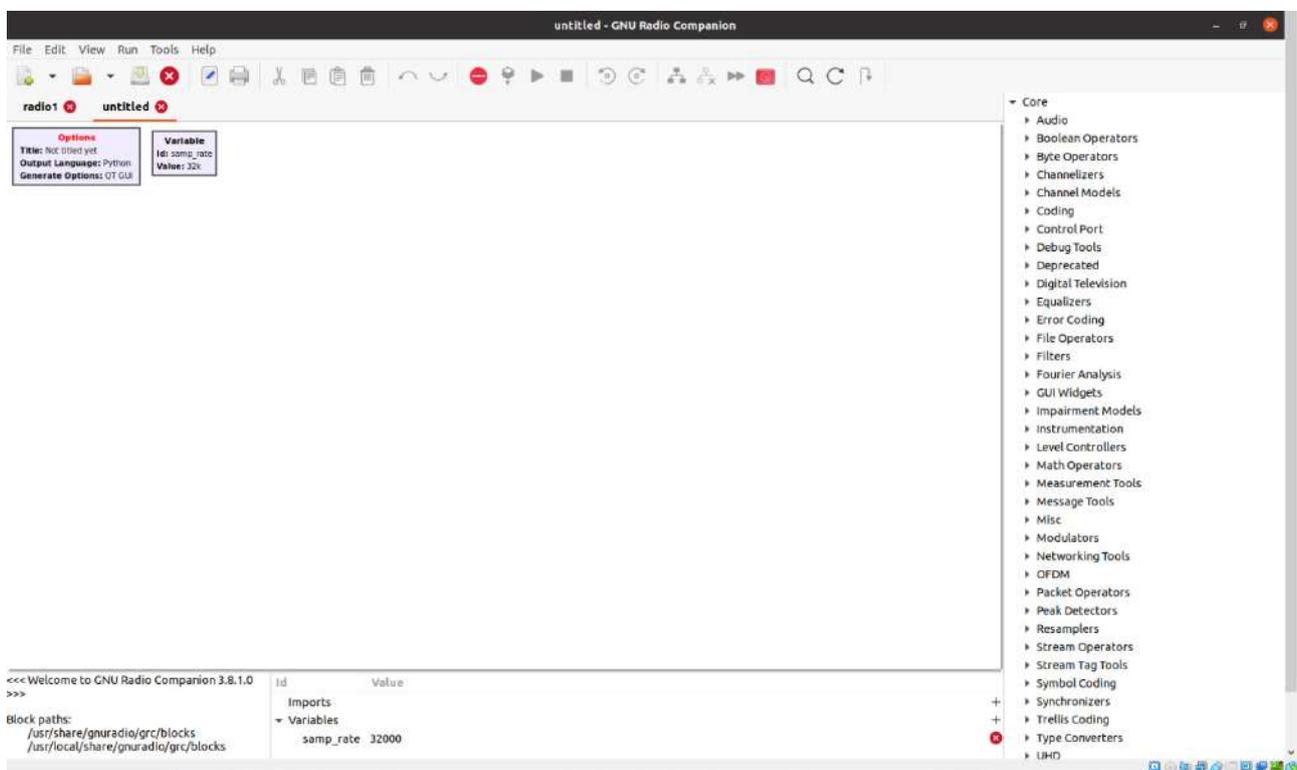


Figura A.1: Interfaz de GNU Radio.

Como se puede observar, existen 3 partes fundamentales en la interfaz de la herramienta: La barra superior, que cuenta con botones para generar el código del diagrama y ejecutarlo, la parte central, donde se colocan los bloques del diagrama, y la parte derecha, donde se ubican los bloques en sus respectivos directorios. Otro aspecto importante a la hora de utilizar la herramienta es necesario conocer los tipos de datos con los que trata y se ven en la figura A.2. Los tipos de datos utilizados en el ejemplo son los *complex float 32* con el que se representan los números complejos y las muestras IQ y los *float 32* que representan la parte real de la señal. La explicación

del funcionamiento de los diagramas de flujo se apoyará en el ejemplo de receptor FM recogido en [64], y que se puede ver en la figura A.3.

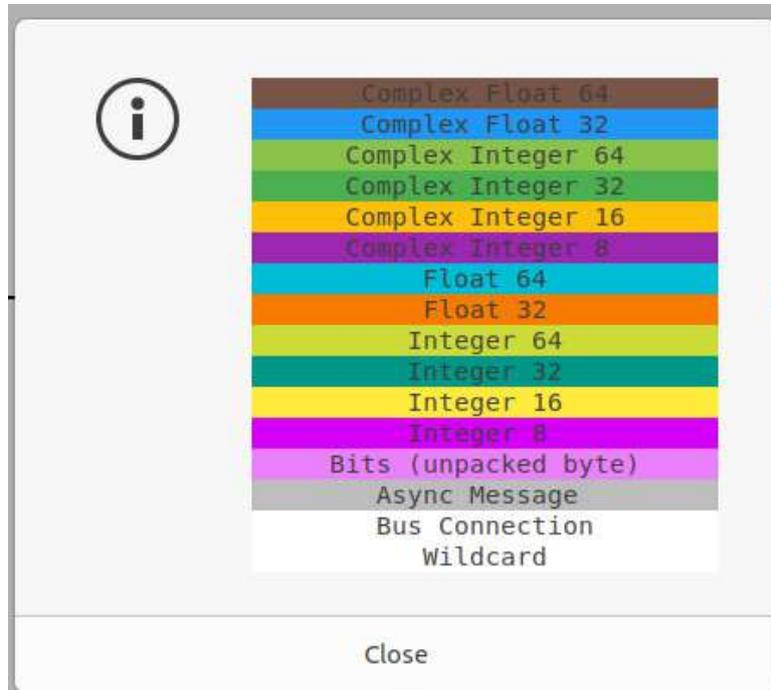


Figura A.2: Tipos de datos en GNU Radio.

Antes de empezar con el análisis del diagrama, cabe señalar los bloques que se encuentran en la esquina superior izquierda. Estos bloques son *Options*, que configura el lenguaje del diagrama cuando se genera y ejecuta, y los bloques de variables. Para este caso se eligió un valor de 2.048 Msps, que es suficiente para el muestreo y acomodará los cálculos más adelante. Siguiendo el orden de izquierda a derecha, el primer elemento es el bloque *PlutoSDR Source*. Todos los diagramas programados desde GNU tienen que tener, como mínimo, dos elementos primordiales, los bloques *source* y los bloques *sink*. Conociendo el significado de sus nombres, fuente y sumidero respectivamente, se entiende que los datos entran al diagrama mediante el bloque *source* y son representados en los bloques *sink*. Para el caso de un transmisor, el bloque *sink* de la plataforma utilizada será el último del diagrama. Dentro de los parámetros de este bloque se encuentran la frecuencia del oscilador local, establecida en 94.4 MHz (Frecuencia del canal “40 Principales”), la tasa de muestreo, ancho de banda de recepción, tamaño del buffer, ganancia y correcciones de componente continua, tanto en la frecuencia de sintonía como en banda base.

El siguiente bloque tras el ADALM-PLUTO es un filtro paso-bajo. Como se ha visto en el capítulo 3, los SDR trabajan en banda base, por lo que es el filtro ideal. Sus parámetros de configuración son el factor de diezmado, que está establecido a 1 para que no modifique la señal, los

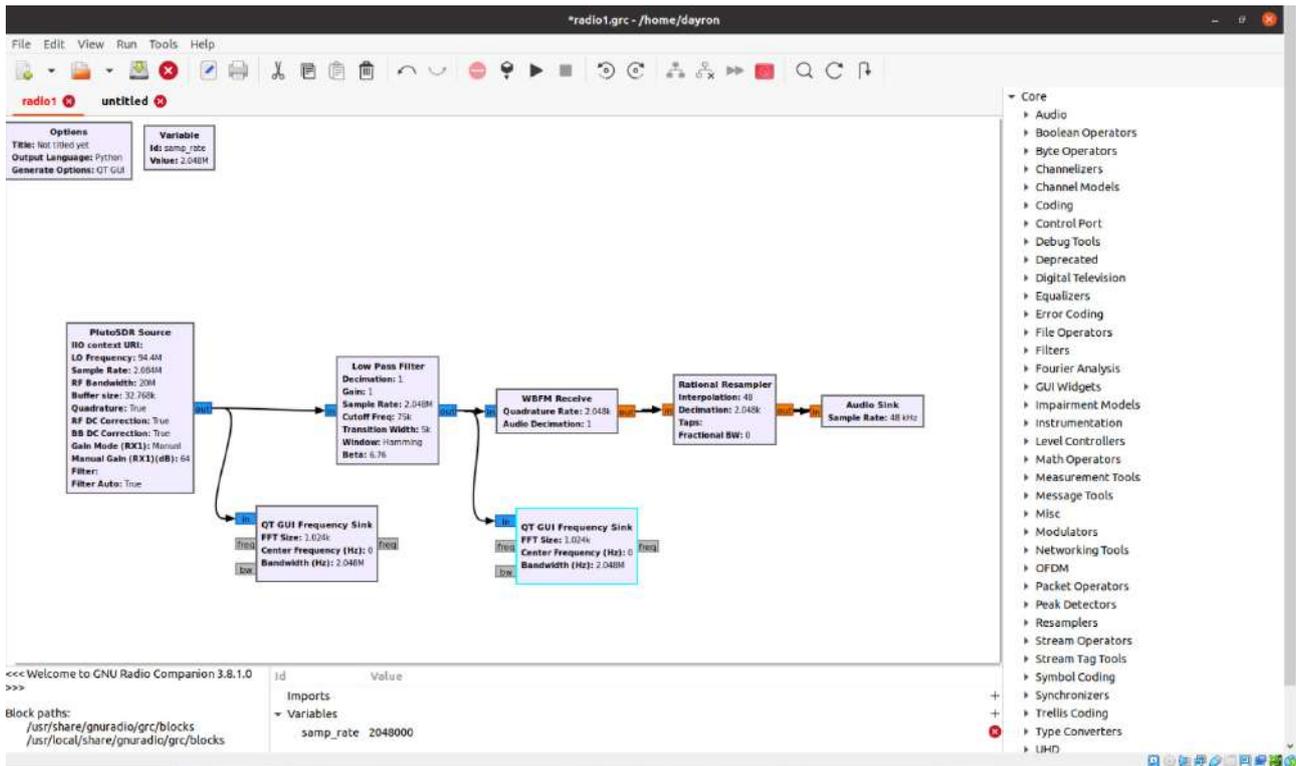


Figura A.3: Diagrama de bloques de un receptor radio FM.

proceso de diezmado e interpolación los realiza un bloque posterior, ganancia, tasa de muestreo y frecuencia de corte del filtro, establecida a 75 KHz, siendo el ancho de banda de señal útil definida para transmisiones de radio FM.

El bloque tras el filtro es el demodulador FM de banda ancha, que transformará la señal de componentes IQ compleja en una señal real. Sus parámetros son el *Quadrature Rate*, cuyo valor debe coincidir con la frecuencia de muestreo del diagrama, y el diezmador de audio, que se dejará a 1 como el caso del filtro.

El penúltimo bloque, el *Rational Resampler*, es otra de las bases de GNU Radio, ya que se encarga de interpolar y diezmarse, cambiando la frecuencia de muestreo según se necesite. En este diagrama tiene los valores de diezmado a 2048 muestras por segundos y el factor de interpolación a 48. Como se quiere recibir una señal de audio, éstas son muestreadas a una frecuencia de 48 KHz, ya que recoge todas las componentes del espectro audible. Para comprobar que el resultado es el deseado, se sigue la ecuación A.1.

$$f_{s_o} = f_{s_i} \cdot \frac{F_I}{F_D} = 2,048 \cdot 10^6 \cdot \frac{48}{2048} = 48000 \text{ muestras por segundo}; \quad (\text{A.1})$$

donde F_I es el factor de interpolación y F_D el factor de diezmado.

Por último, tras preparar la señal para su reproducción, se tiene el bloque *Audio Sink*, que se encarga de reproducir la señal que entra al bloque. Su único parámetro es la frecuencia de muestreo, que tras pasar por el *Rational Resampler* tiene el valor de 48000 muestras por segundo. Con las pruebas realizadas en Linux, la señal no se escucha y es un problema respecto al sistema operativo, ya que con el mismo diagrama pero en Windows, se puede escuchar esta señal.

Además de estos bloques se cuentan con otros bloques *Sink*, que en este caso son analizadores de espectros. Es importante que las frecuencias de muestreo coincidan en cada punto que se quiere medir, ya que de no ser así, la señal se representa mal. En la figura A.4 se puede ver el espectro recibido después y antes de pasar por el filtro.

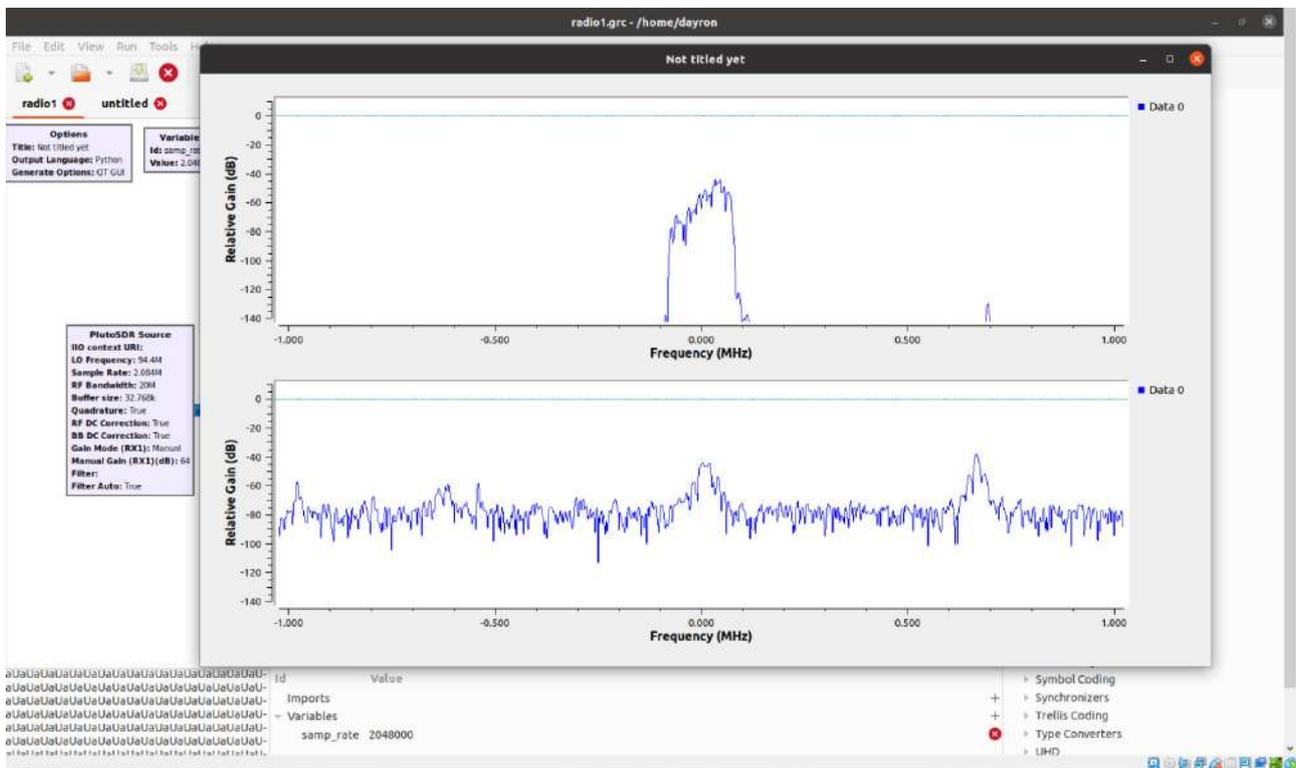


Figura A.4: Señal de radio FM recibida por el ADALM-PLUTO en GNU Radio.

B | Divulgación de resultados

Durante la realización del Trabajo Fin de Máster, titulado "*Diseño de un transceptor para comunicaciones marítimas digitales sobre una plataforma de radio definida por software*" se presentó un artículo en el *Congreso URSI AT-AP-RASC* en junio de 2022, celebrado en Maspalomas, Gran Canaria (España) [65].

En este artículo se evalúa la posibilidad de utilizar una plataforma SDR de bajo costo como el ADALM-PLUTO como base para un prototipo de transceptor implementando los estándares AIS y VDES. Además, se diseñó una pequeña etapa RF para mejorar el rango dinámico del prototipo para la rama de recepción. Aunque estaba diseñada para recepción, también se evaluó en la rama de transmisión. Los resultados expuestos en el artículo revelan que se puede utilizar plataformas SDR de bajo coste, aunque se ha de suplir sus carencias añadiendo hardware adicional y en los algoritmos de transmisión y recepción. A continuación, se adjunta el artículo publicado.



Design and implementation of a prototype with a low-cost SDR platform for the next generation of maritime communications

D. Romero-Godoy⁽¹⁾, N. Molina-Padrón⁽¹⁾, F. Cabrera⁽¹⁾, V. Araña⁽¹⁾ and E. Jiménez⁽¹⁾

(1) Instituto para el Desarrollo Tecnológico y la Innovación en Comunicaciones (IDeTIC), Universidad de Las Palmas de Gran Canaria, Las Palmas, Spain

Abstract

In the last years, the increasing amount of maritime traffic has compromised the security of the AIS (*Automatic Identification System*). The use of two 25 KHz channels is not enough for the global amount of ships at seas. With the purpose of reinforcing the AIS, the VDES (*VHF Data Exchange System*) has been proposed in 2013. This system adds the AIS and four additional services, with challenges in modulation schemes, channel bandwidths and bit rates. The adoption of this new system implies a hardware challenge, because the VDES equipment could be more expensive than current commercial AIS transceivers. The SDR (Software-Defined Radio) technology is a versatile technology which could reduce the cost of the next generation of maritime communications equipment. In this paper, a prototype for the AIS and VDES is implemented on a low-cost SDR platform. Additionally, an RF circuit for filtering and amplifier in transmission and reception has been designed to improve the SDR performance.

1 Introduction

Since the Titanic's accident in 1912, maritime safety measures have been reinforced through a multiple international standardisations. In 1914, the SOLAS (*Safety Of Life At Sea*) convention was a historic milestone where the first international rules for safety at sea were approved [1]. Nowadays, maritime communications tend to transmit more information, optimising the bandwidths. Because of the radio spectrum is a finite and demanded resource, there is a need to replace analog to digital systems to reduce the bandwidth requirements. In relation with this challenge, and to achieve a reduction of human error, a renewal of the GMDSS (*Global Maritime Distress Safety System*) has been proposed [2]. This renewal has meant a digitalisation of the maritime communication systems, where the AIS (*Automatic Identification System*) has been a good example of this transition.

The AIS [3] is a broadcast radio communication system which operates at the VHF maritime band (156.025 MHz – 162.025 MHz). This system, which was developed in 1998 [4], allows the exchange of navigation data between ships and other infrastructures, such as shore stations or aids to

navigation [5]. It operates on two 25 KHz channels, AIS-1 (161.975 MHz) and AIS-2 (162.025 MHz), applying a NRZI line code and a GMSK modulation scheme on the baseband signals. The transmissions are managed by the vessels themselves using a self-organised TDMA (*Time Division Multiple Access*) media access scheme, avoiding a centralised control as it is used in the traditional mobile networks [6]. With the AIS, a vessel can transmit its navigation parameters (position, speed, course, etc.) and receives the same parameters from neighbouring vessels. However, the system's capacity falls on two narrow band channels that share its usage through a TDMA scheme. In a high density of marine traffic scenarios, an increment of packet collisions can be produced, compromising the proper performance of the AIS. The AIS channel saturation has been studied in some areas of the world, such as Mexico or Japan [7].

In order to avoid the AIS channel saturation, the IALA (*International Association of Marine Aids to Navigation and Lighthouse Authorities*) and other authorities proposed a modernisation of the AIS in 2013, called VDES (*VHF Data Exchange System*) [8]. This new system integrates the AIS, and twelve new channels with higher rates and bandwidths are added, with changes on modulation and coding schemes. The VDES services may have fixed 25 KHz channels, as the LR-AIS (*Long Range AIS*) and the ASM (*Application Specific Message*), or variable with 25, 50 or 100 KHz, as the VDE-TER (*VHF Data Exchange - Terrestrial component*) and the VDE-SAT (*VHF Data Exchange - Satellite component*). In addition, the LR-AIS uses the same modulation scheme as the AIS (GMSK), the ASM uses a $\pi/4$ -QPSK scheme, the VDE-TER can use $\pi/4$ -QPSK, 8PSK or 16QAM schemes, and the VDE-SAT uses BPSK, QPSK or 8PSK modulation (in downlink) and QPSK, OQPSK and 16APSK (in uplinks) [9]. The VDE-TER service can achieve transmission rates 32 times higher than the AIS [10]. Despite the positive aspects offered by the VDES, there are still some technical issues to be resolved. Changes in the physical layer of the VDES involve changes in the hardware used for the AIS equipment [11]. In comparison with the current AIS commercial equipment prices, the implementation of the VDES hardware could experiment a prohibitive increment. For this reason, it is necessary to evaluate new alternatives for implementing the

VDES hardware, where the SDR (*Software-Defined Radio*) [12] could be a potential technology to achieve affordable cost for the next generation of maritime communications equipment.

In this paper, a prototype capable of transmitting and receiving signals conformed by the next generation of maritime communications, the AIS and the VDES, is proposed. This prototype is composed by a low-cost SDR platform, which executes a selectable code for the AIS and the VDES services in transmission and reception modes. Moreover, an additional platform for filtering and amplification in the AIS and VDES bands has been implemented to improve the performance of the SDR platform.

2 Prototype's architecture

The prototype's architecture proposed is shown in the block diagram of the figure 1, and it is formed by the following blocks:

- *Control unit*, based-on a personal computer (i7-8750H CPU, 16GB RAM, 1 TB) that is connected to the SDR platform for control tasks by a serial interface. Also, different algorithms for AIS and VDES services are executed from MATLAB.
- *Radio unit*, composed by an SDR platform (ADALM-Pluto by Analog Devices) that converts the digital instructions launched from the control unit in radio signals, received or transmitted in function of the selected mode.
- *RF front-end unit*, where an own designed PCB has been implemented to include a SAW filter (TA0935A) and two low-noise amplifiers (TQP3M9009) connected in cascade topology. This additional hardware is used to remove potential interferences and adjust the gain of both received and transmitted signals.

3 Methodology

As the proposed prototype is intended to transmit and receive messages on the different services like ASM, AIS and VDE-TER, several scripts were programmed. A global flow chart of each process is explained, differentiating the transmission and reception modes which has been implemented on the SDR platform.

3.1 Transmission mode

When a service is selected, the first step is to set up the transmission parameters, such as the symbol rate (9600 for the AIS, 19200 for the ASM), the number of samples per symbol (24 as the optimal value) and the sample rate, which is the product between the symbol rate and samples per

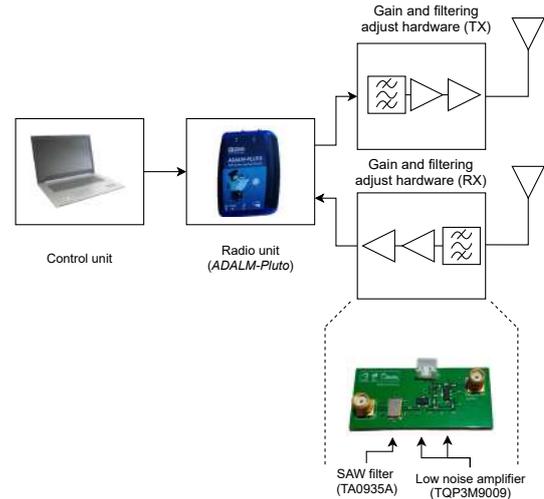


Figure 1. Block diagram of the prototype's architecture

symbol. Then, the SDR platform is configured as a transmitter, setting the desired channel frequency, samples per frame and sample rate. The binary packet is generated, adding the training sequence, payload and CRC. Lastly, an appropriate modulation scheme is applied on the binary packet, that will change in function of the selected service. When the packet is modulated, the SDR platform starts the transmission. The flowchart of the transmission mode is shown in figure 2.

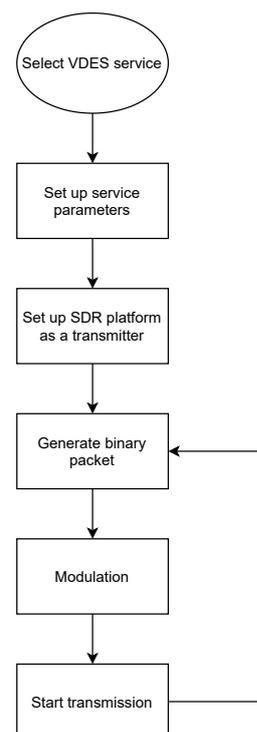


Figure 2. Proposed transmission flow chart

3.2 Reception flow chart

After a service is selected, several reception parameters are set, as the symbol rate (9600 for the AIS, 19200 for the ASM), samples per symbol (24 as the optimal value) and sample rate, conformed by the product between the symbol rate and samples per symbol. When the parameters and the SDR platform are set, the reception is started. Once a transmission is detected, the higher energy window within the received signal is considered as a packet. After storing the packet, DC offset is removed and I/Q imbalance compensation is performed. Later on, a demodulation is applied to retrieve the original signal, depending on the selected service. To retrieve the data within the frame, a synchronization process conformed by the detection and correlation of the training sequence is applied. The message's checksum is computed to check if the packet are valid or not. On the one hand, if the packet is invalid, it is discarded and the reception process will start again. On the other hand, if the packet is valid, payload is retrieved and decoded according to NMEA0183 protocol. The flowchart of the reception mode is shown in figure 3.

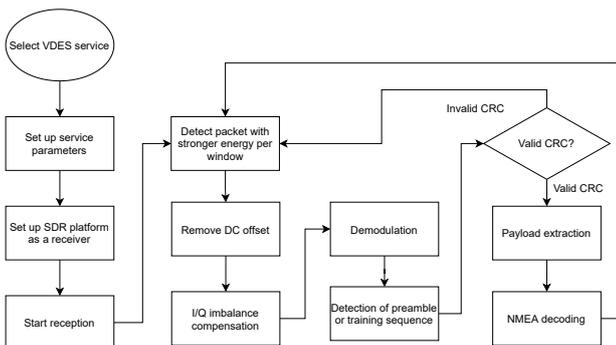


Figure 3. Proposed reception flow chart

4 Results and discussion

To verify the prototype's design, firstly the RF front-end has been tested. For reception mode, the filtering and amplification capacity of our design have been tested connecting the PCB input to a VHF antenna (located at a rooftop) and the output to an spectrum analyser (Advantest R3131A). In figure 4(a), the spectrum of the FM commercial band is attenuated from -21.4 dBm to -31.3 dBm, providing an attenuation of 9.9 dB. An AIS signal received is amplified from -61.3 dBm to -20.5 dBm, obtaining a gain of 40.8 dB. These results show an improvement in the reception capacity of the SDR platform, which also allows to avoid possible close out-band interferences. For transmission mode, an AIS signal at 162.025 MHz is generated by the SDR platform, obtaining a power output of 32.8 dBm, very close to 33 dBm that the AIS class B transceivers achieve. The transmitted signal obtained is shown in figure 4(b).

In figure 5 is shown a time domain representation of the AIS signal received by the SDR platform. It can be identified

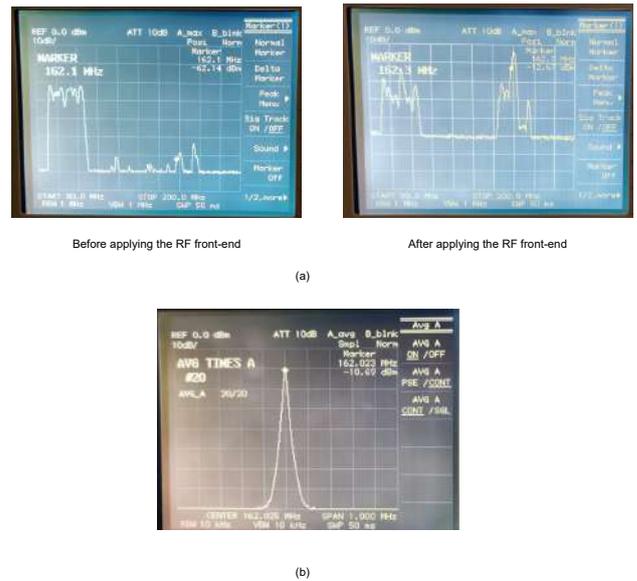


Figure 4. Spectrum without (left) and with (b) filtering and amplification hardware

the ramp-up period and the start and stop bytes, besides the payload.

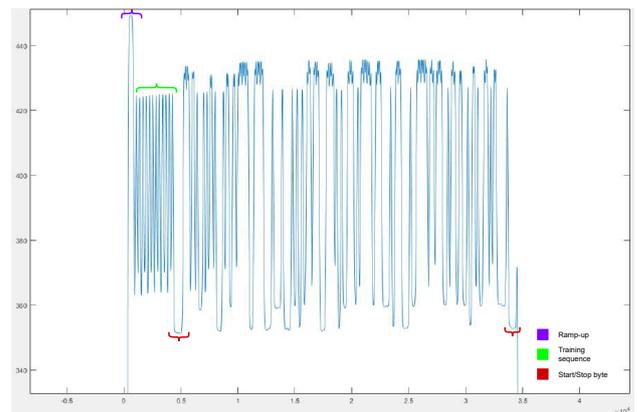


Figure 5. Analysis of an AIS signal received in time domain

5 Conclusions

In this paper, a prototype to transmit and receive AIS and VDES signal with a low-cost SDR platform has been implemented. The algorithms for transmission and reception modes are described. Additionally, a filtering and amplification hardware unit has been implemented to improve the prototype performance, showing the attenuation of the out-band interferences and also, the amplification of the AIS and VDES signals generated and received by the SDR platform. Finally, the AIS and VDES services implementation in a low-cost SDR platform has been verified, showing a potential alternative for the next generation of maritime communications deployment.

6 Acknowledgements

The authors acknowledge the work carried out by Juan Domingo Santana Urbín for his enormous contribution during this research. This work was supported by the Spanish Government under Grant (PID2020-116569RB-C32) Project. In addition, the contribution of Nicolás Molina-Padrón has been supported by the predoctoral fellowship by the Agencia Canaria de Investigación, Innovación y Sociedad de la Información, through the “Canarias avanza con Europa” Program, co-funded by the European Social Fund.

References

- [1] F. Wim van der Heijden, "AIS in a Historic Perspective: A History of the Identification of Ships". IALA, 2020.
- [2] M. Ilcev, "New Aspects for Modernization Global Maritime Distress and Safety System (GMDSS)," 2020.
- [3] International Telecommunication Union, "Technical characteristics for an automatic identification system using time division multiple access in the VHF maritime mobile band (Recommendation ITU-R M.1371-4)," *Int. J. Rock Mech. Min. Sci.*, vol. 42, no. 4, pp. 481–507, 2005, [Online]. Available: https://www.itu.int/dms_pubrec/itu-r/rec/m/REC-M.1371-5-201402-I!!PDF-E.pdf.
- [4] A. Harati Mokhtari, A. Wall, P. Brooks, and J. Wang, "Automatic Identification System (AIS): Data Reliability and Human Error Implications," *J. Navig.*, vol. 60, pp. 373–389, 2007, doi: 10.1017/S0373463307004298.
- [5] A. Felski, K. Jaskolski, and P. Banyś, "Comprehensive Assessment of Automatic Identification System (AIS) Data Application to Anti-collision Manoeuvring," *J. Navig.*, vol. 68, 2015, doi: 10.1017/S0373463314000897.
- [6] T. Gaugel, J. Mittag, H. Hartenstein, S. Papanastasiou, and E. G. Ström, "In-depth analysis and evaluation of Self-organizing TDMA," in 2013 IEEE Vehicular Networking Conference, 2013, pp. 79–86, doi: 10.1109/VNC.2013.6737593.
- [7] S. Fossen and T. I. Fossen, "Extended Kalman Filter Design and Motion Prediction of Ships Using Live Automatic Identification System (AIS) Data," in 2018 2nd European Conference on Electrical Engineering and Computer Science (EECS), 2018, pp. 464–470, doi: 10.1109/EECS.2018.00092.
- [8] ITU-R, "Technical characteristics for a VHF data exchange system in the VHF maritime mobile band," *Recomm. ITU-R*, vol. M.[VDES], no. January, pp. 1–54, 2014.
- [9] T. Eriksen, L. E. Bråten, H. C. Haugli, and F. A. S. Storesund, "VDE-SAT-A new maritime communications system Address." 2016.
- [10] N. M. Padrón, F. Cabrera-Almeida, V. Araña, and M. Tichavska, "An Overview About the Physical Layer of the VHF Data Exchange System (VDES)," *Lecture Notes in Computer Science*, vol. 12013 LNCS. 2020, doi: 10.1007/978-3-030-45093-9_9.
- [11] F. Lázaro, R. Raulefs, W. Wang, F. Clazzer, and S. Plass, "VHF Data Exchange System (VDES): an enabling technology for maritime communications," *CEAS Sp. J.*, vol. 11, no. 1, pp. 55–63, 2019, doi: 10.1007/s12567-018-0214-8.
- [12] M. M. Marques, D. Teles, V. Lobo, and G. Capela, "Low-cost AIS Transponder using an SDR device," in *OCEANS 2019 MTS/IEEE SEATTLE*, 2019, pp. 1–4, doi: 10.23919/OCEANS40490.2019.8962863.

