

**Escola Universitària d'Enginyeria Tècnica
de Telecomunicació La Salle**

Treball Final de Màster

Màster Universitari en Enginyeria de Telecomunicació

Estudio de modulaciones digitales de baja potencia y
anchos de banda variable en la Antártida

Alumne
Josep M. Masó Llinàs

Professor Ponent
Joaquim Porté Jiménez

ACTA DE L'EXAMEN DEL TREBALL FI DE MÀSTER

Reunit el Tribunal qualificador en el dia de la data, l'alumne

D. Josep M. Masó Llinàs

va exposar el seu Treball de Fi de Màster, el qual va tractar sobre el tema següent:

Estudio de modulaciones digitales de baja potencia y anchos de banda variable en la Antártida

Acabada l'exposició i contestades per part de l'alumne les objeccions formulades pels Srs. membres del tribunal, aquest valorà l'esmentat Treball amb la qualificació de

Barcelona,

VOCAL DEL TRIBUNAL

VOCAL DEL TRIBUNAL

PRESIDENT DEL TRIBUNAL

Abstract

Current HF transceivers make use of high transmission powers to establish communications between them. In this final master's project, different modulations are implemented, transmitted, received and studied (under different transmission power conditions and bandwidths) in order to explore their capabilities inside and outside the STANAG and MIL-STD standards. This study aims to know the limits of each modulation in NVIS transmissions of low powers and different bandwidths. It is intended to find out the most optimal configuration in remote sensor data transmission applications, which will require low consumption. Through this study it will be concluded that modulation, bandwidth and transmission power offer us the best relationship between bitrate and consumption. For this, digital modulations will be studied through the results obtained: QAM, PSK and FSK. The different modulations will be transmitted with powers between 1 W to 24 W. On the other hand, they will be analyzed with different bandwidths between 2.3 KHz to 25 KHz. Modulation orders will vary between 2 to 32. In order to ensure a unique analysis of the modulations without the presence of high noise and interference, the study will be conducted in Livingston Island, Antarctica, through the prototype developed in the ENVISERA project.

Key words: HF, NVIS, Low-power transmission, Bandwidth, Bitrate, QAM, PSK, FSK, STANAG, MIL-STD

Resumen

Los transceptores HF actuales hacen uso de altas potencias de transmisión para establecer comunicaciones entre ellos. En este trabajo final de máster se implementan, se transmiten, se reciben y se estudian diferentes modulaciones (en distintas condiciones de potencia de transmisión y anchos de banda) con el fin de explorar sus capacidades dentro y fuera de los estándares STANAG y MIL-STD. Este estudio tiene como fin conocer los límites de cada modulación en transmisiones NVIS de bajas potencias y diferentes anchos de banda. Se pretende averiguar la configuración más óptima en aplicaciones de transmisión de datos de sensores remotos, los cuales requerirán un bajo consumo. A través de este estudio se concluirá que modulación, ancho de banda y potencia de transmisión nos ofrece la mejor relación entre bitrate y consumo. Para ello se estudiarán a través de los resultados obtenidos las modulaciones digitales: QAM, PSK y FSK. Las distintas modulaciones serán transmitidas con potencias entre 1 W y 24 W. Por otra parte, se analizarán con distintos anchos de banda entre 2.3 KHz y 25 KHz. Los órdenes de modulación variaran entre 2 y 32. Con tal de asegurar un análisis único de las modulaciones sin presencia de ruido alto e interferencias, el estudio se realizará en Livingston Island, Antártida, a través del prototipo desarrollado en el proyecto ENVISERA.

Palabras clave: HF, NVIS, Low-power transmission, Bandwidth, Bitrate, QAM, PSK, FSK, STANAG, MIL-STD

Resum

Els transceptors HF actuals fan ús d'altres potències de transmissió per establir comunicacions entre ells. En aquest treball final de màster s'implementen, es transmeten, es reben i s'estudien diferents modulacions (en diferents condicions de potència de transmissió i amplitud de banda) amb la fi d'explorar les seves capacitats dintre i fora dels estàndards STANAG i MIL-STD. Aquest estudi té com a fi conèixer els límits de cada modulació en transmissions NVIS de baixes potències i diferents amplituds de banda. Es pretén esbrinar la configuració més òptima en aplicacions de transmissió de dades de sensors remots, els quals requeriran un baix consum. A través d'aquest estudi es conclourà quina modulació, amplitud de banda i potència de transmissió ens ofereix la millor relació entre bit rate i consum. Per això s'estudiaran a través dels resultats obtinguts les modulacions: QAM, PSK i FSK. Les diferents modulacions seran transmeses amb potències entre 1 W i 24 W. Per una altra banda, s'analitzaran amb diferents amplituds de banda entre 2.3 KHz i 25 KHz. Els ordres de modulació variaran entre 2 i 32. Per tal d'assegurar un anàlisi únic de les modulacions sense presència de soroll alt e interferències, l'estudi es realitzarà a Livingston Island, Antàrtida, a través del prototip desenvolupat en el projecte ENVISERA.

Paraules clau: HF, NVIS, Low-power transmission, Bandwidth, Bitrate, QAM, PSK, FSK, STANAG, MIL-STD

Quod est inferius, est sicut quod est superius.

(Chrysogonus Polydorus, Nuremberg, 1541)

Contenido

1	Introducción	1
1.1	Introducción del proyecto	1
1.2	Motivación del proyecto	2
1.3	Estructura del documento.....	3
2	Fundamentos teóricos	5
2.1	Comunicaciones NVIS.....	5
2.2	Variabilidad del canal ionosférico	6
2.3	Efectos del canal ionosférico.....	8
2.3.1	Dispersión temporal.....	8
2.3.2	Variación temporal.....	10
3	Implementación	13
3.1	Plataforma NIGREDO.....	13
3.1.1	Hardware.....	13
3.1.2	Software	15
3.2	Formato de la trama.....	16
3.2.1	Tono 600 Hz.....	17
3.2.2	Secuencia PN de orden 6.....	19
3.2.3	Modulaciones	20
3.2.4	Test realizados.....	21
3.3	Implementación Matlab.....	24
3.3.1	Cambios realizados.....	24
3.3.2	GenPNbyPolynomial.m	25
3.3.3	GenPNSequence.m.....	26
3.3.4	CreateIQData.m.....	26
3.3.5	Demodtrama.m	32
3.3.6	MassiveDemodTrama.m	40
3.3.7	MassiveGrafiques.m.....	42
4	Resultados	45
4.1	Introducción a los resultados	45
4.2	BER-EbNo.....	45
4.3	EbNo-Time.....	46

4.4	CDF según potencias transmitidas	47
4.5	CDF según anchos de banda transmitidos	50
5	Costes	53
5.1	Temporal	53
5.2	Económico.....	54
6	Conclusiones.....	55
7	Líneas de futuro	57
8	Referencias.....	59
9	Anexo.....	63

Tabla de contenidos

Figura 1. Propagación NVIS y Oblicua [20]	6
Figura 2. Variabilidad diaria de la capa ionosférica [23]	7
Figura 3. Ciclo solar del sol [24].....	7
Figura 4. Efecto multicamino de un canal [25]	8
Figura 5. Perfil de retraso de potencia [26]	9
Figura 6. Fading selectivo en frecuencia	10
Figura 7. Efecto doppler shift [27].....	11
Figura 8. Esquema general de la plataforma NIGREDO [29]	14
Figura 9. Plataforma NIGREDO en Glaciar Rocoso	15
Figura 10. Esquema general de la Red Pitaya [30]	15
Figura 11. Formato de la trama [29]	17
Figura 12. Efecto doppler en trama 600 Hz	19
Figura 13. Correlación de una trama.....	20
Figura 14. Diagrama IQ 2QAM y 2 PSK.....	21
Figura 15. Diagrama IQ 4QAM y 4PSK.....	21
Figura 16. Código genPNbyPolynomial.m	25
Figura 17. Código GenPNSequence.m.....	26
Figura 18. Configuración parámetros CreateIQData.m	27
Figura 19. Generación tono 600 Hz CreateIQData.m.....	28
Figura 20. Creación secuencia PN CreateIQData.m	29
Figura 21. Creación de bits y filtro CreateIQData.m	29
Figura 22. Modulación PSK CreateIQData.m.....	30
Figura 23. Modulación FSK CreateIQData.m.....	30
Figura 24. Modulación QAM CreateIQData.m	31
Figura 25. Escalado de potencia CreateIQData.m.....	31
Figura 26. Creación trama CreateIQData.m	32
Figura 27. Creación ficheros IQ CreateIQData	32
Figura 28. Parámetros Demodtrama.m	33
Figura 29. Cargado de ficheros IQ Demodtrama.m	34
Figura 30. Correlación secuencia PN Demodtrama.m	35
Figura 31. Sincronización, generación de bits y filtro Demodtrama.m.....	36
Figura 32. Cálculo y corrección del efecto doppler Demodtrama.m	36
Figura 33. Punto de control Demodtrama.m	37
Figura 34. Cálculo SNR y EbNo Demodtrama.m.....	37
Figura 35. Ecuación modulaciones Demodtrama.m	38
Figura 36. Cálculo BER PSK Demodtrama.m	38
Figura 37. Cálculo BER FSK Demodtrama.m.....	39
Figura 38. Cálculo BER QAM Demodtrama.m	39
Figura 39. Parámetros MassiveDemodTrama.m.....	40
Figura 40. Bucle general y ajuste de parámetros MassiveDemodTrama.m	41
Figura 41. Ejecución Demodtrama y guardado de datos MassiveDemodTrama.m.....	41

Figura 42. Clasificación datos para CDF MassiveGrafiques.m.....	42
Figura 43. Clasificación datos para BER-EbNo-Time MassiveGrafiques.m.....	42
Figura 44. Generación de gráficas MassiveGrafiques.m.....	43
Figura 45. Gráfica BER-EbNo orden 2 [29]	45
Figura 46. Gráfica BER-EbNo orden 4 [29]	46
Figura 47. Gráfica EbNo-Time orden 8 [29].....	47
Figura 48. Gráfica EbNo-Time orden 32 [29].....	47
Figura 49. Gráfica CDF 24 W orden 2 [29].....	48
Figura 50. Gráfica CDF 3W orden 2 [29].....	49
Figura 51. Gráfica CDF 24W orden 4 [29].....	49
Figura 52. Gráfica CDF 3W orden 4 [29].....	50
Figura 53. Gráfica CDF 5 KHz 24 W orden 4 [29].....	51
Figura 54. Gráfica CDF 10 KHz 3 W orden 4 [29].....	51
Figura 55. Diagrama de Gantt del trabajo final de máster.....	53
Tabla 1. Test de potencia realizados cada hora	23
Tabla 2. Test de ancho de banda realizados cada hora	23
Tabla 3. Costes económicos del estudio	54

Acrónimos

HF: *High Frequency*

NVIS: *Near Vertical Incidence Skywave*

QAM: *Quadrature Amplitude Modulation*

FSK: *Frequency Shift Keying*

PSK: *Phase Shift Keying*

FPGA: *Field-programmable gate array*

DSSS: *Direct Sequence Spread Spectrum*

OFDM: *Orthogonal Frequency-Division Multiple Access*

ISI: *Interferencia intersimbólica*

KSPS: *Kilosamples per Second*

DDC: *Digital Down Converter*

DUC: *Digital Up Converter*

CDF: *Cumulative Distribution Function*

BER: *Bit Error Rate*

EbNo: *Energy per bit to noise power spectral density ratio*

1 Introducción

1.1 Introducción del proyecto

La Universidad La Salle cuenta con numerosos proyectos de diferentes temáticas, entre los cuales dentro del ámbito de las telecomunicaciones destaca por excelencia el proyecto Antártico. El proyecto Antártico se mueve en torno a las comunicaciones HF (High Frequency) de 3 MHz a 30 MHz, concretamente en comunicaciones de rebote ionosférico. Los distintos proyectos que ha habido, han basado la investigación al desarrollo de transceptores de HF de bajo coste económico y al estudio de modulaciones y los efectos producidos por el canal ionosférico.

Los primeros proyectos antárticos se remontan entre el 2003 y el 2012. Durante ese periodo los proyectos no estaban basados en el desarrollo de una plataforma de comunicación en tiempo real sino en una plataforma que permitiera el postprocesado de todas las señales recibidas. La principal razón de este efecto era por el alto coste de la tecnología que pudiera permitir una comunicación en tiempo real. Durante ese periodo, los proyectos que desarrollaron plataformas de bajo coste, de las cuales podemos destacar la primera generación formada por una emisora HF Kenwood TS-870, a partir de la segunda generación se empezó a diseñar e implementar las propias plataformas como con la plataforma con el sobrenombre SODIO. La tercera generación con cuenta con el sobrenombre de POTASIO y finalmente la cuarta generación llamada IRIS. Los principales objetivos de los proyectos durante el periodo del 2003 al 2012 fue el estudio de los efectos del canal ionosférico en transmisiones HF oblicuas realizadas desde la Base Española Juan Carlos I (Livingston Island, Antártida) y l'Observatori de l'Ebre en Roquetes (Tarragona, España). La investigación llevada a cabo durante estos años culminaría con intensos estudios de los efectos de canal en comunicaciones HF oblicuas de muy larga distancia con resultados publicados en varias revistas de alto impacto [1]–[8] y la formulación de tesis doctoral [9]. Por otro lado durante el proyecto, además se hicieron estudios acerca la transmisión de datos con modulaciones de banda ancha como DSSS y OFDM [10]–[13]. El propósito final era la transmisión de datos de sensores pertenecientes a l'Observatori de l'Ebre para disponer de los datos durante todo el año en España dado que la Base Antártica Española Juan Carlos I se mantiene cerrada durante el invierno austral.

A partir del 2016 el nuevo proyecto Antártico se inició con un enfoque diferente al anterior y a través de la necesidad de los científicos de la base Juan Carlos I. Dado que la Antártida es un continente dedicado únicamente a la ciencia según el Tratado Antártico [14], la presencia de científicos es alta. En la base Juan Carlos I, los científicos hacen uso de sensores para tomar medidas diarias para series históricas como por ejemplo serían temperaturas, humedad, luminosidad, movimiento de glaciares, etc. La problemática reside en que los sensores que ayudan en los proyectos de los científicos, estos en algunas ocasiones se encuentra lejos de la base. Para obtener los datos recogidos a lo largo del año, los científicos han de ir hasta los sensores, lo cual dada la orografía de la zona y las condiciones climáticas puede ser una tarea que tome mucho tiempo. Esta problemática se vio como una oportunidad para el nuevo

proyecto, dado que hoy en día podemos encontrar en el mercado tecnológico plataformas con FPGAs y ARMs con interesantes características que permitan el desarrollo de una plataforma de comunicaciones HF en tiempo real a diferencia de unos años atrás que tenían un coste muy alto. Por lo tanto, el nuevo proyecto tiene como fin la comunicación de sensores remotos presentes en la isla de Livingston con la base Juan Carlos I para que los científicos puedan obtener los datos sin necesidad de movilizarse hasta la ubicación de los sensores. En este caso a diferencia del antiguo proyecto la comunicación ionosférica en vez de ser oblicua y de larga distancia, sería NVIS (Near Vertical Incidence Skywave) y de corta distancia (inferior a 250 km). Además, por otra parte, el proyecto también hará un profundo estudio del canal NVIS y de los efectos que pueda comportar en transmisiones de datos. Una vez analizado el canal se efectuará otro estudio acerca de modulaciones de banda estrecha para la comunicación de sensores remotos, con el fin de optimizar las comunicaciones que se efectuaran.

El estudio de modulaciones de banda estrecha tiene una alta importancia dado que uno de los puntos más críticos en su estudio es la potencia de transmisión. El prototipo desarrollado estará en ubicaciones donde no habrá red eléctrica para ser alimentado, por lo tanto, la gestión de la energía es de alta importancia ya que será alimentado a través de baterías. Este trabajo final de máster estará centrado en el estudio de la transmisión de datos con diferentes modulaciones de banda estrecha, potencias de transmisión, ordenes de modulación y anchos de banda fuera de los estándares STANAG [15] y MIL-STD [16], [17]. El estudio como tal se centrará en el consumo de las modulaciones en la plataforma y en la obtención de resultados que permitan conocer que modulación, orden, potencia y ancho de banda es el más óptimo para la transmisión y recepción de datos de sensores remotos.

1.2 Motivación del proyecto

Al empezar este proyecto hacía dos años que formaba parte del proyecto Antártico como integrante por lo que la motivación de este proyecto iba más allá que el Trabajo Final de Máster. Este proyecto constituía una de las fases del proyecto Antártico, que culminaría con un artículo científico para la revista Radio Science. Tras la finalización del estudio realizado en el Trabajo Final de Grado “Modulaciones para sensores remotos con tecnología NVIS” [18] el proyecto Antártico aún había de hacer otro estudio de modulaciones con más aspectos variables: más modulaciones, más ordenes de modulación y distintos anchos de banda en los que transmitir. A través de los conocimientos adquiridos y experiencia reunida en el anterior estudio, muchas de las herramientas, procesos, diseños e implementación utilizadas estaban poco optimizadas. Al tener en mente todas las mejoras que podría tener el nuevo estudio a nivel de diseño e implementación para poder hacer un estudio más perfeccionado me ofrecí para llevarlo a cabo. A partir de ese punto empecé con mucha motivación el estudio que daría lugar a este Trabajo Final de Máster titulado “Estudio de modulaciones digitales de baja potencia y anchos de banda variable en la Antártida”.

1.3 Estructura del documento

Este Trabajo Final de Máster principalmente pretende ofrecer al lector cuales son los cambios más sustanciales respecto al diseño e implementación llevados a cabo respecto el anterior estudio o Trabajo Final de Carrera. Para tener en cuenta los fundamentos básicos de la ionosfera y de las modulaciones utilizadas para la transmisión de datos, se recomienda la previa lectura del Trabajo Final de Carrera “Modulaciones para sensores remotos con tecnología NVIS”.

Este Trabajo Final de Máster está estructurado empezando por un breve resumen acerca de los fundamentos teóricos de todo el estudio que se ha llevado a cabo. En este apartado veremos las bases de la tecnología NVIS, que es la utilizada para este estudio. Posteriormente veremos cómo se comporta el canal ionosférico, cuáles son sus características y que problemas surgen en las señales al ser transmitidas y recibidas. Para finalizar la teoría del asunto se hará una breve explicación de las modulaciones utilizadas: QAM, PSK y FSK.

Tras finalizar una breve explicación de los fundamentos teóricos, se procederá a la argumentación de todo el hardware y software requerido para el estudio llevado a cabo en este Trabajo Final de Máster. Se resumirán las necesidades requeridas, la plataforma escogida junto con sus periféricos asociados para poder transmitir y recibir correctamente las señales. Después de definir el sistema se tratará acerca del diseño de las diferentes tramas a transmitir y el porqué de cada una de las partes que la forman. Por otro lado, se mostrarán todos los códigos Matlab utilizados: el código para la creación de las tramas, el código asociado a la demodulación de una trama, el código para la demodulación de varias tramas seguidas y el código para la generación de todas las gráficas de resultados.

Tras mostrar toda la implementación desarrollado para efectuar el estudio, en el siguiente apartado analizaremos todas las gráficas de mayor interés generadas a través de los resultados obtenidos a lo largo de todas las transmisiones y recepciones. Este apartado es el de mayor importancia dado que a través de los diferentes resultados obtenidos se realizarán diversas conclusiones acerca de todas las pruebas efectuadas. A partir de las conclusiones se dará un resultado como la modulación y orden de modulación, potencia de transmisión y ancho de banda más óptimo para la transmisión de datos de sensores remotos a través de la plataforma desarrollada por el proyecto.

A partir de este punto iremos finalizando los diferentes apartados del estudio tratando los costes que han sido necesarios para llevar a cabo el siguiente Trabajo Final de Máster. Los costes se dividirán en temporales, como el tiempo que ha requerido llevar a cabo todo este Trabajo Final de Máster; y en económicos, como todo el precio que ha costado todo lo necesario para este Trabajo Final de Máster.

Tras tratar los costes, el siguiente apartado tratará todas las conclusiones que se crean oportunas de mencionar tras todo el estudio llevado a cabo. Las conclusiones serán breves para ser lo más conciso posible.

Para ir acabando los diferentes apartados, se mencionarán las líneas de futuro que se llevarán a cabo después de este estudio, cuál será la continuación y posibilidades que ofrecerán los códigos Matlab implementados.

Como penúltimo apartado de este Trabajo Final de Máster se hará mención a todas las referencias que el lector haya encontrado en la lectura.

Para finalizar encontraremos el anexo donde encontraremos el artículo escrito para la publicación en Radio Science y todas las gráficas que no se hayan mostrado en el apartado de resultados por no aportar suficiente información.

2 Fundamentos teóricos

Primeramente, antes de dar explicación a todo el sistema utilizado, el diseño e implementación llevada a cabo se recordarán brevemente algunos de los fundamentos teóricos básicos del estudio. Es importante tener en cuenta que para una explicación más detallada de la teoría básica se puede encontrar en el Trabajo Final de Carrera previo a este estudio “Modulaciones para sensores remotos con tecnología NVIS”.

2.1 Comunicaciones NVIS

Las comunicaciones que basan el proyecto Antártico son aquellas que hacen uso de la ionosfera para ser reflejadas en ella. La ionosfera es una capa de la atmósfera presente en la termosfera. Las frecuencias que permiten este efecto se encuentran en la banda HF (High Frequency) que comprende entre 3 MHz y 30 MHz. Concretamente aquellas que se reflejan suelen ser entre 3 MHz y 15 MHz. Este tipo de comunicaciones son muy utilizadas en el ámbito de radioaficionados como en el sector militar como comunicación alternativa al satélite.

En comunicaciones HF por reflexión ionosférica podemos discernir entre dos métodos de transmisión: Oblicua y NVIS (Near Vertical Incidence Skywave). La primera, la oblicua se basa en la reflexión de las ondas con la capa ionosférica en un ángulo de incidencia menor de 70° . A través de este método se permiten alcanzar distancias mayores de 250 km [19], como fue en el antiguo proyecto antártico de un total de 12.000 km aproximadamente. En este caso para alcanzar tales distancias se hace a través de varios rebotes en los que la onda refleja con la superficie terrestre y posteriormente con la ionosfera varias veces. Por otro lado, el método NVIS, contrariamente, se basa en la reflexión de las ondas con la capa ionosférica en un ángulo mayor de 70° . A través de este método únicamente se puede alcanzar una distancia de 250 km. Este último método de reflexión con la ionósfera, NVIS, es el método en el que se basará todo el estudio dado que es en el que se trabaja en el proyecto Antártico actual. En la Figura 1 podemos ver la diferencia entre la propagación NVIS y la oblicua, en esta figura llamada DX.

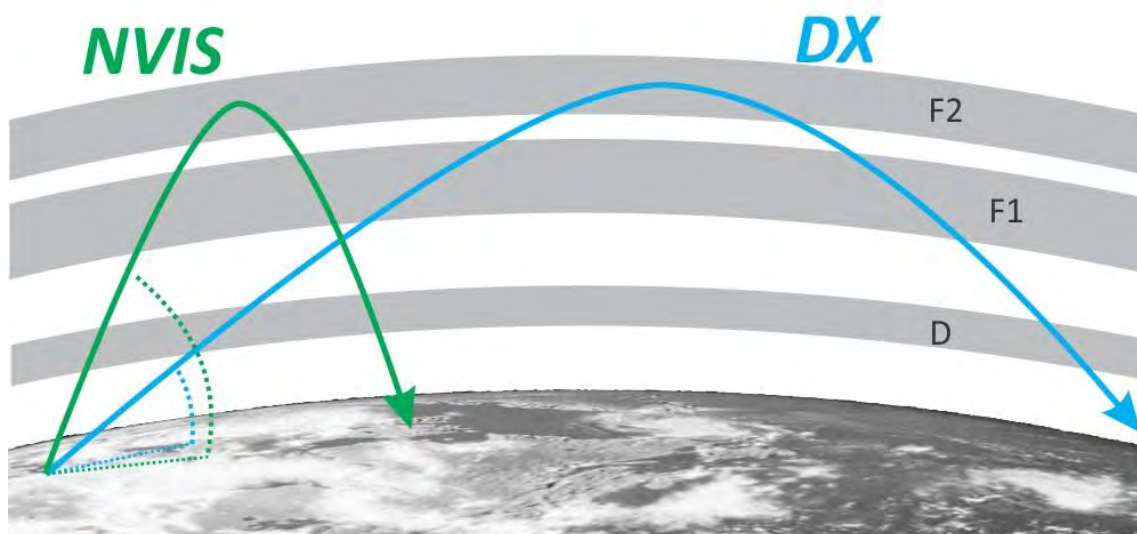


Figura 1. Propagación NVIS y Oblicua [20]

2.2 Variabilidad del canal ionosférico

La principal razón por la cual las ondas electromagnéticas reflejan con las capas de la ionósfera es debido a la radiación solar y en parte al campo magnético terrestre. Concretamente los iones presentes en la capa ionosférica son excitados a través de las radiaciones solares haciendo que su densidad sea mayor y a consecuencia al incidir una onda electromagnética de frecuencias de HF reflejarla. Según la densidad de iones ionizados presentes en la capa ionosférica está directamente relacionado con la frecuencia a la que las ondas ionosféricas se reflejan. A mayor ionización mayor es la frecuencia a la que se reflejan las ondas [21].

En la ionosfera podemos discernir varias capas [22] en las que interactuara nuestra señal:

- Capa D: Aparece únicamente durante el día debido a la ionización provocada por el viento solar. Es sumamente absorbente para frecuencias por debajo de los 10 Mhz
- Capa E: Se conoce también como la capa de Heaviside. Se forma a veces por la ionización del aire; algunos investigadores creen que podría ser por la fricción entre las diferentes capas de la atmósfera. Esta capa refleja las ondas de HF.
- Capa F: Se conocen también como capas de Appleton. Esta es la capa a través de la cual nuestra señal se reflejará para poder ser transmitida a largas distancias. Podemos distinguir la capa F1 y la capa F2. Durante el día están presentes las dos, pero por la noche la capa F1 desaparece, de tal manera que solo se puede comunicar a través de la capa F2.

La variación de las distintas capas de la ionosfera viene dada según la radiación solar, la cual varía diariamente, estacionalmente y anualmente. En la Figura 2 podemos ver la variabilidad de las capas que se da entre el día y la noche. Tal y como podemos ver, la capa F1 y la capa D desaparecen durante la noche.

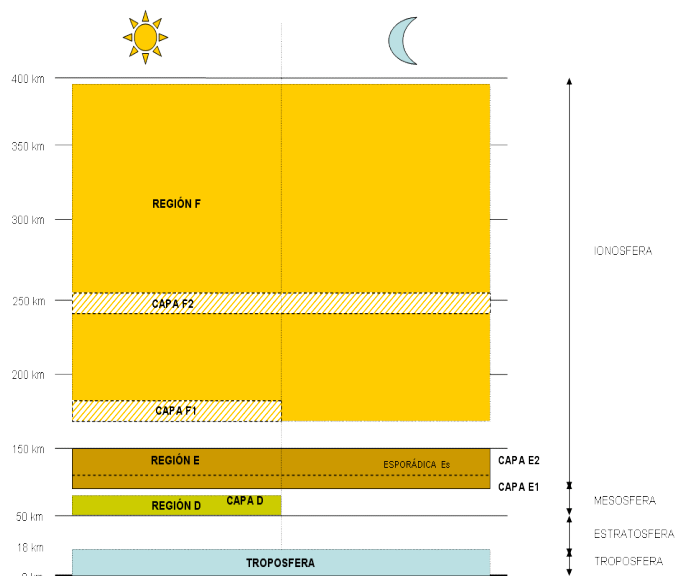


Figura 2. Variabilidad diaria de la capa ionosférica [23]

Por otra parte, la radiación solar incidente en la ionosfera no es constante a lo largo del año debido a la rotación de la Tierra alrededor del Sol. Durante el invierno, la radiación solar incide en la ionosfera de manera oblicua, siendo menor la influencia de ésta. A diferencia, en verano, los rayos solares inciden de manera perpendicular a la atmosfera obteniendo una ionización de esta mayor.

Por otra parte, podemos encontrar también los ciclos de actividad solar que tienen una duración de una década. Cada ciclo solar es único de tal manera que puede ser mayor o menor en cuestiones de radiación transmitida. Cada ciclo solar se compone de cinco años de gran actividad y otros cinco años de baja actividad. En la Figura 3 podemos observar una serie histórica de los diferentes ciclos solares que ha habido a lo largo del tiempo.

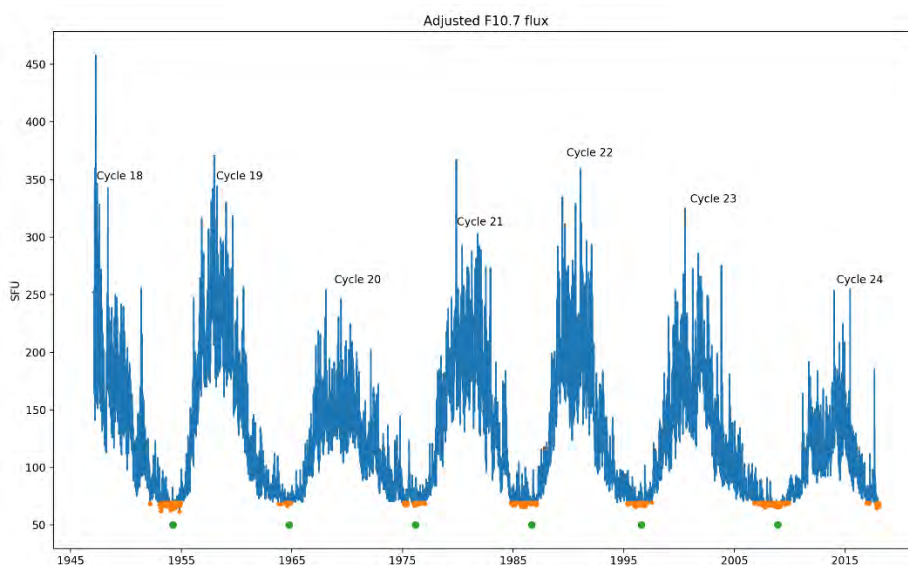


Figura 3. Ciclo solar del sol [24]

2.3 Efectos del canal ionosférico

El canal ionosférico es considerado un canal hostil debido a todas las características que lo forman en sus efectos en las señales que son transmitidas a través de este. En este apartado podremos ver los distintos problemas que podemos encontrarnos al recibir cualquier señal que haya sido reflejada en la ionosfera. Principalmente podemos encontrar dos tipos de efectos de variación del canal a corto plazo: la dispersión temporal y la variación temporal de las señales.

2.3.1 Dispersión temporal

La dispersión temporal es uno de los principales efectos que podemos encontrar en el canal ionosférico. Los principales parámetros que caracterizan la dispersión temporal son el ensanchamiento temporal y el ancho de banda de coherencia del canal.

2.3.1.1 Delay Spread o Ensanchamiento temporal

El ensanchamiento temporal es un efecto producido por el efecto multicamino del canal en las señales que se transmiten. Tal y como podemos observar en la Figura 4, podemos ver como se transmite una señal la cual es recibida por varios caminos de distinta distancia y por lo tanto se recibe la misma señal en distintos momentos de tal manera que la señal recibida queda afectada.

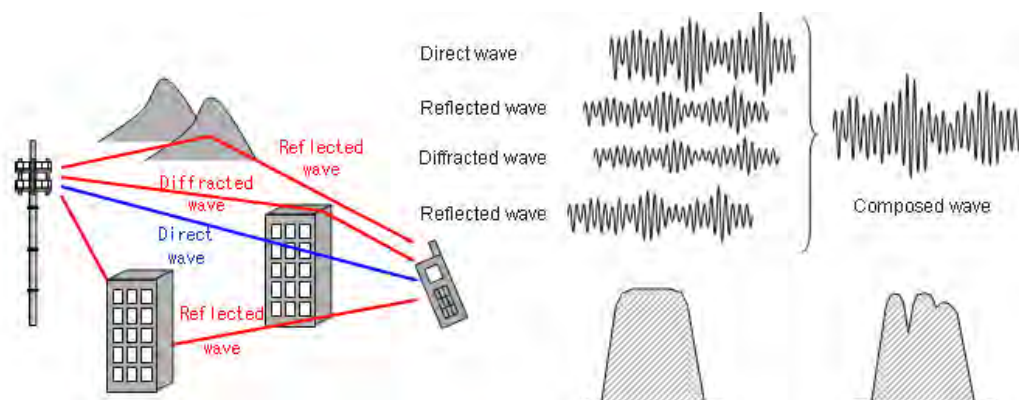


Figura 4. Efecto multicamino de un canal [25]

El efecto multicamino forma parte de la respuesta impulsional de un canal, este sería el perfil de retraso de potencia recibida en función del retraso de propagación: $S(\tau) = E_t[|h(t; \tau)|^2]$

Esta respuesta se puede estimar transmitiendo diferentes señales piloto como podrían ser: pulsos cuadrados de corta duración o señales de banda ancha como secuencias PN. En la Figura 5 podemos observar un ejemplo de la respuesta impulsional de un canal en el cual podemos

observar los diferentes caminos que puede haber y el retraso que tienen respecto la señal recibida. El parámetro de ensanchamiento temporal (T_m) es medido a través de la respuesta impulsional, la cual es la dispersión temporal máxima provocada por el efecto multicamino.

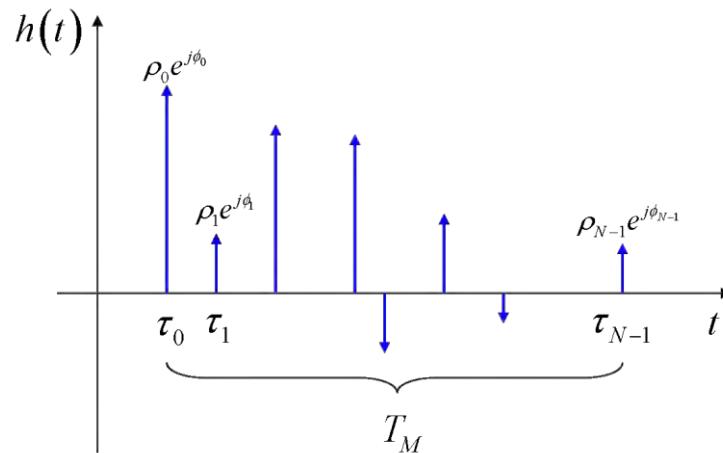


Figura 5. Perfil de retardo de potencia [26]

El tiempo de ensanchamiento temporal es un parámetro muy importante para poder decidir el tiempo de duración de los distintos símbolos que forman una trama. La importancia del tiempo del ensanchamiento temporal reside en que éste no debe ser nunca mayor que el tiempo de transmisión de un símbolo (T_s). En el caso que el tiempo de ensanchamiento temporal fuera mayor al tiempo de símbolo se daría un caso de ISI (Interferencia Intersimbólica) en el que según la respuesta impulsional del canal, el símbolo transmitido se recibiría varias veces con retraso y al mismo tiempo en el que se recibe el siguiente símbolo.

2.3.1.2 Ancho de banda de coherencia

La caracterización del efecto del ensanchamiento temporal también se puede expresar en el dominio frecuencial. En este caso el ancho de banda de coherencia (f_0) se definiría como el rango de frecuencias dentro de las cuales el canal tiene un comportamiento similar. Por normal general el ancho de banda de coherencia es inversamente proporcional al delay spread. A pesar de ello el valor del ensanchamiento temporal se puede calcular como el valor r.m.s. ($\sigma\tau$) del retraso de propagación $S(\tau)$ como función densidad, de tal manera que tiene en cuenta de una manera más exacta la distribución energética de la energía recibida. En este caso existe una constante de proporcionalidad inversa entre f_0 y $\sigma\tau$ que puede variar entre 3.62 y 50.

Desde el punto de vista frecuencial es importante tener en cuenta que el ancho de banda de nuestra señal ($W = 1/T_s$) no debe ser mayor que el ancho de banda de coherencia (f_0). En el caso que nuestra señal fuera mayor, se producirían fadings selectivos en frecuencia en nuestra señal. En la Figura 6 podemos observar como el ancho de banda de nuestra señal es mayor que el

ancho de banda de coherencia haciendo que en algunas frecuencias de nuestra señal haya pérdidas de potencia.

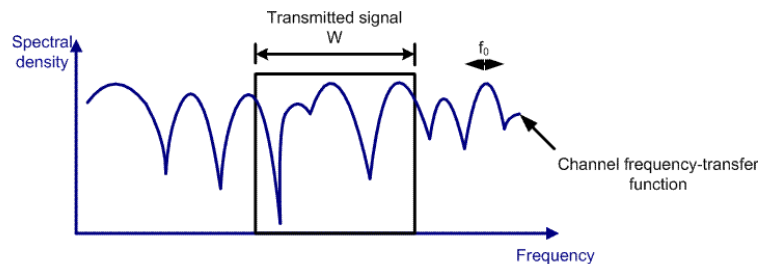


Figura 6. Fading selectivo en frecuencia

Por el contrario, en el caso que el ancho de banda de nuestra señal fuera menor que el ancho de banda de coherencia, todas las frecuencias de la señal sufrirían los mismos efectos del canal de manera simultánea.

2.3.2 Variación temporal

Tal y como hemos observado, el ensanchamiento temporal y el ancho de banda de coherencia, nos aportan información de la dispersión temporal del canal (efectos de ISI y selectividad en frecuencia), pero no describen el tipo de variabilidad en el dominio del tiempo. La variación temporal del canal viene descrita por dos funciones estadísticas las cuales permiten estimar parámetros decisivos para evaluar el tipo de variabilidad en vez de la velocidad de transmisión de los datos. La variabilidad temporal del canal viene dada por los parámetros de doppler shift y tiempo de coherencia.

2.3.2.1 Tiempo de coherencia

El tiempo de coherencia se define como el intervalo de tiempo en el que el canal tiene un comportamiento similar. Es importante que el tiempo de símbolo de nuestra señal (T_s) no sea mayor que el tiempo de coherencia del canal (T_0). En el caso que lo fuera, el canal variaría significativamente de un símbolo a otro, de tal manera que el sistema podría desincronizarse fácilmente de la trama que está recibiendo o incluso perder la divergencia de los sistemas adaptativos de ecualización. Por otro lado, en el caso que el tiempo de símbolo (T_s) sea menor que el tiempo de coherencia del canal (T_0), las variaciones del canal entre símbolo y símbolo serían suficientemente pequeñas para garantizar la sincronización del sistema en cada símbolo y de los ecualizadores. En este caso si existen pérdidas de SNR en la señal durante varios símbolos de tal manera que hace aumentar el BER, la manera de combatir este efecto es a través de interleaving y código corrector de errores.

2.3.2.2 Doppler Shift y Doppler Spread

Por otro lado, la variación temporal del canal se puede estudiar desde el punto de vista frecuencial. Los efectos doppler shift y doppler spread, son efectos de variabilidad temporal rápida del canal. El doppler spread (B_d) se caracteriza por la dispersión frecuencial que puede tener la frecuencia portadora de la señal. Este efecto puede deberse a variaciones en la absorción de las capas ionosféricas, al aumento de la distancia de transmisión debido al movimiento de las capas ionosféricas o debido a los cambios de polarización debido a la rotación de Faraday. El doppler spread es inversamente proporcional al tiempo de coherencia.

Por otra parte, el doppler shift (f_d) se debe al movimiento de las capas ionosféricas, el cuál varía la frecuencia portadora recibida. El parámetro de doppler shift (f_d), describe la máxima variabilidad frecuencial del canal, asociado directamente a la velocidad máxima de las capas ionosféricas. A diferencia, el doppler spread es la suma de todas las variabilidades que existen en un canal. El doppler shift se puede calcular como $f_d = v / \lambda$, donde v es la velocidad de movimiento de la capa ionosférica en la que se refleja y λ es la longitud de onda de la portadora. En la Figura 7 podemos ver un ejemplo del efecto doppler shift en el que la frecuencia portadora varía según se alejan o se acercan el transmisor y el receptor.

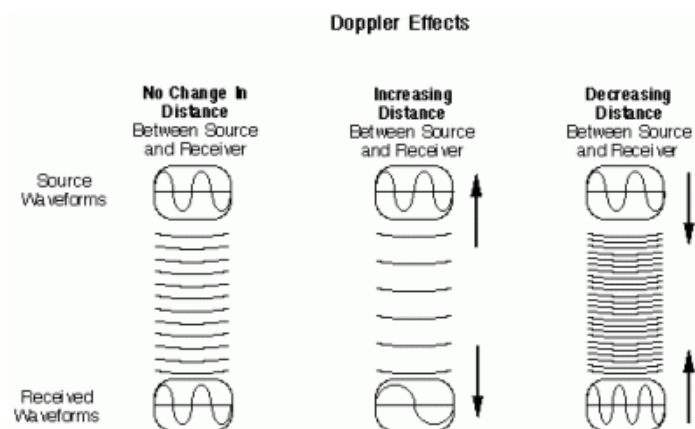


Figura 7. Efecto doppler shift [27]

El efecto del doppler shift y doppler spread deben ser corregidos en todo momento dado que este efecto puede producir fadings en nuestra señal portadora de tal manera que no se pueda demodular correctamente.

3 Implementación

Tras una breve explicación teórica de los distintos conceptos teóricos de los efectos que pueden producir el canal ionosférico a las señales que se transmiten, en este apartado se explicará detalladamente toda la implementación llevada a cabo. Se explicará brevemente la plataforma utilizada desde el punto de vista del hardware como desde el punto de vista del software, se explicará detalladamente el formato de la trama utilizado para tratar los efectos del canal ionosférico y se explicará el software implementado en Matlab para el uso de la plataforma y la realización del estudio.

3.1 Plataforma NIGREDO

La plataforma NIGREDO es fruto del trabajo de tres años del proyecto Antártico iniciado en 2016 y finalizado en 2019 con el objetivo de ser una plataforma de bajo coste económico. Esta plataforma es el conjunto de diferentes hardware conectados entre sí y operados por diferentes softwares como veremos brevemente a continuación. Todo el conjunto permite llevar a cabo diferentes funcionalidades según su configuración. Primeramente, la transmisión de tramas expresadas en ficheros con los valores IQ de estas y la recepción de las tramas en el caso que otra plataforma las haya transmitido. Para este primer caso esta funcionalidad permite el sondeo de canales y estudios de distintas señales, como es el caso de este estudio, el cual ha transmitido diferentes tramas con distintas modulaciones. Por otra parte, la plataforma NIGREDO puede configurarse para funcionar como un sistema de transmisión y recepción de paquetes de datos en tiempo real half dúplex. Este caso de transmisión y recepción en tiempo real es la característica de uno de los objetivos finales del proyecto Antártico 2016-2019 para la comunicación de sensores remotos con la Base Antártica Española Juan Carlos I.

3.1.1 Hardware

El hardware que compone la plataforma NIGREDO es un total de 9 componentes de mayor y menor importancia. Principalmente el corazón del sistema lo compone la plataforma Red Pitaya [28], esta plataforma está compuesta por una Zynq 7010 y un procesador DUAL CORE ARM CORTEX A9 de tal manera que se puede programar y se le puede cargar un sistema operativo, el cual en este caso es Linux. No obstante, una de las características que la hacen tener una gran relevancia son sus conversores ADC y DAC de 14 bits, la cual le permiten transmitir señales de HF.

Paralelamente a la plataforma Red Pitaya, existe un soporte realizado por la plataforma Raspberry Pi 3. Esta plataforma se compone por un microprocesador en el cual se ha cargado un sistema operativo Linux. A través de esta plataforma se gestiona el uso del resto de

periféricos y del control de la Red Pitaya. En el caso del tiempo real es la encargada de modular y demodular las tramas correctamente junto con la codificación y interleaving.

Finalmente, la plataforma NIGREDO cuenta con una serie de periféricos que complementan el funcionamiento de las dos plataformas anteriores. De estos periféricos podemos encontrar el amplificador de potencia de las señales que se transmiten con una potencia máxima de 24 W. Por otra parte, encontramos el amplificador de recepción con 30 dB de amplificación. Encontraremos también en la plataforma un filtro en recepción de 2 MHz a 7 MHz para atenuar las señales o interferencias. También forma parte el relé de antena, el cual nos permite conmutar la antena según nos conviene en recepción o transmisión. El relé de antena y el amplificador de potencia viene controlado por la Teensy, una plataforma diseñada por el Radioclub de La Salle. Está plataforma además lleva el control de ROE y temperatura del amplificador de potencia y lo gestiona según estos parámetros. En este caso la Teensy viene controlada por la Raspberry Pi 3. También en la plataforma NIGREDO podemos encontrar un módulo GPS Adafruit Breakout v3, el cual nos permite obtener la hora exacta para que la Raspberry Pi 3 pueda tener conocimiento de que hora debe dar orden a Red Pitaya de transmitir o recibir en las configuraciones de sondeo. Finalmente, uno de los elementos más importantes de los periféricos es la antena utilizada en el sistema acompañada de un balun de $\lambda/2$. La antena utilizada para este estudio, dado que se ha hecho uso de una sola frecuencia, 5 MHz, se ha utilizado una antena V invertida dada sus características.

En la Figura 8 podemos ver un esquema general de la plataforma NIGREDO. Por otra parte, en la Figura 9, podemos observar una imagen tomada de la plataforma en una de las ubicaciones de la Antártida, Glaciar Rocoso.

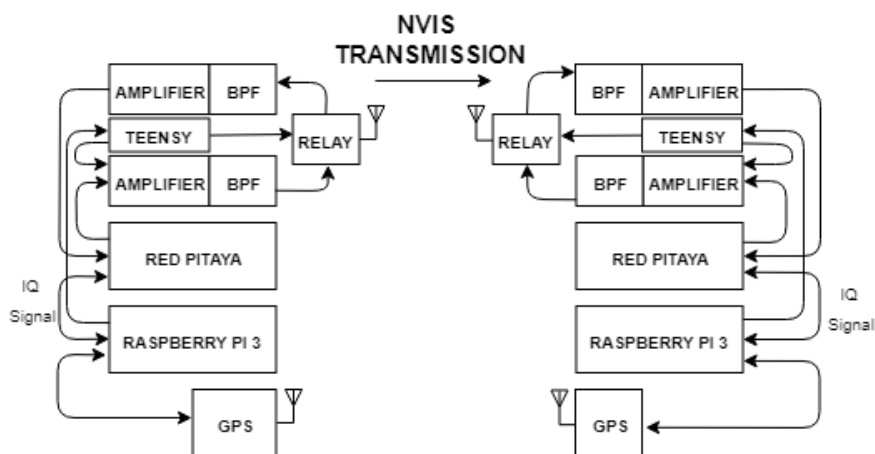


Figura 8. Esquema general de la plataforma NIGREDO [29]

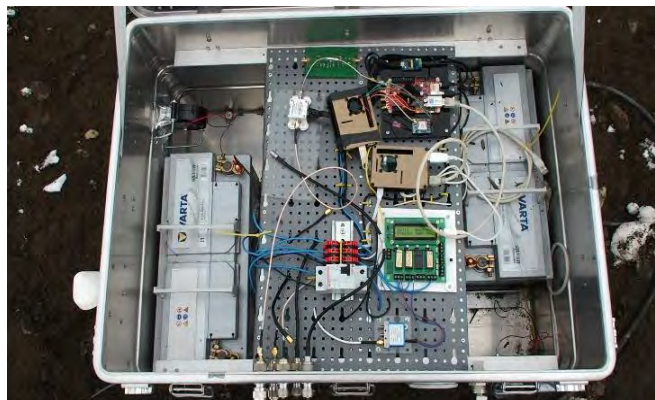


Figura 9. Plataforma NIGREDO en Glaciar Rocoso

3.1.2 Software

La plataforma NIGREDO está compuesta por un conjunto de tres softwares. Principalmente el software más complejo es el que forma parte de la Red Pitaya, el cuál esta explicado detalladamente en el Trabajo Final de Máster de Joaquim Porté [30]. Esté software contiene la comunicación entre la FPGA y el sistema operativo Linux de la Red Pitaya, y por otra parte implementa todo el proceso de DUC (Digital Up Converter) hasta el DAC y el DDC (Digital Down Converter) hasta el ADC. Por otra parte, todo el procesado de señal llevado a cabo a través de estas operaciones, es guardado en una trama IQ o es leído para ser transmitido en la Red Pitaya. En la Figura 10 podemos ver un esquema que resumiría brevemente el software de la plataforma Red Pitaya.

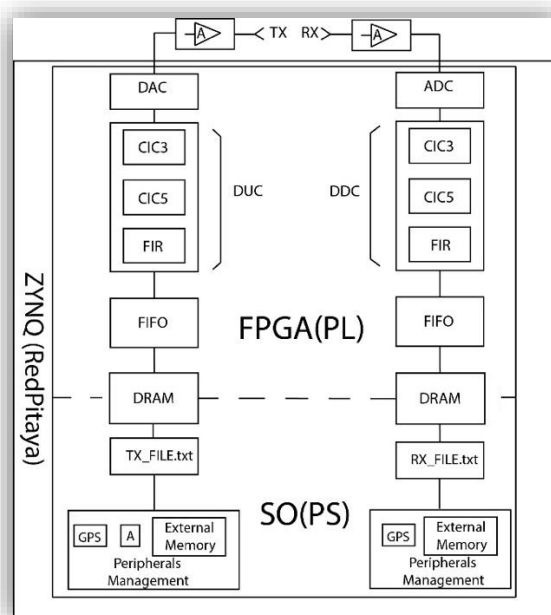


Figura 10. Esquema general de la Red Pitaya [30]

Por otra parte, encontramos el software correspondiente a la Raspberry Pi 3, este software permite la gestión del hardware Teensy de la plataforma, de tal manera que tiene en cuenta cuando activar el amplificador de potencia para transmitir. Por otra parte, lleva el control de la Red Pitaya para decidir cuándo debe transmitir o recibir datos. La conexión con esta plataforma es por vía ethernet, de tal manera que todos los datos son transferidos por este método. En el caso de la transmisión en tiempo real, la Raspberry Pi 3 se ocupa de recibir los bits que se deben transmitir o almacenar los bits recibidos, se ocupa de codificar y decodificar todos los bits junto con el interleaving o deinterleaving. Finalmente se ocupa de modular y demodular todos los bits. En el caso de que la plataforma NIGREDO esté configurado en modo de sondeo, la Raspberry Pi 3 se ocupa de leer los ficheros IQ y transmitirlos a Red Pitaya, o recibir las muestras IQ y preparar el fichero para ser guardado.

Por último, encontramos el software dedicado al control de la Teensy, la cual a través de la conexión con la Raspberry Pi 3 llevará el control directo del amplificador de potencia. A través de sensores de temperatura y de ROE podrá conocer en todo momento según los valores de estos, si debe apagar el amplificador de potencia para preservar la vida útil de este.

3.2 Formato de la trama

En este Trabajo Final de Máster el formato de la trama es uno de los puntos que han cambiado con importancia respecto al último estudio realizado en el Trabajo Final de Grado. El formato que siguen las tramas que se transmitirán son de suma importancia para que el postprocesado sea posible. Además, en este caso siendo una comparación de distintos casos de transmisión, cada paquete de datos debe ser comparado bajo las mismas condiciones que el resto. Para que todos estos aspectos sean posibles se ha diseñado la trama a enviar con sumo cuidado para evitar cualquier problema que no se pueda resolver en postprocesado.

En la Figura 11 podemos ver el esquema general que se ha utilizado para la transmisión de las tramas. Tal y como podemos ver, cada paquete de trama lo forman 3 paquetes de modulación. Un paquete de modulación está formado por un tono de 600 Hz, una secuencia PN de orden 6 y una modulación que puede ser una PSK, FSK o una QAM. Los paquetes de trama están formados por tres paquetes de modulación que incluirá: tres tonos de 600Hz, tres secuencias PN de orden 6 y tres modulaciones distintas que siempre tendrán el mismo orden y mismo ancho de banda. Todas las tramas están formadas por 50 paquetes de trama, que serán 150 paquetes de modulación.

Por otra parte, hay que destacar que la Figura 11 es el esquema general de las tramas que se han utilizado para la transmisión durante la primera parte. Podemos ver que los bloques de modulaciones contienen 250 símbolos, con un remuestreo de 42 símbolos, lo que nos daría un ancho de banda de 2.3 KHz. Hay que tener en cuenta que, en la segunda parte de tramas transmitidas, el esquema general de tramas es el mismo exceptuando el número de símbolos en los bloques de datos. En este último caso el remuestreo dependerá del ancho de banda con el que se quiera transmitir y el tiempo que durará el paquete de datos siempre será el mismo.

En los siguientes apartados analizaremos detalladamente cada una de las partes de la trama, se explicará el porqué de su presencia y cuál es su función en la transmisión y recepción.

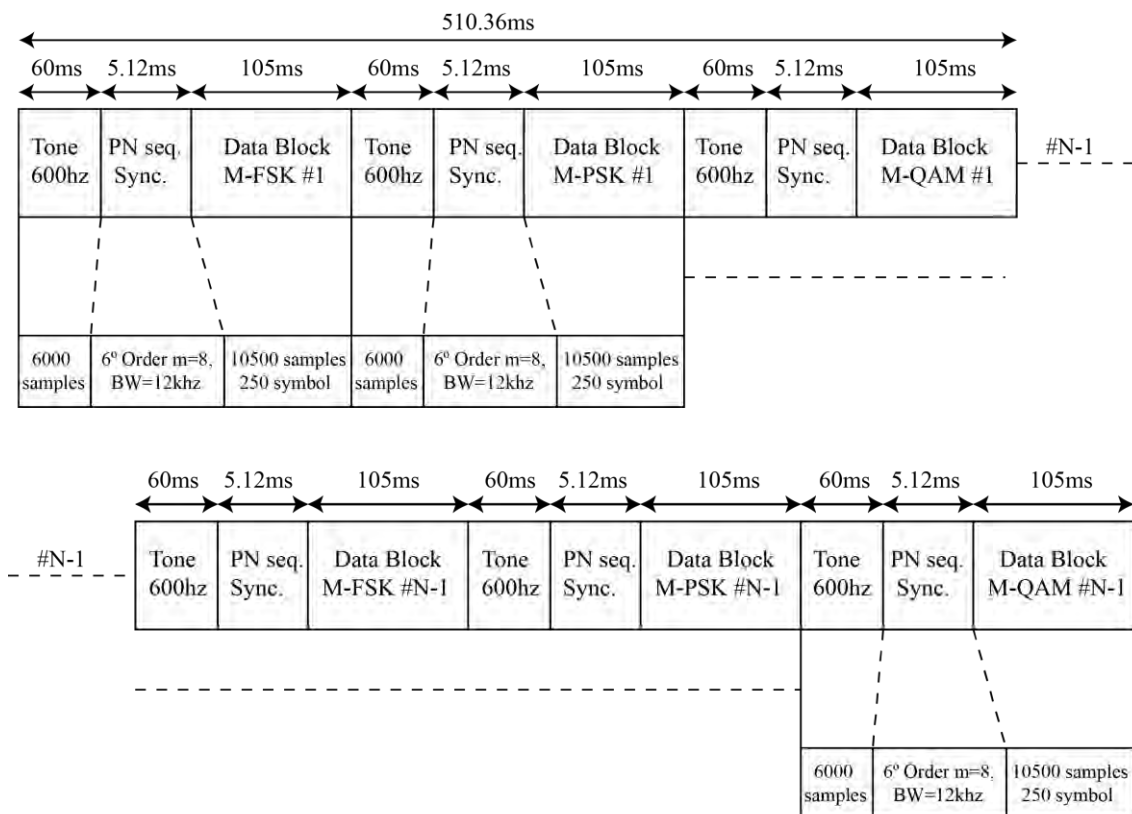


Figura 11. Formato de la trama [29]

3.2.1 Tono 600 Hz

Primeramente, todos los paquetes de modulación están compuestos por un tono de 600 Hz que ocupa un total de 6000 muestras. La función principal de este bloque es detectar y medir el efecto doppler originado por el canal y la plataforma.

A diferencia del estudio anterior, este bloque, en esta ocasión lo forma un tono de 600 Hz en vez de una señal continua. La principal razón por la que se ha decidido transmitir un tono de 600 Hz es para ser más óptimo. En el estudio anterior existía un tono constante inicial de 150.000 muestras, un bloque bastante largo de duración 1.5 segundos. Además, cada paquete de modulación iba precedido de otro tono constante de 50.000 muestras con una duración de 500 ms. Tal y como podemos ver en este estudio, el tono contiene 6.000 muestras y una duración de 60 ms, mucho más corto que los anteriores. La clave de esta optimización reside en el cambio de un tono constante a un tono de 600 Hz.

Para entender este cambio, debemos basarnos primeramente en el efecto doppler. El efecto doppler que queremos caracterizar es del orden de 50 Hz. En nuestro caso el efecto doppler es tan alto debido a los relojes internos de la plataforma Red Pitaya, los cuales no son muy exactos y frecuentemente este efecto viene dado como un efecto doppler en nuestra señal. En el caso que los relojes internos de la plataforma Red Pitaya funcionarán con total precisión, seguiríamos enfrentándonos al efecto doppler, pero en este caso de menor magnitud siendo entre 1 Hz y 9 Hz.

Teniendo en cuenta que podemos enfrentarnos a un efecto doppler de 50 Hz como máximo, para poder medir un doppler de 1 Hz, será necesario como mínimo un tono constante de 1 segundo, en el peor de los casos de 500 ms para medir medio periodo. Aun así, teniendo en cuenta el canal ionosférico y la inexactitud de los relojes, en general el efecto doppler no suele ser menor de 5 Hz, por lo que un tono constante de 500 ms nos permite medir dos periodos y medio. A diferencia del tono de 600 Hz que se utiliza en la trama del nuevo estudio, en el peor de los casos, si afecta a nuestra señal un efecto doppler de 50 Hz, el tono de 600 Hz modificaría su frecuencia a 550 Hz o 650 Hz. En el caso de 550 Hz si el bloque tiene una duración de 60 ms podremos medir el efecto doppler con un total de 33 periodos de manera que el cálculo de la frecuencia será más exacto que en el caso anterior.

Una vez calculada la frecuencia que está afectando a la trama, a partir de esta se podrá corregir el efecto doppler de paquete de modulación que lo forma. Hay que tener en cuenta que el paquete de modulación y el paquete de trama estarán dentro de los intervalos de tiempo de coherencia por lo que en todo un paquete de trama se verá afectado de igual manera por el canal ionosférico. Para corregir el efecto doppler se aplicará la siguiente fórmula a la señal:

$$datacorrect = datadoppler * (2 * j * pi * freqdoppler)$$

Si no corregimos el efecto doppler correctamente no seremos capaces en las siguientes operaciones detectar correctamente la secuencia PN del paquete de modulación y por lo tanto no demodular correctamente.

En la Figura 12 podemos ver en la derecha una trama de 600 Hz al ser transmitida y podemos ver en la izquierda el cambio de frecuencia debido al efecto doppler.

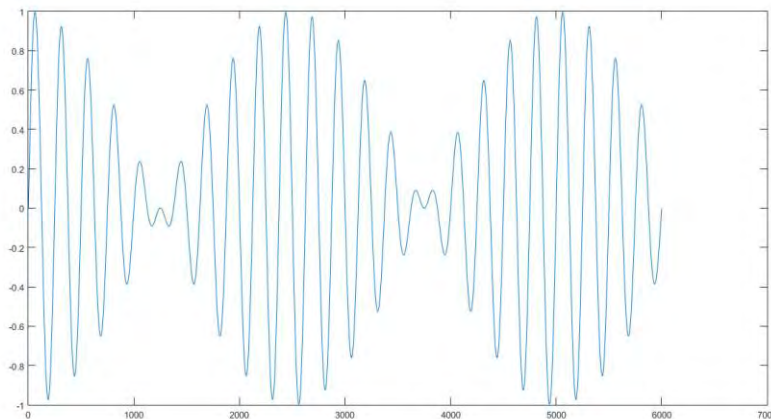


Figura 12. Efecto doppler en trama 600 Hz

3.2.2 Secuencia PN de orden 6

Las secuencias PN son una parte muy importante del paquete de modulación, ya que estas son las que nos permite sincronizarnos con el primer símbolo del bloque de modulación. En este nuevo estudio, la secuencia PN es de orden 6 con 64 chips, con un remuestreo de 8 conteniendo un total de 512 muestras y una duración de 5.12 ms. En el estudio anterior la secuencia PN inicial para sincronizar el inicio de la trama era de orden 10 con 1024 chips, con un remuestreo de 42 conteniendo un total de 43.008 muestras. Dado que estaba formado por tres secuencias PN juntas el contenido total es de 129.008 muestras, con una duración de 1.29 segundos. Por otra parte, antes de cada modulación había otra secuencia PN de orden 8 con 256 chips, con un remuestreo de 42 conteniendo un total de 10.752 muestras y una duración de 100 ms.

En el nuevo estudio se ha decidido implantar la misma secuencia PN que la que se ha utilizado para transmisiones y recepciones en tiempo real con el prototipo desarrollado. Una de las principales razones es para poder utilizar el estudio con más eficacia simulando el uso en tiempo real. La plataforma Red Pitaya para poder correlar la secuencia PN sin tener problemas de consumo y eficiencia, está principalmente no debe exceder un total de 500 muestras en total. Para la secuencia PN de orden 6 y remuestreo de 8 podemos apreciar que el

Por otra parte, se ha decidido disminuir el orden de la PN ya que no es necesario un orden tan alto. Si tenemos en cuenta la secuencia PN de orden 8 con un remuestreo de 42, podemos calcular que la correlación será posible con un SNR de -26 dB. En nuestro caso, aunque fuéramos capaces de sincronizar con los datos modulados con un SNR de -26 dB, no podríamos demodular. Para ello con una secuencia PN de orden 6 y un remuestreo de 8 la correlación será posible con SNR de -16 dB. En este caso podemos ver que la diferencia de duración de la trama es muy grande. La secuencia PN de orden 6 utilizada está formada por el polinomio [6 5 0].

A través de esta secuencia PN, tras haber corregido el efecto doppler, correlaremos la secuencia PN con la trama recibida y a consecuencia se mostrarán todos los picos de correlación

correspondientes al número de muestra donde se encuentra la primera muestra del bloque de modulación PSK, FSK o QAM. En Figura 13 podemos ver un ejemplo de la correlación de una trama.

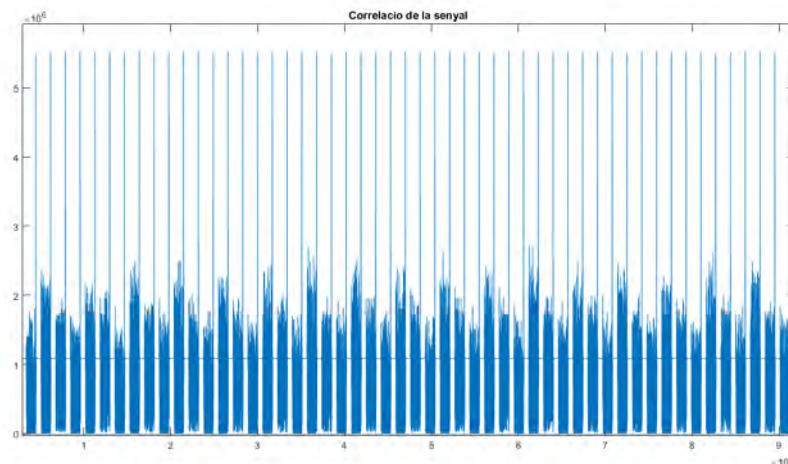


Figura 13. Correlación de una trama

3.2.3 Modulaciones

Las modulaciones utilizadas son el objeto principal de todo el estudio realizado. De todas las modulaciones posibles se ha hecho uso de modulaciones de espectro discreto que formen parte de los estándares SATANAG y MIL-STD. Aun así, parte del estudio está dedicado a salirse de los anchos de banda especificados por estos estándares.

En cuanto a las modulaciones utilizadas, nos encontramos con la PSK, FSK y QAM, siendo la QAM la utilizada dentro de los estándares. Lo que respecta a los órdenes de modulación se han hecho tests de modulaciones de orden 2 hasta orden 32. Hay que tener en cuenta que el aumento del orden de modulación hace disminuir el E_b/N_0 de la señal. Este parámetro es de más importancia que el SNR, ya que en este caso nos indica la energía de bit respecto al ruido.

Por otra parte, hay que tener en cuenta como han sido ubicados los símbolos en el diagrama IQ en las modulaciones QAM y PSK. Para las modulaciones PSK y QAM de orden 2 y 4, los símbolos pueden ser ubicados de igual forma en el diagrama IQ. En este estudio se ha querido probar la eficiencia de la 2QAM con sus dos símbolos en el eje real del diagrama IQ. Por otro lado, la 2PSK se ha probado con sus símbolos en los ejes real e imaginario. De esta forma se pretende testear que modulación es más eficaz, en un solo eje, o utilizando sus dos ejes. En la Figura 14 podemos ver el diagrama IQ de la modulación 2QAM a la izquierda y la modulación 2PSK a la derecha.

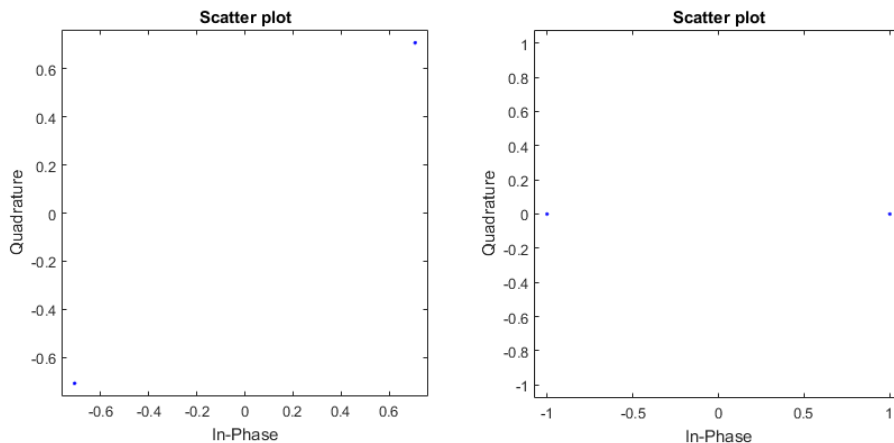


Figura 14. Diagrama IQ 2QAM y 2 PSK

Por lo que respecta a las modulaciones 4QAM y 4PSK, en este caso se ha decidido insertar los símbolos en los ejes real e imaginario tal y como se puede apreciar en la Figura 15. En este caso uno de los objetos de estudio es comprobar que se comportan de igual manera y que el tiempo de coherencia de canal se cumple correctamente en todas las transmisiones. Por otra parte, se pretenden demostrar que los ejes real e imaginario son totalmente independientes y se obtendrán los mismos resultados como si utilizáramos una 2PSK.

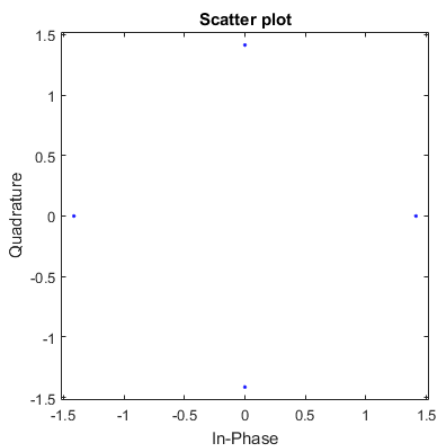


Figura 15. Diagrama IQ 4QAM y 4PSK

3.2.4 Test realizados

Para poder realizar el estudio con todas las posibilidades de transmisiones, se han efectuado 60 test diferentes. Por razones de logística, los test se han llevado a cabo en dos partes. La razón por la que se ha llevado a cabo en dos partes es debido a la presencia de una ionosonda en la

Base Antártica Española Juan Carlos I. La ionosonda operada por l'Observatori de l'Ebre transmite secuencias PN en todas las frecuencias comprendidas entre 1 MHz y 30 MHz. Como nuestro ámbito de operación se encuentra en 5 MHz, se decidió partir temporalmente los minutos del día en los que transmitiríamos. Para el caso de l'Observatori de l'Ebre, necesitaba todos aquellos minutos comprendidos entre el 1 y el 4, ambos incluidos. Para realizar nuestros test hicimos uso de todos aquellos minutos comprendidos entre 5 y 9, ambos incluidos. Dado la división de minutos imposibilitaba la capacidad de hacer todos los test cada hora del día, por lo que se dividió en dos partes.

La primera parte de test se mantiene dentro de los anchos de banda establecidos por los estándares, 2.3 KHz, pero varía las potencias de transmisión. Tal y como podemos ver en la Tabla 1 las potencias de transmisión son de 1 W, 3 W, 6 W, 12 W, 24 W. Esta primera parte del test se efectuó durante un total de 14 días. Por otro lado, la segunda parte de test, tal y como podemos observar en la Tabla 2, mantiene una única potencia de transmisión, 24 W. Esta potencia de transmisión se decidió en base a los resultados obtenidos en la primera parte, tal y como veremos más adelante en el apartado de Resultados. En cuanto a los test realizados en la segunda parte, se varían los anchos de banda fuera de los estándares, en 4.7 KHz, 10 KHz, 16.6 KHz, 20 KHz y 25 KHz.

N-Test	Minuto	Orden mod.	Bandwidth	TX Potecia
1-Test	5	2{PSK,FSK,QAM}	2.3 KHz	1 W
2-Test	6	4{PSK,FSK,QAM}	2.3 KHz	1 W
3-Test	7	8{PSK,FSK,QAM}	2.3 KHz	1 W
4-Test	8	16{PSK,FSK,QAM}	2.3 KHz	1 W
5-Test	9	32{PSK,FSK,QAM}	2.3 KHz	1 W
6-Test	15	2{PSK,FSK,QAM}	2.3 KHz	3 W
7-Test	16	4{PSK,FSK,QAM}	2.3 KHz	3 W
8-Test	17	8{PSK,FSK,QAM}	2.3 KHz	3 W
9-Test	18	16{PSK,FSK,QAM}	2.3 KHz	3 W
10-Test	19	32{PSK,FSK,QAM}	2.3 KHz	3 W
11-Test	25	2{PSK,FSK,QAM}	2.3 KHz	6 W
12-Test	26	4{PSK,FSK,QAM}	2.3 KHz	6 W
13-Test	27	8{PSK,FSK,QAM}	2.3 KHz	6 W
14-Test	28	16{PSK,FSK,QAM}	2.3 KHz	6 W
15-Test	29	32{PSK,FSK,QAM}	2.3 KHz	6 W
16-Test	35	2{PSK,FSK,QAM}	2.3 KHz	12 W
17-Test	36	4{PSK,FSK,QAM}	2.3 KHz	12 W
18-Test	37	8{PSK,FSK,QAM}	2.3 KHz	12 W
19-Test	38	16{PSK,FSK,QAM}	2.3 KHz	12 W
20-Test	39	32{PSK,FSK,QAM}	2.3 KHz	12 W
21-Test	45	2{PSK,FSK,QAM}	2.3 KHz	24 W
22-Test	46	4{PSK,FSK,QAM}	2.3 KHz	24 W
23-Test	47	8{PSK,FSK,QAM}	2.3 KHz	24 W
24-Test	48	16{PSK,FSK,QAM}	2.3 KHz	24 W
25-Test	49	32{PSK,FSK,QAM}	2.3 KHz	24 W
26-Test	55	2{PSK,FSK,QAM}	2.3 KHz	6 W

27-Test	56	4{PSK,FSK,QAM}	2.3 KHz	6 W
28-Test	57	8{PSK,FSK,QAM}	2.3 KHz	6 W
29-Test	58	16{PSK,FSK,QAM}	2.3 KHz	6 W
30-Test	59	32{PSK,FSK,QAM}	2.3 KHz	6 W

Tabla 1. Test de potencia realizados cada hora

N-Test	Minuto	Orden mod.	Bandwidth	
31-Test	5	2{PSK,FSK,QAM}	2.3 KHz	24 W
32-Test	6	4{PSK,FSK,QAM}	2.3 KHz	24 W
33-Test	7	8{PSK,FSK,QAM}	2.3 KHz	24 W
34-Test	8	16{PSK,FSK,QAM}	2.3 KHz	24 W
35-Test	9	32{PSK,FSK,QAM}	2.3 KHz	24 W
36-Test	15	2{PSK,FSK,QAM}	4.7 KHz	24 W
37-Test	16	4{PSK,FSK,QAM}	4.7 KHz	24 W
38-Test	17	8{PSK,FSK,QAM}	4.7 KHz	24 W
39-Test	18	16{PSK,FSK,QAM}	4.7 KHz	24 W
40-Test	19	32{PSK,FSK,QAM}	4.7 KHz	24 W
41-Test	25	2{PSK,FSK,QAM}	10 KHz	24 W
42-Test	26	4{PSK,FSK,QAM}	10 KHz	24 W
43-Test	27	8{PSK,FSK,QAM}	10 KHz	24 W
44-Test	28	16{PSK,FSK,QAM}	10 KHz	24 W
45-Test	29	32{PSK,FSK,QAM}	16.6 KHz	24 W
46-Test	35	2{PSK,FSK,QAM}	16.6 KHz	24 W
47-Test	36	4{PSK,FSK,QAM}	16.6 KHz	24 W
48-Test	37	8{PSK,FSK,QAM}	16.6 KHz	24 W
49-Test	38	16{PSK,FSK,QAM}	16.6 KHz	24 W
50-Test	39	32{PSK,FSK,QAM}	16.6 KHz	24 W
51-Test	45	2{PSK,FSK,QAM}	20 KHz	24 W
52-Test	46	4{PSK,FSK,QAM}	20 KHz	24 W
53-Test	47	8{PSK,FSK,QAM}	20 KHz	24 W
54-Test	48	16{PSK,FSK,QAM}	20 KHz	24 W
55-Test	49	32{PSK,FSK,QAM}	20 KHz	24 W
56-Test	55	2{PSK,FSK,QAM}	25 KHz	24 W
57-Test	56	4{PSK,FSK,QAM}	25 KHz	24 W
58-Test	57	8{PSK,FSK,QAM}	25 KHz	24 W
59-Test	58	16{PSK,FSK,QAM}	25 KHz	24 W
60-Test	59	32{PSK,FSK,QAM}	25 KHz	24 W

Tabla 2. Test de ancho de banda realizados cada hora

3.3 Implementación Matlab

La implementación de todo el software de creación de ficheros y postprocesado es una de las partes más extensas que han formado este estudio. La creación de todos los ficheros que serán transmitidos por la plataforma han de implementarse con un diseño muy bien estudiado dado que todas las transmisiones que se efectúen con ellos no se podrán repetir una vez finalizada la campaña antártica española. Todas las tramas creadas están diseñadas a consciencia. Por otra parte, todos los códigos de postprocesado siguen estrictamente los criterios utilizados para la transmisión de las tramas y así obtener la mayor eficiencia y veracidad en los resultados. Aun así, cabe destacar que el postprocesado puede ser modificado para aumentar a mejor sus resultados.

Para este estudio, como en el estudio anterior, ha sido todo implementado en Matlab a través de su software Matlab R2015a [31]. Se ha utilizado este tipo de programación debido a que Matlab es el entorno más sencillo y productivo para el trabajo de procesado de señales que no son en tiempo real. Por otra parte, el fin de este estudio es en la obtención de resultados que puedan ser divulgativos a la comunidad científica, por lo que a través de Matlab es fácil la creación de graficas que puedan ofrecer los resultados de una manera sencilla a entender.

En cuanto a lo que hace referencia a los códigos implementados, en este apartado veremos con detalle cada una de las funciones y partes que han hecho posible este estudio.

3.3.1 Cambios realizados

Antes de entrar en detalle en cada uno de los códigos que forman parte este estudio, cabe mencionar todos los cambios más importantes que han sido modificados respecto el anterior estudio.

Uno de los mayores cambios a nivel de eficiencia en el postprocesado para generar todas las gráficas es la eliminación del uso de Microsoft Excel. En el antiguo estudio se hacía uso de Microsoft Excel para guardar todos los datos de las tramas demoduladas. Este proceso de guardar los datos de cada paquete de una trama demodulada en una plantilla, a nivel de software era una operación que llevaba poco menos de un segundo, pero que en el cúmulo de todos los paquetes y tramas hacía el proceso muy largo. Además, en múltiples ocasiones la escritura de los datos generaba errores que hacían tener que reiniciar el código manualmente. Por otra parte, para la creación de las gráficas de resultados, el código Matlab debía leer todos los datos del fichero Excel, añadiendo más retraso en el código. Para evitar el uso de Microsoft Excel, para este estudio, se han guardado directamente los datos en ficheros Matlab .mat, de tal manera que todas las operaciones efectuadas son casi instantáneas reduciendo de manera vertical los tiempos dedicados al guardado de todos los paquetes demodulados. Para evitar cualquier pérdida de datos, en caso que el programa se cierre, se efectúan diversos guardados cada cierto tiempo de manera que se guarda el fichero .mat bajo el nombre de "TheMatrix.mat".

Otro de los mayores cambios efectuados en este estudio es el diseño de las tramas que serán transmitidas. En el diseño de las tramas, en el estudio anterior, éstas estaban formadas primeramente por todos los paquetes de una modulación, y finalmente por todos los paquetes de otra modulación. En este caso cada uno de los mismos datos de un paquete eran transmitidos en dos momentos temporales distintos que estaban fuera del tiempo de coherencia del canal. Este efecto podía influir en los resultados dado que no se estaban comparando dos paquetes de datos transmitidos en el mismo instante. Para el nuevo estudio, los paquetes de las tres modulaciones son transmitidas en un intervalo menor al tiempo de coherencia del canal, de manera que se pueden comparar en las mismas condiciones de canal. Cada paquete de trama se transmite en tiempos menores al tiempo de coherencia.

Algunos otros cambios que es importante mencionar, pero no con tan importancia, es la adición de una modulación nueva (PSK), de modificaciones de los anchos de banda de las modulaciones, el aumento de órdenes de modulación y la transmisión a potencias menores.

3.3.2 GenPNbyPolynomial.m

El código genPNbyPolynomial.m, tal y como podemos ver en la Figura 16 es la función que se encarga de generar las secuencias PN a través del polinomio que se le indique. Esta función nos retorna la secuencia PN resultado del polinomio y el número de chips que la forman.

```

10 %polynomial=[6 1 0]; %example: AX^3+BX^1+1 is represented as [X3 X2 X1 X0]->[A B 0]
11 N=2^polynomial(1)-1;
12
13 %convert polynomial representation to BnX^n+Bn-1x^n-1+...B1X^1+B0X^0 form
14 temp=zeros(1,max(polynomial)+1);
15 for i=1:length(polynomial)
16     temp(length(temp)-polynomial(i))=1;
17 end
18
19 polynomial=temp;
20 if nargin<2
21     initialCondition=[1 zeros(1,length(polynomial)-2)];
22 end
23
24 %initialize the initial state of the LFSR
25 x=initialCondition;
26
27 y=zeros(1,N);
28 for i=1:N
29     y(i) = x(end); %output;
30     xi = mod(sum(x.*polynomial(2:end)),2);
31     x(2:end)=x(1:end-1);
32     x(1)=xi;
33 end
34 pnSequence = 2*y-1;
35 chipIndex = 1:N;

```

Figura 16. Código genPNbyPolynomial.m

3.3.3 GenPNSequence.m

La función GenPNSequence es la función encargada de escoger que polinomios serán utilizados para cada orden de secuencia PN. Esta función a través del orden de secuencia PN indicado, nos retornará la secuencia PN con el orden correspondiente. Tal y como podemos ver en la Figura 17, la función recibe el orden de la secuencia y a través de un switch ejecuta la función genPNbyPolynomial.m con un polinomio correspondiente al orden de la secuencia PN deseada. A través de este código, se puede escoger cambiando para cada orden de secuencia PN que polinomio será el generador de está.

```
1 function [PN]=genPNSequence(order)
2
3     switch order
4     case 2
5         PN = genPNbyPolynomial([2 1 0]);
6     case 3
7         PN = genPNbyPolynomial([3 2 0]);
8     case 4
9         PN = genPNbyPolynomial([4 3 0]);
10    case 5
11        PN = genPNbyPolynomial([5 3 0]);
12    case 6
13        PN = genPNbyPolynomial([6 5 0]);
14    case 7
15        PN = genPNbyPolynomial([7 6 0]);
16    case 8
17        PN = genPNbyPolynomial([8 6 5 4 0]);
18    case 9
19        PN = genPNbyPolynomial([9 5 0]);
20    case 10
21        PN = genPNbyPolynomial([10 7 0]);
22    case 11
23        PN = genPNbyPolynomial([11 9 0]);
24    case 12
25        PN = genPNbyPolynomial([12 11 8 6 0]);
```

Figura 17. Código GenPNSequence.m

3.3.4 CreateIQData.m

Esta función es la última que forma parte de los generadores de tramas del estudio. Dentro de las funciones generadoras, este es el código más importante dado que es donde esta implementado todo el diseño descrito anteriormente en otros apartados. A través de este código, tal y como podemos ver en la Figura 18, se configuran todos los parámetros que caracterizan la trama resultante. Al ejecutar el código CreateIQData.m teniendo en cuenta todos los parámetros, se generará un fichero IQ preparado para ser leído por la plataforma desarrollada en el proyecto.

```

23 %Parametres variables
24 %-----
25 system.param.to.length = 6000;
26 system.param.to.frequency = 600; %(Hz)
27 system.param.pn.ordre = 6; %OPCION: 2 - 30
28 system.param.pn.Rm = 8; %Solo remuestreo en secuencia PN OPCIONS: 2.38KHz(Rm=42) 4.76KHz(Rm=21) 10KHz(Rm=10) 12.5 KHz(Rm=
29 system.param.mod.type = [1 1 1]; %OPCION: [PSK FSK QAM]
30 system.param.mod.M = 2; %OPCION: 2 4 8 16 32 64 128 256 512
31 system.param.mod.paquetsondeig = 50; %Cada paquet de sondeig inclou 3 modulacions (to, seq. PN i dades)
32 system.param.mod.eqsymbols = 0; %
33 system.param.mod.mostres = 10500; %EVITAR TOCAR Mostres de modulacio amb remostreig
34 system.param.mod.Rm = 42; %Solo remuestreo en modulacion OPCIONS: 2.38KHz(Rm=42) 5KHz(Rm=20) 10KHz(Rm=10) 16.66KHz(Rm=6)
35 system.param.raised = 1; %OPCION: 1-ON 0-OFF
36 system.param.Fm = 100000; %Frecuencia mostreig sistema
37 system.param.Power = 0; %OPCION: Màxim 0 dBm
38 system.param.PowerRef = 14090; %EVITAR TOCAR CALCULAT AMB RED PITAYA
39 system.param.rename = 'false'; %True rename file.IQ False automatic name file.IQ
40 system.param.bitsize = 'int16'; %EVITAR TOCAR bits por cada muestra del fichero
41 system.param.plots = 1; %OPCION: 1-ON 0-OFF
42 %-----
43 % SEQ. PN | To | SEQ. PN | BLOC DADES | To | SEQ. PN | BLOC DADES |
44 % ORDRE 6 | 600 Hz | ORDRE 6 | PREAMBUL | FSK | 600 Hz | ORDRE 6 | FSK |
45 % 504 mostres | 6000 mostres | 504 mostres | EQUALITZACIO | 250 mostres | 2000 mostres | 504 mostres | 250 mostres |
46 % Rm=8 | Rm=0 | Rm=8 | | sense Rm=? | Rm=0 | Rm=8 | sense Rm=? |
47 %-----

```

Figura 18. Configuración parámetros CreatelQData.m

En la Figura 18 vemos todos los parámetros que se pueden configurar en este código. Todos los parámetros podemos apreciar que están dentro de la variable system.param. A partir de este punto cada parámetro se diferencia según su función y pasa a formar parte de otro punto de esta variable. Por orden de variables podemos describirlas como:

- System.param.to.length: Nos permite configurar la longitud total del tono que se transmite al principio de cada paquete de modulación.
- System.param.to.frequency: Nos permite modificar la frecuencia del tono.
- System.param.pn.ordre: Nos permite configurar el orden que queremos en cada secuencia PN que aparece en cada paquete de modulación.
- System.param.pn.Rm: Nos permite caracterizar el remuestreo de nuestra secuencia PN. A partir de este remuestreo podremos modificar el ancho de banda de nuestra secuencia PN siendo el máximo 50 KHz, que sería poco eficaz.
- System.param.mod.type: Nos permite configurar que modulaciones queremos que se transmitan. A través del array, con un '1' indicamos que queremos que la modulación esté presente y con un '0' indicamos que no queremos que esté presente. En el array se indicarían las modulaciones de la siguiente manera: [PSK FSK QAM].
- System.param.mod.M: Nos permite configurar el orden de modulación de las modulaciones que serán transmitidas.
- System.param.mod.paquetsondeig: Nos permite configurar el número de paquetes de trama que queremos que contenga nuestra trama.
- System.param.mod.eqsymbols: Nos permite activar o desactivar la ecualización de las modulaciones que la contengan. En este caso, la QAM y la PSK.
- System.param.mod.mostres: Nos permite configurar la longitud en muestras de los bloques de modulación teniendo en cuenta los remuestreos posibles.
- System.param.mod.Rm: Nos permite configurar el remuestreo que se efectuara en los símbolos de las modulaciones de la trama. A través de este remuestreo podemos modificar el ancho de banda de los datos modulados. Como máximo podremos tener un ancho de banda de 50 KHz, aunque sería demasiado rápido.

- System.param.raised: Nos permite activar o desactivar la aplicación del filtro raised cosinus que se efectúa en los símbolos modulados y la secuencia PN.
- System.param.Fm: Esta variable se utiliza para configurar la frecuencia de muestreo a la que se procesaran todas las señales.
- System.param.Power: Esta variable nos permite configurar la potencia en dBm a la que queremos que nuestra trama sea transmitida por la salida de RedPitaya. Se ha de tener en cuenta el amplificador al que estará conectado para calcular la potencia de transmisión. Por otra parte, RedPitaya a partir de los 4 dBm se satura la salida, por lo que no es aconsejable superar los 0 dBm.
- System.param.PowerRef: Esta variable contiene el valor de referencia utilizado para calcular la potencia a la RedPitaya transmite las señales por su salida. Este valor nos permite calcular la potencia a través de la amplitud de las señales generadas en Matlab.
- System.param.rename: En el caso que esta variable sea 'false', el fichero generado por el código será nombrado de manera genérica con las características de la trama. En el caso de ser 'true', al generarse el fichero, el código nos pedirá con que nombre queremos llamar al fichero creado.
- System.param.bitsize: A través de esta variable se configura el número de bits que contiene cada paquete del fichero IQ. Para que sea compatible con la configuración de RedPitaya, éste es int16, ya que cada paquete es de 16 bits siendo 32 bits la I y la Q.
- System.param.plots: Nos permite activar con '1' y desactivar con '0' la proyección de todos los plots del código. En el caso de estar activo, se mostrarán todas las señales de la trama en los distintos momentos de montar la trama junto con información adicional.

En todos los casos, sea cuál sea la configuración, se respecta en todo momento la longitud de muestras de todos los bloques configurados. A partir de toda la configuración de todas las variables, el código hará uso de estas para formar la trama.

Primeramente, a partir de las variables referentes al tono y de la frecuencia de muestreo del sistema, se generará el tono en la variable TO, tal y como podemos ver en la Figura 19.

```

49 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
50 %                                CREACIO DEL TO                                %
51 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
52
53 %Creem el to de la trama
54 - TO = exp(-i*pi*system.param.to.frequency*(0:1/system.param.Fm:system.param.to.length/system.param.Fm));
55 %Mostrem el to
56 - if system.param.plots == 1
57 -     figure
58 -     plot(TO);
59 -     title('To')
60 - end
61 - disp('To creat');
```

Figura 19. Generación tono 600 Hz CreateIQData.m

Como siguiente paso, se procede a crear la secuencia PN a través de las variables configuradas referentes a ésta. Para generar la secuencia PN se hace uso de la función genPNSequence la cuál retorna la secuencia PN generada. Una vez generada, se crea los coeficientes del filtro de coseno alzado según el remuestreo configurado. Una vez creado se procesa la secuencia PN a través de la función upfirdn con los coeficientes creados del filtro de coseno alzado. Al finalizar todas las

operaciones efectuadas en la secuencia PN nos resultara la secuencia PN remuestreada, la cual se deben eliminar las primeras y últimas muestras que no forman parte de la secuencia PN sino del filtro de coseno alzado. En la Figura 20 se puede ver el código perteneciente a la creación de la secuencia PN.

```

62 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
63 %                                CREACIO DE LA PN                                %
64 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
65
66 %Creem la secuencia PN de la trama
67 - PN = genPNSequence(system.param.pn.ordre);
68
69 %Remostrejem la secuencia PN
70 - h = roosdesign(0.35,system.param.pn.Rm,system.param.pn.Rm);
71 - if system.param.raised == 1
72 -     PN_Rm = upfirdn(PN,h,system.param.pn.Rm,1);
73 -     retail = (length(PN_Rm)-length(PN)*system.param.pn.Rm)/2;
74 -     PN_Rm = PN_Rm(retail+system.param.pn.Rm/2:end);
75 -     PN_Rm = PN_Rm(1:end-retail+system.param.pn.Rm/2);
76 - end
    
```

Figura 20. Creación secuencia PN CreateIQData.m

Una vez creado el tono y la secuencia PN se procede a la creación de las modulaciones. Tal y como podemos ver en la Figura 21, primeramente se crean a través de la semilla rng(1) todos los bits que serán modulados. Posteriormente se crean los coeficientes del filtro de coseno alzado que se utilizara para remuestrear los símbolos de las modulaciones.

```

95 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
96 %                                CREACIO DE LES MODULACIONS                    %
97 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
98
99 %Creem els bits a modular
100 - rng(1);
101 - tx_size = (system.param.mod.mostres/system.param.mod.Rm)*system.param.mod.paquetsondeig;
102 - bitstx = randi([0 system.param.mod.M-1],tx_size,1);
103
104 %Filtre raised cosine per modulacions
105 - h = roosdesign(0.35,system.param.mod.Rm,system.param.mod.Rm);
    
```

Figura 21. Creación de bits y filtro CreateIQData.m

A partir de este punto se modularán los datos en tres modulaciones distintas: PSK, FSK y QAM. Tal y como podemos ver en la Figura 22, primeramente, se modulará la PSK teniendo en cuenta los parámetros de orden de modulación con los que se hayan configurado. En el caso de PSK, la modulación, en el caso de ser de orden 2, contiene un error en el que aparecen unos pulsos muy pequeños en la parte imaginaria de la señal. En el momento que se hace el escalado, estos pulsos obtienen una amplitud suficiente mayor para ser emitida por la Red Pitaya. Por ello se eliminan en las siguientes líneas. Una vez obtenida la señal modulada se procederá al remuestreo de la señal y su filtrado de coseno alzado. Todo este procesado lo hacemos únicamente con la función upfirdn cargando los coeficientes del filtro. Una vez remuestreada y filtrada la señal eliminamos las muestras pertenecientes al inicio del filtro.

```

107 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% PSK %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
108
109 %Creem la modulacio PSK
110 FSK = pskmod(bitstx,system.param.mod.M);
111
112 %Controlem error de pskmod: insereix polsos quadrats molt baixos a la part
113 %imaginaria
114 if system.param.mod.M == 2
115     FSK = complex(real(FSK));
116 end
117
118 %Remostrejem la modulacio PSK
119 %PSK_Rm = repelem(PSK,system.param.mod.Rm);
120
121 if system.param.raised == 1
122     FSK_Rm = upfirdn(FSK,h,system.param.mod.Rm);
123     retail = (length(FSK_Rm)-length(FSK)*system.param.mod.Rm)/2;
124     FSK_Rm = FSK_Rm(retail+system.param.mod.Rm/2:end);
125     FSK_Rm = FSK_Rm(1:end-retail+system.param.mod.Rm/2);
126
127 end
    
```

Figura 22. Modulaci3n PSK CreateIQData.m

Por lo que respecta al resto de modulaciones, la FSK, tal y como podemos ver en la Figura 23, no hace falta pasarlo por un filtro de coseno alzado porque no est3 formada por pulsos cuadrados sino por componentes frecuenciales. Al utilizar fskmod, indicamos todos los par3metros configurados de remuestreo, datos a modular, orden de la modulaci3n y frecuencia de muestreo.

```

158 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% FSK %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
159
160 %Creem la modulacio FSK i remostrejem
161 FSK_Rm = fskmod(bitstx,system.param.mod.M,(system.param.Fm/system.param.mod.Rm)/system.param.mod.M,system.param.mod.Rm...
162 ,system.param.Fm);
163
164 %Mostrem el remostreig
165 if system.param.plots == 1
166     figure
167     subplot(2,1,1)
168     plot(real(FSK_Rm))
169     title('FSK Real');
170     legend('FSK Rm');
171     subplot(2,1,2)
172     plot(imag(FSK_Rm))
173     title('FSK Imaginaria')
174     legend('FSK Rm');
175     figure
176     obw(FSK_Rm,100000)
177     %title('FFT FSK')
178 end
    
```

Figura 23. Modulaci3n FSK CreateIQData.m

Finalmente, la modulaci3n QAM sigue el mismo proceso que en el caso de la PSK. Tal y como podemos ver en la Figura 24, se hace us3 de la funci3n qammod para modular los datos, donde indicamos el orden de la modulaci3n y la fase de est3. Este 3ltimo par3metro para el estudio realizado es muy importante dado que en el caso de la modulaci3n QAM de orden 2 y orden 4 se ha utilizado una fase inicial de pi/4, a diferencia del resto. Por otra parte, una vez modulada la funci3n, igual que con la PSK, se remuestrea toda la se3al y se filtra con un filtro de coseno alzado para disminuir el ancho de banda.


```

181 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% QAM %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
182
183 %Creem la modulacio QAM
184 if(system.param.mod.M==2||system.param.mod.M==4)
185     QAM = qammod(bitstx,system.param.mod.M,pi/4);
186 else
187     QAM = qammod(bitstx,system.param.mod.M);
188 end
189
190 %Remostrejem la modulacio QAM
191 if system.param.raised == 1
192     QAM_Rm = upfirdn(QAM,h,system.param.mod.Rm,1);
193     retail = (length(QAM_Rm)-length(QAM)*system.param.mod.Rm)/2;
194     QAM_Rm = QAM_Rm(retail+system.param.mod.Rm/2:end);
195     QAM_Rm = QAM_Rm(1:end-retail+system.param.mod.Rm/2);
196 end
    
```

Figura 24. Modulaci3n QAM CreateIQData.m

Una vez generado todas las se1ales necesarias para transmitir la trama, tal y como podemos ver en la Figura 25, se procede al escalado de la se1al de acuerdo con la potencia que se ha configurado para que sea transmitida a trav3s de Red Pitaya. Podemos ver que se hace el escalado independientemente para cada se1al generada. Para las modulaciones el escalado se tiene que hacer primeramente para la parte real y posteriormente para la parte imaginaria tal y como podemos observar.

```

225 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% ESCALEM ELS SENYALS SEGONS POTENCIA %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
226 %
227 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
228
229 system.param.Voltage = sqrt(((10^(system.param.Power/10))/1000)*50);
230 Esc = system.param.Voltage*system.param.PowerRef;
231
232 %Escalem el to
233 TO = TO.*(Esc/max(TO));
234
235 %Escalem la sequencia PN
236 PN_Rm_max = max(abs(max(PN_Rm)),abs(min(PN_Rm)));
237 PN_Rm = PN_Rm.*(Esc/PN_Rm_max);
238
239 %Escalem la modulacio PSK
240 PSK_Rm_max = max(abs(max(real(PSK_Rm))),abs(min(real(PSK_Rm))));
241 if PSK_Rm_max ~= 0
242     PSK_Rm = real(PSK_Rm)*(Esc/PSK_Rm_max) + li*imag(PSK_Rm);
243 end
244 PSK_Rm_max = max(abs(max(imag(PSK_Rm))),abs(min(imag(PSK_Rm))));
245 if PSK_Rm_max ~= 0
246     PSK_Rm = li*imag(PSK_Rm)*(Esc/PSK_Rm_max) + real(PSK_Rm);
247 end
    
```

Figura 25. Escalado de potencia CreateIQData.m

Tras el escalado, las diferentes modulaciones se ordenan en paquetes de 10500 muestras a trav3s de un reshape para posteriormente tratar los distintos paquetes con m3s facilidad. Para ir acabando podemos ver en la Figura 26 un loop en el que se van a1adiendo los diferentes paquetes de trama seg3n las modulaciones que se han indicado que deben ser transmitidas. Este loop se efect3a tantas veces como paquetes se hayan configurado para que forme la trama.

```

278 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
279 %                                CREACIO DE LA TRAMA I FITXERS                                %
280 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
281
282 %Trama = [PN_Rm PN_Rm PN_Rm PN_Rm];
283 Trama = PN_Rm;
284
285 for npaquet = 1:1:system.param.mod.paquetsondeig
286     if system.param.mod.type == [1 1 1]
287         Trama = horzcat(Trama, TO, PN_Rm, FSK_Rm(npaquet,:), TO, PN_Rm, FSK_Rm(npaquet,:), TO,...
288             PN_Rm, QAM_Rm(npaquet,:));
289     end
290     if system.param.mod.type == [1 0 0]
291         Trama = horzcat(Trama, TO, PN_Rm, FSK_Rm(npaquet,:));
292     end
293     if system.param.mod.type == [0 1 0]
294         Trama = horzcat(Trama, TO, PN_Rm, FSK_Rm(npaquet,:));
295     end
296     if system.param.mod.type == [0 0 1]
297         Trama = horzcat(Trama, TO, PN_Rm, QAM_Rm(npaquet,:));
298     end
299     if system.param.mod.type == [0 0 0]
300         Trama = horzcat(Trama, TO, PN_Rm);
301     end
302 end
    
```

Figura 26. Creación trama CreateIQData.m

Finalmente, una vez creada la trama final, esta se forma intercalando muestras reales con imaginarias dado que es el formato a través del cual Red Pitaya podrá leer correctamente la trama que se quiere transmitir como podemos ver en la Figura 27. Una vez creada la trama con el protocolo de lectura de Red Pitaya se procede a crearla en un fichero que podrá ser llamado de manera genérica o renombrar como el usuario quiera. Es importante al crear la trama que cada una de las muestras este formada por 16 bits dado que es el número de bits que leerá Red Pitaya para cada muestra.

```

315 %Trama fichero
316 Tramafile(1:2:length(Trama)*2) = real(Trama);
317 Tramafile(2:2:length(Trama)*2) = imag(Trama);
318
319 switch system.param.rename
320     case 'true'
321         prompt = '%Nombre del fichero?';
322         file = input(prompt,'s');
323         file = strcat(file, '.iq')
324     otherwise
325         file = strcat('Trama_', 'TO', num2str(system.param.to.length), '_', 'PN', num2str(system.param.pn.ordre), ...
326             '_', num2str(system.param.pn.Rm), '_', 'MOD', num2str(system.param.mod.M), '[_]', num2str(system.param.mod.type(1)), ...
327             '[_]', num2str(system.param.mod.type(2)), '[_]', num2str(system.param.mod.type(3)), '[_]', num2str(system.param.mod.Rm), ...
328             '_PAQ', num2str(system.param.mod.paquetsondeig), '_PWR', num2str(system.param.Power), '_R', ...
329             num2str(system.param.raised), '.iq')
330     end
331
332 fid = fopen (file, 'w');
333 fwrite(fid, Tramafile, system.param.bitsize);
334 fclose(fid);
335 disp('Trama creada');
    
```

Figura 27. Creación ficheros IQ CreateIQData

3.3.5 Demodtrama.m

La función Demodtrama.m es la función más importante y compleja de todo el software creado con Matlab. Esta función es la cual nos permitirá leer y demodular cualquier señal recibida en Red Pitaya. Tal y como veremos a continuación, podremos entender el funcionamiento de todo este código. Primeramente, como podemos ver en la Figura 28, el código tiene dos modos de funcionamiento, el cual se activa a través de la variable MODE. Esta variable debe valer 1 cuando se quiere demodular el fichero indicado. En el caso que se vaya a utilizar la función

MassiveDemodTrama.m como veremos más adelante, esta variable deberá configurarse en 0. Tal y como vemos, si el modo es el de una simple demodulación, a continuación, aparecen todo de parámetros para ser configurados según la señal que vayamos a demodular. Los parámetros que aparecen son los mismo que en el caso de la creación de las tramas en CreateIQData.m a diferencia de los siguientes que son añadidos:

- System.param.file: Esta variable es donde indicamos el path en el que se encuentra el fichero IQ que vamos a demodular.
- System.param.mod.eqsymbols: Esta variable nos permite configurar si queremos que no se ecualicen las modulaciones recibidas.
- System.param.PassbandFrequencypn: Esta variable nos permite configurar la frecuencia de corte del filtro que se utilizara para filtrar las secuencias PN.
- System.param.StopbandFrequencypn: Esta variable nos permite configurar la frecuencia de la banda de rechazo del filtro que se utilizara para filtrar las secuencia PN.
- System.param.PassbandFrequencyto: Esta variable nos permite configurar la frecuencia de corte del filtro que se utilizara para filtrar los tonos.
- System.param.StopbandFrequencyto: Esta variable nos permite configurar la frecuencia de la banda de rechazo del filtro que se utilizara para filtrar los tonos.
- System.param.PassbandFrequencymod: Esta variable nos permite configurar la frecuencia de corte del filtro que se utilizara para filtrar las modulaciones.
- System.param.StopbandFrequencymod: Esta variable nos permite configurar la frecuencia de la banda de rechazo del filtro que se utilizara para filtrar las modulaciones.

En el caso que MODE este configurado para utilizarse para el MassiveDemodTrama.m, los parámetros de orden de modulación, filtros, nombre del fichero y remuestreo serán variables de entrada de la función. Más adelante lo veremos con más detalle en MassiveDemodTrama.m.

```

10 - if(MODE == 1)
11     %Parametres variables
12     %-----c1-----
13     %system.param.file = 'Trama_TO6000_PN6_8_MOD16[1_1_1]_42_FAQ50_PWR-10.71_R1.iq';
14     system.param.file = 'D:\Documents\Dades Antartida\Caleta_Arentina\2019-2-7-11-2.txt';
15     system.param.file = '2019-9-25-15-10.txt';
16
17     %system.param.file = 'C:\Users\envis\Desktop\MatlabV3\1.txt';%D:\Documents\Dades Antartida\Caleta_Arentina\2019
18     system.param.PowerRef = 1; %EVITAR TOCAR CALCULAT AMB RED PITAYA
19     system.param.to.frequency = 600; %(Hz)
20     system.param.to.length = 6000;
21     system.param.pn.ordze = 5; %OPCIONES: 2 - 30
22     system.param.pn.Rm = 16; %Solo reemostreo en secuencia PN OPCIONS: 2.38KHz(Rm=42) 4.76KHz(Rm=21) 10KHz(Rm=10)
23     system.param.mod.type = [1 1 1]; %OPCIONES: [PSK FSK QAM]
24     system.param.mod.M = 2; %OPCIONES: 2 4 8 16 32 64
25     system.param.mod.paquetsondeig = 50; %Cada paquet de sondeig inclou 3 modulacions (to, seq. PN i dades)
26     system.param.mod.eqsymbols = 0;
27     system.param.mod.Rm = 42; %Solo reemostreo en modulacion OPCIONS: 2.38KHz(Rm=42) 4.76KHz(Rm=20) 10KHz(Rm=10) 16.
28     system.param.mod.mostres = 10500; %EVITAR TOCAR Mostres de modulacio amb reemostreig
29     system.param.raised = 1; %OPCIONES: 1-ON 0-OFF
30     system.param.Fm = 100000; %Frecuencia mostreig sistema
31     system.param.Power = -7.71; %OPCIONES: Máxim 0 dBm
32     system.param.rename = 'false'; %True rename file.IQ False automatic name file.IQ
33     system.param.bitsize = 'int16'; %EVITAR TOCAR bits por cada muestra x del fichero
34     system.param.PassbandFrequencypn = 1000; %10500
35     system.param.StopbandFrequencypn = 2300; %11000
    
```

Figura 28. Parámetros Demodtrama.m

Posteriormente encontraremos una serie de variables que en este caso no son configurables. Estas variables son aquellas que almacenaran todos los datos que deban tenerse en cuenta para mostrar los resultados de la demodulación.

- System.param.numpaquetmod: Nos indica cuantos bloques de modulación han sido demodulados.
- System.param.numpaquet: Nos indica cuantos paquetes de modulación han sido demodulados.
- System.param.snr: Es un array en el que se guardara el SNR de cada bloque de modulación.
- System.param.EbNo: Es un array en el que se guardara el EbNo de cada bloque de modulación.
- System.param.multipathkm: Es un array en el que se guardara la distancia en km de los multicaminos calculados en las correlaciones de las secuencias PN. El código perteneciente a este cálculo no ha sido finalizado.
- System.param.snrmultipath: Es un array en el que se guardara la distancia el SNR de los multicaminos calculados en las correlaciones de las secuencias PN. El código perteneciente a este cálculo no ha sido finalizado.
- System.param.BER; Es un array en el que se guardan todos los BER resultantes de cada demodulación.
- System.param.PotRx: Es un array en el que se guardan los valores de potencia recibidos de las modulaciones de la señal.
- System.param.Cat: Es un array en el que se guardara el número de categoría del multicamino según el SNR que contenga, de manera que se podrá conocer la magnitud de sus efectos y caracterizarlo. El código perteneciente a este cálculo no ha sido finalizado.
- System.param.marca: Es una variable que nos permite marcar la trama recibida en el caso que no se haya podido demodular a pesar de contener señal.
- System.param.piccorr: Es un array en el cuál se guardan todos los picos de correlación para mostrar posteriormente la respuesta del canal.

Una vez establecido todos los parámetros del código se iniciará el postprocesado de la señal como podemos ver en la Figura 29. Primeramente, se cargarán los datos del fichero IQ teniendo en cuenta que cada muestra son 16 bits del fichero y que están ordenados por real e imaginario consecutivamente. Una vez cargada toda la trama de la señal recibido, se procederá al escalado según la potencia recibida en la Red Pitaya. Una vez cargados los datos se filtrará toda la señal a través del filtro de secuencias PN para eliminar todas aquellas interferencias que no nos interesan.

```

94 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
95 %                               CARREGUEM DADES DE FITXER IQ                               %
96 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
97
98 %Carreguem les dades del fitxer IQ
99 fid=fopen(system.param.file,'r');
100 data = fread(fid,system.param.bitsize);
101 %data = fread(fid,'float');
102
103 fclose(fid);
104 data = data(1:4:end)+1i*data(2:4:end);
105 Esc = 10^((system.param.Power-4.15)/10)*system.param.PowerRef;
106 data = data./Esc;
107
108 %Filtrem les dades
109 filterpn = designfilt('lowpassfir', 'PassbandFrequency',system.param.PassbandFrequencypn, 'StopbandFrequency', system
110 data = filter(filterpn,data);

```

Figura 29. Cargado de ficheros IQ Demodtrama.m

Posteriormente se procederá a la creación de la secuencia PN que se ha utilizado en las tramas con su remuestreo correspondiente como se puede ver en la Figura 30. Una vez creada la secuencia PN se correlara está con toda la trama recibida para encontrar todos los picos de correlación, y concretamente el primer pico de correlación donde se encuentra el inicio de la trama y de sus datos. Además de encontrar el pico de correlación, respecto el primero se revisará cuantas muestras hay entre el primer pico y el segundo validando que hay la correcta distancia entre ellos (según la trama recibida) y por lo tanto que corresponde a picos de correlación. En el caso que no sea correcto el cálculo de distancia realizado, se indicara que la señal PN inicial no es válido y el código acabara añadiendo una marca al señal recibido a través de system.param.marca.

```

146 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
147 %                                PROCESSAT DEL SENYAL                                %
148 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
149
150 %Creem la secuencia PN de la trama
151 PN = genPNSequence(system.param.pn.ordre);
152
153 %Remostrejem la secuencia PN
154 h = rcoosdesign(0.35,system.param.pn.Rm,system.param.pn.Rm);
155 if system.param.raised == 1
156     PN_Rm = upfirdn(PN,h,system.param.pn.Rm,1);
157     retail = (length(PN_Rm)-length(PN)*system.param.pn.Rm)/2;
158     PN_Rm = PN_Rm(retail+system.param.pn.Rm/2:end);
159     PN_Rm = PN_Rm(1:end-retail-system.param.pn.Rm/2);
160 end
161
162 %Correlem total la trama per detectar l'inici
163 dataxcorr = xcorr((PN_Rm),real(data));
164 dataxcori = xcorr((PN_Rm),imag(data));
165 datacorr = abs(complex(dataxcorr,dataxcori));
166 datacorr = datacorr(1:floor(size(datacorr)/2));
167
168 [pot,pos] = findpeaks(flip(datacorr),'NPeaks',system.param.mod.paquetsondeig*sum(system.param.mod.type)+1 ...
169 , 'SortStr','descend','MinPeakDistance',900);
170
171 pos = sort(pos);
    
```

Figura 30. Correlación secuencia PN Demodtrama.m

En la Figura 31 podemos ver una vez verificada la señal que estamos procesando que se procede a obtener un conjunto de muestras del principio de la señal a la que corresponde ruido. Este ruido obtenido será utilizado para caracterizar el SNR de la señal. Por otra parte, nos posicionaremos en el punto en el que hemos encontrado el primer pico de correlación, punto en el cuál siguientemente encontraremos el tono y los datos de la primera modulación. Además como podemos ver se crea un filtro de coseno alzado para interpolar los símbolos de las modulaciones y se crea un array con los datos que se han enviado dado que se ha utilizado la semilla de bits aleatorios rng(1). Finalmente se entrará en un bucle que acabará cuando todos los paquetes hayan sido demodulados o en alguna parte del código se vea la imposibilidad de seguir demodulando debido a un EbNo muy bajo o a una interferencia.

```

205 - disp('Senyal PN inicial valid')
206 - %filterto = designfilt('lowpassfir', 'PassbandFrequency',system.param.PassbandFrequencyto, 'StopbandFrequency', syste
207 - %filtermod = designfilt('lowpassfir', 'PassbandFrequency',system.param.PassbandFrequencymod, 'StopbandFrequency', sys
208 -
209 - system.param.noise = data(pos(1)+length(PN_Rm)+1-system.param.mod.mostres:pos(1)+length(PN_Rm));
210 -
211 - %Ens posicionem davant del primer paquet
212 - data = data(pos(1)+length(PN_Rm)+1:end);
213 -
214 - %Creem el raised cosinus
215 - h = rcosdesign(0.35,system.param.mod.Rm,system.param.mod.Rm);
216 -
217 - %Generem les dades a demodular
218 - rng(1);
219 - tx_size = (system.param.mod.mostres/system.param.mod.Rm)*system.param.mod.paquetsondeig;
220 - bitstx = randi([0 system.param.mod.M-1],tx_size,1);
221 -
222 - while system.param.numpaquet<system.param.mod.paquetsondeig-1

```

Figura 31. Sincronización, generación de bits y filtro Demodtrama.m

Una vez iniciado el bucle, la primera modulación que se demodula es la PSK. Para ello lo primero que se debe hacer es corregir el efecto doppler shift, dado que si no está corregido no podremos demodular correctamente la señal. Para ello a través de la función fftshift analizaremos las muestras correspondientes al tono para medir la frecuencia que tiene y conocer el doppler en hercios que hay que corregir. En la Figura 32 podemos ver el proceso de corrección. Una vez medida la frecuencia esta se corregirá directamente en la señal recibida tal y como podemos ver en la línea de código 239. Una vez corregida la frecuencia se volverá a correlar la trama para encontrar el inicio exacto de los datos, dado que el doppler shift puede afectar en la sincronización de los datos.

```

224 - %***** PSK *****
225 - %Trobem el doppler de PSK
226 - datafreq = fftshift(fft(data(1:system.param.to.length),system.param.Fm*100));
227 - [~,pos] = max(datafreq);
228 -
229 - if((pos-system.param.Fm*100/2)/(-100)<0)
230 -     freq = system.param.to.frequency+(pos-system.param.Fm*100/2)/(-100)
231 - else
232 -     freq = system.param.to.frequency-(pos-system.param.Fm*100/2)/(-100)
233 - end
234 -
235 - t = 1:(system.param.Fm):400;
236 - t = t(1:length(data(1:end)));
237 - p1 = exp(2*(1j)*pi*(freq).*t);
238 - %Corregim la fase
239 - data_demod = data(1:end).*p1;
240 -
241 - %Correlem per trobar l'inici de les dades
242 - dataxcorr = xcorr((PN_Rm),real(data(1:2*length(PN_Rm)+2*system.param.to.length+2*system.param.mod.mostres)));
243 - dataxcorr_i = xcorr((PN_Rm),imag(data(1:2*length(PN_Rm)+2*system.param.to.length+2*system.param.mod.mostres)));
244 - datacorr = abs(complex(dataxcorr,dataxcorr_i));
245 - datacorr = datacorr(1:floor(size(datacorr)/2));

```

Figura 32. Cálculo y corrección del efecto doppler Demodtrama.m

A partir de la nueva correlación efectuada para identificar el inicio exacto de los datos modulados, también se utiliza para poder identificar el siguiente pico de correlación de la trama, en este caso correspondiente al bloque de datos modulados en FSK. A través de esta correlación podemos medir el número de muestras que hay entre el primer pico de correlación y el próximo correspondiente al siguiente paquete. En el caso que el número de muestras sea el esperado se considerará que no hay problemas con la demodulación y se seguirá. En el caso que no haya el número de muestras esperadas se efectuará un return en el código retornando todos los datos de la trama medidos hasta el momento. En la Figura 33 podemos ver el código correspondiente a la parte descrita.

```

247 - [pot,pos] = findpeaks(flip(datacorr),'NPeaks',2,'SortStr','descend','MinPeakDistance',900);
248 - [pos, B] = sort(pos);
249
250 - pos(2)-pos(1)
251 - length(FN_Rm)+system.param.to.length+system.param.mod.mostres+10
252 - if ((pos(2)-pos(1)>length(FN_Rm)+system.param.to.length+system.param.mod.mostres+10) || (pos(2)-pos(1)<(length(FN
253 - disp('Senyal FSK no valid')
254 - if(system.param.numpaquetmod>0)
255 -     system.param.numpaquetmod = system.param.numpaquetmod -1;
256 -     end
257 -     return
258 - end
    
```

Figura 33. Punto de control Demodtrama.m

En el caso que los cálculos realizados en cuanto a la distancia entre los picos de correlación sean correctos, se procederá a los cálculos pertinentes al SNR y del EbNo de cada paquete a demodular como podemos ver en la Figura 34. Para calcular el SNR se hace uso de la función snr a través de la cual únicamente se dará el ruido recibido en el inicio de la trama y posteriormente las muestras pertenecientes a la modulación, en total 10500. A través del valor del SNR recibido en ese paquete, se podrá calcular el EbNo a través de la fórmula: $EbNo = SNR + 10\log(BWc) - 10\log\left(\frac{Fm}{Rm}\right)$. Una vez obtenido los valores del SNR y el EbNo del bloque recibido, se procederá al downsampling de los símbolos de la modulación recibida. Para hacer el downsampling se hará uso de la función upfirdn en la que se le cargaran los coeficientes utilizados en el filtro de coseno alzado en el upsampling y se le indicara el número de muestras utilizado en el remuestreo. Una vez hecho el downsampling se deberán eliminar aquellas muestras correspondientes al inicio del filtrado, esas muestras no corresponden a la señal remuestreada por lo que se deben eliminar.

```

300 - %Ens posicionem davant les dades
301 - data_demod = data_demod(pos(1)+length(FN_Rm)+1:pos(1)+length(FN_Rm)+system.param.mod.mostres);
302 - %data_demod = filter(filtermod,data_demod);
303
304 - %Calculuem potencia i system.param.EbNo
305 - system.param.snr(system.param.numpaquet*3+1) = snr((abs(data_demod(1:system.param.mod.mostres))),...
306 -     abs(system.param.noise))/2;
307 - X = ['system.param.SNR ', num2str(system.param.mod.M), 'PSK Paquet ', num2str(system.param.numpaquet), ...
308 -     ': ', num2str(system.param.snr(system.param.numpaquet*3+1))];
309 - disp(X)
310 - system.param.EbNo(system.param.numpaquet*3+1) = system.param.snr(system.param.numpaquet*3+1)+...
311 -     10*log10(system.param.StopbandFrequencypt)-10*log10((100000/system.param.mod.Rm)*log2(system.param.mod.M));
312 - X = ['system.param.EbNo ', num2str(system.param.mod.M), 'PSK Paquet ', num2str(system.param.numpaquet), ...
313 -     ': ', num2str(system.param.EbNo(system.param.numpaquet*3+1))];
314 - disp(X)
315
316 - %Fem downsample de les dades PSK
317 - data_demod = upfirdn(data_demod,h,1,system.param.mod.Rm);
318
319 - data_demod = data_demod(system.param.mod.Rm/2+1:end-system.param.mod.Rm/2);
320
321 - %Eliminem els 3 primers símbols i els dos ultims degut a que el
322 - %filtre FIR no ho fa be per els primers
323 - data_demod= data_demod(4:end-2);
324 - %Escalem PSK a 1
325 - PSK_Rm_max = max(abs(max(real(data_demod))),abs(min(real(data_demod)))));
    
```

Figura 34. Cálculo SNR y EbNo Demodtrama.m

Una vez recuperada la señal modulada en su formato inicial antes de ser remuestreada, se procederá al escalado de esta para que pueda ser demodulada en todos sus aspectos iniciales. Para hacer el escalado tal y como podemos observar en la Figura 35, esté se hace tanto de la parte real como de la imaginaria. Tras el escalado de la señal antes de ser demodulados los símbolos de esta, se deberán ecualizar. Este es un proceso muy importante dado que los símbolos de las modulaciones PSK y QAM se encontrarán desfasados y por lo tanto sin

corresponder a los bits que pretenden representar. En el caso de la FSK, el desfase de la señal no afecta a los símbolos que representa cada frecuencia. Para la ecualización se hará uso de un ecualizador RLS, para su correcto funcionamiento se necesitan algunos símbolos conocidos para ecualizar correctamente. Para este caso se ha utilizado los 50 primeros símbolos que haya presentes. Tras la ecualización de los diferentes símbolos que componen la señal modulada, se deberán eliminar las 9 muestras iniciales dado que no son pertenecientes a la señal, sino al inicio del ecualizador.

```

327 -         if (system.param.mod.type(1)==1)
328 -             if PSK_Rm_max ~= 0
329 -                 data_demod = real(data_demod)*(1/PSK_Rm_max) + li*imag(data_demod);
330 -             end
331 -             PSK_Rm_max = max(abs(max(imag(data_demod))),abs(min(imag(data_demod))));
332 -             if PSK_Rm_max ~= 0
333 -                 data_demod = li*imag(data_demod)*(1/PSK_Rm_max) + real(data_demod);
334 -             end
335 -
336 -             %Equalitzacio
337 -             numFFtaps = 1;
338 -             numFBTaps = 1;
339 -             eql = dfe(numFFtaps,numFBTaps,xls(0.99,0.99));
340 -             %eql = linearreq(1,lms(0.01*test));
341 -             eql.ResetBeforeFiltering = 0;
342 -
343 -             eql.SigConst = pskmod(0:system.param.mod.M-1,system.param.mod.M);
344 -             txSymbols = pskmod(bitstrx(system.param.numpaquet*system.param.mod.mostres/system.param.mod.Rm+1+3:...
345 -                 (system.param.mod.mostres/system.param.mod.Rm)-2),system.param.mod.M);
346 -             data_demod_eq = equalize(eql,data_demod,txSymbols(1:50));
347 -
348 -
349 -             %Eliminem els tres primers símbols deguts a lequalitzador
350 -             data_demod_eq = data_demod_eq(9:end);
351 -             data demod = data demod(9:end);

```

Figura 35. Ecualización modulaciones Demodtrama.m

Para ir acabando, una vez ecualizado todos los símbolos se procederá a la demodulación de la señal de acuerdo a como se ha modulado tal y como podemos ver en la Figura 36. Una vez demodulados todos los símbolos, los bits resultantes recibidos se compararán con los bits transmitidos con tal de obtener el BER de cada paquete de la trama. Para obtener la máxima cantidad de datos de la trama, también se demodula la trama sin ser ecualizada para ver cuáles son los efectos del desfase producidos en los símbolos.

```

353 -             %Demodulacio sense equalitzar
354 -             rxbits = (pskdemod(data_demod,system.param.mod.M));
355 -
356 -             system.param.BER(system.param.numpaquet*3+1) = biterr(bitstrx(system.param.numpaquet*...
357 -                 system.param.mod.mostres/system.param.mod.Rm+1+3+8:(system.param.numpaquet*...
358 -                 (system.param.mod.mostres/system.param.mod.Rm)+(system.param.mod.mostres/system.param.mod.Rm)-2),...
359 -                 rxbits)/((system.param.mod.mostres/system.param.mod.Rm)*log2(system.param.mod.M));
360 -             X= ['system.param.BER ',num2str(system.param.mod.M),'PSK Paquet no eq ',num2str(system.param.numpaquet),...
361 -                 ':',num2str(system.param.BER(system.param.numpaquet*3+1))];
362 -             disp(X)
363 -
364 -             %Demodulacio equalitzada
365 -             rxbits = (pskdemod(data_demod_eq,system.param.mod.M));
366 -
367 -             system.param.BER(system.param.numpaquet*3+1) = biterr(bitstrx(system.param.numpaquet*...
368 -                 system.param.mod.mostres/system.param.mod.Rm+1+3+8:(system.param.numpaquet*...
369 -                 (system.param.mod.mostres/system.param.mod.Rm)+(system.param.mod.mostres/system.param.mod.Rm)-2),rxbits)/...
370 -                 ((system.param.mod.mostres/system.param.mod.Rm)*log2(system.param.mod.M));
371 -             X= ['system.param.BER ',num2str(system.param.mod.M),'PSK Paquet eq ',num2str(system.param.numpaquet),...
372 -                 ':',num2str(system.param.BER(system.param.numpaquet*3+1))];
373 -             disp(X)

```

Figura 36. Cálculo BER PSK Demodtrama.m

Una vez demodulado el paquete y obtenido todos los datos necesarios para añadirlos al array de datos de la trama, se procederá al siguiente paquete, que en este caso será la modulación

FSK y posteriormente la QAM. Este proceso se repetirá hasta salir del bucle inicial en el que se tienen en cuenta todos los paquetes que se van a demodular. En la Figura 37 podemos observar la demodulación de la modulación FSK. En este caso se ha seguido el mismo proceso que la PSK de sincronización y corrección de efecto doppler shift. A diferencia de la PSK y la QAM en este caso como podemos ver no hace falta ecualizar antes de demodular sus datos.

```

497 %Escalem FSK a 1
498 FSK_Rm_max = max(abs(max(real(data_demod))),abs(min(real(data_demod))));
499 if (system.param.mod.type(2)==1)
500     if FSK_Rm_max ~= 0
501         data_demod = real(data_demod)*(1/FSK_Rm_max) + li*imag(data_demod);
502     end
503     FSK_Rm_max = max(abs(max(imag(data_demod))),abs(min(imag(data_demod))));
504     if FSK_Rm_max ~= 0
505         data_demod = li*imag(data_demod)*(1/FSK_Rm_max) + real(data_demod);
506     end
507
508 rxbits = (fskdemod(data_demod,system.param.mod.M,(system.param.Fm/system.param.mod.Rm)/...
509 system.param.mod.M,system.param.mod.Rm,system.param.Fm));
510 rng(1);
511 tx_size = (system.param.mod.mostres/system.param.mod.Rm)*system.param.mod.paquetsondeig;
512 bitstx = randi([0 system.param.mod.M-1],tx_size,1);
513 system.param.BER(system.param.numpaquet*3+2) = biterr(...
514 bitstx(system.param.numpaquet*system.param.mod.mostres/system.param.mod.Rm+1:...
515 (system.param.numpaquet*(system.param.mod.mostres/system.param.mod.Rm)+...
516 (system.param.mod.mostres/system.param.mod.Rm)),rxbits)/...
517 ((system.param.mod.mostres/system.param.mod.Rm)*log2(system.param.mod.M));
518 X= ['system.param.BER ',num2str(system.param.mod.M),'FSK Paquet ',num2str(system.param.numpaquet),...
519 ':',num2str(biterr(bitstx(system.param.numpaquet*system.param.mod.mostres/system.param.mod.Rm+1:...
520 (system.param.numpaquet*(system.param.mod.mostres/system.param.mod.Rm)+...
521 (system.param.mod.mostres/system.param.mod.Rm)),rxbits)/...
522 ((system.param.mod.mostres/system.param.mod.Rm)*log2(system.param.mod.M))];
523 disp(X)

```

Figura 37. Cálculo BER FSK Demodtrama.m

Finalmente, en la Figura 38 podemos observar el código en el que se demodula la QAM. En el caso de la QAM como hemos mencionado, se debe ecualizar antes de ser demodulada. Como podemos observar, una vez acabada la demodulación de los símbolos se aumentan los contadores de las variables system.param.numpaquet y system.param.numpaquetmod para llevar el conteo del número de paquetes que se han demodulado en el bucle general de las tres modulaciones.

```

746 if(system.param.mod.M==2||system.param.mod.M==4)
747     rxbits = (qamdemod(data_demod_eq,system.param.mod.M,pi/4));
748 else
749     rxbits = (qamdemod(data_demod_eq,system.param.mod.M));
750 end
751
752 system.param.BER(system.param.numpaquet*3+3) = biterr...
753 (bitstx(system.param.numpaquet*system.param.mod.mostres/system.param.mod.Rm+1+3+8:...
754 (system.param.numpaquet*(system.param.mod.mostres/system.param.mod.Rm)+(system.param.mod.mostres/...
755 system.param.mod.Rm)-2),rxbits)/((system.param.mod.mostres/system.param.mod.Rm)*...
756 log2(system.param.mod.M));
757 X= ['system.param.BER ',num2str(system.param.mod.M),'QAM Paquet eq ',...
758 num2str(system.param.numpaquet),':',num2str(system.param.BER(system.param.numpaquet*3+3))];
759 disp(X)
760
761 if system.param.plots == 1
762     scatterplot(data_demod_eq);
763     title('QAM equalitzada');
764 end
765 end
766
767 %Ens preparem pel següent paquet
768 data = data(pos(1)+length(FN_Rm)+system.param.mod.mostres+1:end);
769 system.param.numpaquet = system.param.numpaquet + 1;
770 system.param.numpaquetmod = system.param.numpaquetmod + 1;
771
772 end

```

Figura 38. Cálculo BER QAM Demodtrama.m

3.3.6 MassiveDemodTrama.m

El código MassiveDemodTrama.m es el código encargado de demodular todas las tramas situadas entre los días indicados en el código. Este código ejecuta DemodTrama.m con diferentes ficheros de entrada, una vez demodulados los ficheros, DemodTrama.m retorna todos los datos referentes a la trama demodulada, estos datos se guardan en una variable común llamada TheMatrix. En esta variable se guardan los datos resultantes de todas las tramas.

En la Figura 39 podemos ver el inicio del código donde se deben introducir los datos referentes al día, mes y año de inicio con las horas y minutos en el que se quiere que se empiecen a demodular los ficheros. Por otra parte, se debe introducir los datos correspondientes a la fecha en la que se debe parar de demodular ficheros. Finalmente se debe configurar las variables de uso internas para saber qué tipo de ficheros se van a demodular. Según la variable MODE, si vale '1' se demodularan ficheros en los que hay cambios de potencias de transmisión. Por otra parte, si vale '0' el código demodulara los ficheros teniendo en cuenta que sus anchos de banda varían. Por otro lado, en el caso que MODE este a '1', la variable Rm contiene la información del remuestreo que se ha efectuado en los ficheros para configurar su ancho de banda. Finalmente encontramos las variables PBPN y SBPN las cuales nos permitirán configurar los filtros digitales que se utilizarán en las tramas de acuerdo a sus anchos de banda. En este caso, se pueden configurar introduciendo la frecuencia de los filtros, PBPN para modificar la frecuencia de corte del filtro y la frecuencia de la banda de rechazo del filtro.

```
30 %Inici de demodulacio
31 -   yeari = 2019;
32 -   monthi = 2;
33 -   dayi =10;
34 -   houri =0;
35 -   mini = 5;
36
37 %Fi de demodulacio
38 -   yearf = 2019;
39 -   monthf = 2;
40 -   dayf = 23;
41 -   hourf = 23;
42 -   minf = 59;
43
44 %Variables internes
45 -   pointer = 1;
46 -   year = yeari;
47 -   month = monthi;
48 -   day = dayi;
49 -   hour = houri;
50 -   min = mini;
51 -   m = 1;
52 -   M=2;
53 -   MODE = 0; % 0->POWER STUDY 1->BW STUDY
54 -   Rm = 42;
55 -   PBPN = 7000;
56 -   SBPN = 7500;
```

Figura 39. Parámetros MassiveDemodTrama.m

Tal y como podemos observar en la Figura 40, podemos observar el bucle general del código en el que se ejecutan todas las fechas que se han configurado. Al inicio del bucle podemos ver cómo según las variables de fecha se cambian los nombres de los ficheros que se van a demodular para que estén de acorde a la fecha. Tal y como podemos observar también podemos ver cómo según el minuto que corresponde demodular se varía la variable del remuestreo.

```

59 - try
60 -     while month < monthf + 1
61 -         while day < dayf + 1
62 -             while hour < hourf + 1
63 -                 while min < minf + 1
64 -
65 -                     Path = sprintf('D:/Documentos/Dades Antartida/GlaciAr_Rocoso/%d-%d-%d-%d.txt', year, ...
66 -                                     month, day, hour, min);
67 -                     X = ['Open file ', Path];
68 -                     disp(X)
69 -                     if(MODE == 1)
70 -                         if(floor(min/10)==0)
71 -                             Rm = 20;
72 -                             PBPn = 7000;
73 -                             SBPN = 7500;
74 -                         end
75 -                         if(floor(min/10)==1)
76 -                             Rm = 10;
77 -                             PBPn = 7000;
78 -                             SBPN = 7500;
79 -                         end
80 -                         if(floor(min/10)==2)
81 -                             Rm = 6;
82 -                             PBPn = 9000;
83 -                             SBPN = 9500;
84 -                         end
85 -                         if(floor(min/10)==3)

```

Figura 40. Bucle general y ajuste de parámetros MassiveDemodTrama.m

En la Figura 41 podemos ver como se procede a llamar la función DemodTrama.m con todos los parámetros configurados automáticamente. Una vez demodulada a posteriori de la función podemos ver la variable TheMatrix donde se guardarán en diferentes posiciones los diferentes arrays de datos extraídos de la función DemodTrama.m y a consecuencia del fichero demodulado.

```

92 - try
93 -     system = DemodTrama (Path, M, Rm, PBPn, SBPN) ;
94 -
95 -     %Gestio de dades
96 -     TheMatrix(pointer:pointer+system.param.numpaquetmod, 1) = year;
97 -     TheMatrix(pointer:pointer+system.param.numpaquetmod, 2) = month;
98 -     TheMatrix(pointer:pointer+system.param.numpaquetmod, 3) = day;
99 -     TheMatrix(pointer:pointer+system.param.numpaquetmod, 4) = hour;
100 -    TheMatrix(pointer:pointer+system.param.numpaquetmod, 5) = min;
101 -    TheMatrix(pointer:pointer+system.param.numpaquetmod, 6) = 0:system.param.numpaquetmod;
102 -    TheMatrix(pointer:pointer+system.param.numpaquetmod, 7) = M;
103 -    TheMatrix(pointer+3:pointer+system.param.numpaquetmod, 8) = 1;
104 -    TheMatrix(pointer+1:3:pointer+system.param.numpaquetmod, 8) = 2;
105 -    TheMatrix(pointer+2:3:pointer+system.param.numpaquetmod, 8) = 3;
106 -    TheMatrix(pointer:pointer+system.param.numpaquetmod, 9) = system.param.Fm/system.param.mod.Rm;
107 -    TheMatrix(pointer:pointer+system.param.numpaquetmod, 10) = system.param.Power+48.5;
108 -    TheMatrix(pointer:pointer+system.param.numpaquetmod, 11) = system.param.PotRx;
109 -    TheMatrix(pointer:pointer+system.param.numpaquetmod, 12) = system.param.snrr;
110 -    TheMatrix(pointer:pointer+system.param.numpaquetmod, 13) = system.param.EbNo';
111 -    TheMatrix(pointer:pointer+system.param.numpaquetmod, 14) = system.param.multipathkm;
112 -    TheMatrix(pointer:pointer+system.param.numpaquetmod, 15) = system.param.snrmultipath;
113 -    TheMatrix(pointer:pointer+system.param.numpaquetmod, 16) = system.param.Cat;
114 -    TheMatrix(pointer:pointer+system.param.numpaquetmod, 17) = system.param.BER;
115 -    TheMatrix(pointer:pointer+system.param.numpaquetmod, 18) = system.param.marca;
116 -    TheMatrix(pointer:pointer+system.param.numpaquetmod, 19) = system.param.piccorr';
117 -    TheMatrix(pointer:pointer+system.param.numpaquetmod, 20) = system.param.dopplerspread';

```

Figura 41. Ejecución Demodtrama y guardado de datos MassiveDemodTrama.m

Una vez finalizado el guardado de todos los datos extraídos del fichero demodulado, se actualizarán todas las variables con fechas correspondientes para prepararse a demodular el siguiente fichero y actualizar todas las variables de filtros y remuestreo. Se ha podido observar, que muchas partes del fichero están ejecutadas utilizando un try. La razón por la que se utiliza es para evitar en el caso que una parte de la demodulación falle y cause un error, antes de cerrar la ejecución de todo el código, pueda guardar la variable TheMatrix y así no perderse.

3.3.7 MassiveGrafiques.m

Finalmente, el último código utilizado en el postprocesado de todos los test efectuados es MassiveGrafiques.m. Este código es el encargado de cargar los datos de la variable TheMatrix donde se guardan todos los datos pertenecientes a todas las demodulaciones realizadas en un periodo de tiempo. Una vez cargados los datos gestiona y procesa los datos con tal que estos puedan resultar en las gráficas a través de las cuales se analizaran los datos y se llevaran las conclusiones referentes a cada modulación.

En la Figura 42, podemos ver el proceso en el que se ordenan los diferentes datos en diversas variables a través de donde se mostraran en plots. Podemos ver en el bucle que se diferencian los distintos parámetros que pueda tener las distintas partes de una trama: orden de modulación, modulación, minuto y por lo tanto potencia de transmisión o ancho de banda.

```

27 %CDF
28 for min = 1:59
29     for mod_type = 1:3
30         k = 1;
31         for i = 1:length(TheMatrix(:,1))
32             if(TheMatrix(i,5)==min)
33                 if(TheMatrix(i,8)==mod_type)
34                     if(isnan(TheMatrix(i,17))==0)
35                         %if((isnan(TheMatrix(i,17))==0) && (TheMatrix(i,4)>=12&&TheMatrix(i,4)<=19))
36                             cdf_vector(k) = TheMatrix(i,17);
37                             ber_time_vector(k) = TheMatrix(i,17);
38                             if(ber_time_vector(k)==0)
39                                 %ber_time_vector(k) = 0.00000001;
40                             end
41                             ebno_time_vector(k) = 10^(TheMatrix(i,13)/10);
42                             hour_vector_1(k) = TheMatrix(i,4);
43                             hour_vector_2(k) = TheMatrix(i,4);
44                             k = k + 1;
45                             %Sumador de bits
46                             Bits = log2(TheMatrix(i,7)) * (10500 / (100000 / TheMatrix(i,9))) + Bits;
47                         end
48                     end
49                 end
50             end

```

Figura 42. Clasificación datos para CDF MassiveGrafiques.m

Como podemos ver en la Figura 43, a través de distintos bucles, se clasifican las distintas modulaciones y se guarda el BER y el EbNo en distintas arrays con el nombre que corresponde en base al punto en el que se encuentra del bucle. Estas arrays permiten posteriormente a través de la función cdfplot o haciendo la media de los BER por un EbNo dado, obtener las gráficas CDF o las gráficas BER-EbNo que veremos posteriormente en los siguientes apartados.

```

52 %BER vs Time
53 [hour_vector_1,B] = sort(hour_vector_1);
54 ber_time_vector = ber_time_vector(B);
55 [hour_vector_1,~,idx] = unique(hour_vector_1);
56 ber_time_vector = accumarray(idx,ber_time_vector,[],@mean,1);
57 paq_vector = accumarray(idx,1);
58
59 %EbNo vs Time
60 [hour_vector_2,B] = sort(hour_vector_2);
61 ebno_time_vector = ebno_time_vector(B);
62 [hour_vector_2,~,idx] = unique(hour_vector_2);
63 ebno_time_vector = accumarray(idx,ebno_time_vector,[],@mean,1);
64 ebno_time_vector = 10*log10(ebno_time_vector);
65
66 cdf_vector = sort(cdf_vector);
67 X = ['min_',num2str(min),'_mod_type_',num2str(mod_type)];
68 CDF(X) = cdf_vector;
69 X = ['min_',num2str(min),'_mod_type_',num2str(mod_type)];
70 BERTime.BER(X) = ber_time_vector;
71 X = ['min_',num2str(min),'_mod_type_',num2str(mod_type)];
72 BERTime.hour.(X) = hour_vector_1;
73 X = ['min_',num2str(min),'_mod_type_',num2str(mod_type)];
74 BERTime.paq.(X) = paq_vector;
75 X = ['min_',num2str(min),'_mod_type_',num2str(mod_type)];
76 EbNoTime.hour.(X) = hour_vector_2;
77 X = ['min_',num2str(min),'_mod_type_',num2str(mod_type)];
78 EbNoTime.EbNo.(X) = ebno_time_vector;

```

Figura 43. Clasificación datos para BER-EbNo-Time MassiveGrafiques.m

En el código se repite el mismo proceso de clasificación una segunda vez. El hecho que se repita esta clasificación es para la creación de otros tipos de gráficas en las que se requiere otro tipo de clasificación de los datos. Una vez acabada la clasificación se tratarán las variables según la gráfica que se quiera obtener. Tal y como podemos ver en la Figura 44, en el código aparecen después de la clasificación códigos para generar la mayoría de las gráficas genéricas, pero en caso que se quiera generar una nueva, el usuario deberá tratar los variables según le convengan o iniciar otros tipos de clasificación de datos. En el caso de la figura, el código nos permite dibujar la gráfica correspondiente a la CDF de las modulaciones 2PSK, 2FSK y 2QAM.

```

129 %GRAFICA CDF
130 x1 = cdfplot(CDF.min_17_mod_type_1);
131 hold on;
132 x2 = cdfplot(CDF.min_17_mod_type_2);
133 hold on;
134 x3 = cdfplot(CDF.min_17_mod_type_3);
135 xlabel('Xo');
136 ylabel('P(BER<Xo)');
137 legend('2PSK','2FSK','2QAM');
138 set(gca,'Yscale','log');
139 set(gca,'Xscale','log');
140 set(gca,'FontSize',18);
141 set(x1,'LineWidth',2);
142 set(x2,'LineWidth',2);
143 set(x3,'LineWidth',2);
144 xlim([0.0001 1]);
145 ylim([0.8 1]);
146 hold off;
147
148 %GRAFICA CDF POR POTENCIAS
149 x1 = cdfplot(CDF.min_5_mod_type_1);
150 hold on;
151 x2 = cdfplot(CDF.min_15_mod_type_1);
152 hold on;
153 x3 = cdfplot(CDF.min_25_mod_type_1);
154 hold on;
155 x4 = cdfplot(CDF.min_35_mod_type_1);
156 hold on;

```

Figura 44. Generación de gráficas MassiveGrafiques.m

4 Resultados

4.1 Introducción a los resultados

En este apartado de este trabajo final de máster veremos los resultados obtenidos en este estudio. A continuación, se mostrarán los resultados más relevantes que se han obtenido a través de todas las transmisiones efectuadas en términos de modulaciones, anchos de banda y potencias de transmisión. Todos los datos son pertenecientes a los datos obtenidos durante la Campaña Antártica Española 2018-2019 en Livingston Island, concretamente entre el 1 de febrero y el 2 de marzo de 2019.

4.2 BER-EbNo

Primeramente, en el estudio se han analizado los resultados de BER y EbNo de las tramas recibidas. En la Figura 45 podemos ver la comparación de las modulaciones PSK, FSK y QAM de segundo orden. Tal y como podemos ver, el BER mínimo viene dado por el número de tramas recibidas y de bits demodulados en total, de tal manera que no podemos asegurar ningún BER que este por debajo de 10^{-8} . Tal y como podemos ver en la gráfica resultante, las modulaciones PSK y QAM son más robustas que la FSK por un mismo EbNo. Por otro lado, hay que destacar que en este caso las modulaciones 2PSK y 2 QAM no son iguales. En este caso se ha querido demostrar que el hecho que la 2QAM utilice componentes reales e imaginarias para la modulación de sus símbolos la hace menos robusta que la 2PSK, la cual utiliza únicamente componentes reales para la modulación de sus símbolos. El mismo hecho se podría aplicar en el caso que sus símbolos fueran modulados únicamente con componentes imaginarias.

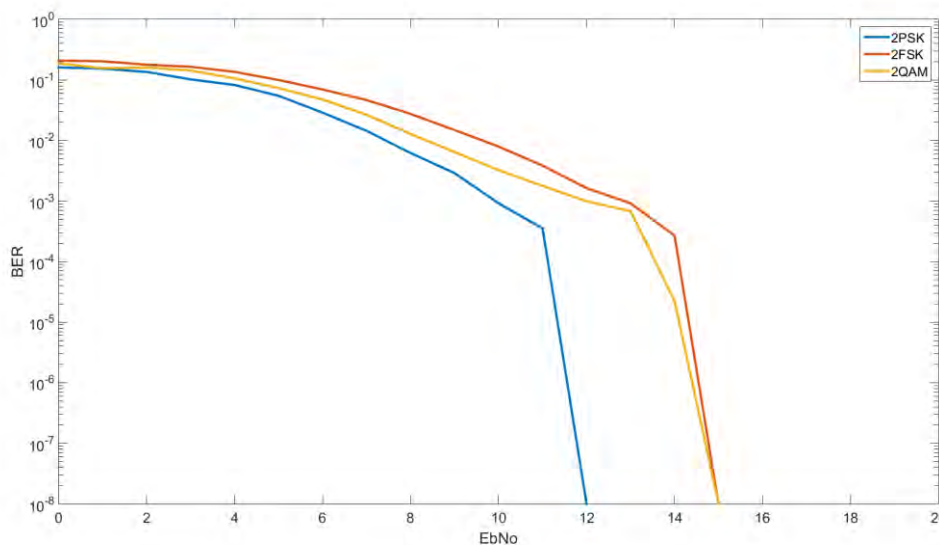


Figura 45. Gráfica BER-EbNo orden 2 [29]

En la Figura 46 podemos ver la comparación de las modulaciones PSK, FSK y QAM de cuarto orden. En este caso podemos apreciar que la QAM y la PSK son más robustas que la FSK. Por otra parte, podemos observar que la PSK y la QAM en este caso son exactamente iguales. El hecho de esta igualdad se debe a que han sido transmitidas de igual manera. Sus símbolos en este caso han sido modulados unos en el eje real y otros en el eje imaginario. Teniendo en cuenta que al doblar el bitrate respecto la modulación de orden dos el E_b/N_0 disminuye 2 dB, podríamos demostrar que la modulación de los ejes reales e imaginarios los hace totalmente independientes como si de dos modulaciones de orden 2 se tratara siendo igual de robusto.

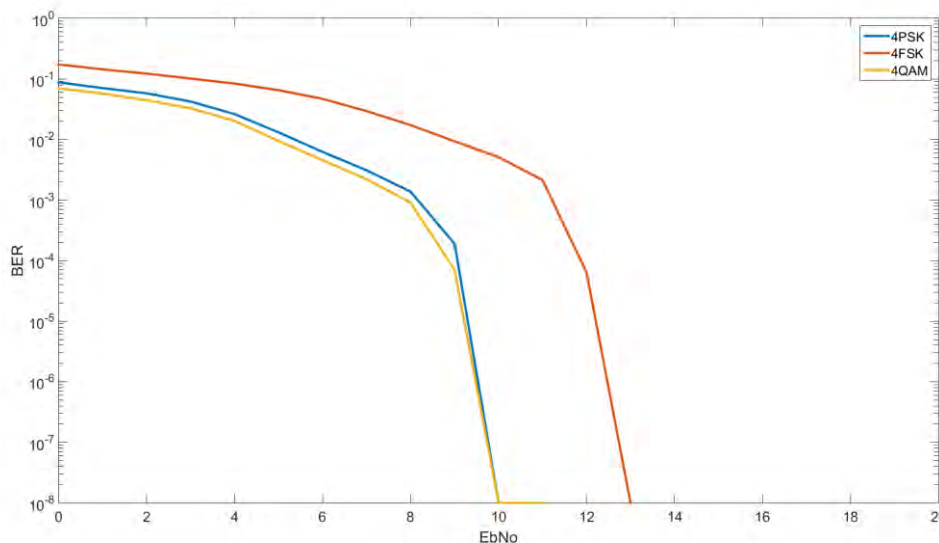


Figura 46. Gráfica BER-EbNo orden 4 [29]

Las modulaciones de un orden superior a cuatro, los símbolos recibidos han sido demasiado bajos a nivel de E_b/N_0 para ser demodulados correctamente. En el siguiente apartado podemos ver las gráficas que nos dan los resultados correspondientes.

4.3 E_b/N_0 -Time

En este apartado analizaremos las razones que han hecho que las modulaciones de un orden superior a cuatro no hayan podido ser demoduladas correctamente. En la Figura 47 y Figura 48 podemos ver las gráficas correspondientes al E_b/N_0 medio recibido de todas las tramas con potencias de transmisión de 24 W y ordenes de modulación 8 y 32. En las gráficas cabe destacar que las horas van desde las 12:00 UTC a las 19:00 UTC dado que el resto de horas la ionosfera no permitía la reflexión de las señales transmitidas. A través de estas gráficas podemos apreciar que la ventana de horas disponibles para la transmisión a la frecuencia de 5 MHz era de 7 horas cada día. Tal y como podemos observar en las gráficas, para las modulaciones de orden 8, los valores de E_b/N_0 varían entre 3 dB y 5 dB según la modulación y en el caso de las modulaciones de orden 32 varían entre 0 dB y 3 dB dependiendo también de la modulación. Los resultados obtenidos nos demuestran los valores bajos de E_b/N_0 en las demodulaciones realizadas, dando

a entender la incapacidad para demodular correctamente los datos y obtener resultados de gráficas BER-EbNo con EbNo altos.

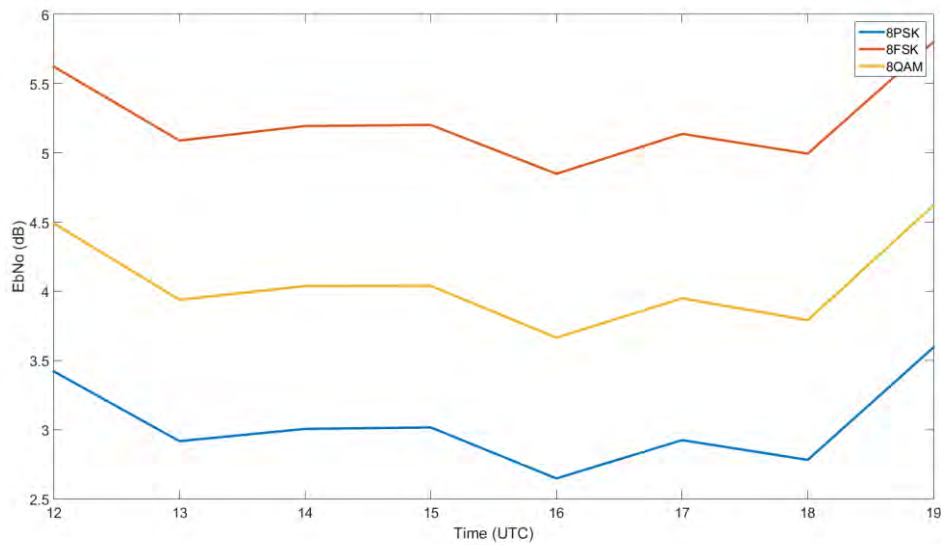


Figura 47. Gráfica EbNo-Time orden 8 [29]

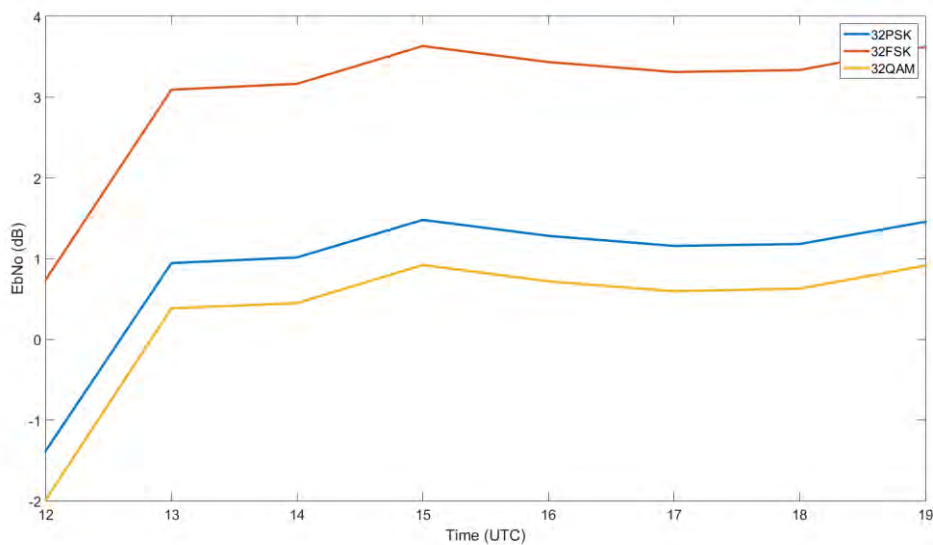


Figura 48. Gráfica EbNo-Time orden 32 [29]

4.4 CDF según potencias transmitidas

Tras el análisis de los resultados anteriores podemos entender la razón por la que los resultados de modulaciones de orden superiores a 4 han sido malos. Por otra parte, se ha querido analizar los resultados a través de CDF (Cumulative Distribution Function). Esta distribución nos indica la

probabilidad de recibir una trama con un BER inferior a un valor dado. En la Figura 49 podemos ver la CDF correspondiente a las modulaciones PSK, FSK y QAM de orden 2 con una potencia de transmisión de 24W. Podemos observar que la PSK y la QAM tienen una probabilidad de 95.2% y 95.4% respectivamente para obtener un BER de 10^{-8} , que es el mínimo que podemos asegurar en este estudio. Para el mismo BER, en el caso de la FSK la probabilidad es de 92.8% siendo menor que en el caso de la PSK y la QAM.

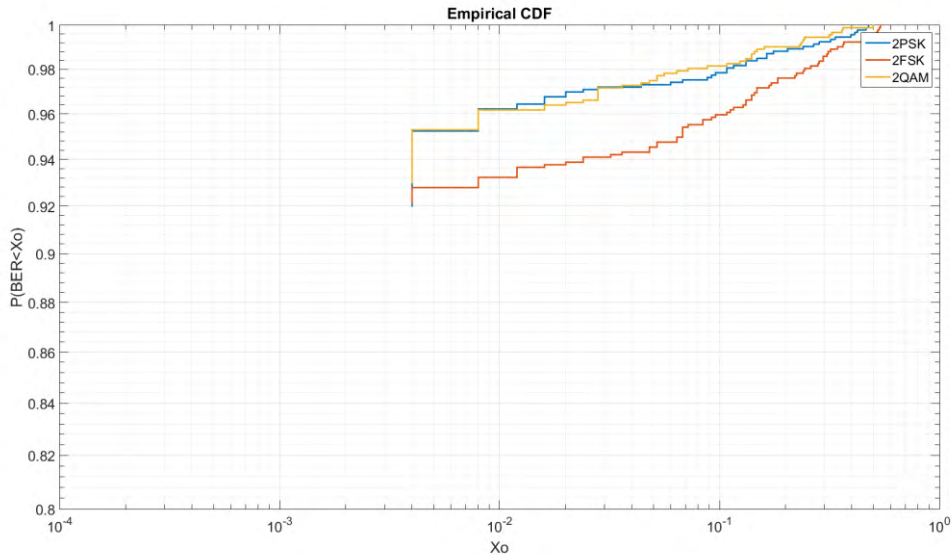


Figura 49. Gráfica CDF 24 W orden 2 [29]

En la Figura 50 podemos ver la CDF de las modulaciones PSK, FSK y QAM de orden 2 para una potencia de transmisión de 3 W. En este caso la diferencia entre las diferentes modulaciones no es notable. Podemos apreciar que las modulaciones PSK y QAM tienen una probabilidad de 94.3% y 94.4% respectivamente de obtener un BER de 10^{-8} . A diferencia de la FSK, está tiene una probabilidad del 93.8% de obtener un BER de 10^{-8} . En este caso no hay mucha diferencia entre las tres modulaciones.

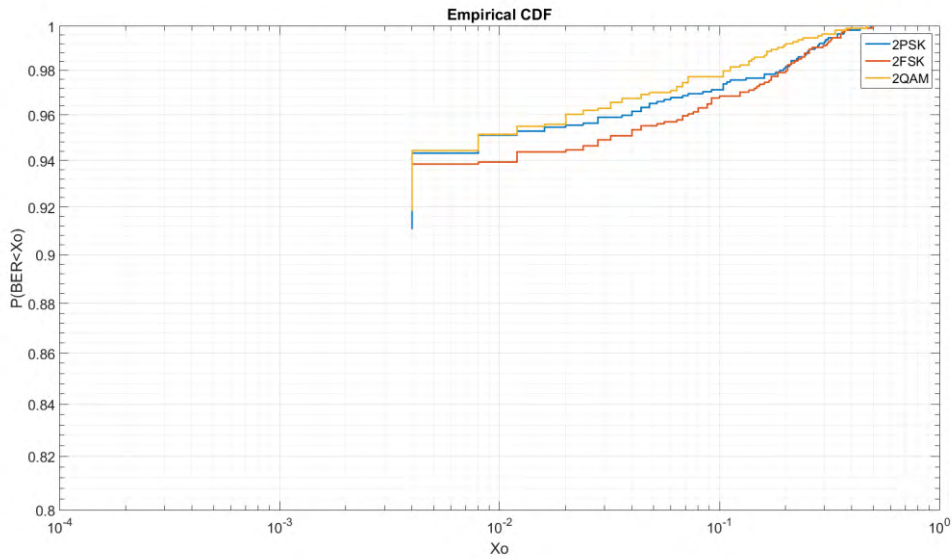


Figura 50. Gráfica CDF 3W orden 2 [29]

Por otro lado, podemos observar las gráficas CDF correspondientes a las transmisiones efectuadas con órdenes de modulación 4 y potencias de transmisión de 24 W. En la Figura 51 podemos observar que la probabilidad de la 4FSK disminuye al 77.3%. La 4PSK y la 4QAM tienen un comportamiento similar con probabilidades de 91.1% y 92.8% respectivamente.

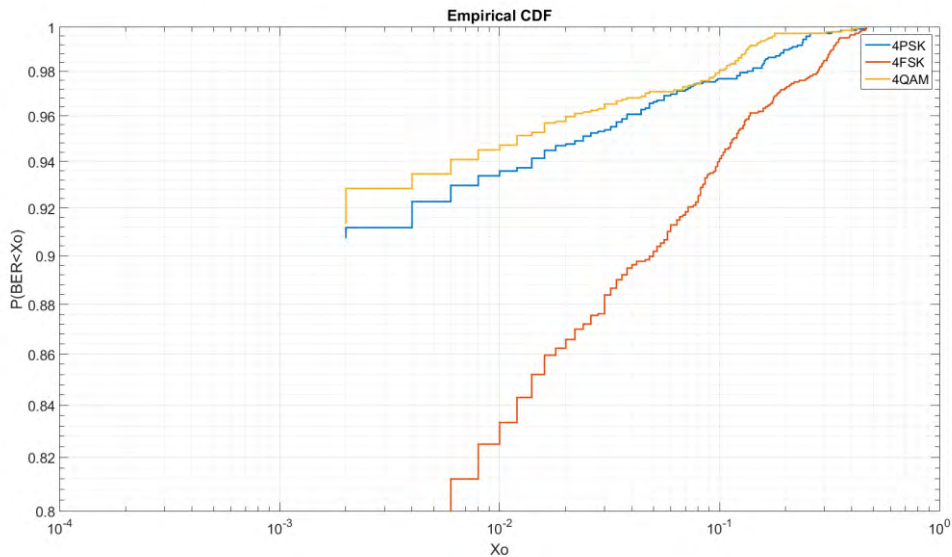


Figura 51. Gráfica CDF 24W orden 4 [29]

En la Figura 52 podemos observar en este caso, la gráfica CDF correspondiente a las modulaciones con órdenes de modulación 4 y potencias de transmisión de 2 W. En este caso podemos ver que la PSK y la QAM superan las capacidades de la FSK con probabilidades de 90.1%

y 91.1% respectivamente. En este caso la probabilidad de obtener BER inferiores a 0.004 disminuyen hasta el 80%.

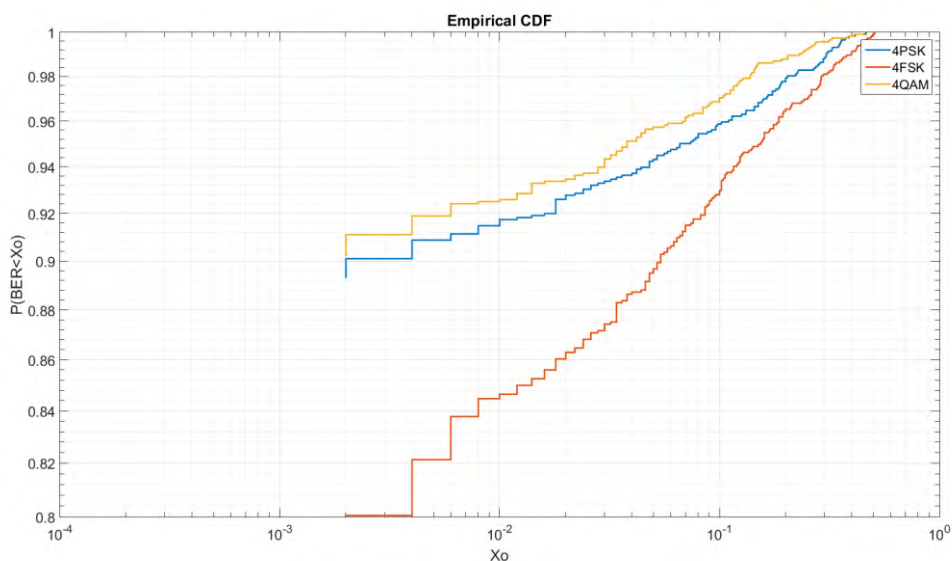


Figura 52. Gráfica CDF 3W orden 4 [29]

4.5 CDF según anchos de banda transmitidos

En los resultados anteriores se han mostrado las gráficas correspondientes a los resultados de los test realizados con anchos de banda estándares de 2.3 KHz. En este apartado se pretende mostrar los resultados de todos aquellos test que se han realizado fuera de los anchos de banda estándares. Para una modulación dada, el ancho de banda es directamente proporcional y como mínimo la velocidad de bitrate de está, de tal manera que podemos alcanzar bitrates de hasta 18.4 Kbps con un ancho de banda de 10 KHz.

En la Figura 53 podemos observar la gráfica resultante de la transmisión de modulaciones de orden 4 con un ancho de banda de 5 KHz y una potencia de transmisión de 24W. Tal y como podemos ver en la gráfica, la probabilidad de obtener un BER menor a 0.001 es del 73.7% tanto para la QAM como para la PSK, a diferencia de la FSK que obtiene un valor del 63.8%.

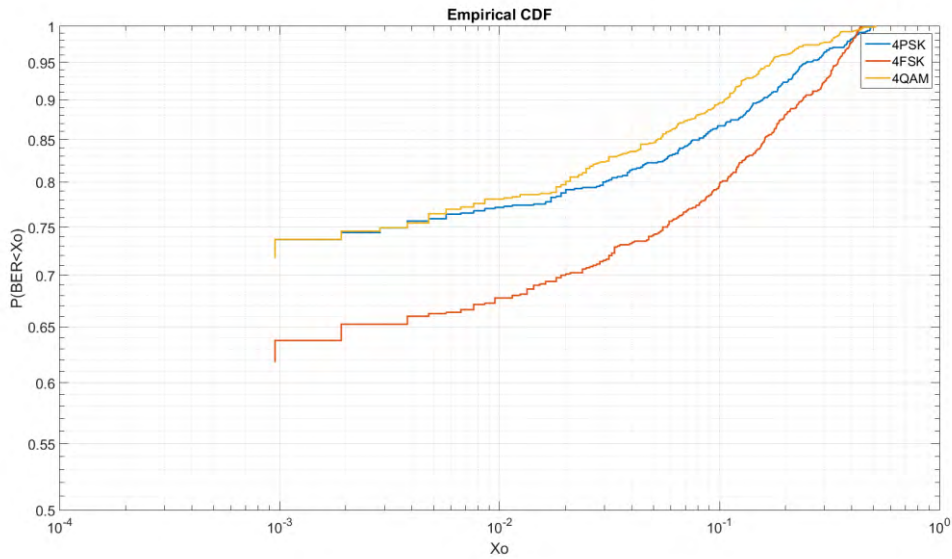


Figura 53. Gráfica CDF 5 KHz 24 W orden 4 [29]

Los conjuntos de test se han realizado para modulaciones con un ancho de banda de 10 KHz, ordenes de modulación 4 y potencias de transmisión de 24 W. En la Figura 54, tal y como podemos ver, la probabilidad de obtener un BER menor a 0.001 es del 66% para la PSK y la QAM, mientras que para la FSK la probabilidad es del 43.9%.

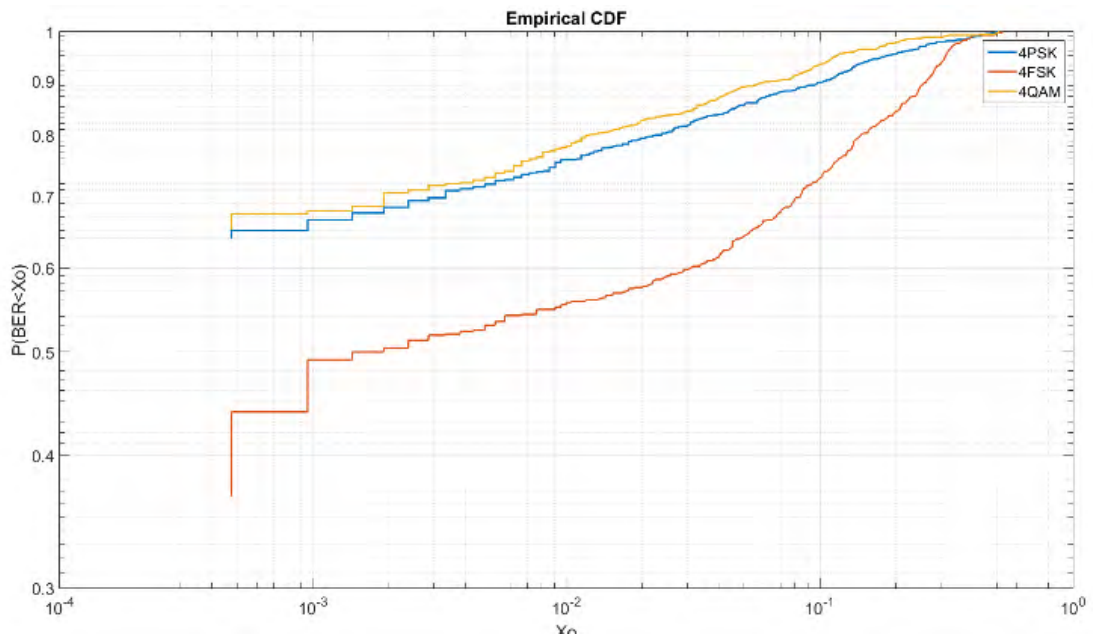


Figura 54. Gráfica CDF 10 KHz 3 W orden 4 [29]

5 Costes

En este apartado veremos los costes generales que ha conllevado el estudio realizado en este trabajo final de máster. Primeramente, veremos el coste temporal del estudio a través de un diagrama de Gantt y el cómputo total de horas que se han requerido. Por otra parte, veremos el coste económico que requiere llevar a cabo este estudio a nivel de hardware y software, aunque muchos elementos no se han tenido que comprar dado que han tenido otra utilidad en el proyecto antártico.

5.1 Temporal

El estudio realizado en este trabajo final de máster se ha llevado a lo largo de un año, aunque se ha estado trabajando en el estudio un total de siete meses aproximadamente. El trabajo realizado se ha dividido en varias etapas, donde primeramente encontraríamos el planteamiento de toda la implementación que se ha tenido que hacer, un estudio del canal y el diseño de la trama de acuerdo al canal. Esta primera etapa ha conllevado un total de 40 horas aproximadamente. Posteriormente, se ha llevado a cabo la implementación de todo el software necesario para la creación de todas aquellas tramas que se han transmitido, el software enfocado a la demodulación de todas las tramas recibidas, el software de gestión para la demodulación masiva de ficheros y finalmente el software que permite la clasificación de todos los datos extraídos y de su representación en gráficas. Esta etapa de implementación ha conllevado un total de 640 horas aproximadamente. Posteriormente a esta etapa, nos encontramos con el análisis de todos los resultados. En esta etapa se ha llevado a cabo a través del software implementado, la creación de todas las gráficas referentes a los datos extraídos de los test realizados. Por otra parte, se ha realizado un análisis de las gráficas para extraer conclusiones del estudio realizado. La etapa de análisis de resultados a tenido un coste temporal de 60 horas aproximadamente. Finalmente, la última etapa, la creación de toda la documentación pertinente al estudio en el trabajo final de máster ha llevado a cabo un total de 84 horas. En conjunto el coste temporal que se ha requerido para la realización de este trabajo final de máster es de un total de 824 horas. En la Figura 55 se puede observar el diagrama de Gantt en el que están desglosados en semanas el tiempo dedicado al trabajo final de máster.

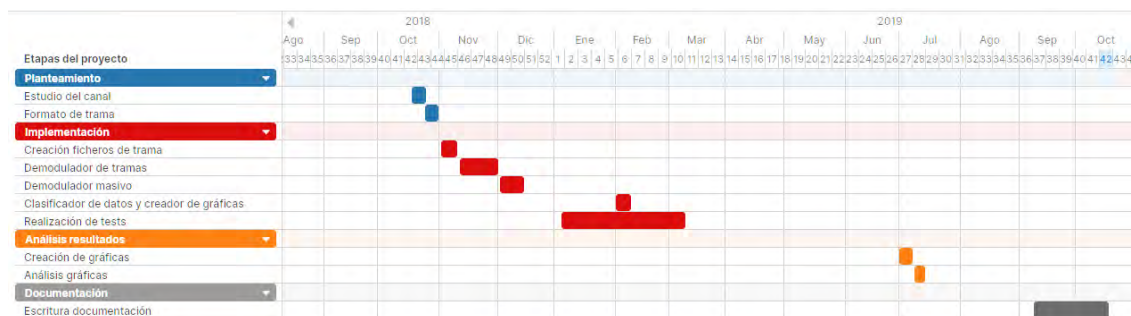


Figura 55. Diagrama de Gantt del trabajo final de máster

5.2 Económico

Desde el punto de vista económico el estudio llevado a cabo a tenido un total de costes muy bajo dado que la mayoría de los elementos ya estaban presentes del estudio realizado en el trabajo de final de carrera o del proyecto antártico. En la Tabla 3 podemos observar el desglose de todos los elementos implicados en el proyecto. En la tabla, todos aquellos precios que se encuentran en cursiva significa que ya estaban presentes en el proyecto antártico.

Componente hardware	Precio (€)
Red Pitaya 125-14 (Starter Kit)	<i>319 €</i>
Raspberry Pi 3	<i>39.99 €</i>
Plataforma Teensy	<i>60 €</i>
Adafruit Ultimate GPS Breakout v3	<i>53,02 €</i>
Mini-Circuits ZHL-32A	<i>203,48 €</i>
PCB filtro pasabanda y componentes	<i>60 €</i>
Bonn BLWA 0103-250	<i>5.000 €</i>
Antena y balun	<i>60 €</i>
Cables radiofrecuencia asociados	<i>100 €</i>
USB 128 GB	<i>26.39 €</i>
Matlab	<i>69 €</i>
Coste total nodo	<i>6.073,07 €</i>
Coste total dos nodos	<i>12.146,14 €</i>
Coste real total	0 €

Tabla 3. Costes económicos del estudio

6 Conclusiones

En este trabajo final de máster a través de todos los test efectuados, se han explorado las características que pueden ofrecer las principales modulaciones de banda estrecha con diferentes órdenes de modulación, anchos de banda y potencias de transmisión para su aplicación en sensores remotos.

Para sensores de muy bajo consumo, en este entorno a través de los resultados obtenidos podemos ver que la opción más recomendable es el uso de anchos de banda de 2.3 KHz, potencias de transmisión de 3W y modulaciones 4QAM y 4PSK. A través de estas podemos obtener un bitrate de 4.6 Kbps con una probabilidad del 90% en obtener un BER 0 en los paquetes recibidos sin tener en cuenta el uso de un código corrector de errores.

En el caso que el escenario este enfocado a sensores con un consumo más elevado y por lo tanto que se pueda no tener tanto en cuenta los aspectos energéticos, la mejor opción recomendada sería el uso de una modulación 2 FSK con una potencia de transmisión de 24 W y un ancho de banda estándar de 2.3 KHz. En este caso se recomienda el uso de la FSK dado que se pueden utilizar amplificadores con un mayor consumo, aunque dado que la FSK no requiere de cambios de amplitud en sus símbolos, en este caso la eficiencia energética sería muy alta. En el caso que se requieran bitrates más altos, sería recomendable aumentar el ancho de banda a 5 KHz. En ese caso se podría también utilizar una 4QAM la cual tendría una probabilidad del 73.7% de obtener un bitrate menor de 0.001 y un bitrate de 9.2 Kbps.

Por otra parte, para el estudio que se ha llevado a cabo se ha definido una trama de transmisiones NVIS en la que está incluida un bloque para la corrección de doppler shift, sincronización de datos y cada uno de los bloques es de un tamaño concreto para estar dentro de los tiempos de coherencia del canal.

Por último, cabe mencionar que el estudio llevado a cabo en este trabajo final de máster ha sido utilizado para el escrito de un artículo científico el cual actualmente está bajo revisión menores en la revista Radio Science [32] con un factor de impacto Q3.

Finalmente, todos los test realizados demuestran la factibilidad de una red de sensores remotos conectados a través de NVIS situados en distancias de hasta 250 km alejados del nodo central. Esta red no necesitaría el uso de operadores de satélite siendo una alternativa al satélite en zonas remotas y con orografías complicadas.

7 Líneas de futuro

Tras la realización de este estudio como trabajo final de máster se pueden contemplar varias líneas de futuro. Estas líneas de futuro están previstas para la publicación de otro artículo y la ampliación del software para que permita aumentar la diversidad de los tipos de test.

En primer lugar, a través de los test realizados, en este estudio nos hemos centrado en averiguar los límites de cada modulación en transmisiones NVIS de bajas potencias con diferentes anchos de banda bajo los efectos del canal ionosférico. A través de estas transmisiones, cada trama contiene un tono de 600 Hz y una secuencia PN, elementos que nos pueden ayudar a estudiar el canal. Como línea de futuro, a través de los test realizados, concretamente a través de los tonos y secuencias PN transmitidas, se pretende hacer un estudio del canal ionosférico con el fin de extraer información de él. Este estudio del canal ionosférico, nos permitirá extraer conclusiones que serán publicadas como un artículo para una revista de divulgación científica con un factor de impacto mínimo.

Por otra parte, el estudio realizado en este trabajo final de máster, solo se transmite a una sola frecuencia, a 5 MHz. En un futuro estudio se pretende cambiar la antena de banda estrecha de $\lambda/2$ por una antena de banda ancha que pueda transmitir desde 1 MHz hasta 30 MHz con el fin de realizar los test en diferentes frecuencias. Por otra parte, este estudio también permitiría de la misma manera que el estudio realizado en este trabajo de máster, el estudio del canal ionosférico en diferentes frecuencias de uso. Hasta el momento, no se han realizado test con diferentes frecuencias, dado que el banco de pruebas actual ha sido muy amplio y no habría suficientes datos para cada caso concreto de estudio.

8 Referencias

- [1] R. M. Alsina-Pagès, M. Hervás, D. Altadill, J. Calduch, and E. Blanch, "Vertical and oblique ionospheric soundings performance comparison over the 12,760 km transequatorial HF link between Antarctica and Spain," *Radio Sci.*, vol. 52, no. 4, pp. 498–510, Apr. 2017.
- [2] A. G. Ads *et al.*, "Vertical and oblique ionospheric soundings over the long haul HF link between Antarctica and Spain," *Radio Sci.*, vol. 50, no. 9, pp. 916–930, Sep. 2015.
- [3] C. Vilella, D. Miralles, and J. L. Pijoan, "An Antarctica-to-Spain HF ionospheric radio link: Sounding results," *Radio Sci.*, vol. 43, no. 4, p. n/a-n/a, Aug. 2008.
- [4] A. G. Ads *et al.*, "A comprehensive sounding of the ionospheric HF radio link from Antarctica to Spain," *Radio Sci.*, vol. 48, pp. 1–12, 2012.
- [5] M. Hervás, R. Alsina-Pagès, F. Orga, D. Altadill, J. Pijoan, and D. Badia, "Narrowband and Wideband Channel Sounding of an Antarctica to Spain Ionospheric Radio Link," *Remote Sens.*, vol. 7, no. 9, pp. 11712–11730, Sep. 2015.
- [6] M. Hervas, J. Pijoan, R. Alsina-Pagès, M. Salvador, and D. Altadill, "CHANNEL SOUNDING AND POLARIZATION DIVERSITY FOR THE NVIS CHANNEL," 2013.
- [7] C. Vilella *et al.*, "Vertical and oblique ionospheric soundings over a very long multihop HF radio link from polar to midlatitudes: Results and relationships," *Radio Sci.*, vol. 44, no. 2, p. n/a-n/a, Apr. 2009.
- [8] M. Hervás, J. L. Pijoan, R. M. Alsina-Pagès, M. Salvador, and D. Altadill, "Channel sounding and polarization diversity for the NVIS channel," *Proc. Nord. HF Conf.*, no. August, 2013.
- [9] P. Bergadà Caramés, "Oblique Sounding and HF Communication Techniques for Very Long Haul Ionospheric Links," Universitat Ramon Llull, 2015.
- [10] P. Bergadà *et al.*, "Digital transmission techniques for a long haul HF link: DSSS versus OFDM," *Radio Sci.*, vol. 49, no. 7, pp. 518–530, Jul. 2014.
- [11] M. Hervás, R. M. Alsina-Pagès, J. L. Pijoan, M. Salvador, and D. Badia, "Advanced modulation schemes for an Antarctic Long Haul HF Link Performance comparison between SC-FDE, OFDMA and SC-FDMA in a hostile environment," *Telecommun. Syst.*, vol. 62, no. 4, pp. 757–770, Aug. 2016.
- [12] J. Pijoan, D. Altadill, J. Torta, R. Alsina-Pagès, S. Marsal, and D. Badia, "Remote Geophysical Observatory in Antarctica with HF Data Transmission: A Review," *Remote Sens.*, vol. 6, no. 8, pp. 7233–7259, Aug. 2014.
- [13] R. M. Alsina-Pagès, M. Salvador, M. Hervás, P. Bergadà, J. L. Pijoan, and D. Badia, "Spread spectrum high performance techniques for a long haul high frequency link," *IET Commun.*, vol. 9, no. 8, pp. 1048–1053, May 2015.
- [14] "ATS - The Antarctic Treaty." [Online]. Available: <https://www.ats.aq/e/ats.htm>. [Accessed: 23-Sep-2019].
- [15] North Atlantic Treaty Organization (NATO), "STANAG 5066: The Standard for Data Applications over HF Radio," 2009.

- [16] MIL-STD-188-110B, "DEPARTMENT OF DEFENSE INTERFACE STANDARD INTEROPERABILITY AND PERFORMANCE STANDARDS FOR DATA MODEMS," 2000.
- [17] MIL-STD-188-110D, "DEPARTMENT OF DEFENSE INTERFACE STANDARD: INTEROPERABILITY AND PERFORMANCE STANDARDS FOR DATA MODEMS," 2017.
- [18] J. M. Masó, "Modulaciones para sensores remotos con tecnología NVIS," La Salle Universitat Ramon Llull, 2018.
- [19] "10sd156 Leon Guanajuato Mexico: Propagación de las ondas de radio." [Online]. Available: <http://10sd156.blogspot.com/p/propagacion-de-las-ondas-de-radio.html>. [Accessed: 27-Sep-2018].
- [20] "(G9D02) NVIS Antenna | Ham Radio School.com." [Online]. Available: <https://hamradioschool.com/g9d02-nvis-antenna/>. [Accessed: 13-Oct-2019].
- [21] "Ionosfera - Wikipedia, la enciclopedia libre." [Online]. Available: <https://es.wikipedia.org/wiki/Ionosfera>. [Accessed: 25-Sep-2018].
- [22] "Rebotando en la ionosfera | Proyecto Antártida." [Online]. Available: <http://blogs.salleurl.edu/antartida/rebotando-en-la-ionosfera/>. [Accessed: 27-Sep-2018].
- [23] "Capas de la Ionósfera." [Online]. Available: https://qsl.net/xs2pna/html_files/capas_de_la_inosfera.html. [Accessed: 13-Oct-2019].
- [24] "El ciclo solar se precipita hacia el mínimo. - Grupo Amateur de Meteorología Espacial GAME." [Online]. Available: <http://blog.meteorologiaespacial.es/2018/02/09/ciclo-solar-se-precipita-hacia-minimo/>. [Accessed: 13-Oct-2019].
- [25] "ShareTechnote." [Online]. Available: https://www.sharetechnote.com/html/Handbook_LTE_Fading.html. [Accessed: 14-Oct-2019].
- [26] "Multipath propagation - Wikipedia." [Online]. Available: https://en.wikipedia.org/wiki/Multipath_propagation. [Accessed: 14-Oct-2019].
- [27] "Pengertian Efek Doppler - Budisma." [Online]. Available: <https://budisma.net/2015/04/pengertian-efek-doppler.html>. [Accessed: 14-Oct-2019].
- [28] "Red Pitaya STEMLab board." [Online]. Available: <https://www.redpitaya.com/f130/STEMlab-board>. [Accessed: 27-Sep-2018].
- [29] J. Porte, J. M. Maso, J. L. Pijoan, and D. Badia, "Sensing system for remote areas in Antarctica," *Radio Sci.*, no. Minor Revisions, 2019.
- [30] J. Porté, "Desenvolupament d'una plataforma de baix cost per a sensors remots amb tecnologia NVIS," Universitat Ramon Llull- La Salle.
- [31] "R2015a - MATLAB & Simulink - MathWorks España." [Online]. Available: <https://es.mathworks.com/help/matlab/release-notes-R2015a.html>. [Accessed: 29-Sep-2019].
- [32] American Geophysical Union, International Union of Radio Science, United States. Environmental Science Services Administration, and U.S. National Committee of the

International Scientific Radio Union, *Radio science*. American Geophysical Union.

9 Anexo

Radio Science

Sensing system for remote areas in Antarctica

Joaquim Porte¹, Josep M. Maso¹, Joan Lluís Pijoan^{1,*}, David Badia¹

¹La Salle Ramon Llull; {joaquim.porte, josep.maso, david.badia}@salle.url.edu.

*Correspondence: joanlluis.pijoan@salle.url.edu;

Key Points:

- **IoT network communications study for small HF antennas based on low power transmission.**
- **Study of narrowband modulations varying its power transmission and bandwidth.**
- **QAM is the best modulation for low-power transmission with NVIS system on Antarctica environment**

Abstract

Abstract: Every year the number of Internet of Things (IoT) devices is grows exponentially. The current IoT technology to support the connectivity of such a huge number of devices is limited by the coverage of the base stations deployed. In case of remote areas without coverage of any operator, the use of a satellite connection is such high cost option. The only alternative option for very remote sensors are High Frequency (HF) communications with ionospheric reflection. The HF band (3-30 MHz) with Near Vertical Incidence Skywave (NVIS) allows a large coverage area (up to 250 Km) without the need of line of sight. The HF radio links usually need higher power transmissions with larger antennas supported by a mast. In this paper, we explore a new transmission scheme for low-power transmissions, which is equivalent to use small and low gain HF antennas. We study the performance of several digital modulations using different bandwidths and transmission power. The field tests have been done around the Spanish Antarctic Base at Livingston Island, to ensure the availability of the system even in polar areas where the behavior of the ionosphere is quite different from lower latitudes. However, the proposed physical layer fits well with any other remote location that requires low power data communication.

1. Introduction

Every year the number of IoT connected devices is increasing constantly. In 2019 the number of IoT devices are about 26.66 billion and it is expected to be increase exponentially up to 75.44 billion in 2025 ("Statista - Internet of Things - Number of connected devices worldwide 2015-2025," n.d.). Some radio technologies for communicating this exorbitant number of devices have emerged, however the coverage areas (offered by Sigfox, Lora, Neul, etc) are quite limited (Lauridsen et al., 2017).

In case of remote sensors which are far (more than 10 Km) from the nearest base station, the only alternatives without line of sight are satellites and HF communications. Other solutions such as repetition of the signal following the orography will solve the problem. However, this

Radio Science

solution has a high economic cost. The main drawbacks of satellite connections are the high costs of the service and the operator dependency. Moreover, the coverage area of satellites is limited at polar locations, since the geostationary satellites usually do not have line of sight and a it is required a network of polar satellites.

As an alternative to the satellite connection, a radio link in the HF band (3 MHz - 30 MHz) (Davies, 1990) can be established. HF communications do not need Line of Sight (LoS) between the transmitter and the receiver. The wave is reflected by the upper layers of the atmosphere, which are ionized by the ultraviolet radiation coming from the sun. Depending on the angle of incidence in the ionosphere, two types of communication can be achieved. In the case of long-distance communications (DX), the incidence angle is lower than 70 degrees and we can cover a distance of 3000 Km per hop. In practice, we can communicate with any part of the world using 4 or 5 hops(Vilella et al., 2009). In the case of an incidence angle between 70 degrees and 90 degrees, we refer to Near Vertical Incidence Skywave (NVIS) communications, with only one reflection achieving a radio coverage up to 250 km see in Figure 2, assuming 350km as the maximum height of the ionosphere, (Witvliet & Alsina-Pagès, 2017). In Figure 1 we can see an hypothetical scenario with the different radio technologies for remote IoT.

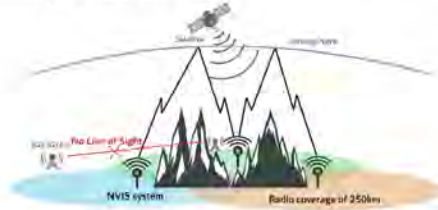


Figure 1. NVIS Technology

Antarctica is one of the most remote and isolated places in the world. Many research activities are developed in the scientific bases distributed throughout the continent among the year (AEMET, 2019). Most of the scientific studies are based on data collected from sensors. Some of them are far from the base and the downloading of the data has to be made manually. In that context, the data transmission is only possible using satellite links or HF radio communications. The services of the most common satellite operators (Iridium, Immarsat, Thuraya, etc) are usually expensive and need line of sight.

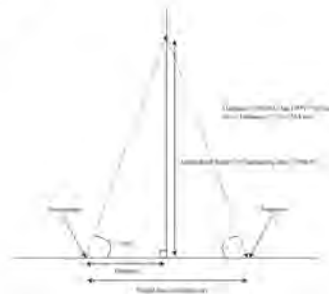


Figure 2: NVIS signal propagation coverage

If we want to use HF communications for remote sensors, there are two key factors to be taken into account: first, the power consumption of the system, to increase the autonomy of the batteries and reduce the size as much as possible. Second, the size of the antenna, since we are dealing with wavelengths up to 100 m. For instance, a typical half-wave dipole at 4 MHz can reach 30 meters and needs a mast of at least 12 meters high. If a smaller antenna is installed, it will introduce some losses, which the receiver will have to handle with.

In this paper, we present a solution for a new physical layer for NVIS communications under two different hypotheses. First, in the case of limited transmission power (or lossy antennas) for battery powered systems, we find the best modulation scheme with a transmission power around 1 W. That will allow us to install small antennas, whenever the losses remain below 15 dB. Second, in the case of equipment connected to the mains, we try to maximize the bit-rate using the best combination of modulation and bandwidth for a given transmission power.

The performance tests were carried out during the 2018-2019 Spanish Antarctic campaign in Livingston Island. We established a NVIS testbed from a remote location of the Spanish Antarctic Base Juan Carlos I. The remote node was placed at Rocky Glacier, 5.7 Km away from the base. In Figure 2 and Figure 3 we can see the node location and the elevation profile between the two nodes. The testbed was intended to study both the channel response and the physical layer in non real-time mode, as well as the operation of the system in real-time.

This paper is organized as follows. In section 2, a description of the system is presented. In section 3, we detail the test bench and the frame distribution of every transmission. In section 4, the results obtained of the different tests are presented. Finally, section 6 contains the conclusions.



Figure 3. NVIS communications at Livingston Island



Figure 4. Elevation profile from the base to Rocky Glacier

2. System description

The main goal of the testbed is to test the influence of the transmission power, carrier frequency, hour of the day, bandwidth and modulation scheme in the performance of a NVIS link. For that purpose, we need a complete reconfigurable software defined radio platform.

2.1 Hardware description

The hardware platform (see Figure 5) is based on available devices in the market, but all the programming code has been developed by our group in order to transmit and receive any kind of signal in both real-time and non real-time (Porte, Maso, Pijoan, & Badia, 2019). It can also be programmed as a radio-modem in real-time as we can see at (Maso, Porte, Pijoan, & Badia, 2019). Moreover, it allows optional GPS synchronization to help the timing adjust for the testing mode.

The hardware can be programmed to act as either a transmitter or a receiver. The system is composed principally by a Red Pitaya platform ("Red Pitaya," n.d.) and a Raspberry Pi 3 together with some peripherals described below. The Red Pitaya platform is the core of the system, essentially composed by a Zynq 7010 chip. This chip is an ARM (Advanced RISC Machine) + FPGA (Field-Programmable Gate Array) and includes two analog to digital converters (ADC) and two digital to analog converters (DAC). It is in charge of upsampling the baseband signals and downsampling the HF signal carrier (Kodali, Bopppana, & Kondapalli, 2013). A Linux operative system is running in the ARM and it is programmed in C code, while the FPGA is programmed directly in VHDL code. This architecture allows the ARM and the FPGA to communicate in a very efficient way, and to have a full control of all the parameters of the transmission.

On the other hand, the Raspberry Pi 3 is the microprocessor that interacts with the rest of the peripherals. This platform is also used with a Linux operative system and has been programmed in C code. It is connected to the Red Pitaya using an Ethernet connection, and to the GPS, a Teensy module and an external hard drive using a serial port. In reception mode, the Raspberry receives the HF signal after being downconverted to baseband and stores the IQ samples in the hard drive. In transmission mode, a crontab is set-up to decide the hour and minute when every test frame has to be transmitted. In the right moment, the Raspberry sends the IQ frames via serial to the Red Pitaya and a message to the Teensy module. The Teensy is a microcontroller in charge of the control of the signal amplifier and the antenna relay. When it receives a message from Raspberry Pi 3 to go to TX mode, it switches on the amplifier and commutes the antenna relay to the power amplifier. The GPS module, endowed with an external GPS antenna, provides time-synchronization between transmitter and receiver for sounding proposes.

Regarding the amplifiers, the system is composed by an LNA (Low Noise Amplifier) for the receiving HF signal and a power amplifier for the transmitting HF signal from Red Pitaya. The maximum transmission power amplifier is limited to 250 W, but the tests have been done at much lower values.

Radio Science

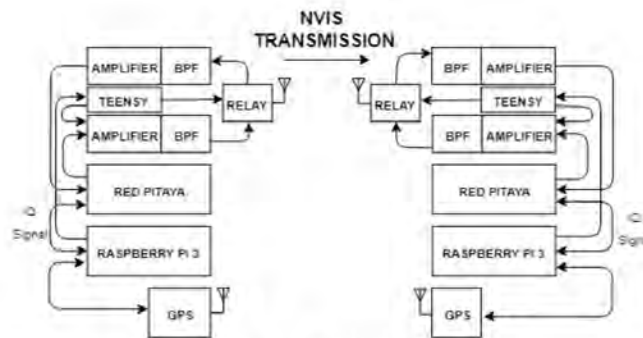


Figure 5. Schematic of the developed system

2.2 Antenna

A small HF antenna will present negative gains from -15dBi to -10dBi. Instead of using such type of antenna, we will test our platform using a low power signal transmission, from 1W to 24W, which values are very low compared to others HF transceivers of the market (“HF 3000 | Thales Group,” n.d.; “R&S®M3SR Series4100 Software Defined Radios | System Components | Rohde & Schwarz,” n.d.), where the power transmission of these products is from 20W to 400W.

For the modulation tests, we chose an inverted-V antenna as the best option as we can see in (Porte, Pijoan, et al., 2019). This antenna is easy to install, as it only needs one central mast and has a gain of around 2 dB. As indicated above, in NVIS the maximum of the radiation diagram of the antenna has to be between 70° and 90° to maximize the efficiency. In Figure 6 we can see how the simulation of an inverted-V over permafrost soil meets the specification.

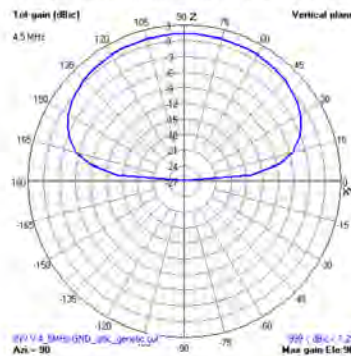


Figure 6. Radiation diagram of V-inverted at permafrost soil.

As the size of the antenna is directly related to the wavelength, we have to decide first the transmission frequency. For vertical incidence, the optimum frequency is selected as the 85 % of the critical frequency of the F2 layer (foF2). The critical frequency is highly dependent on the

Radio Science

season and the solar activity. After checking the historical records of the Ebre Observatory (“Ionogrames de Livingston,” n.d.) and Lowell Digisonde International (“LDI, Lowell Digisonde International Station List (Digisonde.com),” n.d.), we found out an average foF2 of about 5.3 MHz, so a transmission frequency of 4.3 MHz was selected. The size of the antenna has been calculated as $L = c/4f$, resulting a branch length of 15.63 m. Other factors as the mast height and the angle of aperture of the branches are key to achieve the optimum gain (see Figure 7). The height of the mast will configure the height of the antenna beyond the ground, this distance can create variations of the gain antenna because of reflections of the electromagnetic wave with the ground. By the other hand, the angle of aperture of the branches of the antenna also variates the antenna radiation diagram and the gain. These parameters have to be adjusted via simulation for a given type of soil (Witvliet et al., 2015). In Table 1 we can see the optimum values of the antenna dimensions for different types of soil.

The whole system (filters, matching networks) was optimized to operate at 4.3 MHz, which is the best frequency during the day. For night hours, the frequency should decrease and a longer antenna should be used.

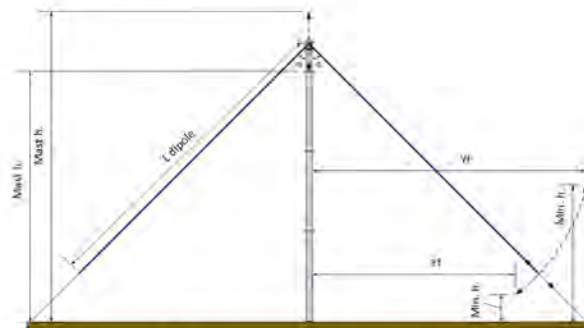


Figure 7. Dimensions of the inverted-V antenna

Soil type	Optimization algorithm	Gain (dBi)	SWR	Impedance (Ω)	Mast h. (m)	Miu h. (m)	Yf (m)
Ideal	Evolve	6.8	1.96	25.6+3.2j	11.01	2.00	12.39
Rural	Evolve	3.8	1.05	47.7+0.4j	10.81	1.87	12.39
Permafrost	Evolve	1.3	1.27	63.3-1.0j	13.08	2.00	11.51

Table 1. Optimum dimensions of the inverted-V antenna

2.3 Amplifier

Although the linearity of the power amplifier is required for non-constant envelope modulations, we wanted a linear amplifier to be able to test any kind of modulation. We installed an amplifier model BLWA 0103-250 from the manufacturer BONN in Holzkirchen, Germany (Bonn Elektronik, n.d.), a class-A amplifier which can deliver up to 250 W for an input power

lower than 0 dBm. It can be controlled through a RS232 serial port, managed by the Teensy platform.

For the final prototype of the system, another type of power amplifier should be used, depending on the modulation selected. For constant envelope modulations such as FSK and PSK, a class C amplifier is the best option.

3. Tests performed

In this section, we describe the tested combinations of power values and modulation scheme. We also detail the frame design to ensure that the channel coherence bandwidth is respected in each of the tests (Hervás, Pijoan, Alsina-Pagès, Salvador, & Altadill, 2013).

3.1 Test bench

The entire set of tests has been classified in two groups (see Table 2 and Table 3). In both groups, the PSK (Phase Shift Keying), QAM (Quadrature Amplitude Modulation) and FSK (Frequency Shift Keying) modulations are tested, with a number of constellation symbols ranging from 2 to 32. In the first group, the bandwidth is fixed to 2.38 KHz and the transmission power varies from 1 W to 24 W for every modulation so the bit rate ranges from 2.38 to 11.9 kbps. In one hour, we perform 30 different tests as we can see in Table 2. In the second group, transmission power is fixed to 24 W and the bandwidth varies from 2.38 KHz to 25 KHz (see Table 3). In this case, the bit rate varies from 5 kbps and 125 kbps.

Order of modulation	Power transmission	Minute
2,4,8,16,32	1 W	05,06,07,08,09
2,4,8,16,32	3 W	15,16,17,18,19
2,4,8,16,32	6 W	25,26,27,28,29
2,4,8,16,32	12 W	35,36,37,38,39
2,4,8,16,32	24 W	45,46,47,48,49
2,4,8,16,32	6 W	55,56,57,58,59

Table 2. Power transmission tests

Order of modulation	Bandwidth transmission	Minute
2,4,8,16,32	5 KHz	05,06,07,08,09
2,4,8,16,32	10 KHz	15,16,17,18,19
2,4,8,16,32	16.66 KHz	25,26,27,28,29
2,4,8,16,32	25 KHz	35,36,37,38,39
Other tests	Other tests	45,46,47,48,49
Other tests	Other tests	55,56,57,58,59

Table 3. Bandwidth transmission tests

Every group of transmissions has been working every hour for 15 days. The total amount of transmitted bits has been 16 MB and 20.1 MB for the first and the second group respectively. The second group involves more data transmitted due to the increase of bandwidth.

3.2 Frame design

The structure of the frame is shown in Figure 7. Every test is composed by 50 packets (N=50), that include three different types of modulation. Each modulation is preceded by a tone of 600 Hz and a PN sequence. The tone is used for frequency synchronization to mitigate the effect of both the Doppler shift of the channel and the mismatch between the transmitter and receiver clocks. The PN sequence is a 6th order m-sequence used for time synchronization, so a correlation has to be performed in real-time continuously at the receiver. The length of the frame has to be consistent with the coherence time of the channel (Proakis, 1995). Taking the most restrictive coherence time for the ordinary wave as 1.46s (corresponding to a Doppler spread of 0.6816Hz) (Hervás et al., 2013), we set the length of the packet to 510.36ms. In this interval, we make the test of three modulations (PSK, QAM, FSK) with the same order of modulation, power transmission and bandwidth. Hence, each modulation is tested under the same channels conditions.

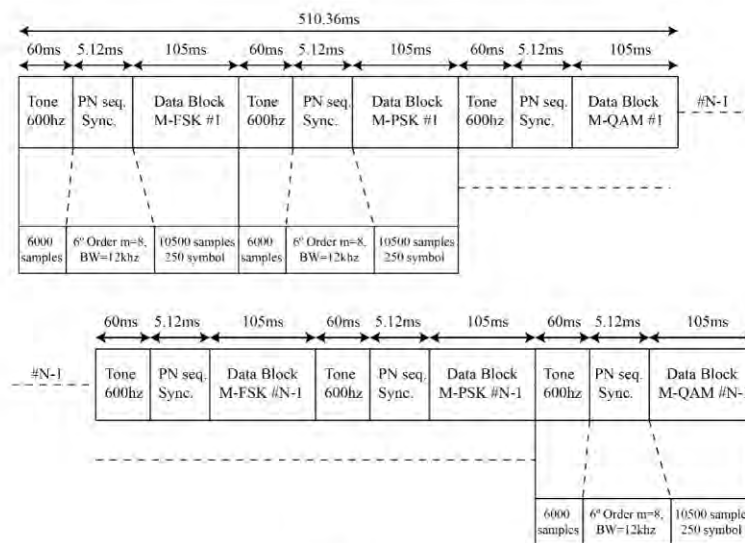


Figure 8: Frame design

4. Results

In this section we introduce the most relevant results of the tests performed in terms of modulation, bandwidth and transmission power. All the data was collected during the 2018-2019 Spanish Antarctic Campaign at Livingston Island, particularly from February 1st to March 2nd of 2019.

4.1 Power transmission tests

First of all, we measured the real relationship between the BER (Bit Error Rate) and E_b/N_0 (Bit energy to noise spectral density) of the received frames. The E_b/N_0 of the signal received is calculated as:

$$(1) (SNR) = \frac{E_b \cdot R_b}{B \cdot N_0}$$

$$(2) \frac{E_b}{N_0} = (SNR) \cdot \frac{B}{R_b}$$

$$(3) \frac{E_b}{N_0} (dB) = (SNR)(dB) + 10 \cdot \log_{10}(B) - 10 \cdot \log_{10}(R_b)$$

The received SNR is calculated in post-processing, R_b is the bit/rate of the signal (it must vary according to the modulation order of each test), and B is the noise bandwidth where we measure the received BER, in this case, 5KHz (the baseband signal BW is limited to 5KHz). Finally, the BER is calculated comparing the transmitted and received bits from a total of 25Mb for each trace.

In Figure 9 we can see the comparison between a second order PSK, FSK and QAM modulations. Due to the number of transmitted bits, our system is not able to measure BER values lower than 10^{-8} . We can see that QAM and PSK are more robust than FSK because of a given E_b/N_0 . The difference between 2PSK and 2QAM is shown in Figure 9. In 2PSK, both symbols are located on the real axis, while in 2QAM are located in the IQ plane. For a given bit energy, the real and complex components of the 2QAM are lower, so the performance is poorer. In order to achieve a BER of 10^{-3} , an E_b/N_0 of 10 dB is enough for 2PSK.

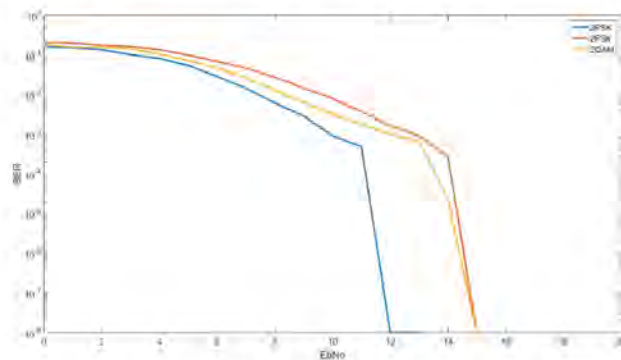


Figure 9. BER vs E_b/N_0 for 2nd order modulations

In figure 9, we can see the comparison of 4th order modulations. In this case, QAM and PSK modulations remain more robust than the FSK. 4QAM and 4PSK have exactly the same

Radio Science

constellation so they behave in the same way. In order to achieve a BER of 10^{-3} , an E_b/N_0 of 8 dB is enough for 4PSK and 4 QAM.

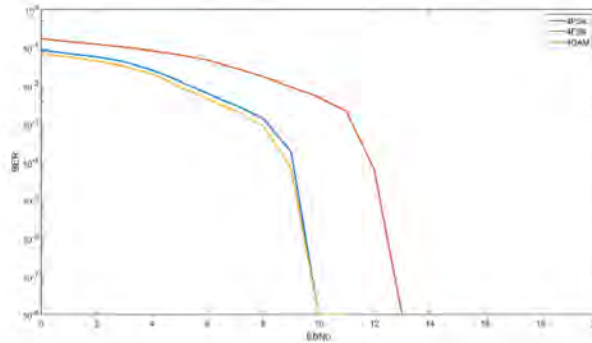


Figure 10. BER vs E_b/N_0 for 4th order modulations

For modulation orders higher than 4, the received symbols are very weak for being demodulated correctly. In Figure 11 and Figure 12, we can see the measured E_b/N_0 as a function of time of day for a transmission power of 24 W. The ionosphere reflected the signal mainly from 12:00 UTC to 19:00 UTC, so the link was available during a time window of 7 hours each day. If we look at the E_b/N_0 values, we can see values from 3 dB to 5dB for 8th order modulations and from 0 dB to 3dB to 32nd order modulations. None of these values is enough to guarantee a proper demodulation of the data.

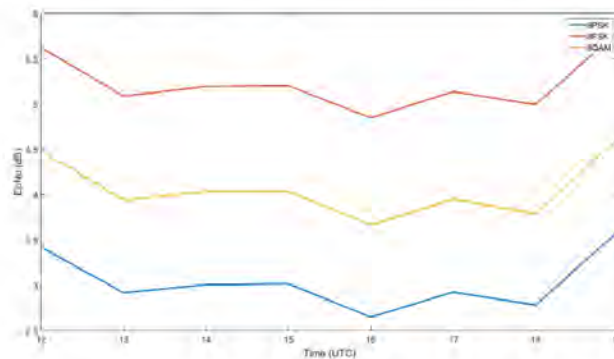


Figure 11. E_b/N_0 vs Time for 8th order modulations

Radio Science

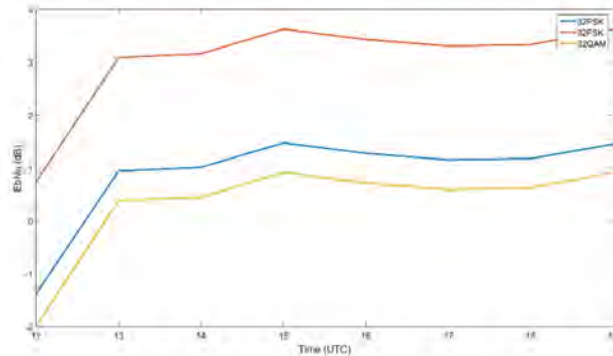


Figure 12. Eb/N0 vs Time order 32 modulations

Now, we are going to study the probability to have a given BER value. For that purpose, we will make use of the CDF (Cumulative Distribution Function) of the BER, that is, the probability of achieving a BER lower than a given value. From Figure 13 to Figure 18, the axis Y shows us the probability $P(\text{BER} < X_0)$ of obtain a BER less or equal than the value X_0 of axis X. For example, the probability $P(\text{BER} < X_0)$ if $X_0 = 0.01$ is 90%, so we can assert that the 90% of the packets transmitted have a BER lower than 0.01. In Figure 13 we can see the CDF of a 2nd order modulation PSK, QAM and FSK for a transmission power of 24 W. We can observe that the PSK and QAM have a probability of 95.2% and 95.4% respectively to achieve a BER lower than 0.004, while that probability is 92.8% for the FSK modulation.

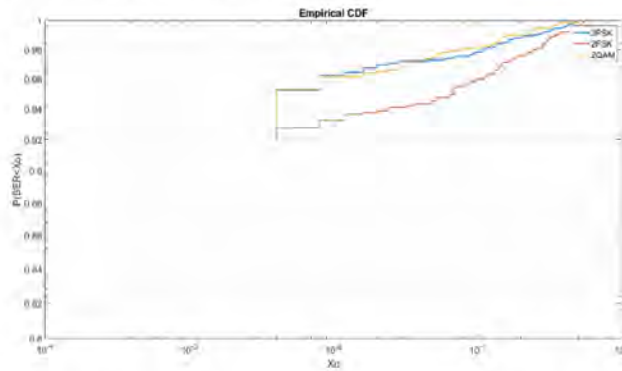


Figure 13. CDF of 2nd order modulations for 24 W

Radio Science

In Figure 14, we can see the probability in case of a power transmission of 2 W. In this case we can see that there is not a great difference between the three modulations. PSK and QAM modulations have a probability of 94.3% and 94.4% to achieve a BER lower than 0.004 respectively, in front of FSK with a probability of 93.8%.

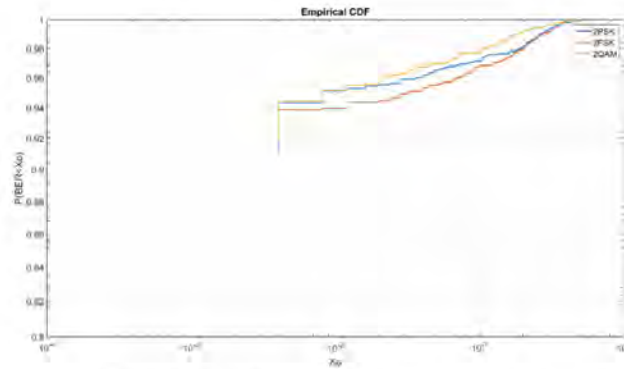


Figure 14. CDF of order 2 modulations for 2 W

In the case of a power transmission of 24 W for 4th order modulations (see Figure 15), the probability for 4FSK decreases to 77.3%. The 4PSK and 4QAM have a similar behavior with a probability of a 91.1% and 92.8% respectively.

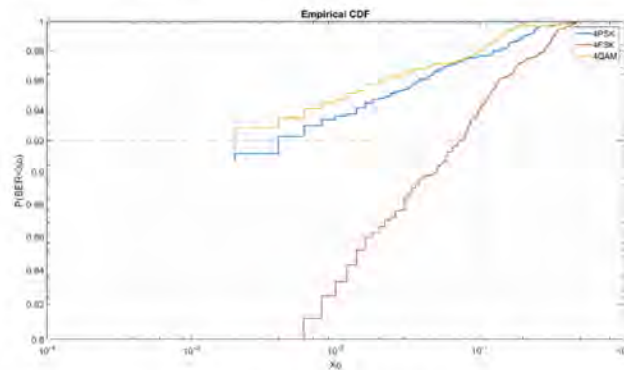


Figure 15. CDF of 4th order modulations for 24 W

On the other hand, Figure 16 shows the CDF for a transmission power of 2 W. In this case, we can see PSK and QAM modulation still outperform FSK with a probability of 90.1% and 91.1% respectively. The probability to have a BER lower than 0.004 decreases to 80 %.

Radio Science

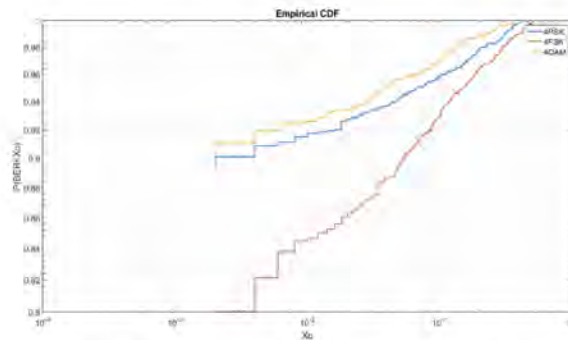


Figure 16. CDF of 4th order modulations for 2 W

Based on the results obtained, we can conclude that a 4QAM can be used with a transmission power of 2 W, since we will have a probability about 90 % to have a BER lower than 0.004. This is good enough to take into account the improvement due to the error correcting codes and the retransmissions of the frames. If we use 2PSK instead, there is a slight increase of the probability up to 94 %, but the bit rate is halved. If we use a transmission power of 24 W, we could also use a simpler modulation such as 2FSK with the same performance.

4.2 Bandwidth transmission tests

The previous tests have been performed using a standard baseband bandwidth of 2.3 KHz. We have also tested the use of higher bandwidths out of standard. For a given modulation scheme, the bit rate is directly proportional to the bandwidth, so we can reach bit rates up to 18.4 Kbps for a 10 KHz bandwidth.

In Figure 17, we can see the CDF of the BER of 4th order modulations for a bandwidth of 5 KHz with a transmission power of 24 W. The probability to have a BER lower than 0.001 is 73.7 % for both 4PSK and 4QAM, while it is 63.8 % for the FSK modulation.

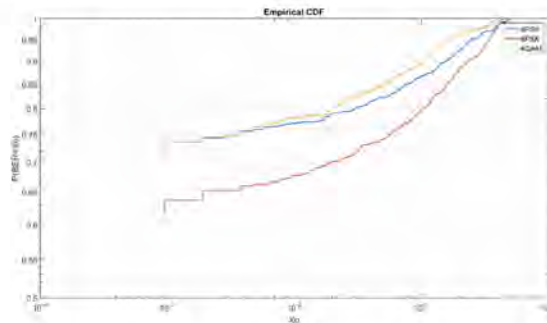


Figure 17. CDF of 4th order modulations for BW=5 KHz

Radio Science

The same test for a bandwidth of 10 KHz is shown in Figure 18. Here, the probability to have a BER lower than 0.001 is around 66 % for both 4PSK and 4QAM, in front of 43.9% for the FSK modulation.

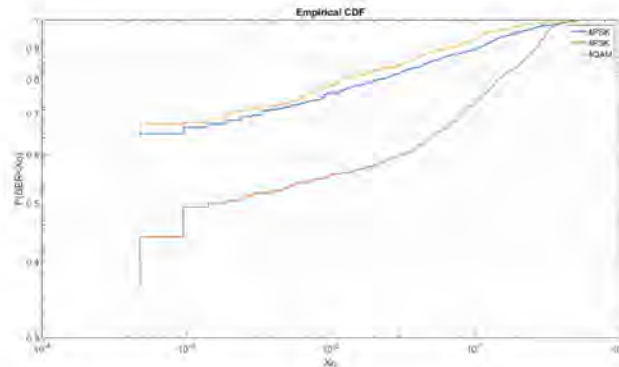


Figure 18. CDF of 4th order modulations for BW=10 KHz

Although the probabilities are lower when we use higher bandwidths, we have to take into account that the bit rate is increased in a larger proportion, so the overall performance is better. If our application needs higher throughputs, we could also use greater bandwidths without increasing the power consumption.

5. Conclusions

In this paper, we have introduced a new solution for remote sensors using the NVIS communication technique. We have explored the influence of the type of modulation, the transmission power and the bandwidth for a NVIS link between sensors located around the Spanish Antarctic Base in Livingston Island.

For low power consumption sensors, for a transmission power of 2 W, we recommend the use of the standard bandwidth of 2.3 KHz with either a 4QAM or 2PSK modulation That yields to 4.6 kbps or 2.3 kbps with a probability around 90 % to have a BER lower than 0.004 prior to any correcting code.

For sensors with higher power consumption (i.e. 24 W), we could also use the standard bandwidth with simpler modulations such as 2FSK, such a simpler amplifier could be used. If we need higher bit-rates without increasing the power, we could also increase the bandwidth up to 5 KHz. In that case we could use a 4QAM with a probability around 73.7 % to have a BER lower than 0.001 for a bit rate of 9.2 kbps.

We have also defined a frame structure for NVIS communications. The frame includes a preamble for frequency and time synchronization, as well as a maximum length for each data block.

Radio Science

The tests prove the feasibility of NVIS for sensors located up to 250 Km away from the central node, without the need of any telecom operator and it is an interesting alternative to satellite for places without infrastructures and complicated terrain.

As we can see, the results obtained show us the specifications to achieve the robust transmission for low transmissions of sensors out of the standards. In case of an equalized 4QAM or 4PSK of 2 W and 2.4 KHz bandwidth, as we have seen, the 90% of the packets transmitted have a BER lower than 0.004 in a 12.500 symbols frame and 25.000 bits. In the worst-case scenarios, we will have 100-bit errors. If we apply an interleaving method and an FEC convolutional code with a rate code 1/2 as the standard MIL-STD-188-110B shows (MIL-STD-188-110B, 2000) , we will improve the probability of a BER less than 0.001 to a 94%. The applying of the FEC will reduce the bitrate to 2.4 Kbps, this is enough for the transmission of sensor data.

The results obtained during the Antarctic campaign, have been implemented for a real-time transmission of sensor data to the Spanish Antarctic Base. The implemented system did not use code correcting and interleaving methods and a 24 W power transmission and 2.4 KHz bandwidth was used. More information about the implementation done can be found at (Maso et al., 2019). In a future line of the real-time implementation of the system, a real-time ionospheric sounding, interleaving and FEC methods will be implemented in the same platform.

This work performs a study of the greatest physical layer definition for remote areas using NVIS communication. In such places, the consumption of power is one of the most critical issues in a telecommunication system design. Most of the current products in the market have a RF signal output power from 20W to 1500W ("HF 3000 | Thales Group," n.d.; "R&S®M3SR Series4100 Software Defined Radios | System Components | Rohde & Schwarz," n.d.). In these kinds of areas, the system should be powered by batteries which are too large. Comparing the results shown in this paper with the most commonly used international standards (MIL-STD-188-110B, 2000; MIL-STD-188-110D, 2017; North Atlantic Treaty Organization (NATO), 2009; STANAG 4539, n.d.), we obtain a better performance for the case of very low power consumption and without applying any interleaving method or an FEC convolutional code.

Compared to others scientific works, we can assume that the performance, in terms of BER achieved, in our system is much better than the one described in similar previous works. In the PhD thesis (Wilson, 2012) , we can see a BER of 10^{-2} for an E_b/N_0 of 11dBm, with a 2-QAM and a power signal transmission of 5W. Finally, in (Saraç, Kara, & Vural, n.d.) we can find a real time implementation of the standard STANAG 4539 (STANAG 4539, n.d.) that achieves the minimum performance of the standard, but without improving it.

Acknowledgments

This work was funded by the Ministry of Economy and Competitiveness and the European Regional Development Fund under the contract CTM2015-68902-R (MINECO/FEDE) and RTI2018-097066-B (MINECO/FEDER).

The data used to develop our study and generate all the figures can be obtained online in <https://doi.org/10.5281/zenodo.3523219> and in <http://doi.org/10.5281/zenodo.3523281>.

References

- AEMET. (2019). Jornada sobre las actividades de AEMET en la Antártida - State Meteorological Agency - AEMET - Spanish Government. Retrieved June 4, 2019, from

Radio Science

- http://www.aemet.es/en/noticias/2019/02/jornada_antartida
- Bonn Elektronik. (n.d.). Bonn Elektronik Power amplifier 9kHz-40GHz. Retrieved from <https://alice.de/wp-content/uploads/2017/03/Bonn-Katalog-Web.pdf>
- Davies, K. (1990). *Ionospheric Radio*. The Institution of Engineering and Technology, Michael Faraday House, Six Hills Way, Stevenage SG1 2AY, UK: IET. <https://doi.org/10.1049/PBEW031E>
- Hervás, M., Pijoan, J. L., Alsina-Pagès, R. M., Salvador, M., & Altadill, D. (2013). Channel sounding and polarization diversity for the NVIS channel. *Proceedings Nordic HF Conference*. (August).
- HF 3000 | Thales Group. (n.d.). Retrieved October 24, 2019, from <https://www.thalesgroup.com/en/worldwide/defence/hf-3000-skyfst>
- Ionogramas de Livingston. (n.d.). Retrieved July 6, 2018, from <http://www.obsebre.url.edu/ca/ionogramas-de-livingston>
- Kodali, R. K., Boppana, L., & Kondapalli, S. R. (2013). DDC and DUC filters in SDR platforms. In *2013 15th International Conference on Advanced Computing Technologies (ICACT)* (pp. 1–6). Rajampet, India: IEEE. <https://doi.org/10.1109/ICACT.2013.6710526>
- Lauridsen, M., Nguyen, H., Vejlgard, B., Kovacs, I. Z., Mogensen, P., & Sorensen, M. (2017). Coverage Comparison of GPRS, NB-IoT, LoRa, and SigFox in a 7800 km² Area. In *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)* (pp. 1–5). Sydney, NSW, Australia: IEEE. <https://doi.org/10.1109/VTCSpring.2017.8108182>
- LDI, Lowell Digisonde International Station List (Digisonde.com). (n.d.). Retrieved July 6, 2018, from <http://www.digisonde.com/stationlist.html>
- Maso, J., Porte, J., Pijoan, J. L., & Badia, D. (2019). Internet of things communications for remote sensors in Antarctica using NVIS. In *HF Nordic*. Fårö, Sweden.
- MIL-STD-188-110B. (2000). *DEPARTMENT OF DEFENSE INTERFACE STANDARD INTEROPERABILITY AND PERFORMANCE STANDARDS FOR DATA MODEMS*.
- MIL-STD-188-110D. (2017). *DEPARTMENT OF DEFENSE INTERFACE STANDARD: INTEROPERABILITY AND PERFORMANCE STANDARDS FOR DATA MODEMS*. Retrieved from http://everyspec.com/MIL-STD/MIL-STD-0100-0299/MIL-STD-188-110D_55856/
- North Atlantic Treaty Organization (NATO). (2009). *STANAG 5066: The Standard for Data Applications over HF Radio*. Retrieved from <https://www.isode.com/whitepapers/stanag-5066.html>
- Porte, J., Maso, J. M., Pijoan, J. L., & Badia, D. (2019). Design, implementation, and test of an SDR for NVIS communications. *International Journal of Circuit Theory and Applications*, 47(9), 1502–1512. <https://doi.org/10.1002/cta.2670>
- Porte, J., Pijoan, J. L., Masó, J. M., Badia, D., Zaballos, A., & Alsina-Pagès, R. M. (2019). Advanced HF communications for remote sensors in Antarctica. In *Antartica - A Key To Global Change* (1st ed.). IntechOpen.
- Proakis, J. G. (1995). *Digital communications*. McGraw-Hill.
- R&S®M3SR Series4100 Software Defined Radios | System Components | Rohde & Schwarz. (n.d.). Retrieved October 24, 2019, from <https://www.rohde-schwarz.com/es/producto/m3srseries4100-productos->

Radio Science

secundarios_63489-9903.html

Red Pitaya. (n.d.). Retrieved June 4, 2019, from <https://www.redpitaya.com/>

Saraç, S., Kara, F., & Vural, C. (n.d.). Real-Time Implementation of STANAG 4539 High-Speed HF Modem. In *World Academy of Science, Engineering and Technology* (p. 6).

STANAG 4539. (n.d.). *North Atlantic Treaty Organization*. Retrieved from <https://www.rapidm.com/standard/stanag-4539/>

Statista - Internet of Things - Number of connected devices worldwide 2015-2025. (n.d.). Retrieved May 22, 2019, from <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>

Vilella, C., Miralles, D., Altadill, D., Acosta, F., Solé, J. G., Torta, J. M., & Pijoan, J. L. (2009). Vertical and oblique ionospheric soundings over a very long multihop HF radio link from polar to midlatitudes: Results and relationships. *Radio Science*, *44*(2), n/a-n/a. <https://doi.org/10.1029/2008RS004001>

Wilson, J. M. (2012). *A low power HF communication system*. Retrieved from <https://www.escholar.manchester.ac.uk/api/datastream?publicationPid=uk-ac-man-scw:156816&datastreamId=FULL-TEXT.PDF>

Witvliet, B. A., & Alsina-Pagès, R. M. (2017). Radio communication via Near Vertical Incidence Skywave propagation: an overview. *Telecommunication Systems*, *66*(2), 295–309. <https://doi.org/10.1007/s11235-017-0287-2>

Witvliet, B. A., van Maanen, E., Petersen, G. J., Westenberg, A. J., Bentum, M. J., Slump, C. H., & Schiphorst, R. (2015). Near Vertical Incidence Skywave Propagation: Elevation Angles and Optimum Antenna Height for Horizontal Dipole Antennas. *IEEE Antennas and Propagation Magazine*, *57*(1), 129–146. <https://doi.org/10.1109/MAP.2015.2397071>