

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA



Trabajo Fin de Máster

**Análisis y diseño en VHDL de un
ecualizador CMA para enlaces de
datos**

**Analysis and design of a
equalizer in VHDL**

Para acceder al Título de

***Máster Universitario en
Ingeniería de Telecomunicación***

Autor: Aizkibel Salcedo Arrugaeta

Mayo - 2018



E.T.S. DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACION

MASTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

CALIFICACIÓN DEL TRABAJO FIN DE MÁSTER

Realizado por: Aizkibel Salcedo Arrugaeta

Director del TFM: David Díez Hijano

Título: Análisis y diseño en VHDL de un ecualizador CMA para enlaces de datos

Title: Analysis and design of a CMA equalizer in vhdl

Presentado a examen el día: 23 de mayo del 2018

para acceder al Título de

MASTER UNIVERSITARIO EN INGENIERÍA DE TELECOMUNICACIÓN

Composición del Tribunal:

Presidente (Apellidos, Nombre): Antonio Tazón Puente

Secretario (Apellidos, Nombre): Beatriz Aja Abelán

Vocal (Apellidos, Nombre): Tomás Fernández Ibáñez

Este Tribunal ha resuelto otorgar la calificación de:

Fdo.: El Presidente

Fdo.: El Secretario

Fdo.: El Vocal

Fdo.: El Director del TFM
(sólo si es distinto del Secretario)

Vº Bº del Subdirector

Trabajo Fin de Máster Nº
(a asignar por Secretaría)

Agradecimientos

Resumen

El avance de los últimos veinte años en cuestiones de potencia de cómputo de las FPGAs, de los algoritmos de inteligencia artificial y de las baterías ha permitido que los UAV (*Unmanned Aerial Vehicle*) se hayan ido extendiendo de forma progresiva tanto en el ámbito militar como en el civil.

Aunque la mayoría de los UAVs comunes en la vida cotidiana se caracterizan por tener un tamaño reducido, una gran parte de los utilizados con fines gubernamentales presentan, en cambio, grandes dimensiones y proporcionan video en tiempo real, necesitando, por lo tanto, enlaces de datos con un gran ancho de banda.

En este contexto, el presente proyecto aborda el diseño e implementación de un ecualizador CMA eficiente para bandas anchas en VHDL, para posteriormente analizar su comportamiento en simulación frente a distintos canales.

El trabajo finaliza con la integración del ecualizador CMA desarrollado en un sistema completo, analizándose su comportamiento en tiempo real en distintos escenarios.

Abstract

In the last twenty years the computing power of the FPGAs, the artificial intelligence algorithms and the batteries have improved significantly. The simultaneous advance on those three fields has made possible the development of UAVs that are used both in military and civil fields.

The most common UAVs are small, but this project is focused on big UAVs used for governmental and non-commercial purposes, that require high bandwidth for real-time video applications.

In this context, this project addresses the design and VHDL implementation of an efficient CMA equalizer for wide bandwidth. As a part of this work, its performance in different channels has been deeply analyzed in simulation.

The last part of this project was the validation of the equalizer. To achieve this objective, it was integrated in the receiver of a system. The performance of the equalizer was tested and analyzed in different real-time scenarios.

Índice

1. Introducción y objetivos del Proyecto	1
1.1. Motivación y objetivos.....	1
1.2. Estructura del Proyecto	2
2. Estado del arte	3
2.1. Comunicaciones inalámbricas	3
2.1.1. Efectos de la propagación.....	4
2.1.2. Canales.....	6
2.1.3. Ecuilibradores	8
2.1.4. Conclusión.....	14
2.2. Procesado de señal.....	14
2.3. FPGA.....	16
2.3.1. Limitaciones de una FPGA	18
3. Desarrollo realizado	20
3.1. Implementación.....	20
3.2. Cross Domain Crossing	24
4. Depuración, verificación y análisis	26
4.1. Depuración y validación.....	26
4.2. Verificación	27
4.2.1. Análisis teórico de la implementación.....	28
4.2.2. Simulación del CMA ante distintos canales	32
5. Comportamiento en canales reales	40
5.1. Banda base.....	40
5.2. RF mediante cable	41
5.3. Transmisión al aire.....	44
5.4. Otras pruebas	46
5.4.1. Recuperación de fase de la portadora.....	46
5.4.2. IQ desbalanceadas	48
5.4.3. Cambios rápidos en el canal	49
6. Conclusiones y líneas futuras	51

Índice de Figuras

Figura 2.1: Transmisión inalámbrica	3
Figura 2.2: Señal en banda base y RF	4
Figura 2.3: Pérdidas en espacio libre.....	4
Figura 2.4: Respuesta al impulso de un canal típico	5
Figura 2.5: Difracción y zonas de fresnel.....	5
Figura 2.6: Efecto Doppler	6
Figura 2.7: Función de densidad de probabilidad del canal AWGN	7
Figura 2.8: Respuesta del canal Rayleigh en tiempo y frecuencia	7
Figura 2.9: Estructura de un filtro adaptativo	9
Figura 2.10: Superficie de coste del LMS.....	10
Figura 2.11: Velocidad de convergencia del LMS vs RLS	11
Figura 2.12: Ecuador GSA	12
Figura 2.13: Representación IQ	14
Figura 2.14: Evolución de la tecnología de fabricación	16
Figura 2.15: Arquitectura de una FPGA.....	17
Figura 2.16: CLB de una Spartan6.....	18
Figura 2.17: DSP de Xilinx	18
Figura 2.18: Multiplicación eficiente de dos números complejos	18
Figura 3.1: Esquema general del ecualizador.....	21
Figura 3.2: Máquina de estados del ecualizador.....	22
Figura 3.3: Cálculo del error del ecualizador.....	23
Figura 3.4: Problema al cambiar de dominio de reloj	24
Figura 3.5: CDC con memoria de doble reloj.....	24
Figura 3.6: Esquema del ecualizador a nivel frecuencial.....	25
Figura 4.1: Test con Matlab	26
Figura 4.2: Proceso de verificación del ecualizador	27
Figura 4.3: El problema de la cuantificación	28
Figura 4.4: Diferencia entre algoritmo e implementación por la cuantificación	29
Figura 4.5: Respuesta al impulso de un canal normal.....	30
Figura 4.6: Velocidad de convergencia en función del paso	30
Figura 4.7: Canal $h = [0.2, 0.5, 1, -1]$ para 20 dB y 5 dB de SNR.....	31
Figura 4.8: Canal $h = [.9 -0.3 .4 -0.7]$ para 60 dB y 5 dB de SNR	31
Figura 4.9: BER LMS,RLS,CMA	32
Figura 4.10: Respuesta al impulso de un canal común	33
Figura 4.11: Señal antes y después de pasar por el canal	33
Figura 4.12: Señal antes y después de pasar por el ecualizador	34
Figura 4.13: Señal ecualizada	34
Figura 4.14: Coeficientes del ecualizador.....	35
Figura 4.15: Señal ecualizada con atenuación de 25 dBs.....	35
Figura 4.16: Señal ecualizada con atenuación de 25 dBs y AGC digital	36
Figura 4.17: Coeficientes del ecualizador con AGC digital	36

Figura 4.18: Velocidad de convergencia para distintas atenuaciones	37
Figura 4.19: Señal ecualizada con AWGN.....	37
Figura 4.20: Señal ecualizada con 25 dBs de atenuación y AWG.....	38
Figura 4.21: Velocidad de convergencia en función de la SNR	38
Figura 4.22: Ecualizador frente a canales Rice (k=4 y k=12) y Rayleigh	39
Figura 5.1: Prueba en banda base	40
Figura 5.2: Resultados en banda base.....	41
Figura 5.3: Prueba en RF mediante cable.....	42
Figura 5.4: Resultados en RF con cable	43
Figura 5.5: Resultados en RF con cable y máxima atenuación.....	43
Figura 5.6: Prueba en RF con antenas	44
Figura 5.7: Resultados de la prueba con antenas	45
Figura 5.8: Resultados de la prueba con antenas y NLOS	45
Figura 5.9: Resultados de la prueba con antenas y NLOS (2).....	46
Figura 5.10: Recuperación de fase	47
Figura 5.11: Resultados sin/con recuperación de fase.....	47
Figura 5.12: Desbalanceo en ganancia	48
Figura 5.13: Desbalanceo en fase.....	48
Figura 5.14: Resultados con el bloque de balanceo IQ desactivado	49
Figura 5.15: Resultados con cambios bruscos en el canal.....	49

Índice de Tablas

Tabla 2.1: Multiplicaciones de señales reales y señales en IQ.....	15
Tabla 3.1: Recursos Spartan6 XC6SLX100T	20
Tabla 3.2: Entradas y salidas del ecualizador	23
Tabla 3.3: Recursos del ecualizador para 16 etapas	23
Tabla 3.4: Figuras de mérito del ecualizador	23
Tabla 5.1: Parámetros del escenario con cables	42
Tabla 5.2: Parámetros del escenario con antenas	44

Capítulo 1

Introducción y objetivos del Proyecto

Durante la última década, la utilización de UAVs, bien en el ámbito militar, bien en el civil, ha aumentado de forma considerable gracias tanto a las increíbles prestaciones que ofrecen como al bajo coste que suponen, dado que en este momento nos encontramos ya ante una tecnología madura.

Si nos centramos en concreto en el último año, los UAVs de uso recreativo se han convertido en un dispositivo asentado y ampliamente difundido. Su corto alcance, baja velocidad y banda estrecha les ha permitido aprovecharse del conocimiento adquirido en sistemas de radio control; no obstante, no es este el objeto del trabajo, sino que el presente proyecto se centra en enlaces de datos de larga distancia diseñados para UAVs de gran tamaño, en entornos más hostiles, con requisitos de QoS (*Quality of Service*) estrictos, utilizados principalmente con fines gubernamentales. Son sistemas SISO (*Single Input Single Output*), con gran autonomía y alcance de cientos de kilómetros en condiciones de LoS (*Line of Sight*), que tienen que enfrentarse en algunos casos al efecto Doppler y que, además, necesitan una velocidad de transmisión considerable.

1.1. Motivación y objetivos

A la hora de diseñar el *firmware* de un DL (*DataLink*) hay una serie de elecciones clave que hay que tomar en función de los requisitos del enlace. Y aun así, ni siquiera en teoría se podría afirmar sin lugar a duda qué o qué no implementar con el fin de obtener un rendimiento óptimo.

La corrección de errores, la codificación de canal, la modulación, el espectro ensanchado... son, todos ellos, bloques que repercutirán decisivamente en el rendimiento y eficiencia del enlace, pero en los que el diseñador debe enfrentarse a un *trade-off* entre resistencia al ruido, banda necesaria y complejidad de implementación.

Por otro lado, hay un bloque que influye de manera significativa en la recepción y que no supone ningún coste de ancho de banda sino sólo de recursos de la FPGA (*Field Programmable Gate Array*). Ese bloque es el ecualizador, tema principal de este proyecto.

Un ecualizador no es más que un filtro digital adaptativo que intenta compensar el multitrayecto causado por el canal, tomando como criterio alguna información disponible a priori.

Sin embargo, el número de recursos dentro de una FPGA es limitado y las implementaciones con un amplio ancho de banda requieren un esfuerzo considerable.

En resumen, los objetivos del presente proyecto son los siguientes:

- Análisis teórico de los ecualizadores y principalmente, del ecualizador CMA (*Constant Module Algorithm*).
- Diseño e implementación de un ecualizador CMA en VHDL (*VHSIC Hardware Description Language*).
- Análisis y validación de la implementación anterior.
- Integración del bloque CMA en un sistema real.

1.2. Estructura del Proyecto

El trabajo se compone de seis capítulos, incluida la presente introducción. Tras ella, se inicia el Capítulo 2 con una presentación del estado del arte en relación a las comunicaciones inalámbricas y al procesado de señal, como preámbulo a un análisis teórico del conjunto de ecualizadores existentes en la técnica.

En el Capítulo 3 se describe con rigor y precisión el desarrollo realizado obviando, en la medida de lo posible, los detalles del código desarrollado. En el 4 se presentan los procesos de depuración, verificación y validación, concluyendo este último con un análisis teórico a nivel de simulación del rendimiento del CMA.

La prueba real del rendimiento es objeto del Capítulo 5, en el que se integra el ecualizador dentro de un sistema comercial y se monitoriza la entrada y la salida bajo distintos escenarios.

Finalmente, en el Capítulo 6 se sintetizan las aportaciones realizadas evaluando su utilidad práctica y sus ventajas y desventajas respecto a otras implementaciones, para terminar con un compendio de futuras líneas de investigación.

Capítulo 2

Estado del arte

La implementación práctica de un ecualizador en VHDL es una tarea que se extiende horizontalmente a lo largo de muchas disciplinas de las telecomunicaciones como, por ejemplo, el procesado de señal, las FPGAs, o las comunicaciones inalámbricas. Para comprender adecuadamente el desarrollo posterior, es necesario entender algunos conceptos clave de todos los ámbitos abarcados.

2.1. Comunicaciones inalámbricas

Desde que en 1888, el físico alemán Rudolf Hertz lograra la primera transmisión sin cables a través de ondas electromagnéticas hasta que hemos sido capaces de recibir datos de sondas en el espacio profundo, la humanidad ha recorrido un arduo camino. Este proyecto está enfocado hacia los enlaces de datos, que son aquellas comunicaciones inalámbricas entre dos puntos, ya sean móviles o fijos, que requieren un ancho de banda importante.

En este tipo de enlaces, un emisor transmite información mediante una señal modulada por un canal a través de una antena, pudiendo, a su vez, recibir información, ya sea al mismo tiempo (full-dúplex) o utilizando distintas técnicas de compartición del canal (half-dúplex).

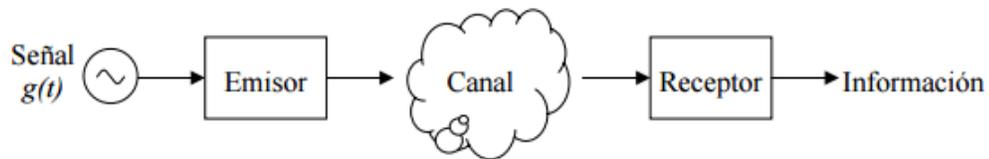


Figura 2.1: Transmisión inalámbrica

El movimiento de la onda a través del canal es conocido como propagación. Los efectos que sufre una onda al ser propagada por un canal pueden ser críticos a la hora de reconstruir la señal en el receptor, pero teniendo información a priori sobre ellos, se pueden implementar diversas soluciones para mitigarlos.

Para conseguir que la onda se propague es necesario modularla correctamente, es decir, trasladar el espectro de la señal desde DC (0 Hz) hasta la frecuencia de transmisión:

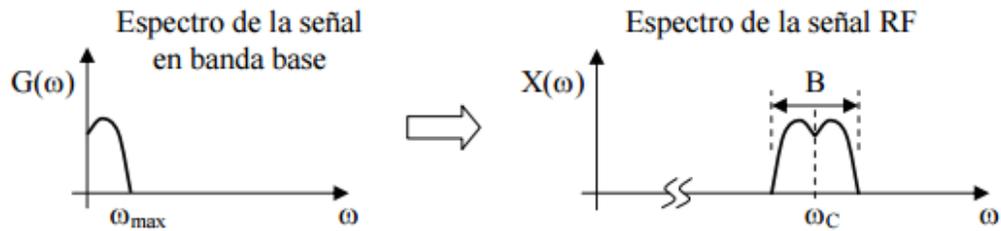


Figura 2.2: Señal en banda base y RF

La primera señal es una señal paso bajo, mientras que la segunda es una señal paso banda de anchura B, donde B es el ancho de banda.

2.1.1. Efectos de la propagación

Aunque los efectos que sufre una señal al propagarse por la atmósfera dependen en gran medida de la frecuencia que se esté usando, se pueden reducir a los siguientes tipos:

2.1.1.1 Atenuación por distancia

Dado que la atmósfera dispersa la energía de una onda a lo largo de la distancia del enlace, este resulta ser el factor más limitante a la hora de diseñar enlaces de largas distancias, pues una vez que la señal se acerque al fondo de ruido correspondiente a su ancho de banda, ya no podrá recuperarse. La UIT-R P.525-2 presenta un método comúnmente aceptado para calcular la atenuación que sufre la señal en condiciones óptimas atmosféricas:

$$L_{bf} = 20 \log \left(\frac{4\pi d}{\lambda} \right) \quad \text{dB}$$

donde:

L_{bf} : pérdida básica de transmisión en el espacio libre (dB)

d : distancia

λ : longitud de onda

d y λ se expresan en las mismas unidades.

Figura 2.3: Pérdidas en espacio libre

2.1.1.2 Atenuación por desvanecimiento

La propagación multitrayecto, intrínseca a los medios no guiados, se produce cuando la señal rebota con los objetos del medio produciéndose copias que, debido a que recorren más distancia, llegan retardadas respecto a la señal principal, provocando lo que se conoce como distorsión temporal. Además, estas ondas reflejadas producen interferencias, constructivas o destructivas, en la señal recibida, siendo imposibles de prever. El comportamiento del multitrayecto depende de la frecuencia y de la longitud de onda, y es mucho más grave en entornos interiores. La respuesta al impulso del canal identifica perfectamente el efecto causado por este fenómeno, pero sólo durante el tiempo de coherencia del canal, es decir, durante el periodo en el que el canal se puede considerar invariante.

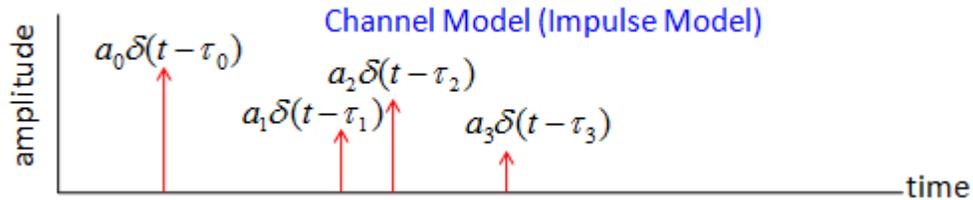


Figura 2.4: Respuesta al impulso de un canal típico

2.1.1.3 Difracción

Cualquier obstáculo que se encuentre situado cerca de la línea de visión del transmisor al receptor puede llegar a influir en la potencia que este último reciba.

Concretamente afecta si dicho obstáculo está localizado dentro de la primera zona de Fresnel. Llamamos zonas de Fresnel al volumen de espacio entre el emisor y un receptor en el cual el desfase de las ondas no supera los 180°.

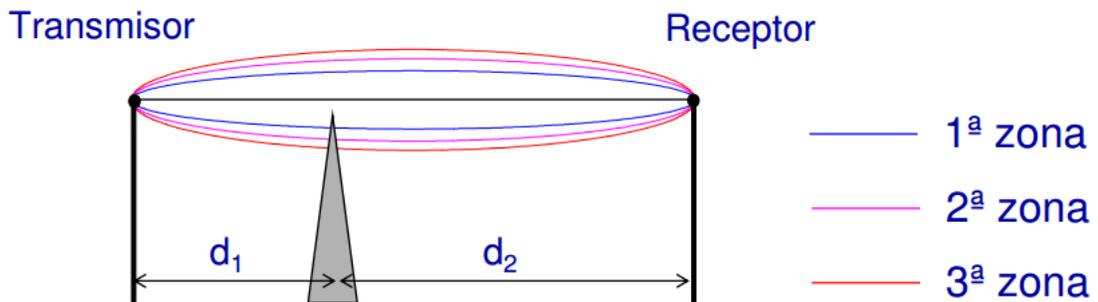


Figura 2.5: Difracción y zonas de Fresnel

2.1.1.4 Efecto Doppler

En el caso de que transmisor y receptor tengan un movimiento relativo uno respecto al otro, o que algún objeto del medio se esté moviendo y reflejando la onda transmitida, la portadora tiende a cambiar la frecuencia, afectando a la señal en banda base, que se dispersa.

Es posible corregirlo procesando adecuadamente la señal.

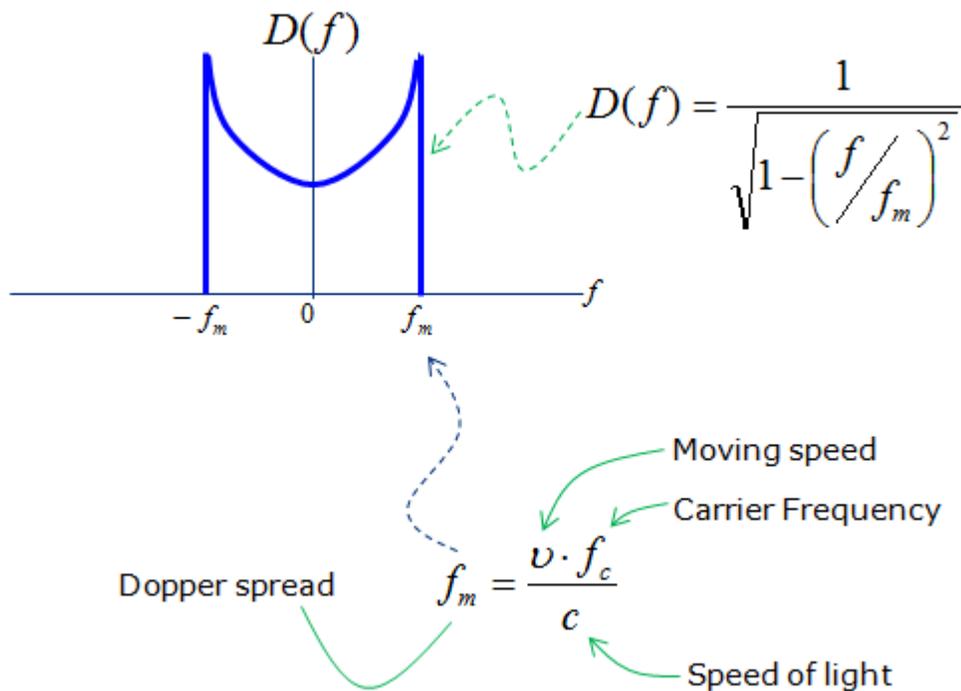


Figura 2.6: Efecto Doppler

2.1.2. Canales

Con el fin de poder simular y adaptar el sistema de comunicaciones al canal en el que va a ser utilizado, a lo largo de los años se han ido desarrollando modelos estadísticos que permiten caracterizar un canal. Los más conocidos son los siguientes:

2.1.2.1 AWGN

Es un canal puramente teórico que añade ruido blanco gaussiano; es uniforme en el tiempo y en el espacio; y no contempla selectividad temporal ni frecuencial, ni multicamino. Su densidad de probabilidad es la siguiente:

$$f(n) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{1}{2} \left(\frac{n}{\sigma}\right)^2}$$

Figura 2.7: Función de densidad de probabilidad del canal AWGN

2.1.2.2 Rayleigh

Representa un canal urbano o interior en el cual la antena receptora se mueve constantemente como, por ejemplo, en el caso de las comunicaciones móviles. No hay línea de vista y la potencia recibida instantánea se puede modelar como una variable aleatoria, consecuencia de las interacciones destructivas debido al multitrayecto. Es selectivo en frecuencia y varía con el tiempo.

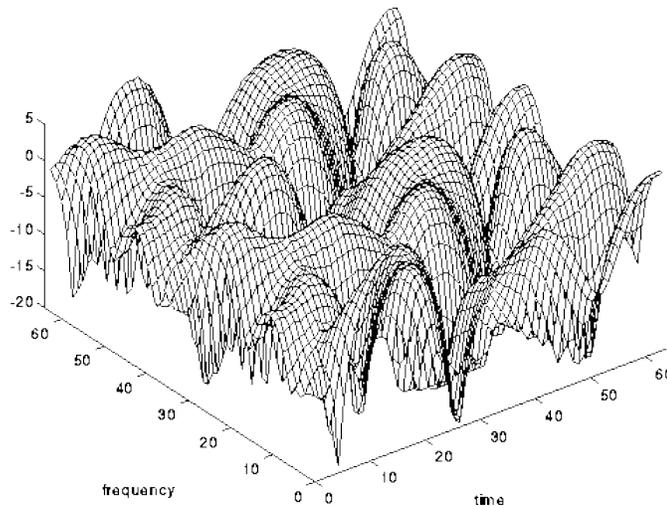


Figura 2.8: Respuesta del canal Rayleigh en tiempo y frecuencia

2.1.2.3 Rician

Es un modelo similar al de Rayleigh, pero en el que existe una componente predominante debido a que hay línea de vista. Modelos más completos, como los utilizados para simular comunicaciones por satélite, incluyen también desvanecimientos en dicha señal.

La diferencia de potencia entre la componente predominante y el resto de señales recibidas debido al multitrayecto, se define como factor K .

El medio no guiado al que nos enfrentamos es un canal RICIAN variante con el tiempo, capaz de deteriorar la señal hasta hacer imposible su demodulación. De todos los efectos que puede causar este canal, el presente proyecto se centra en el multitrayecto. El resto de distorsiones deben ser corregidas por otros bloques ajenos al demodulador y su análisis queda fuera del contexto de esta obra.

En la mayoría de los sistemas de comunicaciones inalámbricos, este efecto es variante con el tiempo, haciéndose necesario la presencia de un filtro adaptativo, es decir, un filtro capaz de cambiar en el tiempo en función de la respuesta que presente el canal.

Los mejores filtros para implementar con un comportamiento como el descrito anteriormente son los filtros digitales con etapas, cuyos coeficientes van cambiando a lo largo del tiempo debido a un criterio establecido. Son conocidos como ecualizadores adaptativos.

2.1.3. Ecualizadores

Durante los últimos 50 años se han diseñado y estudiado muchos tipos de ecualizadores; sin embargo, en la actualidad se utilizan en el mercado dos estrategias: los ecualizadores adaptativos lineales con secuencia de entrenamiento y los ecualizadores ciegos.

2.1.3.1 Ecualizadores con secuencia de entrenamiento

La estrategia más utilizada, sobre todo en sistemas complejos, son los ecualizadores con secuencia de entrenamiento. Estos ecualizadores necesitan una secuencia periódica conocida para poder hallar la distorsión que produce el canal en la señal, con el fin de compensarla.

2.1.3.1.1 LMS

Dentro de los ecualizadores con secuencia de entrenamiento, el más utilizado es el LMS (*Least Mean Square*) debido tanto a su simplicidad matemática, puesto que no requiere medir las funciones de correlación ni invertir las matrices de auto-correlación, como a las buenas propiedades que presenta. La estructura de un filtro adaptativo con secuencia de entrenamiento es la siguiente:

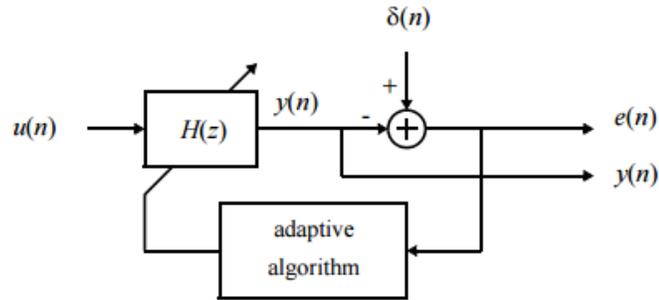


Figura 2.9: Estructura de un filtro adaptativo

donde:

- $u(n)$ y $y(n)$ son la entrada y la salida, respectivamente.
- $\delta(n)$ y $e(n)$ son la señal de entrenamiento y el vector de error.

El ecualizador LMS simplifica el cómputo del error, disminuyendo la complejidad numérica al evitar calcular el error cuadrático, con la siguiente ecuación:

$$e(n) = \delta(n) - y(n)$$

Si llamamos $w(n)$ a los coeficientes del filtro, la ecuación que rige el algoritmo adaptativo del LMS es la siguiente:

$$w(n + 1) = w(n) + 2 \cdot \mu \cdot e(n) \cdot u(n) \cdot w(n)$$

Desde un punto de vista puramente matemático, el LMS es un algoritmo de descenso por gradiente, que intenta buscar el mínimo de una superficie multidimensional (el error obtenido para determinados pesos) dando saltos de tamaño μ , siendo μ el paso del algoritmo.

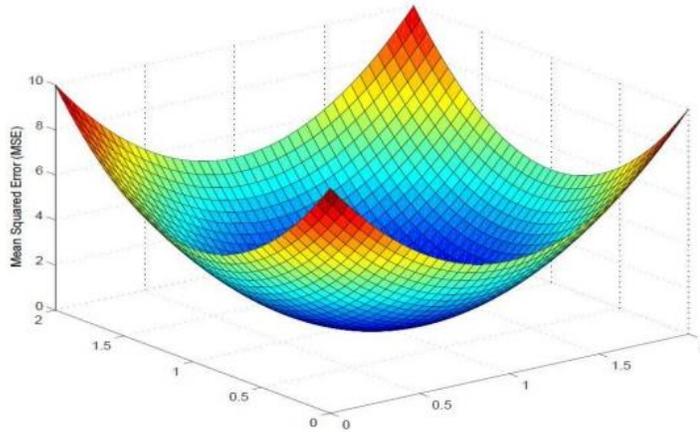


Figura 2.10: Superficie de coste del LMS

En resumen, el ecualizador LMS presenta las siguientes propiedades:

- Fácil implementación y baja carga ($2M+1$ multiplicaciones y $2M$ sumas por iteración)
- Realiza un movimiento aleatorio alrededor de la solución óptima, alejándose como mucho μ de ella.
- Tiene una velocidad de convergencia muy buena.
- Para μ pequeños (menores que el doble de la inversa de la potencia de entrada), es incondicionalmente estable si suponemos i.i.d. entre señales.

En contraposición a sus numerosas ventajas, este ecualizador adolece de dos desventajas enormes. Por un lado, requiere enviar secuencias de entrenamiento periódicas para ajustarlo, perdiendo un alto porcentaje de eficiencia del canal; y, por el otro, un fading en mitad de la secuencia puede impedir la recepción de una gran cantidad de información.

2.1.3.1.2 Ecualizador RLS

Frente al ecualizador LMS, que se rige por un algoritmo lineal adaptativo de descenso por gradiente estocástico, el ecualizador RLS es un algoritmo recursivo de descenso por gradiente determinístico que, a cambio de una complejidad computacional más grande, $4M^2+4M$ multiplicaciones y $3M^2+M-1$ sumas por iteración, asegura la convergencia del mismo.

La actualización de los coeficientes del filtro sigue la siguiente ecuación:

$$\hat{\underline{R}}_{n+1}^{-1} = \alpha^{-1} (\hat{\underline{R}}_n^{-1} - \underline{k}_{n+1} \underline{X}_{n+1}^H \hat{\underline{R}}_n^{-1})$$

siendo \underline{k}_{n+1} :

$$\underline{k}_{n+1} = (\alpha^{-1} \hat{\underline{R}}_n^{-1} \underline{X}_{n+1}) (I + \alpha^{-1} \underline{X}_{n+1}^H \hat{\underline{R}}_n^{-1} \underline{X}_{n+1})^{-1}$$

Tiene, además, la particularidad de utilizar la información de todas las muestras de entrada, aunque aplicando un factor de olvido (α), tal y como se aprecia en la ecuación superior. A mayor valor de esa constante, más memoria tendrá el algoritmo y mejor funcionará; por el contrario, será menos sensible a los cambios bruscos del canal.

La mayor ventaja que ofrece el RLS frente al LMS es su velocidad de convergencia, varios órdenes de magnitud mayor. No obstante, en determinadas situaciones puede presentar inestabilidad numérica.

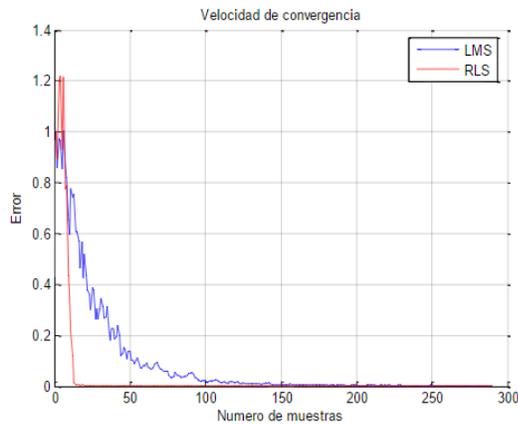


Figura 2.11: Velocidad de convergencia del LMS vs RLS

2.1.3.2 Ecualesadores ciegos

Los ecualizadores ciegos son aquellos en los que no se dispone de ningún dato del transmisor, aunque sí que conocen información estadística de la modulación empleada. La velocidad de convergencia que presentan es varios órdenes menor que la de los ecualizadores LMS, pero a cambio, ofrecen otras características que los hacen enormemente utilizados en el mercado.

Han sido investigados durante muchos años, habiendo en la actualidad gran cantidad de algoritmos para distintas aplicaciones. Con el fin de proporcionar una base adecuada, es

necesario conocer los tres más comunes: el GSA (*Generalized Sato Algorithm*), el MMA (*Multimodulus Algorithm*) y el CMA (*Constant Module Algorithm*), objetivo de este proyecto.

2.1.3.2.1 GSA

Desarrollado por Sato en 1975, fue el primer ecualizador ciego para señales multinivel. El investigador se encontraba trabajando en una determinada aplicación en la que necesitaba conectar y desconectar rutas, y se tuvo que enfrentar al hecho de que se producían cambios drásticos del canal. Así surgió el GSA. Su función de error es la siguiente:

$$J_{sato} = E\{y(k) - \gamma \cdot \text{sign}(y(k)^2)\}$$

donde $y(k)$ es la salida del ecualizador, sign es la función signo y:

$$\gamma = \frac{E[x(k)^2]}{E[|x(k)|]}$$

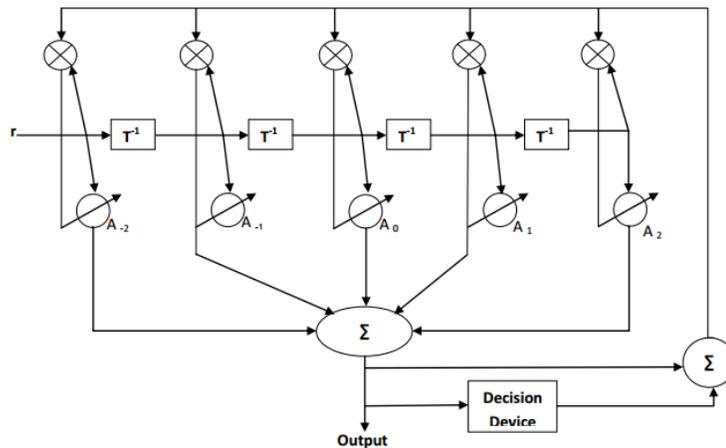


Figura 2.12: Ecualizador GSA

La velocidad de convergencia de este algoritmo es muy baja, y el ruido residual que deja una vez ha convergido, muy alto. Por todo ello no se usa en la actualidad, aunque casi todos los algoritmos ciegos se han desarrollado partiendo del mismo.

2.1.3.2.2 CMA

Basándose en el algoritmo GSA, Dominique Godard [1] extendió los equalizadores ciegos a señales de valores complejos. Una particularización de dicho algoritmo es el CMA.

El CMA es un algoritmo muy utilizado en la actualidad, tanto por su simplicidad matemática como por las prestaciones que ofrece. La ecuación de error que rige su funcionamiento es la siguiente:

$$e(n) = |y(n)|^2 - A^2$$

siendo A el nivel de señal requerido, es decir, el módulo de la constelación. Por lo tanto, el CMA optimiza los coeficientes del filtro para minimizar la siguiente función de coste:

$$J_{cm} = E\{|A^2 - y^2(n)|^2\}$$

Debido a dicha ecuación, el algoritmo puede converger a una solución subóptima, además de presentar una rotación de fase; es decir, la constelación puede salir rotada respecto a la transmitida. Para eliminar este problema se puede implementar un bloque de recuperación de fase o, mejor aún, usar una codificación diferencial. Otra desventaja que presenta es que permanece un error residual incluso después de converger; para reducirlo se suele usar una técnica que minimiza el paso gradualmente.

Por último, la ecuación de actualización de los pesos del filtro es la siguiente:

$$w(n+1) = w(n) - \mu \cdot e(n) \cdot y^*(n) \cdot x(n)$$

En general, el CMA se suele utilizar para modulaciones QAM, que presentan un módulo constante para todos sus símbolos; en caso contrario, existe una variante, el algoritmo MMA, que ofrece buenos resultados para modulaciones multimódulo más complejas.

El estudio matemático del algoritmo como problema de optimización queda fuera de los límites de este trabajo. En la bibliografía se encuentran citados distintos estudios sobre su convergencia; Shalvi y Weinstein [2] determinaron que el algoritmo CMA, a diferencia del LMS, no siempre converge para señales gaussianas.

2.1.3.2.3 MMA

Para modulaciones con símbolos con módulos distintos, el CMA es insuficiente. Nace así el algoritmo MMA, que divide la constelación en regiones, tomando como dato estadístico a priori la dispersión de la constelación:

$$R_m^2 = \frac{E\{a(n)^4\}}{E\{a(n)^2\}}$$

La función de coste que minimiza el MMA de orden dos es la siguiente:

$$J_{mma} = E\{(y(n)^2 - R_m^2)^2 + (\hat{y}(n)^2 - R_m^2)^2\}$$

En la práctica, el MMA de segundo orden es el que mejor compromiso entre rendimiento y complejidad ofrece. Frente al CMA, que es una función de coste bidimensional, el MMA es la suma de dos funciones de coste unidimensionales.

Para constelaciones con gran densidad de símbolos (64 QAM, 256QAM), prácticamente cualquier ecualizador ciego ofrece un bajo rendimiento. Sin embargo, existen versiones avanzadas, por ejemplo, el GMMA, con las que sí se pueden obtener buenos resultados.

2.1.4. Conclusión

Como se ha adelantado en el apartado anterior, la necesidad de implementar un ecualizador es debida a la propagación multicamino que sufre la señal al atravesar el canal. En general, los ecualizadores requieren de necesidades de cómputo relativamente altas que, en la actualidad, son fácilmente alcanzables gracias a las FPGAs. Sin embargo, el procedimiento para realizar operaciones con la señal recibida dentro de las mismas no es sencillo: el conjunto de técnicas y métodos para tratar adecuadamente una señal se conoce como procesamiento de señal.

2.2. Procesado de señal

La mejor manera de muestrear y tratar una señal analógica compleja a nivel digital es usar la representación I/Q, que se obtiene muestreando una señal con dos osciladores desfasados 90°. Dicha representación permite plasmar en coordenadas polares el estado de una onda sinusoidal en un instante concreto, pues la I (*In-phase*) determinará el ángulo de fase y la Q (*Quadrature*) la magnitud.

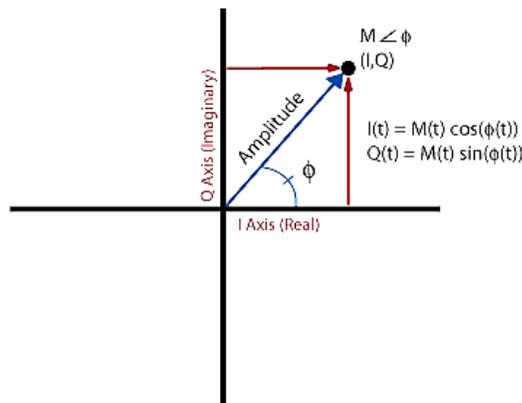


Figura 2.13: Representación IQ

Se puede establecer un paralelismo matemático entre un valor I/Q y un número complejo, permitiendo realizar todas las operaciones aplicables a estos últimos a las señales muestreadas:

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

$$\phi = \tan^{-1}\left(\frac{Q}{I}\right)$$

$$I = A \cdot \cos(\phi)$$

$$Q = A \cdot \sin(\phi)$$

El procesamiento de una señal I/Q se conoce como tratamiento en banda base y presenta varias ventajas respecto al procesado de señales reales:

- Debido a que la señal en dos dimensiones (I/Q) no tiene una frecuencia fija, se puede convertir en una señal en RF de una sola dimensión sin ninguna pérdida de información:

$$Portadora\ Modulada = I \cdot \cos(2\pi ft) + Q \cdot \sin(2\pi ft)$$

- Pese a que los osciladores del receptor y del transmisor no estén sincronizados, la amplitud y el ángulo relativo entre I y Q siempre son correctos.
- Debido a que la portadora tiene una frecuencia mucho más alta que la señal en banda base, no hay que preocuparse de las frecuencias negativas, pues al mezclar siempre se generará una portadora positiva.
- Una señal I/Q permite distinguir una frecuencia positiva de una negativa, distinción irrealizable con señales reales. A la hora de multiplicar los dos tipos de señales, esta propiedad es muy importante:

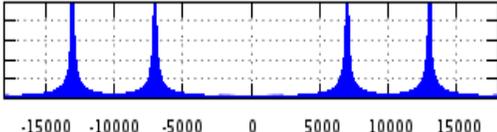
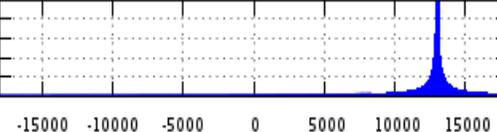
Señal real	Señal I/Q
$\pm f_1 \otimes \pm f_2 = (\pm)f_1 \pm f_2$	$\pm f_1 \otimes \pm f_2 = f_1 + f_2$
	

Tabla 2.1: Multiplicaciones de señales reales y señales en IQ

La representación IQ, en resumen, no es sólo una construcción matemática puesto que se puede considerar la señal de RF como una señal compleja, y la señal real como una simple proyección de la misma.

En este apartado se ha realizado un breve análisis sobre cómo representar una señal y sobre las posibles operaciones que se pueden realizar con ella. Sin embargo, la implementación en hardware de dichas operaciones puede ser en muchos casos tan costosa que resulte inaceptable.

2.3. FPGA

Pese a que el estudio exhaustivo de la arquitectura de una FPGA y los componentes que la forman no sea el objeto de este trabajo, es necesario proporcionar unos conocimientos básicos del tema para poder exponer después los límites con los que nos vamos a encontrar a la hora de implementar el ecualizador.

Una FPGA es un circuito integrado configurable, el cual contiene bloques funcionales que pueden ser conectados unos con otros mediante un lenguaje de descripción de hardware (VHDL, Verilog).

Surge de un proceso natural de convergencia entre los antiguos PLD (*Programmable Logic Devices*) y los circuitos programables de aplicación específica (*Application-Specific Integrated Circuit*), con el fin de intentar unificar los puntos fuertes de ambas tecnologías: lógica reprogramable y una importante potencia de cálculo. Fueron introducidas por Xilinx en el mercado en 1985 y apenas contaban con 9000 puertas lógicas, nada comparable con las millones disponibles en la actualidad.

Este enorme crecimiento sólo ha sido posible gracias a un increíble descenso del tamaño de los transistores que componen las FPGAs, que ha evolucionado desde los 2000 nm que medían en 1985 hasta los actuales 16 nm de la serie UltraScale de Xilinx. Para hacerse una idea del tamaño del que estamos hablando, un átomo con su nube de electrones mide, de media, 0.1 nm.

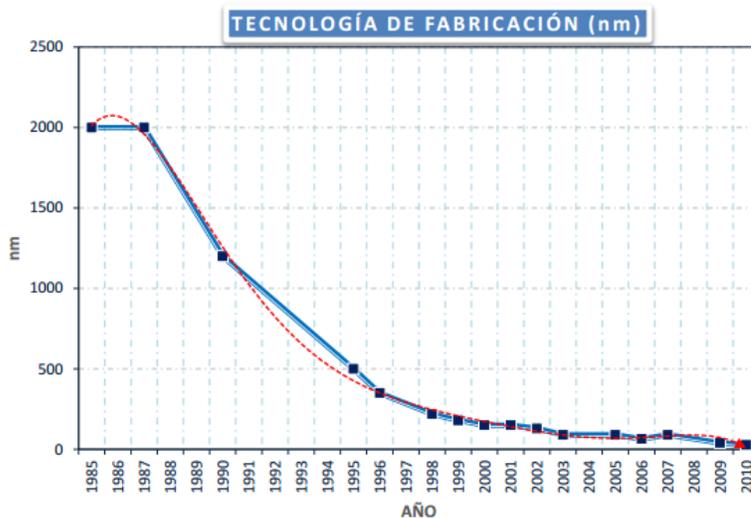


Figura 2.14: Evolución de la tecnología de fabricación

En general, la arquitectura de una FPAG se puede dividir en tres partes: una parte lógica, las interconexiones programables que hay entre los distintos bloques lógicos y unos módulos de entrada y salida.

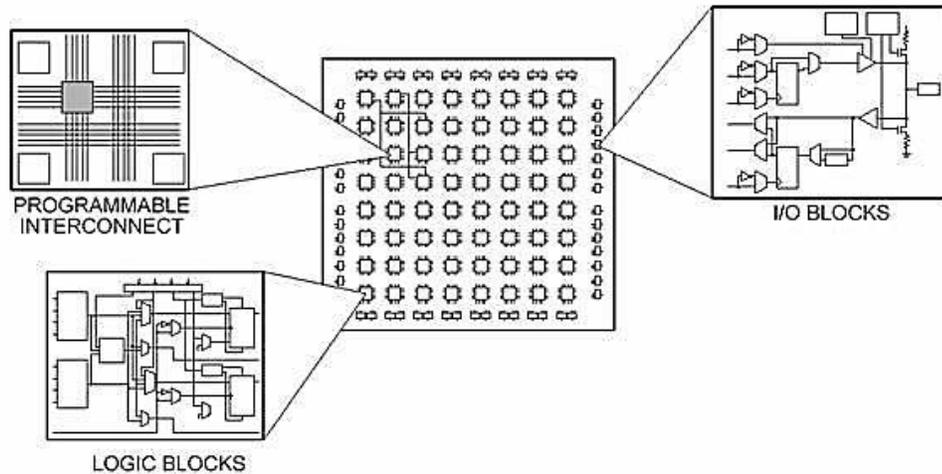


Figura 2.15: Arquitectura de una FPGA

Las FPGAs disponen de gran cantidad de tipos de bloques, muchos de ellos de gran complejidad y aplicación específica, como *transceivers* de alta velocidad, controladores de memoria o bloques RAM. Sin embargo, la mayor funcionalidad de los circuitos se logra a través de pequeños grupos de puertas lógicas, llamadas CLB (*Configurable Logic Block*). En cada grupo hay varias tablas de búsqueda, slices, flip-flops, multiplexadores... La velocidad, potencia y versatilidad a la hora de realizar operaciones booleanas sencillas es enorme.

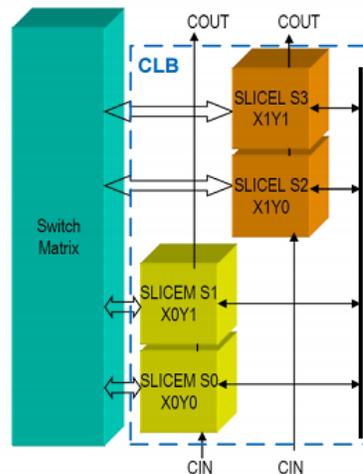


Figura 2.16: CLB de una Spartan6

Sin embargo, cuando se hace necesario realizar operaciones matemáticas más complejas, como pueden ser multiplicaciones con punto fijo, los CLBs se muestran muy limitados. Para esos cálculos, todas las FPGAs incluyen unos bloques llamados DSPs, que contienen, por ejemplo, en el caso de la Spartan6, dos sumadores y un multiplicador de 18 bits, permitiendo realizar una multiplicación con más facilidad.

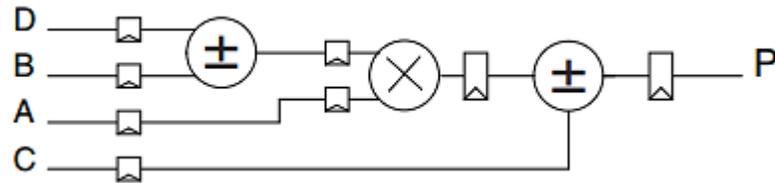


Figura 2.17: DSP de Xilinx

Para la implementación del ecualizador es necesario realizar gran cantidad de multiplicaciones de números complejos; la manera óptima de lograrlo con el mínimo de DSPs es la siguiente:

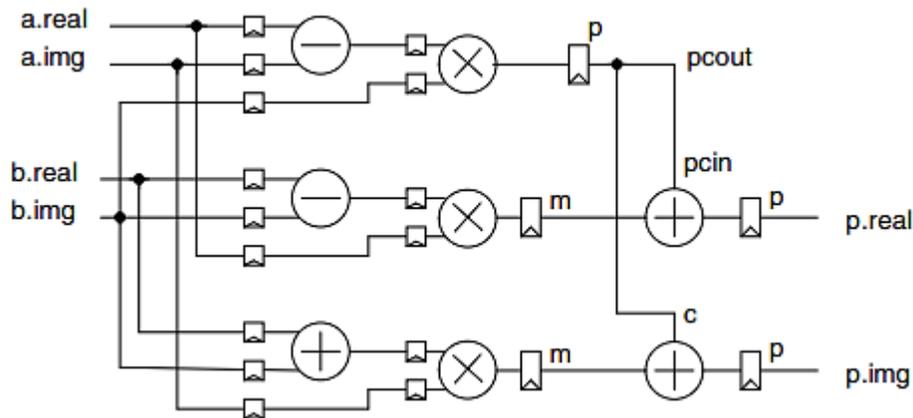


Figura 2.18: Multiplicación eficiente de dos números complejos

La estructura anterior consigue realizar una multiplicación de números complejos en un ciclo de reloj utilizando tres DSPs. Se representaría con la siguiente fórmula:

$$(a + bi) \cdot (c + di) = [d \cdot (a - b) + a \cdot (c - d)] + [b \cdot (c + d) + d \cdot (a - b)] i$$

2.3.1. Limitaciones de una FPGA

En el apartado anterior se ha explicado la variedad de recursos de los que dispone una FPGA. Sin embargo, rara vez el número de recursos limita una implementación; el factor

más influyente suele ser las restricciones de tiempo que se dan a la hora de realizar el enrutado de los componentes.

Para entender este problema, primero es necesario comprender el proceso que convierte un código VHDL en una configuración de una FPGA, y que consta de cuatro etapas:

- **Síntesis**

Es el proceso en el que el compilador interpreta el código con el fin de optimizarlo al máximo posible para reducir los recursos a utilizar.

- **Traducción**

En este procedimiento, el compilador lee los resultados de la síntesis y los traduce a bloques de la FPGA.

- **Mapeado**

Este paso consiste en elegir qué recursos de la FPGA se van a utilizar. Es decir, se intentará que las posiciones de los bloques más críticos se encuentren en los lugares más accesibles, que las operaciones concatenadas se realicen con bloques colindantes, etc.

- **Enrutado**

Es el proceso más restrictivo, puesto que debe poder realizar todas las conexiones entre los bloques que se han situado durante el mapeado con las limitaciones de tiempo que le haya fijado el usuario o el reloj. En caso de no lograrlo, puede que los datos que deberían estar disponibles en un ciclo de reloj no lo estén hasta el siguiente, causando un mal funcionamiento de todo el sistema. Este es un error crítico conocido como *timing constraint*.

Teniendo en cuenta todo lo anterior, se pueden sacar las siguientes conclusiones que repercuten directamente a la hora de implementar un ecualizador:

- El reloj al que funciona un bloque de código es una decisión crucial: a mayor velocidad, mayor será el número de problemas que surjan.
- Debido a las limitaciones de enrutado de la FPGA, no es posible aprovechar la totalidad de los recursos de los que dispone la misma.
- Realizar varias operaciones concatenadas en el mismo ciclo de reloj dificulta enormemente las tareas de enrutado y de mapeo.

Capítulo 3

Desarrollo realizado

La plataforma en la que se va a integrar y testear el ecualizador funciona sobre una FPGA Spartan-6 de Xilinx. Es una FPGA de 45nm de gama media, que salió al mercado en el año 2007 para aplicaciones con una necesidad de cómputo relativamente alta y coste intermedio. Sus características básicas se pueden observar en la siguiente tabla:

XC6SLX100T	
Puertas lógicas	101261
Memoria (Kb)	4824
DSPs	180
Entradas/salidas	498

Tabla 3.1: Recursos Spartan6 XC6SLX100T

Desde el punto de vista de diseño de hardware en VHDL, hay dos características a priori que debemos conocer a la hora implementar un ecualizador: el número de símbolos por segundo que debe procesar y su modulación. En nuestro sistema se utiliza una tasa de símbolo de 8Msp/s, y la constelación es DQPSK. A partir de estos dos factores, debe seleccionarse una frecuencia de reloj que permita calcular el resultado, es decir, que su tiempo de cálculo sea menor al tiempo transcurrido entre dos símbolos. Dicha decisión será clave a la hora de enrutar el hardware, pues a menor velocidad de reloj, menor cantidad de problemas de *timing constraints* aparecerán. El conjunto de bloques de recepción, conocidos como *datapath*, funciona a 80 MHz, y trabaja con señales IQ de 16 bits de longitud, pero se dispone de otros relojes más pequeños (20, 40, 50 MHz) a los que también puede trabajar el ecualizador.

3.1. Implementación

Las ecuaciones que debemos implementar son las siguientes:

$$y[n] = x[n] \cdot W[n]$$

$$W[n + 1] = W[n] + \mu \cdot e[n] \cdot x^*[n]$$

$$e[n] = y[n] \cdot (Y_c - |y[n]|^2)$$

siendo:

$W[n]$: Pesos del filtro FIR en cada etapa.

μ : Paso del ecualizador.

$e[n]$: Señal de error del ecualizador

γ_c : Módulo al que debe converger el ecualizador.

Para implementar las ecuaciones superiores es necesario realizar dos multiplicaciones complejas por cada etapa. Con el fin de evitar duplicar los DSPs necesarios, se diseña una arquitectura que reutiliza los multiplicadores:

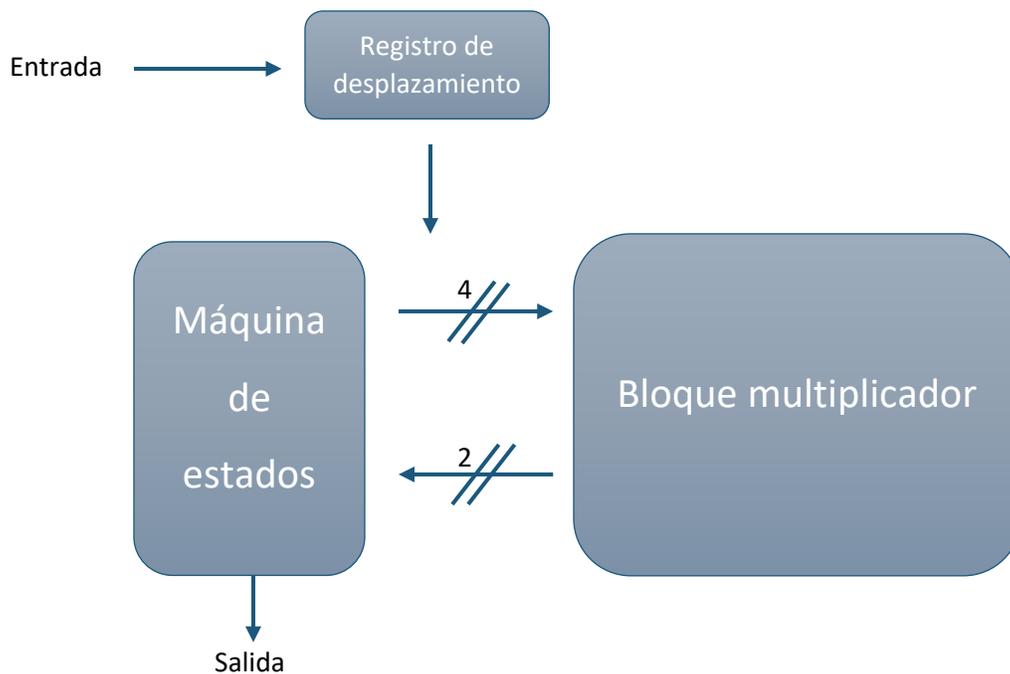


Figura 3.1: Esquema general del ecualizador

El bloque multiplicador tiene como entradas dos señales complejas, y obtiene como salida el resultado de la siguiente fórmula:

$$(a + bi) \cdot (c + di) = [d \cdot (a - b) + a \cdot (c - d)] + [b \cdot (c + d) + d \cdot (a - b)]i$$

Es decir, multiplica las señales y utiliza para ello cinco sumadores y dos multiplicadores. El resultado se obtiene con una latencia de 2 ciclos de reloj.

La máquina de estados tiene tres etapas que van cambiando en cada ciclo de reloj, y que se pueden resumir en el siguiente esquema:

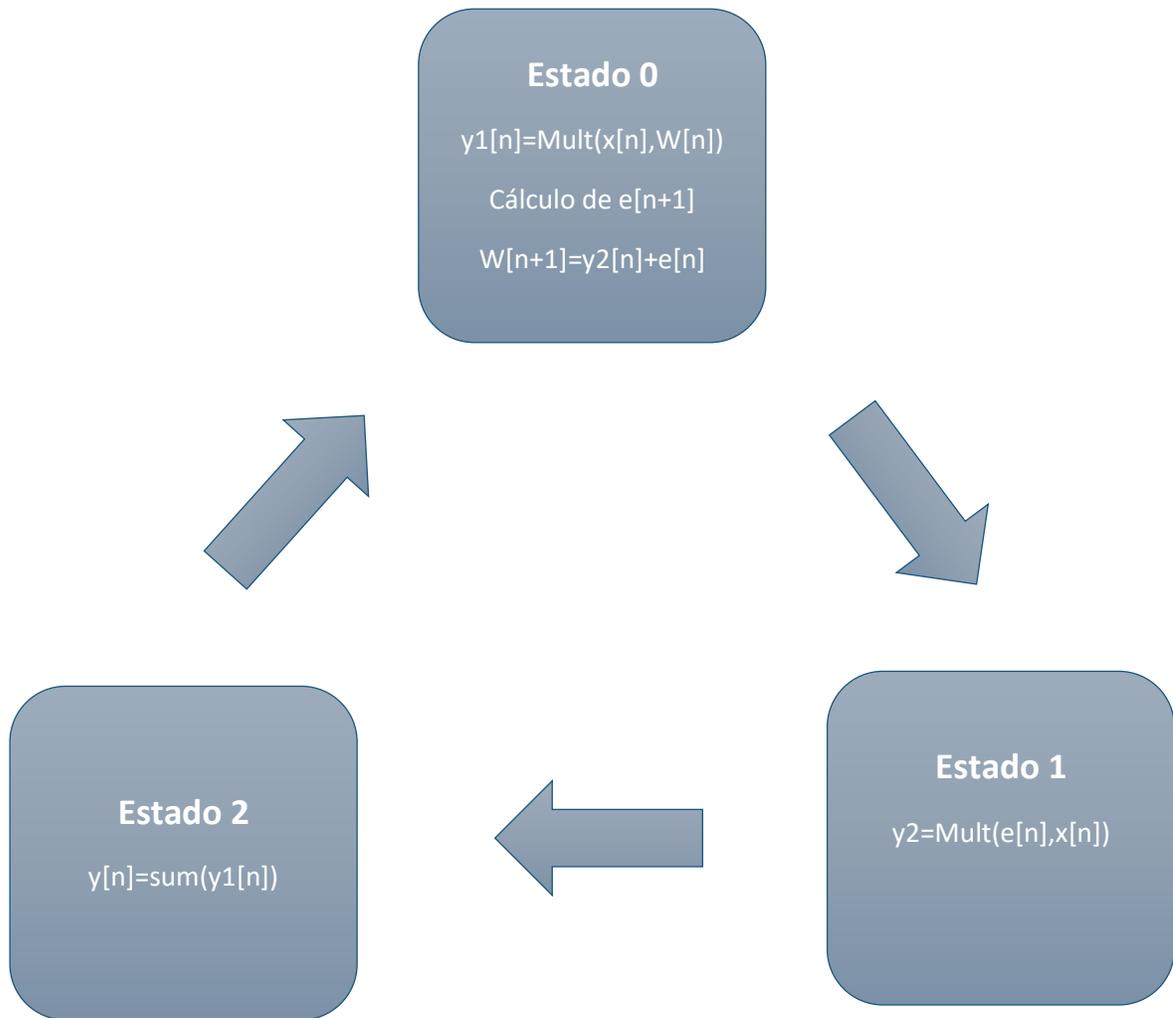


Figura 3.2: Máquina de estados del ecualizador

Para calcular el error $e[n]$, las operaciones a realizar son las siguientes:

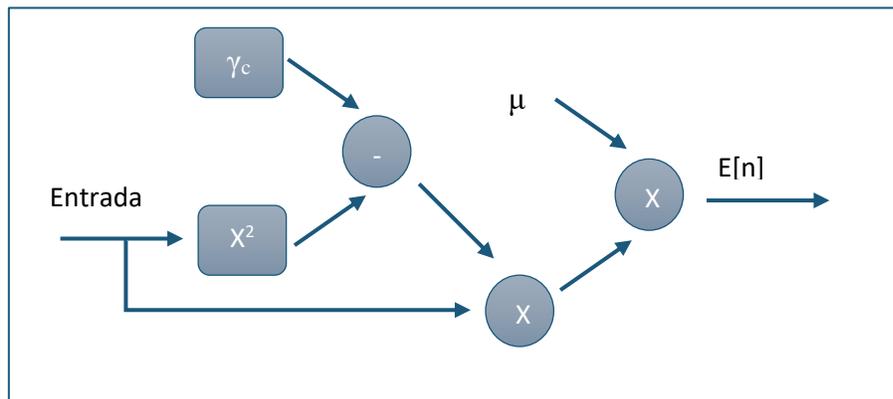


Figura 3.3: Cálculo del error del ecualizador

El número de etapas y la constante γ_c , es decir, el módulo de la señal a la que debe converger el ecualizador, son configurables externamente. El bloque tiene las siguientes entradas:

Señal	Tipo	Descripción
Clk	Std_logic	Reloj
Reset	Std_logic	Reset
Mu	Std_logic_vector(17)	U[17 17]. Paso del ecualizador
Gamma	Std_logic_vector(2)	U[2 2]. Selección de módulo
In_Valid_data	Std_logic	Dato válido, pero no es símbolo
In_Valid_symbol	Std_logic	Dato válido y símbolo
In_i	Std_logic_vector(17)	S[17 16]. Entrada I
In_q	Std_logic_vector(17)	S[17 16]. Entrada Q
Out_Valid_data	Std_logic	Dato válido, pero no es símbolo
Out_Valid_symbol	Std_logic	Dato válido y símbolo
Out_i	Std_logic_vector(17)	S[17 16]. Salida I
Out_q	Std_logic_vector(17)	S[17 16]. Salida Q

Tabla 3.2: Entradas y salidas del ecualizador

Para 16 etapas utiliza los siguientes recursos:

Recurso	Utilizados	Disponibles
Registros	1047	126576
LUTs	4453	63288
Flip-flops	4907	-
DSPs	86	180

Tabla 3.3: Recursos del ecualizador para 16 etapas

Con la implementación superior se consigue compilar correctamente todo el firmware utilizando un reloj de 20 MHz. Se obtienen las siguientes figuras de mérito:

Latencia	1 símbolo
Etapas	4-32
Símbolos/s Máximos	0.5 *Frecuencia

Tabla 3.4: Figuras de mérito del ecualizador

A primera vista, la utilización de un reloj de 20 MHz puede sorprender. El motivo es que la FPGA ya utiliza para otros bloques dicho reloj, y era más eficiente utilizarlo que crear uno nuevo de 16 MHz, que es el mínimo necesario para procesar 8Msps con esta implementación.

3.2. Cross Domain Crossing

Para conseguir que un bloque que funciona a 20 MHz procese adecuadamente una señal asociada a un reloj de 80 MHz, es necesario integrar dos bloques de entrada y salida que adapten ambas señales. Este problema, conocido en la literatura como CDC (*Cross Domain Crossing*) o, más comúnmente, problema de metaestabilidad, ocurre en circuitos síncronos cuando una señal pasa de un determinado dominio de reloj a otro diferente.

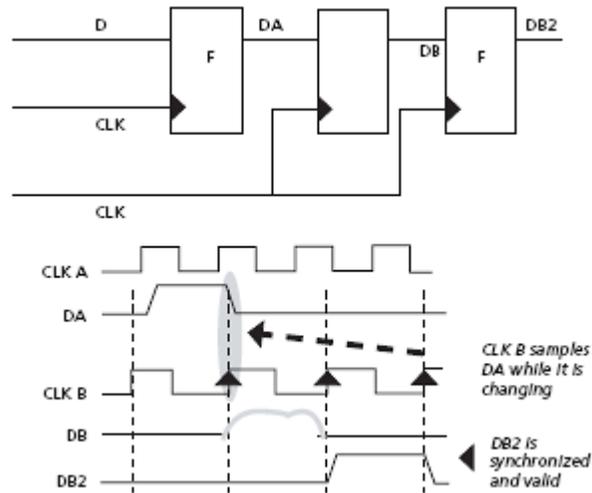


Figura 3.4: Problema al cambiar de dominio de reloj

Las diferentes soluciones que se pueden implementar para superar este problema están ampliamente tratadas en la bibliografía [3][4][5]. El procedimiento más común suele consistir en registrar una vez la señal con el primer reloj y dos veces con el segundo, obteniendo así una probabilidad de error muy baja.

Como esta solución se demostró insuficiente, se decidió implementar dos FIFOs de doble reloj, que, además, permitían adaptar la señal a una tasa constante.

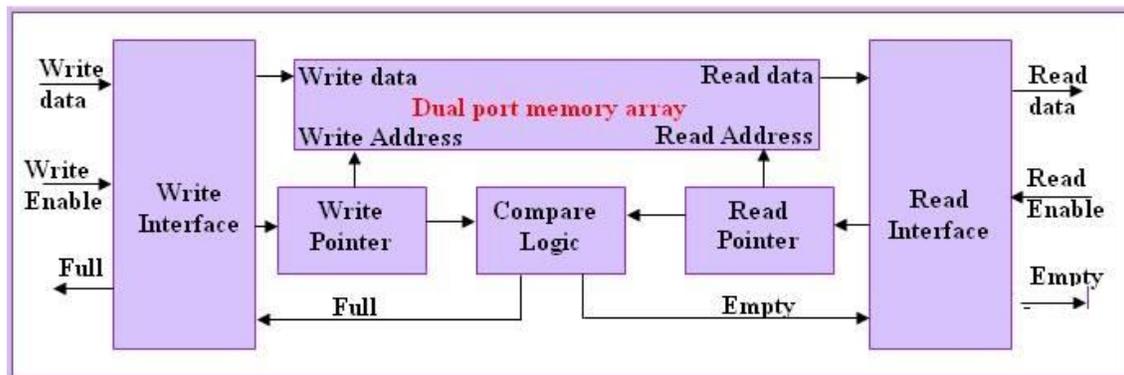


Figura 3.5: CDC con memoria de doble reloj

A nivel de relojes, el diagrama de bloques del ecualizador quedaría de la siguiente manera:

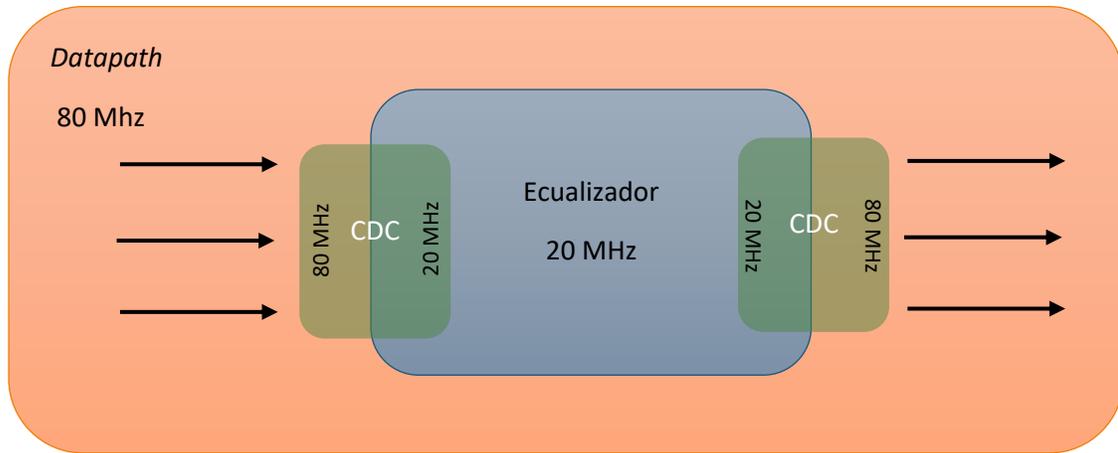


Figura 3.6: Esquema del ecualizador a nivel frecuencial

Capítulo 4

Depuración, verificación y análisis

El ecualizador, programado en VHDL, ha sido simulado y verificado antes de integrarlo en la plataforma con el software Simulink. Para ello, primero se programó un ecualizador CMA en MATLAB y, a continuación, un script que ejecutaba la simulación de Simulink y comparaba ambos resultados. El siguiente paso consistió en enfrentar el bloque a una señal real capturada en la plataforma y verificar su correcto funcionamiento. Por último, se modelaron distintos canales y se estudió el comportamiento del ecualizador en cada uno de ellos.

4.1. Depuración y validación

Con el fin de depurar correctamente el ecualizador durante el desarrollo, se programó un ecualizador CMA y una función de depuración, ambas en MATLAB.

El diagrama de bloques de la función programada en MATLAB es el siguiente:

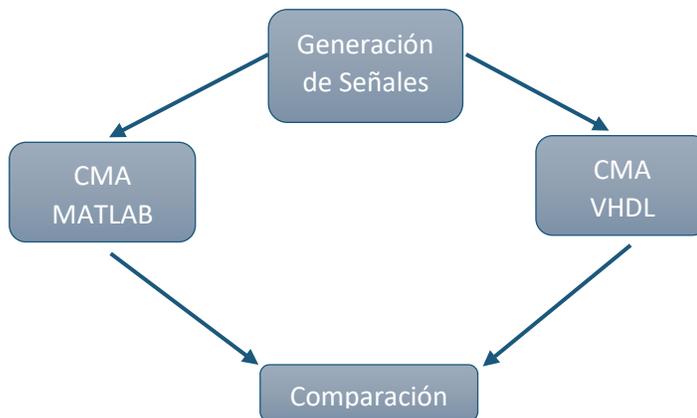


Figura 4.1: Test con Matlab

El script CMA de Matlab ecualiza una entrada en IQ para un módulo seleccionable. Para sacarle el máximo partido, se ha diseñado otro script que, por encima, genera una señal IQ QPSK y compara los resultados que obtiene de la función ecualizador superior y de una simulación del bloque VHDL.

Una vez validado a nivel de funcionamiento, se realizaron unas capturas reales en el punto de la cadena de recepción donde iba a ir insertado el ecualizador y se validó su correcto

funcionamiento. El proceso de depuración y validación que se llevó a cabo se muestra en el siguiente esquema:

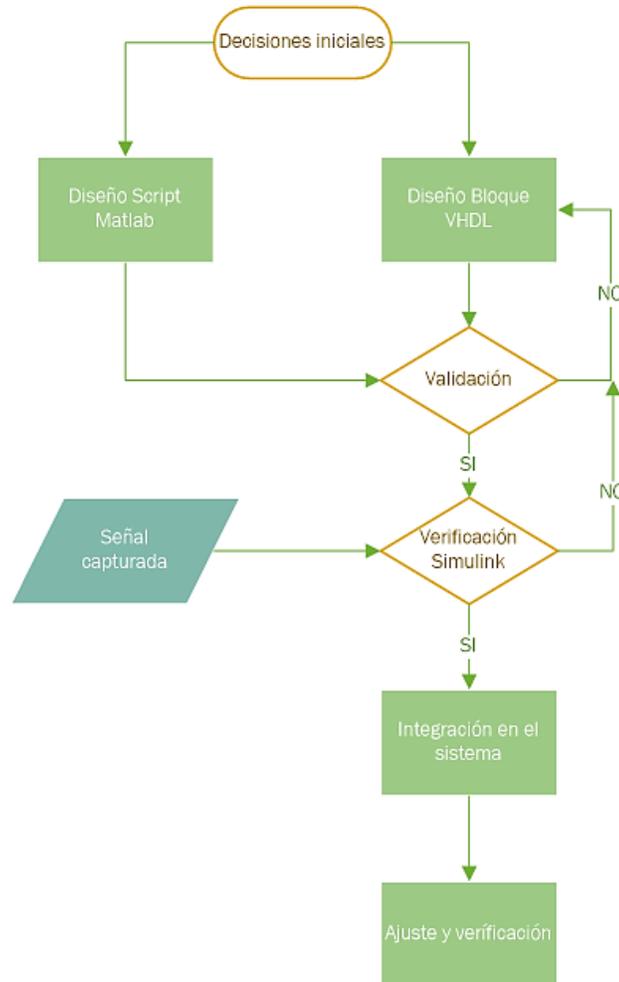


Figura 4.2: Proceso de verificación del ecualizador

Una vez testeados en Simulink los bloques tanto de adaptación como del propio ecualizador, el último paso consistió en integrarlos en el sistema. Y fue precisamente en este punto en el que se pudieron apreciar las enormes diferencias existentes entre la simulación y el funcionamiento real: problemas de metaestabilidad, valores que no eran válidos en los instantes en los que se utilizaban y resultados incongruentes, que fueron resueltos usando las herramientas que previamente se habían desarrollado.

4.2. Verificación

Una vez que el ecualizador ya se encontraba perfectamente integrado, el siguiente paso fue verificar que su comportamiento era el adecuado, acorde con la literatura y con el estudiado con anterioridad en el script CMA de MATLAB para todos los escenarios posibles.

Para ello, se realizó un pequeño estudio general para, posteriormente, ir enfrentando el ecualizador en simulación a distintos canales.

4.2.1. Análisis teórico de la implementación

4.2.1.1 Pérdida de eficiencia debido a la cuantificación

El tratamiento de una señal analógica a nivel digital debe realizarse a posteriori de un proceso de cuantificación, es decir, es necesario asignar a cada nivel de amplitud un valor discreto. La cuantificación de señales supone, inherentemente, una pérdida de información, que afecta a la SNR de la señal de información.

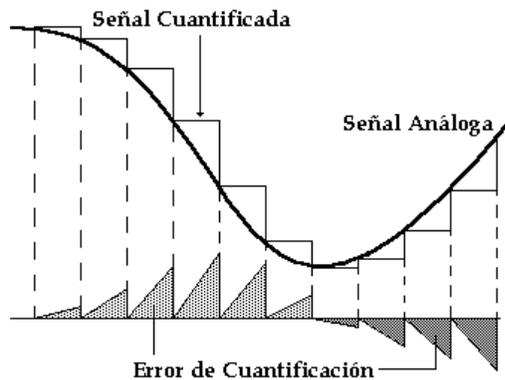


Figura 4.3: El problema de la cuantificación

En el caso del ecualizador el problema es todavía más grave, pues se multiplican dos valores cuantificados, aumentando el error. La siguiente gráfica muestran la diferencia entre el bloque de MATLAB (sin cuantificación) y el bloque VHDL simulado en Simulink (16 bits de cuantificación).

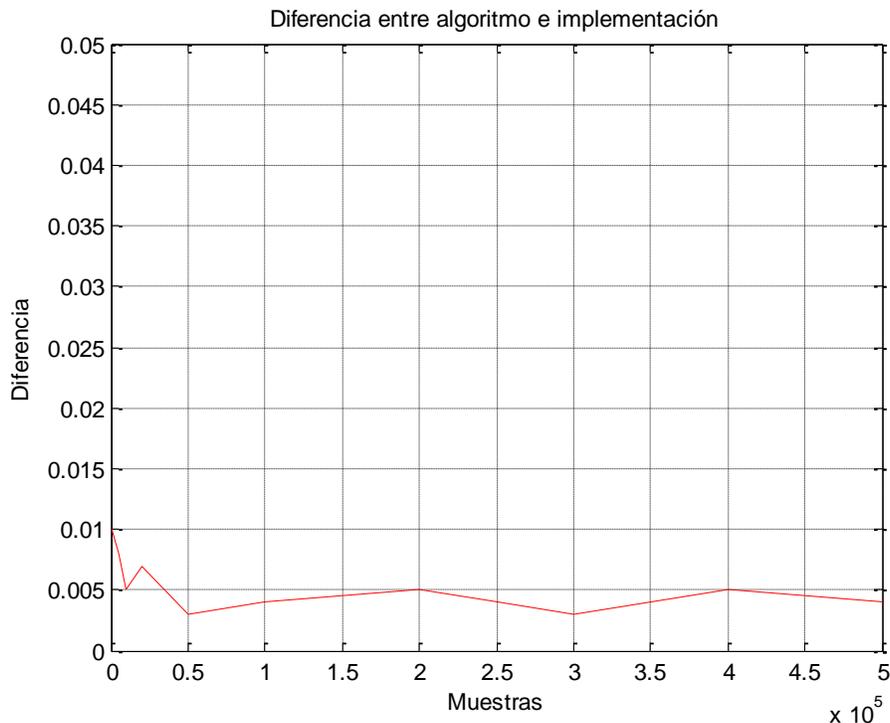


Figura 4.4: Diferencia entre algoritmo e implementación por la cuantificación

Para obtenerla, se ha computado un gran número de veces la diferencia entre las salidas del script de Matlab y el ecualizador en VHDL ante las mismas señales y se ha calculado la media. Se puede concluir que la pérdida por cuantificación es casi despreciable e inferior al ruido de cualquier receptor.

4.2.1.2 Velocidad de convergencia

Se conoce como velocidad de convergencia el número de muestras necesario para que el ecualizador llegue a una solución óptima. Es un factor crítico dentro del ecualizador y depende tanto del paso del ecualizador como del canal al que se enfrente.

Dentro del proceso de convergencia propiamente dicho, se debe diferenciar entre los momentos en los que el ecualizador trabaja como AGC (*Automatic Gain Control*) y cuando realmente ecualiza. Al principio, lo único que hace el sistema es multiplicar por un factor la señal entrante, malgastando un número de muestras considerable en un procedimiento completamente ajeno a su verdadero propósito.

Con el fin de estudiar la velocidad de convergencia, el módulo de las señales de entrada antes de enfrentarlas al canal se ha fijado a un valor cercano al módulo al que debe converger el algoritmo. Para estimar de manera correcta lo acertada que es la solución del

ecualizador, utilizamos como variable la diferencia, en media, del módulo de los símbolos de salida respecto al módulo óptimo.

Las pruebas se han realizado pasando la señal por varios canales Rice, con una respuesta al impulso similar a la de la siguiente gráfica:

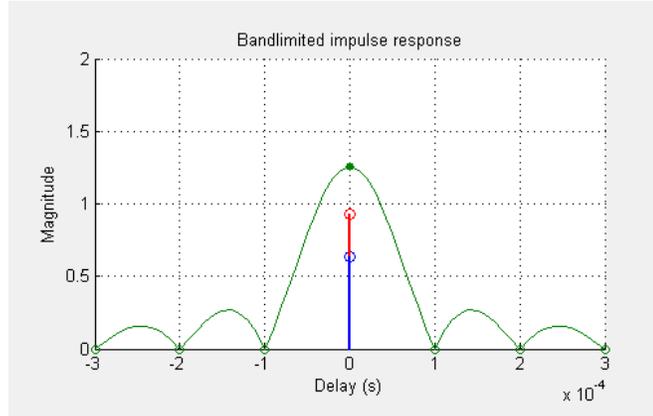


Figura 4.5: Respuesta al impulso de un canal normal

Los resultados obtenidos se presentan a continuación:

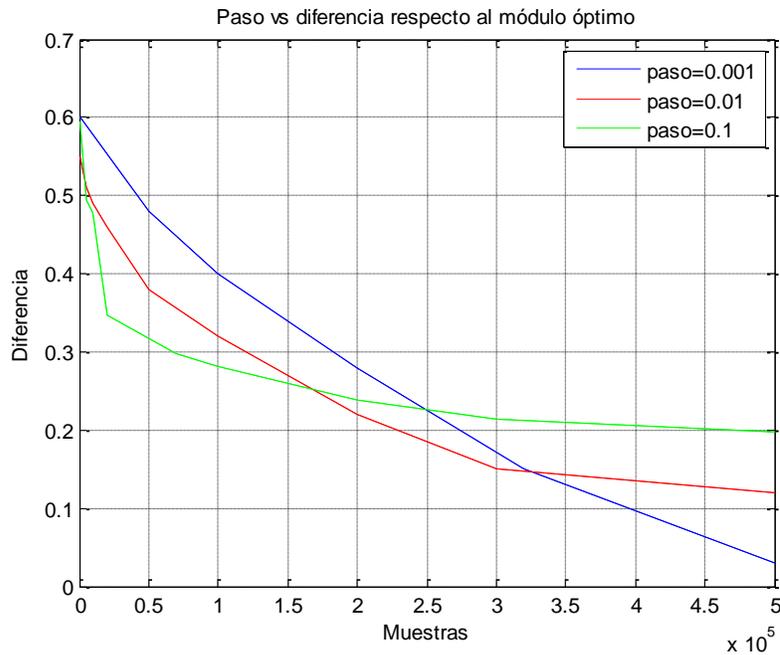


Figura 4.6: Velocidad de convergencia en función del paso

La conclusión ya se adelantó en el marco teórico. Un paso grande hace que converja muy rápidamente, pero dejando un error residual enorme. Según se va reduciendo el paso,

ese error residual se va minimizando, a costa de utilizar más muestras para lograr valores asumibles.

4.2.1.3 Soluciones no óptimas

El algoritmo CMA usa el método de descenso estocástico por gradiente para minimizar la diferencia entre los símbolos recibidos y la constelación ejemplar. Sin embargo, una de las debilidades de este método es que puede quedarse bloqueado en algún mínimo local. La mejor manera de entender este problema es dibujando la superficie sobre la que el algoritmo busca la solución. Las imágenes inferiores representan dichas superficies. Para facilitar la comprensión del problema, el ecualizador sólo tendrá dos etapas, quedando, por lo tanto, una superficie en dos dimensiones:

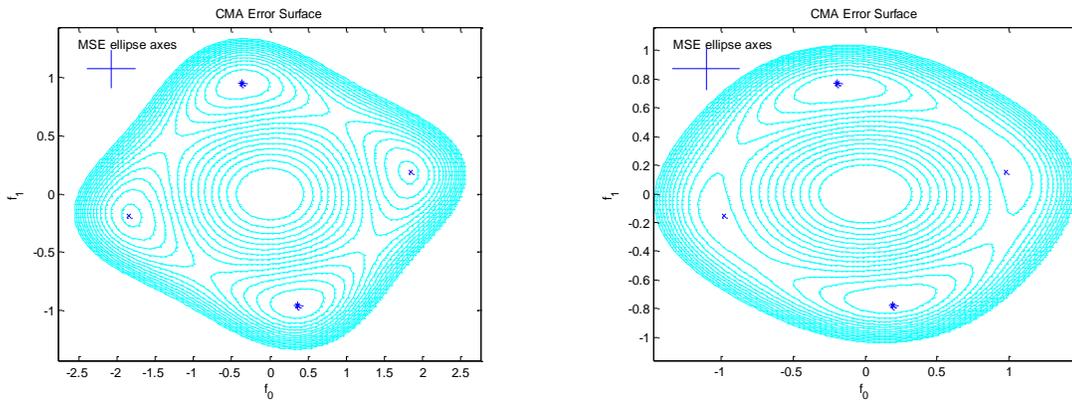


Figura 4.7: Canal $h = [0.2, 0.5, 1, -1]$ para 20 dB y 5 dB de SNR

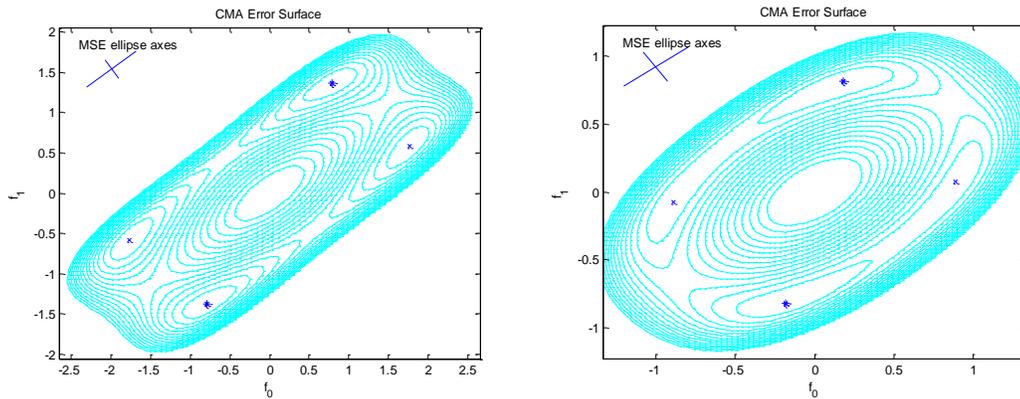


Figura 4.8: Canal $h = [.9, -0.3, .4, -0.7]$ para 60 dB y 5 dB de SNR

Observando las figuras anteriores se puede deducir lo siguiente:

- El ecualizador puede converger a una solución subóptima y permanecer en ella si el paso es muy pequeño.
- Un paso grande implica un error residual de magnitud considerable. Dicho error es producido por la inexactitud que presenta el algoritmo después de converger, debido a que la solución obtenida está constantemente moviéndose estocásticamente alrededor de la solución óptima. Esta es una de las desventajas que presenta el CMA respecto al LMS o RLS en la mayoría de los canales, permitiendo estos últimos alcanzar unos niveles de BER mínimos significativamente menores:

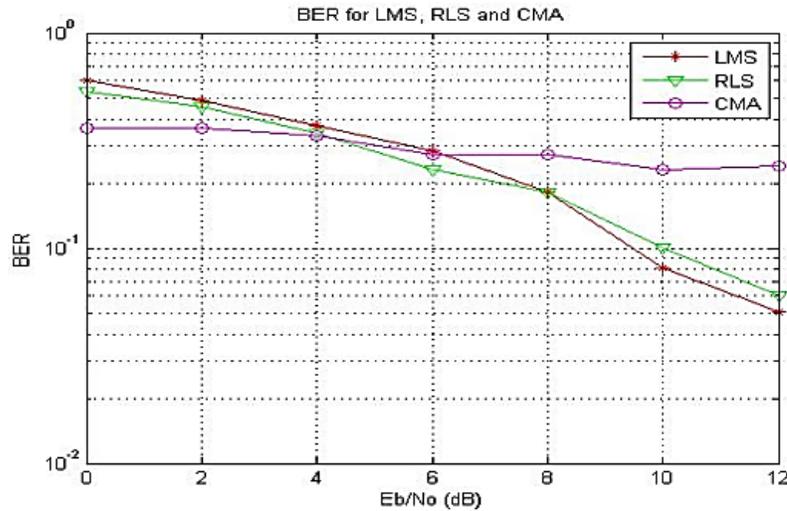


Figura 4.9: BER LMS, RLS, CMA

4.2.2. Simulación del CMA ante distintos canales

Con el fin de comprobar el rendimiento del ecualizador ante distintos escenarios, se ha simulado la cadena de transmisión/recepción completa mediante Simulink.

Los parámetros variables durante el proceso de prueba han sido los siguientes:

- Atenuación: Pérdida de potencia en la señal recibida respecto a la máxima que puede aceptar la cadena de recepción. Simula la distancia del enlace.
- Canal: Respuesta al impulso del medio a testear.
- SNR: Relación señal a ruido de la onda recibida. Se modifica añadiendo ruido gaussiano blanco entre la transmisión y la recepción.
- Paso: Parámetro interno del ecualizador que fija la velocidad de convergencia.

4.2.2.1 Canal común

El propósito de esta prueba es enfrentar el CMA al canal más común que se puede encontrar, un canal sin multitrayecto, que presenta la siguiente respuesta al impulso:

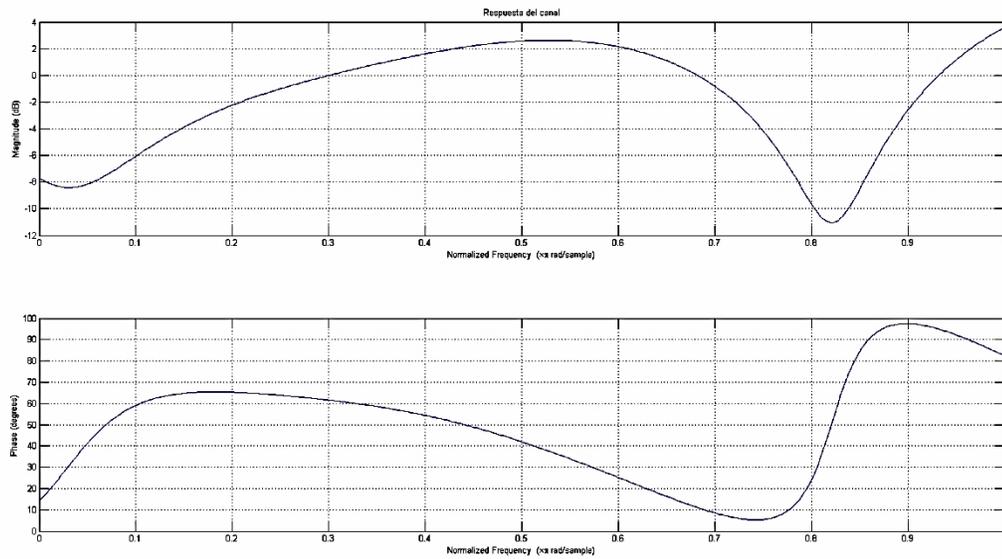


Figura 4.10: Respuesta al impulso de un canal común

Además, fijamos la atenuación en 10 dBs. En la siguiente imagen se puede observar la señal original y la señal degradada:

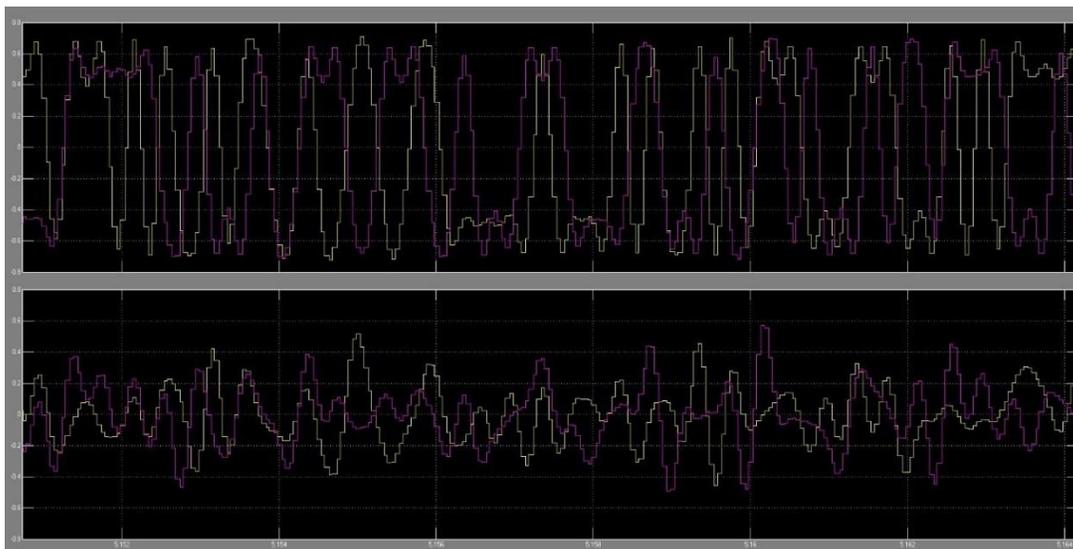


Figura 4.11: Señal antes y después de pasar por el canal

Por último, podemos observar la señal previa y posterior al ecualizador después de 500000 símbolos, con un paso de 10^{-2} :

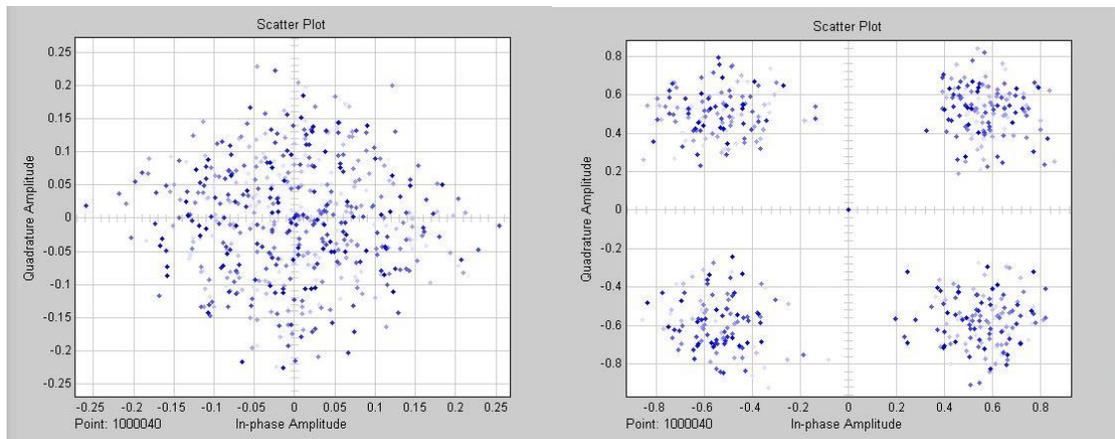


Figura 4.12: Señal antes y después de pasar por el ecualizador

Si manteniendo las mismas condiciones reducimos el paso a 10^{-4} , nos encontraremos con el siguiente resultado:

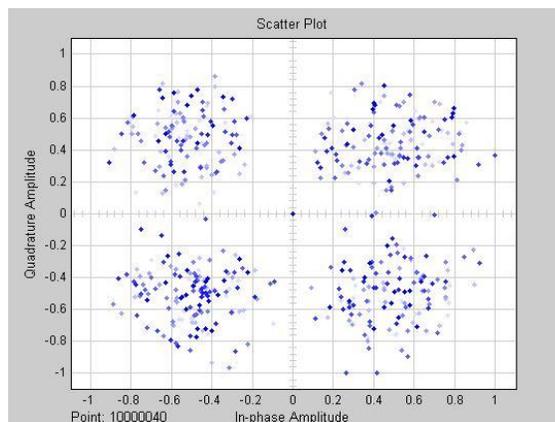


Figura 4.13: Señal ecualizada

El resultado obtenido es sorprendente, ya que, pese haber reducido el paso y entrenado el ecualizador con un número de muestras un orden por encima de los valores de convergencia típicos que da la bibliografía, la solución es peor. Al fijarnos en los coeficientes de la solución del filtro, observamos que el ecualizador no está trabajando como ecualizador, sino como control de ganancia digital:

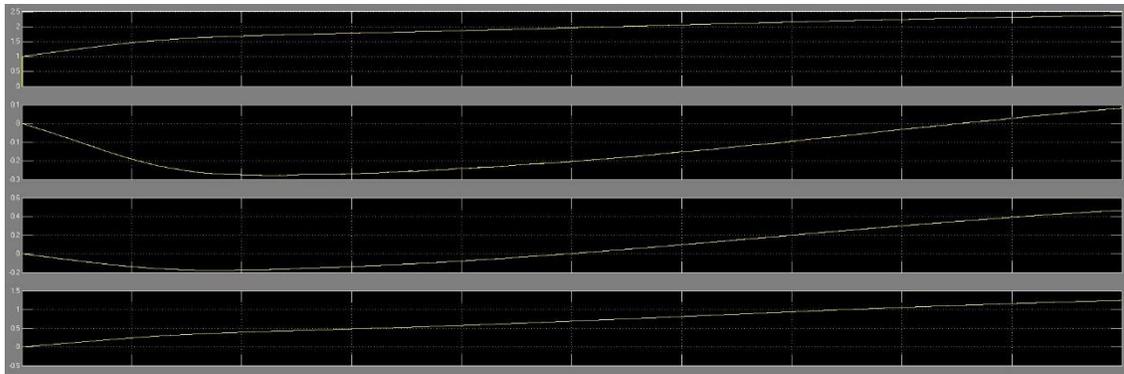


Figura 4.14: Coeficientes del ecualizador

La captura previa representa los coeficientes $h_i[0]$, $h_i[1]$, $h_q[0]$ y $h_q[1]$. Si aumentamos la atenuación a 25 dBs, los resultados son aún peores:

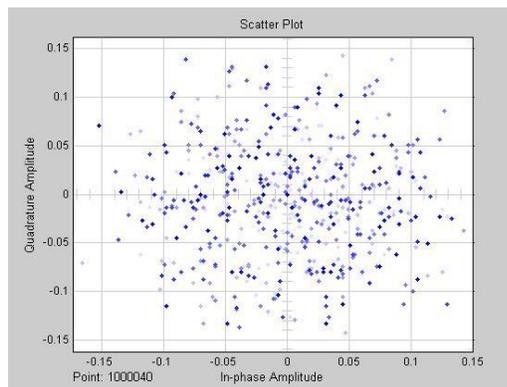


Figura 4.15: Señal ecualizada con atenuación de 25 dBs.

Para acelerar la velocidad de convergencia e impedir que el ecualizador malgaste muestras en multiplicar la señal por una constante, implementamos un control digital de ganancia que calcula la media de la señal y la multiplica por una constante para obtener un resultado en media similar al punto en que converge el ecualizador.

Con dicha mejora obtenemos los siguientes resultados para 25 dBs de atenuación:

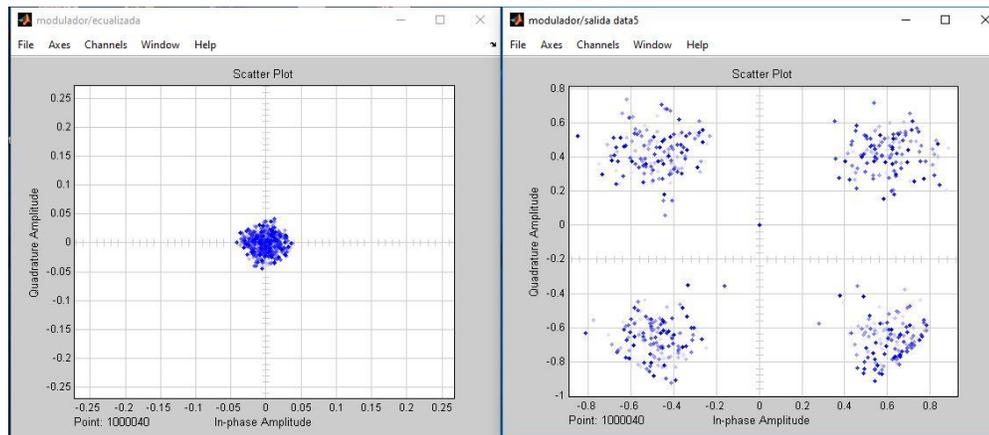


Figura 4.16: Señal ecualizada con atenuación de 25 dBs y AGC digital

Al analizar la evolución de los coeficientes del ecualizador, podemos observar cómo se obtiene una solución aceptable notablemente más rápido que sin el control digital de ganancia:

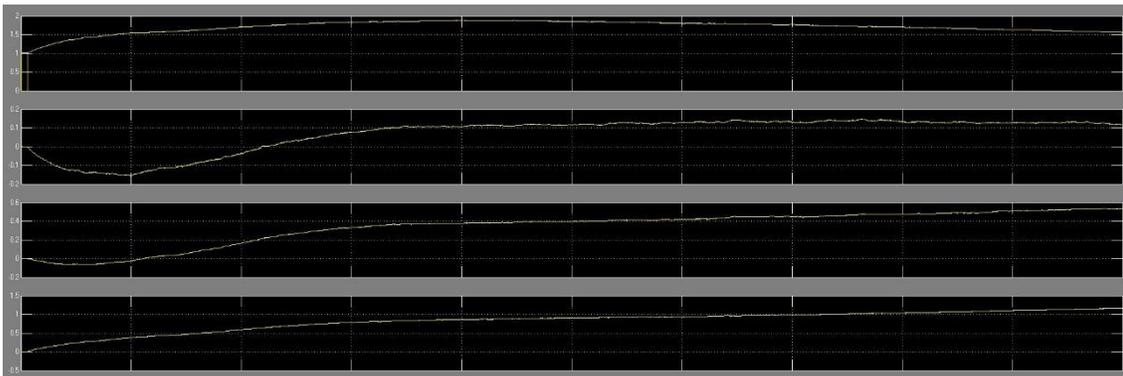


Figura 4.17: Coeficientes del ecualizador con AGC digital

La conclusión a la que llegamos es que la velocidad de convergencia del filtro depende no sólo de la complejidad del canal, sino también, y de manera crítica, de la diferencia entre el módulo medio de la señal de entrada y el módulo óptimo al que debe converger el ecualizador. Además, con pasos tan pequeños el ruido residual que queda es aceptable y apenas afecta a la solución.

En la siguiente imagen podemos observar la velocidad de convergencia del ecualizador ante señales de distintas atenuaciones:

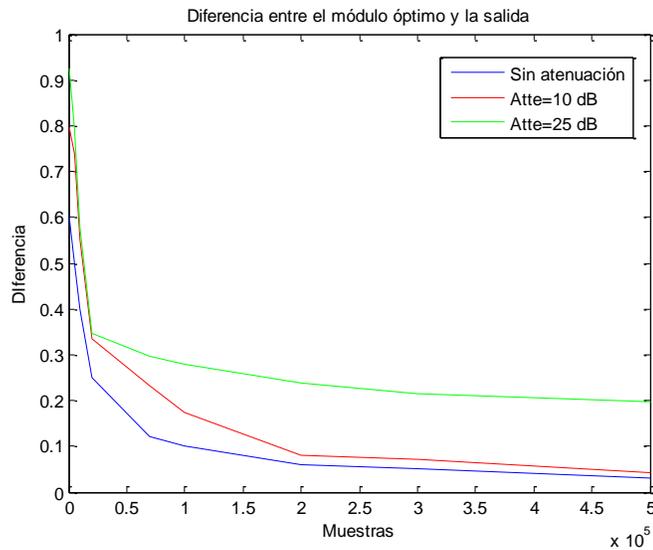


Figura 4.18: Velocidad de convergencia para distintas atenuaciones

4.2.2.2 Canal estándar con AWGN

En las pruebas anteriores ha quedado ya demostrado el rendimiento del ecualizador ante las distintas atenuaciones. Pero en este momento, es preciso plantearse una nueva pregunta: hasta qué punto afecta el ruido gaussiano aditivo blanco a la velocidad de convergencia.

Si añadimos ruido a una señal con una atenuación de 10 dBs hasta obtener una SNR de 12 dBs, obtenemos los siguientes resultados:

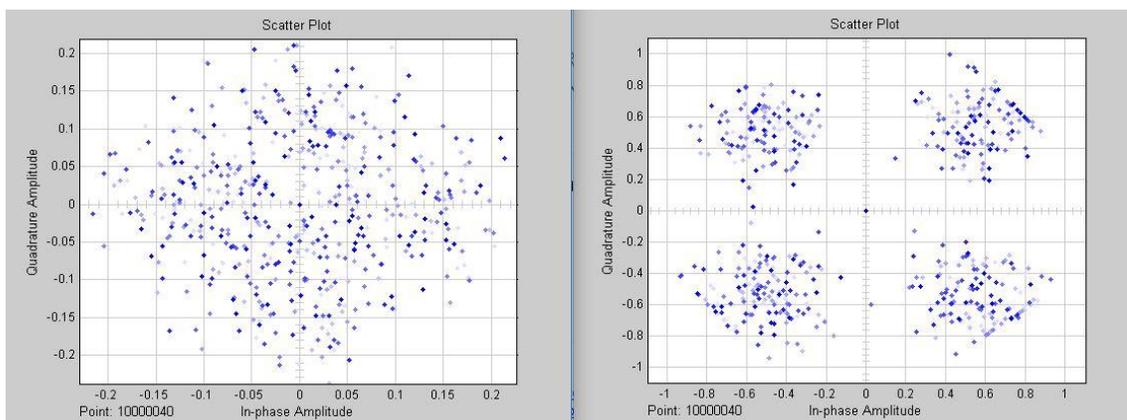


Figura 4.19: Señal ecualizada con AWGN

Se puede observar la disgregación que ocurre en la constelación debido al AWGN.

Si aumentamos la atenuación hasta 25 dBs, la salida del ecualizador es la siguiente:

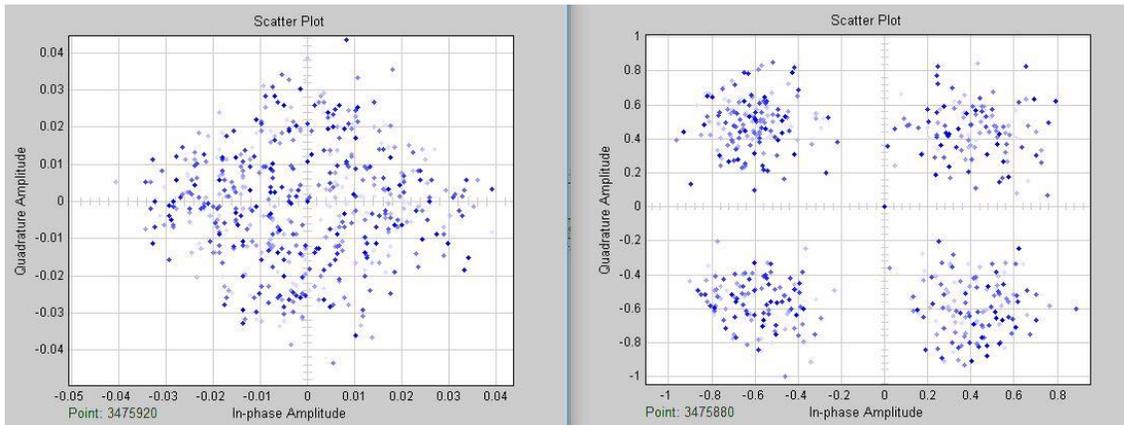


Figura 4.20: Señal ecualizada con 25 dBs de atenuación y AWG.

Ambas pruebas han sido realizadas con el control digital de ganancia activado.

En la gráfica inferior se puede observar cómo afecta el SNR de la señal a la velocidad de convergencia. El valor en el eje de ordenadas representa el incremento en porcentaje de muestras respecto a la señal con SNR óptima.

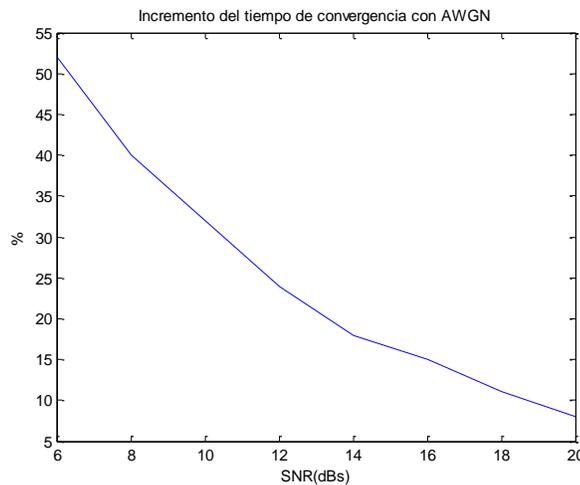


Figura 4.21: Velocidad de convergencia en función de la SNR

4.2.2.3 Canales Rayleigh y Rice

En las pruebas previas hemos enfrentado el ecualizador a canales estáticos. Sin embargo, al transmitir una señal al aire la respuesta del canal varía con el tiempo. El CMA es un ecualizador que funciona correctamente con LOS; no obstante, presenta un

rendimiento discutible en condiciones NLOS. Para demostrarlo se ha enfrentado el ecualizador a tres canales: Rice $k=4$ dBs, Rice $k=12$ dBs y Rayleigh (Doppler=100Hz), con los siguientes resultados:

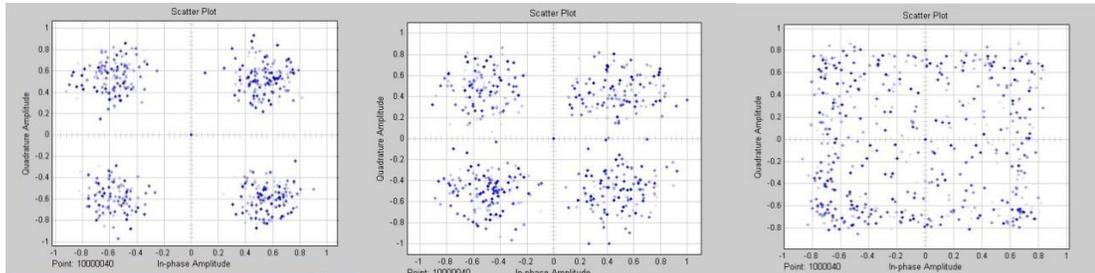


Figura 4.22: Ecualizador frente a canales Rice ($k=4$ y $k=12$) y Rayleigh

Ante las capturas anteriores, se puede deducir que mientras haya una componente de señal predominante relativamente fuerte y ante cambios lentos del canal, el comportamiento del CMA es adecuado. Sin embargo, si el canal varía muy rápidamente o no hay una componente predominante, el ecualizador CMA se muestra demasiado rígido para encontrar la solución.

Capítulo 5

Comportamiento en canales reales

En los apartados anteriores se han analizado las prestaciones que el CMA ofrece a nivel teórico y de simulación. Naturalmente, el siguiente paso de estudio consiste en enfrentar al ecualizador a un canal real.

Para ello se ha integrado en un sistema comercial, y se ha diseñado un software en Labview que permite monitorizar en tiempo real los símbolos a la entrada y a la salida del ecualizador y que, además, muestra las constelaciones en representación IQ.

5.1. Banda base

La prueba inicial para comprobar que el sistema y el ecualizador funcionan correctamente debe hacerse en banda base, es decir, conectando la salida del DAC a la entrada de un ADC de otra placa, tal y como muestra la imagen inferior:

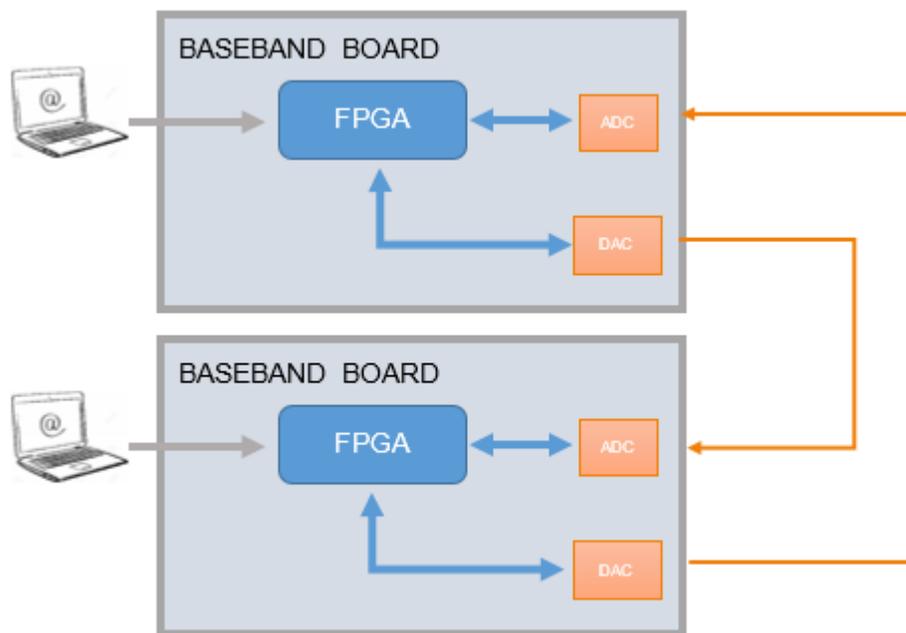


Figura 5.1: Prueba en banda base

El resultado obtenido es el siguiente:

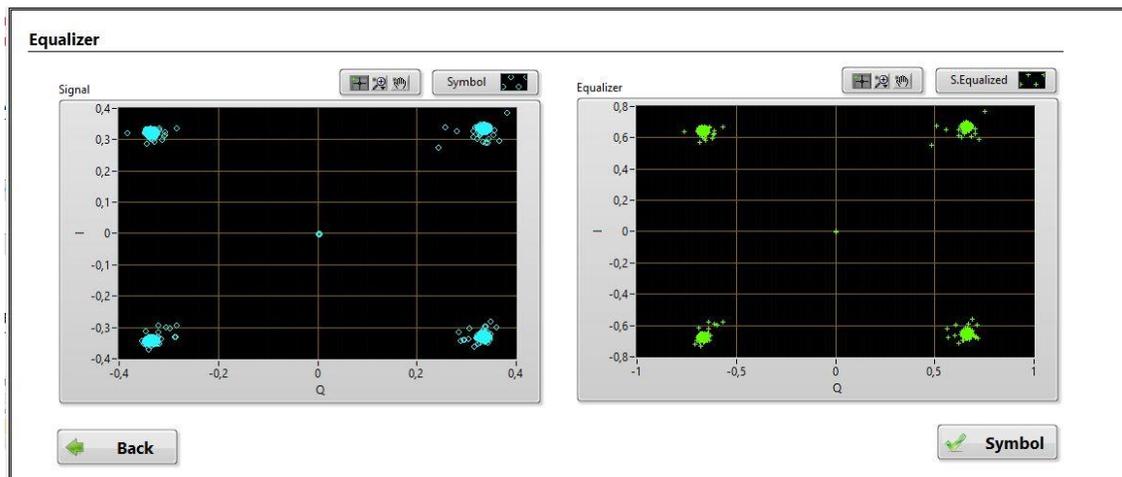


Figura 5.2: Resultados en banda base

A la izquierda se han representado los símbolos antes del ecualizador, y a la derecha, los símbolos después de procesarlos. Puesto que no hay realmente canal, lo único que hace el ecualizador es multiplicar por un valor la entrada para obtener el símbolo óptimo en módulo.

5.2. RF mediante cable

Una vez comprobado que el sistema funciona en banda base, el siguiente paso consiste en añadir toda la cadena de RF, conectando recepción y transmisión mediante un cable y añadiendo atenuadores para aumentar el ruido y reducir la potencia de la señal.

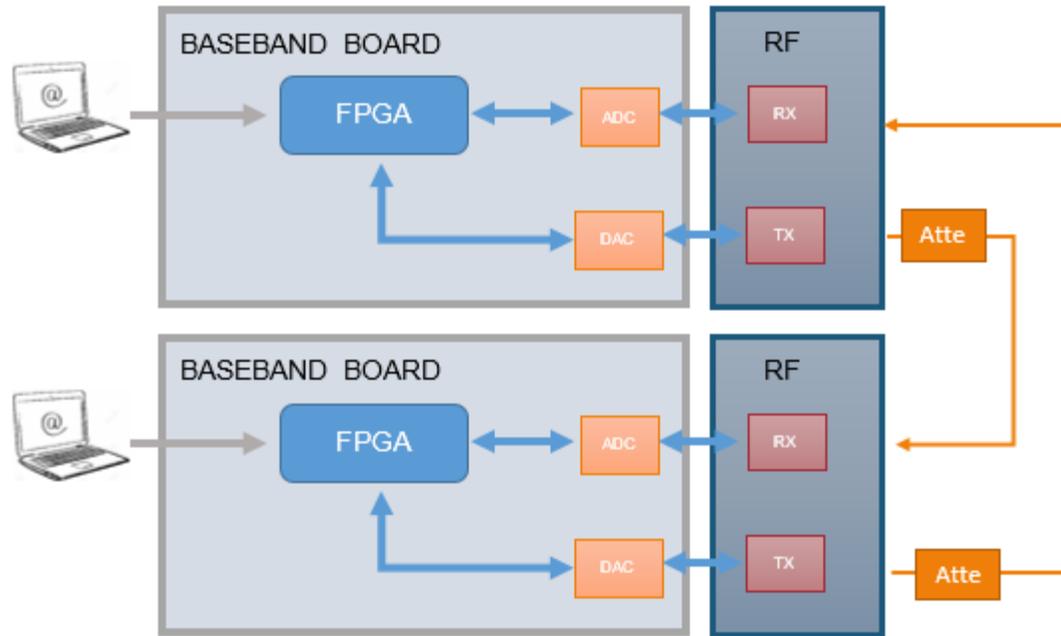


Figura 5.3: Prueba en RF mediante cable

Las características de la cadena de RF que se ha utilizado en la prueba se pueden observar en la tabla inferior:

Parámetro	Valor
Frecuencia	2.3 GHz
BW	8 MHz
P_{tx}	0 dBm
NF	2 dB

Tabla 5.1: Parámetros del escenario con cables

En la primera prueba añadimos 83 dBs de atenuación, con los siguientes resultados:

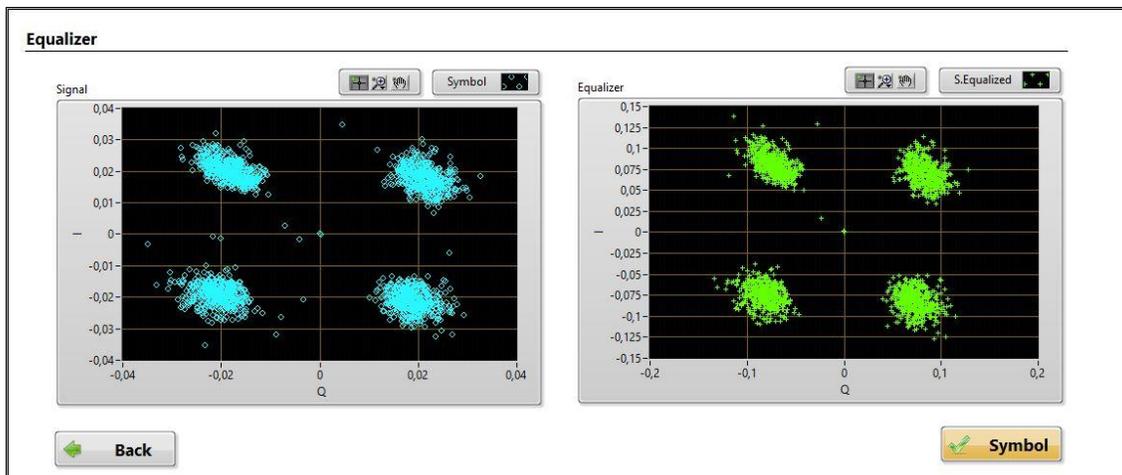


Figura 5.4: Resultados en RF con cable

Se puede observar cómo el ecualizador es capaz de acercar algunos de los símbolos que están fuera de los círculos con más densidad. En cualquier caso, su rendimiento aquí es limitado pues al ser el canal un cable no hay propagación multicamino.

Para concluir este apartado, fijamos ahora 93 dBs de atenuación, reduciendo la potencia prácticamente a la sensibilidad del sistema:

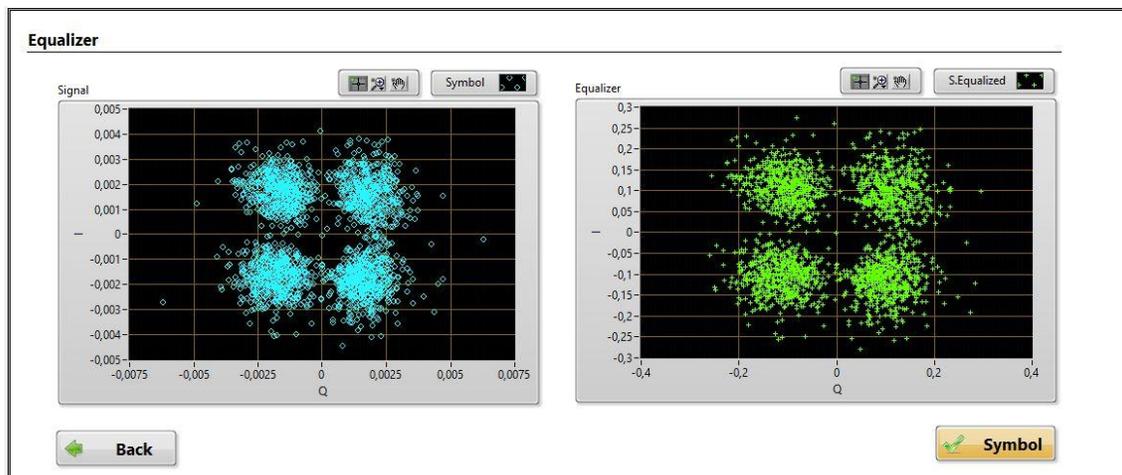


Figura 5.5: Resultados en RF con cable y máxima atenuación

Al igual que en la prueba anterior, la mejora que aporta el ecualizador es bastante limitada, aunque se puede observar que algunos símbolos críticos han sido desplazados.

5.3. Transmisión al aire

El último canal al que vamos a enfrentar el ecualizador es el verdaderamente real: consiste en sacar al aire la señal a través de unas antenas y realizar cambios al canal para observar cómo se comporta. Para ello, situamos las antenas a dos metros de distancia y añadimos 40 dBs de atenuación en recepción. La posición de esta atenuación, en recepción en lugar de en transmisión, puede parecer sorprendente. El motivo es que, al añadirlos en recepción, atenúas la señal y todas las interferencias, pero si los añadimos en transmisión, sólo atenúamos nuestra señal, quedando las interferencias cercanas, por ejemplo, 802.11g (wifi) a 2.4 GHz, con una potencia superior.

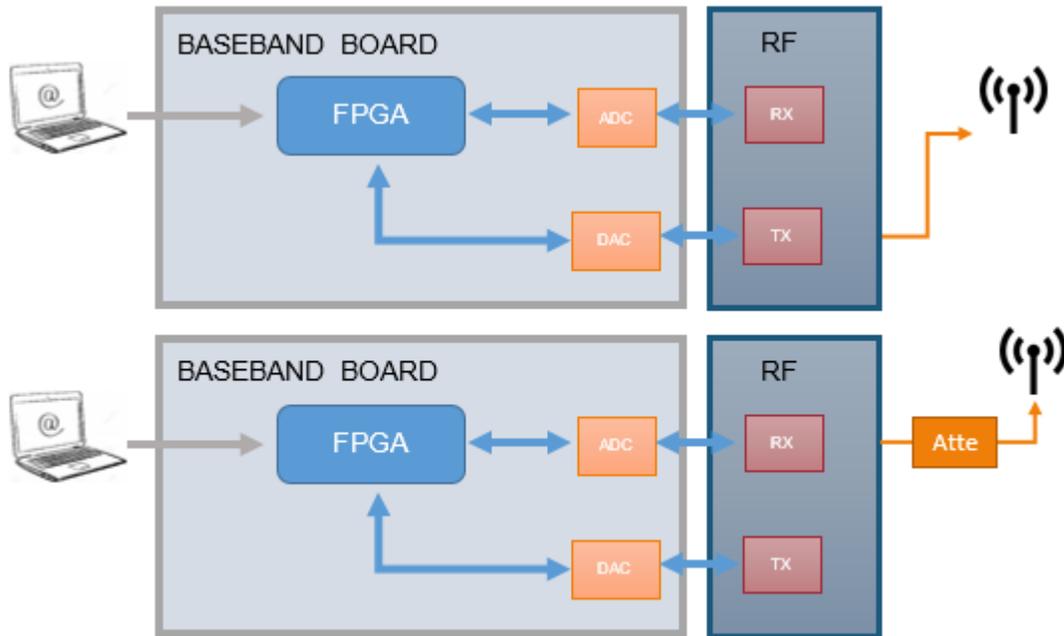


Figura 5.6: Prueba en RF con antenas

Los parámetros más importantes del entorno de prueba son:

Parámetro	Valor
Ganancia Antena	5 dBi
Polarización	Vertical
Diagrama de radiación	Omnidireccional
Atenuador	40 dBs
Pérdidas espacio libre	46 dBs

Tabla 5.2: Parámetros del escenario con antenas

El primer test consiste estudiar el ecualizador existiendo línea de vista entre ambas antenas, en un canal que se puede equiparar a un canal Rice con una fuerte componente principal. Visualmente se puede apreciar una mejora en muchos puntos, sobre todo en las zonas cercanas a los ejes.

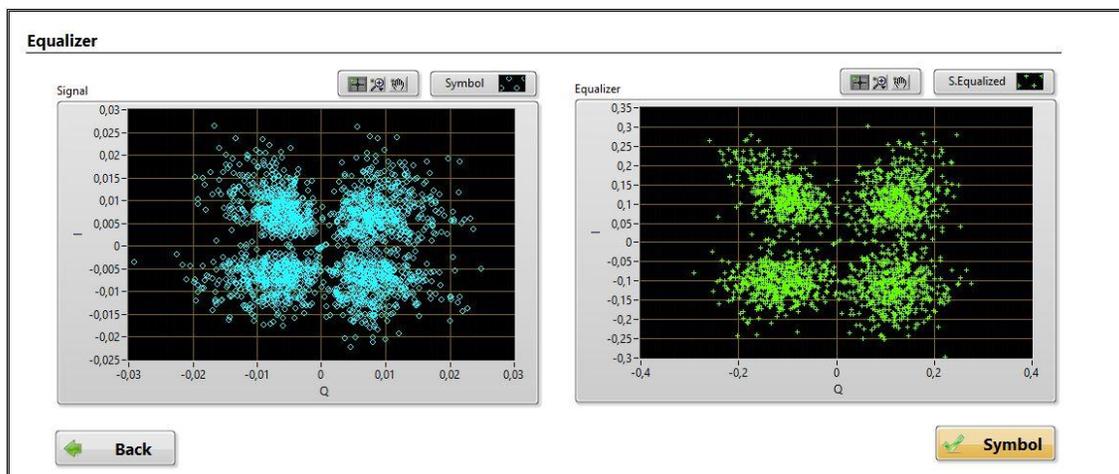


Figura 5.7: Resultados de la prueba con antenas

A continuación situamos un panel conductor entre las dos antenas bloqueando completamente la línea de vista, y buscamos un punto en el que recibamos correctamente. Las constelaciones de entrada y salida serían las siguientes:

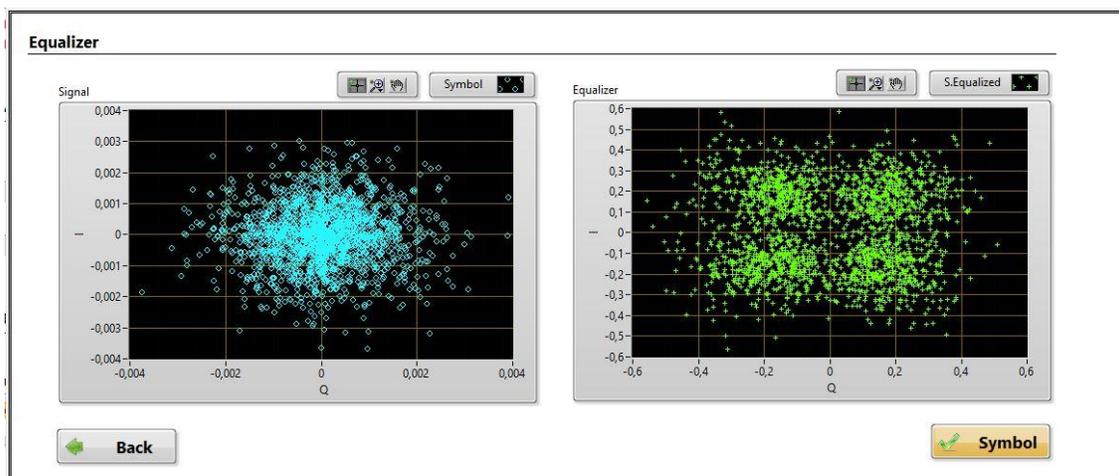


Figura 5.8: Resultados de la prueba con antenas y NLOS

En la captura superior sí que se puede apreciar, notablemente, la mejora que aporta el ecualizador al sistema. Consigue, con unos símbolos aparentemente inconexos, sacar una constelación en la que más o menos se pueden definir las regiones de decisión.

Es posible buscar otras configuraciones sin línea de vista en las que el ecualizador no sea capaz de encontrar una solución debido a que se queda bloqueado en una solución subóptima, o bien oscilando en algún punto:

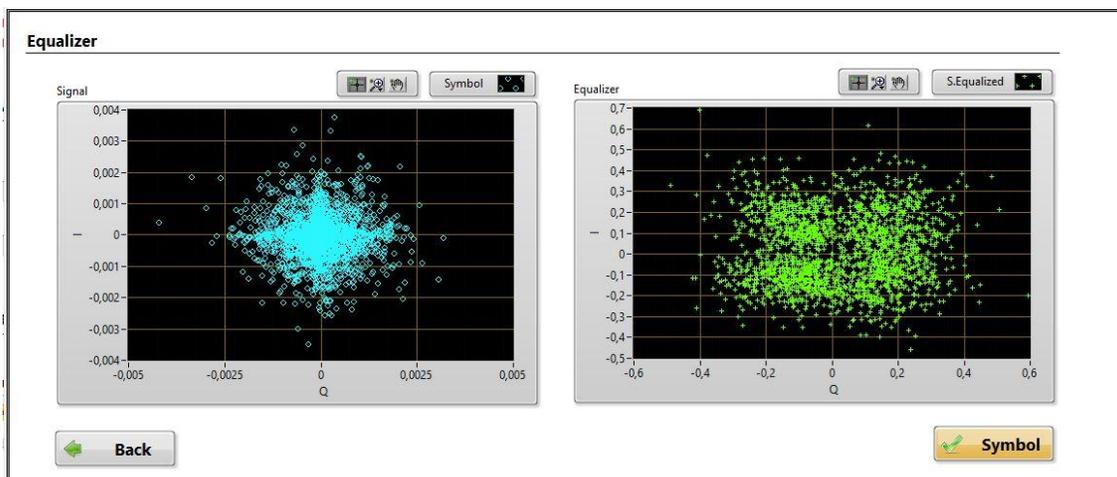


Figura 5.9: Resultados de la prueba con antenas y NLOS (2)

En definitiva, las conclusiones que se pueden obtener de todas las pruebas son las siguientes:

- Tal y como se podía intuir, en banda base o por cable el ecualizador prácticamente sólo funciona como AGC.
- En enlaces con LOS, pese a ser probado en un entorno interior con una fuerte componente multitrayecto, la mejora que ofrece es pequeña.
- En enlaces sin línea de vista es donde se aprecia realmente la mejora clara de rendimiento que ofrece el ecualizador.

5.4. Otras pruebas

La aplicación desarrollada para estudiar el ecualizador puede servir también para analizar otros bloques o fenómenos que ocurren al realizar una comunicación a través del aire.

5.4.1. Recuperación de fase de la portadora

El primer efecto que se puede observar es la recuperación de fase. La recuperación de fase es una operación necesaria para modulaciones PSK, en las que el receptor debe

estimar la fase de la portadora transmitida para poder asignar el valor correspondiente a cada símbolo. La siguiente imagen muestra teóricamente el fenómeno:

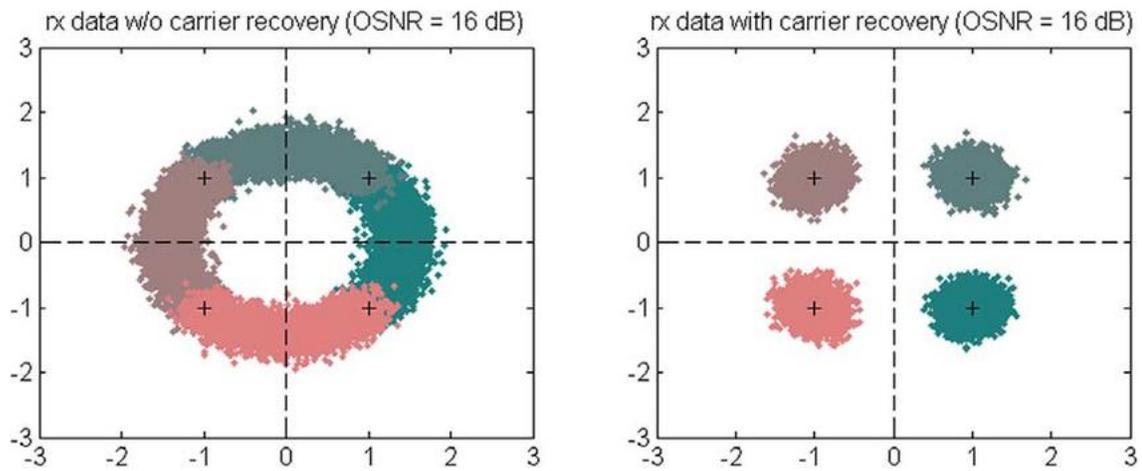


Figura 5.10: Recuperación de fase

Para observar este efecto, anulamos la recuperación de fase en una transmisión a través del aire. La señal obtenida se observa en la captura inferior:

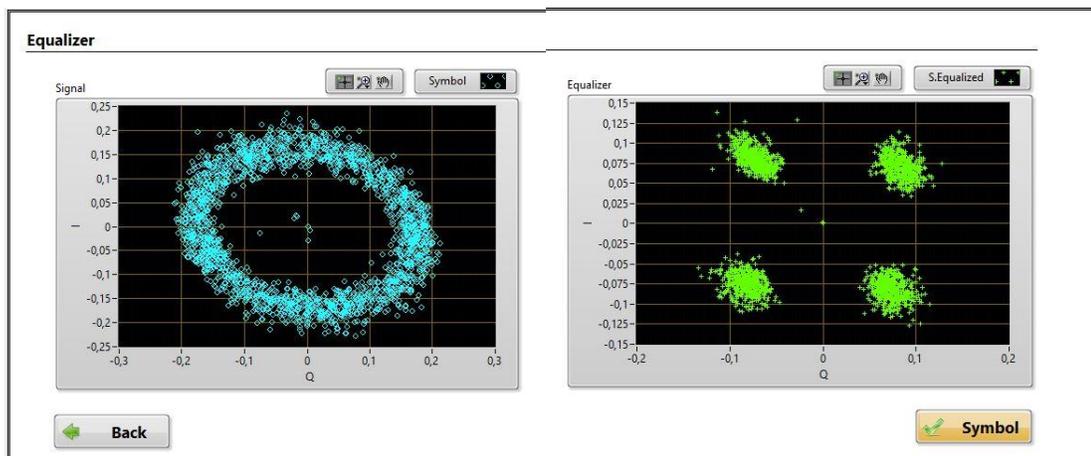


Figura 5.11: Resultados sin/con recuperación de fase

5.4.2. IQ desbalanceadas

Al recibir y transmitir una señal y demodularla con un receptor directo, es decir, que pasa de la frecuencia de la portadora directamente a banda base, uno de los efectos limitantes es el desbalanceo que se produce en la señal I y en la señal Q al seguir caminos paralelos pero distintos, con ganancias y adaptaciones ligeramente diferentes. La variación entre los caminos suele depender del receptor y, normalmente, basta con ajustarla una sola vez.

Este error se puede dividir en dos tipos:

- Error de ganancia: modifica la amplitud de una de las señales respecto a la otra.

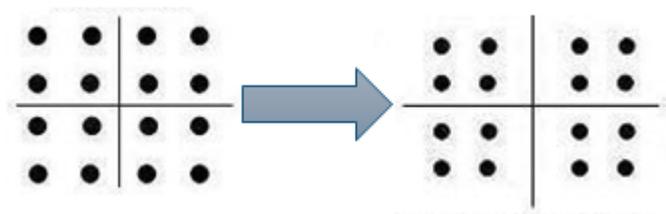


Figura 5.12: Desbalanceo en ganancia

- Error de fase: tiende a inclinar la constelación hacia algún lado.

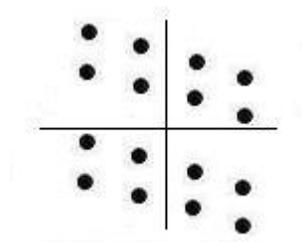


Figura 5.13: Desbalanceo en fase

Se puede observar este fenómeno si además de la recuperación de fase, anulamos el bloque que lo corrige:

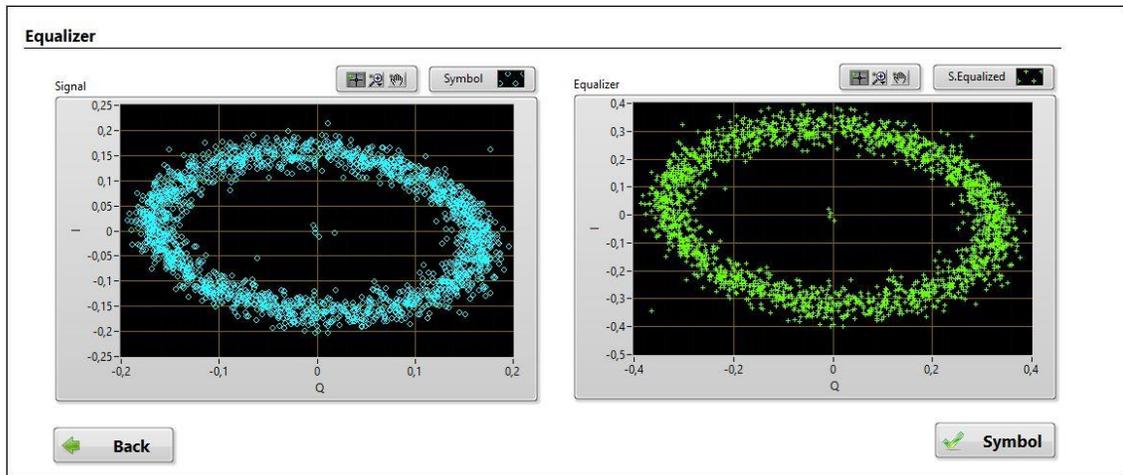


Figura 5.14: Resultados con el bloque de balanceo IQ desactivado

5.4.3. Cambios rápidos en el canal

Uno de los problemas más críticos en las comunicaciones aéreas, conocido con el nombre de desvanecimiento profundo, se produce cuando hay cambios bruscos del canal que reducen de manera drástica la potencia recibida.

En la práctica es casi imposible seguir recibiendo durante el desvanecimiento, pero un buen sistema debe ser capaz de volver a funcionar rápidamente tras restablecerse de nuevo el canal. Bloques como el AGC, el demodulador o el ecualizador tienen que estar diseñados para minimizar el daño producido por los desvanecimientos rápidos.

En el laboratorio, estas caídas bruscas de la potencia recibida se pueden simular bloqueando momentáneamente la línea de vista, con los siguientes resultados:

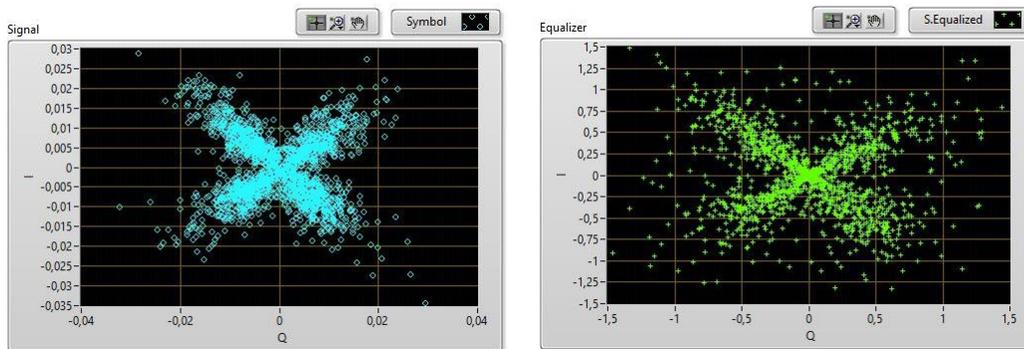


Figura 5.15: Resultados con cambios bruscos en el canal

En las capturas anteriores se puede apreciar visiblemente que el ecualizador se recupera rápidamente de los desvanecimientos profundos.

Capítulo 6

Conclusiones y líneas futuras

Todo parece indicar que a medio y largo plazo los UAVs de largo alcance van a ir ganando protagonismo tanto en el ámbito militar como civil, debido, entre otros motivos, a una mejora de los algoritmos de inteligencia artificial, de las baterías que los alimentan y de la capacidad de cómputo de las FPGAs.

No obstante, estas mejoras requieren, de manera indefectible, un mayor flujo de datos, ya sean de telemetría o de video, y suponen un aumento drástico de la complejidad de los bloques de procesamiento de señal implementados en la FPGA.

Y es en este marco en el que se desarrolla el ecualizador CMA del que es objeto este proyecto, cuyo propósito es procesar adecuadamente una señal con un ancho de banda considerable y con unos recursos muy limitados debido a la complejidad del resto del firmware.

Para aumentar la eficiencia del ecualizador se han implementado unos bloques llamados CDC, que le permiten funcionar a una frecuencia de reloj distinta a la del resto de bloques colindantes, reduciendo los problemas que pueden llegar a ocasionar los bloques complejos matemáticamente a altas frecuencias. Además, se ha diseñado un bloque AGC digital que disminuye de manera drástica la velocidad de convergencia del mismo.

Con el fin de validar completamente su funcionamiento, se ha integrado en un sistema comercial y se han monitorizado en tiempo real sus entradas y salidas en distintos escenarios.

Las futuras líneas de trabajo que se plantean a la redacción de este proyecto se pueden organizar en tres vertientes diferentes:

- Implementación de distintas funciones de coste para constelaciones multimódulo.
- Implementación del ecualizador MMSE, un ecualizador ciego basado en el LMS.
- Análisis del algoritmo CMA en pruebas de vuelo real.

Acrónimos

ADC	Analog to Digital Conversor
AGC	Automatic Gain Control
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BW	BandWidth
CDC	Cross Domain Crossing
CLB	Configurable Logic Block
CMA	Constant Module Algorithm
DAC	Digital to Analog Conversor
DC	Direct Current
DL	DataLink
DQPSK	Differential Quadrature Phase Shift Keying
DSP	Digital Signal Processor
FIFO	First In, First Out
FPGA	Field Programmable Gate Array
GSA	Generalized Sato Algorithm
I	In-phase
LMS	Least Mean Square
LOS	Line of Sight
MMA	Multimodulus Algorithm
MMSE	Minimum Mean Square Error
NF	Noise Figure
NLOS	Non Line of Sight
PLD	Programmable Logic Devices
PSK	Phase Shift Keying
Q	Quadrature
QAM	Quadrature Amplitude Modulation
QOS	Quality of Service
RF	Radio Frequency
RLS	Recursive Least Squares
SISO	Single Input Single Output
SNR	Signal to Noise Ratio
UAV	Unmanned Aerial Vehicle
VHDL	VHSIC Hardware Description Language

Bibliografía

- [1] D. N. Godard, Self-recovering equalization and carrier tracking in two-dimensional data communication systems. IEEE Transaction on Communications, 1980.
- [2] Flávio R. M. Pavan, Magno T. M. Silva, Maria D. Miranda, Avoiding divergence in the constant modulus algorithm for blind equalization of MIMO systems, Sensor Array and Multichannel Signal Processing Workshop (SAM), 2016.
- [3] Clifford E. Cummings, Clock Domain Crossing (CDC) Design & Verification Techniques Using System Verilog, SNUG, 2008.
- [4] Charles Fulks, Best FPGA Development Practices, 2014.
- [5] Philippe Garrault y Brian Philofsky, HDL Coding Practices to Accelerate Design Performance, Xilinx White Paper, 2006.
- [6] Haykin, Simon, *Adaptive Filter Theory*, Third Ed., Upper Saddle River, N.J., Prentice-Hall, 1996.
- [7] Kevin Banovic, Mohammed A.S. Khalid . FPGA Implementation of a Configurable Complex Blind Adaptive Equalizer. IEEE International Symposium on Signal Processing and Information Technology, 2007.
- [8] N. M. Pinto, L. M. Pessoa, J. C. Ferreira y H. M. Salgado. FPGA Implementation of Signal Processing Algorithms in Coherent Optical Systems. IEEE International Symposium on Signal Processing and Information Technology, 2009.
- [9] Robert G. Schumacher. An efficient FPGA implementation of a constant modulus algorithm equalizer for Wireless telemetry. Tesis doctoral, University of Dayton, 2014.
- [10] Tara.Saikumar, R. Nirmaa Devi, Dr. K. Kishna Rao , Adaptive MMSE Equalizer for Blind Fractional Spaced CMA Channel Equalization through LMS Algorithm. IJASUC, 2012.
- [11] Johnson, Richard C. Jr., Philip Schniter, Thomas. J. Endres, et al., Blind Equalization Using the Constant Modulus Criterion: A Review, IEEE, October 1998.
- [12] 1980. Kevin Banovic, Raymond Lee, et al., Computationally-Efficient Methods for Blind Adaptive Equalization, University of Windsor, 2005.

- [13] D. J. Allred, H. Yoo, V. Krishnan, W. Huang, and D. V. Anderson, A novel high performance distributed arithmetic adaptive filter implementation on an FPGA, Proc. ICASSP, Monreal, Quebec, 2004.
- [14] Ying Xiao, Zhenxing Li, Yuhua Dong, Constant modulus algorithms using variable step-size. Commun Technol , 2008.
- [15] M Yang D., Chen D., Kuang W., Constant Modulus Blind Equalization Analysis for High Speed Implementation. Proceedings of the 26th Conference of Spacecraft TT&C Technology in China, 2013.
- [16] Kevin Banović, .Mohammed A.S. Khalid, Esam Abdel-Raheem J.L. A configurable fractionally-spaced blind adaptive equalizer for QAM demodulators, Digital Signal Processing, 2007.
- [17] William Jenkins, Andrew W. Hull, Jeffrey C. Strait, Bernard A. Schnaufer, Xiaohui Li , *Advanced Concepts in Adaptive Signal Processing*, The Springer International Series in Engineering and Computer Science, 1996.
- [18] Catherine Elizabeth Manlove , Length requirement of the fractionally-spaced equalizer under CMA, Cornell University, 1994.
- [19] Ray Andraka P.E., Modulation and Demodulation techniques for FPGAs, DesignCon, 2000.
- [20] Página web de Xilinx: <https://www.xilinx.com>
- [21] Página web de Altera: <https://www.altera.com>