

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS
INDUSTRIALES Y DE TELECOMUNICACIÓN

UNIVERSIDAD DE CANTABRIA



Trabajo Fin de Máster

**Diseño de interfaz Linux y caracterización
de frontends RF para un emulador de canal
inalámbrico sobre plataformas de alto
rendimiento MPSoC**

**(Linux interface design and RF frontends
characterization for a Wireless channel
emulator on MPSoC high performance
platforms)**

Para acceder al Título de

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

Autor: Lucía López Martínez
Director: Iñaki Val Beitia

Octubre - 2019

**MASTER UNIVERSITARIO EN INGENIERÍA DE
TELECOMUNICACIÓN**

CALIFICACIÓN DEL TRABAJO FIN DE MÁSTER

Realizado por: Lucía López Martínez

Director del TFM: Iñaki Val Beitia

Título: “Diseño de interfaz Linux y caracterización de frontends RF para un emulador de canal inalámbrico sobre plataformas de alto rendimiento MPSoC”

Title: “Linux interface design and RF frontends characterization for a Wireless Channel emulator on MPSoC high performance platforms”

Presentado a examen el día: 17 de Octubre de 2019

para acceder al Título de

**MASTER UNIVERSITARIO EN INGENIERÍA DE
TELECOMUNICACIÓN**

Composición del Tribunal:

Presidente (Apellidos, Nombre): Valle López, Luis

Secretario (Apellidos, Nombre): Aja Abelán, Beatriz

Vocal (Apellidos, Nombre): García García, José Angel

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)

Resumen

Se espera que las comunicaciones inalámbricas sean uno de los principales impulsores de la nueva revolución industrial, conocida como la Industria 4.0. Las comunicaciones industriales son diferentes de las comunicaciones en aplicaciones convencionales y presentan una complejidad técnica elevada. Entre sus desafíos se encuentran: la necesidad de tiempo real, baja latencia, alta fiabilidad, y un entorno inalámbrico con una gran cantidad de interferencias y pérdidas de señal por multicamino.

Además de la problemática propia de la comunicación inalámbrica, la validación del funcionamiento de un sistema inalámbrico de comunicaciones en un entorno industrial es muy compleja y costosa. Por ello, es necesario buscar alternativas que permitan realizar dicha validación en laboratorio. La principal alternativa es utilizar un emulador de canal.

En el presente documento, se presenta el diseño de un emulador de canal inalámbrico específicamente diseñado para solventar los problemas de validación de los sistemas de comunicaciones industriales. De todos los elementos y fases que consta el diseño e implementación, en este proyecto se ha abordado la implementación de la interfaz gráfica del emulador de canal, y la caracterización de los frontend de RF del emulador.

Agradecimientos

Tras varios meses de trabajo puedo dar por finalizado mi trabajo fin de máster, y con él, una etapa muy importante en mi vida. El primer reconocimiento de este trabajo es para mí director Iñaki Val, por haberme dado la oportunidad de volver a Ikerlan para realizar el TFM y haber podido trabajar con un gran equipo donde he aprendido más de lo que me imaginaba. Muchas gracias por ser como eres y por estar dispuesto siempre a orientarme.

Este agradecimiento también se extiende a todos los miembros del grupo de KOS. Es un placer trabajar rodeado del compañerismo que se siente nada más entrar en el grupo. He de agradecer tanto a Raúl como a Víctor por su gran paciencia y porque siempre están dispuestos a echarme una mano, aunque tenga mil cosas que hacer. Y no por ello menos importante, a una persona muy especial que me está ayudando a mirar la vida con otros ojos. Muchas gracias por todo el tiempo que has dedicado para aconsejarme, ayudarme a aprender cómo se deben hacer las cosas y estar siempre para que todo me sea un poco más fácil.

También quiero agradecer a mi tutor Tomás Fernández por haberme apoyado y guiado con este proyecto.

Finalmente, gracias a mis amigos y familia; especialmente a mis padres y mis tíos, por apoyarme siempre y confiar en mí.

Índice

1.	Introducción	14
1.1.	Descripción del proyecto	15
1.1.1.	Objetivos del proyecto	16
1.2.	Enmarque del proyecto	17
2.	Marco teórico.....	18
2.1.	Emulador de canal inalámbrico	18
2.2.	Canales inalámbricos.....	20
2.2.1.	Modelado de canal	21
2.2.1.1.	Large Scale Fading.....	23
2.2.1.1.1.	Path loss	23
2.2.1.1.2.	Shadowing.....	23
2.2.1.2.	Small Scale Fading.....	24
2.2.1.2.1.	Dispersión temporal del canal.....	25
2.2.1.2.2.	Distribuciones fading	29
2.2.1.2.3.	Variación temporal del canal debido al movimiento.....	30
2.2.1.2.4.	Distribuciones Doppler	31
3.	Emulador de canal inalámbrico.....	34
3.1.	Antecedentes emulador de canal	34
3.1.1.	Modelo de frontend	36
3.1.2.	FPGA.....	36
3.1.3.	Interfaz de control	37
3.2.	Nueva arquitectura emulador de canal	37
3.2.1.	Zynq UltraScale + MPSoC	39
3.2.2.	Frontend RF	40
4.	Herramientas de desarrollo	42
4.1.	Matlab R2018a	42
4.2.	Oracle VM VirtualBox	42

4.2.1.	Guía instalación máquina virtual y cross-compiler ARM.....	43
4.2.1.1.	Requisitos previos.....	43
4.2.1.2.	Cross-compilation Qt5.....	44
5.	Interfaz emulador de canal.....	46
5.1.	Primera versión interfaz emulador de canal.....	46
5.1.1.	Interfaz usuario.....	48
5.1.2.	Interfaz A53-R5.....	50
5.2.	Segunda versión interfaz emulador de canal.....	51
5.2.1.	Interfaz de usuario.....	52
5.2.2.	Interfaz A53-R5.....	57
6.	Caracterización Frontend RF.....	64
6.1.	Diseño Frontends RF.....	64
6.2.	Configuración elementos programables del Frontend.....	70
6.2.1.	Atenuador variable RUDAT-6000-110.....	70
6.2.2.	Amplificador de IF ADL5202.....	71
6.2.3.	Osciladores locales ADF4355.....	71
6.3.	Hardware y mediciones para la caracterización.....	73
6.3.1.	Hardware empleado.....	73
6.3.2.	Campaña de medidas.....	74
6.3.2.1.	Medidas etapa Entrada/Salida.....	74
6.3.2.2.	Medidas etapas down-conversion y up-conversion.....	76
6.4.	Resultados caracterización Frontend.....	79
6.4.1.	Banda 434 MHz.....	79
6.4.1.1.	Etapa Entrada/Salida & Etapa 2 de down-conversion.....	80
6.4.1.2.	Etapa Entrada/Salida y Etapa 1 de up-conversion.....	82
6.4.2.	Banda 868 MHz.....	85
6.4.2.1.	Etapa Entrada/Salida y Etapa 2 de down-conversion.....	85
6.4.2.2.	Etapa Entrada/Salida y Etapa 1 de up-conversion.....	87
6.4.3.	Banda 2.45 GHz.....	90

6.4.3.1. Etapa Entrada/Salida	91
6.4.3.2. Etapa Entrada/Salida y down-conversion 1-2.....	92
6.4.3.3. Etapa Entrada/Salida y up-conversion 1-2	95
7. Conclusiones	97

Índice de figuras

<i>Figura 1.1. Logo de Ikerlan.....</i>	<i>17</i>
<i>Figura 2.1. Arquitectura emulador de canal.....</i>	<i>18</i>
<i>Figura 2.2. Esquema canal inalámbrico.....</i>	<i>20</i>
<i>Figura 2.3. Efectos de la propagación en un canal inalámbrico [11].....</i>	<i>21</i>
<i>Figura 2.4. Tapped delay line model [2].....</i>	<i>24</i>
<i>Figura 2.5. Respuesta al impulso de un canal multitrayecto.....</i>	<i>25</i>
<i>Figura 2.6. Dispersión temporal.....</i>	<i>25</i>
<i>Figura 2.7. Señal recibida con ISI.....</i>	<i>26</i>
<i>Figura 2.8. Señal recibida sin ISI.....</i>	<i>26</i>
<i>Figura 2.9. Perfil potencia-retardo (PDP).....</i>	<i>27</i>
<i>Figura 2.10. Distorsión selectiva en frecuencia.....</i>	<i>28</i>
<i>Figura 2.11. Función de correlación y ancho de banda de coherencia.....</i>	<i>28</i>
<i>Figura 2.12. Canal sin distorsión en frecuencia.....</i>	<i>29</i>
<i>Figura 2.13. Potencia espectro Doppler.....</i>	<i>31</i>
<i>Figura 2.14. Jakes Doppler Spectrum.....</i>	<i>31</i>
<i>Figura 2.15. Bell Doppler Spectrum.....</i>	<i>33</i>
<i>Figura 3.1. Diagrama de bloques emulador de canal.....</i>	<i>34</i>
<i>Figura 3.2. Diagrama de bloques etapa RF/IF conversion [4].....</i>	<i>35</i>
<i>Figura 3.3. Sistema MIMO 4x4 emulador de canal.....</i>	<i>38</i>
<i>Figura 3.4. Arquitectura nuevo emulador de canal.....</i>	<i>38</i>
<i>Figura 3.5. Arquitectura elementos programables del frontend.....</i>	<i>40</i>
<i>Figura 4.1. Logotipo Matlab R2018a.....</i>	<i>42</i>
<i>Figura 4.2. Logotipo Oracle VM VirtualBox y QT Ctreator.....</i>	<i>43</i>
<i>Figura 4.3. Cross-compilation para ARM [32].....</i>	<i>43</i>
<i>Figura 5.1. Interfaz de control emulador de canal [36].....</i>	<i>47</i>
<i>Figura 5.2. Primera versión interfaz usuario.....</i>	<i>48</i>
<i>Figura 5.3. Parámetros de canal.....</i>	<i>49</i>
<i>Figura 5.4. Parámetros path.....</i>	<i>49</i>
<i>Figura 5.5. Parámetros de configuración frontend.....</i>	<i>49</i>
<i>Figura 5.6. Estructura interfaz de usuario.....</i>	<i>53</i>
<i>Figura 5.7. Pestaña de entradas interfaz.....</i>	<i>54</i>
<i>Figura 5.8. Pestaña Small Scale Fading.....</i>	<i>54</i>
<i>Figura 5.9. Parámetros distribución de envolvente Rice y Nakagami.....</i>	<i>55</i>
<i>Figura 5.10. Configuración de taps.....</i>	<i>55</i>
<i>Figura 5.11. Pestaña Large Scale Fading.....</i>	<i>56</i>
<i>Figura 5.12. Pestaña generador de ruido Gaussiano.....</i>	<i>56</i>
<i>Figura 5.13. Pestaña salidas interfaz.....</i>	<i>57</i>
<i>Figura 6.2. Esquema frontend RF completo.....</i>	<i>65</i>
<i>Figura 6.3. Etapa Entrada/Salida.....</i>	<i>68</i>

<i>Figura 6.4. Etapas down-conversion.....</i>	<i>69</i>
<i>Figura 6.5. Etapas up-conversion.....</i>	<i>69</i>
<i>Figura 6.6. Software control atenuador variable RUDAT 6000 [28].</i>	<i>70</i>
<i>Figura 6.7. Software control amplificador de IF [27].</i>	<i>71</i>
<i>Figura 6.8. Programación OL1.</i>	<i>72</i>
<i>Figura 6.9. E5080A ENA Vector Network Analyzer.....</i>	<i>73</i>
<i>Figura 6.10. E4404B ESA-E Spectrum Analyzer.....</i>	<i>74</i>
<i>Figura 6.11. Caracterización S11 de la etapa down-conversion 2 y entrada salida para la banda de 434 MHz con un ancho de banda de 5 MHz con ganancia amplificador de IF a 0 dB.....</i>	<i>80</i>
<i>Figura 6.12. Caracterización de las pérdidas de conversión de la etapa down-conversion 2 y entrada salida para la banda de 434 MHz con un ancho de banda de 5 MHz con ganancia del amplificador de IF a 0 dB.....</i>	<i>81</i>
<i>Figura 6.13. Caracterización de las pérdidas de conversión de la etapa down-conversion 2 y entrada salida para la banda de 434 MHz con un ancho de banda 5 MHz. (a) con ganancia de amplificador de IF = 10 dB, (b) con ganancia de amplificador de IF = 20 dB.....</i>	<i>82</i>
<i>Figura 6.14. Caracterización SINR de la etapa down-conversion 2 y entrada salida para la banda de 434 MHz con un ancho de banda de 5 MHz con ganancia amplificador de IF a 0 dB.....</i>	<i>82</i>
<i>Figura 6.15. Caracterización S11 de la etapa up-conversion 1 y entrada salida para la banda de 434 MHz con un ancho de banda de 5 MHz.</i>	<i>83</i>
<i>Figura 6.16. Caracterización de las pérdidas de conversión de la etapa up-conversion 1 y entrada salida para la banda de 434 MHz con un ancho de banda de 5 MHz con ganancia del amplificador de IF a 0 dB.</i>	<i>84</i>
<i>Figura 6.17. Caracterización SINR de la etapa up-conversion 1 y entrada salida para la banda de 434 MHz con un ancho de banda de 5 MHz con ganancia del amplificador de IF a 0 dB.....</i>	<i>84</i>
<i>Figura 6.18. Caracterización S11 de la etapa down-converion 2 y entrada salida para la banda de 868 MHz con un ancho de banda de 5 MHz y ganancia del amplificador de IF a 0 dB.....</i>	<i>86</i>
<i>Figura 6.19. Pérdidas de conversión de la etapa down-conversion 2 y entrada salida para la banda de 868 MHz con un ancho de banda de 5 MHz con ganancia del amplificador de IF a 0 dB.....</i>	<i>86</i>
<i>Figura 6.20. Pérdidas de conversión de la etapa down-conversion 2 y entrada salida para la banda de 868 MHz con un ancho de banda de 5 MHz. (a) con ganancia de amplificador de IF = 10 dB, (b) con ganancia de amplificador de IF = 20 dB.....</i>	<i>87</i>
<i>Figura 6.21. Caracterización SINR de la etapa down conversion 2 y entrada salida para la banda de 868 MHz con ancho de banda de 5 MHz con ganancia del amplificador de IF a 0 dB.....</i>	<i>87</i>

<i>Figura 6.22. Caracterización S11 para la banda de 868MHz con un ancho de banda de 5 MHz para la etapa Entrada/Salida y up-conversion 1.....</i>	<i>88</i>
<i>Figura 6.23. Caracterización de las pérdidas de conversión para la banda de 868 MHz con un ancho de banda de 5 MHz para la etapa Entrada/Salida y up-conversion1.....</i>	<i>89</i>
<i>Figura 6.24. Caracterización SINR para la banda de 868 MHz con un ancho de banda de 5 MHz para la etapa Entrada/Salida y up-conversion 1.....</i>	<i>89</i>
<i>Figura 6.25. Caracterización Parámetros-S para etapa entrada salida en la banda de 2.45 GHz.....</i>	<i>91</i>
<i>Figura 6.26. Caracterización S11 para la etapa down-conversion 1-2 y entrada salida en la banda de 2.45 GHz con un ancho de banda de 110 MHz y ganancia del amplificador de IF a 0dB.....</i>	<i>92</i>
<i>Figura 6.27. Caracterización de las pérdidas de conversión para la etapa down-conversion 1-2 y entrada salida en la banda de 2.45 GHz con un ancho de banda de 110 Mhz y ganancia del amplificador de IF a 0dB.....</i>	<i>93</i>
<i>Figura 6.28. Caracterización de las pérdidas de conversión para la etapa de down-conversion 1-2 y entrada salida en la banda de 2.45 GHz con un ancho de banda de 110 Mhz. (a) con ganancia de amplificador de IF = 10 dB, (b) con ganancia de amplificador de IF = 20 dB.....</i>	<i>94</i>
<i>Figura 6.29. Caracterización SINR para la etapa down-conversion 1-2 y entrada salida en la banda de 2.45 GHz con un ancho de banda de 110 MHz.....</i>	<i>94</i>
<i>Figura 6.30. Caracterización S11 para la etapa up-conversion 1-2 y entrada salida en la banda de 2.45 GHz con un ancho de banda de 110 MHz.....</i>	<i>95</i>
<i>Figura 6.31. Caracterización de las pérdidas de conversión para la etapa up-conversion 1-2 y entrada salida en la banda de 2.45 GHz con un ancho de banda de 110 MHz y ganancia del amplificador de IF a 0dB.....</i>	<i>95</i>
<i>Figura 6.32. Caracterización SINR para la etapa up-conversion 1-2 y entrada salida en la banda de 2.45 GHz con un ancho de banda de 110 MHz.....</i>	<i>96</i>

Índice de tablas

<i>Tabla 2.1. Comparación de emuladores de canal [2].</i>	19
<i>Tabla 6.1. Caracterización Potencia de activación del switch.</i>	92
<i>Tabla 6.2. Caracterización Retardo de activación del switch.</i>	92

Capítulo 1. Introducción

Hoy en día, gracias a la presencia de procesos de automatización y dispositivos inteligentes en la industria se está dando una cuarta revolución industrial, conocida como la Industria 4.0. La Industria 4.0 es la promesa de una nueva revolución marcada por la aparición de nuevas tecnologías como la robótica, inteligencia artificial, el Internet of Things (IoT), entre otros.

Bajo este contexto, se espera que las comunicaciones inalámbricas desempeñen un papel esencial en la Industria 4.0, ya que traen nuevas oportunidades para la automatización industrial [1]. En un futuro, se espera que las redes industriales inalámbricas puedan llegar a sustituir las redes industriales cableadas, debido a que proporcionan numerosas ventajas, entre las que se encuentran: alta escalabilidad, despliegues más baratos, nuevos casos de uso en escenarios móviles, etc.[2]. Por ese motivo, existe un gran interés en sustituir las redes industriales cableadas por redes industriales inalámbricas [1].

Sin embargo, las comunicaciones inalámbricas en entornos industriales también presentan múltiples desafíos que impiden que se puedan desplegar de forma masiva. El desafío principal se encuentra en el medio de propagación, usualmente llamado canal de comunicaciones inalámbrico. Los problemas como el ruido, la interferencia, propagación por múltiples trayectos, variación del canal a lo largo del tiempo y la falta de línea de visión directa (LoS) modifican las propiedades de la señal de Radio Frecuencia (RF) en el receptor haciendo que la probabilidad de perder paquetes sea muy elevada. Primero, la propagación inalámbrica se caracteriza por tener normalmente una relación señal a ruido baja (20 dB o menos). Además, la dispersión temporal provocada por el multicamino distorsiona la señal y es necesario añadir elementos adicionales de protección a las señales transmitidas para que puedan ser correctamente demoduladas. Todos estos efectos son especialmente críticos en entornos industriales, en los que suelen existir muchos elementos metálicos, elementos ruidosos / interferentes [3].

Debido a todas estas razones, surge la necesidad de validar los sistemas de comunicación inalámbrica industriales sobre entornos reales. Sin embargo, la validación de los sistemas en la mayoría de los casos suele ser costosa y conlleva mucho tiempo. Para contrarrestar este problema, se puede realizar una validación del sistema en un laboratorio utilizando un dispositivo llamado emulador de canal. El emulador de canal es un dispositivo que permite generar un canal inalámbrico y, que, por lo tanto, permite testear el rendimiento de los sistemas de comunicaciones en una situación controlada y con cierta repetibilidad.

Por lo tanto, un emulador de canal inalámbrico es de gran utilidad para la temprana evaluación y validación de sistemas de comunicaciones inalámbricas, ya que ofrece un entorno controlado y repetible, donde se pueden simular diferentes entornos de propagación. La posibilidad de una validación temprana es importante para las redes inalámbricas que se implementan en entornos donde los escenarios tienen elementos metálicos que favorecen la aparición de multicamino y de desvanecimientos. En el caso de redes inalámbricas en entornos industriales como fábricas o sistemas de transporte, el uso de un emulador inalámbrico permite optimizar el diseño de las capas física (PHY) y de acceso al medio (MAC) de la red inalámbrica mientras se prueba con un modelo emulado del canal real. Para obtener las características de propagación del canal real primero se deben realizar mediciones de canal y generar un modelo matemático que represente lo más fielmente al canal real. Estos resultados serán críticos tanto en el diseño de la capa física de la red como para el modelo del canal real [4].

Un método alternativo para validar sistemas inalámbricos es hacer uso de simulaciones de capa física y MAC junto con diferentes modelos de canal. Sin embargo, esta aproximación está lejos de ser realista ya que las imperfecciones del hardware, que no son normalmente consideradas en este tipo de simulaciones, pueden modificar de forma sustancial los resultados. Por ello, un emulador de canal va a ser, en la mayoría de los casos, una aproximación mucho más realista que este tipo de simulaciones.

1.1. Descripción del proyecto

El proyecto se enmarca en la continuación de un diseño estable de un emulador de canal inalámbrico multipunto en banda base basado en una FPGA Kintex-7 con un ancho de banda de 100 MHz que cubre la banda ISM de 2.45GHz. Con objeto de mejorar el diseño del emulador de canal se decidió realizar un nuevo diseño del emulador. Actualmente se dispone de un diseño del emulador de canal banda base basado en una Zynq UltraScale+ MPSoC con un ancho de banda de 100 MHz que cubre el rango de frecuencias de 400 MHz a 6 GHz.

A continuación, se describe el contenido de cada uno de los capítulos que conforman el presente trabajo fin de máster.

En el Capítulo 1, se hace una introducción sobre la temática que se va a tratar en los siguientes capítulos y también se describen los objetivos del proyecto.

En el Capítulo 2, se ha realizado una revisión del marco teórico sobre emuladores de canal, es decir, los diferentes tipos de emulador de canal que existen,

como es la estructura interna de un emulador de canal y porque es de gran utilidad disponer de un emulador de canal en un laboratorio. Por otro lado, se ha realizado un estudio sobre las diferentes distorsiones que puede experimentar una señal electromagnética tras su paso por un canal inalámbrico y el modelado utilizado para implementar emuladores de canal.

En el Capítulo 3 en primer lugar, se ha llevado a cabo un estudio del emulador de canal basado en la FPGA Kintex-7, es decir, desde el estudio de que partes conforman el emulador hasta como se desarrolla la implementación del modelo de canal banda base complejo. En segundo lugar, se hace una descripción de la arquitectura del emulador de canal, donde se distingue la arquitectura correspondiente para realizar la comunicación entre los diferentes elementos del emulador y poder simular el entorno de un canal inalámbrico. Y, por otro lado, de qué forma se estructura la señal de control del emulador.

En el Capítulo 4 se muestran las diferentes herramientas de desarrollo empleadas en este proyecto, así como una guía de instalación de una máquina virtual Linux sobre el sistema operativo de Windows. Dicha máquina virtual se utiliza como entorno de desarrollo para interfaces de usuario tanto para PC como para la arquitectura ARM.

En el Capítulo 5 se describe el diseño y funcionalidad de las dos interfaces de usuario realizadas en este proyecto.

En el Capítulo 6 en primer lugar, se realiza un estudio del diseño de las partes que componen los frontend RF. En segundo lugar, el software empleado para realizar la caracterización de los frontend RF, así como el setup de medidas que se llevan han realizado. Y, por último, los resultados obtenidos de la caracterización de uno de los frontend RF en las bandas de 434 MHz, 868 MHz y 2.45 GHz.

En el Capítulo 7 se describen las conclusiones y líneas futuras del proyecto.

1.1.1. Objetivos del proyecto

Como se ha comentado anteriormente, para validar los sistemas de comunicación inalámbrica industriales sobre entornos hostiles, sirve de gran utilidad disponer de un sistema de validación en un laboratorio. Por lo tanto, diseñar un emulador de canal para sistemas de validación en un laboratorio tiene múltiples ventajas debido a que permite testear el rendimiento de un sistema de comunicación para posteriormente aplicarlo en un entorno industrial real.

Debido a estos factores, el objetivo principal del proyecto es construir un emulador de canal inalámbrico banda base con el que se puedan validar los

diferentes efectos que sufre una señal electromagnética tras su paso por el canal inalámbrico.

Dentro del objetivo principal del proyecto en este trabajo fin de máster se ha realizado:

1) la puesta en marcha de la máquina virtual como entorno de programación para el desarrollo de la interfaz de control Linux.

2) El desarrollo de la interfaz de control del emulador usando la herramienta QT, que se ejecuta en uno de los procesadores ARM A53 de la tarjeta Zynq UltraScale+.

3) La caracterización de los cuatro frontend de RF en las bandas de radio ISM, en concreto, en la banda de 2.45GHz, y en las frecuencias intermedias de 868MHz y 434MHz.

1.2. Enmarque del proyecto

El proyecto se enmarca en la actividad del centro de investigación IKERLAN en el campo de las Tecnologías de la Información y Comunicación. En concreto, es un proyecto propio del grupo de Sistemas de Comunicaciones.



Figura 1.1. Logo de Ikerlan.

Para cumplir con los objetivos del proyecto, se han repartido las tareas entre los integrantes del equipo de trabajo. En este documento se recoge la descripción general del proyecto, junto con el marco teórico necesario para la comprensión de la motivación del proyecto, y las partes específicas que han sido realizadas en este trabajo fin de máster.

Capítulo 2. Marco teórico

2.1. Emulador de canal inalámbrico

Como se ha descrito en el capítulo anterior, disponer de un emulador en el laboratorio que genere de forma controlada y repetible entornos hostiles resulta de gran utilidad para realizar una validación temprana de los sistemas de comunicaciones inalámbricas industriales. Por lo tanto, es necesario conocer cual es el comportamiento típico de los canales inalámbricos, como se modelan, y como se puede aplicar dicho modelado para la construcción de un emulador de canal.

En la Figura 2.1 se muestra el diagrama de bloques de un emulador de canal general. En primer lugar, un frontend de RF se encarga de convertir la frecuencia de operación en RF de la señal de entrada a frecuencia intermedia (IF). En segundo lugar, el denominado Complex-baseband channel model se encarga de efectuar el modelo equivalente de canal en banda base y de distorsionar la señal según el canal programado. Por último, la señal digital se convierte a analógica y con el frontend RF de salida se hace la up-conversion de IF a RF. Respecto al método de emulación de canal, lo más común es utilizar modelos de canal estocásticos basados en campañas de medida. Los modelos de canal estocásticos se suelen basar en el análisis de la respuesta al impulso (CIR - Channel Impulse Response) del canal y, por tanto, se suelen implementar usando filtros de respuesta finita (FIR - Finite Impulse Response), con coeficientes variantes. Este método también suele ser denominado tapped delay line, el cual es un caso particular del filtrado FIR.

Existen varios emuladores disponibles en el mercado y, en general, tienen características similares. Algunos de estos modelos junto con sus características más relevantes están recogidos en la Tabla 2.1.

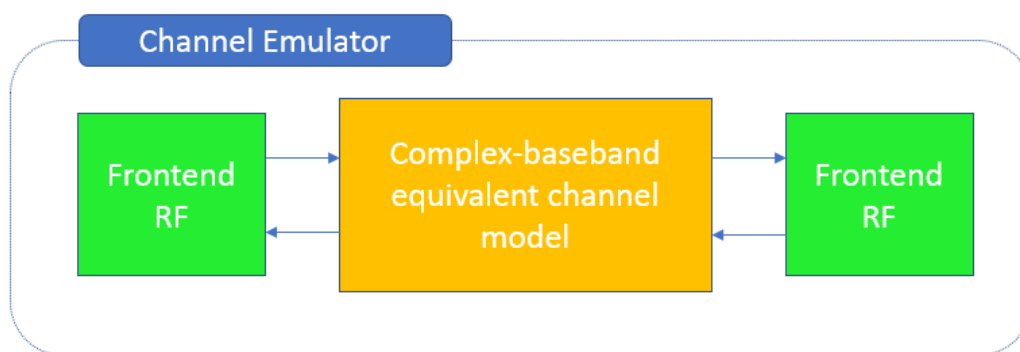


Figura 2.1. Arquitectura emulador de canal.

Channel emulator	Emulator Capacities			
	Max RF ports	Frequency Range (GHz)	Max Bandwidth (MHz)	Doppler Spread (kHz)
Anite Propsim F8 [5]	16	0.22-6	160	1.500
Spirent VR5 HD [6]	16	0.38-3.85	100	4
RFnest Digital D500 series [7]	96	0.02-6	250	0.200

Tabla 2.1. Comparación de emuladores de canal [2].

Como se ve en la tabla, estos emuladores cubren los rangos de frecuencia de hasta 6 GHz, excepto el modelo proporcionado por Spirent que cubre hasta los 3.85GHz y todos ellos soportan anchos de banda superiores a 100 MHz. Son capaces de recrear desvanecimiento y efectos de interferencias para testear sistemas Single Input Single Output (SISO) y Multiple Input Multiple Output (MIMO).

Sin embargo, los emuladores de canal son dispositivos con un coste monetario elevado, especialmente cuando su configuración es de alto rendimiento (gran ancho de banda, varios puertos I/O, gran rango de frecuencias). Es por ello, que una alternativa interesante a los emuladores comerciales es construir el emulador de canal.

Un emulador de canal, como se ha mencionado anteriormente, esta típicamente compuesto por los siguientes elementos: un frontend de entrada que realice la down-conversión de la señal RF a IF/ banda base, un dispositivo que se encargue de aplicar el modelo equivalente de canal banda base y por último un frontend de salida que realice la up-conversion de la señal IF/banda base a RF. Además, el frontend también consta de uno o varios conversores Analógico-Digital (ADC - Analog to Digital Converter) ADC que discretizan la señal recibida, y uno o varios conversores Digital-Analógico (DAC - Digital to Analog Converter) que transforman la señal del dominio discreto al continuo. Por último, el emulador posee un dispositivo programable que recibe la señal discreta generada por el ADC, aplica los modelos de canal sobre dicha señal discreta, y las envía al DAC. Para emular correctamente un canal en condiciones de tiempo real, el hardware del emulador debe tener poca latencia y ser reconfigurable. El dispositivo idóneo para realizar estas operaciones es una Field Programmable Gate Array (FPGA), ya que son dispositivos que permiten realizar una gran cantidad de operaciones en paralelo con una mínima latencia [8].

En la siguiente subsección se explica detalladamente las características de los canales inalámbricos, cuáles son las herramientas matemáticas utilizadas para modelar dichos canales, y de estas herramientas, cuáles pueden ser utilizadas en tiempo real, por lo que son convenientes para implementar un emulador de canal.

2.2. Canales inalámbricos

Un medio de transmisión inalámbrico es aquel en que la señal viaja desde un emisor a un receptor sin la necesidad de utilizar un medio físico para su propagación. Dicha propagación de la señal se realiza mediante el uso de ondas electromagnéticas no guiadas. Una característica de este tipo de canales es que son afectados de manera impredecible por los cambios que pudiesen producirse en su entorno de propagación [9]. En la Figura 2.2 se muestra el esquema de un canal inalámbrico, donde una fuente envía información a un destino.

De los elementos presentados en la Figura 2.2, el elemento de más relevancia en este caso es el canal inalámbrico. La fuente envía un mensaje que tiene que ser modulado para que la señal pueda ser enviada por el canal. La señal durante su propagación por el canal se ve deteriorada por consecuencia de diversos efectos, afectando tanto en el alcance como en la capacidad de la comunicación inalámbrica. Una vez la señal llega al receptor, es necesario demodular la señal para recuperar el mensaje emitido.

La señal transmitida va a ser atenuada y distorsionada por el canal de comunicaciones [10]. El primero de los efectos encontrados en un canal de comunicaciones es el ruido. El ruido puede surgir de varios elementos: ruido generado dentro del receptor, como ruido térmico, ruido de disparo en componentes pasivos y activos, etc. Por otro lado, el multicamino surge por las diferentes trayectorias de la señal transmitida hasta llegar al receptor. Algunos de los numerosos efectos que puede sufrir la señal son: Absorción, Difracción y Dispersión. Además, cuando la señal viaja por un medio inalámbrico puede sufrir diferentes efectos, como pueden ser una línea de visión directa entre el transmisor y el receptor (Pathloss), sufrir desvanecimientos lentos (Shadowing) o también

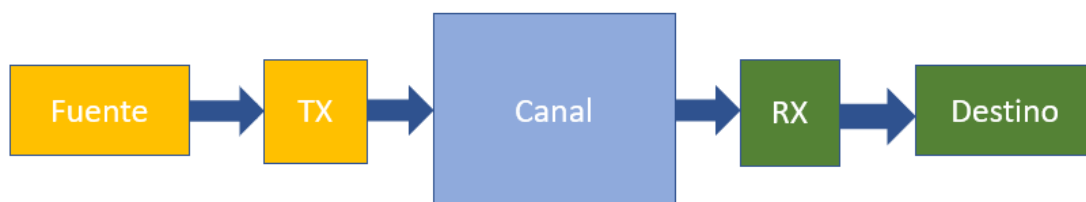


Figura 2.2. Esquema canal inalámbrico.

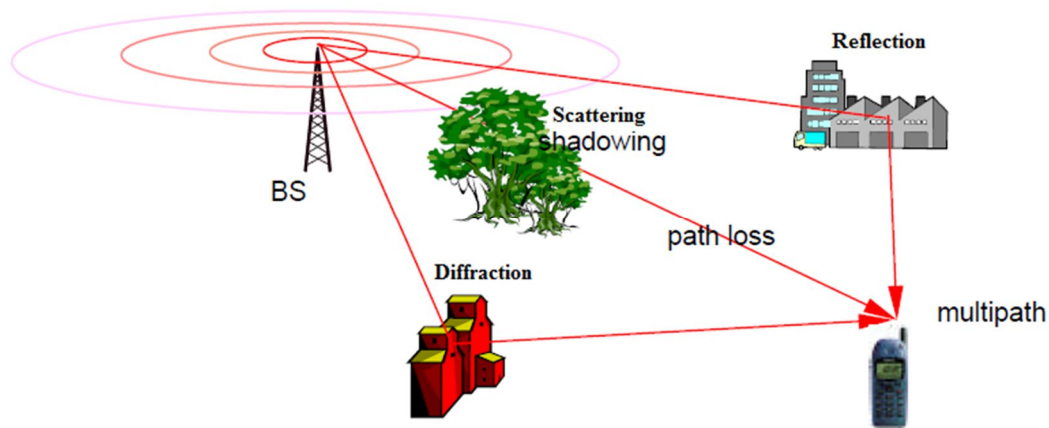


Figura 2.3. Efectos de la propagación en un canal inalámbrico [11].

puede sufrir múltiples rebotes lo que es conocido como desvanecimiento rápido o multipath. Cuando la señal viaja a través de un medio inalámbrico se ve sometida a las distorsiones indicadas en la Figura 2.3 y que son descritas a continuación:

- Pathloss: decaimiento de la señal provocada por la distancia entre el transmisor y receptor.
- Superpuesta a la pérdida de trayectoria se encuentran los desvanecimientos lentos también conocido como shadowing. En este caso el nivel de potencia o pérdidas experimentan variaciones significativas sobre el valor mediano extraído con los modelos de propagación.
- Los desvanecimientos rápidos (fast fading) ocurren cuando la señal llega al receptor a través de diferentes vías debido a las reflexiones de la señal en diferentes elementos del entorno (multicamino o multipath). La interferencia constructiva y destructiva de las componentes del multicamino dan lugar a variaciones significativas en la amplitud y fase de la señal recibida. Estas variaciones de amplitud se dan con pequeños cambios de posición relativa entre el transmisor y receptor.

Teniendo en cuenta estos efectos, es de gran utilidad realizar pruebas sobre estos canales inalámbricos de forma controlada, tanto para diseñar redes de comunicaciones inalámbricas como predecir la cobertura e interferencias de la señal, etc. En la siguiente sección se explican los diferentes métodos que existen para modelar un canal radio, centrandolo en los modelos estocásticos, que son los de interés para implementar emuladores de canal.

2.2.1. Modelado de canal

El modelado de canal inalámbrico tiene como objetivo estudiar los diferentes entornos en los cuales tienen lugar comunicaciones inalámbricas. Los modelos de

canal tratan de predecir el canal inalámbrico para conocer cómo va a ser distorsionada una señal electromagnética. Se distinguen dos principales enfoques en el modelado de canales: los modelos de canales deterministas o analíticos y los modelos de canal estadísticos o estocásticos.

Los *modelos deterministas* constan de reproducir todos los parámetros geométricos y electromagnéticos asociados a un escenario. Posteriormente esos datos se almacenan en bases de datos utilizadas para simular en un ordenador la respuesta al impulso del canal. Para simular la respuesta al impulso del canal se hace uso de las ecuaciones de Maxwell. Dada su complejidad, estas ecuaciones se suelen resolver utilizando métodos numéricos como el trazado de rayos o el método de diferencias finitas [11]. Además, este método no es práctico para modelar escenarios generales puesto que se necesita un gran conocimiento del medio de propagación, propiedades de las antenas y posee un coste computacional muy alto lo que hace que sea impracticable en tiempo real.

Los *modelos estocásticos* describen el entorno según distribuciones estadísticas a partir de campañas de medidas realizadas en distintos entornos de propagación. Los datos obtenidos son analizados usando las funciones de densidad de probabilidad (PDF – Probability Density Function) de los parámetros de transmisión de cada entorno. Se caracterizan por requerir un menor esfuerzo computacional y ser capaces de proporcionar modelos de canal para condiciones más generales. Dentro de los modelos estocásticos se distinguen dos grupos, los modelos estocásticos geométricos y los no geométricos.

- Los *modelos estocásticos geométricos* se derivan de una distribución estadística predefinida de la dispersión mediante la aplicación de las leyes fundamentales de la propagación de ondas. Dichos modelos se pueden adaptar a diferentes escenarios cambiando la forma de la región de dispersión y su correspondiente función densidad de probabilidad (FDP) [12].
- Los *modelos estocásticos no-geométricos* determinan las características de un canal de manera completamente estocástica sin suponer ninguna geometría subyacente. En el siguiente capítulo nos centraremos en el modelado de canal estocástico no geométrico, puesto que son los modelos que más se utilizan para caracterizar el comportamiento del canal dentro de un entorno industrial [13][14].

De estos dos modelos, el más interesante para realizar emulación de canal es el segundo, ya que permite evaluar un canal inalámbrico de forma general, y no es necesario tener detalles geométricos sobre el entorno de propagación.

2.2.1.1. Large Scale Fading

El Large Scale Fading es la atenuación producida por la distancia existente entre el transmisor y el receptor y la pérdida de trayectoria de la señal debido a objetos de gran tamaño. Dentro del Large Scale Fading, se suelen distinguir dos categorías, el Path Loss, y el shadowing.

2.2.1.1.1. Path loss

El fenómeno de path loss es la atenuación que sufre la señal transmitida en el trayecto entre el emisor y el receptor, relacionada con la distancia entre ambos. En telecomunicaciones, el modelo de propagación en espacio libre se refiere a la transmisión con una línea de visión directa y se modela según la ecuación de Friis [15]

$$\frac{P_r}{P_t} = \left(\frac{\lambda}{4\pi d} \right)^2 G_t G_r, \quad (2.1)$$

donde P_r se corresponde con la potencia recibida, P_t con la potencia transmitida, G_t la ganancia de la antena transmisora, G_r la ganancia de la antena receptora, d la distancia entre la antena transmisora y receptora, y por último λ representa la longitud de onda de la portadora.

2.2.1.1.2. Shadowing

El shadowing provoca una atenuación en la señal debida a elementos de gran tamaño que generan zonas de sombra. Se caracteriza por tener una variación lenta en canales móviles. El shadowing se suele modelar utilizando una función log-normal, según la siguiente expresión [16]

$$f(\gamma_p) = \frac{1}{\sqrt{2\pi}\sigma_{dB}} \exp - \left(\frac{(\gamma_p - L_p(dB))^2}{2\sigma_{dB}^2} \right). \quad (2.2)$$

Las pérdidas $\gamma_p(dB)$ a la distancia d se modela como una variable aleatoria Gaussiana con media $L_p(dB)$ y varianza σ_{dB}^2 la cual puede ser determinada empíricamente a partir de campañas de medidas y se relaciona con la variabilidad de los emplazamientos. Los valores de la desviación de acuerdo a modelos de propagación en centros urbanos toman valores en el rango 5-12 dB, siendo mayor en áreas suburbanas y menor en áreas rurales [16]. Las pérdidas por path loss y shadowing típicamente cambian de forma lenta con el tiempo.

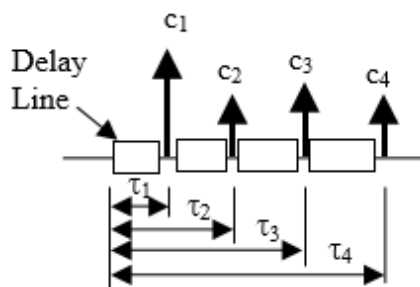


Figura 2.4. Tapped delay line model [2].

2.2.1.2. Small Scale Fading

Los desvanecimientos rápidos o Small Scale Fading es un efecto que se produce a pequeña escala y se da cuando las señales llegan al receptor por dos o más caminos diferentes (multicamino). Dependiendo de la fase de estas señales y de su retardo relativo, estas señales pueden ser sumadas de forma constructiva o destructiva causando rápidas fluctuaciones en la intensidad de la señal recibida.

El modelo equivalente de canal más utilizado es el basado en el modelo de línea tapped delay. El modelo tapped delay (TDL) consiste en representar la respuesta al impulso de un canal $h(t, \tau)$ con múltiples rutas mediante un número discreto de impulsos según la siguiente expresión [17]

$$h(t, \tau) = \sum_{i=0}^{I-1} c_i(t) \delta(t - \tau_i(t)), \quad (2.3)$$

donde $\delta(\cdot)$ es la función delta de Dirac, I es el número de paths, c_i es el peso complejo de la delta i y τ_i es el retardo de cada i -th paths. Por ejemplo, como se muestra en la Figura 2.4, se tiene un modelo de tapped delay con cuatro tap donde cada línea presenta su propio delay.

Por simplicidad, el retardo τ_n para cada path se supone constante y bajo la condición de que solo se tiene en cuenta el efecto del small scale fading; con lo que la expresión anterior se reduce a:

$$h(t) = \sum_{i=0}^{I-1} c_i(t) \delta(t - \tau_i). \quad (2.4)$$

En la Figura 2.5 se muestra la respuesta al impulso de un canal que sufre desvanecimientos rápidos. Por un lado, se ve el efecto de dispersión temporal donde se tienen réplicas de la señal transmitida atenuadas y desplazadas según el retardo

de propagación. Por otro lado, el efecto de varianza temporal es consecuencia del movimiento de los nodos que se comunican o por el movimiento de los elementos del escenario. Dichos efectos son explicados con más detalle en las siguientes secciones.

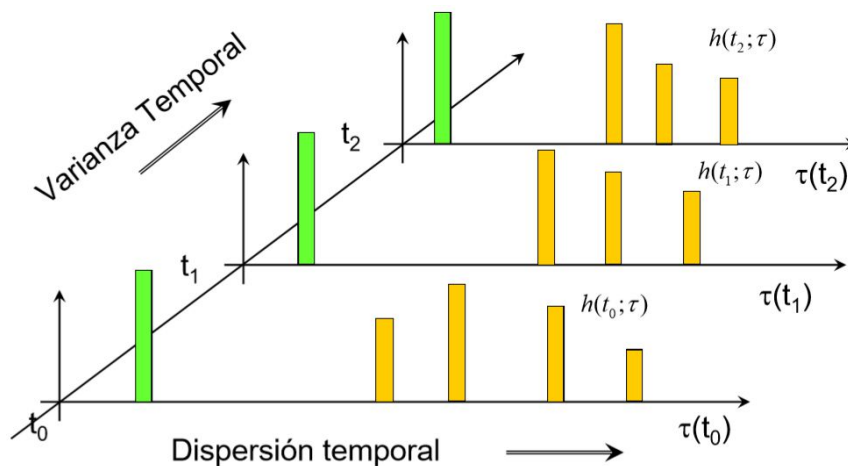


Figura 2.5. Respuesta al impulso de un canal multirrayecto.

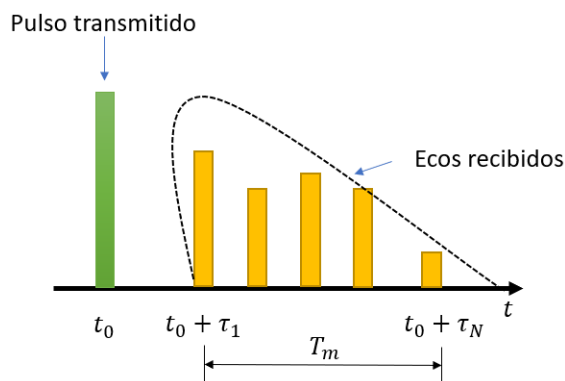


Figura 2.6. Dispersión temporal.

2.2.1.2.1. Dispersión temporal del canal

La dispersión temporal del canal se define como la dispersión temporal de la señal en numerosos ecos que llegan a la antena receptora mediante señales retardadas. Este tipo de efecto provoca variaciones en los tiempos de llegada de la señal, es decir, se tiene la misma señal replicada un determinado número de veces. La dispersión temporal está causada por el multipath del canal.

Esta dispersión temporal implica tener interferencia entre símbolos, que degrada la calidad de la señal recibida y aumenta la probabilidad de error de bit (BER, bit error rate).

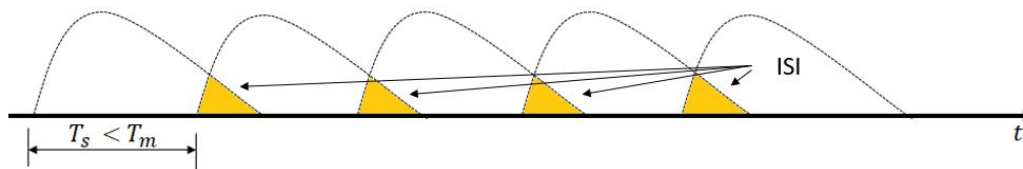


Figura 2.7. Señal recibida con ISI.

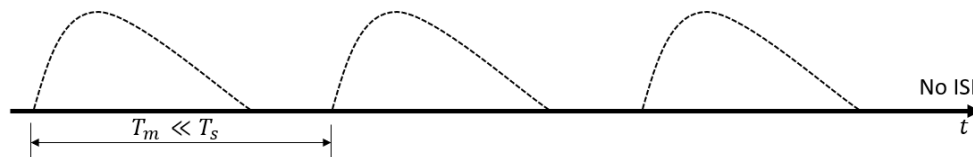


Figura 2.8. Señal recibida sin ISI.

En la Figura 2.6 se observa como de un pulso transmitido se tienen diferentes ecos recibidos cada uno de ellos retardados τ . Se define T_m como el máximo retraso, es el retraso entre la primera y última componente de la señal. En un canal con dispersión temporal, la relación entre el máximo retraso T_m y el periodo de símbolo T_s puede verse como dos categorías de degradación diferente: desvanecimiento selectivo en frecuencia y no selectividad en frecuencia o flat fading.

Un canal se dice que presenta **selectividad en frecuencia** cuando $T_m > T_s$, las componentes multitrayecto del receptor de un símbolo se extienden más allá de la duración de los símbolos, causando interferencia entre símbolos (ISI) (Figura 2.7).

Un canal se dice que presenta **flat fading** si $T_m \ll T_s$. En este caso, todas las componentes multitrayecto del receptor llegan dentro del periodo de símbolo y, por lo tanto, no se generan interferencias (Figura 2.8).

Sin embargo, este tipo de análisis no permite extraer conclusiones que permitan modelar el canal inalámbrico, sino que representan las características del canal inalámbrico en un instante temporal determinado. Para extraer conclusiones más generales, se caracteriza el conjunto de CIR del canal de forma estocástica.

Para describir el comportamiento del multitrayecto en el **dominio del tiempo** se hace uso del perfil potencia-retardo (PDP – Power Delay Profile). La PDP representa la potencia media del canal en función del retardo (τ), y se obtiene a partir de calcular la media de un conjunto de respuestas al impulso. En la Figura 2.9 se muestra la potencia de la señal recibida $P(\tau)$ la cual se suele expresar en potencia normalizada (dB) frente al máximo delay T_m .

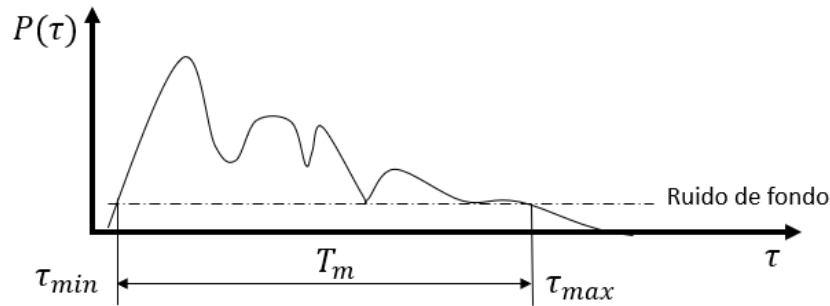


Figura 2.9. Perfil potencia-retardo (PDP).

A partir de la PDP se puede calcular la dispersión media del retardo o *mean delay spread* mediante la siguiente expresión [18]

$$\bar{\tau} = \frac{\int_{\tau_{min}}^{\tau_{max}} \tau P(\tau) d\tau}{\int_{\tau_{min}}^{\tau_{max}} P(\tau) d\tau}. \quad (2.5)$$

A continuación, se puede calcular el *rms delay spread* mediante la siguiente expresión

$$\tau_{rms} = \sqrt{\frac{\int_{\tau_{min}}^{\tau_{max}} \tau^2 P(\tau) d\tau}{\int_{\tau_{min}}^{\tau_{max}} P(\tau) d\tau} - (\bar{\tau})^2}. \quad (2.6)$$

τ_{rms} es una figura de mérito que permite evaluar como de dispersa es la PDP. Si la PDP tiene poca dispersión, τ_{rms} será bajo, mientras que, si la PDP es muy dispersa, τ_{rms} será alto.

Para describir el comportamiento del multitrayecto en el **dominio de la frecuencia** se hace uso del concepto de coherencia del canal. Realizando la transformada de Fourier de $P(\tau)$ se obtiene la función de correlación en el dominio de la frecuencia (Ver Figura 2.11), representa la correlación entre la respuesta del canal a dos señales en función de la separación en frecuencia entre ambas [19].

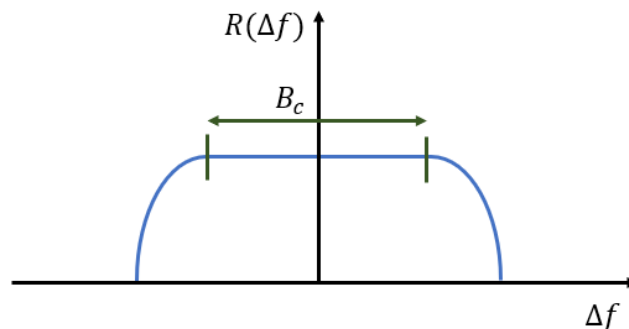


Figura 2.11. Función de correlación y ancho de banda de coherencia.

El ancho de banda de coherencia B_c es una medida estadística del rango de frecuencias sobre las cuales el canal afecta por igual a todas las componentes espectrales. El carácter selectivo del canal en frecuencia depende del ancho de banda de la señal transmitida comparado con el inverso del delay spread. La expresión utilizada comúnmente para evaluar el ancho de banda de coherencia es la siguiente [20].

$$B_c = \frac{1}{5\tau_{rms}}. \tag{2.7}$$

Un canal es **selectivo en frecuencia** cuando $B_s > B_c$, donde B_s es el ancho de banda de la señal transmitida. Por lo tanto, el canal se distorsiona y no conserva el espectro de la señal transmitida, este efecto ocurre cuando las componentes de la señal

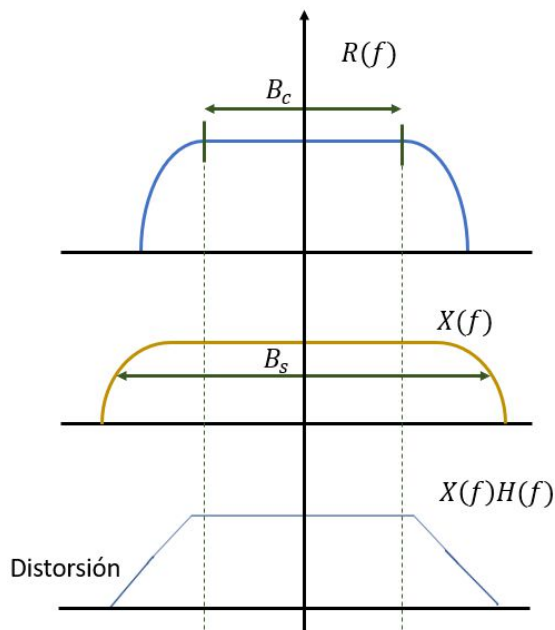


Figura 2.10. Distorsión selectiva en frecuencia.

espectral no se ven afectadas de igual manera por el canal.

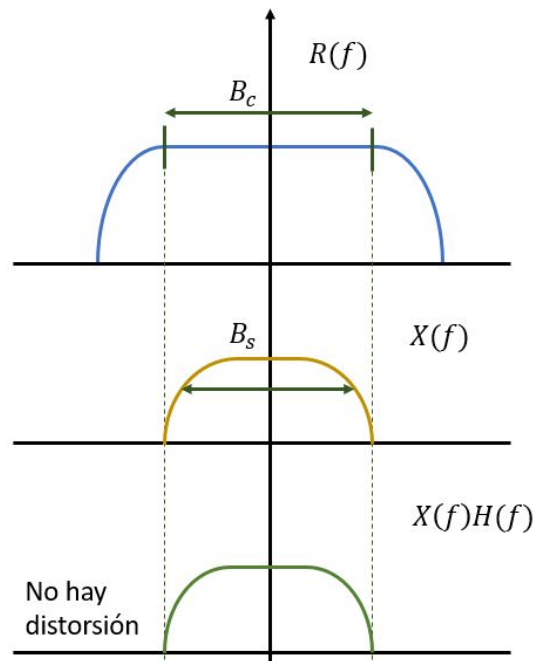


Figura 2.12. Canal sin distorsión en frecuencia.

En cambio, si $B_s < B_c$, se trata del caso de un canal no selectivo en frecuencia (no existe distorsión), donde el canal conserva el espectro de la señal transmitida ya que trata todas las frecuencias por igual.

2.2.1.2.2. Distribuciones fading

Debido a la variación del entorno la atenuación de cada uno de los taps y su fase es variante con el tiempo. La envolvente compleja de cada tap se puede modelar mediante el uso de distribuciones estadísticas. Las distribuciones que mejor se ajustan a los canales inalámbricos debido a sus propiedades son la distribución Rayleigh, la distribución Rice, y la distribución Nakagami.

- Distribución de envolvente tipo Rayleigh

El desvanecimiento Rayleigh suele aparecer en entornos donde no hay línea de visión (NLOS) directa entre el transmisor y receptor. En esa situación, la amplitud compleja de cada uno de los taps se puede modelar como una gaussiana bidimensional

$$h \sim CN(0,1), \tag{2.8}$$

donde $CN(0,1)$ se denota como una variable aleatoria normal compleja con media $\mu = 0$ y varianza $\sigma^2 = 1$. La magnitud de esta distribución da como resultado la ganancia del canal, y dicha distribución sigue una distribución Rayleigh,

- Distribución de envolvente tipo Rice

Si el canal tiene una fuerte componente dominante (LOS), el canal con desvanecimiento puede ser modelado una variable aleatoria Gaussiana con media distinta de cero.

$$h \sim CN(h_{LOS}, \sigma^2), \quad (2.9)$$

donde la media h_{LOS} es un escalar complejo el cual se fija en diferentes usos del canal. Tiene una probabilidad baja de sufrir interferencias destructivas fuertes. La proporción de potencia entre las componentes LOS y NLOS se representa según el factor K de la distribución de Rice según la siguiente expresión

$$K = \frac{|h_{LOS}|^2}{\sigma^2}. \quad (2.10)$$

Cuando la amplitud del pico de la señal predominante tiende a 0, la distribución Rice tiende a la distribución Rayleigh. Es decir, la distribución Rayleigh es un caso particular de la distribución Rice para $K = 0$. En cambio, si la amplitud de la señal toma valores muy grandes, la distribución Rice tiende a ser una constante, y el canal resultante se modelaría como un canal gaussiano [16].

- Distribución de envolvente tipo Nakagami

La distribución de tipo Nakagami proporciona un buen ajuste de medidas empíricas de desvanecimientos en áreas urbanas. Esta distribución está relacionada con la distribución gamma Γ , un parámetro para controlar la propagación w y un parámetro de forma μ . Su función de densidad sigue la siguiente formulación:

$$f(x) = \frac{2\mu^\mu}{\Gamma(\mu)w^\mu} x^{2\mu-1} e^{-\frac{\mu x^2}{w}}. \quad (2.11)$$

Para $\mu = 1$ la distribución de Nakagami coincide con la Rayleigh y en el caso que $\mu > 1$ la distribución coincide con una Rice.

2.2.1.2.3. Variación temporal del canal debido al movimiento

La variación temporal del canal es consecuencia del movimiento de la antena transmisora o receptora, u objetos que provocan cambios en la propagación y, en consecuencia, en el canal. Este efecto de variación temporal del canal también es conocido como efecto Doppler el cual es directamente proporcional a la velocidad del móvil con respecto a la frecuencia de onda recibida. La máxima desviación de la frecuencia Doppler se muestra en la siguiente expresión:

$$f_d = \frac{vf_c}{c}, \quad (2.12)$$

donde v es la velocidad relativa entre el transmisor y el receptor móvil en f_c es la frecuencia de la portadora y c la velocidad de la luz.

El análisis de la variación del canal se realiza habitualmente en el dominio de la frecuencia. Para ello, se hace uso de la función de dispersión Doppler o espectro Doppler. El espectro Doppler representa la densidad espectral de potencia en función del desplazamiento de frecuencia Doppler y se obtiene a partir del módulo de la transformada de Fourier de los coeficientes complejos $c_n(t)$. En la Figura 2.14 se muestra la densidad espectral de potencia de un espectro Doppler de tipo Jakes. En este caso, el ensanchamiento espectral es consecuencia de las rápidas variaciones de los coeficientes complejos $c_n(t)$.

Este efecto produce un desplazamiento en las frecuencias espectrales de la señal transmitida (ensanchamiento de la señal transmitida) originando un desvanecimiento selectivo en el tiempo y distorsión de la señal recibida.

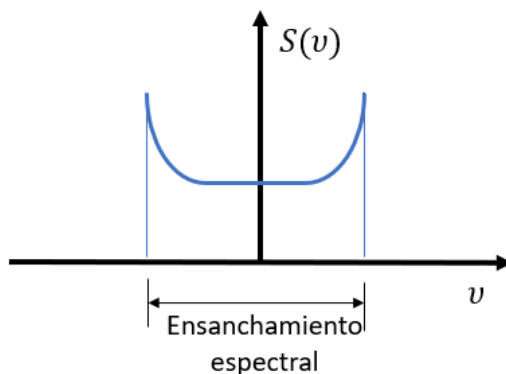


Figura 2.13. Potencia espectral Doppler.

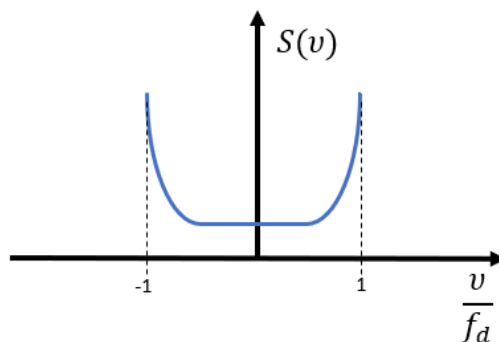


Figura 2.14. Jakes Doppler Spectrum.

2.2.1.2.4. Distribuciones Doppler

Entre los diferentes modelos de espectro Doppler se distinguen los siguientes:

- Distribución Jakes

El espectro Doppler clásico para un modelo de distribución Jakes se muestra en la Figura 2.14, siendo v la velocidad del móvil en (m/s) y f_d es la frecuencia Doppler. Esta distribución se suele dar en canales en los que los nodos que se están comunicando están en movimiento [21].

$$S(v) = \frac{1}{\pi f_d \sqrt{1 - \left(\frac{v}{f_d}\right)^2}}, |v| \leq f_d, \quad (2.13)$$

En el caso del espectro de tipo Jakes, la densidad espectral de potencia es máxima en los extremos coincidiendo con la velocidad máxima del vehículo. La velocidad Doppler se obtiene a partir de la velocidad relativa de los vehículos. Si los vehículos se mueven según la misma dirección y sentido la velocidad Doppler será cero, en cambio, si se mueven en sentidos opuestos la velocidad relativa entre los nodos es máxima ocasionando que mayor parte de la densidad espectral de potencia se concentre en los extremos.

- Distribución Bell

La distribución Bell (Figura 2.15) viene determinada por la siguiente expresión:

$$s(f) = \begin{cases} a_0 + a_2 \left(\frac{v}{f_d}\right)^2 + a_4 \left(\frac{v}{f_d}\right)^4 & |v| \leq f_d, \\ 0 & |v| > f_d, \end{cases} \quad (2.14)$$

donde a_0 , a_2 , y a_4 toman valores reales y son los coeficientes del polinomio del espectro, que dependen del entorno de propagación y v es la velocidad del entorno. Esta distribución se suele dar en escenarios en los que los nodos son estáticos, y los elementos del entorno se mueven.

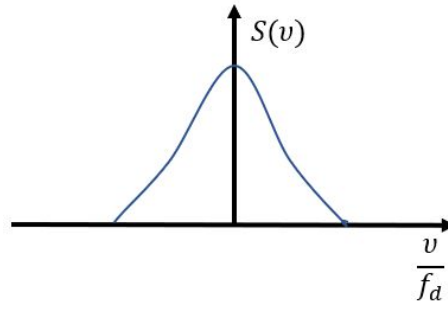


Figura 2.15. Bell Doppler Spectrum.

Capítulo 3. Emulador de canal inalámbrico

Como ha quedado descrito en el capítulo anterior, el medio de propagación inalámbrico es muy complejo e imprevisible, especialmente en escenarios industriales. Por lo tanto, un emulador de canal inalámbrico es de gran utilidad para simular los diferentes entornos de propagación que pueden afectar al canal inalámbrico y de esta forma testear el rendimiento de los sistemas de comunicaciones.

En el presente capítulo se describirá la necesidad que tiene el grupo de comunicaciones de Ikerlan en desarrollar un emulador de canal. En primer lugar, se comentarán los antecedentes, es decir, un primer diseño de emulador realizado por el grupo de comunicaciones y que es previo al emulador objeto de este documento. Dicho emulador tenía ciertas limitaciones en cuanto a capacidad de computo, y por ello, se decidió construir un segundo emulador. En segundo lugar, se explicará la arquitectura del nuevo diseño de emulador.

3.1. Antecedentes emulador de canal

Como parte de antecedentes, se dispone de un diseño estable de un emulador de canal inalámbrico multipunto en banda base basado en una FPGA Kintex-7 con un ancho de banda de 100 MHz y el cual cubre la banda ISM de 2.45GHz. La arquitectura del emulador de canal se muestra en la Figura 3.1 [4].

El emulador de canal se encuentra conectado a dos transceptores RF los cuales forman un enlace inalámbrico. Como se mencionó anteriormente cuenta de una etapa para realizar la conversión de frecuencia tanto descendente como ascendente de RF a IF (2.45 GHz a 61.44 MHz respectivamente) de forma que cualquier transceptor RF que trabaje en la banda ISM puede ser conectado al emulador. Cabe mencionar que la etapa de conversión RF/IF puede ser adaptada a otras frecuencias cambiando la frecuencia de los osciladores locales y de los filtros de RF. En la Figura 3.2 se muestra la etapa conversora, en la cual para dividir el puerto bidireccional de

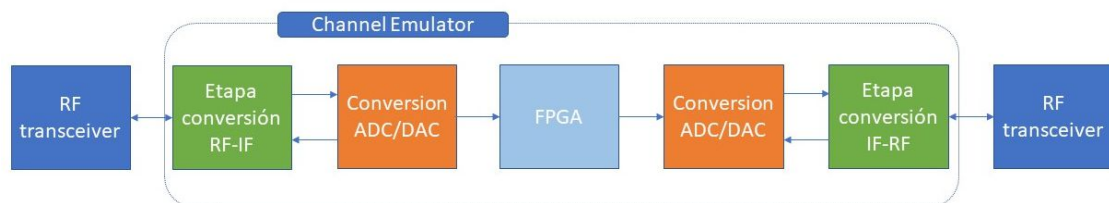


Figura 3.1. Diagrama de bloques emulador de canal.

la señal RF entrada/salida y alimentar la entrada y salida del modelo equivalente de canal se ha utilizado un switch controlado por un detector de potencia.

A la hora de realizar la etapa de RF a IF o down-conversion, la señal de entrada se divide por medio de un splitter. Parte de la señal va hacia el detector de potencia el cual se encarga de controlar el switch de RF, cuando detecta señal de entrada permite la transmisión entre los puertos RF_0 y RF_1 . Después la señal de salida del puerto RF_1 atraviesa un oscilador local el cual baja la señal a 61.44Mhz, y finalmente la señal de IF antes de ser entregada al modelo equivalente de canal pasa por un filtro de IF el cual se encarga de anular los productos de intermodulación.

Respecto a la etapa de IF a RF o up-conversion, la señal proveniente del modelo complejo banda base es filtrada y posteriormente se sube a frecuencia de 2.45 GHz. Antes de que la señal llegue al switch un filtro paso banda filtra la señal según la frecuencia central.

Después de la conversión de la señal de RF a IF, esta se discretiza usando el conversor ADC y se transmite al procesador banda base, es decir, a la FPGA, que se encarga de implementar el modelo de canal. Para aplicar los efectos del canal se utiliza el modelo tapped delay line en el dominio discreto según la siguiente expresión

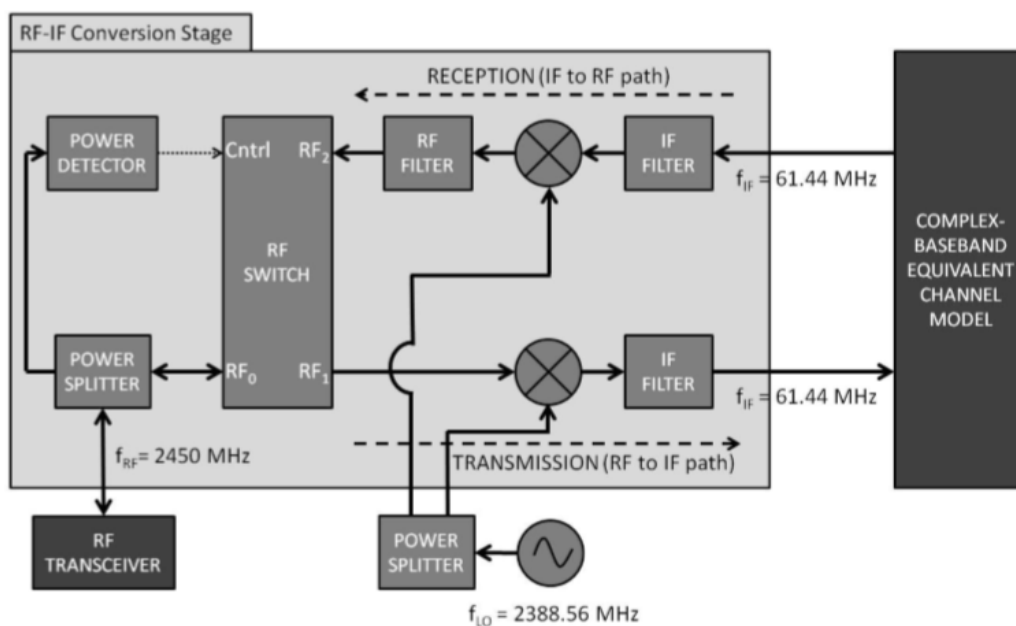


Figura 3.2. Diagrama de bloques etapa RF/IF conversion [4].

$$h[n] = \sum_{i=0}^{I-1} c_i[n] \delta[n - \frac{t_i}{T}], \quad (3.1)$$

donde $\delta[n]$ es la delta de Krocneker, I es el número de paths, $c_i(n)$ es la ganancia compleja de cada path y τ_i es el retardo de cada path, que debe ser múltiplo de T para que $\frac{\tau_i}{T} \in \mathbb{Z}$. Por lo tanto, el canal se representa como la suma de diferentes pesos $c_i(n)$ en diferentes instantes temporales que varían según los cambios del espectro Doppler.

Una vez se ha aplicado los efectos del canal a la señal de entrada esta se convierte del dominio digital al dominio analógico a través del DAC. Finalmente, la señal se convierte de IF a RF mediante el frontend de salida y la señal resultante en RF habrá sido distorsionada según el modelo de canal implementado en la FPGA. Pese a ser completamente funcional, este emulador presenta algunas limitaciones, que se describen en las siguientes subsecciones.

3.1.1. Modelo de frontend

Presenta un modelo de frontend que es funcional para la banda de 2.45 GHz, las etapas de Up-Conversion y Down-Conversion se realizan en un solo paso de frecuencia intermedia, generando problemas con el filtrado de señales adyacentes y espurios. Además, los perfiles de shadowing se tienen que aplicar en digital, lo que limita el rango de shadowing aplicable, ya que el rango dinámico de los DAC es limitado. Es decir, no es posible aplicar un shadowing elevado sin perder gran parte de la precisión numérica de la señal. Debido a estos factores, se decidió realizar un nuevo diseño del frontend de RF el cual permita cubrir las bandas frecuenciales comprendidas entre 400MHz y 6GHz, con dos etapas de frecuencia intermedia, un amplificador de IF en la etapa de down-conversion y atenuador variable en la etapa de up-conversion.

3.1.2. FPGA

Como se ha mencionado anteriormente, el emulador de canal estaba basado en una FPGA Kintex-7 la cual cuenta con una serie de limitaciones en cuanto a recursos hardware.

En primer lugar, no dispone de los recursos necesarios para realizar una emulación precisa de banda ancha debido a que el número de Procesadores Digitales de señal (DSP - *Digital Signal Processor*), que se utilizan principalmente para implementar filtros FIR y los modelos de canal, es reducido. De los recursos

disponible de la FPGA, se cuenta con una total de 840 DSP [22], los cuales 814 son utilizados [23]. Por lo tanto, los recursos que quedan disponibles en la FPGA son escasos. Si se tiene en cuenta que para implementar un solo tap es necesario hacer una multiplicación compleja y esto supone 3 multiplicaciones complejas (es decir, 3 DSP), es evidente que la implementación del emulador de canal tiene un consumo de DSP muy alto. Esto provoca una limitación a la hora de implementar taps por canal. Esta limitación provoca que solo se puedan emular un tapped delay line de 2 taps por canal en caso de que el sistema trabaje en modo MIMO. Dependiendo del canal de comunicaciones, esta cantidad de taps no permite emular de forma realista un canal, que puede tener 16 o más taps [24]. Además, la FPGA no cuenta con un microcontrolador, sino que el microcontrolador es implementado en la lógica programable. Esto provoca serias limitaciones en la capacidad de computo del microcontrolador, ya que su velocidad de reloj no puede ser elevada. Debido a estas limitaciones se decidió realizar el porting a la nueva plataforma Zynq UltraScale +, la cual dispone de un número mayor de DSP (un total de 2520 [25]) y varios cores físicos haciendo posible mayores velocidades de procesamiento. Gracias a la mejora en la plataforma se podrán emular un número mayor de taps por canal y se dispondrá de una mayor precisión en los cálculos numéricos. Por otro lado, el hecho de disponer de varios microcontroladores permite que en cada uno de ellos se ejecute un programa en función de las funcionalidades que se quieran aportar al emulador.

3.1.3. Interfaz de control

La interfaz que se utiliza para el control del emulador es una interfaz Windows mediante la cual se ejecuta en un PC externo al emulador y el control de la comunicación se realizaba a través del puerto ethernet. Por lo tanto, disponer en la Zynq UltraScale+ de varios procesadores permite disponer una interfaz integrada dentro de uno ellos. Dicha interfaz Linux incluyen tanto la integración del uso de periféricos como el control completo del emulador.

3.2. Nueva arquitectura emulador de canal

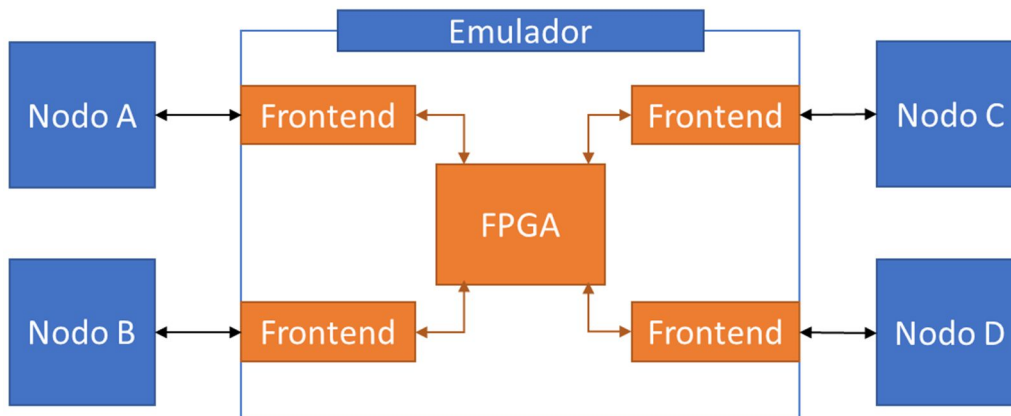


Figura 3.3. Sistema MIMO 4x4 emulador de canal.

En cuanto al diseño e implementación del nuevo emulador de canal, se dispone de un emulador de canal inalámbrico multipunto en banda base basado en una Zynq UltraScale+ MPSoC con una frecuencia de muestreo de la señal digital banda base de 122.88 MHz [4]. Dicho emulador de canal posee un ancho de banda de 110 MHz y cubre el rango de frecuencias de 400 MHz a 6 GHz. El diseño del emulador de canal se ha realizado de forma que se puedan conectar 4 nodos inalámbricos bidireccionales a los 4 frontends que componen el emulador. El hecho de poder conectar cuatro nodos al emulador hace que se puedan utilizar diferentes topologías de red hasta un sistema MIMO 4x4 (Figura 3.3) y que se puedan emular hasta 6 canales lógicos. Como se muestra en la Figura 3.3 se pueden conectar hasta 4 nodos (nodo A, nodo B, nodo C y nodo D) que se conectan a los 4 frontends.

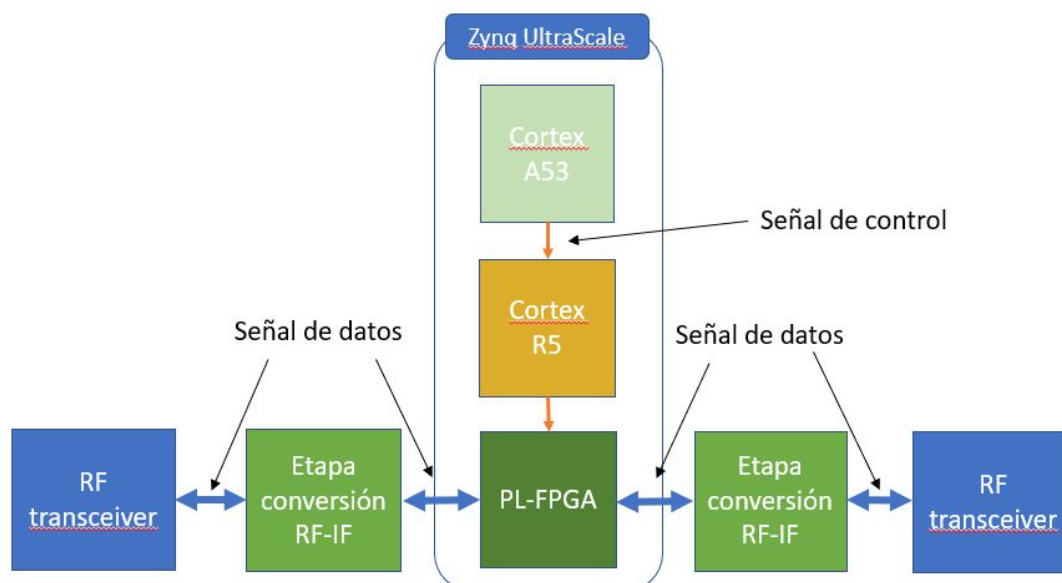


Figura 3.4. Arquitectura nuevo emulador de canal.

La arquitectura del nuevo emulador de canal se puede observar en la Figura 3.4. Por un lado, se distingue la arquitectura correspondiente a la transmisión/recepción de señal de datos, es decir, la comunicación entre los diferentes elementos para simular los efectos de un canal inalámbrico. Y, por otro lado, la arquitectura de control del emulador.

Como se ha explicado anteriormente, el emulador de canal recibe una señal de datos a través de los transceptores RF. Posteriormente un frontend RF realiza la etapa de conversión de RF/IF. En este caso se ha realizado un frontend superheterodino debido a las ventajas que presenta en cuanto a espurios reducidos y calidad de filtrado. La primera conversión se realiza de la frecuencia central hasta 645 MHz, y la segunda conversión de 645 MHz a 61.44 MHz, respectivamente. El diseño del frontend se explica dentro del capítulo correspondiente a su caracterización. A continuación, el ADC discretiza la señal recibida. A continuación, la señal llega a la Zynq UltraScale+ donde se aplican los modelos de canal sobre la señal discreta recibida, y las envía al DAC. Finalmente, el frontend de salida convierte la señal de IF a RF.

En las siguientes secciones se describe la arquitectura correspondiente al control del emulador. En primer lugar, como se realiza la comunicación entre los procesadores de la Zynq UltraScale y en segundo lugar como se lleva a cabo la programación de los frontend de RF.

3.2.1. Zynq UltraScale + MPSoC

La Zynq UltraScale está compuesta de dos partes: una unidad de lógica programable (FPGA) y un sistema de procesamiento (microcontrolador). Dentro del sistema de procesamiento se encuentran dos partes, una unidad de procesamiento de aplicación la cual está formada por cuatro procesadores ARM Cortex-A53, y una unidad de procesamiento tiempo real formada por dos procesadores ARM Cortex-R5 [25]. El sistema implementado en la Zynq UltraScale consta de lo siguiente (Figura 3.4):

- Una aplicación Linux ejecutándose en el procesador Cortex-A53, en el que se ejecuta la interfaz de control a nivel usuario del emulador de canal, permitiendo así configurar las características del canal a emular y los parámetros para configurar la electrónica de los frontend. Dicha interfaz se encarga de adquirir todos los datos que introduce el usuario, realizar operaciones intermedias y enviar los mensajes de configuración al procesador de tiempo real
- Dentro del procesador Cortex-R5 se encuentra ejecutándose la aplicación de control de bajo nivel del emulador de canal. Esta aplicación recibe los mensajes

de configuración provenientes de la interfaz del Cortex-A53 y genera la

Envío de datos Cortex R5

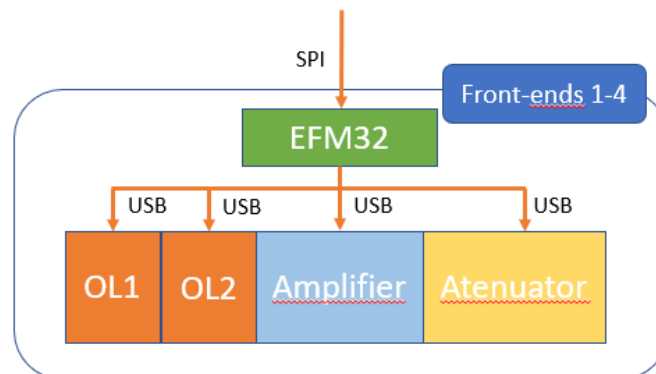


Figura 3.5. Arquitectura elementos programables del frontend.

configuración a nivel de registros necesaria para la operación del emulador de canal. La comunicación entre los dos procesadores se realiza mediante el protocolo de comunicación entre cores RMsg. Dicho protocolo de comunicación se describe en el capítulo 6 de la interfaz de control del emulador.

- Por último, en la parte lógica programable se encuentra la implementación del emulador de canal, que es el elemento que ejecuta la emulación de los modelos de canal. La lógica programable y el Cortex-R5 se comunican a través del bus AXI y registros.

Además, otra función de la tarjeta Zynq UltraScale+ será la programación de cada uno de los elementos programables de los frontend. Para enviar la información a los frontend cada uno de ellos dispone del microcontrolador EFM32 [26], el cual se comunica con la FPGA a través de una conexión mediante SPI. El microcontrolador EFM32 [26] tiene la función de programar los componentes del frontend, que se realiza por una conexión USB. En primer lugar, el microcontrolador procede a iniciar todas las tareas correspondientes con la inicialización de los dispositivos a programar. Es decir, se realiza la detección de los elementos USB a programar, y los inicializa. Una vez el proceso de inicialización ha finalizado, la EFM32 está lista para recibir los parámetros correspondientes a la configuración del frontend.

3.2.2. Frontend RF

Respecto al diseño de los 4 frontend de RF, además de la circuitería de RF, que se describe de forma más detallada en los siguientes capítulos, cada frontend tiene 4 elementos programables (Figura 3.5).

Estos elementos se controlan mediante comunicación USB a partir de la información comandada a través de la EFM32. A continuación, se describe la función que debe realizar cada uno de los elementos programables que componen los frontend de RF:

- Dos osciladores locales (OL) (EV-ADF4355) [27] para realizar el paso a frecuencia intermedia. En el diseño del antiguo emulador solo se disponía de un oscilador para realizar las etapas de down-conversion/up-conversion generando así problemas con el filtrado de señales adyacentes y espurios.
- Atenuador variable (Rudat 6000-110) [28] con el cual realizar la emulación de un perfil de atenuación.
- Amplificador de IF (AD5202) [29] con el cual se optimiza el nivel de potencia de entrada a los ADC de la Zynq UltraScale.

Capítulo 4. Herramientas de desarrollo

Para el desarrollo de este proyecto, han sido necesarias las siguientes herramientas de desarrollo.

4.1. Matlab R2018a

Matlab es un sistema de cómputo numérico que ofrece un entorno de desarrollo integrado con un lenguaje de programación propio. Entre sus prestaciones básicas se hallan la representación de datos y funciones, la implementación de algoritmos, la comunicación con programas en otros lenguajes y con otros dispositivos hardware [30]. Dicha herramienta se utilizará para controlar tanto el VNA (E5080A ENA Vector Network Analyzer) como el analizador de espectro E4404B ESA-E Spectrum Analyzer y obtener los datos de cada una de las etapas que componen los frontend de RF.



MATLAB R2018a

Figura 4.1. Logotipo Matlab R2018a.

4.2. Oracle VM VirtualBox

Oracle VM VirtualBox es un software de virtualización para arquitecturas x86/amd64 [31]. Por medio de esta aplicación es posible instalar sistemas operativos adicionales, dentro de otro sistema operativo, cada uno con su propio ambiente virtual, además de ejecutar máquinas virtuales de forma remota.

Dentro de Oracle VM VirtualBox se instalará una máquina virtual en la cual se realizará la instalación de la multiplataforma QT Creator para el desarrollo de programas software que utilizan una interfaz de usuario.

A continuación, se describirá una guía de instalación de la máquina virtual, así como la instalación de QT Creator y la compilación de las librerías de QT para una arquitectura ARM.



Figura 4.2. Logotipo Oracle VM VirtualBox y QT Creator.

4.2.1. Guía instalación máquina virtual y cross-compiler ARM

En primer lugar, es necesario instalar una máquina virtual Linux sobre el sistema operativo de Windows. Dentro de la máquina virtual se instala QT creator para PC con la que se diseña e implementa la interfaz para posteriormente compilar la implementación de la interfaz a la arquitectura ARM [31]. Como uno de los objetivos del presente Trabajo Fin de Máster es el diseño de una interfaz Linux ejecutada dentro de uno de los procesadores ARM de la tarjeta Zynq UltraScale, es necesario compilar las librerías de QT Creator para la arquitectura ARM dentro de la máquina virtual. Es decir, es necesario crear un entorno de desarrollo que permita cross-compilar desde windows-x64 para ARM. La cross-compilación es el proceso de compilación y creación de código ejecutable para una plataforma distinta de aquella en la que se ejecuta el compilador [32].



Figura 4.3. Cross-compilation para ARM [32].

A continuación, se muestra una guía para compilar las librerías de QT Creator para la arquitectura ARM [33].

4.2.1.1. Requisitos previos

En primer lugar, se debe tener en cuenta que el compilador ARM tiene que formar parte del Kit de desarrollo de Software (SDK) de Xilinx. En este caso, la Zynq UltraScale+ tiene instalada una imagen Linux que se ha creado con Petalinux 2017.4 [34]. Petalinux es un kit de desarrollo de sistemas embebidos dirigido a las FPGAs de Xilinx basado en diseños System on Chip (SoC). Según la guía de Petalinux, el compilador para la Zynq UltraScale+ es el compilador aarch64. Por lo tanto, dentro

de la máquina virtual hay que instalar el mismo compilador para que las versiones sean compatibles.

La máquina virtual incluye un cross-compilador ARM en concreto un compilador de 64 bit aarch64-linux-gnu. Para instalar el compilador basta con ejecutar la siguiente línea

```
bash> source/opt/pkg/petalinux/settings.sh
```

En segundo lugar, se exporta el cross-compilador y se indica el path donde se encuentra localizado

```
bash> export CROSS_COMPILE=aarch64-linux-gnu
```

```
bash> export PATH=/opt/pkg/petalinux/tools/linux-i386/aarch64-linux-gnu/bin
```

En tercer lugar, se crean dos directorios de trabajo uno para el área de instalación y otro para el de compilación como se muestra a continuación

```
bash> export ZYNQ_QT_BUILD=/home/[user]/qt/build/
```

```
bash> export ZYNQ_QT_INSTALL=/home/[user]/qt/install/
```

```
bash> export PATH=$ZYNQ_QT_INSTALL/bin
```

donde [user] es el usuario de la máquina virtual.

4.2.1.2. Cross-compilation Qt5

A continuación, se descargan las fuentes de Qt y se extraen dentro del área de compilación

```
bash> cd $ZYNQ_QT_BUILD
```

```
bash> tar xzfv qt-everywhere-opensource-src-5.4.2.gz
```

```
bash> cd qt-everywhere-opensource-src-5.4.2
```

Antes de configurar el sistema de destino se necesita crear un mkspec el cual indique a qmake qué herramientas debe hacer referencia cuando cree los Makefiles. El directorio mkspec contiene dos ficheros los cuales se descargan de [33]:

- **qmake.conf** – contiene una lista de variables que le indican a qmake qué compilador usar
- **qplatformdefs.h** – archivo de encabezado el cual incluye definiciones específicas de la plataforma

```
bash> mkdir -p qtbase/mkspecs/aarch64-linux-gnu-g++
```

```
bash> mv /path/to/downloaded/qmake_5.4.2.conf qtbase/mkspecs/aarch64-linux-gnu/qmake.conf
```

```
bash> mv /path/to/downloaded/qplatformdefs_5.4.2.h qtbase/mkspace/ aarch64-linux-gnu/qplatformdefs.h
```

A continuación, se procede a configurar Qt con los paquetes requeridos incluyendo los paquetes EGLFS, LinuxFB que están diseñados para sistemas embebidos. Para la ver la extensión de todos los paquetes, ver la referencia [33].

Por último, se ejecuta make para comenzar la cross-compilación de Qt.

```
bash> make
```

Una vez finaliza la compilación se ejecuta make install para comenzar la instalación.

```
bash> make install
```

Cuando acaba la instalación de las librerías de Qt se dispone de un entorno de programación para la arquitectura ARM. Cuando se cree un proyecto solo es necesario añadir los kits de compilación instalados y copiar las librerías dentro de la tarjeta SD donde se ejecutará un Linux una vez se enciende la Zynq UltraScale.

Capítulo 5. Interfaz emulador de canal

En el presente capítulo se presenta el desarrollo de las dos versiones de interfaz del emulador de canal que se han realizado. Estas interfaces han sido creadas utilizando como elemento de desarrollo QT Creator, que es una herramienta de desarrollo multiplataforma orientada a crear interfaces de usuario. Las tareas de la interfaz son: i) efectuar el control del emulador de una manera sencilla con la que se puedan extraer los parámetros necesarios para la configuración del emulador, ii) realizar una serie de cálculos para transformar la información de alto nivel en información interpretable por el programa ejecutado en el Cortex R5, y iii) enviar los datos de una forma estructurada al Cortex R5 de la FPGA para que el programa ejecutado en el Cortex R5 aplique la configuración solicitada desde la interfaz.

Como al emulador de canal se pueden conectar 4 nodos inalámbricos se ha realizado el diseño de la interfaz según este requisito. En la primera versión de interfaz se pueden conectar los nodos A, B, C, y D a los frontend RF (A, B, C, y D). En la segunda versión de interfaz se decidió cambiar dicha nomenclatura, se pueden conectar los nodos 1, 2, 3, y 4 a los frontend 1, 2, 3, y 4. Resultando equivalente el canal A|B al canal 1|2, canal A|C al canal 1|3, canal A|D al canal 1|4, canal B|C al canal 2|3, canal B|D al canal 2|4 y canal C|D al canal 3|4.

5.1. Primera versión interfaz emulador de canal

Esta primera versión de interfaz de control es una reconstrucción heredada de un diseño de interfaz del emulador de canal inalámbrico basado en la FPGA Kintex-7. Originariamente, esta interfaz no se ejecutaba directamente sobre el Cortex A53, sino que se ejecutaba en un ordenador con sistema operativo Windows y los comandos se enviaban a la FPGA Kintex-7 mediante un puerto Ethernet [35]. Por ello, la primera tarea en el desarrollo de las interfaces fue rehacer la interfaz del emulador de canal inalámbrico antiguo para el emulador nuevo, y que se ejecutara sobre el microcontrolador Cortex A53 incluido en la Zynq UltraScale+.

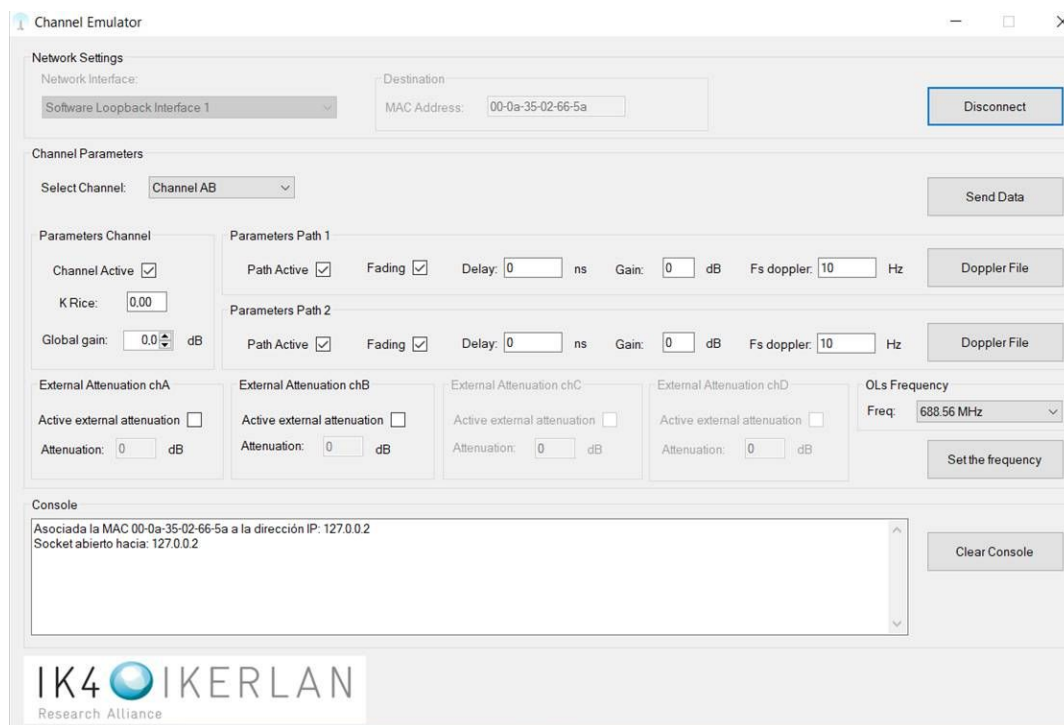


Figura 5.1. Interfaz de control emulador de canal [36].

Esta primera versión de interfaz se ha adaptado a las necesidades del emulador nuevo una de ellas es el cambio de la FPGA Kintex-7 a la plataforma ZCU102 – Zynq UltraScale+ MPSoC. La diferencia principal entre ellas es que no es necesario introducir la información de configuración de la comunicación Ethernet, ya que la interfaz de usuario se ejecuta directamente sobre el ARM Cortex A53.

Para proceder a la configuración del emulador de canal, es necesario tener en cuenta que se debe configurar, por una parte, los elementos programables del frontend RF y por otro lado las propiedades del canal que se quieran emular. En primer lugar, para realizar la configuración del canal es necesario definir sus características, es decir, como va a ser el comportamiento de ese canal. Por lo tanto, se especificará el canal lógico, si presenta fading y como es su distribución estadística (Rayleigh o Rice); y el número de paths por canal para ver el comportamiento de la señal cuando se le somete el efecto de multicamino. Por otro lado, hay que indicar las características de la señal en cada tap, es decir, cuál va a ser el peso o ganancia de cada tap, el delay correspondiente a cada tap, y por último la frecuencia Doppler como el modelo de espectro Doppler que se a aplicar a la señal.

En segundo lugar, para realizar la configuración de los frontends, es necesario indicar a que valores se configurarán los elementos programables, es decir, el amplificador de IF, los dos OLs y el valor de atenuación del atenuador que se encuentra en la etapa de subida. Por lo tanto, se indicará la frecuencia de la

portadora para posteriormente ajustar los osciladores que se encargan de hacer la conversión a frecuencias intermedias, el valor de atenuación para el caso del atenuador y la ganancia a la que se programará el amplificador de IF.

5.1.1. Interfaz usuario

En la Figura 5.2 se muestra la interfaz de control del emulador. Esta interfaz consta de cuatro partes diferenciadas. En primer lugar, y marcado en color azul, hay un desplegable de tipo combo utilizado para seleccionar cuales son los frontend que se quieren comunicar. En segundo lugar, e indicado en verde, se tienen los parámetros de configuración del modelo de fading. En tercer lugar, e indicado en rojo, los parámetros correspondientes para cada tap incluyendo la distribución Doppler. Por último, en naranja, los parámetros de configuración de los elementos programables del frontend.

Como se describió en el capítulo anterior al emulador de canal se pueden conectar 4 nodos inalámbricos resultando que se pueda utilizar un sistema MIMO 4x4 como modo de acceso al canal. Por lo tanto, en la interfaz se implementarán 6 canales lógicos que son (Figura 3.3): Canal AB/BA, Canal AC/CA, Canal AD/DA, Canal BC/CB, Canal BD/DB y Canal CD/DC.

En segundo lugar, se puede elegir si se quiere activar o desactivar el canal, además de la distribución de la envolvente (Figura 5.3). En el caso de que se aplique una $K = 0$ el canal tendrá un comportamiento Rayleigh y para $K > 0$ se tendrá un canal Rice. Además, se puede controlar cual es la atenuación que se aplica en el procesador banda base (atenuación digital).

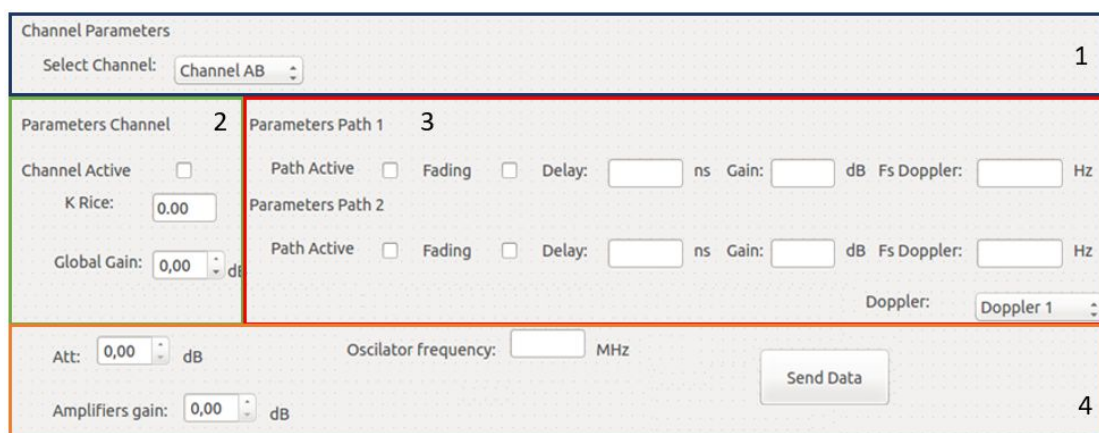


Figura 5.2. Primera versión interfaz usuario.

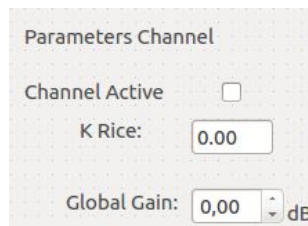


Figura 5.3. Parámetros de canal.

En tercer lugar, se encuentra los parámetros referentes a cada path, dando la posibilidad de controlar cada path de forma independiente. Primero se podrá activar/desactivar el path al igual que con el desvanecimiento. En el caso que solo se active un path, el canal no presentará dispersión temporal. En el caso en el que se activen más taps, el canal tendrá dispersión temporal. El primer tap se puede configurar como una distribución Rice, en caso de que exista línea de visión directa, o Rayleigh ($K = 0$), en caso de que no exista línea de visión directa. Por otro lado, los taps restantes están configurados para seguir un fading de tipo Rayleigh, ya que son taps provocados por reflexiones. La configuración de los taps incluye la frecuencia Doppler ($F_s Doppler$) en Hz, el retardo ($Delay$) del tap en ns y su ganancia ($Gain$) en dB. Se debe mencionar que el retardo del tap en ns debe ser múltiplo del periodo de muestreo de la señal digital banda base, en este caso 8.14 ns.



Figura 5.4. Parámetros path.

En cuarto lugar, la configuración los dispositivos que componen los frontend. Por un lado, la frecuencia central de operación en MHz a partir de la cual se fija la frecuencia de oscilación de los OLs, la ganancia que se debe aplicar a los amplificadores, la atenuación del atenuador variable para generar el efecto del pathloss.



Figura 5.5. Parámetros de configuración frontend.

Una vez configurada toda la información, dicha información es enviada desde el A53 al R5 cuando el usuario presiona sobre el botón “*Send Data*”. En la siguiente sección se mostrará como se recoge la información y posteriormente como se procederá a su envío.

5.1.2. Interfaz A53-R5

Como se ha indicado anteriormente una vez se pulsa sobre el botón “*Send Data*” de la interfaz, el procesador A53 envía la estructura de datos al procesador tiempo real R5 sobre el cual se está ejecutando la aplicación de emulador de canal en bare-metal y donde se genera la estructura de configuración de bajo nivel que es utilizada para generar la emulación de canal.

La estructura de datos que enviará el A53 al R5 es la siguiente:

```
struct datasend {  
    int type_canal;  
    int canal_active;  
    int path_active[2];  
    int fading_active[2];  
    int delay [2];  
    float ganancia [2];  
    int doppler_type;  
    float att_atenuator;  
    float gain_amplifier;  
    float gain_total;  
    int fdoppler[2];  
    float freq_oscilator;  
    int version_interfaz;  
};
```

Donde cada campo de la estructura `datasend` se explica de forma detallada a continuación:

`int type_canal`: indica que canal se ha seleccionado

```
type_canal = 0 -> Canal AB  
type_canal = 1 -> Canal AC  
type_canal = 2 -> Canal AD  
type_canal = 3 -> Canal BC  
type_canal = 4 -> Canal BD  
type_canal = 5 -> Canal CD
```

`int canal_active`: si su valor es ‘0’ indicará que el canal está desactivado, en cambio sí es ‘1’ el canal está activado.

`int path_active[2]`: indica si el tap está activo o no.

`int fading_active[2]`: indica si los desvanecimientos están activados o desactivados.

`int delay [2]`: indica el delay de cada tap, en ns.

`float ganancia [2]`: indica la ganancia de cada tap, en dB.

`int doppler_type`: indica el tipo de filtro Doppler.

float att_atenuator: valor de atenuación al que se debe configurar el atenuador analógico. Presenta valores entre 0 y 110 dB con un paso de 0.25 dB.

float gain_amplifier: indica la ganancia de los amplificadores de IF.

float gain_total: ganancia global del sistema, en dB.

int fdoppler[2]: frecuencia Doppler de cada tap.

float freq_oscilator: frecuencia de los osciladores locales.

int version_interfaz: indica la versión de la interfaz.

La comunicación entre ambos procesadores se realiza mediante un canal RPMsg (*Remote Processor Messaging*). RPMsg es una API implementada sobre el framework VirtIO para que un procesador local se comunice con los procesadores remotos disponibles en el sistema. VirtIO proporciona una capa de transporte basada en anillos, los cuales se utilizan para enviar/recibir mensajes para/desde un procesador remoto sobre una memoria compartida. Los anillos se caracterizan por ser unidireccionales, es decir, un anillo se dedica a enviar mensajes desde un procesador y el resto de los anillos se usan para recibir los mensajes desde el procesador remoto [36].

Por lo tanto, la funcionalidad completa del botón “*Send Data*” es: i) reservar espacio en memoria según el tamaño de la estructura de datos, ii) abrir el canal de comunicaciones RPMsg y iii) escribir los datos de configuración para que el procesador remoto los pueda procesar.

5.2. Segunda versión interfaz emulador de canal

La segunda versión de interfaz del emulador se ha diseñado de forma que se puedan utilizar todas las funcionalidades del nuevo emulador. Dichas funcionalidades son:

- Implementar un número mayor de taps por canal, permitiendo emular un canal inalámbrico de forma más realista.
- Incluir perfiles de shadowing para simular el efecto de las variaciones lentas del canal provocado por elementos de gran tamaño que ocasionan zonas de sombra.
- Nuevas distribuciones de modelado de canal. Para modelar los desvanecimientos causados por el fading se puede elegir entre los tres tipos de distribución de envolvente Rayleigh, Rice y Nakagami. Y, por otro lado, para modelar el desvanecimiento causado por la variación temporal del canal producido por el movimiento se incluyen las distribuciones de espectro Doppler Jakes y Bell.

- Incluir propiedades de la señal transmitida como ancho de banda y relación entre la potencia media de la señal y potencia pico.
- Además de incluir nuevas opciones de configuración, la interfaz también se debe encargar de enviar la configuración de bajo nivel al microcontrolador R5. Para ello, la interfaz se encarga de realizar una serie de cálculos intermedios que transforman la configuración de alto nivel en estructuras de configuración de bajo nivel.

Igual que en el caso de la primera versión de interfaz, a la hora de configurar el emulador de canal, es necesario configurar 2 tipos de elementos: los frontend y las propiedades del canal. Para realizar la configuración de los frontend es necesario indicar, por parte del usuario del emulador, la siguiente información sobre la señal:

- Frecuencia portadora
- Ancho de banda de la señal
- Relación entre la potencia media y pico de la señal (PAPR- Power average Power ratio), también denominado comúnmente factor de cresta.

Con esta información se configuran los OLs, a la frecuencia de mezcla adecuada y se fijan los valores de ganancia/atenuación de los amplificadores de IF.

Para realizar la configuración del canal, el usuario debe definir primero sus características:

- Los canales lógicos que se desean configurar (Figura 3.3), que son:
 - Canal 1-2/ Canal 2-1
 - Canal 1-3/ Canal 3-1
 - Canal 1-4/ Canal 4-1
 - Canal 2-3/ Canal 3-2
 - Canal 2-4/ Canal 4-2
 - Canal 3-4/ Canal 4-3
- Indicar como es el perfil de desvanecimiento mediante los diferentes tipos de envolvente (Rayleigh, Rice o Nakagami).
- Número de paths que se quieren emular por canal y las propiedades que presenta cada path. Como su ganancia, delay, frecuencia doppler, tipo de filtro doppler, y coeficientes del filtro.

Por otro lado, el emulador también debe poder emular los efectos de pathloss, shadowing y ruido gaussiano.

5.2.1. Interfaz de usuario

Dicha interfaz se encarga de recoger los datos que el usuario introduce en la interfaz gráfica. Como se muestra en la Figura 5.6 la interfaz gráfica consta de 3 partes diferenciadas:



Figura 5.6. Estructura interfaz de usuario.

- La primera zona, marcada en azul, representa mediante botones las entradas, salidas y canales.
- La segunda zona, marcada en verde, se corresponde con la configuración de los parámetros de los frontends / canal. Esta zona de la interfaz es variable y lo que aparece en ella depende de que botón del primer elemento de la interfaz haya sido pulsado previamente.
- La tercera zona, marcada en naranja, se utiliza para el control de la ejecución del emulador. En esta zona se han añadido los botones para el inicio, pausa y parada de la emulación; y también el botón “*Send Data*” que ejecuta la comunicación entre los cores A53 y R5.

Para realizar la configuración de los parámetros de las entradas, cuando se pulse sobre el botón correspondiente se mostrará la pestaña “*Inputs*”. En dicha pestaña se realiza la configuración de los frontend, que tiene las siguientes opciones:

- Activar/desactivar el frontend.
- Introducir los parámetros de configuración del frontend: como la potencia media emitida de la señal RF y el factor de cresta. La frecuencia central de la portadora para posteriormente calcular la frecuencia de oscilación de los OLs y, por último, las pérdidas que introducen los cables de RF utilizados para interconectar las etapas del frontend.
- Por último, se tiene un botón “*Apply*” donde se guardan los valores introducidos en una estructura correspondiente a cada entrada antes de proceder a configurar el resto.

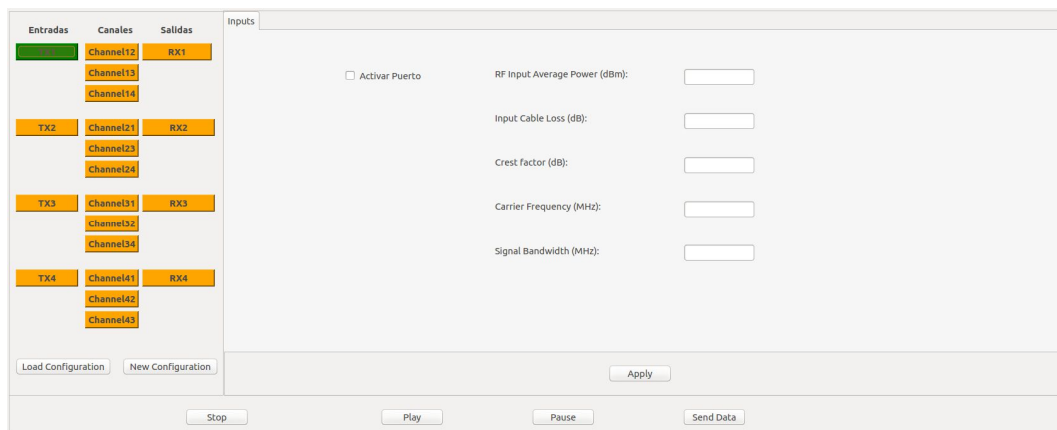


Figura 5.7. Pestaña de entradas interfaz.

Cuando se pulsa sobre los botones correspondientes al canal se puede configurar el “*Small Scale Fading*”, “*Large Scale Fading*” y la “*SNR*”.

La pestaña “*Small Scale Fading*” dispone de las siguientes funcionalidades:

- Activar tanto el path como el fading.
- Introducir valores de delay, máxima frecuencia doppler, tipo de filtro doppler (Jakes, Bell) y distribución de la envolvente para el fading (Rayleigh, Rice o Nakagami).

Dependiendo de la distribución de envolvente que se escoja, se habilitará una ventana para la configuración de cada una de ellas.

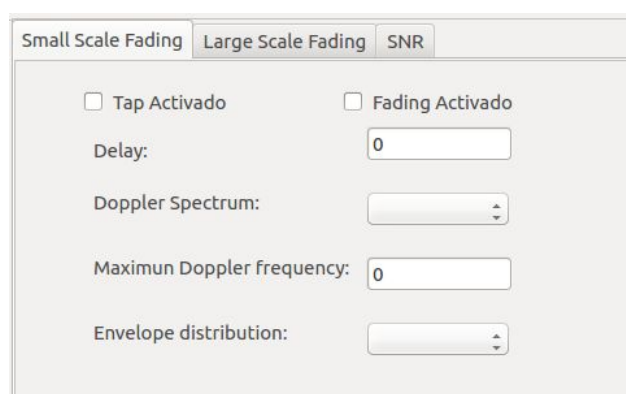


Figura 5.8. Pestaña Small Scale Fading.

The image shows two side-by-side configuration panels. The left panel is titled 'Rice Distribution' and contains a label 'K Rice:' followed by an empty text input field. The right panel is titled 'Nakagami Distribution' and contains two labels: 'm' and 'Ω', each followed by an empty text input field.

Figura 5.9. Parámetros distribución de envolvente Rice y Nakagami.

Una vez se introducen los valores para el tap, pulsando sobre el botón “AddTap” se guardan en la tabla antes de pasar a la configuración del siguiente tap. Además, si el usuario ha cometido un error al introducir los parámetros, se podrán modificar a través de los campos de la tabla haciendo doble click. Por último, una vez se han configurado todos los taps, se representará su respuesta al impulso en la gráfica superior a la tabla de configuración de taps.

The image shows a complex software interface for configuring taps. On the left, there is a table with columns 'Entradas', 'Canales', and 'Salidas'. It lists four transmitter-receiver pairs (TX1-RX1 to TX4-RX4) with associated channel numbers (Channel13 to Channel14, Channel21 to Channel24, Channel31 to Channel34, and Channel41 to Channel43). Below this table are buttons for 'Load Configuration', 'New Configuration', and 'Add Tap'. The main area is divided into three tabs: 'Small Scale Fading', 'Large Scale Fading', and 'SNR'. The 'Small Scale Fading' tab is active, showing checkboxes for 'Tap Activado' and 'Fading Activado', and input fields for 'Delay (ns)', 'Ganancia (dB)', 'Doppler Spectrum', 'Maximun Doppler frequency (Hz)', and 'Envelope distribution'. The 'Large Scale Fading' tab is also visible, showing a 'Rice Distribution' section with a 'K Rice' input field. On the right, there is a graph with a grid. Below the graph is a table with the following data:

Delay	Gain	Doppler Spectrum	Envelope Distribution	Parameters Distributio
1	0	Jakes	Rice	0.5

At the bottom of the interface are buttons for 'Stop', 'Play', 'Pause', and 'Send Data'.

Figura 5.10. Configuración de taps.

Si el usuario pulsa sobre la pestaña “Large Scale Fading” dispone de las siguientes funcionalidades:

- Configuración tipo PathLoss para simular la atenuación de la señal entre la distancia que separa al transmisor del receptor.
- Configuración de los perfiles de Shadowing para emular la atenuación provocada en la señal debida a elementos de gran tamaño que generan zonas de sombra.

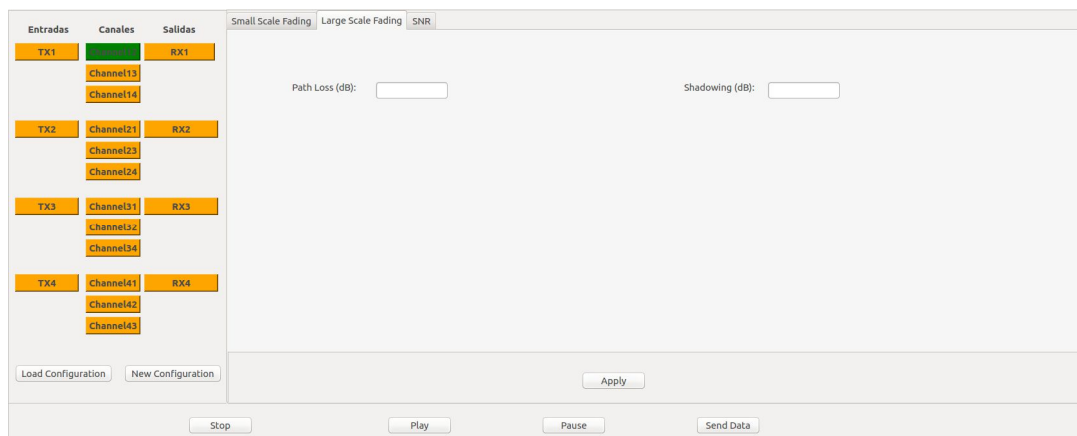


Figura 5.11. Pestaña Large Scale Fading.

Por otro lado, la pestaña “SNR” dispone de las siguientes funcionalidades:

- Introducir ancho de banda y valor de la señal/ruido, con dichos parámetros se especifica la potencia de ruido del generador de ruido Guassiano implementado en la FPGA.

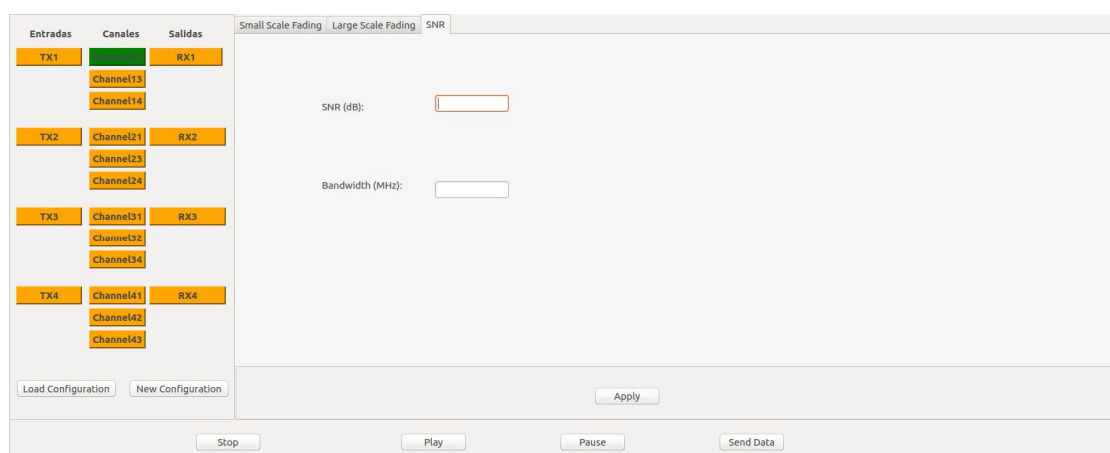


Figura 5.12. Pestaña generador de ruido Gaussiano.

Por último, cuando se presiona sobre un botón correspondiente a la sección de salidas se habilita la pestaña “Outputs”. En dicha pestaña, se introducen las pérdidas de cables, frecuencia de la portadora y ancho de banda de la señal (estas 2 últimas opciones estarán desactivadas/ocultas por defecto, pero se han incluido pensando en futuras versiones del emulador). Cuando el usuario presione sobre el botón “Apply” se guardarán dichos valores en la estructura correspondiente a cada salida.

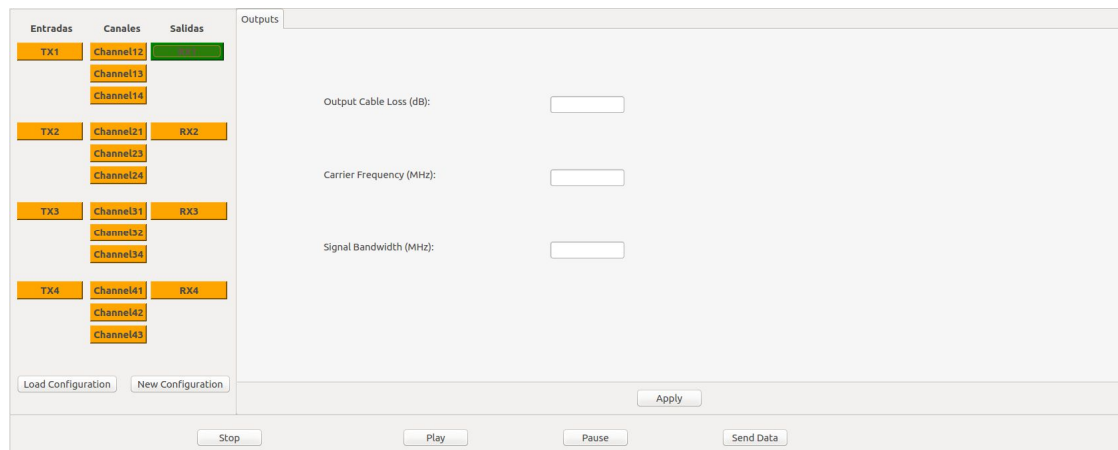


Figura 5.13. Pestaña salidas interfaz.

5.2.2. Interfaz A53-R5

Los requisitos que debe cumplir la interfaz A53-R5 son los siguientes:

- Guardar de forma estructurada los parámetros que introduce el usuario a través de la interfaz. Para ello, se definirá una estructura para cada sección de datos que se quieren guardar, como son las entradas, salidas y canales.
- Acceder a la información de las estructuras y realizar una serie de cálculos intermedios y conversiones que posteriormente se enviarán al R5.
- Enviar tanto la configuración del frontend como del canal al R5 a través del canal de comunicaciones RMPsg.

Como se ha mencionado anteriormente, se definen las estructuras donde se guardarán todos los datos. Dicha definición se muestra a continuación:

- Estructura configuración entradas:

En primer lugar, se define la estructura para la configuración de las entradas, en la cual se incluye un campo por cada parámetro que ha añadido el usuario como las pérdidas de los cables de RF, frecuencia central, ancho de banda y factor de cresta de la señal y potencia media de entrada.

```
struct paraminput{
    int input_active;
    float input_cable_loss;
    float input_avg_power;
    float input_cress_factor;
    float input_carrier_freq;
    float input_signal_bandwidth;
};
```

Una vez el usuario pulse sobre el botón “Apply” se guardarán los parámetros dentro de la estructura correspondiente a cada entrada: **struct**

`paraminputconfig_input[ELEMENTOS_INPUT]` donde `ELEMENTOS_INPUT` es el número total de entradas en este caso 4.

A partir de estos valores se realizarán los cálculos correspondientes para las frecuencias de los osciladores y la ganancia/atenuación de los amplificadores de IF.

Por un lado, la ganancia de los amplificadores de IF se puede calcular según la siguiente expresión:

$$if_{gain} = node_{avgtxpower} - input_{cable_{loss}} - frontend_{downconversionloss} - output_{powerfrontend} - cress\ factor, \quad (5.1)$$

donde los valores de “`outputpowerfrontend`” y de “`frontenddownconversionloss`” son valores conocidos que no necesita introducir el usuario. El valor de “`outputpowerfrontend`” viene fijado por la potencia máxima que se puede introducir en las tarjetas FMC150 sin saturar el ADC, que es de -3 dBm.

Por otro lado, a partir de la frecuencia de la portadora se obtienen las frecuencias a las que se configurarán los OLs. Para ello hay que tener en cuenta que a frecuencias de portadora mayores de 1400 MHz se utilizarán las etapas 1 y 2 de down-conversion y up-conversion, en cambio, si está entre 400 MHz y 1400 MHz estarán activas la 2ª etapa de down-conversion y la 1ª etapa de up-conversion.

```
if (400 <input_carrier_freq< 1400) {  
    //Etapas 2 de down-conversion y etapa 1 de up-conversion  
    //input_carrier_freq -> 61.44 MHz/input_carrier_freq -> 645MHz  
    f_OL2 = input_carrier_freq - 61.44 MHz;  
} else if (input_carrier_freq > 1400) {  
    //Etapas 1 y 2 de down-converion y up-conversion  
    //input_carrier_freq -> 645 MHz -> 61.44 MHz  
    f_OL1 = input_carrier_freq - 645 MHz;  
    f_OL2 = 645 MHz - 61.44 MHz;  
};
```

- Estructura configuración salidas:

En segundo lugar, se define la estructura para la configuración de las salidas, en la cual se incluye un campo para las pérdidas de los cables de RF, frecuencia central y ancho de banda de la señal.

```
struct paramoutput{  
    float output_cable_loss;  
    float output_carrier_freq;
```

```
float output_signal_bandwidth;  
};
```

Una vez el usuario pulse sobre el botón “Apply” se guardarán los parámetros dentro de la estructura correspondiente a cada salida: `struct paramoutputconfig_output[ELEMENTOS_OUTPUT]` donde `ELEMENTOS_OUTPUT` es el número total de salidas en este caso 4.

- Estructura configuración *Large Scale Fading*.

En tercer lugar, se define la estructura para configurar los efectos que se producen en la señal debido a los desvanecimientos por pathloss y shadowing.

```
struct large_scale_fading {  
float path_loss;  
float shadowing;  
}
```

A partir de estos valores introducidos en la interfaz se pueden deducir los valores de atenuación y factor de corrección que se debe aplicar a cada canal para compensar la atenuación que no se aplica con el valor de atenuación del frontend.

En un caso ideal la potencia de un nodo transmisor viene determinada por las pérdidas de pathloss junto con el shadowing y la potencia del nodo receptor, como se muestra en la siguiente expresión.

$$node_{avgtxpower} = factor_{pathloss+shadowing} + node_{avgrxpower} \quad (5.2)$$

A partir de los parámetros introducidos en la interfaz se puede deducir la expresión de la potencia del nodo transmisor:

$$node_{avgtxpower} = node_{avgrxpower} + input_{cable_{loss}} + frontend_{downconversion_{loss}} - if_{gain} + ADC/DAC_{correction} + frontend_{upconversion_{loss}} + att_{salida} + output_{cable_{loss}} \quad (5.3)$$

donde las pérdidas de down-conversion y up-conversion son un valor conocido y extraído a través de la caracterización de los frontend, y el valor $ADC/DAC_{correction}$ extraído de la datasheet. Una vez conocidos todos los parámetros se despeja de la expresión $node_{avgtxpower}$ la atenuación de salida para un canal que se enviará posteriormente al R5.

$$\begin{aligned}
 att_{salida} = & node_{avgtxpower} - node_{avgrxpower} - input_{cable_{loss}} \\
 & - frontend_{downconversionloss} + if_{gain} - \frac{ADC}{DAC}_{correction} \\
 & - frontend_{upconversionloss} - output_{cable_{loss}}.
 \end{aligned} \tag{5.4}$$

Sin embargo, dado que el atenuador de salida es compartido por 3 canales, se requiere que cada uno tenga un valor de att_{salida} diferente. Por ello, hay que buscar el valor común que sea aplicable a los 3 canales, y después realizar para cada canal de manera independiente una corrección en digital.

A continuación, se muestra un ejemplo de configuración. Si se introduce señal por la entrada 1, habría que obtener la atenuación de salida para cada uno de los correspondientes canales para canales 1-2, 1-3 y 1-4.

$$\begin{aligned}
 att_{salidachannel12} = & node_{avgtxpower} - node_{avgrxpower} - input_{cable_{loss}} - \\
 & frontend_{downconversionloss} + if_{gainchannel\ 12} - ADC/DAC_{correction} - \\
 & frontend_{upconversionloss} - output_{cable_{loss}}.
 \end{aligned} \tag{5.5}$$

$$\begin{aligned}
 att_{salidachannel13} = & node_{avgtxpower} - node_{avgrxpower} - input_{cable_{loss}} - \\
 & frontend_{downconversionloss} + if_{gainchannel\ 13} - AD \frac{C}{DAC}_{correction} - \\
 & frontend_{upconversionloss} - output_{cable_{loss}}.
 \end{aligned} \tag{5.6}$$

$$\begin{aligned}
 att_{salidachannel14} = & node_{avgtxpower} - node_{avgrxpower} - input_{cable_{loss}} - \\
 & frontend_{downconversionloss} + if_{gainchannel\ 14} - ADC/DAC_{correction} - \\
 & frontend_{upconversionloss} - output_{cable_{loss}}.
 \end{aligned} \tag{5.7}$$

Y el mínimo valor de atenuación es el que hay que aplicar en el frontend

$$att_{salida} = \min(att_{salidachannel12}, att_{salidachannel13}, att_{salidachannel14}). \tag{5.8}$$

Por último, se calcula el factor de corrección que se debe aplicar a cada canal de la siguiente forma:

$$att_{correctionchannel12} = att_{salidachannel12} - att_{salida} \tag{5.9}$$

$$att_{correctionchannel13} = att_{salidachannel13} - att_{salida} \tag{5.10}$$

$$att_{correctionchannel14} = att_{salidachannel14} - att_{salida} \tag{5.11}$$

- Estructura configuración canales:

En cuarto lugar, se define la estructura donde se guardarán los datos correspondientes a cada canal, como son el identificador de canal configurado, el número de taps que se han introducido y las características de cada tap (delay, ganancia, tipo de filtro Doppler, distribución de la envolvente de la señal, etc).

```
struct channel {  
    int ID_channel;  
    int num_tap;  
    tap array_tab[N_TAP];  
};
```

Después de introducir los datos de configuración en la interfaz de usuario y se guarden en sus correspondientes estructuras de configuración, el procesador A53 procederá a realizar los cálculos intermedios. El resultado de dichos cálculos son la frecuencia de los osciladores locales, ganancia del amplificador de IF, y valor de atenuación del atenuador. Por último, se guardará toda la información para configurar tanto los frontend como el canal en sus correspondientes estructuras, y una vez se pinche sobre el botón “*Send Data*”, el cortex A53 enviará los datos al otro procesador de la Zynq UltraScale.

El R5 recibirá dos tipos de estructuras, una estructura de tipo configuración frontend y por otro lado una estructura de tipo configuración de canal. Para que el R5 pueda distinguir la información que recibe se incluirá en cada estructura un identificador que indique el tipo de información que se está enviando. Por lo tanto, el R5 deberá leer primero el identificador *ID_str*. Si tiene un valor de 0 la estructura recibida será del tipo configuración del frontend y si en cambio tiene valor de 1 habrá recibido una estructura para la configuración del canal.

Las estructuras que recibirá el R5 para realizar la configuración de los frontend presenta la siguiente estructura, en este caso al R5 le llegarán 4 estructuras de configuración frontend ya que se disponen de un total de 4 frontend:

```
struct configuration_frontend{  
    int ID_str;  
    int ID_frontend;  
    float center_freq;  
    float f_OL1;  
    float f_OL2;  
    float if_gain;  
    float at_salida;  
};
```

Donde cada campo de la estructura se explica a continuación:

int ID_str: indica el tipo de mensaje mediante un identificador, en este caso como se trata de la estructura del tipo configuración frontend tendrá un valor de 0.

int ID_frontend: mediante un identificador se indica el número de frontend que se quiere configurar.

El frontend1 se corresponde con el identificador 0.

El frontend2 se corresponde con el identificador 1.

El frontend3 se corresponde con el identificador 2.

El frontend4 se corresponde con el identificador 3.

float center_freq: se corresponde con la frecuencia de la portadora, en MHz

float f_OL1: frecuencia a la que hay que configurar el primer oscilador, en MHz (calculado previamente).

float f_OL2: frecuencia a la que hay que configurar el segundo oscilador, en MHz (calculado previamente).

float if_gain: ganancia global del sistema, en dB (calculado previamente).

float at_salida: valor de atenuación al que se debe configurar el atenuador. Presenta valores entre 0 y 110 dB con un paso de 0.25 dB (calculado previamente).

Por otro lado, las estructuras que enviará el A53 para la configuración de cada uno de los canales presentan la siguiente forma:

```
struct send_channel {
    int ID_str;
    int ID_channel;
    int tap_active[N_TAP];
    int ID_tap;
    int fading_active[N_TAP];
    int delay[N_TAP];
    float gain[N_TAP];
    float max_fdoppler[N_TAP];
    float rice_c1;
    float rice_c2;
    float gain_total;
    float doppler_coeficient_Num[N_TAP][GRADE];
    float doppler_coeficient_Den[N_TAP][GRADE];
};
```

donde cada campo de la estructura se explica a continuación:

int ID_str: indica el tipo de mensaje mediante un identificador, en este caso como se trata de una estructura para la configuración del canal el identificador tendrá un valor de 1.

int ID_channel: indica que canal se va a configurar.

int tap_active[N_TAP]: indica si el tap está activo o no.

int ID_tap: indica el número del tap, '0' para el tap1 y '1' para el tap2.

int fading_active[N_TAP]: indica si el fading está activado o desactivado.

int delay[N_TAP]: indica el delay de cada tap.

float gain[N_TAP]: indica la ganancia de cada tap incluyendo corrección de pathloss. Esta corrección es necesaria porque diferentes canales comparten una misma salida.

float max_fdoppler[N_TAP]: indica la frecuencia Doppler de cada tap.

float rice_c1: indica el valor de la K de rice conjuntamente conrice_c2.

float rice_c2: indica el valor de la K de rice conjuntamente conrice_c1.

float gain_total: Indica el factor de corrección que se debe aplicar a cada canal, este depende de la atenuación y de las pérdidas por pathloss y shadowing, en dB con paso de 0.1 dB.

float doppler_coeficient_Num[N_TAP][grado]: recoge el numerador del coeficiente del filtro doppler para un polinomio de grado 9.

float doppler_coeficient_Den[N_TAP][grado]: recoge el denominador del coeficiente del filtro doppler para un polinomio de grado 9.

Capítulo 6. Caracterización Frontend RF

En el presente capítulo se describen en primer lugar cada una de las etapas que componen el nuevo diseño de los frontend RF, así como el software utilizado para el control de los elementos programables que lo componen. En segundo lugar, se explican los diferentes setups utilizados para realizar la caracterización de cada una de las etapas y por último se muestran los resultados obtenidos de la caracterización en las bandas de 434 MHz, 868 MHz y 2.45 GHz.

La caracterización del frontend es de gran utilidad para comprobar el correcto funcionamiento del diseño y extraer resultados tanto de las pérdidas como espurios que puedan introducir la circuitería de RF. Por otro lado, también es necesario realizar la caracterización de los frontend para calibrar el emulador y ofrecer una emulación real lo más parecida a la emulación configurada por el usuario. Se ha realizado la caracterización completa de los 4 frontend, pero en este documento solo se muestra la caracterización completa de uno de ellos para cada banda.

6.1. Diseño Frontends RF

El emulador de canal se ha diseñado de forma que cubre las bandas frecuenciales de 400 MHz a 6 GHz y con un ancho de banda de 100 MHz, con lo que cualquier transceptor RF que trabaje en ese rango de frecuencias y con un ancho de banda igual o menor al del emulador puede ser conectado al emulador. Por otro lado, el frontend se ha diseñado para que realice la conversión de frecuencia en dos etapas. La primera conversión RF/IF se realiza a la frecuencia de 645 MHz, para posteriormente reducir la frecuencia de 645 MHz a 61.44 MHz. Además, dependiendo de la banda en la que se quiera transmitir hace falta un salto o dos para realizar la conversión de frecuencia. Por otro lado, el frontend también realiza la conversión inversa, pasando de 61.44 MHz a 645 MHz y finalmente a la banda deseada.

En la **¡Error! No se encuentra el origen de la referencia.** se muestra el diseño de cada uno de los 4 frontend RF los cuales se componen de cinco etapas diferenciadas. A continuación, se explican cada una de las etapas del diseño.

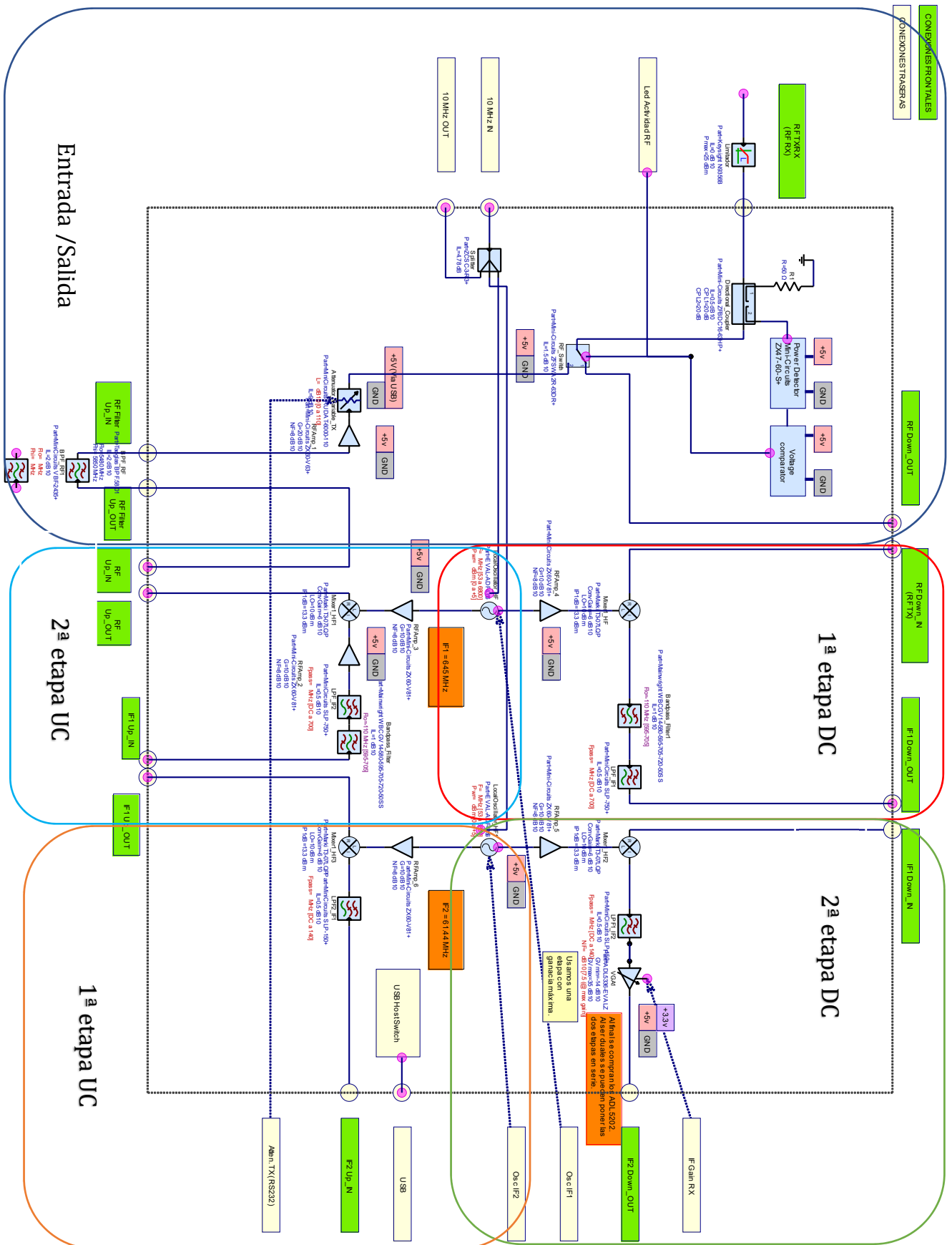


Figura 6.1. Esquema frontend RF completo.

El frontend consta de las siguientes etapas: una etapa de Entrada/Salida resaltado en azul, dos etapas correspondientes a la conversión de la señal de RF a IF resaltado en rojo y verde y por último dos etapas para realizar la conversión de la señal de IF a RF (resaltadas en naranja y azul). A continuación, se explica la función de cada etapa que compone el diseño del frontend:

La primera de ellas, la etapa de Entrada/Salida, está representada en la Figura 6.2. El puerto de entrada/salida del emulador está denotado como '1', el puerto '2' va hacia el ADC, y el puerto '3' viene del DAC. El puerto '1' hace tanto de transmisor como receptor de la señal RF, y es el puerto donde se conecta el transceptor de comunicaciones a medir. La señal RF procedente del puerto '1' atraviesa un acoplador direccional el cual distribuye la señal parte a un detector de potencia y a un Switch RF. La función del detector de potencia es distinguir si existe señal de entrada. En caso de que exista, el comparador de voltaje enviará una señal al Switch RF y el puerto RF se pondrá en modo "recepción" (la señal entra al frontend). La señal RF atravesará la puerta RF1, pasando entonces las etapas de down-conversion hacia el puerto '2'. En cambio, si el detector de potencia no detecta señal se supone que la señal está saliendo por defecto y proviene de las etapas de up-conversion (puerto '3'). Además, en la Figura 6.2, se dispone de una etapa intermedia con la cual a partir de un filtro de RF se puede escoger la banda de la señal que se recibe en el puerto '3'. Por último, entre el puerto '3' y el puerto '1' (salida del emulador a RF), existe un atenuador variable que se usa para implementar el path loss y el shadowing, y cuyo control es realizado por el microcontrolador EFM32.

El umbral de activación de la detección de potencia se ha elegido de forma empírica a través de la caracterización del frontend. Dicha caracterización se especifica en la sección 6.3 de este capítulo. En base a estas medidas, el umbral de detección de potencia se fijó a 0 dBm.

En segundo lugar, a la hora de realizar conversión de frecuencia de RF a IF esta se puede hacer en dos etapas dependiendo de cuál sea la frecuencia de la señal de entrada (Figura 6.3). Si la frecuencia de la portadora es mayor que 1400 MHz se utilizan las etapas 1 y 2 de down-conversion, en cambio, si está entre 400 MHz y 1400 MHz solo se utiliza la 2 etapa de down-conversion. La señal proveniente de RF_{Down_IN} a la frecuencia central f_c se mezcla junto con la frecuencia de oscilación del OL1 $f_{OL1} = f_c - 645 \text{ MHz}$ a través de un mezclador RF. Resultando la frecuencia de mezcla a la salida I del mezclador $IF1 = 645 \text{ MHz}$. Los amplificadores RF son necesarios para adecuar la potencia del tono emitido por el OL a la entrada L del mezclador. Posteriormente, se filtra la señal para seleccionar la banda de 645 MHz a través de un filtro paso banda con un bandwidth de 110 MHz y se eliminan los

productos de intermodulación con un filtro de IF. Una vez la señal sale por el puerto *IF1 Down_OUT* a 645 MHz se procede a realizar la segunda etapa de down-conversion para obtener la señal a la salida *I* del mezclador a $IF2 = 61.44 \text{ MHz}$. El diseño de esta segunda etapa es similar al de la primera salvo que se tiene un amplificador de IF el cual actúa tanto como amplificador y atenuador para adecuar la potencia de entrada al ADC. Finalmente, el ADC convierte la señal del dominio continuo al dominio discreto y se envía a la FPGA.

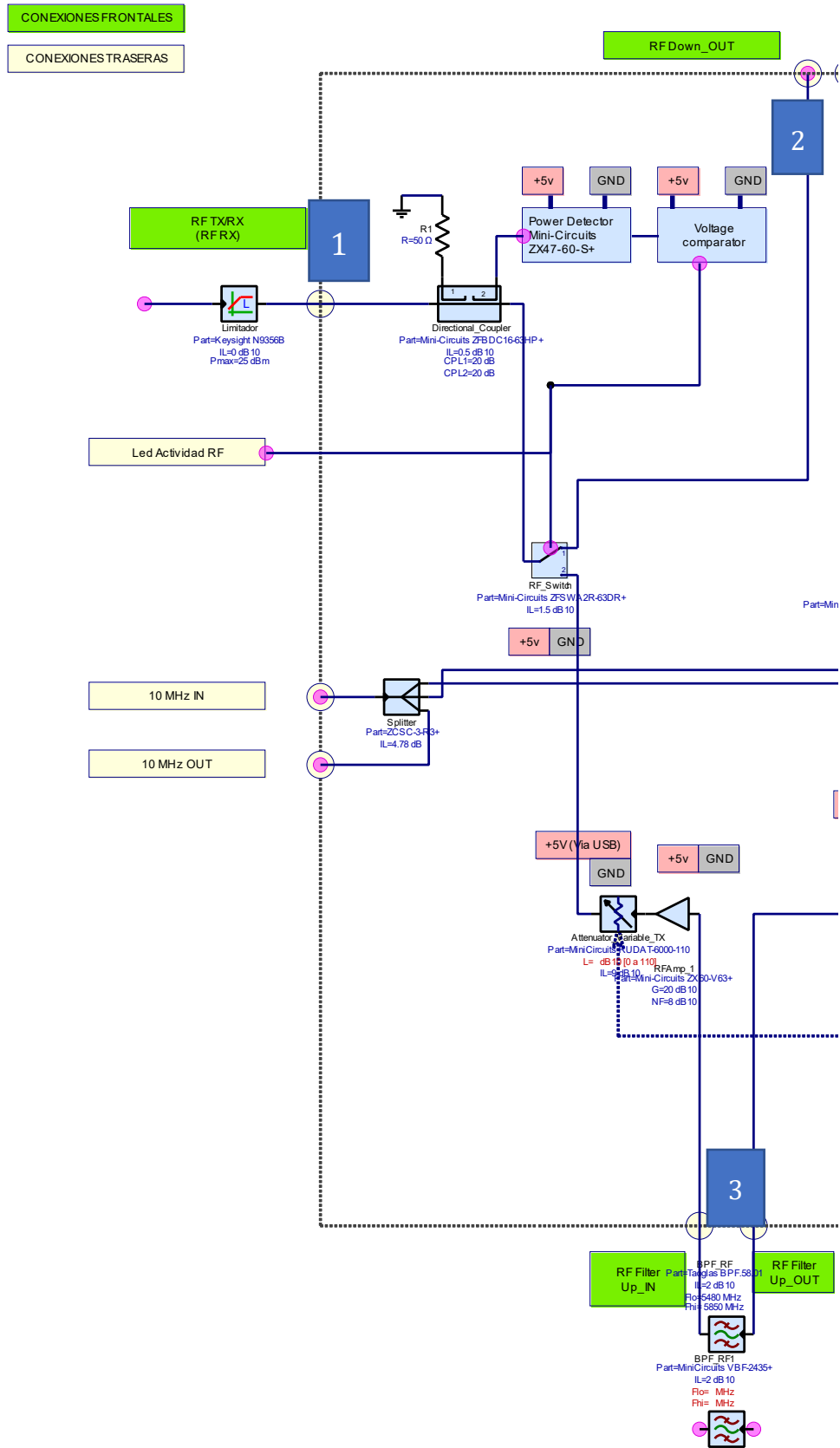


Figura 6.2. Etapa Entrada/Salida.

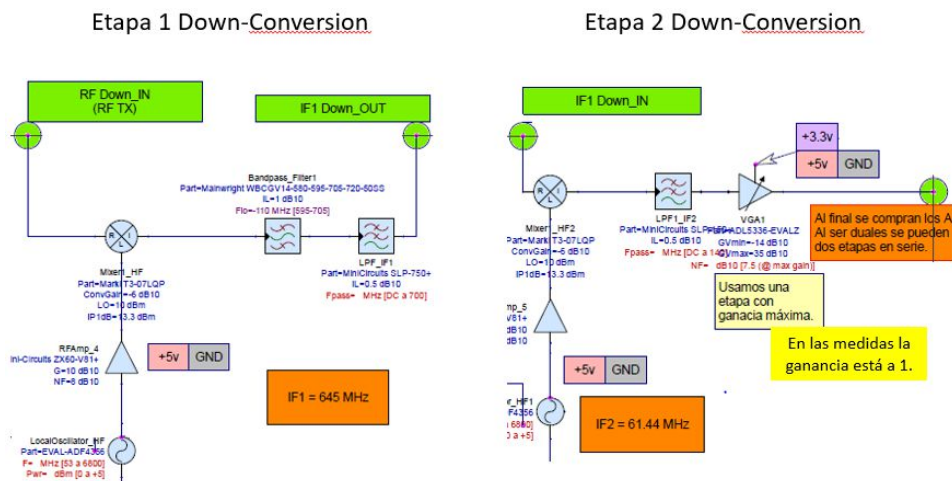


Figura 6.3. Etapas down-conversion.

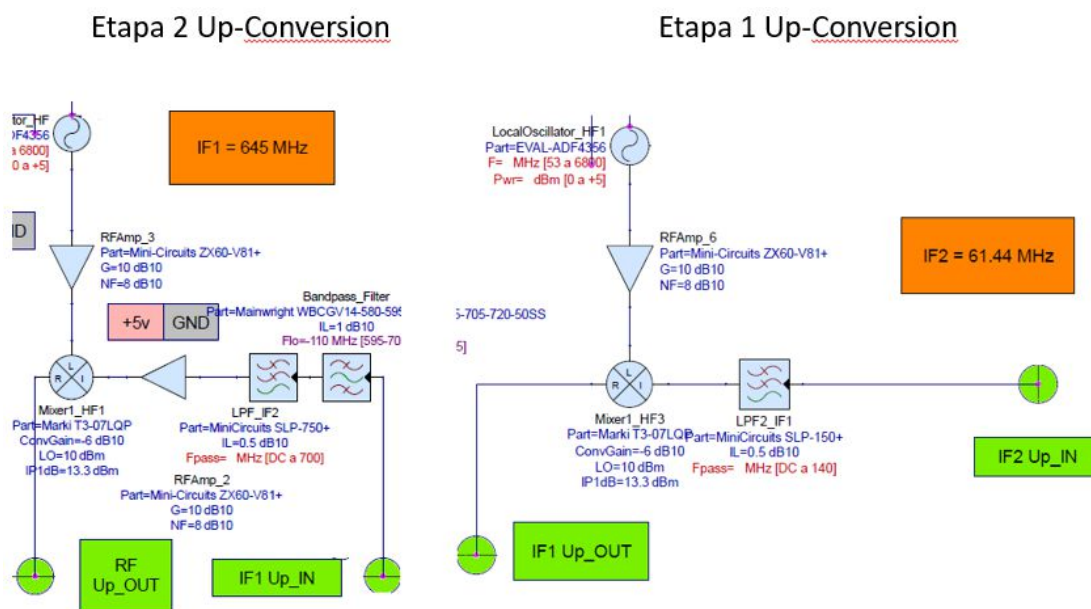


Figura 6.4. Etapas up-conversion.

En tercer lugar, se tienen las etapas de conversión de la señal IF a RF (Figura 6.4). En este caso la conversión también se realiza en dos fases. Para frecuencias mayores de 1400 MHz se utilizarán las etapas 1 y 2 up-conversion, en cambio, si está entre 400 MHz y 1400 MHz solo está activa la 1ª etapa de up-conversion. La señal digital que proviene de la FPGA se convierte del dominio discreto al continuo y posteriormente entra por la primera etapa de up-conversion.

Primero se filtra la señal que proviene de la FPGA a $IF2 = 61.44 \text{ MHz}$ y tras el paso de la señal por el mezclador, se obtiene a la salida del puerto $IF1 \text{ Up_Out}$ una

mezcla de la frecuencia $IF2$ y la frecuencia de oscilación del OL2 ($f_{OL2} = 645 \text{ MHz} - 61.44 \text{ MHz}$). Es decir, se obtiene la señal a la frecuencia intermedia $IF1 = 645 \text{ MHz}$. Cuando la señal llega a la 2 etapa de up-conversion esta se filtra y se mezcla con la frecuencia del OL1 ($f_{OL1} = f_c - 645 \text{ MHz}$) para obtener la frecuencia de RF transmitida en el puerto $RF \text{ Up_Out}$.

Antes de comenzar con la caracterización de los frontend es necesario configurar los elementos programables de los cuales dispone. A continuación, se describe como programar los dispositivos del frontend.

6.2. Configuración elementos programables del Frontend

El frontend contiene 4 elementos programables: 2 OLs, el amplificador de IF, y el atenuador variable. Para programar cada uno de los elementos se utiliza un software específico que proporciona el fabricante de cada uno de los elementos.

6.2.1. Atenuador variable RUDAT-6000-110

El software para el control del atenuador variable lo proporciona el fabricante del atenuador, Mini-Circuits [28]. Dentro de la sección “*Manual Attenuation*” se

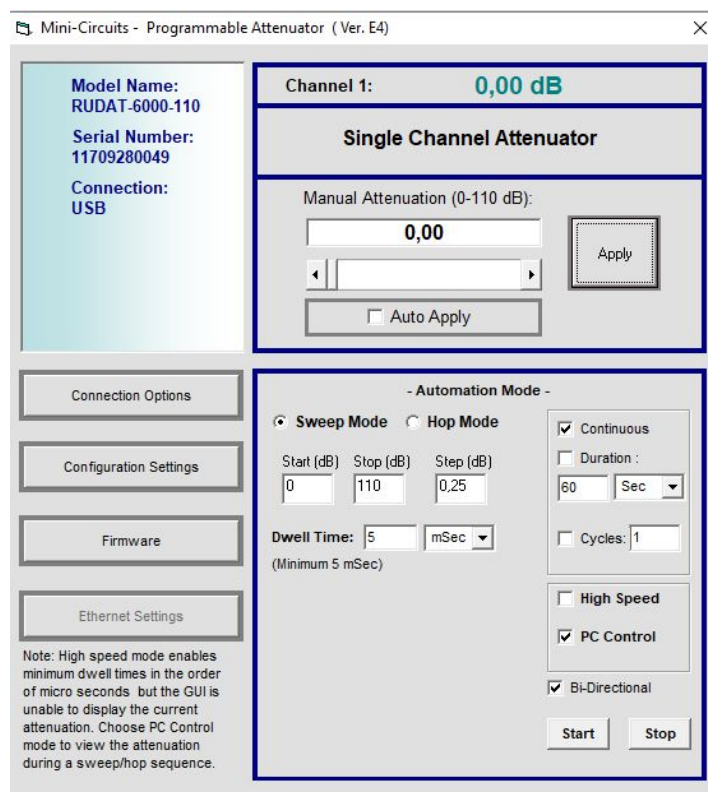


Figura 6.5. Software control atenuador variable RUDAT 6000 [28].

encuentra un campo para introducir el valor de atenuación, que admite valores de atenuación desde 0 a 110 dB. Una vez se introduce el valor de atenuación se pulsa sobre el botón “*Apply*” y el atenuador ya se encuentra configurado. Para realizar la caracterización de los frontend siempre se ha tomado un valor de atenuación de 0dB.

6.2.2. Amplificador de IF ADL5202

El software para el control del amplificador de IF lo proporciona el fabricante del amplificador, Analog Devices [27]. En este caso dentro del apartado de Device se selecciona el dispositivo a programar ADL5202 y se habilitan dos canales para introducir la ganancia. Cada canal admite valores desde -20 dB a 20 dB. Una vez se introduce el valor de ganancia para cada amplificador, se pulsa sobre el botón “*Program*” para configurar los dispositivos. Para realizar la caracterización de los frontend siempre se ha tomado unos valores de ganancia de 0dB, 10 dB, 20 dB, 30 dB y 40 dB.

6.2.3. Osciladores locales ADF4355

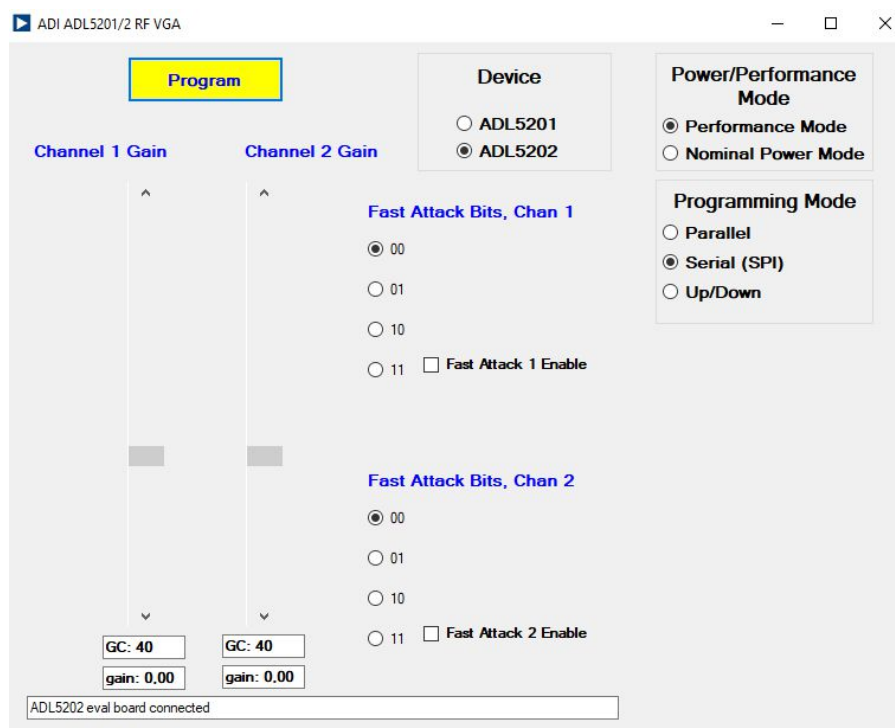


Figura 6.6. Software control amplificador de IF [27].

El software para el control del amplificador de IF lo proporciona el fabricante del oscilador, Analog Devices [37]. Para configurar los OLs es necesario abrir dos ventanas independientes para cada oscilador. En primer lugar, se procede a configurar el OL de la 1ª etapa de down-conversion/2ª etapa de up-conversion. Dentro de *RF Settings* hay que fijar la frecuencia de referencia o *Reference freq* a 10 MHz y después se introduce el valor del campo VCOout en MHz. La frecuencia de oscilación del OL se calcula mediante la siguiente expresión:

$$f_{OL1} = f_{central} - 645 \text{ MHz}, \quad (6.1)$$

donde $f_{central}$ es la frecuencia de operación y 645 MHz es la frecuencia a la que se realiza la primera conversión. Dicho valor de f_{OL1} se corresponde con el valor de RFout de la interfaz. Por lo tanto, en el campo de VCOout hay que introducir múltiplos del valor de RFout. Respecto a los campos *REFin mode* del registro 4 hay que fijarlo a modo Single y dentro del registro 6 habilitar tanto *RFoutA Enable* y *RFoutB Enable* y fijar la potencia *RFoutA Power* y *RFoutB Power* a 5 dBm.

Para proceder a configurar el OL2 o la 2ª etapa down-conversion/1ª etapa up-conversion, se deben hacer los mismos cambios tanto en el registro 4 y 6 como se

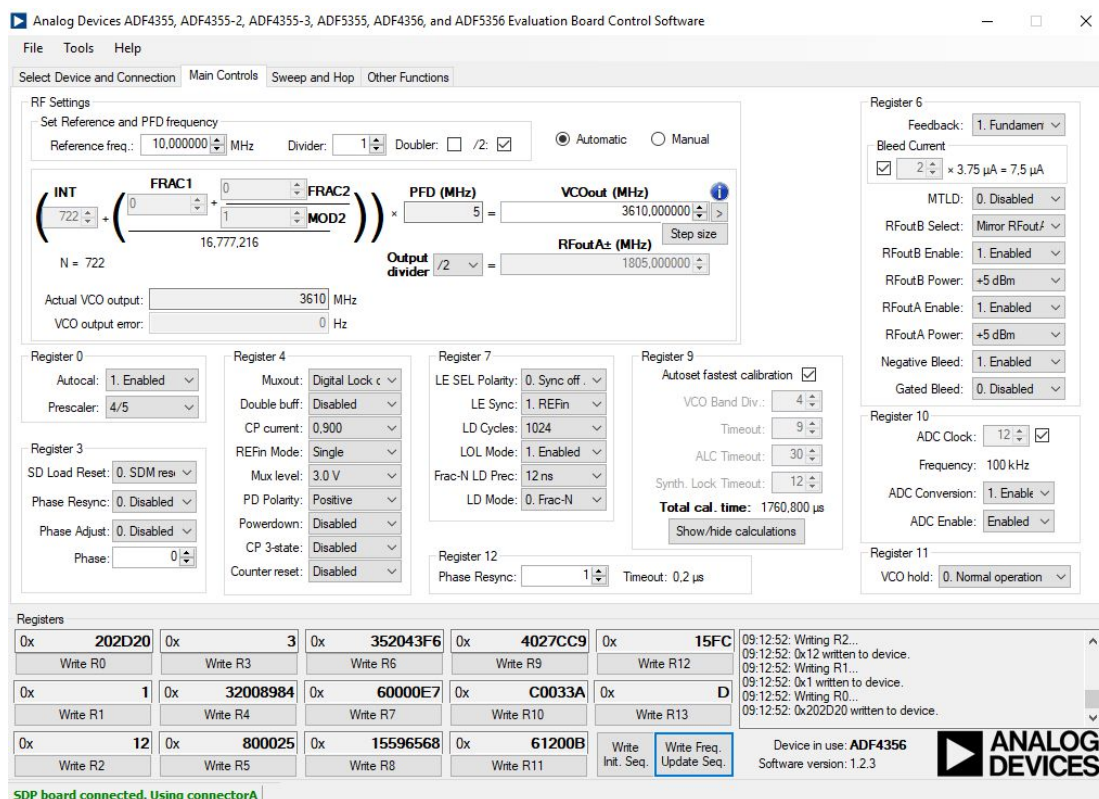


Figura 6.7. Programación OL1.

hizo en el caso del OL1. Y lo único que cambiará es el valor de VCOout ya que la segunda etapa de conversión la frecuencia baja a 61.44 MHz.

6.3. Hardware y mediciones para la caracterización

En el siguiente apartado se muestran los dispositivos utilizados para llevar a cabo la caracterización de los frontend así como la campaña de medidas.

6.3.1. Hardware empleado

- E5080A ENA Vector Network Analyzer

El analizador vectorial (VNA) se utiliza como generador de señal. Abarca el rango de frecuencias de 9 kHz hasta 6.5 GHz y con una potencia de salida máxima de 15 dBm. Con él se pueden medir los parámetros-S de componentes de hasta 4 puertos. En este caso ha sido utilizado para caracterizar cableado RF usado para conectar los diferentes puertos del frontend y como generador de señal.

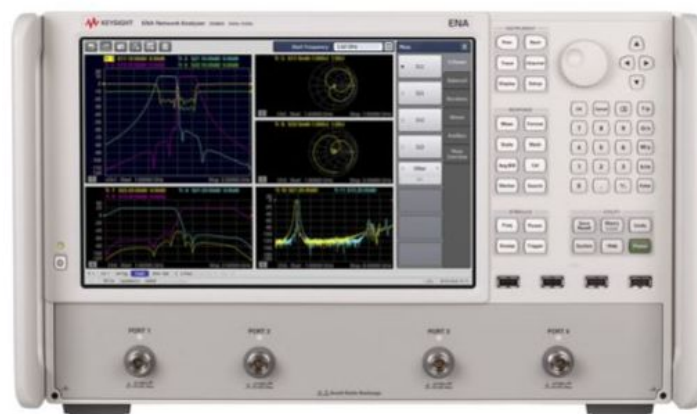


Figura 6.8. E5080A ENA Vector Network Analyzer.

- E4404B ESA-E Spectrum Analyzer

El analizador de espectros E4404B ESA-E de Keysight con el cual se puede visualizar el espectro de la señal en el rango de frecuencias de 9 kHz hasta 6.7 GHz. Se ha utilizado para ver el comportamiento del tono de frecuencia que inyecta el VNA una vez atraviesa los componentes del frontend. En este caso, se han medido las señales de los diferentes puertos de salida del frontend dependiendo de las etapas que se quieran caracterizar.



Figura 6.9. E4404B ESA-E Spectrum Analyzer.

6.3.2. Campaña de medidas

A continuación, se describen el conjunto de medidas que se han llevado a cabo para realizar la caracterización de cada una de las etapas que componen el frontend de RF.

6.3.2.1. Medidas etapa Entrada/Salida

Para esta etapa se han realizado las siguientes mediciones:

- Parámetros-S entre los puertos 1, 2 y 3.

Para medir los parámetros-S de esta etapa se ha utilizado el VNA tanto como generador como para visualizar los parámetros-S. Por otro lado, se han medido los siguientes parámetros:

- El parámetro S_{11} para conocer la relación entre la potencia enviada y reflejada.
- El parámetro S_{21} para obtener las pérdidas entre los puertos 2 y 1.
- El parámetro S_{13} para obtener las pérdidas entre los puertos 3 y 1.
- El parámetro S_{23} para evaluar el aislamiento de los puertos 2 y 3.

Se han caracterizado dichos parámetros para todas las bandas de interés de (400 – 6000 MHz) de manera simultánea siguiendo el siguiente procedimiento:

- En primer lugar, se ha configurado el VNA. La potencia se ha fijado a $P_{VNA} = 0 \text{ dBm}$, la frecuencia central a $F_C = 3200 \text{ MHz}$ con un ancho de banda $BW = 5610 \text{ MHz}$, un ancho de banda de IF de $BW_{IF} = 100 \text{ kHz}$ y una resolución de 56101 puntos.

- En segundo lugar, se procedió a la calibración de los cables que se usaron para conectar el VNA al emulador. Una vez los cables fueron calibrados, se conectó cada puerto del VNA a los puertos de la etapa entrada/salida.
- En tercer lugar, se barrió toda la banda de 395-6005 MHz, midiendo magnitud y fase de los parámetros-S. Por último, se guardaron los resultados en formato .s2p para posteriormente graficar las respuestas de las bandas de interés.

Para medir la potencia de activación del switch se ha utilizado el VNA como generador de señal y un osciloscopio para visualizar la señal. Por otro lado, se van a medir los siguientes parámetros:

- La potencia de entrada en el puerto 1, para saber cuál es la potencia óptima de activación de la etapa de entrada.
- La potencia de entrada del puerto 3, para saber cuál es la potencia máxima de la señal de salida del emulador sin que ésta active el switch de entrada.

Para caracterizar la potencia de activación de entrada del puerto 1 se siguió el siguiente procedimiento:

- En primer lugar, se configuró el VNA con una potencia $P_{VNA} = 0 \text{ dBm}$, una $F_{VNA} = F_C$ según la frecuencia central de la banda que se quiera estudiar y transmitiendo señal sinusoidal.
- Se realizaron las conexiones pertinentes, en este caso, se conecta un cable RF al VNA y al puerto 1 del frontend. El segundo puerto del frontend se conecta al puerto 2 del VNA y se pone una carga de 50Ω en el puerto 3 del frontend.
- En tercer lugar, se hace un barrido en potencia con el VNA visualizando el parámetro S21. Es decir, se comienza introduciendo potencias bajas con el VNA y se va aumentando la potencia hasta que se encienda el led del frontend. El led encendido indica que se ha activado el circuito de entrada y se puede visualizar las pérdidas del parámetro S21 en el VNA.

Para caracterizar la potencia de activación del puerto 3 se realiza el mismo procedimiento salvo que ahora la señal se introduce por el puerto 3 del frontend y se visualizará en el osciloscopio a través del puerto 1.

- Retardo de activación del switch entre los puertos 1 y 2 (Figura 6.2), es decir, el retardo en el que switch cambia de estado.

Para medir el retardo de activación que introduce la etapa entrada/salida igual que con la potencia de activación se ha utilizado el VNA y un osciloscopio para

visualizar la señal. Se medirá el retardo de activación entre la entrada de la señal por el puerto 1 y la salida de la señal por el puerto 2. Para caracterizar el retardo se ha seguido el siguiente procedimiento:

- En primer lugar, se configura el VNA con una potencia $P_{VNA} = 0 \text{ dBm}$, una $F_{VNA} = F_c$ según la frecuencia central de la banda que se quiera estudiar y una señal sinusoidal.
- En segundo lugar, se conecta el VNA a un cable RF y este a la entrada de un splitter. A su vez, la salida 1 del splitter se conecta a un cable RF, y este al puerto 1 del frontend. La salida 2 del splitter se conecta a un cable RF y este al canal 1 del osciloscopio. Posteriormente el puerto 2 del frontend se conecta al canal 2 del osciloscopio.
- En tercer lugar, se configura el osciloscopio con un trigger activado por el flanco de subida de la señal que le llega al splitter. A continuación, se habilita la señal del VNA y se mide el delay entre las dos señales visualizadas en el osciloscopio, es decir, la señal que atraviesa la salida 2 del splitter y la señal del puerto 2 del frontend.

6.3.2.2. Medidas etapas down-conversion y up-conversion

De estas etapas se han medido el parámetro S11, y las pérdidas de conversión.

Para ello, se ha utilizado el VNA como generador de señal y el analizador de espectros para visualizar el tono de frecuencia que se está emitiendo.

Antes de realizar las medidas es necesario programar el amplificador de IF, el atenuador y los osciladores locales con su respectivo software. Por otro lado, para tener conseguir una medida fiable, se ha sincronizado el reloj interno del front-end y el reloj del VNA con un reloj de rubidio de 10 MHz. El procedimiento para caracterizar el parámetro S11 en cualquier banda es el siguiente:

- En primer lugar, se configura el VNA para que emita la señal en la banda a caracterizar. Se fijará la potencia $P_{VNA} = 0 \text{ dBm}$, con una frecuencia $f_{VNA} = f_c$ igual a la frecuencia de operación, un ancho de banda $BW = 110 \text{ MHz}$, un ancho de banda de IF $BW_{IF} = 100 \text{ kHz}$ y con una resolución de 1101 puntos. Una vez el VNA está configurado se procede a calibrar el cable RF que se ha conectado en el puerto 1 del VNA.
- En segundo lugar, se conecta el cable RF al respectivo puerto del frontend donde se quiera introducir la señal y en el puerto de salida del frontend se conecta una carga de 50Ω . A continuación, se activa la señal del VNA y se barre toda la frecuencia midiendo la magnitud y fase del parámetro S11.

- Por último, se guardan los resultados en formato. s1p o .csv para posteriormente procesarlos en Matlab.

Para caracterizar las pérdidas de conversión en cualquier banda se puede realizar tanto de forma manual como automática con Matlab, pero en este caso se ha realizado de manera automática debido a que el proceso de caracterización es más rápido. Las pérdidas de conversión en el caso de realizar la conversión de RF a IF, se definen como la relación de potencia entre la señal de RF de entrada a la etapa de down-conversion y la potencia de la señal de salida IF a la salida de la etapa de down-conversion. Por el contrario, cuando se realiza la conversión de IF a RF, se define como la relación de potencia entre la señal de IF de entrada en la etapa de up-conversion y la potencia de la señal de IF a la salida de la etapa de up-conversion.

El procedimiento es el siguiente:

- En primer lugar, es necesario dar una señal de reloj de rubidio de 10 MHz para que todos los equipos estén sincronizados. Posteriormente se conecta el PC con los equipos los cuales se van a controlar desde Matlab. El analizador de espectros se conecta con el PC vía cable GPIB-USB y el VNA se conecta al PC vía cable USB.
- En segundo lugar, se caracterizan los cables RF los cuales son utilizados para conectar las etapas del frontend. De todos los cables RF se extraen sus pérdidas para posteriormente incluirlo en el script de Matlab.
- En tercer lugar, se realizan las conexiones pertinentes de los dispositivos. Se conecta el puerto 1 del VNA el cual tiene la función de generador de señal a un cable RF y este a la entrada de la etapa determinada del frontend por la cual se quiera inyecta la señal. Después se conectan con cables RF los puertos del frontend y se conecta el puerto de salida del frontend donde se quiera observar el tono de frecuencia tras la conversión a la entrada del analizador de espectros.
- El script de Matlab realiza las siguientes funciones:
 - o Cargar las pérdidas de los cables RF.
 - o Configurar los parámetros correspondientes a la señal RF que genera el VNA. Esto es, una potencia de transmisión de $P_{VNA} = 0 \text{ dBm}$, la frecuencia de la señal $f_{VNA} = f_c$ será la frecuencia de operación, un ancho de banda $BW = 110 \text{ MHz}$, un ancho de banda de IF $BW_{IF} = 100 \text{ kHz}$ y con una resolución de 1101 puntos.
 - o A continuación, se configura la frecuencia de los osciladores según la etapa que se quiera caracterizar. Tanto para la etapa 1 de down-

conversion como la etapa 2 de up-conversion el oscilador se configura según la siguiente expresión

$$f_{OL1} = f_{central} - 645 \text{ MHz}, \quad (6.2)$$

En cambio, para configurar el oscilador de la 2 etapa de down-conversion y la etapa 1 de up-conversion, la frecuencia del OL será

$$f_{OL2} = 645 \text{ MHz} - 61.44 \text{ MHz}, \quad (6.3)$$

- En cuanto al analizador de espectros, se fija la frecuencia central $f_{E4404B} = f_{VNA} - f_{OL}$ con un ancho de banda de $BW = 110 \text{ MHz}$, un ancho de banda de IF $BW_{IF} = 100 \text{ kHz}$ 1101 puntos de resolución. Respecto a la potencia de entrada del analizador de espectros debe ser aquella en la que no se sature el equipo.
- Una vez los dos equipos están configurados se procede a la adquisición de datos. Para ello, se hace un barrido en frecuencia con en el VNA y en el analizador de espectro se muestra el tono emitido para cada frecuencia junto con la potencia recibida. Los datos se guardan en una matriz, donde para cada punto de frecuencia se tiene su correspondiente valor de potencia. Finalmente, con los resultados obtenidos y con las pérdidas de los cables de RF se grafican las pérdidas de conversión.

Para analizar la potencia de los espurios respecto a la señal principal, se debe realizar una medida de la Signal to Interference Noise Ratio (SINR). Para ello se ha medido la señal del primer máximo presente en la banda tras el proceso de down-conversion/up-conversion y el valor del segundo máximo presenta en la banda (espurio de mayor potencia). El procedimiento para caracterizar los espurios es similar al que se ha realizado para las pérdidas de conversión. Los pasos a seguir son los siguientes:

- Inyectar una señal de reloj de rubidio de 10 MHz a todos los equipos para que estén sincronizados y posteriormente conectar el PC tanto al VNA como al analizador de espectros. Proceder con la caracterización de los cables de RF para posteriormente incluir sus pérdidas en el script de Matlab. Y realizar las conexiones pertinentes de las etapas del frontend que se quiera caracterizar.
- El script de Matlab tiene que realizar las siguientes funciones:

- Cargar las pérdidas de los cables RF que se caracterizaron con anterioridad.
- Configurar los parámetros correspondientes a la señal RF que genera el VNA. Esto es, una potencia de transmisión de $P_{VNA} = 0 \text{ dBm}$, la frecuencia de la señal $f_{VNA} = f_c$ será la frecuencia de operación, un ancho de banda $BW = 110 \text{ MHz}$, un ancho de banda de IF $BW_{IF} = 100 \text{ kHz}$ y con una resolución de 1101 puntos.
- A continuación, se configura la frecuencia de los osciladores según la etapa que se quiera caracterizar.
- En cuanto al analizador de espectros, se fija la frecuencia central $f_{E4404B} = f_{VNA} - f_{OL}$ con un ancho de banda de $BW = 110 \text{ MHz}$, un ancho de banda de IF $BW_{IF} = 100 \text{ kHz}$ 1101 puntos de resolución. Respecto a la potencia de entrada del analizador de espectros debe ser aquella en la que no se sature el equipo.
- Una vez los dos equipos están configurados se procede a la adquisición de datos. Se realiza un barrido en frecuencia con el VNA y en el analizador de espectros se visualizará el valor máximo de la señal recibida presente en la banda tras el proceso de conversión junto con los espurios de mayor potencia. Para cada valor de frecuencia se guardarán los valores de la SINR recibida en el analizador de espectros. La SINR se calculará como la diferencia de potencia entre el máximo de la señal y la potencia del espurio predominante en la banda. Posteriormente se graficarán los resultados obtenidos de SINR frente a la frecuencia.

6.4. Resultados caracterización Frontend

Aunque se han caracterizados los 4 frontend RF en la presente sección se muestran los resultados obtenidos únicamente para el frontend 1, ya que los resultados en los 4 frontend han sido similares. En la presente sección se muestran las caracterizaciones de las etapas de entrada/salida, down-conversion y up-conversion para la banda de 434 MHz, 868 MHz, y 2.4 GHz.

6.4.1. Banda 434 MHz

Como la frecuencia central se encuentra entre 400 MHz y 1400 MHz para realizar la conversión de frecuencia RF/IF solo se necesita un salto. Es decir, se utilizan la etapa 2 de down-conversion y la etapa 1 de up-conversion. Para caracterizar la banda de 434 MHz se emplea un ancho de banda de 5 MHz, una

resolución en frecuencia de 10 kHz y con un barrido de 501 puntos. Se caracteriza con un ancho de banda de 5 MHz debido a que el filtro que se utiliza para seleccionar la banda RF tras la segunda etapa de up-conversion presenta un ancho de banda de 5 MHz.

Para realizar la caracterización de esta banda se han configurado los elementos programables del frontend con los siguientes valores:

- El atenuador variable con una atenuación de 0 dB.
- La frecuencia del OL2 a $F_{OL2} = 372.56 \text{ MHz}$.
- La ganancia del amplificador de IF a 0 dB, 10 dB y 20 dB para observar el efecto de dichos valores en las pérdidas de conversión.

A continuación, se describen los resultados obtenidos de la caracterización en la banda de 434 MHz, en primer lugar, la caracterización de la etapa Entrada/Salida – down-conversion 2 y, en segundo lugar, la caracterización de la etapa Entrada/Salida – up-conversion1. Para cada una de las etapas se realizan medidas del parámetro S11, las pérdidas de conversión y los efectos de los espurios causado por la SINR como está descrito en la subsección 6.3.2.

6.4.1.1. Etapa Entrada/Salida & Etapa 2 de down-conversion

- Caracterización S11

En la Figura 6.10 se observa la magnitud del parámetro S11 en función de la frecuencia. En la frecuencia central de 434 MHz se obtiene una magnitud relativa del

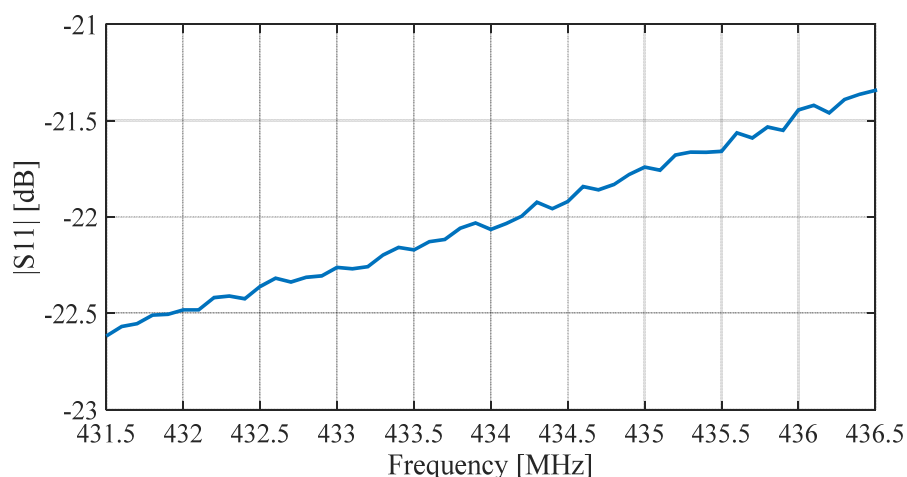


Figura 6.10. Caracterización S11 de la etapa down-conversion 2 y entrada salida para la banda de 434 MHz con un ancho de banda de 5 MHz con ganancia amplificador de IF a 0 dB.

coeficiente de reflexión aproximadamente de -22.05 dB. Esto quiere decir que

menos de un 1% de la potencia introducida al sistema es reflejada, lo que indica una adecuada adaptación.

- Caracterización pérdidas de conversión

En la Figura 6.11 se muestra las pérdidas de conversión en dB frente a la frecuencia en MHz. Para la frecuencia central de 434 MHz se obtiene unas pérdidas de conversión aproximadamente de 15.55 dB y un rizado de la señal en la banda de interés de 0.03 dB. Como se puede observar la respuesta de las pérdidas de conversión es plana en todo el ancho de banda por lo que se puede decir que no hay pérdida de potencia en la señal de salida.

Como se puede observar en la Figura 6.12, se ha caracterizado las pérdidas de conversión para diferentes valores de la ganancia del amplificador de IF y ver el efecto que produce dicha ganancia sobre las pérdidas de conversión. El valor de la ganancia del amplificador de IF se ha ido aumentando desde 0 a 20 dB. Para una frecuencia central de 434 MHz y con una ganancia de 10 dB se obtienen unas pérdidas de conversión de 5.45 dB aproximadamente, para 20 dB de -3.3 dB y con valor de rizado plano en la banda. Comparando los resultados de la Figura 6.12 con el resultado de la Figura 6.11 las pérdidas de conversión disminuyen a medida que aumenta la ganancia. Por lo tanto, el amplificador de IF es de utilidad para adecuar la potencia de salida de la etapa de conversión a IF y que está sea la suficiente para no saturar los ADC.

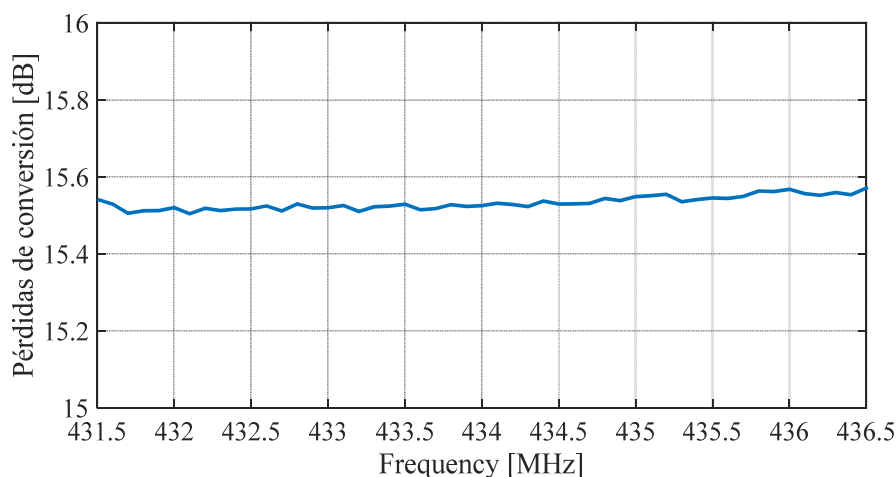


Figura 6.11. Caracterización de las pérdidas de conversión de la etapa down-conversion 2 y entrada salida para la banda de 434 MHz con un ancho de banda de 5 MHz con ganancia del amplificador de IF a 0 dB.

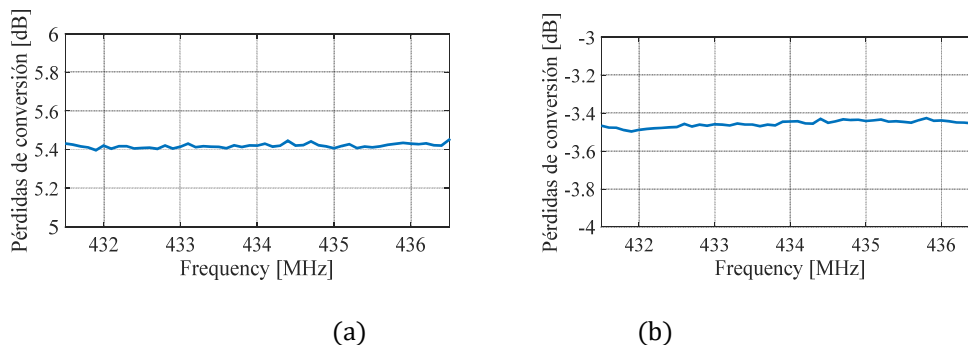


Figura 6.12. Caracterización de las pérdidas de conversión de la etapa down-conversion 2 y entrada salida para la banda de 434 MHz con un ancho de banda 5 MHz. (a) con ganancia de amplificador de IF = 10 dB, (b) con ganancia de amplificador de IF = 20 dB.

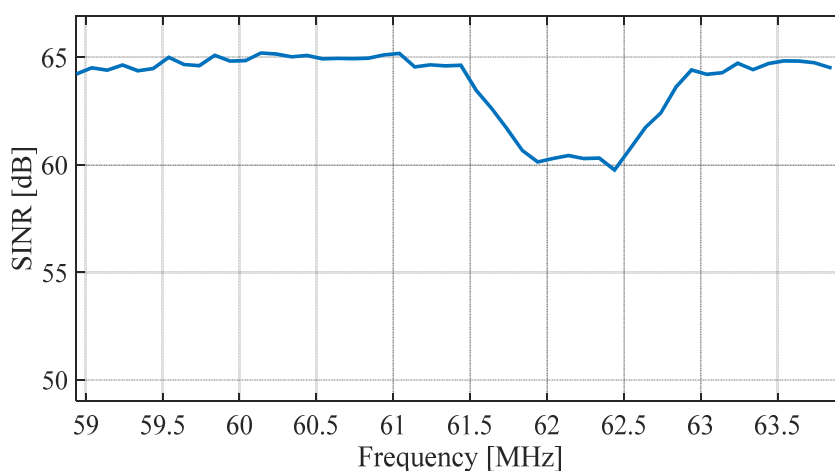


Figura 6.13. Caracterización SINR de la etapa down-conversion 2 y entrada salida para la banda de 434 MHz con un ancho de banda de 5 MHz con ganancia amplificador de IF a 0 dB.

- Caracterización espurios y SINR

A continuación, se realiza la caracterización de los espurios respecto a la señal principal a partir de la SINR. En la Figura 6.13 se muestra la medida de la SINR en dB frente a la frecuencia en MHz el cual toma un valor de 58.47 dB. Esta medida se realiza para ver los espurios que aparecen en la salida, además de la señal de mezcla. Con este parámetro se extrae información sobre la no linealidad de los mezcladores y el amplificador de salida. Este efecto de no linealidad se produce cuando la potencia de la señal de entrada es muy grande y llega un momento que el amplificador comienza a saturarse causando lo que se denomina productos de intermodulación de tercer orden.

6.4.1.2. Etapa Entrada/Salida y Etapa 1 de up-conversion

- Caracterización S11

En la Figura 6.14 se observa la magnitud del parámetro S11 en dB en función de la frecuencia en Hz. En la frecuencia central de 61.44 MHz se obtiene una magnitud relativa del coeficiente de reflexión aproximadamente de -15.5 dB, esto quiere decir que a esta frecuencia presenta buena adaptación.

- Caracterización pérdidas de conversión

En la Figura 6.15 se muestra las pérdidas de conversión en dB frente a la frecuencia en MHz. Para la frecuencia central de 61.44 MHz se obtiene unas pérdidas de conversión aproximadamente de -7 dB y un rizado de la señal en la banda de interés de 0.27 dB. Como se puede observar la respuesta de las pérdidas de conversión es plana en toda la banda y dichas pérdidas son mínimas.

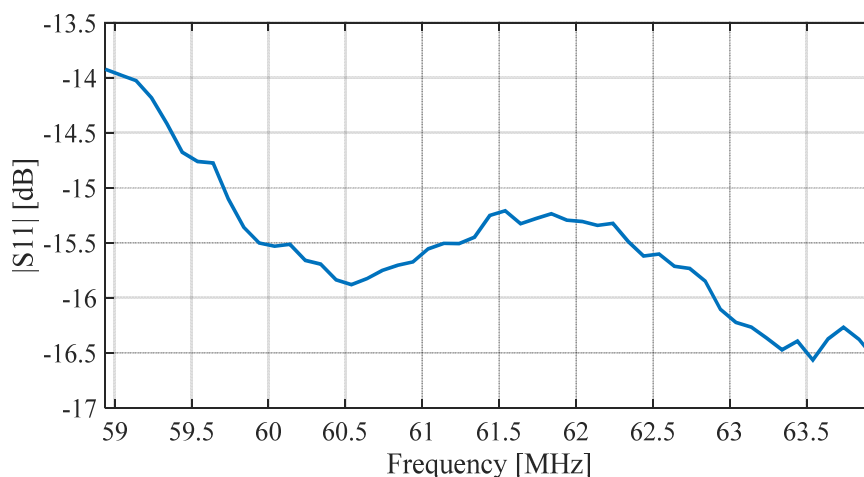


Figura 6.14. Caracterización S11 de la etapa up-conversion 1 y entrada salida para la banda de 434 MHz con un ancho de banda de 5 MHz.

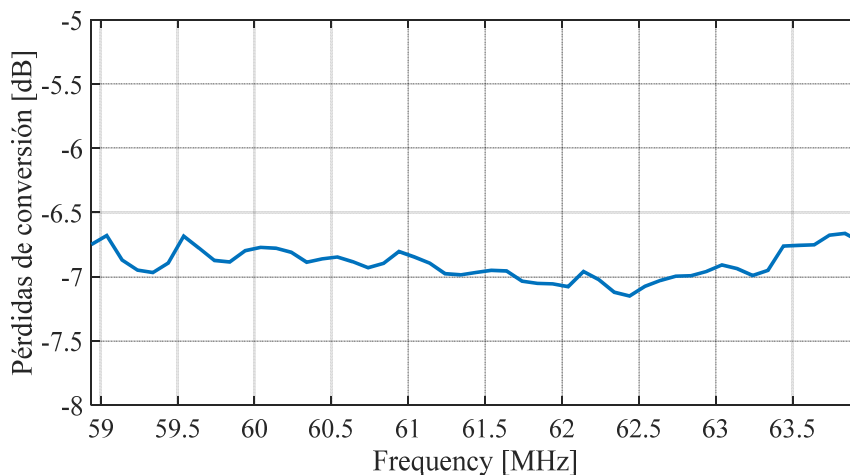


Figura 6.15. Caracterización de las pérdidas de conversión de la etapa up-conversion 1 y entrada salida para la banda de 434 MHz con un ancho de banda de 5 MHz con ganancia del amplificador de IF a 0 dB.

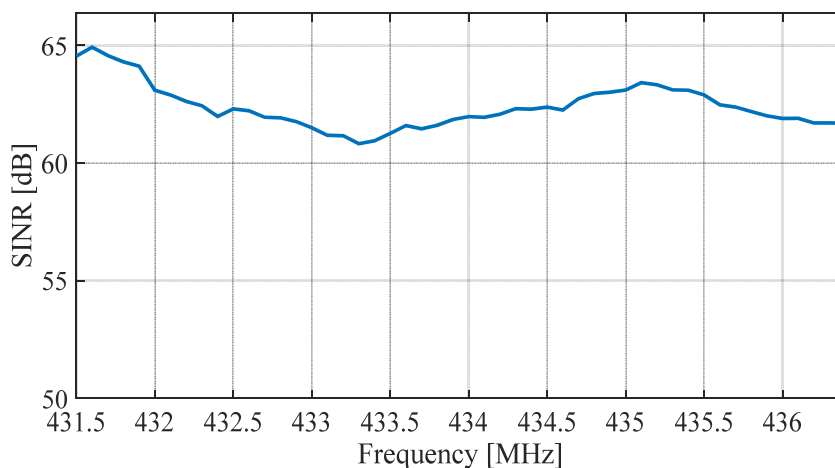


Figura 6.16. Caracterización SINR de la etapa up-conversion 1 y entrada salida para la banda de 434 MHz con un ancho de banda de 5 MHz con ganancia del amplificador de IF a 0 dB.

- Caracterización espurios y SINR

A continuación, se realiza la caracterización de los espurios respecto a la señal principal a partir de la SINR. En la Figura 6.16 se muestra la medida de la SINR en dB frente a la frecuencia en MHz que toma un valor de 70.95 dB.

6.4.2. Banda 868 MHz

Igual que ocurre en la caracterización de la banda de 434 MHz solo se utiliza un salto para convertir la señal de RF a IF o viceversa. Debido a que el ancho de banda del filtro RF de salida de la etapa de up-conversion presenta un ancho de banda de 5 MHz, se ha caracterizado la banda con un ancho de 5 MHz, una resolución en frecuencia de 10 kHz y con un barrido de 501 puntos.

Para realizar la caracterización de esta banda se han configurado los elementos programables del frontend con los siguientes valores:

- El atenuador variable con una atenuación de 0 dB
- La frecuencia del OL2 según la siguiente expresión

$$F_{OL2} = 868 \text{ MHz} - 61.44 \text{ MHz} = 806.56 \text{ MHz}$$

- La ganancia del amplificador de IF a 0 dB, 10 dB y 20 dB para observar el efecto de dichos valores en las pérdidas de conversión.

A continuación, se describen los resultados obtenidos de la caracterización en la banda de 868 MHz, en primer lugar, la caracterización de la etapa Entrada/Salida – down-conversion² y, en segundo lugar, la caracterización de la etapa Entrada/Salida – up-conversion¹. Para cada una de las etapas se realizan medidas del parámetro S11, las pérdidas de conversión y los efectos de los espurios causado por la SINR como se explicó en la subsección 6.3.2.

6.4.2.1. Etapa Entrada/Salida y Etapa 2 de down-conversion

- Caracterización S11

En la Figura 6.17 se observa la magnitud del parámetro S11 en función de la frecuencia. Para la frecuencia central de 868 MHz se obtiene una magnitud relativa del coeficiente de reflexión con unas pérdidas mínimas aproximadamente de -35.5 dB.

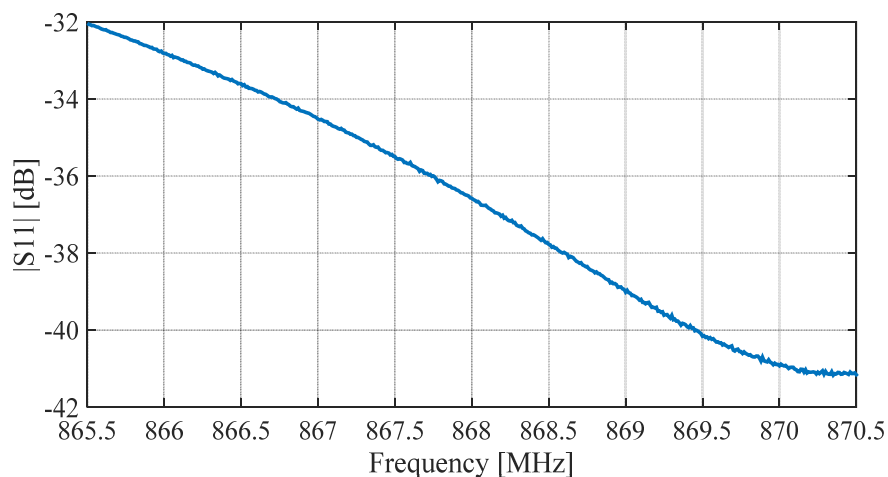


Figura 6.17. Caracterización S11 de la etapa down-conversion 2 y entrada salida para la banda de 868 MHz con un ancho de banda de 5 MHz y ganancia del amplificador de IF a 0 dB.

- Caracterización de las pérdidas de conversión

En la Figura 6.18 se muestra las pérdidas de conversión en dB frente a la frecuencia en MHz. Para la frecuencia central de 868 MHz se obtiene unas pérdidas de conversión aproximadamente de 35.95 dB y un rizado de la señal en la banda de interés de 0.07 dB.

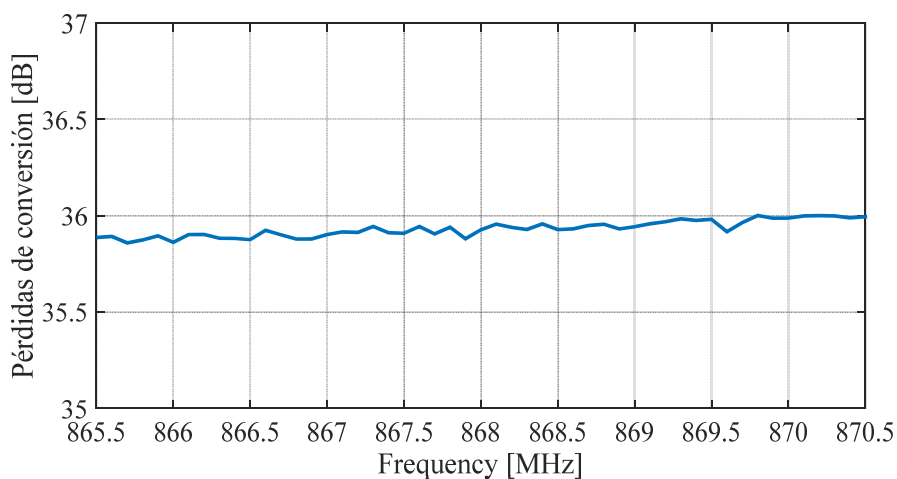


Figura 6.18. Pérdidas de conversión de la etapa down-conversion 2 y entrada salida para la banda de 868 MHz con un ancho de banda de 5 MHz con ganancia del amplificador de IF a 0 dB.

Como se puede observar en la Figura 6.19, se han caracterizado las pérdidas de conversión para diferentes valores de la ganancia del amplificador de IF, es decir, para 10 dB y 20 dB. Para una ganancia de 10 dB se obtienen unas pérdidas de

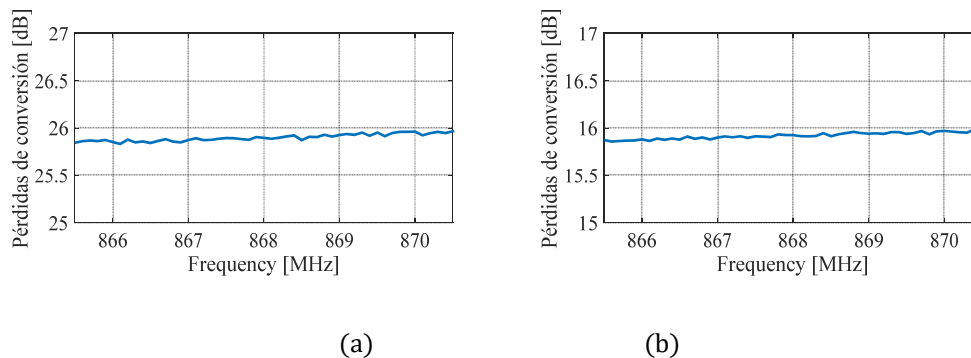


Figura 6.19. Pérdidas de conversión de la etapa down-conversion 2 y entrada salida para la banda de 868 MHz con un ancho de banda de 5 MHz. (a) con ganancia de amplificador de IF = 10 dB, (b) con ganancia de amplificador de IF = 20 dB.

conversión de 25.87 dB aproximadamente y para 20 dB de 15.92 dB. Además, presentan un rizado plano en la banda aproximadamente de 0.07 dB.

- Caracterización espurios y SINR

A continuación, se realiza la caracterización de los espurios y SINR. En la Figura 6.20 se muestra la medida de la SINR en dB frente a la frecuencia en MHz el cual toma un valor de 57.84 dB.

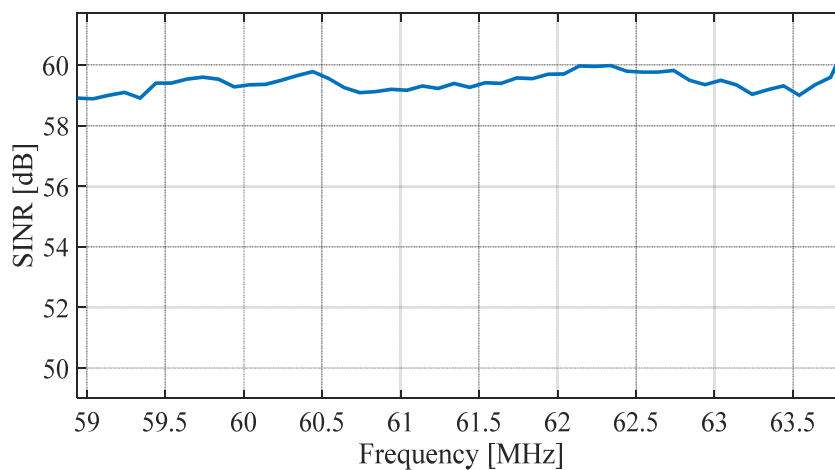


Figura 6.20. Caracterización SINR de la etapa down conversion 2 y entrada salida para la banda de 868 MHz con ancho de banda de 5 MHz con ganancia del amplificador de IF a 0 dB.

6.4.2.2. Etapa Entrada/Salida y Etapa 1 de up-conversion

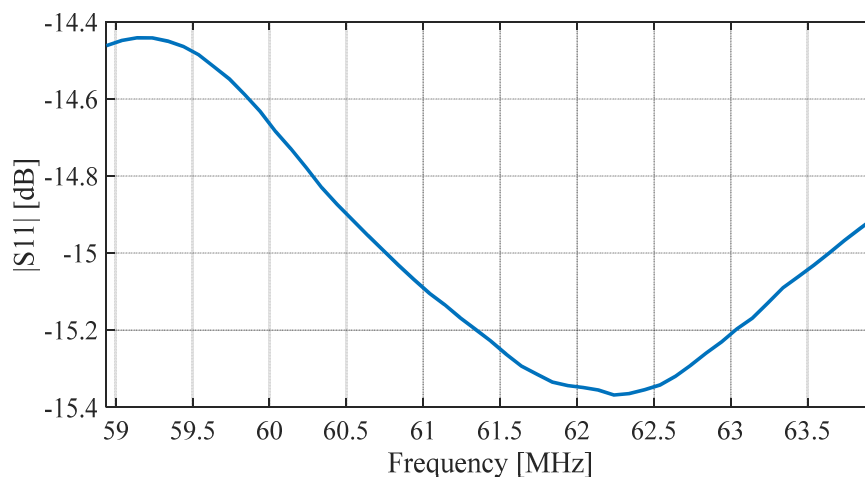


Figura 6.21. Caracterización S11 para la banda de 868MHz con un ancho de banda de 5 MHz para la etapa Entrada/Salida y up-conversion 1.

- Caracterización S11

En la Figura 6.21 se observa la magnitud del parámetro S11 en dB en función de la frecuencia en Hz. En la frecuencia central de 61.44 MHz se obtiene una magnitud relativa del coeficiente de reflexión aproximadamente de -15.1 dB, esto quiere decir que a esta frecuencia presenta buena adaptación.

- Caracterización de las pérdidas de conversión

En la Figura 6.22 se muestra las pérdidas de conversión en dB frente a la frecuencia en MHz. Para la frecuencia central de 61.44 MHz se obtiene unas pérdidas de conversión aproximadamente de -4.24 dB y un rizado de la señal en la banda de interés de 0.12 dB. Como se puede observar la respuesta de las pérdidas de conversión es plana en toda la banda y dichas pérdidas son mínimas.

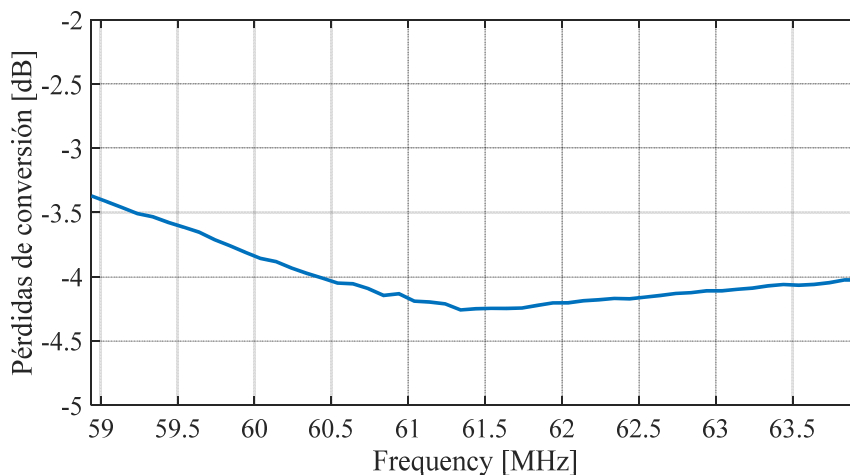


Figura 6.22. Caracterización de las pérdidas de conversión para la banda de 868 MHz con un ancho de banda de 5 MHz para la etapa Entrada/Salida y up-conversion1.

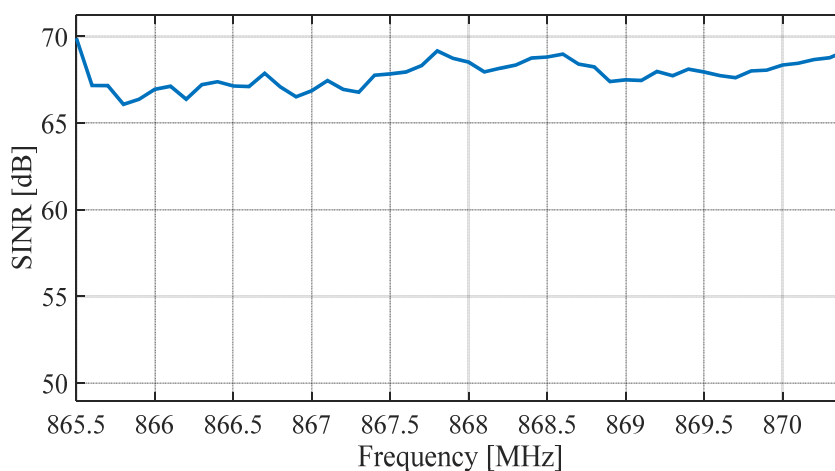


Figura 6.23. Caracterización SINR para la banda de 868 MHz con un ancho de banda de 5 MHz para la etapa Entrada/Salida y up-conversion 1.

- Caracterización espurios y SINR

A continuación, se realiza la caracterización de los espurios respecto a la señal principal a partir de la SINR. En la Figura 6.23 se muestra la medida de la SINR en dB frente a la frecuencia en MHz el cual toma un valor de 63.94 dB.

6.4.3. Banda 2.45 GHz

Como la frecuencia central de 2.45 GHz es mayor que 1400 MHz para transformar la señal de RF a IF o viceversa, es necesario utilizar las dos etapas que dispone el frontend para realizar la conversión. Es decir, en la primera conversión la señal RF a 2.45 GHz pasa a una frecuencia $IF_1 = 645 \text{ MHz}$ y en la segunda etapa de 645MHz a $IF_2 = 61.44 \text{ MHz}$. Para caracterizar la banda de 2.45 GHz se emplea un ancho de banda de 110 MHz, una resolución en frecuencia de 100 kHz y con un barrido de 1101 puntos. Se caracteriza con un ancho de banda de 110 MHz debido a que el filtro centrado a 2.45 GHz que se utiliza para seleccionar la banda RF tras la segunda etapa de up-conversion presenta un ancho de banda de 110 MHz.

Para realizar la caracterización de esta banda se han configurado los elementos programables del frontend con los siguientes valores:

- El atenuador variable con una atenuación de 0 dB
- La frecuencia del OL1 según la siguiente expresión

$$F_{OL1} = 2.45 \text{ GHz} - 654 \text{ MHz} = 1805 \text{ MHz}$$

- La frecuencia del OL2 según la siguiente expresión

$$F_{OL2} = 645 \text{ MHz} - 61.44 \text{ MHz} = 583.56 \text{ MHz}$$

- La ganancia del amplificador de IF a 0 dB, 10 dB y 20 dB para observar el efecto de dichos valores en las pérdidas de conversión.

A continuación, se describen los resultados obtenidos de la caracterización en la banda de 2.45 GHz, en primer lugar, la caracterización de la etapa Entrada/Salida, en segundo lugar, las etapas Entrada/Salida y down-conversion1-2 y Entrada/Salida y up-conversion1-2; y, por último, la caracterización del frontend completo.

6.4.3.1. Etapa Entrada/Salida

Para caracterizar la etapa Entrada/Salida se han seguido los pasos descritos en la sección 6.3.2.1.

- Caracterización Parámetros-S

En la Figura 6.24 se muestra la magnitud y fase de los parámetros-S respecto a la frecuencia en MHz. A la frecuencia central de 2.45 GHz la etapa de Entrada/Salida, presenta unas pérdidas de reflexión aproximadamente de -29 dB. Respecto al parámetro S21, presenta unas pérdidas de -3.4 dB, por lo que se puede decir que gran parte de la potencia introducida por el puerto '1' pasa a las etapas de down-conversion. Para obtener las pérdidas entre los puertos '3' y '1', se mide el parámetro S13 aproximadamente de 5.25 dB. Y, por último, se mide el parámetro S23 para obtener cuanta potencia se filtra a las etapas de down-conversion, cuyo valor es de -30 dB.

- Caracterización Potencia de activación

En la Tabla 6.1 se muestra la potencia mínima a la que se activa el switch de la etapa de entrada para el ancho de banda a caracterizar, es decir, la frecuencia central centrada a 2.45 GHz y con un ancho de banda de 110 MHz. Se observa que para potencias mínimas de entrada se activa el switch, por lo tanto, se ha decidido tomar como valor de referencia de potencia de entrada 0 dBm para la caracterización de todos los frontend.

- Caracterización del Retardo de activación

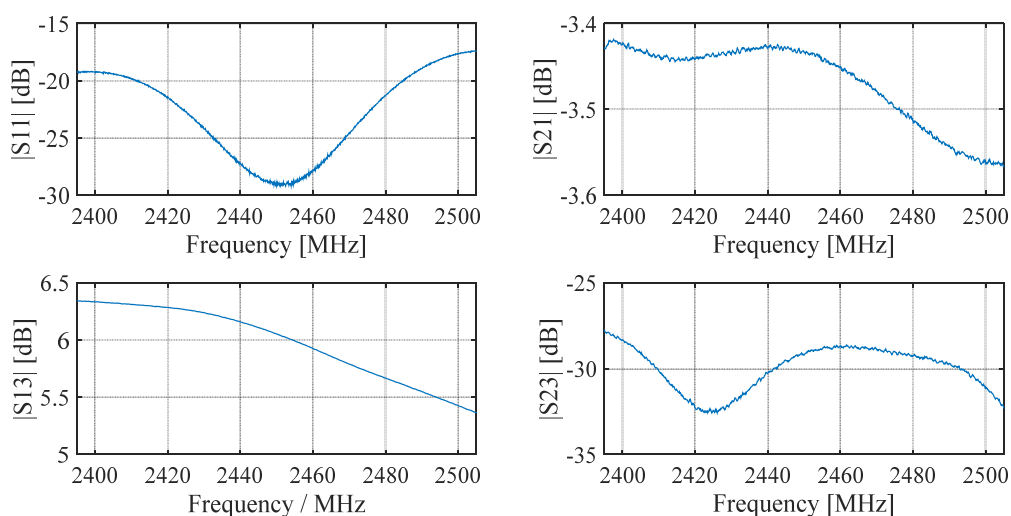


Figura 6.24. Caracterización Parámetros-S para etapa entrada salida en la banda de 2.45 GHz.

<i>Frecuencia</i>	2.395 GHz	2.45 GHz	2.505 GHz
<i>Potencia de activación</i>	-4.8 dBm	-5.2 dBm	-5.5 dBm

Tabla 6.1. Caracterización Potencia de activación del switch.

<i>Frecuencia</i>	2.395 GHz	2.45 GHz	2.505 GHz
<i>Retardo de activación</i>	130 ns	130 ns	130 ns

Tabla 6.2. Caracterización Retardo de activación del switch.

Por otro lado, se ha realizado la medida del retardo de activación del switch RF. Dicho retardo debe ser pequeño (del orden de nanosegundos), para evitar que se pierda parte de la información de la señal transmitida. En la Tabla 6.2 se recogen los valores de retardo de activación de la etapa de entrada para todo el ancho de banda a caracterizar. Como se recoge en la tabla, el retardo de activación del switch es para todas las bandas testeadas de únicamente 130 nanosegundos, y está dentro de los valores esperables.

6.4.3.2. Etapa Entrada/Salida y down-conversion 1-2

- Caracterización S11

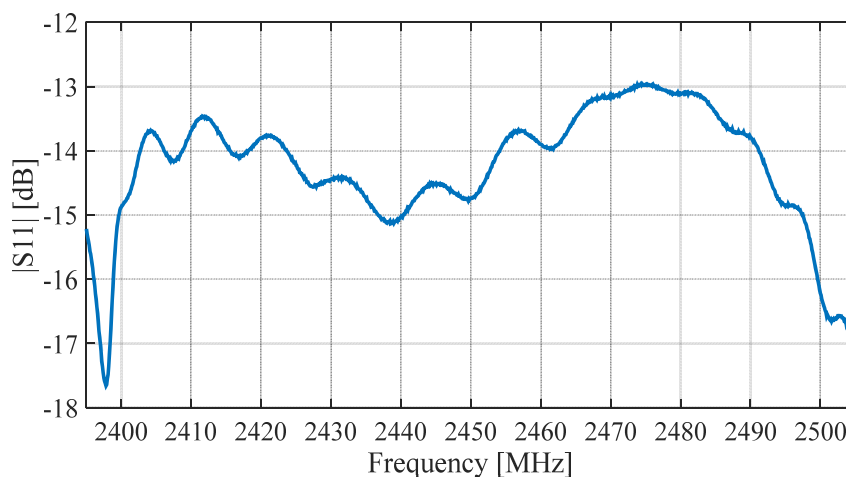


Figura 6.25. Caracterización S11 para la etapa down-conversion 1-2 y entrada salida en la banda de 2.45 GHz con un ancho de banda de 110 MHz y ganancia del amplificador de IF a 0dB.

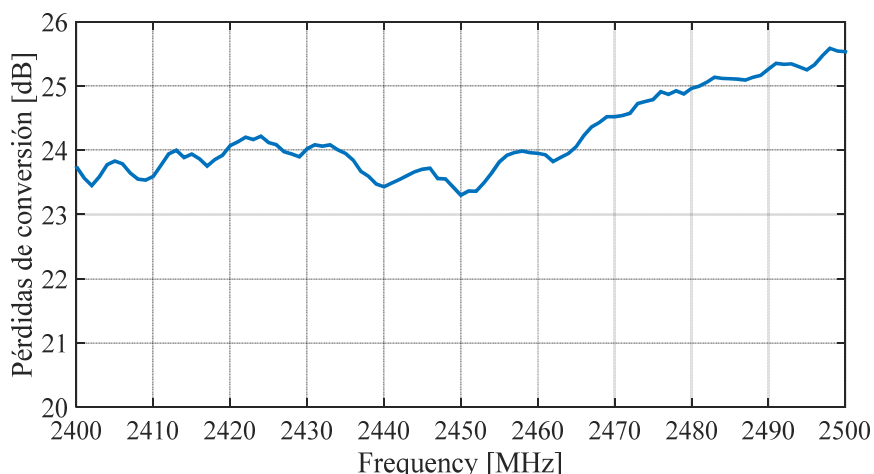


Figura 6.26. Caracterización de las pérdidas de conversión para la etapa down-conversion 1-2 y entrada salida en la banda de 2.45 GHz con un ancho de banda de 110 Mhz y ganancia del amplificador de IF a 0dB.

En la Figura 6.25 se observa tanto la magnitud del parámetro S11 en dB en función de la frecuencia como el parámetro S11 representado en la carta de Smith. A la frecuencia central de 2.45 GHz se obtiene una magnitud relativa del coeficiente de reflexión aproximadamente de -15 dB, y como se puede observar en la carta de Smith dicho parámetro presenta buena adaptación.

- Caracterización pérdidas de conversión

En la Figura 6.26 se muestra las pérdidas de conversión en dB frente a la frecuencia en MHz. Para la frecuencia central de 2.45 GHz se obtiene unas pérdidas de conversión aproximadamente de 23.2 dB y un rizado de la señal en la banda de interés de 2.62 dB. Como se puede observar a partir de la frecuencia de 2.47 GHz las pérdidas de conversión aumentan debido a que los elementos del frontend presentan unas mayores pérdidas a altas frecuencias.

En la Figura 6.27 se muestra la caracterización de las pérdidas de conversión para diferentes valores de la ganancia del amplificador igual a 10 dB y 20 dB. Para una ganancia de 10 dB se obtienen unas pérdidas de conversión de 13 dB aproximadamente con un rizado de 2.69 dB. Y para una ganancia del amplificador de IF de 20 dB las pérdidas de conversión toman un valor de 2.4 dB y un rizado de 2.71 dB.

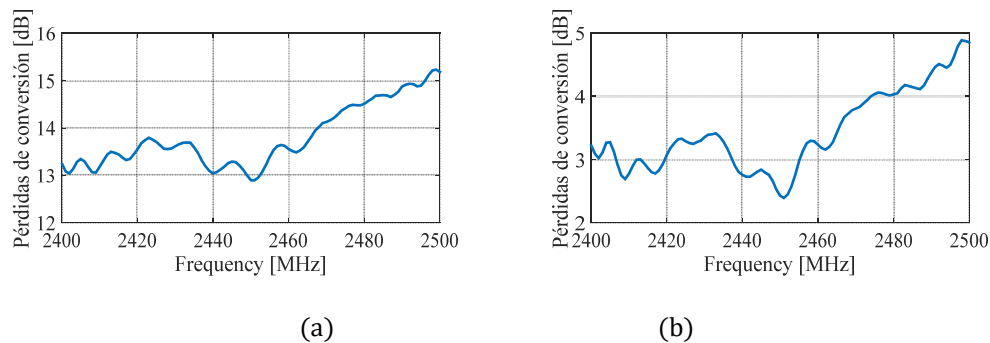


Figura 6.27. Caracterización de las pérdidas de conversión para la etapa de down-conversion 1-2 y entrada salida en la banda de 2.45 GHz con un ancho de banda de 110 Mhz. (a) con ganancia de amplificador de IF = 10 dB, (b) con ganancia de amplificador de IF = 20 dB.

Comparando los resultados de la Figura 6.26 con el resultado de la Figura 6.27 las pérdidas de conversión disminuyen a medida que aumenta la ganancia del amplificador de IF.

- Caracterización SINR

A continuación, se realiza la caracterización de los espurios respecto a la señal principal a partir de la SINR. En la Figura 6.28 se muestra la medida de la SINR en dB frente a la frecuencia en MHz que toma un valor mínimo de 35.37 dB.

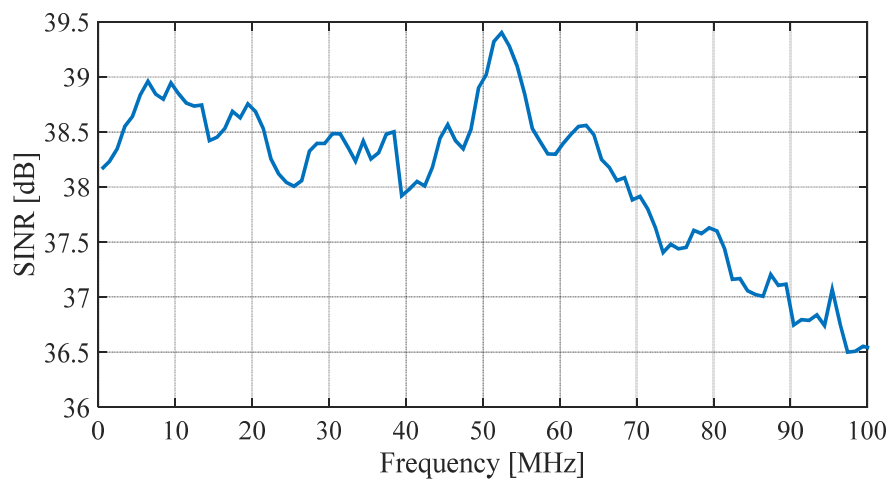


Figura 6.28. Caracterización SINR para la etapa down-conversion 1-2 y entrada salida en la banda de 2.45 GHz con un ancho de banda de 110 MHz.

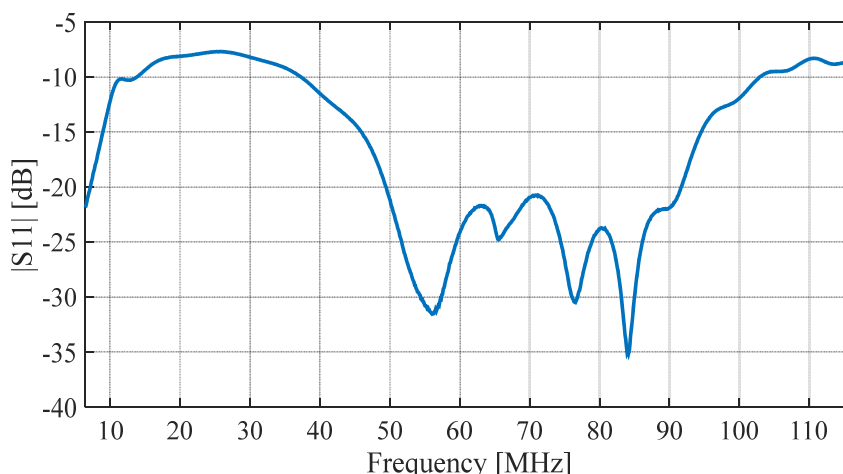


Figura 6.29. Caracterización S11 para la etapa up-conversion 1-2 y entrada salida en la banda de 2.45 GHz con un ancho de banda de 110 MHz.

6.4.3.3. Etapa Entrada/Salida y up-conversion 1-2

- Caracterización S11

En la Figura 6.29 se observa tanto la magnitud del parámetro S11 en dB en función de la frecuencia como el parámetro S11 representado en la carta de Smith. A la frecuencia central de 61.44 MHz se obtiene una magnitud relativa del coeficiente de reflexión aproximadamente de -26 dB. Sin embargo, la adaptación no es excesivamente buena para el resto de la banda, presentando un máximo de -8 dB en 25 MHz.

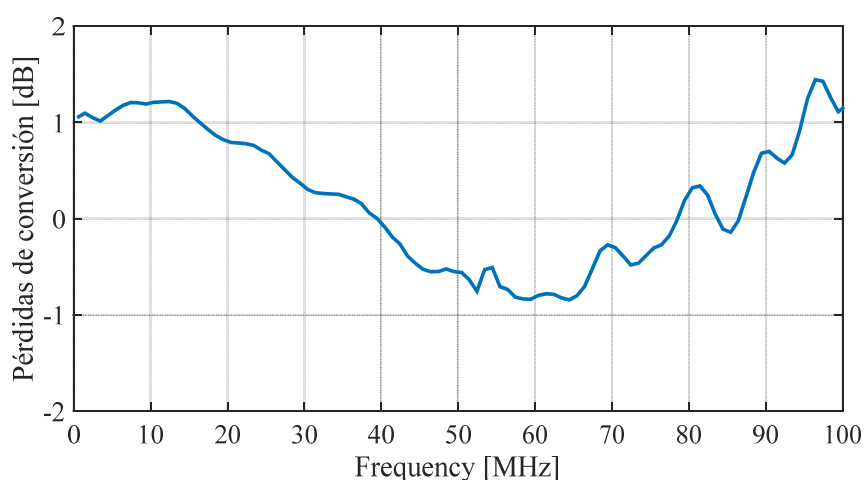


Figura 6.30. Caracterización de las pérdidas de conversión para la etapa up-conversion 1-2 y entrada salida en la banda de 2.45 GHz con un ancho de banda de 110 MHz y ganancia del amplificador de IF a 0dB.

- Caracterización de las pérdidas de conversión

En la Figura 6.30 se muestra las pérdidas de conversión en dB frente a la frecuencia en MHz. Para la frecuencia central de 61.44 MHz se obtiene unas pérdidas de conversión aproximadamente de -1.2 dB y un rizado de la señal en la banda de interés de 2.28 dB.

- Caracterización SINR

A continuación, se realiza la caracterización de los espurios respecto a la señal principal a partir de la SINR. En la Figura 6.31 se muestra la medida de la SINR en dB frente a la frecuencia en MHz el cual toma un valor de 40.39 dB y además se puede observar que a la frecuencia de 2.44 GHz se produce un salto generado por la presencia de un espurio.

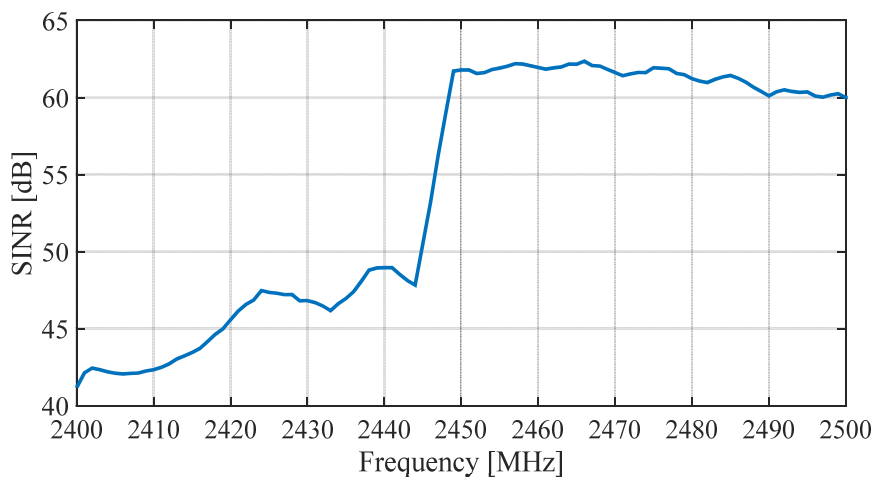


Figura 6.31. Caracterización SINR para la etapa up-conversion 1-2 y entrada salida en la banda de 2.45 GHz con un ancho de banda de 110 MHz.

Capítulo 7. Conclusiones

En este trabajo se ha realizado una descripción general sobre las comunicaciones inalámbricas y sus requisitos en la Industria 4.0. Se ha visto que el elemento más importante de una comunicación inalámbrica es el canal inalámbrico y una metodología de validación en el laboratorio es mediante el uso de un emulador de canal. Se ha detallado cómo caracterizar los principales problemas de propagación de la señal electromagnética en entornos industriales y emularse su comportamiento en un laboratorio.

En este trabajo se ha presentado el diseño de un emulador de canal banda base implementado en la Zynq UltraScale+ MPSoC. Dicha plataforma es capaz de emular canales estáticos, generar el efecto de multipath y desvanecimiento Doppler. Se ha realizado el estudio de la arquitectura del emulador de canal desde las etapas para la conversión de RF/IF y el procesamiento interno de la señal que se realiza para generar un canal banda base. Se ha visto que el emulador se puede conectar a 4 nodos inalámbricos comerciales, debido a que presenta un ancho de banda de 100 MHz que cubre las bandas de 400 MHz hasta 6 GHz. Pudiendo así probar la eficiencia de enlaces MIMO bidireccionales.

Por otro lado, disponer de la plataforma como la Zynq UltraScale+ sobre la que se puedan implementar un número mayor de funcionalidades para caracterizar el canal inalámbrico conlleva un gran número de ventajas. Algunas de estas ventajas, entre otras, son: disponer de una interfaz de control Linux que sea capaz de controlar dispositivos embebidos, extraer información correspondiente a una interfaz de usuario y comande todas las estructuras de configuración de bajo nivel a otro procesador remoto disponible en la tarjeta.

Además, se ha realizado la caracterización del diseño de los frontend RF para diferentes bandas ISM donde se han extraído unas pérdidas de conversión tanto en la etapa de down-conversion como up-conversion lo suficientemente bajas como para comprobar el correcto funcionamiento del sistema. A partir de estos resultados se puede realizar una correcta calibración del emulador para un entorno real.

Desafortunadamente, la integración de los elementos que componen el emulador no se pudo realizar en el periodo de este trabajo fin de máster, debido a que no se finalizó la validación de la implementación del procesador banda base, que, por otro lado, es el elemento más complejo del sistema. Por ello, como líneas futuras, queda pendiente finalizar la implementación de dichos elementos, la integración de las diferentes partes del emulador, y la validación del sistema completo.

Referencias

- [1] Z. Fernandez, O. Seijo, M. Mendicute, and I. Val, "Analysis and Evaluation of a Wired/Wireless Hybrid Architecture for Distributed Control Systems With Mobility Requirements," *IEEE Access*, vol. 7, pp. 95915–95931, Jul. 2019.
- [2] V. Díez, A. Arriola, I. Val, and M. Velez, "Validation of RF communication systems for Industry 4.0 through channel modeling and emulation," *Proc. 2017 IEEE Int. Work. Electron. Control. Meas. Signals their Appl. to Mechatronics, ECMSM 2017*, pp. 2–7, 2017.
- [3] E. Egea-Lopez, A. Martinez-Sala, J. Vales-Alonso, J. Garcia-Haro, and J. Malgosa-Sanahuja, "Wireless communications deployment in industry: A review of issues, options and technologies," *Comput. Ind.*, vol. 56, no. 1, pp. 29–53, 2005.
- [4] I. Val, F. Casado, P. M. Rodriguez, and A. Arriola, "FPGA-based wideband channel emulator for evaluation of Wireless Sensor Networks in industrial environments," in *19th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA 2014*, 2014.
- [5] "Versatile Channel Emulator for Advanced Performance Testing Keysight Technologies Prosim F8 Channel Emulator."
- [6] "Spirent Wireless Channel Emulator VR5 HD Spatial Channel Emulator."
- [7] "RFnest™ API RFview 2.11+, D5-series (rev1), and A208 www.i-a-i.com/rfnest."
- [8] "ZedBoard Zynq™ Evaluation and Development Hardware User's Guide," 2012.
- [9] "Wireless communications : principles and practice / Theodore S. Rappaport. - Version details - Trove." [Online]. Available: <https://trove.nla.gov.au/work/16129438?q&versionId=45005968>. [Accessed: 28-Sep-2019].
- [10] S. R. Saunders, "ANTENNAS AND PROPAGATION FOR WIRELESS COMMUNICATION SYSTEMS Second Edition."
- [11] O. Fernández Fernández Directores and M. P. Domingo Gracia Rafael Torres Jiménez, "UNIVERSIDAD DE CANTABRIA DEPARTAMENTO DE INGENIERÍA DE COMUNICACIONES TESIS DOCTORAL Caracterización Experimental y Modelado de Canal MIMO para aplicaciones WLAN y WMAN."
- [12] X. Yin and X. (Researcher in wireless communication) Cheng, *Propagation channel characterization, parameter estimation, and modeling for wireless communications*. .
- [13] M. Cheffena, "Industrial wireless sensor networks: channel modeling and performance evaluation," *EURASIP J. Wirel. Commun. Netw.*, vol. 2012, no. 1, Dec. 2012.
- [14] E. Tanghe, W. Joseph, L. Martens, H. Capoen, K. Van Herwegen, and W. Vantomme, "Large-scale fading in industrial environments at wireless communication frequencies," in *IEEE Antennas and Propagation Society, AP-S International Symposium (Digest)*, 2007, pp. 3001–3004.
- [15] "Antenna-Theory.com - Friis Fórmula Transmisión." [Online]. Available: <http://www.antenna-theory.com/spanish/basics/friis.php>. [Accessed: 29-Sep-2019].
- [16] "Wireless Channels."
- [17] J. G. Proakis, "Digital communications / John G. Proakis,," 2001.
- [18] Matlab, "M-SEQUENCE GENERATION - galois sequences."
- [19] J. M. 1940- Hernando Rábanos, L. Mendo Tomás, and J. M. Riera Salís, *Comunicaciones móviles*. Editorial Universitaria Ramón Areces, 2015.
- [20] L. DE Banda, R. O. Bettancourt, H. E. Carrasco, and R. L. Feick, "MEDICIÓN DE ANCHO DE BANDA DE COHERENCIA Y DISPERSIÓN TEMPORAL EN," 2005.

- [21] M. Mfeze and E. Tonye, "Comparative Approach of Doppler Spectra for Fading Channel Modelling by the Filtered White Gaussian," vol. 6, no. 11, 2015.
- [22] "Xilinx Kintex-7 FPGA KC705 Evaluation Kit." [Online]. Available: <https://www.xilinx.com/products/boards-and-kits/ek-k7-kc705-g.html>. [Accessed: 04-Oct-2019].
- [23] G. Joseba, "ESCUELA POLITÉCNICA SUPERIOR DE MONDRAGON MONDRAGON UNIBERTSITATEKO GOI ESKOLA POLITEKNIKOA Trabajo fin de máster presentado para la obtención del título de Titulua eskuratzeko master bukaerako lana MÁSTER UNIVERSITARIO EN SISTEMAS EMBEBIDOS SISTEMA TXERT," 2019.
- [24] L. Schumacher *et al.*, "IEEE P802.11 Wireless LANs TGN Channel Models," 2004.
- [25] Xilinx and Inc, "Zynq UltraScale+ MPSoC Product Tables and Product Selection Guide," 2016.
- [26]. "the world's most energy friendly microcontrollers EFM32GG Reference Manual Giant Gecko Series."
- [27] "EVAL-ADL5201 Evaluation Board | Analog Devices." [Online]. Available: <https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/eval-adl5201.html#eb-documentation>. [Accessed: 05-Oct-2019].
- [28] "Programmable Attenuators - Software & Documentation Resources | Mini-Circuits." [Online]. Available: <https://ww2.minicircuits.com/softwaredownload/patt.html>. [Accessed: 05-Oct-2019].
- [29] "ADL5202 Datasheet and Product Info | Analog Devices." [Online]. Available: <https://www.analog.com/en/products/adl5202.html#>. [Accessed: 04-Oct-2019].
- [30] "R2018a - Actualizaciones de las familias de productos MATLAB y Simulink - MATLAB & Simulink." [Online]. Available: https://es.mathworks.com/products/new_products/release2018a.html. [Accessed: 04-Oct-2019].
- [31] "Get Qt - Download now." [Online]. Available: <https://www.qt.io/download>. [Accessed: 05-Oct-2019].
- [32] "How to cross compile for ARM." [Online]. Available: <http://www.firmcodes.com/how-to-cross-compile-for-arm/>. [Accessed: 05-Oct-2019].
- [33] "Qt & Qwt Build Instructions (Qt 5.4.2, Qwt 6.1.2) - Xilinx Wiki - Confluence." [Online]. Available: <https://xilinx-wiki.atlassian.net/wiki/spaces/A/pages/18842110/Qt+Qwt+Build+Instructions+Qt+5.4.2+Qwt+6.1.2>. [Accessed: 05-Oct-2019].
- [34] "PetaLinux Tools Documentation Reference Guide," 2017.
- [35] J. G. Hidalgo, "ESCUELA POLITÉCNICA SUPERIOR DE MONDRAGON POLITEKNIKOA Trabajo presentado para la obtención del título de Titulua eskuratzeko lana GRADO EN INGENIERÍA DE SISTEMAS DE TELECOMUNICACIÓN TELEKOMUNIKAZIO SISTEMETAKO INGENIARITZA GRADUA Título del Trabajo Lanar," 2017.
- [36] "Linux RPMMsg framework overview - stm32mpu." [Online]. Available: https://wiki.st.com/stm32mpu/wiki/Linux_RPMMsg_framework_overview. [Accessed: 30-Sep-2019].
- [37] "EVAL-ADF4355 Evaluation Board | Analog Devices." [Online]. Available: <https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/eval-adf4355.html#eb-overview>. [Accessed: 05-Oct-2019].

