



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

IMPLEMENTACIÓN DE PROCESADOR BANDA BASE MIMO

JOAQUÍN ANDRÉS VENEGAS JARA

Tesis para optar al grado de
Magíster en Ciencias de la Ingeniería

Profesor Supervisor:
MARCELO GUARINI

Santiago de Chile, Diciembre 2015

© MMXV, JOAQUÍN ANDRÉS VENEGAS JARA



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

IMPLEMENTACIÓN DE PROCESADOR BANDA BASE MIMO

JOAQUÍN ANDRÉS VENEGAS JARA

Tesis presentada a la Comisión integrada por los profesores:

MARCELO GUARINI

CHRISTIAN OBERLI

PABLO ZEGERS

ÁNGEL ABUSLEME

Para completar las exigencias del grado de
Magíster en Ciencias de la Ingeniería

Santiago de Chile, Diciembre 2015

© MMXV, JOAQUÍN ANDRÉS VENEGAS JARA

*En el fondo, los científicos somos gente con suerte: podemos,
jugar a lo que queramos durante toda la vida.*

LEE SMOLIN

AGRADECIMIENTOS

Quisiera agradecer a mi profesor supervisor Marcelo Guarini y al profesor Christian Oberli por darme la oportunidad de trabajar con ellos durante el transcurso de mi postgrado. Ambos me han alentado y dado las herramientas para poder desarrollarme como profesional.

También, agradezco a mi familia por su cariño y apoyo a la distancia. Igualmente, a las personas que fueron mi segunda familia: Eliana Leiva, Alejandro Venegas, Viviana Jara por acogerme y permitirme ser parte de su hogar durante algunos años.

Además, agradezco a mis compañeros de trabajo durante mi postgrado: Santiago Barros, Jean Paul de Villers-Grandchamps, Joaquín Aldunate, Diego Kaulen y Felipe Kettlun, por sus ánimos y consejos que hicieron más ameno el transcurso del postgrado.

Mi mayor gratitud va hacia mi futura novia Clara Morales por darme su apoyo incondicional durante todo mis años de estudio, especialmente en aquellos momentos difíciles.

Finalmente, al proyecto FONDEF IT13i20015 bajo el cual fue posible desarrollar esta tesis y a CONICYT Chile por el financiamiento de mi postgrado mediante la beca CONICYT-PCHA Magíster Nacional 2014-22141527.

ÍNDICE GENERAL

AGRADECIMIENTOS	IV
ÍNDICE GENERAL	V
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	XI
RESUMEN	XII
ABSTRACT	XIII
1. Introducción	1
1.1 Motivación	1
1.2 Objetivos y Contribuciones principales	3
1.3 Organización del documento	4
2. Plataforma experimental	5
2.1 Estado del arte en plataforma de pruebas MIMO	6
2.2 Plataforma Acuario	7
2.3 Módulos de radio frecuencia	8
3. Comunicaciones inalámbricas	10
3.1 Modulación digital	10
3.1.1 Forma de pulso	11
3.2 Canal inalámbrico	12
3.2.1 Canal MIMO	12
3.3 Sistemas de comunicaciones reales	13
3.3.1 Desplazamiento en frecuencia de la portadora	13
3.3.2 Control automático de ganancia	14
4. Metodología	18
4.1 Flujo de diseño y desarrollo	18

4.2	Aplicación	19
4.3	Simulación	20
5.	Diseño de procesador banda base	22
5.1	Arquitectura a nivel de sistema	23
5.1.1	Máquina de control	25
5.2	Banda-Base	26
5.2.1	Filtros	27
5.2.2	Métodos de implementación	29
6.	Algoritmos	31
6.1	Control automático de ganancia	31
6.1.1	Implementación del AGC	32
6.2	Detección de paquete	34
6.2.1	Implementación de la detección de paquete	36
6.3	Estimación gruesa del desplazamiento en frecuencia de la portadora	37
6.3.1	Implementación del estimador grueso de CFO	38
6.4	Estimación fina del desplazamiento en frecuencia de la portadora	39
6.4.1	Implementación estimador fino de CFO	40
6.5	Estimación de canal	42
6.5.1	Extensión MIMO	42
6.5.2	Implementación del estimador de canal	43
6.6	Estimación de desplazamiento de fase de la portadora	44
6.6.1	Estructura de paquete	46
6.6.2	Máquina de estado	47
6.6.3	Diagrama de bloques a nivel de procesamiento	48
7.	Simulaciones	51
7.0.4	Pérdida de desempeño producto de la estructura de paquete	51
7.1	Estimación de la tasa de error de bit	52
7.1.1	Hardware de emulación de canal discreto	52

7.1.2	Resultados	54
8.	Conclusiones y Trabajo Futuro	57
8.1	Discusión y resultados	57
8.2	Trabajo futuro	59
8.3	Observaciones sobre la plataforma	59
	BIBLIOGRAFÍA	60
	ANEXOS	65
A.	PRIMER ANEXO	66

ÍNDICE DE FIGURAS

1.1.	Tendencias de las distintas tecnologías inalámbricas. Actualmente existe un vacío en las comunicaciones <i>wide area ubiquitous networks</i> (Saito, Kagami, Umehira, y Kado, 2008).	2
1.2.	Diagrama de investigaciones en LATINA UC	2
2.1.	Fotografía de la plataforma Acuario detallando las placas que la componen. . .	8
2.2.	Diagrama de bloques de la rama de recepción.	9
3.1.	Ilustración del efecto del filtro IF y el AGC sobre la potencia de la señal y la escala de la señal. El color gris representa potencia de la señal de interés. La trama de puntos con fondo blanco indica potencia de ruido. La trama de franjas horizontales negras representa el filtro IF ideal y las franjas rojas indican el ancho de banda adicional producto de la implementación real.	16
4.1.	Flujo clásico de diseño y desarrollo (Rupp, Burg, y Beck, 2003).	19
4.2.	Nuevo flujo de diseño y desarrollo aplicado (Rupp et al., 2003).	20
5.1.	Arquitectura del procesador banda-base.	24
5.2.	Diagrama de estados de la máquina de control de módulo banda-base.	25
5.3.	Diagrama de estados de la máquina de control externa del procesador.	26
5.4.	Cadena de filtros de recepción.	27
5.5.	Respuesta en frecuencia de los filtros.	28
5.6.	Esquema de funcionamiento de un procesamiento por entrelazado.	30
6.1.	Diagrama de bloques del control automático de ganancia (AGC).	33
6.2.	Codificación y decodificación diferencial.	35
6.3.	Correlación de un preámbulo BPSK de 32 símbolos transmitido bajo un canal blanco gaussiano aditivo con SNR de 4 dB.	35

6.4.	Diagrama del algoritmo de detección de paquete para 4 antenas receptoras. . .	36
6.5.	(A) Valor promedio del <i>peak</i> a la salida del correlacionador (B) Varianza del valor <i>peak</i> a la salida del correlacionador	37
6.6.	(A) Valor medio del estimador de CFO grueso para un SNR de 4 dB y distintos CFO (B) Varianza del estimador de CFO grueso para distintas SNR	38
6.7.	(A) Valor medio del estimador de CFO fino para un SNR de 4 dB y distintos CFO (B) Varianza del estimador de CFO fino para distintas SNR	40
6.8.	Implementación del estimador fino de CFO para $N = 4$	41
6.9.	Constelación 64-QAM (original y rotada) representada en punto fijo con 8 bits enteros y 7 fraccionarios.	43
6.10.	Implementación del estimador de canal.	44
6.11.	Implementación del estimador de desplazamiento de fase de la portadora. . .	45
6.12.	Estimación del desplazamiento de fase de la portadora a lo largo de un paquete QPSK transmitido en un canal aditivo blanco gaussiano con SNR de 4 dB y un desfase de $\pi/4$	45
6.13.	Pérdida de razón señal ruido dada la cantidad de pilotos consecutivos en el PLL implementado bajo un canal aditivo blanco gaussiano sin desplazamiento en frecuencia de la portadora	46
6.14.	Estructura del paquete.	47
6.15.	Máquina de estados para generar paquete.	48
6.16.	Arquitectura del módulo de procesamiento banda-base.	50
7.1.	Pérdida de SNR para distintos valores de DEP (datos entre pilotos) y números de pilotos consecutivos.	52
7.2.	Segmentación jerárquica aplicada a IGPDF.	54
7.3.	(A) Distribución de probabilidad de 10^7 muestras gaussianas obtenidas de la unidad implementada. (B) Distribución de probabilidad para 10^6 muestras entre 6σ y $7,6\sigma$	55

7.4. Comparación de BER entre el módulo banda-base simulado en MATLAB y el implementado en FPGA. 56

A.1. Recepción de símbolos bajo un canal AWGN con SNR de 4 dB y un desfase de $\pi/4$ (A) Diagrama de dispersión antes de la corrección de fase (B) Diagrama de dispersión después de la corrección de fase. 66

ÍNDICE DE TABLAS

2.1.	Especificaciones técnicas de la plataforma Acuario.	7
5.1.	Requerimientos de funciones del procesador banda-base.	22
5.2.	Tabla resumen de la aritmética de cada filtro en caso de ser implementado en forma paralela.	28
5.3.	Tabla resumen de los largos de palabra utilizados en los filtro implementados (I = <i>input</i> , O = <i>output</i> , C = <i>coefficient</i> , WL = <i>word length</i>).	29
6.1.	Codificación de aumento y disminución de ganancias.	33
7.1.	Tiempo de simulación por paquete de 200 bits transmitido.	55

RESUMEN

Actualmente no existe un desarrollo adecuado o suficiente de la tecnología en el ámbito de las comunicaciones de ancho de banda angosto y largo alcance. Sin embargo, este segmento de las comunicaciones se muestra especialmente adecuado para las redes inalámbricas de sensores que cubren grandes extensiones geográficas. Varios autores han propuesto utilizar algoritmos que emplean múltiples antenas (conocida como MIMO por sus siglas en inglés) para minimizar la potencia total transmitida. No obstante, el desarrollo de las plataformas experimentales que permiten validar las investigaciones han apuntado en la dirección de esquemas de gran ancho de banda como WiMAX o LTE.

Esta tesis propone una metodología para diseñar e implementar un procesador banda-base y comprobar su funcionamiento en un dispositivo programable, en este caso una FPGA, para la investigación en redes inalámbricas de sensores MIMO. El procesador fue diseñado bajo las especificaciones del *testbed* desarrollado en el Laboratorio de Tecnologías Inalámbricas (LATINA) de la Pontificia Universidad Católica de Chile (PUC).

El procesador propuesto incluyó la características mínimas de un transceptor comercial, tales como: registros de configuración, interfaz de comunicación y procesamiento de señales en banda-base. Además, utilizando algoritmos desarrollados por investigadores en LATINA que solucionan aspectos de sincronización mediante el uso múltiples antenas, se diseñó una arquitectura de procesamiento que permitirá la posterior comprobación empírica de técnicas MIMO.

Finalmente, para validar el procesador se desarrolló en FPGA un emulador de canal discreto como medio para realizar simulaciones de Monte Carlo, y así comparar bajo un medio controlado la tasa de error de bit para distintas razones de señal a ruido entre la implementación y las simulaciones realizadas en MATLAB.

Palabras Claves: MIMO, FPGA, PROCESADOR, BANDA-BASE, HDL

ABSTRACT

Nowadays there is no adequate and sufficient technology development in the field of narrow bandwidth and long reach communications. However, this field of the communications is key for low density wireless sensor networks that cover large geographical areas. Several authors have proposed using algorithms that employ multiple antennas (known as MIMO) in order to minimize the total power transmitted. Nonetheless, the development of experimental platforms to validate the use of MIMO have been directed to high bandwidth schemes, such as WiMAX or LTE.

This thesis proposes a methodology to design, implement and test a baseband processor in a programmable device, in this case an FPGA, for research in MIMO wireless sensor networks. The proposed baseband processor was designed under the specifications of the testbed developed at the Wireless Technologies Laboratory (LATINA, by its Spanish acronym) of the Pontificia Universidad Católica de Chile (PUC).

The proposed baseband processor includes the requirements of any commercial transceiver, such as: configuration registers, a standard communication interface and baseband signal processing . In addition, using synchronization algorithms developed by researchers at LATINA, an architecture that allows subsequent empirical verification of MIMO techniques was designed.

Finally, to validate the proposed baseband processor performance a discrete channel emulator was developed as a way to perform Monte Carlo simulations and estimate the bit error rate for different signal-to-noise ratios. The results show that the proposed methodology can be used to implement a baseband processor with a performance comparable to MATLAB simulations.

Keywords: MIMO, FPGA , PROCESSOR, BASEBAND, IMPLEMENTATION, HDL, SYSTEM DESIGN

1. INTRODUCCIÓN

Las redes inalámbricas de sensores (RIS) son un sistema autónomo que permite la transmisión de información a través de una red de dispositivos de bajo consumo llamados nodos hacia un nodo maestro que finalmente envía la información a una base de datos. En el diseño de las RIS se tiende a buscar implementaciones de baja complejidad lo que limita, junto con la energía disponible en las baterías de los nodos, el rango de alcance de las comunicaciones.

Recientemente, se ha propuesto utilizar la tecnología MIMO ¹ como alternativa para poder realizar comunicaciones más eficientes en términos energéticos (Cui, Goldsmith, y Bahai, 2004; Liu, Li, y Chen, 2005; Rosas y Oberli, 2012). Empleando este método sería posible reducir el vacío tecnológico en las llamadas *wide area ubiquitous networks* (ver Figura 1.1), que tienen por objetivo transmitir un pequeño ancho de banda a largas distancias (Saito et al., 2008).

Esta investigación fue desarrollada bajo el proyecto FONDEF IT13i20015: “Redes inalámbricas de sensores: desarrollo de soluciones precompetitivas”, en el Laboratorio de Tecnologías Inalámbricas (LATINA) de la Pontificia Universidad Católica de Chile (PUC).

1.1 Motivación

Actualmente existen varias plataformas experimentales o *testbeds* que permiten realizar pruebas empíricas sobre nuevos algoritmos propuesto en torno a las comunicaciones MIMO. Sin embargo, éstas han sido desarrolladas para el espectro de los 2.4 y 5.1 GHz y gran ancho de banda. Además, la mayoría son utilizadas en conjunto con un computador encargado de procesar los paquetes de la capa física.

En LATINA se han desarrollado investigaciones que solucionan aspectos de sincronización obteniendo ventaja de las comunicaciones de múltiples antenas. Esas investigaciones han sido registradas en los documentos de Feres (2013); Kaulen (2015b); Kettlun

¹Comunicaciones utilizando múltiples antenas. MIMO: *Multiple-input multiple-output*

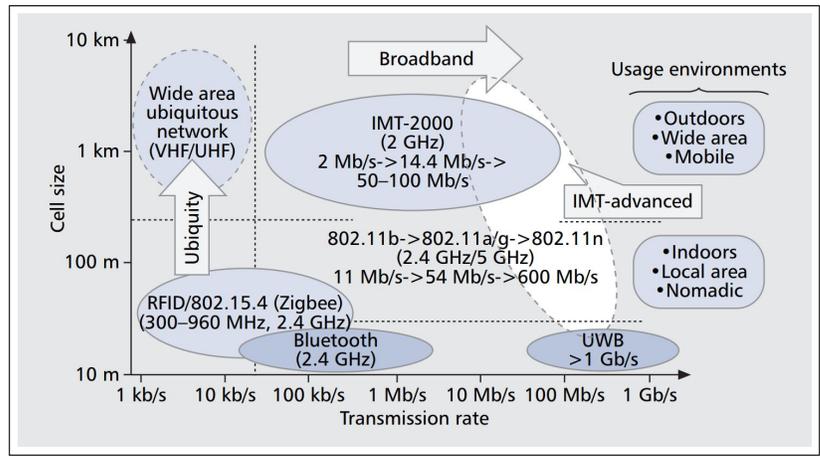


FIGURA 1.1. Tendencias de las distintas tecnologías inalámbricas. Actualmente existe un vacío en las comunicaciones *wide area ubiquitous networks* (Saito et al., 2008).

(2014); Villers-Grandchamps (2014). La Figura 1.2 muestra el resumen y la ubicación de los aportes de cada investigador en un diagrama simplificado de un módulo de procesamiento banda-base de comunicaciones digitales. Los cuadros delineados con azul implican que dichas investigaciones han sido comprobadas a través de simulaciones teóricas. Por otro lado, los cuadros delineados con rojo son aportes que ya se encuentra implementados en FPGA. Este trabajo implica la implementación de los cuadros azules y la integración de todas las investigaciones para ser utilizadas en conjunto dentro de un procesador banda-base.

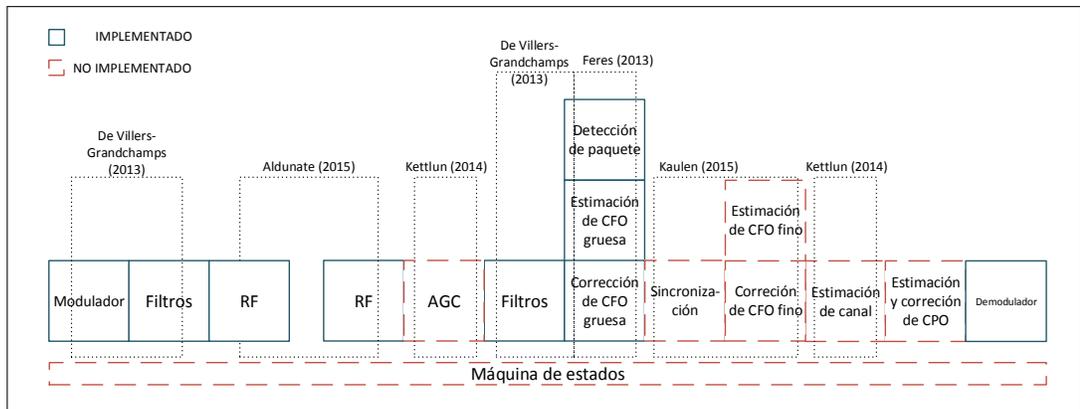


FIGURA 1.2. Diagrama de investigaciones en LATINA UC

Sin embargo, para probar si la implementación de los algoritmos no incurre en pérdidas de desempeño es necesario crear un medio de pruebas que permita realizar mediciones estadísticas sobre parámetros de la recepción. En particular, interesa probar el sistema completo bajo condiciones controladas con el objetivo de tener una medición de la tasas error de bit para distintas razones de señal a ruido.

1.2 Objetivos y Contribuciones principales

La contribución principal de esta tesis es diseñar e implementar un procesador banda-base en FPGA que integre los algoritmos desarrollados por los investigadores de LATINA y que permita obtener estadísticas de variables. Para ello se ha propuesto cumplir los siguientes objetivos:

- Establecer una metodología que permita el desarrollo conjunto entre teoría, simulación e implementación considerando las restricciones dentro de un laboratorio de investigación universitario.
- Diseñar e implementar un procesador que permita aumentar el nivel de abstracción a la hora de realizar pruebas estadísticas de algoritmos MIMO implementados.
- Crear un simulador que integre los algoritmos propuestos en LATINA, considerando las especificaciones de la plataforma, y que simule la transmisión y recepción de datos binarios.
- Implementar los algoritmos en *hardware* considerando los recursos disponibles en la FPGA Virtex 5 LX50T.
- Implementar un emulador de canal que permite generar muestras gaussianas aleatorias.
- Obtener la curva de tasa de error de bit promedio (BER, por sus siglas en inglés) dada una razón señal a ruido (SNR, por sus siglas en inglés).

1.3 Organización del documento

El presente documento se organiza de la siguiente manera: el Capítulo 2 define qué es un *testbed* y muestra las plataformas existentes. Se presentan las especificaciones del *testbed* de LATINA bajo las cuales se diseñará el procesador banda-base. El Capítulo 3 introduce los principales conceptos y desafíos de una esquema real de transmisión y recepción inalámbrico, que deberán ser incluidos en el simulador. Por otro lado, el Capítulo 4 describe la metodología que permitirá unificar las simulaciones realizadas en código y las simulaciones en *hardware* con el objetivo de acelerar el proceso de desarrollo y facilitar el uso de la plataforma para otros investigadores.

El Capítulo 5 establece la arquitectura del procesador y sus partes, tales como: registros de configuración, máquinas de control, modulador, demodulador y procesamiento de banda-base. Luego, el Capítulo 6 detalla los aspectos de implementación que permiten llevar a *hardware* los algoritmos de sincronización MIMO desarrollados en LATINA. El Capítulo 7 establece los requerimientos para poder realizar una medida de desempeño del procesador banda-base implementado. Finalmente, el Capítulo 8 resume las principales contribuciones y conclusiones de la tesis, indicando sugerencias para futuras investigaciones.

2. PLATAFORMA EXPERIMENTAL

Hoy en día realizar un prototipo para sistemas de comunicaciones inalámbricos, entendido en su concepto más amplio, parece impensable debido a los grandes costos en recursos y tiempo para llevar a cabo tal tarea (Rupp et al., 2006). Es por ello que hace algunos años atrás la industria pensó en una forma de re-integrar el prototipado a la cadena de diseño, naciendo de esta forma los llamados *testbeds*. Éstos ofrecen la capacidad de hacer experimentos en tiempo real con hardware muy parecido al del producto final, requiriendo un pequeño grupo de ingenieros.

Los términos: demostradores, *testbed* y prototipo, a menudo son usados indistintamente tanto para referirse a pruebas de conceptos como a implementaciones (Chockalingam y Rajan, 2014), por lo tanto, conviene dar algunas claves para diferenciarlos:

Demostradores: Tienen por objetivo mostrar una nueva tecnología a los consumidores. Implementa una nueva idea, concepto o estándar que ya ha sido establecido o finalizado. Se prefiere la funcionalidad y tiempo de desarrollo por sobre la escalabilidad.

Plataformas o Testbed: Es un conjunto de *hardware* diseñado para validar el funcionamiento de sistemas en escenarios más realistas. Usado principalmente por investigadores para verificar el funcionamiento de nuevos algoritmos. Requiere de escalabilidad, modularidad y extendibilidad.

Prototipo: Realización inicial de un idea de investigación o estándar como referencia para pruebas de concepto.

Aunque tanto los *testbeds* como los prototipos son parte esencial en la fase de investigación y desarrollo, la diferencia fundamental está en que el prototipo necesariamente debe operar en tiempo real y se encuentra más cerca del producto final, mientras que el *testbed* puede procesar los datos en tiempo real o no, siendo esta última la tendencia predominante en los usos de las plataformas ya que pueden existir limitaciones en los recursos

del *testbed*. Por otro lado, los demostradores son más utilizados por las compañías para dar a conocer a los consumidores el potencial de sus nuevas ideas.

Esta investigación se centrará en el desarrollo de un *testbed* para comunicaciones inalámbricas MIMO en redes de sensores. A continuación se presenta el estado del arte en relación a las plataformas disponibles en el mercado o diseñadas por investigadores.

2.1 Estado del arte en plataforma de pruebas MIMO

Son pocos los autores que han explorado el estado actual de los *testbeds* existentes en el mundo. Por un lado, Rao et al. (2004) describe las capacidades académicas tanto en educación como investigación que pueden tener los *testbeds* y los clasifica desde la perspectiva del usuario bajo los conceptos de flexibilidad, tiempo de desarrollo, eficiencia y costos. Por ejemplo, los dispositivos de radio definidos por *software* serían una plataforma muy flexible en uso, pero limitada en procesamiento. Por otro lado, un sistema basado en FPGA permite una mayor personalización del sistema y capacidad de procesamiento. Sin embargo, el tiempo de desarrollo y los costos asociados aumentarían.

Por otro lado, García-Naya, González-López, y Castedo (2008) presenta un análisis del *hardware* necesario para construir las principales partes de un *testbed*, por ejemplo: componentes de *bandabase*, *front-end* RF y *software* de desarrollo. Además, crea una tabla resumen de las características técnicas de las plataformas existentes hasta el 2008. Sin embargo, hoy en día es más común encontrar en investigaciones el uso de *hardware* o equipos comprados a empresas como: National Instruments, Pentek, Sundance, Nallatech, ICS y Nutaq, en vez de *hardware* desarrollado por las propias universidades. Las herramientas de desarrollos incluyen Matlab de Mathworks, Code Composer de Texas Instruments e ISE Design Suite de Xilinx.

Cabe destacar que la mayoría de las investigaciones revisadas utilizan los *testbed* para obtener datos y luego procesarlos en un computador, dejando en duda la capacidad de la plataforma para implementar algoritmos en tiempo real. Además, debido al continuo interés de la industria en comunicaciones de gran ancho de banda no se han diseñado o

presentado plataformas para comunicaciones de ancho de banda angosto que es el caso de interés en las redes de sensores. Por lo tanto, en LATINA se decidió diseñar módulos propios de RF para comunicaciones MIMO y elaborar una plataforma propia.

2.2 Plataforma Acuario

Acuario es el *testbed* desarrollado en LATINA para la investigación en redes inalámbricas de sensores MIMO. Las características técnicas se resumen la Tabla 2.1.

TABLA 2.1. Especificaciones técnicas de la plataforma Acuario.

Parámetro	Valor
Resolución ADC-DAC	8 bits
Modulaciones	QPSK , 16-QAM. 64-QAM
Ancho de banda de datos	13.24 kHz
Factor de sobre-muestreo bandabase	4
Factor de sobre-muestreo IF digital	512
Ancho de banda interior	19.75 kHz
Frecuencia intermedia	6.75 MHz
Frecuencia de la portadora	915-930 MHz
Frecuencia Oscilador Local	27 MHz
ppm Oscilador Local	+/- 10

La Figura 2.1 muestra la plataforma Acuario con el detalle de las placas que la componen. A continuación se detallarán las funciones de las placas dentro de la plataforma:

Cangrejo: Placa desarrollada en LATINA. Posee los módulos transceptores de radio frecuencia que operan en la banda de 915-930 MHz. Además, entrega directamente a la placa Digilent Genesys las muestras obtenidas por ADC que cuantiza la señal recibida en 8 bits a una tasa de 27 MHz.

Nexys 3: Tarjeta de desarrollo creada por Digilent Inc. la cual posee una FPGA Spartan-6. Se utiliza esta placa para configurar por medio del protocolo SPI los circuitos integrado en la placa Cangrejo.

Genesys: Tarjeta de desarrollo creada por Digilent Inc. la cual posee una FPGA Virtex 5 LX50T. Se utiliza esta placa para el procesamiento digital de las señales

en banda base. Además, permite el descargar datos hacia el computador mediante un módulo ethernet.

Placa Calamar : Placa desarrollada en LATINA. Distribuye el reloj de sistema de 27 MHz hacia los transceptores y hacia la placa Genesys.

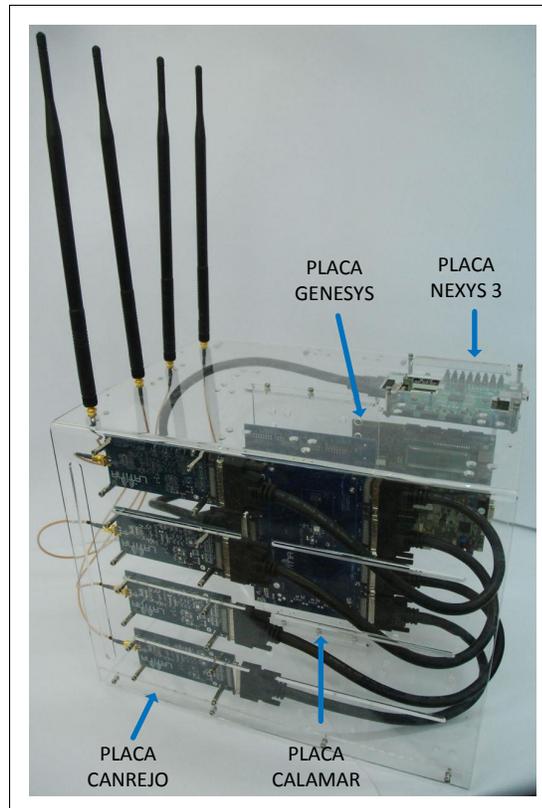


FIGURA 2.1. Fotografía de la plataforma Acuario detallando las placas que la componen.

2.3 Módulos de radio frecuencia

En esta sección se describen los bloques más importantes que componen la rama de recepción (Ver Figura 2.2) de los módulos de radio frecuencia (RF) con el objetivo de dar al lector nociones de los bloques para poder entender las siguientes secciones de la tesis.

El módulo RF en modo de recepción, recibe la señal entrante hacia un amplificador de bajo ruido (LNA, por sus siglas en inglés). La señal resultante es filtrada por un filtro pasa-banda, atenuando cualquier señal fuera de la banda ISM 902-928 MHz. La señal filtrada es multiplicada por una senoide de 600 MHz y posteriormente filtrada para eliminar el lado inferior de la banda. A continuación, la señal se multiplica por una senoide de 321.75 MHz para luego filtrar su banda superior. Finalmente, la señal antes de llegar al ADC es acondicionada por un filtro anti-aliasión pasa-bajos y por un amplificador de ganancia variable (Aldunate, 2013).

En el lado digital es común especificar la frecuencia intermedia como 1/4 de la tasa de muestreo con el objetivo de evitar efectos de fase truncada en el oscilador por control numérico (NCO, por su siglas en inglés). El NCO es implementado por un contador que elimina, mantiene o niega una muestra. Cabe destacar que hace falta el retardo de una muestra para que ocurra un desfase de 90° . Luego, a la salida de la demodulación digital el control automático de ganancia (AGC, por sus siglas en inglés) estima el nuevo valor de ganancia y lo envía al amplificador para que la señal digital no sature.

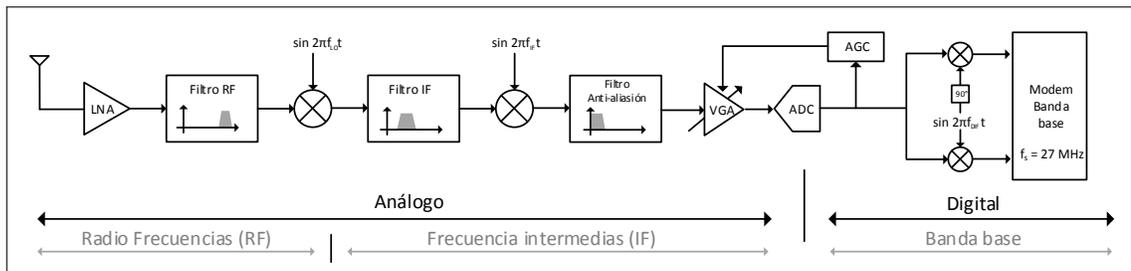


FIGURA 2.2. Diagrama de bloques de la rama de recepción.

3. COMUNICACIONES INALÁMBRICAS

Para diseñar un modem banda base capaz superar las dificultades que presenta el canal inalámbrico, primero se debe establecer el modelo matemático y los supuestos sobre los cuales se debe diseñar.

3.1 Modulación digital

El proceso de modulación consiste en representar la información a transmitir, en este caso bits, en una señal adecuada para ser propagada por el canal físico (Cherubini y Benvenuto, 2003). El dispositivo que realiza el mapeo de una secuencia binaria a una señal de amplitud $a(t)$, frecuencia de la portadora f_c y fase $\phi(t)$ se denomina “modulador”. La señal modulada puede ser descrita por la siguiente ecuación:

$$s(t) = a(t)\sqrt{2} \cos [2\pi f_c t + \phi(t)] \quad (3.1)$$

que puede reescribirse en términos de su componentes en fase y cuadratura

$$s(t) = \sqrt{2}s_I \cos [2\pi f_c t] - \sqrt{2}s_Q \sin [2\pi f_c t] \quad (3.2)$$

donde $s_I = a(t) \cos [\phi(t)]$ y $s_Q = a(t) \sin [\phi(t)]$ son llamados las componentes en fase y cuadratura de $s(t)$ respectivamente.

La representación compleja en bandabase de $s(t)$ suele ser útil en receptores que procesan la señal por su componente en fase y cuadratura, como es el caso de esta investigación, por lo tanto, conviene definirla:

$$s(t) = \sqrt{2}\Re \{ \tilde{s}(t)e^{j2\pi f_c t} \} \quad (3.3)$$

donde $\tilde{s}(t) = s_I(t) + js_Q(t)$ es la envolvente compleja de $s(t)$.

En modulaciones lineales tales como las M-arias, la información binaria es codificada en amplitud y fase sobre el intervalo T .

$$\tilde{s}(t) = s_I(t) + js_Q(t) \quad (3.4)$$

$$= a_I + ja_I \quad 0 \leq t \leq T \quad (3.5)$$

$$= \tilde{a} \quad 0 \leq t \leq T \quad (3.6)$$

donde \tilde{a} es un número complejo denominado “símbolo” perteneciente a la constelación M-aria de M símbolos donde cada símbolo es la codificación de $\log_2(M)$ bits. El mapeo y codificación a utilizar está detallado en el estándar 802.11a (WiFi) .

3.1.1 Forma de pulso

Con el objetivo de limitar el ancho de banda de la transmisión y garantizar un decaimiento rápido para aquellas frecuencias que no están en la banda de interés se utilizará un filtro de forma de pulso para evitar la interferencia intersimbólica (ISI, por sus siglas en inglés). Por ello, la respuesta en el tiempo de dicho filtro debe ser cero para múltiplos enteros de la duración de un tiempo de símbolo.

Una de las respuestas de filtros más utilizadas es la Raíz de Coseno Elevado (RRC, por sus siglas en inglés). Posee un parámetro de diseño llamado “ factor de roll-off ”, β , que determina el exceso de ancho de banda y la tasa de decaimiento de la respuesta en frecuencia del filtro. El uso de este filtro maximiza la razón señal a ruido (SNR, por sus siglas en inglés) y minimiza la ISI de la señal recibida.

La señal transmitida se convierte en la Ecuación (3.7)

$$\tilde{s}(t) = \sqrt{E_s} \sum_k \tilde{a}_k g(t - kT) \quad (3.7)$$

donde $g(t)$ es la forma de pulso con energía unitaria y \tilde{a}_k depende de la modulación y posee energía promedio unitaria.

3.2 Canal inalámbrico

La mayoría de los canales de radio pueden ser caracterizados por una propagación multitrayectoria donde un gran número de rayos afectados por *scattering*¹ llegan al receptor. Cada rayo puede ser caracterizado por una atenuación o ganancia de amplitud, un cambio de fase y un retardo. La ganancia de amplitud, variante en el tiempo, es modelada generalmente mediante una distribución Rayleigh y la velocidad a la cual fluctúa depende del tiempo de coherencia del canal.

En este trabajo se asumirá que el ancho de banda de coherencia de canal es mucho más grande de que ancho de banda de la señal transmitida. Este supuesto implicará que la transmisión se enfrentará a un canal con desvanecimiento plano en frecuencia durante la transmisión de un paquete, por consiguiente, el canal puede ser modelado por un número complejo \tilde{h} que distorsiona en fase y magnitud a la señal de manera constante a lo largo de varios paquetes. La señal a la entrada del receptor, $\tilde{r}(t)$, puede ser modelada por

$$\tilde{r}(t) = \tilde{h} \tilde{s}(t) + \tilde{n}(t) \quad (3.8)$$

donde $\tilde{n}(t)$ es la representación compleja en bandabase del ruido aditivo blanco gaussiano (AWGN, por sus siglas en inglés), dentro del ancho de banda B de la señal. Posee media cero y varianza $N_0 B$, donde $\frac{N_0}{2}$ es la densidad espectral de potencia del ruido. Este ruido es generado por la agitación de los electrones en el conductor (Goldsmith, 2005).

3.2.1 Canal MIMO

Las comunicaciones de múltiples antenas transmisoras o receptoras (MIMO, por sus siglas en inglés) son utilizadas para mejorar el desempeño de un sistema de comunicaciones bajo un canal inalámbrico. En el presente trabajo, un sistema MIMO de N_t antenas transmisoras y N_r antenas receptoras puede ser modelado por

¹El *scattering* ocurre cuando los canales de radio contienen objetos cuyas dimensiones están más o menos en el orden de la longitud de onda de la onda de propagación. El *scattering*, el cual sigue los mismos principios físicos de la difracción, hace que la energía desde el transmisor, se propague en múltiples direcciones (Andersen, Rappaport, y Yoshida, 1995).

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \quad (3.9)$$

donde el vector columna $\mathbf{x} \in \mathbb{C}^{N_t}$ representa los símbolos transmitidos, $\mathbf{y} \in \mathbb{C}^{N_r}$ es el vector columna de los símbolos recibidos, $\mathbf{n} \in \mathbb{C}^{N_r}$ es el vector columna de ruido blanco gaussiano i.i.d con media cero y varianza N_0 , y $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$ es la matriz de canal Rayleigh donde sus componente $h_{i,j}$ representan las ganancias complejas desde la antena transmisora j hasta la antena receptora i . Los elementos $h_{i,j}$ se asumen que son variables gaussianas complejas circularmente simétricas i.i.d con media cero y varianza unitaria.

3.3 Sistemas de comunicaciones reales

Hasta ahora se han presentado modelos y supuestos que describen el comportamiento de una comunicación inalámbrica estándar. Sin embargo, aquellas no-idealidades de las transmisiones que no se han descrito aún, pueden implicar un deterioro significativo del sistema cuando esté implementado.

Por otra parte, se debe considerar el módulo de RF utilizado y los efectos que podría tener su diseño sobre la señal recibida en banda-base. Un típico receptor heterodino posee una etapa que amplifica la señal recibida tratando de no aumentar el ruido. Luego, la señal es filtrada y convertida a una frecuencia intermedia (IF, por sus siglas en inglés). La señal resultante es tratada por un filtro pasabanda de IF idealmente con un ancho de banda igual al ancho de banda de interés de la señal. Finalmente, un convertidor análogo-digital (ADC) cuantifica la señal resultante para su posterior procesamiento (Razavi, 1998).

3.3.1 Desplazamiento en frecuencia de la portadora

En el receptor, el equivalente en banda base de una transmisión se obtiene multiplicando la señal por una referencia local de la frecuencia de la portadora (f_r). En el caso ideal, al llevar la señal a banda base se debería eliminar la portadora (f_c), pero en la realidad la señal oscila debido a la diferencia entre la portadora del transmisor y receptor. Este problema se denomina desplazamiento en frecuencia de la portadora (CFO, por sus siglas en inglés). Entonces, se redefine la señal recibida $r(t)$ como

$$r(t) = \sqrt{2}\Re \{ \tilde{r}(t)e^{j2\pi f_r t} \} \quad (3.10)$$

La razón por la cual no es posible eliminar totalmente la portadora de la señal entrante es porque las frecuencia de los osciladores del transmisor y receptor no son totalmente iguales, sino que existe un pequeña diferencia, $\Delta f = f_c - f_r$. La variación máxima que puede experimentar un oscilador con respecto a su valor nominal se mide en partes por millón o ppm. Además, a dicha variación se le debe agregar el desplazamiento adicional en frecuencia producto de que el transmisor y/o el receptor se encuentren en movimiento por lo cual la señal recibida se ve afectada por el efecto Doppler.

El equivalente en banda base de la señal recibida se transforma en

$$\tilde{r}(t) = e^{j2\pi(\Delta f)t} \tilde{h} \tilde{s}(t) + \tilde{n}(t) \quad (3.11)$$

3.3.2 Control automático de ganancia

En un receptor inalámbrico la potencia promedio de la señal recibida no siempre es la misma. Por un lado, si la señal recibida supera el rango de entrada del ADC se obtendrá una señal saturada, distorsionándola. Por otro lado, si la señal tiene una magnitud pequeña la señal digitalizada se encontrará distorsionada debido a la falta de resolución (Middleton, 2007). Por ello, se debe agregar un control automático de ganancia (AGC, por sus siglas en inglés) que permita mantener la señal a la entrada del ADC en un rango de voltaje apropiado. En otras palabras, el AGC asegura que la representación digital de la señal muestreada posea suficiente precisión digital para ser usada por algoritmos de estimación de parámetros.

El filtro de IF en la cadena de recepción (ver Figura 2.2) permite rechazar la potencia de ruido externa al ancho de banda transmitido, logrando un mejor rendimiento del sistema. Sin embargo, el ruido térmico, la función de transferencia del filtro y la acción del AGC pueden tener un efecto significativo en los algoritmos de estimación, como es el

caso de los algoritmos de adquisición basados en umbrales (Noneaker, Raghavan, y Baum, 2001). Por ende, conviene modelar el efecto del AGC.

Sea $\psi(t)$ el filtro IF,

$$\psi(t) = \sqrt{2} \Re \left\{ \tilde{\psi}(t) e^{j2\pi f_r t} \right\} \quad (3.12)$$

La señal a la salida del AGC es $\hat{r}(t)$,

$$\hat{r}(t) = \sqrt{\alpha(t)} (r * \psi)(t) \quad (3.13)$$

donde $1/\alpha(t)$ es la potencia promedio estacionaria a la entrada del sistema de AGC en el tiempo t y $*$ es el operador convolución.

Calculando la potencia promedio de la señal a la entrada del AGC se obtiene la Ecuación (3.14), donde $G(f)$ y $\Psi(f)$ son las transformadas de Fourier de la forma de pulso $g(t)$ y del filtro $\psi(t)$, respectivamente.

$$E \left[\frac{1}{T} \int_{t=kT}^{(k+1)T} |(r * \psi)(t)|^2 dt \right] \approx \frac{E_s}{T} |h|^2 \int_0^{+\infty} |(G * \tilde{\Psi})(f)|^2 df + \int_{-\infty}^{+\infty} N_0 |\tilde{\Psi}(f)|^2 df \quad (3.14)$$

Se utilizará los parámetros adimensionales γ_s y γ_n definidos por Noneaker et al. (2001) para caracterizar el comportamiento de filtro IF y AGC sobre la señal.

La fracción de la potencia de la señal que pasa a través del filtro IF está dada por γ_s

$$\gamma_s = \frac{\frac{E_s}{T} |h|^2 \int_0^{+\infty} |(G * \tilde{\Psi})(f)|^2 df}{E_s |h|^2} = \int_0^{+\infty} \left| \frac{1}{T} (G * \tilde{\Psi})(f) \right|^2 df \quad (3.15)$$

El parámetro γ_n representa la razón entre la potencia de ruido que pasa a través del filtro IF y la potencia de ruido que pasaría a través de un filtro ideal de ganancia unitaria.

$$\gamma_n = \frac{\int_0^{+\infty} \left| \frac{1}{T} (G * \tilde{\Psi})(f) \right|^2 df}{N_0 \frac{1}{T}} = T \int_{-\infty}^{+\infty} N_0 |\tilde{\Psi}(f)|^2 df \quad (3.16)$$

En el estado estacionario la potencia promedio a la entrada del AGC durante la recepción puede ser expresada como:

$$E \left[\frac{1}{T} \int_{t=kT}^{(k+1)T} [(r * \psi)(t)]^2 dt \right] = \gamma_s \left(E_s + \frac{\gamma_n N_0}{\gamma_s T} \right) \quad (3.17)$$

Para poder explicar el efecto del parámetro γ_n sobre la señal recibida se usará la ilustración de la Figura 3.1. El tamaño de cada rectángulo representa la potencia de la señal. A la entrada del módulo RF la potencia recibida está compuesta por la señal de interés y por el ruido en todas las demás frecuencias. El filtro IF ideal (franja horizontal) elimina el ruido que no se encuentra en el rango de frecuencia de interés. Sin embargo, un filtro IF real tiene un ancho de banda mayor (franja horizontal roja) permitiendo que a la entrada del AGC se observe una mayor potencia de ruido. Luego, el sistema de AGC atenúa y amplifica la señal para que el ADC no sature.

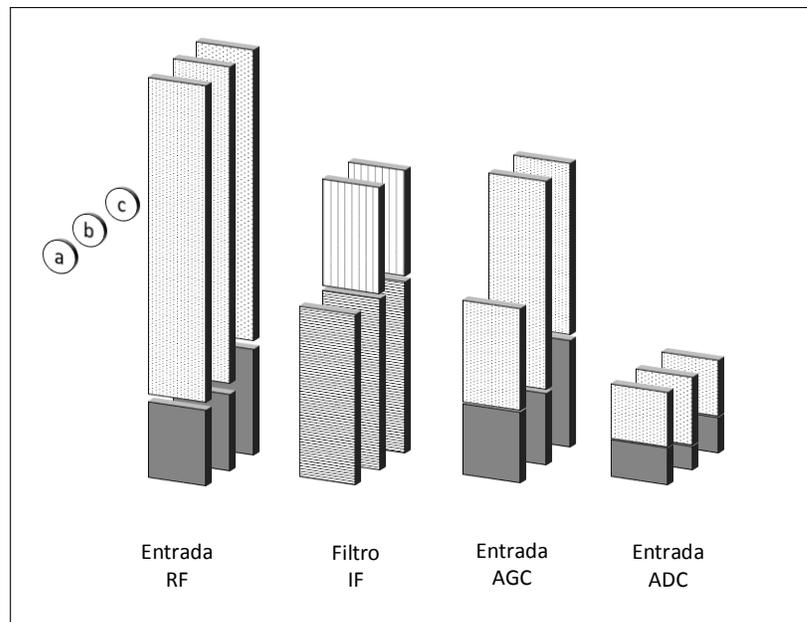


FIGURA 3.1. Ilustración del efecto del filtro IF y el AGC sobre la potencia de la señal y la escala de la señal. El color gris representa potencia de la señal de interés. La trama de puntos con fondo blanco indica potencia de ruido. La trama de franjas horizontales negras representa el filtro IF ideal y las franjas rojas indican el ancho de banda adicional producto de la implementación real.

Finalmente, una forma de simular el comportamiento del AGC está dada por la función $\alpha(t)$ presentada en (Noneaker et al., 2001), donde $t = 0$ es el instante en que la señal transmitida llega al receptor .

$$\alpha(t) = \begin{cases} \gamma_s^{-1} \left(\frac{\gamma_n N_0}{\gamma_s T} \right)^{-1} & t < 0 \\ \gamma_s^{-1} \left(E_s + \frac{\gamma_n N_0}{\gamma_s T} \right)^{-1} & t \geq 0 \end{cases} \quad (3.18)$$

Si se observa la imagen (a) y (b) de la Figura 3.1 se puede concluir que debido al ancho de banda adicional del filtro IF a la entrada del ADC se tiene una señal con una SNR mucho menor. Por otro lado, si se compara la imagen (b) y (c) se puede apreciar el efecto aislado del AGC. Dado que el AGC intenta ajustar la potencia promedio a la entrada del ADC, éste tendrá como efecto la disminución de la amplitud de la señal de interés a la salida del filtro adaptado.

4. METODOLOGÍA

4.1 Flujo de diseño y desarrollo

Diseñar un modem para comunicaciones inalámbricas típicamente requiere tres grupos de trabajo (Rupp et al., 2003):

Equipo de investigación: Trabajan sobre nuevas ideas tecnológicas para incluir nuevas características o mejorar las existentes. Realizan simulaciones, por ejemplo en MATLAB, solo modelando aquellas partes que son esenciales para la transmisión de datos. Típicamente solo los resultados son reportados en publicaciones o reportes técnicos internos.

Equipo de diseño de sistema: Proporcionan la especificaciones técnicas y la arquitectura del sistema. Deciden qué recursos (*hardware* o *software*) se utilizarán para mapear cada algoritmo.

Equipo de implementación: Equipo con conocimientos en ASICs, DSPs y/o FPGAs que recrea el ambiente de precisión de punto flotante del equipo de investigación y los transforma en uno de punto fijo.

Dentro de un grupo de investigación es inevitable que los integrantes se encuentren en más de uno de los tres equipos de trabajo. Además, al contrario de empresas, en las universidades la rotación de alumnos/investigadores es mucho más rápida y problemas como la falta de documentación por parte del equipo de investigación y la incompatibilidad entre herramientas de trabajo entre grupos son más recurrentes.

El flujo clásico o intuitivo de diseño y desarrollo posee una estructura intrínsecamente “hacia adelante”(ver Figura 4.1). La responsabilidad del desarrollo se va heredando en una sola dirección y las instancias de discusión ocurren al final del trabajo, generalmente cuando suceden problemas inesperados obligando la discusión entre los tres grupos de trabajo (Rupp et al., 2003). Finalmente, lo que hace innecesariamente más lento este tipo de flujo es la falta de compatibilidad o encuentro común entre los equipos, y cada vez que surge un problema, entre el modelo y la implementación, toma un tiempo no despreciable

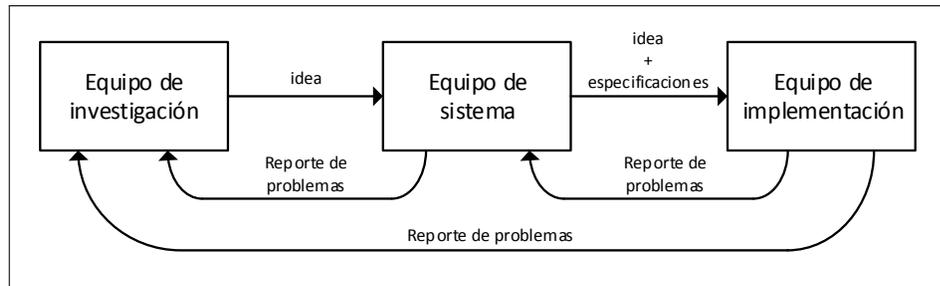


FIGURA 4.1. Flujo clásico de diseño y desarrollo (Rupp et al., 2003).

que uno de los grupo alcance el nivel de conocimiento del otro para poder entender y solucionar dicho problema.

Rupp et al. (2003) propone una nueva metodología basada en cinco nuevos enfoques, los cuales pueden ser utilizados para realizar un prototipado más rápido y eficiente:

- Un solo ambiente de trabajo.
- Una documentación por especificación.
- Una herramienta de revisión de compatibilidad.
- Un código para ser trabajado en mejoras.
- Un equipo.

Sin embargo, los puntos que señala Rupp et al. (2003) deben ser modificados y llevados a la realidad de un laboratorio de investigación universitario considerando la alta rotación de personas. La Figura 4.2 representa la metodología de Rupp aplicada al trabajo de esta investigación.

4.2 Aplicación

La herramienta predilecta en el ámbito académico y de procesamiento de datos es MATLAB, por lo tanto, se puede asumir que será la usada por el equipo de investigación. Si bien no existe tanta variedad de herramientas que ayuden en la tarea del equipo de diseño de sistema, Simulink de MATLAB es una herramienta gráfica que permite abstraer al

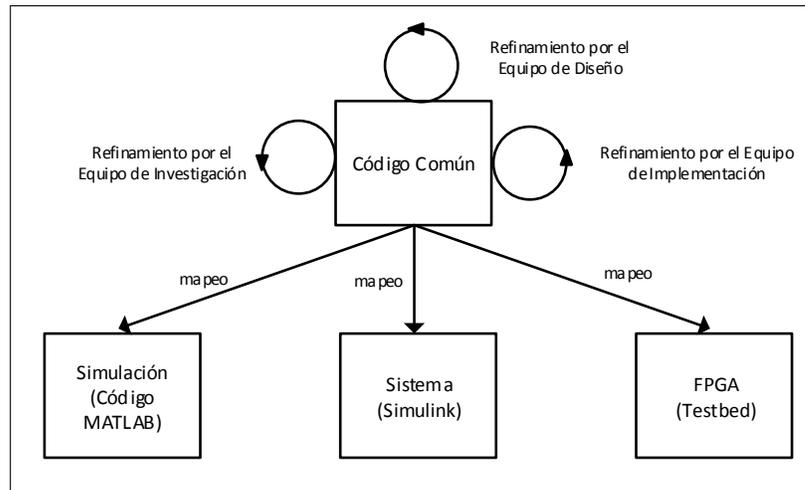


FIGURA 4.2. Nuevo flujo de diseño y desarrollo aplicado (Rupp et al., 2003).

usuario y acercar el código a la implementación. Finalmente, debido a las dos herramientas anteriores se decidió utilizar System Generator de Xilinx para implementar el diseño en FPGA.

Para poder aplicar la metodología anterior primero se debe elegir un archivo, documento o código que pueda ser usado por cualquier equipo de trabajo. Además, debe tener la posibilidad de llevar un sistema de control de versión y revisión. El documento que se eligió fue un archivo de función de MATLAB, el cual crea una estructura de datos que posee los parámetros del sistema. Algunos de los parámetros que contiene son: tasas de muestreo, especificación de punto fijo, coeficientes de los filtros, preámbulo, estructura del paquete de capa física, parámetros de algoritmos de estimación.

4.3 Simulación

El proceso de diseño fue realizado mediante 3 etapas de simulación: alto nivel, bajo nivel y hardware. Las simulaciones de alto nivel se realizaron mediante MATLAB en punto flotante y utilizando la ventaja de operaciones vectoriales para procesar los datos. Mediante iteración se obtienen los parámetros para los diferentes algoritmos tal que se logra un desempeño determinado. La siguiente etapa corresponde a una simulación de bajo nivel. MATLAB, Simulink y System Generator permiten realizar simulaciones con

“precisión de bit” que permiten determinar la cantidad de bits con las cuales se realizan las operaciones y determinar si ocurre una pérdida de rendimiento, y con ello iterar en conjunto con la simulación de alto nivel para ajustar parámetros. Finalmente, el diseño se implementa en un lenguaje de descripción de *hardware* (HDL, por sus siglas en inglés), en este caso, Verilog. Sin embargo, realizar simulaciones para obtener estadísticas de un parámetro a nivel de HDL es impensable debido a los tiempos necesarios para simular. Por lo tanto, se establecerá en capítulos posteriores la forma de realizar simulaciones en hardware.

5. DISEÑO DE PROCESADOR BANDA BASE

Un procesador banda base es un módulo capaz de transmitir y recibir información binaria dada por un usuario. Sin considerar el proceso de acondicionamiento de señal para que sea correctamente transmitida y recibida, el procesador debe poseer una inteligencia mínima que le permita recibir e interpretar comandos o sentencias de un usuario, entre ellas: modificar parámetros, recibir y transmitir información de estado, ejecutar acciones determinadas como por ejemplo auto-reiniciarse, encender o apagar módulos RF, borrar memorias, etc.

Pérez (2012), basándose en la arquitectura del integrado CC2420 (Instruments, 2006), presenta los módulos básicos para implementar un transceptor, como por ejemplo: memoria, interfaz de comunicación, máquina de control, decodificador, etc. Sin embargo, Pérez sacrificó modularidad por eficiencia en recursos de implementación. Por ello, se ha propuesto una nueva arquitectura que sea fácil de entender, modificar y utilizar por futuros investigadores.

TABLA 5.1. Requerimientos de funciones del procesador banda-base.

Configuración	Debe permitir la configuración remota de parámetros del sistema
Recursos	Las implementaciones de algoritmos deben ocupar la menor cantidad de recursos con el objetivo que todo el sistema entre en una sola FPGA
Capacidad de procesamiento	Debe tener la posibilidad de procesar los datos en tiempo real y/o <i>offline</i>
Diseño modular	Capacidad para seguir extendiéndose
Experimentos	Agregar la inteligencia necesaria para poder obtener información estadística del parámetros

5.1 Arquitectura a nivel de sistema

La Figura 5.1 muestra un diagrama de bloques del procesador banda-base desarrollado. A continuación se describe la funcionalidad de cada bloque. Es importante destacar que los módulos se comunican entre sí a través de una señal habilitadora y otra de datos.

Interfaz de comunicación: Módulo pensado para enviar o recibir información en paquete de bytes. En el caso de esta tesis se utilizó el integrado Marvell Alaska Tri-mode PHY 88E1111 para implementar la capa física de Ethernet.

Registros: Guardan parámetros que configuran el módulo banda base y los módulos RF.

FIFO: es una memoria *First in, first out* encargada de guardar los bits de información que se desean enviar o que fueron recibidos. Además, permite la interacción entre dos módulos que funcionan a distintas tasas, como es caso entre los datos provenientes de la interfaz digital y el modulador.

Modulador y Demodulador: Convierte datos binarios a una representación en fase y cuadratura de 8 bits , y viceversa.

Dispatcher: Se encarga de decodificar la instrucción enviada y dirigir la información desde o hacia la máquina de control, la memoria FIFO o registros.

Banda-base: Empaqueta los datos del modulador y los procesa para llevarlos a la tasa de IF digital a la cual funcionan los módulos RF. También, actúa de receptor y es capaz de estimar parámetros de sincronización para alinearse a la portadora del transmisor y detectar si existe un paquete en el aire.

Control: Es una máquina de estado que interpreta las instrucciones enviadas por el usuario y las ejecuta habilitando o deshabilitando señales conectadas a otros módulos.

Logger: Permite tomar datos de cualquier parte del sistema y reportarlos mediante la interfaz de comunicación. Cambia según el tipo de interfaz de comunicación. En el caso de comunicación vía Ethernet aprovecha el sistema de paquete para observar varias señales al mismo tiempo.

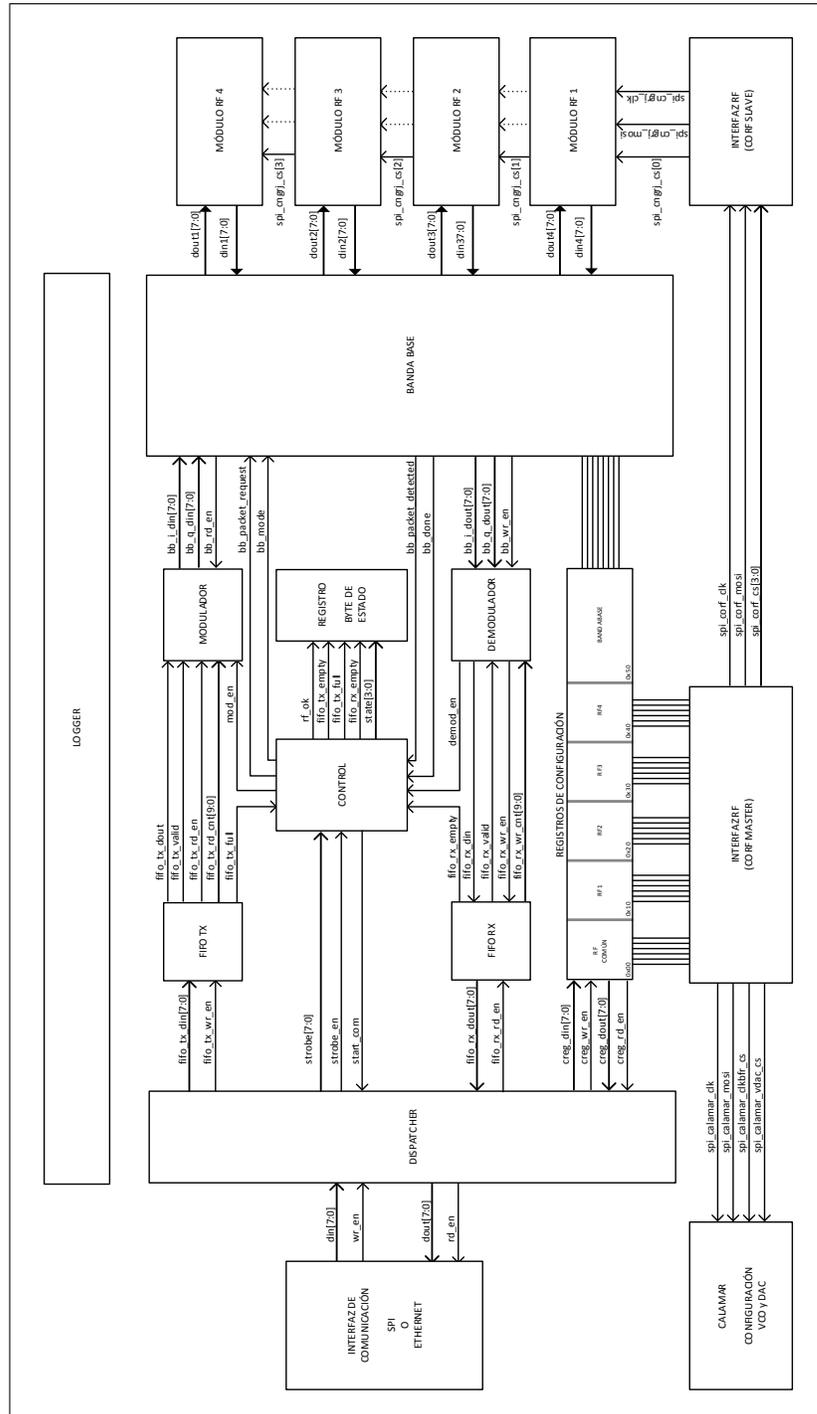


FIGURA 5.1. Arquitectura del procesador banda-base.

5.1.1 Máquina de control

Dada la necesidad de que el procesador banda-base pueda seguir desarrollándose en forma paralela mediante módulos independientes, conviene que el módulo de control esté compuesto por varias máquina de estados y no solo una. La ventaja de esta decisión se puede observar al querer obtener información del estado del procesador sin interrumpir, por ejemplo, la transmisión de un paquete.

La Figura 5.2 muestra la máquina de control que interactúa con el módulo banda-base. Recibe dos tipos de comando: TX_INIT y RX_INIT, los cuales inician la transmisión o recepción respectivamente. Los cambios de estados ocurren por variaciones en las señales provenientes del módulo banda-base.

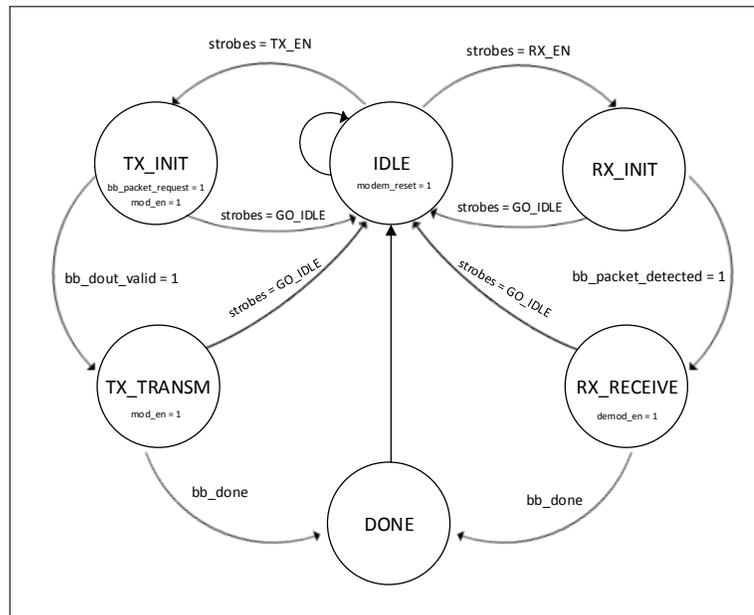


FIGURA 5.2. Diagrama de estados de la máquina de control de módulo banda-base.

La Figura 5.3 muestra la máquina principal del procesador. La idea es mantener esta máquina de estado lo más sencilla posible para poder agregar nuevos estados sin dificultad al interior del código Verilog. Además, se ha propuesto que cada estado debe recibir la respuesta del módulo que debe modificar. Por ejemplo, si se desea encender o apagar

los módulos RF, una vez que el usuario ha enviado el comando, la máquina de control espera una señal proveniente del módulo RF antes de volver al estado IDLE. Esta práctica evita que cualquier retardo de procesamiento al interior de un módulo afecte al sistema completo.

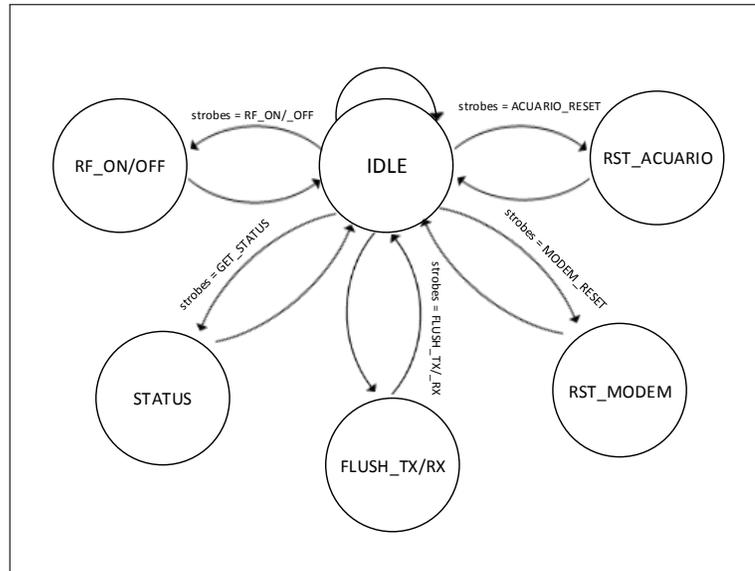


FIGURA 5.3. Diagrama de estados de la máquina de control externa del procesador.

5.2 Banda-Base

El módulo de banda-base es el encargado de realizar la transmisión de los datos en la FIFO y la detección de símbolos de información para poder ser posteriormente demodulados. Los datos binarios provenientes de la FIFO son procesados por el modulador a una tasa 2048 veces menor que la tasa de IF digital, denominada tasa de símbolo. Para lograr transmitir de manera eficiente se utiliza una cadena de filtros que incluye un filtro de forma de pulso (Sección 3.1.1) y filtros de interpolación para aumentar la tasa de muestreo. Por otro lado, la cadena de recepción debe disminuir la tasa de muestreo del ADC a la tasa de símbolo original y detectar los símbolos enviados maximizando la SNR. Además, el receptor debe ser capaz de estimar parámetros de sincronización para detectar de mejor manera la información transmitida en forma de símbolos.

5.2.1 Filtros

Villers-Grandchamps (2014) plantea una implementación eficiente en recursos para la cadena de filtros de interpolación y decimación. Mediante la utilización de filtros CIC es posible sobremuestrear o submuestrear utilizando solo sumadores y acumuladores, al contrario de la arquitectura típica de un filtro FIR que utiliza multiplicadores. Sin embargo, el filtro CIC no es plano en el ancho de banda de los datos, sino que presenta una caída. Para compensar dicha caída se puede utilizar un filtro FIR.

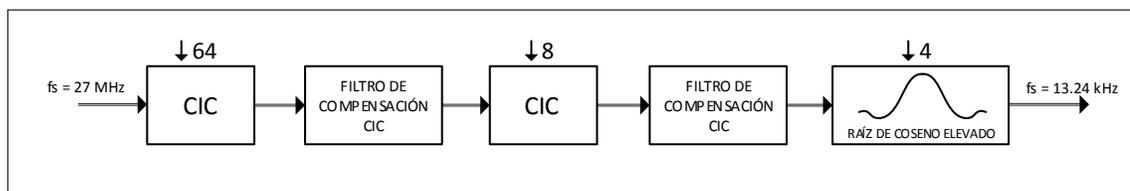


FIGURA 5.4. Cadena de filtros de recepción.

En LATINA interesa que el módulo banda-base sea capaz de conectarse a los módulos RF con componentes discretos (Cangrejos) cuya tasa es 2048 veces la tasa de símbolo, y a futuro circuito integrado RF (Prometeo) cuya tasa es 32 veces la tasa de símbolo, por ello se decidió realizar el submuestreo y el sobremuestro en dos etapas CICC (filtro CIC más filtro FIR de compesación).

La Figura 5.4 muestra la cadena de filtros para el caso de la recepción, donde destacan las dos etapas CICC y el filtro de forma de pulso cuya respuesta es una raíz de coseno elevado que submuetrea la señal 4 veces para llegar finalmente a la tasa de símbolo. La Figura 5.5 representa la respuesta en frecuencia de la cadena de filtros. En dicha figura se destaca como las imágenes del espectro del filtro de raíz de coseno elevador (RRC, por sus siglas en inglés) son eliminadas por los filtros siguientes. Además, es posible observar que la respuesta en conjunto del filtro CIC con el FIR de compensación logran una respuesta plana en frecuencia.

La Tabla 5.2 muestra el costo aritmético de implementar cada filtro en su forma paralela. Aunque se ha puesto esfuerzo en disminuir los recursos utilizados para implementar

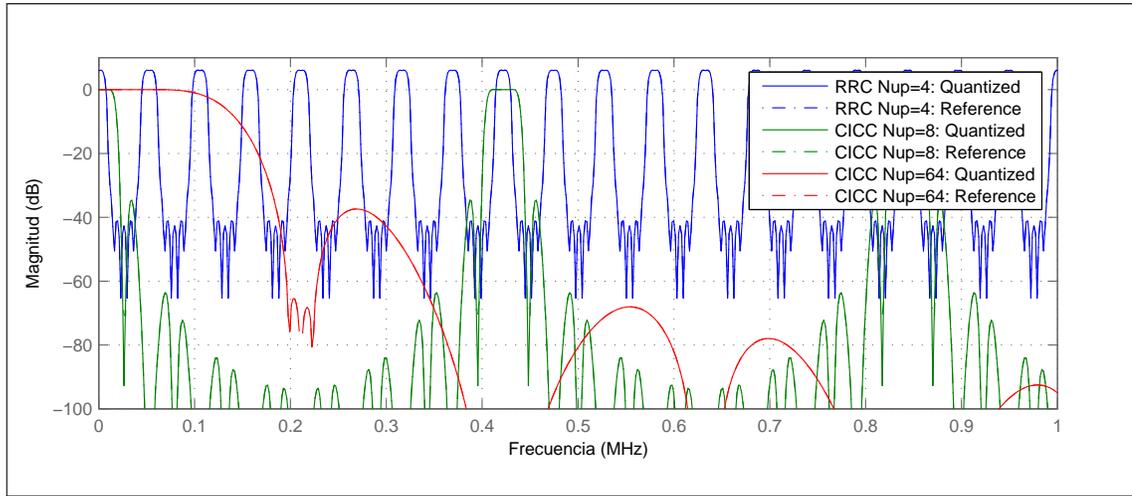


FIGURA 5.5. Respuesta en frecuencia de los filtros.

la interpolación y decimación de los datos, la cantidad de multiplicadores necesarios solo para la cadena de recepción no es despreciable. Por ello se ha optado por implementar cada filtro con una arquitectura serial, utilizando solo un multiplicador y un sumador.

TABLA 5.2. Tabla resumen de la aritmética de cada filtro en caso de ser implementado en forma paralela.

Filtro	Multiplicadores	Sumadores	Factor de sobre muestreo
RRC	25	21	4
Prometeo CIC	0	10	8
Prometeo CIC FIR Comp	15	14	1
Prometeo CIC Normalización	1	0	1
Acuario CIC	0	10	512
Acuario CIC FIR Comp	10	9	1
Acuario CIC Normalización	1	0	1
Total	52	64	2048

Tras realizar simulaciones de Monte-Carlo para distintos valores de SNR se obtuvieron los largos de palabra mínimos en los que no se incurre en una pérdida de implementación (ver Tabla 5.3).

TABLA 5.3. Tabla resumen de los largos de palabra utilizados en los filtro implementados (I = *input*, O = *output*, C = *coefficient*, WL = *word length*).

	IWL	IFL	OWL	ODL	CWL	CFL
RCC	8	7	8	7	16	15
Prometeo CIC	16	15	28	15		
Prometeo CIC FIR Comp	16	15	16	15	16	15
Prometeo CIC Normalización	28	15	16	15	16	15
Acuario CIC	16	15	40	15		
Acuario CIC FIR Comp	16	15	16	15	16	15
Acuario CIC Normalización	40	16	8	7	31	30

5.2.2 Métodos de implementación

Es intuitivo pensar que el largo de las palabras en bits que representa cada variable de procesamiento es proporcional al área utilizada dentro de un ASIC o FPGA (Meyer, Moeneclaey, y Fechtel, 1997), por lo tanto, una de las primera precauciones que se debe tener al implementar físicamente el módulo es definir el largo correcto para cada sumador, multiplicador, bus, registro, etc . La disminución de la razón señal a ruido debido los efectos de cuantización se denominan “pérdida por implementación”. Por otro lado, la degradación con respecto a la varianza de los estimadores de los parámetros de sincronización se denomina “pérdida de detección”.

Por otra parte, dado que se pretende probar el procesador dentro de una FPGA, los recursos para implementar módulos son limitados. Por ello, para ahorrar recursos, se consideraron los siguientes puntos:

Serialización: La mayor parte del procesamiento digital ocurre a una tasa 2048 veces menor que el reloj más rápido del sistema. La implementación de los filtros se realizó con una arquitectura serial ya que la velocidad no es crítica. Aunque el uso de flip-flops y memoria es incrementado, el costo de bloques aritméticos es mayor dentro de la FPGA¹.

Entrelazado: Se puede ahorrar bastante *hardware* si un mismo bloque o procesamiento es utilizado en varios algoritmos. Por ejemplo, un módulo CORDIC se usa

¹El costo de celdas necesarias para implementar un multiplicador aumenta al cuadrado por bit.

para rotar o calcular ángulos. Al entrelazar distintos canales de datos utilizando un tasa mayor se logra procesar distintos canales al mismo tiempo. La Figura 5.6 grafica como funciona el entrelazamiento.

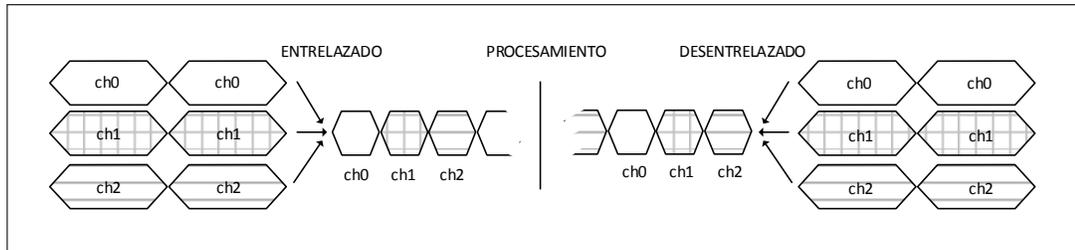


FIGURA 5.6. Esquema de funcionamiento de un procesamiento por entrelazado.

6. ALGORITMOS

Con el objetivo de enfrentar las adversidades del canal inalámbrico se debe incluir en el bloque de banda-base una serie de algoritmos que sean capaces de enfrentar de forma individual o conjunta los problemas descritos en el Capítulo 3. En la mayoría de los libros e investigaciones se tiende a aislar cada efecto de la transmisión-recepción y resolverlo en un programa de computación científica con precisión de 32 o 64 bits en punto flotante, sin embargo, a la hora de realizar una implementación en FPGA no se dispone de memoria y bloques lógicos infinitos. Además, los efectos del AGC sobre la amplitud de la señal a la salida del filtro adaptado tienden a degradar las estimaciones cuando se utiliza una precisión finita en punto fijo.

6.1 Control automático de ganancia

Kettlun (2014) propone un AGC consistente en dos controles: uno grueso, que disminuye radicalmente la ganancia del amplificador cuando se detectan fuertes cambios de energía a la entrada receptor, por ejemplo, el instante de llegada de un paquete, y uno fino, basado en la comparación de energía según una referencia.

La corrección gruesa observa dentro de una ventana de 32 muestras si todas ellas se encuentran en saturación positiva o negativa, en caso de ser afirmativo, el control pide disminuir la ganancia del amplificador en 10 dB.

Por otro lado, la rama de corrección fina utiliza un detector de ley cuadrada, usada comúnmente en los sistemas de AGC. El cuadrado de las muestras de IF es directamente proporcional a la potencia de entrada. Al pasar dicha información por un filtro pasa-bajos se obtiene una estimación de la potencia promedio de la señal recibida. El filtro pasa-bajos es implementado utilizando la siguiente ecuación

$$P[k] = \omega |r[k]|^2 + (1 - \omega)P[k - 1] \quad (6.1)$$

donde $P[k]$ es la muestra del ADC elevada al cuadrado en el instante k .

La diferencia entre la estimación de la potencia recibida y la referencia P_{ref} permite obtener el error instantáneo. Luego, el error es multiplicado por una constante e integrado. Si el error es mayor que cero se debe disminuir la ganancia en 1 dB. Por otro lado, si el error es menor que cero se debe aumentar la ganancia en 1 dB. Entonces, el error se representa como:

$$e_p[k] = K_{fine} (P_{ref} - P[k]) \quad (6.2)$$

donde $K_{fine} \in \mathbb{R}^+$ es una constante. Y el error integrado se expresa como:

$$e_i[k] = e_i[k - 1] + e_p[k] \quad (6.3)$$

6.1.1 Implementación del AGC

La ganancia del amplificador antes del ADC se modifica a través de una interfaz SPI hacia el integrado. Debido al retardo generado al calcular una nueva ganancia y luego transmitirla hacia el amplifica, Kettlun (2014) propuso que la ganancia solo se actualizara cada N_{UR} muestras.

La Figura 6.1 grafica la implementación del AGC. Por un lado, la corrección gruesa se aplica al comprobar si la señal se encuentra en algún riel, si es así se acumula en un registro las veces que ha ocurrido esto de forma consecutiva. Por otro lado, la corrección fina se implementa utilizando sumadores, restadores, multiplicadores y registros que actúan como acumuladores. Cada tipo de corrección genera una señal de control que es utilizada para codificar la dirección de un multiplexor (Tabla 6.1) que contiene la disminución o aumento de ganancia correspondiente a cada corrección.

TABLA 6.1. Codificación de aumento y disminución de ganancias.

Saturación	Error Negativo	Error Positivo	Multiplexación
0	0	0	0
0	1	0	1
0	0	1	2
1	X	X	3

Adicionalmente, se deben considerar los siguientes puntos en la implementación del AGC:

- La ganancia de salida no puede ser 0.
- El AGC debe congelar su ganancia cuando se ha detectado un paquete.
- Los registros deben tener un valor por defecto cada vez que se reinician.
- El AGC debe funcionar *standalone*

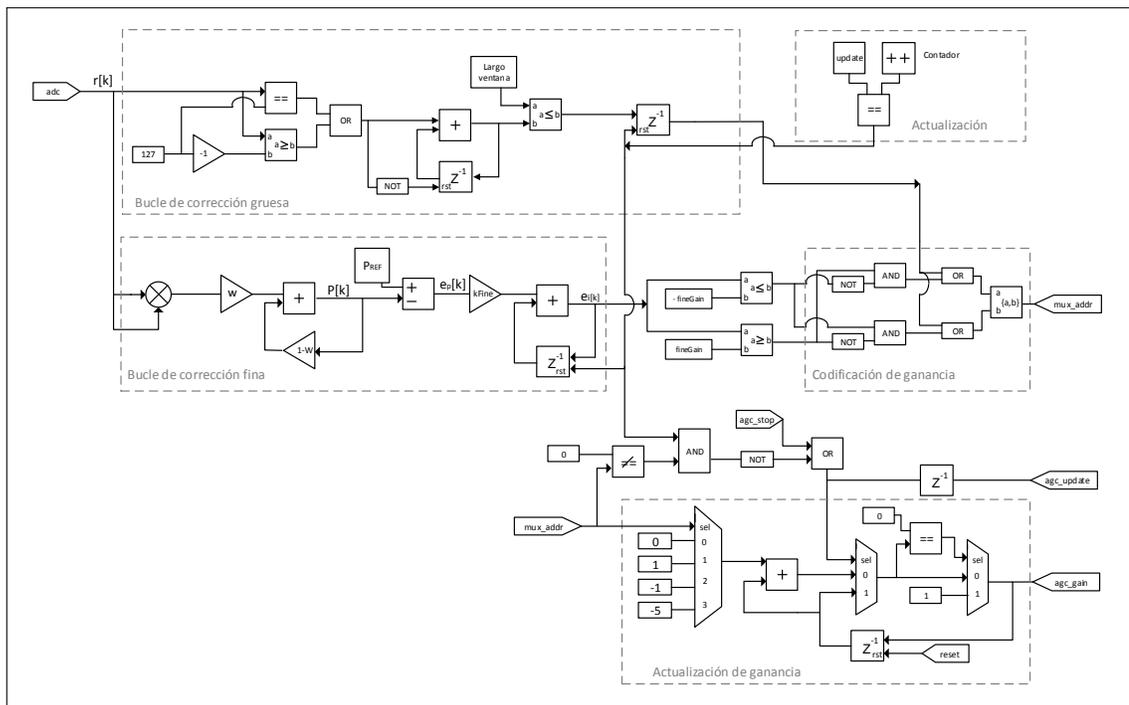


FIGURA 6.1. Diagrama de bloques del control automático de ganancia (AGC).

En el caso de tener múltiples antenas receptoras, el procesador banda-base debe transmitir las nuevas ganancias por la interfaz SPI a cada módulo RF, siendo imposible realizarlo de forma simultánea. Por ello, se decidió que cada rama receptora tuviese un valor distinto al interior del bloque “update”, dentro del cuadro de actualización.

6.2 Detección de paquete

Feres (2013) propone un método de detección de paquete utilizando un algoritmo de correlación y codificación diferencial, que se resume a continuación. Sea a_m , $1 \leq m \leq L-1$, un preámbulo conocido de largo L e $y^p[m]$ el preámbulo recibido a la salida del filtro adaptado. En el instante k_0 , el preámbulo transmitido comienza a recibirse en el receptor y la correlación efectuada durante la recepción genera su punto más alto, esto es

$$R_{SISO}[k_0] = \sum_{m=1}^L y[m - k_0 - 1]a_m^* = \sum_{m=1}^L y^p[m]a_m^* \quad (6.4)$$

Además, para hacer más robusta la detección de paquete ante cambios de fase y desplazamiento en frecuencia de la portadora, Feres (2013) propone que el preámbulo sea codificado diferencialmente siguiendo la idea propuesta por Nagaraj, Khan, Schlegel, y Burnashev (2009). La codificación diferencial permite que la información transmitida esté en los cambios de fase relativos entre símbolos consecutivos y no en la amplitud/fase como en los esquemas clásicos. La Figura 6.2A muestra cómo el preámbulo b_m es codificado diferencialmente para obtener a_m y la Figura 6.2B muestra el diagrama de bloques de la decodificación diferencial.

En resumen, el proceso de detección de paquete es el siguiente: el transmisor toma un preámbulo b_m , lo codifica diferencialmente y lo transmite. En el receptor, a la salida del filtro adaptado se decodifica la señal y se correlaciona con el preámbulo conocido a_m . La métrica de decisión utilizada para detectar si existe un paquete en aire es un umbral sobre la correlación al cuadrado. Para maximizar la diferencia entre el valor pico de la correlación y los valores en tiempos adyacentes se utilizó uno de los preámbulos BPSK

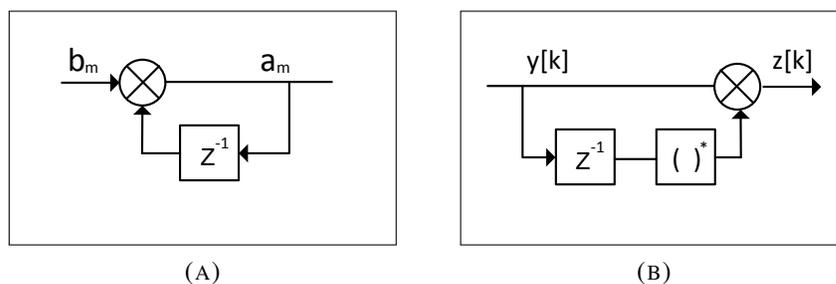


FIGURA 6.2. Codificación y decodificación diferencial.

propuesto por Kaulen (2015b) para el control de acceso al medio en un red inalámbrica de sensores MIMO usando el algoritmo de Feres.

La Figura 6.3 muestra la correlación obtenida tras enviar un preámbulo de 32 símbolos bajo un canal aditivo blanco gaussiano con una SNR de 4 dB. Al simular la transmisión se hizo considerando un AGC y un espacio de solo ruido antes del preámbulo.

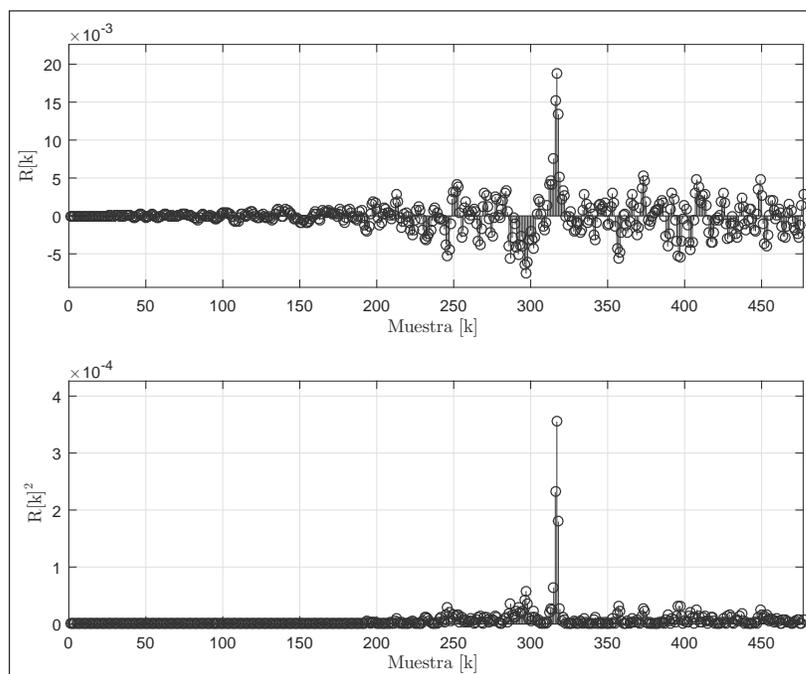


FIGURA 6.3. Correlación de un preámbulo BPSK de 32 símbolos transmitido bajo un canal blanco gaussiano aditivo con SNR de 4 dB.

La extensión a MIMO del algoritmo de detección requiere de la suma de las correlaciones de cada antena, dada por la Ecuación (6.5).

$$R_{MIMO}[k_0] = \sum_{i=1}^{N_r} \sum_{m=1}^L y_i[m - k] a_m^* \quad (6.5)$$

6.2.1 Implementación de la detección de paquete

Aprovechando la ventaja de enviar un preámbulo BPSK se puede simplificar la implementación de la detección de paquete considerando lo propuesto por Villers-Grandchamps (2014), y reducirlo a un bloque que utiliza solo sumas e inversiones.

La codificación diferencial tiene la ventaja de ser robusta a desplazamiento en frecuencia de la portadora, sin embargo, su implementación requiere de multiplicadores. Si consideramos que cada antena necesita su propio multiplicador, el detector de paquete comienza a ser costoso en término de compuertas lógicas. Por ello, aprovechando que el procesamiento de datos ocurre a una tasa 512 veces menor que el reloj del sistema se puede utilizar el procesamiento entrelazado propuesto en la Sección 5.2.2 para solo utilizar un multiplicador complejo.

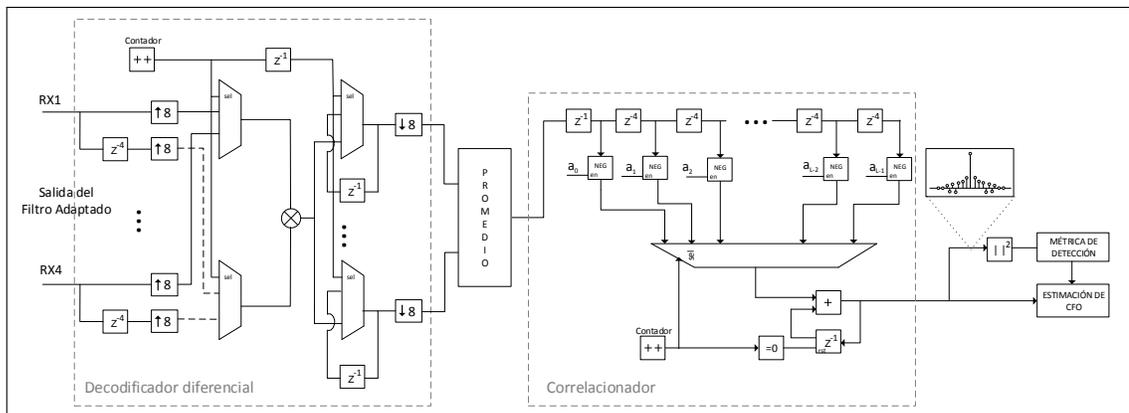


FIGURA 6.4. Diagrama del algoritmo de detección de paquete para 4 antenas receptoras.

Para determinar el largo de cada bus de datos se utilizaron simulaciones de Monte Carlo a objeto de observar el valor de la correlación al cuadrado promedio para distintas SNR, ya que debido al efecto del AGC, el valor pico será al disminuir la SNR. La curva de la Figura 6.5 muestra el valor pico promedio y la varianza de la correlación al cuadrado para distintas SNR.

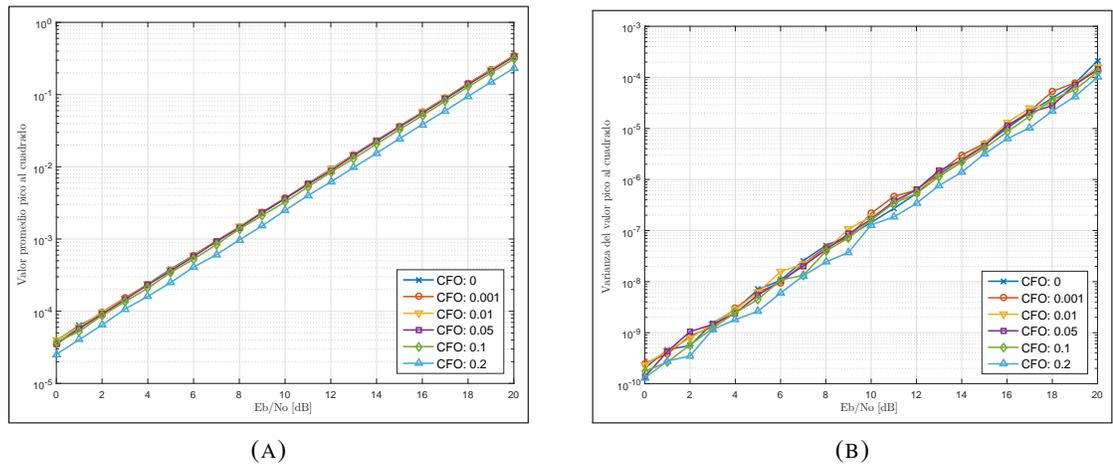


FIGURA 6.5. (A) Valor promedio del *peak* a la salida del correlacionador (B) Varianza del valor *peak* a la salida del correlacionador

6.3 Estimación gruesa del desplazamiento en frecuencia de la portadora

Los estimadores de desplazamiento en frecuencia de la portadora (CFO, por sus siglas en inglés) que son más prácticos de implementar están basados en los trabajos de (Fitz, 1991; Kay, 1989; Luise y Reggiannini, 1995). Poseen un buen comportamiento a baja SNR, a excepción del algoritmo de Kay que funciona a partir de los 9 dB. Cabe destacar que todos estos algoritmos y sus derivados asumen que no existe desplazamiento en el tiempo de muestreo y que el desplazamiento de la portadora es mucho más pequeño que la tasa de símbolo.

Utilizando el resultado de la correlación de la detección de paquete, Feres (2013) propone calcular el ángulo de la correlación en el instante de la detección de paquetes para obtener una estimación del CFO (Ecuación (6.6)).

$$\Delta \hat{f} = \frac{1}{2\pi T} \angle \sum_{m=1}^L y^p[m] a_m^* \quad (6.6)$$

La Figura 6.6 muestra el valor esperado bajo una SNR de 4 dB para distintos CFO y varianza del los estimadores clásico y el propuesto por Feres (2013). Los gráficos fueron obtenidos considerando una representación en punto fijo de 16 bits enteros y 13 bits fraccionarios para los estimadores.

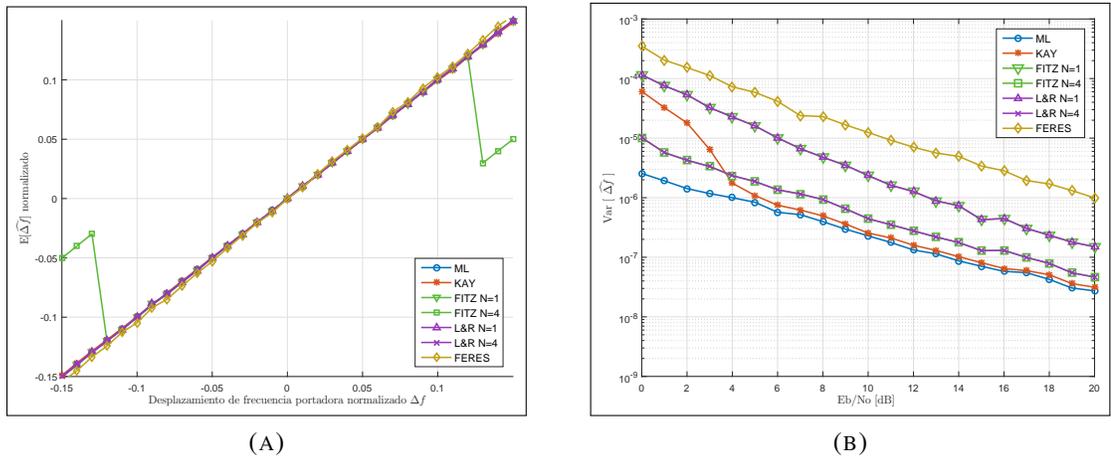


FIGURA 6.6. (A) Valor medio del estimador de CFO grueso para un SNR de 4 dB y distintos CFO (B) Varianza del estimador de CFO grueso para distintas SNR

6.3.1 Implementación del estimador grueso de CFO

Las especificaciones del oscilador local (Tabla 2.1) considera un error de +/- 10 ppm. Por lo tanto, la desviación más grande que puede ocurrir entre el transmisor y el receptor es de 20 ppm. En el módulo RF del sistema el error se ha acotado a 2.5 ppm. Así el desplazamiento máximo de frecuencia de la portadora está dado por la Ecuación (6.7).

$$CFO_{max} = 2 \cdot (2,5) \cdot 10^{-6} \cdot 915 \cdot 10^6 = 4575 Hz \quad (6.7)$$

Si se normaliza el valor anterior con respecto a la tasa de símbolo se obtiene:

$$\Delta f_{max} = \frac{4575 \text{ Hz}}{13240 \text{ Hz}} = 0,3455 \quad (6.8)$$

Se puede observar que el estimador de Feres no es muy bueno con respecto los algoritmos clásicos. Sin embargo, la ventaja del estimador de Feres radica en su implementación, ya que solo requiere un bloque CORDIC capaz de calcular el ángulo de la correlación.

6.4 Estimación fina del desplazamiento en frecuencia de la portadora

En la sección anterior se estableció una estimación gruesa del desplazamiento de frecuencia de la portadora aprovechando el resultado de la correlación de la detección de paquete. Aunque se podría mejorar el estimador grueso, aún no se ha asegurado el instante óptimo de muestreo del filtro adaptado. En consecuencia, se propone utilizar un estimador de CFO fino a continuación del algoritmo de corrección de tiempo de muestreo, no incluido en esta tesis. La ventaja de hacer una corrección de dos pasos es que se puede utilizar un estimador de CFO que posea un rango más acotado de operación pero con mejor varianza y/o menores recursos de *hardware*.

Kaulen (2015a)¹ (DKL) elaboró una simplificación del estimador de CFO de Luise y Reggiannini (1995) (L&R), donde se utiliza la resta en vez de la multiplicación para calcular la diferencia de fase entre símbolos consecutivos. Se define $\theta[m]$ como la resta de ángulos entre el preámbulo recibido y el preámbulo conocido, de la siguiente forma

$$\theta[m] = \angle y^p[m] - \angle a_m^*, \quad 1 \leq m \leq L - 1 \quad (6.9)$$

Al igual que el algoritmo de L&R, el estimador DKL definido como

¹Informe técnico interno de LATINA UC.

$$\Delta \hat{f} = \frac{1}{\pi T_s N(N+1)} \sum_{k=1}^N \sum_{i=k+1}^L \frac{1}{L-k} (\theta[i] - \theta[i-k]) \quad (6.10)$$

tiene dos parámetros para configurar. Uno es el parámetro L dado por el largo del preámbulo, y el otro, es N que define la cantidad de sumas acumuladas que se desean realizar sobre el preámbulo. Al aumentar N la estimación mejora debido a que se obtienen diferencias de fase entre símbolos no consecutivos.

La Figura 6.7A muestra el valor medio de estimador DKL en comparación con los algoritmos clásicos para distintos valores de CFO bajo un canal blanco gaussiano cuya SNR es de 4 dB. Se observa que el estimador DKL para $N = 4$ funciona en un rango menor de CFO que el resto, sin embargo, posee buen comportamiento en su varianza como se muestra en la Figura 6.7B.

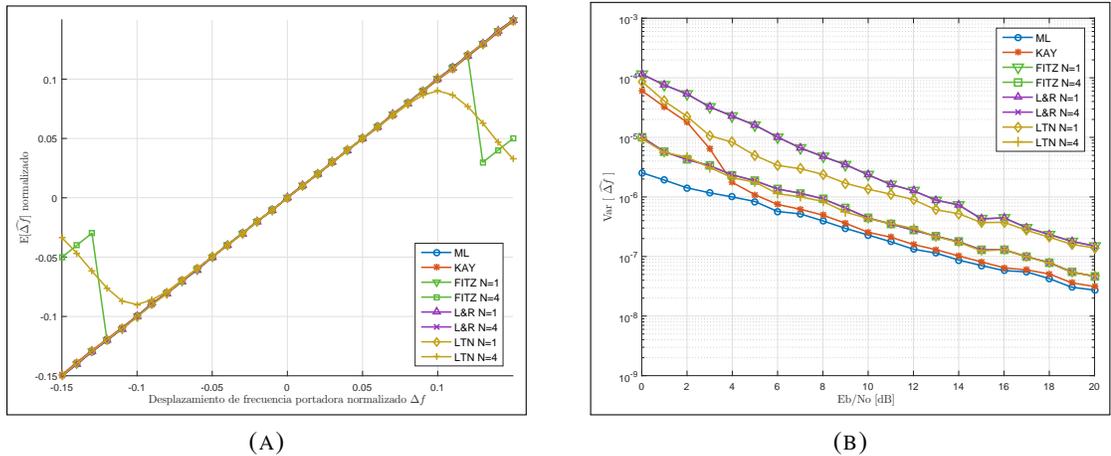


FIGURA 6.7. (A) Valor medio del estimador de CFO fino para un SNR de 4 dB y distintos CFO (B) Varianza del estimador de CFO fino para distintas SNR .

6.4.1 Implementación estimador fino de CFO

La Figura 6.8 muestra la implementación del algoritmo DKL para $M = 4$. El bloque “limitador de rango” restringe el valor de la fase al rango $(-\pi, \pi]$. Los multiplicadores por

$1/(L - k)$ y el bloque para calcular la fase de los preámbulos son los que más recursos utilizan en la implementación de este algoritmo.

En el caso MIMO se debe replicar el bloque de la Figura 6.8 por cada antena para luego promediar las estimaciones. Observando la estructura del estimador se puede utilizar el método de procesamiento por entrelazado a partir del limitador de rango hasta antes del acumulador, y con ello reducir el *hardware* utilizado.

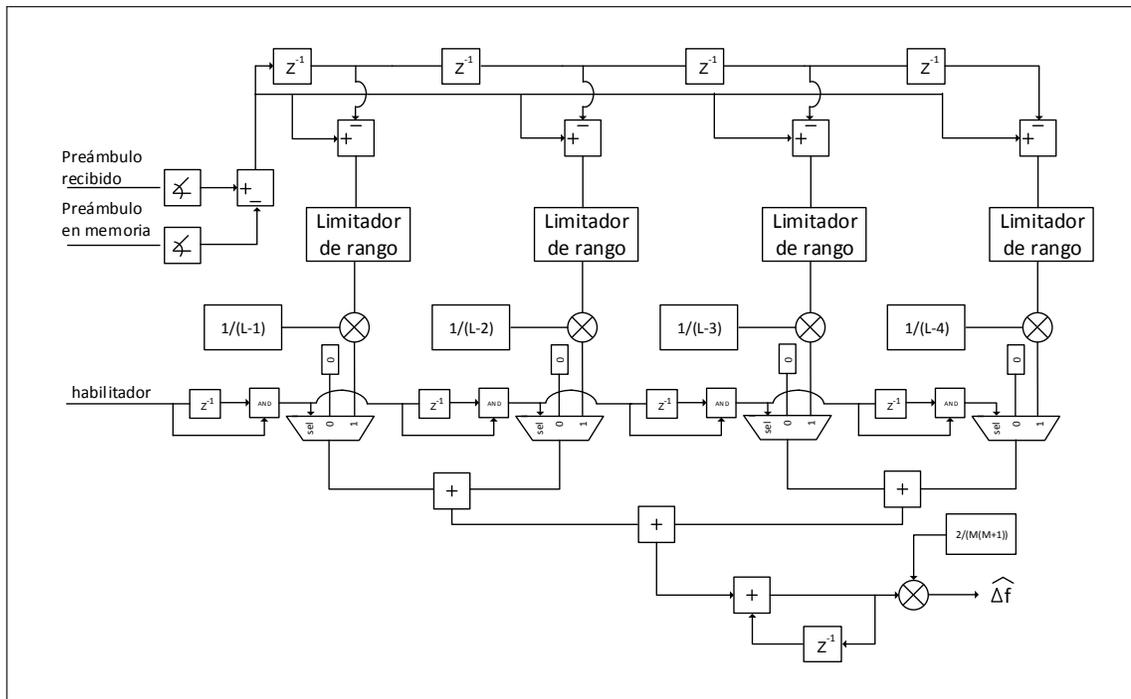


FIGURA 6.8. Implementación del estimador fino de CFO para $N = 4$.

6.5 Estimación de canal

La estimación de canal es un aspecto importante del módulo banda base debido a que su resultado puede mejorar bajo diversos escenarios el rendimiento del procesador banda-base. Además, es requisito esencial para el uso de técnicas con múltiples antenas.

La siguiente ecuación muestra el preámbulo recibido después de ser corregido por los algoritmos de estimación grueso y fino CFO.

$$y^p[m] = e^{j2\pi(\Delta f_r)} \tilde{h} a_m + \tilde{n}[m], \quad 1 \leq m \leq L - 1 \quad (6.11)$$

Los términos que afectan al preámbulo recibido incluyen el CFO residual proveniente de la varianza de los estimadores, el canal, al cual se le ha adjudicado cualquier desplazamiento de fase, y el ruido a la salida del filtro adaptado.

Considerando el caso SISO, Feres (2013) propone el estimador

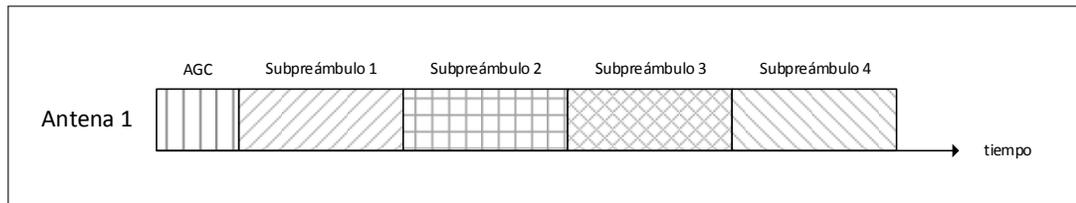
$$\hat{h} = \sum_{m=1}^L y^p[m] c_m \quad (6.12)$$

$$c_m = \frac{a_m^*}{\|a\|^2}, \quad 1 \leq m \leq L \quad (6.13)$$

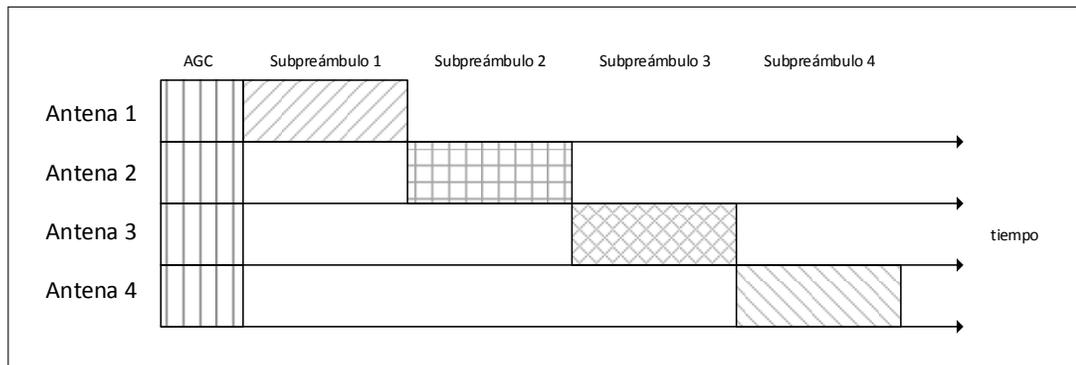
donde los coeficientes c_m corresponden al preámbulo codificado diferencialmente y normalizado por su energía al cuadrado.

6.5.1 Extensión MIMO

La Figura 6.9 muestra la secuencia propuesta por Feres (2013) en la cual se divide el preámbulo de entrenamiento en N_t subpreámbulos. Con el objetivo de estimar el canal entre cada antena transmisora y receptora, al transmitir los subpreámbulos en forma de escalera set evita que exista una combinación lineal entre antenas transmisoras permitiendo la llamada “estimación vectorial”.



(A)



(B)

FIGURA 6.9. Constelación 64-QAM (original y rotada) representada en punto fijo con 8 bits enteros y 7 fraccionarios.

6.5.2 Implementación del estimador de canal

Uno de los propósitos de crear una plataforma de pruebas para comunicaciones de múltiples antenas es la posibilidad de comparar distintos esquemas de transmisión y recepción: SISO, SIMO o MIMO. En consecuencia, la implementación del estimador de canal debe seguir la línea anterior y debe ser fácil de configurar dependiendo del esquema de comunicación.

La Figura 6.10 muestra la implementación propuesta. Considerando un preámbulo de 32 símbolos y 4 antenas, el módulo utiliza un contador de 5 bits para poder direccionar una ROM que contiene los coeficientes c_m . Los 3 primeros bits son utilizados para poder procesar cada sub-prámbulo por separado. En caso de estar en modo SISO o SIMO, basta con promediar las estimaciones por cada antena transmisora para mejorar la estimación.

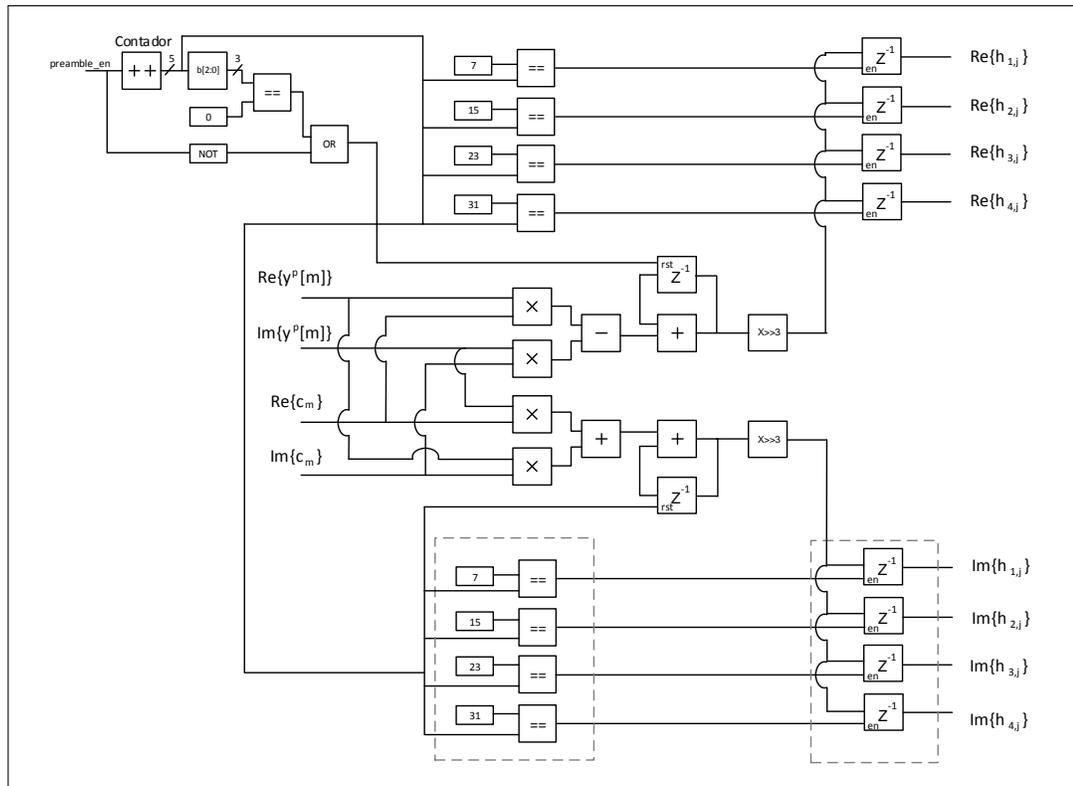


FIGURA 6.10. Implementación del estimador de canal.

6.6 Estimación de desplazamiento de fase de la portadora

Hasta ahora se ha sincronizado la portadora en frecuencia. Sin embargo, errores de fase o pequeños errores de frecuencia pueden hacer rotar lo suficiente al espacio de señal para que un símbolo se encuentre en una región de decisión errónea. El método típico para enfrentar los errores residuales de los estimadores de frecuencias y, además, ajustar el desfase de la portadora, es la implementación de un lazo de seguimiento de fase (PLL, por sus siglas en inglés) implementado como un bucle de segundo orden (Mengali y D'Andrea, 1997), el cual permite el seguimiento y corrección a lo largo de todo el paquete.

La Figura 6.11 muestra el diagrama de bloques de un bucle de segundo orden. Dado que el PLL solo corregirá la fase del campo de datos, se ha propuesto utilizar el preámbulo para evitar restricciones estrictas sobre el transiente y *overshoot* de la respuesta en el tiempo.

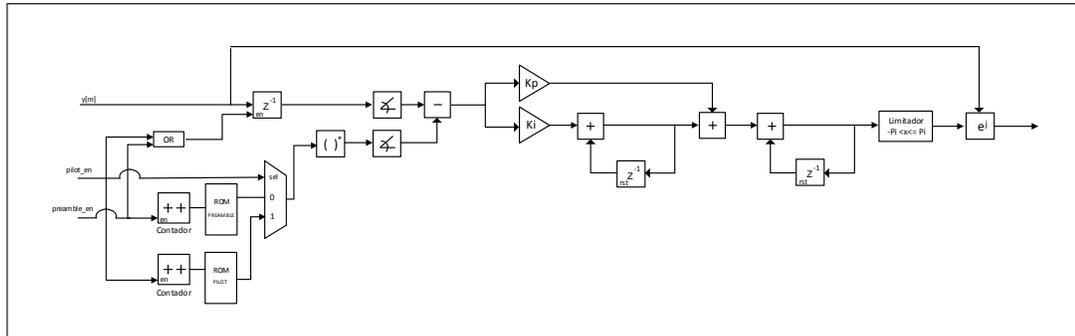


FIGURA 6.11. Implementación del estimador de desplazamiento de fase de la portadora.

La Figura 6.12 muestra la respuesta en el tiempo del PLL para una transmisión bajo un canal aditivo blanco gaussiano con SNR de 4 dB. Se diseñó el control para que el transiente no dure más de los 32 símbolos que corresponden al preámbulo. La Figura A.1, en el Anexo, muestra el resultado de la corrección en el diagrama de dispersión de los símbolos de datos recibidos.

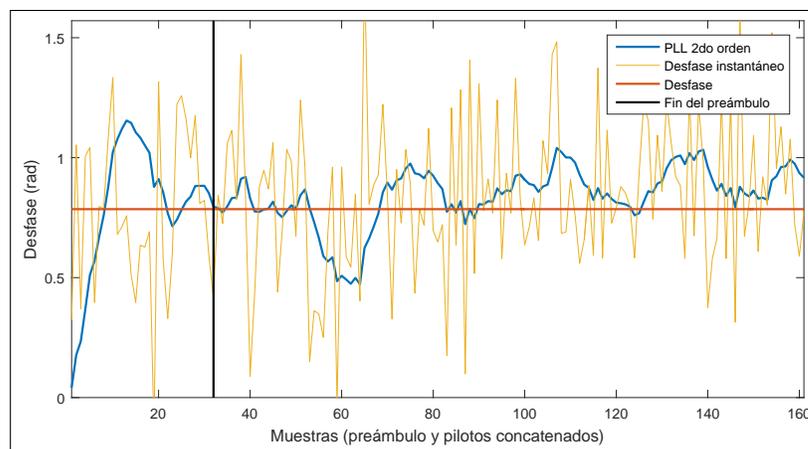


FIGURA 6.12. Estimación del desplazamiento de fase de la portadora a lo largo de un paquete QPSK transmitido en un canal aditivo blanco gaussiano con SNR de 4 dB y un desfase de $\pi/4$.

Los parámetros de diseño del PLL son el factor de amortiguación y el ancho de banda del filtro, los cuales fueron ajustados según la pérdida de SNR en decibeles de la tasa de

error de bit promedio bajo un canal blanco gaussiano. La Figura 6.13 muestra la pérdida de razón señal cuando no existe desplazamiento en frecuencia de la portadora.

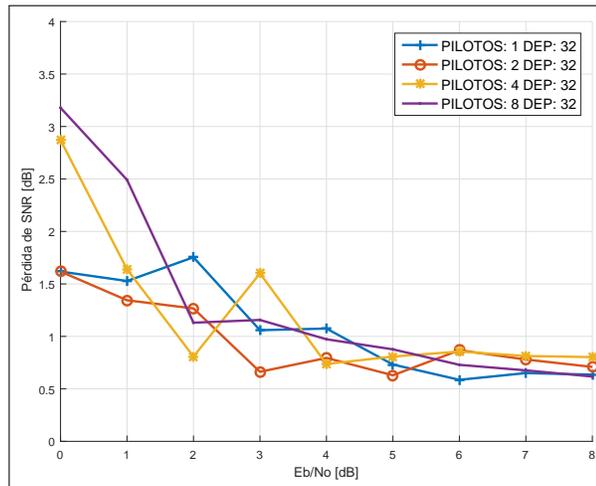


FIGURA 6.13. Pérdida de razón señal ruido dada la cantidad de pilotos consecutivos en el PLL implementado bajo un canal aditivo blanco gaussiano sin desplazamiento en frecuencia de la portadora

6.6.1 Estructura de paquete

En cada transmisión el paquete enviado tiene cierta estructura formada por distintos “campos de símbolos“ que permiten el correcto funcionamiento de cada algoritmo propuesto. A continuación se detalla cada uno de los campos:

Campo AGC: contiene los símbolos necesarios para que el algoritmo de AGC se encuentre en régimen permanente. Debido al diseño del algoritmo implementado se requiere un largo mínimo de 4 símbolos. Cabe destacar que la mayoría de los estándares MIMO (IEEE 802.11n, WiMAX, LTE) utilizan un preámbulo para la estabilización del AGC.

Campo Preámbulo: Considera los símbolos conocidos por el receptor que permiten realizar la detección de paquete y las estimaciones de otros parámetros como el CFO.

Campo Señal: Contiene la información de la cantidad de símbolos enviados en el paquete y el número M-ario de la modulación.

Campo Pilotos: Son los símbolos necesarios para la compensación de la fase de los símbolos correspondientes a los datos. El número de pilotos consecutivos tiene un gran efecto en el desempeño del sistema en baja SNR.

Campo Datos: Contiene los símbolos de información y establece la distancia entre pilotos, determinando la frecuencia residual máxima permitida.

La Figura 6.14 muestra cómo los campos de símbolos se ordenan al interior de una paquete. Cabe mencionar que el campo de señal lleva antes de él un campo de pilotos, ya que se debe corregir su fase para posteriormente demodular y decodificar su información.

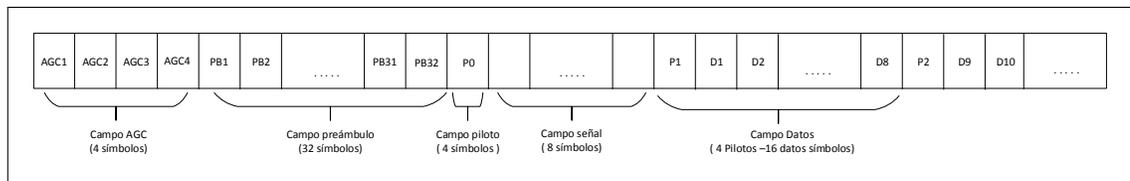


FIGURA 6.14. Estructura del paquete.

6.6.2 Máquina de estado

La Figura 6.15 muestra el diagrama de estados que permite generar un paquete. La salida de la máquina permite habilitar otros bloques según se necesite, por ejemplo, habilitar un contador que direcciona una ROM que contiene el preámbulo.

La máquina comienza y se mantiene en el estado “RESET” a menos que se requiera enviar un paquete o si se ha detectado un paquete. Además, es necesario reconocer si el procesador se está utilizando como transmisor o receptor, ya que en la recepción los símbolos de AGC distorsionados no son necesarios para ningún tipo de procesamiento.

Si la entrada *bb_packet_request* cambia a 1 y la entrada “mode” está en 0 (procesador en modo transmisor) significa que el usuario ha enviado un comando para iniciar la transmisión de los datos contenidos en la memoria FIFO. La máquina de estados

cambiará del estado “RESET” a “AGC”. Por otro lado, si la entrada “mode” está en 1 (procesador en modo receptor), cuando se haya detectado un paquete en el aire la señal “bb_packet_detected” cambiará a 1 y la máquina de estados pasará directamente del estado “RESET” a “TRAIN”.

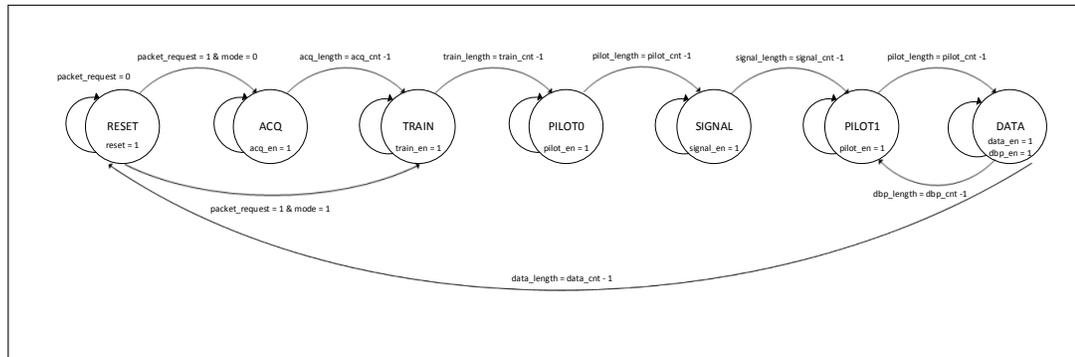


FIGURA 6.15. Máquina de estados para generar paquete.

6.6.3 Diagrama de bloques a nivel de procesamiento

La Figura 6.16 muestra un diagrama de bloques a nivel de procesamiento considerando una recepción MIMO de 4 antenas. A continuación se describirá el flujo de datos de señal en los bloques y el porqué de algunas decisiones:

- A la entrada, las muestras del ADC se dirigen directamente al bloque de control de ganancia automática (AGC). El bloque envía cada cierto tiempo la ganancia que se debe actualizar en el amplificador de ganancia variable. El AGC se mantiene funcionando hasta que detecta un paquete, en dicho caso, se congela su procesamiento.
- Desde el módulo *downconverter* hasta el filtro adaptado, observando el camino inferior del multiplexor, se tiene la misma cadena de filtros de la Sección 5.5. Una vez que se ha detectado un paquete se utiliza el camino superior, el cual posee un *buffer* que ha guardado el preámbulo para luego ajustar las muestras desde el instante de detección de paquete hasta el inicio de un periodo de símbolo.

Además, corrige el CFO con la información obtenida de la estimación realizada durante la detección de paquetes.

- El bloque de sincronización de muestreo no se ha incluido en esta tesis pero se agregará en un futuro en la posición reservada en la figura.
- A partir del bloque de *down sampling* por 4 las muestras se encuentran a tasa de símbolo.
- La estimación de CFO fina debe ir acompañada de un *buffer* que permite la posterior corrección del preámbulo y con ello mejorar la estimación de canal.
- A partir de la corrección de CFO fino se puede utilizar cualquier método o algoritmo MIMO para mejorar la recepción de los datos utilizando diversidad.
- A continuación, el preámbulo y los pilotos se utilizan para estimar el desfase de la señal para poder corregirlo. Finalmente, los símbolos corregidos en fase entran al demodulador.

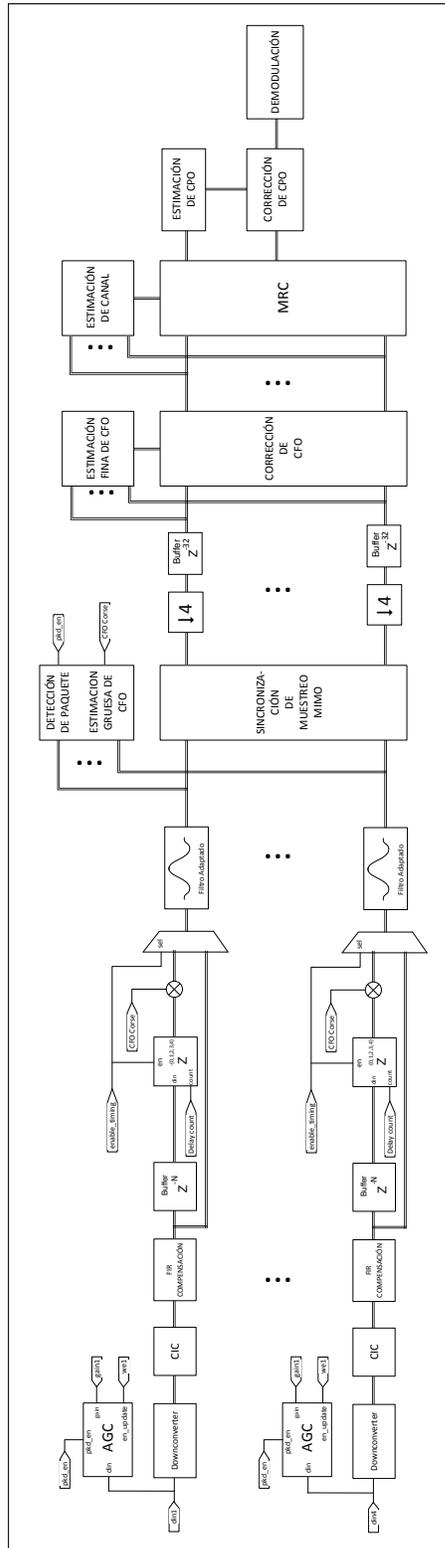


FIGURA 6.16. Arquitectura del módulo de procesamiento banda-base.

7. SIMULACIONES

El diseño de sistemas de comunicaciones termina inevitablemente en un proceso iterativo que determina un punto de operación donde se espera un balance entre complejidad y rendimiento. La sinergia entre algoritmos y la gran cantidad de decisiones que se deben tomar en el diseño del procesamiento banda-base (largo del paquete, número de símbolos en el preámbulo, cuantización, elección del estimador de corrimiento de frecuencia y fase) hacen necesario observar alguna figura de mérito como la tasa de error de bit (BER, por sus siglas en inglés) o la tasa error de paquete (FER, por sus siglas en inglés), las cuales son usadas comúnmente para predecir el rendimiento del sistema (Proakis, 2001). El método tradicional para obtener una estimación de BER es a través de simulaciones de Monte-Carlo.

7.0.4 Pérdida de desempeño producto de la estructura de paquete

Uno de los aspectos que afecta la tasa de error de bit de la comunicación es la estructura del paquete. El tamaño del preámbulo determina en gran medida la varianza de los parámetros de sincronización en la recepción. No obstante, se decidió mantener el largo de 32 símbolos definido en las simulaciones del algoritmo *Pin-Pong Payload* desarrollado por (Kettlun, 2014) en LATINA con el objetivo de comprobar su funcionamiento real en un futuro.

Otro factor que se debe considerar es la cantidad de pilotos consecutivos que se utilizan para la estimación de error de fase y la corrección del desplazamiento en frecuencia de la portadora residual. Al mismo tiempo, se debe considerar la cantidad de símbolos de datos que existirán entre los pilotos. Por un lado, si se tiene demasiados datos entre pilotos (DEP) se utilizará una misma corrección de fase para corregir muchos más datos, aumentando la tasa de error de bit. Por otro lado, si se tiene demasiados pilotos consecutivos la transmisión será muy ineficiente y el paquete enviado sería muy largo.

La Figura 7.1 muestra la pérdida en decibelios de SNR con respecto a una transmisión ideal QPSK bajo un canal aditivo blanco gaussiano. En otras palabras se mide las

cantidad de decibeles en la SNR necesarios para que la comunicación realizada tenga un comportamiento igual a la curva teórica.

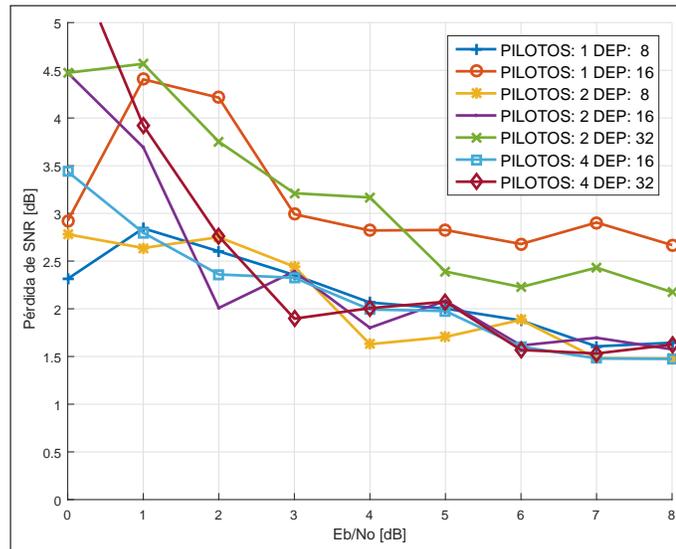


FIGURA 7.1. Pérdida de SNR para distintos valores de DEP (datos entre pilotos) y números de pilotos consecutivos.

7.1 Estimación de la tasa de error de bit

La tasa de error de bit resulta ser un buen punto de comparación entre las simulaciones realizadas en MATLAB y la implementación debido a que es una medida de desempeño del sistema. Sin embargo, realizar simulaciones de Monte-Carlo en *hardware* implica tener un método con el cual generar muestras aleatorias gaussianas a la tasa de IF digital (27 MHz). Por ello es necesario implementar un emulador de canal que replique el canal descrito en el código MATLAB.

7.1.1 Hardware de emulación de canal discreto

El término de *hardware* de emulación de canal discreto fue acuñado por Boutillon, Tang, Marchand, y Bomel (2010) para referirse a un dispositivo capaz de imitar los efectos de un canal banda base. La complejidad de un circuito digital capaz de realizar dicha tarea está en la generación de variables aleatorias que sigan una distribución determinada.

Los métodos clásicos para obtener la realización de un variable aleatoria normal son el método de Box Muller (Box y Muller, 1958) , el método de Wallace (Wallace, 1996) y la función de distribución inversa gaussiana acumulada (Chhikara, 1988). Además, se han realizado investigaciones para implementar eficientemente en *hardware* dichos métodos (Boutillon et al., 2010; Lee, Cheung, Villasenor, y Luk, 2006; Lee, Luk, Villasenor, Zhang, y Leong, 2005; Lee, Villasenor, Luk, y Leong, 2006).

En esta tesis se utilizó la implementación del método de evaluación de la función de distribución inversa gaussiana acumulada (IGCDF, por sus siglas en inglés) mediante el uso del esquema de segmentación jerárquica propuesta por (Lee, Cheung, et al., 2006). La función IGCDF es altamente no-lineal, y Lee propuso un esquema de evaluación de aproximaciones polinomiales (*splines*) por partes con un esquema de rangos de segmentación variables que disminuyen en tamaño en potencia de dos en aquellas regiones no-lineales. Se decidió por el método de Lee debido a que posee un esquema de implementación que depende de una ROM con los coeficientes del polinomio cuadrático. En caso de querer implementar otra distribución a futuro basta con cambiar los coeficientes de la memoria.

La generación de una variable aleatoria normal tiene por objetivo emular ruido blanco y realizar transmisiones a distintas razones señal a ruido . Las características del módulo implementado son las siguientes:

- Valor máximo de $7,6\sigma$, donde σ es la desviación estándar.
- Muestras gaussianas con resolución de punto fijo. Palabras de 16 bits con 11 bits fraccionarios.
- La función IGCDF es aproximada por segmentación jerárquica y polinomios de grado 2 con un error máximo de aproximación de $0,3 \times 2^{-11}$.
- Periodicidad de 10^{13} .
- 45 segmentos externos cuyo largo va disminuyendo en potencias de 2.
- 120 segmentos internos con tamaño uniforme.

Un generador de números uniformes (URNG, por sus siglas en inglés) es usado como entrada x y adicionalmente para obtener un signo aleatorio en las muestras gaussianas. La

unidad de evaluación de IGPDF aproxima la función $y = |F^{-1}(x/2)|$ y un multiplexor selecciona si la muestra es positiva o negativa. La Figura 7.2 muestra la segmentación realizada a la función IGPDF.

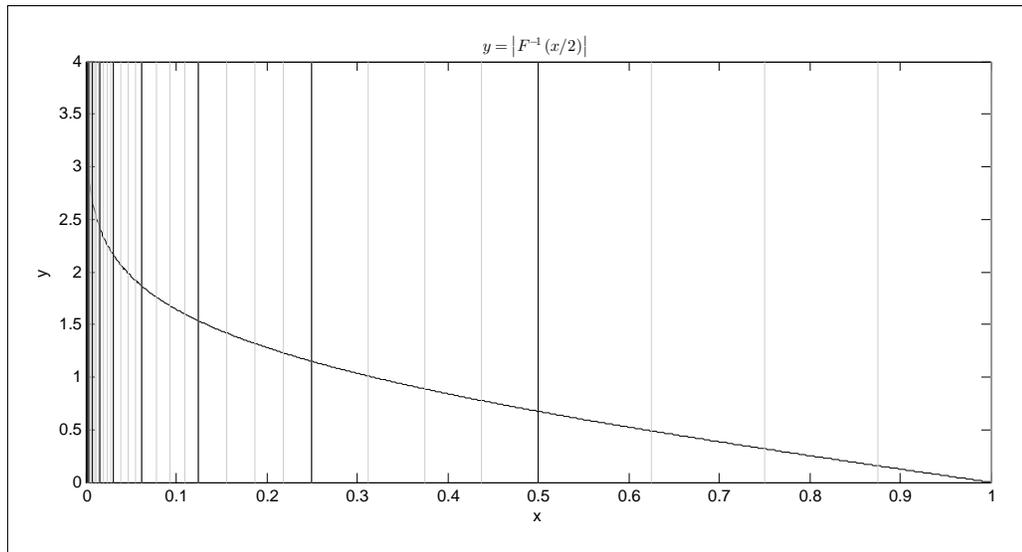


FIGURA 7.2. Segmentación jerárquica aplicada a IGPDF.

Finalmente, realizando una prueba de bondad de ajuste de Chi-cuadrado con un nivel de significancia de $\alpha = 0,000005$ se comprobó que la distribución de los datos corresponden a una variable aleatoria gaussiana de media 0 y varianza 1. La Figura 7.3 muestra la distribución de las muestras obtenidas de la implementación realizada utilizando la arquitectura propuesta por Lee, Cheung, et al. (2006). La unidad fue implementada en FPGA utilizando el programa “System Generator” de Xilinx en MATLAB.

7.1.2 Resultados

Se implementaron dos procesadores banda-base en *hardware* conectados por medio del emulador de canal. Desde MATLAB es posible comunicarse con el procesador vía ethernet y con ello modificar registros, cargar la memoria FIFO con datos binarios e iniciar la transmisión y recepción. Una vez recibido un paquete el procesador que hace de receptor envía los bits recibidos al computador con MATLAB.

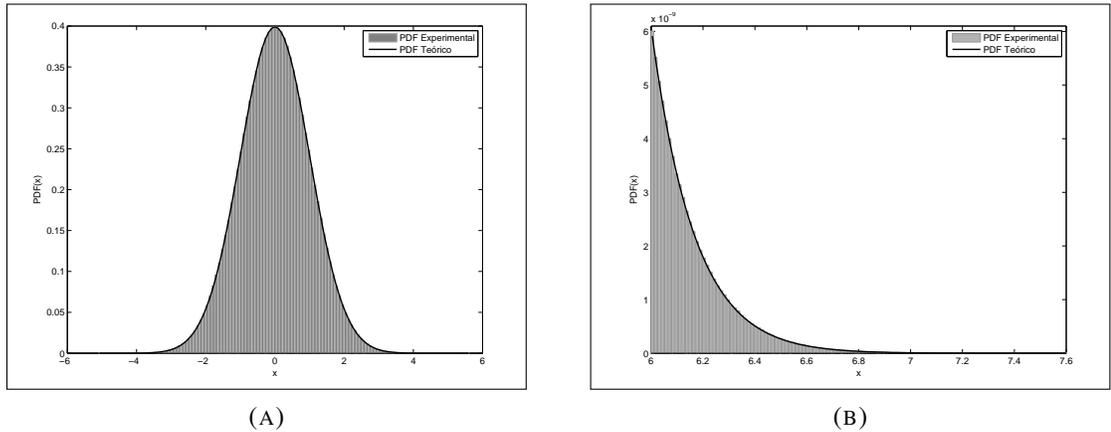


FIGURA 7.3. (A) Distribución de probabilidad de 10^7 muestras gaussianas obtenidas de la unidad implementada. (B) Distribución de probabilidad para 10^6 muestras entre 6σ y $7,6\sigma$

La Tabla 7.1 muestra el tiempo que demora la transmisión y recepción de un paquete mediante una simulación en MATLAB, otra en Simulink/System Generator en la cual se utilizan bloques que modelan código HDL y finalmente en el sistema procesador y emulador de canal implementado en una FPGA. . Cabe destacar que el tiempo indicado para la simulación por FPGA incluye los comandos por Ethernet para cargar los bits de información , iniciar la transmisión y obtener los bits recibidos. El paquete transmitido tiene la estructura definida en la Sección 6.6.1 utilizando 8 símbolos de datos cada 1 piloto y enviando 200 bits de datos útiles.

TABLA 7.1. Tiempo de simulación por paquete de 200 bits transmitido.

Tipo de simulación	Tiempo (s)
Código	23
Simulink/System Generator	810
FPGA	¡1

La Figura 7.4 muestra la tasa de error de bit estimada dado un valor de razón de señal a ruido para el procesador banda-base implementado en FPGA y una simulación realizada en MATLAB. En ambos casos se transmitieron paquetes bajo un canal aditivo

blanco gaussiano con un CFO normalizado de 0,01 y con una estructura de paquete de 8 símbolos de datos cada 1 pilotos consecutivos.

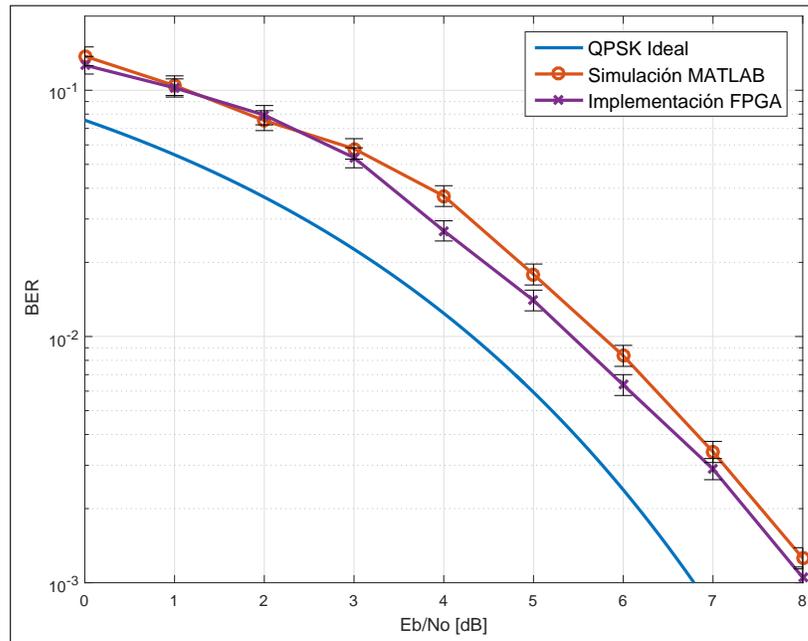


FIGURA 7.4. Comparación de BER entre el módulo banda-base simulado en MATLAB y el implementado en FPGA.

8. CONCLUSIONES Y TRABAJO FUTURO

8.1 Discusión y resultados

El estado del arte de las plataformas experimentales o *testbed* se encuentra enfocado hacia tecnologías de múltiples antenas y de gran ancho de banda como WiMax y LTE. Por este motivo, en LATINA se decidió crear una plataforma propia para la investigación en las llamadas *wide area ubiquitous networks*, esto es, comunicaciones de largo alcance y ancho de banda angosto. El procesador banda-base presentado en esta tesis podrá ser utilizado en el *testbed* de LATINA para probar bajo condiciones reales algoritmos que utilicen múltiples antenas.

Utilizando el enfoque propuesto por Rupp et al. (2003) se propuso una metodología que consistió en el uso de un único archivo que contiene los parámetros de todo el sistema, buscando unificar los tres tipos de equipos de trabajo existentes en un proyecto (investigación, diseño e implementación) y así evitar caer en errores como la falta de documentación e incompatibilidad entre ambientes de trabajo.

La arquitectura del procesador banda-base presentada y su interfaz de comunicación vía ethernet permite a investigadores abstraerse de las implementaciones de bajo nivel. Por ejemplo, mediante un *software* como MATLAB es posible comunicarse con varios Acuarios desde un mismo computador y así probar algoritmos de red. Además, considerando una tasa de transferencia de 100 Mb/s de la comunicación por ethernet es posible muestrear y recibir en tiempo real información de cualquier señal al interior del módulo banda-base.

Utilizando algoritmos desarrollados en LATINA se creó un esquema de banda-base capaz de estimar y corregir los efectos del canal inalámbrico. La arquitectura propuesta consiste en una corrección de tres partes. Primero, se utiliza una estimación gruesa del CFO proveniente de la correlación de la detección de paquetes. Segundo, una vez corregido el paquete con la estimación gruesa se vuelve a pasar por el filtro adaptado. Tercero, el paquete vuelve a tasa de símbolo y se utiliza para realizar una estimación del desfase.

Luego, paralelamente ocurre la estimación de canal. Esta arquitectura tiene por objetivo minimizar el tamaño del paquete optando por preámbulos más cortos.

Cabe destacar que un aspecto que afecta el desempeño del sistema, en particular la tasa de error de bit, es la estructura de paquete. Por ello, se realizaron simulaciones para obtener la pérdida de SNR dada la cantidad de símbolos pilotos consecutivos y la cantidad de símbolos de datos entre pilotos, en otras palabras, la tasa de actualización del algoritmos de estimación de fase de la portadora. Se decidió por una configuración de 8 datos cada 1 piloto, modificable mediante los registros de configuración.

Para comprobar el procesador implementado en FPGA se estimó la tasa de error de bit para distintos valores de razón señal a ruido y se comparó con simulaciones realizadas en MATLAB. Para poder realizar simulaciones de Monte Carlo en *hardware* fue necesario implementar un emulador de canal que replicara las condiciones de las simulaciones hechas en el computador. Por esa razón, fue necesario crear un generador de variables aleatorias en base a la evaluación de la función de distribución inversa gaussiana acumulada y segmentación jerárquica. Se obtuvo una disminución del 95 % del tiempo de simulación por FPGA con respecto a la simulación por código.

El procesador diseñado tiene una pérdida de razón señal a ruido de 2 decibeles comparada con el caso de una transmisión QPSK ideal en un canal aditivo aleatorio gaussiano. La pérdida de desempeño se debe principalmente a la varianza del estimación fina del desplazamiento de portadora produciendo un valor residual de CFO el cual debe ser corregido por el estimador de desfase de la portadora. La arquitectura propuesta permite modificar ambos bloques a futuro, en caso de querer mejorar dicho aspecto, sin tener que entrar a conocer al detalle todo el procesador. Sin embargo, el desempeño obtenido y cuantificado es un resultado suficiente para poder corroborar de forma empírica los beneficios de uso de múltiples antenas.

En resumen, la contribución principal es la implementación de un procesador banda-base, como plataforma de investigación, utilizando algoritmos de baja complejidad desarrollados en LATINA, que permite comunicaciones MIMO de ancho de banda angosto.

8.2 Trabajo futuro

En base a lo realizado en esta tesis se sugieren las siguientes líneas de trabajo:

- Crear una estimación del número de compuertas lógicas necesarias para implementar los diferentes algoritmos de estimación de CFO en la literatura. Verificar la ganancia respecto a la varianza en comparación a los recursos utilizados.
- Obtener la curva de la tasa de error de bit según la SNR, mediante el uso de los módulos de RF. Comprobar si existe alguna pérdida de desempeño o fenómeno que no haya sido considerado en las simulaciones.
- Investigar sobre diferentes métodos de estimación de la razón señal a ruido implementables en *hardware*.
- Usando dos plataformas, medir las variaciones del canal bajo distintos escenarios y verificar los límites de los supuestos sobre el ancho de banda de coherencia y reciprocidad del canal.

8.3 Observaciones sobre la plataforma

En esta última sección se presentan algunas observaciones sobre mejoras que podrían ser integradas a la plataforma:

- El *testbed* no posee puertos típicos para posicionar nuevos *hardware* o módulos RF *front-ends*. La mayoría de las plataformas comerciales utilizan buses como USB, PCI o CP. En futuras mejoras de la plataforma se debe tener en cuenta las especificaciones de “PCI Alliance”(García-Naya et al., 2008).
- Considerar implementar una interfaz SPI y reemplazarla por el módulo de comunicación ethernet. Agregar la inteligencia adicional, de ser necesario, tal que el procesador pueda ser usado mediante cualquier interfaz de comunicación digital.
- Reestructurar la plataforma con el objetivo de que la FPGA con la cual se procesan los datos sea fácil de cambiar. De esta forma se podría aumentar la capacidad de procesamiento sin tener que modificar demasiado el *testbed*.

BIBLIOGRAFÍA

Aldunate, J. (2013). *Reporte técnico RT2 RF Discreto (Cangrejo)* (Inf. Téc. n.º v1.0). LATINA UC.

Andersen, J., Rappaport, T., y Yoshida, S. (1995, Jan). Propagation measurements and models for wireless communications channels. *Communications Magazine, IEEE*, 33(1), 42-49. doi: 10.1109/35.339880

Boutillon, E., Tang, Y., Marchand, C., y Bomel, P. (2010, June). Hardware discrete channel emulator. En *High Performance Computing and Simulation (HPCS), 2010 International Conference on* (p. 452-458). doi: 10.1109/HPCS.2010.5547099

Box, G. E., y Muller, M. E. (1958). A note on the generation of random normal deviates. *The annals of mathematical statistics*(29), 610–611.

Cherubini, G., y Benvenuto, N. (2003). *Algorithms for communications systems and their applications*. Wiley.

Chhikara, R. (1988). *The inverse gaussian distribution: Theory: Methodology, and applications* (Vol. 95). CRC Press.

Chockalingam, A., y Rajan, B. S. (2014). *Large mimo systems*. Cambridge University Press.

Cui, S., Goldsmith, A., y Bahai, A. (2004, Aug). Energy-efficiency of mimo and cooperative mimo techniques in sensor networks. *Selected Areas in Communications, IEEE Journal on*, 22(6), 1089-1098. doi: 10.1109/JSAC.2004.830916

Feres, C. (2013). *Algoritmos de adquisición para redes inalámbricas de sensores con tecnología de múltiples antenas* (Tesis de Magíster). Pontificia Universidad Católica de Chile.

Fitz, M. (1991, Dec). Planar filtered techniques for burst mode carrier synchronization. En *Global Telecommunications Conference, 1991. GLOBECOM '91. 'Countdown to the New Millennium. Featuring a Mini-Theme on: Personal Communications Services* (p. 365-369 vol.1). doi: 10.1109/GLOCOM.1991.188412

Garcia-Naya, J. A., González-López, M., y Castedo, L. (2008). An overview of MIMO testbed technology. En *Proceedings of the 4th International Symposium on Image and Video Communications over Fixed and Mobile Networks (ISIVC'08)*.

Goldsmith, A. (2005). *Wireless communications*. Cambridge university press.

Instruments, T. (2006). Cc2420: 2.4 ghz ieee 802.15. 4/zigbee-ready rf transceiver. Available <http://www.ti.com/lit/gpn/cc2420>, 53.

Kaulen, D. (2015a). *CFO and Timing Loops Documentation* (Inf. Téc.). LATINA UC.

Kaulen, D. (2015b). *Medium Acces Control for MIMO Wireless Sensor Networks* (Tesis de Magíster). Pontificia Universidad Católica de Chile.

Kay, S. (1989, Dec). A fast and accurate single frequency estimator. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 37(12), 1987-1990. doi: 10.1109/29.45547

Kettlun, F. (2014). *SVD-Based Beamforming communications over narrow-band quasi-static MIMO channels* (Tesis de Magíster). Pontificia Universidad Católica de Chile.

Lee, D.-U., Cheung, R., Villasenor, J., y Luk, W. (2006, Dec). Inversion-based hardware gaussian random number generator: A case study of function evaluation via hierarchical segmentation. En *Field Programmable Technology, 2006. FPT 2006. IEEE International Conference on* (p. 33-40). doi: 10.1109/FPT.2006.270388

Lee, D.-U., Luk, W., Villasenor, J., Zhang, G., y Leong, P. (2005, Aug). A hardware gaussian noise generator using the wallace method. *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on*, 13(8), 911-920. doi: 10.1109/TVLSI.2005.853615

Lee, D.-U., Villasenor, J., Luk, W., y Leong, P. (2006, June). A hardware gaussian noise generator using the box-muller method and its error analysis. *Computers, IEEE Transactions on*, 55(6), 659-671. doi: 10.1109/TC.2006.81

Liu, W., Li, X., y Chen, M. (2005, March). Energy efficiency of MIMO transmissions in wireless sensor networks with diversity and multiplexing gains. En *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on* (Vol. 4, p. iv/897-iv/900 Vol. 4). doi: 10.1109/ICASSP.2005.1416154

Luise, M., y Reggiannini, R. (1995, Feb). Carrier frequency recovery in all-digital modems for burst-mode transmissions. *Communications, IEEE Transactions on*, 43(2/3/4), 1169-1178. doi: 10.1109/26.380149

Mengali, U., y D'Andrea, A. N. (1997). *Synchronization techniques for digital receivers*. Springer Science & Business Media.

Meyer, H., Moeneclaey, M., y Fechtel, S. A. (1997). *Digital communications receivers*. Wiley-Interscience.

Middleton, G. B. (2007). *On the impact of receiver quantization on the performance of wireless communication systems* (Tesis Doctoral no publicada). Rice University.

Nagaraj, S., Khan, S., Schlegel, C., y Burnashev, M. (2009, Feb). Differential preamble detection in packet-based wireless networks. *Wireless Communications, IEEE Transactions on*, 8(2), 599-607. doi: 10.1109/TWC.2009.071169

Noneaker, D., Raghavan, A., y Baum, C. (2001, Jul). The effect of automatic gain control on serial, matched-filter acquisition in direct-sequence packet radio communications. *Vehicular Technology, IEEE Transactions on*, 50(4), 1140-1150. doi: 10.1109/25.938588

Pérez, H. (2012). *Diseño e implementación en fpga de un módem de banda base para redes inalámbricas de sensores* (Tesis de Magíster). Pontificia Universidad Católica de Chile.

Proakis, J. (2001). *Digital communications*. McGraw-Hill. Descargado de <https://books.google.cl/books?id=sbr8QwAACAAJ>

Rao, R. M., Zhu, W., Lang, S., Oberli, C., Browne, D., Bhatia, J., ... others (2004). Multi-antenna testbeds for research and education in wireless communications. *Communications Magazine, IEEE*, 42(12), 72–81.

Razavi, B. (1998). *RF microelectronics* (Vol. 1). Prentice Hall New Jersey.

Rosas, F., y Oberli, C. (2012, Sept). Energy-efficient MIMO SVD communications. En *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on* (p. 1588-1593). doi: 10.1109/PIMRC.2012.6362601

Rupp, M., Burg, A., y Beck, E. (2003). Rapid prototyping for wireless designs: the five-ones approach. *Signal Processing*, 83(7), 1427–1444.

Rupp, M., Mehlführer, C., Caban, S., Langwieser, R., Mayer, L. W., y Scholtz, A. L. (2006). Testbeds and rapid prototyping in wireless system design. *EURASIP Newsletter*, 17(3), 32–50.

Saito, H., Kagami, O., Umehira, M., y Kado, Y. (2008, December). Wide area ubiquitous network: the network operator's view of a sensor network. *Communications Magazine, IEEE*, 46(12), 112-120. doi: 10.1109/MCOM.2008.4689217

Villers-Grandchamps, J. D. (2014). *Diseño e implementación de un módem digital para redes inalámbricas de sensores mimo compatible* (Tesis de Magíster). Pontificia Universidad Católica de Chile.

Wallace, C. S. (1996). Fast pseudorandom generators for normal and exponential variates. *ACM Transactions on Mathematical Software (TOMS)*, 22(1), 119–127.

ANEXOS

A. PRIMER ANEXO

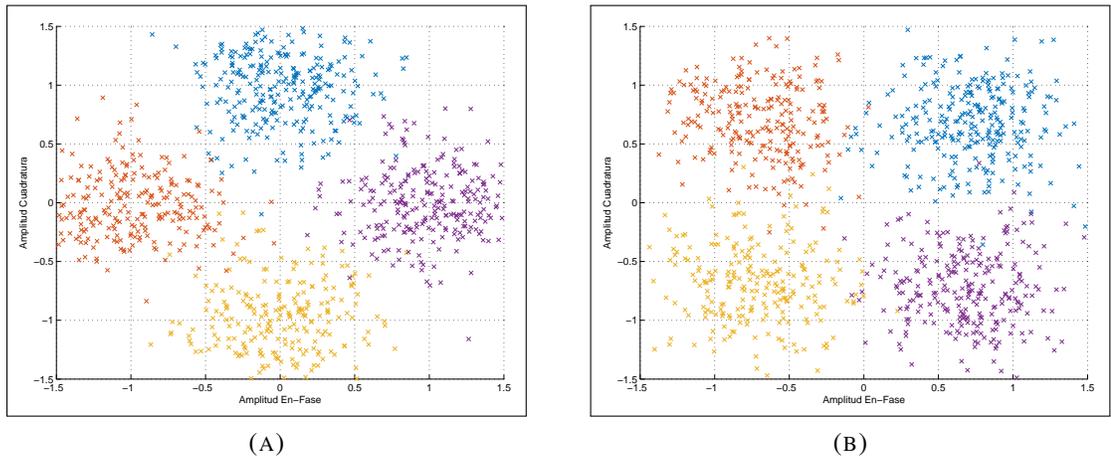


FIGURA A.1. Recepción de símbolos bajo un canal AWGN con SNR de 4 dB y un desfase de $\pi/4$ (A) Diagrama de dispersión antes de la corrección de fase (B) Diagrama de dispersión después de la corrección de fase.