



**UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA**

**PROGRAMACIÓN DE UNA PLACA ROACH COMO UN CORRELADOR PARA
UN INTERFERÓMETRO CERCANO AL INFRARROJO**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

PEDRO ABRAHAM SÁNCHEZ MORALES

**PROFESOR GUÍA:
ERNEST A. MICHAEL**

**MIEMBROS DE LA COMISIÓN
CLAUDIO ESTEVEZ MONTERO
MARCOS ORCHARD CONCHA**

**SANTIAGO DE CHILE
ABRIL 2013**

**RESUMEN DE LA MEMORIA
PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL ELÉCTRICO
PEDRO SÁNCHEZ MORALES
ABRIL 2013**

El presente trabajo de memoria se enmarca en el ámbito de la radioastronomía, específicamente en la interferometría heterodina, procedimiento por el cual varias antenas funcionan como si fueran una sola de gran tamaño; siendo el experimento realizado en la Universidad de Chile sólo con dos antenas. El objetivo fundamental del proyecto es lograr programar un correlador, dispositivo que se encargará de correlacionar las señales de las antenas en una FPGA montada en una placa, la cual fue diseñada exclusivamente para esta tarea: la placa ROACH.

El proceso de diseño del correlador se hará mediante herramientas desarrolladas por el grupo CASPER, las cuales tienen como función principal la programación de la ROACH mediante la compilación de diversos modelos de correlador diseñados en MATLAB-Simulink®. Se implementaron dos modelos, uno obtenido a partir de realizar modificaciones a un modelo propuesto por CASPER, y el otro correspondiente a un espectrómetro que realiza internamente la correlación entre las señales, para luego entregar el espectro observado por las antenas.

Las pruebas en laboratorio pudieron realizarse sólo sobre el primer diseño, ya que para probar el espectrómetro se necesita hardware adicional que no se encuentra en el laboratorio; razón por la cual su uso quedará propuesto para cuando se cuente con el hardware necesario.

El correlador se sometió a diversos experimentos basados en tres montajes, entre los cuales uno es de naturaleza óptica, para verificar la funcionalidad y caracterizar el comportamiento de éste.

Los resultados de las primeras pruebas hechas al correlador funcional, revelaron que la mayoría del montaje no estaba aislado del ruido externo; además, las señales generadas para efecto de probar el sistema estaban contaminadas con un alto nivel de ruido blanco producido desde la fuente. Aun así, los espectros analizados tanto por su auto-correlación, correlación cruzada y fase de las señales, demuestran que el correlador se encuentra operando como es debido.

Luego se procedió al segundo experimento, el cual presentó como objetivo principal eliminar la mayor cantidad de componentes externas de ruido; estudiando así el comportamiento del correlador bajo condiciones cercanas a las ideales. Los resultados arrojados de este segundo montaje reafirmaron el correcto funcionamiento del sistema.

Al realizar el tercer y último experimento, se pudo apreciar el funcionamiento del diseño sobre un montaje óptico, cuya finalidad fue demostrar el correcto funcionamiento del detector heterodino en conjunto, sobre condiciones cercanas a las pruebas de campo en una primera aproximación.

Finalmente se puede concluir que el objetivo principal se cumple, y que existen diversos factores que afectaron en los resultados pero sin tener consecuencia directa en el funcionamiento del correlador. Aun así existen ciertas recomendaciones para disminuir los niveles de ruido y obtener espectros en ausencia de componentes externas al experimento.

Tabla de Contenido

1.	Introducción	1
1.1.	Motivación	1
1.2.	Objetivos	1
1.3.	Estructura de la Memoria.....	1
2.	Contextualización.....	3
2.1.	Radioastronomía	3
2.1.1.	Fuentes de Emisión Electromagnética	4
2.1.2.	Sistemas de Recepción	6
2.1.3.	Sistemas de Análisis	11
2.1.4.	Correlador	14
2.1.5.	Desarrollo de la Radioastronomía.....	17
2.1.6.	Marco del Experimento.....	18
2.2.	Infraestructura Disponible	19
2.2.1.	Espacio Físico.....	19
2.2.2.	Hardware.....	20
2.2.2.1.	Placa ROACH.....	20
2.2.2.2.	Convertor Análogo Digital.....	22
2.2.2.3.	Instrumentos de Análisis y Generación de Señales.....	24
2.3.	Metodología	24
2.3.1.	Programación de Hardware	25
2.3.2.	Obtención de Datos.....	26
2.4.	Reseña Bibliográfica	26
3.	Implementación	28
3.1.	Estructura General	28
3.1.1.	Convertor Análogo Digital (ADC).....	28
3.1.2.	Banco de Filtros Polifásicos (PFB).....	28
3.1.3.	Proceso de Multiplicar/Acumular (MAC)	29
3.1.4.	Registro	30
3.2.	Diseño del Correlador	30
3.2.1.	Correlador de 700 [MHz].....	30

3.2.1.1.	Implementación del Modelo en MATLAB-Simulink®	30
3.2.1.2.	Software de Control	34
3.2.2.	Espectrómetro VEGAS.....	35
3.2.2.1.	Funcionamiento del espectrómetro.....	35
3.2.2.2.	Análisis del Modelo en MATLAB-Simulink®	36
3.2.2.3.	Software de Control	39
3.3.	Prueba de los Diseños	43
3.3.1.	Hardware a Utilizar en los Experimentos.....	44
3.3.2.	Montaje de los Experimentos	46
3.3.3.	Condiciones de las Pruebas.....	48
3.3.4.	Procedimiento de Ejecución de Pruebas.....	51
4.	Análisis de Resultados.....	52
4.1.	Primer Montaje	52
4.2.	Segundo Montaje.....	61
4.3.	Tercer Montaje.....	70
5.	Conclusión.....	74
5.1.	Conclusiones.....	74
5.2.	Soluciones a Implementar.....	75
5.3.	Trabajos Propuestos.....	75
6.	Bibliografía	77
7.	Anexos.....	79

Índice de Figuras

Figura 1: Recreación artística de Nuestra Galaxia (NASA/JPL-Caltech).	3
Figura 2: Espectro Electromagnético.	4
Figura 3: Espectro de emisión del Hierro.	4
Figura 4: Relación entre frecuencia y brillo de un “cuerpo negro” según la ley de Planck (Wilson & Rohlfs, 2009).	5
Figura 5: Curva normalizada de Planck junto a las curvas de Wien y de Rayleigh-Jeans (Wilson & Rohlfs, 2009).	6
Figura 6: Esquema básico de un receptor heterodino óptico (COSETI, 1990).	6
Figura 7: (a) Antena móvil tipo Vertex usada en ALMA. (b) Antena fija usada en el observatorio de Arecibo.	8
Figura 8: Tipo de OMT.	9
Figura 9: Amplificador HEMT de bajo ruido (LNA) utilizado en ALMA entre 4-8 [GHz].	10
Figura 10: Receptor con 2 salidas de distinta polarización en Frecuencia Intermedia (IF) (Wilson & Rohlfs, 2009).	10
Figura 11: Imagen que ilustra diferencia de tamaños del espejo del E-ELT versus el VLT (ESO, 2011).	11
Figura 12: Arreglo compacto de ALMA, donde cada antena es representada por un círculo (ALMA, 2011).	11
Figura 13: A la derecha se tiene la configuración de 6 antenas y a la izquierda su respectivo plano UV.	12
Figura 14: Esquema de dos antenas actuando como un interferómetro.	12
Figura 15: (a) Frente de ondas perpendicular a las antenas. (b) Frente de ondas con desfase temporal en la recepción.	13
Figura 16: Interferómetro de dos antenas con señal retardada.	14
Figura 17: Esquema de un correlador (ALMA, 2011).	14
Figura 18: Diagrama de bloques de un correlador tipo XF (NCRA, 2012).	15
Figura 19: Diagrama de bloques de un correlador tipo FX (NCRA, 2012).	15
Figura 20: Configuración de las antenas de ALMA. A la izquierda se aprecia el ACA (ALMA, 2011).	17
Figura 21: Esquema del experimento óptico usando interferometría heterodina.	18
Figura 22: Placa ROACH.	20
Figura 23: Diagrama de bloques de la placa ROACH con sus distintos componentes y conexiones (CASPER Website).	21
Figura 24: Diagrama de bloques del conversor ADC083000 (National Instruments).	23
Figura 25: Placa ROACH con 2 tarjetas ADC montadas en la izquierda de la imagen.	23
Figura 26: Diagrama de bloques de un espectrómetro hecho en MATLAB-Simulink®.	25
Figura 27: Software de control del sintetizador incluido en la ROACH.	26
Figura 28: Sitio web de CASPER.	27
Figura 29: Diagrama de bloques del diseño general de un correlador.	28
Figura 30: División de una señal en 4 <i>taps</i>	29
Figura 31: Representación de los bloques PFB FIR y de FFT en MATLAB-Simulink®.	29

Figura 32: Bloques de comunicación y generador de señales implementados en MATLAB-Simulink®	31
Figura 33: Representación del bloque ADC en MATLAB-Simulink®	31
Figura 34: Bancos de Filtros Polifásicos implementados en MATLAB-Simulink®	31
Figura 35: Etapa MAC implementada en MATLAB-Simulink®	32
Figura 36: Registros RMS implementados en MATLAB-Simulink®	32
Figura 37: Detector de saturación implementado en MATLAB-Simulink®	33
Figura 38: Subsistema de control de LEDs implementado en MATLAB-Simulink®	33
Figura 39: Esquema de buffers de memoria para modo HB (izquierda) y modo LB (derecha).	35
Figura 40: Formato SPEAD de los paquetes en modo HB.	36
Figura 41: Bloque “stokes” que realiza las multiplicaciones entre las 8 señales resultantes de la FFT.	36
Figura 42: Uno de los 4 acumuladores implementados en MATLAB-Simulink®	37
Figura 43: Bloque encargado de concatenar las salidas de los acumuladores.	37
Figura 44: Subsistema de reestructuración implementado en MATLAB-Simulink®	38
Figura 45: Subsistema de formateo SPEAD implementado en MATLAB-Simulink®	38
Figura 46: Subsistema de direccionamiento de la data al puerto de 10Gbe en la ROACH.....	38
Figura 47: Ventana de monitoreo del software de VEGAS.	42
Figura 48: Gráfico de correlación cruzada.	43
Figura 49: Uno de los sintetizador de frecuencias para generar señal de prueba.....	44
Figura 50: Amplificador LNA de 39 [dB] de ganancia.....	45
Figura 51: Sintetizador láser de precisión, con longitud de onda ajustable en pasos de 1 [pm].....	45
Figura 52: Esquema del primer experimento.....	46
Figura 53: Esquema del segundo montaje.....	46
Figura 54: Esquema del tercer montaje.....	47
Figura 55: Montaje de experimento óptico implementado en el laboratorio de Fotónica.....	48
Figura 56: Divisor T simple implementado con conectores SMA/BNC.	49
Figura 57: Espectro obtenido generando un tono a 200 [MHz].	49
Figura 58: Espectro obtenido generando un tono a 400 [MHz].	50
Figura 59: Espectro obtenido generando un tono a 600 [MHz].	50
Figura 60: Auto-correlación del ruido interno de ambos ADC.....	52
Figura 61: Auto-correlación del ruido externo amplificado.....	53
Figura 62: Correlación cruzada y fase del ruido externo amplificado.....	54
Figura 63: Auto-correlación de una señal de entrada a 200 [MHz].	55
Figura 64: Correlación cruzada y fase de una señal de entrada a 200 [MHz].	56
Figura 65: Auto-correlación de una señal de entrada a 300 [MHz].	57
Figura 66: Correlación cruzada y fase de una señal de entrada a 300 [MHz].	57
Figura 67: Auto-correlación de una señal de entrada a 400 [MHz].	58
Figura 68: Correlación cruzada y fase de una señal de entrada a 400 [MHz].	58
Figura 69: Auto-correlación de una señal de entrada a 500 [MHz].	59
Figura 70: Correlación cruzada y fase de una señal de entrada a 500 [MHz].	59
Figura 71: Auto-correlación de una señal de entrada a 600 [MHz].	60

Figura 72: Correlación cruzada y fase de una señal de entrada a 600 [MHz].	60
Figura 73: Auto-correlación de ruido blanco en ambos ADC.	61
Figura 74: Correlación cruzada y fase de ruido blanco.	62
Figura 75: Auto-correlación de una señal de entrada a 100 [MHz].	63
Figura 76: Correlación cruzada y fase de una señal de entrada a 100 [MHz].	63
Figura 77: Auto-correlación de una señal de entrada a 300 [MHz].	64
Figura 78: Correlación cruzada y fase de una señal de entrada a 300 [MHz].	64
Figura 79: Auto-correlación de una señal de entrada a 500 [MHz].	65
Figura 80: Correlación cruzada y fase de una señal de entrada a 500 [MHz].	65
Figura 81: Auto-correlación de una señal de entrada a 100 [MHz].	66
Figura 82: Correlación cruzada y fase de una señal de entrada a 100 [MHz].	67
Figura 83: Auto-correlación de una señal de entrada a 200 [MHz].	67
Figura 84: Correlación cruzada y fase de una señal de entrada a 250 [MHz].	68
Figura 85: Auto-correlación de una señal de entrada a 400 [MHz].	68
Figura 86: Correlación cruzada y fase de una señal de entrada a 400 [MHz].	69
Figura 87: Auto-correlación de una señal mezclada con una diferencia en longitud de onda de 5 [pm].	70
Figura 88: Correlación cruzada de una señal mezclada con una diferencia en longitud de onda de 5 [pm].	71
Figura 89: Auto-correlación de una señal mezclada con una diferencia en longitud de onda de 10 [pm].	71
Figura 90: Correlación cruzada de una señal mezclada con una diferencia en longitud de onda de 10 [pm].	72
Figura 91: Espectro del tono generado por el sistema a la salida de los amplificadores.	73

Lista de Acrónimos

ADC: Analog-to-Digital Converter.
ALMA: Atacama Large Millimeter/submillimeter Array.
CASPER: Collaboration for Astronomy Signal Processing and Electronics Research.
CMOS: Complementary Metal-Oxide Semiconductor.
CPU: Central Processing Unit.
DAS: Departamento de Astronomía.
DFT: Discrete Fourier Transform.
DIE: Departamento de Ingeniería Eléctrica.
E-ELA: European Extremely Large Telescope.
FCFM: Facultad de Ciencias Físicas y Matemáticas.
FFT: Fast Fourier Transform.
FIR: Finite Impulse Response.
FITS: Flexible Image Transport System.
FPGA: Field-Programmable Gate Array.
GBT: Green Bank Telescope.
GPU: Graphics Processing Unit.
HB: High Bandwidth.
HDL: Hardware Description Language.
HEMT: High-Electron-Mobility Transistor
HPC: Hardware of PC.
IF: Intermediate Frequency.
LB: Low Bandwidth.
LNA: Low Noise Amplifier.
MAC: Multiply and Accumulate.
OL: Oscilador Local.
OMT: Ortho-Mode Transducer.
PC: Personal Computer.
PFB: Polyphase Filter Bank.
PLL: Phase-Locked Loop.
RF: Radio Frequency.
ROACH: Reconfigurable Open Architecture Computing Hardware.
SIS: Superconductor-Insulator-Superconductor.
VEGAS: VErsatile GBT Astronomical Spectrometer.
VLT: Very Large Telescope.

1. Introducción

El trabajo de memoria se realiza en el marco de desarrollos asociados a la disciplina de la Radioastronomía, y consiste principalmente en implementar un correlador para un interferómetro que operará en una banda de frecuencia cercana al infrarrojo. Durante el desarrollo de la memoria, se expondrán los conceptos necesarios para entender la finalidad del trabajo y el marco en el que se encuentra. Cabe destacar que el hardware usado para la implementación del proyecto es de carácter innovador, transformando a la Universidad de Chile en una de las dos instituciones del país en usar dicha tecnología.

1.1. Motivación

En los últimos años, hubo un considerable desarrollo en el ámbito de la Radioastronomía y un gran interés por parte de la Astronomía en la observación del Universo en frío. Lo anterior ha llevado a numerosos países desarrollar radiotelescopios de mayor alcance y precisión; junto con el desarrollo de la tecnología necesaria para la interpretación de los datos recibidos, tanto en hardware como en software. Uno de los proyectos que ha contado con el apoyo internacional para su ejecución ha sido el proyecto ALMA, el cual está actualmente en operación en nuestro país. La tecnología impulsada por este proyecto a generado grandes oportunidades para el desarrollo de ideas científicas. Dentro de éstas se encuentra el experimento realizado por el Departamento de Ingeniería Eléctrica en conjunto con el Departamento de Astronomía, que consiste básicamente en realizar un sistema de detección heterodina, montando un interferómetro de dos telescopios, con el fin de obtener mediciones espectrales con una alta resolución y de muy bajo costo de implementación. Dentro de los componentes que participan en el montaje de dicho dispositivo de detección heterodina se encuentra el correlador, el cual se encargará principalmente de realizar la correlación entre los telescopios que participan en el interferómetro. Este equipo será diseñado e implementado en el proceso de la memoria.

1.2. Objetivos

Dentro de los objetivos de la memoria, el principal se define como el diseño y la implementación de un correlador en una placa ROACH para un interferómetro cercano al infrarrojo. Los objetivos secundarios son establecer un procedimiento detallado para la instalación del software necesario para el control de los modelos de correlador y aprender a utilizar el software asociado al desarrollo en base a la ROACH, destacando tanto su funcionamiento como además, su proceso de instalación.

1.3. Estructura de la Memoria

El primer capítulo entrega una breve introducción al tema, destacando los objetivos y la estructura de ésta. El segundo capítulo se desarrolla contextualizando la memoria desde un nivel global hasta el experimento en el cual participará el correlador una vez diseñado y probado, pasando por entregar el marco teórico necesario para la comprensión de los procesos asociados al experimento. También se entregará información sobre la infraestructura disponible para el proyecto, la metodología a usar en el proceso de implementación del correlador para finalizar con

la reseña bibliográfica indicando el estado del arte de las bases en las que se dará forma al proyecto. El tercer capítulo trata sobre la implementación del correlador, detallando la estructura general y explicando el proceso de diseño de éste. También se discutirá en que consiste el procedimiento para evaluar la correcta operación del correlador, el montaje del experimento y las condiciones en las que serán llevadas a cabo las pruebas. En el cuarto capítulo se exponen y analizan los resultados obtenidos durante las pruebas, separando el contenido de acuerdo a los tres tipos de montajes usados para llevarlas a cabo. El quinto capítulo constituye principalmente la conclusión de la memoria, además de plantear las posibles soluciones a los problemas encontrados durante el proceso de pruebas, y también se expondrán ideas de trabajos futuros a realizar. El Capítulo 6 expone la bibliografía usada para el diseño del marco teórico y las fuentes que fueron usadas para extraer imágenes e información sobre el proyecto. El capítulo final, de anexos, contiene los códigos desarrollados en Python para su uso en el proceso de implementación del correlador.

2. Contextualización

El hombre desde el comienzo de los tiempos se ha preguntado qué hay más allá del cielo, de dónde venimos y hacia dónde vamos. Con el paso del tiempo ha desarrollado numerosas herramientas para poder responder de alguna forma dichas interrogantes; siendo una de ellas la astronomía: ciencia que estudia el Universo, su comportamiento y estructura, para poder comprender mejor el lugar donde la Tierra se encuentra. Y a su vez, esta ciencia ha ido evolucionando en la forma de observar el cielo, llegando a desarrollar una rama conocida como la radioastronomía. Como se ha descrito en el Capítulo 1, el trabajo de la memoria se enmarca principalmente en el ámbito de la observación radioastronómica y posteriormente se procederá a explicar en qué consiste y cómo se lleva a cabo.

2.1. Radioastronomía

Desde que se realizan observaciones astronómicas, la principal fuente de obtención de datos ha sido el rango visible del espectro electromagnético, dado que la instrumentación disponible y la teoría vigente fortalecen dicha forma de medición y obtención de información sobre el Universo. Pero a medida que la tecnología fue avanzando, comenzó a ampliarse de forma significativa el área del tratamiento de señales electromagnéticas, y se tuvo que para comienzos de 1930 empezaron a detectarse por primera vez ondas de radio de origen extraterrestre. Fue allí que Karl Guthe Jansky realizó una de las primeras investigaciones de origen radioastronómico; donde primero detectó que una de las fuentes de emisión de ondas de radio era el centro de nuestra Galaxia, seguido por el Sol.



Figura 1: Recreación artística de Nuestra Galaxia (NASA/JPL-Caltech).

Entonces, la radioastronomía se puede definir como la disciplina que realiza observaciones en el rango de radio del espectro electromagnético, es decir, las señales recibidas oscilan entre los 10[MHz] y 950[GHz], abarcando dentro de esta ventana las microondas y el infra-rojo cercano.

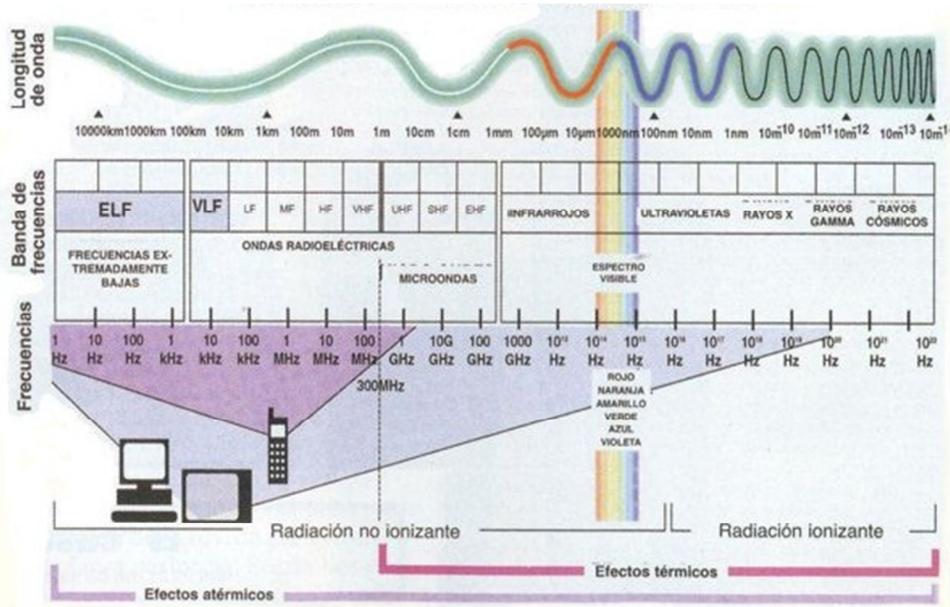


Figura 2: Espectro Electromagnético.

2.1.1. Fuentes de Emisión Electromagnética

Una fuente de emisión es particularmente un cuerpo celeste que emite ondas electromagnéticas en todo el espectro o en una parte de él. En cuanto a la radioastronomía, los cuerpos de interés son:

- Estrellas
- Galaxias
- Nubes moleculares
- Zonas de formación de estrellas
- Discos galácticos

El espectro se puede definir como la “firma electromagnética” del cuerpo celeste, el cual permite identificar los procesos químicos y electromagnéticos que ocurren en el momento que se ha realizado la observación. También se pueden determinar los componentes químicos que participan en las reacciones observadas, junto con las condiciones en donde éstas están llevándose a cabo. En la Figura 3 se puede observar un ejemplo del espectro de emisión del Hierro, y se pueden apreciar las líneas producidas a distintas frecuencias.

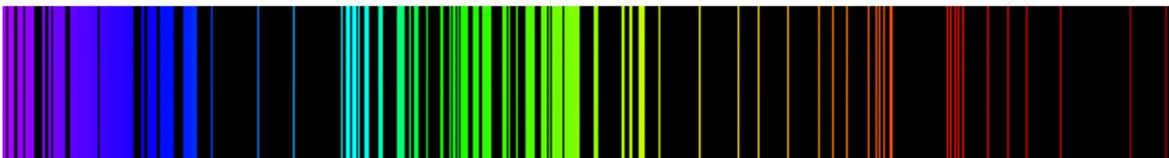


Figura 3: Espectro de emisión del Hierro.

Ahora bien, una de las principales características de las ondas electromagnéticas emitidas por dichos cuerpos viene dada por la Ley de Planck, que relaciona la intensidad del brillo del “cuerpo negro”, con su temperatura y la frecuencia de la radiación emitida; tal como se puede observar en la Figura 4. Existen otros modelos que incluyen algunas correcciones para casos específicos, como los son la Ley de Wien y la Ley de Rayleigh-Jeans.

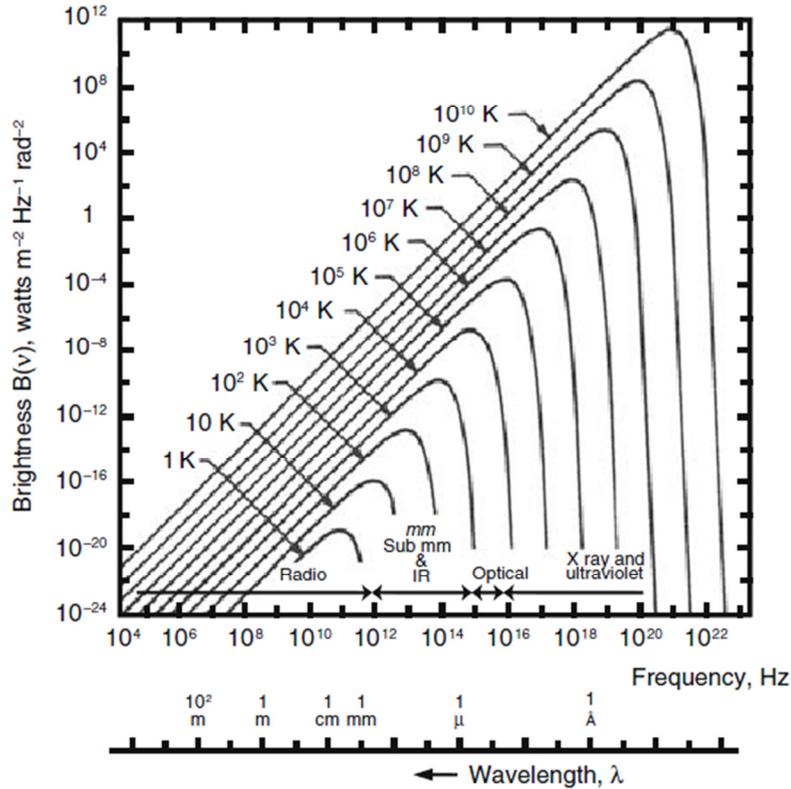


Figura 4: Relación entre frecuencia y brillo de un “cuerpo negro” según la ley de Planck (Wilson & Rohlfs, 2009).

Como fue mencionado, el espectro de la figura anterior puede ser obtenido de acuerdo a la Ley de Planck:

$$B_{\nu}(T) = \frac{2h\nu^3}{c^2} \frac{1}{e^{h\nu/kT} - 1}.$$

En radioastronomía se usa una muy buena aproximación, ya mencionada como la Ley de Rayleigh-Jeans. Si se cumple que $h\nu \ll kT$, se puede hacer una expansión de Taylor de segundo orden, quedando la ecuación de Plank de la forma:

$$B_{\nu}(T) = \frac{2\nu^2}{c^2} kT,$$

donde estrictamente esta aproximación requiere que $\nu [\text{GHz}] \ll 20T[\text{K}]$.

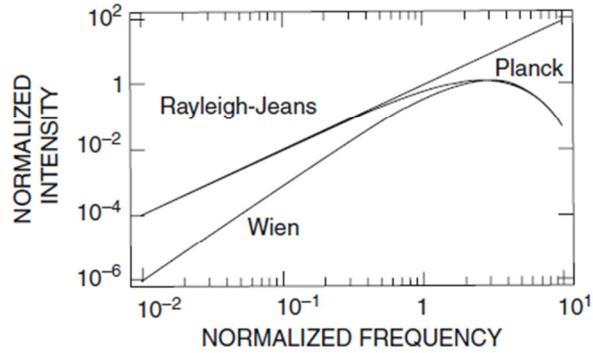


Figura 5: Curva normalizada de Planck junto a las curvas de Wien y de Rayleigh-Jeans (Wilson & Rohlf, 2009).

En la práctica, al observar el espectro continuo de emisión del cuerpo celeste, no sólo se ve la radiación producida por la temperatura del objeto observado, sino que también existen líneas espectrales producto de eventos energéticos no relacionados con la temperatura; efectos de interacción electromagnéticas de los electrones con su entorno, como el efecto Zeeman (origen magnético) y el efecto Stark (origen eléctrico); corrimientos de la frecuencia producidos por el efecto Doppler a nivel local y el *redshift* debido a la expansión del Universo a gran escala.

2.1.2. Sistemas de Recepción

En contexto del trabajo de la memoria, sólo se describirá el concepto de detección heterodina. Ésta consiste en un receptor que recibe una señal de alta frecuencia (RF) y se encarga de disminuirla a un rango aceptable amplificándola, para después ser procesada y analizada; dado que electrónicamente es más sencillo trabajar con señales de menor frecuencia. Esto se logra mediante un mezclador (Mixer), el cual tiene una señal de referencia provista por un oscilador local (OL) y una señal de entrada recibida desde una fuente de radio-emisión. El mezclador debe tener una respuesta no lineal frente a la amplitud, ya que en dicho caso, al hacer una expansión de Taylor sobre la relación corriente-voltaje del mezclador junto con la señal de entrada, se pueden determinar de manera sencilla las componentes armónicas de la señal resultante de la mezcla. El esquema en bloques es el siguiente:

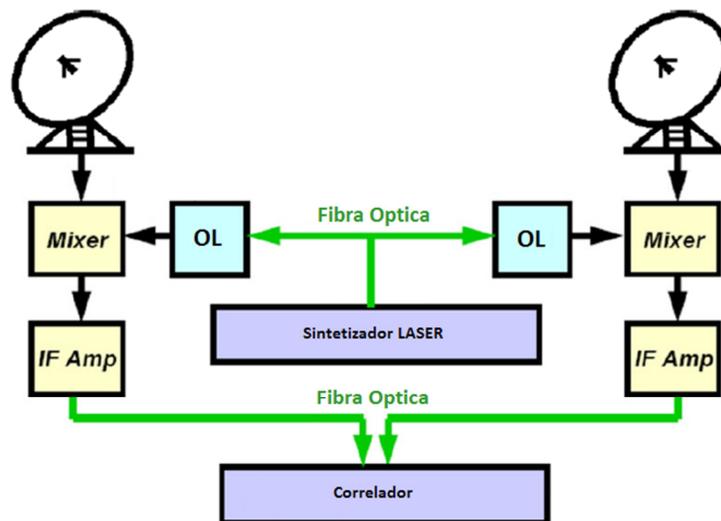


Figura 6: Esquema básico de un receptor heterodino óptico (COSETI, 1990).

Se puede observar que en el mezclador ambas señales se suman, lo cual algebraicamente será definido a continuación, teniendo en cuenta algunos supuestos sobre las señales que interactúan en el sistema.

Primero supondrá que la señal recibida es de la forma $E_{sig} \cos(\omega_{sig}t + \varphi)$, donde E_{sig} , ω_{sig} y φ son respectivamente la amplitud, frecuencia y fase de dicha señal. Luego se tiene que la señal de referencia producida por el oscilador local es $E_{LO} \cos(\omega_{LO}t)$, donde E_{LO} es la amplitud y ω_{LO} es la frecuencia de la señal de del oscilador local.

Se tiene que la señal de salida I_{out} del detector es directamente proporcional al cuadrado de la amplitud, es decir $I_{out} \propto (E_{sig} \cos(\omega_{sig}t + \varphi) + E_{LO} \cos(\omega_{LO}t))^2$, entonces:

$$I_{out} = \frac{E_{sig}^2}{2} (1 + \cos(2\omega_{sig}t + 2\varphi)) + \frac{E_{LO}^2}{2} (1 + \cos(2\omega_{LO}t)) + E_{sig}E_{LO} [\cos((\omega_{sig} + \omega_{LO})t + \varphi) + \cos((\omega_{sig} - \omega_{LO})t + \varphi)].$$

Si dicho resultado se elabora un poco más se pueden distinguir las siguientes partes:

- Componente constante:

$$\frac{E_{sig}^2 + E_{LO}^2}{2}$$

- Componente de alta frecuencia:

$$\frac{E_{sig}^2}{2} \cos(2\omega_{sig}t + 2\varphi) + \frac{E_{LO}^2}{2} \cos(2\omega_{LO}t) + E_{sig}E_{LO} \cos((\omega_{sig} + \omega_{LO})t + \varphi)$$

- Componente de baja frecuencia:

$$E_{sig}E_{LO} \cos((\omega_{sig} - \omega_{LO})t + \varphi)$$

Como se desea trabajar con una menor frecuencia, se utiliza un filtro pasa-bajos para amortiguar de la componente de alta frecuencia, y además, la componente constante puede ser eliminada con un sustractor del mismo orden. De esta forma se tiene una señal IF de frecuencia $\omega_{sig} - \omega_{LO}$ conservando la fase y la amplitud de la onda recibida inicialmente, agregando una amplitud extra de E_{LO} , correspondiente a la ganancia del oscilador local.

A nivel de arquitectura, los sistemas de recepción consisten principalmente en 2 partes:

a) Antena

La antena es la primera etapa en el proceso de captar la señal que se quiere analizar y son del tipo parabólica. Su propósito es principalmente concentrar la energía recibida en un solo punto (foco), de esta forma se direcciona la potencia aumentando el flujo por unidad de área; facilitando así tener una “mayor cantidad de información” sobre el fenómeno que se quiere observar. Además la antena debe tener otras características relacionadas con la precisión de su movimiento, nivel de perfección alcanzado en la curvatura y en la superficie de la antena, grados de libertad

asociada a los ejes móviles de ésta y también poseer la habilidad de poder desplazarse para generar distintas líneas bases o “*baselines*”, concepto que será explicado más adelante.



(a)



(b)

Figura 7: (a) Antena móvil tipo Vertex usada en ALMA. (b) Antena fija usada en el observatorio de Arecibo.

b) Receptor

Esta parte es la encargada de preparar la señal recibida para ser analizada. El diseño del receptor depende de la frecuencia de la onda de entrada, afectando principalmente en las características constructivas de cada componente activo y pasivo; ya sean éstos filtros y amplificadores. También influye en el tipo de enfriamiento criogénico utilizado para disminuir la temperatura del equipo y así evitar el ruido generado por el calentamiento de los dispositivos activos en el receptor. En la siguiente tabla se puede observar a modo de ejemplo, el comportamiento típico de la relación temperatura–frecuencia en los receptores de ALMA y el tipo de tecnología ocupada.

Banda de ALMA	Rango de Frecuencia [GHz]	Temperatura del Receptor sobre el 80% de la banda de Frecuencia de Radio (RF) [K]	Tecnología del Receptor
1	31 - 45	26	HEMT
2	67 - 90	47	HEMT
3	84 - 116	60	SIS
4	125 - 163	82	SIS
5	162 - 211	105	SIS
6	211 - 275	136	SIS
7	275 - 373	219	SIS
8	385 - 500	292	SIS
9	602 - 720	261	SIS
10	787 - 950	344	SIS

Tabla 1: Características generales de los receptores para cada banda (ALMA Website, 2011).

Ahora bien, los elementos básicos presentes en el receptor se pueden clasificar en dos grupos: elementos activos y elementos pasivos.

Elementos Pasivos

- **Atenuadores:** aparecen deliberadamente en los circuitos para reducir la amplitud de señales de entrada muy amplias o simplemente para representar las pérdidas de piezas “sueltas” como conectores y juntas.
- **Aisladores:** son elementos no recíprocos, es decir, dejan que la potencia pase en solo una dirección. Se usan principalmente entre dispositivos para prevenir que las reflexiones de la señal ocurridas en una parte del receptor se propaguen y afecten a otros dispositivos del sistema.
- **Acopladores Direccionales:** consisten en las piezas que interconectan guías de ondas de distintas dimensiones, además pueden ser usados para acoplar dos dispositivos de características similares en la transferencia de la onda.
- **Sistemas Enclavados en Fase (PLL):** su propósito es proveer una señal estable tanto en frecuencia como en fase. Esto es necesario en el proceso de disminuir la frecuencia, de la señal observada, en un esquema de un receptor heterodino.
- **Filtros Pasivos:** se utilizan para filtrar los rangos de frecuencias indeseados producto de procesos de amplificación o armónicas debido al tratamiento de la señal en diferentes etapas.
- **Transductor Ortomodal (OMT):** es el elemento encargado de separar las polarizaciones vertical y horizontal de la señal de entrada, para ser tratadas en forma independiente en la etapa de análisis.

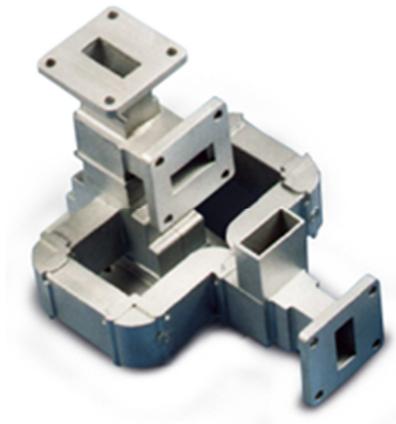


Figura 8: Tipo de OMT.

Elementos Activos

- **Amplificadores:** son necesarios para amplificar la señal recibida entre 80 y 100 [dB]. Se suelen conectar en cascada y son la principal fuente de generación de ruido térmico, por lo que se deben enfriar para evitar alteraciones indeseadas en la señal amplificada.

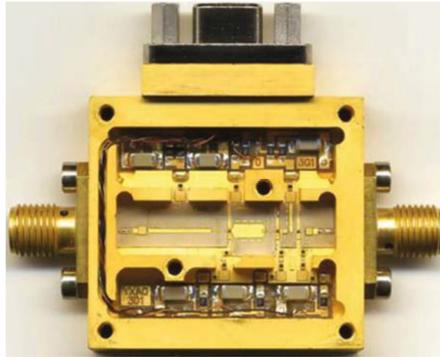


Figura 9: Amplificador HEMT de bajo ruido (LNA) utilizado en ALMA entre 4-8 [GHz].

- **Mezcladores:** se encargan de desplazar la frecuencia de la señal de salida a una deseada, lo cual es necesario para poder procesar más adelante la onda que viene en frecuencia de radio.
- **Oscilador Local:** generan una señal de referencia para los mezcladores y otros dispositivos, con una frecuencia determinada. Se caracterizan por ser altamente precisos.

Teniendo en cuenta los elementos ya mencionados, un esquema típico de un receptor es mostrado en la siguiente figura:

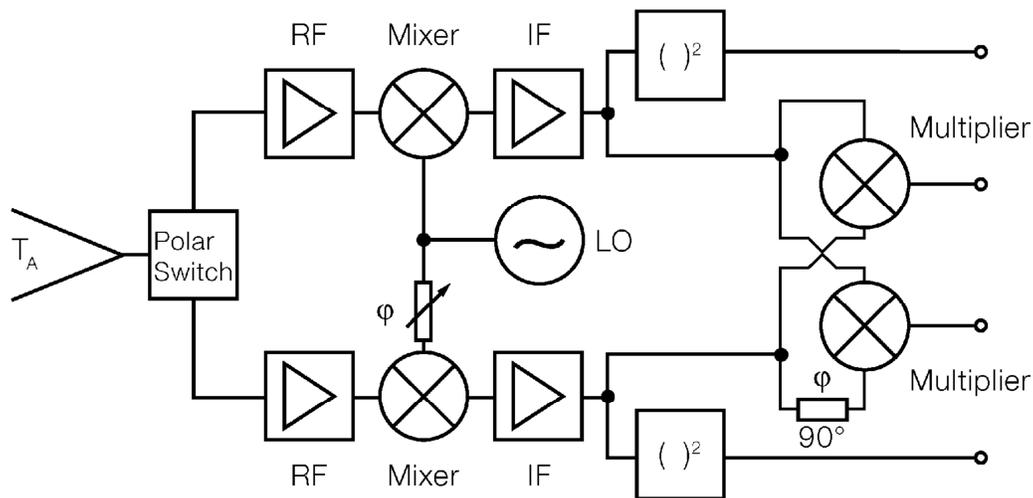


Figura 10: Receptor con 2 salidas de distinta polarización en Frecuencia Intermedia (IF) (Wilson & Rohfs, 2009).

Una estructura común usada en el receptor es tener en la entrada un *horn* circular que entrega una ganancia entre 15 y 25 [dB] que luego es conectado a un “Polar Switch” que puede ser un OMT para separar las polarizaciones. Después cada señal polarizada pasa por un conjunto de amplificadores y aisladores para aumentar la ganancia entre 80 y 100 [dB] para finalizar con una etapa de filtrado. Hasta acá tenemos la señal en RF (Frecuencia de Radio), la cual al pasar por los mezcladores, es convertida a IF (Frecuencia Intermedia) para llegar más adelante al correlador.

2.1.3. Sistemas de Análisis

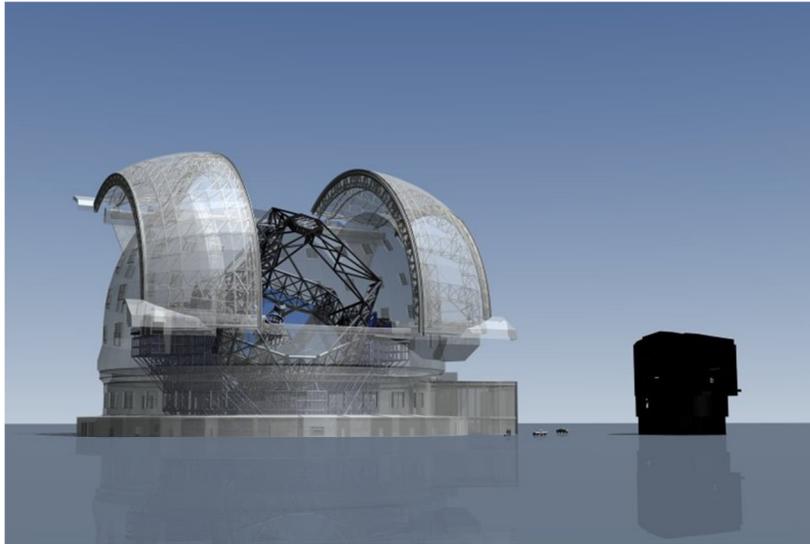


Figura 11: Imagen que ilustra diferencia de tamaños del espejo del E-ELT versus el VLT (ESO, 2011).

Teóricamente se tiene que mientras mayor es la dimensión del telescopio mejor es la resolución de la imagen obtenida, en contra de una mayor longitud de onda, por ende los primeros telescopios de alta resolución en el rango visible del espectro electromagnético poseían diámetros enormes, como el VLT de 8.3 [m] de diámetro o el futuro E-ELT de 39.3 [m] (ver imagen superior). Pero con la interferometría se logra que un grupo de antenas de un diámetro no mayor a los 12 [m], en conjunto operen como un radiotelescopio de grandes proporciones llamado interferómetro. Entonces, el sistema de análisis consta principalmente por un interferómetro constituido por 2 o más radiotelescopios. Teóricamente, la interferometría es posible gracias a que los radiotelescopios se encuentran separados una distancia llamada "*baseline*", donde cada par de radiotelescopios en el interferómetro presenta una *baseline* definida. En la siguiente ilustración se aprecia las *baselines* demarcadas en azul entre las antenas de ALMA en su configuración compacta.

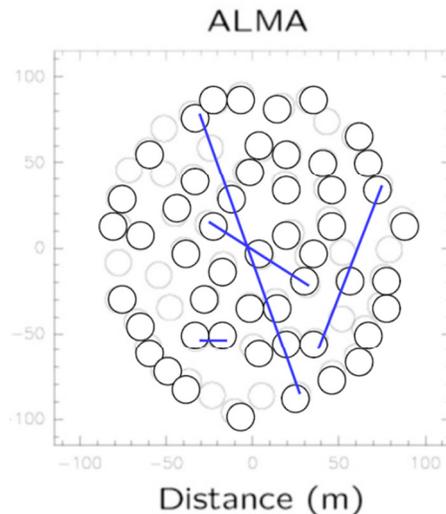


Figura 12: Arreglo compacto de ALMA, donde cada antena es representada por un círculo (ALMA, 2011).

Un interferómetro funciona de la siguiente manera. Supongamos que se tiene un cuerpo celeste que se quiere observar, la imagen obtenida es una fotografía bidimensional en un plano X-Y. A la señal que genera dicha imagen se le aplica una transformada de Fourier para obtener su espectro; el cual se traduce en un mapa bidimensional del espectro de coordenadas U-V, llamado el “plano U-V”.

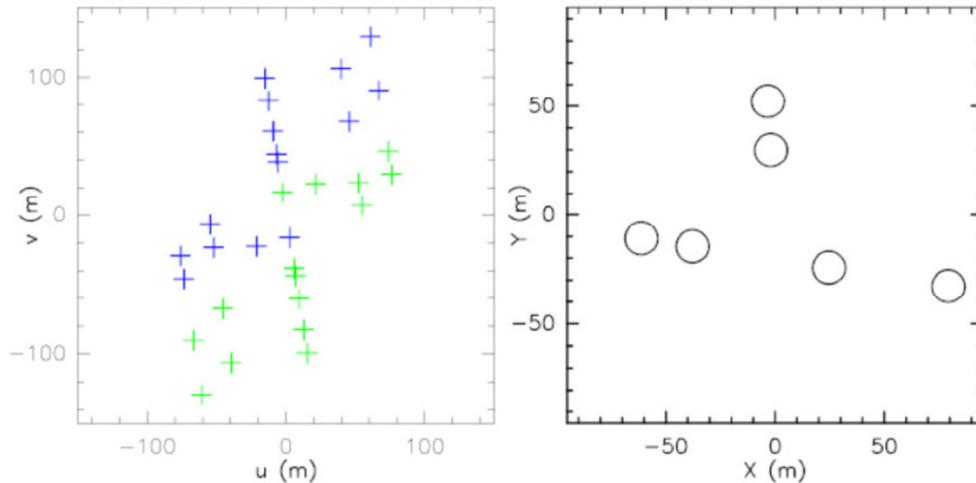


Figura 13: A la derecha se tiene la configuración de 6 antenas y a la izquierda su respectivo plano UV.

Cada *baseline* representa un punto en el plano U-V, el cual tendrá definida su distancia al centro del plano mediante la longitud de la *baseline*, lo cual en el fondo se traduce que para tener mayor resolución en la imagen se debe tener una mayor cantidad de *baselines*. Las líneas base pueden aumentar considerando un número mayor de antenas, o bien, variando las distancias de separación entre éstas; ya que a mayor longitud se tienen puntos concentrados en el centro del plano espectral. Además como la transformada de Fourier es simétrica, se tiene que cada punto posee su par simétrico central (ver Figura 13) respecto al origen del plano U-V; teniendo un total de $n(n - 1)$ puntos en el plano con un interferómetro de n antenas. Como se indicó anteriormente, la resolución de la imagen depende también de la longitud de onda percibida, que se traduce en la estructura del patrón de la radiación de la antena y en el del interferómetro. Un esquema de interferómetro simple con 2 antenas es el siguiente:

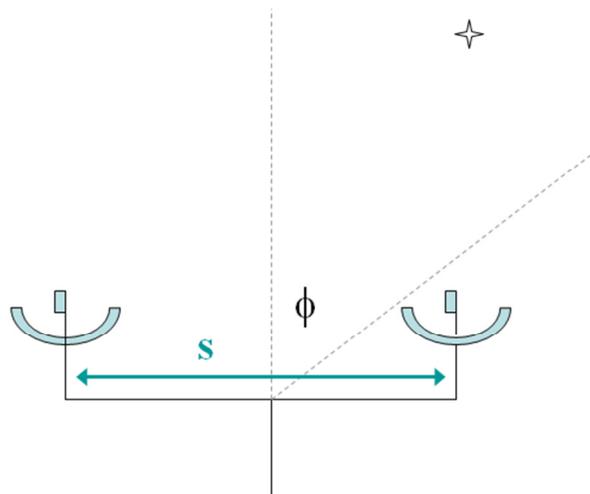


Figura 14: Esquema de dos antenas actuando como un interferómetro.

Se puede apreciar que en el esquema presentado, la variable S corresponde a la longitud de la *baseline* y ϕ es el ángulo horario definido por la normal a la superficie y el frente de ondas incidente en las antenas. Si la antena es de radio R y la longitud de onda observada es λ , la ecuación que representa el patrón de radiación del interferómetro en función del ángulo horario es:

$$P_2(\phi) = e^{-\left(\frac{\phi\pi R}{\lambda}\right)^2} \cos\left(\frac{\pi S}{\lambda} \text{sen}(\phi)\right)^2.$$

Y para contrastar, con una antena en observación directa, la potencia es:

$$P_1(\phi) = e^{-\left(\frac{\phi\pi R}{\lambda}\right)^2}.$$

Como se puede apreciar, la ecuación del interferómetro presenta un mayor “detalle” en cuanto al término extra dependiente de ϕ . Para observar dicho fenómeno hay que entender en mayor profundidad el desarrollo de la transformada de Fourier aplicada a la señal, y luego la transformada inversa para obtener la imagen. Pero este proceso no será visto ya que no es el fin último de este trabajo.

Hasta ahora se tiene una noción general del comportamiento del interferómetro y su funcionalidad, pero hay un punto bastante importante que tratar. En la realidad se tiene que si la fuente está lo suficientemente alejada de los receptores, los rayos emitidos por ésta llegan en forma paralela entre sí a las antenas que se encuentran relativamente cercanas; entonces la información que llega a una antena, llegará a la otra con un desfase temporal.

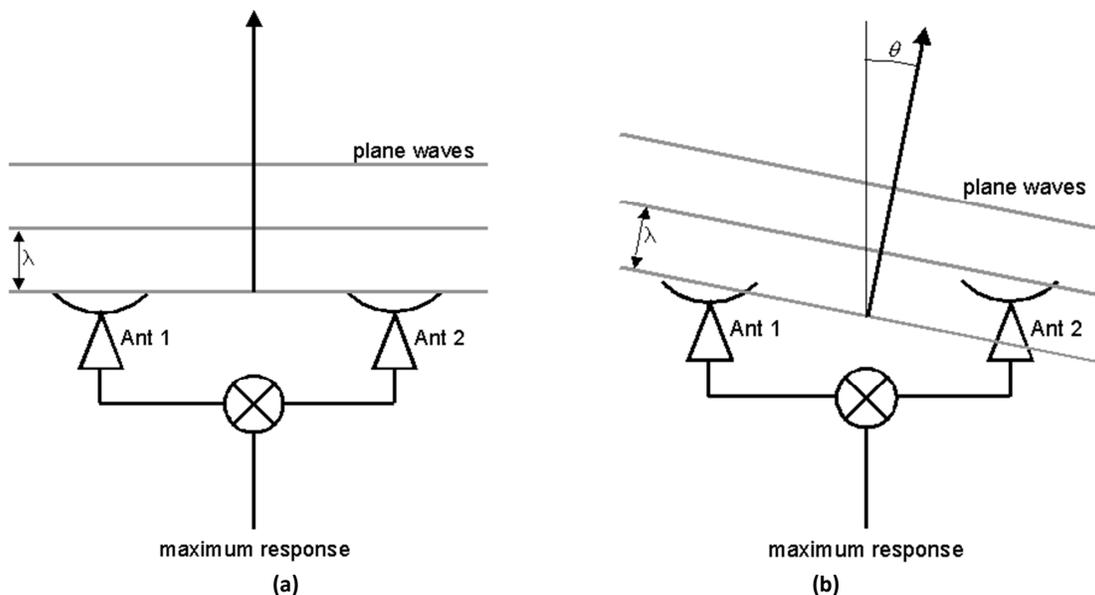


Figura 15: (a) Frente de ondas perpendicular a las antenas. (b) Frente de ondas con desfase temporal en la recepción.

Una solución al problema de la fuente no perpendicular a la superficie en donde están las antenas, es colocar un retraso temporal llamado “retraso geométrico” en la primera señal que es recibida, tal que al momento de ser correlacionada con la señal desfasada, ambas contengan la misma información del frente de ondas observado. La siguiente figura muestra un esquema de lo planteado anteriormente:

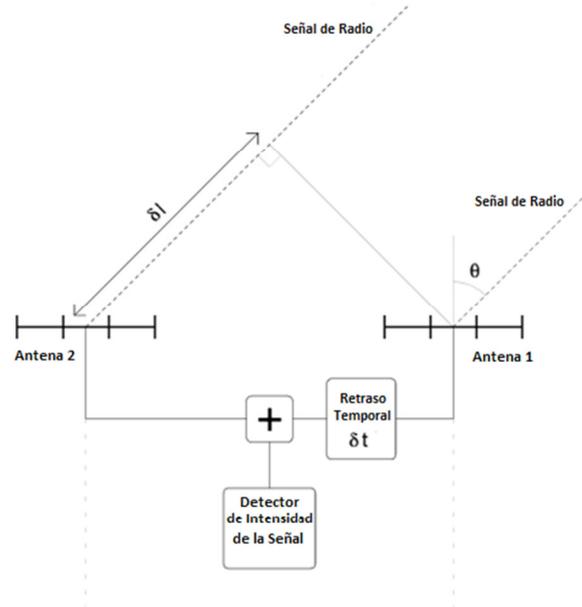


Figura 16: Interferómetro de dos antenas con señal retardada.

Se tiene que el retraso δt está directamente relacionado con el retraso espacial δl , ya que la finalidad del sistema es representar al mismo tiempo los datos del fenómeno observado.

2.1.4. Correlador

En la práctica, el correlador es el encargado de añadir los desfases temporales de cada señal recibida por cada antena, además de añadir también el desfase producto de los distintos largos de las rutas de la fibra óptica entre los receptores y la central de procesamiento de la información. También se encarga de entregar la correlación cruzada entre las salidas de los voltajes de cada par de antenas.

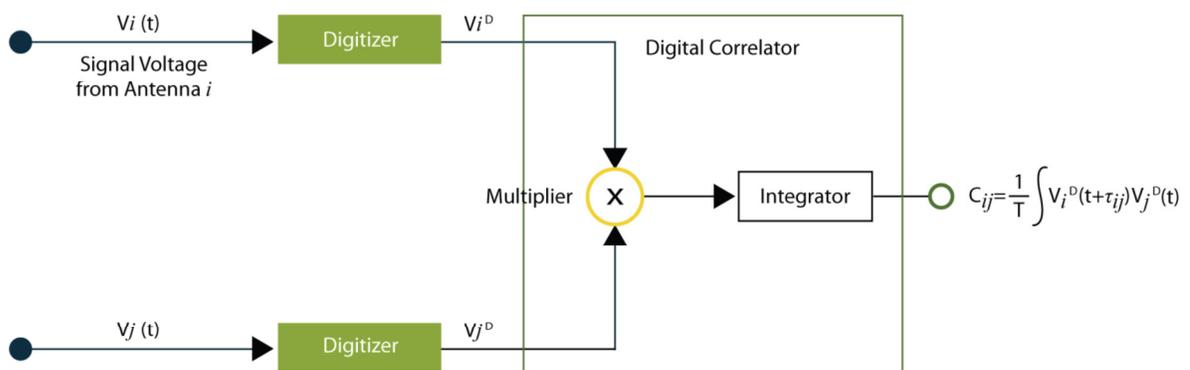


Figura 17: Esquema de un correlador (ALMA, 2011).

En el esquema anterior se aprecia que los voltajes de cada antena V_j y V_i son digitalizados y luego multiplicados al interior del correlador, para que finalmente sean integrados considerando el retraso geométrico entre ambas señales. La salida del correlador corresponde a la correlación cruzada $C_{ij} = \frac{1}{T} \int V_i^D(t + \tau_{ij}) V_j^D(t)$. Para obtener la correlación cruzada del espectro existen dos

arquitecturas en el diseño del correlador: XF y FX. Dichos nombres provienen del tipo de operación que el sistema realiza primero a las señales de entrada; si se calcula la correlación (X) o se le aplica la Transformada de Fourier (F).

a) Arquitectura XF

Consiste principalmente en aplicar primero la correlación cruzada a la señal digitalizada en intervalos de tiempo determinados, para luego calcular la FFT (Fast Fourier Transform) de los productos ya correlacionados y así obtener el espectro cruzado de potencia.

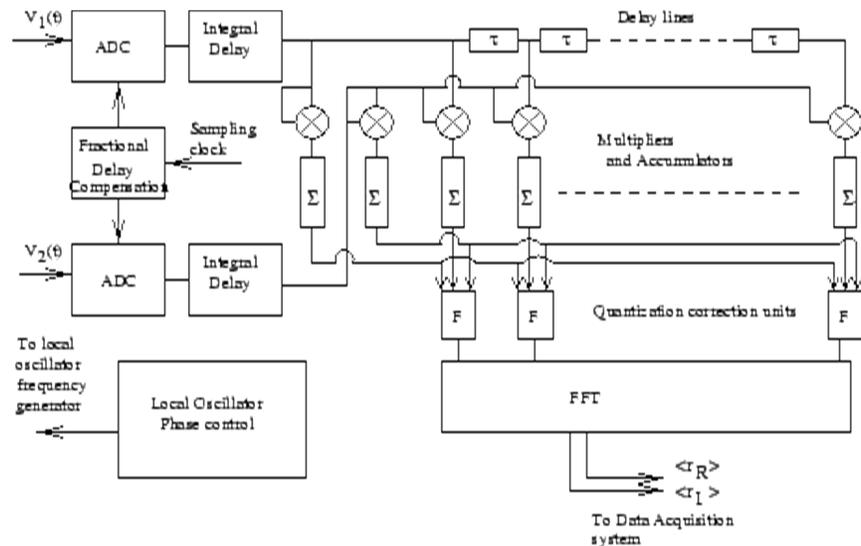


Figura 18: Diagrama de bloques de un correlador tipo XF (NCRA, 2012).

b) Arquitectura FX

A grandes rasgos, la señal digitalizada primero se le aplica una FFT para luego pasar por un multiplicador complejo por cada canal de frecuencia.

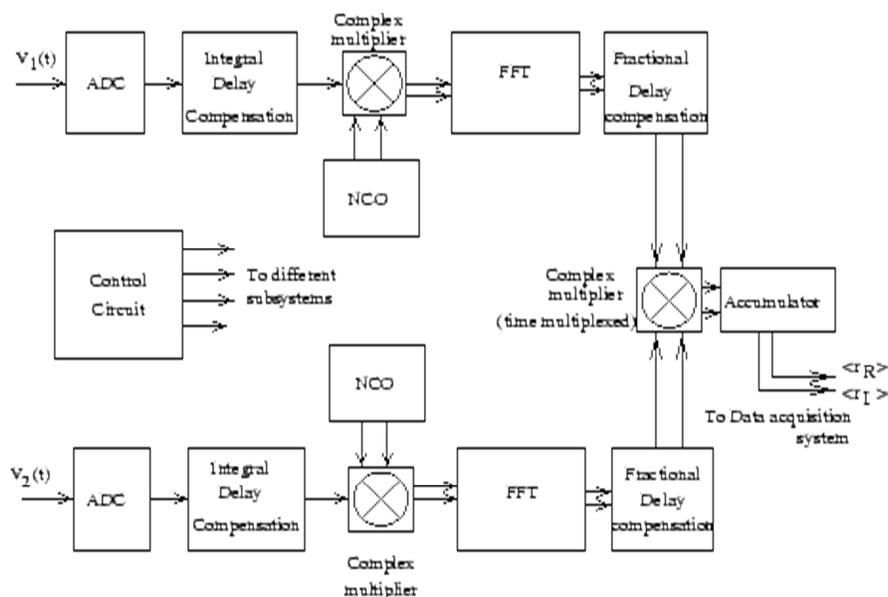


Figura 19: Diagrama de bloques de un correlador tipo FX (NCRA, 2012).

Dentro de las comparaciones que se pueden hacer entre ambas arquitecturas, la más importante es el número de multiplicaciones/acumulaciones cruzadas (MACs) que llevan a cabo cada una en el proceso de correlación, ya que esta cantidad influye directamente en la cantidad de recursos a utilizar para llevar a cabo dicho proceso. Para ejemplificar dicha comparación, primero se debe tener una idea de cuáles son los parámetros principales que influyen en el diseño de un correlador y de cómo éstos interactúan en el proceso de cálculo de la correlación. Primero se encuentra la tasa de muestreo del ADC representada por S , luego se tiene el ancho de banda deseado a observar del espectro BW y el número de canales a dividir dicha banda estaría representado por n . Dependiendo los datos a muestrear e ignorando el efecto de superposición o *aliasing*, se tienen las siguientes relaciones:

- a) Datos puramente reales (Muestreo de datos de Nyquist)

$$BW = \frac{S}{2}$$

- b) Datos puramente complejos

$$BW = S$$

- c) Resolución de frecuencia

$$\Delta f = \frac{BW}{n}$$

Nótese que la resolución de frecuencia indica qué tan preciso se puede medir una frecuencia determinada dentro del espectro observado, por lo tanto, mientras menor sea Δf implica que se tiene una observación mucho más precisa de la frecuencia deseada.

Regresando a la comparación de las arquitecturas que consideran sólo muestreo datos complejos, se debe tener en cuenta que en un correlador FX la señal de entrada es filtrada en n bandas de frecuencias y para cada banda se requiere calcular solo una correlación cruzada. Además se considerará un correlador XF con el mismo ancho de banda BW y tasa de muestreo S que el FX, pero si la resolución de frecuencia Δf es la misma, se deben considerar los $k * n$ retrasos incluidos en el esquema de la arquitectura XF, donde k es una constante que depende del tipo de ventana requerida, ya sea rectangular ($k = 1$) o de Blackman ($k \approx 1.7$) en el caso que se quiera eliminar fugas en el espectro. Resumiendo los datos en base a una *baseline* en una tabla se tiene que:

	Bandas de frecuencia	Tasa de muestreo por banda de frecuencia	Tasa de muestreo total	Retrasos por banda de frecuencia	MACs ops/s
Correlador XF	1	S	S	$k * n$	$k * n * S$
Correlador FX	n	S/n	S	1	S

Tabla 1: Comparación entre arquitecturas XF y FX.

Se puede apreciar que en un correlador FX las MACs son en un factor de $k * n$ veces menor que uno del tipo XF, por lo que la arquitectura FX es ideal para disminuir recursos destinados al cálculo de las operaciones.

2.1.5. Desarrollo de la Radioastronomía

La radioastronomía en los últimos 15 años ha experimentado tremendos avances en cuanto a la tecnología usada para la recepción y análisis de los datos, pasando por enormes antenas inmóviles a los más avanzados interferómetros que aprovechan la rotación de la tierra en la “síntesis de apertura” para llenar el plano U-V de manera más sencilla y elegante. El proceso de disminución de frecuencia de la señal (interferometría heterodina) ha sufrido grandes cambios desde el uso de amplificadores con transistores BJT hasta llegar a los LNA usando HEMT o junturas SIS. El proyecto más ambicioso que se encuentra actualmente en desarrollo, es el ya mencionado Proyecto ALMA (Atacama Large Millimeter/submillimeter Array), que se lleva a cabo en San Pedro de Atacama, Chile. Consiste en un interferómetro heterodino de 50 antenas de 12 metros de diámetro más un arreglo compacto (ACA) de otras 12 antenas de 7 metros de diámetro y otras 4 de 12 metros, el cual proporcionará datos de frecuencias espaciales para la calibración del arreglo grande y una sola antena de 12 metros.

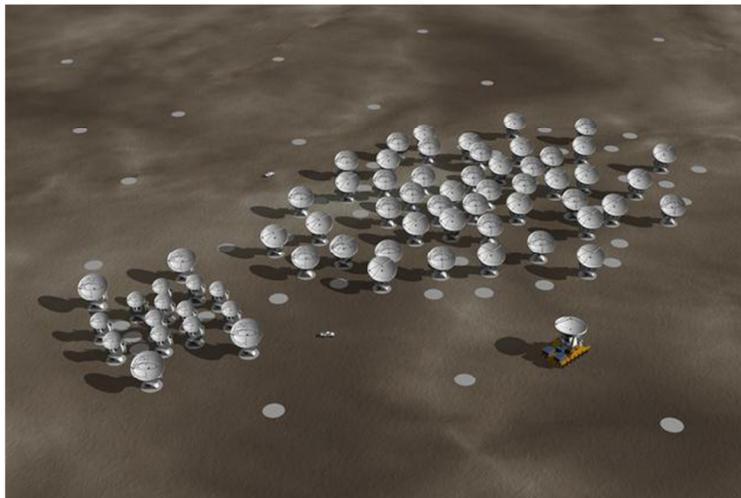


Figura 20: Configuración de las antenas de ALMA. A la izquierda se aprecia el ACA (ALMA, 2011).

Además del mega proyecto, existen otros grupos de Universidades e instituciones que llevan el plano de la radioastronomía a términos económicos más accesibles para que organizaciones educativas puedan desarrollar e integrar la radioastronomía dentro de sus campos de investigación y desarrollo. Además permite un desarrollo más personalizado a las necesidades y objetivos que se propongan dichos grupos a alcanzar. Este es el caso del grupo CASPER (Collaboration for Astronomy Signal Processing and Electronics Research) que fue fundado por un diverso grupo de profesores, graduados y estudiantes de la Universidad de California, Berkeley. El objetivo principal de dicha agrupación es optimizar, simplificar y hacer más eficiente el flujo de diseño de la instrumentación radioastronómica promoviendo la reutilización de diseños de software y hardware a través de plataformas independiente y de código libre (open-source).

Dentro de los diseños creados por CASPER se encuentra la placa ROACH, la cual será detallada más adelante en el capítulo de infraestructura disponible, sección de hardware. Dicha placa es parte medular para el diseño del correlador como objetivo principal de esta memoria; allí será programado y probado para el proyecto de un interferómetro heterodino de bajo costo del área de Instrumentación Astronómica del Departamento de Ingeniería Eléctrica en la Facultad de Ciencias Físicas y Matemáticas de la Universidad de Chile.

En el esquema anterior se tiene un láser altamente coherente de frecuencia cercana al infrarrojo, operando como un OL, cuyo haz será distribuido vía fibra óptica mono-modal por dos caminos ópticos, para después ser combinados con una señal estelar. La fase y la polarización del OL debe conservarse en el mezclador, lo cual implica el uso de controladores de polarización, un controlador de fase y un sistema LLC (*Line Length Corrector*), el cual está compuesto por un *fiber stretcher* facilitado por ALMA, un corredor de frecuencia (*frequency shifter*) y un espejo rotador de Faraday. Como en el sistema fotónico OL de ALMA, se propone, una vez que el interferómetro se encuentre en funcionamiento para *baselines* cortas de dos metros de longitud, la implementación de un LLC para utilizar líneas bases de mayor longitud. En un brazo óptico (el de mayor longitud, el otro puede quedar sin cambios, como se puede apreciar a la izquierda de la Figura 21) la señal del OL es corrida en frecuencia al final del montaje, antes de ser devuelta por la misma fibra para generar la señal de control, junto con la señal original del OL en un fotodiodo. Luego, contando los máximos (*fringes*) en esta señal, se recupera la información de la longitud del camino óptico, y estabilizando en un *fringe* se puede estabilizar el largo completo de la fibra. La necesidad de este procedimiento yace en que los cambios mecánicos y térmicos en el camino óptico de la fibra deben ser compensados, en orden de mantener un desfase fijo entre la señal del OL que fue dividida para ser mezclada con la señal recibida por cada telescopio.

Para poder realizar mediciones con *baselines* de mayor longitud, se propone probar la aplicabilidad del esquema corrector de la longitud del camino óptico en fibra de ALMA, el cuál en términos técnicos, se encuentra altamente desarrollado.

Volviendo al esquema del montaje, se tiene que las señales pasarán por un mezclador, en dónde las señales estelares junto a la del OL, generan una señal con una frecuencia mucho menor ($100 \text{ [MHz]} - 3 \text{ [GHz]}$), tal que las relaciones de fase y amplitud entre las señales estelares se mantenga.

En uno de los brazos del esquema, se debe añadir una línea extra para compensar el retraso geométrico producto del ángulo de observación al que se encuentre el interferómetro. Y finalmente, se requerirá de un correlador para medir el grado de coherencia de la fuente observada. Dicho sistema entregará tanto la información del espectro auto-correlacionado y de correlación cruzada, además de entregar información sobre la fase entre las señales de cada telescopio.

2.2. Infraestructura Disponible

2.2.1. Espacio Físico

Dentro del espacio físico, el trabajo de memoria se realiza en dos lugares pertenecientes a la Facultad de Ciencias Físicas y Matemáticas (FCFM) de la Universidad de Chile. La ubicación principal donde se llevarán a cabo los experimentos y el trabajo de diseño del correlador es en el Laboratorio de Fotónica, ubicado en el quinto piso del Departamento de Ingeniería Eléctrica en la FCFM. En este lugar se encuentran la mayoría del hardware a utilizar, así como los implementos ópticos y electrónicos, para desarrollar los distintos montajes de las pruebas contempladas. El segundo espacio es el Laboratorio de Ondas Milimétricas en el Departamento de Astronomía, ubicado en Cerro Calán. Allí se realizarán algunas pruebas secundarias en cuanto al comportamiento del correlador bajo condiciones cercanas a las ideales.

2.2.2. Hardware

En esta sección se describirá de manera detallada el equipo a utilizar, para tener las características principales en cuenta a la hora de diseñar lo que se requiere. Además de listar los equipos a utilizar en el experimento.

2.2.2.1. Placa ROACH

La placa ROACH (Reconfigurable Open Architecture Computing Hardware) es una placa que contiene principalmente una FPGA (Virtex 5 de Xilinx) que puede ser programada para que cumpla distintas funciones en hardware. Además posee un Power PC embebido y distintos bloques de memoria que facilitan el acceso y distribución de tareas de la FPGA al resto de dispositivos periféricos, para que los recursos de la Virtex 5 no sean usados en procesos secundarios y el chip se pueda focalizar sólo a la tarea prediseñada; en el caso de este trabajo de memoria corresponde a la de un correlador.

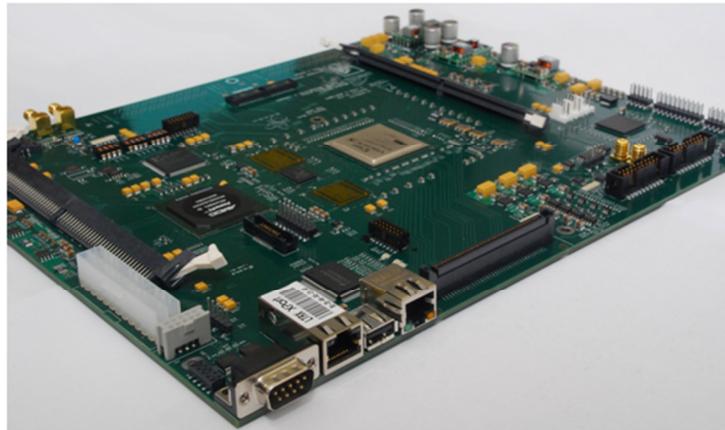


Figura 22: Placa ROACH.

La ROACH posee además otro conjunto de características, dentro de las cuales destaca la posibilidad de conectarle en 2 slots (Z-DOK) conversores Análogo-Digital (ADC) o Digital-Análogo (DAC) dependiendo del objetivo en la programación de la FPGA. El siguiente diagrama de bloques representa la placa con sus dispositivos:

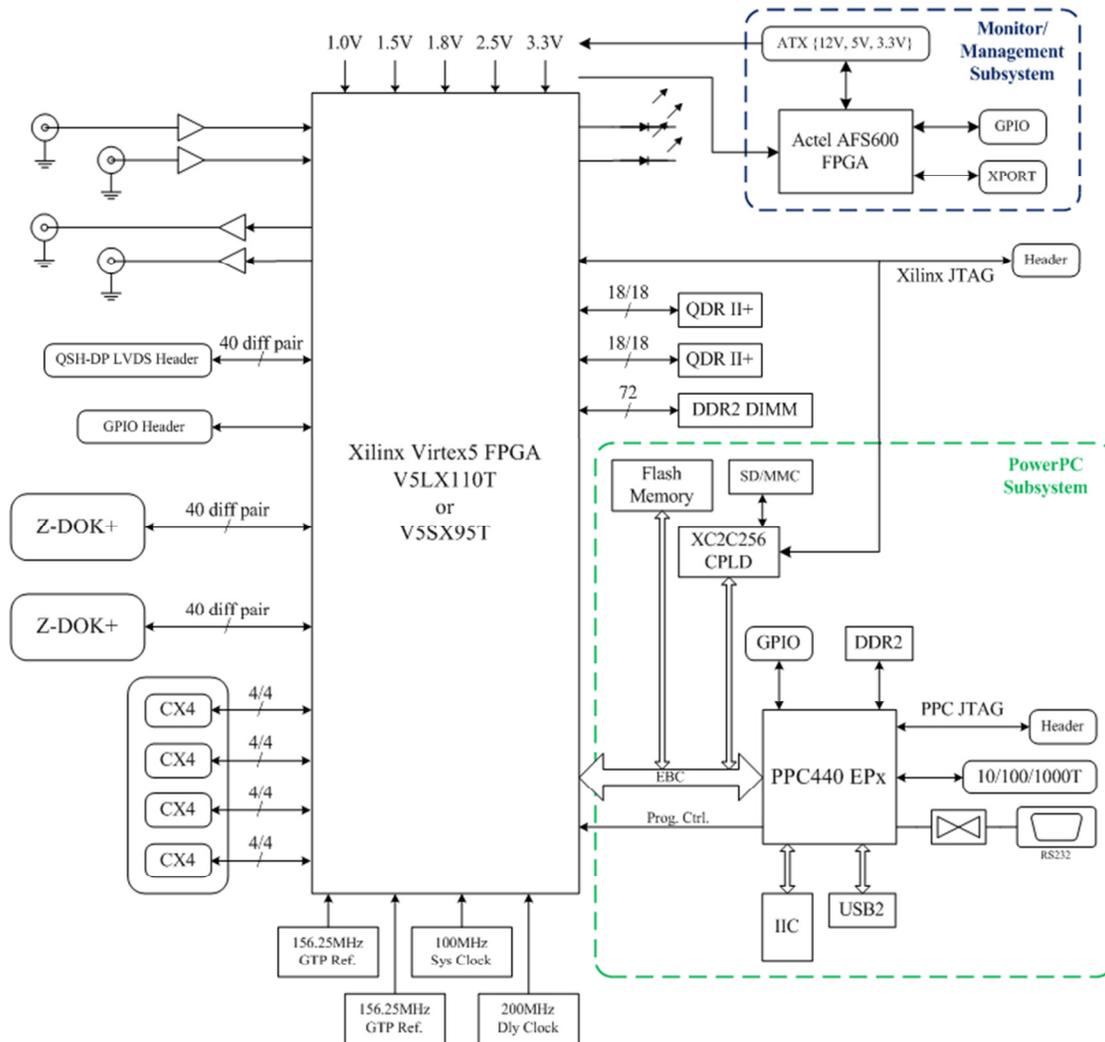


Figura 23: Diagrama de bloques de la placa ROACH con sus distintos componentes y conexiones (CASPER Website).

Las especificaciones de la placa son las siguientes:

- FPGA
 - 1x Xilinx Virtex-5 XC5VLX110T-1FF1136 FPGA
- Interfaces de la FPGA
 - 2x Z-DOK+ 40 conectores de par diferencial
 - 4x CX4 10Gbps conectores seriales de alta velocidad
 - 1x QSH 40 conector de par diferencial
 - 16x GPIO
 - 4x SMA IO (2x clock-capable)
- Periféricos de la FPGA
 - 2x 2M x 18-bit QDRII+ SRAMs
 - 1x DDR2 DRAM DIMM
 - CPU
 - 1x AMCC PowerPC 440EPx Procesador Embebido

- Interfaces de la CPU
 - 1x RS232 DB9 Puerto serial
 - 1x 10/100/1000Mbit RJ45 Ethernet
 - 1x USB2.0
 - 1x MMC/SD Socket para tarjeta

- Monitoreo y manejo de los sistemas internos
 - Temperaturas del Xilinx Virtex5, PowerPC y Actel Fusion.
 - Voltajes de 12V, 5V, 3.3V, 2.5V, 1.8V, 1.5V, 1V y 1.2V para carriles auxiliares.
 - Apagado automático ante eventos de sobre temperatura y sobre o baja tensión, con registro del evento después del apagado.
 - Sistema de encendido/apagado remoto.
 - Un puerto 100Mbps Ethernet separado para control independiente de control de placa y monitoreo de salud.

Una característica fundamental de la ROACH es que el código de programación necesaria no se lleva a cabo en lenguaje Verilog o algún otro HDL, sino que se realiza en el MATLAB-Simulink® mediante la conexión de bloques prefabricados por CASPER y Xilinx. Cuando se realiza algún diseño de hardware en MATLAB-Simulink®, el sistema comienza a compilarlo y a transformar el lenguaje usado en los diagramas de bloques de Matlab en lenguaje de máquina, para finalmente entregar un archivo de extensión *bof*. Una vez conectada la ROACH con un PC externo mediante un puerto RJ45 Ethernet, se procede a copiar el archivo *bof* en un directorio predeterminado, para finalmente ser ejecutado con una instrucción prediseñada y de esta forma programar la FPGA como se diseñó en MATLAB-Simulink®.

Aun teniendo el proceso “automático” de conversión de archivos y programación de la FPGA, uno de los objetivos secundarios de la memoria es entender a cabalidad el proceso de compilación y de diseño de los bloques de hardware en MATLAB-Simulink®, para poder crear herramientas más personalizadas y tener los bloques creados “a medida” de las necesidades presentes en cualquier proyecto relacionado con la ROACH.

Para obtener información más detallada del proceso, revisar la Sección 2.3 dentro de este capítulo, y revisar además a Sección 2.4 de la referencia bibliográfica que se entrega al final del mismo.

2.2.2.2. Conversor Análogo Digital

El conversor ocupado es un ADC083000 de National Instruments. Es un conversor análogo-digital basado en CMOS de baja potencia y alto rendimiento que digitaliza señales a 8 bits de resolución a tasas de muestreo de 3.4 [GSPS]. Típicamente consume una potencia de 1.9 [W] a 3 GSPS de una fuente a 1.9 [V], este dispositivo garantiza la no pérdida de códigos dentro del rango completo de temperaturas en que puede funcionar.

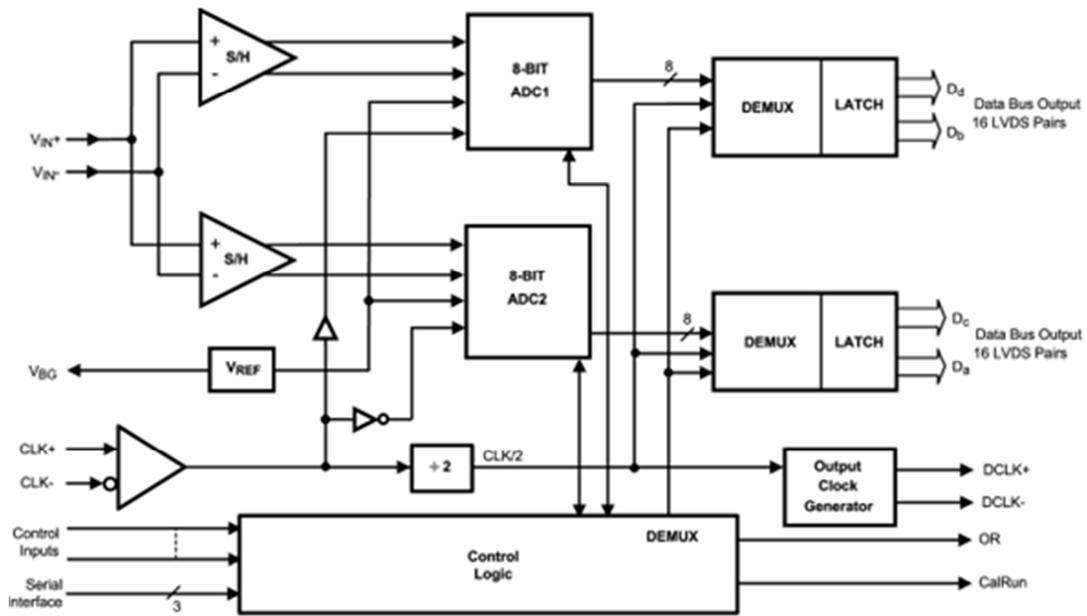


Figura 24: Diagrama de bloques del conversor ADC083000 (National Instruments).

Dentro de las características se puede encontrar que posee una arquitectura de interpolación, un diseño de comparador diferencial, un innovador diseño de un amplificador interno que toma las muestras y las retiene, y además posee un esquema de auto-calibración. Todo esto permite que la ADC posea una excelente respuesta para todos los parámetros dinámicos hasta Nyquist, produciendo un alto número efectivo de bits de 7.0 (ENOB) con una señal de entrada a 748 [MHz] y una tasa de muestreo de 3 [GHz] suministrando una tasa de error de palabra entre 10 y 18 y además, la relación señal-ruido (SNR) alcanza los 44.5 [dB] a los mismos 748 [MHz]. El ADC083000 logra las 3 [GSPS] utilizando tanto los cantos de subida y de bajada del reloj de entrada a 1.5 [GHz]. El formato de la salida es de offset binario y el conector de salida LVDS digital es compatible con IEEE 1596.3-1996.

Esta ADC posee un demultiplexor 1:4 que alimenta 4 buses LVDS y reduce la data de salida a la cuarta parte de la tasa de muestreo para cada bus.

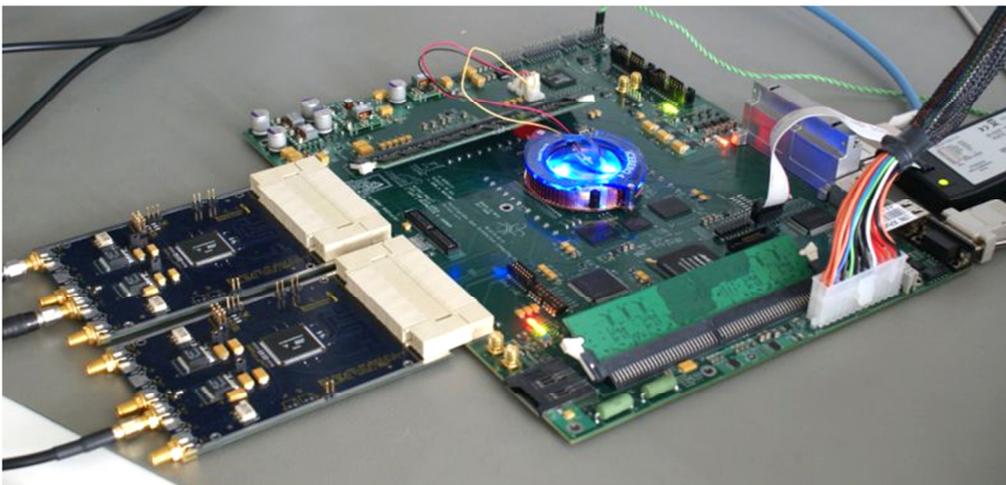


Figura 25: Placa ROACH con 2 tarjetas ADC montadas en la izquierda de la imagen.

2.2.2.3. Instrumentos de Análisis y Generación de Señales

El laboratorio cuenta con los siguientes equipos para desarrollar los experimentos necesarios:

- Sintetizadores de señales: el equipado en la ROACH (Valon 5007) y el sintetizador de frecuencias (Phase Matrix).
- Cables y conectores SMA.
- Analizador de espectro.
- Conectores BNC/SMA requeridos para generar divisores de señales.
- Amplificadores LNA de ganancia 35 [dB] y ruido de 1.5 [dB].
- Fuente de alimentación de los amplificadores.
- Cables de alimentación.
- Sintetizadores láser.
- Fotodiodos.
- Cables de fibra óptica en diversos largos.

2.3. Metodología

El trabajo de memoria principalmente consiste en la programación de una FPGA para que se comporte como un correlador. Para ello se utilizarán diversas herramientas de software que cumplirán distintos roles en el transcurso del trabajo, es decir, unos se requerirán permanentemente en su funcionamiento, mientras otros se utilizarán sólo durante la etapa de diseño. Además, dado que todo el contenido en software de la iniciativa es de acceso libre y gratuito, la mejor plataforma de desarrollo en la implementación y mantención del software asociado al manejo de la ROACH es el sistema operativo (SO) Linux. Por lo que el Power PC embebido en la placa ROACH está administrado por Linux, lo que fuerza de alguna manera, a que el usuario externo al querer acceder a la ROACH, también debe hacerlo desde algún sistema montado en Linux, en este caso se usará el SO Ubuntu. Dicha plataforma solo se utilizará como un sistema operativo dónde se instalarán el resto de aplicaciones de control y diseño.

Es necesario tener instalado además el programa ISE de Xilinx, el cual es parte vital en la etapa de compilación de los diagramas de bloques generados. No es necesario acceder a éste en forma sustancial, pero aun así se debe tener instalado y con las librerías necesarias.

Cabe hacer hincapié que para el correcto funcionamiento del software detallado a continuación, se deben tener la versión de Ubuntu y de Xilinx definida por CASPER, además de las bibliotecas necesarias para la funcionalidad del sistema, las cuales son:

- Ubuntu o CentOS (se trabajó con Ubuntu 12.04)
- Matlab 2012a
- Xilinx ISE 14.2 (Incluyendo las herramientas DSP)
- Librerías de CASPER
- Archivos de arranque de CASPER

Se debe destacar, que el trabajo realizado para llevar a cabo la correcta instalación del software necesario, en conjunto con las librerías de CASPER, requiere de bastante tiempo e información detallada por el grupo CASPER, ya que pueden presentarse diversos problemas al momento de configurar el sistema para que arranque con las librerías deseadas, o bien, para que

los modelos compilen. La mayoría de estos problemas provienen de la configuración inicial que tenga el sistema operativo; además de la versión del *kernel*, o núcleo, que éste posea. En caso de necesitar una nueva instalación del sistema, o actualizar las librerías de CASPER, se recomienda visitar la página web de este grupo, y revisar el archivo de correos para encontrar información relativa al proceso de instalación de las nuevas librerías; cómo también el apartado de *toolflow*. En la sección 2.4 se puede encontrar información sobre el sitio web de CASPER.

2.3.1. Programación de Hardware

- **MATLAB-Simulink®**

Esta herramienta se utilizará principalmente para el diseño del correlador mediante diagramas de bloques, los cuales están prediseñados en el lenguaje ocupado por MATLAB-Simulink®. Luego serán compilados para después ser copiados en la ROACH, y desde allí ser ejecutados para que la FPGA se programe de acuerdo a lo estipulado en el diagrama de bloques hecho en MATLAB-Simulink®. En la figura que se muestra a continuación se observa el diseño de un espectrómetro mostrado en uno de los tutoriales adjuntos en el sitio web de CASPER. Los bloques representan funciones de hardware, como por ejemplo hay bloques que realizan la FFT, otros son sumadores o bien un conversor análogo digital (ADC).

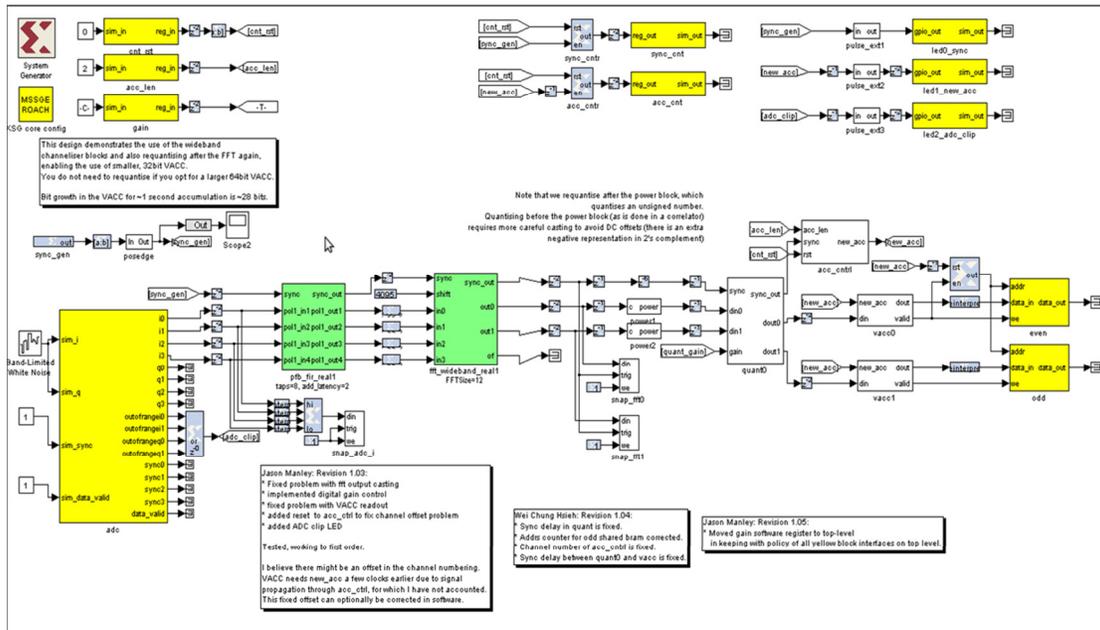


Figura 26: Diagrama de bloques de un espectrómetro hecho en MATLAB-Simulink®.

- **Software Especifico**

Existen piezas de hardware que poseen software específico para su control, por lo que se dependerá de éstos para el correcto funcionamiento de dichas partes. Cabe destacar que algunos de éstos están hechos solo para ejecutarlos en un sistema operativo Microsoft Windows, por lo que se utilizará una máquina virtual dentro del OS Ubuntu para simular Microsoft Windows y

poder utilizar dichos controladores. En la siguiente imagen se puede apreciar la interfaz gráfica del software de control del sintetizador incluido en la ROACH.

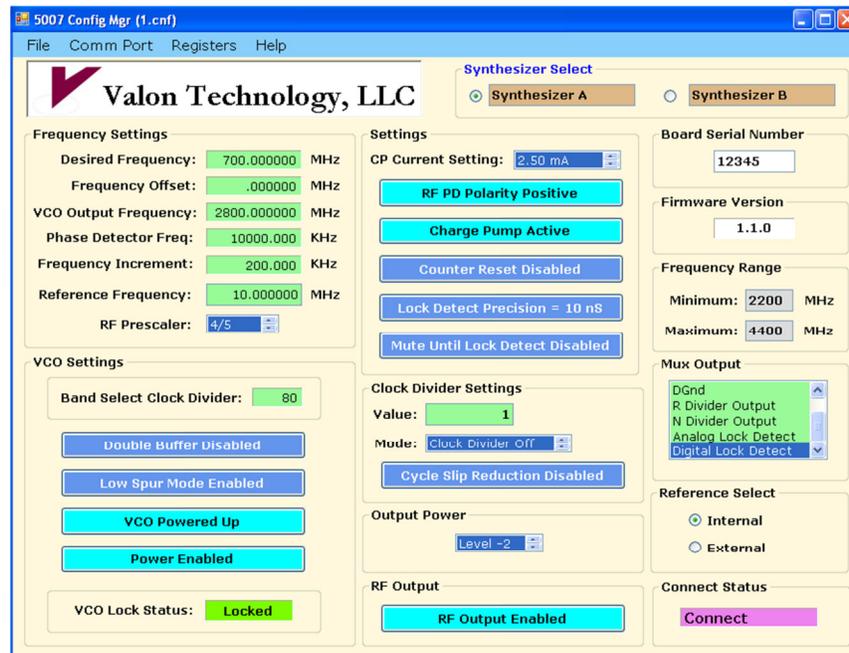


Figura 27: Software de control del sintetizador incluido en la ROACH.

2.3.2. Obtención de Datos

- **Scripts en Python**

Una vez que esté el hardware programado en la FPGA, los datos de las mediciones deben ser extraídos y analizados desde los puertos de la ROACH. Para ello se utilizarán archivos que contendrán los conjuntos de instrucciones necesarias, o scripts, programados en Python. Los scripts se encargarán de obtener la data, ordenarla y graficarla para después ser analizada y comparada con los resultados esperados. El proceso de instalación de las librerías necesarias para el desarrollo de dichos scripts se detallará más adelante en el Capítulo 3.

2.4. Reseña Bibliográfica

El estado del arte del tema es prácticamente nuevo, dado que la iniciativa del desarrollo de radioastronomía de bajo costo es reciente. Aunque se dispone de una amplia gama de información en un sitio web generado para esta iniciativa, dónde se comparte toda la información generada en el estudio y desarrollo de diversos proyectos relacionados con el rubro. Además, el sitio contiene numerosos tutoriales y guías para comprender mejor el funcionamiento del software y hardware desarrollado por CASPER, los cuales son los facilitadores de las herramientas básicas para implementar la iniciativa ya explicada en el capítulo anterior. El sitio se puede visitar en https://casper.berkeley.edu/wiki/Main_Page.

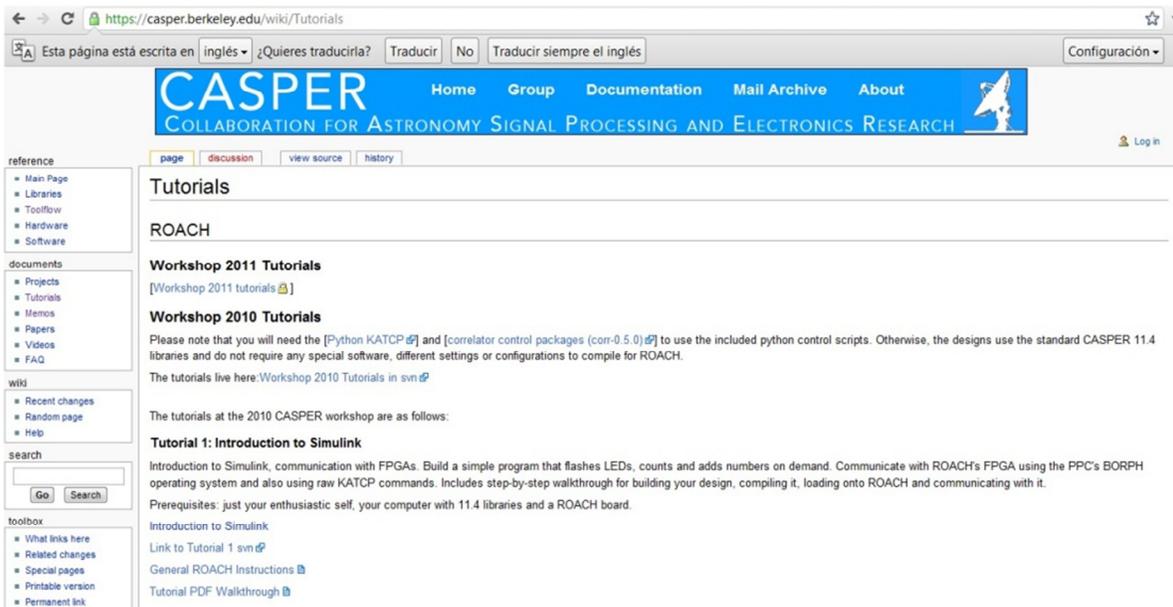


Figura 28: Sitio web de CASPER.

También existe un archivo de correos, donde integrantes de numerosas instituciones a nivel mundial intercambian información con el resto sobre problemas técnicos de hardware y de software; así como también se dan “consejos” sobre ciertos procedimientos a seguir en la mantención y el correcto funcionamiento del equipamiento. En Chile, se tiene que sólo la Universidad de Chile y la Universidad de Concepción, en conjunto están desarrollando proyectos de radioastronomía con la metodología desarrollada por CASPER.

3. Implementación

En este capítulo se procederá a examinar el diseño de dos tipos de correladores en MATLAB-Simulink®, detallando la estructura general que tendrán para luego dar paso a una explicación más profunda de los bloques principales que participan en el diseño de cada uno; y así poder definir las ventajas y desventajas que tendrán los diseños. Además se dará a conocer el experimento a realizar para probar la funcionalidad del correlador, el montaje de éste y las condiciones en las que se realizará. Cabe destacar que la implementación del sistema obtuvo la colaboración por parte de estudiantes de postgrado del DAS y practicantes de la Universidad Federico Santa María.

3.1. Estructura General

Como fue mencionado en el capítulo 2, existen principalmente dos formas de hacer un correlador, el tipo FX o XF. En este caso el desarrollo se hará en base a la arquitectura FX ya que presenta una ventaja considerablemente mayor a la arquitectura XF en cuanto al uso de recursos ofrecidos por la FPGA, como fue explicado en la Sección 2.1.4; por lo tanto, los dos diseños mencionados al comienzo de este capítulo se basarán en la arquitectura FX. La estructura general que será ocupada en los diseños, se puede apreciar en el siguiente diagrama:



Figura 29: Diagrama de bloques del diseño general de un correlador.

A continuación se procederá a explicar cada bloque del diagrama, y su respectiva función en el correlador.

3.1.1. Conversor Análogo Digital (ADC)

Como dice el nombre, este bloque se encarga de convertir la señal análoga de entrada en una señal digitalizada con una resolución de 8 bits. Aun cuando se poseen los ADC en forma física conectadas a la ROACH, se necesita una representación de éstos en MATLAB-Simulink®, dado que el proceso de compilación del modelo requiere los parámetros con los que trabajarán realmente los ADCs. La frecuencia con la que operará el ADC, dada sus características y los requerimientos del ancho de banda será de 700 [MHz]. Además este bloque está configurado para representar 2 ADCs, por lo que en los diseños se utilizará solo uno. Se debe tener en cuenta que cada ADC toma dos muestras por ciclo de reloj.

3.1.2. Banco de Filtros Polifásicos (PFB)

El proceso de filtrado polifásico es esencial para obtener la correlación de las señales de entrada de los ADC con una mejor resolución espectral, en gran medida libre de posibles problemas producidos por el *aliasing* y otras fuentes en el proceso de aplicación de la Transformada de Fourier a la señal de entrada, si ésta fuera convertida directamente al dominio de frecuencia. Por motivos del tema en que se basa este trabajo, no se entrará a explicar detalladamente el proceso de filtrado polifásico pero a grandes rasgos, se tiene que el filtro

subdivide la señal de entrada en un cierto número de partes de igual tamaño, o *taps*, en forma paralela para después aplicar a cada *tap* un filtro FIR y así dejar cada sección de la señal original filtrada para luego aplicarles la FFT.

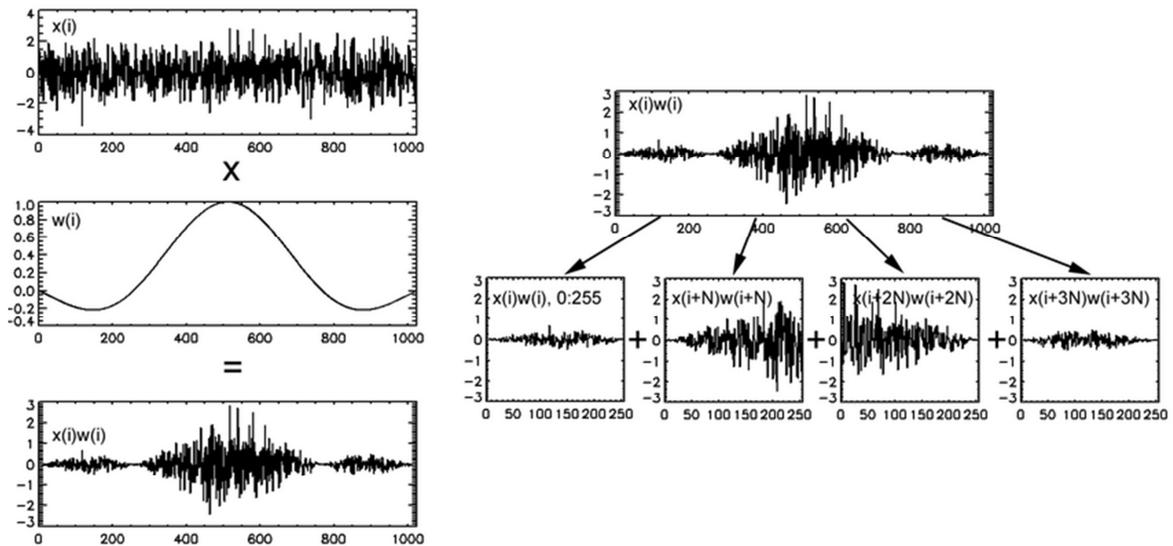


Figura 30: División de una señal en 4 *taps*.

Para efectos del modelo, se tiene que el banco de filtros polifásico tiene dos componentes principales. El primer componente es el bloque PFB FIR que se encarga de dividir la señal en el número deseado de *taps* para después en paralelo, aplicarles un filtrado FIR. El segundo componente corresponde al bloque que realiza el algoritmo de la FFT a cada canal de entrada en éste. Cada bloque puede ser configurado en diversos aspectos, destacando el tamaño de bits a utilizar en las operaciones, el número de entradas y salidas de los bloques y el tamaño de los datos que recibirán en cada entrada.

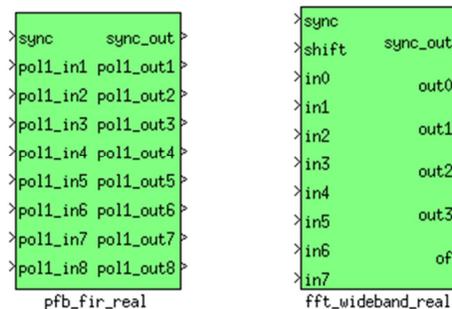


Figura 31: Representación de los bloques PFB FIR y de FFT en MATLAB-Simulink®.

3.1.3. Proceso de Multiplicar/Acumular (MAC)

Durante este proceso se realizan las multiplicaciones necesarias de cada canal para luego poder obtener la correlación cruzada y la auto-correlación de las señales de entrada, y una vez realizadas se van acumulando en un registro donde se irán almacenando hasta que una señal indique que el espectro fue acumulado completamente. De esta forma se sabe cuándo una etapa de integración se ha terminado y así vaciar el registro, dando lugar a una nueva actualización del espectro.

3.1.4. Registro

El registro de los datos se realiza en la BRAM interna de la ROACH, donde cada acumulador guarda las multiplicaciones en un registro previamente definido. Los datos almacenados podrán ser extraídos usando un script basado en Python, con métodos y librerías instaladas previamente en el sistema. Las librerías principalmente contienen métodos, funciones y definiciones de protocolos especializados para la comunicación entre la ROACH y el PC de control del correlador.

3.2. Diseño del Correlador

Como se ha mencionado al comienzo del capítulo, se trabajó con dos diseños de correlador, ambos basados en modificaciones de diseños existentes en la página de CASPER. El primer diseño se basó en un correlador hecho para 4 antenas con un ADC distinto al requerido. El segundo modelo se basó en el proyecto VEGAS (VErsatile GBT Astronomical Spectrometer), el cual es un espectrómetro diseñado para el GBT (Green Bank Telescope). Internamente realiza la correlación entre dos antenas, con una resolución espectral y ancho de banda muy superior al primer diseño mencionado. A continuación se detallarán los modelos en gran medida y se explicarán los pasos requeridos para hacer funcionar ambos modelos, teniendo en cuenta el software necesario y los scripts desarrollados para la obtención de datos.

3.2.1. Correlador de 700 [MHz]

Este correlador tiene como principal característica que el ADC funciona a una frecuencia de 700 [MHz] y la FFT proporciona 1024 canales de frecuencias del espectro. Los datos obtenidos por el correlador se guardan en la memoria interna de la ROACH para después ser extraídos utilizando un script desarrollado para este modelo.

3.2.1.1. Implementación del Modelo en MATLAB-Simulink®

Los valores de cada parámetro necesario en cada bloque se pueden encontrar en el modelo ya implementado, acá sólo se describirán los parámetros más importantes y se enseñarán figuras que indiquen cómo llevar a cabo las conexiones que deben realizarse entre los bloques.

El primer subsistema que debe implementarse son los bloques “*System Generator*” y “*XSG Core Config*”, los cuales se encargarán al momento de compilar el modelo, establecer la comunicación entre Matlab y el software de Xilinx para poder crear el archivo *bof* del diseño y así poder programar la FPGA con el correlador implementado en MATLAB-Simulink®. En esta primera etapa también se debe implementar el sistema que generará la señal de sincronización para todos los bloques del modelo. De esta forma se tiene la precaución de que todos los subsistemas operen en conjunto durante intervalos de tiempo controlados, asegurando así que los datos sean válidos en cuanto al tratamiento de la misma parte de la señal de entrada vista por todos los bloques del modelo. También se ha tenido en cuenta que no todos los bloques trabajan con los datos a la misma rapidez, lo cual puede provocar saturación o pérdidas de información en el procesamiento de las señales, por lo tanto, se han añadido en los casos necesarios bloques de retraso temporal en las entradas y/o salidas de algunos subsistemas. En la figura 32 se pueden apreciar los dos subsistemas ya mencionados.

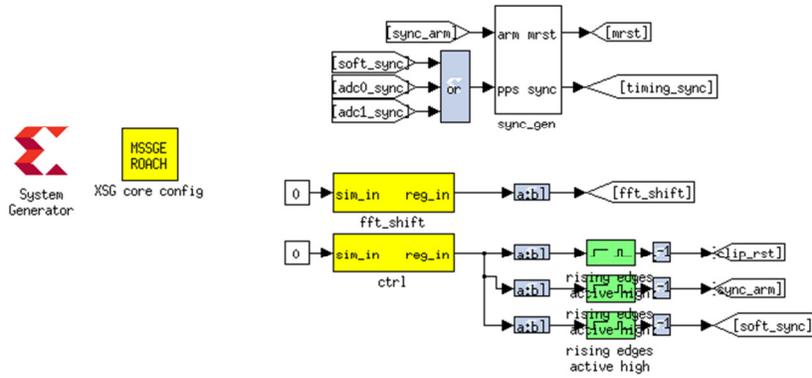


Figura 32: Bloques de comunicación y generador de señales implementados en MATLAB-Simulink®.

Siguiendo la estructura señalada en la Sección 3.1 primero se debe colocar el ADC correspondiente, configurándolo a 700 [MHz] en modo *interleave*.

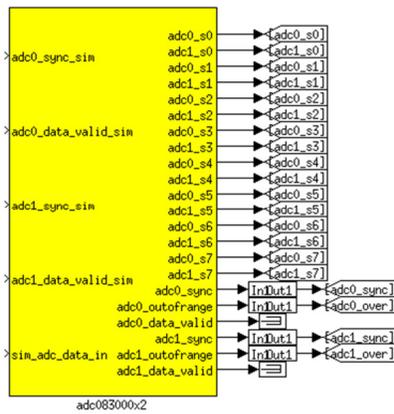


Figura 33: Representación del bloque ADC en MATLAB-Simulink®.

Cabe destacar que las salidas con el nombre *adc0* corresponden a un ADC y *adc1* corresponden al otro ADC. Ahora se debe crear un PFB por cada ADC. El PFB se implementará como se muestra en la siguiente figura:

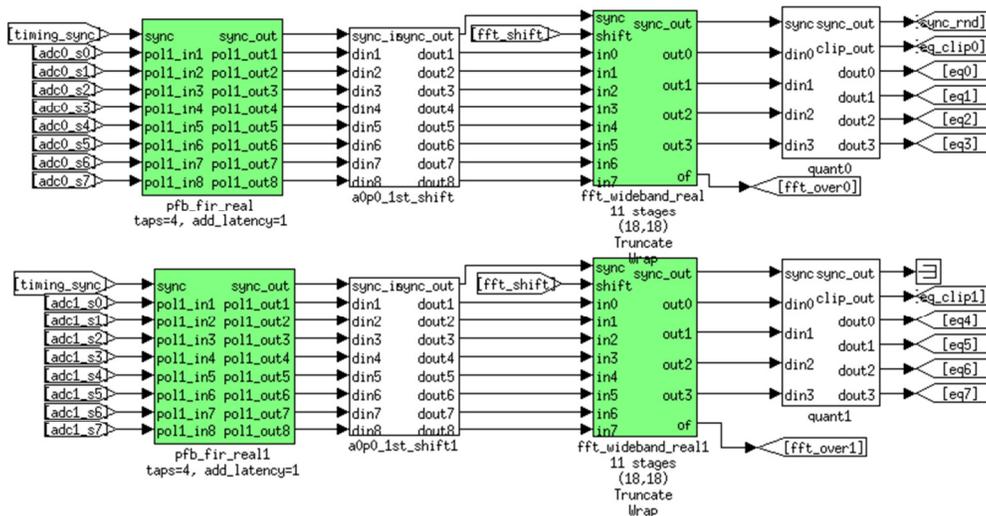


Figura 34: Bancos de Filtros Polifásicos implementados en MATLAB-Simulink®.

Los bloques *pfb_fir_real* deben tener un tamaño de 2^{11} puntos y 4 *taps* y los bloques *fft_wideband_real* también deben tener 2^{11} puntos y coeficientes de 18 bits. Además se ha agregado un bloque cuantificador por cada rama (*quant*) para remover el bit extra que ha aparecido en el proceso de PFB y FFT. Luego cada parte de la señal llevada al dominio de frecuencia será multiplicada y acumulada por el siguiente subsistema:

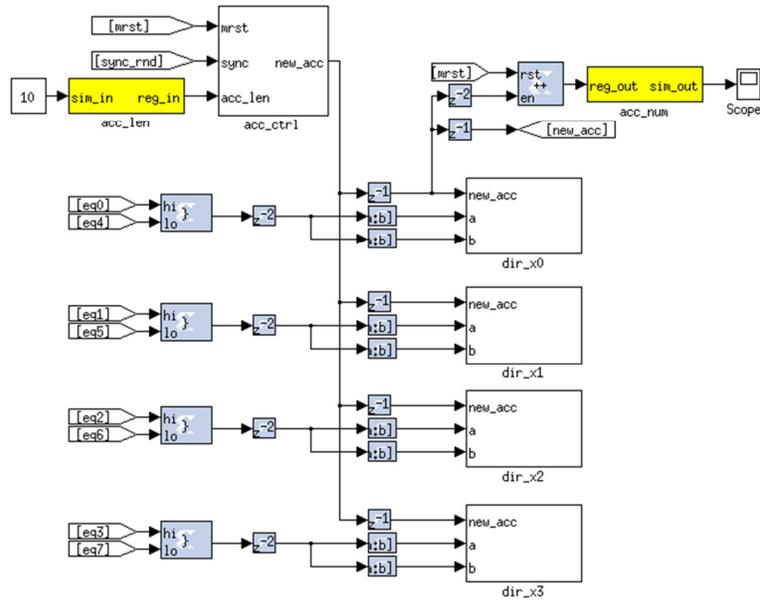


Figura 35: Etapa MAC implementada en MATLAB-Simulink®.

La auto-correlación y la correlación cruzadas se llevan a cabo en los bloques *dir_x*, donde cada uno es configurado con 512 canales. Los nombres de los registros de las multiplicaciones son, como por ejemplo, *dir_x0_ab_real* o *dir_x0_ab_imag* para el caso de la correlación cruzada, o bien *dir_x0_aa_real* o *dir_x0_bb_real* para el caso de la auto-correlación. Esta información se debe tener en cuenta para desarrollar el script que extraerá la información de la ROACH.

Además de tener implementada la estructura general del correlador, se deben usar estructuras extras que permiten tener un control más específico sobre éste y obtener otros parámetros relacionados con la señal. Uno de estos subsistemas es el que permite calcular los valores RMS de las señales de entrada de los ADCs, el cual se puede apreciar en la siguiente figura:

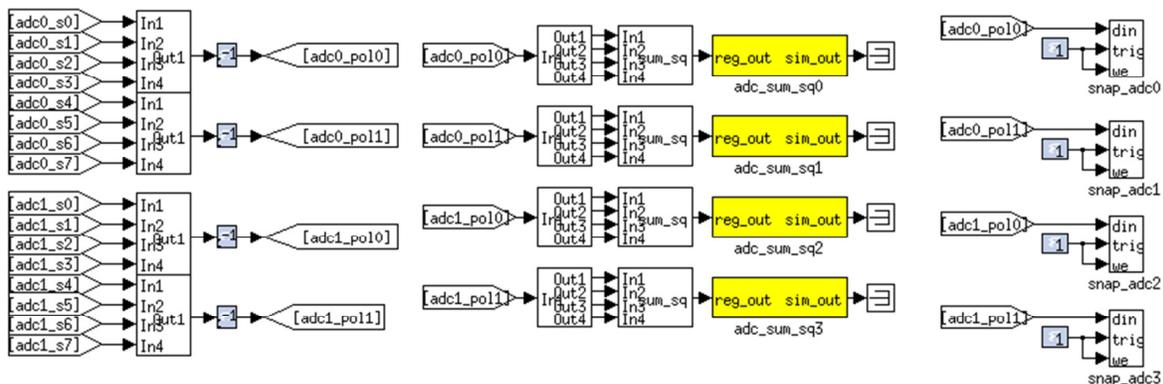


Figura 36: Registros RMS implementados en MATLAB-Simulink®.

Para detectar y reportar la saturación de las señales (*clipping*) al software, se desarrolló creando un subsistema que contiene un enclavamiento de todas las entradas, dicho sistema se muestra a continuación:

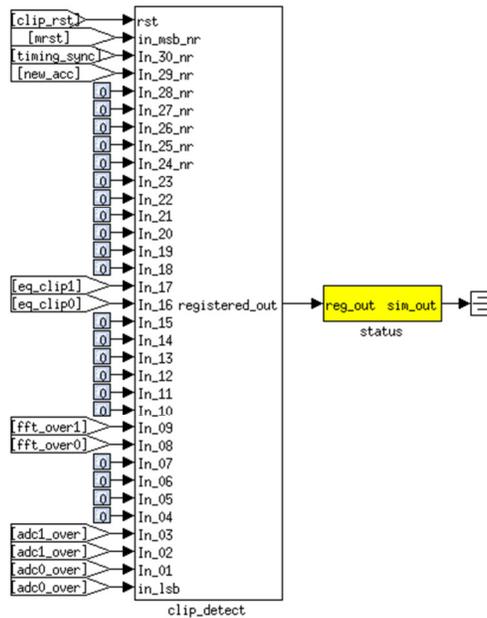


Figura 37: Detector de saturación implementado en MATLAB-Simulink®.

Finalmente, se agregará un subsistema que administre los LEDs disponibles en la ROACH (Figura 38) para monitorear si se produce *clipping*, el pulso de sincronización, al completar una acumulación y la señal que resetea la memoria (*mrst*).

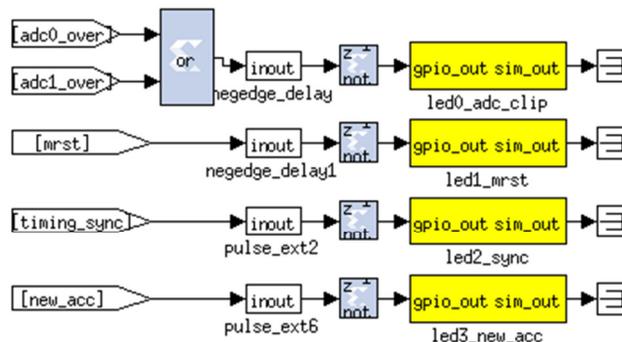


Figura 38: Subsistema de control de LEDs implementado en MATLAB-Simulink®.

Con los bloques interconectados y el diseño finalizado, se procede a compilar el modelo para luego obtener el archivo *bof* y programarlo en la ROACH. Este procedimiento se realizará mediante un script programado en Python que configurará automáticamente la ROACH con el archivo *bof* deseado.

3.2.1.2. Software de Control

Para controlar el correlador y obtener los datos guardados en la ROACH se diseñaron básicamente tres scripts programados en Python. Es importante destacar que para su correcto funcionamiento se deben los siguientes módulos instalados en el sistema operativo:

- Python 2.7.3
- SHM 2.1.1
- Python Setuptools 2.1.1
- NumPy
- PyGTK2
- PyLab
- Matplotlib
- Corr
- KATCP Python Client
- Construct
- INPARSE-04
- GNUplot
- Telnetlib

Una vez instalados los módulos, se podrán ejecutar sin inconvenientes los scripts. En caso de que haya algún error, en la consola de Ubuntu se podrá recopilar información por si existen módulos faltantes, y cuales son.

Los scripts desarrollados tienen como referencia los proporcionados por CASPER para entender el uso de los módulos y las librerías necesarias, además de servir como ejemplo para comprender el código que permite la conexión entre el PC y la ROACH. Para una mayor comprensión de dichos scripts se adjuntarán los códigos en el capítulo de anexos de la memoria y se procederá a describir en forma general la estructura de cada uno.

a) Script de configuración del correlador (*corr_conf.py*)

Este es el script encargado de programar la FPGA en la ROACH con el modelo de correlador ya compilado. La primera parte del código corresponde a importar las librerías necesarias, para luego definir las variables importantes en la conexión con la ROACH como lo es la dirección MAC de ésta, la IP y el puerto de escucha, además de definir el archivo *bof* a programar en la ROACH. Después se definen los parámetros del modelo ya compilado como lo son la ganancia de los canales y la cantidad de acumulaciones a realizar. La segunda etapa corresponde a inicializar la escritura del *bof* en la FPGA, configurar internamente el período de acumulación y resetear los registros de los contadores en la placa. Finalmente se escribe en los registros del correlador la ganancia de todos los canales previamente definidos en los parámetros iniciales.

b) Scripts para graficar (*plot_corr_auto.py*, *plot_corr_cross.py*)

Para graficar los espectros tanto de la auto-correlación de cada antena como el de la correlación cruzada entre ambas, además de añadir la gráfica de la fase de cada canal; se diseñaron dos scripts con estructuras muy similares. En la primera etapa se definen los parámetros de la ROACH a conectar. Luego se definen los métodos que extraerán los datos de los registros de

la ROACH y que graficarán dichos datos. Finalmente se genera el código principal que se conectará a la ROACH ya definida, extraerá y graficará los datos con los métodos ya creados y estará atento a la interrupción del teclado para frenar el proceso.

3.2.2. Espectrómetro VEGAS

Este espectrómetro fue diseñado para el GBT (Green Bank Telescope) para alcanzar un mayor ancho de banda y una mejor resolución espectral ante las nuevas necesidades presentadas por la adquisición de nuevos instrumentos. Para efectos de la memoria, dicho espectrómetro posee internamente un correlador que realiza la auto-correlación y la correlación cruzada de los datos recibidos por dos antenas, además de presentar un ancho de banda máximo determinado solamente por la capacidad del ADC, en este caso de 1.5 [GHz].

En este punto se explicará el funcionamiento de VEGAS, para luego presentar la estructura general del modelo de éste implementado en MATLAB-Simulink® y se detallará la instalación y ejecución del software necesario para el funcionamiento del espectrómetro. Si se desea entender de forma más profunda y detallada el proyecto de VEGAS o se requiere profundizar en un concepto expuesto en la información entregada a continuación, se recomienda revisar la reseña bibliográfica al final de la memoria que lo alude.

3.2.2.1. Funcionamiento del espectrómetro

El espectrómetro está diseñado internamente de forma similar a la estructura propuesta al comienzo del Capítulo 3, sólo que existen cambios sustanciales en el manejo de los datos a nivel interno, los registros y la forma en que se procesa la información una vez ya obtenidas las correlaciones.

Para comenzar, el espectrómetro puede funcionar de 17 modos, de los cuales se utilizará el modo 01 correspondiente a un espectrómetro para 2 antenas. Los modos se pueden dividir en dos subcategorías: el primero para un ancho de banda mayor o igual a 250 [MHz] o HB, y el segundo para un ancho de banda menor a 250 [MHz] o LB. Para los modos HB, la FPGA en la ROACH funciona como un espectrómetro, calculando la FFT, correlacionando, integrando y enviando el espectro integrado al hardware (CPU y GPU) en el PC (HPC) para más tarde realizar un post-procesamiento de los datos y almacenamiento respectivo. En modo LB, la FPGA simplemente empaqueta la señal IF procesada por los ADCs antes de enviarla al HPC, donde allí la GPU se encargará de realizar la FFT antes de escribir en el disco.

La información enviada por la ROACH al HPC se realiza por medio de una conexión de 10Gbe, la cual una vez ingresada al sistema es almacenada en buffers de memoria compartidos para más tarde poder ser procesadas tanto por la GPU como por la CPU.

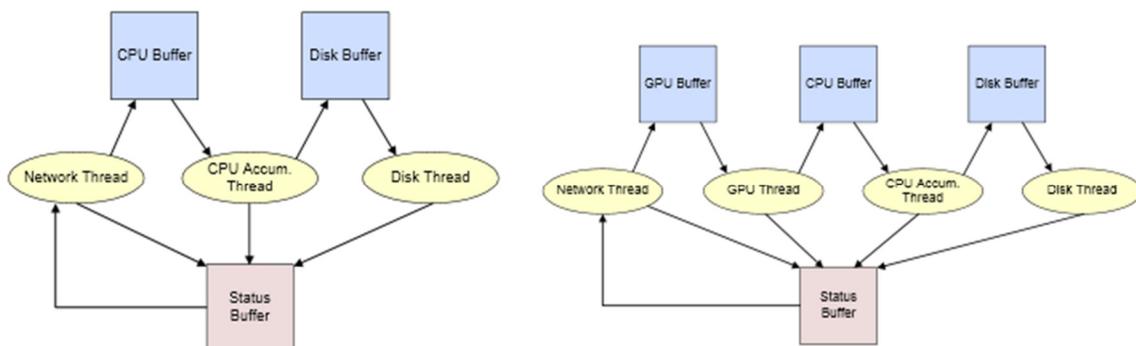


Figura 39: Esquema de buffers de memoria para modo HB (izquierda) y modo LB (derecha).

Los datos son enviados por la ROACH al HPC mediante paquetes UDP, los cuales son formateados de acuerdo al protocolo SPEAD (Streaming Protocol for Exchanging Astronomical Data). El software encargado del procesamiento de la señal por la GPU y CPU será detallado posteriormente en la Sección 3.2.2.3.

0x53	0x04	0x03	0x05	0x0000	0x000A
1	0x001	Counter			
1	0x002	Heap size			
1	0x003	Offset			
1	0x004	Packet payload length			
1	0x020	Time stamp			
1	0x021	Spectrum counter			
1	0x022	Integration size			
1	0x023	Mode			
1	0x024	Status bits (4 bits)			
0	0x025	Payload data offset			
Payload data packet					

Figura 40: Formato SPEAD de los paquetes en modo HB.

3.2.2.2. Análisis del Modelo en MATLAB-Simulink®

El diseño de VEGAS en MATLAB-Simulink® no se aleja demasiado del correlador de 700 [MHz] en cuanto a las ramas del PFB concatenados inmediatamente después del bloque que representa al ADC; pero existen diferencias en cuanto a los bloques implementados para realizar nuevas tareas o modificar las ya existentes, como es el caso del proceso MAC.

El nuevo bloque añadido en este diseño para realizar las multiplicaciones de las diferentes señales resultantes después de aplicar la FFT se muestra a continuación:

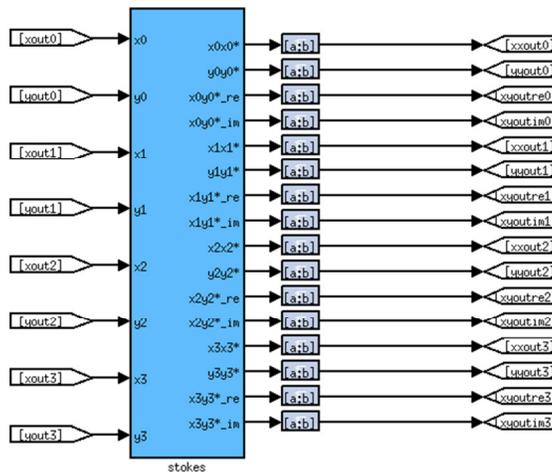


Figura 41: Bloque "stokes" que realiza las multiplicaciones entre las 8 señales resultantes de la FFT.

Se puede apreciar que las salidas del bloque de la Figura 41 corresponden a los productos entre las partes reales e imaginarias de ambas señales, junto con la multiplicación cruzada de éstas, entregando tanto la parte real como imaginaria. Luego en la etapa de acumulación participan cuatro bloques idénticos, los cuales además de estar conectados a cada salida del bloque “*stokes*”, poseen una señal de sincronización idéntica para controlar los tiempos de acumulación. Uno de los bloques se puede apreciar en la siguiente figura:

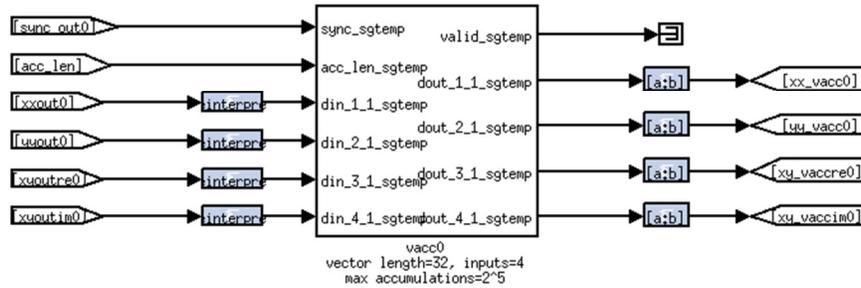


Figura 42: Uno de los 4 acumuladores implementados en MATLAB-Simulink®.

Una vez acumulado el número de vectores previamente definido, todas las salidas de los cuatro acumuladores serán concatenadas en serie por el siguiente bloque:

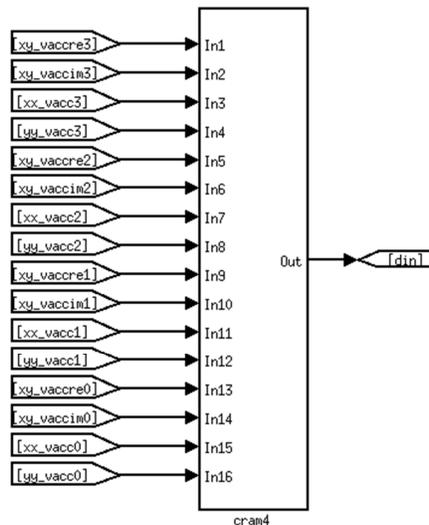


Figura 43: Bloque encargado de concatenar las salidas de los acumuladores.

Al tener todas las señales enlazadas a una salida, ésta podrá ser reestructurada por el subsistema “*reshaper*”; el cual primero recoge toda la información que viene en serie y la reestructura con un número menor de bits (o manteniendo el tamaño original) para luego ser separada en varios paquetes, tal que entremedio de éstos se pueda insertar un encabezado que los identifique o simplemente añada información adicional a la entregada por los acumuladores. El esquema del subsistema encargado de realizar la reestructuración se detalla en la siguiente ilustración:

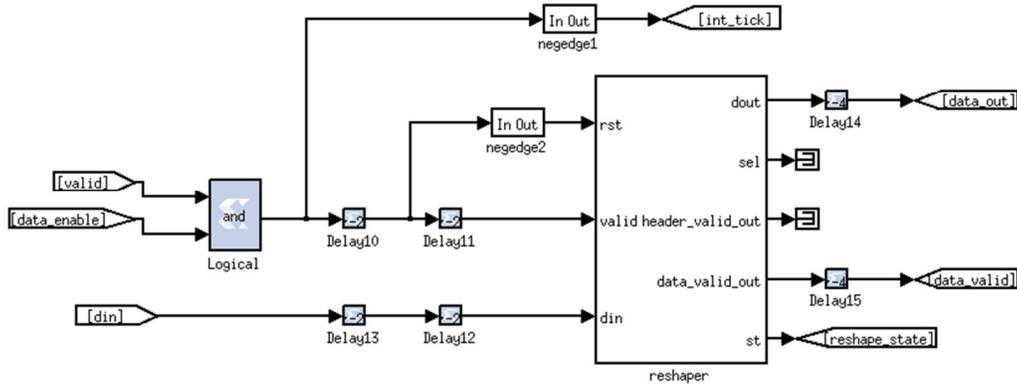


Figura 44: Subsistema de reestructuración implementado en MATLAB-Simulink®.

Los datos ahora pasarán por el proceso encargado de darle el formato SPEAD requerido para su posterior extracción y post-procesamiento. El bloque “speader” es el encargado de dicho proceso, y se puede apreciar directamente que sus entradas corresponden a los encabezados del formato expuesto en la Figura 40. El subsistema se puede apreciar a continuación:

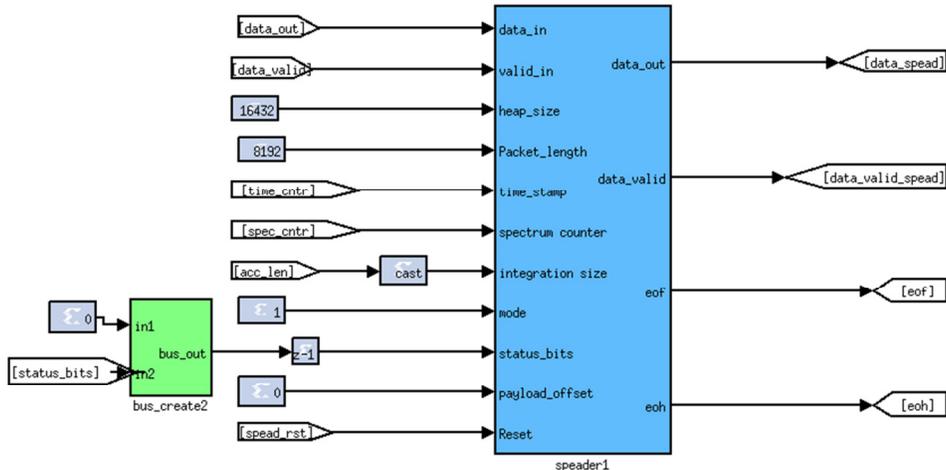


Figura 45: Subsistema de formato SPEAD implementado en MATLAB-Simulink®.

Finalmente se tiene el subsistema encargado de enviar los datos mediante el puerto de 10Gbe, el cual se expone en el siguiente esquema:

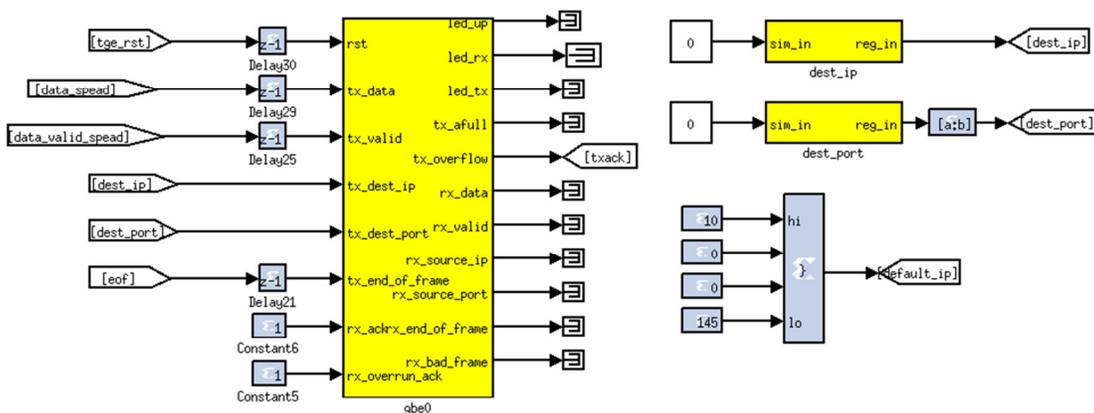


Figura 46: Subsistema de direccionamiento de la data al puerto de 10Gbe en la ROACH.

3.2.2.3. Software de Control

A diferencia del primer diseño de correlador, VEGAS posee un software bastante complejo, ya que no solamente se tratan de scripts de control, ahora más bien se tendrán una variedad de librerías, módulos y programas encargados de procesar los datos recibidos mediante un puerto de 10Gbe. El software se puede dividir en dos grupos, uno que posee la finalidad de suministrar los paquetes y módulos necesarios para la correcta instalación del software de control de VEGAS; y el otro que tiene como objetivo controlar y preparar tanto el PC como la ROACH para recibir los datos.

Cabe destacar que el procedimiento de instalación y ejecución del software, aun siendo indicado por la página del proyecto VEGAS, ésta lo hace a un nivel muy superficial; el cual puede presentar diversos problemas al momento de su ejecución. Por lo tanto, gran parte de éste punto será utilizado en señalar el correcto procedimiento de instalación del software y poder indicar las consideraciones necesarias para afrontar posibles fallos en la instalación de los programas.

Los programas y módulos necesarios se encuentran detallados e indicando su función principal en el siguiente listado:

- **Python 2.5 y 2.7**

Python es un lenguaje ampliamente conocido por su estructura que lo hace ideal para programación de métodos numéricos y otras aplicaciones. Se necesitarán ambas versiones ya que cada una posee ciertas características únicas y necesarias para el proceso de obtención y procesamiento de datos del correlador programado en la ROACH.

- **Librerías de Python**

Python utilizará varias librerías que no se encuentran por defecto en Ubuntu, en las cuales se encuentran *python-setuptools*, *shm-1.2.2*, *python2.5-dev*, *numpy*, *pyfits* y *pygtk2*. Todas estas librerías participan en el mismo proceso que Python, es decir en las etapas de obtención y procesamiento de datos.

- **Python MySQL**

A diferencia de las librerías previamente listadas, Python MySQL es un paquete que permite el control y manejo de bases de datos de forma rápida y efectiva aprovechando las poderosas ventajas que posee Python en el manejo numérico. Participa principalmente en la etapa de procesamiento de datos.

- **CFITSIO**

CFITSIO es un conjunto de librerías que contiene subrutinas programadas en C y en Fortran que permiten la escritura y lectura de archivos de datos en formato FITS (Flexible Image Transport System).

- **pySLALIB**

Es una librería especial que contiene todas las funciones en Fortran de la librería SLALIB (librería de astronomía posicional) traspasadas a Python. Es esencial para el funcionamiento

correcto de la mayoría de los programas que participan en el proceso de extracción y procesamiento de los datos obtenidos por el correlador.

- **FFTW3**

Contiene una librería cuyas subrutinas están programadas en C, y sirve principalmente para calcular la DFT (Discrete Fourier Transform) en una o más dimensiones con una entrada de tamaño arbitrario, ya sea datos complejos o reales. Su aplicación es necesaria en la etapa de procesamiento de datos.

- **Controladores CUDA y el kit de desarrollo de CUDA**

Para poder usar la GPU en la etapa de post-procesamiento de datos, se necesitan los controladores correctos y las herramientas para poder programar y desarrollar cálculos usando los procesadores CUDA que vienen en el chip de procesamiento de video.

Ahora bien, se procederá a explicar el proceso de instalación de cada elemento sin entrar en los detalles del código para hacerlo (para ello se requiere un conocimiento básico de sistemas operativos basados en Linux), además de plantear los posibles problemas y las soluciones que éstos pueden tener. Además se recomienda fuertemente instalar todos los componentes del software enumerados, en el mismo directorio.

1. Primero se debe comenzar con la instalación de ambas versiones de Python. Se recomienda bajar las últimas actualizaciones de cada versión, ya que pueden contener algunas modificaciones en algunos módulos necesarios para la instalación posterior de otras librerías. Además se deben instalar en conjunto los módulos *dev* de ambos Python (*pytho2.5-dev* y *python27-dev*). Es muy importante que una vez instaladas ambas versiones de Python, la mayoría de los módulos posteriores se deben instalar usando Python 2.5, ya que existen procesos incompatibles con la versión 2.7 que puede generar errores. En caso de ser necesario, se informará si el paquete debe ser instalado, además, en la versión 2.7.
2. Instalar el módulo *shm-2.1.1*. Este proceso no debería presentar problemas.
3. Ahora se deben instalar el paquete *python-setuptools* para ambas versiones de Python, y para ello se recomienda descargar los códigos fuentes del paquete para ambas versiones en la página web oficial. Durante este proceso se pueden encontrar algunas complicaciones; la primera de ellas es que el instalador puede no encontrar algunas librerías instaladas en el sistema, por lo que se deben instalar los paquetes *zlib* y *lib32* manualmente. El segundo problema es que el instalador no encuentra el módulo *_sha254*, por lo que hay que descargar e instalar la librería *hashlib*.
4. Se procederá a instalar el paquete *MySQL-python-1.2.3* para ambas versiones de Python en forma manual, descargando el código de fuente del sitio web oficial. En esta etapa no debería haber errores en el instalador, pero en el caso de que se produzcan, se recomienda revisar que el paquete *python-setuptools* se encuentre correctamente instalado.

5. Uno de los paquetes más importantes es *numpy* ya que posee un lugar alto en la jerarquía de dependencias para la instalación y el funcionamiento del resto del software de VEGAS, el proceso de instalación debe realizarse para ambas versiones desde el código de fuente. La instalación en si no debería generar errores, pero existe la posibilidad de que haya que hacerle ciertas reparaciones más adelante.
6. Ahora corresponde instalar *pyfits-2.3.1*, el cual no deberá presentar problemas.
7. Se procederá con la instalación de *pygtk-2.24.0*. Es muy probable que el paquete no encuentre algunas librerías en el sistema, por lo que se deberán instalar manualmente, las cuales son: *libglib2.0-0*, *python-gobject-2-dev* y *python-gi-dev*.
8. El proceso de instalación continúa con *pyslab*, el cual no presentará problemas. Una vez instaladas las librerías se debe copiar el archivo *libsla.so* ubicado en el directorio raíz del instalador, en la carpeta *site-packages* del directorio donde se encuentra instalado Python 2.5.
9. Se instalará el paquete *fftw-3.2.2* el cual no deberá presentar errores.
10. Para terminar con la instalación de paquetes necesarios para el buen funcionamiento del software de VEGAS se deben instalar los controladores CUDA de la GPU, descargando la versión más reciente del sitio web oficial de NVIDIA junto con las instrucciones de su instalación.
11. Entrando al proceso de instalación del software de control de VEGAS, primero se debe descargar el código fuente de los instaladores en el sitio web del proyecto. Se añadirá una nueva variable de entorno en el script *.bash_profile* con el nombre de VEGAS, y su ruta direccionando a la carpeta *src* contenida en el directorio raíz del código fuente descargado.
12. En la carpeta *vegas_hpc* se encuentra un script (*vegas_spec-hpc_udch.bash*) que crea las variables de entorno necesarias para el proceso de instalación del software de control. Dicho script debe ser modificado de acuerdo a las nuevas rutas de las librerías instaladas en los puntos anteriores.
13. Para proceder en el proceso de instalación se debe ejecutar exactamente el siguiente código:

```
source vegas_spec-hpc_udch.bash
python2.5 setup.py install --install-lib=$VEGAS/lib/python
python2.5 setup.py install --install-lib=$VEGAS/lib/python
```
14. Luego, en el directorio raíz, se ubica la carpeta *src* contenida en *vegas_hpc* para proceder a modificar el archivo *Makefile*. Primero, en la línea donde se definen las librerías a usar se debe eliminar *"-lg2c"*. Ahora se crea una variable que haga referencia al compilador Fortran de Ubuntu, se recomienda escribir: *"FC = gfortran"*. Finalmente en la parte principal del código se cambian todos los compiladores de C (CC) por el compilador Fortran (FC). Con estos cambios se procede a instalar el *Makefile*.

15. Para probar que el software haya sido correctamente instalado se procede nuevamente a ejecutar el script *vegas_spec-hpc_udch.bash*, y luego se ejecutará el script encargado de generar los semáforos y los bloques de memoria compartidos por los distintos *threads*: *vegas_setup_shmem_mode01*. Si se encuentran problemas al ejecutar dicho código los problemas pueden tener diversas razones. Una de ellas es que el script se ejecute con Python 2.7 en vez del 2.5, entonces se abre el código y se cambia la versión de Python. Otra razón se puede deber a que la instalación de *numpy* haya quedado defectuosa en los módulos “*curses*”, y para solucionarlo se debe obtener una instalación “limpia” de repositorio de Python 2.5, para luego buscar los módulos *_curses.so* y *_curses_panel.so*, los cuales deberán ser copiados en la carpeta *lib_dynload* del directorio de instalación de Python 2.5. Finalmente se recomienda revisar todos los scripts para que se ejecuten con Python 2.5 y no 2.7. Si los problemas persisten y no se generan los bloques de memoria compartida o los semáforos necesarios, se deben cambiar las siguientes líneas (o añadir en el caso que no existan) en el archivo de sistema “*sysctl.conf*”

```
net.core.rmem_max = 2147483648
net.core.rmem_default = 536870912
net.core.netdev.max_backlog = 50000
kernel.shmmax = 2147483648
kernel.shmall = 2097152
kernel.sem = 1024 32000 32 32767
```

16. Ahora se procede a ejecutar el script *vegas_status_monitor_py*, el cual debería mostrar la siguiente imagen si el software de VEGAS está correctamente instalado:

```
Current VEGAS status:
SRC_NAME : unknown          OBSERVER : root
RA_STR   : 00:00:00.0      DEC_STR  : +00:00:00.0
TELESCOP : GBT            FRONTEND : none
PROJID   : test           OBSFREQ  : 1200.0
OBSBW    : 800.0         CAL_MODE : OFF
INSTRUME : VEGAS         PKTFMT  : SPEAD
DATAHOST : 192.168.0.101  DATAPORT : 60000
CAL_FREQ : 25.0          CAL_DCYC : 0.5
CAL_PHS  : 0.0          OBSNCHAN : 2048
NCHAN    : 1024         CHAN_BW  : 585938.0
NPOL     : 2            NBITS    : 8
NBITSADC : 8            DATADIR  : .
TSYS     : 0            NSUBBAND : 1
ELEV     : 0            OBJECT   : unknown_obj
BMAJ     : 0            BMIN    : 0
BPA      : 0            EXPOSURE : 2e-05
SUB0FREQ : 2400000000.0  SUB1FREQ : 1000000000.0
SUB2FREQ : 1000000000.0  SUB3FREQ : 1000000000.0
SUB4FREQ : 1000000000.0  SUB5FREQ : 1000000000.0
-- Increase window size -- -- Increase window size --
Last update: Sun Mar 10 16:59:38 2013 - Press 'q' to quit
```

Figura 47: Ventana de monitoreo del software de VEGAS.

17. Finalmente se procede a ejecutar el script de configuración de la ROACH “*vegas_config.py*”, el cual requiere las mismas librerías que el script de configuración del correlador de 700 [MHz]. Una vez la ROACH esté programada con VEGAS, se procede a inicializar los programas *vegas_hpc_hbw* o *vegas_hpc_lb* dependiendo del modo que se vaya a utilizar, ya sea conectado a la FPGA o en modo simulación. Se recomienda revisar el número de procesadores que tiene el PC donde VEGAS está instalado, ya que es necesario

corregir los siguientes scripts para el correcto funcionamiento del sistema: *vegas_accum_thread*, *vegas_net*, *vegas_pfb* y *vegas_sdfits*.

Con el software de VEGAS instalado correctamente es posible obtener el espectro ya correlacionado en formato FITS para luego ser graficado o post-procesado. Dichos procesos pueden ser llevados por cualquier programa capaz de leer e interpretar archivos en formato FITS. Queda a discreción del usuario qué software utilizar para recuperar el espectro.

3.3. Prueba de los Diseños

En el contexto del experimento global que está insertado el correlador, previamente especificado en la Sección 2.1.6 del Capítulo 2, se tienen tres tipos de montajes, uno óptico y dos eléctricos. Posteriormente se explicarán dichos montajes y en qué consisten las pruebas que fueron llevadas a cabo en éstos. Además se señalarán las condiciones en que se realizaron las pruebas, tanto del estado de los instrumentos como las posibles fuentes de interferencia.

A grandes rasgos, todas las pruebas a ejecutar se basan en generar tonos de distintas frecuencias y observar la gráfica del espectro en la auto-correlación y en la correlación cruzada, además de observar el comportamiento de la fase en los distintos canales.

Dentro de los resultados esperados se pretende observar principalmente un “*spike*” en la frecuencia del tono de entrada en la auto-correlación de ambas antenas, y en la correlación cruzada el mismo *spike* que en el gráfico anterior. Ambos resultados deben exponer un nivel de ruido relacionado tanto con el diseño de las pruebas, como con las fuentes. En la Figura 48 se puede apreciar el espectro cruzado entre dos antenas:

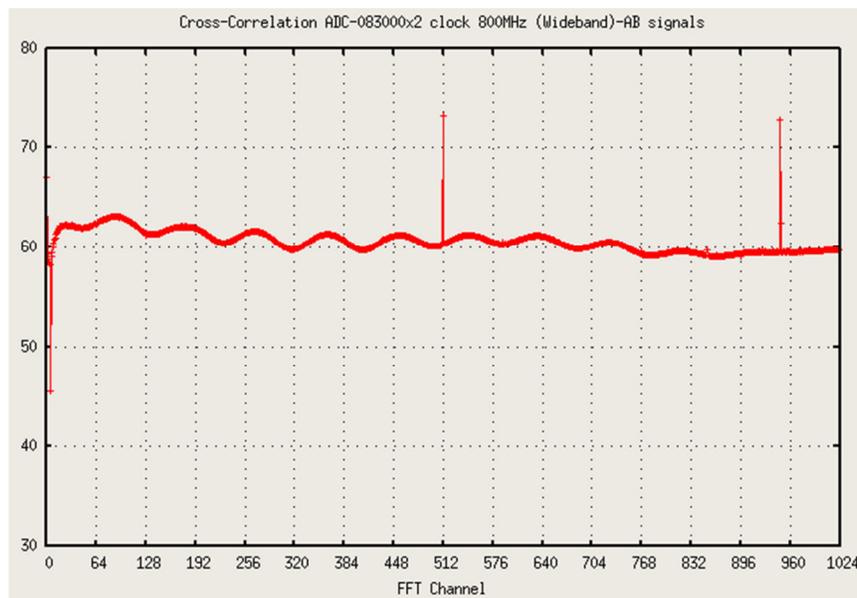


Figura 48: Gráfico de correlación cruzada.

En la ilustración anterior se pueden apreciar dos *spikes*, el primero en el canal central y el segundo cerca del canal 960. El *spike* central se debe a un problema interno de los ADCs, su acoplamiento con la ROACH y la señal de reloj que los alimenta, por lo que siempre se tendrá una alta potencia en la frecuencia central de la banda observada. El segundo *spike* es el producto de

una señal monocromática idéntica a la entrada de los ADC, lo que se traduce en que el correlador ha detectado dos señales de igual frecuencia en cada ADC ubicadas en un canal cercano al 960 en la FFT. También se puede observar que existe un nivel de ruido cercano a las 60 unidades de potencia, el cual posee características de ruido blanco dado que las señales de prueba fueron mezcladas con un generador de este tipo de interferencia. Las unidades de potencia son arbitrarias y de gran magnitud dado que el proceso de acumulación interna de la ROACH, suma y multiplica un gran número de veces el espectro acumulado de potencia, antes de entregar el espectro final correspondiente a una iteración. En el caso de que se desee tener una medida real de la potencia, se recomienda calibrar el *script* que extrae y grafica los datos del correlador, con un valor conocido de potencia de entrada. De esta forma se determina un nivel de referencia para futuras mediciones de la potencia real en las entradas.

Cabe destacar que de los dos modelos implementados, sólo se le podrán realizar las pruebas al correlador de 700 [MHz], dado que el hardware necesario de VEGAS para la comunicación entre la ROACH y el PC aún no se encuentra disponible en el laboratorio, por lo que se pospondrán su uso y las pruebas de desempeño. Por lo tanto la ROACH y el PC estarán conectados a la red mediante un cable de Ethernet de 100 Mbps de acuerdo a las especificaciones del modelo de correlador de 700 [MHz] para producir el intercambio de datos, los cuales serán procesados y graficados por los scripts correspondientes.

3.3.1. Hardware a Utilizar en los Experimentos

El equipo que se usará en el montaje del experimento eléctrico consta de:

- Sintetizadores de señales: el equipado en la ROACH (Valon 5007), y dos sintetizadores externos extras.

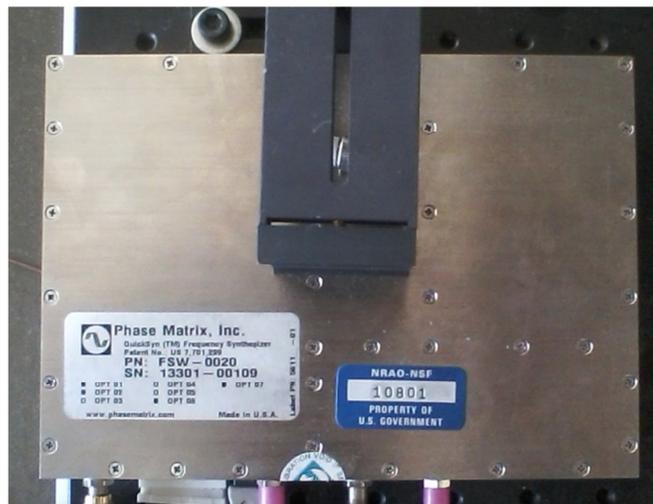


Figura 49: Uno de los sintetizador de frecuencias para generar señal de prueba.

- Cables y conectores SMA para conexiones entre los componentes.
- Conectores BNC/SMA requeridos para generar el divisor T.
- Amplificadores LNA de ganancia 35 [dB] y de 13 [dB].



Figura 50: Amplificador LNA de 39 [dB] de ganancia.

- Fuente de alimentación para los amplificadores.
- Cables de alimentación.
- Divisores de potencia 50/50 de conectores SMA.
- Fotodiodos.
- Fuente de ruido blanco entre los 10 [MHz] y 3 [GHz].
- Sintetizadores láser, uno de alta precisión con frecuencia fija, y otro de frecuencia ajustable.



Figura 51: Sintetizador láser de precisión, con longitud de onda ajustable en pasos de 1 [pm].

- Conectores de fibra óptica del tipo FC/APC y FC/UPC.
- Acopladores 50/50 de fibra óptica.

3.3.2. Montaje de los Experimentos

Las pruebas serán realizadas utilizando un sintetizador de señales controlado por software desde un PC junto con una etapa de amplificación previa a la entrada de la ROACH. El primer experimento tiene como objetivo evaluar el funcionamiento del correlador en su primera puesta en marcha, y caracterizar los posibles fenómenos asociados a las condiciones en las que se realizaron las pruebas. El siguiente esquema ilustra el primer experimento a realizar:

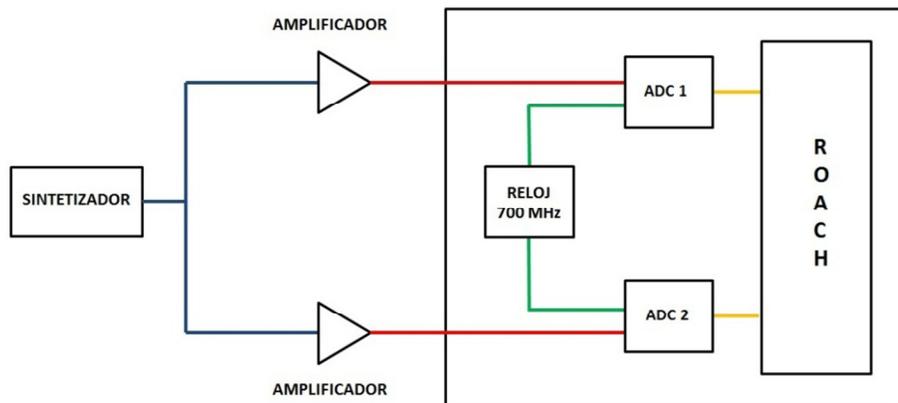


Figura 52: Esquema del primer experimento.

Al comienzo el sintetizador genera una señal de frecuencia fija entre los 170 [MHz] a los 700 [MHz] con una intensidad de -25 [dBm], luego la señal es dividida por un divisor simple tipo T para luego pasar por amplificadores activos de bajo ruido (LNA) con una ganancia de 35 [dB] Posteriormente ambas señales se dirigen respectivamente a cada ADC, los cuales están funcionando con un reloj externo de 700 [MHz] generado por un segundo sintetizador (Valon 5007). Finalmente cada señal entrará en la ROACH para su correlación.

En el segundo montaje se realizarán dos experimentos, ambos con el objetivo de estudiar el comportamiento del correlador bajo condiciones cercanas a las ideales, considerando equipos que produzcan una menor cantidad de ruido, y un ambiente aislado de las posibles señales externas que puedan introducirse en el sistema. Además se pretende estudiar el comportamiento de la fase en la correlación cruzada de las señales de entradas, manipulando las longitudes de los cables SMA usados en su transmisión. Para este experimento se utilizará el siguiente esquema:

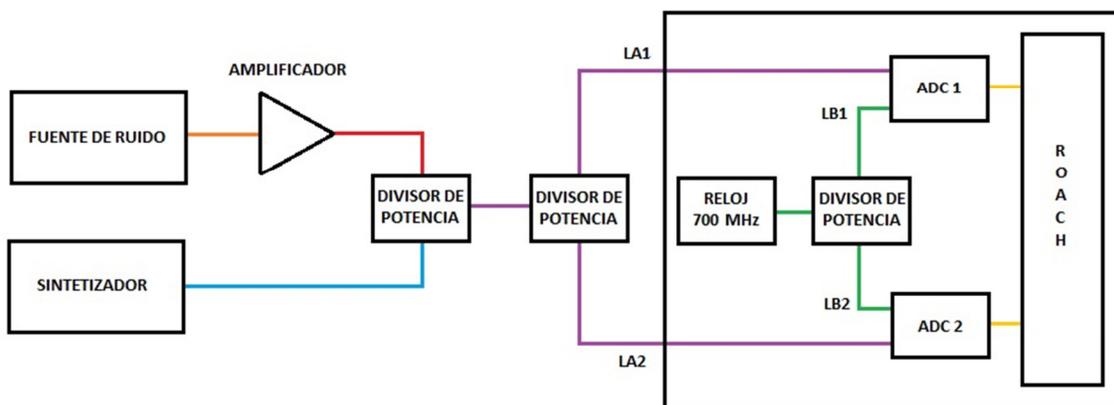


Figura 53: Esquema del segundo montaje.

El primer experimento a realizar en este montaje es similar al expuesto para el primer set de pruebas. La diferencia radica en los equipos a utilizar y las conexiones que se deben llevar a cabo. En la primera etapa se tiene una fuente de ruido blanco operando entre los 10 [MHz] y los 3 [GHz], cuya señal será amplificada en 3 pasos de ganancia 13 [dB] montado en un único circuito de amplificación. Luego dicha señal será mezclada en un divisor de potencia, con el tono generado por el sintetizador de frecuencias; para luego ser dividida nuevamente por un divisor y transportadas a cada ADC mediante los cables SMA LA1 y LA2 respectivamente. Dichos cables poseen distintas longitudes, con una diferencia aproximada de 10 [cm] entre éstos. Otra diferencia particular con respecto al primer montaje, está relacionada con la forma en que se genera la señal de reloj para cada ADC. En éste montaje se tendrá solo una salida del sintetizador interno de la ROACH generando la señal de 700 [MHz] requerida por los conversores análogo-digital, para luego ser dividida por un divisor de potencia, y de esta forma ingresar a cada ADC por las líneas LB1 y LB2 respectivamente. Estos cables SMA poseen entre ellos una diferencia de 2 a 3 [cm].

El segundo experimento a realizar, consiste en cambiar las líneas que transmiten tanto las señales de entrada a los ADC (LA1 y LA2), cómo la señal de reloj (LB1 y LB2) por pares de cables que posean la misma longitud. De esta manera, se pretende estudiar el comportamiento de la fase del correlador, en ausencia de condiciones que generen desfases en las señales y en el procesamiento de datos.

El tercer montaje tendrá como principal objetivo probar el correlador en un experimento de diseño óptico. De esta forma el correlador estará inserto en un experimento más cercano a su función principal, la cual será correlacionar la señal recibida por dos telescopios apuntando hacia alguna fuente luminosa en el cielo. El esquema correspondiente al montaje para el desarrollo del cuarto experimento es el siguiente:

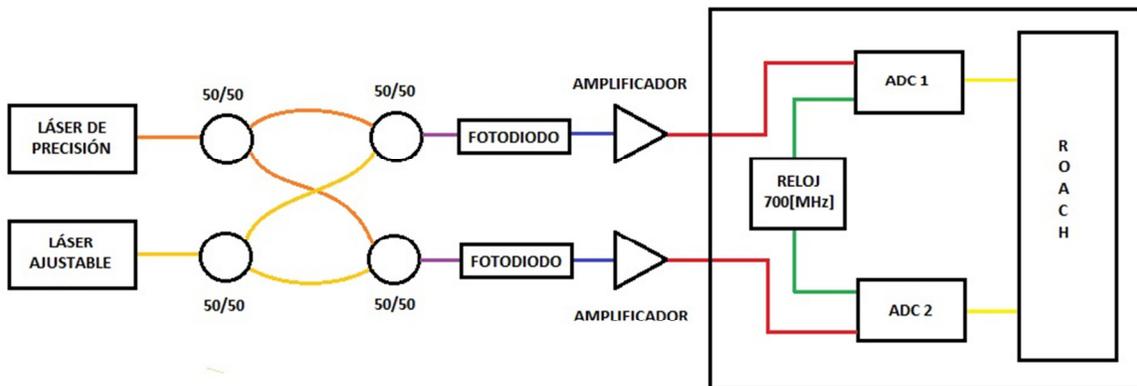


Figura 54: Esquema del tercer montaje.

El cuarto experimento se llevará a cabo utilizando dos sintetizadores láser para generar dos señales de frecuencias muy cercanas; el láser ajustable aportará una señal monocromática para el sistema, mientras que el láser de precisión generará un haz de frecuencia ajustable muy cercana a la anterior. Luego ambas señales se transmitirán mediante un acoplador de fibra óptica para cada una, el cual dividirá la potencia en 2 partes casi idénticas, para posteriormente ser mezcladas en otro conjunto del mismo tipo de acopladores. El haz resultante de la mezcla será guiado en su primera etapa por fibras del tipo FC/APC para luego cambiar a fibras del tipo FC/UPC hasta alcanzar a los fotodiodos; los cuales operan con señales de hasta 2 [GHz]. La razón de este cambio del tipo de conector en las fibras se debe a que los fotodiodos presentan una mayor receptividad de la señal cuando la fibra posee un pulido plano (UPC), en contra de un pulido con una pendiente

de 8° (APC). En los fotodiodos la señal lumínica será transformada a una señal eléctrica, la cual estará caracterizada por la misma frecuencia que el haz incidente. Además ésta no poseerá la presencia de ruido externo, dado que el origen de la señal es netamente del tipo óptico, a diferencia de las señales eléctricas generadas en los dos montajes anteriores. Finalmente cada señal recibida por los fotodiodos será amplificada con una ganancia de 35 [dB], y luego ingresarán a cada ADC para su procesamiento. En la siguiente imagen se puede apreciar el montaje realizado en el laboratorio:

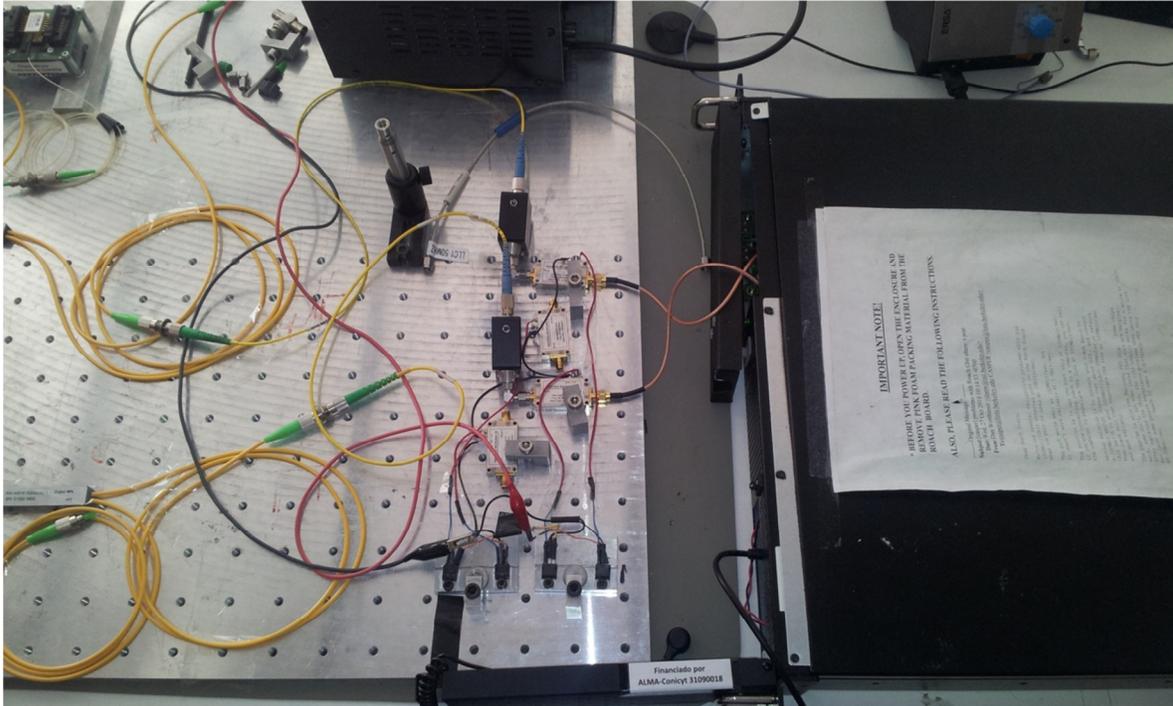


Figura 55: Montaje de experimento óptico implementado en el laboratorio de Fotónica.

3.3.3. Condiciones de las Pruebas

El experimento se realizará en el quinto piso del Departamento de Ingeniería Eléctrica de la Universidad de Chile, en las dependencias del Laboratorio de Fotónica. Dicho laboratorio no cuenta con un aislamiento acorde a la sensibilidad de los instrumentos, pero se espera que las posibles componentes de alta frecuencia proveniente de fuentes de interferencia externas, como antenas de comunicaciones móviles ubicadas en los techos de los edificios circundantes, no influyan considerablemente en las señales o los instrumentos de prueba.

La mayoría de los componentes relacionados con las conexiones entre los equipos (amplificadores, sintetizadores, ADCs) no se encuentran en buenas condiciones dado el continuo uso y tiempo presente en el laboratorio.

Algunos conectores de SMA/BNC no se encuentran en buen estado, estando algunos un poco gastados y quedando desajustados para formar el divisor T, por lo que podrían variar las razones de transmisión/reflexión entre ellos e introducir pequeñas diferencias en las señales de entradas de los ADC, las cuales se esperan que sean relativamente idénticas.



Figura 56: Divisor T simple implementado con conectores SMA/BNC.

Los cables SMA se pueden observar un poco deteriorados, además de no poseer una cobertura aislante apropiada, lo cual puede ser perjudicial si existen señales de bajas frecuencias en el ambiente, las cuales podrían interferir fuertemente en la forma de la señal de entrada de los ADC.

Los amplificadores cuentan con un único sistema de refrigeración en base a conducción por metal, la cual a largo plazo puede provocar saturación térmica por parte del equipo y aumentar la presencia de ruido térmico en las mediciones.

El modelo de correlador es sabido que se satura rápidamente en cuanto a potencia recibida, por lo que si existe una señal con muy alta potencia, y otra cercana al nivel de saturación de los ADC, ambas aparecerán al mismo nivel en el espectro. Dicha saturación se debe a que el tamaño de los datos usados por el correlador no es lo suficientemente grande para representar niveles de potencia muy elevados.

Para verificar la cantidad de ruido presente en la señal de prueba que se utilizará en el primer montaje, se realizó una medición del espectro generado por el sintetizador de frecuencia; usando el analizador de espectros disponible en el laboratorio. Los resultados obtenidos se pueden apreciar en la siguiente figura:

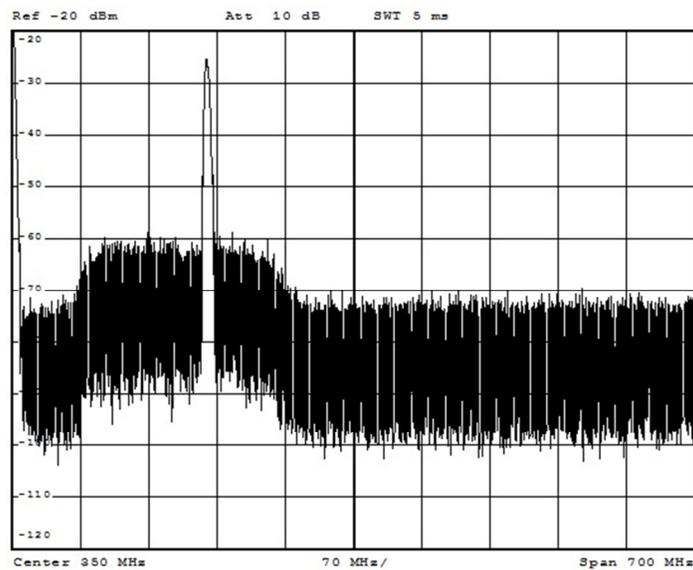


Figura 57: Espectro obtenido generando un tono a 200 [MHz].

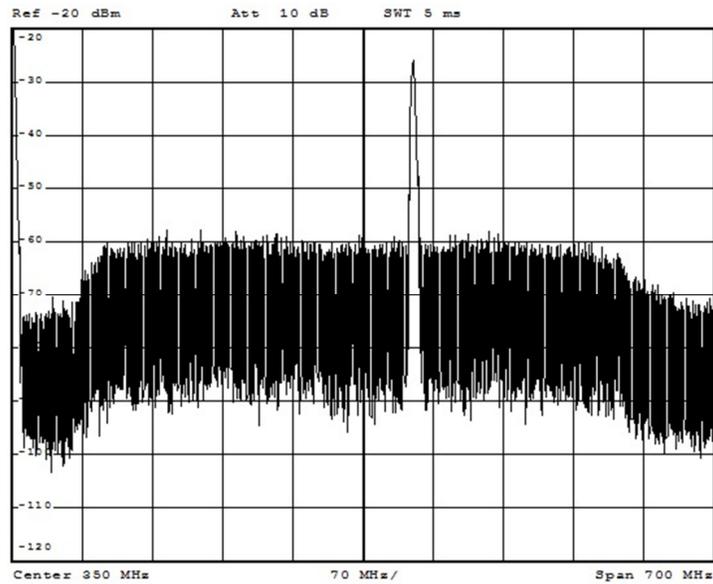


Figura 58: Espectro obtenido generando un tono a 400 [MHz].

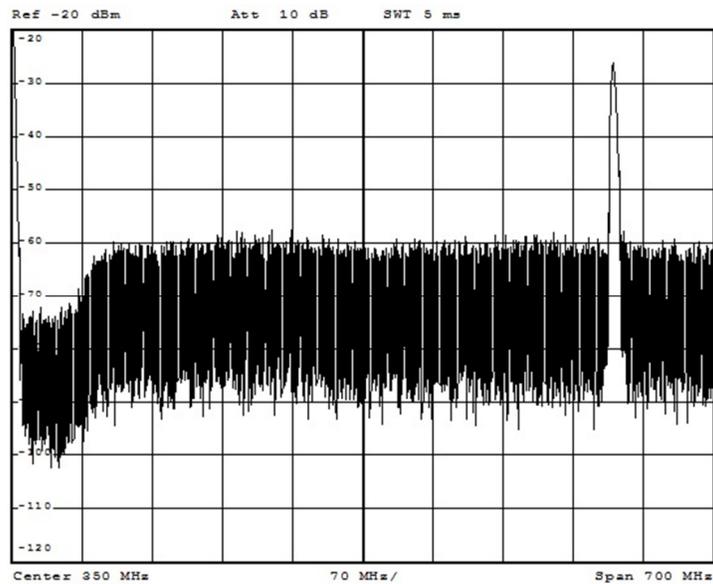


Figura 59: Espectro obtenido generando un tono a 600 [MHz].

Se puede apreciar de los gráficos obtenidos, que a grandes rasgos, el tono generado se encuentra a 30 [dB] por sobre un pedestal de ruido blanco que aparece tanto en los 200, 400 y 600 [MHz]. En la práctica, dicha potencia es aceptable para trabajar con esa relación señal/ruido, pero en el caso del correlador, como fue mencionado anteriormente, éste se satura rápidamente. Por lo que dicho nivel de ruido blanco debería aparecer con magnitudes importantes en los gráficos de las correlaciones.

Finalmente se tiene que al igual que el laboratorio, el hardware que participará en el experimento no cuenta con ningún tipo de blindaje, lo cual puede ser perjudicial si existen componentes de bajas frecuencias en el ambiente, la cual será amplificada y podría deteriora las formas de ondas o los espectros observados.

3.3.4. Procedimiento de Ejecución de Pruebas

Para comenzar a ejecutar las pruebas se debe seguir el siguiente procedimiento, tanto para el primer y segundo montaje, antes de la obtención de datos. Dichos pasos son necesarios para no dañar los equipos durante el experimento, y se asume que la persona encargada de realizarlos debe estar permanentemente utilizando una pulsera anti-estática, la cual estaría conectada a la tierra de los montajes.

1. Se debe asegurar de tener: todos los equipos apagados, las conexiones hechas correctamente, y también que el cableado diseñado para la alimentación de los amplificadores se encuentre en buen estado.
2. Se procede a encender la ROACH y los PC de control del hardware.
3. Se configurará el controlador de los sintetizadores que generará las señales de reloj para los ADC. Se establecen las frecuencias de ambas salidas en los 700 [MHz], con el menor nivel de potencia permitido (-2). Una vez ingresados los parámetros se deben apagar y volver a encender ambos sintetizadores.
4. Se configurará la ROACH con el archivo *bof* a utilizar; para ello se debe ejecutar el script de configuración predeterminado.
5. Se revisarán nuevamente los amplificadores y su alimentación. Una vez en orden se procede a encender la fuente que los energiza.
6. Para finalizar, se configura el sintetizador que generará la señal de prueba. Se debe tener especial cuidado con no activar la salida de éste en el software, hasta disminuir la intensidad de la señal al mínimo (-25 dB). Al no hacerlo se pueden quemar los amplificadores.

Una vez realizado el procedimiento, se pueden ejecutar los scripts para extraer la información de la ROACH y graficar los resultados. De la misma forma, se deben seguir instrucciones similares para el tercer montaje, las cuales son:

1. Se debe asegurar de tener: todos los equipos apagados, las conexiones hechas correctamente, y también que el cableado diseñado para la alimentación de los amplificadores se encuentre en buen estado. Además se pone especial énfasis en revisar que las fibras ópticas no posean ningún tipo de curvatura aguda, ya que éstas las deterioran en forma considerable.
2. Se tendrá un procedimiento idéntico desde el segundo paso hasta el cuarto, indicados anteriormente para el primer y segundo montaje.
3. Se procede a energizar los fotodiodos, insertándoles las baterías y atornillándolas con el cuidado de no dañar el hardware.
4. Se revisarán nuevamente los amplificadores y su alimentación. Una vez en orden se procede a encender la fuente que los energiza.

5. Se activarán tanto el láser ajustable como el de precisión. Se ajustarán ambos a la misma frecuencia (o longitud de onda en su defecto), y se encontrará el tono de forma manual, ajustando sólo el láser de precisión, desplazando su longitud de onda unos pocos nanómetros.

4. Análisis de Resultados

En este capítulo se expondrán los resultados obtenidos en las pruebas realizadas con los montajes previamente detallados en el Capítulo 3. Se comenzará con el análisis del primer montaje de prueba, luego se detallarán los resultados obtenidos en los dos experimentos del segundo montaje, y para finalizar, se analizará el experimento final utilizando el tercer montaje.

4.1. Primer Montaje

En primer lugar se procede a graficar el estado de la ROACH antes de encender los amplificadores, para tener una idea del nivel de ruido interno en los ADC, lo cual es necesario tener en cuenta al momento de interpretar los datos y calibrar el correlador.

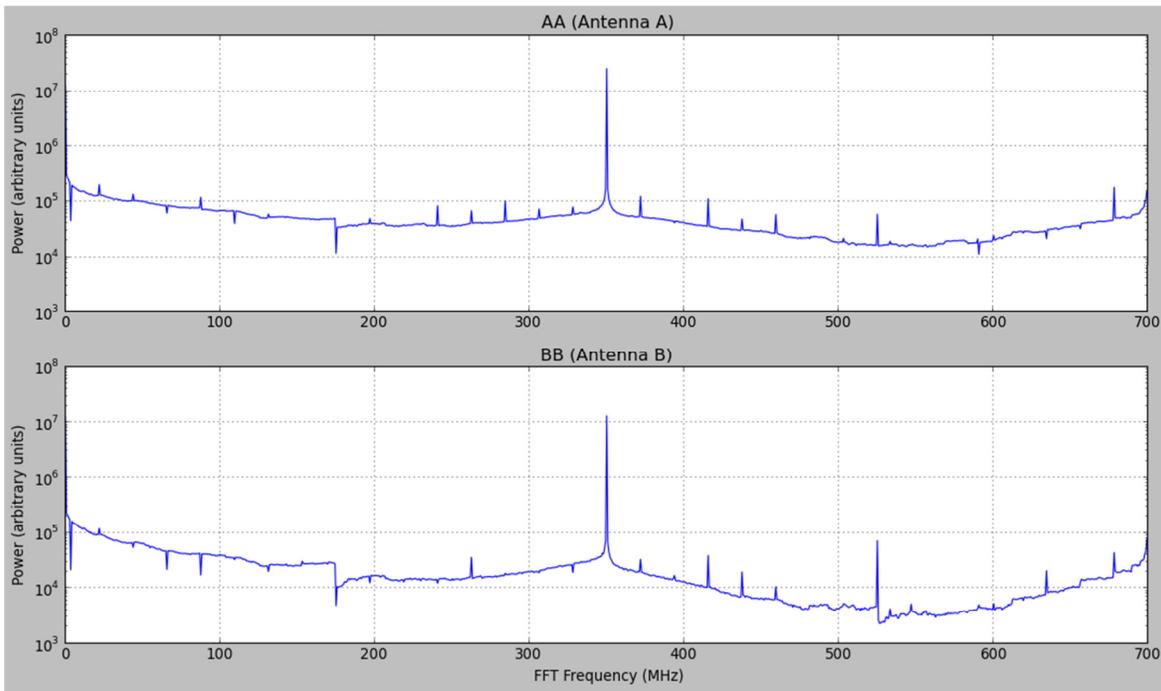


Figura 60: Auto-correlación del ruido interno de ambos ADC.

En la figura de la auto-correlación de ambas antenas se puede apreciar el *spike* del canal central mencionado anteriormente, el cual se espera tener en cada muestra de datos y se debe ignorar. Además se puede apreciar que el nivel de ruido por lo general se encuentra bajo las 10^5 unidades en potencia, y específicamente en este caso se ha configurado la ganancia interna del modelo en un nivel alto simplemente para exponer la estructura del ruido interno. En las futuras

mediciones se bajará la ganancia a un nivel normal (la señal no sobrepasará el nivel de potencia de 10^3 y así poder apreciar los resultados de forma natural. Para esta prueba no se puede considerar la correlación cruzada ni la fase de las señales debido a que existen “ceros” en los datos dado que no hay señal de entrada en los ADC, por lo que no se pueden graficar ya que el script calcula el logaritmo de esos datos y se tienen puntos fuera del dominio de aquella operación.

La segunda muestra está compuesta por la señal detectada al encender los amplificadores, la cual se compone principalmente de ruido externo y el generado por los amplificadores:

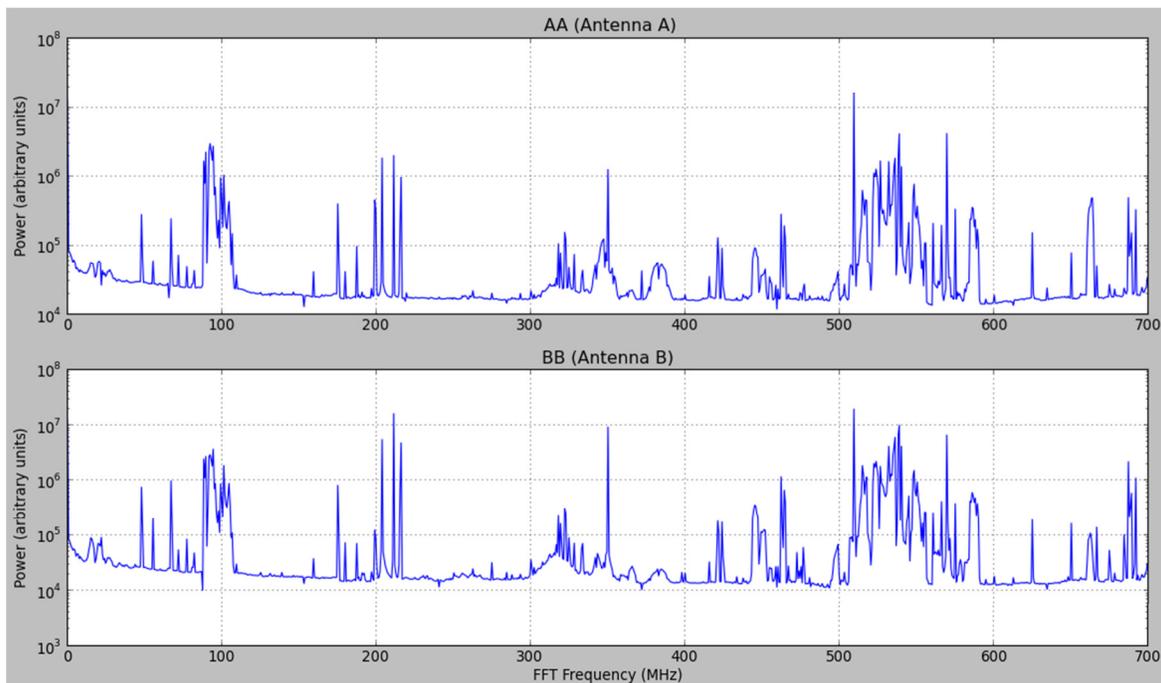


Figura 61: Auto-correlación del ruido externo amplificado.

La Figura 61 corresponde al gráfico de la auto-correlación para cada ADC. Dado que se tienen dos entradas prácticamente idénticas, ya que los amplificadores son de características muy similares; se tienen dos espectros que presentan la misma forma y además con *spikes* muy similares para una frecuencia determinada. Se observa que dichos máximos alcanzan magnitudes cercanas entre 10^6 y 10^7 unidades, mientras que otros *peaks* de gran magnitud bordean inferiormente las 10^6 unidades. Inspeccionando con más detalle se puede apreciar que la entrada B del correlador posee un ligero incremento frente a la potencia medida para todos los canales del espectro con respecto a la entrada A. Se debe a que el amplificador de la entrada B entrega una ganancia levemente mayor a de la entrada A.

El nivel alto de ruido en las mediciones se puede producir debido a que la mayoría del equipo está situado sobre una placa metálica de gran tamaño, la cual en conjunto con todo el montaje funciona como una gran antena. Además no existe ningún tipo de cobertura ni aislación sobre los componentes para evitar la recepción de señales externas, dentro de las cuales se tienen las Radios FM (cercanas a los 100 MHz); canales policiales y otros servicios de emergencia (cercanos a los 200 MHz), etc. También pueden aparecer componentes de alta frecuencia en canales de baja frecuencia debido al fenómeno de *aliasing*.

Este ruido debe ser considerado en futuras mediciones, ya que de esta forma se podrán conocer que componentes en las mediciones de la señal generada por el sintetizador son ruido interno o externo; y así se podrán interpretar de mejor manera los datos.

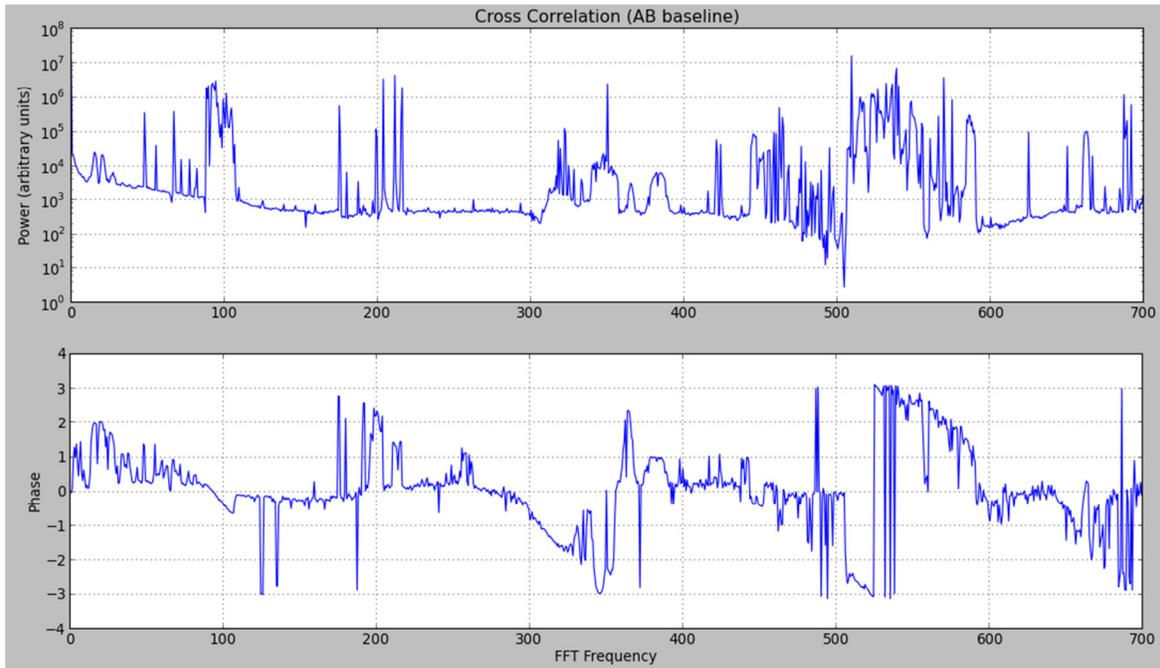


Figura 62: Correlación cruzada y fase del ruido externo amplificado.

En la Figura 62 se puede observar la correlación cruzada entre ambas señales de entrada, generando un espectro muy similar al de éstas, lo cual es de esperarse debido a que el correlador detecta las mismas componentes de frecuencia en el espectro de ambos ADC en todos sus canales, por lo que al cruzar la información, se encontrará con que los ADC están observando el mismo fenómeno.

El gráfico inferior indica la fase (en radianes) que existe en cada canal, el cual se observa muy disperso debido a que las distancias a la que se encuentran las fuentes de ruido deben ser muy variadas, por lo que los desfases de las señales incidentes en cada equipo del montaje funcionando como antena, son diferentes. Además se pueden introducir componentes al desfase considerando que parte de los cables son de largos muy similares, pero no idénticos, con algunos milímetros de diferencia.

La tercera prueba se realizó frente a una señal de entrada sintetizada a una frecuencia de 200 [MHz], cuyos resultados se presentará a continuación:

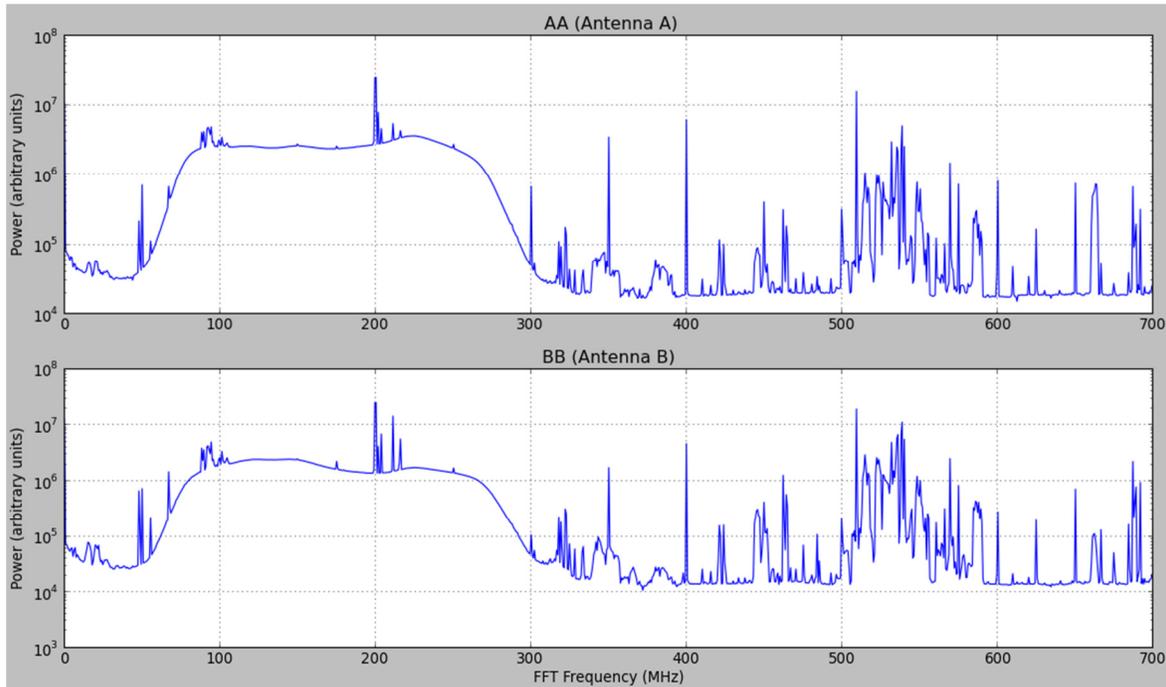


Figura 63: Auto-correlación de una señal de entrada a 200 [MHz].

En la Figura 63 se pueden apreciar diversos fenómenos relacionados con la auto-correlación de las señales de entrada. Primero se puede apreciar el *spike* esperado de la señal marcado en los 200 [MHz] del espectro. Se puede apreciar que junto con ello, apareció un “pedestal” de ruido blanco asociado con el sintetizador de señales (fenómeno esperado según lo discutido en la Sección 3.3.3), el cual cubre un ancho de banda de aproximadamente 260 [MHz], comenzando en una frecuencia cercana a los 50 [MHz] y terminando en una frecuencia cercana a los 310 [MHz], además dicho ruido se encuentra casi completamente por sobre el nivel de ruido base, a excepción de unos máximos que éste tiene por sobre los 200 [MHz] hasta los 225 [MHz] aproximados; además del nuevo *spike* generado en los 100 [MHz].

Se observa que la interferencia base se mantuvo inmutable hasta los 50 [MHz], como era de esperarse, ya que el pedestal de ruido añadido por la nueva señal de entrada interfiere por sobre los 50 [MHz]. El resto del espectro por sobre los 300 [MHz] posee una estructura muy similar correspondiente al ruido base con una diferencia fundamental, la cual se puede observar a simple vista en los canales de 100, 300, 400, 500 y 600 [MHz] del espectro. Dichos *spikes* fueron introducidos por el sintetizador de señales, y la mayoría puede deberse a componentes armónicas de ruido interno del generador de la señal, pero se recomendaría realizar más experimentos para detectar la fuente de dichas interferencias.

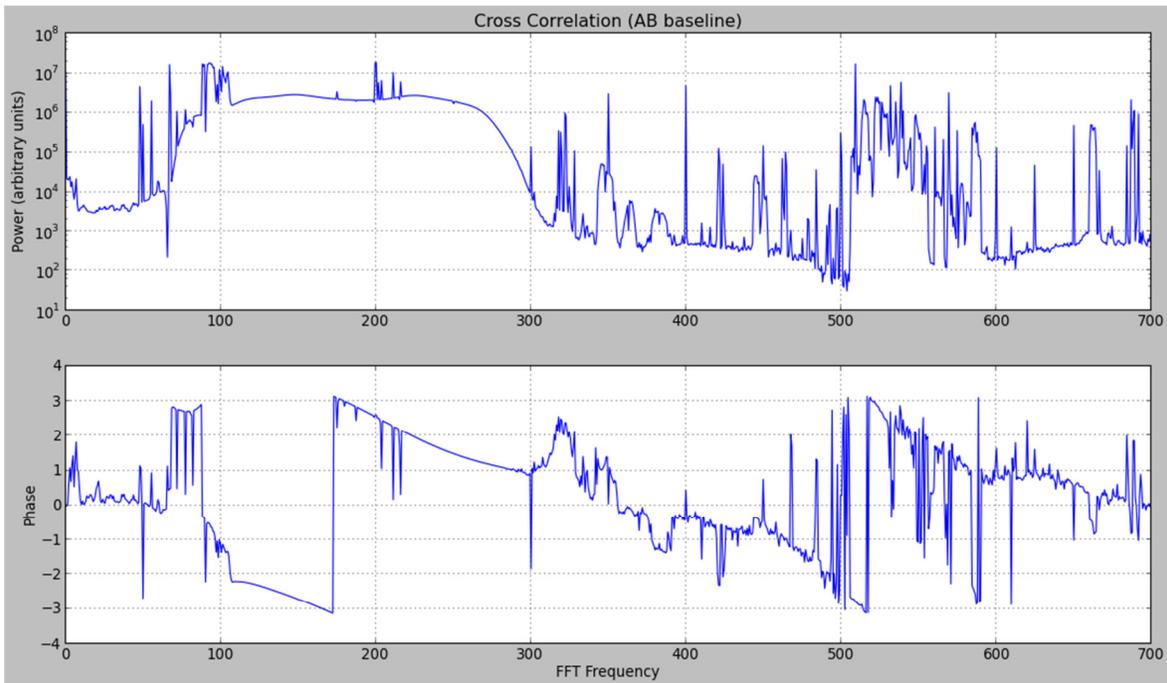


Figura 64: Correlación cruzada y fase de una señal de entrada a 200 [MHz].

El gráfico superior de la Figura 64 indica la correlación cruzada de las señales de entrada, la cual presenta un espectro muy parecido al de las auto-correlaciones de dichas entradas; lo cual está dentro de lo esperado dada las razones expuestas previamente en el análisis del ruido externo. El gráfico que representa la fase (en radianes), muestra una propiedad interesante. Es sabido que el ruido blanco posee un comportamiento de fase “lineal”, por lo que se observa que a medida que aumenta la frecuencia, la fase disminuye hasta llegar a -180° , donde al siguiente punto vuelve a aparecer en los 180° y así recursivamente. Dicha estructura es conocida como “dientes de sierra”, y en el caso del gráfico presentado se puede apreciar una parte de este fenómeno en el ancho de banda donde se presenta el ruido blanco. Las distorsiones que se pueden observar son producidas por la señal generada y los componentes del ruido exterior no blanco. Una vez disipada la componente blanca del ruido, la fase vuelve a tomar una estructura dispersa.

A continuación se expondrán consecutivamente los resultados de los experimentos realizados con señales de frecuencia 300, 400, 500 y 600 [MHz], debido a que el patrón de comportamiento de los datos obtenidos no cambia. Al final de la exposición de gráficos se procederá a detallar los cambios sustanciales en los espectros sin entrar a profundizar en las razones, ya que éstas fueron mencionadas previamente en el análisis de los resultados anteriores. Los resultados se ilustrarán en las siguientes figuras:

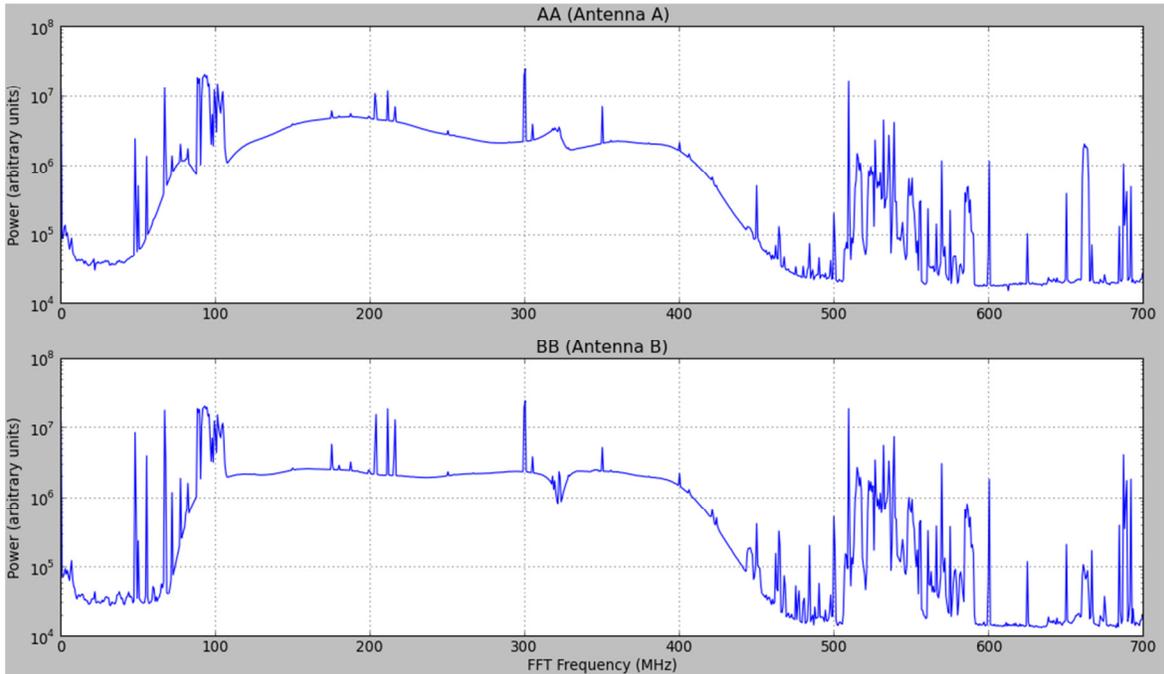


Figura 65: Auto-correlación de una señal de entrada a 300 [MHz].

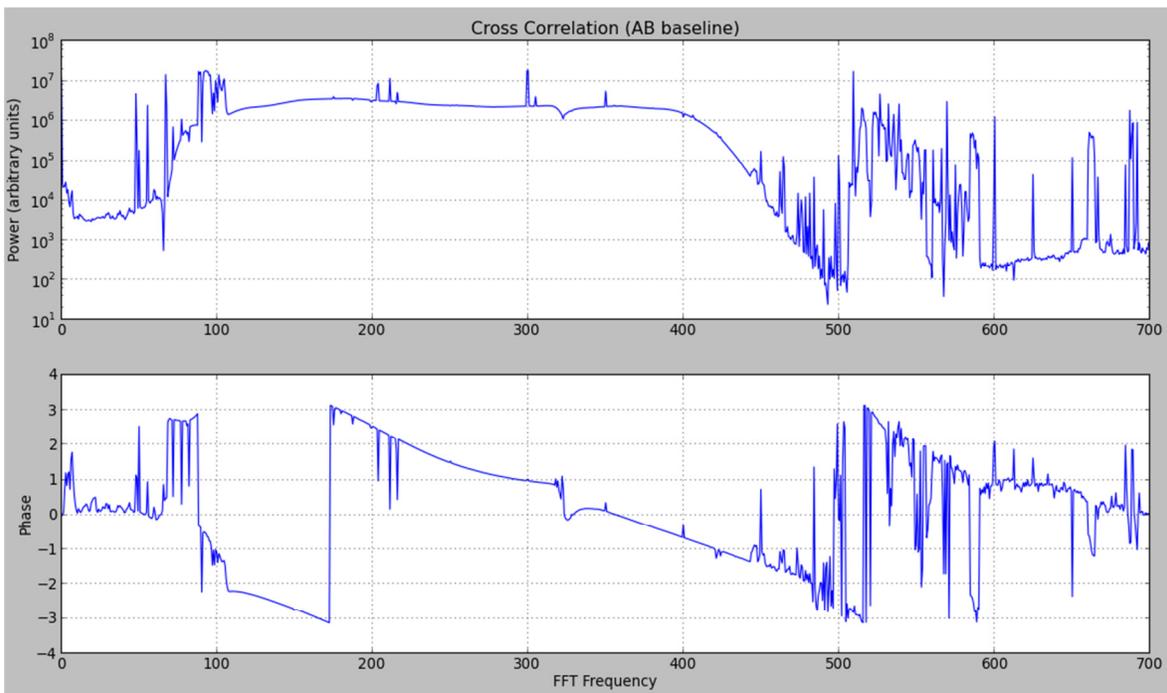


Figura 66: Correlación cruzada y fase de una señal de entrada a 300 [MHz].

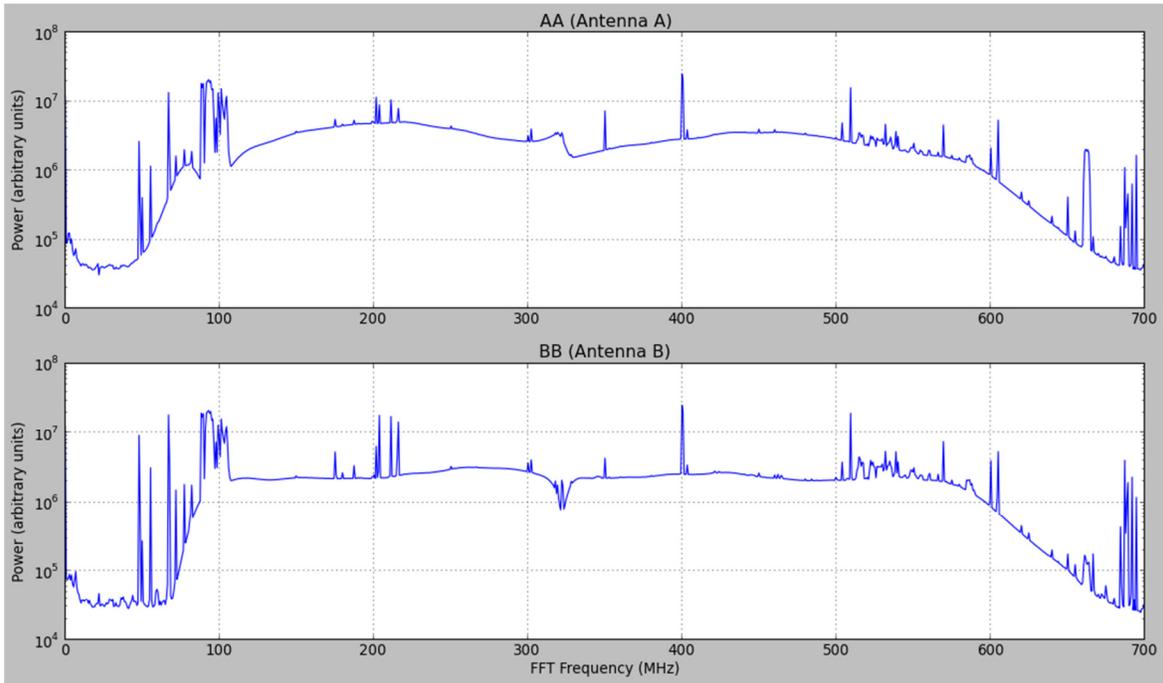


Figura 67: Auto-correlación de una señal de entrada a 400 [MHz].

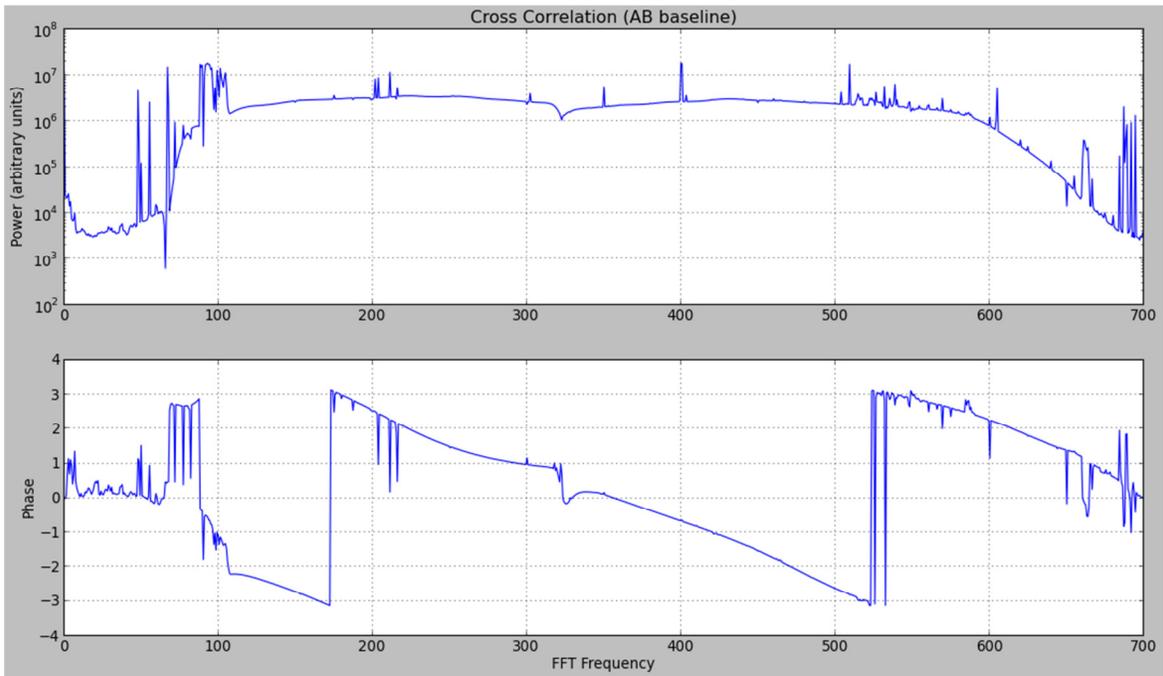


Figura 68: Correlación cruzada y fase de una señal de entrada a 400 [MHz].

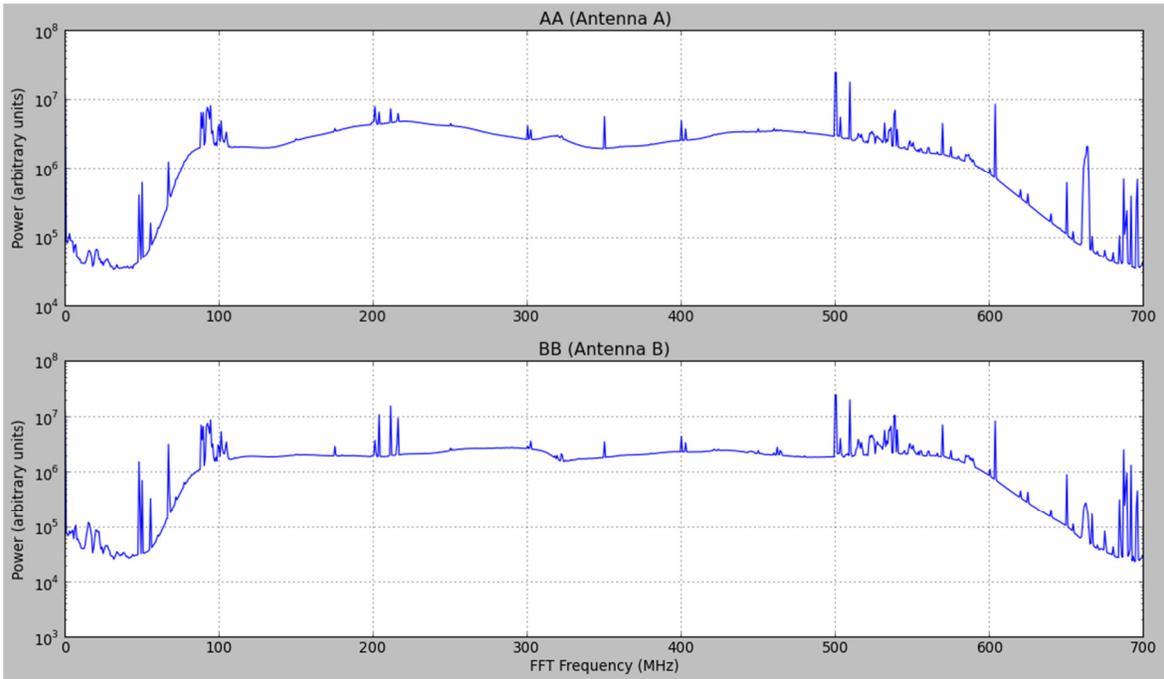


Figura 69: Auto-correlación de una señal de entrada a 500 [MHz].

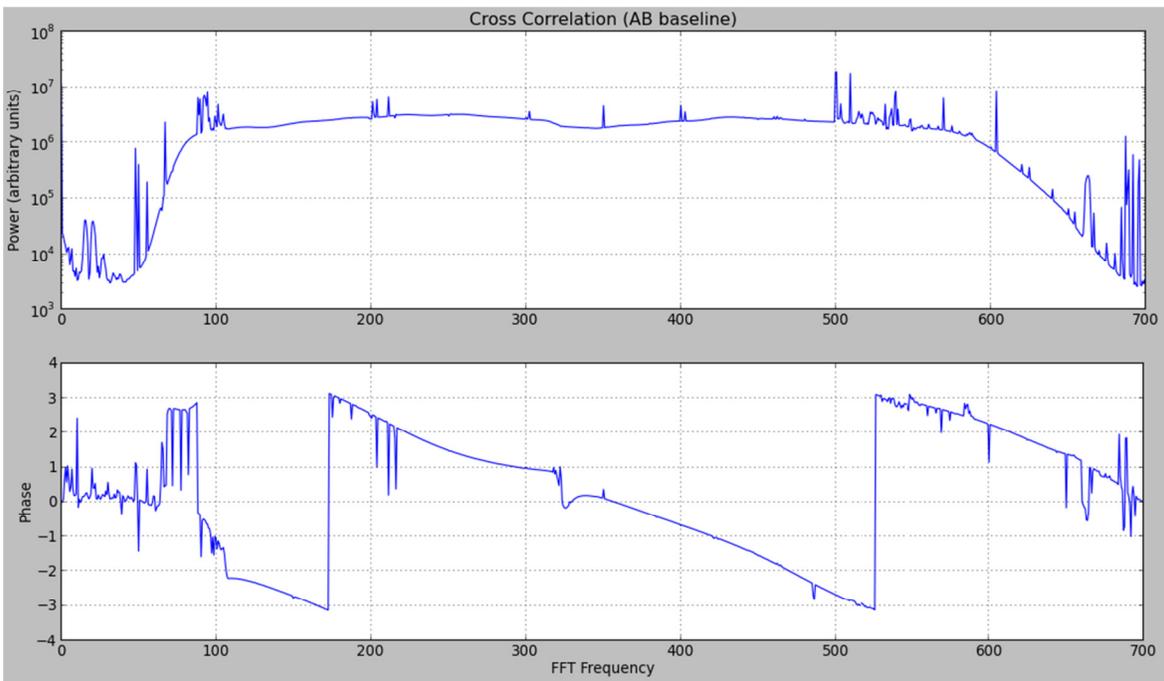


Figura 70: Correlación cruzada y fase de una señal de entrada a 500 [MHz].

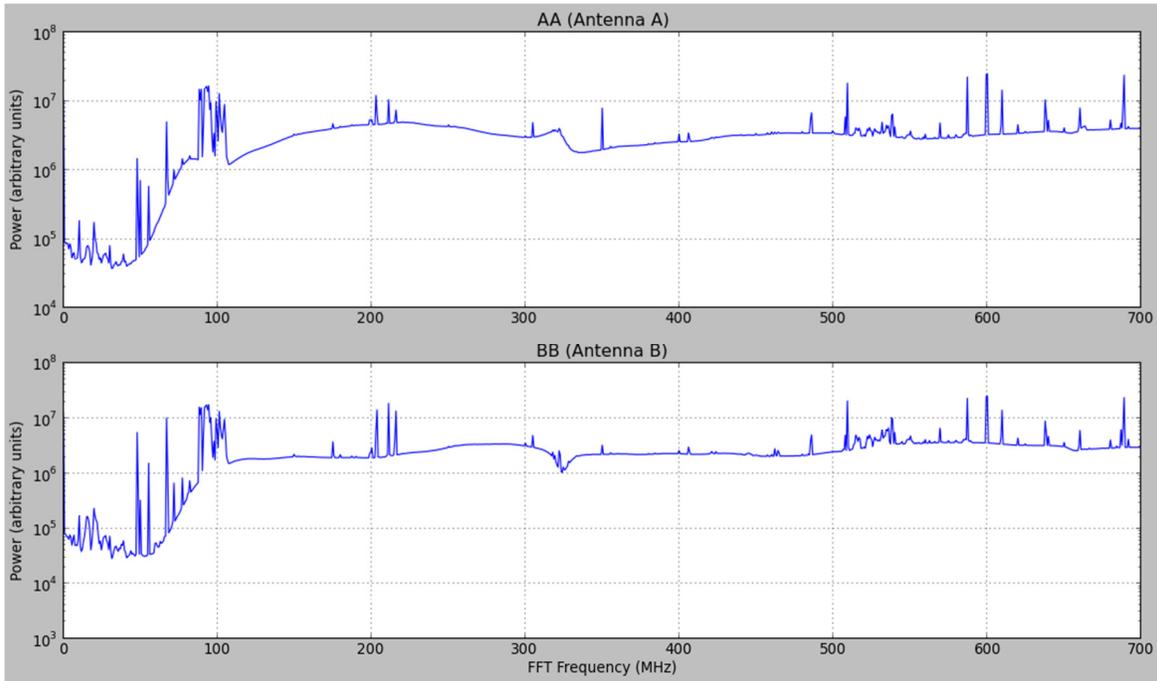


Figura 71: Auto-correlación de una señal de entrada a 600 [MHz].

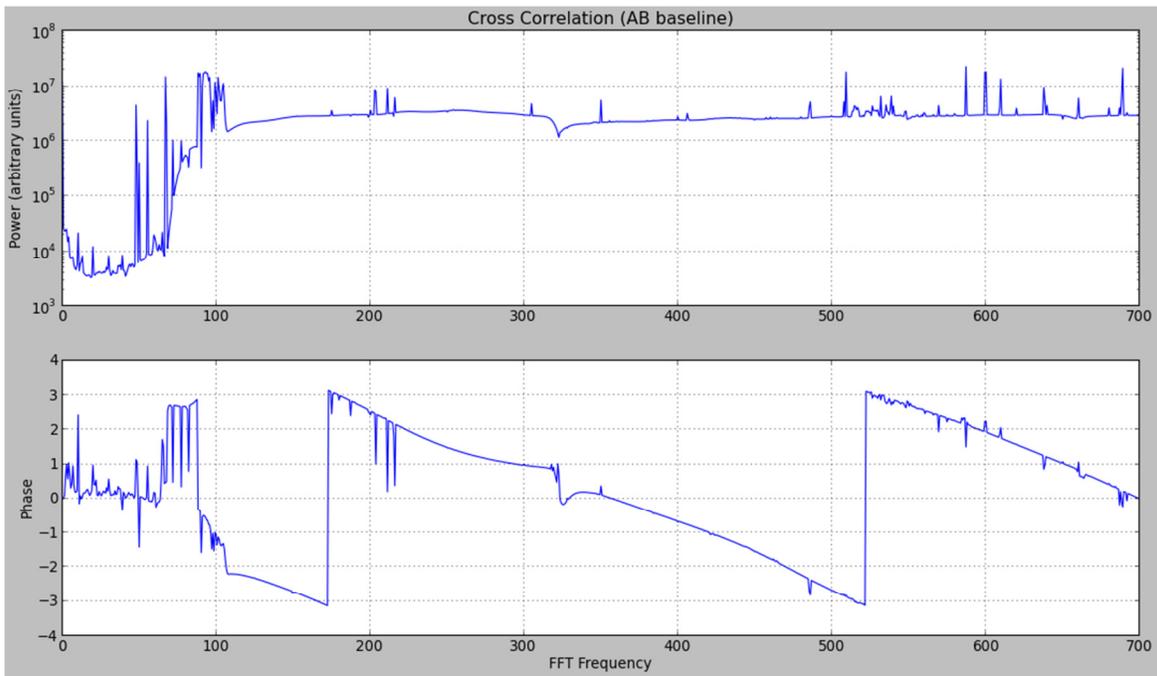


Figura 72: Correlación cruzada y fase de una señal de entrada a 600 [MHz].

Dentro de las características generales de las pruebas finales se pueden rescatar múltiples puntos:

- Tanto en los resultados de auto-correlación y correlación cruzada se pueden apreciar los *spikes* correspondientes a cada una de las frecuencias principales generadas por el sintetizador, es decir, las señales de 300, 400, 500 y 600 [MHz] aparecen con una mayor

amplitud en su potencia (alcanzando la saturación del modelo), por sobre la mayoría de los máximos del ruido externo e interno generados por el ambiente y el sintetizador de frecuencias respectivamente.

- La correlación cruzada sigue manteniendo una estructura muy similar al espectro observado en las auto-correlaciones de las señales en cada frecuencia.
- A medida que aumenta la frecuencia de la señal de prueba, se tiene que el nivel de ruido blanco aumenta junto a ésta, incrementando el ancho de banda en el que está presente. Cabe destacar que dicho efecto no es muy notorio en el cambio de frecuencia de 400 a 500 [MHz], manteniéndose en el mismo rango de frecuencia. La razón de este fenómeno se desconoce y probablemente esté relacionado con el comportamiento del sintetizador de señales.
- En los gráficos de fase, se puede observar que al ir aumentando la frecuencia de la señal, se hace más notorio el efecto “diente de sierra”, quedando la fase más estable y sólo con variaciones debidas a las frecuencias que presentan una alta amplitud en la potencia. Este fenómeno se debe a que el ruido blanco va incrementando su ancho de banda, dominando la mayoría del espectro observado.

4.2. Segundo Montaje

Las pruebas realizadas en el segundo montaje se dividen en dos experimentos principalmente, uno sin control de desfases, y otro con éste control. Se comenzará con los resultados obtenidos en el primer caso, analizando sólo con una señal de ruido blanco ingresando en ambos ADC, los cuales se pueden apreciar en el siguiente gráfico:

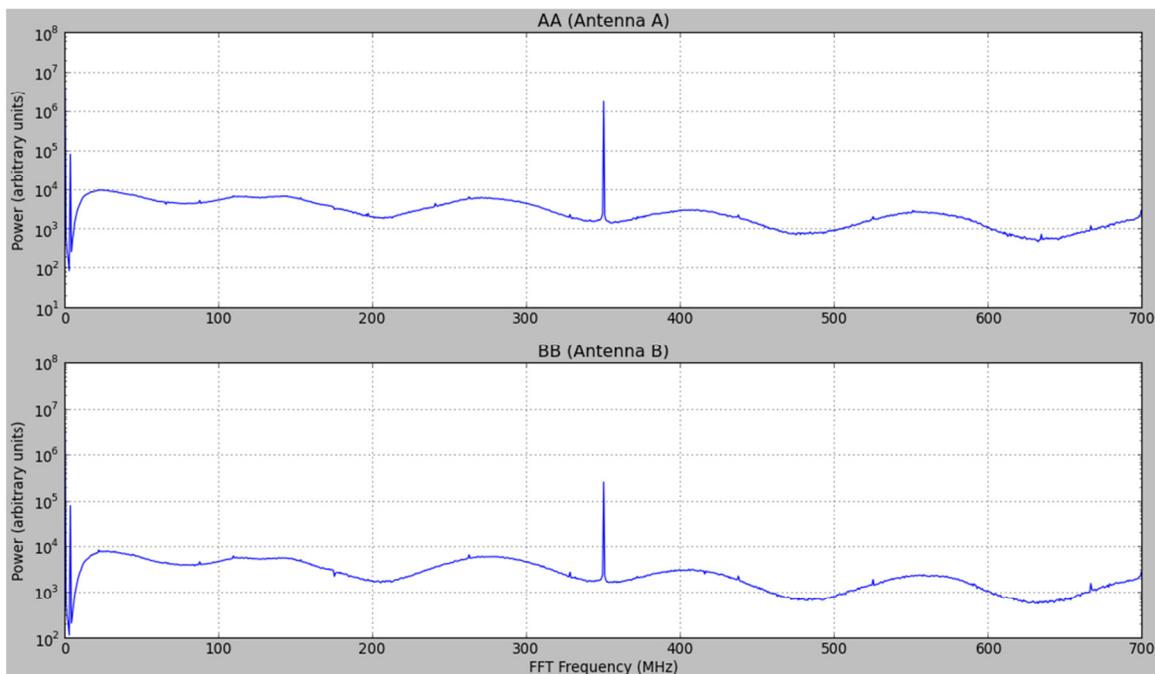


Figura 73: Auto-correlación de ruido blanco en ambos ADC.

En la Figura 73 se puede apreciar la auto-correlación del ruido blanco, el cual aparece en una frecuencia cercana a los 10 [MHz], tal como se indica en las especificaciones del generador de ruido. También se tiene el tono característico del correlador a 350 [MHz], tal como se es esperado. Ahora bien, la correlación cruzada y fase entre las señales de cada ADC se presentan en la siguiente figura:

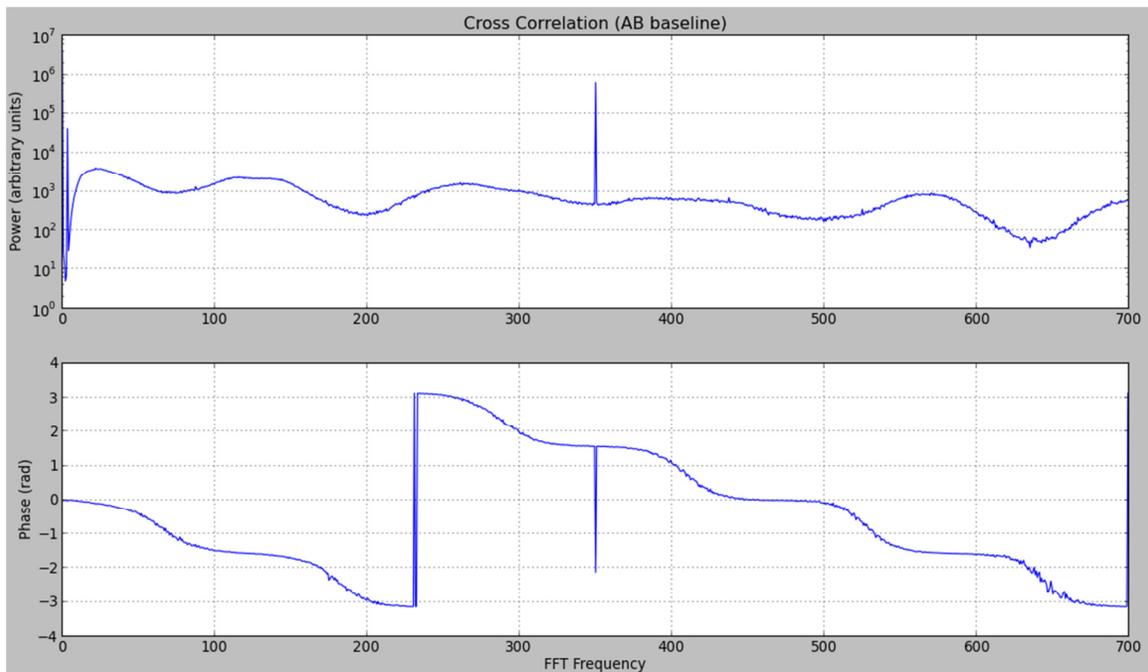


Figura 74: Correlación cruzada y fase de ruido blanco.

Como se aprecia en la ilustración anterior, se puede observar nuevamente el tono de 350 [MHz], y la forma característica de la correlación cruzada de dos entradas prácticamente idénticas; fenómeno expuesto previamente al comienzo de la Sección 3.3. Con respecto al gráfico de la fase entre las señales, se tiene un fenómeno netamente asociado a la presencia de ruido blanco dominando el espectro observado. Nuevamente se tiene un “diente de sierra” como forma principal en la fase, conteniendo un cierto nivel de curvatura; el cual puede estar relacionado con alguna característica interna del sistema asociada a los cables y conectores SMA utilizados en el montaje. La depresión asociada al tono de 350 [MHz] puede ser ignorada, dado que se trata de una señal no válida generándose en el sistema.

A continuación se expondrán los resultados obtenidos del experimento realizado con señales de distinta frecuencia (100, 300 y 500 [MHz]). Al finalizar la exposición, se procederá a entregar un análisis general de los datos presentados.

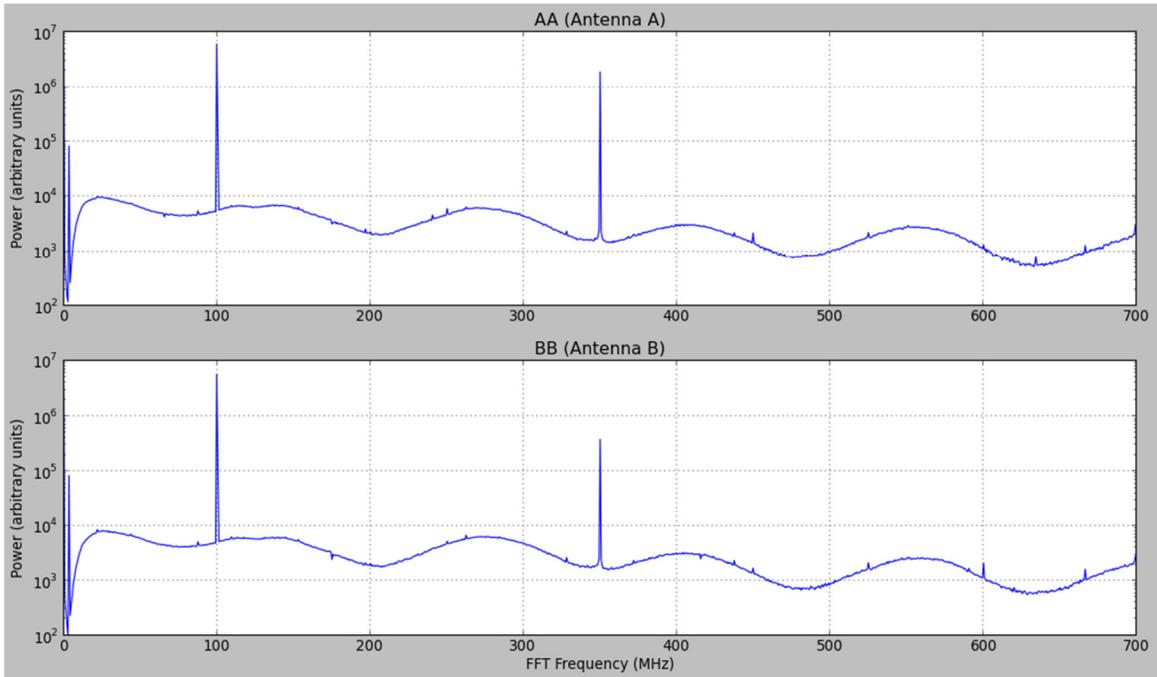


Figura 75: Auto-correlación de una señal de entrada a 100 [MHz].

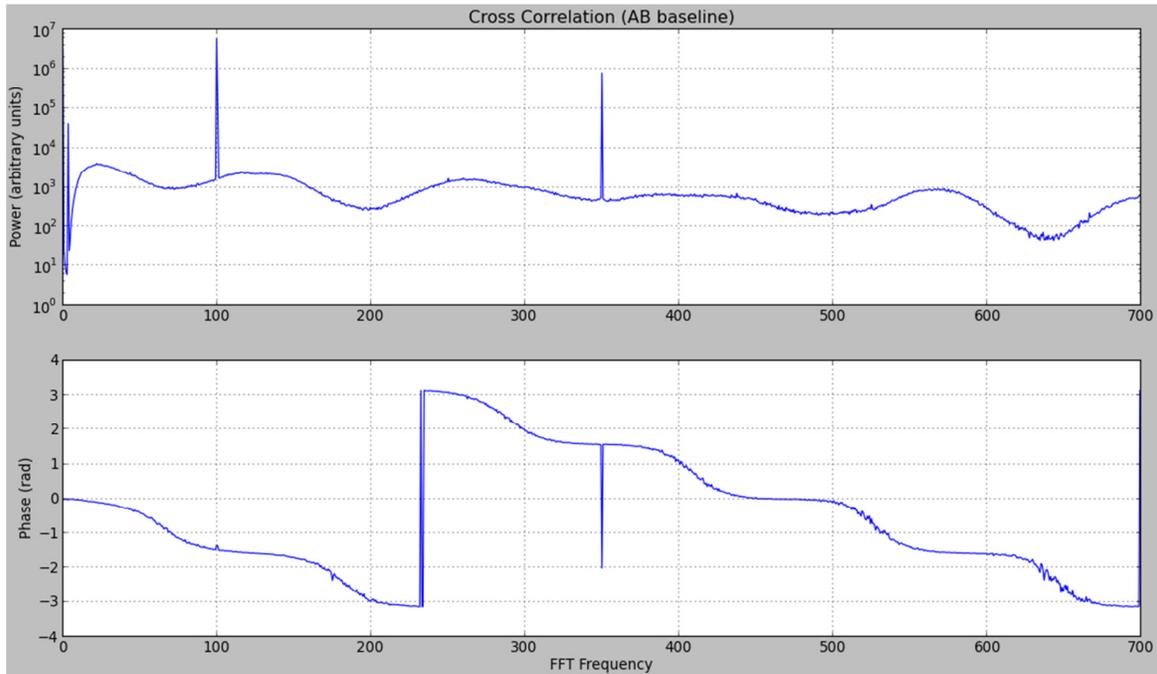


Figura 76: Correlación cruzada y fase de una señal de entrada a 100 [MHz].

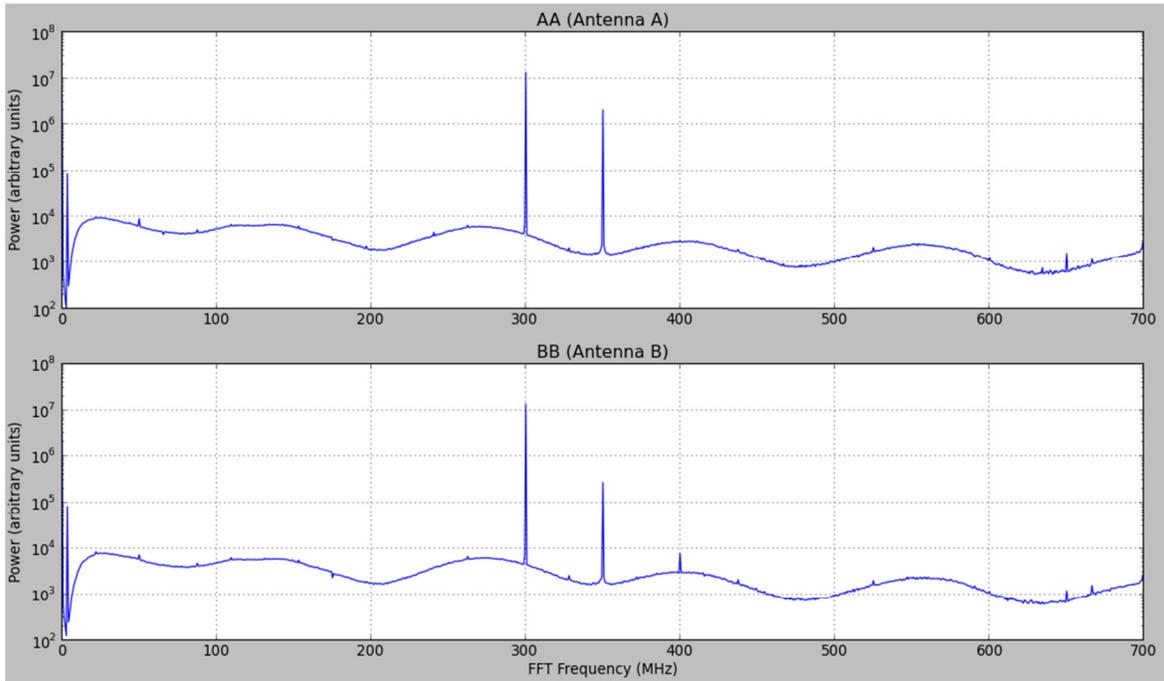


Figura 77: Auto-correlación de una señal de entrada a 300 [MHz].

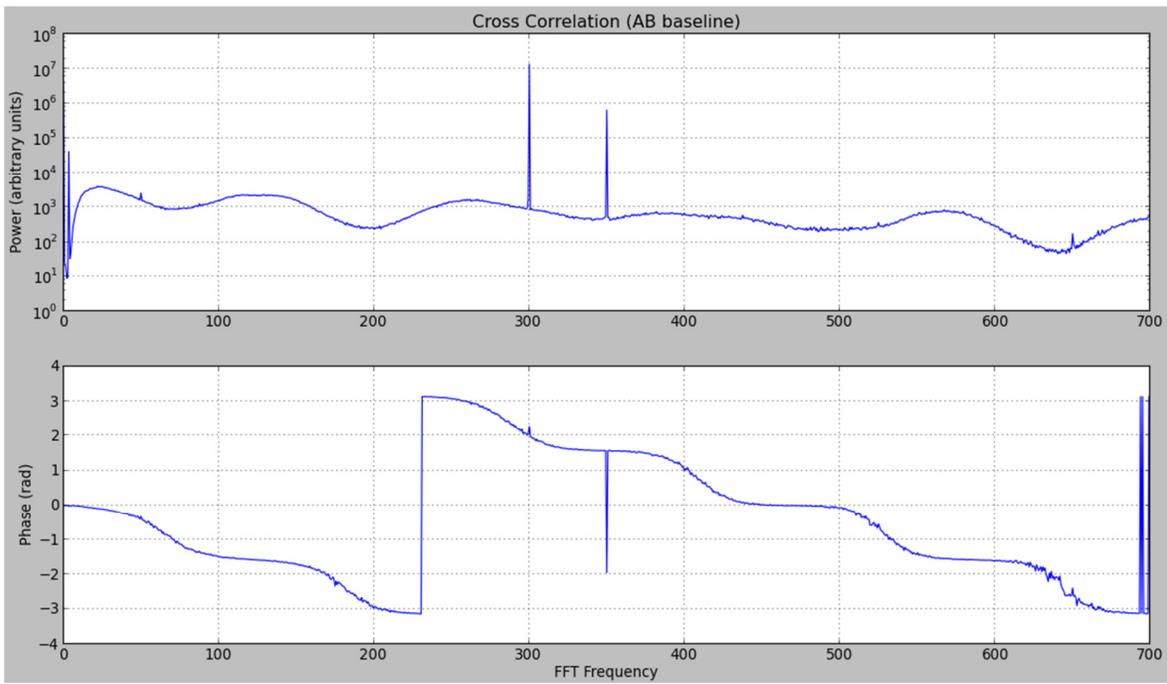


Figura 78: Correlación cruzada y fase de una señal de entrada a 300 [MHz].

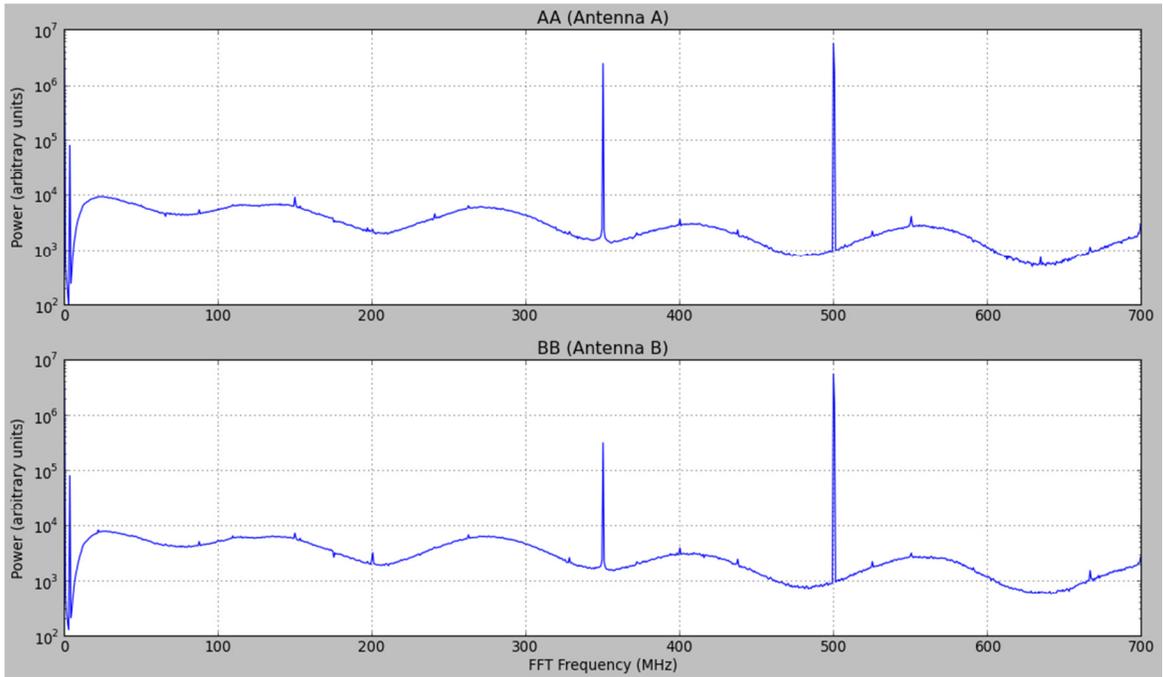


Figura 79: Auto-correlación de una señal de entrada a 500 [MHz].

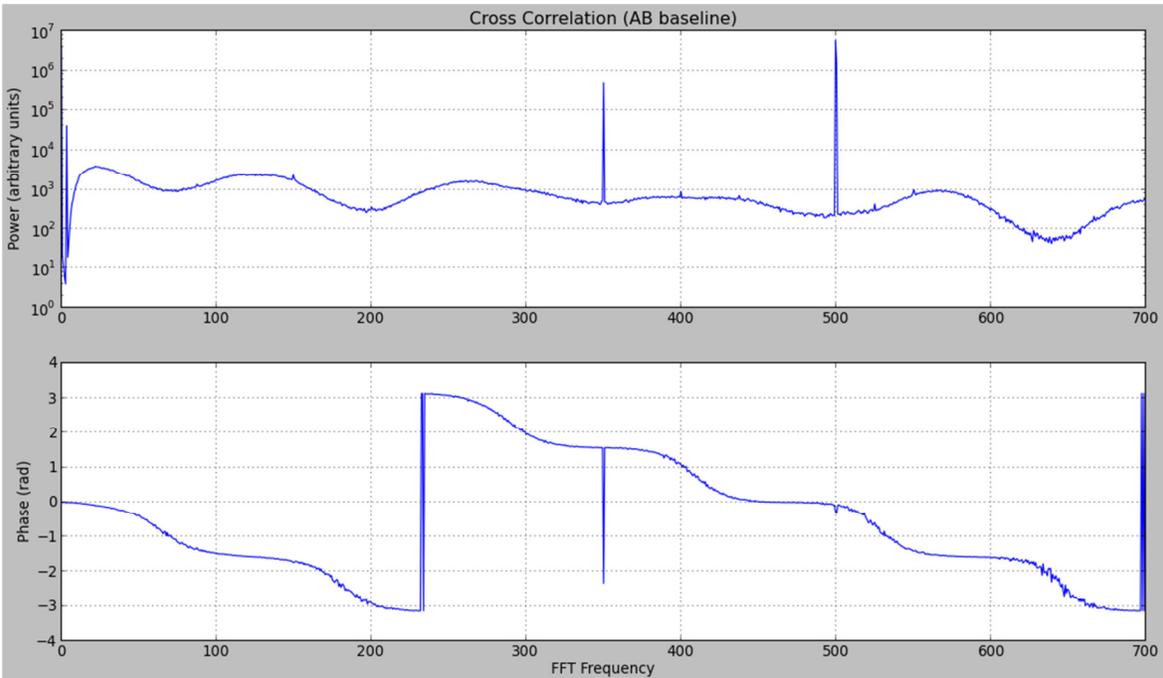


Figura 80: Correlación cruzada y fase de una señal de entrada a 500 [MHz].

En general, se pueden rescatar las siguientes características de los ensayos realizados:

- El comportamiento del correlador sigue demostrando su funcionalidad tal como se presentó en el primer montaje. La diferencia radica en que las señales de entrada no poseen componentes de ruido externo, por lo que se obtuvieron espectros mejor definidos en cuánto a las señales de entrada.
- La fase de cada gráfico posee el mismo comportamiento antes y después de la presencia del tono en la mezcla con el ruido blanco; salvo algunos *spikes* de mínima intensidad relacionados con fenómenos internos de la ROACH.

Como fue planteado anteriormente, en éste montaje se llevará a cabo un segundo experimento que pretende observar el comportamiento del correlador ante una fase controlada de las señales, mediante cables SMA de la misma longitud. De ésta forma se tendría que el desfase de las señales debería ser cero en todo el espectro, ya que cada ADC recibiría la misma señal en el mismo instante de tiempo, dado que el trayecto a recorrer por las señales mezcladas sería el mismo. Lo anterior también se aplica para la señal de reloj que controla la tasa de muestreo de cada ADC. Si existe un desfase en los relojes, se tiene que un ADC construye una señal digitalizada distinta a la del otro, pese a que ambos están recibiendo los mismos datos simultáneamente. Este fenómeno se debe a que el segundo ADC está tomando una muestra desfasada con respecto a la muestra que está tomando el primero, debido a que las señales de reloj no se encuentran perfectamente sincronizadas. A continuación se presentarán los resultados obtenidos con señales de 100, 250 y 400 [MHz] en este experimento; y finalizada la exposición, se presentará el análisis general de éstos.

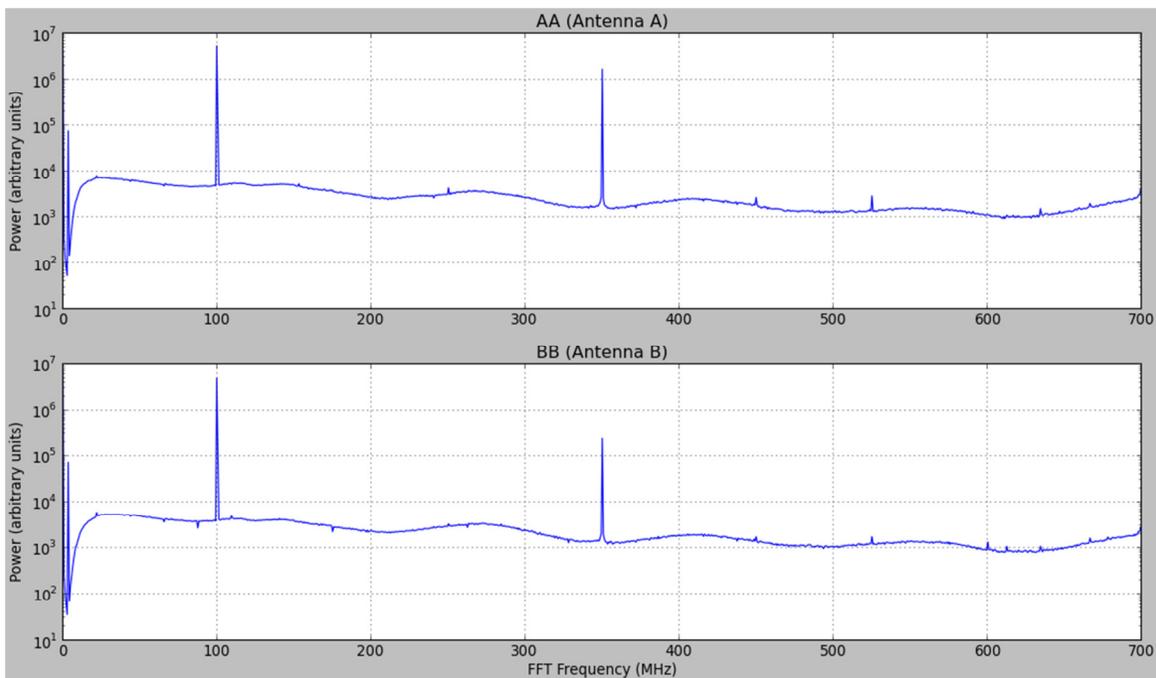


Figura 81: Auto-correlación de una señal de entrada a 100 [MHz].

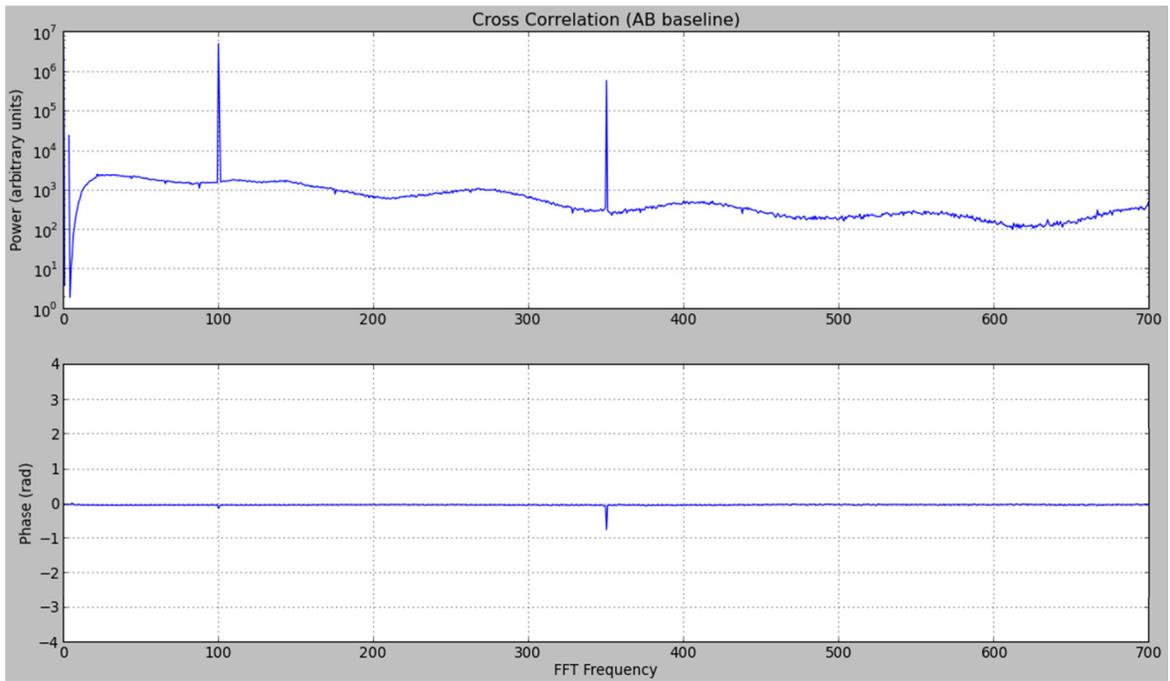


Figura 82: Correlación cruzada y fase de una señal de entrada a 100 [MHz].

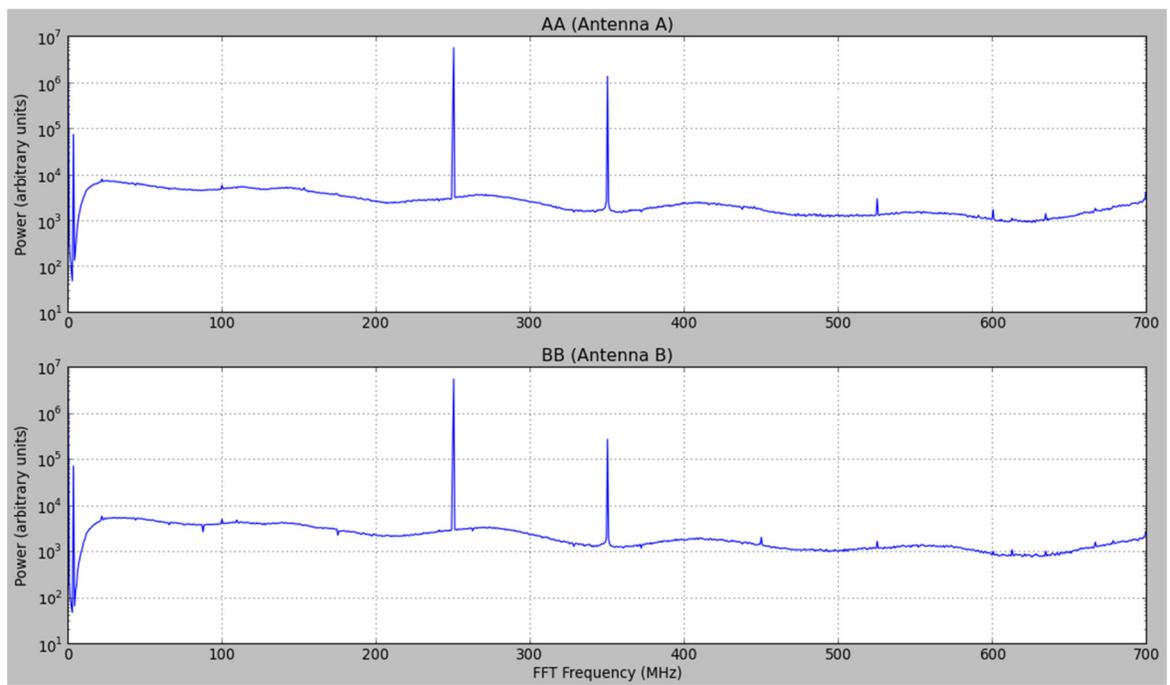


Figura 83: Auto-correlación de una señal de entrada a 200 [MHz].

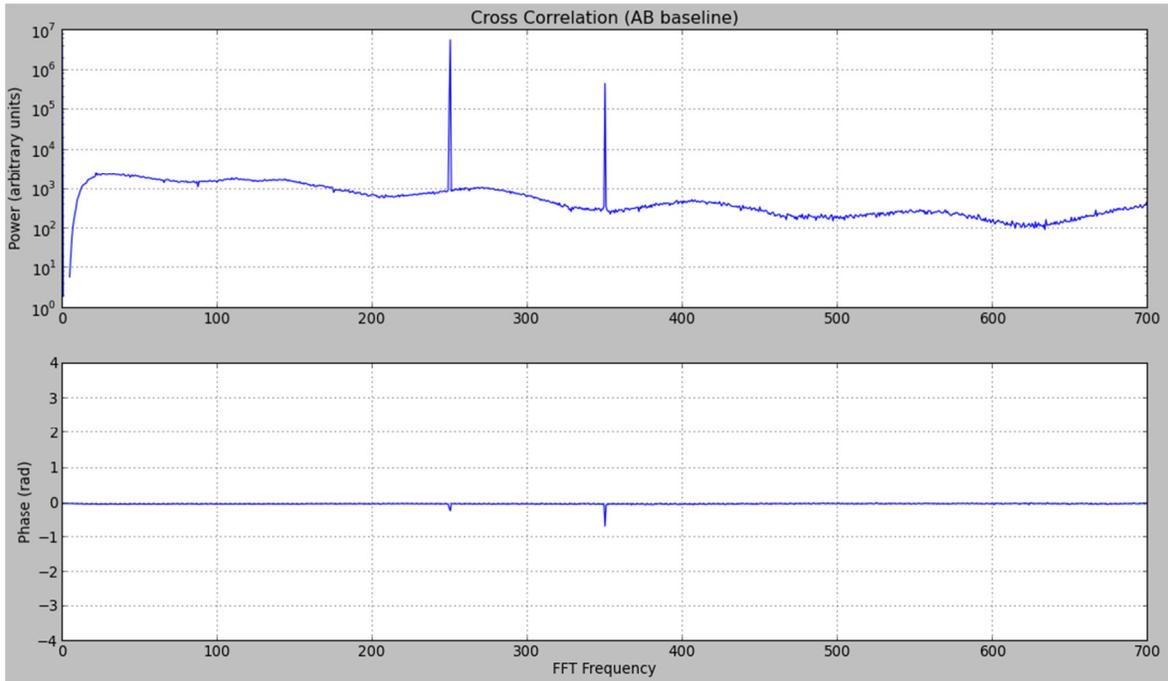


Figura 84: Correlación cruzada y fase de una señal de entrada a 250 [MHz].

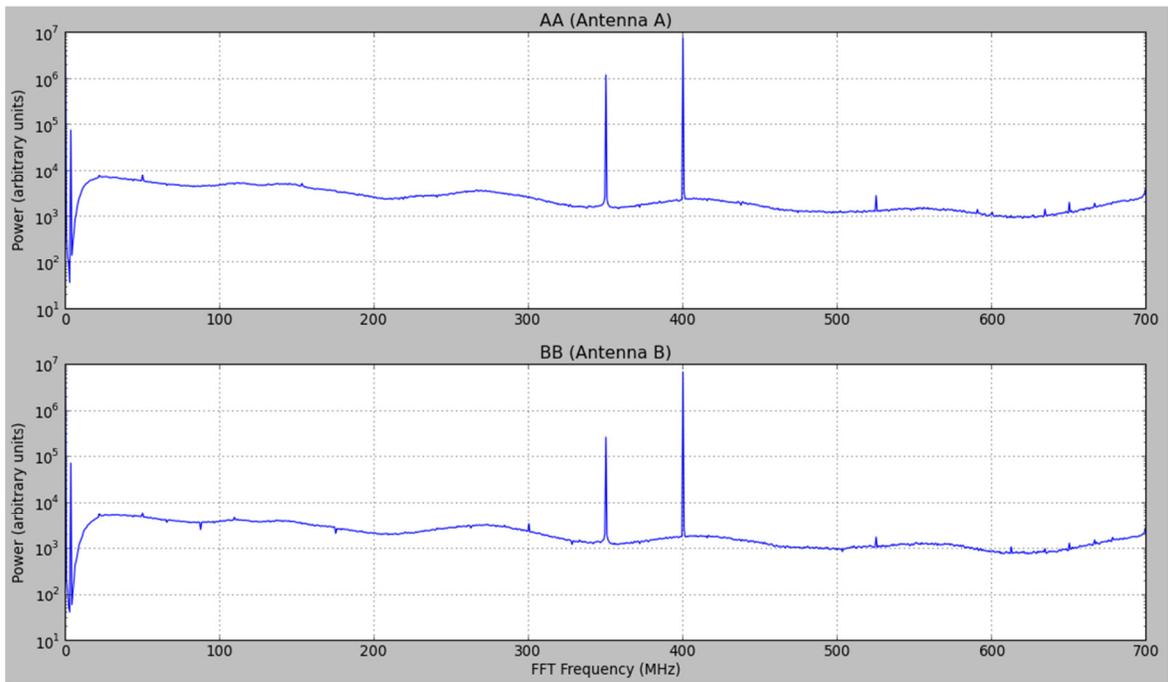


Figura 85: Auto-correlación de una señal de entrada a 400 [MHz].

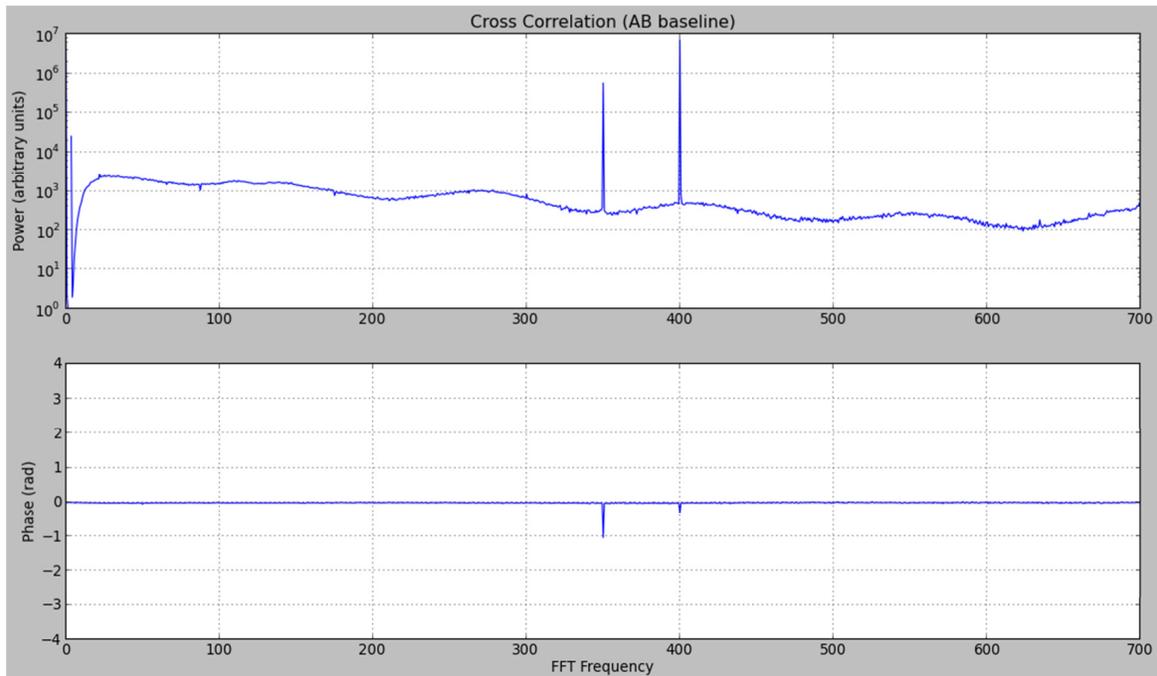


Figura 86: Correlación cruzada y fase de una señal de entrada a 400 [MHz].

Se puede apreciar como características generales de los resultados, los siguientes puntos:

- Los gráficos de auto-correlación se comportan de forma similar a los obtenidos en el primer experimento sin control del desfase. La diferencia yace en que la potencia del ruido blanco se encuentra más estable durante este experimento, lo cual se debe principalmente al cambio de cables SMA y sus conectores, dado que éste fue el único hardware que se varió en ambos experimentos. Posiblemente en el primer experimento, existían pérdidas para determinadas frecuencias del ruido blanco debido a reflexiones internas y atenuaciones de esta señal, al interior del cable y en el contacto de los conectores.
- Los gráficos de correlación cruzada siguen demostrando el mismo comportamiento del correlador frente a los experimentos ya realizados.
- Los gráficos de fase entre las señales de entrada muestran un cambio importante. Anteriormente se tenían gráficos que mostraban principalmente una forma dentada, debido al desfase existente en el sistema. En cambio en este experimento, al controlar el desfase corrigiendo las diferencias de longitud en los cables SMA, se tiene que la fase entre las entradas del correlador es cero. Además de la variación natural observada a la frecuencia central, se tienen pequeñas fases negativas para cada tono de prueba en su frecuencia respectiva; fenómeno que puede ser el resultado de operaciones internas en el cálculo de la correlación y fase por parte de la ROACH.

4.3. Tercer Montaje

Cómo se explicó anteriormente, el tercer montaje es del tipo óptico, en donde la señal de prueba será un haz de luz generado por la mezcla de dos señales luminosas producto de sintetizadores láser. Cabe destacar que para cambiar la frecuencia del haz de prueba, se aumentará la longitud de onda del sintetizador láser de precisión en 5 [pm] por paso, para que de esta forma varíe la frecuencia de la señal resultante de la mezcla entre ambas fuente, como fue explicado en el Capítulo 2. Finalmente, dentro de los resultados no se considerará la correlación de las señales de ruido, ya que al ser un montaje óptico, al no tener ningún tipo de señal captada por los fotodiodos, en la entrada de los ADC se tendrán sólo una señal nula. Los resultados obtenidos fueron los siguientes:

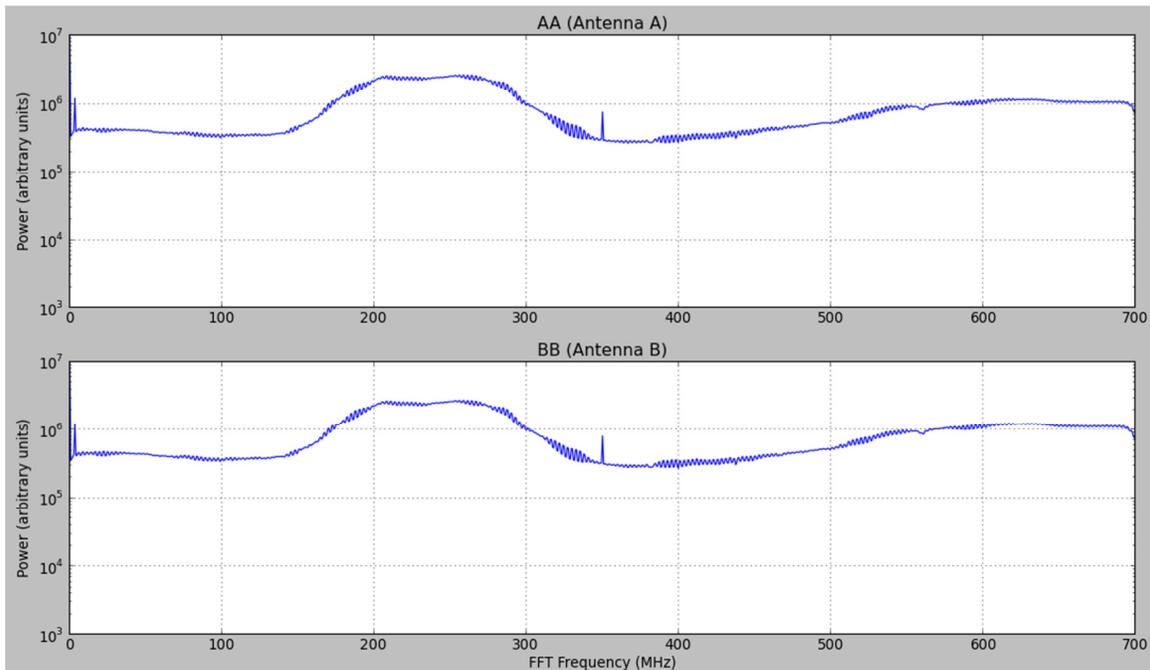


Figura 87: Auto-correlación de una señal mezclada con una diferencia en longitud de onda de 5 [pm].

Se puede apreciar que el correlador en ambas entradas ha recibido la misma señal, en donde el tono se puede apreciar entre los 160 [MHz] y 300 [MHz]. Al contrario de la señal monocromática fija en una sola frecuencia, se tiene una señal con un ancho de banda de 140 [MHz] aproximadamente. Este fenómeno será explicado al final de esta sección, dado que se realizaron mediciones posterior para determinar su causa. También se observa el *spike* en la frecuencia central del espectro, lo cual se encuentra dentro de lo esperado. Además se tiene que el espectro posee múltiples vibraciones en su estructura, las cuales se deben principalmente a que la mezcla de los sintetizadores, no es una señal puramente de una sola frecuencia; sino que existe un rango de varios [kHz] en que los sintetizadores desplazan los haces que están siendo generados. Por lo tanto, la señal observada no es estática como en los casos anteriores, sino que las frecuencias van variando acorde a las fluctuaciones en la precisión de los sintetizadores láser. La siguiente figura expondrá la correlación cruzada entre éstas dos señales:

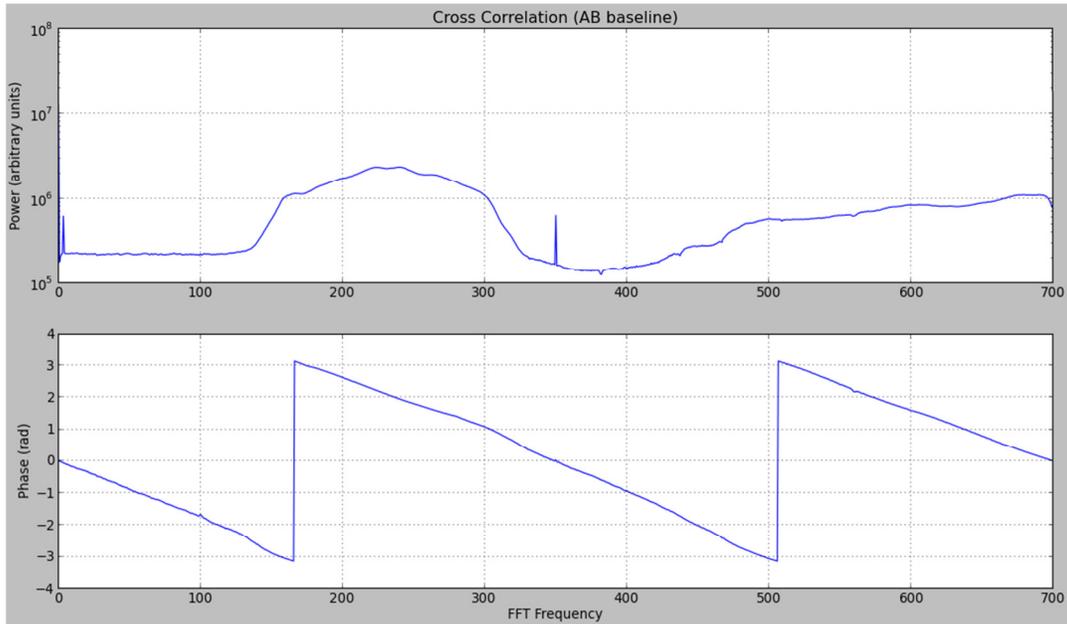


Figura 88: Correlación cruzada de una señal mezclada con una diferencia en longitud de onda de 5 [pm].

En la figura XX se puede observar que la correlación cruzada corresponde a la misma estructura del espectro observado en las auto-correlaciones de cada entrada. Las pequeñas vibraciones presentes anteriormente no aparecen en el gráfico dado que la imagen fue capturada una fracción de segundos después cuando el láser se encontraba en un momento de estabilidad.

En la fase se puede apreciar un comportamiento lineal (fenómeno de “diente de sierra”), similar al del ruido blanco, sólo que en este experimento el ruido proviene de los sintetizadores y el desfase sólo está presente en la diferencia de longitud de los cables SMA, que conectan las señales de reloj entre el sintetizador interno de la ROACH con cada ADC. Al aumentar en 5 [pm] más la longitud de onda del sintetizador de precisión, se tiene la siguiente auto-correlación de cada entrada:

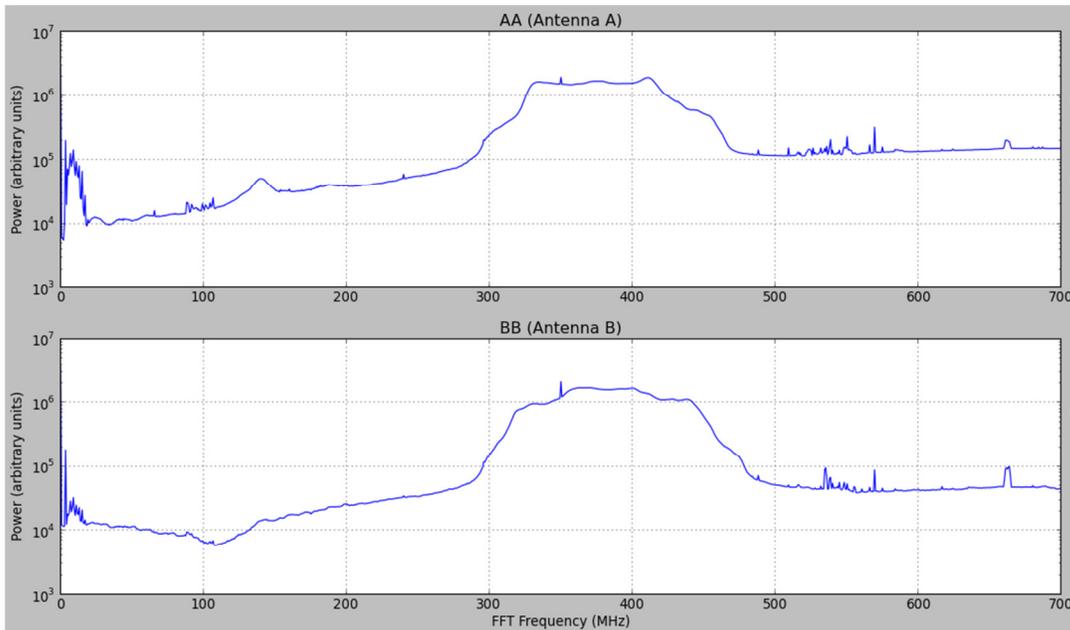


Figura 89: Auto-correlación de una señal mezclada con una diferencia en longitud de onda de 10 [pm].

En la ilustración anterior se pueden apreciar las auto-correlaciones de cada ADC, en dónde se comienzan a observar fluctuaciones en la estructura de la señal, con respecto al resultado obtenido anteriormente. En el espectro comienzan a aparecer un mayor número de *spikes* a lo largo del ancho de banda del correlador, relacionados con la inestabilidad de los sintetizadores, además de tener un aumento de temperatura en éstos debido al periodo de operación. Se vuelve a enfatizar que los láser no se encuentran enclavados en una frecuencia fija; por lo que naturalmente el espectro oscila, de acuerdo a la diferencia de frecuencia que se vaya generando en el haz resultante, tras el proceso de mezclado en los acopladores de fibra óptica. Finalmente, se puede apreciar el tono en la banda correspondiente entre los 320 [MHz] y los 460 [MHz] aproximadamente en cada ADC, pero en éste caso, las estructuras de dichos tonos presenta algunas diferencias. Éstas se deben probablemente a algún proceso relacionado con reflexiones internas en las fibras ópticas o en los acopladores, las cuales pueden producir diferencias operacionales en los fotodiodos. El espectro cruzado de ambas señales se puede apreciar en la siguiente figura:

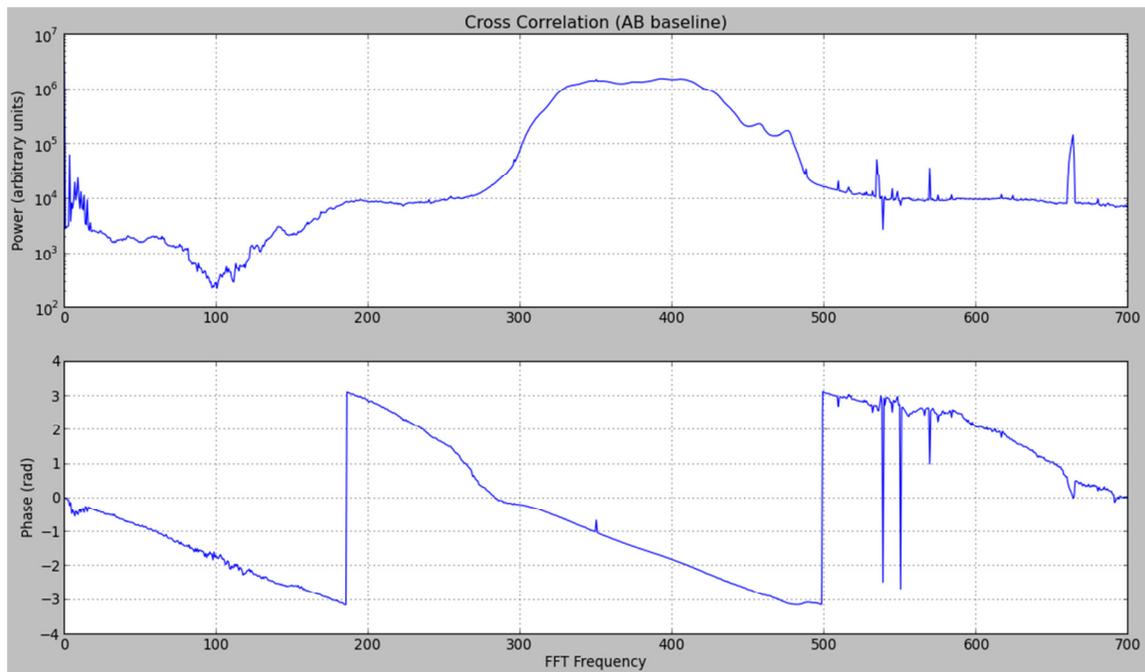


Figura 90: Correlación cruzada de una señal mezclada con una diferencia en longitud de onda de 10 [pm].

En el gráfico de la correlación cruzada de la Figura 90 se tiene el tono generado entre las frecuencias de 330 [MHz] y 470 [MHz] aproximadamente, sólo que la estructura es distinta a la de auto-correlación, debido a que ambas señales no son exactamente idénticas. En el gráfico de la fase se puede apreciar una estructura similar al de la Figura 88, sólo que en éste existe una mayor componente relacionada con la interferencia, producto de fenómenos internos en el montaje del experimento relacionado, como ya fue mencionado previamente, con reflexiones internas del láser tanto en las fibras, como en los acopladores y en los fotodiodos.

Cabe destacar que no se realizaron más pruebas ópticas debido a la inestabilidad de las señales, donde al aumentar nuevamente la longitud de onda en 5 [pm], el tono sale del rango observable para el correlador, generando una señal con *aliasing* entre los 600 y los 700 [MHz].

Una de las posibles causas por la que se tiene un tono con un ancho de 140 [MHz], es que en las fibras ópticas se debe estar generando una cavidad interna, la cual actúa frente al láser como una cavidad externa, generando un espacio para que los rayos del haz láser se reflejen internamente, e interactúen de forma aleatoria con el haz principal de cada sintetizador. El problema principal con la cavidad interna en las fibras ópticas, es que no hay control sobre éstas y dependen de variables ambientales, tales que el resultado del fenómeno de interferencia entre los rayos viajeros y reflejados, sea en gran parte aleatorio. Las componentes armónicas del proceso de reflexión interna se pueden observar mediante el uso del analizador de espectro, ya que con ésta herramienta se puede tener una mejor resolución espectral, a diferencia del correlador. El resultado obtenido a la salida de la etapa de amplificación (previa a la entrada a los ADC) se presenta en la siguiente figura:

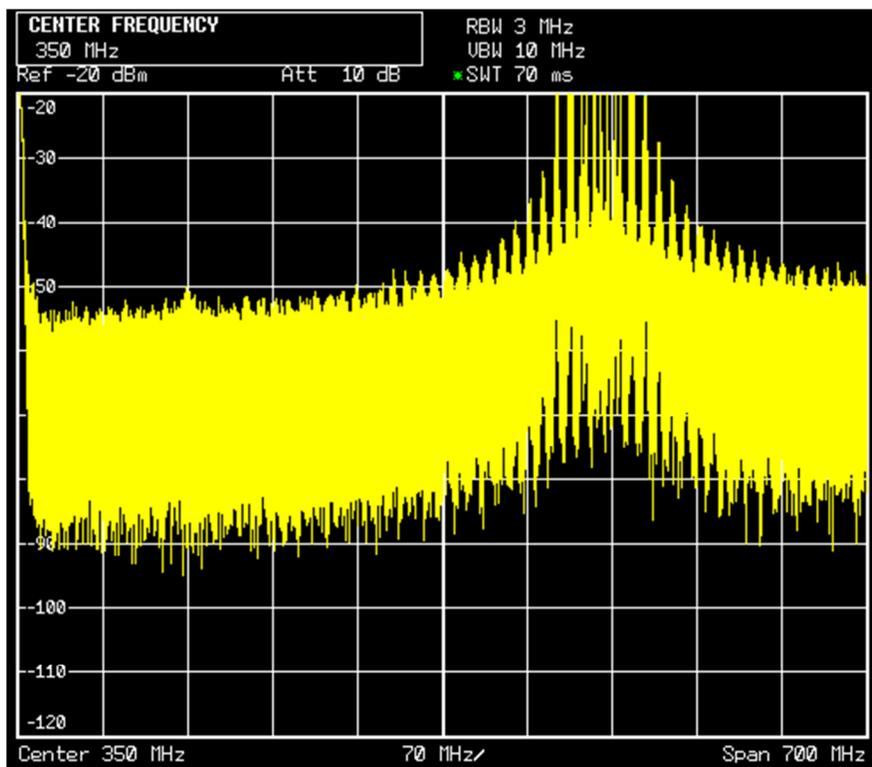


Figura 91: Espectro del tono generado por el sistema a la salida de los amplificadores.

Se pueden observar las distintas componentes armónicas entregadas por el tono, las cuales se pueden producir tanto por la aparición de cavidades internas en el sistema, como por el tipo de tono que generan los sintetizadores láser. Para comprender mejor este fenómeno, e ir descartando posibles elementos que lo generen, se recomienda realizar el mismo experimento especificado para el tercer montaje, sólo que las conexiones entre los sintetizadores y las fibras, debe hacerse mediante aisladores. De esta forma se aíslan las cavidades externas al láser, y por lo tanto se tendrían tonos de mayor resolución.

5. Conclusión

En este capítulo se presentarán las conclusiones referentes a los resultados obtenidos y al desarrollo en general de la memoria, además se expondrán posibles soluciones para mejorar los resultados obtenidos durante el proceso de experimentación y finalmente se darán a conocer posibles trabajos propuestos y experimentos a realizar con el hardware y software implementados en el laboratorio.

5.1. Conclusiones

En cuanto al marco de la memoria cabe destacar que el laboratorio ahora cuenta con el software necesario para el uso de la ROACH. Su potente FPGA permitirá con basta flexibilidad el desarrollo de diversos experimentos en el ámbito de la radioastronomía y el procesamiento de señales. Este documento proporciona la información necesaria para la reproducción de los modelos implementados, además de servir como guía para futuras reinstalaciones del software encargado de controlar dichos modelos. También el marco teórico presenta la información necesaria para poder entender a grandes rasgos lo que es la radioastronomía y cuál es su herramienta principal para el estudio del Universo: la interferometría.

Se puede afirmar que el objetivo principal de la memoria fue cumplido y se pudo implementar un modelo de correlador en la placa ROACH, además de definir paso a paso el proceso de diseño y de la instalación del software necesario para su control, el cual es uno de los objetivos secundarios perseguidos

Se consiguió con éxito la instalación del software de diseño implementado por CASPER para el desarrollo de nuevo hardware. Al cumplir dicho objetivo secundario de esta memoria, permitirá la implementación rápida y flexible de nuevos sistemas usando MATLAB-Simulink® como herramienta principal; prescindiendo de ésta manera entrar a programar una FPGA directamente usando lenguajes como Verilog o VHDL.

Sobre la implementación de los modelos cabe decir que el proceso de compilación de los mismos suele ser problemático, y se entregan las referencias necesarias del sitio de desarrollo de la ROACH para encontrar posibles soluciones.

Aun cuando hubo un modelo que no pudo ser probado por la falta de hardware necesario para su funcionamiento, el software de control quedó correctamente instalado, y la guía presentada para dicho procedimiento detalla cuidadosamente todo lo necesario para llevar a cabo nuevamente la instalación de forma exitosa.

Los resultados obtenidos se alejaron del propuesto al comienzo en cuanto a las condiciones de la señal generada, pero esto se debió principalmente a que la interferencia presente en el ambiente del experimento, y las malas condiciones de algunas de las piezas del montaje, generaron ruido externo y ruido blanco en magnitudes no despreciables, las cuales influyeron fuertemente en la forma del espectro que se observaba; descartando de esta manera la posibilidad de producir un espectro limpio con sólo un *spike* en la frecuencia del tono de entrada. Pero aún con la presencia de ruido en las mediciones, se pudo probar con éxito el modelo desarrollado, ya que los resultados arrojados eran acordes a lo esperado de un correlador en presencia de cualquier señal de entrada. Además la presencia de ruido blanco generó la oportunidad de probar el comportamiento del correlador en la detección de la fase entre las señales de entrada. Con el segundo montaje se pudo corroborar el correcto funcionamiento del correlador, ya que al encontrarse en un ambiente sin fuentes externas de ruido, fue posible observar espectros con mayor claridad, y libres de interferencias. Además se probó la respuesta

ante el control de la fase del correlador, la cual entregó resultados tales como los esperados. Finalmente en el tercer montaje, nuevamente el correlador operó según lo estimado, pero surgieron fenómenos asociados al montaje, los cuales pueden ser reparables si se cuenta con la implementación descrita al final de la Sección 4.3.

En resumen, el objetivo principal de la memoria y sus objetivos secundarios fueron llevados a cabo.

5.2. Soluciones a Implementar

Para mejorar los resultados del experimento se plantearán algunas soluciones de acuerdo a lo observado.

Una solución al ruido externo consiste principalmente en aislar las piezas de hardware que participan en los experimentos. Se podría construir un blindaje que contuviera la etapa de amplificación y conectores metálicos en su interior, para que ninguna de estas piezas pueda actuar como antena e introducir señales externas en forma de ruido al experimento.

Se plantea tener cableado de mejor calidad en cuanto a cubierta aislante e instalación de sus conectores, dado que pequeñas filtraciones en la señal de entrada de los ADC por parte del ruido externo, puede deformar considerablemente el espectro esperado; especialmente si el ruido ingresó a la señal de prueba previamente a la etapa de amplificación. Además se debe procurar que los cables usados para generar las ramas de cada ADC sean del mismo largo, lo cual puede influir en valores erróneos al momento de detectar la fase.

También se recomienda usar otro sintetizador de señales para las pruebas del correlador, dado que el disponible posee un nivel de ruido blanco muy elevado, el cual influye determinantemente en la forma del espectro como se puede apreciar en los resultados obtenidos. Un candidato a sintetizador de señal de prueba puede ser una de las salidas que posee el sintetizador del reloj incluido en la ROACH. Dicho componente cuenta con 2 salidas como es sabido; una de ellas puede ajustarse a la frecuencia necesaria para operar los ADC (dicha señal pasará por medio de un divisor o *splitter* y de esta forma se tendrán ambos ADC conectados al reloj), y la otra salida puede ser utilizada como señal de entrada de los ADC (también pasando por un *splitter*). Se sabe que el espectro de aquellos sintetizadores es bastante aceptable y proporcionarían un nivel muy bajo de ruido, pero aun así, habría que observar cuidadosamente el efecto de las componentes armónicas generadas por dicho sintetizador.

5.3. Trabajos Propuestos

Dentro de los trabajos propuestos para desarrollar con la ROACH y el software disponible son los siguientes:

Se puede utilizar el correlador para realizar los experimentos ópticos presentados en la Sección 2.1.6 del Capítulo 2. Dicha actividad requiere que se arme el montaje descrito y para ello se deben tener todos los componentes de hardware necesarios en buen estado y con sus especificaciones definidas mediante mediciones en el laboratorio. Luego se puede proceder a capturar señales cercanas al infrarrojo, y mezclarlas para luego, ser analizadas utilizando el correlador.

En los resultados del segundo experimento descrito en la Sección 4.2, se pudo observar que, aun teniendo una fase nula para casi todos los canales de frecuencia, existe un desfase menor en el canal de la frecuencia perteneciente al tono generado por el sintetizador. En caso de ser necesaria su eliminación del sistema, se recomienda realizar un estudio de las posibles causas que pueden provocarlo, y corregir el montaje del experimento en caso de que allí se encuentre el origen de dicho problema.

En el montaje planteado para el experimento realizado con el correlador de 700 [MHz] no contenía cambios de fase planeados para el estudio del desfase entre las señales de entrada y la respuesta del correlador ante estas. Se sugiere realizar el experimento con cables de largos estándar para generar desfases de 45°, 90° y 180° para estudiar el comportamiento del correlador ante cambios de fases controlados de pendientes definidas.

Un punto importante del correlador, es que debería ser capaz de ajustar los retrasos geométricos de forma automática sin la necesidad de cambiar las longitudes de las fibras de ondas de cada antena, para realizar el ajuste requerido por el desfase espacial producido por el cambio de longitud de la *baseline* o simplemente por cambiar el ángulo de observación de ambos telescopios. Este punto es fundamental para desarrollar la interferometría heterodina, por lo que se recomienda fuertemente añadir al modelo ya probado en este trabajo, un control de desfases geométricos. El sistema puede ser desarrollado en MATLAB-Simulink®, y si es necesario ajustar parámetros de dicho modelo se debería tener un script para redefinirlos sin que fuese necesario compilar el modelo nuevamente; ya que dicho proceso puede tomar varias horas en estar completado. Dentro de esta posibilidad, también se debe tener en cuenta si los recursos disponibles en la FPGA son capaces de albergar este nuevo sistema, dado que el correlador es un modelo que consume gran parte de dichos recursos. En el caso de que sea estrictamente necesario, y no se dispongan de los recursos suficientes en la ROACH, se recomienda la adquisición de un modelo más reciente de este hardware, ya que éstos contienen una versión más actualizada de la FPGA incluida en la placa con la que se trabaja en el laboratorio.

Para finalizar, esta memoria expuso el modelo del espectrómetro VEGAS como un correlador que ofrece mayores ventajas en cuanto al ancho de banda y a la resolución espectral para realizar observaciones. Dado que parte del trabajo consistió en dejar el sistema apto para el correcto funcionamiento de VEGAS, instalando el software de control para su operación; se recomienda realizar las pruebas necesarias de dicho diseño cuando el hardware que requiere se encuentre disponible. De esta forma se tendrá una herramienta con mayores capacidades para el análisis espectral requeridos, considerando 2 o más antenas dependiendo del número de placas ROACH que hayan disponibles.

6. Bibliografía

Se procederá a presentar la bibliografía que se ha utilizado para el desarrollo de la memoria. Se pueden encontrar los enlaces a las páginas web de dónde se extrajeron imágenes y la información obtenida para el diseño del marco teórico de este documento.

- ALMA. (2011). *ALMA Website*. Recuperado el 5 de Diciembre de 2011, de <http://www.almaobservatory.org/es/tecnologia/front-end>
- ALMA. (2011). *ALMA Website*. Recuperado el 28 de Enero de 2012, de <http://www.almaobservatory.org/es/sobre-alma/origenes-del-proyecto-alma>
- ALMA. (2011). *ALMA Website*. Recuperado el 10 de Diciembre de 2011, de http://www.almaobservatory.org/science_articles/05_how_will_alma_make_images.pdf
- ALMA. (2011). *ALMA Website*. Recuperado el 14 de Diciembre de 2011, de http://www.almaobservatory.org/science_articles/07_the_ALMA_correlators.pdf
- CASPER. (2008). *CASPER Website*. Recuperado el 28 de Enero de 2012, de <https://casper.berkeley.edu/wiki/ROACH>
- CASPER. (2008). *CASPER Website*. Recuperado el 12 de Noviembre de 2011, de <https://casper.berkeley.edu/wiki/Tutorials>
- COSETI. (1990). *The Columbus Optical SETI Observatory*. Recuperado el 12 de Diciembre de 2011, de <http://www.coseti.org/9006-026.htm>
- ESO. (2011). *ESO*. Recuperado el 5 de Enero de 2012, de http://www.eso.org/public/images/elt_vlt/
- Goldsmith, P. F. (1998). *Quasioptical Systems*. IEEE Press.
- Instruments, N. (s.f.). *National Instruments Website*. Recuperado el 28 de Enero de 2012, de <http://www.national.com/pf/AD/ADC083000.html#Overview>
- Instruments, T. (s.f.). *Texas Instruments Website*. Recuperado el 3 de Enero de 2012, de <http://www.ti.com/product/ADC083000#parametrics>
- Meschede, D. (2007). *Optics, Light and Lasers: The Practical Approach to Modern Aspects of Photonics and Laser Physics* (Second ed.). Wiley.
- NCRA. (2012). *National Centre for Radio Astrophysics Website*. Recuperado el 4 de Enero de 2012, de http://www.ncra.tifr.res.in/gmrt_hpage/Users/doc/WEBLF/LFRA/node76.html
- NCRA. (2012). *National Centre for Radio Astrophysics Website*. Recuperado el 3 de Enero de 2012, de http://www.ncra.tifr.res.in/gmrt_hpage/Users/doc/WEBLF/LFRA/node77.html

NRAO. (s.f.). *National Radio-Astronomy Observatory Website*. Recuperado el 3 de Noviembre de 2011, de <http://www.aoc.nrao.edu/events/synthesis/2010/lectures10.html>

Richard Thompson, A., Moran, J., & George W. Swenson, J. (2001). *Interferometry and Synthesis in Radio Astronomy* (Second ed.). Wiley-Interscience.

Saleh, B., & Teich, M. (2007). *Fundamentals of Photonics* (Second ed.). Wiley.

Wilson, T., & Rohlfs, K. (2009). *Tools of Radio Astronomy* (Fifth ed.). Springer.

7. Anexos

En esta sección se encontrarán los códigos mencionados en el capítulo 3, para la implementación del software de configuración y extracción de datos de los correladores diseñados. En orden se encontrarán *corr_config.py*, *plot_corr_auto.py*, *plot_corr_cross.py* y finalmente *vegas_conf.py*.

corr_config.py

```
#Se procede a importar las librerías necesarias
import corr,time,numpy,struct,sys,logging,pylab

#Se define la ROACH a utilizar
roach = 'roach040209'

# IPs de los equipos del laboratorio
dest_ip = 192*(2**24)+168*(2**16)+101
src_ip = 192*(2**24)+168*(2**16)+150

#Definición de parámetros internos de la ROACH
dest_port = 60000
mac = 2199023518217
fabric_port = 60000
acc_len=(2**28)/1024
gain = 250

#Inicio de comunicaciones con la ROACH
fpga=corr.katcp_wrapper.FpgaClient(roach,7147)
time.sleep(1)

#Se define el archive a programar en la FPGA
boffile = 'correlator_2013_Jan_29_1426.bof'

#Desprogramar FPGA
fpga.progdev('')
time.sleep(2)

# Se programa el dispositivo y se entrega información del proceso
print 'Programando la FPGA con el siguiente bof: correlator_2013_Jan_29_1426.bof'
fpga.progdev(boffile)
print 'Configurando el fft_shift...',
fpga.write_int('fft_shift',(2**32)-1)
print 'listo'
print 'Configurando el periodo de acumulación',
fpga.write_int('acc_len',acc_len)
print 'listo'
```

```

print 'Reseteando la tarjeta, triggers del software y contadores de errores...',
fpga.write_int('ctrl',0)
fpga.write_int('ctrl',1<<17) #arm
fpga.write_int('ctrl',0)
fpga.write_int('ctrl',1<<18) #software trigger
fpga.write_int('ctrl',0)
fpga.write_int('ctrl',1<<18) #segundo trigger
print 'listo'

print 'Seteando la ganancia en todos los canales de todas las entradas a %i...'%gain,
fpga.write_int('quant0_gain',gain) # escribe la misma ganancia para todas las entradas y canales
fpga.write_int('quant1_gain',gain) #escribe la misma ganancia para todas las entradas y canales
for chan in range(1024):
    for input in range(2):
        fpga.blindwrite('quant%i_addr'%input,struct.pack('>I',chan))
print 'done'

print "OK, todo listo. Trata de graficar usando plot_corr_auto.py o plot_corr_cross.py"

```

plot_corr_auto.py

```

#Se procede a importar las librerías necesarias
import corr,time,numpy,struct,sys,logging,pylab,matplotlib

#Se define la ROACH y el puerto a utilizar
katcp_port=7147
roach = 'roach040209'

#Se define función en caso de fallas
def exit_fail():
    print 'FAILURE DETECTED. Log entries:\n',lh.printMessages()
    try:
        fpga.stop()
    except: pass
    raise
    exit()

#Se define función que cierra el script
def exit_clean():
    try:
        fpga.stop()
    except: pass
    exit()

```

#Se define función que extrae los datos de los registros del correlador
def get_data():

```
acc_n = fpga.read_uint('acc_num')
print 'Grabbing integration number %i'%acc_n

a_0=struct.unpack('>512l',fpga.read('dir_x0_aa_real',2048,0))
b_0=struct.unpack('>512l',fpga.read('dir_x0_bb_real',2048,0))
a_1=struct.unpack('>512l',fpga.read('dir_x1_aa_real',2048,0))
b_1=struct.unpack('>512l',fpga.read('dir_x1_bb_real',2048,0))
a_2=struct.unpack('>512l',fpga.read('dir_x2_aa_real',2048,0))
b_2=struct.unpack('>512l',fpga.read('dir_x2_bb_real',2048,0))
a_3=struct.unpack('>512l',fpga.read('dir_x3_aa_real',2048,0))
b_3=struct.unpack('>512l',fpga.read('dir_x3_bb_real',2048,0))
```

#Definición de arreglos que guardarán la información extraída de la ROACH

```
interleave_a=[]
interleave_b=[]
```

```
for i in range(256):
    interleave_a.append(a_0[i])
    interleave_a.append(a_1[i])
    interleave_b.append(b_0[i])
    interleave_b.append(b_1[i])
    interleave_a.append(a_2[i])
    interleave_a.append(a_3[i])
    interleave_b.append(b_2[i])
    interleave_b.append(b_3[i])
```

```
return acc_n,interleave_a,interleave_b
```

#Se define función encargada de graficar los datos guardados en los arreglos

def drawDataLoop():

```
matplotlib.pyplot.ion()
fig = matplotlib.pyplot.figure(num = 1, figsize=(18,12))
acc_n,interleave_a,interleave_b = get_data()

ax1 = fig.add_subplot(211)
if ifch:
    plot1, = matplotlib.pyplot.semilogy(interleave_a)
    matplotlib.pyplot.xlim(0,1024)
else:
    plot1, = matplotlib.pyplot.semilogy(xaxis,interleave_a)
matplotlib.pyplot.grid()
matplotlib.pyplot.title('Auto-Correlation \nIntegration: %ith cycle \nAA (Antenna A)'%acc_n)
matplotlib.pyplot.ylabel('Power (arbitrary units)')
```

```
matplotlib.pyplot.subplot(212)
```

```

if ifch:
    plot2, = matplotlib.pyplot.semilogy(interleave_b)
    matplotlib.pyplot.xlim(0,1024)
else:
    plot2, = matplotlib.pyplot.semilogy(xaxis,interleave_b)
matplotlib.pyplot.grid()
matplotlib.pyplot.ylabel('Power (arbitrary units)')
matplotlib.pyplot.title('BB (Antenna B)')
matplotlib.pyplot.xlabel('FFT Frequency (MHz)')

#Esta parte del código se genera en caso de que se encuentre alguna interrupción en el
#teclado para que se cierre el script
while(1):
    try:
        acc_n,interleave_a,interleave_b = get_data()
        ax1.set_title('Auto-Correlation \nIntegration: %ith cycle \nAA (Antenna A)'%acc_n)
        plot1.set_ydata(interleave_a)
        matplotlib.pyplot.hold(False)
        matplotlib.pyplot.draw()
        matplotlib.pyplot.hold(True)
        plot2.set_ydata(interleave_b)
        matplotlib.pyplot.hold(False)
        matplotlib.pyplot.draw()
        matplotlib.pyplot.hold(True)

    except (KeyboardInterrupt, SystemExit):
        print '\nKeyboard interrupt caught, exiting...\n'
        break

#Se define el main del script, el cual ejecutará las funciones definidas y recibirá los parámetros
#introducidos por el usuario
if __name__ == '__main__':

#Se importan librerías
    from optparse import OptionParser

#Se definen los parámetros de entradas que pueden ser llamados desde la consola dependiendo
#de lo que el usuario requiera, ya sea la información del eje x con la frecuencia o número de
#canales
    p = OptionParser()
    p.set_description(__doc__)
    p.add_option('-C','--channel',dest='ch',action='store_true',
        help='Set plot with channel number or frequency.')
    p.add_option('-f','--frequency',dest='fr',type='float',default=700.0,
        help='Set plot max frequency.(If -c sets to False)')
    opts, args = p.parse_args(sys.argv[1:])

    if opts.ch !=None:

```

```

    ifch = opts.ch
else:
    ifch = False

if ifch == False:
    if opts.fr != "":
        maxfr = opts.fr
    else:
        maxfr = 700.0
    xaxis = numpy.arange(0.0, maxfr, maxfr*1./1024)

# ifch significa si X será en número de canales

#Continuación del main donde se ejecutarán funciones para graficar.
try:
    loggers = []
    lh=corr.log_handlers.DebugLogHandler()
    logger = logging.getLogger(roach)
    logger.addHandler(lh)
    logger.setLevel(10)

#Se trata de comunicar con la ROACH
print('Connecting to server %s on port %i...'%(roach,katcp_port)),
fpga = corr.katcp_wrapper.FpgaClient(roach, katcp_port, timeout=10,logger=logger)
time.sleep(1)

if fpga.is_connected():
    print 'ok\n'
else:
    print 'ERROR connecting to server %s on port %i.\n'%(roach,katcp_port)
    exit_fail()

prev_integration = fpga.read_uint('acc_num')

# Comienza el proceso de ploteo
drawDataLoop()
print 'Exiting...'

#Detecta interrupciones del teclado en caso de que se quiera cerrar el script
except KeyboardInterrupt:
    exit_clean()
except:
    exit_fail()

exit_clean()

```

plot_corr_cross.py

```
#Importación de librerías necesarias.
import corr,time,numpy,struct,sys,logging,pylab,matplotlib

#Se define la ROACH y el puerto a utilizar
katcp_port=7147
roach = 'roach040209'

#Se define función en caso de fallas
def exit_fail():
    print 'FAILURE DETECTED. Log entries:\n',lh.printMessages()
    try:
        fpga.stop()
    except: pass
    raise
    exit()

#Se define función que cierra el script
def exit_clean():
    try:
        fpga.stop()
    except: pass
    exit()

#Se define función que extrae los datos de los registros del correlador
def get_data():

    acc_n = fpga.read_uint('acc_num')
    print 'Grabbing integration number %i'%acc_n

    #get the data...
    a_0r=struct.unpack('>512l',fpga.read('dir_x0_ab_real',2048,0))
    a_1r=struct.unpack('>512l',fpga.read('dir_x1_ab_real',2048,0))
    a_2r=struct.unpack('>512l',fpga.read('dir_x2_ab_real',2048,0))
    a_3r=struct.unpack('>512l',fpga.read('dir_x3_ab_real',2048,0))
    b_0r=struct.unpack('>512l',fpga.read('dir_x0_ab_real',2048,0))
    b_1r=struct.unpack('>512l',fpga.read('dir_x1_ab_real',2048,0))
    b_2r=struct.unpack('>512l',fpga.read('dir_x2_ab_real',2048,0))
    b_3r=struct.unpack('>512l',fpga.read('dir_x3_ab_real',2048,0))
    a_0i=struct.unpack('>512l',fpga.read('dir_x0_ab_imag',2048,0))
    a_1i=struct.unpack('>512l',fpga.read('dir_x1_ab_imag',2048,0))
    a_2i=struct.unpack('>512l',fpga.read('dir_x2_ab_imag',2048,0))
    a_3i=struct.unpack('>512l',fpga.read('dir_x3_ab_imag',2048,0))
    b_0i=struct.unpack('>512l',fpga.read('dir_x0_ab_imag',2048,0))
    b_1i=struct.unpack('>512l',fpga.read('dir_x1_ab_imag',2048,0))
    b_2i=struct.unpack('>512l',fpga.read('dir_x2_ab_imag',2048,0))
    b_3i=struct.unpack('>512l',fpga.read('dir_x3_ab_imag',2048,0))
```

```

#Se generan arreglos para guardar la información extraída de la ROACH
interleave_a=[]
interleave_b=[]

#Se llenan los arreglos
for i in range(256):
    interleave_a.append(complex(a_0r[i], a_0i[i]))
    interleave_a.append(complex(a_1r[i], a_1i[i]))
    interleave_a.append(complex(a_2r[i], a_2i[i]))
    interleave_a.append(complex(a_3r[i], a_3i[i]))
    interleave_b.append(complex(b_0r[i], b_0i[i]))
    interleave_b.append(complex(b_1r[i], b_1i[i]))
    interleave_b.append(complex(b_2r[i], b_2i[i]))
    interleave_b.append(complex(b_3r[i], b_3i[i]))

return acc_n,interleave_a,interleave_b

#Se genera función encargada de graficar los datos
def drawDataCallback():
    matplotlib.pyplot.clf()
    acc_n,interleave_a,interleave_b = get_data()

    matplotlib.pyplot.subplot(211)
    if ifch == True:
        matplotlib.pyplot.semilogy(numpy.abs(interleave_a))
        matplotlib.pyplot.xlim(0,1024)
    else:
        matplotlib.pyplot.semilogy(xaxis,numpy.abs(interleave_a))
    matplotlib.pyplot.grid()
    matplotlib.pyplot.title('Integration :%ith cycle \nCross Correlation (AB baseline)%acc_n)
    matplotlib.pyplot.ylabel('Power (arbitrary units)')

    matplotlib.pyplot.subplot(212)

    if ifch == True:
        matplotlib.pyplot.plot(numpy.angle(interleave_b))
        matplotlib.pyplot.xlim(0,1024)
        matplotlib.pyplot.xlabel('FFT Channel')
    else:
        matplotlib.pyplot.plot(xaxis,numpy.angle(interleave_b))
        matplotlib.pyplot.xlabel('FFT Frequency')
    matplotlib.pyplot.ylabel('Phase')
    matplotlib.pyplot.ylim(-4,4)
    matplotlib.pyplot.grid()

fig.canvas.manager.window.after(100, drawDataCallback)
matplotlib.pyplot.show()

```

```

#Se define el main del código

if __name__ == '__main__':
    from optparse import OptionParser

#Se generan los posibles parámetros que puede seleccionar el usuario al momento de ejecutar el
#script desde la consola
p = OptionParser()
p.set_description(__doc__)
p.add_option('-c', '--cross', dest='cross', type='str', default='ab',
             help='Plot this cross correlation magnitude and phase. default: ab')
p.add_option('-C', '--channel', dest='ch', action='store_true',
             help='Set plot with channel number or frequency.')
p.add_option('-f', '--frequency', dest='fr', type='float', default=700.0,
             help='Set plot max frequency.(If -c sets to False)')
opts, args = p.parse_args(sys.argv[1:])

if opts.ch != None:
    ifch = opts.ch
else:
    ifch = False

if ifch == False:
    if opts.fr != '':
        maxfr = opts.fr
    else:
        maxfr = 700.0
    xaxis = numpy.arange(0.0, maxfr, maxfr*1./1024)

baseline=opts.cross

#Se trata de conectar a la ROACH
try:
    loggers = []
    lh=corr.log_handlers.DebugLogHandler()
    logger = logging.getLogger(roach)
    logger.addHandler(lh)
    logger.setLevel(10)

    print('Connecting to server %s on port %i...'%(roach,katcp_port)),
    fpga = corr.katcp_wrapper.FpgaClient(roach, katcp_port, timeout=10,logger=logger)
    time.sleep(1)

    if fpga.is_connected():
        print 'ok\n'
    else:
        print 'ERROR connecting to server %s on port %i.\n'%(roach,katcp_port)
        exit_fail()

```

```

# Se definen las variables del gráfico a realizar
fig = matplotlib.pyplot.figure()
ax = fig.add_subplot(2,1,1)

# Se procede a graficar utilizando la función previamente definida
fig.canvas.manager.window.after(100, drawDataCallback)
matplotlib.pyplot.show()
print 'Plotting complete. Exiting...'

#Detecta interrupción en el teclado y cierra el script
except KeyboardInterrupt:
    exit_clean()
except:
    exit_fail()

exit_clean()

```

Vegas_conf.py

```

#Se importan librerías
import corr
import time

#Se define la ROACH a utilizar
roach = 'roach040209'

# Se definen las IPs del laboratorio
dest_ip = 192*(2**24)+168*(2**16)+101
src_ip = 192*(2**24)+168*(2**16)+150

#Definición de parámetros internos de la ROACH
dest_port = 60000
mac = 2199023518217
fabric_port = 60000
acc_len=1023

#Comienza la comunicación con la ROACH
fpga=corr.katcp_wrapper.FpgaClient(roach,7147)
time.sleep(1)

#Se define el archivo a programar en la FPGA
boffile='v01_na_ri_8r4t11f_ver108_2012_Mar_21_0636.bof'

```

```

# Desprograma la FPGA
fpga.progdev("")
time.sleep(2)

# Programa la FPGA
fpga.progdev(boffile)

# Define el puerto y la IP de la 10Gbe NIC
time.sleep(3)
#fpga.tap_start('tap0','gbe0',mac_base+src_ip,src_ip,fabric_port)
#fpga.tap_start('tap0','gbe0',mac,src_ip,fabric_port)
fpga.tap_start('gbe0','tap0',mac,src_ip,fabric_port)
time.sleep(3)

# Escribe la IP de destino
fpga.write_int('dest_ip',dest_ip)

# Escribe el puerto de destino
fpga.write_int('dest_port',dest_port)

# Establece el tamaño de acumulaciones
fpga.write_int('acc_len',acc_len)
time.sleep(1)

# Establece el periodo de sincronización y escribe condiciones iniciales en los registros de control
fpga.write_int('sg_period',2*16*1024*1024/8 -2)
fpga.write_int('sg_sync',0x12)
fpga.write_int('arm',0)
fpga.write_int('arm',1)
fpga.write_int('arm',0)

time.sleep(1)
fpga.write_int('sg_sync',0x13)

#Se define la función reset en caso de que se requiera, para restaurar el correlador con los
#parámetros iniciales
def reset():
    fpga.write_int('sg_sync',0x12)
    fpga.write_int('arm',0)
    fpga.write_int('arm',1)
    fpga.write_int('arm',0)
    fpga.write_int('sg_sync',0x12)
    fpga.write_int('sg_sync',0x11)

reset()

```