

UNIVERSITAT POLITÈCNICA DE CATALUNYA

DEPARTAMENT DE TEORIA DEL SENYAL I COMUNICACIONS

Proyecto Final de Carrera

“ALGORITMOS DE CALIBRACIÓN PARA RADIÓMETROS DIGITALES”

Autor:

Miguel Ángel Guerrero

Directores:

Xavier Bosch i Lluís

Dr. Adriano Camps Carmona

Barcelona, Mayo de 2009

“ALGORITMOS DE CALIBRACIÓN PARA RADIÓMETROS DIGITALES”

This work, conducted as part of the award “Passive Advanced Unit (PAU): A Hybrid L-band Radiometer, GNSS-Reflectometer and IR-Radiometer for Passive Remote Sensing of the Ocean” made under the European Heads of Research Councils and European Science Foundation EURYI (European Young Investigator) Awards scheme in 2004, was supported by funds from the Participating Organisations of EURYI and the EC Sixth Framework Programme.

AGRADECIMIENTOS

Llega el momento de poner fin a una etapa de mi vida para comenzar con toda la ilusión del mundo una nueva etapa que, espero, sea tan plena y satisfactoria como la anterior. Es ahora, justo en mitad del camino, cuando me detengo por unos instantes y, sin perder de vista el camino que todavía me queda por recorrer, vuelvo la vista atrás para darme cuenta del enorme apoyo que he recibido de cuantos me rodean. Por este motivo quiero dedicar las primeras líneas de este proyecto a agradecer a todos el apoyo que me habéis brindado:

... A Adriano Camps, por darme la oportunidad de participar en este proyecto y por los sabios consejos recibidos durante todo este tiempo.

... A Xavi Bosch, por la paciencia que ha tenido conmigo, por no desfallecer nunca por muy duras que fuesen las circunstancias y mantener en todo momento alta la moral y por la inmensa cantidad de horas que hemos trabajado codo con codo persiguiendo un mismo objetivo.

... A todos los miembros del equipo PAU y compañeros de laboratorio: Banqué, Isaac, Benjamín, Nereida, Juan Fernando, Enric, José Miguel, Juan Manuel, Javier, Francesc, Carlos... por estar siempre dispuestos a escuchar mis problemas y aportar soluciones cuando les fue posible y consuelo cuando no.

... A todos mis compañeros y amigos de la universidad: Maruan, Bernat, Jokin, Paco, Cesc, Marc, Paska, Angels, Laia, David, Mercé, Román, Xavi, Javi, Victor, Dani, Oscar, Gemma, Laura... (disculpad que no continúe, no acabaría nunca si no). Porque sin duda alguna, sois lo mejor que me llevo de Barcelona.

... A todos los compañeros que durante estos años han compartido piso conmigo, porque habéis compartido mucho más que un simple piso, y ya sois parte de mi vida.

... A los compañeros de Indra, porque parte de los conocimientos que he necesitado para completar este proyecto los aprendí de vosotros.

... A mis padres y hermanos y en general a toda mi familia, por todo el esfuerzo realizado y porque sin vuestro apoyo durante todos estos años, todo esto no hubiera sido posible.

... A todos mis amigos y familiares de Ibiza por los ánimos que me habéis dado durante todo este tiempo y por interesaros constantemente por este proyecto, aún a sabiendas de que lo único que ibais a recibir era un galimatías incomprensible.

... A Esther, porque por mucho que creas que no mereces figurar en estos agradecimientos, lo cierto es que te lo mereces más que nadie, tanto, que si pudiese quitaría mi nombre de la portada y pondría el tuyo. Gracias por estar a mi lado durante todo este tiempo, por todo el cariño y por tu infinita paciencia conmigo. Te prometo que te recompensaré por todo.

... A ti, porque si por lo que sea estás leyendo estas líneas, sin duda mereces mi agradecimiento más sincero.

Sinceramente,
Miguel Ángel Guerrero.

LISTADO DE ACRÓNIMOS

ADC	Analog to Digital Converter
ALU	Arithmetic Logic Unit
ANSI	American National Standard Institute
API	Application Program Interface
ARMA	Auto Regressive Moving Average
EURYI	EUROpean Young Investigator
FI	Frecuencia Intermedia
FIR	Finite Impulse Response
FPGA	Field-Programmable Gate Array
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
HAL	Hardware Abstraction Layer
IDE	Integrated Development Environment
IIR	Infinite Impulse Response
IP	Intellectual Property
ISR	Interrupt Service Routine
LAURA	L-band Automatic Radiometer
LNA	Low Noise Amplifier
LWIP	Light Weight Internet Protocol
MBE	Main Beam Efficiency
MDDC	Multiplex Digital Down Converter
MIRAS	Microwave Imaging Radiometer using Aperture Synthesis
NIR	Noise Injection Radiometer
NLPS	Nivel de Lóbulo Principal a Secundario
OS	Operating System
PAU	Passive Advanced Unit
PC	Personal Computer
PIO	Parallel Input Output
RF	Radio Frequency
RISC	Reduced Instruction Set Computer
RMS	Root Mean Square
RUT	Receiver Under Test
SDK	Software Development Kit
SDRAM	Synchronous Dynamic Random Access Memory
SICS	Sweedish Institute of Computer Science
SMOS	Soil Moisture and Ocean Salinity
SOPC	System On Programmable Chip
SRAM	Synchronous Random Access Memory
SSS	Sea Surface Salinity
SST	Sea Surace Temperature
STP	Shielded Twisted Pair

TCP/IP	Transport Control Protocol/Internet Protocol
TPR	Total Power Radiometer
TT	Tabla Trigonométrica
UART	Universal Asynchronous Receiver Transmitter
UC	Unidad de Control
UPC	Universidad Politécnica de Cataluña
VHDL	Very high speed Integrated circuit Hardware Description Language

ÍNDICE

1 INTRODUCCIÓN	15
1.1 Antecedentes.....	15
1.2 El proyecto PAU	15
1.3 Motivación de PAU	16
2 PRINCIPIOS DE RADIOMETRÍA.....	19
2.1 Introducción.....	19
2.2 Principios básicos de radiometría	19
2.2.1 Densidad espectral de brillo y potencia captada por una antena.	20
2.2.2 Radiación térmica	20
2.2.3 Radiación de un cuerpo negro.....	21
2.2.4 Ruido térmico captado por una antena.....	23
2.2.5 Radiación de un cuerpo gris	24
2.2.6 Parámetros de Stokes.....	27
2.3 Tipos de radiómetros	28
2.3.1 Radiómetro de potencia total (TPR)	28
2.3.2 Radiómetro de Dicke	32
2.3.3 Radiómetro de inyección de ruido (NIR)	34
2.3.4 Radiómetros polarimétricos	36
2.4 Aplicaciones de la radiometría.....	42
2.4.1 Justificación al uso de frecuencias GPS en radiometría.....	42
3 DESCRIPCIÓN DE PAU-RAD.....	43
3.1 Introducción.....	43
3.2 Estructura de PAU-RAD.....	43
3.2.1 Antena	44
3.2.2 Receptor	47
3.2.3 Conversor Analógico-Digital	50
3.2.4 FPGA	52
4 UNIDAD DE CONTROL E INTERFAZ DE CAPTURA DE DATOS	63
4.1 Introducción.....	63

4.2 Unidad de control de PAU-RAD	64
4.2.1 Descripción	64
4.2.2 Arquitectura	66
4.2.3 Entradas y salidas	70
4.2.4 Parámetros del sistema	71
4.2.5 Estados de la <i>Secuencia Principal</i>	74
4.2.6 Estados de las <i>Secuencias de Validación</i>	79
4.3 Interfaz de captura de datos.....	81
4.3.1 Descripción	81
4.3.2 Arquitectura	85
4.3.3 Comportamiento en función de la secuencia de ejecución.....	87
4.4 Conclusiones	91
5 ALGORITMOS DE CALIBRACIÓN	93
5.1 Introducción.....	93
5.1.1 Importancia de la calibración	93
5.1.2 Descripción general del proceso de calibración	95
5.2 Tipos de algoritmos estudiados	97
5.2.1 Introducción	97
5.2.2 Algoritmos de respuesta finita	99
5.2.3 Algoritmos de respuesta infinita	100
5.2.4 Algoritmos no diferenciales.....	101
5.2.5 Algoritmos diferenciales.....	102
5.2.6 Hardware de calibración diferencial.....	102
5.3 Calibración de fase.....	114
5.3.1 Calibración de fase mediante algoritmo no diferencial.....	114
5.3.2 Implementación en la unidad de control.....	117
5.3.3 Resultados y conclusiones	122
5.3.4 Calibración de fase mediante algoritmo diferencial.....	124
5.3.5 Implementación en la Unidad de Control.....	127
5.3.6 Resultados y conclusiones	130
5.3.7 Pruebas de validación de los algoritmos	132
5.3.8 Pruebas de validación del hardware.....	135
5.4 Calibración de ganancia	142
5.4.1 Calibración de ganancia mediante algoritmo no diferencial	142
5.4.2 Implementación en la Unidad de Control.....	143
5.4.3 Resultados y conclusiones	144
5.4.4 Calibración de ganancia mediante algoritmo diferencial	145

5.4.5 Implementación en la Unidad de Control.....	147
5.4.6 Resultados y conclusiones	150
5.4.7 Pruebas de validación de los algoritmos	151
5.4.8 Pruebas de validación del hardware.....	152
5.5 Escalado del sistema	160
5.5.1 Escalado a 4 receptores.....	160
5.5.2 Consideraciones necesarias para el escalado a 16 receptores.....	165
6 FILTRADO DEL ERROR DE FASE	169
6.1 Introducción.....	169
6.1.1 Mecanismo de calibración de fase	169
6.1.2 Justificación al filtrado	170
6.1.3 Tipos de filtros considerados.....	171
6.2 Filtrado de fase mediante filtros de respuesta impulsional infinita (IIR)	172
6.2.1 Características de los filtros IIR.....	172
6.2.2 Diseño del filtro	173
6.2.3 Resultados obtenidos en el simulador.....	177
6.2.4 Análisis de los resultados y conclusiones	178
6.3 Filtrado de fase mediante filtros de respuesta impulsional finita (FIR).....	180
6.3.1 Características de los filtros FIR.....	180
6.3.2 Diseño del filtro	180
6.3.3 Resultados obtenidos en el simulador.....	182
6.3.4 Análisis de los resultados y conclusiones	183
6.4 Filtrado de fase mediante filtro de Kalman.....	184
6.4.1 Características del filtro de Kalman	184
6.4.2 Diseño del filtro	185
6.4.3 Resultados obtenidos con el simulador	186
6.4.4 Análisis de los resultados y conclusiones	187
6.4.5 Inconvenientes del filtrado Kalman	189
7 MIGRACIÓN DEL SISTEMA A STRATIX II.....	193
7.1 Introducción.....	193
7.2 Dispositivos de la familia Stratix II	194
7.2.1 Placa de desarrollo Stratix II	194
7.2.2 Comparativa Stratix I Vs. Stratix II	195

7.3 Problemática de la migración	196
7.3.1 Incompatibilidad entre sistemas	196
7.3.2 Soluciones consideradas.....	200
7.4 Solución Implementada	204
7.4.1 Nueva Unidad de Control	204
7.4.2 Nueva Interfaz de captura de datos	207
8 CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS.....	221
9 BIBLIOGRAFIA	225

1 INTRODUCCIÓN

1.1 ANTECEDENTES

Desde hace algún tiempo, el Grupo de Radiometría de Microondas de la UPC viene colaborando en proyectos de teledetección pasiva con la Agencia Espacial Europea o ESA (European Space Agency). Uno de los proyectos de la ESA que goza de la participación activa del Grupo de Radiometría de Microondas es la misión SMOS (Soil Moisture and Ocean Salinity), cuyo objetivo es recuperar la humedad del suelo y la salinidad superficial oceánica. Para ello cuenta con un radiómetro de interferometría en banda L, el MIRAS (Microwave Imaging Radiometer using Aperture Synthesis) embarcado en un satélite.

Por otra parte, el Grupo de Ingeniería Electrónica y Fotónica de la UPC, también había realizado trabajos en el campo de la monitorización del estado de la superficie del mar mediante el uso de reflectometría aplicada a las señales de los sistemas de posicionamiento global por satélite GNSS (Global Navigation Satellite System).

En este contexto nace el proyecto PAU (Passive Advanced Unit), como una solución pensada para combinar ambas tecnologías, reflectometría y radiometría, en un único dispositivo. PAU es la materialización de una idea del profesor Adriano Camps, galardonada con el premio EURYI (European Young Investigators Awards) de la European Science Foundation en agosto de 2004 [1].

1.2 EL PROYECTO PAU

PAU es un sistema híbrido de teledetección pasiva, compuesto de tres instrumentos: un radiómetro en banda L (PAU-RAD), un reflectómetro GNSS (PAU-GNSS), y un radiómetro de infrarrojos (PAU-IR).

Tanto PAU-RAD como PAU-GNSSR comparten una misma agrupación de antenas, así como el mismo cabezal de radio frecuencia y los digitalizadores de señal. Mediante el uso de conformación digital del haz, es posible sintetizar para ambos sistemas múltiples haces con diferentes ángulos de incidencia y diferentes niveles de lóbulo principal a secundario.

La finalidad de este sistema es obtener de forma remota:

- El estado de la superficie del mar (PAU-GNSS).

- La salinidad de la superficie del mar (PAU-RAD).
- La temperatura de la superficie del mar (PAU-IR).

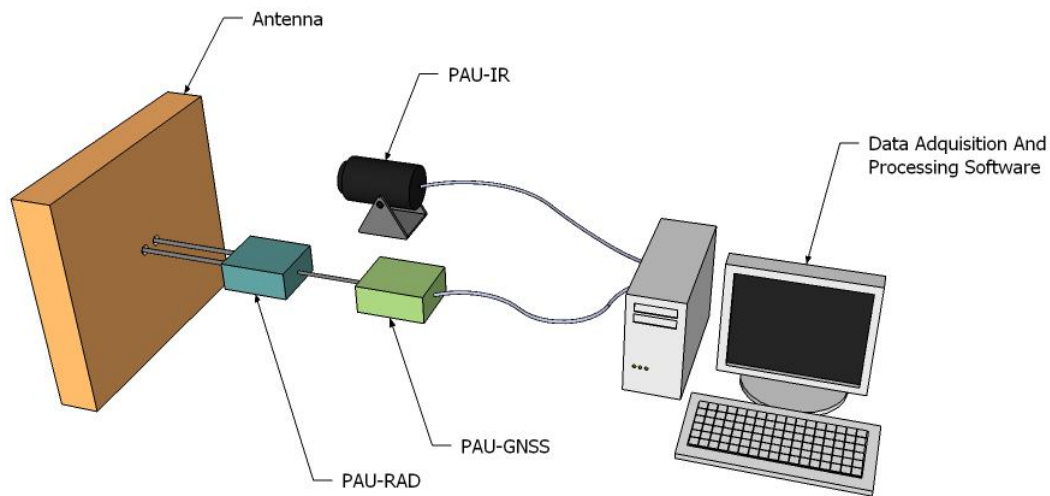


Figura 1-1: Esquema que ilustra los diferentes componentes de PAU y sus interconexiones.

Mediante la combinación de estos tres parámetros es posible obtener información sobre la dinámica conjunta del océano y la atmósfera, lo que permitirá elaborar modelos del comportamiento climático terrestre.

El trabajo desarrollado en este proyecto de final de carrera se enmarca dentro del ámbito de PAU-RAD, y se distribuye entre el software contenido dentro del propio instrumento y el software de adquisición y procesado de datos externo. Por tanto esta memoria se centrará únicamente en dicho instrumento.

1.3 MOTIVACIÓN DE PAU

La salinidad superficial del mar o SSS (Sea Surface Salinity) junto con la temperatura superficial del mar o SST (Sea Surface Temperature) forman el conocido par termo-halino, este constituye uno de los elementos clave a la hora de estimar la influencia de los océanos en el clima, ya que es un indicador del equilibrio termodinámico oceánico, parámetro esencial para comprender la dinámica del clima. La SSS, junto con la SST, constituye una herramienta muy valiosa en lo que a la mejora de las estimaciones de evaporización/precipitación se refiere, lo que permite predecir períodos de sequía. Asimismo, una mejora en la medida de la SSS permitirá cuantificar el papel de la salinidad en los cambios del nivel del mar y mejorar la predicción de fenómenos meteorológicos como el Niño.

A pesar de su importancia, las medidas de la salinidad superficial del mar de que se dispone se hallan muy heterogeneamente distribuidas, por lo que el

conocimiento de la distribución espacial y la variación temporal de la SSS es todavía limitado para la mayor parte de la superficie oceánica.

Los satélites constituyen la plataforma ideal para la monitorización de los parámetros geofísicos de nuestro planeta. Mediante la utilización de sistemas pasivos embarcados en satélites, es posible obtener un gran volumen de información que permitirá elaborar mapas de salinidad de forma global y precisa.

El proyecto PAU constituye una prueba de concepto de la integración en un único dispositivo de dos tecnologías que pueden jugar un papel clave en el futuro de la oceanografía por satélite.

2 PRINCIPIOS DE RADIOMETRÍA

2.1 INTRODUCCIÓN

La radiometría es la rama de la ciencia encargada de la medida de la radiación que, de forma espontánea, emite cualquier superficie que se encuentre a una temperatura diferente de 0 Kelvin, así como del posterior estudio de las relaciones entre esta radiación y los atributos físicos y biológicos de dicha superficie.

Mediante la observación y comprensión de los procesos físicos asociados con la emisión espontánea de energía de las superficies en observación, es posible calcular gran variedad de parámetros atmosféricos y geofísicos (algunos de los cuales se comentan en el apartado 2.4). Por ello, no es de extrañar que el uso de técnicas radiométricas, originalmente empleadas en radioastronomía, haya sufrido una migración progresiva hacia aplicaciones de monitorización de los parámetros del planeta Tierra.

El instrumento encargado de la medición de esta energía es el radiómetro, en esencia, un dispositivo que mide la potencia de microondas captada por una antena con una gran precisión.

En este capítulo se introducirán algunos conceptos fundamentales de radiometría que se emplearán más adelante, se hará una descripción de los tipos de radiómetros más comunes, que permitirá sentar las bases para la descripción de la arquitectura del radiómetro empleado en este proyecto y finalmente se indicarán algunas de las principales aplicaciones radiométricas.

2.2 PRINCIPIOS BÁSICOS DE RADIOMETRÍA

A continuación se definirán algunos de los conceptos básicos de radiometría que constituyen el fundamento teórico a la hora de desarrollar las ecuaciones que se emplearán más adelante.

2.2.1 Densidad espectral de brillo y potencia captada por una antena.

La potencia total captada por una antena se expresa en términos de su diagrama de radiación: $t(\theta, \phi)$ i de la densidad espectral de brillo de la superficie a la que enfoca: $B_f(\theta, \phi)$ que se define a su vez como brillo por ancho de banda y cuyas unidades son $[W \cdot Sr^{-1} \cdot m^{-2} \cdot Hz^{-1}]$.

El brillo de un cuerpo es la potencia emitida por unidad de ángulo sólido y por unidad de superficie radiante. Si se considera una superficie emisiva con un área total radiante A_t que radia una potencia con un determinado diagrama de radiación $F_T(\theta, \phi)$ expresado en vatios por estereorradián, entonces el brillo B se define como:

$$B(\theta, \phi) = \frac{F_T(\theta, \phi)}{A_t} [W \cdot Sr^{-1} \cdot m^{-2}] \quad (2.1)$$

Así pues la potencia total captada por una antena enfocada a dicha superficie radiante se puede expresar como:

$$P = \frac{1}{2} A_r \int_{f_0}^{f_0+B} \iint_{4\pi} B_f(\theta, \phi) t(\theta, \phi) d\Omega df [W] \quad (2.2)$$

Donde A_r es el área efectiva de la antena, $d\Omega$ es el diferencial de ángulo sólido correspondiente al diferencial de potencia recibido por la antena, f_0 es la frecuencia a la que está centrado el ancho de banda B y el factor $\frac{1}{2}$ tiene en cuenta el hecho de que la antena con polarización lineal únicamente captará la mitad de la potencia emitida por la fuente.

2.2.2 Radiación térmica

Cualquier cuerpo que se encuentre a una temperatura absoluta diferente de cero grados Kelvin, emite de manera espontánea energía en forma de ondas electromagnéticas. Este fenómeno, conocido como radiación térmica, se debe al paso de los electrones de una órbita a otra de menor energía. La radiación electromagnética emitida por el electrón al cambiar de órbita presentará una frecuencia determinada por la ecuación de Bohr:

$$f = \frac{\varepsilon_1 - \varepsilon_2}{h} \quad [\text{Hz}] \quad (2.3)$$

Donde ε_1 representa la energía de la órbita inicial (estado de excitación), ε_2 es la energía de la órbita final (estado de reposo) y h es la constante de Planck ($h = 6,63 \cdot 10^{-34} \text{ [J} \cdot \text{s]}$).

Para que un electrón ascienda a una órbita con un nivel de energía mayor, es necesario un aporte de energía exterior, ya sea en forma de energía electromagnética (mediante la absorción de un fotón) o en forma de energía térmica (mediante la absorción de un fonón). La probabilidad de absorción de un fonón depende de la densidad volumétrica de átomos así como de su energía cinética. Es posible expresar la energía cinética de los átomos en términos de temperatura absoluta, lo que implica que la cantidad de energía radiada por un cuerpo aumentará con la temperatura.

2.2.3 Radiación de un cuerpo negro

En general, cierta cantidad del total de radiación electromagnética incidente sobre una superficie es absorbida por la misma, reflejando el resto. La energía absorbida por la superficie se transforma, según establecen los principios de la termodinámica, en energía térmica, lo que provoca un incremento de la temperatura de la superficie. Del mismo modo, el proceso inverso cumple la ley de conservación de la energía, efectuando el balance entre la radiación absorbida, emitida y transmitida. Así pues, en equilibrio termodinámico, la energía absorbida por la superficie se vuelve a radiar.

Se denomina cuerpo negro a aquel cuerpo ideal capaz de absorber toda la radiación incidente a cualquier frecuencia. A partir de la ley de radiación de Planck, es posible determinar que un cuerpo negro radia de manera uniforme en todas direcciones y con una densidad espectral de brillo que viene determinada por la expresión:

$$B_f = \frac{2hf^3}{c^2} \frac{1}{e^{\frac{hf}{kT_{ph}}} - 1} \quad [\text{W} \cdot \text{m}^{-2} \cdot \text{Hz}^{-1} \cdot \text{Sr}^{-1}] \quad (2.4)$$

Donde f es la frecuencia de la energía radiada, h es la constante de Planck, k es la constante de Boltzman ($k = 1,28 \cdot 10^{-23} \text{ [J} \cdot \text{K}^{-1}]$), T_{ph} es la temperatura física absoluta en grados Kelvin y c es la velocidad de la luz ($c = 3 \cdot 10^8 \text{ [m} \cdot \text{s}^{-1}]$).

De la ecuación 2.4 se deduce que la densidad espectral de brillo depende tanto de la frecuencia como de la temperatura física absoluta. En la figura 2-1 se representan

varias curvas de la densidad espectral de brillo de un cuerpo negro en función de la frecuencia y de la temperatura física absoluta.

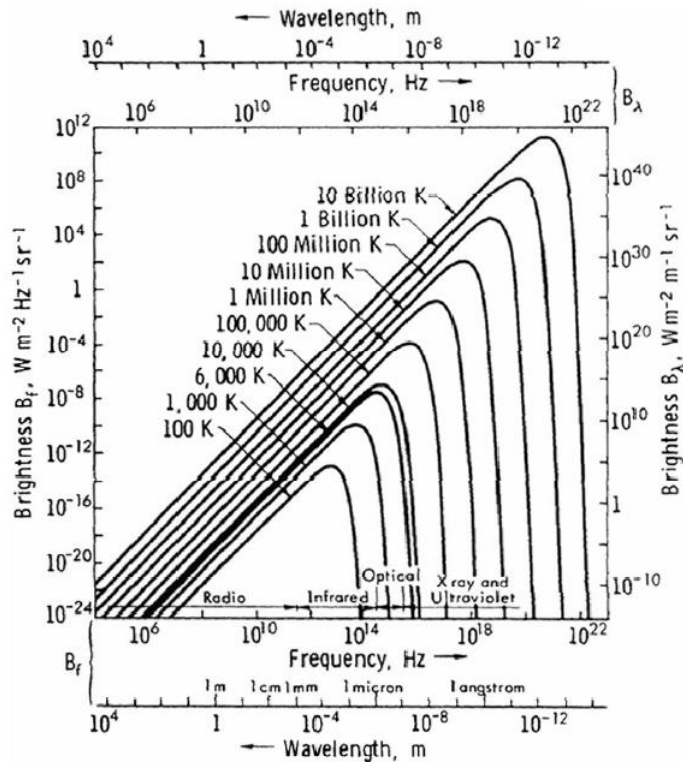


Figura 2-1: Curvas de radiación según la ley de Planck [10].

Como se puede apreciar, todas las curvas de radiación presentan dos zonas claramente diferenciadas: una primera zona correspondiente a frecuencias bajas en la que las curvas poseen un comportamiento creciente; y una segunda zona para frecuencias mayores en la que las curvas son descendientes. En la primera zona, el comportamiento de las curvas se puede aproximar por la ley de radiación de Rayleigh-Jeans, mientras que en la segunda se puede aproximar por la ley de Wien.

Para las frecuencias de microondas y en particular para la banda L (comprendida entre 1 y 2 GHz) en la que trabaja PAU-RAD, las curvas de radiación se encuentran en la primera zona, y por tanto es posible aproximarlas por la ley de Rayleigh-Jeans. Puesto que en banda L se cumple que:

$$\frac{h \cdot f}{k \cdot T_{ph}} \rightarrow 0 \tag{2.5}$$

Es posible aplicar una serie de Taylor del término exponencial de la ecuación 2.4 es decir:

$$e^x - 1 \cong 1 + x + \frac{x^2}{2} + \dots - 1 \quad \text{para } x \ll 0 \quad (2.6)$$

Y por tanto, aplicando esta aproximación a la ecuación 2.4 se tiene que:

$$B_f \cong \frac{2hf^2kT_{ph}}{c^2} = \frac{2kT_{ph}}{\lambda^2} \quad [W \cdot m^{-2} \cdot Hz^{-1} \cdot Sr^{-1}] \quad (2.7)$$

Para temperaturas físicas cercanas a los 300 K y frecuencias inferiores a 117 GHz, el error cometido al emplear esta aproximación es inferior al 1%. Así pues, la ecuación 2.7 pone de manifiesto que existe una relación lineal entre la temperatura física de un cuerpo y su densidad espectral de brillo.

2.2.4 Ruido térmico captado por una antena

Una vez introducidos los conceptos de cuerpo negro y densidad espectral de brillo es posible determinar la potencia captada por una antena. Si la fuente de radiación es un cuerpo negro, la potencia captada por una antena con un diagrama de radiación $t(\theta, \phi)$ enfocada a dicho cuerpo negro se obtiene aplicando la expresión de Rayleigh-Jeans a la ecuación 2.2, es decir:

$$P = \frac{1}{2} A_r \int_{f_0}^{f_0+B} \iint_{4\pi} \frac{2kT_{ph}}{\lambda^2} t(\theta, \phi) d\Omega df \quad [W] \quad (2.8)$$

Puesto que la potencia captada por la antena estará limitada por el ancho de banda de la misma (B), si se considera un ancho de banda lo suficientemente pequeño, es posible suponer que la densidad espectral de brillo de un cuerpo negro que rodee la antena será constante para todo el margen de frecuencias del sistema. Así pues es posible simplificar la ecuación 2.8 de tal manera que:

$$P = kT_{ph} B \frac{A_r}{\lambda^2} \iint_{4\pi} t(\theta, \phi) d\Omega \quad [W] \quad (2.9)$$

Si además se expresa el ángulo sólido de la antena en función de su área efectiva:

$$\Omega_{ef} = \iint_{4\pi} t(\theta, \phi) d\Omega = \frac{\lambda^2}{A_r} \quad (2.10)$$

Es posible obtener una forma más simplificada de la ecuación 2.9 para la determinación de la potencia captada por la antena:

$$P = kT_{ph}B \frac{A_r}{\lambda^2} \Omega_{ef} = kT_{ph}B \quad [W] \quad (2.11)$$

Esta expresión pone de manifiesto la relación lineal existente entre la temperatura física de un cuerpo negro y la potencia captada por una antena que estuviese rodeada por éste. Esta es exactamente la misma relación que obtuvieron en 1928 Johnson y Nyquist para el caso de un receptor ideal de ancho de banda B con una resistencia adaptada a la entrada que estuviese a la misma temperatura física que la antena, como se muestra en la figura 2-2:

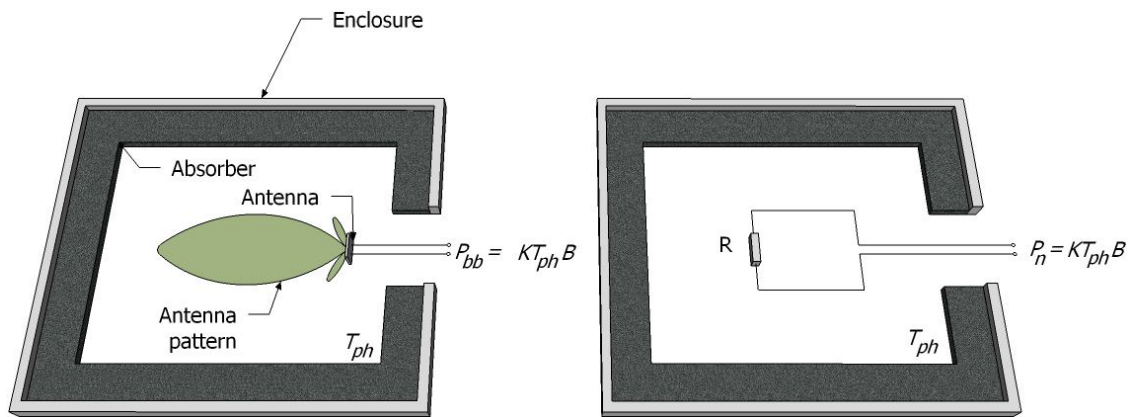


Figura 2-2: Antena en el interior de un cuerpo negro y resistencia en el interior de un cuerpo negro, ambas a la misma temperatura T .

Esto implica que a efectos de un receptor ideal de ancho de banda B , la antena se comporta como una resistencia adaptada al receptor que se encuentra a una temperatura física igual a la temperatura de la antena.

2.2.5 Radiación de un cuerpo gris

Los resultados obtenidos para el caso de un cuerpo negro no aplican de manera general a la mayoría de los materiales reales, ya que se trata de un caso ideal. Los materiales reales no absorben toda la energía que incide sobre ellos y por lo tanto emiten menos energía que un cuerpo negro. A estos materiales se les denomina de forma genérica cuerpos grises.

Como se ha mencionado, el brillo de un cuerpo gris es inferior al de un cuerpo negro. La relación entre ambos se define como emisividad $e(\theta, \phi)$:

$$e(\theta, \phi) = \frac{B(\theta, \phi)}{B_{cn}} = \frac{T_B(\theta, \phi)}{T_{ph}} \quad (2.12)$$

Donde B_{cn} representa el brillo de un cuerpo negro, T_B es la temperatura de brillo y T_{ph} es la temperatura física.

Puesto que $B(\theta, \phi) \leq B_{cn}$, por definición, la emisividad es un valor comprendido entre 0 y 1, siendo 1 el valor de emisividad de un cuerpo negro (absorbente perfecto) y 0 el valor de emisividad de un reflector perfecto (metal sin pérdidas). De lo obtenido en la expresión 2.12 es posible deducir que la temperatura de brillo de un material será siempre menor que su temperatura física.

A continuación se define otro parámetro importante a la hora de trabajar con cuerpos grises: la temperatura aparente de brillo.

2.2.5.1 Temperatura aparente de brillo y temperatura de antena

Lejos de proceder únicamente de la temperatura de brillo del cuerpo al que se enfoca, la potencia captada por una antena constituye el resultado de la suma de las diferentes contribuciones de todas las fuentes que radian sobre ésta. La figura 2-3 ilustra este efecto:

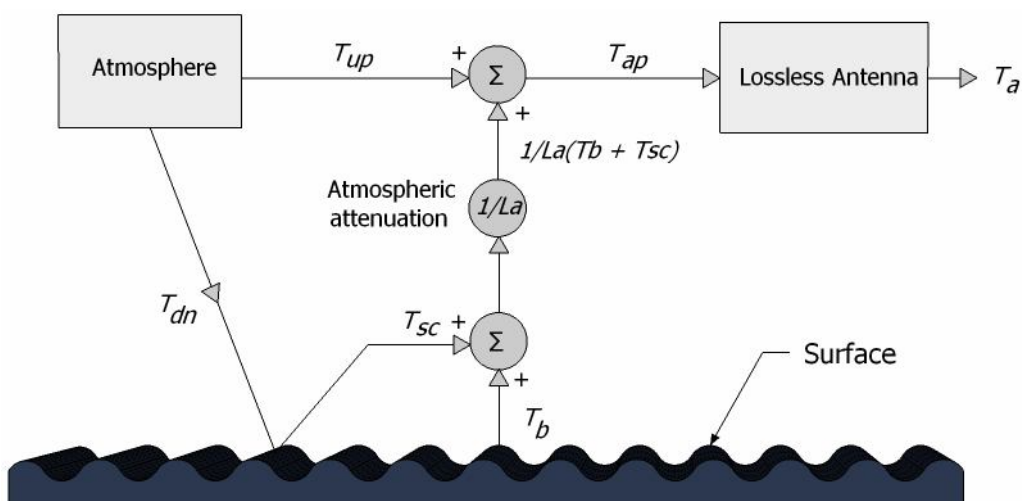


Figura 2-3: Esquema que ilustra la procedencia de las diferentes contribuciones que componen la potencia captada por una antena [10].

Como se puede observar, la radiación captada por antena es el resultado de las siguientes contribuciones:

T_B : Temperatura de brillo de la superficie bajo observación.

T_{UP} : Emisión atmosférica ascendente.

T_{SC} : Emisión atmosférica descendente reflejada en la superficie.

L_A : Atenuación atmosférica.

Por tanto, la temperatura aparente de brillo se puede calcular como:

$$T_{ap} = T_{UP} + \frac{1}{L_a} (T_{SC} + T_B) \quad [K] \quad (2.13)$$

A partir de aquí es posible obtener la temperatura de antena integrando la temperatura aparente $T_{AP}(\theta, \phi)$ ponderada por el diagrama de radiación normalizado de la antena $t(\theta, \phi)$ en un ángulo sólido de 4π y normalizando el resultado por el modelo de ángulo sólido Ω_p , es decir:

$$T_A = \frac{1}{\Omega_p} \iint_{4\pi} T_{AP}(\theta, \phi) t(\theta, \phi) d\Omega \quad [W] \quad (2.14)$$

La temperatura aparente de brillo es un parámetro importante a tener en cuenta en aplicaciones radiométricas, ya que la antena se encuentra ubicada a mucha distancia de la superficie radiante y por tanto habrá muchas contribuciones diferentes sobre la potencia captada total.

A parte de las contribuciones a la potencia captada por la antena ya mencionadas, existe también una serie de contribuciones que es necesario tener en cuenta cuando se trabaja en aplicaciones radiométricas. Las más importantes son: la temperatura de brillo de procedencia extraterrestre, que depende del ruido cósmico i del ruido galáctico; la temperatura de brillo procedente del sol ("Sun Glint") y otros efectos atmosféricos. Una descripción detallada de cómo afectan estas contribuciones a las medidas de un radiómetro se puede encontrar en [2].

2.2.6 Parámetros de Stokes

Las ondas electromagnéticas son ondas transversales y la polarización del campo eléctrico se puede representar en un sistema de coordenadas formado por dos vectores ortogonales y unitarios en el plano ortogonal a la dirección de propagación de la onda. De manera general, la radiación electromagnética emitida por un objeto se encuentra parcialmente polarizada, lo que significa que la potencia de la componente horizontal es diferente de la potencia de la componente vertical.

El vector de Stokes es una herramienta empleada en aplicaciones radiométricas para el estudio de las ondas electromagnéticas parcialmente polarizadas, de manera que, normalmente, las medidas radiométricas se expresan en términos de los parámetros de Stokes. El vector de Stokes se define como:

$$\vec{S} = \begin{bmatrix} S_0 \\ S_1 \\ S_2 \\ S_3 \end{bmatrix} = \begin{bmatrix} I \\ Q \\ U \\ V \end{bmatrix} = \begin{bmatrix} |E_U|^2 + |E_W|^2 \\ |E_U|^2 - |E_W|^2 \\ 2\text{Re}\langle E_U \cdot E_W^* \rangle \\ 2\text{Im}\langle E_U \cdot E_W^* \rangle \end{bmatrix} \left[\frac{W}{\text{m}^2 \cdot \text{Hz}} \right] \quad (2.15)$$

Donde S_0 es la potencia total disponible, E representa el campo eléctrico, Re e Im constituyen las partes real e imaginaria respectivamente y los subíndices U y W denotan la proyección del campo eléctrico sobre dos vectores ortogonales unitarios ubicados en plano ortogonal a la dirección de propagación de la onda.

De manera habitual, en aplicaciones radiométricas tanto el primer como el segundo elemento del vector de Stokes se suelen expresar de forma equivalente, al mismo tiempo que se sustituyen los vectores unitarios U y W , más generales, por las componentes horizontal y vertical del campo eléctrico. Así pues, el vector de Stokes para aplicaciones radiométricas está compuesto por los cuatro parámetros de Stokes, y adopta la siguiente forma:

$$\vec{S} = \begin{bmatrix} |E_h|^2 \\ |E_v|^2 \\ 2\text{Re}\langle E_v \cdot E_h^* \rangle \\ 2\text{Im}\langle E_v \cdot E_h^* \rangle \end{bmatrix} \left[\frac{W}{\text{m}^2 \cdot \text{Hz}} \right] \quad (2.16)$$

También es posible expresar los cuatro parámetros de Stokes como temperaturas de brillo a través de la siguiente expresión:

$$\overline{T}_B = \begin{bmatrix} T_h \\ T_v \\ T_U \\ T_V \end{bmatrix} = \frac{\lambda^2}{k \cdot z} \cdot \begin{bmatrix} |E_h|^2 \\ |E_v|^2 \\ 2\text{Re}\langle E_v \cdot E_h^* \rangle \\ 2\text{Im}\langle E_v \cdot E_h^* \rangle \end{bmatrix} \quad [K] \quad (2.17)$$

Donde z representa la impedancia del medio por el cual se propaga la onda, λ es la longitud de onda a la frecuencia central a la que se llevan a cabo las medidas y k es la constante de Boltzman. A partir de la ecuación 2.17 es posible demostrar que el vector de Stokes admite una tercera expresión:

$$\overline{T}_B = \begin{bmatrix} T_h \\ T_v \\ T_U \\ T_V \end{bmatrix} = \begin{bmatrix} T_h \\ T_h \\ T_{45^\circ} - T_{-45^\circ} \\ T_L - T_R \end{bmatrix} \quad [K] \quad (2.18)$$

Donde T_{45° y T_{-45° representan las temperaturas de brillo de las componentes ortogonales obtenidas a partir de una polarización lineal en un sistema de cordenadas rotado $\pm 45^\circ$ respecto de las polarizaciones vertical y horizontal mientras que T_L y T_R representan las dos temperaturas de brillo obtenidas al descomponer el campo incidente según una polarización circular a izquierdas y a derechas respectivamente.

2.3 TIPOS DE RADIÓMETROS

A continuación se realizará una breve descripción de los tipos de radiómetros más comúnmente empleados que puede resultar esclarecedora a la hora de presentar la topología de PAU-RAD.

2.3.1 Radiómetro de potencia total (TPR)

La figura 2-4 muestra el diagrama de bloques simple de un radiómetro de potencia total o TPR (Total Power Radiometer). Como se puede observar, el TPR es, en esencia, un receptor superheterodino de ancho de banda B y ganancia G conectado directamente a una antena. El TPR se puede descomponer en dos etapas fundamentales: una etapa de pre-detección y una etapa de detección de potencia.

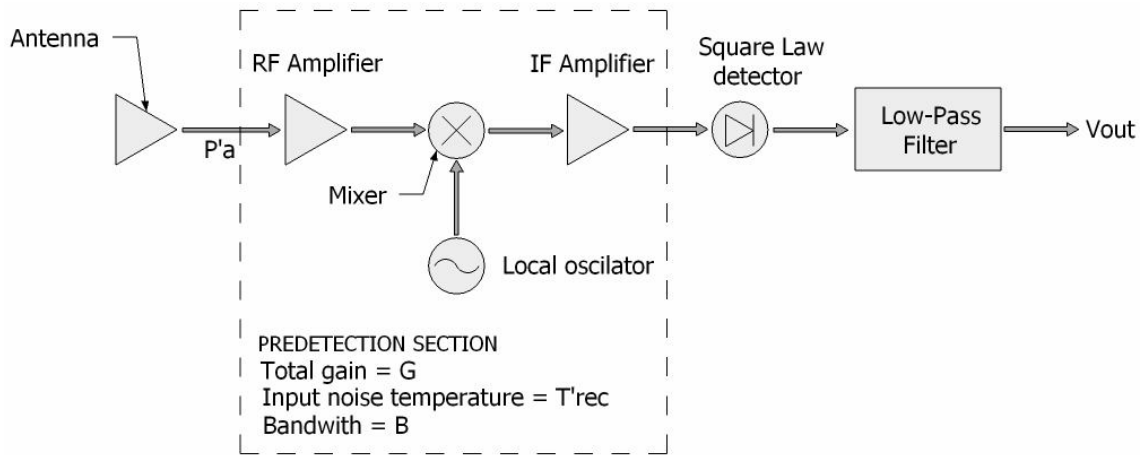


Figura 2-4: Diagrama de bloques de un radiómetro de potencia total [10].

La etapa de pre-detección está formada por un amplificador de radio-frecuencia, cuya función es filtrar la señal de entrada (ruido térmico blanco de banda ancha), amplificando la banda frecuencial del receptor (B); un mezclador que realiza la conversión de la señal de radio-frecuencia y ancho de banda B a una de frecuencia intermedia con el mismo ancho de banda y finalmente un amplificador de frecuencia-intermedia. Por su parte, la etapa de detección de potencia se compone de un detector de potencia (formado por diodo cuadrático) y de un filtro paso-bajo que actúa como integrador.

Es posible obtener un modelo equivalente del TPR sustituyendo la antena por una fuente de ruido con una potencia $P'_A = k \cdot B \cdot T'_A$ y el receptor por un receptor ideal sin ruido y con una entrada de ruido con una potencia $P'_{REC} = k \cdot B \cdot T'_{REC}$, donde T'_{REC} es la temperatura equivalente del ruido introducido por el conjunto de la línea de transmisión que une una antena con receptor y el propio receptor. De esta forma es posible obtener la potencia total de ruido en el sistema mediante la expresión:

$$P_{SYS} = P'_A + P'_{REC} = k \cdot T_{SYS} \cdot B = k(T'_A + T'_{REC}) \cdot B \quad [W] \quad (2.19)$$

Puesto que a la entrada del sistema se tiene ruido térmico, la tensión instantánea a la salida del amplificador de frecuencia intermedia sigue una distribución de probabilidad Gausiana de media cero y desviación típica σ , cuya envolvente obedece a una distribución de Rayleigh del tipo:

$$p(V_e) = \begin{cases} \frac{V_e}{\sigma^2} e^{-\frac{V_e^2}{2\sigma^2}}, & V_e \geq 0 \\ 0, & V_e < 0 \end{cases} \quad (2.20)$$

Por tanto, el valor medio de V_e coincidirá la potencia disponible a la salida del amplificador de frecuencia intermedia sobre una resistencia unitaria, es decir:

$$P_{FI} = \overline{V_e^2} = 2\sigma^2 \quad [W] \quad (2.21)$$

Si se relaciona la salida del diodo cuadrático (V_d) con la entrada, se tiene que:

$$V_d = C_d V_e^2 \quad [V] \quad (2.22)$$

Donde C_d representa la constante de sensibilidad de potencia del detector, con unidades de conversión $\left[\frac{V}{W}\right]$. El valor medio de V_d se calcula como:

$$\overline{V_d} = C_d \overline{V_e^2} = 2C_d \sigma^2 = C_d P_{FI} = C_d GkBT_{SYS} \quad [V] \quad (2.23)$$

Por otra parte, la tensión a la salida del filtro paso bajo consta de dos componentes: una componente continua $\overline{V_{out}} = \overline{V_d}$; y una componente alterna $V_{ac}(t)$. La componente $V_{ac}(t)$ corresponde a la desviación típica de V_d (σ_d) y representa la incertidumbre estadística asociada a la media de P_{SYS} . La componente continua se relaciona directamente con la potencia de entrada P_{SYS} , y por tanto con la temperatura radiométrica T_{SYS} a través de la ecuación 2.24:

$$\overline{V_{out}} = G_{LPF} \overline{V_d} \quad [V] \quad (2.24)$$

Donde G_{LPF} constituye la ganancia del filtro paso-bajo. Por tanto, la tensión a la salida de un TPR es proporcional a la temperatura radiométrica y su valor viene determinado por la siguiente expresión:

$$V_{out}(t) = \overline{V_{out}} + V_{ac}(t) = G_{LPF} C_d GkBT_{SYS} + V_{ac}(t) = G_S T_{SYS} + V_{ac}(t) \quad [V] \quad (2.25)$$

Para una distribución de Rayleigh, se cumple que el valor medio al cuadrado coincide con la varianza, lo que significa que la desviación típica y el valor medio de la salida del diodo cuadrático son iguales, es decir:

$$\frac{\sigma_d}{\overline{V_d}} = 1 \rightarrow \sigma_d = \overline{V_d} \quad [V] \quad (2.26)$$

Esto significa que la incertidumbre sobre la medida será del mismo orden de magnitud que lo que se pretende medir, lo que invalidaría por completo las medidas. La función principal del filtro paso-bajo es evitar este inconveniente actuando como integrador que promedia el valor de V_d durante un intervalo de tiempo τ (que se corresponde con la constante de tiempo del filtro) de manera que la varianza de la medida se reduzca en un factor $N = B\tau$, donde N es el número de muestras

independientes utilizadas en el promediado. Por tanto, la relación entre la desviación típica y el valor medio de la medida a la salida del filtro adoptan la siguiente forma:

$$\frac{\sigma_{out}}{V_{out}} = \frac{1}{\sqrt{B\tau}} \rightarrow \sigma_{out} = \frac{\overline{V_{out}}}{\sqrt{B\tau}} \quad (2.27)$$

Si se supone que todos los parámetros de la ecuación 2.23 son constantes, la relación anterior puede reescribirse en función de la desviación típica asociada al valor medio de T_{SYS} (ΔT_{SYS}), es decir:

$$\frac{\Delta T_{SYS}}{T_{SYS}} = \frac{1}{\sqrt{B\tau}} \quad (2.28)$$

A partir del resultado anterior es posible determinar la sensibilidad o resolución radiométrica del radiómetro (ΔT), que se define como la mínima variación de temperatura que es capaz de detectar el radiómetro o lo que es lo mismo, la sensibilidad constituye el menor cambio posible de temperatura a la entrada que provoca un cambio de tensión a la salida del receptor. La sensibilidad radiométrica para un TPR viene dada por:

$$\Delta T_N \equiv \Delta T_{SYS} = \frac{T_{SYS}}{\sqrt{B\tau}} = \frac{T'_A + T'_{REC}}{\sqrt{B\tau}} \quad [K] \quad (2.29)$$

No obstante, esta última expresión únicamente tiene en cuenta las fluctuaciones del ruido, de manera que se obvian las posibles fluctuaciones de la ganancia del receptor, que se producen en un caso real y que llevan asociada una incertidumbre tal que:

$$\Delta T_G = T_{SYS} \left(\frac{\Delta G_S}{G_S} \right) \quad [K] \quad (2.30)$$

Donde G_S es la ganancia total de la cadena receptora y ΔG_S es la variación RMS (Root Mean Square) de la potencia detectada. Teniendo en cuenta que las fluctuaciones debidas al ruido son estadísticamente independientes de las de la ganancia y por tanto incorreladas, la sensibilidad total del sistema se puede expresar como:

$$\Delta T = [(\Delta T_N)^2 + (\Delta T_G)^2]^{\frac{1}{2}} = T_{SYS} \left[\frac{1}{B\tau} + \left(\frac{\Delta G_S}{G_S} \right)^2 \right]^{\frac{1}{2}} \quad [K] \quad (2.31)$$

De la ecuación anterior se desprende que la sensibilidad de un TPR se verá fuertemente influenciada por las posibles fluctuaciones de ganancia que pueda sufrir el radiómetro durante el proceso de medida.

2.3.2 Radiómetro de Dicke

El radiómetro de Dicke constituye una variación del TPR cuya finalidad es la de eliminar la dependencia que tiene la sensibilidad de éste último con las variaciones de ganancia en la cadena de recepción.

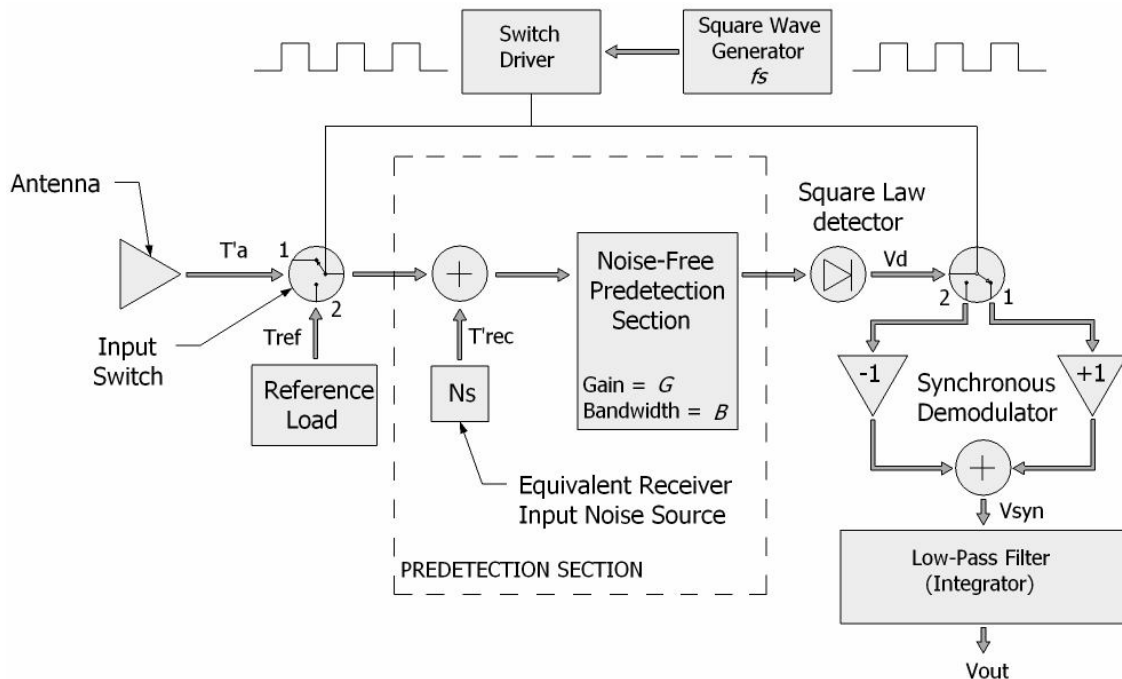


Figura 2-5: Diagrama de bloques simple de un radiómetro de Dicke [10].

En la figura 2-5, que muestra el diagrama de bloques de un radiómetro de Dicke, se puede observar como la estructura de éste es esencialmente la de un TPR pero con las siguientes adiciones:

- Un selector de entrada, que permite conmutar entre la antena y una carga adaptada que actúa como fuente de ruido de referencia.
- Un demodulador síncrono a la salida del detector cuadrático, formado por un selector i dos amplificadores de ganancia unitaria y signo opuesto.

- Un generador de onda cuadrada a la frecuencia de muestreo que mantiene sincronizados los dos selectores anteriores.

El selector de entrada, o conmutador de Dicke, permite llevar a cabo una modulación de la señal de entrada conmutando periódicamente la entrada del receptor entre la antena durante el primer semiperiodo y la fuente de ruido durante el segundo. Para eliminar la dependencia de la sensibilidad con la ganancia, es necesario que ésta última se mantenga constante durante un periodo. Por tanto, la frecuencia de muestreo, f_s , tiene que ser mayor que la máxima frecuencia de las fluctuaciones de ganancia.

Las tensiones detectadas a la salida del diodo cuadrático para un periodo completo, obedecen a las siguientes expresiones:

$$\begin{aligned}\overline{V_{dANT}} &= C_d G k B (T'_A + T'_{REC}) \quad \text{para } 0 \leq t \leq \frac{\tau_s}{2}, \quad [V] \\ & \\ \overline{V_{dREF}} &= C_d G k B (T_{REF} + T'_{REC}) \quad \text{para } \frac{\tau_s}{2} \leq t \leq \tau_s. \quad [V]\end{aligned}\tag{2.32}$$

Donde T_{REF} es la temperatura de ruido de la fuente de referencia, τ_s es el periodo de muestreo y T'_{REC} es la temperatura de ruido del receptor incluyendo la contribución de ruido del selector de entrada.

Por su parte el demodulador síncrono consta de un segundo conmutador que opera de manera sincronizada con el conmutador de Dicke. Éste conmutador, alimenta dos amplificadores de ganancia unitaria y signos opuestos, de manera que uno de ellos recibe $\overline{V_{dANT}}$ y el otro $\overline{V_{dREF}}$. Las salidas de los amplificadores se suman y pasan a un filtro que actúa como integrador, dando lugar a la señal:

$$V_{SYN} = \frac{1}{2} (\overline{V_{dANT}} - \overline{V_{dREF}}) = \frac{1}{2} C_d G k B (T'_A - T'_{REF}) \quad [V] \tag{2.33}$$

Como se puede observar, la tensión de salida de un radiómetro de Dicke es proporcional al término $T'_A - T'_{REF}$. Esto supone que, a las incertidumbres debidas a las variaciones de la temperatura de antena y de la ganancia, haya que añadir la incertidumbre en la medida de la temperatura de referencia. Por tanto, la sensibilidad de un radiómetro de Dicke se obtiene a través de la expresión:

$$\Delta T = \left[\frac{(T'_A + T'_{REC})^2}{\frac{B\tau}{2}} + \frac{(T_{REF} + T'_{REC})^2}{\frac{B\tau}{2}} + \left(\frac{\Delta G_S}{G_S} \right) (T'_A - T_{REF})^2 \right]^{\frac{1}{2}} [K] \tag{2.34}$$

Si particularizamos para el caso ideal en que la temperatura de antena sea igual a la de la fuente de referencia $T'_A = T_{REF}$, obtenemos lo que se denomina: radiómetro de Dicke balanceado o equilibrado, cuya expresión de la sensibilidad queda reducida a:

$$\Delta T = \frac{2(T'_A + T'_{REC})}{\sqrt{B\tau}} = 2\Delta T_{IDEAL}. \quad [K] \quad (2.35)$$

Donde ΔT_{IDEAL} , representa la sensibilidad de un TPR en el caso de ausencia total de fluctuaciones de ganancia. Como se puede observar a partir de los resultados de la ecuación anterior, la dependencia con las fluctuaciones de la ganancia ha desaparecido. Como contrapartida, la sensibilidad del radiómetro ha empeorado en un factor 2, debido a que el tiempo de observación de antena se ha reducido a la mitad.

2.3.3 Radiómetro de inyección de ruido (NIR)

El radiómetro de inyección de ruido constituye un caso particular del radiómetro de Dicke, optimizado para que su salida sea independiente tanto de las fluctuaciones de ganancia como de la temperatura de ruido del propio receptor. Para conseguir esto, la estructura de un NIR se basa en la red de realimentación que se muestra en la figura 2-6.

La finalidad de la red de realimentación es la de balancear el radiómetro (al igual como ocurriera en el radiómetro de Dicke balanceado) mediante la inyección de ruido en la línea de antena a través de un acoplador direccional, de tal manera que:

$$T''_a = T_{REF} = 0 \quad [K] \quad (2.36)$$

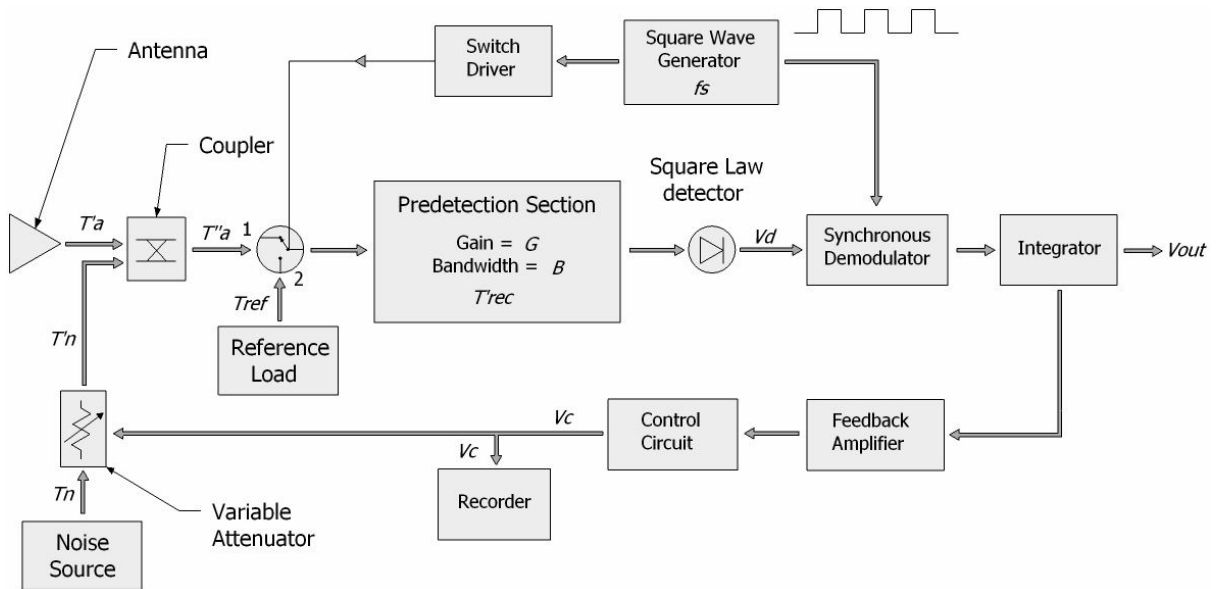


Figura 2-6: Diagrama de bloques simple de un radiómetro de inyección de ruido [10].

La cantidad de potencia inyectada se controla mediante un atenuador variable, que a su vez está gobernado por la red de realimentación, de manera que la temperatura de ruido resultante que entra al sistema es:

$$T''_a = \left(1 - \frac{1}{F_c}\right) T'_a + \frac{T'_N}{F_c}. \quad [K] \quad (2.37)$$

Donde F_c representa el factor de acoplamiento del acoplador direccional y T'_N es la temperatura de ruido inyectada una vez atenuada en el atenuador variable.

La tensión V_c , que controla el valor de atenuación del atenuador variable es proporcional a la diferencia de temperaturas entre la temperatura de antena (T'_a) y la temperatura física del radiómetro, que se aproxima por la temperatura ambiente ($T_0 \cong 290^\circ K$):

$$V_c = \frac{F_c - 1}{T_N - T_0} (T_0 - T'_a). \quad [V] \quad (2.38)$$

Mediante la aplicación de esta tensión de control al atenuador variable, se consigue que la salida del radiómetro sea independiente de las fluctuaciones de ganancia así como del ruido introducido por el propio receptor. La sensibilidad radiométrica para un NIR en el caso de que se cumpla que $T'_a = T_0$, es:

$$\Delta T = \frac{2(T_0 + T'_{REF})}{\sqrt{B\tau}} = 2\Delta T_{IDEAL}. \quad [K] \quad (2.39)$$

Que como se puede observar, coincide con la de un radiómetro de Dicke balanceado, pero con la ventaja de no depender de la temperatura del receptor.

2.3.4 Radiómetros polarimétricos

Tal y como ya se ha comentado en el apartado 2.2.6 de este mismo capítulo, de manera habitual, las medidas radiométricas se proporcionan en términos de los cuatro parámetros de Stokes. Los radiómetros polarimétricos son aquellos destinados a proporcionar a su salida el valor de los cuatro parámetros de Stokes. Su implementación se basa en la del radiómetro de Dicke, con la particularidad de que la antena posee separadas las polarizaciones horizontal y vertical. A continuación se llevará a cabo una breve descripción de algunos de los tipos fundamentales de radiómetros polarimétricos.

2.3.4.1 Radiómetro de combinación secuencial de polarizaciones

Recibe este nombre debido a que los cuatro parámetros de Stokes se obtienen de forma secuencial a partir de las medidas de las polarizaciones vertical y horizontal proporcionadas por la antena.

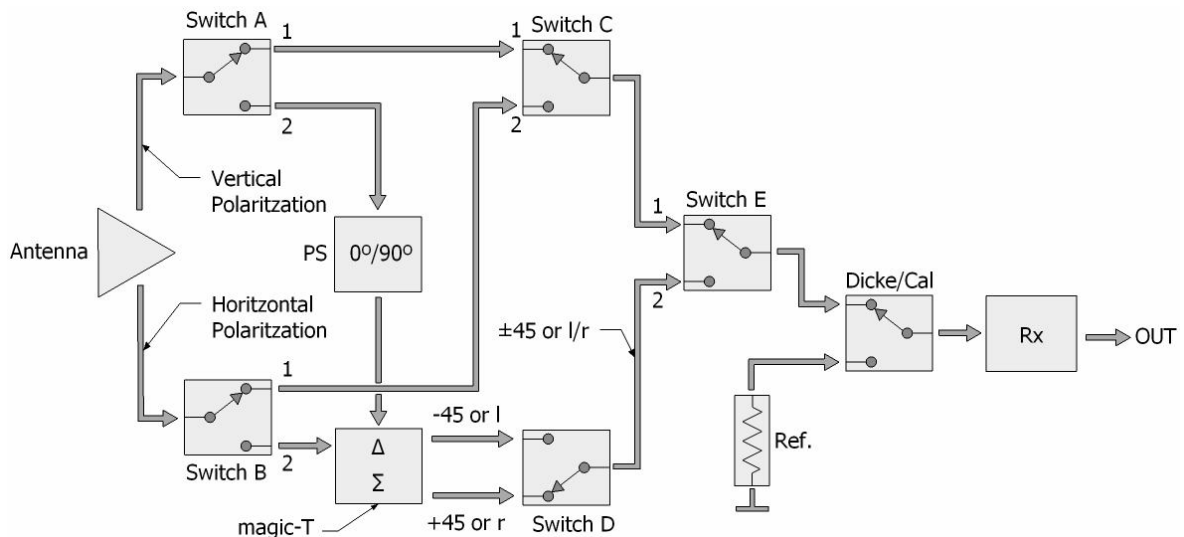


Figura 2-7: Diagrama de bloques simple de un radiómetro polarimétrico de combinación secuencial de polarizaciones [11].

La figura 2-7 muestra el diagrama de bloques para este tipo de radiómetro. Los cuatro parámetros de Stokes se obtienen a través de la combinación de los 6 conmutadores con un rotador de fase y una T-mágica. De esta forma, combinando las dos polarizaciones en la T-mágica con el rotador fijado a 0° se obtienen las polarizaciones de $\pm 45^\circ$ que se emplearán para formar el tercer y cuarto parámetro de Stokes. Si se configura el rotador para un desfase de 90° , a la salida de la T-mágica se obtienen polarizaciones circulares a izquierda y derecha, mediante las cuales se obtendrán el primer y segundo parámetro.

La principal ventaja de esta implementación es el hecho de requerir únicamente una etapa demoduladora (Rx en la figura), mientras que el inconveniente fundamental son las pérdidas introducidas por los conmutadores, que derivan en una sensibilidad radiométrica baja debida al bajo rendimiento por ciclo y al ruido añadido por los diferentes componentes.

2.3.4.2 Radiómetro de combinación paralela de polarizaciones

Este tipo de radiómetro se basa en la implementación descrita para el tipo anterior, tal y como se puede observar en la figura 2-8, que muestra el diagrama de bloques. No obstante, en esta ocasión, los parámetros de Stokes se obtienen de forma paralela gracias a la utilización de 6 receptores.

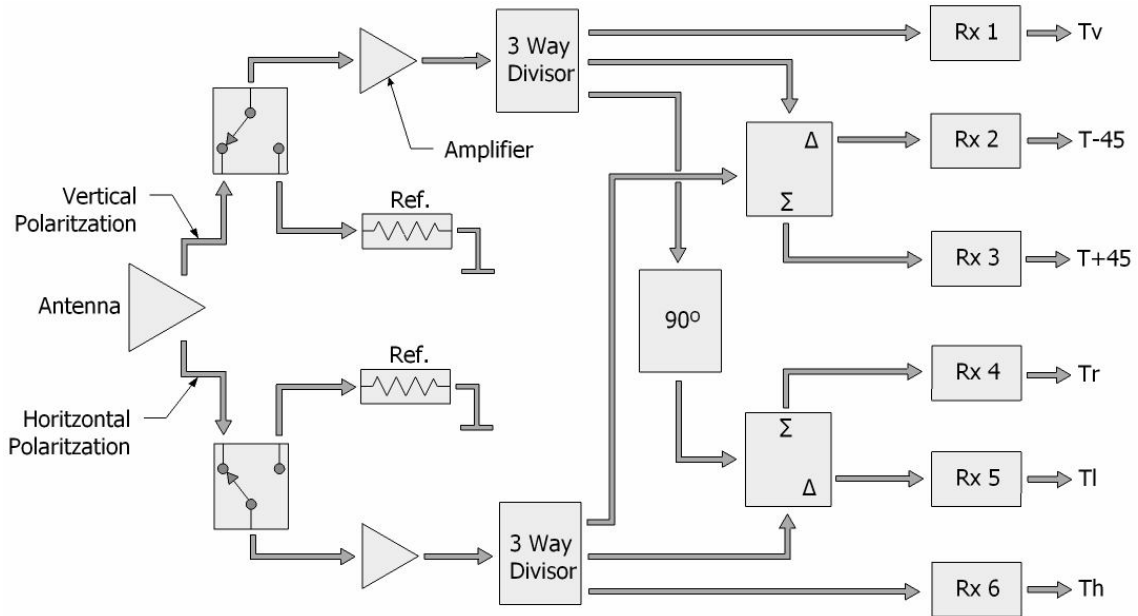


Figura 2-8: Diagrama de bloques simple de un radiómetro polarimétrico de combinación paralela de polarizaciones [11].

Mediante esta implementación se consigue reducir las pérdidas debidas a los conmutadores, mejorando por tanto la sensibilidad del radiómetro de combinación secuencial. El inconveniente fundamental de esta aproximación es que requiere seis receptores, con el consiguiente incremento de los costes así como de la complejidad del sistema.

2.3.4.3 Radiómetro de correlación elemental

La figura 2-9 muestra el diagrama de bloques de este tipo de radiómetro. Como se puede observar consta de dos receptores coherentes entre si, que proporcionan a su salida T_h y T_v , y de un correlador complejo que multiplica las componentes en fase y cuadratura de las señales de entrada y las integra para proporcionar el tercer y cuarto parámetro de Stokes.

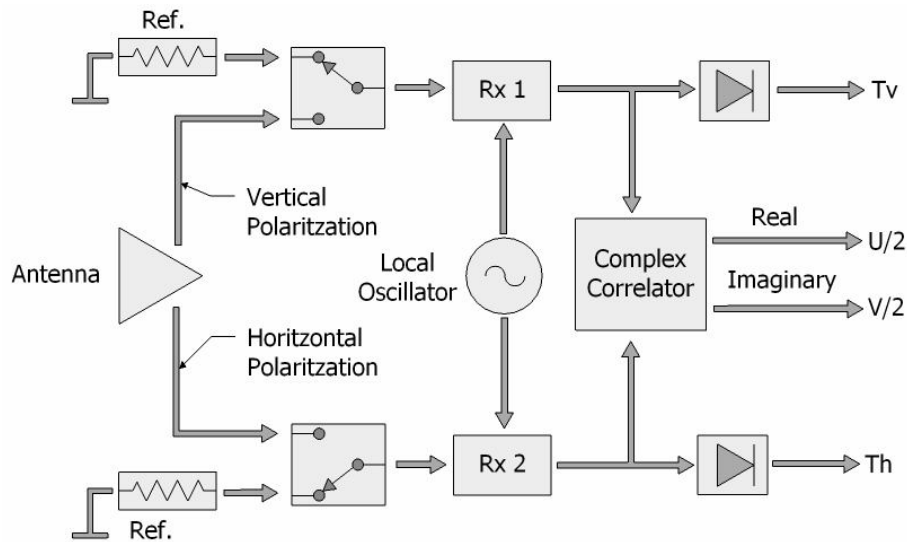


Figura 2-9: Diagrama de bloques simple de un radiómetro de correlación elemental.

Esta solución constituye un compromiso entre las dos aproximaciones anteriores, ya que presenta una buena sensibilidad radiométrica y requiere tan sólo del uso de dos receptores. Sin embargo, es necesario tener en cuenta que la precisión del tercer y cuarto parámetro de Stokes dependerá de la calidad del correlador complejo empleado.

2.3.4.4 Radiómetro de correlación conmutado

El radiómetro de correlación conmutada presenta una estructura similar a la descrita para el radiómetro de correlación elemental, tal y como se puede observar en la figura 2-10.

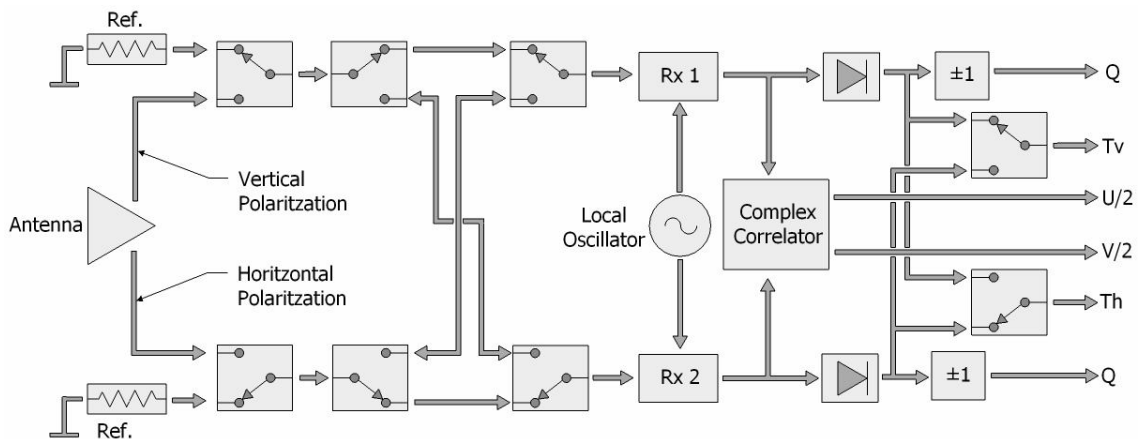


Figura 2-10: Diagrama de bloques simple de un radiómetro polarimétrico de correlación conmutada.

Gracias a la incorporación al sistema de un conmutador, se distribuyen las señales de ambas polarizaciones de manera alternativa, es decir, durante el primer semiperiodo cada receptor recibe una de las dos polarizaciones, mientras que durante el segundo semiperiodo recibe la polarización ortogonal. Mediante esta estructura se consigue aumentar la sensibilidad en un factor $\sqrt{2}$, ya que las dos salidas Q presentan la sensibilidad de un radiómetro de Dicke, y si se suponen estadísticamente independientes es posible combinarlas. La contrapartida es la complejidad que representa el control de la conmutación.

2.3.4.5 Radiómetro de pseudo-correlación

Este tipo de radiómetro es el implementado en PAU-RAD. En este apartado tan solo se enunciará el principio básico de funcionamiento de un radiómetro de pseudo-correlación. En el siguiente capítulo se abordará con más detalle la estructura física de la implementación final del radiómetro de PAU. En cualquier caso se puede hallar una descripción exhaustiva de dicha implementación en [2] y en [12].

El radiómetro de pseudo-correlación, tal y como se puede apreciar en la figura 2-11, se divide en dos ramas. Cada una de estas ramas se compone de tres elementos básicos: un divisor de Wilkinson, dos amplificadores (uno para cada cadena) y un correlador. El divisor de Wilkinson constituye una modificación al esquema teórico de este tipo de radiómetro, que originalmente utiliza un híbrido de 180° , ya que la longitud de onda de trabajo de PAU hace que las dimensiones del mismo sean inviables. A la salida de los divisores de cada una de las ramas se obtienen las señales:

$$S_{1H,V} = \frac{S_{AH,V} + S_R}{\sqrt{2}} \quad S_{2H,V} = \frac{S_{AH,V} - S_R}{\sqrt{2}} \quad (2.40)$$

Donde $S_{AH,V}$ es la señal de antena para cada una de las dos polarizaciones y S_R es el ruido introducido por la resistencia interna del divisor.

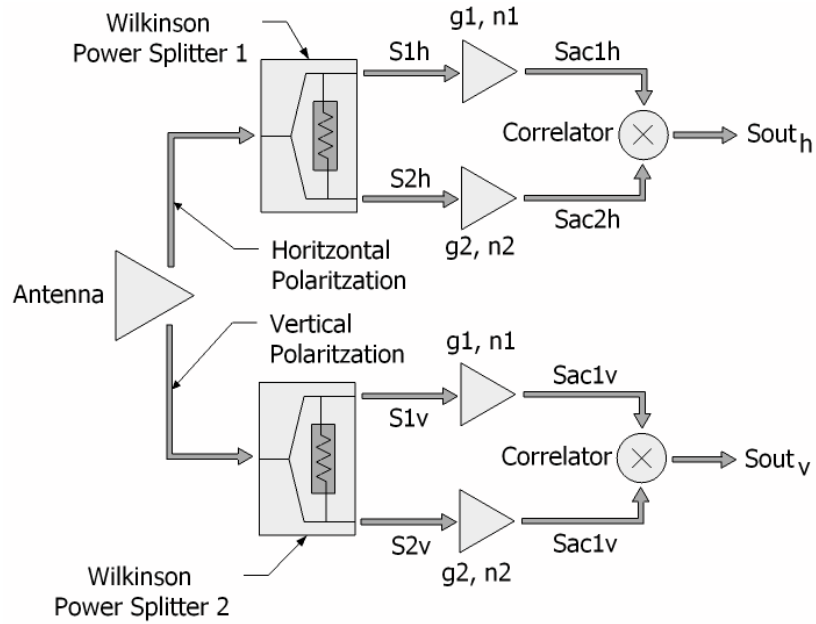


Figura 2-11: Diagrama de bloques simple de un radiómetro de pseudo-correlación.

Las señales a la salida de los divisores pasan por los amplificadores que se encargan de acondicionar y filtrar, de forma que a su salida se obtienen las siguientes señales:

$$S_{ac1_{H,V}} = \left(\frac{S_{A_{H,V}} + S_R}{\sqrt{2}} + n_1 \right) \cdot \sqrt{g_1} \tag{2.41}$$

$$S_{ac2_{H,V}} = \left(\frac{S_{A_{H,V}} - S_R}{\sqrt{2}} + n_2 \right) \cdot \sqrt{g_2}$$

Finalmente, estas señales entran en un multiplicador que hace las veces de correlador, de ahí que se denomine radiómetro de pseudo-correlación. A la salida de cada uno de los multiplicadores se obtiene:

$$S_{OUT_{H,V}} = \frac{(S_{A_{H,V}})^2 - (S_R)^2}{2} \cdot \sqrt{g_1 g_2} \tag{2.42}$$

Puesto que la señal de antena está incorrelada con respecto al ruido del divisor, los términos cruzados resultantes de la multiplicación son nulos. De esta forma, se obtiene a la salida la señal de antena para cada una de las polarizaciones, a partir de las cuales se extraen el primer y el segundo parámetro de Stokes. Para obtener el tercer y cuarto parámetro, basta con hacer los productos cruzados entre las señales de las dos polarizaciones.

2.4 APLICACIONES DE LA RADIOMETRÍA

En la tabla 2-1 se recogen algunas de las aplicaciones fundamentales de la radiometría en función de la frecuencia de observación. Como se puede comprobar todas ellas están destinadas a la medida de parámetros atmosféricos o de la superficie terrestre, que tienen múltiples aplicaciones fundamentalmente meteorológicas y medioambientales.

Tabla 2-1: Aplicaciones de la radiometría en función de la frecuencia de trabajo.

Aplicación	Frecuencia [GHz]
Contenido de agua en las nubes	21, 37, 90
Clasificación de hielo	10, 18, 37
Seguimiento de manchas de petróleo en el océano	6.6, 37
Lluvia sobre el suelo	18, 37, 55, 90, 180
Lluvia sobre el mar	10, 18, 21, 37
Concentración de hielo en el mar	18, 37, 90
Salinidad superficial marina	1.4, 6.6
Temperatura superficial marina	6.6, 10, 18, 21, 37
Velocidad del viento en la superficie marina	10, 18
Cobertura de nieve	6.6, 10, 18, 37, 90
Humedad del suelo	1.4, 6.6
Perfiles de temperatura	21, 37, 55, 90, 180
Vapor de agua	21, 37, 90, 180

2.4.1 Justificación al uso de frecuencias GPS en radiometría.

Tal y como se verá en el siguiente capítulo, la frecuencia de trabajo de PAU-RAD es de 1.57542 GHz, que se corresponde con la banda L1 de GPS (Global Positioning System). A priori la elección de esta frecuencia de trabajo puede resultar desconcertante, ya que lo que se pretende medir es esencialmente ruido, y la presencia en la banda frecuencial de medida de señales procedentes de otros sistemas puede causar interferencias.

La modulación empleada por el sistema GPS es de espectro ensanchado, lo que significa que las señales GPS únicamente pueden ser detectadas mediante la correlación de la señal recibida con un código. Por tanto la presencia de señales GPS en la banda de medida no afecta a las medidas radiométricas, y la utilización de esta frecuencia permite que tanto el radiómetro como el reflectómetro compartan un mismo cabezal de RF, simplificando el diseño.

3 DESCRIPCIÓN DE PAU-RAD

3.1 INTRODUCCIÓN

Tal y como se ha comentado en el capítulo 1, PAU-RAD es uno de los tres subsistemas que componen PAU. Se trata de la etapa de procesado digital de señal de un radiómetro polarimétrico con conformación de haz digital, que tiene por misión procesar los datos procedentes de los cuatro canales de cada uno de los 16 receptores que compondrán el sistema completo, y corregir los errores de fase, ganancia y offset. De esta forma, una vez corregidos los errores, será posible llevar a cabo la conformación del haz.

PAU-RAD es en esencia un radiómetro de pseudo-correlación, cuyo objetivo es obtener los 4 parámetros de Stokes, que proporcionan información sobre la temperatura de brillo de la superficie a la cual se enfoca (en este caso, la superficie del mar). Los resultados obtenidos, combinados debidamente con los resultados proporcionados por los otros dos subsistemas de PAU, permitirán la obtención de la salinidad superficial del mar o SSS (Sea Surface Salinity).

Tanto la parte analógica de PAU-RAD como la parte analógica de PAU-GNSS comparten un mismo circuito integrado receptor de GPS. Debido a esto, PAU-RAD trabaja a una frecuencia de 1.57542 GHz, que corresponde a la banda L1 de GPS. Las ventajas e inconvenientes de esta configuración ya han sido discutidas en el capítulo 2.

3.2 ESTRUCTURA DE PAU-RAD

A grandes rasgos, PAU-RAD se puede descomponer en dos etapas: una primera etapa analógica, encargada de la adquisición de los datos; y una segunda etapa digital encargada de procesar estos datos.

La etapa analógica está formada por una antena y un receptor analógico [3], mientras que la etapa digital la compone una FPGA (Field Gate Programmable Array) que implementa una unidad aritmético-lógica o ALU (Arithmetic-Logic Unit) [2] y un procesador embebido Nios que actúa como UC (Unidad de Control) [4]. La conexión entre ambas etapas se lleva a cabo mediante un conversor analógico digital o ADC (Analog to Digital Converter) [5]. La figura 3-1 muestra un esquema que ilustra la interconexión de estos cuatro elementos que componen PAU-RAD.

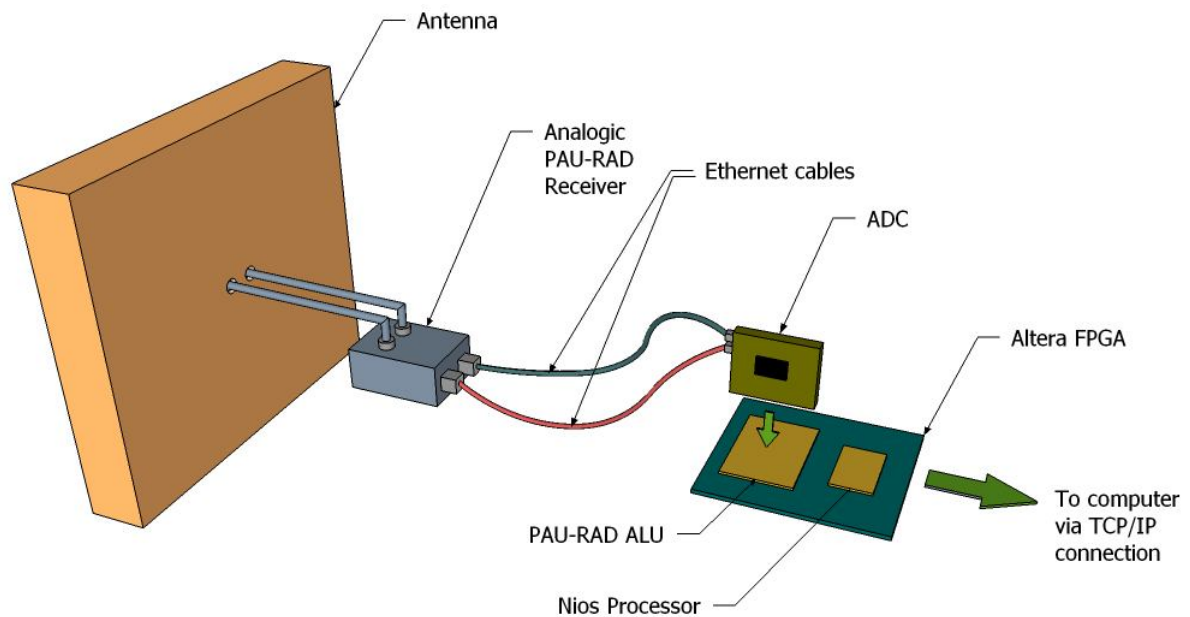


Figura 3-1: Esquema que ilustra la ubicación de los cuatro componentes principales del sistema PAU-RAD, para un único receptor.

3.2.1 Antena

La antena constituye el primer elemento de la estructura del radiómetro, y su función es la de captar el ruido térmico de la superficie a la que enfoca. Dado que la magnitud que se quiere captar presenta una potencia muy reducida, es necesario que el lóbulo principal del diagrama de radiación de esta antena sea lo más estrecho posible. Esto se traduce en un valor de directividad elevado, que evitará que se capte a través de los lóbulos secundarios del diagrama de radiación potencia procedente de otras direcciones diferentes a la dirección de observación, cosa que invalidaría las medidas.

Para conseguir los valores de directividad en banda L requeridos para esta aplicación, se podría implementar una antena de grandes dimensiones. Esta solución presenta sin embargo dos inconvenientes importantes que hacen que sea inviable: en primer lugar una antena de grandes dimensiones implica una movilidad reducida y limita el número de ubicaciones en las que poder llevar a cabo campañas de medida; en segundo lugar, al tratarse de una superficie grande, resulta complicado llevar a cabo un control térmico adecuado, con lo que es más difícil controlar la cantidad de ruido térmico añadido por la antena. A todo esto cabe añadir las complicaciones mecánicas propias de una antena de estas características.

Otra posible solución consiste en la formación de una agrupación de antenas básicas con diferentes alimentaciones, de manera que el diagrama de radiación resultante se adapte a las características requeridas. Esta solución tiene la ventaja de

haber sido implementada satisfactoriamente en el radiómetro LAURA (L-band AUTomatic RAdiometer).

3.2.1.1 Antena básica

La antena básica elegida para la formación de la agrupación es la misma que la empleada en el radiómetro LAURA, pero adaptada a la frecuencia de trabajo de PAU es decir: 1575.42 MHz (la banda L1 de GPS), ya que la frecuencia de trabajo de LAURA es de 1400 MHz.

Se trata de una antena de tipo parche, es decir: una impresión radiante sobre un sustrato dieléctrico (fibra de vidrio) colocada sobre un plano de masa a cierta distancia. Este tipo de antenas presentan un coste de producción reducido debido a su sencillez.

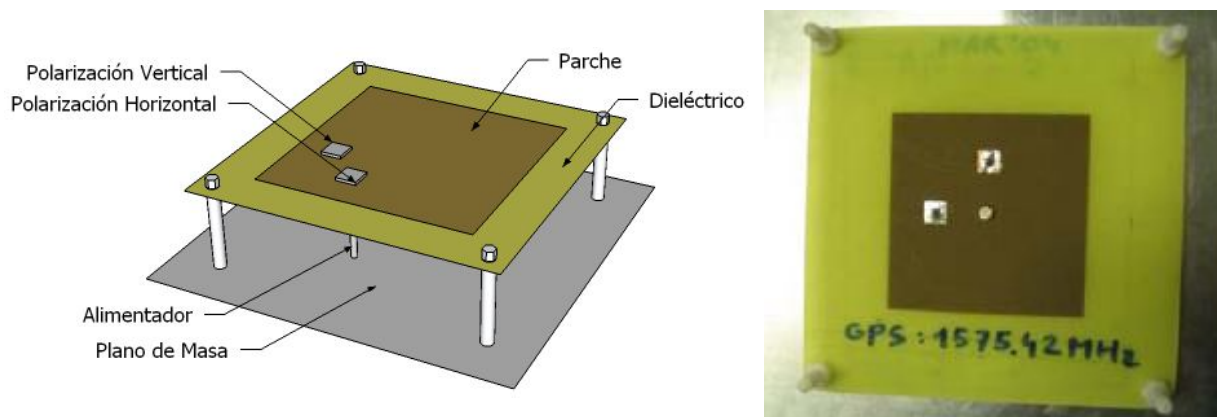


Figura 3-2: A la izquierda, diagrama de los componentes de la antena básica; a la derecha, fotografía de una de las antenas básicas empleadas.

Puesto que PAU-RAD es un radiómetro de tipo polarimétrico, esta antena básica es de doble polarización (vertical y horizontal). La figura 3-3 muestra los diagramas de radiación en los planos E y H.

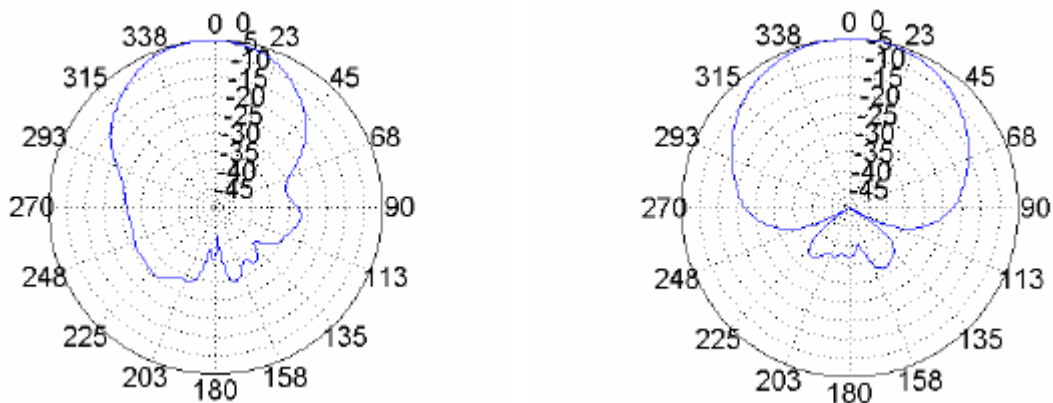


Figura 3-3: Diagramas de radiación de la antena unitaria de PAU-RAD. Izquierda: diagrama de radiación en el plano E; derecha: diagrama de radiación en el plano H.

A parte de las características mencionadas, la antena básica de PAU-RAD presenta una buena adaptación a la entrada, lo que permite aprovechar al máximo la cantidad de potencia incidente.

3.2.1.2 Agrupación

La antena completa de PAU-RAD, es una agrupación de 16 antenas básicas dispuestas en una estructura cuadrada de 4x4 elementos. Las 16 antenas básicas están ubicadas de forma que la antena resultante presenta simetría a lo largo de sus dos ejes principales, tal y como se puede apreciar en la figura 3-4:

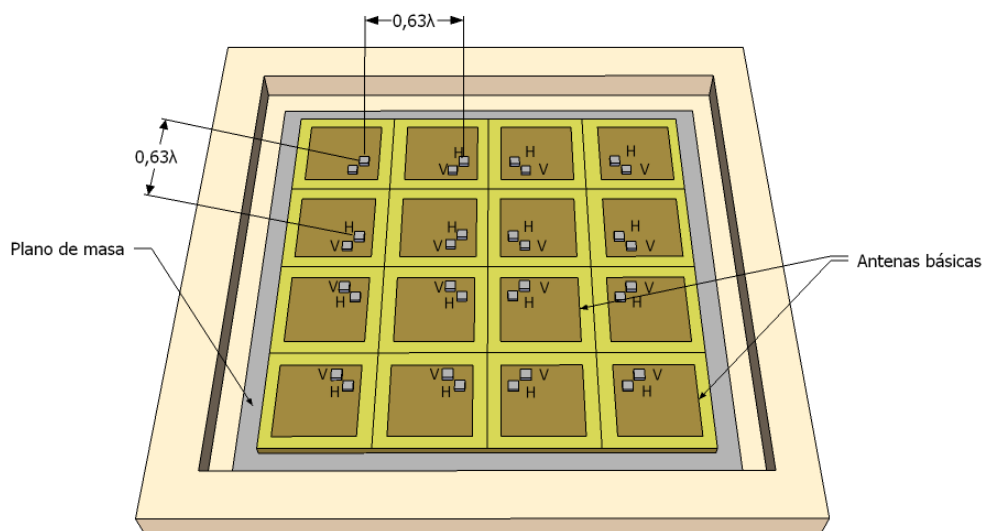


Figura 3-4: Distribución de las antenas básicas que conforma la agrupación.

Gracias a esta distribución se consigue cancelar la componente cross-polar en los planos principales del diagrama de radiación resultante. Esto es de vital importancia, ya que al tratarse de un radiómetro polarimétrico es necesario separar bien ambas polarizaciones para que los parámetros de Stokes obtenidos sean lo más precisos posible.

La iluminación que se emplea en la conformación del haz es de tipo triangular de pesos 1:2:2:1, tanto para las filas como para las columnas, lo que permite reducir el Nivel de Lóbulo Principal a Secundario (NLPS) hasta situarlo en un nivel menor o igual a -20 dB.

La separación de $0,63\lambda$ entre antenas básicas es el resultado de un estudio teórico realizado en [2] y constituye un punto de equilibrio en el compromiso entre reducción del ancho de haz a -3 dB y maximización de la eficiencia del lóbulo principal o MBE (Main Beam Efficiency).

Todas estas características hacen que la antena resultante cumpla los requisitos necesarios para ser empleada en un sistema de medida radiométrico.

3.2.2 Receptor

El diseño y la implementación del receptor de PAU-RAD escapan al ámbito del presente proyecto. Una descripción exhaustiva tanto del diseño como de la implementación y funcionamiento del mismo se pueden encontrar en [3]. No obstante en este apartado se describirán ciertas características del receptor que son imprescindibles para la correcta comprensión del funcionamiento global de PAU.

El receptor de PAU-RAD es un receptor polarimétrico, esto implica que trabaja tanto con la polarización horizontal como con la vertical de la antena y que, por tanto, presentará una topología separada en dos ramas. Además de ser polarimétrico, es un radiómetro de pseudo-correlación, por tanto, cada una de estas dos ramas se bifurcará a su vez en dos canales, dando lugar a los cuatro canales de salida del receptor.

Además de las dos entradas de antena (una para cada polarización) el receptor presenta una tercera entrada (entrada de calibración) destinada a conectar una señal de referencia, que en este caso vendrá proporcionada por una fuente de ruido gaussiano blanco. La función de esta entrada de calibración se abordará con detalle en el capítulo 5.

La función del receptor es la de captar las señales en banda L1 de GPS procedentes de la antena y trasladarlas en frecuencia mediante chips específicos de GPS (GP2015 de la marca Zarlink) a una frecuencia intermedia centrada en 4,3 MHz, con un ancho de banda de 2,2 MHz. De esta forma, las señales pueden ser digitalizadas y enviadas a la etapa digital de PAU-RAD, donde son procesadas.

La estructura física del receptor se compone de dos etapas claramente diferenciadas: una etapa de radio-frecuencia, y una etapa de frecuencia intermedia. A continuación se describen brevemente la topología y funciones de ambas etapas.

3.2.2.1 Etapa Radio-Frecuencia

La etapa de Radio-Frecuencia (RF) de PAU-RAD obtiene las señales correspondientes a las polarizaciones horizontal y vertical (S_{aH} , S_{aV}) procedentes de la antena. Ambas señales se inyectan a sendos divisores de Wilkinson, dando como resultado las señales de cada uno de los cuatro canales:

$$\frac{[S_a + S_r]_H}{\sqrt{2}}, \frac{[S_a - S_r]_H}{\sqrt{2}}, \frac{[S_a + S_r]_V}{\sqrt{2}}, \frac{[S_a - S_r]_V}{\sqrt{2}} \quad (3.1)$$

Donde: S_a denota la señal procedente de la antena, S_r representa el ruido térmico añadido por la resistencia del divisor de Wilkinson por el hecho de encontrarse a una temperatura física diferente a cero Kelvin y los subíndices H y V indican polarización horizontal y vertical respectivamente .

Estas cuatro señales pasan a un bloque acondicionador de señal compuesto por un amplificador de bajo ruido (LNA), un filtro paso banda centrado en la frecuencia L1 de GPS y finalmente un amplificador que adecua los niveles de señal a los requerimientos de la siguiente etapa.

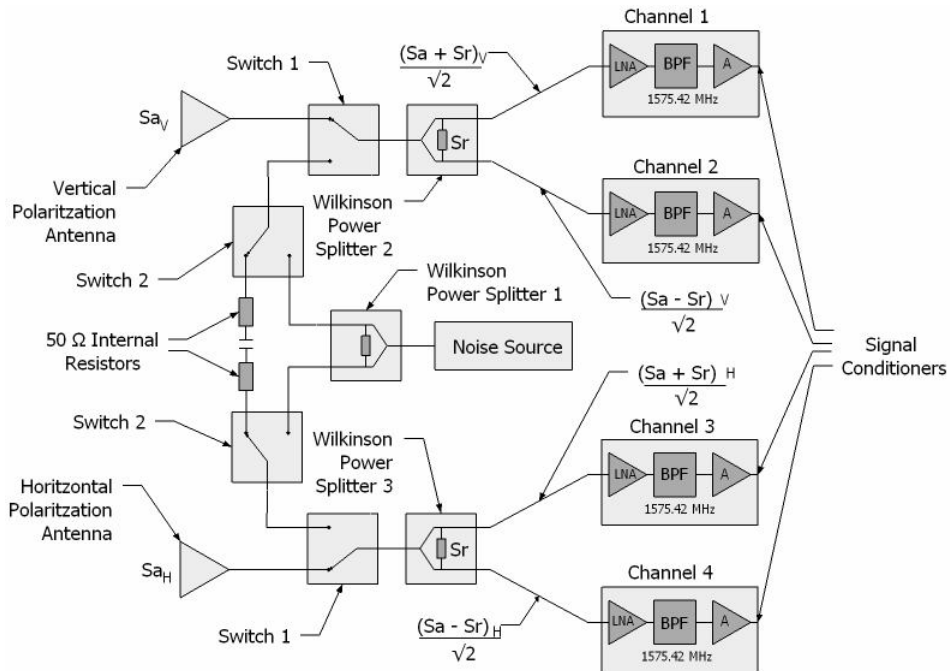


Figura 3-5: Diagrama de bloques de la etapa de RF del receptor PAU-RAD [3].

Por otra parte, la etapa de RF del receptor incorpora dos conmutadores (Switch 1 y Switch 2) que permiten seleccionar la señal de entrada del receptor. Así pues es posible conmutar la señal de entrada entre tres posibles: una entrada de antena empleada en el cálculo de los parámetros de Stokes; una entrada de ruido correlado, que obtiene señal procedente de una fuente de ruido blanco a través de un tercer divisor de Wilkinson y que se emplea en la calibración de amplitud y fase; y una entrada de ruido incorrelado, que obtiene el ruido térmico procedente de dos resistencias de 50 Ω y que se emplea en la calibración del offset (ver figura 3-5).

3.2.2.2 Etapa Frecuencia Intermedia

Las señales procedentes de la etapa de RF pasan a la etapa de Frecuencia Intermedia (FI). La etapa de FI incorpora cuatro chips (uno para cada canal) receptores de GPS modelo GP2015 de la marca Zarlink. La función principal de estos circuitos integrados es la de convertir en frecuencia las señales RF a la entrada de cada canal, de manera que a la salida de cada chip se obtiene una señal centrada en una frecuencia intermedia de 4,3 MHz con un ancho de banda de 2,2 MHz.

A la salida de los integrados GP2015, cada canal dispone de una resistencia ajustable (R_{adj}) que permite llevar a cabo un ajuste aproximado (el ajuste fino se lleva a cabo de manera digital en la FPGA) de la ganancia de cada canal, de forma que la potencia de salida de los cuatro canales sea lo más igualada posible.

A continuación, las señales de cada canal pasan a un amplificador de video que proporciona a su salida señales diferenciales. Estas señales pasan por una red de adaptación de impedancias y finalmente van a parar a un conector de tipo RJ-45, al que se conecta un cable STP (Shielded Twisted Pair) Ethernet que hace las veces de transporte hasta el conversor analógico-digital.

La figura 3-6 muestra un esquema simplificado de los elementos que componen la etapa de FI de un receptor:

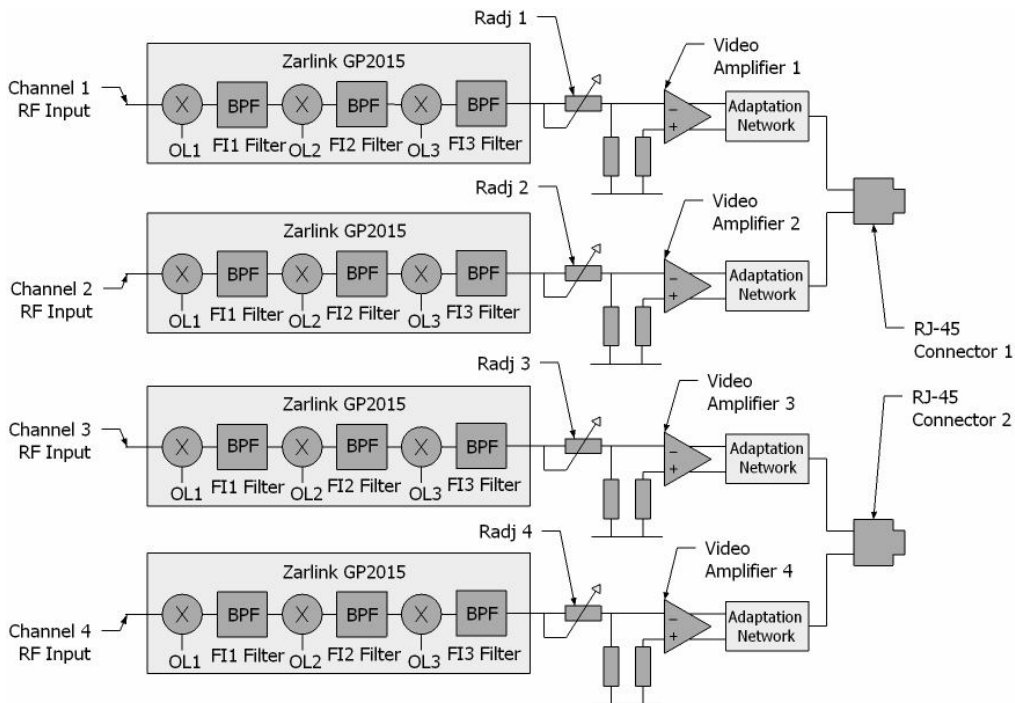


Figura 3-6: Diagrama de bloques de la etapa de FI del receptor PAU-RAD [3].

Finalmente las figuras 3-7 y 3-8 muestran fotografías de las etapas de RF y FI que forman parte de la implementación definitiva del receptor de PAU-RAD.

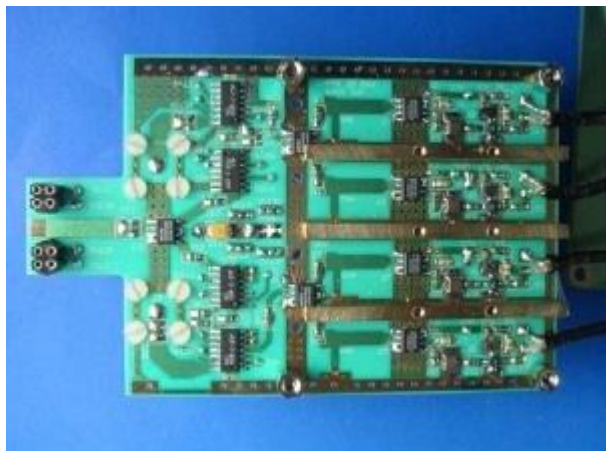


Figura 3-7: Fotografía de la etapa de RF del receptor PAU-RAD [3].



Figura 3-8: Fotografía de la etapa de FI del receptor de PAU-RAD [3].

3.2.3 Conversor Analógico-Digital

El conversor analógico-digital o ADC (Analog to Digital Converter) constituye el nexo de unión entre la parte analógica y la parte digital de PAU-RAD. Su función es la de digitalizar las señales procedentes de la etapa de FI del receptor a fin de poder ser procesadas en la FPGA.

Al igual que ocurriera en el caso del receptor, el diseño e implementación del ADC de PAU no forman parte de los objetivos de este proyecto. En este apartado se realizará únicamente una breve descripción del funcionamiento del ADC. Para una descripción exhaustiva del mismo consultar [5].

El núcleo del ADC está formado por el circuito integrado AD9201 de la marca Analog Devices, que se encarga de muestrear las señales procedentes del receptor a 10 bits. De estos 10 bits la FPGA utiliza tan sólo los 8 de mayor peso. El motivo por el cual se lleva a cabo este truncamiento de bits es que el factor de agrupación resultante de la digitalización a 8 bits de las señales, cumple los requisitos necesarios para aplicaciones radiométricas (esencialmente un $NLPS \leq -20$ dB), mientras que el uso de señales digitalizadas a 10 bits no supone una mejora apreciable. Por tanto en el diseño original se optó por emplear tan sólo 8 bits, lo que permite reducir el coste computacional de la FPGA así como el número de pines de entrada/salida empleados. El estudio completo sobre el impacto de la digitalización de las señales a 8 bits puede consultarse en [7].

Las señales que llegan al ADC procedentes del receptor presentan el espectro que se muestra en la siguiente figura:

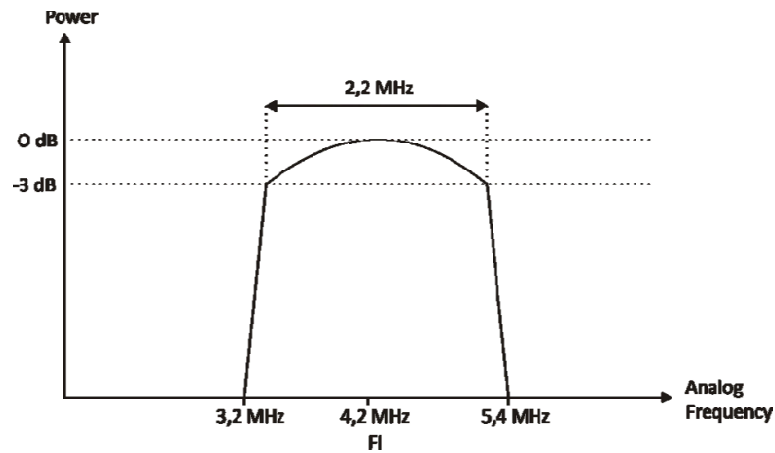


Figura 3-9: Aspecto que presenta el espectro de la señal de entrada al ADC.

Las señales son digitalizadas a una frecuencia de muestro F_s tal que la frecuencia digital resultante sea de 0,25, es decir: $F_s = 5,745$ MHz. Esta técnica es conocida con el nombre de "*band-pass sampling*" y facilita enormemente el procesado digital de la señal en banda base, puesto que para llevar a cabo la demodulación en fase y cuadratura únicamente será necesario multiplicar por la secuencia 1, 0, -1, 0.

Finalmente, el espectro digital a la salida del ADC presenta el siguiente aspecto:

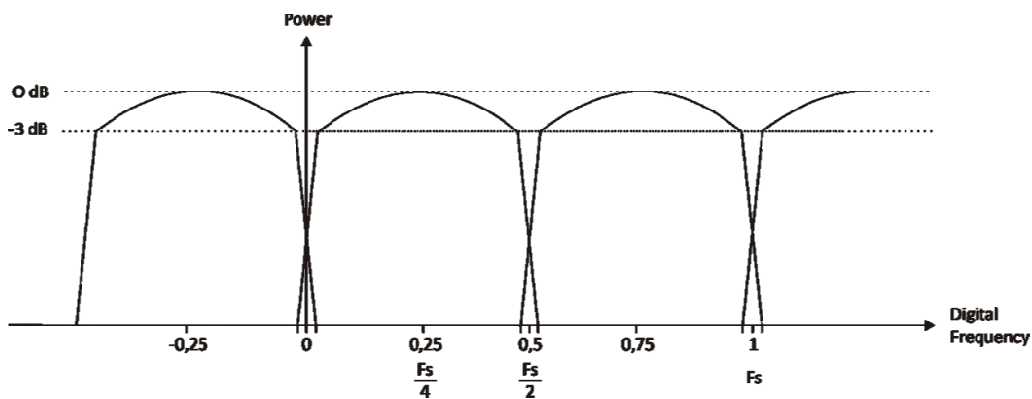


Figura 3-10: Aspecto que presenta el espectro de la señal digitalizada a la salida del ADC.

El diseño e implementación definitivos del ADC, vienen determinados por la limitación de pines de entrada de la FPGA. Cada uno de los 16 receptores que componen la estructura final de PAU-RAD requiere su propio ADC, el cual se conecta a los pines de entrada/salida de la FPGA. Teniendo en cuenta que cada receptor tiene cuatro canales y que cada canal se codifica con 8 bits, en total son necesarios 512 pines de entrada/salida para conectar toda la parte analógica de PAU-RAD a la parte digital.

No obstante la FPGA dispone tan solo de 107 pines accesibles por el usuario, lo cual imposibilita la entrada de las señales digitales procedentes de los 16 ADC en paralelo de forma simultánea. Para solucionar este inconveniente se implementó una solución basada en el multiplexado temporal de las señales, aprovechando las características del integrado AD9201:

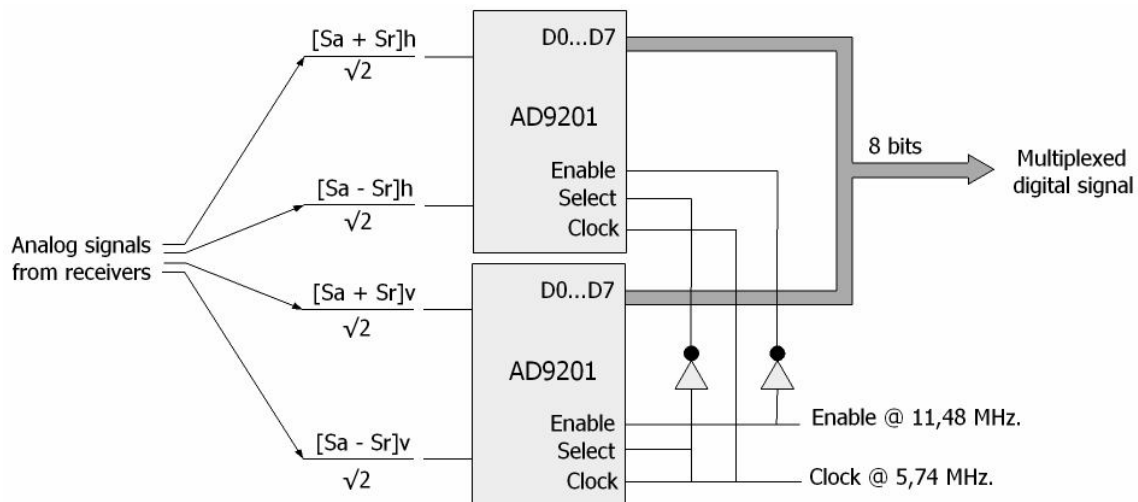


Figura 3-11: Esquema que ilustra la conexión de dos integrados AD9201 para formar la señal digitalizada multiplexada de entrada a la FPGA.

La figura 3-11 muestra un esquema de la solución implementada. Empleando esta configuración, se consigue un multiplexado de factor 4, lo que supone una reducción del número de pines de entrada/salida necesarios de 512 hasta 128. A pesar de esto, el número de pines sigue siendo insuficiente para conectar los 16 ADC (recordar que la placa dispone tan sólo de 107 pines). Para solucionar este inconveniente se optó por distribuir el sistema en dos FPGA iguales, una de las cuales actuará como master y la otra como slave. De esta forma se distribuye equitativamente entre las dos placas el número de ADCs a conectar (8 en cada FPGA), con lo que el número de pines deja de ser un impedimento.

3.2.4 FPGA

La etapa digital de PAU-RAD se encuentra implementada en una FPGA modelo EP1S10F780C6-ES de la marca Altera. Esta FPGA se incluye en la placa de evaluación que se muestra en la figura 3-12, y que contiene todos los dispositivos y conectores necesarios para la implementación del sistema.

Como se ha comentado en el apartado anterior, será necesario el uso de dos de estas placas de evaluación debido fundamentalmente a la limitación en el número

de pines de entrada/salida de la placa así como al elevado coste computacional que supone el diseño de PAU-RAD.

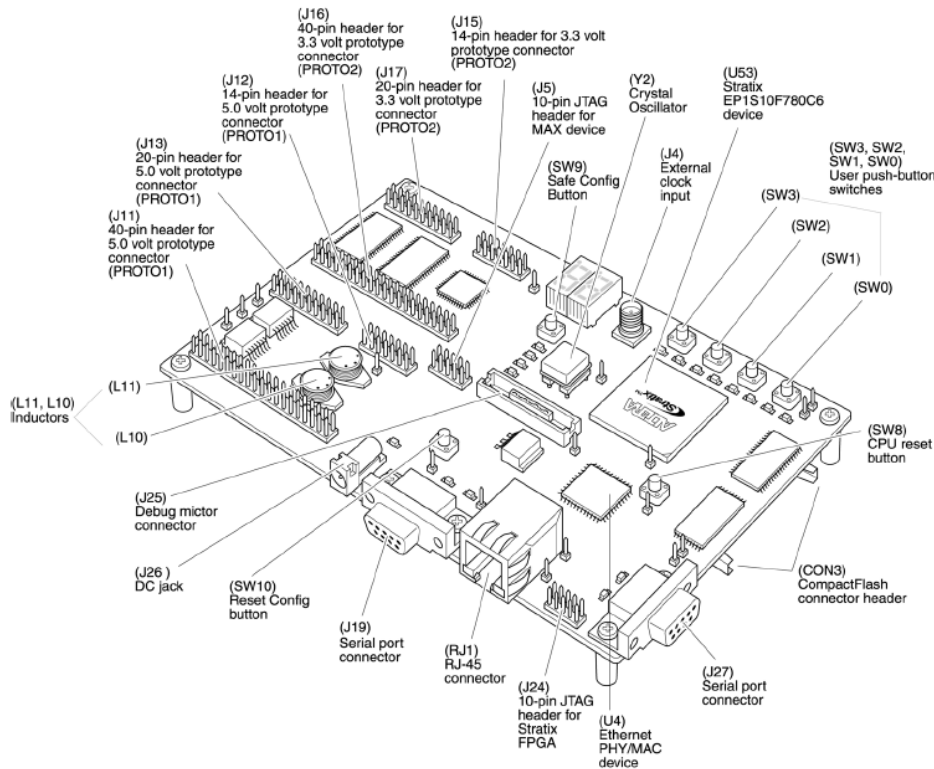


Figura 3-12: Diagrama de la placa de evaluación EP1S10F780C6-ES de Altera.

La figura 3-12 muestra los diferentes elementos que componen la placa de evaluación. El componente fundamental de esta placa es el dispositivo de la FPGA (U53 en la figura), se trata de un dispositivo de programación volátil, lo que significa que si se desconecta la alimentación, al volver a conectarla la FPGA arranca con la configuración de fábrica por defecto.

La tabla 3-1 muestra algunas de las características fundamentales de las FPGA de la familia EP1S10:

Tabla 3-1: Características fundamentales de las FPGA de la familia EP1S10 de Altera.

Stratix EP1s10 Device Features	
LEs	10570
M512 RAM blocks (32 x 18 bits)	94
M4k RAM blocks (128 x 36 bits)	60
M-RAM blocks (4k x 144 bits)	1
Total RAM bits	920448
DSP blocks	6
Embedded multipliers	48
PLLs	6
Maximum user I/O pins	426

De entre estas características, cabe destacar la primera: el número de elementos lógicos (LE) de los que dispone la FPGA, (en este caso 10.570) ya que es el parámetro que determina la capacidad de la FPGA.

A pesar de tratarse de un dispositivo de programación volátil, la EP1S10 incorpora 8 Mbytes de memoria flash AMD AM29LV065D, que pueden emplearse o bien como memoria de propósito general no volátil para el procesador, o bien para almacenar una configuración por defecto que se cargue cada vez que se alimente la placa, de manera que no sea necesario cargar continuamente la programación.

Además de la memoria flash, la placa dispone de 16 Mbytes de memoria SDRAM (Synchronous Dynamic Random Access Memory) Micron MT48LC4M32B2 y de 2 chips de memoria SRAM (Static Random Access Memory) asíncrona de 512 Kbytes x 16 bits que pueden ser empleados por el procesador como memoria de propósito general y pueden ser configurados para trabajar en paralelo implementando una memoria de 512 Kbytes x 32 bits o bien de 1 Mbyte x 16 bits.

La comunicación entre la FPGA y el PC corre a cargo del controlador Ethernet LAN91C111 (U4 en la figura). Se trata de un dispositivo que implementa protocolos de 10 Mbps y 100 Mbps y al cual se puede acceder a través de un conector RJ-45. Alternativamente la placa dispone de dos conectores DB9 estándar (J19 y J27 en la figura) que permiten una comunicación RS-232 con el PC a través de una UART (Universal Asynchronous Receiver-Transmitter).

El reloj central de la placa está implementado mediante un oscilador de cristal, que, por defecto, es de 50 MHz. No obstante, como ya se ha comentado en este mismo capítulo, la frecuencia de muestreo de PAU-RAD es de 5,745 MHz y teniendo en cuenta que los datos se multiplexan por un factor 4, es necesaria una frecuencia de trabajo de la FPGA de 22,98 MHz.

Por motivos de compatibilidad con el dispositivo de grabación de la placa, se optó por elegir un oscilador con una frecuencia múltiplo de 5,745 MHz lo más próxima a 50 MHz posible. El valor elegido fue 46,98 MHz, lo que significa que los datos son procesados durante la primera mitad del ciclo de trabajo de la FPGA. El tiempo restante del ciclo de trabajo de la FPGA se aprovecha para realizar el procesamiento de los datos para un segundo haz de la antena.

3.2.4.1 Unidad Aritmético-Lógica

La unidad aritmético-lógica (ALU) de PAU-RAD es la encargada de llevar a cabo el procesamiento de los datos del radiómetro que llegan a la FPGA digitalizados y multiplexados a través de los ADC. La ALU ha sido implementada mediante el lenguaje

de descripción de hardware digital VHDL (Very high speed integrated circuit Hardware Description Language), se trata por tanto de hardware digital sintetizado a través de conexiones entre los elementos lógicos de la FPGA.

El diseño e implementación de la ALU han sido desarrollados en [2], donde puede encontrarse una descripción exhaustiva de todos sus componentes. Aquí se describirán tan sólo algunos de los elementos más importantes, aquellos que por su función se hallan íntimamente ligados al desarrollo del presente proyecto.

Como se ha comentado, el diseño de la ALU se ha distribuido en dos FPGA iguales con el fin de disponer de un número de pines de entrada-salida suficiente para dar cabida a todos los ADC que componen el sistema. La figura 3-13 ilustra la interconexión de las dos FPGA:

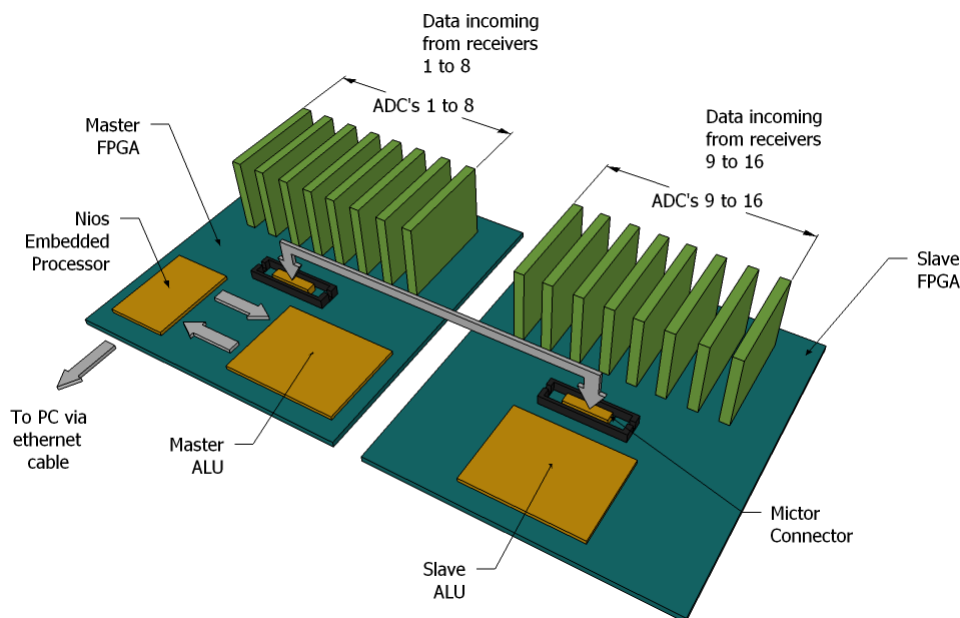


Figura 3-13: Ilustración del montaje de interconexión de las dos FPGA.

Cada una de las dos FPGA procesa los datos procedentes de 8 de los 16 ADC. Los datos adquiridos por la placa *slave* se envían, a través de un conector de 25 pines (conector Mictor), a la placa *master*. Debido a esto, la ALU *master* es ligeramente más compleja que la ALU *slave*, ya que es la encargada de obtener el resultado final a partir de los datos combinados de las dos FPGA y además debe incorporar los sistemas de comunicación con el procesador Nios embebido en la misma FPGA.

A grandes rasgos la ALU se puede descomponer en tres bloques: el bloque *InterCom*, encargado de llevar a cabo la comunicación *master-slave*; el bloque *Radiometer*, que constituye la implementación de la parte digital del radiómetro y el bloque *NiosCom*, presente únicamente en la ALU *master* y que realiza la comunicación

entre la FPGA y el procesador Nios. La figura 3-14 muestra el diagrama de bloques simplificado de las dos ALU:

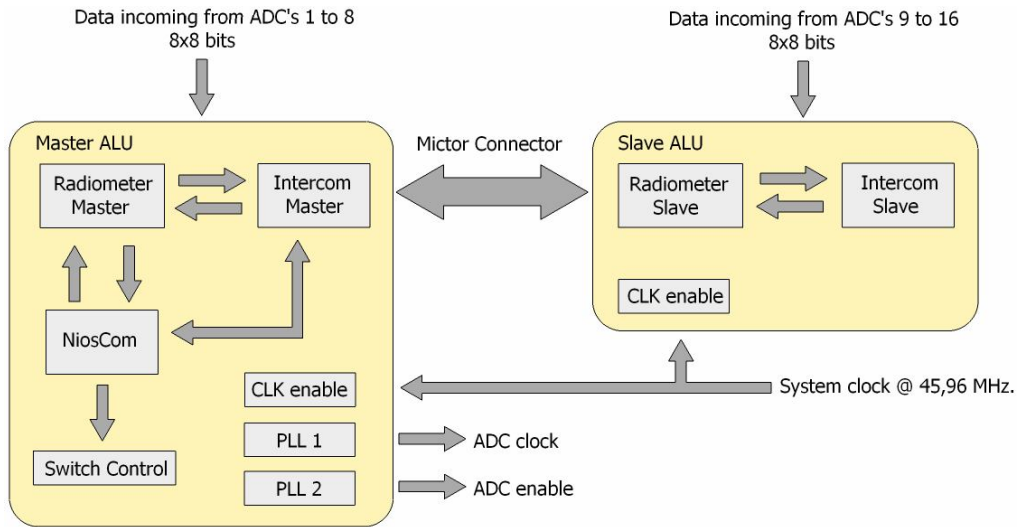


Figura 3-14: Diagrama de bloques de la interconexión entre los diferentes elementos de las dos ALU.

A continuación se describirá brevemente la funcionalidad de los bloques *Radiometer* y *NiosCom*, puesto que se hallan íntimamente ligados al desarrollo del presente proyecto:

- o **Radiometer**

El bloque *Radiometer* constituye la implementación de la parte digital del radiómetro. Proporciona a su salida los parámetros de Stokes, así como los datos necesarios para que el procesador Nios estime los errores de fase, ganancia y offset relativos entre canales. Se trata, por tanto, del componente más importante de la ALU.

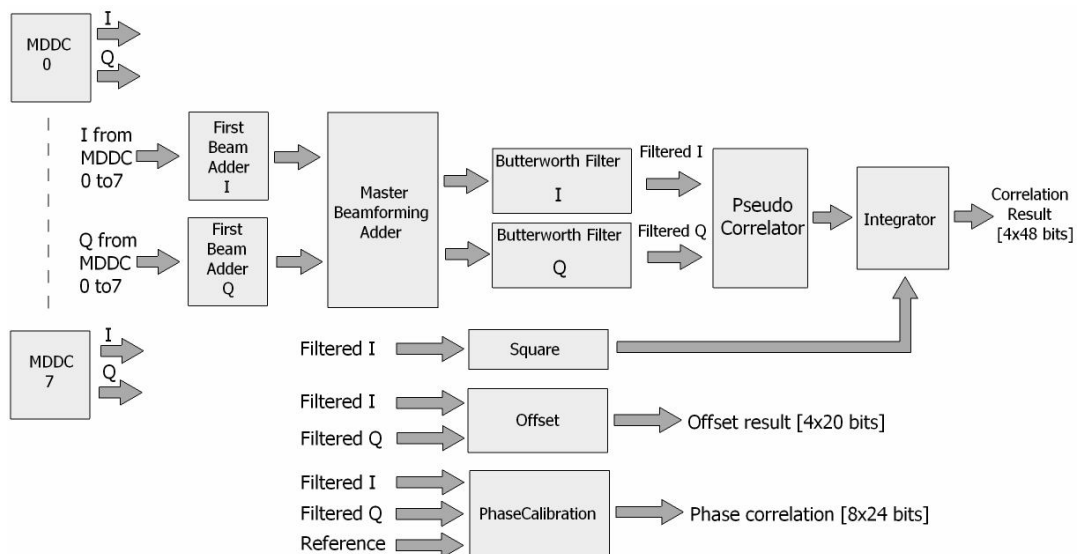


Figura 3-15: Diagrama de bloques del bloque Radiometer [2].

La figura 3-15 muestra un diagrama de bloques simplificado del bloque *Radiometer* de la placa *master*. A pesar de que el bloque *Radiometer* de la placa *master* difiere en algunos aspectos del implementado en la placa *slave*, se describirá únicamente el primero, puesto que es el de mayor complejidad.

El primer elemento del bloque *Radiometer* es el denominado *MDDC* (Multiplexed Digital Down Converter). El *MDDC* es uno de los elementos clave del sistema, puesto que es el encargado de pasar a banda base tanto en fase como en cuadratura, las señales multiplexadas procedentes de los ADC. Por tanto, son necesarios 8 *MDDC*, uno para cada ADC.

Al mismo tiempo que se lleva a cabo la conversión en frecuencia a banda base, se realizan las correcciones de los errores de fase y ganancia relativos entre canales así como el desfase necesario para el conformado del haz. En este proceso interviene un sub-bloque del *MDDC* denominado *TrigonometricTable*.

El sub-bloque *TrigonometricTable* es básicamente un registro de 16 posiciones de 9 bits. La información contenida en este registro se conoce como: TT (Tablas Trigonómicas). Las TT son un conjunto de funciones trigonométricas sinusoidales encargadas de la corrección del módulo i la fase de cada una de las cadenas receptoras así como de la asignación de la fase correspondiente a la conformación de los dos haces de la antena. También desempeñan la función de convertir en frecuencia a banda base las señales digitalizadas en los ADC, que legan a la FPGA centrados en una frecuencia digital $F = 0,25$.

Como resultado de la utilización de esta frecuencia digital, se obtienen funciones trigonométricas de periodo 4, de tal forma que tan solo es necesario almacenar en el registro *TrigonometricTable* los 4 primeros valores de la función y recorrerlo de forma cíclica.

A esto último se le añade el hecho de que no es necesario almacenar tanto senos como cosenos para convertir en frecuencia las señales en fase y en cuadratura, sino que basta con utilizar únicamente cosenos y obtener los senos aplicando un desfase de $\frac{\pi}{2}$. Este desfase representa un retraso de una muestra en la frecuencia digital, de forma que, recorriendo la tabla con un retardo de una muestra, se pueden obtener los senos.

Finalmente, la información contenida en cada una de las posiciones de las TT se corresponde con el valor de una ecuación trigonométrica, digitalizado a 9 bits y expresado en complemento a 2, por tanto el rango de valores va de -256 a 255. La tabla 3-2 muestra las ecuaciones que corresponden a cada una de las posiciones de la tabla:

Tabla 3-2: Ecuaciones que se corresponden con cada uno de los valores almacenados en las 16 posiciones de las TT, para un receptor y un único haz.

POSICIÓN	ECUACIÓN TRIGONOMÉTRICA
1	$A_1 \cos(\theta_1)$
2	$A_2 \cos(\theta_2)$
3	$A_3 \cos(\theta_3)$
4	$A_4 \cos(\theta_4)$
5	$A_1 \cos(\pi/2 + \theta_1)$
6	$A_2 \cos(\pi/2 + \theta_2)$
7	$A_3 \cos(\pi/2 + \theta_3)$
8	$A_4 \cos(\pi/2 + \theta_4)$
9	$A_1 \cos(\pi + \theta_1)$
10	$A_2 \cos(\pi + \theta_2)$
11	$A_3 \cos(\pi + \theta_3)$
12	$A_4 \cos(\pi + \theta_4)$
13	$A_1 \cos(3\pi/2 + \theta_1)$
14	$A_2 \cos(3\pi/2 + \theta_2)$
15	$A_3 \cos(3\pi/2 + \theta_3)$
16	$A_4 \cos(3\pi/2 + \theta_4)$

Donde A_i (con $i = 1, 2, 3, 4$) representa el factor de corrección de ganancia, mientras que θ_i se corresponde con el desfase que es necesario aplicar para corregir el error de fase y asignar el valor de fase correspondiente al conformado de haz.

Una vez realizada la conversión en frecuencia y la corrección de los errores, las señales pasan por el bloque *FirstBeamAdder*, encargado de asignar a los diferentes canales los pesos necesarios para la conformación del haz con una iluminación triangular 1:2:2:1, de manera que la ponderación de los 16 elementos que componen la agrupación de antenas sea la siguiente:

$$\begin{bmatrix} 1 & 2 & 2 & 1 \\ 2 & 4 & 4 & 2 \\ 2 & 4 & 4 & 2 \\ 1 & 2 & 2 & 1 \end{bmatrix} \quad (3.2)$$

Una vez asignados los pesos, las señales en fase y cuadratura se filtran mediante un filtro paso-bajo Butterworth para eliminar las componentes de alta frecuencia resultantes de la conversión en frecuencia y obtener la componente en banda base.

El siguiente bloque en la cadena de procesado es un pseudo-correlador que realiza las correlaciones cruzadas que proporcionan a su salida los cuatro componentes del vector instantáneo de Stokes:

$$\begin{aligned}
T_1 &\propto \langle I_{V^+} I_{V^-} \rangle \propto \langle |E_V|^2 \rangle \\
T_2 &\propto \langle I_{H^+} I_{H^-} \rangle \propto \langle |E_H|^2 \rangle \\
T_3 &\propto \langle I_{V^+} I_{H^-} \rangle \propto 2\text{Re}\{E_V E_H^*\} \\
T_4 &\propto \langle I_{H^+} Q_{V^+} \rangle \propto 2\text{Im}\{E_V E_H^*\}
\end{aligned}
\tag{3.3}$$

Donde: T_i (con $i = 1, 2, 3, 4$) representa una medida de potencia instantánea dependiente de la temperatura para cada uno de los cuatro componentes del vector de Stokes; I y Q indican señales en fase o cuadratura respectivamente; los subíndices H y V indican la polarización de la señal de entrada y Re e Im se corresponden con el operador parte real e imaginaria respectivamente.

Estos resultados son medidas instantáneas de potencia, por tanto, para obtener los parámetros de Stokes es necesario integrar la salida del pseudo-correlador. Esta es precisamente la función del bloque *Integrator*, un acumulador digital que integra los datos durante un tiempo determinado por una variable que puede ser modificada en tiempo real por el usuario. A su salida se obtienen 4 valores de 48 bits correspondientes a los 4 parámetros de Stokes.

De forma paralela a la cadena principal de procesamiento que se ha descrito, trabajan tres bloques más: *Square*, cuya finalidad es la de estimar el error de ganancia de las señales de los cuatro canales; *Offset*, que proporciona una medida de los offsets añadidos y *PhaseCalibration* que entrega 8 medidas de similitud (correlaciones a un bit) entre las señales en fase y cuadratura de los cuatro canales con respecto al canal de referencia, que serán empleadas para determinar el error de fase relativo.

En resumen, el bloque *Radiometer* proporciona 4 tipos básicos diferentes de información a su salida:

- Cuatro medidas que constituyen el valor de los cuatro parámetros de Stokes.
- Cuatro medidas de potencia, una para cada canal.
- Ocho medidas de similitud de fase: cuatro para la similitud entre la fase del canal de referencia y la fase del resto de canales; y cuatro para la similitud entre la fase del canal de referencia y la cuadratura del resto de canales.
- Cinco medidas de offset: cuatro para las señales en fase de cada canal y una para la cuadratura del canal de referencia.

Es importante hacer hincapié en la naturaleza y cantidad de los datos que proporciona el bloque *Radiometer* a su salida, ya que éstos serán los datos de los que disponga en procesador para realizar las correcciones oportunas, ámbito en el que se centra parte del desarrollo de este proyecto.

o **NiosCom**

El bloque *NiosCom* es el encargado de gestionar las comunicaciones entre la ALU y el procesador Nios que ejecuta el código de la UC (Unidad de Control, cuya arquitectura y funciones se abordarán con detalle en el capítulo 4). Se encarga también de interpretar las instrucciones recibidas desde la UC y generar las señales de control correspondientes.

Según el sentido de la comunicación, *NiosCom* se subdivide en dos bloques: *nioscom_r*: si la comunicación es de la ALU hacia la UC, y *nioscom_w*: si es de la UC hacia la ALU:

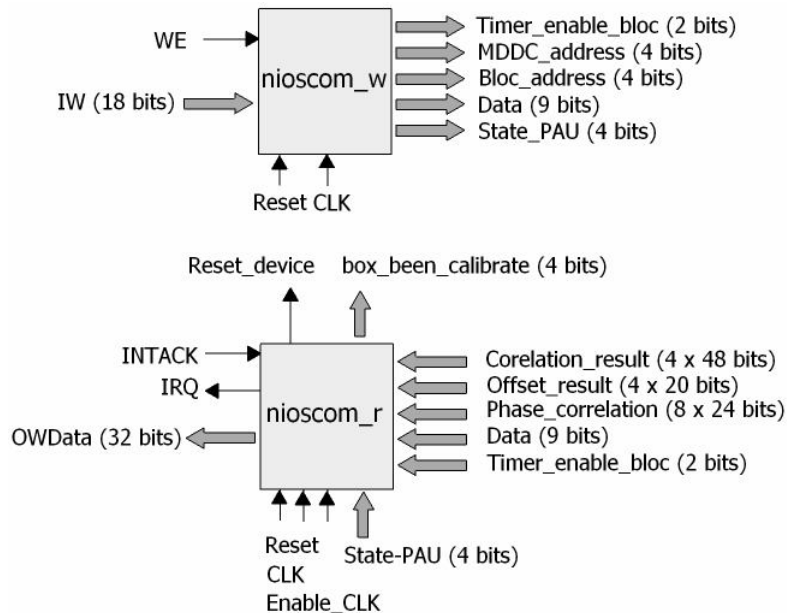


Figura 3-16: Esquema que muestra las diferentes entradas y salidas de los dos sub-bloques que componen el bloque *nioscom* [2].

El sub-bloque *nioscom_w* es el encargado de interpretar el estado de funcionamiento del sistema (salida *State_PAU*), fijado por la UC (a través del registro *IW*). Además determina la posición en la que se escriben los datos en la TT según procedan de uno u otro receptor (mediante la salida *MDDC_address*) y modifica el temporizador que fija el tiempo de integración (*Timer_enable_bloc*) según el valor fijado por el usuario.

Por su parte, el sub-bloque *nioscom_r* implementa un protocolo de transmisión de datos con la UC que se basa en la utilización combinada de señales de confirmación

(entrada INTACK) e interrupciones (entrada IRQ) y que será descrito en el capítulo 4. También lleva a cabo la transmisión de los datos procedentes del bloque *Radiometer* a la UC (a través del registro OWDData).

3.2.4.2 Procesador Nios

El microprocesador Nios constituye el centro de procesamiento de todos los datos procedentes de la ALU. Su función es la de ejecutar el software que implementa la UC y por tanto es el encargado de llevar a cabo, entre otras tareas, la calibración de los datos y su envío al PC (Personal Computer) en el que serán visualizados por el usuario.

Como ya se ha comentado anteriormente, es precisamente en el código que ejecuta el procesador en el que se centra parte del trabajo realizado en el transcurso de este proyecto, de ahí que se dedique un capítulo (el capítulo 4) a su estructura y funcionamiento. En estas líneas se describirán de manera muy breve algunas de las características más destacadas del procesador.

Nios es un procesador de 32 bits embebido, descrito mediante un lenguaje de descripción de hardware digital (VHDL) y optimizado para los dispositivos lógicos programables de la casa Altera. Se trata de un procesador de propósito general de tipo RISC (Reduced Instruction Set Computer), segmentado y mono tarea capaz de ejecutar la mayoría de instrucciones en un solo ciclo de reloj. El número de registros de propósito general de los que dispone el procesador es configurable, habiéndose fijado en 512 para este proyecto.

Tabla 3-3: Características de la arquitectura del procesador Nios de 32 bits.

Nios CPU Details	32-bit Nios CPU
Data bus size (bits)	32
ALU width (bits)	32
Internal register width (bits)	32
Address bus size (bits)	32
Instruction size (bits)	16

El juego de instrucciones que posee el procesador está orientado a soportar programas compilados en C y C++. Incluye un conjunto estándar de operaciones lógicas y aritméticas e instrucciones que proporcionan soporte a operaciones bit a bit, extracción de bytes, modificación del control de flujo e instrucciones de ejecución condicional.

Tanto la arquitectura interna del procesador, como su conexión con el resto de elementos del sistema, son altamente configurables por el usuario gracias al software de Altera SOPC Builder (System on a Programmable Chip Builder). Mediante el SOPC

Builder, se automatiza el conexionado de múltiples bloques pre diseñados o IP (Intellectual Properties), como puedan ser diferentes tipos de memorias, interfaces o periféricos, para crear un sistema computacional completo.

Uno de los periféricos de mayor importancia en el desarrollo del proyecto son los registros de entrada-salida paralela o PIO (Parallel Input-Output). Se trata de registros de longitud variable de 1 a 32 bits cuya función es la de actuar como puentes de comunicación entre el software implementado en el procesador y la lógica creada por el usuario. En el caso del presente proyecto, los PIO son las vías de comunicación entre la UC y la ALU y a través de ellos se transmiten la mayoría de entradas y salidas definidas en los bloques *Radiometer* y *Nioscom*.

4 UNIDAD DE CONTROL E INTERFAZ DE CAPTURA DE DATOS

4.1 INTRODUCCIÓN

En este capítulo se introducirá el funcionamiento y la arquitectura de la UC (Unidad de Control), encargada de llevar a cabo, entre otras, las funciones de calibración y transmisión de datos de PAU-RAD. Para ello se definirán algunos de los parámetros más relevantes del sistema, se describirán los diferentes estados que componen la máquina de estados que es la UC y por último se analizará como se enlazan estos estados durante la ejecución para dar lugar a determinadas secuencias de funcionamiento. Conjuntamente a la descripción de la UC se llevará a cabo una descripción de la interfaz encargada de la captura y visualización de los datos procedentes del radiómetro.

Inicialmente se disponía de una primera versión de la UC funcional para un único receptor [4]. Uno de los objetivos iniciales del proyecto consistía en la ampliación del esta UC, de tal forma que fuese capaz de operar con el sistema completo de 16 receptores. Sin embargo, durante el proceso de ampliación quedó patente la necesidad de mejorar algunos de los aspectos de la calibración, de forma que se desvió el rumbo del trabajo en esa dirección. Así pues, se realizaron modificaciones a los algoritmos de calibración existentes y se implementaron nuevos algoritmos. Al mismo tiempo, se modificó el flujo entre estados de la UC para dar cabida a los nuevos algoritmos y se crearon nuevos estados que permitiesen validar el hardware del sistema. Por este motivo, al término del proyecto se dispone de una versión preliminar que trabaja únicamente con 4 receptores.

De igual forma, se disponía de una implementación inicial de la interfaz externa encargada de recoger y presentar los datos al usuario en un PC. Esta interfaz, implementada en Matlab, mostraba los resultados en forma numérica y los almacenaba en memoria, de tal manera que el acceso a los datos almacenados era posible únicamente una vez detenido el sistema.

Durante el transcurso de las pruebas de los algoritmos de calibración se llegó a la conclusión de que era necesario un sistema de presentación de datos alternativo, que permitiese analizar de forma más cómoda y ágil los resultados obtenidos. Debido a esto se implementó una nueva versión de la interfaz, que representa al mismo tiempo los principales parámetros del sistema, en tiempo real, de forma gráfica y mostrando el progreso durante una ventana temporal de cierta duración. A lo largo del capítulo se

hará referencia a ésta como Interfaz de captura de datos, Interfaz de usuario, o sencillamente Interfaz.

4.2 UNIDAD DE CONTROL DE PAU-RAD

4.2.1 Descripción

La UC es el conjunto de subrutinas software encargado de la adquisición y el procesamiento de los datos procedentes del radiómetro así como de la transmisión de los datos procesados. Los resultados son recogidos por un PC, en el que serán visualizados por el usuario al mismo tiempo que se almacenan para posibles usos futuros. Como ya se comentó en el capítulo anterior, el código que implementa la UC es ejecutado por el procesador Nios I de Altera, que se encuentra embebido en la propia FPGA. De ahí que el lenguaje empleado en el desarrollo de la UC sea C estándar o ANSI (American National Standards Institute) C, ya que el procesador Nios I está especialmente diseñado para la ejecución de programas desarrollados en este lenguaje.

La UC se comporta, en esencia, como una máquina de estados finitos que gestiona toda la secuencia de procesamiento de datos que proporciona los parámetros de Stokes. Básicamente constituye el lazo de unión entre el hardware del sistema (tanto analógico como digital) y el PC que presenta los resultados al usuario.

Las funciones básicas de la UC se resumen a continuación:

- Obtención de los errores de fase y ganancia a partir de las medidas de calibración proporcionadas por la ALU del radiómetro.
- Cálculo de los 32 valores (16 para cada uno de los dos haces) de las 16 Tablas Trigonómicas que forman el sistema completo a partir de los errores obtenidos.
- Control del flujo de la máquina de estados. Gestión de la transición entre los diferentes estados del sistema y de los comandos recibidos desde la interfaz.
- Envío de los datos procesados a la interfaz de captura de datos para su presentación y almacenamiento.

Después de una primera etapa de inicialización en la que se fijan los valores iniciales de las variables del sistema, la UC trabaja según una secuencia establecida que puede dividirse en dos grandes etapas: una etapa de calibración en la que se obtienen y corrigen los diferentes errores inherentes al sistema y una segunda etapa en la que se calculan los parámetros de Stokes y se envían a la interfaz. Esta secuencia se repite

de forma periódica (según un período establecido por el usuario) dando lugar a lo que se ha definido como *Secuencia Principal*, que proporciona los parámetros de Stokes corregidos de forma continua. Esta *Secuencia Principal* se reinicia en el momento en que el usuario introduce en la interfaz alguna modificación a los parámetros iniciales del sistema.

De forma alternativa a la *Secuencia Principal* existen otras secuencias de ejecución que permiten comprobar el correcto funcionamiento tanto del software como del hardware del sistema. A estas secuencias se las ha denominado *Secuencias de Validación* y se accede a ellas únicamente cuando el usuario activa los botones correspondientes en la interfaz. Una vez activada una secuencia de validación, la *Secuencia Principal* se detiene hasta que el usuario la desactiva, momento en que el sistema se reinicia y comienza de nuevo la *Secuencia Principal*.

La figura 4-1 muestra un diagrama que recoge los principales estados de de la UC y las transiciones entre ellos:

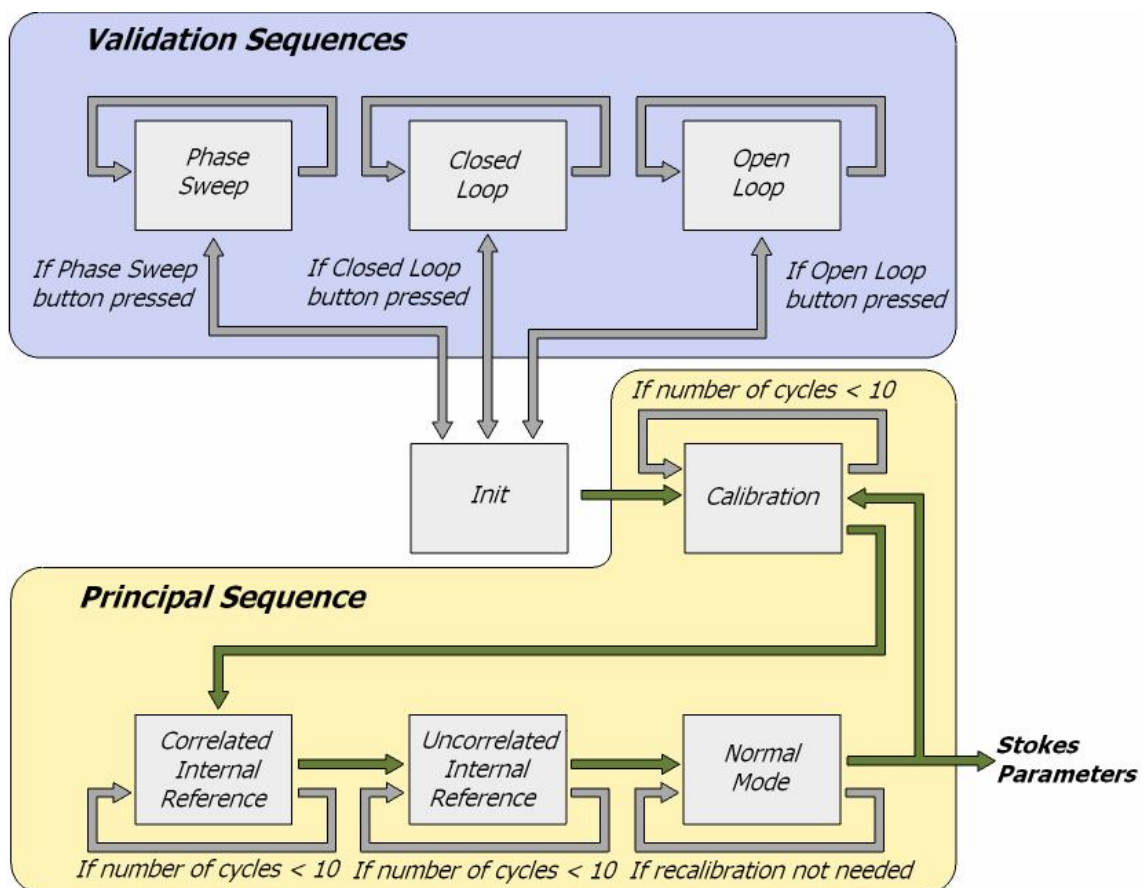


Figura 4-1: Diagrama de estados de la Unidad de Control. En color verde se destaca el flujo entre estados de la secuencia principal.

En la figura se destacan en color verde las líneas de transición entre estados que componen la *Secuencia Principal*. Como se puede observar, los estados que

componen la *Secuencia Principal* se iteran un número determinado de veces. En la figura se muestran 10 iteraciones para cada estado, no obstante este número es configurable, tal y como se verá más adelante. Estas iteraciones permiten promediar los datos de calibración para mejorar los resultados. Por otra parte, el estado denominado *Normal Mode*, encargado de obtener los parámetros de Stokes, se itera durante cierto tiempo hasta que se requiere una nueva calibración, evitando así la degradación de los resultados. Todos los estados de que dispone la UC se analizarán con mayor detalle más adelante en este mismo capítulo.

Como ya se ha mencionado, la función esencial de la UC es la de actuar como intermediaria entre el hardware del sistema y el usuario. La comunicación con el hardware se lleva a cabo a través de la ALU mediante el uso de registros internos de la FPGA, mientras que la comunicación con el PC de usuario se realiza a través de una conexión TCP/IP. Ambos sistemas de comunicación se detallan en este mismo capítulo.

4.2.2 Arquitectura

La UC se ha implementado siguiendo una arquitectura simple y modular, que facilita la incorporación al sistema de nuevos estados y la modificación de los ya existentes según sea necesario. El código se divide en archivos independientes, cada uno de los cuales ejecuta una función determinada. Así pues, el código de la UC se compone de un archivo que implementa la rutina principal (*pau.c*), diversas librerías que implementan las subrutinas necesarias para cada uno de los estados del sistema (*pau_init.h*, *pau_calibracio.h*, *pau_normal.h...*) y finalmente varios archivos que contienen las variables globales, las rutinas de atención a las interrupciones y las inicializaciones del protocolo TCP/IP.

La gestión de la transición entre estados se lleva a cabo en la rutina principal. El diagrama de flujo entre estados viene determinado por la variable global: *status*. Se trata de una variable de tipo enumeración que contiene una lista de todos los posibles estados del sistema, lo que la convierte en uno de los parámetros más importantes del sistema. Una vez iniciada la ejecución de la rutina principal, se entra en un bucle infinito que evalúa de forma continua el valor de *status* mediante una sentencia *switch*. Este *switch* contiene un caso para cada uno de los estados disponibles, de manera que en cada caso se realiza la llamada a la subrutina correspondiente al estado en cuestión. Al final de la ejecución de la subrutina se actualiza el valor de *status*, de forma que en la siguiente evaluación del *switch* se ejecutará la subrutina correspondiente al siguiente estado de la secuencia. La figura 4-2 ilustra este sistema de conformación del diagrama de flujo entre estados:

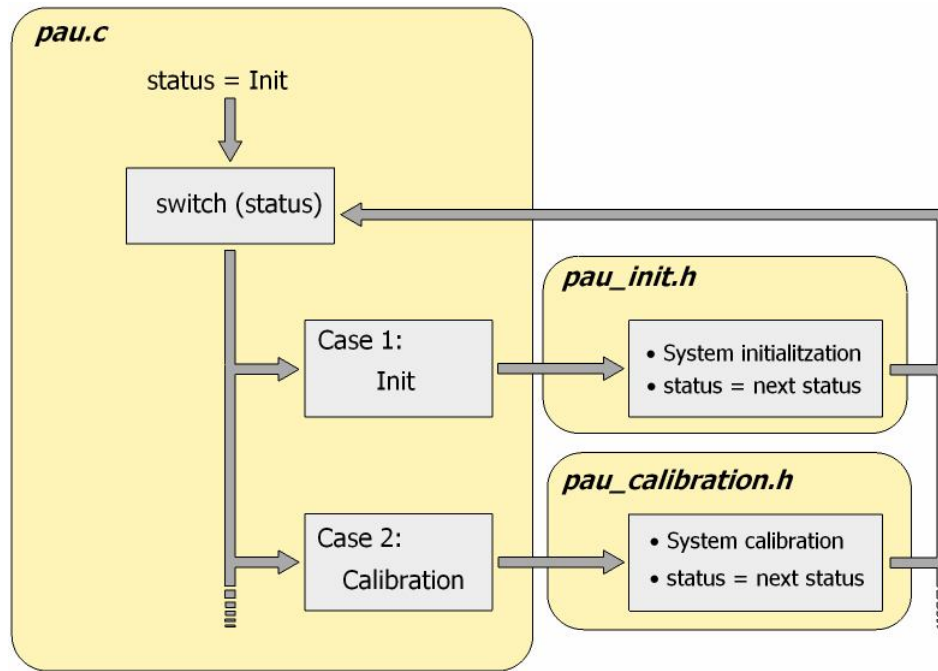


Figura 4-2: Sistema de conformación del diagrama de flujo entre estados. A efectos ilustrativos se muestran tan sólo los dos primeros estados, el funcionamiento para el resto de estados es idéntico.

Por tanto la función de la rutina principal de la UC es únicamente la de determinar el siguiente estado del sistema y llamar a la subrutina pertinente.

La interacción con la interfaz de usuario se lleva a cabo mediante un sistema de interrupciones. Cada vez que el usuario modifica alguno de los parámetros del sistema o bien inicia la ejecución de alguno de los modos de validación en la interfaz, se envía desde el PC un carácter de control. El envío de este carácter genera una interrupción, que es recogida por una rutina de atención a la interrupción o ISR (Interrupt Service Routine). Dentro de esta ISR se recoge el valor del carácter de control, que es almacenado en una variable global, y se actualiza el valor de la variable *status* para que el siguiente estado sea *Init*.

En la ejecución de la subrutina correspondiente al estado *Init*, se evalúa el carácter de control y se realizan las acciones pertinentes (cambio de algún parámetro del sistema o activación/desactivación de los métodos de validación). En la figura 4-3 se ilustra de manera esquematizada el mecanismo de interacción con la interfaz de usuario.

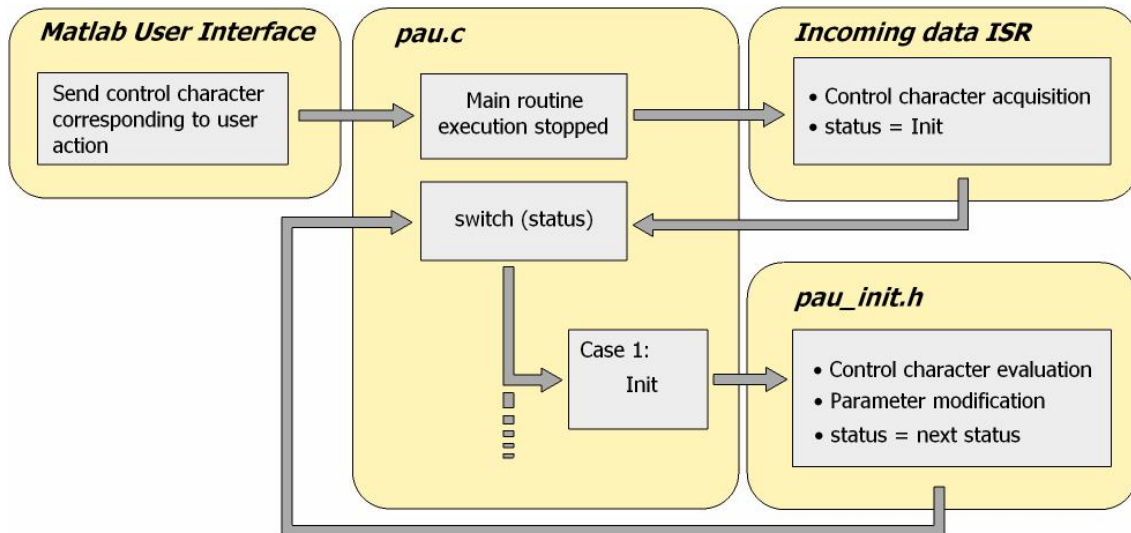


Figura 4-3: Esquemización del mecanismo de interacción con el usuario. La secuencia se inicia con una modificación del sistema introducida por el usuario en la interfaz de captura de datos. Esto se traduce en el envío de un carácter de control que provoca una interrupción de la rutina principal.

4.2.2.1 Mecanismo de comunicación con el radiómetro

La comunicación de la UC con el hardware del radiómetro se lleva a cabo a través de la ALU. Para ello se emplean unos periféricos disponibles en las librerías VHDL de Altera denominados PIO (Parallel Input Output). Los PIO son registros de longitud variable que permiten el flujo de datos directo entre el microprocesador Nios y el resto de hardware implementado en la FPGA, en este caso la ALU del radiómetro.

El registro encargado de llevar a cabo la comunicación desde la UC hacia la ALU recibe el nombre de *Input Word*. Se trata de un registro de 20 bits de longitud encargado del transporte a la ALU de las Tablas Trigonómicas calculadas en la UC. De manera adicional se transmiten también el tiempo de integración y el estado actual del sistema. La siguiente figura muestra la estructura del registro *Input Word*:

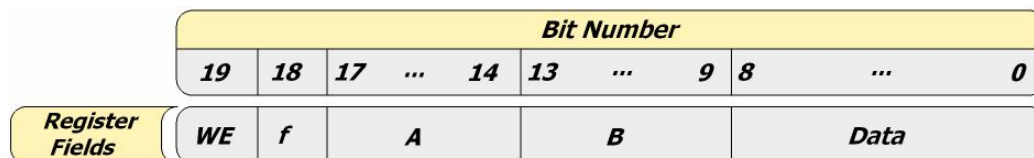


Figura 4-4: Estructura del registro *Input Word*.

Como se puede observar, el registro *Input Word* se divide en cinco segmentos diferentes. Los 9 bits de menor peso forman el primer segmento (*Data*), y en ellos se

almacena la información a transmitir (tiempo de integración o valor de una de las posiciones de la TT).

El siguiente segmento, denominado *B*, tiene una longitud de 5 bits y se emplea para diferenciar entre las tres partes en que se divide el valor del tiempo de integración. Puesto que el tiempo de integración es un entero de 27 bits (ver definición en el siguiente apartado) y la longitud del campo de datos del registro *Input Word* es de 9 bits, se necesitan tres transmisiones consecutivas para enviar el valor completo y la posición de la TT a la que corresponde Data.

El segmento *A*, de 4 bits de longitud, se emplea para transmitir el estado actual a la ALU. Esta información se emplea para accionar los selectores del hardware del radiómetro conmutando así entre las tres entradas disponibles (calibración, antena y carga interna).

Por último, los bits 18 (*f*) y 19 (*WE*) ejercen la función de “flags” de señalización. El bit 18 indica a la ALU que se está transmitiendo un valor de las TT cuando su valor es 0, y para el resto de casos su valor es 1. El bit 19 hace las veces de *Write Enable* (*WE*), de manera que la UC fija su valor a 1 cuando hay datos disponibles en el registro para que la ALU los lea.

La siguiente figura muestra los valores de *Input Word* para tres de los casos más habituales de transmisión UC → ALU: modificación del tiempo de integración, calibración y escritura de un valor en la TT.

		<i>Bit Number</i>										
		19	18	17	...	14	13	...	9	8	...	0
<i>Integration Time Modification</i>		1	1	0	0	0	0	0	0	0	<i>Integration Time LSB [8..0]</i>	
		1	1	0	0	0	0	0	0	1	<i>Integration Time MEDSB [17..9]</i>	
		1	1	0	0	0	0	0	1	0	<i>Integration Time MSB [26..18]</i>	
<i>Calibration Status</i>		1	1	0	0	0	1	x	x	x	x	x
<i>Writing TT Data</i>		1	0	<i>Receiver ID</i>			<i>TT Position</i>			<i>TT Element Value</i>		

Figura 4-5: Configuración del registro *Input Word*. En el último caso, *Receiver ID* es el entero que identifica el número de receptor entre los 16 posibles, *TT Position* es la posición de la Tabla que se va a escribir y *TT Element Value* es el valor a escribir.

La transmisión de datos desde la UC hacia la ALU no requiere de ningún sistema de control del tráfico, ya que la ALU siempre está disponible para recibir datos y ejecuta las operaciones en un ciclo de reloj. No sucede igual con las transmisiones desde la ALU hacia la UC. Puesto que la UC se ejecuta en un microprocesador, no se puede suponer que estará disponible para recibir datos en todo momento ni que procesará los datos en un único ciclo de reloj.

Por este motivo la comunicación ALU → UC se basa en un sistema de interrupciones y confirmaciones formado por tres registros. Los registros implicados en este proceso son: *Output Word*, *INTACK* e *IRQ*.

La información se transmite a través de *Output Word*. Se trata de un registro de 31 bits de longitud dividido en dos segmentos, uno con información que permite identificar el origen de los datos y otro que contiene los datos a transmitir.

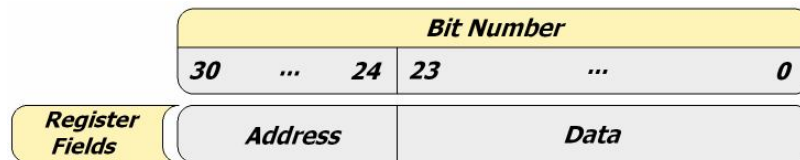


Figura 4-6: Estructura del Registro *Output Word*.

En el momento en que la UC está preparada para recibir un valor de la ALU, fuerza a 1 el valor del registro *INTACK*. Una vez recibida la confirmación, la ALU envía una interrupción a través del registro *IRQ* y deposita los datos en el registro *Output Word*. Cuando la UC recibe la interrupción, fuerza nuevamente un cero en el valor de *INTACK*, evitando así el envío de un nuevo dato desde la ALU. Una vez leída la información contenida en *Output Word*, la UC vuelve a poner a 1 el valor de *INTACK*, permitiendo la llegada de un nuevo dato.

4.2.3 Entradas y salidas

Puesto que la UC ejerce las funciones de elemento de enlace entre el hardware del radiómetro y la interfaz de captura de datos, posee múltiples entradas y salidas desde y hacia ambos. La figura 4-7 muestra las principales entradas y salidas de la UC:

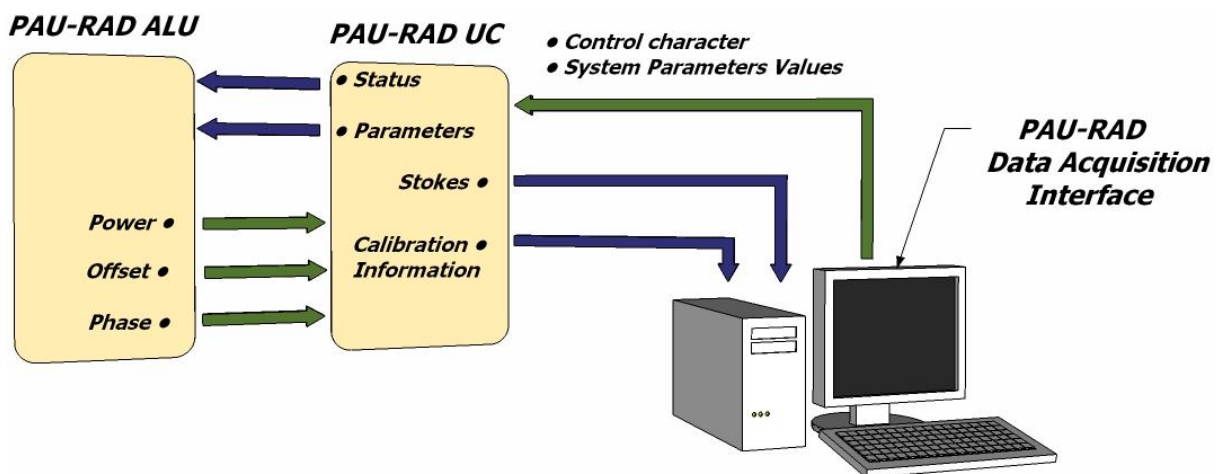


Figura 4-7: Esquematización de las entradas y salidas de la UC. En verde se destacan las entradas, mientras que las salidas se indican en azul.

Como se puede apreciar en la imagen, la UC envía a la Interfaz de adquisición de datos los parámetros de Stokes así como la información de calibración del sistema. Ambos serán representados gráficamente en tiempo real, proporcionando al usuario información sobre los resultados y el estado del sistema. Por su parte, la Interfaz de captura envía a la UC el carácter de control que modifica el estado del sistema y los valores de los parámetros del sistema en caso de que se modifiquen.

Por otra parte, la UC recibe de la ALU la siguiente información:

- **Power:** Cuatro enteros que contienen la potencia de los cuatro canales de cada uno de los receptores. La ALU expresa la potencia de cada canal como un entero de 48 bits. No obstante, el procesador Nios I tan sólo puede trabajar con enteros de 32 bits. Por este motivo la potencia se divide en dos enteros de 24 bits, lo que se traduce en 8 enteros de 24 bits para expresar la potencia de los cuatro canales. Una vez en la UC, los dos enteros de 24 bits se componen para formar un entero de 32 bits, truncando los 16 bits de menor peso.
- **Phase:** Ocho enteros de 32 bits que contienen medidas de similitud. Cuatro medidas de similitud entre la componente en fase del canal de referencia y las componentes en fase del resto de canales; y cuatro medidas de similitud entre la componente en fase del canal de referencia y las componentes en cuadratura del resto de canales.
- **Offset:** Cinco enteros de 8 bits que contienen una estimación del offset introducido por los ADC.

A partir de estos tres tipos de información se obtienen las correcciones necesarias y se calculan los valores de las TT, por tanto estas tres entradas constituyen la información fundamental con la que trabajarán los algoritmos de calibración.

La información que la UC entrega a la ALU es básicamente el estado en el que se encuentra el sistema, dato que se utiliza para conmutar las entradas del radiómetro y los valores de los diferentes parámetros del sistema cada vez que son modificados en la interfaz.

4.2.4 Parámetros del sistema

Es evidente que PAU-RAD constituye un dispositivo complejo tanto por su arquitectura como por su funcionalidad. Como tal, posee gran cantidad de parámetros que determinan tanto su funcionamiento como su rendimiento, todos ellos controlados por la UC. Estos parámetros se han dividido en dos clases: parámetros dinámicos y parámetros estáticos. Como ya se ha mencionado, dada la complejidad de

PAU, el sistema posee infinidad de parámetros, en este apartado se describirán únicamente los más importantes.

4.2.4.1 Parámetros dinámicos

Los parámetros dinámicos son aquellos que admiten modificaciones en tiempo real. Durante el transcurso de una campaña de medidas, el usuario puede modificar estos parámetros a través de la interfaz de captura de datos sin necesidad de detener el instrumento. Esto otorga flexibilidad al instrumento, pero al mismo tiempo complica la interfaz de usuario y el manejo del mismo. Por este motivo la cantidad de parámetros dinámicos se ha reducido al mínimo necesario:

- **Tiempo de integración**

El tiempo de integración es el parámetro más importante del sistema. Determina el tiempo durante el cual el radiómetro integrará la potencia captada por la antena para obtener la información de calibración y los parámetros de Stokes.

El tiempo de integración se almacena en una variable de 27 bits de longitud, y se expresa en función de un número entero de ciclos de reloj de la FPGA. La siguiente tabla establece la relación entre ciclos de reloj y segundos a partir de la frecuencia de trabajo de PAU-RAD (22,98 MHz.):

Tabla 4-1: Equivalencia entre el valor de la variable que determina el tiempo de integración (en ciclos de reloj) y el tiempo de integración (en milisegundos).

System Clocks	Integration Time [ms]
500000	21,75
$1 \cdot 10^6$	43,51
$5 \cdot 10^6$	217,58
$1 \cdot 10^7$	453,16
$5 \cdot 10^7$	2175,8
$1 \cdot 10^8$	4531,6
$1,34218 \cdot 10^8$	5840,64

El primer valor de la tabla corresponde al tiempo de integración mínimo que garantiza una correcta comunicación entre la UC y la interfaz de captura de datos, valor que se ha determinado empíricamente. El último valor corresponde al máximo tiempo de integración seleccionable, es decir: 2^{27} .

- **Tiempo de inter-calibración**

El tiempo de inter-calibración determina el lapso de tiempo durante el cual el sistema permanece en el modo normal proporcionando los parámetros de Stokes a su

salida. Una vez concluido este tiempo, el sistema vuelve al estado de calibración, lo que garantiza la fiabilidad de los datos frente a las derivas del sistema. Al igual que el tiempo de integración se expresa en función de ciclos de reloj.

- **Ángulos de apuntamiento**

Los parámetros β_1 y β_2 corresponden a los ángulos de apuntamiento para cada uno de los dos haces de los que dispone el sistema. Si se introduce un valor cualquiera los dos haces trabajan en modo fijo, orientados según el ángulo deseado; mientras que si el valor introducido es -1, ambos haces comienzan una secuencia de escaneo.

- **Puerto y dirección IP**

Puesto que la comunicación con PAU-RAD se establece a través de una conexión TCP/IP, al inicio de una sesión de medida es necesario proporcionar al sistema tanto la dirección IP de la FPGA como el puerto por el cual se establecerá la comunicación.

4.2.4.2 Parámetros estáticos

Los parámetros estáticos son aquellos cuya modificación se efectúa directamente en el código fuente de la UC. Por tanto, para modificar el valor de uno de éstos parámetros es necesario detener la campaña de medidas y re-compile el código fuente.

- **Número de ciclos de calibración**

A través de este parámetro es posible fijar el número de veces consecutivas que se ejecuta la rutina de calibración durante la secuencia principal. Cuando se inicia el sistema, las señales procedentes de los cuatro canales de cada receptor se adquieren con fases completamente aleatorias. Debido a esto las diferencias de fase entre canales pueden llegar a ser muy elevadas, provocando grandes errores de fase. Para garantizar que los errores residuales se encuentran dentro de los márgenes que permiten el correcto funcionamiento del sistema, es necesario realizar varias calibraciones consecutivas.

- **Número de ciclos de ruido correlado**

Este parámetro determina el número de veces que se ejecuta la rutina de calibración mediante temperatura de referencia interna con ruido correlado.

- **Número de ciclos de ruido incorrelado**

Controla el número de ejecuciones de la rutina encargada de realizar la calibración por temperatura de referencia interna inyectando ruido incorrelado.

4.2.5 Estados de la *Secuencia Principal*

A continuación se describen con más detalle los diferentes estados que componen la Secuencia Principal de funcionamiento de la UC. Los algoritmos de calibración se abordarán con detalle en el siguiente capítulo. Por este motivo, en este capítulo tan sólo se esbozarán los procesos básicos que componen los estados encargados del proceso de calibración.

4.2.5.1 *Init*

Init es siempre el primer estado en ejecutarse en el instante en que se pone en funcionamiento el radiómetro. Se implementa en la librería de nombre: *pau_init.h* y se compone fundamentalmente de una subrutina: *r_init()*. Cumple esencialmente dos funciones: en primer lugar inicializar los valores de todas las variables del sistema al inicio de una nueva campaña de medidas, y en segundo lugar evaluar las secuencias de control que se envían desde la interfaz de usuario y efectuar los cambios oportunos.

La subrutina *r_init()* se basa en una sentencia *switch()* que evalúa la variable global *v_lan_in_control*, que contiene el carácter de control enviado desde la interfaz de usuario. Este *switch()* contempla los siguientes casos:

- **Reset:** Constituye el caso por defecto y, como su propio nombre indica, lleva a cabo un reinicio de PAU. En él se establecen los valores iniciales de todas las variables del sistema (tiempo de integración, tiempo entre calibraciones, errores de fase y ganancia, parámetros del filtrado...) y se realiza la llamada a la función encargada de establecer la conexión TCP/IP con la interfaz de captura de datos. Llegados a este punto el sistema permanece a la espera de que se establezca conexión desde la interfaz de usuario. Una vez establecida la conexión se actualiza la variable de estado y se devuelve el control de la ejecución al bucle principal.
- **Modificación del tiempo de integración:** Se ejecuta en caso de que desde la interfaz de usuario se modifique el tiempo de integración. Básicamente se modifica el valor de la variable que controla el tiempo de integración (*v_settime*) y se llevan a cabo las tres escrituras en la ALU que son necesarias para establecer el nuevo tiempo de integración.
- **Modificación del tiempo de intra-calibración:** Se accede a este caso cuando el usuario modifica el tiempo entre calibraciones. Esencialmente se establece el nuevo valor de la variable *v_timecal*, y se vuelve al estado que se estuviese ejecutando anteriormente. Este caso también contempla la entrada al modo de validación Closed Loop, del cual se hablará más adelante.

- **Modificación del ángulo de apuntamiento β_1 :** Cuando el usuario introduce un nuevo valor para el ángulo de apuntamiento del haz 1 se entra en este caso. Se establece el nuevo valor de ángulo de apuntamiento en la variable *v_beta.B1* y a continuación se pasa al estado de calibración, en el que se determinarán las nuevas tablas trigonométricas para el ángulo seleccionado. En caso de que el valor introducido por el usuario sea -1, se entra en modo escaneo, y se realiza un barrido del ángulo de apuntamiento del haz.
- **Modificación del ángulo de apuntamiento β_2 :** Es el mismo caso que el anterior, pero para el ángulo de apuntamiento del segundo haz. En este caso la variable a modificar es *v_beta.B2*.
- **Activación del modo debug:** En este caso se lleva a cabo la entrada al estado debug. Se trata de un estado interno (de ahí que no se haya incluido en el diagrama de estados general) que se emplea para verificar la comunicación entre la UC y la ALU. En el modo debug se lleva a cabo la transmisión desde la UC a la ALU de los 16 valores que componen una tabla trigonométrica, así como de los tres 3 enteros de 9 bits que componen el tiempo de integración. Si los valores recibidos concuerdan con los enviados es posible concluir que la comunicación es correcta.
- **Activación/desactivación del modo Open Loop:** Se accede a este caso cuando el usuario pulsa el botón de activación del modo Open Loop. En él se modifica el valor de la variable status en función del estado anterior del sistema. Si se accede a este caso desde cualquier estado diferente a Open Loop, se activa el modo Open Loop; en caso contrario se desactiva.
- **Activación/desactivación del modo Phase Sweep:** Este caso implementa el mecanismo de activación/desactivación del modo Phase Sweep, que es el mismo que el descrito para el modo Open Loop.
- **Activación/desactivación del modo Reduced Calibration:** El último caso que contempla la sentencia *switch()* controla la activación y desactivación del modo Reduced Calibration. El protocolo es exactamente el mismo que el de los casos anteriores.

Adicionalmente a la subrutina *r_init()*, la librería *pau_init.h* que implementa este estado, dispone de una subrutina llamada *visualitzar_dades()*, que se encarga de presentar en pantalla los valores asignados a los principales parámetros del sistema.

4.2.5.2 Calibration

Como su propio nombre indica es el estado en el que se ejecutan los algoritmos de calibración, encargados de corregir los errores de fase y ganancia del sistema. El

estado de calibración se implementa en la librería *pau_calibracio.h*, que se compone de una única subrutina llamada *r_calibracio()*.

Cuando se accede a este estado, se ejecuta la subrutina *r_calibracio()*, en la que se lleva a cabo la siguiente secuencia de tareas:

- En primer lugar se procede a la declaración e inicialización del conjunto de variables locales de la subrutina. En este conjunto se incluyen diversas variables que actúan como contadores y auxiliares y las variables que identifican el tipo de datos a transmitir. El resto de variables que se emplean en el proceso de calibración se declaran de manera global en la librería *pau.h* y se inicializan en el estado *Init*.
- Se aumenta el tiempo de integración del sistema. De esta forma se consigue incrementar la precisión de las medidas a partir de las cuales se efectuarán las estimaciones de los errores.
- Si es la primera vez que se accede al estado de calibración, se escriben las tablas trigonométricas de referencia. Las TT de referencia contienen los valores "ideales", es decir aquellos valores que se obtendrían en caso de no haber errores.
- A continuación se configura la ALU para que la entrada del sistema sea la de ruido correlado (fuente de ruido) y se realiza una primera captura de medidas que es desechada. El motivo por el cual se descarta el primer conjunto de medidas es para evitar posibles errores introducidos por la transición de los selectores de entrada.
- Una vez descartada la primera captura, se procede a realizar una segunda captura que esta vez será almacenada. Se obtiene así el conjunto de datos que se emplearán para estimar los errores de amplitud y fase mediante la aplicación de los algoritmos de calibración que se describen en el capítulo 5.
- Los errores estimados son filtrados mediante un filtro de Kalman para reducir su varianza, tal y como se expone en el capítulo 6.
- Finalmente, a partir de los errores filtrados se calculan cada una de las 16 posiciones de las tablas trigonométricas de cada receptor y se envían los resultados a la ALU. Simultáneamente se envían los resultados de la calibración a la interfaz de captura de datos, de forma que el usuario pueda comprobar la evolución de los algoritmos.

Este proceso se itera tantas veces como indique la variable local *iteraciones*. De esta forma se consigue reducir el error residual final, mejorando la precisión de los resultados. Experimentalmente se ha puesto de manifiesto que 10 iteraciones del

proceso de calibración garantizan la reducción de los errores de fase y amplitud hasta ubicarlos dentro de los límites que garantizan el correcto funcionamiento del sistema.

Transcurrido el número de iteraciones fijado, se actualiza el valor de la variable de control de estado, de manera que el siguiente estado en ejecutarse sea el estado *Normal Mode*, en el que se obtienen los parámetros de Stokes.

4.2.5.3 *Internal Reference Correlated*

El estado *Internal Reference Correlated* implementa una calibración por temperatura de referencia interna. Esta calibración utiliza dos niveles diferentes de potencia para determinar los coeficientes de corrección de fase y ganancia de cada canal. Ambos niveles de potencia proceden de la fuente de ruido del sistema, de ahí que se denomine referencia interna correlada, ya que la señal de entrada es la misma para los cuatro canales.

La librería que contiene la implementación de este estado es: *pau_internal_reference.h*. Esta librería se compone de una única función, *internal_reference()*, que lleva a cabo las siguientes tareas:

- En primer lugar se envía a la ALU el valor que la configura en modo *Internal Reference Correlated*. Esto implica seleccionar como entrada la fuente de ruido correlado del sistema.
- El proceso se iterará durante un número de veces determinado por la variable local *iterations*. Durante la primera mitad de las iteraciones se configura la fuente de ruido para que entregue una potencia de ruido con una temperatura de ruido equivalente denominada T_{MAX} ; mientras que durante la segunda mitad de iteraciones se disminuye la potencia, dando lugar a una temperatura de ruido equivalente T_{MIN} . Las medidas de potencia obtenidas para T_{MAX} y T_{MIN} se promedian y se almacenan.
- A partir de las medidas de potencia promediadas para las dos temperaturas de referencia se aplican los algoritmos que determinan los coeficientes de corrección de ganancia de cada canal. Los coeficientes obtenidos se envían a la interfaz de captura de datos para su representación.
- Se calculan las posiciones de la TT aplicando los coeficientes obtenidos y se envían a la ALU.
- Finalmente se actualiza la variable de estado, pasando al siguiente estado del diagrama en caso de que se haya alcanzado el número de iteraciones establecido por el usuario.

4.2.5.4 *Internal Reference Uncorrelated*

Este estado tiene como función determinar el offset de los parámetros de Stokes. A diferencia de lo que ocurría en el estado anterior, en *Internal Reference Uncorrelated* se emplea ruido incorrelado procedente de una resistencia interna de 50 Ω y no de la fuente de ruido.

La implementación de este estado se encuentra en la librería *pau_internal_reference_uncorrelated.h* y su ejecución corre a cargo de la subrutina *r_internal_reference_uncorrelated()*. El proceso es similar al que se sigue en el estado *Normal Mode* (ver siguiente punto) para obtener los parámetros de Stokes. La principal diferencia es que, en este caso, se selecciona como entrada la resistencia de 50 Ω en vez de la entrada de antena.

4.2.5.5 *Normal Mode*

En el estado *Normal Mode* se lleva a cabo el procedimiento mediante el cual se obtienen los valores de los cuatro parámetros de Stokes a partir de la potencia de ruido captada por la antena. El código que implementa este estado se encuentra en la librería *pau_normal.h*, que se compone de una única función: *r_normal()*, encargada de todo el proceso.

Durante la ejecución de la subrutina *r_normal()*, tiene lugar el siguiente proceso:

- Al acceder al estado *Normal Mode* es necesario contabilizar de alguna manera el tiempo transcurrido desde la última calibración. Para este cometido se emplea la variable *cal_cont*, que actúa como contador del número de veces que se ha ejecutado la subrutina. Por tanto, el primer paso es incrementar este contador. También se incrementa la variable *scan_cont*, que controla el tiempo entre cambios de ángulo de apuntamiento en modo scan en caso de que éste esté activado.
- Se configura la ALU para que trabaje en modo *Normal Mode* a través de una escritura en el registro *Input Word* y se habilitan las interrupciones procedentes de la ALU.
- Una vez habilitadas las interrupciones de la ALU, se entra en un bucle *do while* en el que se capturan los datos procedentes de la ALU según el protocolo de interrupción/confirmación descrito en el apartado 4.2.2.1. En total se capturan 13 valores: 4 valores de potencia, divididos en dos enteros de 24 bits cada uno que representan los parámetros de Stokes y 5 valores de offset de 24 bits cada

uno. Cada vez que se recibe uno de los valores, éste es reenviado a la interfaz de captura de datos para su visualización.

- o Finalmente se actualiza la variable de estado en función de los contadores *cal_cont* y *scan_cont*. Si el contador *cal_cont* alcanza el valor fijado por el usuario, el siguiente estado en ejecutarse será *Calibration*. De esta forma se corrigen las posibles derivas de calibración que haya podido experimentar el dispositivo durante la obtención de medidas. En caso de que el modo scan esté activado, si el contador *cal_cont* es igual al tiempo entre ángulos de apuntamiento establecido, se modifica el ángulo de apuntamiento antes de proceder a la siguiente captura de datos.

4.2.6 Estados de las *Secuencias de Validación*

Al margen de los estados que componen la *Secuencia Principal* existe una serie de estados diseñados para verificar el correcto funcionamiento tanto del hardware como del software del sistema. En este apartado se incluye la descripción de algunos estados que no se muestran en el diagrama de la figura 4-1. Estos estados fueron omitidos del diagrama general de estados debido a que se crearon para efectuar pruebas puntuales que no se incluyen dentro del funcionamiento normal del sistema.

4.2.6.1 *Phase Sweep*

El estado *Phase Sweep* (barrido de fase) fue ideado para comprobar el comportamiento en fase de los receptores. Se halla implementado en la librería *barrido_fase.h*, que contiene una única subrutina encargada de todo el proceso: *r_barrido_fase()*.

Durante el proceso de calibración se estiman los errores de fase de cada canal respecto al canal de referencia para corregirlos. En la ejecución del *Phase Sweep* sin embargo, se introducen errores de fase conocidos en una secuencia que va desde los 0° hasta los 360° de error, en saltos fijados por el usuario. Para cada valor de fase introducido se determinan las componentes real e imaginaria (denominadas μ_r y μ_i respectivamente) de la correlación entre el canal de referencia y cada uno de los otros tres canales. Los valores obtenidos se envían a la interfaz de captura de datos, donde se representan en el plano complejo.

El resultado esperado de la representación de μ_r en función de μ_i , en una secuencia completa de 360° es un juego de tres circunferencias perfectas (una para cada canal) concéntricas. Las deformidades en las circunferencias indican un

exceso/defecto de nivel de correlación en el canal, mientras que un desplazamiento en una de las circunferencias señala la presencia de un offset de correlación.

Durante la ejecución de la rutina de barrido de fase tiene lugar la siguiente secuencia de procesos:

- Inicialización de las variables de barrido: *salto_fase*, *ciclos_salto*, y *entrada*. La primera determina el valor en grados sexagesimales de los saltos, de manera que cuanto menor es el valor, mayor es la precisión de los resultados. *Ciclos_salto* determina el número de veces a promediar los resultados para un valor de fase en concreto, lo que permite mejorar la precisión. Por último *entrada* permite seleccionar entre la entrada de calibración (si vale 1) o la entrada de antena (si vale 2).
- Al inicio de cada ciclo de barrido se escriben las TT de referencia y se aumenta el tiempo de integración para mejorar la precisión. Se envían también los comandos necesarios a la ALU para fijar la entrada en concordancia con el valor de *entrada*.
- A partir de este punto se inicia el proceso de barrido. En cada iteración se incrementa el error de fase según el valor de *salto_fase*, y se envían a la ALU las TT que corrigen dicho error. A continuación se lleva a cabo una captura de datos a partir de la cual se calculan los valores de μ_r y μ_i .
- Finalmente los valores de μ_r y μ_i obtenidos se envían a la interfaz de captura de datos para su representación.

La ejecución del Phase Sweep continúa hasta que el usuario la desactiva en la interfaz.

4.2.6.2 Open Loop

Cada receptor incorpora cuatro potenciómetros destinados a controlar la ganancia de cada uno de los canales. Al inicio de una sesión de medidas es recomendable ajustar estos potenciómetros de manera que las potencias de los cuatro canales queden igualadas, facilitando así la tarea de los algoritmos de calibración.

La finalidad del estado *Open Loop* (calibración en lazo abierto) no es otra que la de facilitar la tarea de ajuste de los potenciómetros y determinar la estabilidad de cada receptor. Durante la ejecución del estado *Open Loop* se adquieren de forma continuada datos procedentes de la entrada de calibración y se envían a la interfaz sin aplicar ningún tipo de calibración. Este estado se implementa en la librería *calibracio_validacio.h*, librería que se compone de la subrutina *r_calibracio_validacio()*.

Mediante la representación gráfica de las potencias obtenidas resulta sencillo ajustar de forma manual los cuatro potenciómetros de cada receptor para igualar las potencias de los canales.

4.2.6.3 *Closed Loop*

El estado *Closed Loop* (lazo cerrado) tiene como función comprobar el correcto funcionamiento de los algoritmos de calibración y filtrado del sistema así como su bondad. Se implementa en la librería *calibracio_validacio2.h* y se compone de una única subrutina de nombre *r_calibracio_validacio2()*. Durante la ejecución de este estado se lleva a cabo un proceso continuado de calibración, lo que permite poner a prueba los diferentes algoritmos de calibración y determinar sus tiempos de respuesta y resultados finales.

La secuencia de ejecución de la función *r_calibracio_validacio()* es la que se describe a continuación:

- Al inicio de un ciclo de calibración en lazo cerrado, se envían a la ALU las TT de referencia y se selecciona la entrada de calibración.
- Se realiza una captura de datos de calibración, a partir de los cuales se estiman los errores de fase y ganancia.
- Se procede al filtrado de los errores obtenidos en caso de que se quieran testear los algoritmos de filtrado.
- Una vez obtenidos los errores, se determinan las posiciones de las TT que corrigen dichos errores y se envían a la ALU. Al mismo tiempo, los errores se envían a la interfaz de captura de datos, donde serán representados.

Este proceso se itera de forma continua (a excepción del primer paso, que sólo se lleva a cabo en la primera iteración) hasta que el usuario desactiva el estado. De esta forma los errores se reducen hasta llegar al mínimo permitido por el algoritmo, permitiendo evaluar la bondad y velocidad de los algoritmos.

4.3 INTERFAZ DE CAPTURA DE DATOS

4.3.1 Descripción

Del mismo modo en que la UC actúa como nexo de unión entre el radiómetro y el ordenador, la interfaz de captura de datos constituye el mecanismo de

comunicación final entre el usuario y PAU-RAD. Su función es doble: por un lado se encarga de recoger los datos obtenidos por el radiómetro y entregarlos al usuario al mismo tiempo que los almacena en memoria; y por otro se encarga de transmitir los comandos introducidos por el usuario a la ALU.

Para el desarrollo de la interfaz de captura de datos se empleó como plataforma Matlab. Así pues la implementación completa de la interfaz de captura de datos está comprendida dentro de un script Matlab.

En la figura 4-8, se muestra el aspecto de la interfaz original:

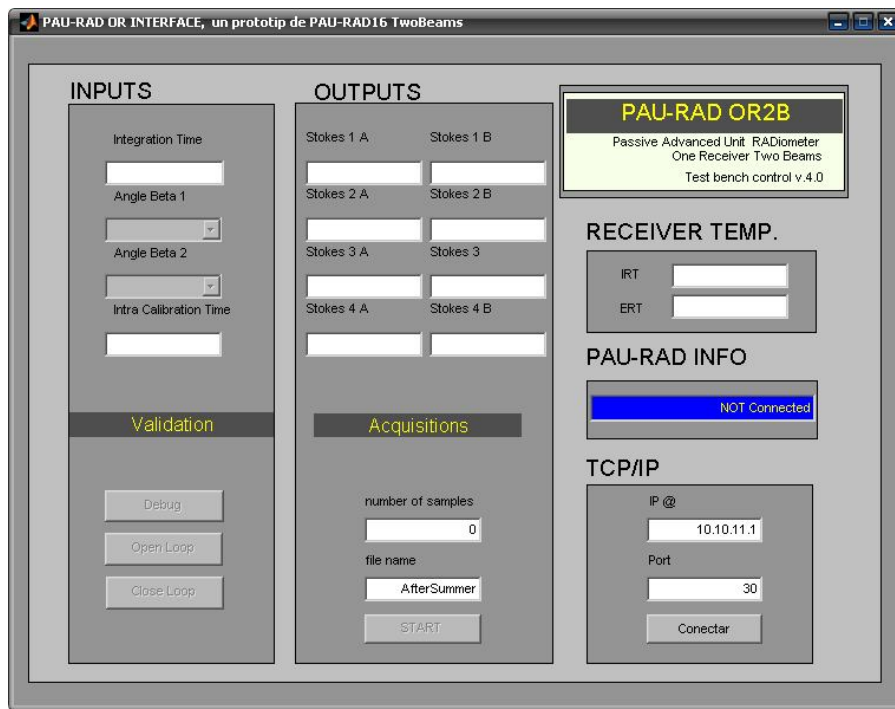
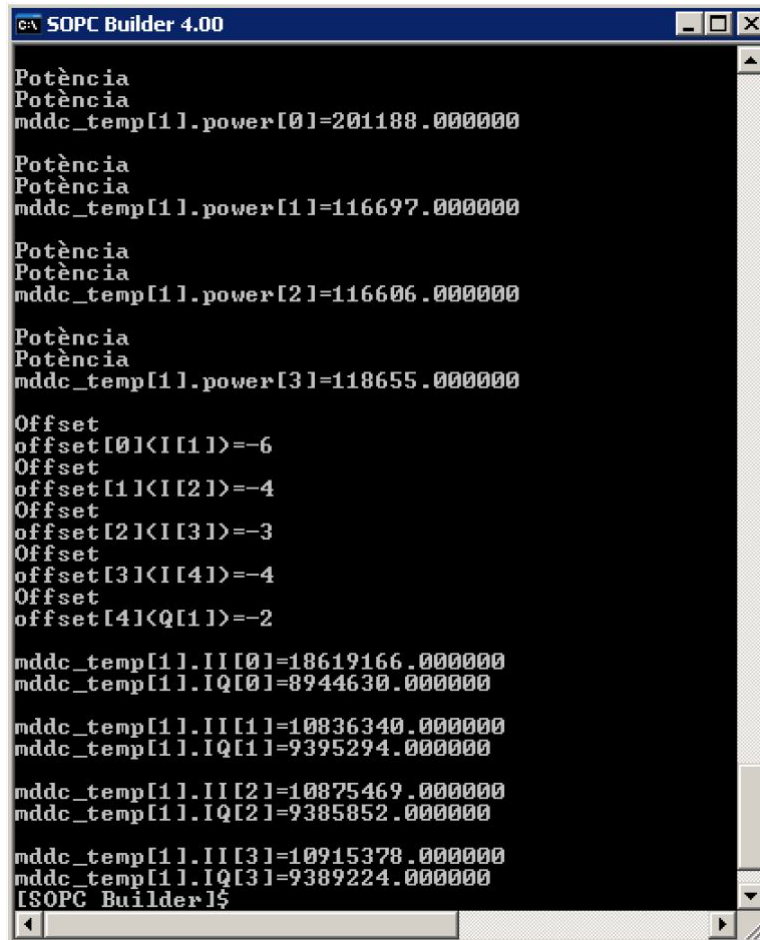


Figura 4-8: Aspecto que presentaba la versión de la interfaz de captura de datos disponible inicialmente.

En esta versión de la interfaz, la presentación de los datos se limita a los cuatro parámetros de Stokes, representados en forma numérica. Esto dificulta la interpretación de los resultados, especialmente si se tiene en cuenta que los valores de los parámetros de Stokes varían a medida que se obtienen nuevas muestras. Para realizar una inspección detallada es necesario detener el proceso de adquisición de datos, abrir el archivo que contiene los datos almacenados y representarlos mediante un segundo script Matlab creado ex profeso. Los datos de calibración del radiómetro no se muestran en la interfaz, sino que son almacenados directamente y su análisis requiere, por tanto, de un nuevo script Matlab específico.

Para paliar de alguna manera la carencia de información en tiempo real, durante las pruebas del radiómetro se complementa el uso de la interfaz de captura de

datos con la consola *Nios SDK Shell*. La consola *Nios SDK Shell* es una utilidad de desarrollo de software de Altera que proporciona un entorno estilo UNIX en forma de línea de comandos y que incorpora, entre otras, las herramientas que se emplean para la compilación, carga y ejecución del código de la unidad de control en el procesador.



```
C:\> SOPC Builder 4.00

Potència
Potència
mddc_temp[1].power[0]=201188.000000

Potència
Potència
mddc_temp[1].power[1]=116697.000000

Potència
Potència
mddc_temp[1].power[2]=116606.000000

Potència
Potència
mddc_temp[1].power[3]=118655.000000

Offset
offset[0]<I[0]>=-6
Offset
offset[1]<I[1]>=-4
Offset
offset[2]<I[2]>=-3
Offset
offset[3]<I[3]>=-4
Offset
offset[4]<Q[1]>=-2

mddc_temp[1].II[0]=18619166.000000
mddc_temp[1].IQ[0]=8944630.000000

mddc_temp[1].II[1]=10836340.000000
mddc_temp[1].IQ[1]=9395294.000000

mddc_temp[1].II[2]=10875469.000000
mddc_temp[1].IQ[2]=9385852.000000

mddc_temp[1].II[3]=10915378.000000
mddc_temp[1].IQ[3]=9389224.000000
[SOPC Builder]$
```

Figura 4-9: Imagen de la consola *Nios SDK Shell* mostrando información adquirida por el radiómetro en tiempo real. En la imagen se pueden ver los datos de potencia, offset y correlación para los cuatro canales durante la etapa de calibración.

La consola *Nios SDK Shell* se comunica con la FPGA a través del puerto serie del ordenador, mientras que los datos que se transmiten a la interfaz de captura de datos viajan a través del puerto Ethernet. De esta forma, mediante el uso de sentencias *printf()* en el código de la UC, es posible visualizar en la consola los valores de las variables que se quieran inspeccionar en tiempo real, tal y como se muestra en la figura 4-9. No obstante este sistema alternativo sigue siendo muy poco práctico a la hora de extraer conclusiones de los resultados.

A medida que se avanzaba en el perfeccionamiento de la calibración del sistema, se hizo patente la necesidad de perfeccionar también la forma en que los resultados se presentaban al usuario. Uno de los objetivos que se fijaron fue el de disponer de una representación gráfica en tiempo real de los datos obtenidos, lo que

permitiría un análisis mucho más cómodo de los resultados. Este hecho, unido a la necesidad de adaptar la interfaz a los cambios efectuados en el código de la UC, dio lugar a la creación de una nueva versión de la interfaz, cuyo aspecto se muestra en la siguiente figura:

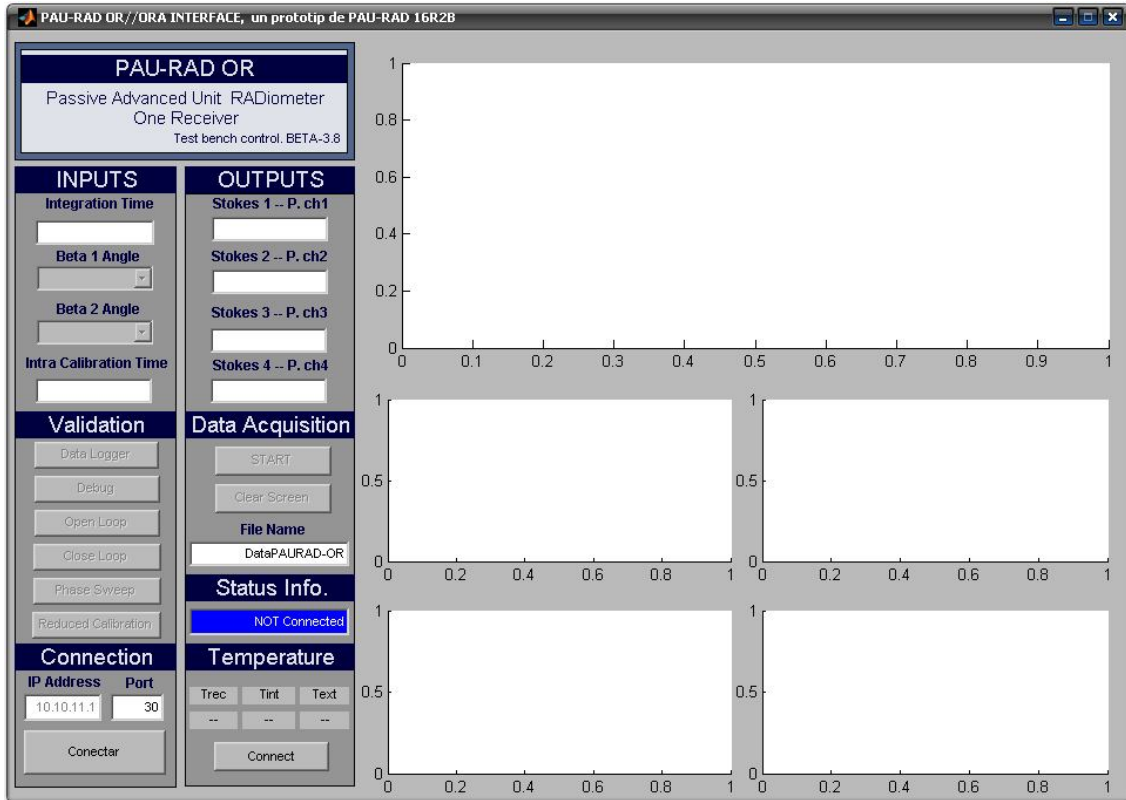


Figura 4-10: Aspecto que presenta la interfaz de captura de datos antes de establecer conexión con el radiómetro.

Como se puede apreciar la interfaz de captura de datos presenta dos zonas claramente diferenciadas: la zona de control (a la izquierda de la imagen); y la zona gráfica.

La zona de control se compone de las siguientes secciones:

- **Inputs:** Contiene los cuadros de inserción de texto en los que el usuario puede introducir los cuatro parámetros dinámicos fundamentales del sistema: tiempo de integración, ángulos de apuntamiento y tiempo de intra calibración.
- **Outputs:** Muestra el valor numérico de los cuatro parámetros de Stokes cuando el sistema se encuentra en estado *Normal Mode*.
- **Validation:** Agrupa los botones de activación/desactivación de los estados de validación.

- **Data Acquisition:** Contiene el botón que inicia la secuencia de captura de datos, un botón que borra el contenido de los paneles de visualización y un cuadro de inserción de texto en el que introducir el nombre del archivo en el que se almacenarán los datos.
- **Status Info:** Proporciona información sobre el estado actual del sistema.
- **Connection:** Dispone de dos cuadros de inserción de texto para introducir la IP y el puerto, y un botón que establece la comunicación con la FPGA.
- **Temperature:** Muestra las temperaturas de tres sensores, dos en el interior del radiómetro y un tercero en el exterior.

Por su parte, la zona gráfica dispone de cinco paneles de representación. El panel superior está destinado a mostrar la progresión de los parámetros de Stokes en modo *Normal Mode*, los siguientes paneles muestran la amplitud y la fase respectivamente durante el estado de calibración y los dos paneles inferiores se utilizan a modo de paneles auxiliares para mostrar diferentes tipos de información según se necesite.

4.3.2 Arquitectura

Como se ha mencionado, todo el código que compone la interfaz de captura de datos se encuentra íntegramente en un solo archivo. La figura 4-11 muestra un esquema del proceso de ejecución de la interfaz de captura de datos:

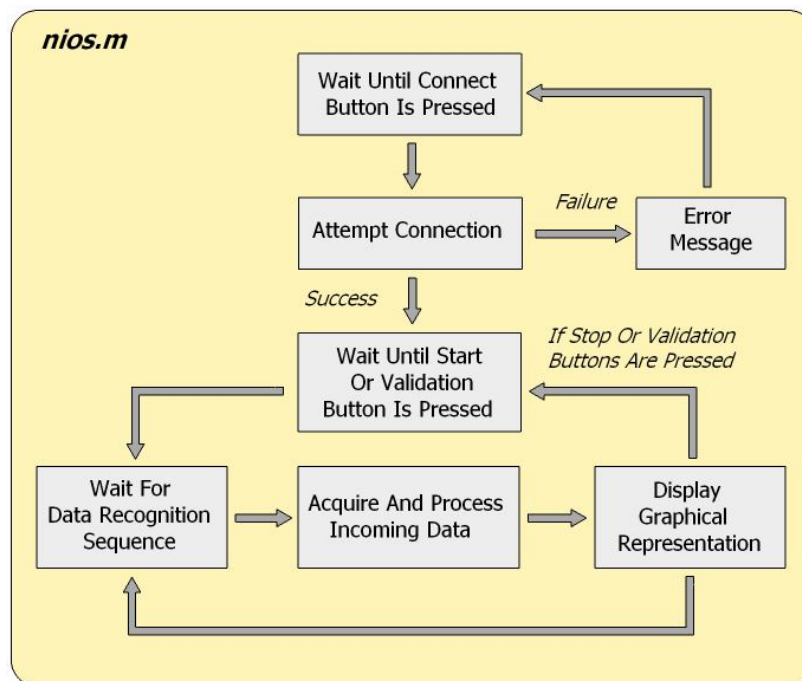


Figura 4-11: Proceso de ejecución del script de la interfaz de captura de datos esquematizado.

El código está estructurado en forma de múltiples funciones de “callback”, funciones que se ejecutan en el momento en que el usuario activa alguno de los botones o introduce alguna modificación en la zona de control.

Inicialmente, la interfaz se mantiene a la espera de que el usuario de la orden de establecimiento de conexión. Una vez dada la orden, se procede a crear la conexión con la FPGA. En caso de que el software de la UC no se halle arrancado y a la espera de recibir la conexión, se muestra un mensaje de error y se vuelve al estado inicial.

Con la conexión establecida, se espera a que el usuario active alguno de los estados de validación, o bien inicie la adquisición de datos. A partir de este punto, se entra en un bucle en el que se espera la llegada de una secuencia de reconocimiento del tipo de datos que va a enviar la UC. En función de dicha secuencia, se adquieren, almacenan y procesan los datos según su naturaleza (parámetros de Stokes, datos de calibración...).

Finalmente, los datos obtenidos se representan gráficamente en uno de los cinco paneles gráficos en función de su naturaleza y de la secuencia de ejecución. Con el fin de mantener una “ventana” de datos de cierta longitud que permita referenciar los resultados actuales a los obtenidos anteriormente, los datos obtenidos en cada iteración se almacenan en la primera posición de un vector de 100 posiciones de longitud. En la siguiente iteración, se desplaza una posición el dato obtenido en la iteración anterior y se coloca el nuevo dato en la primera posición, y así sucesivamente en cada iteración. De esta forma se mantienen las últimas 100 muestras adquiridas en pantalla en todo momento.

Este bucle se mantiene hasta que el usuario aprieta el botón STOP o bien activa alguno de los modos de validación.

4.3.2.1 Mecanismo de sincronización con la UC

Una de las particularidades de la interfaz de captura de datos es que, a diferencia de lo que ocurre con la UC, Matlab carece de un sistema de interrupciones que permitan sincronizar la transferencia de información desde la UC hacia la interfaz.

Con el fin de poder sincronizar la interfaz con la UC, se adoptó un protocolo de transmisión de secuencias de reconocimiento que permitiesen a la interfaz identificar el tipo de datos que la UC envía. Las secuencias están formadas por dos enteros de un byte de longitud. El primero de los bytes siempre tiene por valor 5, valor que identifica la llegada de datos procedentes de la UC. El segundo byte contiene el valor específico que identifica el estado en el que se encuentra la UC y por tanto el tipo de datos que transmitirá.

La tabla 4-2 muestra la correspondencia entre los valores de las secuencias y el tipo de datos:

Tabla 4-2: Valores de las secuencias de reconocimiento en función del tipo de datos a transmitir.

Tipo de datos	Valor
Inicio de transmisión de datos	5
Parámetros de Stokes	85
Datos de Calibración/Open Loop	170
Datos Internal Reference Correlated	130
Datos Internal Reference Uncorrelated	135
Datos Close Loop	125
Datos Phase Sweep	120

De esta forma, cuando la interfaz recibe el byte de inicio de transmisión de datos, evalúa el segundo byte, y en función de su valor se ejecuta el conjunto de instrucciones que almacenan, procesan y representan gráficamente los datos de la forma apropiada.

La comunicación en el sentido contrario (transmisión de datos desde la interfaz hacia la UC) no requiere ningún tratamiento especial, ya que el procesador Nios dispone de un sistema de interrupciones que hace innecesario el uso de un sistema de sincronización. Simplemente se envía el carácter de control que se evaluará en el estado *Init* de la UC, seguido del valor del parámetro que se quiera modificar si se da el caso.

4.3.3 Comportamiento en función de la secuencia de ejecución.

Para terminar se mostrará con ayuda de algunas capturas el comportamiento de la interfaz en función de la secuencia que se esté ejecutando.

- **Secuencia Principal**

El aspecto que presenta la interfaz durante la ejecución de la Secuencia Principal es el que se muestra en la figura 4-12:

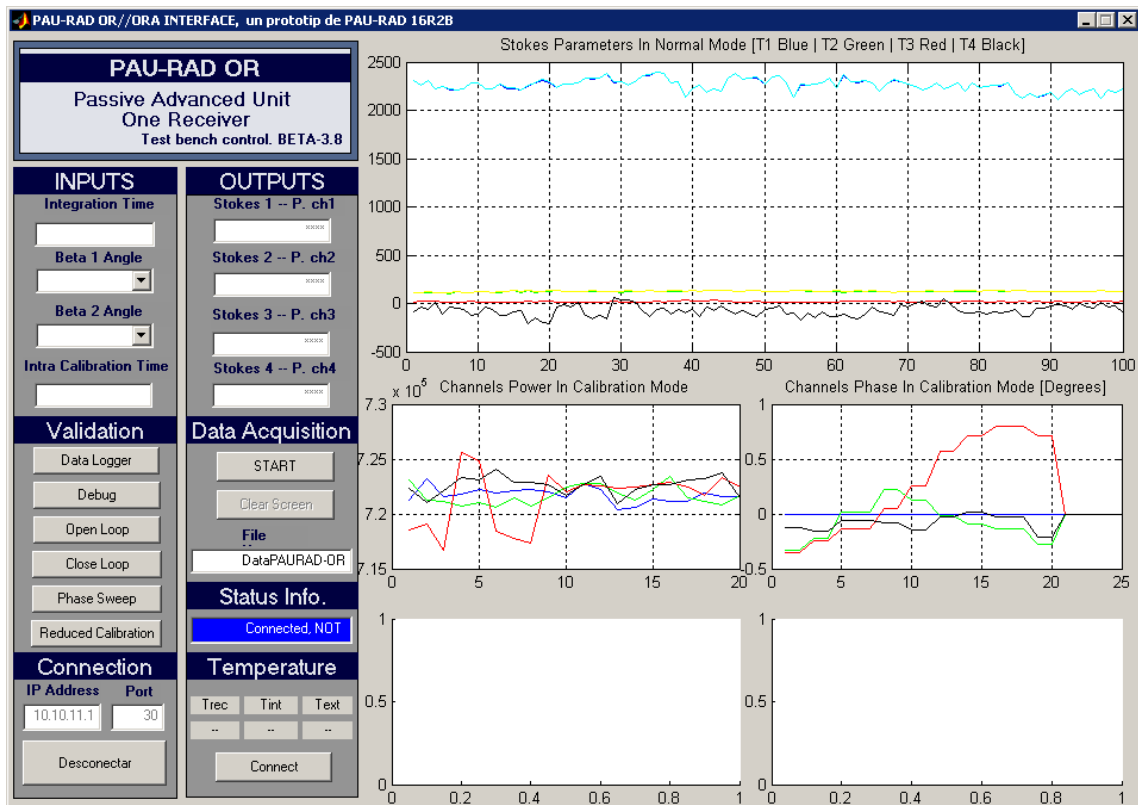


Figura 4-12: Aspecto que presenta la interfaz de captura de datos durante la ejecución de la Secuencia Principal.

En la figura se puede apreciar el resultado de los parámetros de Stokes en el panel superior (cada color representa un canal). Los dos paneles situados debajo del panel principal muestran la potencia de cada canal durante la calibración (izquierda) y el error de fase residual durante la calibración (derecha). En esta captura en concreto, los dos paneles inferiores permanecen desactivados, aunque habitualmente se emplean para mostrar el offset de los parámetros de Stokes obtenido en el estado *Internal Reference Uncorrelated* y la diferencia entre éste offset y los parámetros obtenidos.

- **Secuencia Phase Sweep**

Durante la ejecución de una secuencia de barrido de fase, la interfaz muestra el siguiente aspecto:

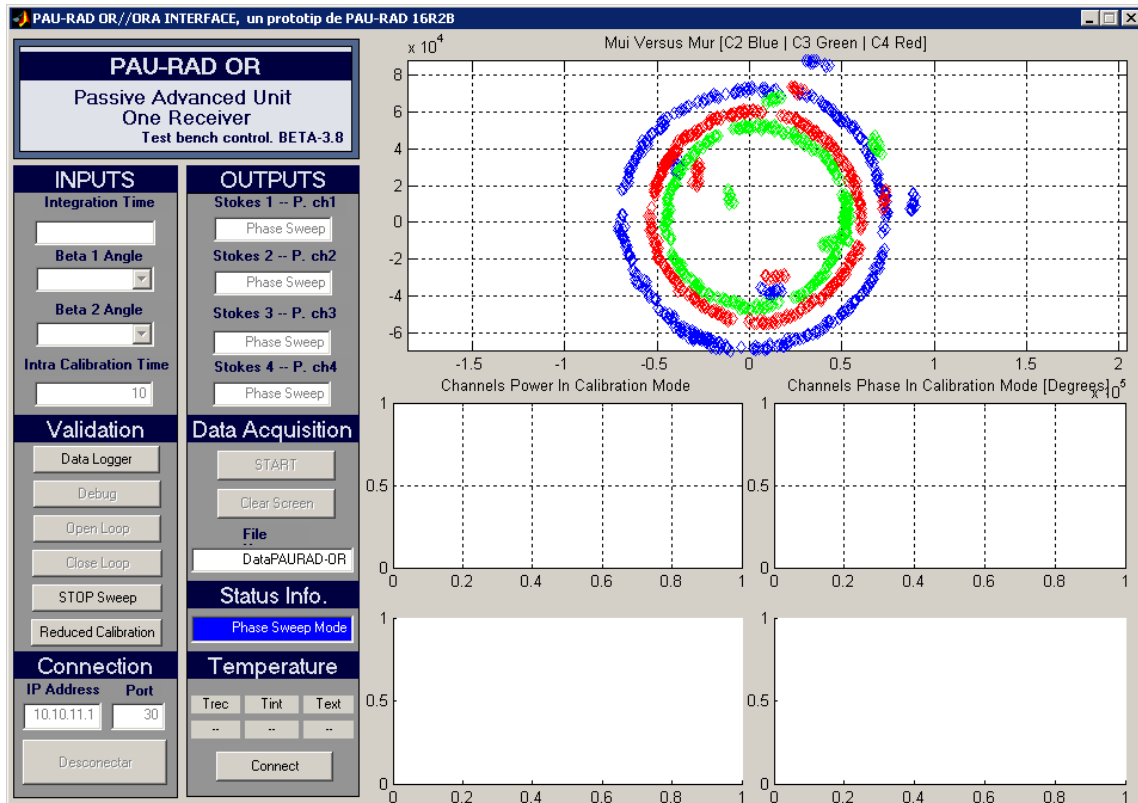


Figura 4-13: Captura de la interfaz una vez completada la ejecución de una secuencia *Phase Sweep*.

En cada iteración del barrido de fase se representa el resultado de la correlación entre el canal de referencia y los otros tres canales en el plano complejo. Cada una de las marcas representa una iteración, de manera que transcurrido un ciclo de 360° , la superposición de todas las marcas da como resultado las circunferencias de colores que se ven en la figura (cada color representa un canal). Esto permite detectar anomalías en los receptores tales como niveles incorrectos de correlación o presencia de offset.

En el caso de esta captura en concreto se pueden observar anomalías en los puntos que corresponden a desfases entre el canal de referencia y el resto de canales de 0° , 90° , 180° y 270° . Gracias a esto se pudo determinar la estrecha relación existente entre el nivel de potencia de la entrada de calibración y el resultado de la calibración, así como mejorar un aspecto del hardware que dificultaba el proceso de calibración.

○ **Secuencia Open Loop**

La figura 4-14 muestra el comportamiento de la interfaz de captura de datos durante la ejecución de una secuencia Open Loop:

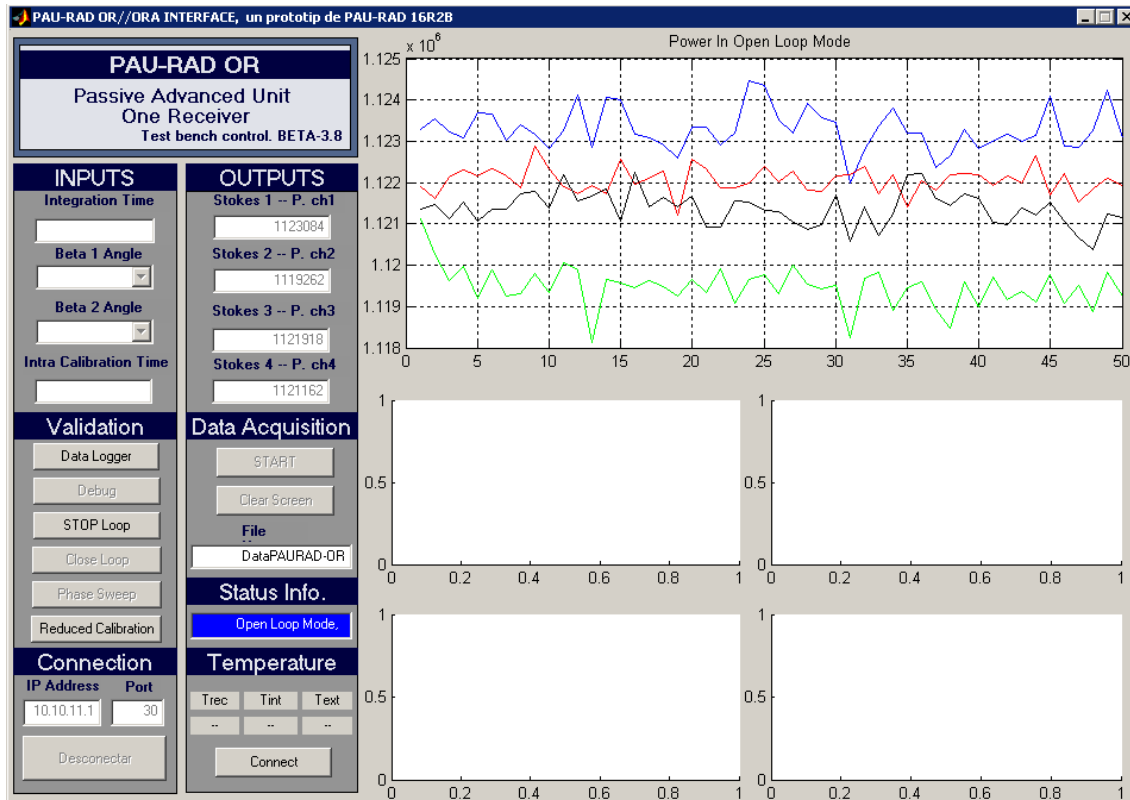


Figura 4-14: Captura de la interfaz durante la ejecución de una secuencia Open Loop.

En el panel principal se puede observar la potencia de calibración de los cuatro canales una vez ajustados manualmente los cuatro potenciómetros que determinan la ganancia de la etapa de RF.

○ **Secuencia Closed Loop**

La finalidad de la secuencia *Closed Loop* es la de determinar la bondad de los algoritmos de calibración y filtrado, por ello durante su ejecución se muestran las potencias de calibración así como los errores de fase residuales tal y como se muestra en la figura 4-15:

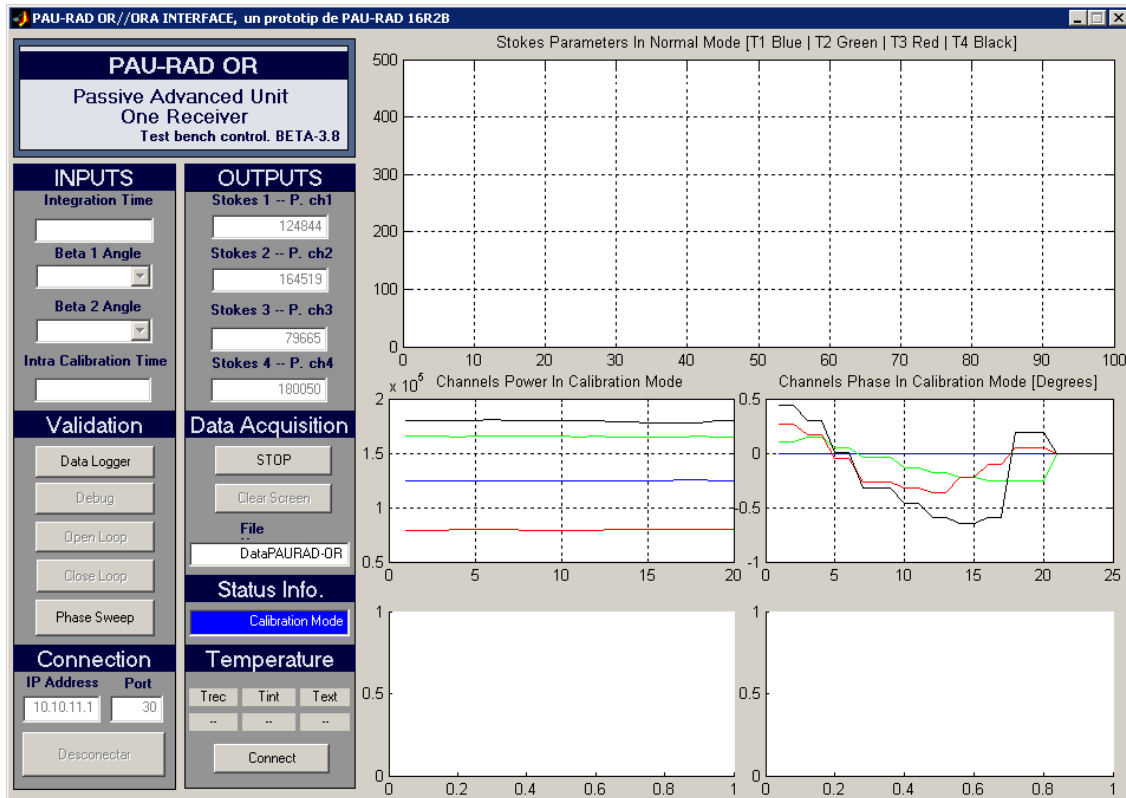


Figura 4-15: Captura de la interfaz durante la ejecución de una secuencia *Cosed Loop*.

4.4 CONCLUSIONES

A lo largo de este capítulo se ha definido el comportamiento de la UC así como de la interfaz de captura de datos. Las versiones originales de ambas fueron modificadas durante el transcurso del proyecto para adaptarse a las necesidades surgidas a partir del trabajo desarrollado en el ámbito de los algoritmos de calibración.

En el caso de la UC se modificó el diagrama de flujo entre estados, se desarrollaron nuevos estados destinados a verificar el correcto comportamiento tanto del software como del hardware del sistema y se añadieron los algoritmos de calibración y filtrado que se describirán en los siguientes capítulos. Por lo que respecta a la interfaz de captura de datos, se incorporó la representación gráfica de datos en tiempo real, que se confirmó como una poderosa herramienta a la hora de extraer conclusiones acerca del funcionamiento del sistema.

5 ALGORITMOS DE CALIBRACIÓN

5.1 INTRODUCCIÓN

En este capítulo se abordan en detalle los algoritmos de calibración implementados en el estado *Calibration* que forma parte de la *Secuencia Principal* de ejecución de la UC de PAU-RAD (ver capítulo 4).

Como plataforma de desarrollo y prueba de los algoritmos se ha empleado PAU-OR (One Receiver), una versión reducida del sistema completo PAU-16R que utiliza un único receptor al que se ha denominado receptor de referencia. No obstante, este hecho no resta generalidad a los algoritmos, que son extensibles al caso de 16 elementos. A no ser que se indique lo contrario, todo lo descrito en este capítulo aplica al caso de un único receptor (PAU-OR).

Al final del capítulo se incluyen resultados de una primera versión que trabaja con 4 receptores, así como algunas consideraciones a tener en cuenta a la hora de extender los algoritmos al sistema completo de 16 elementos.

5.1.1 Importancia de la calibración

Tal y como se muestra en la figura 5-1, cada uno de los receptores analógicos consta de cuatro cadenas, dos para la polarización horizontal y dos para la polarización vertical.

Las señales analógicas proporcionadas por cada uno de los canales (S1, S2, S3 y S4) proceden, por tanto, de cadenas receptoras diferentes. Esto supone que, incluso cuando los cuatro canales tienen a la entrada la misma señal, se producen a la salida ciertas variaciones entre ellos.

Suponiendo que la temperatura de ruido introducida por el receptor es la misma para los cuatro canales estas variaciones se presentan de dos formas distintas: en primer lugar una variación de amplitud, provocada por las diferencia de ganancias entre las cadenas y en segundo lugar una variación de fase, debida a que el reloj del sistema es regenerado en cada cadena por un PLL (Phase Locked Loop) diferente y por

tanto la fase de cada cadena es completamente aleatoria. Es posible afirmar por tanto que el receptor presenta ciertos errores de fase y ganancia inherentes a su estructura.

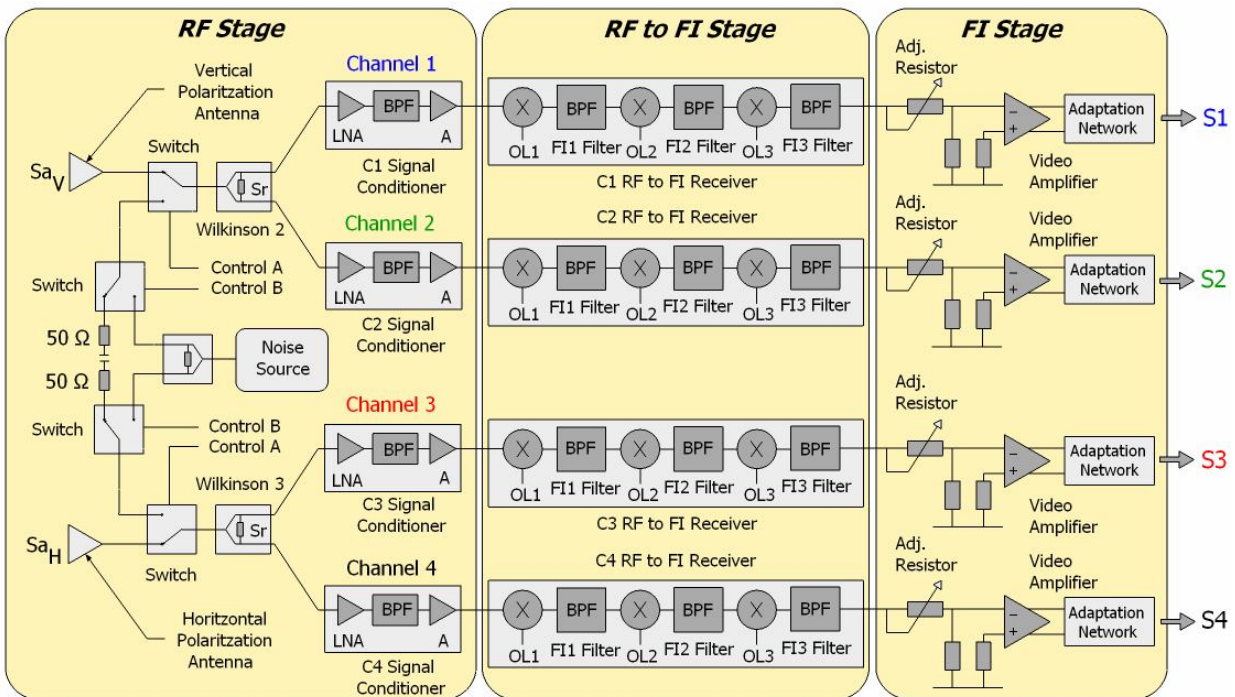


Figura 5-1: Diagrama de bloques del receptor analógico PAU.

La finalidad del proceso de calibración es la de eliminar estos errores de fase y ganancia entre canales. Para ello se fijan la fase y la amplitud de uno de los canales (en nuestro caso el canal 1) como referencia, y se igualan a estas la fase y la ganancia de los otros tres canales mediante los procedimientos que se describirán en este capítulo. Es decir: se lleva a cabo una estimación y corrección del error relativo (tanto de fase como de ganancia) entre canales.

Pero, ¿qué influencia tienen estos errores en el comportamiento global del sistema? Como ya se ha comentado en el capítulo 3, la antena de PAU-RAD es una agrupación bidimensional de 16 elementos. Dos de los parámetros más importantes de esta agrupación son la eficiencia del lóbulo principal o MBE (Main Beam Efficiency) y el Nivel de Lóbulo Principal a Secundario o NLPS.

El MBE es una medida de la cantidad de potencia relativa que entra por el lóbulo principal respecto al total. Se trata de un parámetro de vital importancia en aplicaciones radiométricas, puesto que garantiza que la señal obtenida procede de la superficie enfocada. Por su parte el NLPS es una medida de la atenuación que sufren las señales captadas por el lóbulo secundario del factor de agrupación respecto al lóbulo principal. Los valores de diseño para estos parámetros son $MBE = 0.96$ y $NLPS = 25$ dB.

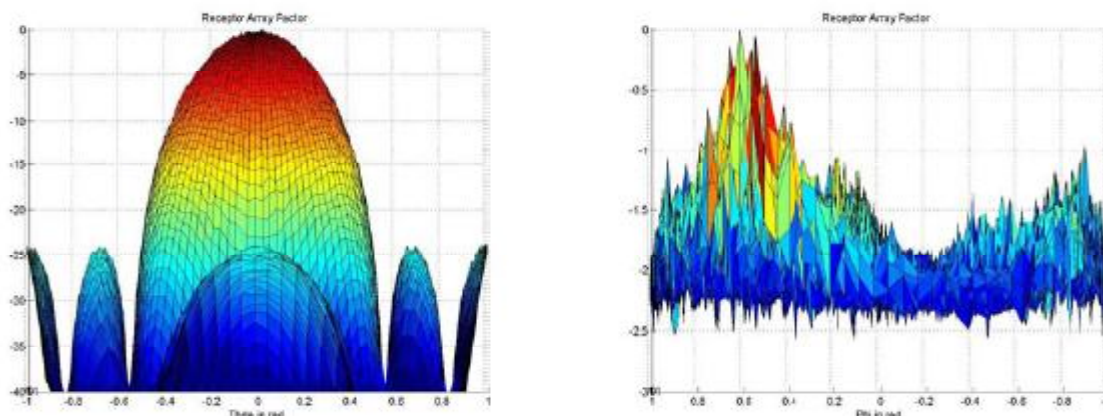


Figura 5-2: A la izquierda el factor de agrupación teórico; a la derecha el mismo factor de agrupación considerando un error aleatorio de fase entre 0 y 2π y de ganancia de ± 1 dB.

La figura 5-2 muestra el resultado de dos simulaciones realizadas para el sistema completo. A la izquierda se puede observar el factor de agrupación teórico de la antena empleada en PAU, mientras que la gráfica de la derecha muestra el mismo factor de agrupación pero considerando un error aleatorio de fase entre 0 y 2π y de ganancia de ± 1 dB.

Como se puede comprobar en la figura de la izquierda, la presencia en el sistema de errores de fase y ganancia deforma por completo el factor de agrupación, de forma que se pierden completamente los valores teóricos de MBE y NLPS. Las medidas adquiridas con el factor de agrupación resultante serían totalmente incorrectas e inservibles. Por tanto resulta imprescindible para el correcto funcionamiento del sistema la calibración de los datos adquiridos en la etapa analógica antes de realizar la conformación del haz.

El estudio realizado por Xavier Bosch en [2] demuestra que el comportamiento del sistema es adecuado para errores residuales de fase de ± 3 grados y de amplitud de $\pm 0,1$ dB. El objetivo de la calibración por tanto, será reducir los errores hasta situarlos dentro de estos márgenes.

5.1.2 Descripción general del proceso de calibración

Tal y como se puede comprobar en la figura 5-1, cada receptor analógico posee tres entradas diferentes: una entrada directa de antena que se emplea en la obtención de los parámetros de Stokes, una entrada de calibración a la que se conecta una fuente de ruido externa y una entrada conectada internamente a una resistencia de 50Ω .

Para poder llevar a cabo la calibración de las cuatro cadenas de un receptor, es necesaria una señal externa que actúe como referencia común. Este es precisamente el propósito de la fuente de ruido externa, actuar como señal de calibración.

Al inicio de cada ciclo de calibración se actúa sobre los controles A y B (ver figura 5-1) de forma que la entrada se conmuta para que los cuatro canales de cada receptor tengan como entrada la señal procedente de la fuente de ruido externa.

Puesto que se inyecta la misma señal por las cuatro cadenas de cada receptor, calculando las correlaciones cruzadas entre la señal a la salida del canal de referencia (S_1) y las señales a la salida del resto de canales (S_2, S_3, S_4), es posible estimar mediante los algoritmos que se definirán en los siguientes apartados las diferencias de fase y amplitud de cada uno de los canales con respecto al canal de referencia.

Una vez estimados los errores de fase y amplitud de cada uno de los canales respecto al canal de referencia, se procede a corregirlos. Para ello se modifican los valores de las ecuaciones trigonométricas contenidas en las TT (ver capítulo 3) de forma que se modifica la fase y la amplitud de los osciladores locales de cada canal para que incluyan estos errores. Finalmente las TT corregidas para un receptor cualquiera presentan el aspecto que se muestra en la tabla 5-1:

Tabla 5-1: Contenido de la TT de un receptor.

POSICIÓN	ECUACIÓN TRIGONOMÉTRICA
1	$A_1 \cos(\theta_1)$
2	$A_2 \cos(\theta_2)$
3	$A_3 \cos(\theta_3)$
4	$A_4 \cos(\theta_4)$
5	$A_1 \cos(\pi/2 + \theta_1)$
6	$A_2 \cos(\pi/2 + \theta_2)$
7	$A_3 \cos(\pi/2 + \theta_3)$
8	$A_4 \cos(\pi/2 + \theta_4)$
9	$A_1 \cos(\pi + \theta_1)$
10	$A_2 \cos(\pi + \theta_2)$
11	$A_3 \cos(\pi + \theta_3)$
12	$A_4 \cos(\pi + \theta_4)$
13	$A_1 \cos(3\pi/2 + \theta_1)$
14	$A_2 \cos(3\pi/2 + \theta_2)$
15	$A_3 \cos(3\pi/2 + \theta_3)$
16	$A_4 \cos(3\pi/2 + \theta_4)$

Donde A_i y θ_i (con $i = 1, 2, 3, 4$) representan el factor de corrección de ganancia y el error de fase estimado respectivamente.

Las TT corregidas se escriben en la ALU del radiómetro, de forma que los datos obtenidos en la siguiente captura del radiómetro ya estarán corregidos en fase y amplitud. Para conseguir mayor precisión en los resultados reduciendo la varianza asociada a las medidas, el proceso de estimación y corrección de errores se repite un número de veces fijado por el usuario. Se considera que 10 iteraciones del proceso son suficientes para reducir de manera considerable la varianza asociada a las medidas.

Una vez transcurrido el número de iteraciones fijado, se selecciona la entrada de antena, de tal manera que las modificaciones realizadas en los osciladores locales de cada canal a través de las TT, permiten la correcta extracción de los parámetros de Stokes.

La figura 5-3 muestra un diagrama simplificado del proceso de estimación y corrección de errores:

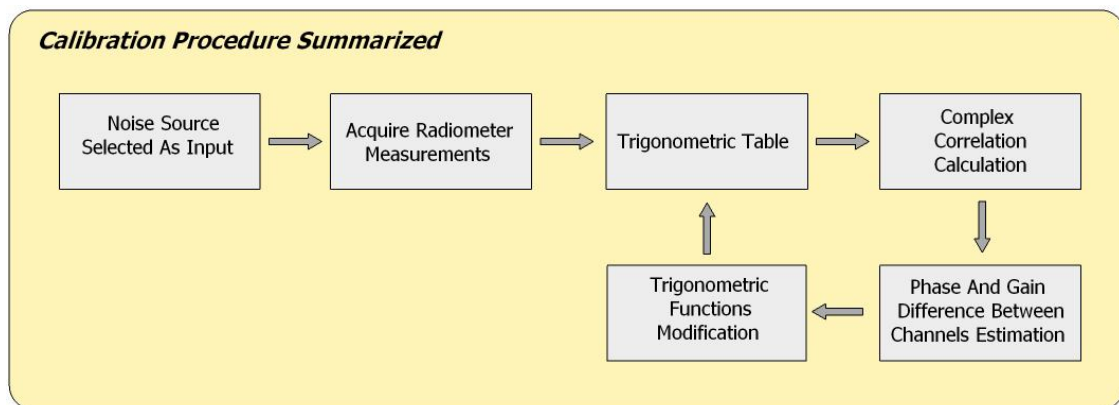


Figura 5-3: Proceso de estimación y corrección de errores simplificado.

En los siguientes apartados se abordan en profundidad los diferentes tipos de algoritmos sometidos a estudio así como su implementación y resultados a la hora de estimar y corregir los errores de fase y ganancia.

5.2 TIPOS DE ALGORITMOS ESTUDIADOS

5.2.1 Introducción

Durante el proceso de calibración se inyecta a cada canal la misma señal procedente de la fuente de ruido. Esta señal se verá afectada, de diferente forma para cada canal, por las tolerancias de los elementos que lo componen. El efecto provocado es la introducción de variaciones de amplitud y fase entre canales. Como resultado, las

salidas de los cuatro canales no serán iguales entre sí, a pesar de que el diseño hardware de los canales es idéntico y la señal de entrada es la misma.

Es posible expresar las salidas de cada uno de los canales de un receptor como:

$$\begin{aligned}
 S_1 &= I_1 + jQ_1 = |A_1|e^{j\theta_1} = a_1Te^{j\theta_1} \\
 S_2 &= I_2 + jQ_2 = |A_2|e^{j\theta_2} = a_2Te^{j\theta_2} \\
 S_3 &= I_3 + jQ_3 = |A_3|e^{j\theta_3} = a_3Te^{j\theta_3} \\
 S_4 &= I_4 + jQ_4 = |A_4|e^{j\theta_4} = a_4Te^{j\theta_4}
 \end{aligned}
 \tag{5.1}$$

Donde: I_i y Q_i (con $i = 1, 2, 3, 4$) representan respectivamente las componentes en fase y cuadratura de cada señal, $|A_i|$ y θ_i son el módulo y la fase, a_i representa el factor de ganancia de cada canal y T es la temperatura de ruido equivalente de la fuente de ruido correlado.

A la salida de los cuatro canales habrá, por tanto, cuatro señales diferentes. La misión de los algoritmos de calibración es la de igualar los factores de ganancia de los canales 2, 3 y 4 (a_2, a_3 y a_4) al factor de ganancia del canal de referencia (a_1), así como igualar las fases θ_2, θ_3 y θ_4 a la fase del canal de referencia θ_1 .

Durante el transcurso del proyecto se estudiaron dos tipos de algoritmos encargados de ejecutar esta tarea. Estos algoritmos corresponden a dos grados diferentes de complejidad del modelo del sistema: los algoritmos no diferenciales, que se basan en un modelo sencillo del sistema y los diferenciales, creados a partir de un modelo que considera la presencia de offsets en las señales de salida de cada canal.

Los dos tipos de algoritmos considerados se estudiaron además en dos variantes diferentes en función de la duración de su respuesta: respuesta finita y respuesta infinita. La inclusión de estas dos variantes de los algoritmos en el estudio obedece a una voluntad de disponer de un mayor número de opciones a la hora de llevar a cabo la calibración del sistema, lo que permite cubrir un amplio espectro de posibles situaciones.

En este apartado se definirán los conceptos de respuesta finita e infinita, ambos aplicables tanto a los algoritmos diferenciales como a los no diferenciales, y se introducirán las particularidades de los dos tipos de algoritmos considerados. El desarrollo matemático en profundidad de los algoritmos se abordará en los siguientes apartados.

5.2.2 Algoritmos de respuesta finita

Los algoritmos de respuesta finita son aquellos en los que se reescriben las tablas trigonométricas de referencia (aquellas que contienen los valores ideales que se corresponden con la ausencia de errores en el sistema) cada vez que se lleva a cabo una calibración. Por tanto, la estimación de los errores carece de memoria, es decir que no se tienen en cuenta las estimaciones de error realizadas en ciclos de calibración anteriores.

La siguiente figura muestra un esquema del funcionamiento de este tipo de algoritmos:

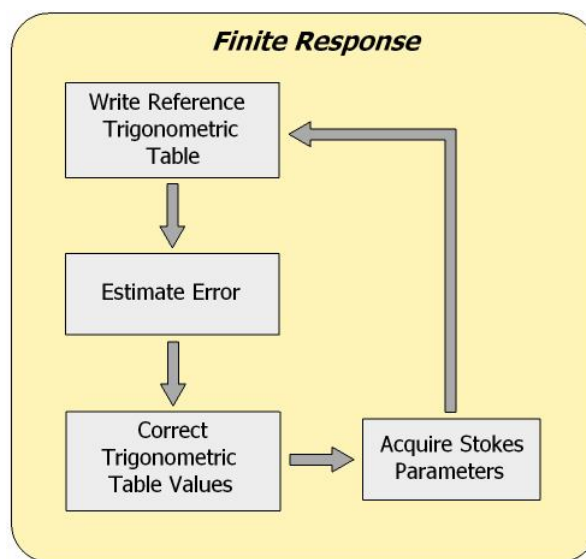


Figura 5-4: Esquema que ilustra el funcionamiento de los algoritmos de respuesta finita.

Las principales características de los algoritmos de respuesta finita son:

- En caso de que se produzca una variación brusca del error estimado, el sistema se recuperará con rapidez al no depender de los errores estimados en instantes anteriores.
- Requiere dos escrituras de las tablas trigonométricas en cada ciclo de calibración (las de referencia al inicio y las corregidas al final del ciclo).
- El error residual resultante presentará una variación rápida en torno a cero en régimen permanente.

5.2.3 Algoritmos de respuesta infinita

Por contraposición a los algoritmos de respuesta finita, los algoritmos de respuesta infinita, son aquellos en los que las tablas trigonométricas de referencia se escriben únicamente la primera vez que lleva a cabo un ciclo de calibración en una campaña de medidas. El error estimado en cada ciclo de calibración es acumulado, de manera que la estimación actual conserva memoria de estimaciones pasadas.

En la figura 5-5 se presenta un esquema del funcionamiento de los algoritmos de respuesta infinita:

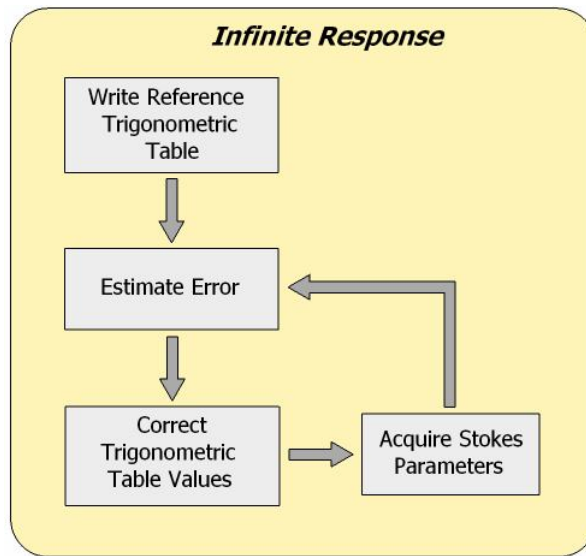


Figura 5-5: Esquema que ilustra el mecanismo de los algoritmos de respuesta infinita.

Para el caso de los algoritmos de calibración de fase, la implementación de la versión de respuesta infinita pasa por la acumulación de los errores instantáneos o deltas de error ($\Delta\theta_{error}$) obtenidos en sucesivas iteraciones del proceso, es decir:

$$\theta_{error} = \theta_{error} + \Delta\theta_{error} \quad (5.2)$$

En el caso de los algoritmos de calibración de ganancia, los coeficientes calculados en cada iteración dependen de los obtenidos en la iteración anterior y del error de ganancia instantáneo (α):

$$A_n = A_{n-1} \frac{1}{\alpha} \quad (5.3)$$

Este tipo de algoritmos posee las siguientes características fundamentales:

- Recuperación lenta del sistema en caso de que se produzca una variación súbita del error estimado al depender de errores anteriores.
- Requiere una única escritura de las tablas trigonométricas en cada ciclo de calibración, lo que reduce la carga computacional del sistema y aumenta la rapidez del mismo.
- La variación en torno al cero del error residual resultante en régimen permanente será más lenta que para el caso de los algoritmos de respuesta finita.

Puesto que las señales que se pretenden medir no presentan variaciones rápidas y una oscilación lenta del error residual resultante una vez aplicados los algoritmos proporciona estabilidad a los parámetros de Stokes obtenidos, de forma habitual se empleará esta versión en la implementación de los algoritmos.

5.2.4 Algoritmos no diferenciales

Los algoritmos no diferenciales son aquellos en los cuales el nivel de señal procedente de la fuente de ruido externa se mantiene constante durante todo el proceso de calibración. Es decir: la temperatura equivalente de ruido que es detectada por cada receptor en la entrada de calibración permanece constante.

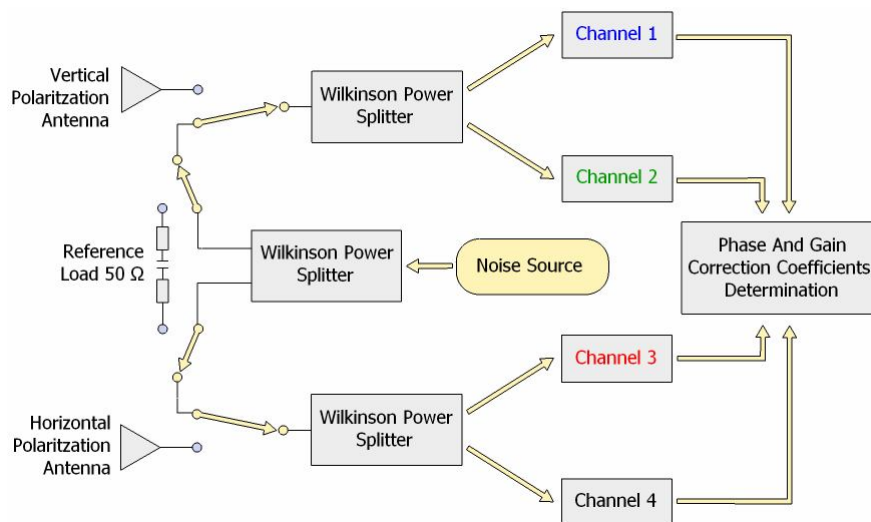


Figura 5-6: Esquema simplificado de un receptor durante el proceso de calibración mediante algoritmo no diferencial.

La figura 5-6 muestra una representación esquemática de un receptor durante el proceso de calibración no diferencial.

Estos algoritmos se desarrollaron a partir de un modelo sencillo del sistema que no contempla la presencia de offsets en las señales. Más adelante se presentan las ecuaciones del modelo y se desarrollan en profundidad los algoritmos.

5.2.5 Algoritmos diferenciales

Los algoritmos diferenciales hacen uso de una fuente de ruido conmutable que es capaz de entregar a los receptores diferentes niveles de potencia de ruido. A partir de las diferencias entre las señales obtenidas para dos niveles de potencia de ruido distintos, es posible determinar los coeficientes de corrección de fase y ganancia.

La figura 5-7 esquematiza un receptor durante el proceso de calibración mediante algoritmos diferenciales:

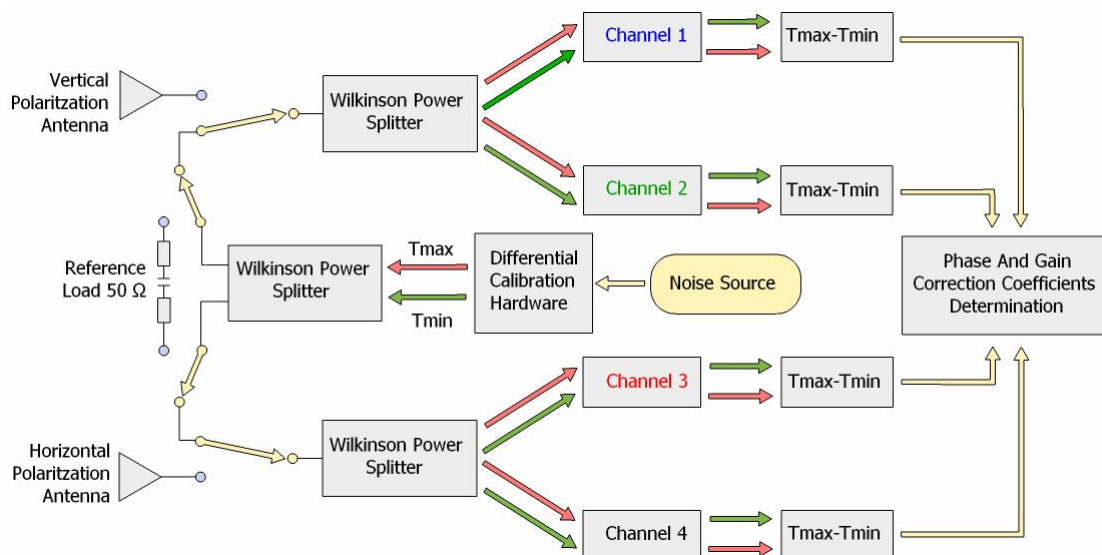


Figura 5-7: Esquema simplificado de un receptor durante el proceso de calibración mediante algoritmo diferencial.

Los algoritmos diferenciales fueron concebidos como solución a un nuevo modelo del sistema, más complejo, en el que se considera la presencia de offsets en las señales de salida los cuatro canales de cada receptor. Más adelante se presentarán las ecuaciones del modelo, se definirá con mayor detalle este tipo de algoritmo y se expondrán las razones que dieron lugar a su desarrollo.

5.2.6 Hardware de calibración diferencial

La implementación de los algoritmos diferenciales requiere un sistema que permita controlar de forma digital la potencia entregada por la fuente de ruido,

permitiendo así establecer varios niveles de potencia a partir de los cuales se extraerán los coeficientes de calibración. En este apartado se aborda con detalle el proceso de diseño e implementación de este sistema de calibración por temperatura de referencia interna.

5.2.6.1 Descripción del sistema

A continuación se muestra el esquema básico que llevará a cabo la función de calibración por temperatura de referencia interna. El sistema se compone de una fuente de ruido modelo NC346, un atenuador fijo (ATT) cuyo valor será discutido más adelante, un atenuador variable (ZX73-2500) cuyo valor de atenuación se controla mediante un voltaje de control, un switch (MSP2T-18), una fuente de alimentación conectada a la red que proporciona 28V y un sistema de control que actúa de interfaz entre la FPGA y el sistema.

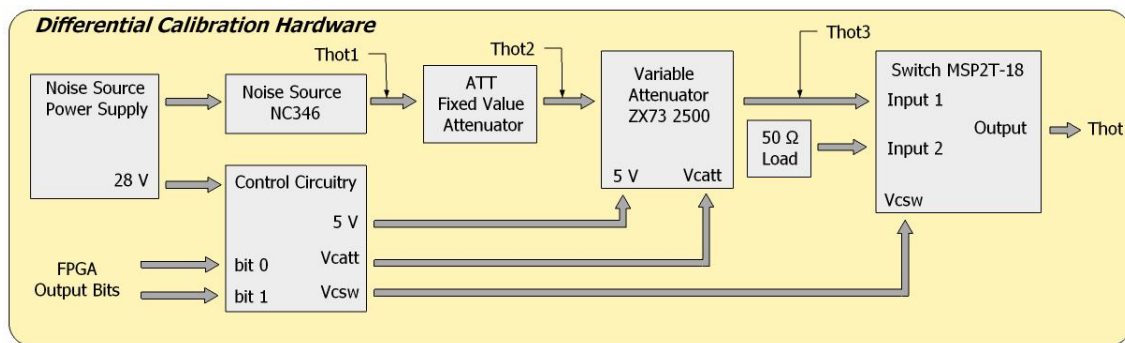


Figura 5-8: Diagrama de bloques del hardware de calibración diferencial.

La finalidad de este esquema es la de disponer en la salida T_{HOT} de tres temperaturas de referencia ($T_{MAX} = 500$ K, $T_{MED} = 400$ K y $T_{MIN} = 300$ K) que permitan, por una parte extraer los coeficientes de calibración de los algoritmos diferenciales; y por otra obtener la recta de calibración del radiómetro. Para ello se han estudiado los valores de atenuación necesarios de manera que a través de los bits b_0 y b_1 el software de la unidad de control cargado en el procesador Nios sea capaz de conmutar entre estas tres temperaturas.

Según las especificaciones de la fuente de ruido, esta proporciona a su salida una ENR (Excess Noise Ratio) de 21,41 dB, lo cual se traduce en una temperatura de ruido equivalente T_{HOT1} de:

$$ENR = 10 \log \left(\frac{T_{HOT1} - T_{COLD}}{T_0} \right) = 21,41 \text{ dB} \rightarrow T_{HOT1} = 40413 \text{ K} \quad (5.4)$$

Donde se ha aproximado la temperatura de fuente apagada (T_{COLD}) a la temperatura ambiente (T_0) y se ha considerado para esta última un valor de 290 K. Esta aproximación se utilizará de manera regular a lo largo de los siguientes apartados.

La potencia de ruido proporcionada por la fuente es demasiado elevada para las especificaciones del sistema (se pretende ver a la salida temperaturas de 300, 400, y 500 K, muy por debajo de los 40413 K que entrega la fuente) lo que implica que el atenuador variable tenga que atenuar entre 40 y 20 dB, trabajando en una zona de polarización en la que una variación de mili voltios (ver figura 5-9) en la tensión de control provocaría una variación de decenas de dB de atenuación.

Para solucionar este inconveniente se introduce en el sistema un segundo atenuador, cuya función es la de reducir la potencia de ruido a la salida de la fuente, desplazando la zona de trabajo del atenuador variable hacia una zona más lineal y de menor pendiente.

5.2.6.2 Cálculo de las atenuaciones

El primer paso del diseño consiste en determinar las atenuaciones que deberá aplicar el atenuador variable a la señal procedente de la fuente de ruido en función de diferentes valores del atenuador fijo y determinar los voltajes de control que requieren dichas atenuaciones. Se han estudiado tres posibles valores de atenuación que corresponden con atenuadores disponibles en el laboratorio: 20 dB, 12 dB y 6 dB.

o Atenuaciones para ATT = 20 dB

Considerando un atenuador fijo de 20 dB de atenuación, tenemos que la temperatura de ruido equivalente a la salida del atenuador es:

$$T_{HOT2} = T_{HOT1}L + (1 - L)T_0 \rightarrow T_{HOT2} = 691 K \quad (5.5)$$

Donde, de nuevo se han aplicado las mismas aproximaciones aplicadas en la ecuación 5.4.

Si se procede de igual forma para la temperatura de ruido equivalente a la salida del atenuador variable T_{HOT3} :

$$T_{HOT3} = T_{HOT2}L_{VAR} + (1 - L_{VAR})T_0 \quad (5.6)$$

Es posible determinar la atenuación necesaria para obtener a la salida una temperatura equivalente de ruido tal que $T_{HOT3} = T_{med} = 400 K$:

$$L_{VAR} = \frac{110}{(T_{HOT2} - 290)} = 5,62 \text{ dB} \quad (5.7)$$

De igual forma, para que $T_{HOT3} = T_{max} = 500 \text{ K}$:

$$L_{VAR} = \frac{210}{(T_{HOT2} - 290)} = 2,81 \text{ dB} \quad (5.8)$$

La figura 5-9 muestra los valores de tensión de control necesarios para obtener los valores de atenuación calculados:

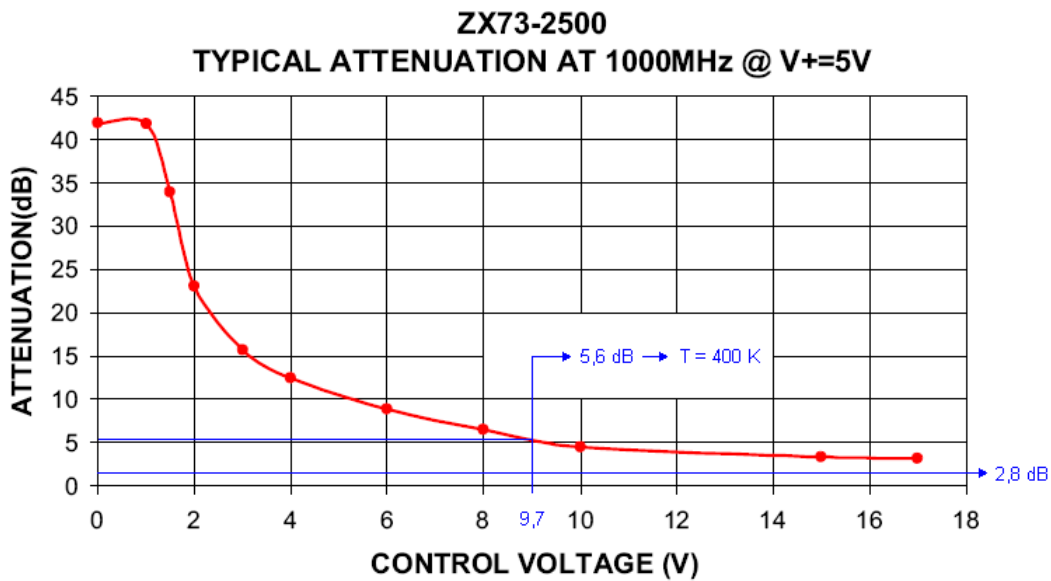


Figura 5-9: Curva de atenuación del atenuador variable con valores de atenuación para T_{max} y T_{med} con un atenuador de 20 dB.

Como se puede apreciar, el atenuador variable no proporciona valores de atenuación de 2,8 dB, por tanto esta configuración no es válida.

- **Atenuaciones para ATT = 12 dB**

Siguiendo el mismo procedimiento que para el caso de 20 dB de atenuación, se tiene que:

$$T_{HOT2} = T_{HOT1}L + (1 - L)T_0 \rightarrow T_{HOT2} = 2821 \text{ K} \quad (5.9)$$

Para obtener una temperatura equivalente de ruido tal que $T_{HOT3} = T_{med} = 400 \text{ K}$:

$$L_{VAR} = \frac{110}{(T_{HOT2} - 290)} = 13,62 \text{ dB} \quad (5.10)$$

De igual forma, para que $T_{HOT3} = T_{max} = 500 \text{ K}$:

$$L_{VAR} = \frac{210}{(T_{HOT2} - 290)} = 10,81 \text{ dB} \quad (5.11)$$

La figura 5-10 muestra los valores de tensión de control necesarios para obtener los valores de atenuación calculados:

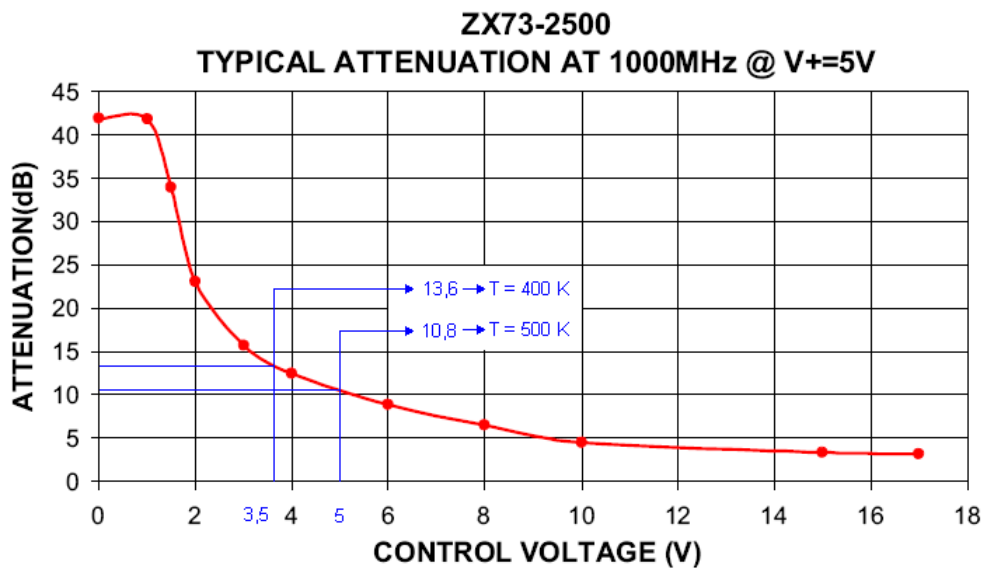


Figura 5-10: Curva de atenuación del atenuador variable con valores de atenuación para T_{max} y T_{med} con un atenuador de 12 dB.

Como se puede apreciar en la figura, para un atenuador de 12 dB, se requieren valores de tensión de control de 3,5 y 5 voltios para obtener T_{max} y T_{med} .

○ **Atenuaciones para ATT = 6 dB**

De nuevo, adaptando las ecuaciones de los dos casos anteriores para el caso de un atenuador de 6 dB, tenemos que:

$$T_{HOT2} = T_{HOT1}L + (1 - L)T_0 \rightarrow T_{HOT2} = 10368,54 \text{ K} \quad (5.12)$$

Para conseguir el nivel de temperatura correspondiente a $T_{HOT3} = T_{med} = 400 \text{ K}$:

$$L_{VAR} = \frac{110}{(T_{HOT2} - 290)} = 19,62 \text{ dB} \quad (5.13)$$

Asimismo, para el nivel de temperatura de valor $T_{HOT3} = T_{max} = 500 \text{ K}$:

$$L_{VAR} = \frac{210}{(T_{HOT2} - 290)} = 16,81 \text{ dB} \quad (5.14)$$

Los valores de tensión de control necesarios para obtener estos valores de atenuación se muestran en la figura 5-11:

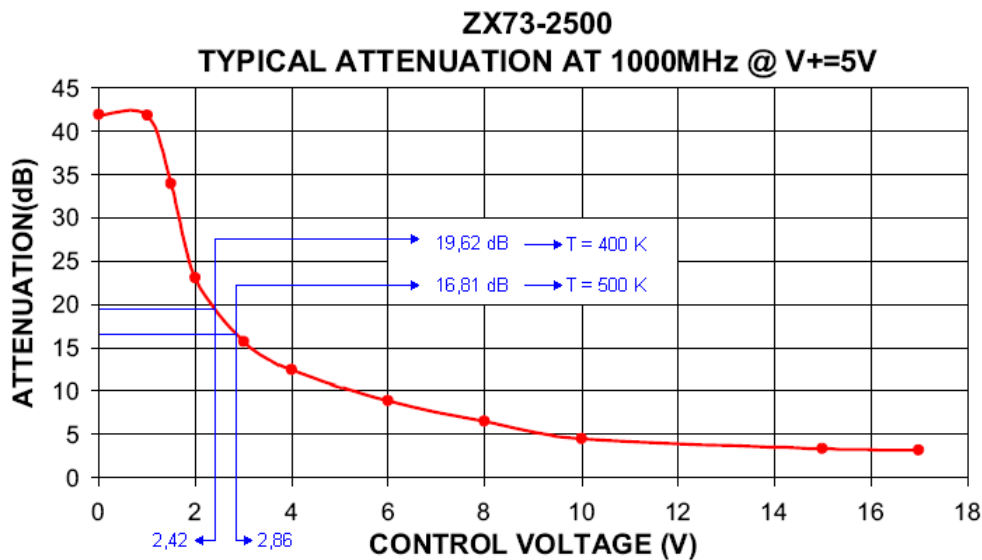


Figura 5-11: Curva de atenuación del atenuador variable con valores de atenuación para T_{max} y T_{med} con un atenuador de 6 dB.

En este caso se obtienen valores de tensión de control demasiado próximos entre sí, lo que dificulta la implementación de la circuitería de control, invalidando esta configuración.

Finalmente el atenuador empleado es el 12 dB de atenuación, ya que proporciona los mejores valores de atenuación y valores de la tensión de control fácilmente alcanzables con una implementación de la circuitería de control sencilla.

5.2.6.3 Circuitería de control

A continuación se muestra el diseño de la circuitería de control, cuya función es doble: por una parte permitir al usuario conmutar entre las tres temperaturas equivalentes de ruido empleadas como referencia; y por otra adaptar la fuente de alimentación disponible (la misma que se emplea para alimentar la fuente de ruido) a

los requerimientos de alimentación de los elementos que componen el hardware de conmutación. La figura 5-12 muestra el esquemático de la circuitería de control implementada:

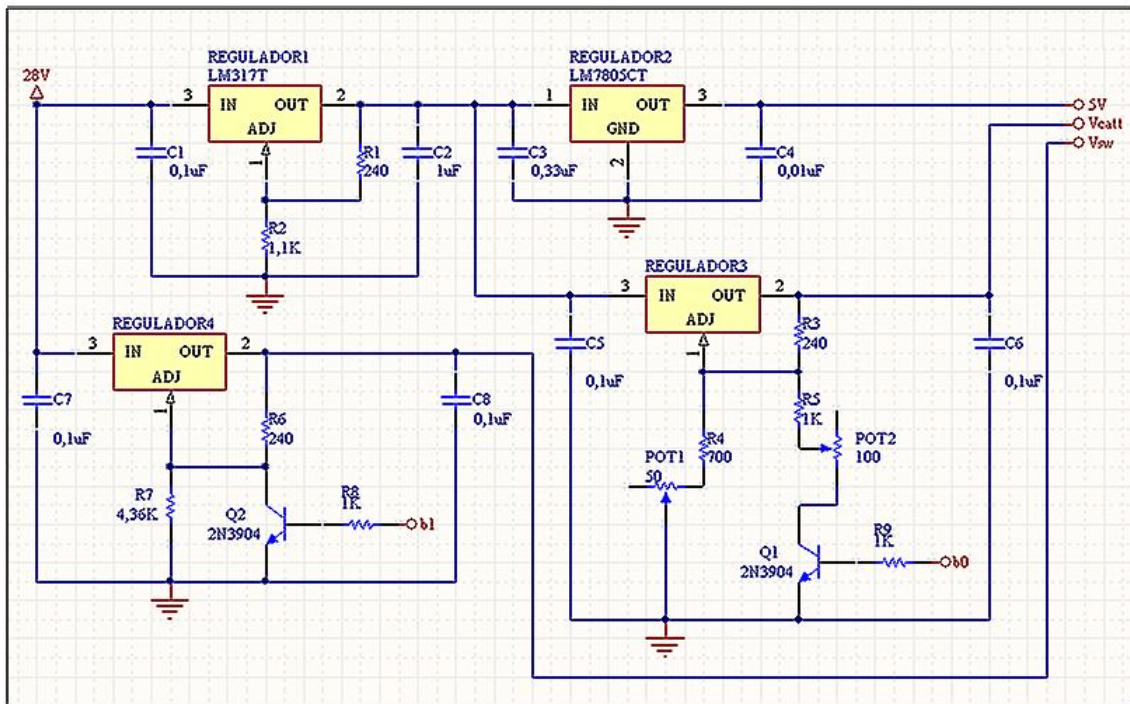


Figura 5-12: Esquemático de la circuitería de control del hardware de conmutación de temperatura de referencia interna.

La salida etiquetada con el nombre 5 V proporciona los 5 voltios constantes que alimentan al atenuador variable, mientras que Vcatt es la tensión de control del atenuador variable que conmuta entre 3.5 V ($T_{med} = 400\text{ K}$) y 5 V ($T_{max} = 500\text{ K}$) en función del valor lógico del bit 0 (b0 en la figura). Por otra parte, la tensión de salida Vsw, corresponde a la alimentación del switch, que conmuta entre 1.3 V y 24 V permitiendo seleccionar, en función del bit 1 (b1 en la figura), la fuente de ruido atenuada o bien una carga adaptada que proporcionará la tercera temperatura de referencia ($T_{min} = 300\text{ K}$).

La tabla 5-2, resume el comportamiento de la circuitería de control en función de los valores de los bits de entrada:

Tabla 5-2: Tensiones de salida y temperaturas correspondientes en función de los bits de entrada.

Equivalent Noise Temperature [K]	Bit 0	Bit 1	Vcatt [V]	Vsw [V]
300	0	0	5	24
300	1	0	3,5	24
400	1	1	3,5	1,3
500	0	1	5	1,3

5.2.6.4 Pruebas experimentales

En primer lugar se procedió a comprobar que los niveles de atenuación introducidos, tanto por el atenuador variable, como por el sistema completo se corresponden con los calculados teóricamente. Para ello se empleó un analizador de factor de ruido que previamente había sido calibrado empleando la fuente de ruido del sistema. Es importante notar que el analizador de factor de ruido empleado en esta prueba tiene un rango de funcionamiento que alcanza 1,5 GHz, mientras que PAU trabaja a una frecuencia de 1,575 GHz.

Para llevar a cabo la conmutación entre las temperaturas T_{\max} y T_{med} se empleó una fuente de alimentación conectada al bit 0 que simulaba los valores de tensión correspondientes a niveles lógicos alto y bajo que se habían medido previamente en los pines de la FPGA.

Como se puede apreciar en la figura 5-13, la atenuación introducida por el atenuador variable cuando se selecciona T_{\max} (bit 0 = 0) es de 10,62 dB, valor muy aproximado a los 10,69 dB calculados teóricamente para una temperatura de 500 K.

Figura 5-13: Atenuación introducida por el atenuador variable para $T=500$ K a una frecuencia de 1,5 GHz.

En la siguiente figura se muestra la atenuación introducida por el atenuador variable para una temperatura T_{med} (bit 0 = 1). Como se puede comprobar, el valor medido de atenuación es de 13,68 dB, valor que coincide con los 13,62 dB calculados teóricamente para una temperatura de 400 K.

Figura 5-14: Atenuación introducida por el atenuador variable para $T=400\text{K}$ a una frecuencia de 1,5 GHz.

En las dos gráficas siguientes se muestran los niveles de atenuación correspondientes a T_{\max} y T_{med} para el sistema de conmutación al completo (fuente de ruido, atenuador de 12 dB, atenuador variable y switch).

En la figura 5-15 se puede comprobar que la atenuación del sistema completo para la temperatura de referencia T_{\max} es de 23,26 dB:

Figura 5-15: Atenuación del sistema completo (Fuente de ruido + Atenuador 12 dB + Atenuador Variable + Switch) para una temperatura $T_{\max} = 500\text{ K}$.

De la misma forma, seleccionando como temperatura de referencia T_{med} se obtiene una atenuación de 26,17 dB:

Figura 5-16: Atenuación del sistema completo (Fuente de ruido + Atenuador 12 dB + Atenuador Variable + Switch) para una temperatura $T_{med} = 400$ K.

Ambos valores de atenuación son correctos, ya que se corresponden con los calculados para el atenuador variable más los 12 dB introducidos por el atenuador fijo.

A continuación se comprobó el comportamiento del sistema completo conectado al radiómetro. De nuevo se simularon los bits 0 y 1 mediante una fuente de alimentación y se realizaron medidas de potencia para los diferentes canales durante la ejecución de una secuencia *Open Loop* (ver capítulo 4) conmutando manualmente la fuente de alimentación para conseguir las tres temperaturas de referencia.

Como se puede comprobar en la figura 5-17, se producen tres saltos en la potencia captada por el radiómetro cuando se seleccionan las tres temperaturas. No obstante, en la escala vertical se puede apreciar que los saltos son de aproximadamente 0,1 dB.

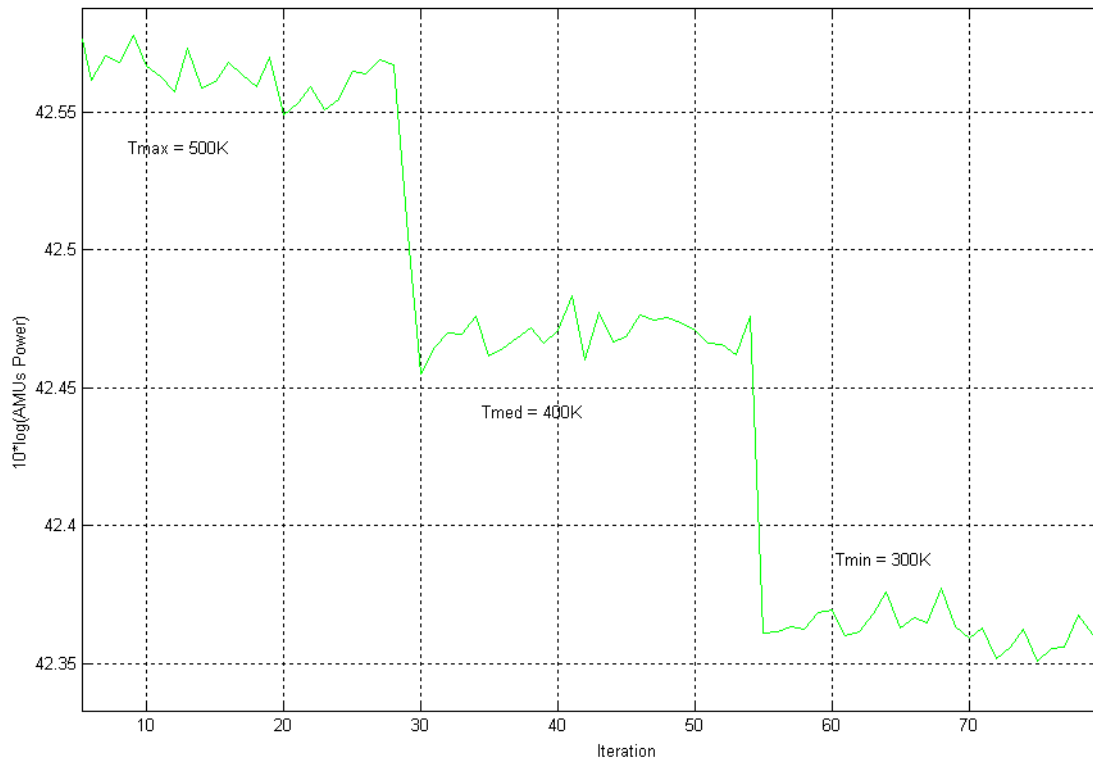


Figura 5-17: Potencia captada por el receptor para las tres temperaturas de referencia.

La respuesta en potencia por tanto no es la esperada intuitivamente, ya que, teóricamente, se deberían de producir saltos de 3 dB, puesto que la potencia captada por el radiómetro para T_{max} es la potencia de ruido emitida por la fuente, atenuada 23,26 dB mientras que para T_{med} es la misma atenuada 26,17 dB.

Para verificar el correcto funcionamiento de la parte digital de PAU en el rango de temperaturas del hardware de calibración diferencial, se conectó un generador de funciones a la entrada del ADC mediante una placa de adaptación, y se inyectó ruido por los cuatro canales del ADC.

Con esta configuración se realizó un barrido de potencias desde -50 dBm hasta 3 dB. Los resultados obtenidos se muestran en la figura 5-18.

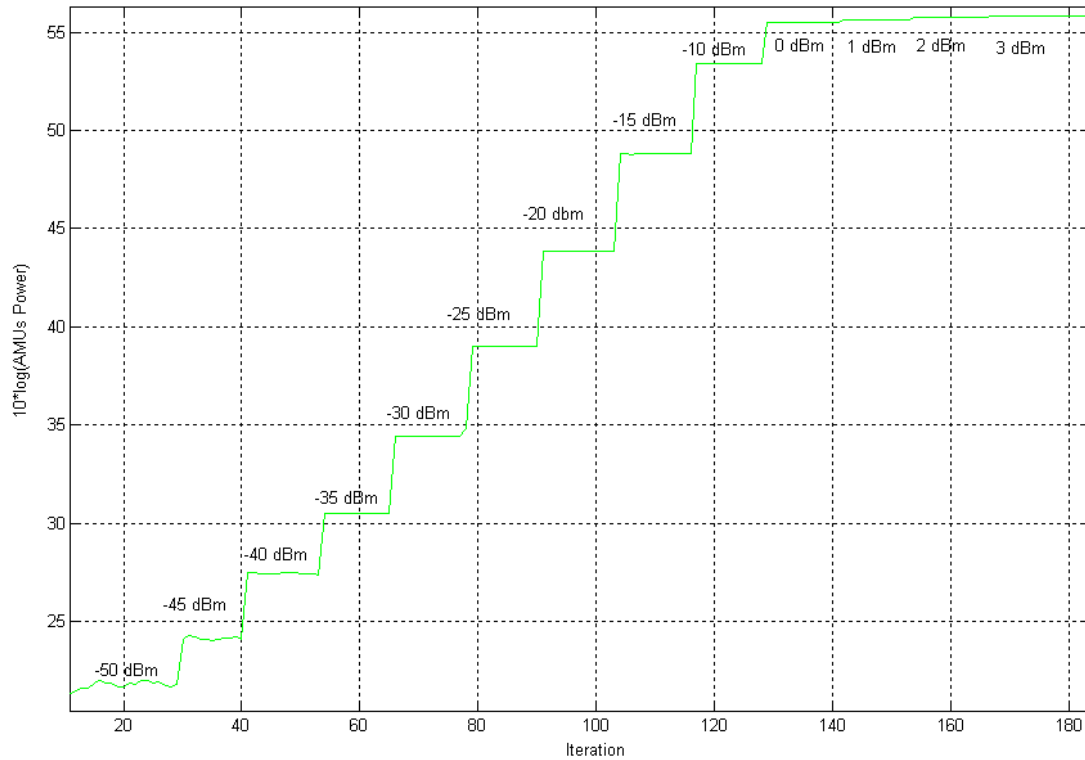


Figura 5-18: Respuesta en potencia del canal 1 para un barrido de potencia de ruido de - 50 dBm a 3 dBm.

Como se puede observar, la parte digital del sistema se comporta de forma lineal en todo el rango de potencias de entrada.

Tras varias comprobaciones se llegó a la conclusión de que los saltos de potencia captados por el receptor no se corresponden con los saltos de 3 dB esperados debido a que en los cálculos iniciales no se tuvo en consideración la atenuación introducida por los divisores Wilkinson.

Cada uno de los dos divisores Wilkinson que atraviesa la señal procedente de la fuente de ruido introduce una atenuación de 3 dB, por tanto, la potencia de ruido que finalmente llega a la entrada de cada canal es 6 dB menor de lo esperado. Teniendo en cuenta la atenuación de los divisores, se tiene que las temperaturas de ruido equivalentes para cada nivel de potencia son:

$$T_{HOT3|T=T_{MAX}} = T_{HOT2}L_W + (1 - L_W)T_0 \rightarrow T_{HOT3} = 342,5 K \quad (5.15)$$

$$T_{HOT3|T=T_{MED}} = T_{HOT2}L_W + (1 - L_W)T_0 \rightarrow T_{HOT3} = 317,5 K$$

A la luz de estos valores los resultados obtenidos en la figura 5-17 cobran significado ya que, realmente, las temperaturas de ruido equivalentes para los niveles T_{MAX} y T_{MED} no se corresponden con las esperadas (500 K y 400 K respectivamente),

sino con los valores obtenidos en la ecuación 5.13. De ahí que las potencias captadas por el receptor para los niveles de potencia de T_{MAX} y T_{MED} estén tan próximas al nivel de potencia captado para T_{MIN} que corresponde a una temperatura equivalente de 300 K.

Para corregir el efecto de los divisores Wilkinson se modificaron los valores de las resistencias de la circuitería de control de tal manera que los valores de la tensión de control del atenuador variable pasaron a ser: 7,6 dB para T_{MED} y 4,8 dB para T_{MAX} . De esta forma se compensan los 6 dB de pérdidas introducidos por los dos divisores de potencia.

5.3 CALIBRACIÓN DE FASE

5.3.1 Calibración de fase mediante algoritmo no diferencial

El proceso de calibración de fase consiste en la determinación los coeficientes de corrección que, aplicados a la salida de cada canal, igualan la fase del canal a la fase del canal de referencia (de ahora en adelante, por motivos de simplicidad, se considerará el canal 1 como canal de referencia).

El primer paso para la determinación de los coeficientes de corrección de fase consiste en el cálculo de las correlaciones entre la señal del canal de referencia y las señales del resto de canales conjugadas. La correlación de la señal del canal de referencia consigo misma conjugada se calcula como:

$$\langle S_1 S_1^* \rangle = \langle (I_1 + jQ_1)(I_1 - jQ_1) \rangle = I_1^2 + Q_1^2 \quad (5.16)$$

Expresando la correlación en forma de módulo y fase:

$$\langle S_1 S_1^* \rangle = \sqrt{I^2 + Q^2} e^{j0} = |A_{11}| e^{j\theta_{11}} = |A_1|^2 \quad (5.17)$$

Si se procede de igual manera para las correlaciones de las señales del canal de referencia con las del resto de canales se tiene que:

$$\begin{aligned}
\langle S_1 S_2^* \rangle &= (\langle I_1 I_2 \rangle + \langle Q_1 Q_2 \rangle) + j(\langle Q_1 I_2 \rangle - \langle I_1 Q_2 \rangle) \\
\langle S_1 S_3^* \rangle &= (\langle I_1 I_3 \rangle + \langle Q_1 Q_3 \rangle) + j(\langle Q_1 I_3 \rangle - \langle I_1 Q_3 \rangle) \\
\langle S_1 S_4^* \rangle &= (\langle I_1 I_4 \rangle + \langle Q_1 Q_4 \rangle) + j(\langle Q_1 I_4 \rangle - \langle I_1 Q_4 \rangle)
\end{aligned} \tag{5.18}$$

De nuevo, expresando las correlaciones como módulo y fase y aplicando las propiedades del ruido gaussiano blanco se tiene que:

$$\begin{aligned}
\langle S_1 S_2^* \rangle &= |A_{12}| e^{j\theta_{12}} = |A_1| |A_2| e^{j(\theta_1 - \theta_2)} \\
\langle S_1 S_3^* \rangle &= |A_{13}| e^{j\theta_{13}} = |A_1| |A_3| e^{j(\theta_1 - \theta_3)} \\
\langle S_1 S_4^* \rangle &= |A_{14}| e^{j\theta_{14}} = |A_1| |A_4| e^{j(\theta_1 - \theta_4)}
\end{aligned} \tag{5.19}$$

Es posible calcular la fase de cada una de estas correlaciones como:

$$\begin{aligned}
\theta_{11} &= \arctg \left(\frac{\langle Q_1 I_1 \rangle - \langle I_1 Q_1 \rangle}{\langle I_1 I_1 \rangle + \langle Q_1 Q_1 \rangle} \right) \\
\theta_{12} &= \arctg \left(\frac{\langle Q_1 I_2 \rangle - \langle I_1 Q_2 \rangle}{\langle I_1 I_2 \rangle + \langle Q_1 Q_2 \rangle} \right) \\
\theta_{13} &= \arctg \left(\frac{\langle Q_1 I_3 \rangle - \langle I_1 Q_3 \rangle}{\langle I_1 I_3 \rangle + \langle Q_1 Q_3 \rangle} \right) \\
\theta_{14} &= \arctg \left(\frac{\langle Q_1 I_4 \rangle - \langle I_1 Q_4 \rangle}{\langle I_1 I_4 \rangle + \langle Q_1 Q_4 \rangle} \right)
\end{aligned} \tag{5.20}$$

De tal forma que, aplicando las propiedades del ruido blanco Gaussiano ($\langle I_1 I_2 \rangle = \langle Q_1 Q_2 \rangle$ y $\langle Q_1 I_2 \rangle = -\langle I_1 Q_2 \rangle$) se tiene que, para los tres canales a corregir:

$$\theta_{12} = \arctg\left(\frac{-\langle I_1 Q_2 \rangle}{\langle I_1 I_2 \rangle}\right) = \theta_1 - \theta_2 = \alpha_2$$

$$\theta_{13} = \arctg\left(\frac{-\langle I_1 Q_3 \rangle}{\langle I_1 I_3 \rangle}\right) = \theta_1 - \theta_3 = \alpha_3 \quad (5.21)$$

$$\theta_{14} = \arctg\left(\frac{-\langle I_1 Q_4 \rangle}{\langle I_1 I_4 \rangle}\right) = \theta_1 - \theta_4 = \alpha_4$$

Donde α_i (con $i = 2, 3, 4$) representan los coeficientes de corrección de fase para cada uno de los canales que de desea corregir.

Estos coeficientes de corrección de fase son, en definitiva, la diferencia de fase existente entre la señal a la salida del canal de referencia y la señal a la salida del canal a corregir. Por tanto, para corregir las diferencias de fase basta con aplicar los coeficientes obtenidos a las señales de salida de cada canal (representadas en la ecuación 5.1), de tal manera que:

$$S_2' = S_2 e^{j\alpha_2} = |A_2| e^{j\theta_1}$$

$$S_3' = S_3 e^{j\alpha_3} = |A_3| e^{j\theta_1} \quad (5.22)$$

$$S_4' = S_4 e^{j\alpha_4} = |A_4| e^{j\theta_1}$$

Donde S_i' (con $i = 2, 3, 4$) son las señales a la salida de los canales 2, 3 y 4 una vez aplicados los coeficientes de corrección de fase.

Los resultados obtenidos en la ecuación 5.22 demuestran que, aplicando el algoritmo de calibración descrito, es posible igualar la fase de los canales 2, 3 y 4 a la fase del canal de referencia, calibrando así el sistema en fase. Notar que durante el proceso únicamente se ha trabajado con la fase, por lo que la ganancia permanece sin calibrar en la ecuación 5.22.

El siguiente paso será el cálculo de estos coeficientes de calibración de fase en la unidad de control de PAU.

5.3.2 Implementación en la unidad de control

5.3.2.1 Implementación mediante correlaciones a

1 bit

Los procedimientos descritos en los apartados anteriores para la obtención de los coeficientes de corrección son puramente teóricos, ahora bien ¿cómo se implementan estos procedimientos en la unidad de control de PAU?

Como se ha expuesto en los apartados anteriores, todo el proceso de determinación de los coeficientes de corrección de fase y amplitud se basa en el cálculo de las correlaciones cruzadas entre la señal del canal de referencia y las señales del resto de canales.

Las señales captadas por el receptor son digitalizadas en el ADC y enviadas a la ALU. Una vez allí, son trasladadas a banda base, filtradas paso bajo, y digitalizadas a un solo bit: el bit de signo. A partir de estas medidas se obtienen los coeficientes Z_r y Z_i tal y como se muestra en la ecuación 5.23:

$$Z_{r_{1,n}} = \frac{1}{N} \sum_{i=1}^N XNOR(i_{1_i} i_{n_i}) \quad (5.23)$$

$$Z_{i_{1,n}} = \frac{1}{N} \sum_{i=1}^N XNOR(i_{1_i} q_{n_i})$$

Donde i_1 denota la componentes en fase de la señal del canal de referencia mientras que i_n y q_n denotan las componentes en fase y cuadratura de las señales del resto de canales, con $n = 2, 3$ ó 4 .

Debido a que las señales empleadas en el cálculo están digitalizadas a un solo bit, estos coeficientes no son exactamente correlaciones, sino que constituyen medidas de similitud entre la fase del canal de referencia y la fase del resto de canales y entre la fase del canal de referencia y la cuadratura del resto de canales respectivamente, es decir:

$$I_1 I_1, I_1 Q_1, I_1 I_2, I_1 Q_2, I_1 I_3, I_1 Q_3, I_1 I_4, I_1 Q_4 \quad (5.24)$$

Además de estas 8 medidas de similitud, la ALU proporciona a la unidad de control 4 medidas de potencia (una para cada canal) y 5 medidas de offset (4 correspondientes a las componentes en fase de cada canal y 1 correspondiente a la cuadratura del canal de referencia).

A partir de estas medidas de similitud, podemos calcular correlaciones mediante las expresiones [6]:

$$\mu_{r,n} = \sin\left(\frac{\pi}{2}(2Z_r - 1)\right) = \frac{R_{i_1 i_n}(0)}{\sqrt{R_{i_1 i_1}(0)R_{i_n i_n}(0)}} \quad (5.25)$$

$$\mu_{i,n} = \sin\left(\frac{\pi}{2}(2Z_i - 1)\right) = \frac{R_{i_1 q_n}(0)}{\sqrt{R_{i_1 i_1}(0)R_{q_n q_n}(0)}}$$

Donde el subíndice n (con $n = 2, 3, 4$) denota el canal al cual corresponde la correlación.

Por otra parte, es posible obtener la correlación entre la señal del canal de referencia y el resto de canales como:

$$R_{1i} = \frac{1}{2} E[S_1(t)S_i^*(t)] \quad (5.26)$$

Donde el subíndice i (con $i = 2, 3, 4$) representa el canal al cual corresponde la correlación.

Considerando las señales S_1 y S_2 como señales complejas, es posible expresarlas en función de sus componentes en fase y en cuadratura y desarrollar la ecuación anterior como:

$$R_{1i} = \frac{1}{2} (R_{i_1 i_i}(0) + R_{q_1 q_i}(0)) + j \frac{1}{2} (R_{q_1 i_i}(0) + R_{i_1 q_i}(0)) \quad (5.27)$$

Donde los subíndices i y q , identifican las componentes en fase y en cuadratura respectivamente.

Las señales obtenidas por el radiómetro durante la fase de calibración proceden de la fuente de ruido. Son, por tanto, ruido blanco gaussiano de media cero y cumplen las siguientes propiedades:

$$R_{i_1 i_i}(0) = R_{q_1 q_i}(0)$$

$$R_{q_1 i_i}(0) = -R_{i_1 q_i}(0) \quad (5.28)$$

$$R_{1i} = R_{i_1 i_i}(0) - jR_{i_1 q_i}(0)$$

Aplicando las propiedades del ruido blanco gaussiano descritas en la ecuación 5.28, es posible reescribir la ecuación 5.27 de la siguiente forma:

$$R_{1i} = \sqrt{R_{i_1 i_1}(0)R_{q_i q_i}(0)}(\mu_{r_i} - j\mu_{i_i}) \quad (5.29)$$

Finalmente, para obtener el desfase relativo entre el canal de referencia y el resto de canales α_i , es necesario calcular el arcotangente de los dos elementos de R_{1i} :

$$\theta_1 - \theta_i = \tan^{-1}\left(\frac{-\mu_{i_i}}{\mu_{r_i}}\right) = \alpha_i \quad (5.30)$$

A continuación se detalla este proceso tal y como se ha implementado en la UC:

- o Toda la información referente a la calibración de cada uno de los receptores se almacena en una estructura de datos denominada *mddc* (multiplex digital down converter). Mediante un bucle *while* se capturan las medidas de similitud y se almacenan en las variables correspondientes, separadas según su canal de procedencia:

Pseudocode

```
while calibration data available:
    mddc[receiver].II[channel] = acquire data
    mddc[receiver].IQ[channel] = acquire data
```

- o A partir de las medidas de similitud obtenidas se determinan los coeficientes μ_i y μ_r :

Pseudocode

```
For each channel
    mddc[receiver].Ur[channel] = sin((pi/2)*(2*(mddc[receiver].II[channel] - 0,5)))
    mddc[receiver].Ui[channel] = sin((pi/2)*(2*(mddc[receiver].IQ[channel] - 0,5)))
```

- Mediante estos dos coeficientes se obtiene el error de fase relativo para cada canal:

Pseudocode

```

For each channel
  mddc[receiver].Oer[channel] = arctg(-mddc[receiver].Ui[channel]/mddc[receiver].Ur[channel])
    
```

- El error obtenido es acumulado, es decir se emplea el algoritmo de calibración no diferencial en la variante de respuesta infinita:

Pseudocode

```

For each channel
  acumulated_error[channel] = acumulated_error[channel] + mddc[receiver].Oer[channel]
  mddc[receiver].Onm[channel] = acumulated_error[channel]
    
```

- Finalmente se modifican las funciones de las tablas trigonométricas para corregir el error estimado y se envían a la ALU las nuevas tablas:

Pseudocode

```

For each position of the trigonometric table
  k = dephase coefficient corresponding to the current table position
  tt=128*cos(k*(pi/2)- mddc[receiver].Onm[channel])
  send_to_ALU (tt)
    
```

Una vez completado el proceso de estimación y corrección del error, la siguiente captura de datos del radiómetro se efectuará con las nuevas tablas trigonométricas y, por tanto, el error de fase estará corregido.

5.3.2.2 Implementación mediante correlaciones a 8 bits

Como alternativa a la implementación mediante correlaciones a 1 bit, se desarrolló una versión que emplea 8 bits para calcular las correlaciones. Las ecuaciones básicas que rigen el proceso de estimación y corrección del error de fase mediante correlaciones a 8 bits son exactamente las mismas que para el caso de 1 bit, no obstante la implementación en la unidad de control presenta algunas diferencias.

- En primer lugar se capturan los datos del radiómetro, que en esta ocasión no son medidas de similitud sino directamente correlaciones, ya que se han calculado empleando 8 bits:

Pseudocode

```
while calibration data available:
    mddc[receiver].II[channel] = acquire data
    mddc[receiver].IQ[channel] = acquire data
```

- A partir de los valores de correlación obtenidos se estima, para cada canal, el error de fase relativo al canal de referencia:

Pseudocode

```
For each channel
    mddc[receiver].Oer[channel] = arctg(-mddc[receiver].IQ[channel]/mddc[receiver].II[channel])
```

- El error es acumulado, para aplicar el algoritmo en su forma de respuesta infinita:

Pseudocode

```
For each channel
    acumulated_error[channel] = acumulated_error[channel] + mddc[receiver].Oer[channel]
    mddc[receiver].Onm[channel] = acumulated_error[channel]
```

- Finalmente se modifican los valores de las tablas trigonométricas que controlan los osciladores locales de cada canal, de manera que la siguiente captura de datos ya estará corregida en fase:

Pseudocode

```
For each position of the trigonometric table
    k = dephase coefficient corresponding to the current table position
    tt=128*cos(k*(pi/2)- mddc[receiver].Onm[channel])
    send_to_ALU (tt)
```

Por tanto, la diferencia fundamental de la versión de 1 bit con respecto a la versión de 8 bits, consiste en que en esta última no es necesario calcular las correlaciones a partir de los coeficientes μ_i y μ_r ya que las medidas proporcionadas por la ALU son directamente correlaciones.

El desarrollo de esta versión supone realizar modificaciones en la ALU del radiómetro y su implementación requiere un mayor número de elementos lógicos de

la FPGA. No obstante ofrece una mayor precisión así como una mayor comodidad a la hora de desarrollar los algoritmos en la UC. A no ser que se especifique lo contrario, por defecto se empleará la implementación de 8 bits.

5.3.3 Resultados y conclusiones

Para verificar el correcto funcionamiento del algoritmo de calibración, se utilizó PAU-OR como plataforma de pruebas. Al tratarse de un sistema con un único receptor, la validación de los algoritmos pasa por comprobar que, al inicio del estado de calibración, se estiman correctamente las fases de los cuatro canales del receptor y que, transcurrido un cierto número de iteraciones, estas fases tienden a cero (es decir que la fase del canal de referencia y las fases del resto de canales se han igualado).

Durante el transcurso de la prueba, las dos entradas de antena del receptor (una para la polarización horizontal y otra para la polarización vertical) se conectaron a dos cargas de 50Ω .

Antes de efectuar las pruebas de calibración de fase, se ajustaron los potenciómetros de cada canal en modo *Open Loop* para garantizar que los cuatro canales presentasen la misma ganancia. De esta forma se consigue minimizar el impacto de las variaciones de ganancia sobre los resultados de la calibración de fase.

La figura 5-19 muestra los resultados de la fase resultante después del proceso de calibración para el receptor de referencia empleando la implementación a 8 bits con respuesta infinita. El tiempo de integración para cada muestra empleado para esta prueba (y para el resto de pruebas a no ser que se especifique lo contrario) es el tiempo de integración por defecto para calibración: 0,54 segundos.

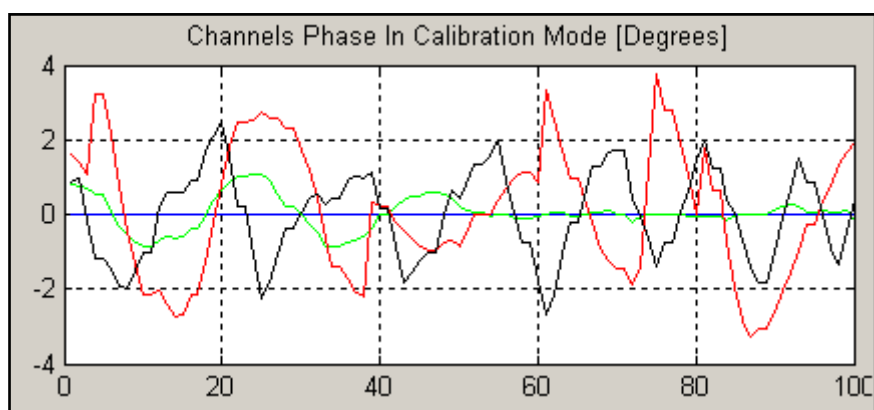


Figura 5-19: Fase residual resultante del proceso de calibración.

Como se puede observar, las fases resultantes del proceso de calibración para los cuatro canales oscilan en torno al cero, que constituye la fase del canal de

referencia. Se puede afirmar por tanto que el error de fase relativo al canal de referencia ha sido corregido.

También es necesario comprobar que los parámetros de Stokes obtenidos para estos valores de error de fase residual son correctos. La figura 5-20 muestra los parámetros de Stokes una vez aplicada la calibración:

Figura 5-20: Partes real e imaginaria del primer (azul y rojo) y segundo (verde y negro) parámetros de Stokes y módulos del primer (cian) y segundo (amarillo) parámetros de Stokes.

Las trazas de color azul y rojo corresponden a las partes real e imaginaria respectivamente del primer parámetro de Stokes, mientras que las trazas de color verde y negro corresponden a las partes real e imaginaria del segundo parámetro de Stokes. Por otra parte, las trazas cian y amarillo corresponden a los módulos del primer y segundo parámetro de Stokes respectivamente.

Para garantizar que los parámetros de Stokes obtenidos están calibrados en fase, es necesario que las partes reales de ambos (trazas azul y verde) sean distintas de cero y que las partes imaginarias (trazas roja y negra) sean iguales a cero. Como se puede comprobar, no se cumplen estas condiciones, por tanto el primer y segundo parámetro de Stokes, no están calibrados en fase.

Es posible calcular las fases correspondientes a cada uno de los parámetros de Stokes a partir de los valores obtenidos para la parte real e imaginaria como:

$$\begin{aligned} \alpha_{S_1} &= \tan^{-1} \left(\frac{3072}{1479} \right) = 64^\circ \\ \alpha_{S_2} &= \tan^{-1} \left(\frac{-1789}{-1154} \right) = 57^\circ \end{aligned} \tag{5.31}$$

Los valores de fase obtenidos distan del valor teórico que deberían tener, ya que si los cuatro canales están calibrados en fase tal y como sugiere la figura 5-19, la fase de los dos primeros parámetros de Stokes debería ser próxima a cero.

Estos mismos resultados se repiten para cualquiera de las posibles variaciones del algoritmo, es decir: con respuesta finita o infinita, o empleando una implementación de 1 u 8 bits.

Después de numerosas pruebas se consideró la posibilidad de que la presencia de un cierto nivel de offset fuese la responsable de los resultados obtenidos. En caso de que los niveles de offset fuesen mayores que los niveles de señal, el error de fase estimado correspondería en su mayor parte a la fase introducida por el offset, con lo que el verdadero error de fase relativo quedaría enmascarado por la fase del offset. Para corregir el efecto del offset en la estimación de fase, se desarrollaron los algoritmos diferenciales.

5.3.4 Calibración de fase mediante algoritmo diferencial

El desarrollo de los algoritmos de calibración diferenciales surge a partir de la hipótesis de la presencia de cierto nivel de offset en las medidas obtenidas del radiómetro. El proceso de calibración diferencial se basa en la utilización de dos niveles diferentes de potencia de calibración para eliminar este offset.

Si se considera el nuevo modelo de sistema, que contempla la presencia de offsets, se tiene que las ecuaciones que expresan la salida de cada uno de los canales de un receptor son:

$$\begin{aligned}
 S_1 &= I_1 + I_{1OFF} + j(Q_1 + Q_{1OFF}) = |A_1|e^{j\theta_1} + |O_1|e^{j\theta_{o1}} \\
 S_2 &= I_2 + I_{2OFF} + j(Q_2 + Q_{2OFF}) = |A_2|e^{j\theta_2} + |O_2|e^{j\theta_{o2}} \\
 S_3 &= I_3 + I_{3OFF} + j(Q_3 + Q_{3OFF}) = |A_3|e^{j\theta_3} + |O_3|e^{j\theta_{o3}} \\
 S_4 &= I_4 + I_{4OFF} + j(Q_4 + Q_{4OFF}) = |A_4|e^{j\theta_4} + |O_4|e^{j\theta_{o4}}
 \end{aligned}
 \tag{5.32}$$

Donde I y Q denotan las componentes en fase y cuadratura respectivamente de la señal, el subíndice OFF indica que la componente en cuestión pertenece al offset, $|O|$ representa el módulo del offset y θ_o la fase del offset.

Se definen los dos niveles de potencia para cada canal como:

$$S_{i_{MAX}} = I_{i_{MAX}} + I_{i_{OFF}} + j(Q_{i_{MAX}} + Q_{i_{OFF}}) \quad (5.33)$$

$$S_{i_{MIN}} = I_{i_{MIN}} + I_{i_{OFF}} + j(Q_{i_{MIN}} + Q_{i_{OFF}})$$

Donde el subíndice i denota el canal al cual corresponden las señales, el subíndice MAX indica señal correspondiente al nivel de potencia alto (para una temperatura de referencia interna $T_{max} = 500$ K) i el subíndice MIN indica señal correspondiente al nivel de potencia bajo (para una temperatura de referencia interna $T_{min} = 300$ K).

Se ha considerado que el offset introducido por el sistema es constante, y no varía con la potencia de entrada, por tanto los offsets para los dos niveles de potencia son iguales. En el caso de que esto no fuese así y el sistema presentase efectos no lineales, el problema se complicaría sobremanera, haciéndolo irresoluble mediante la aplicación de ningún tipo de algorítmica.

Calculando las diferencias entre ambas señales para cada uno de los cuatro canales, es posible cancelar las componentes en fase y cuadratura del offset:

$$S_{i_{MAX}} - S_{i_{MIN}} = I_{i_{MAX}} - I_{i_{MIN}} + j(Q_{i_{MAX}} - Q_{i_{MIN}}) = \Delta I_i + j\Delta Q_i = \Delta S_i \quad (5.34)$$

Una vez cancelados los offsets, si se calculan las correlaciones cruzadas entre la señal diferencial del canal de referencia y las señales diferenciales del resto de canales conjugadas, tal y como se hizo para el caso del algoritmo no diferencial, se tiene que:

$$\langle \Delta S_1 \Delta S_2^* \rangle = (\langle \Delta I_1 \Delta I_2 \rangle + \langle \Delta Q_1 \Delta Q_2 \rangle) + j(\langle \Delta Q_1 \Delta I_2 \rangle - \langle \Delta I_1 \Delta Q_2 \rangle)$$

$$\langle \Delta S_1 \Delta S_3^* \rangle = (\langle \Delta I_1 \Delta I_3 \rangle + \langle \Delta Q_1 \Delta Q_3 \rangle) + j(\langle \Delta Q_1 \Delta I_3 \rangle - \langle \Delta I_1 \Delta Q_3 \rangle) \quad (5.35)$$

$$\langle \Delta S_1 \Delta S_4^* \rangle = (\langle \Delta I_1 \Delta I_4 \rangle + \langle \Delta Q_1 \Delta Q_4 \rangle) + j(\langle \Delta Q_1 \Delta I_4 \rangle - \langle \Delta I_1 \Delta Q_4 \rangle)$$

Aplicando las propiedades de la correlación del ruido gaussiano de media cero, es posible obtener las fases de las correlaciones cruzadas de la ecuación 5.35 a través de la inversa de la tangente:

$$\theta_{12} = \arctg\left(\frac{-\langle \Delta I_1 \Delta Q_2 \rangle}{\langle \Delta I_1 \Delta I_2 \rangle}\right) = \theta_1 - \theta_2 = \alpha_2 \quad (5.36)$$

$$\theta_{13} = \arctg\left(\frac{-\langle\Delta I_1 \Delta Q_3\rangle}{\langle\Delta I_1 \Delta I_3\rangle}\right) = \theta_1 - \theta_3 = \alpha_3$$

$$\theta_{14} = \arctg\left(\frac{-\langle\Delta I_1 \Delta Q_4\rangle}{\langle\Delta I_1 \Delta I_4\rangle}\right) = \theta_1 - \theta_4 = \alpha_4$$

De forma que a partir de las correlaciones cruzadas de las señales diferenciales es posible obtener los coeficientes de corrección de fase siguiendo el mismo procedimiento que en el caso de los algoritmos no diferenciales. La diferencia radica que en que mediante la aplicación de este método se elimina la contribución de la fase del offset a la estimación del error de fase.

Una vez obtenidos los coeficientes de corrección de fase, modificando las tablas trigonométricas se puede calibrar la fase de los cuatro canales de la misma forma que para el caso del algoritmo no diferencial:

$$\begin{aligned} S_2' &= S_2 e^{j\alpha_2} = |A_1| e^{j\theta_1} + |O_2| e^{j(\theta_{o_2} + \theta_1 - \theta_2)} \\ S_3' &= S_3 e^{j\alpha_3} = |A_3| e^{j\theta_1} + |O_3| e^{j(\theta_{o_3} + \theta_1 - \theta_2)} \\ S_4' &= S_4 e^{j\alpha_4} = |A_4| e^{j\theta_1} + |O_4| e^{j(\theta_{o_4} + \theta_1 - \theta_2)} \end{aligned} \quad (5.37)$$

Los cuatro canales de cada receptor estarán, por tanto, igualados en fase a la fase del canal de referencia. Esta solución corrige las diferencias de fase relativas al canal de referencia, pero no elimina el offset (como se puede apreciar en la ecuación 5.37), por tanto, para obtener una calibración completa, es necesario estimar y eliminar el offset de los parámetros de Stokes obtenidos mediante este proceso de calibración.

Para estimar el offset introducido por el sistema, se actúa sobre el conmutador de entrada, seleccionando como entrada para cada una de las dos cadenas la resistencia interna de 50 Ω (ver figura 5-1). Puesto que la salida depende de la diferencia entre la temperatura equivalente de la entrada (en este caso la resistencia interna de 50 Ω) y la temperatura equivalente del divisor de Wilkinson, y ambos elementos se encuentran a la misma temperatura, cualquier medida diferente de cero a la salida corresponderá al offset. Este proceso de estimación del offset de los parámetros de Stokes se lleva a cabo en el estado *Internal Reference Uncorrelated* de la *Secuencia Principal* de ejecución de la UC (ver capítulo 3).

5.3.5 Implementación en la Unidad de Control

La implementación del algoritmo diferencial en la Unidad de Control de PAU-RAD se lleva a cabo en el estado *Internal Reference Correlated* de la *Secuencia Principal* de ejecución de la UC (ver capítulo 3).

Se testearon dos implementaciones del algoritmo: una primera versión que ejecuta la secuencia de procesos descrita en el apartado anterior y una segunda versión que lleva a cabo una estimación inicial del offset y lo elimina de las medidas para finalmente aplicar el mismo procedimiento que en la primera versión.

5.3.5.1 Primera implementación

La secuencia de acciones que se ejecuta durante la implementación inicial de la calibración diferencial es la siguiente:

- En primer lugar se selecciona la temperatura de referencia de manera que durante la primera mitad de las iteraciones del estado la temperatura de referencia se corresponda con T_{\max} y durante la segunda mitad con T_{\min} :

Pseudocode

```
Every new calibration cycle:
  if ( numero_ciclos_cal <= iteraciones/2 )
  {
    select Tmax reference temperature
  }

  else if ( numero_ciclos_cal > iteraciones/2 )
  {
    select Tmin reference temperature
  }
```

- A continuación se lleva a cabo la captura de los datos procedentes de ALU del radiómetro. La información se almacena en la estructura PCE, que guarda por separado las correlaciones de fase y cuadratura obtenidas para T_{\max} y las obtenidas para T_{\min} . Estos valores de correlación para ambos niveles de potencia se acumulan para posteriormente promediar por el número de muestras obtenidas:

Pseudocode

```

For each channel:
  if ( (numero_ciclos_cal) <= (iteraciones/2) )
  {
    PCE.tmaxii[channel] = acquire_data_from_ALU();
    PCE.tmaxiimean[channel] = PCE.tmaxiimean[channel] + PCE.tmaxii[channel];
    PCE.tmaxiq[channel] = acquire_data_from_ALU();
    PCE.tmaxiqmean[channel] = PCE.tmaxiqmean[channel] + PCE.tmaxiq[channel];
  }
  else if ( (numero_ciclos_cal) > (iteraciones/2) )
  {
    PCE.tminii[channel] = acquire_data_from_ALU();
    PCE.tminiimean[channel] = PCE.tminiimean[channel] + PCE.tminii[channel];
    PCE.tminiq[channel] = acquire_data_from_ALU();
    PCE.tminiqmean[channel] = PCE.tminiqmean[channel] + PCE.tminiq[channel];
  }
  }
    
```

- En la última iteración del estado se lleva a cabo el promediado de las muestras obtenidas para ambas temperaturas de referencia, en función del número de iteraciones efectuadas:

Pseudocode

```

For each channel:
  if (numero_ciclos_cal == iteraciones)
  {
    PCE.tmaxiimean[channel] = PCE.tmaxiimean[channel]/iteraciones/2;
    PCE.tmaxiqmean[channel] = PCE.tmaxiqmean[channel]/iteraciones/2;

    PCE.tminiimean[channel] = PCE.tminiimean[channel]/iteraciones/2;
    PCE.tminiqmean[channel] = PCE.tminiqmean[channel]/iteraciones/2;
  }
  }
    
```

- A partir de los valores promediados se calculan las diferencias entre las correlaciones obtenidas para ambas temperaturas de referencia y se obtienen el error de fase instantáneo:

Pseudocode

```

if (numero_ciclos_cal == iteraciones)
  For each channel:
    PCE.numerator[channel] = PCE.tmaxiqmean[channel] - PCE.tminiqmean[channel];
    PCE.denominator[channel] = PCE.tmaxiimean[channel] - PCE.tminiimean[channel];

    mddc[receiver].Oer[channel] = atan(PCE.numerator[channel]/PCE.denominator[channel]);
    
```

- El error instantáneo obtenido es acumulado para aplicar un algoritmo de respuesta infinita:

Pseudocode

```

if (numero_ciclos_cal == iteraciones)
  For each channel:
    total_acumuladed_error[channel] = total_acumuladed_error[channel] + mddc[receiver].Oer[channel];
    mddc[receiver].Onm[channel] = total_acumuladed_error[channel];

```

- Finalmente se modifican los valores de las TT con los coeficientes obtenidos, de manera que la siguiente captura de datos se llevará a cabo con las TT que corrigen el error de fase:

Pseudocode

```

For each position of the trigonometric table
  k = dephase coefficient corresponding to the current table position
  tt=128*cos(k*(pi/2)- mddc[receiver].Onm[channel])
  send_to_ALU (tt)

```

5.3.5.2 Segunda implementación

Como complemento a la primera implementación y para dotar al sistema de un mayor rango de posibilidades y mayor flexibilidad, se desarrolló una segunda versión del algoritmo.

La principal diferencia con respecto a la primera versión consiste en que se lleva a cabo una estimación inicial del offset de la correlación. El valor de offset obtenido se resta a las correlaciones obtenidas para cada una de las temperaturas de referencia y finalmente se aplica el algoritmo diferencial empleando correlaciones libres de offset.

A continuación se detalla la secuencia de procesos para esta implementación:

- Se selecciona como entrada la resistencia interna de 50Ω y se capturan las medidas de correlación que corresponderán al offset. Para conseguir una estimación de mayor precisión del offset, el proceso se itera un determinado número de veces definido por el usuario (por defecto se realizan 10 iteraciones) y se promedia el resultado:

Pseudocode

```

Select internal reference uncorrelated as input
for(i=0; i<iteraciones_offset; i++)
  For each channel
    PCE.ii_offset[channel] = acquire_data_from_ALU();
    PCE.ii_offset_mean[channel] = PCE.ii_offset_mean[channel] + PCE.ii_offset[channel];

    PCE.iq_offset[channel] = acquire_data_from_ALU();
    PCE.iq_offset_mean[channel] = PCE.iq_offset_mean[channel] + PCE.iq_offset[channel];

PCE.ii_offset[channel] = PCE.ii_offset_mean[channel]/iteraciones_offset;
PCE.iq_offset[channel] = PCE.iq_offset_mean[channel]/iteraciones_offset;

```

- A continuación se selecciona la entrada de ruido correlado y se lleva a cabo la captura de las correlaciones proporcionadas por la ALU para ambas temperaturas de referencia. El proceso empleado es el mismo que el descrito para los dos primeros pasos de la primera implementación.
- A estas correlaciones se les restan los valores de offset obtenidos en el primer paso:

Pseudocode

```

For each channel
  PCE.tmaxii[channel] = PCE.tmaxii[channel] - PCE.ii_offset[channel];
  PCE.tmaxiq[channel] = PCE.tmaxiq[channel] - PCE.iq_offset[channel];

  PCE.tminii[channel] = PCE.tminii[channel] - PCE.ii_offset[channel];
  PCE.tminiq[channel] = PCE.tminiq[channel] - PCE.iq_offset[channel];

```

- A partir de este punto el algoritmo es idéntico al descrito para la primera implementación, pero se aplica sobre las correlaciones a las que se les ha eliminado el offset.

Evidentemente esta nueva implementación es mucho más lenta que la primera, ya que al inicio de cada nuevo ciclo de calibración requiere un cierto número de iteraciones (10 por defecto) para estimar el offset de las correlaciones.

5.3.6 Resultados y conclusiones

Los resultados obtenidos para ambas implementaciones del algoritmo diferencial coinciden con los obtenidos para el algoritmo no diferencial. La siguiente figura muestra el error de fase residual resultante de aplicar la primera implementación:

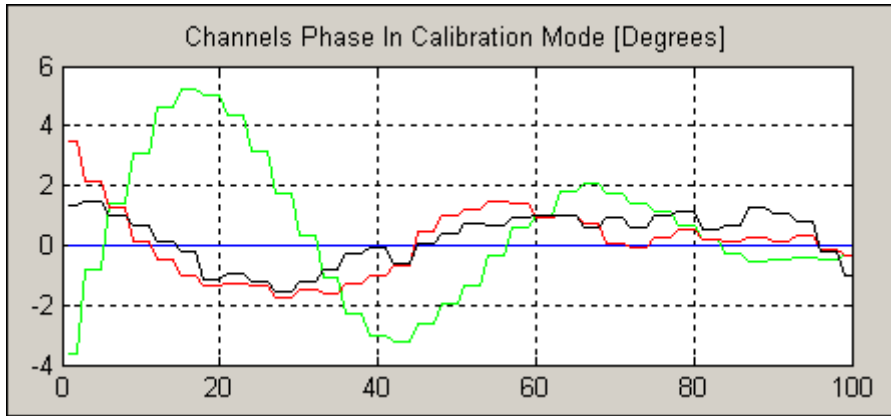


Figura 5-21: Error de fase residual tras aplicar un algoritmo de calibración diferencial.

El primer hecho a destacar es que la conversión del algoritmo es mucho más lenta que para el caso no diferencial. Esto no es de extrañar, ya que, a pesar de que cada ciclo de calibración se itera, por defecto, el mismo número de veces que en el caso del algoritmo no diferencial, estas iteraciones se emplean en obtener las medidas de ambos niveles de referencia y es únicamente en la última iteración cuando se lleva a cabo la estimación y corrección del error de fase. En el algoritmo no diferencial, por el contrario, el error se estima y se corrige en cada iteración.

Al margen de la velocidad de convergencia del algoritmo se puede observar como el error residual tiende a cero para los cuatro canales, es decir que la fase de los cuatro canales se ha igualado y por tanto el sistema está calibrado.

A pesar de esto, se produce el mismo efecto que para el caso del algoritmo no diferencial, es decir: las fases de los parámetros de Stokes obtenidos con las TT corregidas no se corresponden con los valores del error de fase residual, y por tanto los parámetros de Stokes no están calibrados en fase. Este efecto se ilustra en la siguiente figura, que muestra una captura de los parámetros de Stokes obtenidos empleando las fases residuales de la figura 5-21:

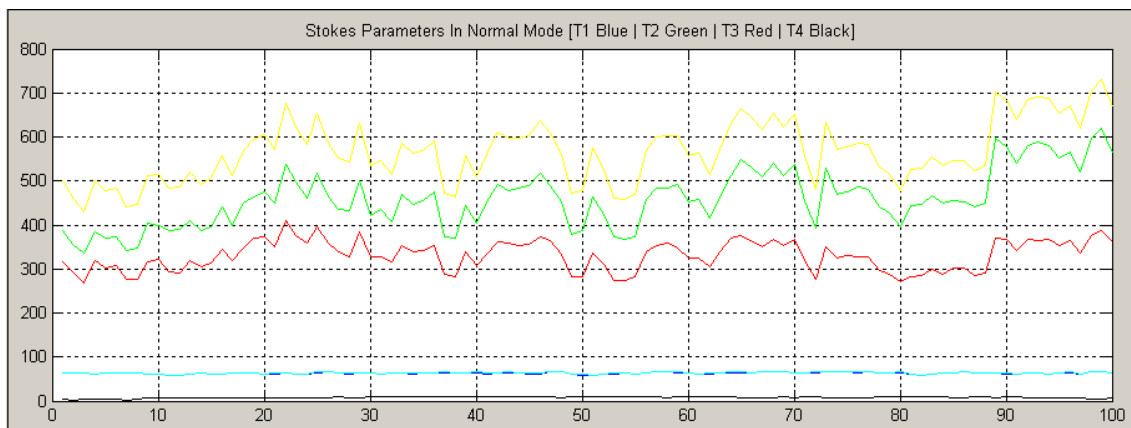


Figura 5-22: Parámetros de Stokes obtenidos tras aplicar la corrección del error de fase.

Los resultados obtenidos son los mismos tanto para la primera como para la segunda implementación del algoritmo.

El resultado final de las pruebas es que, todos los algoritmos testeados, ya sean diferenciales o no diferenciales, en cualquiera de sus variantes e implementaciones, son capaces de estimar y corregir, con mayor o menor precisión y rapidez, el error de fase relativo. No obstante, una vez finalizado el proceso de calibración y conmutada la entrada para obtener medidas procedentes de la antena, esta corrección no se aprecia en los parámetros de Stokes obtenidos, que continúan presentando errores de fase.

5.3.7 Pruebas de validación de los algoritmos

Llegados a este punto, se llevaron a cabo diversas pruebas con el fin de determinar si los errores obtenidos en el proceso de calibración obedecían a una implementación incorrecta de los algoritmos o por el contrario se debían a la utilización de un modelo impreciso del sistema.

Para verificar el correcto funcionamiento de los algoritmos de estimación y corrección de fase, se aisló la parte digital de PAU (ADC y FPGA) de la analógica (antena y receptor). Para ello se sustituyó el receptor por una placa de adaptación, a la cual se inyectó señal procedente de un generador de funciones. La figura 5-23 muestra una fotografía de la placa de adaptación [5]:



Figura 5-23: Placa de adaptación que permite inyectar señal sintética a cada uno de los cuatro canales por separado y llevarla al ADC.

Mediante este montaje la señal procedente del generador llega a los cuatro canales del ADC con la misma fase. Para forzar un error de fase conocido, se insertó un cable de cierta longitud entre la salida del generador de funciones y la entrada del

canal cuatro de la placa de adaptación. El desfase del cable a la frecuencia de trabajo de PAU, medido con el analizador de redes vectorial, es de aproximadamente 149° .

La figura 5-24 muestra el error de fase residual resultante aplicando un algoritmo no diferencial de respuesta infinita para el montaje descrito:

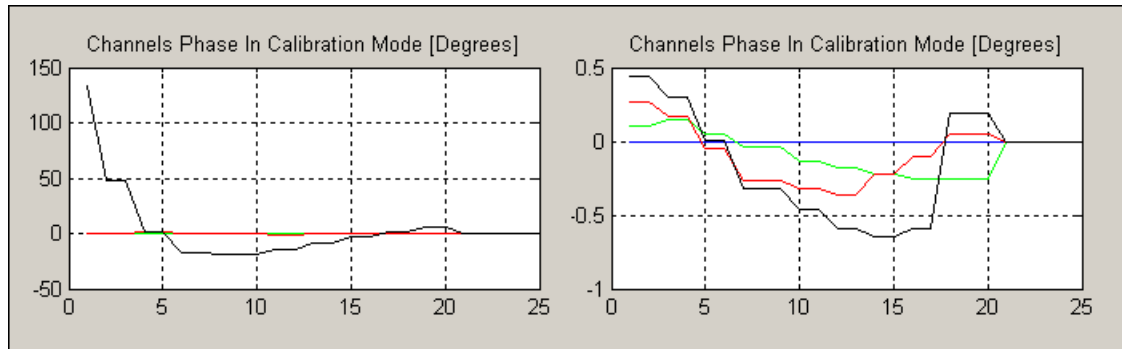


Figura 5-24: Error de fase residual aplicando algoritmo no diferencial de respuesta infinita en ausencia del receptor. A la izquierda al inicio de la secuencia *Closed Loop* (calibración continua) y a la derecha después de un cierto número de iteraciones de la secuencia.

Como se puede apreciar en la figura de la izquierda, la estimación del error de fase del canal cuatro al inicio de la secuencia *Closed Loop* (traza negra) coincide con los 149° introducidos por el cable, mientras que para los otros tres canales el error de fase es cero, puesto que las señales llegan al ADC en fase. Transcurridas 20 iteraciones del proceso de calibración, el error de fase del cuarto canal se reduce hasta quedar comprendido entre $0,5^\circ$ y $-0,5^\circ$. Por tanto, se puede afirmar que tanto la estimación como la corrección del error son satisfactorias.

Los parámetros de Stokes obtenidos para esta configuración se muestran en la siguiente figura:

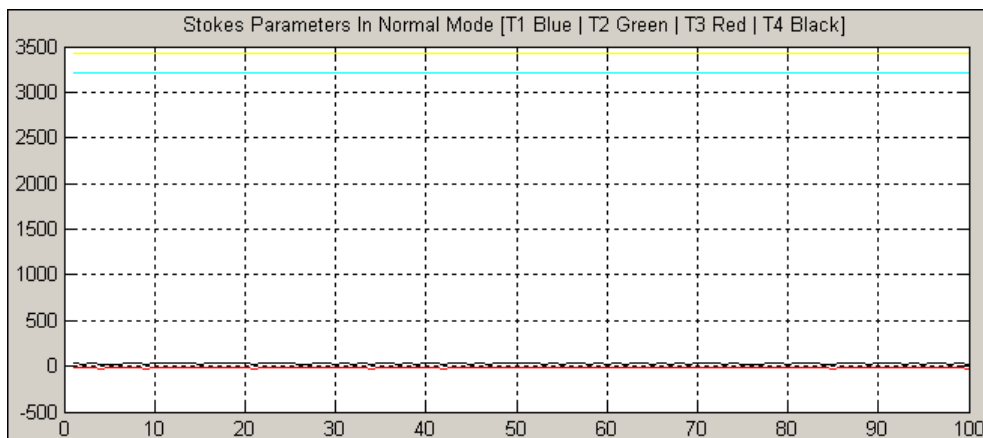


Figura 5-25: Parámetros de Stokes obtenidos en ausencia del receptor aplicando un algoritmo de corrección de fase no diferencial.

Como se puede apreciar, las trazas que corresponden las partes imaginarias del primer y segundo parámetro de Stokes (trazas roja y negra respectivamente) tienen valores próximos a cero, mientras que las trazas correspondientes a las partes reales del primer y segundo parámetro (trazas azul y verde) presentan valores diferentes de cero y próximos entre sí. En la figura no se aprecian las trazas azul y verde debido a que se hallan enmascaradas por las trazas cian y amarillo (que corresponden al módulo del primer y segundo parámetro de Stokes respectivamente). La siguiente figura muestra una captura en la que se pueden leer los valores de las cuatro trazas (en orden descendente: azul, verde, rojo y negro):

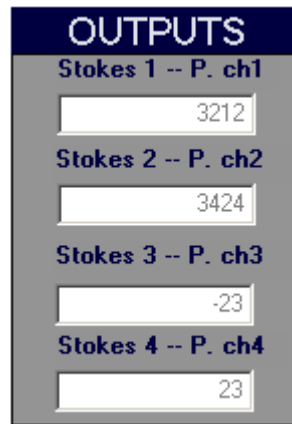


Figura 5-26: Valores numéricos de los parámetros de Stokes representados en la figura 5-25.

A partir de estos valores es posible determinar la fase del primer y segundo parámetro de Stokes como:

$$\alpha S_1 = \tan^{-1}\left(\frac{-23}{3212}\right) = -0,41^\circ \quad (5.38)$$

$$\alpha S_2 = \tan^{-1}\left(\frac{23}{3424}\right) = 0,38^\circ$$

Estos resultados de fase coinciden con los valores de error de fase residual obtenidos en la figura 5-25. Por tanto, se puede afirmar que el algoritmo estima y corrige de forma adecuada el error de fase relativo, es decir, lleva a cabo la calibración de fase del sistema.

Como ya se ha mencionado, estos resultados se obtuvieron empleando un algoritmo no diferencial. No obstante, se obtienen resultados similares empleando un algoritmo diferencial, lo que prueba la implementación de los algoritmos es correcta y que el modelo empleado en su desarrollo no se corresponde con el hardware del receptor.

5.3.8 Pruebas de validación del hardware

Verificado el correcto funcionamiento de los algoritmos de estimación y corrección del error de fase en ausencia del receptor, se trabajó bajo la hipótesis de que los modelos del receptor considerados a la hora de desarrollar las ecuaciones de calibración no eran del todo correctos. Resultaba evidente que existía un aspecto del hardware del radiómetro que introducía una particularidad que no había sido considerada.

5.3.8.1 Determinación de los coeficientes de desfase entre entradas

Para poder determinar esta particularidad del hardware que invalidaba los algoritmos desarrollados, se llevó a cabo la siguiente experiencia: se sustituyó el control de los conmutadores de entrada, gobernado de forma automática por la ALU, por un micro switch hardware externo que permitía realizar las conmutaciones de forma manual. Mediante esta configuración es posible mantener durante toda la secuencia principal de ejecución una única entrada, evitando así la conmutación entre las entradas de calibración, ruido incorrelado y antena que, en circunstancias normales se lleva a cabo automáticamente.

La figura 5-27 muestra los resultados para una de las secuencias de conmutación manual:

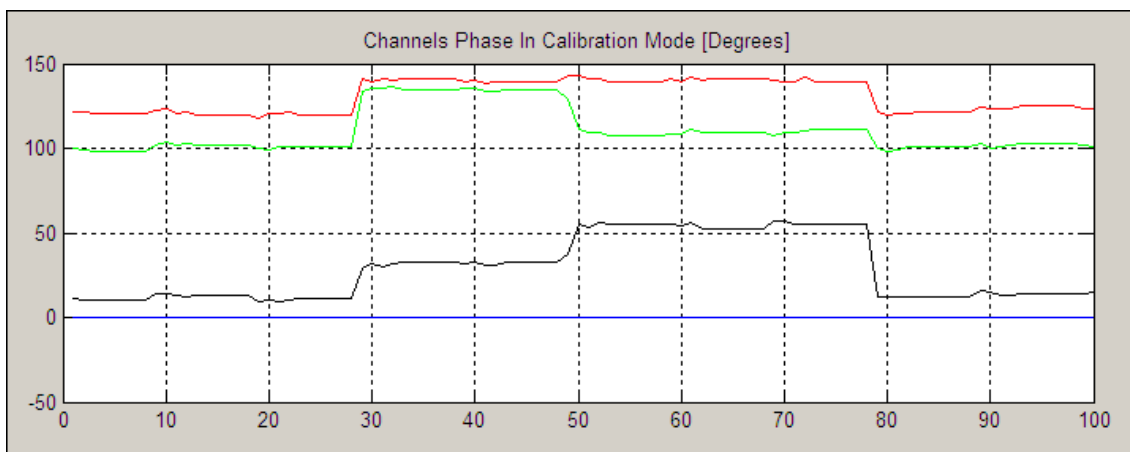


Figura 5-27: Errores de fase estimados durante las pruebas de conmutación manual. Los saltos que se aprecian corresponden a la secuencia de selección de entrada: entrada de calibración con ruido correlado – ruido incorrelado procedente de carga interna – entrada de antena – entrada de calibración con ruido correlado.

Durante el transcurso de esta prueba no se efectuó ningún tipo de calibración, de manera que los resultados obtenidos constituyen directamente el error de fase estimado para cada una de las tres entradas posibles.

Como se puede apreciar en la figura, se producen saltos en el valor de la fase estimada en función de la entrada seleccionada. El primer nivel de fase estimada corresponde a la entrada de calibración con ruido correlado, el segundo a la entrada de ruido incorrelado procedente de una carga de 50Ω interna, el tercero a la entrada de antena y, por último, el cuarto nivel de fase corresponde de nuevo a la entrada de calibración.

Por tanto, el funcionamiento incorrecto de los algoritmos de calibración de fase se debe a las diferencias en la fase estimada en función de la entrada. Una inspección detallada de los receptores reveló que estas diferencias de fase se deben a diferencias en las longitudes de las pistas para cada una de las tres entradas.

Si el error de fase estimado durante el proceso de calibración no se corresponde con el error de fase real existente durante el proceso de adquisición de datos a través de la antena, es lógico que los parámetros de Stokes obtenidos sean incorrectos, puesto que no han sido calibrados.

La solución a este problema pasa por determinar el coeficiente que corrige la diferencia de fase existente entre la entrada de correlación y la entrada de antena para los cuatro canales de cada receptor. Esta diferencia de fase entre entradas es constante, puesto que se trata de una característica del hardware.

Los coeficientes de desfase entre entradas para cada canal fueron estimados a partir de las medidas de fase obtenidas para las entradas de correlación y antena. Para cada entrada se calculó la diferencia entre la fase obtenida para cada canal y la fase del canal de referencia. Finalmente se calculó la diferencia entre la fase obtenida para la entrada de calibración y la fase obtenida para la entrada de antena. Para proporcionar mayor robustez al resultado este proceso se iteró diez veces y los resultados se promediaron.

A continuación se muestran los resultados obtenidos para el receptor de referencia utilizado durante las pruebas:

Tabla 5-3: Diferencias de fase entre las entradas de calibración y antena para el receptor de referencia.

	$\theta_{CORR_2} - \theta_{ANT_2}$	$\theta_{CORR_3} - \theta_{ANT_3}$	$\theta_{CORR_4} - \theta_{ANT_4}$
Iteración 1	-4,83 °	-55,30 °	-65,00 °
Iteración 2	-5,51 °	-58,61 °	-64,10 °
Iteración 3	-0,30 °	-54,31 °	-60,66 °

Iteración 4	-1,28 °	-52,69 °	-62,42 °
Iteración 5	-3,03 °	-54,90 °	-64,20 °
Iteración 6	-4,51 °	-57,21 °	-61,90 °
Iteración 7	-3,57 °	-54,22 °	-62,68 °
Iteración 8	-3,40 °	-54,50 °	-59,64 °
Iteración 9	-5,40 °	-53,93 °	-59,49 °
Iteración 10	-3,66 °	-55,06 °	-60,68 °

Promediando los resultados obtenidos para las diez iteraciones, se tiene que los coeficientes de desfase entre entradas para cada uno de los tres canales a corregir son:

$$\Delta\theta_2 = -3,55^\circ$$

$$\Delta\theta_3 = -55,07^\circ \quad (5.39)$$

$$\Delta\theta_4 = -62,08^\circ$$

Y las varianzas de cada uno de los coeficientes, determinadas a partir de las diez iteraciones son:

$$\text{var}(\Delta\theta_2) = 2,87^\circ$$

$$\text{var}(\Delta\theta_3) = 2,86^\circ \quad (5.40)$$

$$\text{var}(\Delta\theta_4) = 3,79^\circ$$

Estos valores fueron confirmados midiendo los desfases del receptor mediante un analizador de redes vectorial.

Sumando estos coeficientes al error de fase obtenido durante el proceso de calibración y corrigiendo en las TT la suma de ambos desfases, la calibración de fase resultante es correcta.

A continuación se muestran los resultados obtenidos aplicando un algoritmo de calibración de fase no diferencial y respuesta infinita conjuntamente con los coeficientes de desfase entre entradas obtenidos empíricamente:

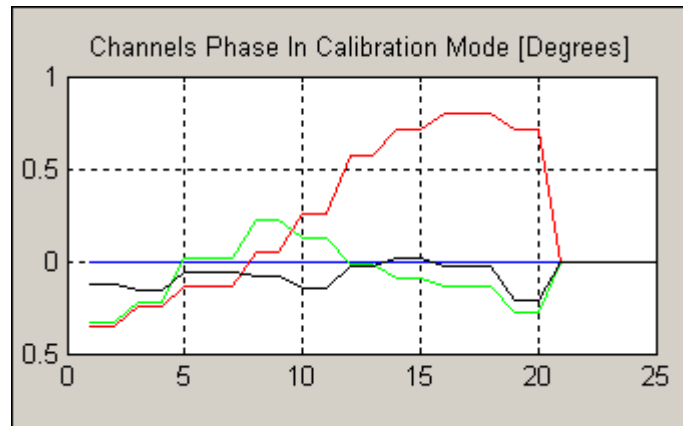


Figura 5-28: Error de fase residual resultante después de aplicar un algoritmo de calibración de fase no diferencial y de respuesta infinita considerando los coeficientes de desfase entre entradas.

Como se puede comprobar, el error de fase residual de los cuatro canales oscila en torno al cero, por tanto los errores de fase relativos a la fase del canal de referencia han sido corregidos.

Los parámetros de Stokes obtenidos para los valores de fase de la figura anterior se muestran en la figura 5-29:

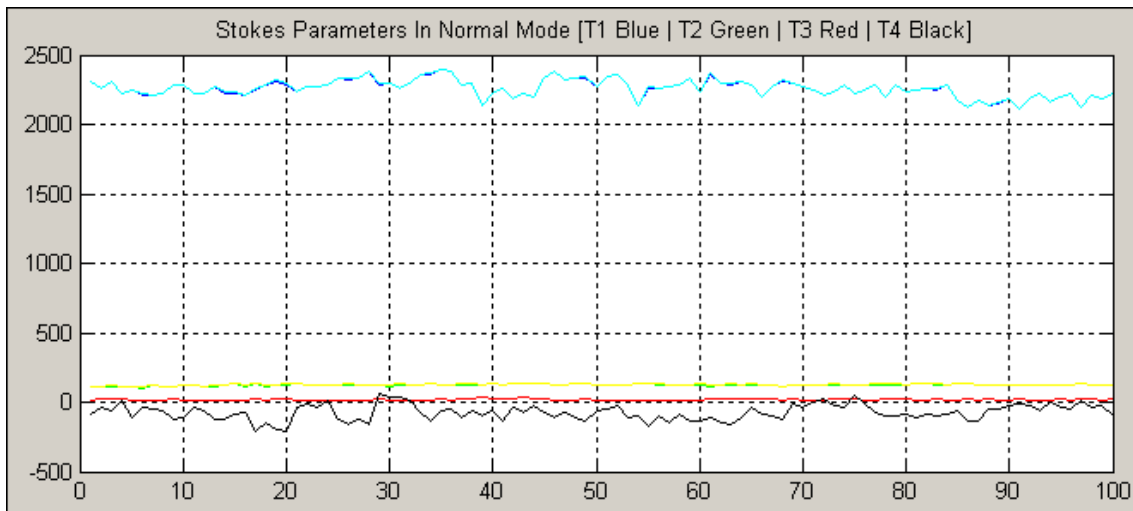


Figura 5-29: Partes real e imaginaria del primer (azul y rojo) y segundo (verde y negro) parámetro de Stokes y módulos del primer (cian) y segundo (amarillo) parámetro de Stokes.

En esta ocasión las partes imaginarias de ambos parámetros de Stokes (trazas roja y negra) son próximas a cero, mientras que las partes reales (trazas azul y verde) son diferentes de cero, por tanto la fase de los parámetros de Stokes será próxima a cero, coincidiendo con los valores de error de fase residual de la figura 5-28. Se puede afirmar, por tanto, que el sistema se encuentra calibrado en fase.

Por tanto, antes de llevar a cabo una campaña de medidas será necesario parametrizar cada uno de los receptores para determinar sus coeficientes de desfase entre entradas, y añadir estos coeficientes a la implementación del algoritmo.

5.3.8.2 Variación de la fase estimada en función de la potencia de entrada

Otro efecto significativo del hardware a tener en cuenta es que la fase estimada durante el proceso de calibración presenta cierta dependencia con respecto a la potencia de entrada.

La figura 5-30 muestra la diferencia de fase estimada entre el canal de referencia y el resto de canales (la traza verde para el canal 2, la traza roja para el canal 3 y la traza negra para el canal 4) del receptor de referencia de PAU-OR, en función de la temperatura de ruido equivalente inyectada a través de la entrada de antena del receptor de referencia.

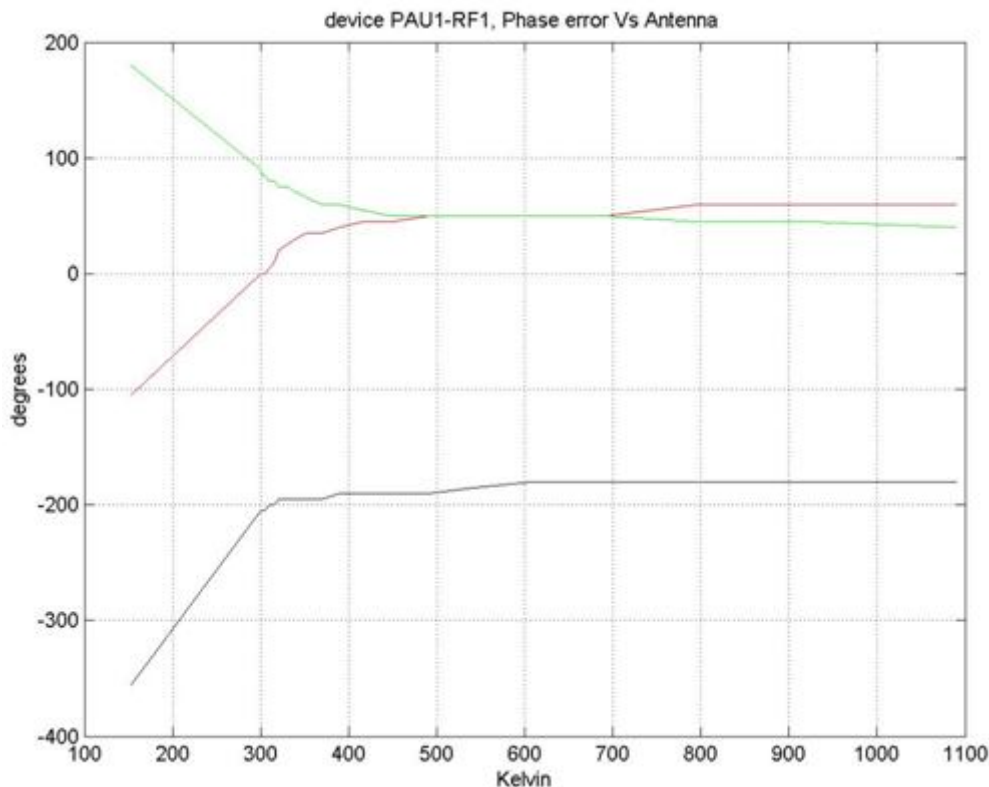


Figura 5-30: Diferencias de fase relativas al canal de referencia estimadas en función de la temperatura de ruido equivalente inyectada a través de la antena de PAU-OR. La traza verde corresponde a la diferencia de fase del canal 2, la traza roja al canal 3 y la negra al canal 4.

La zona de temperaturas equivalentes inferiores a 300 Kelvin fue creada empleando una carga fría activa [13], mientras que para el resto de temperaturas se utilizó la fuente de ruido conjuntamente con un atenuador variable. Como se puede observar, las diferencias de fase estimadas experimentan un salto de aproximadamente 180° en la zona comprendida entre los 200 K y los 400 K para los tres canales. Alcanzados los 700 K, las diferencias de fase se estabilizan en sus valores reales.

Si se repite el mismo procedimiento, pero en este caso inyectando potencia a través de la entrada de calibración en lugar de hacerlo a través de la entrada de antena, se obtienen los mismo resultados, tal y como se muestra en la siguiente figura:

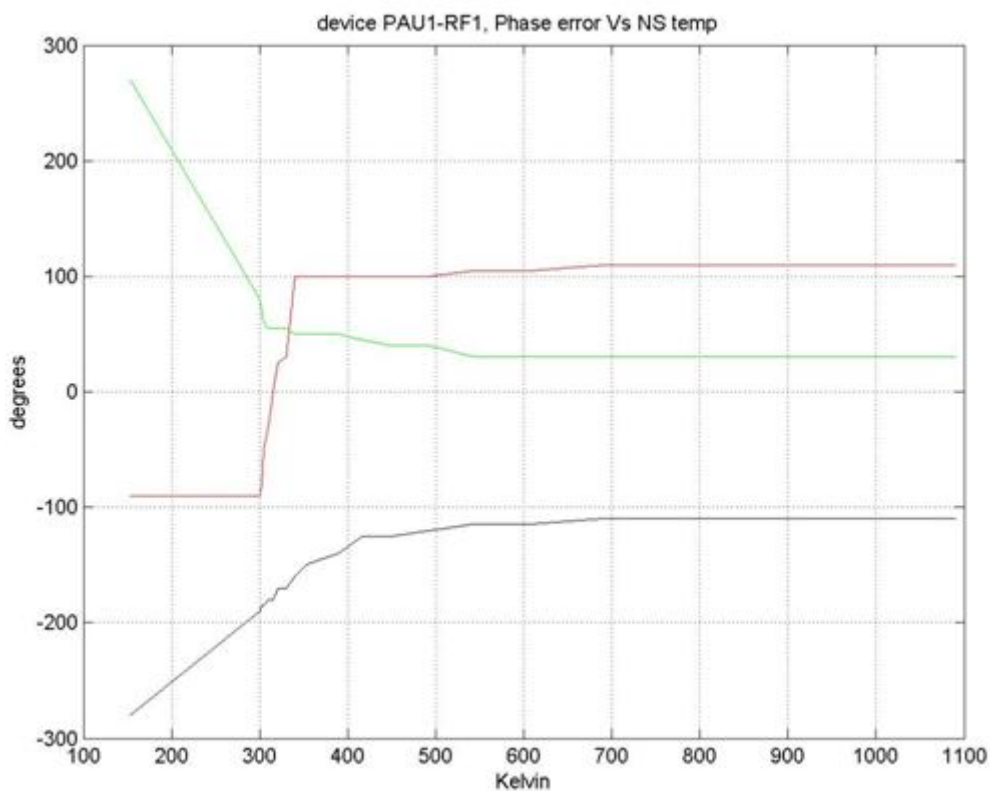


Figura 5-31: Diferencias de fase relativas al canal de referencia estimadas en función de la temperatura de ruido equivalente inyectada a través de la entrada de calibración de PAU-OR. La traza verde corresponde a la diferencia de fase del canal 2, la traza roja al canal 3 y la negra al canal 4.

De nuevo, para valores de temperatura equivalente de ruido inferiores a aproximadamente 300 Kelvin, las diferencias de fase estimadas presentan un salto de 180° .

Este efecto es debido a que la salida del radiómetro es proporcional a la diferencia entre la temperatura equivalente de ruido de la entrada (ya sea la entrada de calibración o de antena) y la temperatura equivalente de ruido generada por el divisor de Wilkinson, es decir:

$$S \propto (T_{IN} - T_W) e^{j\phi} \quad (5.41)$$

Puesto que la temperatura equivalente de ruido generada por el divisor de Wilkinson es de aproximadamente 300 Kelvin, si se inyecta en la entrada de calibración una temperatura equivalente de ruido menor de 300 K, el signo negativo hace que la diferencia de fase medida sea la real menos 180°. En cambio para temperaturas de entrada superiores a 300 Kelvin la diferencia de fase medida es la que realmente se quiere estimar, es decir que:

$$T_{IN} \ll T_W \rightarrow S \propto a e^{j(\phi - \pi)} \rightarrow \phi - 180^\circ \quad (5.42)$$

$$T_{IN} \gg T_W \rightarrow S \propto a e^{j\phi} \rightarrow \phi$$

Por tanto, para valores cercanos a 300 Kelvin no es posible llevar a cabo medidas de fase, sino que es necesario un cierto nivel de potencia a la entrada (a partir de 700 K) para garantizar que la respuesta de fase obtenida será la correcta.

Este efecto explicaría los resultados obtenidos para los algoritmos de calibración diferencial, ya que el hardware de calibración diferencial utiliza como temperaturas de referencia 300, 400 y 500 K, todas ellas dentro de la zona de transición de la fase. Por tanto las fases obtenidas serían distintas para cada nivel de potencia y en consecuencia no se estaría llevando a cabo la calibración correctamente.

Así pues, de la misma forma que es necesario determinar los coeficientes de desfase entre entradas para cada receptor al inicio de una campaña de medidas, también es necesario determinar para cada receptor y para cada canal, la temperatura de ruido equivalente mínima que garantiza que la fase estimada es correcta.

Si se quiere utilizar una calibración diferencial de fase, es necesario modificar los valores de atenuación del hardware de calibración diferencial para garantizar que la potencia que se inyectará a la entrada de calibración resulte en una temperatura equivalente de ruido suficientemente elevada. Esto mismo también es aplicable para el caso de los algoritmos no diferenciales.

5.4 CALIBRACIÓN DE GANANCIA

5.4.1 Calibración de ganancia mediante algoritmo no diferencial

El proceso de calibración de ganancia consiste en el cálculo del coeficiente de corrección de ganancia que es necesario aplicar a cada canal para igualar su amplitud a la amplitud del canal de referencia.

Para determinar estos coeficientes de corrección de ganancia se parte de las correlaciones entre el canal de referencia y el resto de canales, correlaciones que ya se obtuvieron para el caso de la calibración de fase (ver ecuaciones 5.16 a 5.19):

$$\begin{aligned}\langle S_1 S_2^* \rangle &= |A_{12}| e^{j\theta_{12}} = |A_1| |A_2| e^{j(\theta_1 - \theta_2)} \\ \langle S_1 S_3^* \rangle &= |A_{13}| e^{j\theta_{13}} = |A_1| |A_3| e^{j(\theta_1 - \theta_3)} \\ \langle S_1 S_4^* \rangle &= |A_{14}| e^{j\theta_{14}} = |A_1| |A_4| e^{j(\theta_1 - \theta_4)}\end{aligned}\tag{5.43}$$

Es posible obtener los coeficientes de corrección de ganancia deseados a partir de los módulos de las correlaciones de la ecuación 5.43, de tal forma que:

$$\begin{aligned}C_2 &= \frac{|A_{11}|}{|A_{12}|} = \frac{|A_1|^2}{|A_1| |A_2|} = \frac{|A_1|}{|A_2|} \\ C_3 &= \frac{|A_{11}|}{|A_{13}|} = \frac{|A_1|^2}{|A_1| |A_3|} = \frac{|A_1|}{|A_3|} \\ C_4 &= \frac{|A_{11}|}{|A_{14}|} = \frac{|A_1|^2}{|A_1| |A_4|} = \frac{|A_1|}{|A_4|}\end{aligned}\tag{5.44}$$

Una vez obtenidos los coeficientes, es posible igualar la amplitud de cada uno de los canales a la amplitud del canal de referencia si se multiplica la señal a la salida de cada canal por su respectivo coeficiente, es decir:

$$S_2' = C_2 S_2 = \frac{|A_1|}{|A_2|} |A_2| e^{j\theta_2} = |A_1| e^{j\theta_2}$$

$$S_3' = C_3 S_3 = \frac{|A_1|}{|A_3|} |A_3| e^{j\theta_3} = |A_1| e^{j\theta_3} \quad (5.45)$$

$$S_4' = C_4 S_4 = \frac{|A_1|}{|A_4|} |A_4| e^{j\theta_4} = |A_1| e^{j\theta_4}$$

Donde S_i' (con $i = 2, 3$ y 4) representa las señales a la salida de cada uno de los canales, una vez aplicados los coeficientes de corrección de ganancia.

Como se puede observar en la ecuación 5.45, las amplitudes de los tres canales corregidos serán iguales entre sí e iguales a la amplitud del canal de referencia. Se puede afirmar por tanto que los cuatro canales están calibrados en ganancia.

5.4.2 Implementación en la Unidad de Control

La implementación del algoritmo de calibración de ganancia descrito en el apartado anterior se lleva a cabo a partir de las correlaciones que la ALU del radiómetro envía a la unidad de control. El proceso que lleva a cabo la calibración de ganancia se compone de los siguientes pasos:

- En primer lugar se capturan las correlaciones calculadas por la ALU:

<i>Pseudocode</i>
<pre>while calibration data available: mddc[receiver].II[channel] = acquire data mddc[receiver].IQ[channel] = acquire data</pre>

- A partir de las correlaciones obtenidas en el primer paso, se calcula el coeficiente de corrección de ganancia para cada canal. Notar que se aplica el algoritmo en su versión de respuesta infinita, por tanto se conserva el valor del coeficiente para emplearse en la siguiente iteración del algoritmo:

Pseudocode

```

For each channel:
  numerator[channel] = mddc[receiver].II[0]^2 + mddc[receiver].IQ[0]^2
  denominator[channel] = mddc[receiver].II[channel]^2 + mddc[receiver].IQ[channel]^2
  PCE.alpha[channel] = sqrt(numerator[channel]/denominator[channel])
  mddc[receiver].A[channel] = mddc[receiver].Aant[channel]*PCE.alpha[channel];
  mddc[receiver].Aant[channel] = mddc[receiver].A[channel];
    
```

- o Finalmente, se aplica el factor de corrección a las expresiones contenidas en las tablas trigonométricas, de manera que la siguiente captura de datos ya estará corregida en ganancia. Notar que la expresión empleada incluye únicamente corrección de ganancia y no de fase.

Pseudocode

```

For each position of the trigonometric table
  k = dephase coefficient corresponding to the current table position
  tt=(128*mddc[receiver].A[channel])*cos(k*(pi/2));
  send_to_ALU (tt)
    
```

5.4.3 Resultados y conclusiones

Los resultados obtenidos para la implementación descrita del algoritmo de calibración de ganancia se muestran en la figura 5-32:

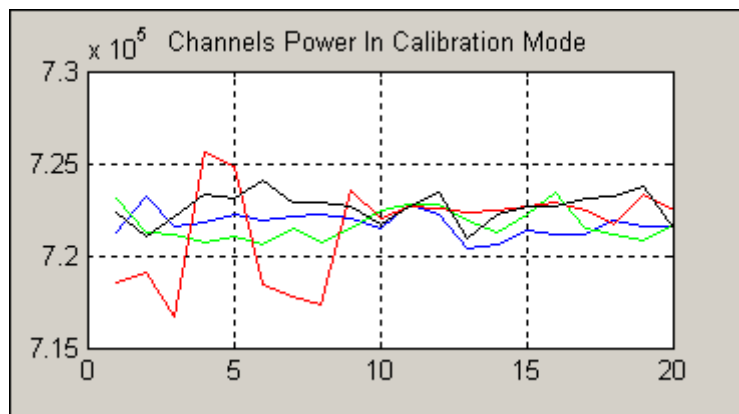


Figura 5-32: Potencias obtenidas durante el proceso de calibración tras aplicar un algoritmo de corrección de ganancia no diferencial.

Como se puede observar los canales 2 (traza verde), 3 (traza roja) y 4 (traza negra) presentan unas potencias de calibración iguales a la potencia de calibración del canal de referencia (traza azul). Por tanto, se puede afirmar que se han igualado los

coeficientes de ganancia de los cuatro canales del receptor, y que por tanto este se encuentra calibrado en ganancia.

5.4.4 Calibración de ganancia mediante algoritmo diferencial

Puesto que ya se disponía del hardware de calibración diferencial implementado para la calibración de fase, y los mecanismos de adquisición de los datos procedentes de la ALU del radiómetro ya habían sido adaptados para trabajar con dos niveles de potencia, se implementó una versión diferencial del hardware de calibración de ganancia.

Las señales a la salida de cada uno de los cuatro canales puede expresarse como:

$$\begin{aligned}
 S_1 &= \frac{1}{\sqrt{2}} \left(\frac{1}{\sqrt{2}} (S_{CNS} + S_{W1}) + S_{W2} \right) + S_{REC1} \\
 S_2 &= \frac{\sqrt{\alpha_2}}{\sqrt{2}} \left(\frac{1}{\sqrt{2}} (S_{CNS} + S_{W1}) - S_{W2} \right) + S_{REC2} \\
 S_3 &= \frac{\sqrt{\alpha_3}}{\sqrt{2}} \left(\frac{1}{\sqrt{2}} (S_{CNS} - S_{W1}) + S_{W3} \right) + S_{REC3} \\
 S_4 &= \frac{\sqrt{\alpha_4}}{\sqrt{2}} \left(\frac{1}{\sqrt{2}} (S_{CNS} - S_{W1}) - S_{W3} \right) + S_{REC4}
 \end{aligned} \tag{5.46}$$

Donde: S_{CNS} es el ruido generado por la fuente de ruido conmutable, S_{W1} , S_{W2} y S_{W3} representan el ruido generado por los divisores de Wilkinson 1, 2 y 3 (ver figura 5-1), S_{RECI} constituye el ruido introducido por cada una de las cuatro cadenas receptoras y α_i son los coeficientes de amplitud para cada canal que se pretende estimar y corregir.

Calculando las correlaciones entre la señal del canal de referencia y las señales del resto de canales conjugadas, se obtiene:

$$\langle S_1 S_1^* \rangle = \frac{T_{CNS} + T_{W1}}{4} + \frac{T_{W2}}{2} + T_{REC1} \tag{5.47}$$

$$\langle S_1 S_2^* \rangle = \alpha_2 \left(\frac{T_{CNS} + T_{W1}}{4} - \frac{T_{W2}}{2} \right)$$

$$\langle S_1 S_3^* \rangle = \alpha_3 \left(\frac{T_{CNS} + T_{W1}}{4} \right)$$

$$\langle S_1 S_4^* \rangle = \alpha_4 \left(\frac{T_{CNS} + T_{W1}}{4} \right)$$

Donde: T_{CNS} es la temperatura de ruido equivalente de la fuente de ruido (notar que: $T_{CNS} = \langle S_{CNS} \cdot S_{CNS}^* \rangle$), T_{W1} y T_{W2} constituyen las temperaturas de ruido de los divisores de Wilkinson 1 y 2, y T_{REC1} es la temperatura de ruido de la cadena receptora del canal de referencia.

Si se obtienen estas correlaciones para dos niveles de temperatura de referencia diferentes y se calculan las diferencias, se tiene que:

$$\langle S_1 S_1^* \rangle|_{T_{MAX}} - \langle S_1 S_1^* \rangle|_{T_{MIN}} = 1 \cdot \left(\frac{T_{MAX} + T_{MIN}}{4} \right)$$

$$\langle S_1 S_2^* \rangle|_{T_{MAX}} - \langle S_1 S_2^* \rangle|_{T_{MIN}} = \alpha_2 \cdot \left(\frac{T_{MAX} + T_{MIN}}{4} \right)$$

(5.48)

$$\langle S_1 S_3^* \rangle|_{T_{MAX}} - \langle S_1 S_3^* \rangle|_{T_{MIN}} = \alpha_3 \cdot \left(\frac{T_{MAX} + T_{MIN}}{4} \right)$$

$$\langle S_1 S_4^* \rangle|_{T_{MAX}} - \langle S_1 S_4^* \rangle|_{T_{MIN}} = \alpha_4 \cdot \left(\frac{T_{MAX} + T_{MIN}}{4} \right)$$

Por tanto, a partir de estas diferencias es posible obtener los coeficientes de corrección de ganancia como:

$$\alpha_2 = \frac{\langle S_1 S_2^* \rangle|_{T_{MAX}} - \langle S_1 S_2^* \rangle|_{T_{MIN}}}{\langle S_1 S_1^* \rangle|_{T_{MAX}} - \langle S_1 S_1^* \rangle|_{T_{MIN}}}$$

$$\alpha_3 = \frac{\langle S_1 S_3^* \rangle|_{T_{MAX}} - \langle S_1 S_3^* \rangle|_{T_{MIN}}}{\langle S_1 S_1^* \rangle|_{T_{MAX}} - \langle S_1 S_1^* \rangle|_{T_{MIN}}} \quad (5.49)$$

$$\alpha_4 = \frac{\langle S_1 S_4^* \rangle|_{T_{MAX}} - \langle S_1 S_4^* \rangle|_{T_{MIN}}}{\langle S_1 S_1^* \rangle|_{T_{MAX}} - \langle S_1 S_1^* \rangle|_{T_{MIN}}}$$

Así pues, para corregir la ganancia de los cuatro canales, igualándolos a la ganancia del canal de referencia, basta con multiplicar por la inversa de los coeficientes obtenidos en la ecuación 5.49.

5.4.5 Implementación en la Unidad de Control

Se implementaron dos versiones diferentes del algoritmo: una primera versión en la que las potencias empleadas en el cálculo de los coeficientes corresponden directamente a las entregadas por la ALU del radiómetro y una segunda versión en la que las potencias se obtienen a partir de las correlaciones de fase y cuadratura que proporciona la ALU.

5.4.5.1 Primera implementación

Esta implementación se basa en la utilización de las medidas de potencia que proporciona la ALU del radiómetro para estimar los coeficientes de corrección de ganancia. Para este algoritmo, la secuencia de acciones a ejecutar es la siguiente:

- En primer lugar se selecciona la temperatura de referencia, de forma que durante la primera mitad de las iteraciones el hardware de calibración inyecte a la entrada de calibración el nivel de potencia correspondiente a T_{max} y durante la segunda mitad inyecte el nivel de potencia correspondiente a T_{min} :

Pseudocode

```

Every new calibration cycle:
  if ( numero_ciclos_cal <= iteraciones/2 )
  {
    select Tmax reference temperature
  }

  else if ( numero_ciclos_cal > iteraciones/2 )
  {
    select Tmin reference temperature
  }
    
```

- Se obtienen las potencias calculadas por a ALU para cada nivel de temperatura de referencia y se promedia por el número de iteraciones efectuadas:

Pseudocode

```

For each channel:
  if ( (numero_ciclos_cal) <= (iteraciones/2) )
  {
    PCE.tmaxpower[channel] = acquire_data_from_ALU();
    PCE.tmaxpowermean[channel] = PCE.tmaxpowermean[channel] + PCE.tmaxpower[channel];
  }
  else if ( (numero_ciclos_cal) > (iteraciones/2) )
  {
    PCE.tminpower[channel] = acquire_data_from_ALU();
    PCE.tminpowermean[channel] = PCE.tminpowermean[channel] + PCE.tminpower[channel];
  }
  PCE.tmaxpowermean[channel] = PCE.tmaxpowermean[channel]/iteraciones/2;
  PCE.tminpowermean[channel] = PCE.tminpowermean[channel]/iteraciones/2;
    
```

- Se calcula el coeficiente de corrección de ganancia para cada uno de los canales (mddc[receiver].A[channel]). Puesto que se aplica un algoritmo de respuesta infinita que tiene memoria, se almacena el coeficiente obtenido para emplearlo en el cálculo de la siguiente iteración:

Pseudocode

```

For each channel:
  numerator[channel] = PCE.tmaxpower[0]-PCE.tminpower[0];
  denominator[channel] = PCE.tmaxpower[channel]-PCE.tminpower[channel];
  PCE.alpha[channel] = numerator[channel]/denominator[channel];
  mddc[receiver].A[channel] = mddc[receiver].Aant[channel]*PCE.alpha[channel];
  mddc[receiver].Aant[channel] = mddc[receiver].A[channel];
    
```

- Finalmente se corrigen los valores de las tablas trigonométricas y se envían las nuevas tablas corregidas a la ALU:

Pseudocode

```

For each position of the trigonometric table
  k = dephase coefficient corresponding to the current table position
  tt=(128*mddc[receiver].A[channel])*cos(k*(pi/2));
  send_to_ALU (tt)

```

5.4.5.2 Segunda implementación

Esta implementación es una variante de la anterior, idéntica en todo el proceso excepto en el tercer paso, en el que se utilizan las correlaciones para obtener las potencias que se emplearán en el cálculo de los coeficientes de corrección de ganancia, es decir:

$$P_i = |R_i(0)| = \sqrt{\langle I_i I_i \rangle^2 + \langle I_i Q_i \rangle^2} \quad (5.50)$$

Donde: el subíndice i denota el canal, P es la potencia, $|R_i(0)|$ es la correlación en el origen e I y Q son las componentes en fase y cuadratura respectivamente de la señal del canal.

Teóricamente, la potencia entregada por la ALU y la potencia estimada a partir de las correlaciones según la ecuación 5.48, han de coincidir. No obstante, puesto que las correlaciones son imprescindibles para llevar a cabo la calibración de fase, este método alternativo podría servir para prescindir de las potencias y emplear únicamente correlaciones, aligerando así la carga que soporta todo el sistema (tanto la ALU como la UC). En esta implementación, el tercer punto del algoritmo descrito para la primera implementación (en el que se calculan los coeficientes) debería sustituirse por el siguiente:

- Se calcula el coeficiente de corrección de ganancia para cada uno de los canales ($mddc[receiver].A[channel]$). Para ello se emplean las correlaciones promediadas que se obtuvieron en el algoritmo de calibración de fase ($tmaxiimean$, $tmaxiqmean$, $tminiimean$, $tminiqlmean$):

Pseudocode

For each channel:

```

PCE.tmaxpower[channel] = sqrt(PCE.tmaxiimean[channel]^2 + PCE.tmaxiqmean[channel]^2);
PCE.tminpower[channel] = sqrt(PCE.tminiimean[channel]^2 + PCE.tminiqlmean[channel]^2);
numerator[channel] = PCE.tmaxpower[0]-PCE.tminpower[0];
denominator[channel] = PCE.tmaxpower[channel]-PCE.tminpower[channel];
PCE.alpha[channel] = numerator[channel]/denominator[channel];
mddc[receiver].A[channel] = mddc[receiver].Aant[channel]*PCE.alpha[channel];
mddc[receiver].Aant[channel] = mddc[receiver].A[channel];

```

A excepción de este punto, el resto de pasos descritos para la primera implementación permanecen iguales en esta implementación.

5.4.6 Resultados y conclusiones

La comparativa de los resultados obtenidos para ambas implementaciones del algoritmo de calibración diferencial puso de manifiesto un problema del sistema.

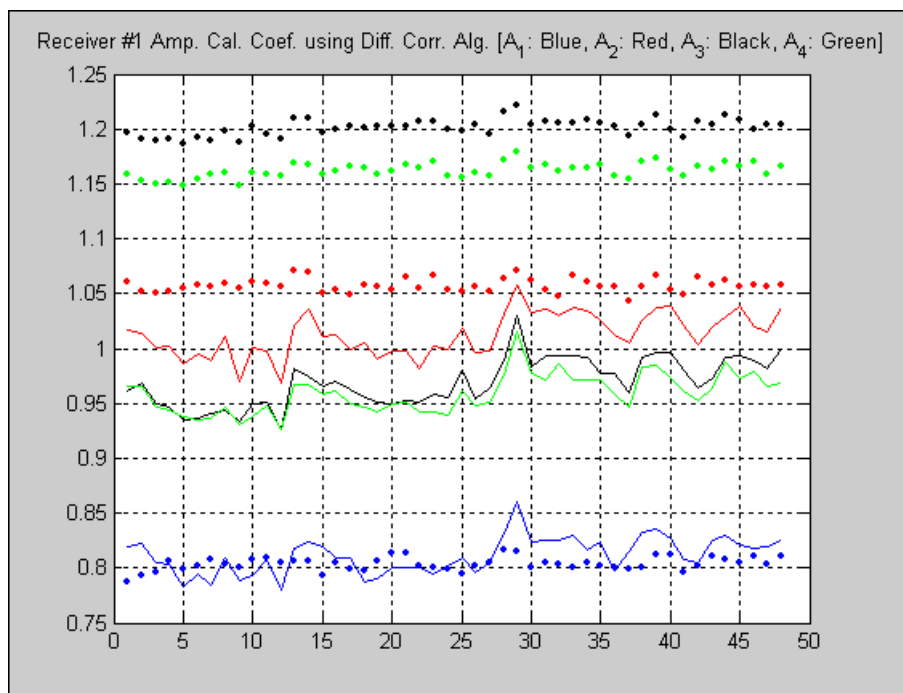


Figura 5-33: Coeficientes de corrección de ganancia obtenidos empleando la primera (líneas sólidas) y segunda (líneas de puntos) implementación del algoritmo de calibración de ganancia diferencial.

En la figura 5-33 se muestra una gráfica con los coeficientes de corrección de ganancia obtenidos aplicando al receptor de referencia ambas implementaciones del algoritmo. Las líneas sólidas corresponden a los coeficientes obtenidos mediante la primera implementación (utilizando las potencias calculadas por la ALU); mientras que

las líneas punteadas corresponden a los coeficientes obtenidos mediante la segunda implementación (empleando las correlaciones para calcular las potencias).

Lejos de lo esperado, los coeficientes calculados a partir de los dos métodos no coinciden, a excepción del coeficiente del canal 1, ya que al tratarse del canal de referencia, la correlación cruzada coincide con la auto-correlación. Estos resultados revelan la presencia de niveles incorrectos de correlación en las señales a la salida de los cuatro canales, es decir que se detecta cierto nivel de correlación en circunstancias en las que la correlación debería ser cero.

5.4.7 Pruebas de validación de los algoritmos

Para verificar el correcto funcionamiento de los algoritmos de calibración de ganancia, de nuevo se recurrió al montaje empleado en la validación de los algoritmos de calibración de fase. Se sustituyó el receptor de referencia por la placa de adaptación y se inyectó señal sintética a la frecuencia de trabajo de PAU-RAD generada mediante un generador de funciones.

La figura 5-34 muestra los resultados obtenidos para una calibración de ganancia no diferencial:

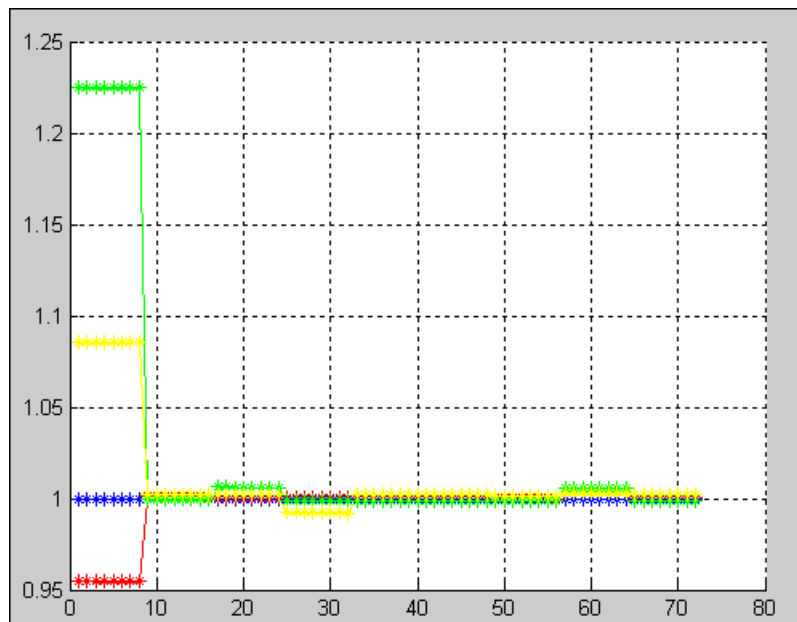


Figura 5-34: Coeficientes de corrección de ganancia obtenidos empleando la primera (asteriscos) y segunda (líneas) implementación del algoritmo de calibración de ganancia diferencial en ausencia del receptor.

Las trazas formadas por asteriscos corresponden a los coeficientes de corrección de ganancia calculados directamente a partir de la potencia obtenida del

radiómetro. Por otra parte, las trazas sólidas corresponden a los mismos coeficientes pero determinados empleando correlaciones.

Como se puede observar, los coeficientes determinados empleando potencias coinciden con los calculados a partir de correlaciones y todos los coeficientes tienden a 1 tras la primera iteración. Por tanto se puede afirmar que los cuatro canales están calibrados en ganancia y que el algoritmo funciona tanto para potencias como para correlaciones.

5.4.8 Pruebas de validación del hardware

Una vez comprobado el correcto funcionamiento de los algoritmos en ausencia del receptor, para determinar la causa por la cual la potencia calculada a partir de la correlación no coincide con la potencia obtenida directamente del radiómetro, se realizaron diversas pruebas con el receptor de referencia para determinar el origen de los niveles incorrectos de correlación.

5.4.8.1 Medida de la respuesta frecuencial del receptor

En primer lugar se analizó la respuesta frecuencial del receptor de referencia. La figura 5-35 muestra la respuesta frecuencial a la salida del amplificador de video de la etapa de FI (ver figura 5-1) con el potenciómetro de control de ganancia ajustado para proporcionar ganancia máxima:

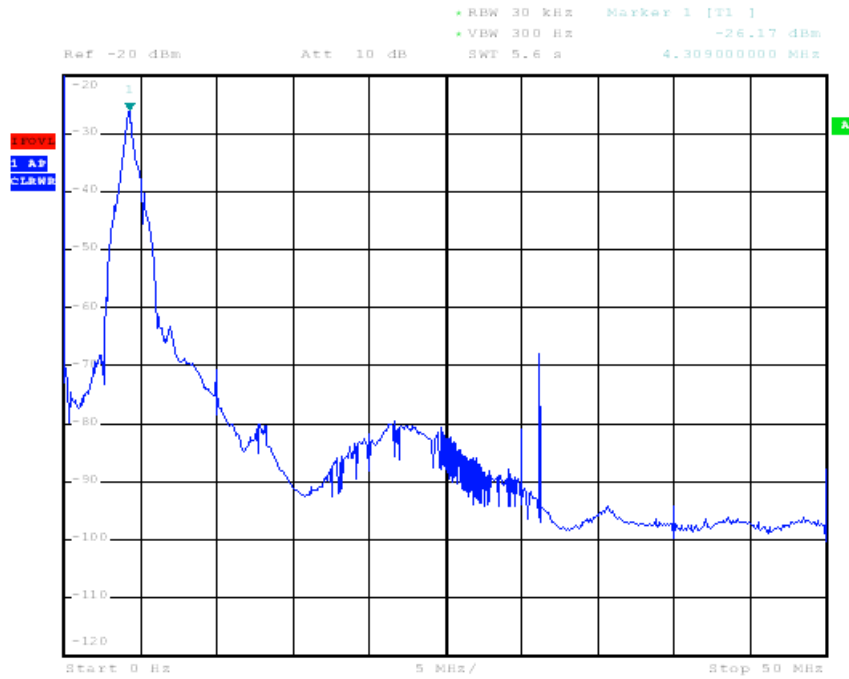


Figura 5-35: Espectro a la salida del receptor de referencia de PAU.

Como se puede observar, la señal convertida a frecuencia intermedia (4,309 MHz), presenta un pico de ganancia que distorsiona toda la banda.

Después de múltiples pruebas, se comprobó que el hecho de actuar sobre los potenciómetros del control de ganancia de la etapa de FI, no sólo modificaba la ganancia del amplificador, sino que también modificaba la banda de la señal. Esto es debido a que el amplificador de video además de amplificar, ejerce funciones de acondicionamiento de señal para el ADC. La etapa de entrada del amplificador de video NE592 ha sido diseñada de manera que, añadiendo elementos reactivos entre los terminales de control de ganancia, el circuito pueda funcionar como un filtro. Este hecho fue aprovechado para implementar un filtro paso banda centrado en la frecuencia intermedia y reducir el nivel de ruido del sistema.

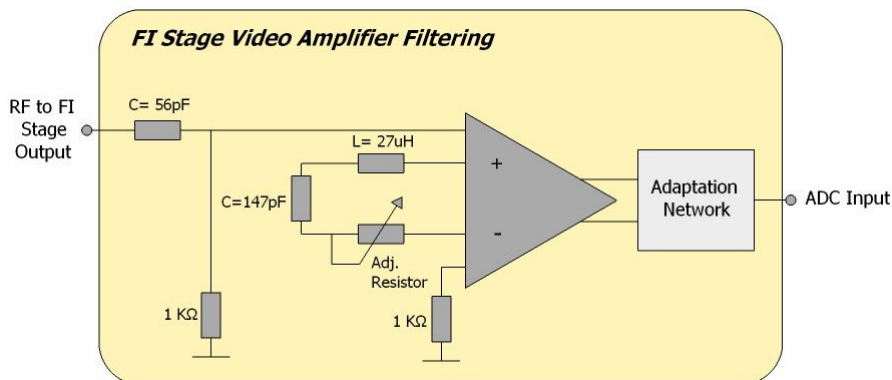


Figura 5-36: Implementación del filtro paso banda en el lazo de control de ganancia del amplificador de video.

La figura 5-36 muestra el esquema de la implementación del filtro paso banda integrado en el amplificador de vídeo.

Notar que esta implementación no se corresponde con la mostrada en la figura 5-1, que muestra una implementación anterior que fue desestimada por introducir desadaptación de impedancias con la etapa de conversión RF a FI.

Al actuar sobre el potenciómetro de ajuste de ganancia del amplificador, además de modificar la ganancia se modifica tanto el ancho de banda como la frecuencia central del filtro pasa banda, creando la forma distorsionada del espectro que se puede observar en la figura 5-35. Inicialmente se señaló esta distorsión como causante de los niveles incorrectos de correlación, y el esquema de la figura 5-37 se modificó para solucionar este problema, dando como resultado el esquema que se muestra en la siguiente figura:

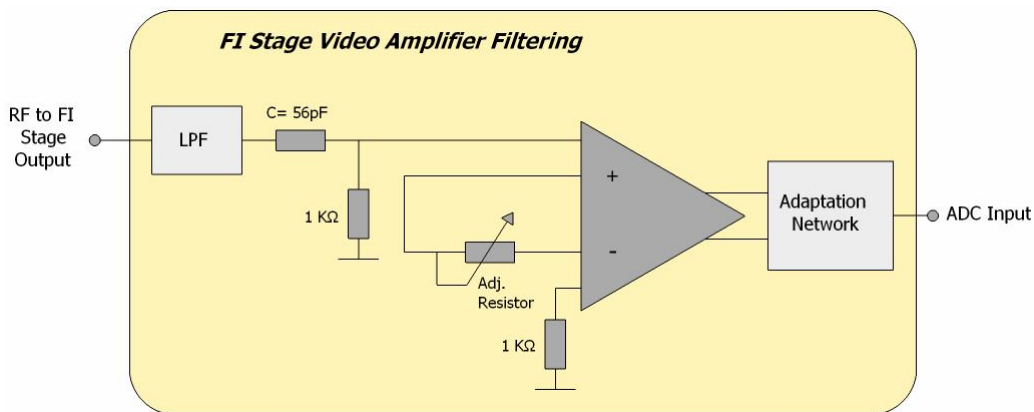


Figura 5-37: Implementación del filtro paso banda fuera del lazo de control de ganancia del amplificador de video.

En la nueva configuración, el filtrado paso banda se lleva a cabo directamente a la entrada de la etapa de amplificación mediante un filtro paso bajo comercial y un filtro paso alto implementado mediante una resistencia y un condensador. De esta forma se elimina la opción de filtrado del amplificador, dejando en el lazo de control de ganancia únicamente el potenciómetro encargado de controlar la amplificación del canal.

El espectro de la señal a la salida del amplificador de video una vez eliminada la opción de filtrado del lazo de control de ganancia y seleccionando amplificación mínima, se muestra en la figura 5-38:

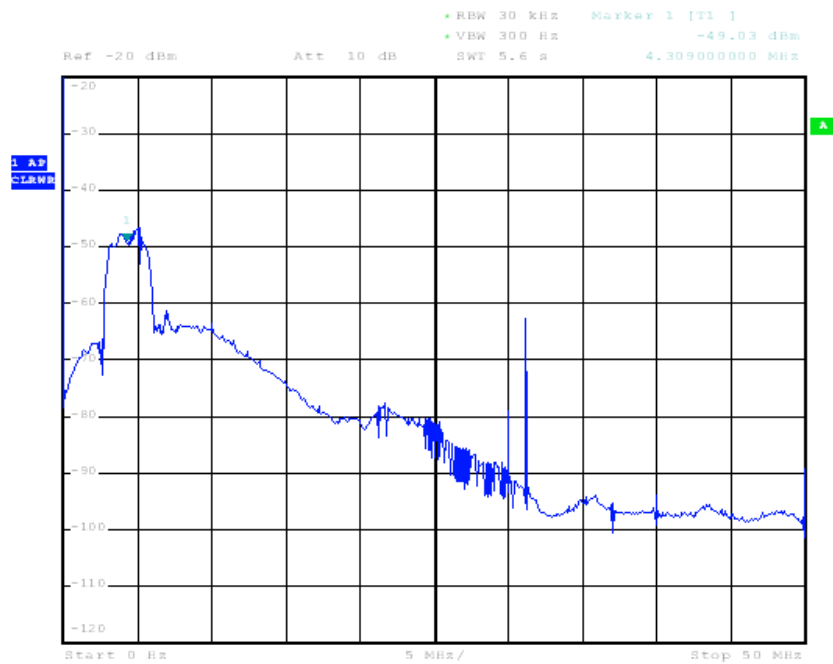


Figura 5-38: Espectro a la salida del receptor de referencia de PAU sin filtrado integrado en el lazo de control de ganancia, con ganancia mínima seleccionada.

Como se puede observar, la distorsión de la banda de paso que se apreciaba en la figura 5-35 ha desaparecido, dando lugar a una respuesta frecuencial más coherente. Si se ajusta el potenciómetro para proporcionar ganancia máxima a la salida del canal, el espectro resultante es el que se muestra en la figura 5-39:

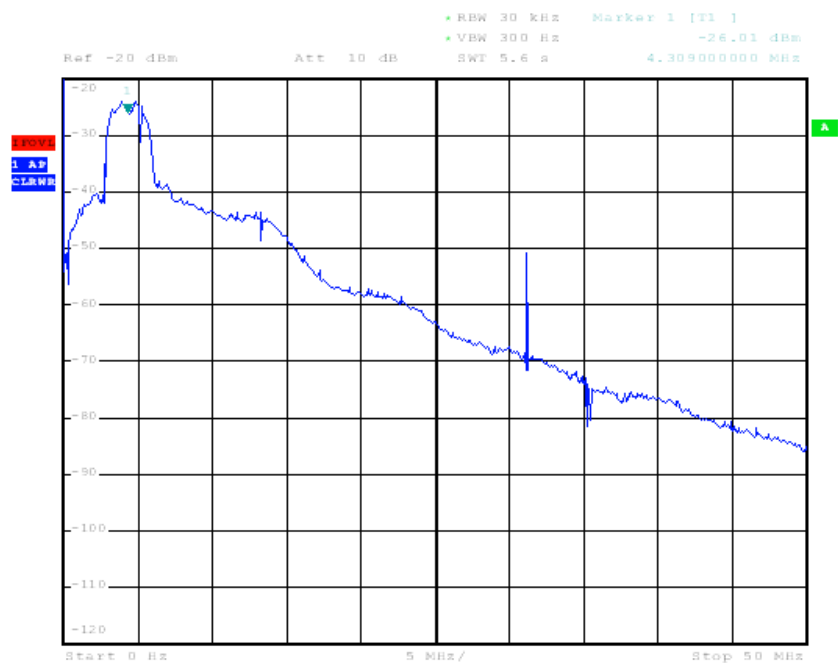


Figura 5-39: Espectro a la salida del receptor de referencia de PAU sin filtrado integrado en el lazo de control de ganancia, con ganancia máxima seleccionada

Se puede observar como el nivel de señal aumenta, manteniendo constante la forma de la banda de frecuencia de FI, sin provocar la distorsión que se apreciaba en la figura 5-35.

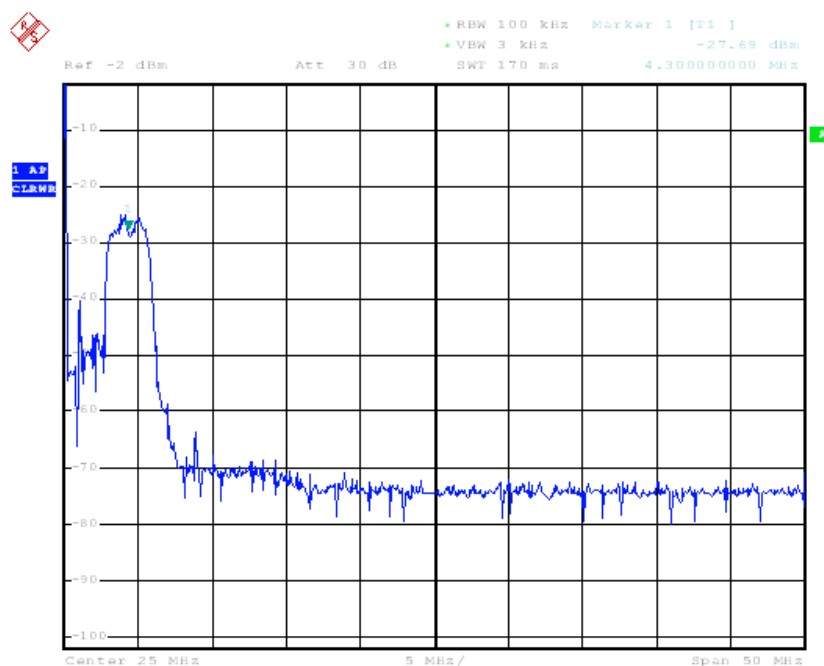


Figura 5-40: Espectro a la salida del receptor de referencia de PAU con la implementación de filtrado y control de ganancia de la figura 5-37.

Añadiendo el filtro paso bajo comercial a la entrada, tal y como se ilustra en la figura 5-37, el espectro de salida es el que se muestra en la figura 5-40. Como se puede observar se reduce drásticamente el nivel de ruido a la salida y mejora la calidad de la señal.

5.4.8.2 Medida de la respuesta en potencia del receptor

Una vez resueltos los problemas de distorsión de la señal debidos al filtrado del amplificador de vídeo, se comprobó que la diferencias existentes entre la potencia medida directamente por el radiómetro y la potencia calculada a partir de la correlación persistían.

Para determinar la procedencia de esta anomalía, se llevó a cabo una medida de la respuesta del receptor de referencia en función de la potencia inyectada a través de la entrada de calibración. En la ejecución de esta prueba se sustituyeron los ADC y la FPGA del sistema PAU por una digitalizadora de la marca National Instruments modelo PXI-5124. Mediante un atenuador variable, se realizó un barrido de potencia

de calibración. Las señales a la salida del receptor se digitalizaron y los datos obtenidos fueron procesados con Matlab para proporcionar las figuras que se presentan a continuación.

La siguiente figura muestra la respuesta del receptor de referencia en función de la potencia de entrada:

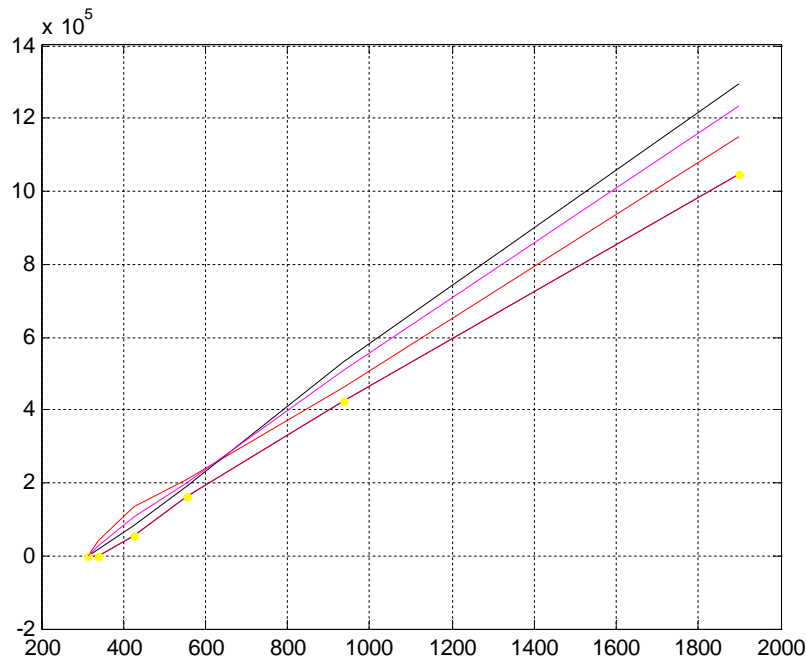


Figura 5-41: Respuesta en potencia del receptor de referencia de PAU.

Los datos que generan la figura 5-41 corresponden a una única polarización, es decir a dos canales adyacentes del receptor. La traza de color negro corresponde a la potencia de salida del canal 1 mientras que la traza de color rojo corresponde a la potencia de salida del canal 2. Por otra parte, la traza de color magenta corresponde a la media geométrica de las trazas negra y roja (es decir la media geométrica de la potencia de los dos canales), mientras que la traza roja con asteriscos amarillos corresponde a la correlación entre la señal de salida del canal 1 y la señal de salida del canal 2.

La señal a la salida del canal 1 puede expresarse como:

$$P_1 = G_1(T_A + T_{W1} + T_{W2} + T_{REC1}) \quad (5.51)$$

Donde G_1 es la ganancia del canal, T_A es la temperatura equivalente de ruido inyectado en la entrada de correlación, T_{W1} es la temperatura equivalente del divisor Wilkinson 1, T_{W2} es la temperatura equivalente del divisor Wilkinson 2 (ver figura 5-1) y T_{REC1} es la temperatura equivalente de la cadena de recepción del canal 1.

De la misma forma la salida del canal 2 puede expresarse como:

$$P_2 = G_2(T_A + T_{W1} - T_{W2} + T_{REC2}) \quad (5.52)$$

Cada uno de los puntos que forman la traza negra, corresponden a un valor de la siguiente ecuación:

$$\Delta P_1 = P_1|_T - P_1|_{T=300 K} = G_1(T_A - 300 K) \quad (5.53)$$

Asimismo, los puntos de conforman la taza roja se obtienen a partir de la ecuación:

$$\Delta P_2 = P_2|_T - P_2|_{T=300 K} = G_2(T_A - 300 K) \quad (5.54)$$

Por otra parte, la correlación de las señales de ambos canales puede expresarse como:

$$C_{12} = \sqrt{G_1 G_2}(T_A + T_{W1} - T_{W2}) \quad (5.55)$$

Calculando la diferencia entre la correlación obtenida para cada valor de la potencia de entrada, y la correlación obtenida para una potencia de entrada correspondiente a una temperatura equivalente de 300 K, se forma la traza magenta, cuyos puntos son valores de la siguiente ecuación:

$$\Delta C_{12} = \sqrt{G_1 G_2}(T_A - 300 K) \quad (5.56)$$

Por último, la traza de color rojo con asteriscos amarillos corresponde a la media geométrica de las trazas roja y negra, es decir:

$$M = \sqrt{\Delta P_1 \Delta P_2} = \sqrt{G_1 G_2}(T_A - 300 K) \quad (5.57)$$

Por tanto se tiene que ΔC_{12} ha de ser igual a M , es decir que las trazas magenta y roja con asteriscos han de ser iguales. No obstante, como se puede observar en la figura 5-40, no coinciden.

Inicialmente se consideró que la causa de este problema residía en acoplamientos indeseados de señal entre los cuatro canales, debidos a que la alta densidad de componentes del receptor y su reducido tamaño hacen que las cuatro cadenas de RF se hallen muy próximas entre sí. Para verificar esta hipótesis se realizó la misma prueba descrita, pero en esta ocasión se empleo un receptor de PAU2.

PAU2 es un proyecto paralelo a PAU que constituye una prueba de concepto de una arquitectura similar a la de PAU, pero que se rige por otros principios de funcionamiento, mientras que PAU es un radiómetro de apertura real, PAU2 es un radiómetro de apertura sintética. Las similitudes con PAU hacen que ambos proyectos puedan compartir un mismo diseño de receptor, no obstante los receptores de PAU2 utilizan únicamente dos de las cuatro cadenas de cada receptor (las dos más alejadas entre sí). Estas características hacen que los receptores de PAU2 sean ideales para comprobar si efectivamente la proximidad de las cuatro cadenas provoca acoplamiento de señal entre cadenas.

El resultado obtenido para un receptor de PAU2 se muestra en la siguiente figura:

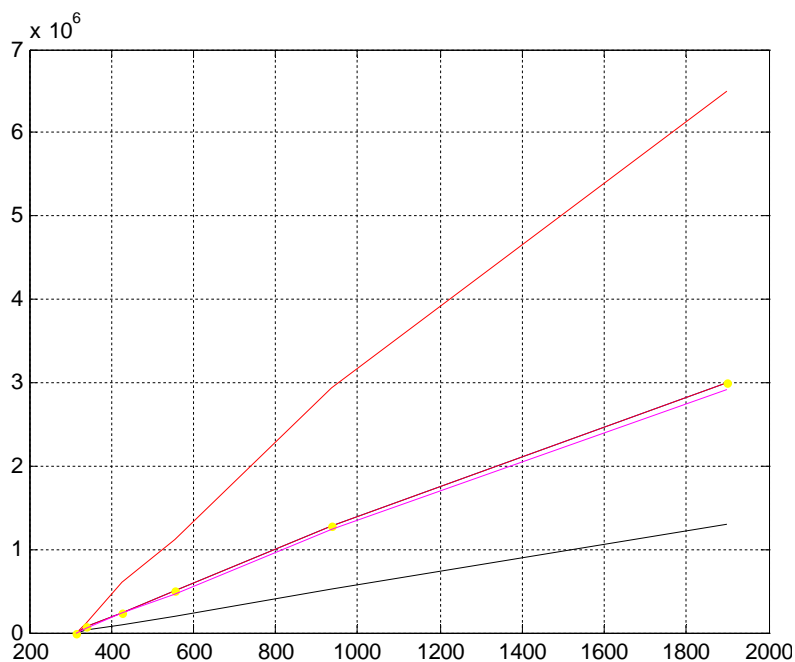


Figura 5-42: Respuesta en potencia de un receptor de PAU2.

Como se puede comprobar, en esta ocasión la coincidencia entre las trazas magenta y roja con asteriscos es mucho mayor que para el caso del receptor de referencia de PAU que se muestra en la figura 5-41.

Para comprobar que efectivamente los resultados obtenidos para el receptor de PAU2 se deben a una mayor separación entre cadenas receptoras, se llevaron a cabo las modificaciones necesarias para que el receptor de referencia de PAU emulase la estructura de un receptor PAU2. Así pues, se anularon dos de las cuatro cadenas de RF y se emplearon únicamente las dos cadenas más alejadas entre sí.

Los resultados de la prueba efectuada para el receptor de PAU modificado son los que se muestran en la figura 5-43:

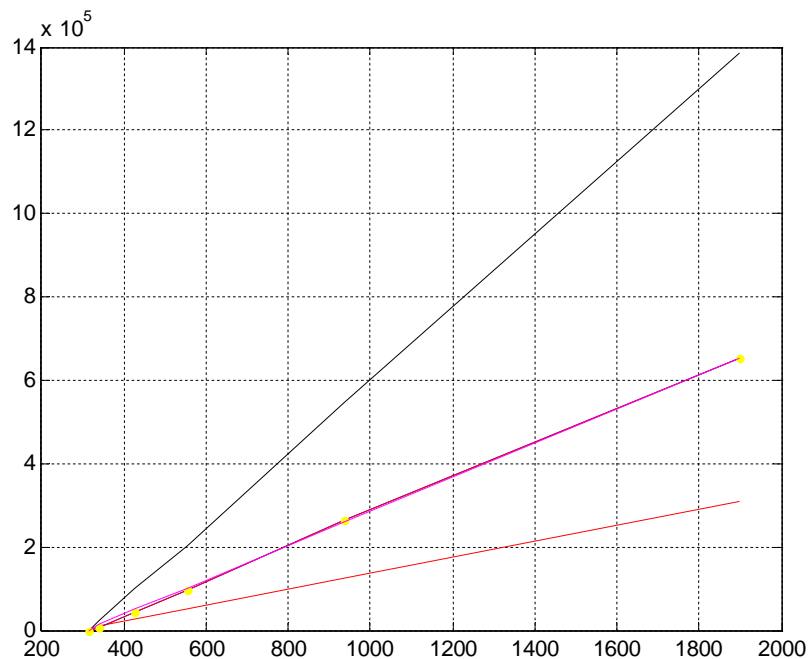


Figura 5-43: Respuesta en potencia del receptor de referencia de PAU, utilizando únicamente dos de las cuatro cadenas receptoras del receptor.

Como se puede comprobar, en esta ocasión las trazas magenta y roja con asteriscos coinciden casi a la perfección. Por tanto, se puede concluir que los niveles incorrectos de correlación causantes de los problemas de calibración de ganancia se deben a acoplamientos indeseados de señal.

Actualmente se está trabajando en una solución que permita corregir estos problemas del hardware de los receptores PAU, adoptando los conocimientos adquiridos durante el desarrollo de PAU2.

5.5 ESCALADO DEL SISTEMA

Uno de los objetivos fijados al inicio del proyecto era el de extender el sistema a su versión final de 16 receptores. No obstante, debido al trabajo desarrollado en otras áreas, al término del proyecto únicamente se ha testeado una versión previa que opera con cuatro receptores.

5.5.1 Escalado a 4 receptores

Para testear el funcionamiento del sistema con cuatro receptores, se empleó el esquema de pruebas que se ilustra en el diagrama de la figura 5-44:

4 Receiver Test Setup

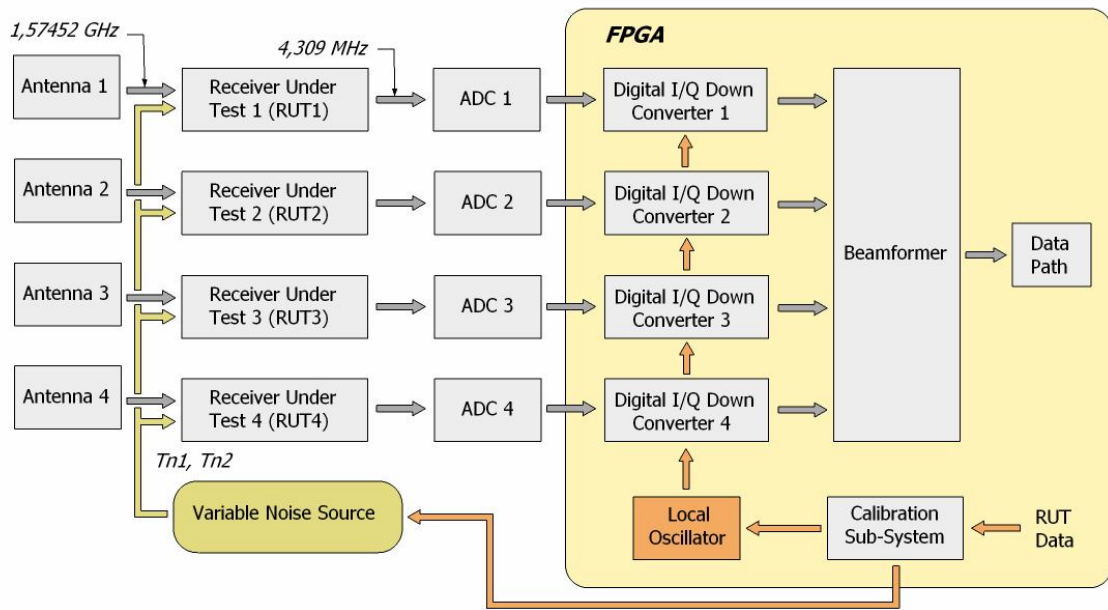


Figura 5-44: Esquema del montaje de pruebas para cuatro receptores.

En la figura 5-45 se muestra una fotografía del montaje efectuado para llevar a cabo la prueba:

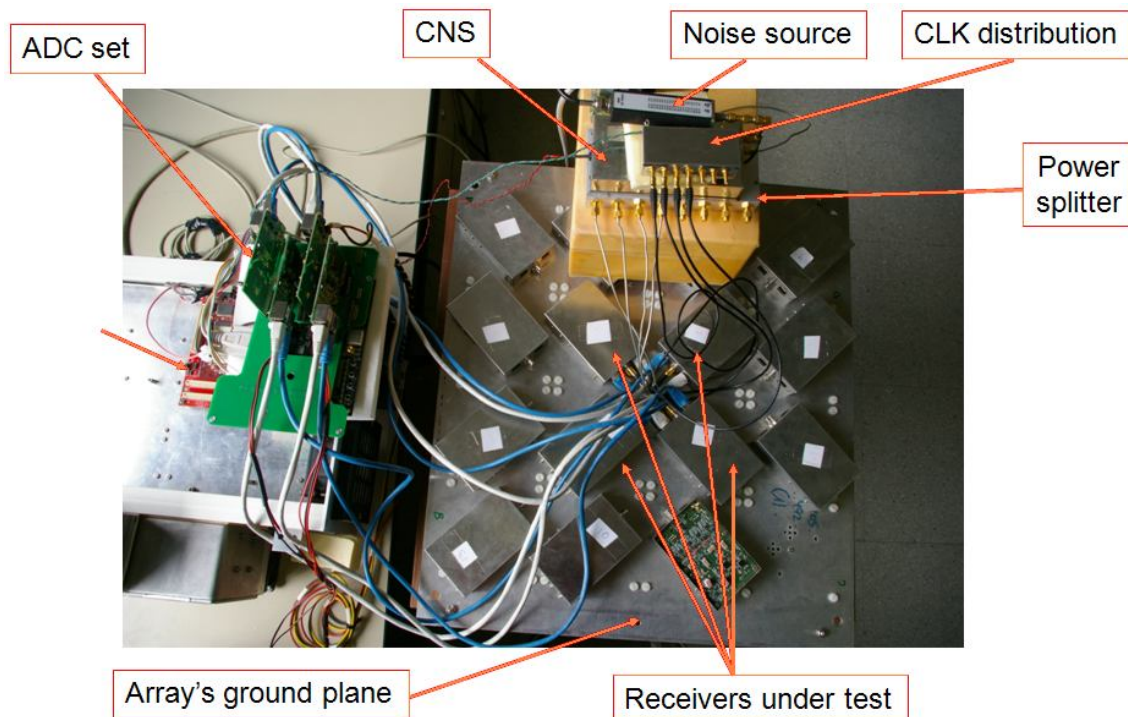


Figura 5-45: Fotografía del montaje empleado para efectuar las pruebas con cuatro receptores.

En esta prueba se llevó a cabo tanto calibración como conformación de haz, de forma que es posible comprobar los efectos de la calibración de los cuatro receptores sobre la conformación de haz.

Para calibrar los cuatro receptores se empleó el esquema de calibración que se muestra en la siguiente figura:

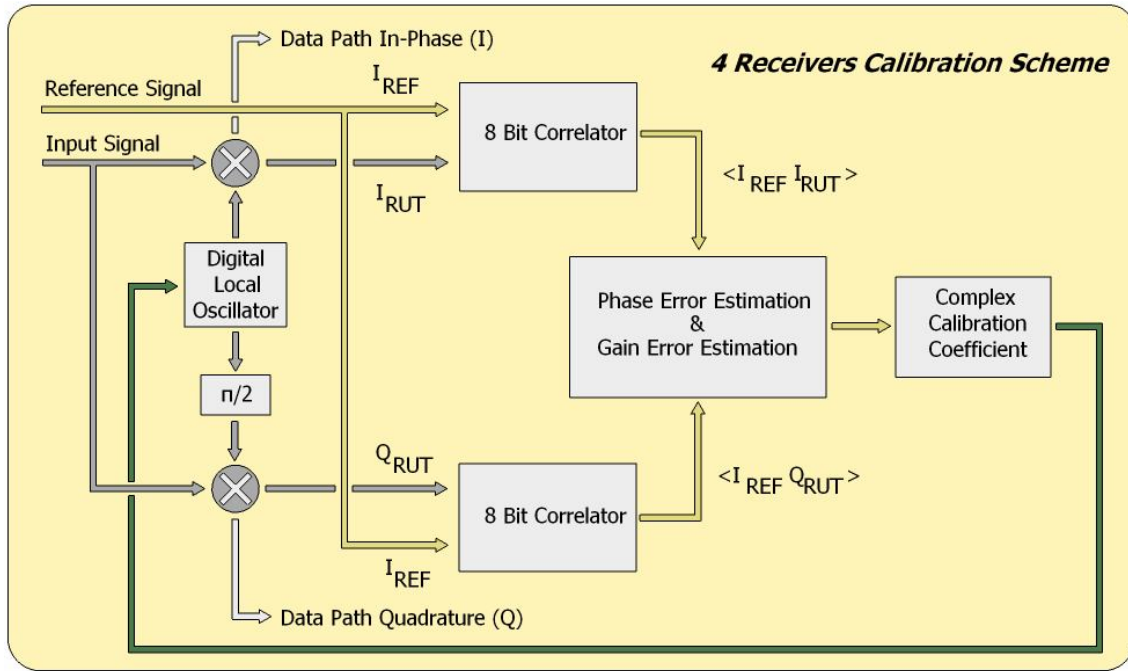


Figura 5-46: Esquema de calibración empleado para el montaje de 4 receptores.

Tanto para la calibración de fase como para la calibración de ganancia se utilizaron implementaciones a 8 bits de los algoritmos de calibración no diferenciales descritos en este mismo capítulo. Como referencia se definió el canal 1 del primer receptor en pruebas o RUT (Receiver Under Test), por lo que, tanto las fases como las ganancias de los cuatro canales de cada receptor se igualarán a la fase y a la ganancia del canal 1 del RUT1.

El coeficiente de calibración complejo obtenido incluye tanto calibración de fase como de ganancia, y se puede expresar como:

$$\alpha = \frac{1}{A} e^{-j\varphi} \quad (5.58)$$

Donde A es el coeficiente de corrección de ganancia, que se obtiene a partir de las correlaciones como:

$$A = \sqrt{\langle I_{REF} I_{RUT} \rangle^2 + \langle I_{REF} Q_{RUT} \rangle^2} \quad (5.59)$$

Mientras que φ representa el coeficiente de corrección de fase, que se calcula a partir de la expresión:

$$\varphi = \tan^{-1} \left(\frac{-\langle I_{REF} Q_{RUT} \rangle}{\langle I_{REF} I_{RUT} \rangle} \right) \quad (5.60)$$

Una vez llevada a cabo la calibración y efectuada la conformación de haz, se obtuvieron las siguientes correlaciones complejas para las polarizaciones horizontal y vertical, que se corresponden con las componentes en fase y cuadratura del primer y segundo parámetro de Stokes:

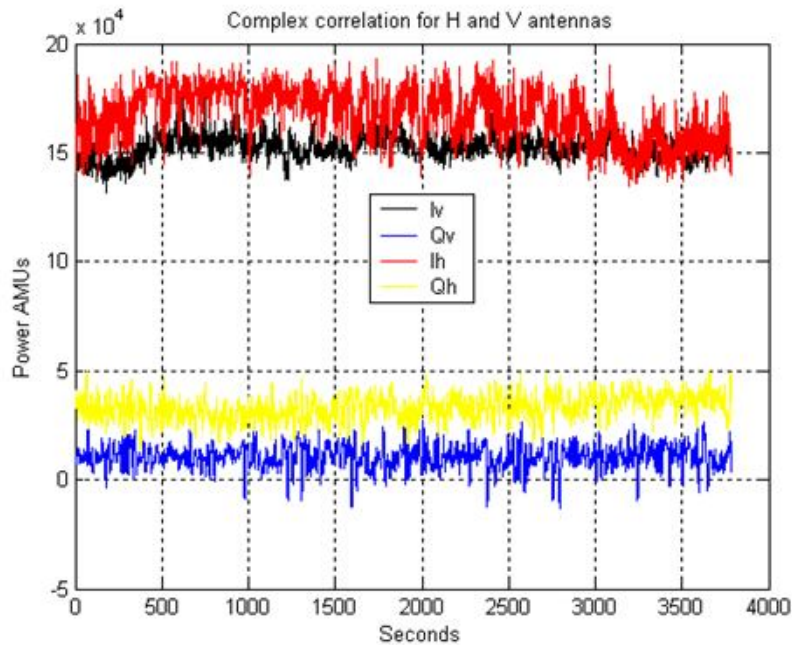


Figura 5-47: Correlaciones complejas obtenidas para las polarizaciones horizontal y vertical. En negro y azul respectivamente, las componentes en fase y cuadratura de la polarización vertical. En rojo y amarillo respectivamente, las componentes en fase y cuadratura de la polarización horizontal.

Los valores de offset obtenidos durante el estado *Internal Reference Uncorrelated* de la UC para esta configuración son los que se muestran en la figura 5-48:

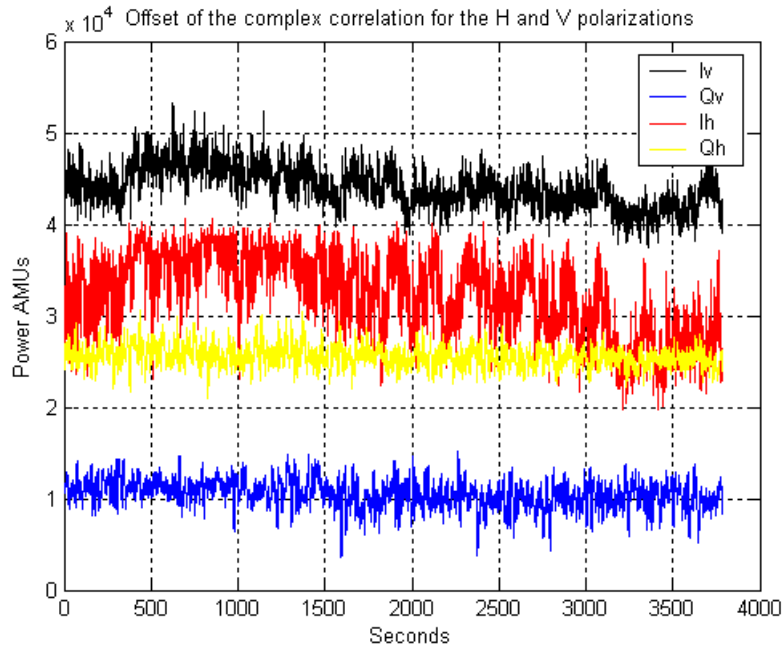


Figura 5-48: Offset de las correlaciones complejas. En negro y azul respectivamente, las componentes en fase y cuadratura de la polarización vertical. En rojo y amarillo respectivamente, las componentes en fase y cuadratura de la polarización horizontal.

Restando estos offsets a las correlaciones complejas de la figura 5-47, se tiene que los valores corregidos son:

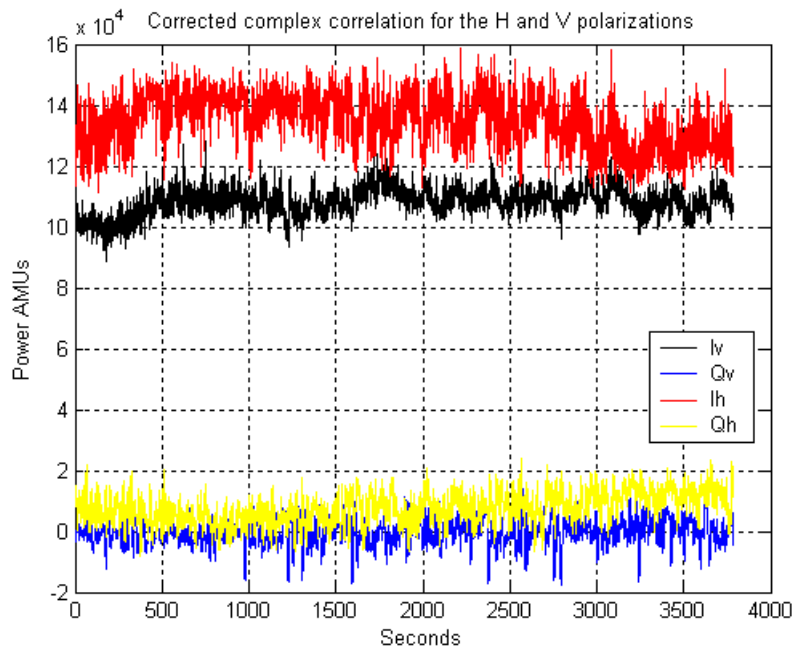


Figura 5-49: Correlaciones complejas con corrección de offset. En negro y azul respectivamente, las componentes en fase y cuadratura de la polarización vertical. En rojo y amarillo respectivamente, las componentes en fase y cuadratura de la polarización horizontal.

Como se puede comprobar, las componentes en cuadratura son cercanas a cero, mientras que las componentes en fase son diferentes de cero y presentan valores próximos entre sí. Se puede afirmar por tanto que, tanto la calibración aplicada a un sistema de 4 receptores, como la conformación de haz funcionan según lo esperado.

5.5.2 Consideraciones necesarias para el escalado a 16 receptores

A la hora de implementar la versión final de PAU-RAD con 16 receptores, hay algunos aspectos que es necesario tener en consideración para garantizar el correcto funcionamiento del sistema.

- **Red de distribución de ruido**

En primer lugar, la versión de 16 receptores del sistema, requiere de la utilización de una red de distribución de ruido, es decir: un sistema encargado de llevar la señal de calibración procedente de la fuente de ruido hasta la entrada de calibración de cada uno de los 16 receptores. Este sistema se compone de un divisor de potencia de 16 salidas y los cables que unen estas salidas a los receptores.

Como es lógico, las señales a la salida del divisor de potencia experimentarán una atenuación, no obstante existen variaciones en los valores de atenuación aplicados a cada una de las salidas. Además de las variaciones en la atenuación, las diferentes longitudes de los cables y el propio divisor introducirán diferentes variaciones de fase con respecto a la fase de la señal de entrada.

Para que la calibración pueda llevarse a cabo satisfactoriamente, es necesario medir estas variaciones de atenuación y fase para cada una de las salidas e incluirlas dentro del proceso de calibración el receptor correspondiente.

Los valores de atenuación y fase medidos para la red de distribución de ruido de PAU-RAD 16 mediante un analizador de redes vectorial se recogen en la siguiente tabla:

Tabla 5-4: Valores de atenuación y fase de la red de distribución de ruido de PAU-RAD 16.

Salida	Atenuación [dB]	Fase [°]	Longitud Cable [cm]	Fase Relativa [°]	Coefficiente Ganancia
1	-12.89	174.63	12	0	1,00
2	-12.85	125.25	14	-49.38	0,98
3	-13.03	-124.19	23.25	50.44	1,03
4	-13.01	146.07	26.50	-28.56	1,02
5	-13.04	-27.50	19.75	-202.13	1,03
6	-12.89	-40.65	20.25	-215.28	0,99
7	-12.85	-157.50	11.50	-332.13	0,99
8	-12.96	-160.60	11.50	-335.23	1,01
9	-13.00	-170.00	11.50	-344.63	1,02
10	-12.89	-157.69	11.50	-332.32	1,00
11	-13.01	-37.96	20.25	-212.59	1,02
12	-13.39	-31.83	19.75	-206.43	1,12
13	-13.04	153.05	26.50	-21.58	1,03
14	-13.23	-123.22	23.25	-297.85	1,08
15	-12.95	128.01	14	-46.62	1,01
16	-12.89	177.88	12	3.25	1,00

La primera columna contiene los valores de atenuación para cada una de las 16 salidas. La segunda columna la fase de la señal a la salida de cada salida. La tercera contiene la longitud del cable que une el divisor de potencia con el receptor. La cuarta columna muestra la fase relativa a la fase de la primera salida que se empleará como referencia y por último, la quinta columna muestra el coeficiente de corrección de ganancia que es necesario aplicar para igualar las atenuaciones de todas las salidas a la atenuación de la primera salida (que de nuevo se emplea como referencia).

Por tanto, es necesario incluir estos valores en el algoritmo de calibración de cada receptor para que la calibración obtenida sea correcta. De la misma forma, si se quiere emplear la calibración diferencial, es necesario incluir las atenuaciones del divisor en los cálculos del hardware de calibración diferencial, de manera que las temperaturas de referencia resultantes sean apropiadas.

○ ***Determinación del desfase entre salidas para cada receptor***

Como ya se vio en el apartado 5.3.8 de este mismo capítulo, cada receptor presenta un cierto coeficiente de desfase entre entradas debido a diferencias en las longitudes de las pistas que unen cada entrada con la etapa de RF.

De la misma manera que es necesario incluir los desfases introducidos por la red de distribución de ruido en el algoritmo de calibración de fase, resulta de vital importancia también incluir estos desfases.

Esto supone que, antes de efectuar medidas con el sistema completo, es necesario medir los desfases entre entradas para cada receptor, ya sea mediante un procedimiento implementado en la UC (tal y como se hizo para el receptor de referencia en el apartado 5.3.8), o bien midiendo cada receptor en un analizador de redes vectorial.

- ***Medida de la respuesta en fase de cada receptor***

En el mismo apartado 5.3.8, se vio también que la fase estimada durante la calibración presenta cierta dependencia con la potencia (y por tanto con la temperatura de ruido equivalente) inyectada a través de la entrada de calibración.

Para garantizar que los 16 receptores operan en la zona de temperaturas estable y que, por tanto, todos se calibran de forma correcta, es conveniente medir las variaciones de fase de cada receptor. De esta forma se puede determinar una potencia de calibración óptima que garantiza la correcta calibración de todo el sistema.

Teniendo en consideración estos tres aspectos del sistema a la hora de llevar a cabo la implementación de 16 receptores, se conseguirá que la calibración del sistema completo se lleve a cabo correctamente.

6 FILTRADO DEL ERROR DE FASE

6.1 INTRODUCCIÓN

6.1.1 Mecanismo de calibración de fase

Antes de comenzar con la descripción de los procesos de filtrado, se recordará brevemente el mecanismo de calibración de fase descrito en el capítulo anterior. A partir de las señales captadas por el receptor, down-convertidas y filtradas paso bajo, se obtienen los coeficientes Z_r y Z_i , que constituyen medidas de similitud entre la fase del canal de referencia y la fase del resto de canales y entre la fase del canal de referencia y la cuadratura del resto de canales respectivamente.

$$Z_{r_1,n} = \frac{1}{N} \sum_{i=1}^N XNOR(i_1 i_{n_i}) \quad (6.1)$$

$$Z_{i_1,n} = \frac{1}{N} \sum_{i=1}^N XNOR(i_1 q_{n_i})$$

Donde i_1 denota la componentes en fase del canal de referencia mientras que i_n y q_n denotan las componentes en fase y cuadratura del resto de canales, con $n = 2, 3$ ó 4 .

Una vez obtenidas las medidas de similitud, podemos calcular correlaciones mediante las expresiones [6]:

$$\mu_{r,n} = \sin\left(\frac{\pi}{2}(2Z_r - 1)\right) = \frac{R_{i_1 i_n}(0)}{\sqrt{R_{i_1 i_1}(0)R_{i_n i_n}(0)}} \quad (6.2)$$

$$\mu_{i,n} = \sin\left(\frac{\pi}{2}(2Z_i - 1)\right) = \frac{R_{i_1 q_n}(0)}{\sqrt{R_{i_1 i_1}(0)R_{q_n q_n}(0)}}$$

El error de fase relativo de un canal con respecto al canal de referencia se obtiene mediante la expresión 6.3:

$$(\phi_1 - \phi_n) = \tan^{-1} \left(\frac{-\mu_{i,n}}{\mu_{r,n}} \right) \quad (6.3)$$

Todo el proceso de estimación y corrección del error de fase relativo entre canales se esquematiza de forma resumida en la siguiente figura:

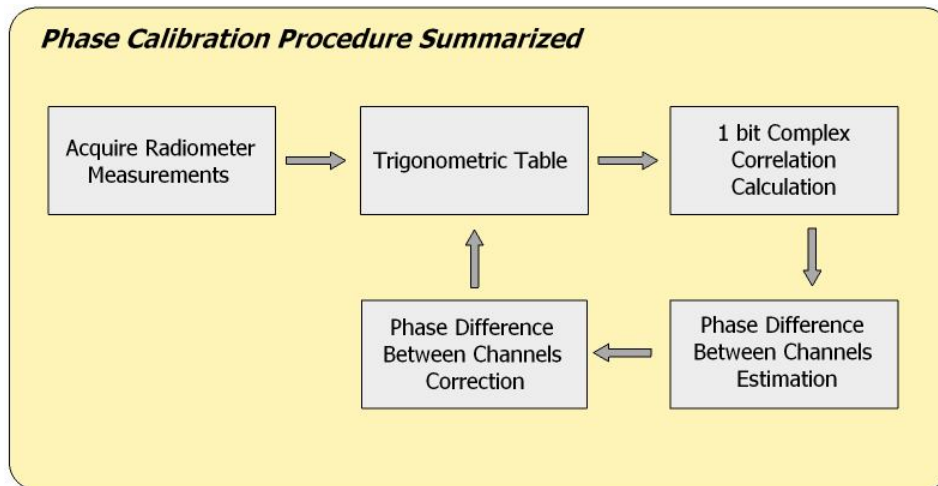


Figura 6-1: Diagrama de flujo que esquematiza de forma simple el proceso de estimación y corrección del error de fase entre canales.

A las medidas obtenidas por el radiómetro se aplican las correcciones de fase y amplitud almacenadas en las tablas trigonométricas. A partir de estas medidas se realiza el cálculo de los valores de correlación utilizando señales de 1 bit, valores que se emplean para determinar la diferencia de fase entre el canal de referencia y el resto de canales. Finalmente la corrección del error de fase para cada canal se escribe en las tablas trigonométricas, que se utilizarán para corregir las siguientes medidas.

El objetivo es incorporar un nuevo bloque a este diagrama de flujo, justo después de la estimación de la diferencia de fase entre canales, en el que se lleve a cabo el filtrado de la estimación de fase.

6.1.2 Justificación al filtrado

La finalidad del filtrado es reducir la desviación típica de la medida del error de fase. Dado que la estimación de fase de cada uno de los cuatro canales lleva asociada una función de distribución de probabilidad gaussiana centrada en el error de fase correspondiente y con una cierta desviación típica, al efectuar la corrección de fase

estas propiedades estadísticas se transmiten al error residual de fase medido. Esto se traduce en un error de fase medido con una función de distribución de probabilidad gaussiana, centrada en cero (puesto que el error ha sido corregido) y con la misma desviación típica que la gaussiana de la estimación de fase.

Reducir la desviación típica del error de fase medido conlleva, no sólo una mejora en la precisión de todo el sistema, puesto que el error residual oscila de forma menos pronunciada en torno a cero, sino que además protege al sistema frente a posibles "glitches" de fase, que pueden inducir errores en el cálculo de los parámetros de Stokes .

Cada una de las 16 posiciones de la tabla trigonométrica encargada de corregir los errores de fase y amplitud contiene una expresión trigonométrica del tipo:

$$TT = \alpha \cos(2\pi f + \theta_{Error} - \theta_{Array}) \quad (6.4)$$

Donde α se corresponde con el factor de corrección de amplitud, θ_{Error} es la corrección de fase y θ_{Array} es la fase necesaria para el beamforming.

En primer lugar se realiza la corrección de fase, de forma que se obtiene un resultado numérico para la expresión trigonométrica correspondiente. Este resultado se digitaliza mediante 9 bits y se escribe en la posición de la tabla trigonométrica que le corresponde. Realizada esta operación, la siguiente captura de datos del radiómetro ya está corregida en fase, y se puede por tanto corregir la ganancia.

No obstante, al corregir la ganancia, se reescriben las posiciones de la tabla trigonométrica, deshaciendo la corrección de fase realizada previamente. Debido a esto resulta de vital importancia proteger el sistema frente a posibles "glitches" de fase, puesto que si se corrige una muestra de fase aberrante la corrección de ganancia no se efectuará correctamente durante varias muestras seguidas hasta que el sistema se estabilice de nuevo, lo que se traduce en un cálculo incorrecto de los parámetros de Stokes durante un cierto período de tiempo.

6.1.3 Tipos de filtros considerados

Dado que la señal a filtrar se puede considerar como una constante (idealmente cero cuando el error ha sido corregido) con una componente de alta frecuencia superpuesta (ruido gaussiano blanco), se han estudiado filtros digitales de tipo paso bajo con el fin de obtener la constante y descartar la mayor cantidad de ruido posible.

Concretamente se han considerado dos tipos distintos de filtros paso bajo calculados según la misma aproximación (aproximación de Butterworth), además de un tercer tipo de filtro adaptativo:

- Filtro de Respuesta Impulsional Infinita (IIR)
- Filtro de Respuesta Impulsional Finita (FIR)
- Filtro de Kalman

Como paso previo a la implementación de los filtros mediante un algoritmo C, ejecutado en la Unidad de Control del radiómetro, se estudió el comportamiento de los tres filtros descritos anteriormente con la ayuda de Matlab.

La finalidad de este estudio es la de obtener los coeficientes necesarios para la implementación de los filtros (en el caso de los filtros IIR y FIR) en la Unidad de Control. Al mismo tiempo, Matlab constituye una plataforma de pruebas excelente para verificar si los resultados obtenidos son apropiados antes de realizar modificaciones al código C de la Unidad de Control.

6.2 FILTRADO DE FASE MEDIANTE FILTROS DE RESPUESTA IMPULSIONAL INFINITA (IIR)

6.2.1 Características de los filtros IIR

Como su propio nombre indica, este tipo de filtros presentan una respuesta impulsional de duración infinita. El filtro implementado es del tipo ARMA (Auto Regressive Moving Average), el más general. La ecuación diferencia que describe el filtro es la siguiente:

$$\sum_{i=0}^N a_i Y[n-i] = \sum_{m=0}^N b_m X[n-m] \quad (6.5)$$

Donde Y representa la señal a la salida del filtro, X representa la señal a la entrada del filtro, N representa el orden del filtro y b_m y a_i son las secuencias de coeficientes del filtro.

La respuesta del filtro es, por tanto, una suma ponderada de valores pasados y presentes de la entrada. De ahí que se denomine: Media en Movimiento (Moving Average).

Por otra parte, el término Auto Regresivo (Auto Regressive), hace referencia a la tendencia estadística de la salida del filtro a tender hacia sus valores pasados. Esta característica es debida a que los filtros ARMA son de tipo recursivo, ya que la salida depende, no solo de la entrada actual, sino que además depende de valores pasados de la salida. Se trata por tanto de filtros realimentados.

Esta ecuación diferencia da lugar a una función de transferencia de la forma:

$$H(z) = \frac{b_0 + b_1 Z^{-1} + \dots + b_N Z^{-N}}{1 + a_1 Z^{-1} + \dots + a_N Z^{-N}} \quad (6.6)$$

Se trata por tanto de una función de transferencia que presenta tanto polos como ceros, lo que hace que los filtros de este tipo puedan ser inestables. Además de su posible inestabilidad, los filtros IIR tienen como inconveniente el hecho de introducir distorsión de fase, es decir que la fase de la salida del filtro no es lineal con la frecuencia.

La contrapartida a estos inconvenientes es el hecho de no requerir un orden excesivo para cumplir con unas especificaciones determinadas, lo que reduce la carga computacional del algoritmo.

6.2.2 Diseño del filtro

Para determinar el orden y los coeficientes del filtro y estudiar su comportamiento, se desarrolló un código Matlab que simula una estimación de fase con las mismas características estadísticas que la estimación de fase real realizada por el radiómetro.

La estimación de fase simulada tendrá, por tanto, una función de distribución de probabilidad Gausiana, centrada en un cierto valor de fase y con una cierta desviación típica.

En el estudio que se presenta a continuación, se fijó un valor de fase estimada de 0 grados y una desviación típica de 2.5 grados, tal y como se muestra en la siguiente figura:

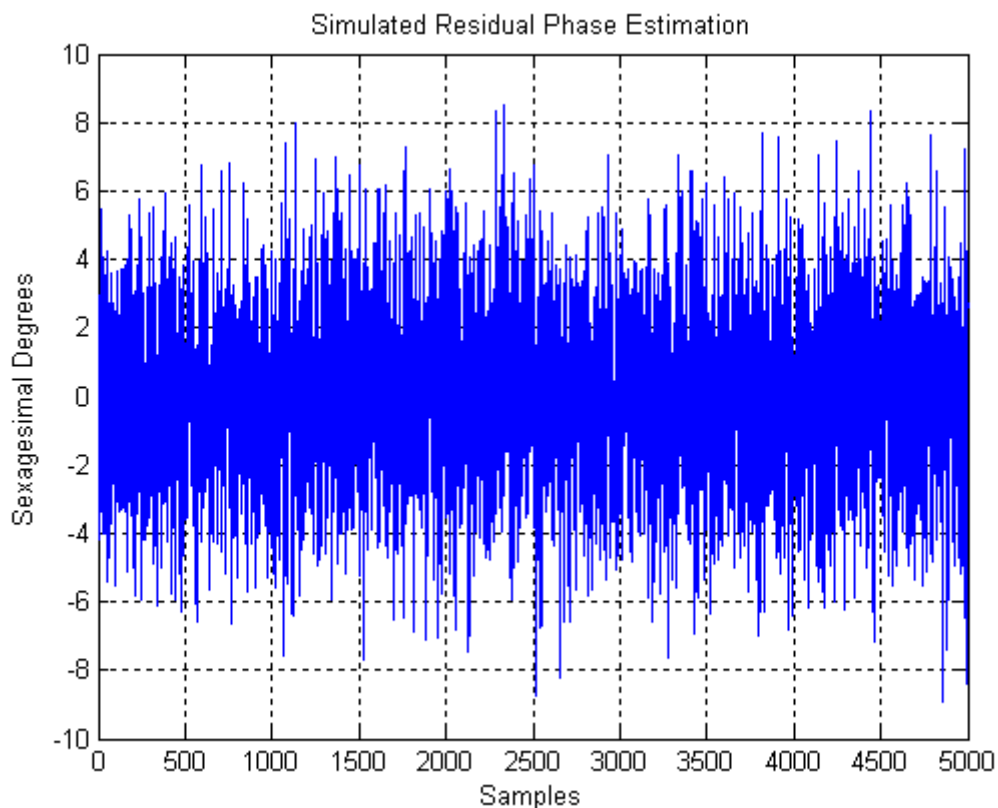


Figura 6-2: Estimación de fase residual simulada empleada como entrada en el estudio de los diversos filtros.

A continuación se utilizaron las herramientas de que dispone Matlab para calcular los coeficientes del filtro y se aplicó el filtro diseñado a esta estimación de fase simulada. Para el caso de los filtros de respuesta Impulsional infinita se estudiaron cuatro filtros de diferente orden: IIR1 con $N = 1$, IIR2 con $N = 2$, IIR3 con $N = 4$ y finalmente IIR4 con $N = 6$.

Con el fin de poder comparar los resultados para los diferentes órdenes, se fijaron las mismas especificaciones para todos los filtros: frecuencia de paso 1 Hz, frecuencia de atenuación 5 Hz, atenuación máxima en la banda de paso 0,1 dB y atenuación mínima en la banda atenuada de 50 dB. Es importante tener en cuenta que estas especificaciones son muy restrictivas tanto en frecuencia como en atenuación, de hecho el orden mínimo que Matlab establece para garantizarlas es de $N = 5$. Por tanto tan sólo uno de los filtros estudiados (el filtro de orden $N = 6$) cumple completamente estas características.

En la figura 6-3 se muestran las respuestas frecuenciales para cada uno de los cuatro órdenes considerados. Como ya se ha comentado anteriormente tan sólo el filtro de orden 6 cumple los requisitos de atenuación mínima en la banda atenuada requeridos debido a la imposición de un orden por debajo del mínimo necesario.

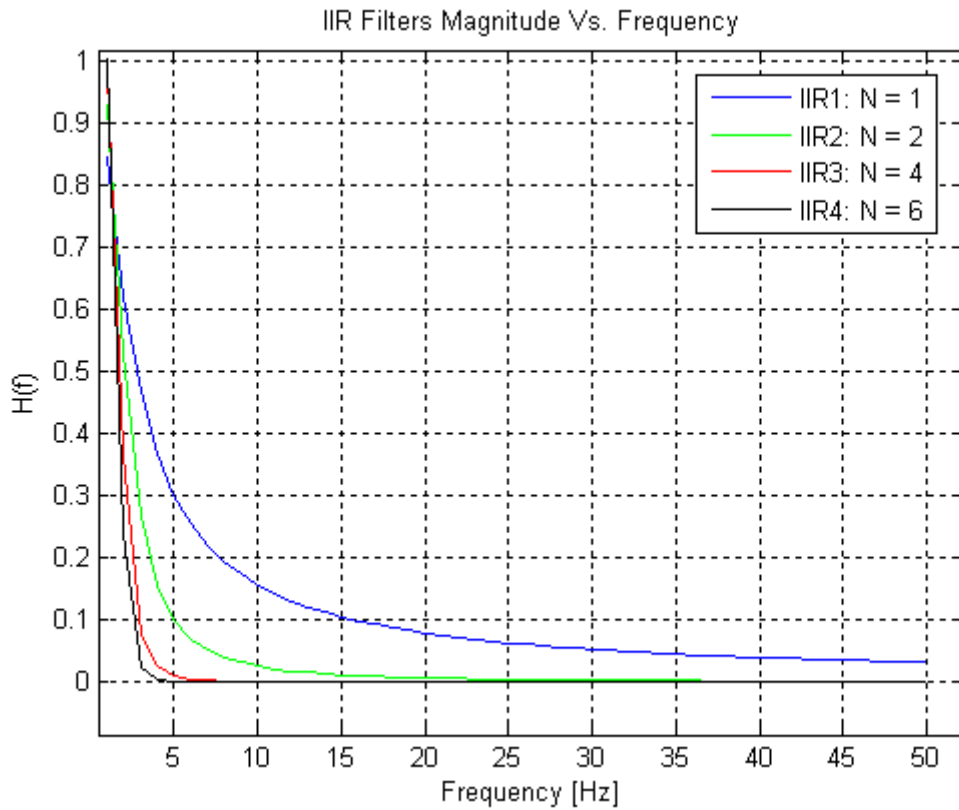


Figura 6-3: Respuesta frecuencial para filtros IIR de diferente orden.

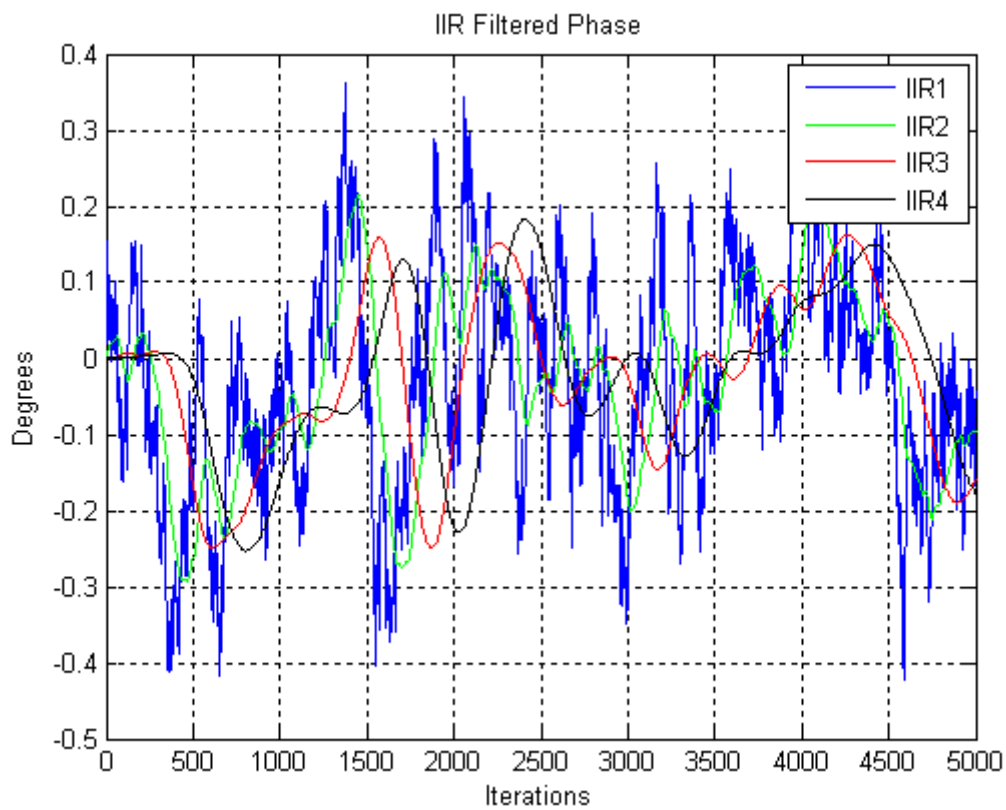


Figura 6-4: Resultado del filtrado de fase mediante filtros IIR de diversos órdenes para una estimación de fase de 0 grados.

La figura 6-4 muestra el resultado de aplicar a la estimación de fase simulada, cada uno de los cuatro filtros por separado. Como se puede comprobar si se comparan las gráficas 6-2 y 6-4 la desviación típica se ha visto considerablemente reducida en los cuatro casos.

Esta reducción de la desviación típica se aprecia de una forma más clara si se observa la figura 6-5, que permite establecer una comparación directa entre el histograma de la estimación de fase sin filtrar (la primera de las gráficas) y los histogramas de las fases filtradas.

Como se puede comprobar, la desviación típica se ha reducido respecto a la desviación típica original en aproximadamente un 94% para el filtro de orden 1, un 95,3% para el filtro de orden 2, un 95,6% para el filtro de orden 4 y un 95,7% para el filtro de orden 6.

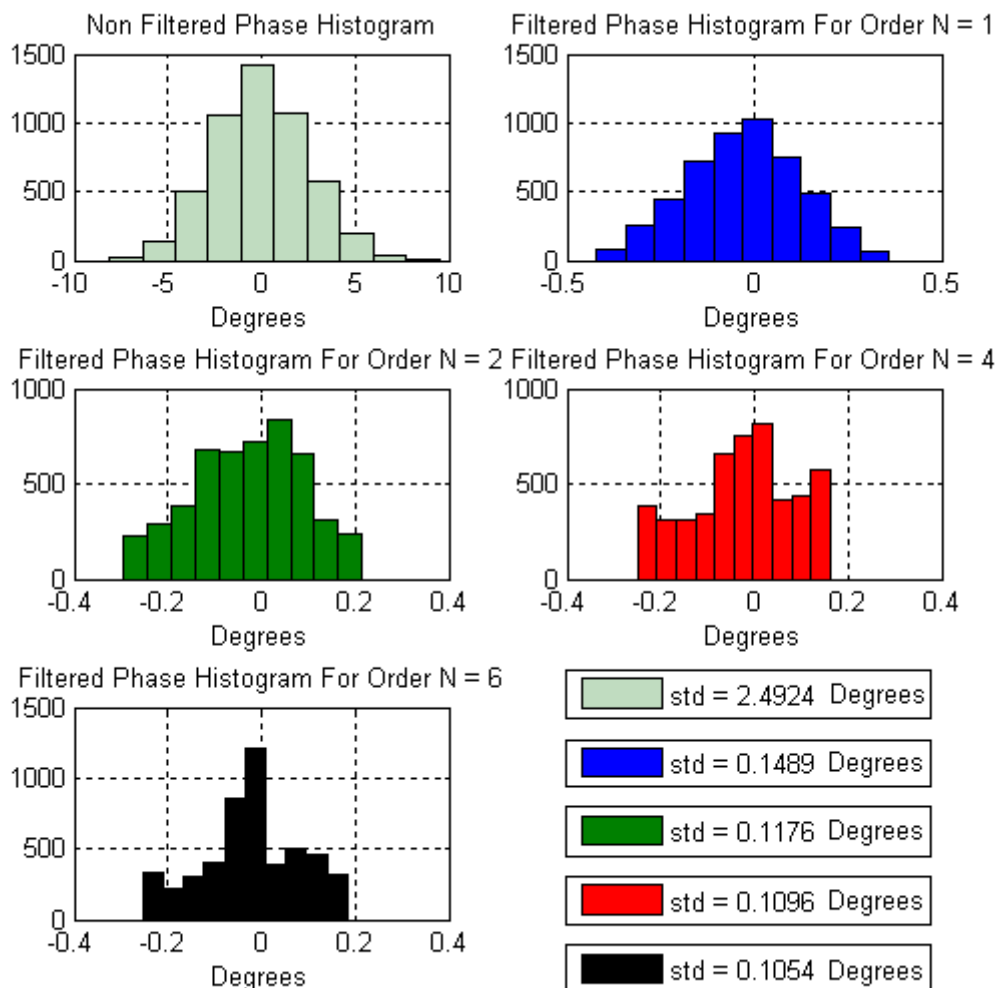


Figura 6-5: Histogramas correspondientes a la estimación de fase sin filtrar y a la estimación de fase filtrada mediante los cuatro filtros IIR considerados.

A la luz de los resultados obtenidos se optó por implementar el filtro de orden $N = 2$ (IIR2). Esta decisión se debe principalmente a que ofrece un buen compromiso entre reducción de varianza y facilidad de implementación, en el sentido de que tan sólo requiere almacenar dos muestras de la fase a filtrar y dos muestras de la fase ya filtrada. De este modo se reduce considerablemente con respecto a los filtros de orden superior la cantidad de variables necesarias para su implementación y al mismo tiempo se aligera la carga computacional del procesador al requerir un menor número de multiplicaciones.

6.2.3 Resultados obtenidos en el simulador

Una vez finalizado el diseño del filtro y como paso previo a su implementación en la unidad de control del radiómetro, se realizaron pruebas para determinar su comportamiento en un escenario similar al que ofrece PAU. Para ello se desarrolló una simulación en Matlab que recrea el proceso de estimación y corrección del error de fase entre dos canales que se lleva a cabo en el radiómetro.

Un esquema simplificado del proceso que se lleva a cabo en este simulador es el siguiente:

- Se generan y se componen las señales de dos canales, uno de los cuales estará desplazado respecto al otro un cierto error de fase.
- Se lleva a cabo la down-conversión de las señales procedentes de los dos canales. Para el canal 2 se aplica la corrección de fase que se estimará más adelante. El resultado se filtra paso-bajo para eliminar las componentes de alta frecuencia resultantes de la down-conversión.
- Se calculan los valores de similitud entre las componentes en fase y cuadratura de los dos canales ($\langle I_1 I_1 \rangle, \langle I_1 Q_1 \rangle, \langle I_1 I_2 \rangle, \langle I_1 Q_2 \rangle$).
- A partir de los valores de similitud normalizados con respecto a $\langle I_1 I_1 \rangle$ se obtienen los valores de pseudo-correlación para el canal 2 (μ_i, μ_r). Mediante estos valores se calcula la estimación de la diferencia de fase del canal 2 respecto del canal 1.
- La diferencia de fase estimada constituye el error de fase, que es acumulado y filtrado mediante cualquiera de los filtros sometidos a estudio.
- Finalmente el error de fase filtrado se realimenta al proceso de down-conversión tal y como se ha comentado en el segundo punto del algoritmo.

La figura 6-6 muestra los resultados de esta simulación para el filtro de orden 2 diseñado en el apartado anterior.

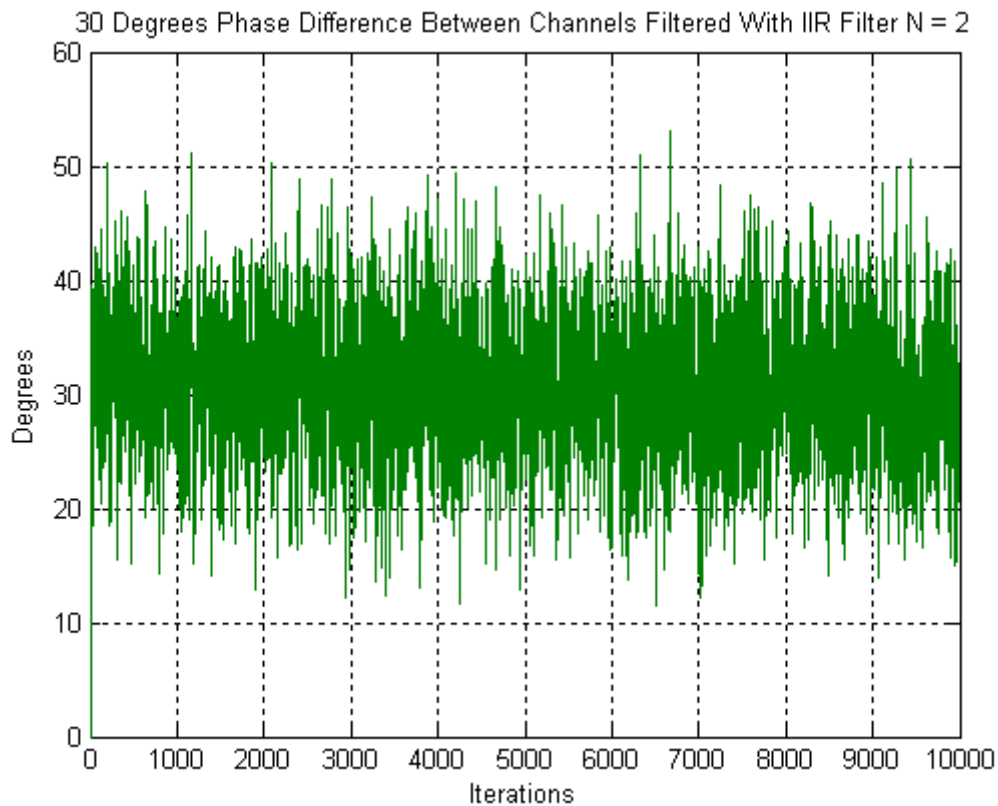


Figura 6-6: Resultados obtenidos mediante simulación del filtrado de fase con un filtro IIR de orden 2 para un error de fase entre canales de 30 grados.

Como se puede comprobar, los resultados distan mucho de los obtenidos en la etapa de diseño. En el simulador se obtiene una desviación típica de 5.9 grados, frente a los 0.11 grados esperados. El filtro diseñado no sólo no reduce la desviación típica, sino que además la aumenta, pasando de aproximadamente 2.5 grados a 5.9 grados.

6.2.4 Análisis de los resultados y conclusiones

A pesar de los malos resultados obtenidos en el simulador, para descartar un mal funcionamiento del mismo, se implementó el filtro en la unidad de control del radiómetro. Para ello se utilizaron los mismos coeficientes obtenidos en la etapa de diseño.

La figura 6-7 muestra el error de fase residual de los cuatro canales del radiómetro, aplicando el filtro diseñado a los datos capturados por el radiómetro durante la ejecución de una secuencia *Closed Loop*:

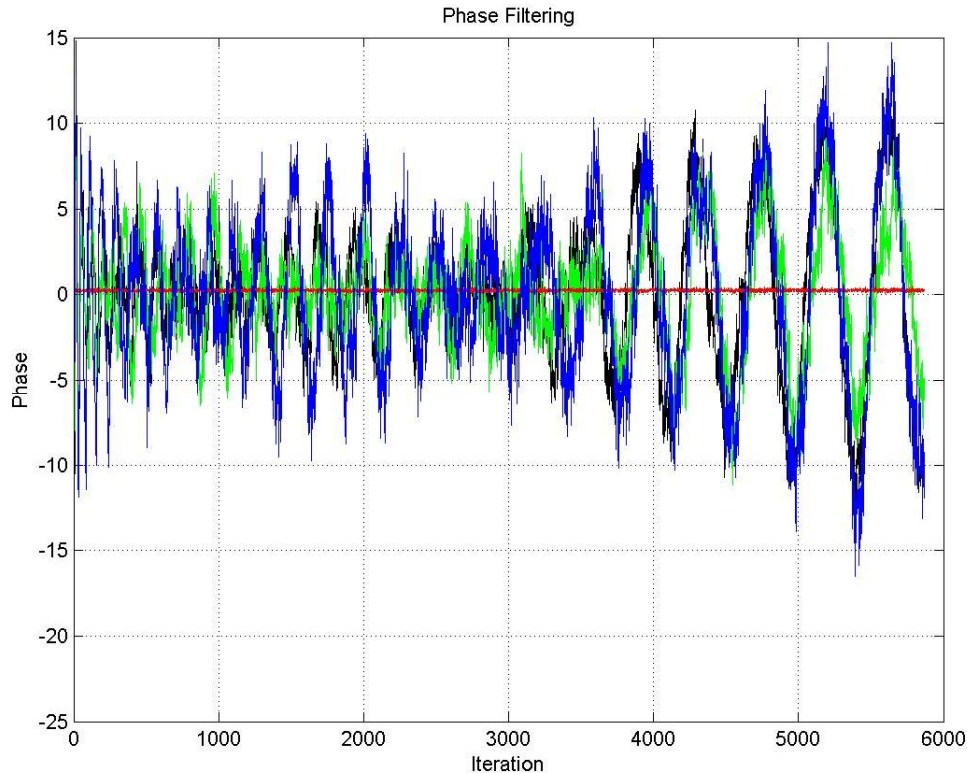


Figura 6-7: Resultados del filtrado del error de fase mediante filtro IIR. Se trata de datos reales obtenidos a partir del radiómetro durante la calibración, no de simulaciones.

En la figura se aprecia cómo, desde el primer instante hay una cierta inestabilidad en los resultados. A pesar de que en la captura que se muestra en la figura 6-7, el error de fase residual se mantiene siempre en la franja que va de -15° a $+15^\circ$, capturas de mayor duración revelan que, lejos de corregirse, esta desestabilización del filtro se va incrementando con el paso del tiempo.

Llegados a este punto se relajaron las especificaciones iniciales de diseño del filtro, con la esperanza de conseguir mejores resultados. Así pues, se redujo la atenuación en la banda atenuada, se aumentó la atenuación en la banda de paso, se aumentó la separación entre las frecuencias de banda de paso y banda eliminada y se incrementó el orden del filtro. Ninguna de las medidas adoptadas mejoró el funcionamiento del filtro.

Después de numerosas pruebas se llegó a la conclusión de que, probablemente, la presencia de polos en la función de transferencia del filtro, junto con la estructura de estimación de fases de PAU, eran las causantes de la desestabilización de la fase. Por tanto se descartó definitivamente el uso de filtros de respuesta impulsional infinita y se procedió al estudio de los filtros de respuesta impulsional finita.

6.3 FILTRADO DE FASE MEDIANTE FILTROS DE RESPUESTA IMPULSIONAL FINITA (FIR)

6.3.1 Características de los filtros FIR

A diferencia de los filtros de respuesta impulsional infinita, los filtros de tipo FIR tienen una respuesta impulsional de duración finita. La relación entre la entrada y la salida del filtro, viene determinada por la siguiente expresión:

$$Y[n] = \sum_{m=0}^N b_m X[n-m] \quad (6.7)$$

Donde Y representa la señal a la salida del filtro, X representa la señal a la entrada del filtro, N representa el orden del filtro y b_m son los coeficientes del filtro. Asimismo, la función de transferencia del filtro obedece a la siguiente expresión:

$$H(z) = b_0 + b_1 z^{-1} + \dots + b_N z^{-N} \quad (6.8)$$

Como se puede observar, la función de transferencia del filtro presenta únicamente ceros, por tanto los filtros de tipo FIR serán siempre estables. Como contrapartida a la estabilidad, este tipo de filtros requiere un orden superior al de los filtros de tipo IIR para proporcionar resultados similares. Esto se traduce en la necesidad de un mayor número de muestras para calcular la salida del filtro en un instante determinado, añadiendo carga computacional al algoritmo de corrección del error de fase. Además el instante de salida de la primera muestra filtrada se verá retardado, ya que son necesarias N muestras de la señal de entrada antes de empezar a filtrar.

6.3.2 Diseño del filtro

Al contrario de lo que ocurre con los filtros de tipo IIR, Matlab carece de herramientas para determinar el orden de los filtros FIR. Por tanto se realizaron pruebas incrementando sucesivamente el orden del filtro hasta alcanzar una reducción de la desviación típica de la fase satisfactoria. Con el fin de poder establecer comparaciones, se mantuvieron los mismos parámetros de atenuación y las mismas frecuencias de corte y de paso que se emplearon en el diseño del filtro IIR.

Como punto de partida se fijó el mismo orden que el calculado para el filtro IIR ($N = 2$). Como se puede apreciar en la figura 6-8, la reducción de la desviación típica es muy inferior a la que se obtuvo mediante el filtro IIR del mismo orden.

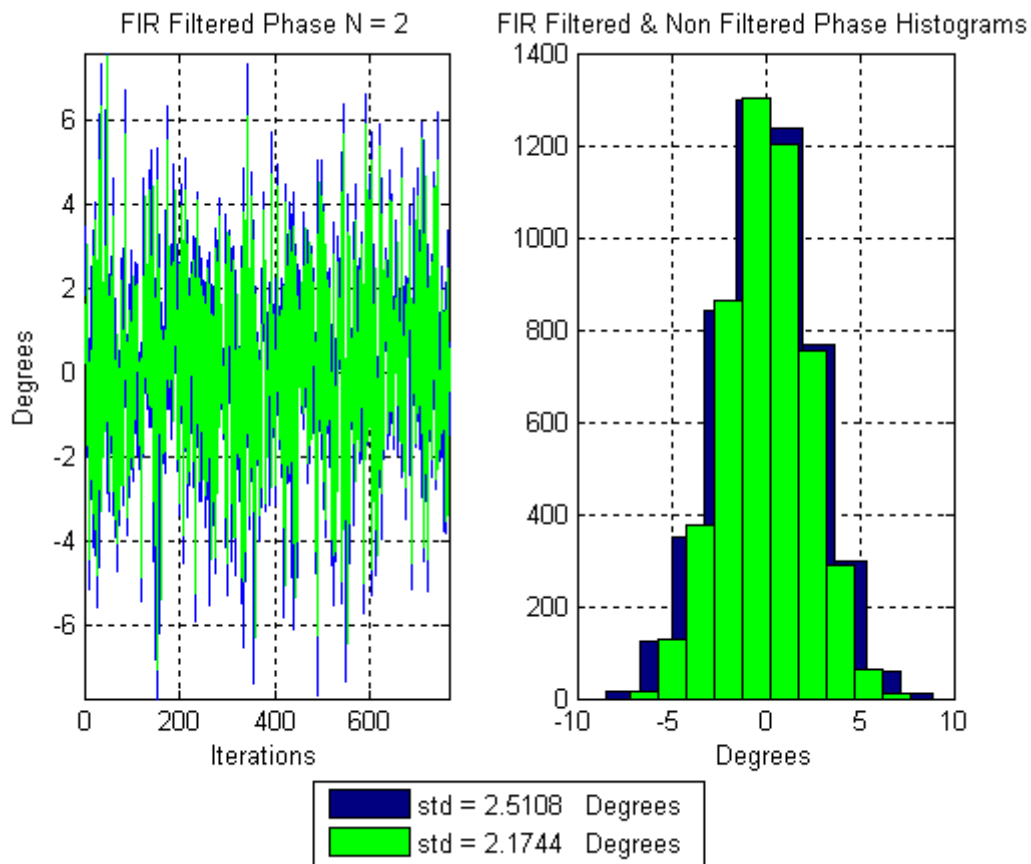


Figura 6-8: Comparativa entre la fase sin filtrar y la fase filtrada mediante un filtro FIR de orden $N = 2$. A la izquierda la fase sin filtrar (azul) y la fase filtrada (verde). A la derecha los histogramas correspondientes.

Tras realizar incrementos sucesivos del orden del filtro y observar los resultados, se llegó a la conclusión de que para obtener una reducción de la varianza de la fase estimada similar a la que se obtuvo para el filtro IIR de orden 2, era necesario un filtro FIR de orden superior a 200. La implementación en la unidad de control de un filtro de semejante orden es complicada y consumiría gran cantidad de recursos del procesador, retrasando todo el proceso de obtención de muestras del radiómetro.

Con el fin de reducir la complejidad de la implementación del filtro, se fijó un orden de $N = 12$. Tal y como puede comprobarse en la figura 6-9, para este orden es posible reducir en aproximadamente un 33% la desviación típica de la fase.

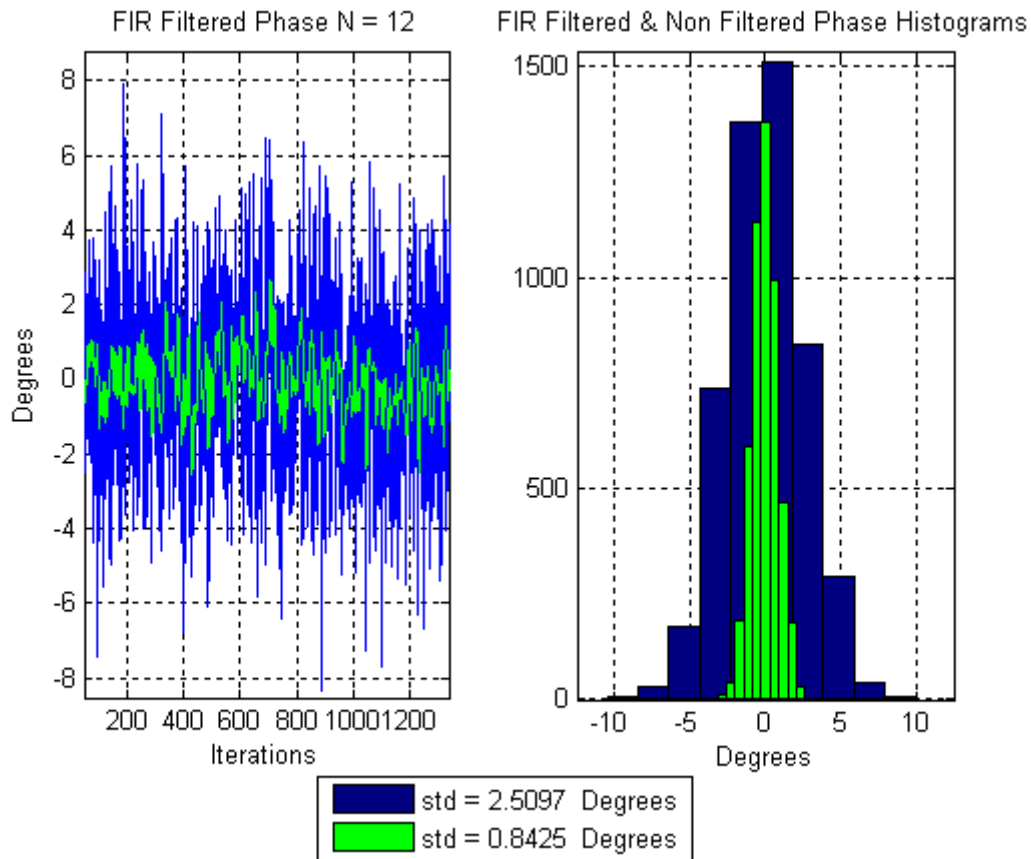


Figura 6-9: Comparativa entre la fase sin filtrar y la fase filtrada mediante un filtro FIR de orden $N = 2$. A la izquierda la fase sin filtrar (azul) y la fase filtrada (verde). A la derecha los histogramas correspondientes.

6.3.3 Resultados obtenidos en el simulador

De nuevo, antes de proceder a la implementación definitiva del filtro en la unidad de control, se validó el funcionamiento del mismo mediante el simulador de PAU (ver apartado 6.2.3).

Los resultados obtenidos se pueden observar en la figura 6-10. Al igual que en el caso del filtro FIR, la diferencia de fase entre canales es de 30 grados. Como se puede comprobar, la desestabilización del filtro es total. En esta ocasión ni siquiera la media de la estimación de fase es correcta (es de aproximadamente 90 grados, cuando debería de ser de 30), y la desviación típica es de aproximadamente 52 grados.

Como ya se hiciera para el filtro FIR, a tenor de los resultados obtenidos en el simulador se intentó aumentar el orden del filtro al mismo tiempo que se relajaban las especificaciones. Ninguna de las variaciones del filtro propuesto inicialmente en la etapa de diseño proporcionó resultados satisfactorios.

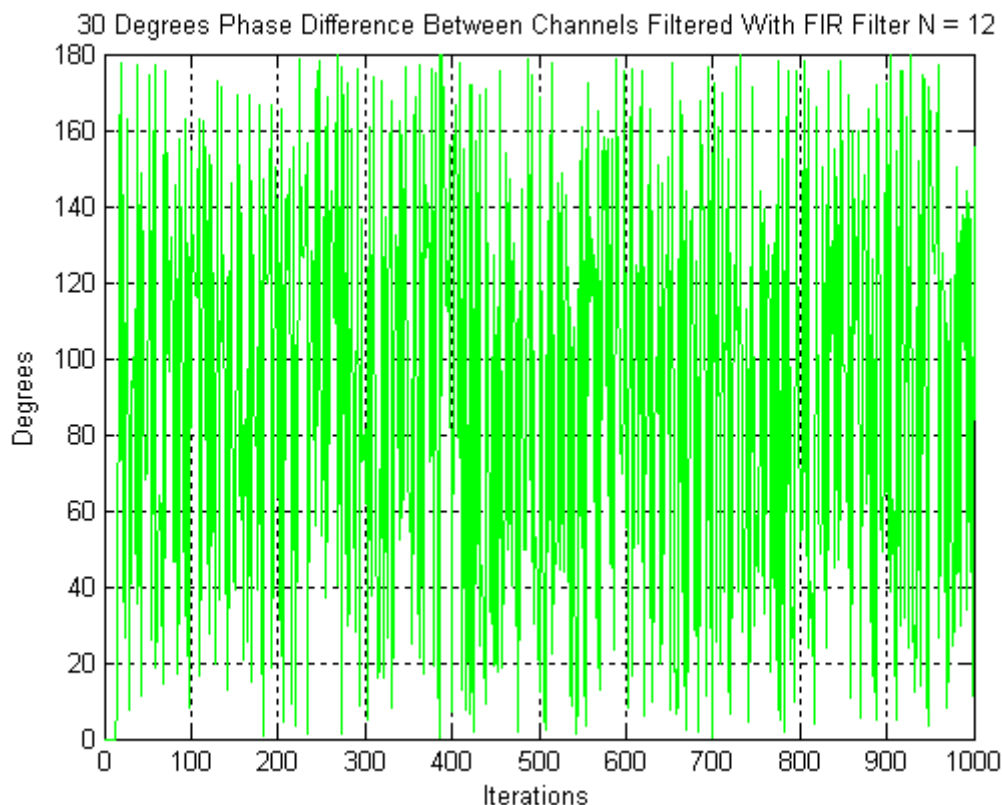


Figura 6-10: Resultados obtenidos mediante simulación del filtrado de fase con un filtro FIR de orden 12 para un error de fase entre canales de 30 grados.

6.3.4 Análisis de los resultados y conclusiones

A pesar de que los resultados de las simulaciones dejaban claro lo inapropiado del filtro, se implementó en la unidad de control un filtro semejante al diseñado, pero de menor orden. La finalidad era, de nuevo, asegurar el correcto funcionamiento del simulador y comprobar que se producía la misma desestabilización.

Tal y como ocurrió con la implementación del filtro IIR, el filtro FIR provocó una desestabilización de la fase transcurridas apenas unas cuantas muestras. Después de efectuar numerosas pruebas con diferentes filtros FIR, todas ellas con idéntico resultado, la hipótesis inicial que había motivado el paso de un diseño de respuesta impulsional infinita a otro de respuesta impulsional finita fue descartada.

Quedaba así probado que la presencia de polos en la función de transferencia del filtro no era la causante de los problemas del filtrado, ya que los filtros FIR son completamente estables y provocaban el mismo efecto. Por tanto la causa de la desestabilización de las fases filtradas debía residir en el propio mecanismo de estimación de fases de PAU.

Esta segunda hipótesis se probó aplicando los filtros diseñados a la estimación de fase, pero sin volver a introducir en el sistema la fase filtrada. Es decir: la estimación de fase utilizada para realizar la corrección no pasaba por ningún filtro, mientras que la fase filtrada se representaba en la interface Matlab sin ser inyectada de nuevo al sistema. En estas condiciones los resultados fueron los esperados: la fase filtrada se correspondía con la obtenida en la etapa de diseño y la fase corregida se mantenía estable en una media de 0 grados.

El problema del filtrado de fase debía radicar por tanto en el hecho de realimentar la fase filtrada al sistema. Llegados a este punto se pensó que la solución al problema podría pasar por el uso de filtros adaptativos, es decir: cuyos coeficientes se calculasen nuevamente a cada iteración del sistema para mantenerlo estable. De ahí surgió la idea de la utilización de un filtro de Kalman.

6.4 FILTRADO DE FASE MEDIANTE FILTRO DE KALMAN

6.4.1 Características del filtro de Kalman

El tercer tipo de filtro que se ha considerado es el filtro de Kalman. A diferencia de los filtros comentados anteriormente, el filtro de Kalman es adaptativo, es decir: sus coeficientes no son fijos, sino que se re calculan a cada iteración.

Las ecuaciones que definen el comportamiento del filtro de Kalman son las siguientes:

$$\phi_f[n] = \phi_f[n - 1] + K[n - 1](\phi_m[n] - \phi_f[n - 1]) \quad (6.9)$$

Donde ϕ_f denota la fase filtrada, ϕ_m la fase medida y K es el coeficiente de Kalman, que se define como:

$$K[n] = \frac{\sigma^2_{\phi_f}[n]}{\sigma^2_{\phi_f}[n] + \sigma^2_{\varepsilon}} \quad (6.10)$$

Donde σ^2_{ε} es la varianza del error asociado a la fase que queremos filtrar (en nuestro caso, la fase medida: ϕ_m) y $\sigma^2_{\phi_f}$ se define como la varianza de la fase filtrada, que se calcula para cada iteración a partir de la ecuación:

$$\sigma^2_{\phi_f}[n] = (1 - K[n - 1]) \sigma^2_{\phi_f}[n - 1] \quad (6.11)$$

La inicialización del filtro de Kalman requiere por tanto dos parámetros: por una parte, una estimación inicial de la varianza de la fase filtrada, es decir: $\sigma^2_{\phi_f}[0]$ y por otra, la varianza del error que se asocia a la estimación de fase que realiza el radiómetro, en nuestro caso: σ^2_{ε} . Es posible fijar a cualquier valor $\sigma^2_{\phi_f}[0]$, ya que se trata de un algoritmo adaptativo y rápidamente convergerá su valor real. Por otra parte el valor de σ^2_{ε} se midió directamente en varias estimaciones de fase realizadas por el radiómetro, fijando su valor en aproximadamente 3.34° .

6.4.2 Diseño del filtro

A diferencia de los dos casos anteriores, el diseño del filtro de Kalman no requiere determinación de un orden ni fijar unas especificaciones de atenuación o frecuencias. El diseño del filtro de Kalman se limita a la implementación mediante Matlab de un bucle con las tres ecuaciones descritas anteriormente y a la inicialización de los parámetros necesarios ($\sigma^2_{\varepsilon} = 3,34^\circ$ y $\sigma^2_{\phi_f}[0] = 0$).

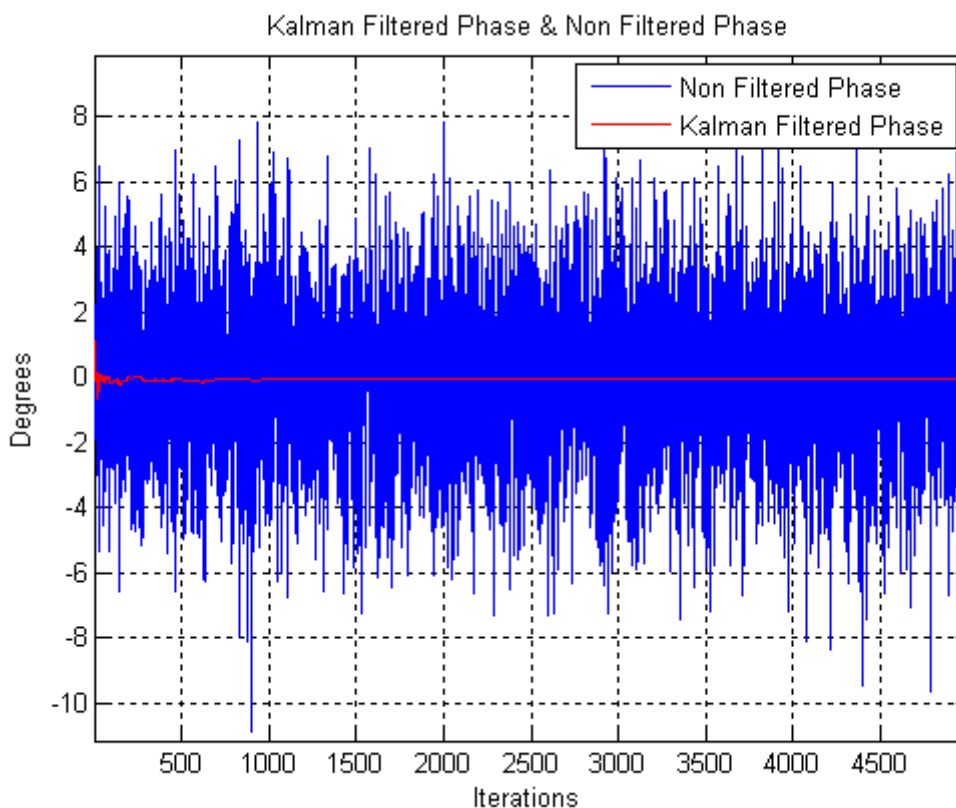


Figura 6-11: Comparativa entre la fase sin filtrar y la fase filtrada mediante un filtro de Kalman.

La figura 6-11 muestra una comparativa entre la fase sin filtrar y la fase filtrada. Como se puede ver, los resultados obtenidos son espectaculares, mejores incluso que los obtenidos para el filtro IIR. Esta mejora se hace más patente si se observa la figura 6-12, en la que se muestran los histogramas y desviaciones típicas para las fases sin filtrar y filtrada.

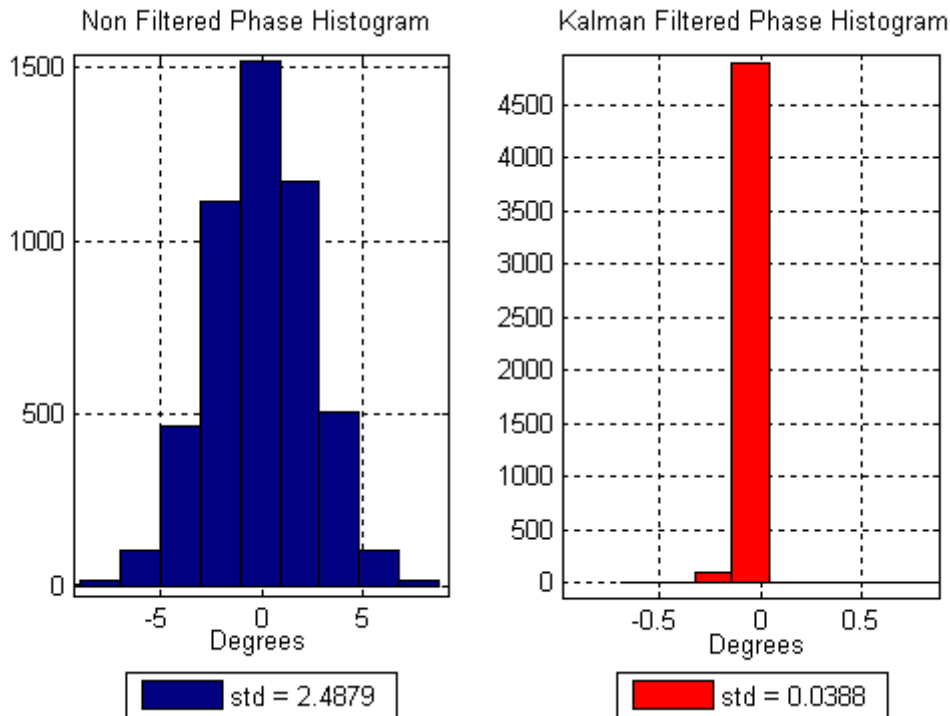


Figura 6-12: Histogramas correspondientes a la fase sin filtrar (azul) y la fase filtrada mediante un filtro de Kalman (rojo).

Se aprecia cómo mediante el uso del filtro de Kalman, es posible reducir la desviación típica de la fase residual de 2.5 grados a 0.0388 grados, lo que supone una gran mejora con respecto al resto de filtros estudiados.

6.4.3 Resultados obtenidos con el simulador

En esta ocasión los resultados de las simulaciones son mucho más alentadores que en el caso de los filtros IIR y FIR. La figura 6-13 muestra la estimación de fase filtrada en el simulador, para una diferencia de fase entre canales de 30 grados.

A pesar de que la desviación típica de la estimación de fase filtrada en el simulador (1.37 grados) dista mucho de la obtenida en la etapa de diseño (0.0388 grados), el hecho de que esta se reduzca y que, al mismo tiempo, la estimación de fase permanezca estable constituye un gran avance con respecto a los casos anteriores.

Además si eliminamos las 2000 primeras muestras que contienen el transitorio del filtro, la desviación típica se reduce a 0.86 grados.

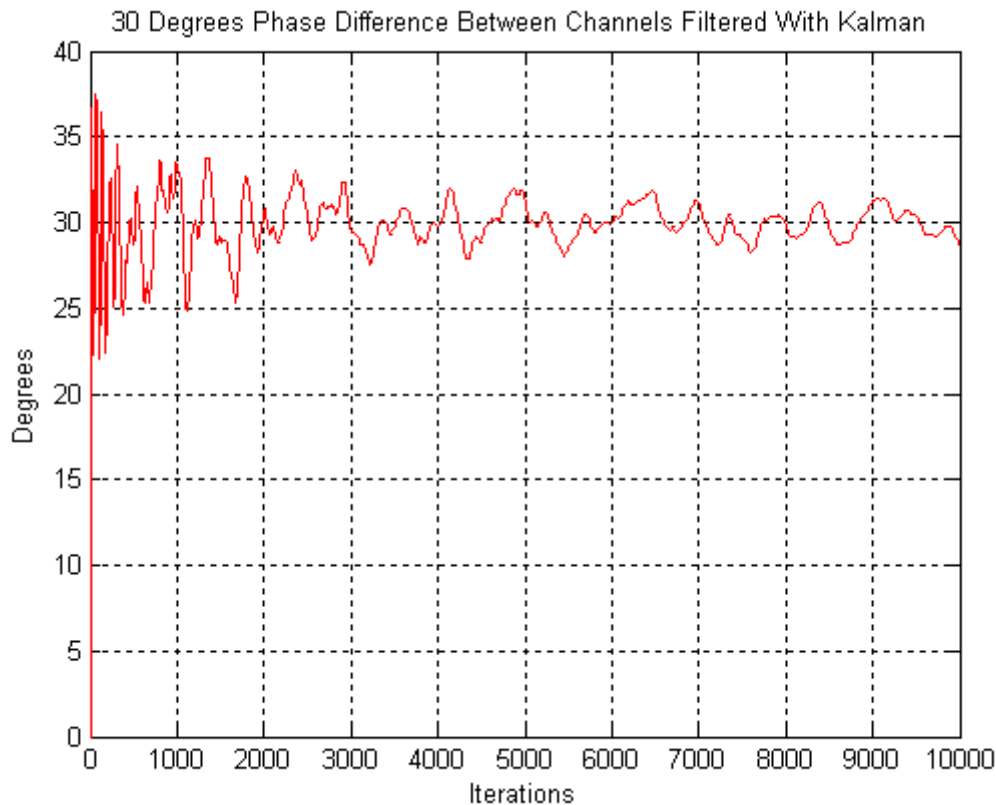


Figura 6-13: Resultados obtenidos mediante simulación del filtrado de fase con un filtro de Kalman, para un error de fase entre canales de 30 grados.

6.4.4 Análisis de los resultados y conclusiones

Desafortunadamente la primera implementación del filtro de Kalman dio lugar a nuevas desestabilizaciones de la fase estimada. Habiendo descartado todas las hipótesis que podían explicar este fenómeno, se consideró como posible causa del mismo la presencia en el bucle de algún tipo de no linealidad.

La única fuente posible de no linealidad es la inversa de la tangente empleada en el cálculo de la estimación de fase, por tanto se adelantó la posición del filtro en el bucle hasta justo antes de este cálculo. De esta forma, en vez de filtrar la estimación de fase, se filtran los coeficientes $\mu_{r,n}$ y $\mu_{i,n}$. Esta aproximación requiere dos filtros de Kalman (uno para cada coeficiente) en vez de uno.

La figura 6-14 muestra un diagrama sencillo de la solución finalmente implementada.

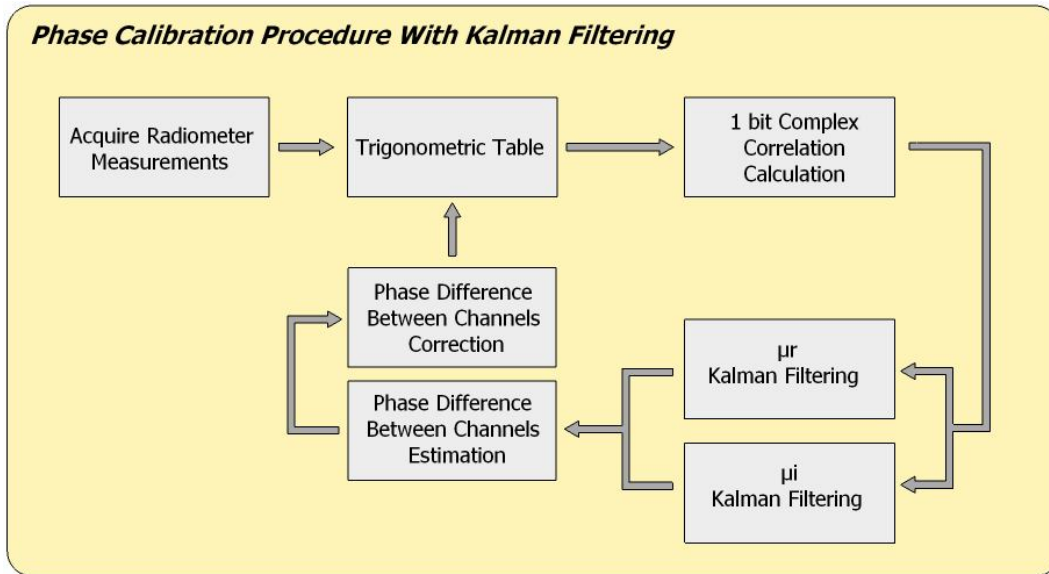


Figura 6-14: Implementación final en la unidad de control del filtrado de fase mediante filtros de Kalman.

Los resultados obtenidos de la implementación del filtro de Kalman en la unidad de control del radiómetro y su aplicación a medidas reales se muestran en la siguiente figura:

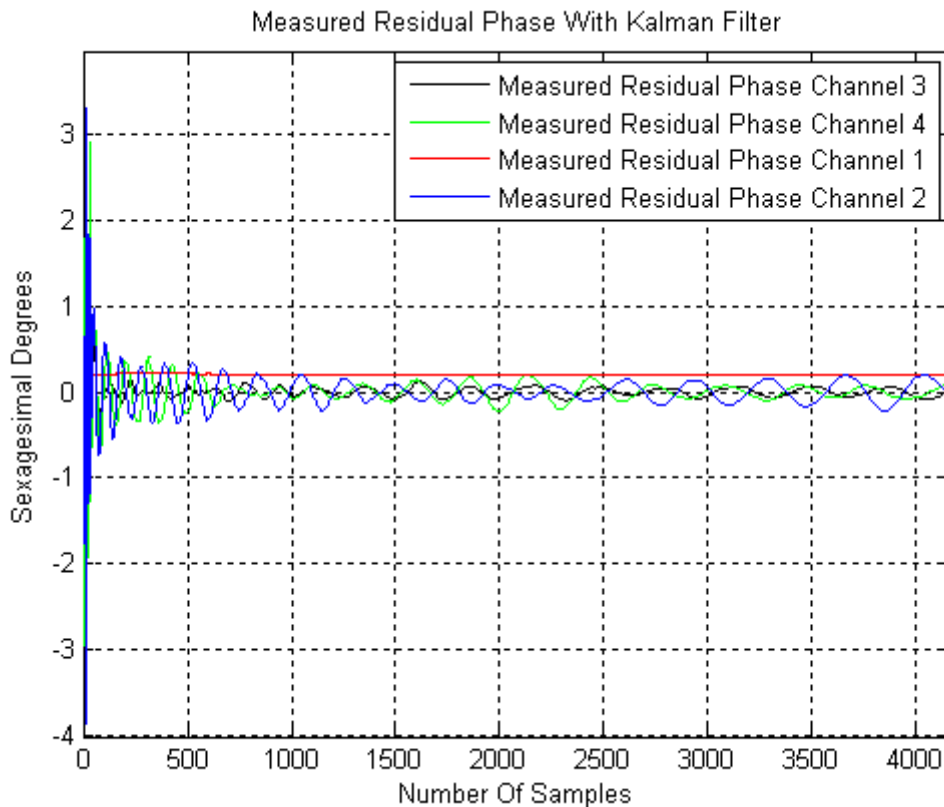


Figura 6-15: Resultados de la implementación del filtro de Kalman en la unidad de control del radiómetro.

Finalmente los resultados obtenidos de la implementación final si son satisfactorios. Como se puede ver en la figura, la fase corregida (fase residual) para los cuatro canales tiene una media de 0 grados (el error de fase entre canales ha sido totalmente corregido) y se mantiene completamente estable. Por otra parte, la fase residual exhibe una desviación típica muy inferior a la que presenta el sistema sin filtrado, incluso si se tienen en consideración las primeras muestras que contienen el transitorio del filtro. Esto último se puede apreciar con claridad en la figura 6-16:

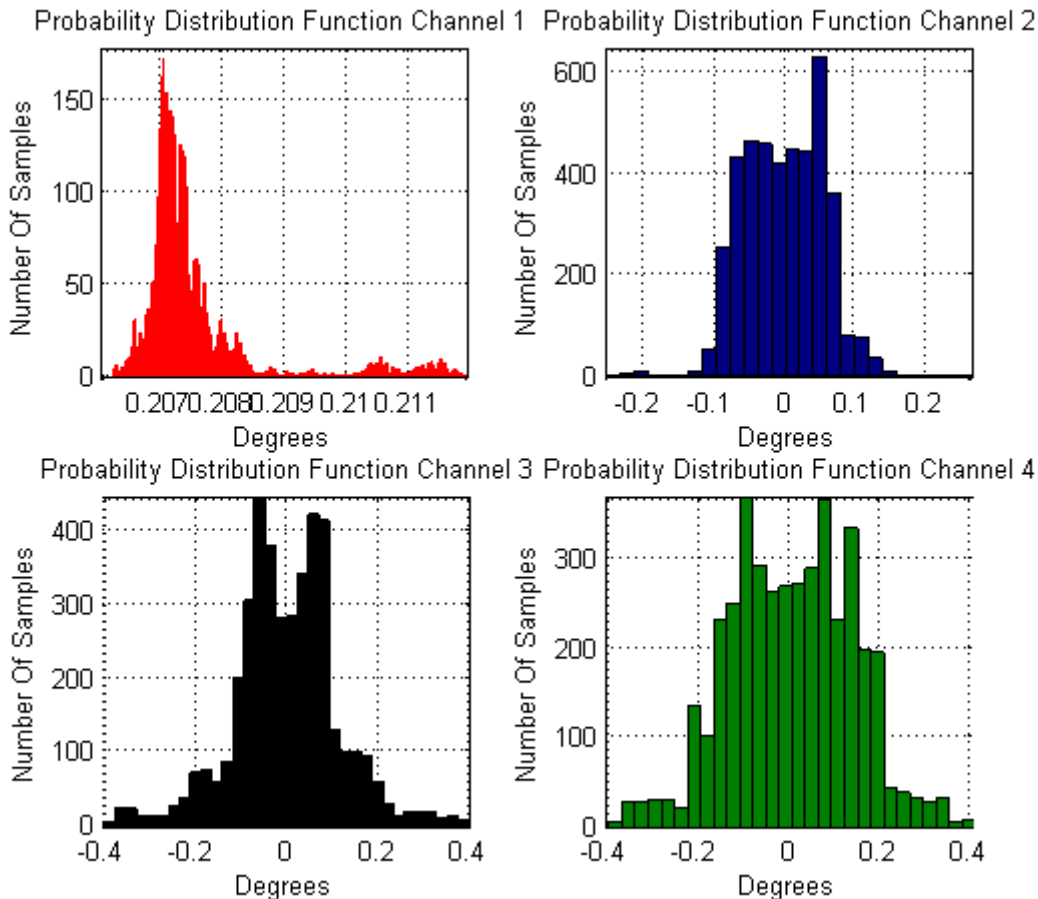


Figura 6-16: Funciones de distribución de probabilidad correspondientes a las fases residuales de los cuatro canales que se muestran en la figura 6-15.

Como se puede observar, las desviaciones típicas para los cuatro canales se han reducido considerablemente, de forma que las varianzas resultantes son aproximadamente: $\sigma_1 = 0.012^\circ$, $\sigma_2 = 0.189^\circ$, $\sigma_3 = 0.227^\circ$, $\sigma_4 = 0.278^\circ$.

6.4.5 Inconvenientes del filtrado Kalman

A pesar de los buenos resultados obtenidos, el filtro de Kalman tiene dos inconvenientes. En primer lugar la fase residual resultante del filtrado presenta cierta

oscilación que, a pesar de estar por debajo de los valores de variación sin filtrado, es en cierta manera antinatural. Por otra parte, la naturaleza del filtro de Kalman hace que su respuesta sea extremadamente lenta frente a variaciones bruscas del escenario de medida.

Para tratar de paliar estos inconvenientes se añadió a la salida del filtro una constante de atenuación que reduciría esta oscilación, reduciendo al mismo tiempo el tiempo del transitorio del filtro y el tiempo de recuperación frente a un cambio brusco en la entrada. La incorporación de esta constante de atenuación supone un aumento de la varianza final resultante, por tanto se hizo necesario hallar un buen compromiso entre los diversos factores.

Los cambios bruscos a la entrada del filtro se simularon desconectando la alimentación del receptor, forzando de esta manera que el oscilador local se reiniciase con una fase completamente diferente a la que tenía inicialmente. El tiempo transcurrido desde este “desenganche” de fase hasta que el filtro recupera el estado estable se ha denominado tiempo de recuperación.

Con el fin de determinar una constante de atenuación óptima, se llevaron a cabo varias pruebas cuyos resultados se ilustran en la siguiente tabla:

Constante De Atenuación	Desviación Típica Media De Los Cuatro Canales [grados]	Tiempo Del Transitorio	Tiempo De Recuperación
1	0.334	4.6 minutos	+20 minutos
0.98	0.352	2.3 minutos	7 minutos
0.95	0.392	1.9 minutos	2.3 minutos
0.9	0.412	42 segundos	1.6 minutos
0.88	0.506	35 segundos	1.2 minutos

Tabla 6-5: Desviación típica media de los cuatro canales y tiempos de transitorio y recuperación en función de la constante de atenuación.

Finalmente se optó por una constante de atenuación de 0.9, que ofrece un buen compromiso entre reducción de la desviación típica y tiempos de transitorio y recuperación. Los resultados obtenidos para este caso son los que se muestran en las figuras 6-17 y 6-18.

Como se puede comprobar en la figura 6-18, la oscilación ha sido prácticamente eliminada, y la fase residual para los cuatro canales se mantiene en su mayoría en la zona comprendida entre -0.5° y 0.5° . Podemos concluir por tanto que esta implementación en particular del filtro es satisfactoria y se cumplen los objetivos fijados inicialmente.

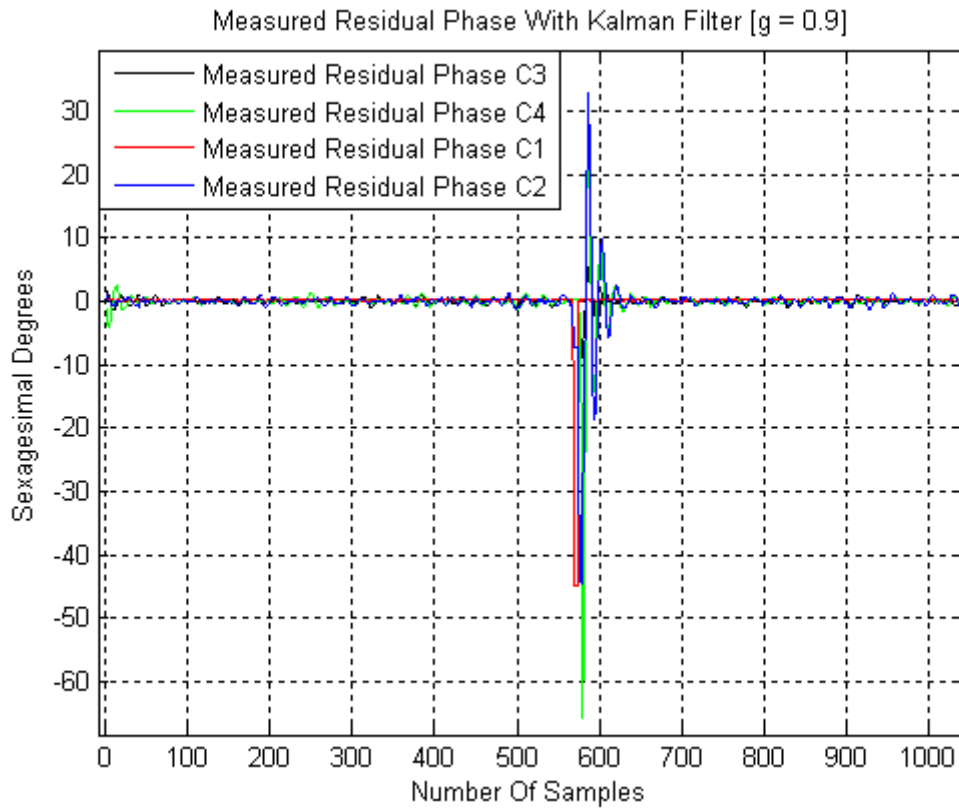


Figura 6-17: Resultados obtenidos para una constante de atenuación 0.9. La variación producida entre las muestras 550 y 650 es debida al desenganche de fase simulado.

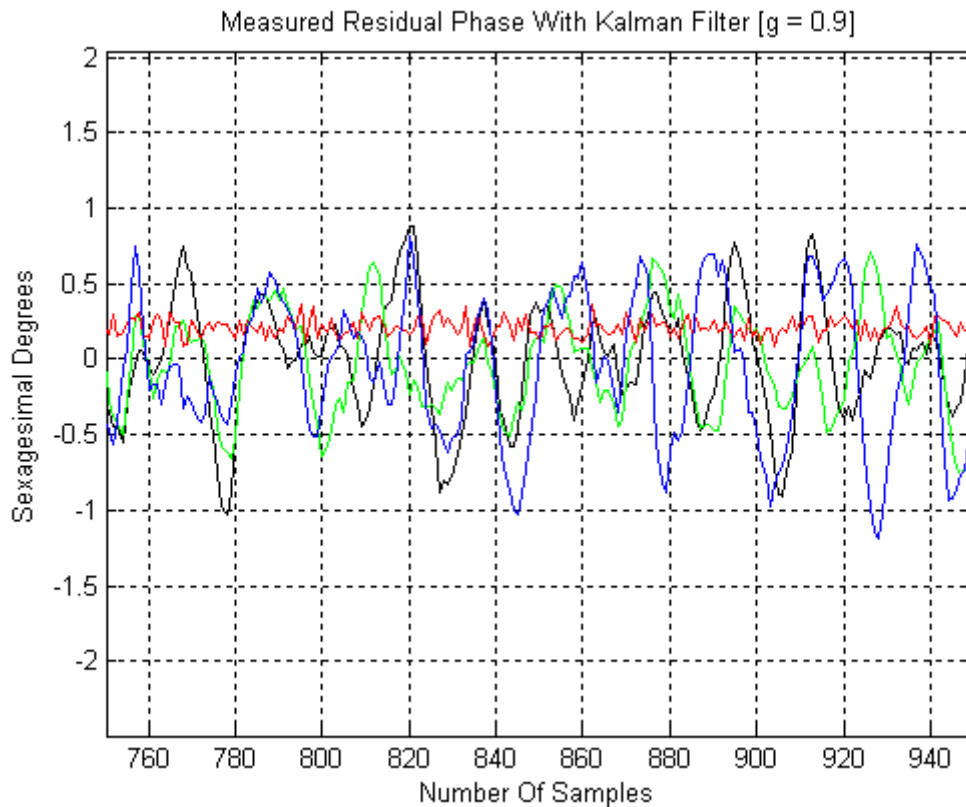


Figura 6-18: Ampliación de la zona comprendida entre las muestras 760 y 940 de la figura 6-17.

7 MIGRACIÓN DEL SISTEMA A STRATIX II

7.1 INTRODUCCIÓN

En este capítulo se abordará la descripción del proceso de migración del sistema PAU-RAD de la FPGA Stratix I a un nuevo modelo de FPGA Stratix II. Entendemos por migración la adaptación a la nueva FPGA de todos los diferentes subsistemas de PAU-RAD originalmente implementados para Stratix I.

A priori puede parecer extraño el hecho de migrar un sistema funcional a una nueva plataforma que, como se verá más adelante, no es compatible con la anterior. Sin embargo existe una poderosa razón por la cual la migración es inevitable. Tal y como ya se ha comentado anteriormente, la implementación completa de PAU-RAD con 16 receptores requiere de dos FPGA que trabajen de manera conjunta. Esto es debido tanto a la limitada capacidad de la FPGA (en términos de elementos lógicos) como al reducido número de pines de entrada/salida que hacen imposible conectar los 16 ADC en una única placa.

Al inicio del proyecto PAU-RAD se disponía de dos placas de evaluación Stratix I como la descrita en el capítulo 3. No obstante, una avería irreparable en una de ellas limitó el desarrollo a una única FPGA, reduciendo por tanto el número de receptores posibles del sistema. Altera, empresa que fabrica las FPGA, ya había descatalogado y retirado el soporte técnico de las FPGA de la familia Stratix I tiempo atrás, imposibilitando la compra de una nueva placa.

Dadas las circunstancias se optó por la compra de varias placas de evaluación de la familia Stratix II para poder completar el proyecto PAU tal y como fue concebido originalmente. Por este motivo, la migración de todo el sistema a la nueva plataforma que ofrece la familia Stratix II resulta inevitable.

El proceso de migración del que se hablará a lo largo de este capítulo, concierne únicamente a lo relacionado con el procesador Nios II (y por lo tanto a la UC), ya que la migración del resto de hardware digital fue realizada de forma paralela por Xavier Bosch en el marco de su tesis doctoral. Se comentarán los inconvenientes que se encontraron durante este proceso y las soluciones finalmente implementadas.

7.2 DISPOSITIVOS DE LA FAMILIA STRATIX II

Los dispositivos de la familia Stratix II constituyen la segunda generación de FPGA de Altera. A pesar de que en la actualidad sus prestaciones han sido superadas por nuevas generaciones, las FPGA Stratix II se mantienen en producción y siguen siendo recomendadas por Altera para determinados proyectos. Esto es debido a que es la única familia de dispositivos de Altera que ha sido completamente adaptada a especificaciones militares, lo que hace que sean ideales para sistemas que requieren fiabilidad, durabilidad y que están sujetos a grandes cambios de temperatura, como en el caso de PAU-RAD.

7.2.1 Placa de desarrollo Stratix II

Al igual que sucedía con Stratix I, la nueva FPGA se encuentra dentro de una placa de desarrollo que contiene todos los elementos hardware necesarios para su configuración.



Figura 7-1: Fotografía de la placa de desarrollo Stratix II empleada.

Entre los elementos hardware, disponibles en la nueva placa de desarrollo, cabe destacar:

- FPGA Stratix II modelo EP2S60F672C3N.
- 16 Mbytes de memoria flash, 2 Mbytes de memoria SRAM síncrona y 32 Mbytes de DDR SDRAM.
- Lógica necesaria para poder llevar a cabo la configuración de la FPGA desde la memoria flash.

- Controlador de acceso al medio Ethernet y conector RJ45.
- Puerto serie RS-232 con conector DB9.
- Conector JTAG de programación.
- Conector Mictor.

En esencia, se puede afirmar que, a excepción de la propia FPGA, los componentes de la nueva placa de desarrollo no distan demasiado de aquellos disponibles en la versión anterior.

7.2.2 Comparativa Stratix I Vs. Stratix II

Las FPGA Stratix II son dispositivos de tecnología de 90 nm, que incorporan una arquitectura de celdas lógicas completamente nueva. Gracias a esta nueva arquitectura, disponen de una mayor capacidad en términos de elementos lógicos y de mejores prestaciones en general.

A continuación se incluye una tabla que compara las principales características de ambas generaciones de dispositivos Stratix:

Tabla 7-1: Comparativa de características Stratix I Vs. Stratix II.

	Dispositivos Stratix I EP1S10	Dispositivos Stratix II EP2S60
Elementos Lógicos	10.570	60.440
Bloques De Memoria M4K	60	255
Bits De RAM Totales	920.448	2.544.192
Bloques Multiplicadores	48	144
PLLs	6	12
Pines I/O De Usuario	426	718

Como se puede comprobar, los dispositivos de la familia Stratix II superan ampliamente en todos los aspectos a sus predecesores. En particular, el incremento en el número de elementos lógicos disponibles hace que la nueva FPGA disponga de capacidad para albergar proyectos de mayor envergadura y complejidad.

Si se tiene en cuenta que el número de elementos lógicos que requiere el sistema PAU-RAD al completo es de 11081, queda claro que la nueva FPGA dispone, a diferencia de lo que ocurría con la anterior, de capacidad suficiente para dar cabida al proyecto completo. No obstante sigue siendo necesaria la utilización de dos placas de desarrollo, ya que, si bien el número de pines de entrada/salida de la FPGA ha aumentado, el número de pines de la placa de desarrollo se mantiene en 82, igual que en la versión anterior.

7.3 PROBLEMÁTICA DE LA MIGRACIÓN

7.3.1 Incompatibilidad entre sistemas

La dificultad de la migración al nuevo sistema viene determinada por la incompatibilidad entre los dispositivos Stratix I y Stratix II. A priori, la solución más sencilla pasaría por sintetizar el procesador Nios I en la nueva FPGA. No obstante a causa de las diferencias entre ambos dispositivos esta solución no es posible.

Los dispositivos de la familia Stratix II requieren el uso de una nueva versión de procesador embebido, el Nios II. Ya que la nueva FPGA no admite la versión anterior del procesador, el primer paso de la migración consistirá por tanto en realizar las adaptaciones necesarias al código de la UC para permitir su ejecución en el procesador Nios II.

7.3.1.1 El procesador Nios II

Antes de abordar las modificaciones que es necesario realizar en la UC, se introducirán de forma breve algunas de las principales características del procesador Nios II.

El Nios II es, al igual que su predecesor, un procesador de propósito general de arquitectura RISC. Se trata de un procesador de 32 bits que dispone de 32 registros de propósito general, 32 fuentes de interrupción externas y capacidad de realizar multiplicaciones y divisiones de 32 x 32 bits en una única instrucción dando como resultado un entero de 32 bits. Dispone además acceso a una gran variedad de periféricos "on-chip" así como de interfaces para acceder a los periféricos de la placa de desarrollo Stratix II.

7.3.1.2 Adaptación de la UC original a Nios II

La implementación de software para Nios II requiere la utilización de un nuevo entorno de desarrollo denominado Nios II IDE (Integrated Development Environment). Esto se debe a que, a diferencia de lo que ocurría con el Nios I, los componentes de memoria del procesador Nios II no se generan mediante el SOPC Builder (ver capítulo 3), sino que se generan directamente en el entorno de desarrollo al realizar la compilación.

Los proyectos generados para Nios I mediante el anterior entorno de desarrollo tienen una estructura de archivos muy distinta a la de los proyectos generados mediante Nios II IDE. Esto hace que no sea posible compilar el proyecto de la UC original en el nuevo entorno. En lugar de ello, es necesario crear un nuevo proyecto y copiar directamente el código de los archivos del proyecto anterior para poder comenzar las modificaciones.

Tal y como se mencionó en el capítulo 4, el procesador Nios I dispone de un kit de desarrollo de software conocido como Legacy SDK (Software Development Kit) que incluye diversas herramientas para facilitar su programación. Dentro de este kit se incluye una serie de librerías que permiten la utilización de funciones específicas para el procesador, tales como operaciones de lectura y escritura en los registros. Estas librerías no se incluyen en Nios II y no es posible importarlas de ninguna manera al nuevo sistema. Por tanto, el primer paso de la adaptación consiste en buscar archivo por archivo, línea por línea, todas las llamadas a funciones de las librerías Legacy SDK en el código de la UC original y sustituirlas por sus equivalentes para Nios II.

Otro de los aspectos que diferencian ambos procesadores es que los archivos que contienen la descripción del hardware del sistema (los periféricos de la placa a los cuales tiene acceso el procesador) tienen estructuras completamente diferentes. Por tanto, será necesario modificar las descripciones del hardware del código de la UC original para adaptarlas a las empleadas en el sistema Nios II. En particular será necesario modificar la descripción de todos los registros PIO (Parallel Input Output), elementos clave del sistema, ya que se utilizan para múltiples tareas de comunicación entre la ALU y la UC.

Teniendo en cuenta todo lo anterior, resulta obvio que la tarea de adaptación del código de la UC al nuevo sistema, no solo resulta larga y compleja, sino que además tiene altas probabilidades de fracasar en la etapa de compilación debido a las profundas diferencias entre ambos sistemas. Por tanto finalmente se adoptó la opción que parecía más razonable: crear una nueva UC desde cero, utilizando como referencia la arquitectura de la UC anterior.

7.3.1.3 Diferencias en los mecanismos de acceso al hardware del sistema

A pesar de las múltiples diferencias entre ambos procesadores, el principal obstáculo a la hora de desarrollar una nueva UC, se encuentra en el mecanismo de acceso al hardware de ambos procesadores.

La figura 7-1 muestra un esquema con las diferentes capas que forman el mecanismo de acceso al hardware para el sistema Nios I:

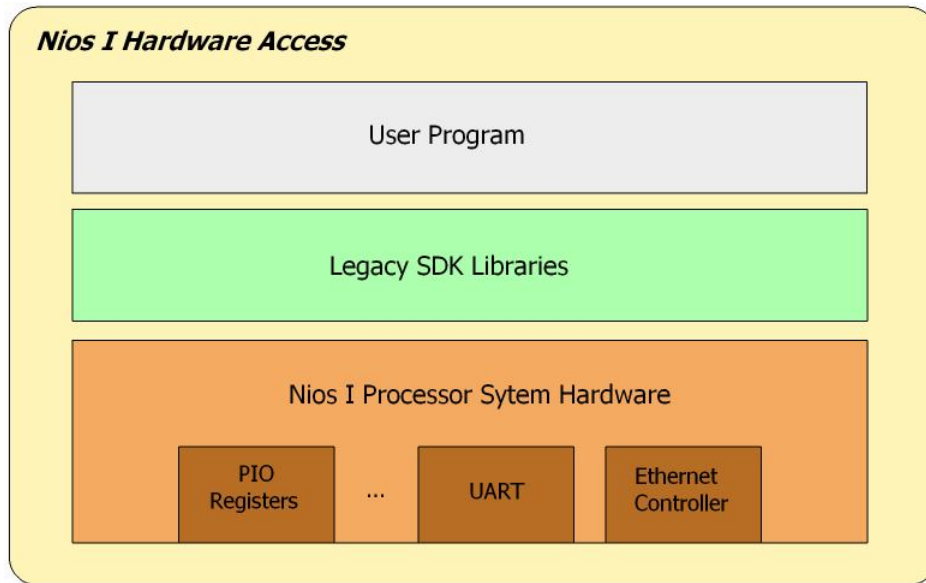


Figura 7-2: Mecanismo de acceso al hardware del sistema Nios I.

Como se puede observar, el acceso al hardware del Nios I es relativamente sencillo y consta únicamente de tres niveles. En el nivel superior se ubica el software de usuario que corre en el procesador (en este caso la UC). El segundo nivel lo ocupa el conjunto de librerías Legacy SDK. Estas librerías contienen las rutinas necesarias para acceder directamente al tercer nivel, compuesto por los diferentes dispositivos hardware disponibles en la placa de desarrollo (los registros PIO, la UART, el chipset LAN91C111 que controla el puerto Ethernet, los displays de 7 segmentos,...).

En particular, el kit de desarrollo de software contiene la librería *plugs_library*, que dispone de las rutinas necesarias para establecer una conexión TCP/IP. Esta librería incluye las funciones: *nr_plugs_send()* y *nr_plugs_listen()* que permiten enviar datos a la interfaz de captura y monitorizar los datos entrantes de una forma sencilla y eficaz.

No obstante, este mecanismo de acceso a los recursos hardware de la placa de desarrollo no se mantiene en los dispositivos de la familia Stratix II. En la figura 7-2 se ilustra el nuevo mecanismo de acceso al hardware implementado en el sistema Nios II.

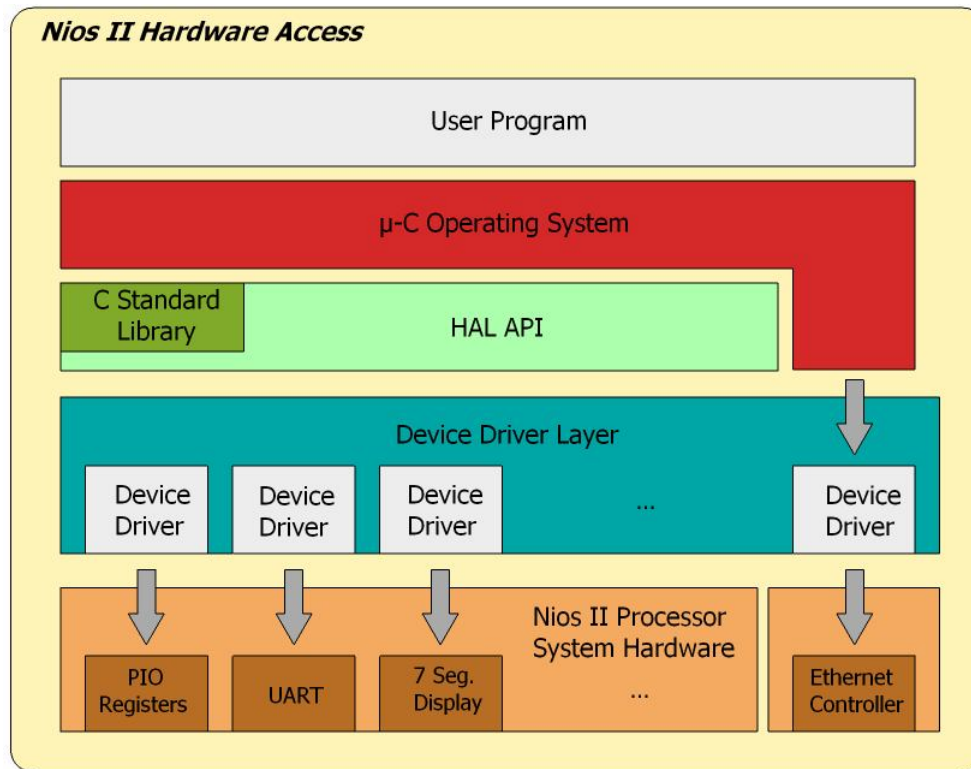


Figura 7-3: Mecanismo de acceso al hardware del sistema Nios II.

El sistema Nios II posee un mecanismo de acceso al hardware más complicado que su predecesor. Se incluyen dos capas nuevas y una de las ya existentes es sustituida por otra con el mismo propósito, pero completamente diferente.

De nuevo, en el nivel superior se encuentra la aplicación de usuario (la UC). En esta ocasión el segundo nivel lo ocupa la capa del sistema operativo μ -C OS (Operating System), del cual se hablará más adelante. El siguiente nivel es el equivalente al segundo nivel de la estructura de acceso al hardware del sistema Nios I. En el sistema Nios II las librerías Legacy SDK se sustituyen por la capa HAL API (Hardware Abstraction Layer Application Programming Interface). La capa HAL es la encargada de proporcionar al procesador Nios II las diferentes rutinas de acceso al hardware de la placa de desarrollo. Presenta la ventaja añadida de incorporar una serie de librerías estándar de C de uso muy extendido, que no son compatibles con el sistema Nios I.

El cuarto nivel está formado por la capa de drivers de los dispositivos. Los drivers de los dispositivos son un conjunto funciones que, mediante el uso de macros de bajo nivel, se encargan de hacer que los dispositivos hardware sean accesibles a la aplicación de usuario. De esta forma, cada dispositivo de la placa de desarrollo dispone de un driver asociado que permite a la aplicación de usuario interactuar con él.

El nivel correspondiente al sistema operativo es opcional. Durante el proceso de creación de un nuevo proyecto Nios II es posible especificar que la aplicación que se va a desarrollar no requiera sistema operativo. No obstante, tal y como queda patente

en el esquema de la figura 7-2, el driver que gobierna el controlador del puerto Ethernet de la placa únicamente es accesible a través del sistema operativo μ -C OS.

7.3.2 Soluciones consideradas

Por tanto no es posible sintetizar el procesador Nios I en las FPGA de la familia Stratix II. Asimismo, el proceso de adaptación a Nios II del software desarrollado para Nios I no es viable. Como ya se ha mencionado, la mejor opción disponible pasa por la creación de una nueva UC que, siguiendo la arquitectura de la UC original, se adapte al nuevo procesador. No obstante las diferencias entre los mecanismos de acceso al hardware de ambos procesadores hacen que sea extremadamente complicado respetar la arquitectura original de la UC.

El punto crítico de la nueva implementación se encuentra en el sistema de transmisión de los datos procesados a la interfaz de captura de datos, que en la UC original se llevaba a cabo mediante el puerto Ethernet de la palca de desarrollo. A continuación se analizarán las tres soluciones consideradas, detallando los inconvenientes que supone la implementación de cada una de ellas.

7.3.2.1 Implementación mediante μ -C OS

El sistema operativo μ -C, es un kernel (núcleo o parte fundamental de un sistema operativo) multitarea de ejecución en tiempo real desarrollado por Micrium Inc. Desde su lanzamiento en 1992 se ha convertido en la base de cientos de aplicaciones comerciales implementadas en más de 40 procesadores diferentes.

Tal y como se ha ilustrado en la figura 7-2, μ -C opera justo por encima de la capa de abstracción de hardware (HAL). La finalidad de esta arquitectura es la de mantener la portabilidad a otros sistemas y proteger las aplicaciones frente a posibles cambios en el hardware. Los servicios básicos que proporciona este sistema operativo son:

- Múltiples Hilos o tareas.
- Flags de señalización de eventos.
- Paso de mensajes.
- Gestión de memoria.
- Implementación de semáforos.
- Gestión de tiempo.

La arquitectura de la UC original (descrita en el capítulo 3) se basa en el establecimiento de una conexión TCP/IP con la interfaz de captura de datos como medio de transmisión de información. Tal y como se ha descrito en el apartado anterior, el driver que gobierna el controlador del puerto Ethernet, requiere la utilización del sistema operativo μ -C. Por este motivo, la primera opción planteada a la hora de iniciar la implementación de la nueva UC pasa por el uso de dicho sistema operativo.

No obstante la naturaleza multitarea del sistema operativo y el hecho de tratarse de un sistema propietario desarrollado por una compañía privada, marcan dos claros inconvenientes para esta posible solución:

- Al tratarse de un sistema operativo multitarea, el código de la nueva UC ha de presentar una estructura concurrente, dividida en múltiples hilos que se ejecutan de forma simultánea (tan sólo el funcionamiento del controlador Ethernet ya requiere tres hilos diferentes). Esto implica modificar toda la arquitectura de la UC original, pasando de una arquitectura de subrutinas que representan los estados del sistema y que se ejecutan en un único proceso, a una estructura compuesta por múltiples procesos (un proceso por cada estado del sistema) corriendo de forma simultánea.

La adaptación de la arquitectura inicial a una nueva arquitectura concurrente es compleja, ya que todo el sistema se basa en una serie de variables globales a las que accede cada uno de los estados del sistema. Sin embargo este mecanismo no es recomendable en un sistema concurrente, ya que el hecho de disponer diferentes hilos accediendo a un único recurso (por ejemplo, una variable) de forma simultánea puede acarrear problemas con el resultado final o incluso llegar a bloquear la ejecución.

Por tanto, la adopción de μ -C en la nueva UC, requiere profundas modificaciones al concepto de la UC original. Esto choca con el principal requisito de la nueva implementación: mantener la arquitectura ya creada para la UC, algo necesario si se tiene en cuenta que modificarla supondría modificar también la arquitectura de la ALU.

- Puesto que es un sistema operativo desarrollado por una empresa privada, es necesario pagar una licencia para poder desarrollar aplicaciones basadas en μ -C. A pesar de que Micrium dispone de licencias gratuitas para universidades, sin duda este es un factor a tener en cuenta de cara a futuras aplicaciones del sistema.

Al margen de estos dos inconvenientes hay que tener en consideración el tiempo y esfuerzo que requiere aprender a programar en un sistema operativo

desconocido. Finalmente, tras varios intentos de adoptar el sistema operativo, esta solución fue descartada por los motivos expuestos.

7.3.2.2 Implementación mediante LWIP

Una vez quedó claro que la adopción de μ -C como sistema operativo añadía demasiadas complicaciones al proceso de desarrollo, se optó por buscar una solución alternativa. Los requisitos que se plantearon para dicha solución fueron dos: que fuese independiente de cualquier sistema operativo y que permitiese el uso del puerto Ethernet para la transmisión de información a la interfaz de captura. La única alternativa disponible que cumple sendos requisitos es el protocolo LWIP (Light Weight Internet Protocol).

El protocolo LWIP es una implementación de la pila de protocolo TCP/IP. Se trata de un protocolo de bajo peso, orientado a reducir el uso de memoria y el tamaño del código, lo que hace que sea ideal para sistemas con recursos limitados como los procesadores embebidos. Originalmente, LWIP, fue creado en el SICS (Swedish Institute of Computer Science), aunque en la actualidad constituye un proyecto abierto en el que trabajan diferentes desarrolladores de todo el mundo. Se trata por tanto de un software abierto, de libre acceso.

LWIP no requiere de un sistema operativo en tiempo real gracias que dispone de un driver de dispositivo propio para el controlador Ethernet. De forma que, para su utilización, es necesario sustituir el componente de la capa HAL correspondiente al controlador LAN91C111 original de Altera, por el que se incluye en el protocolo.

Esta solución presenta dos inconvenientes fundamentales:

- Se trata de un protocolo desarrollado por una entidad ajena a Altera y que sustituye los componentes originales de las librerías de sistema del Nios II. Debido a ello, el entorno de desarrollo Nios II IDE detecta frecuentes errores de compatibilidad (a menudo se trata de errores que no se pueden identificar), dificultando la compilación de los proyectos.
- La documentación disponible es muy limitada. El único documento del que se dispone se centra en una descripción general del funcionamiento del protocolo y presenta unos ejemplos de implementación de comunicaciones que emplean sentencias que no se corresponden con la versión disponible para Nios II. Debido a esto, la principal fuente de información empleada durante el estudio de esta solución fueron los comentarios que se encuentran directamente en el propio código del protocolo.

A pesar de estos inconvenientes se invirtió mucho tiempo en esta solución, al considerarse la única que hacía viable mantener la arquitectura de la UC original.

Teóricamente, la implementación de LWIP descrita en la documentación, dispone de una API (Application Programming Interface) que facilita enormemente el establecimiento de una conexión TCP/IP y la transmisión de información. No obstante no fue posible utilizar esta API en el sistema Nios II. En su lugar, se utilizaron directamente las macros de bajo nivel del protocolo. La utilización de los recursos de bajo nivel de un protocolo requiere amplios conocimientos en materia de programación, especialmente cuando no se dispone de la documentación apropiada.

Tras numerosos intentos se consiguió establecer una conexión y realizar la transmisión de información en ambos sentidos de manera satisfactoria. No obstante, no se consiguió establecer una comunicación full dúplex entre la UC y la interfaz de captura. Por el contrario, una transmisión desde la interfaz a la UC tan sólo podía darse como respuesta a la llegada de un paquete desde la UC, es decir una conexión half dúplex.

Finalmente, debido a los múltiples inconvenientes encontrados durante el desarrollo de esta solución, se optó por descartarla en favor de otra que adoptase un sistema de transmisión alternativo.

7.3.2.3 Transmisión de datos a través de la UART

Dado que el establecimiento de una conexión TCP/IP como medio de transmisión de información representaba efectuar profundas modificaciones a la arquitectura del sistema completo, se optó por emplear otro medio de transmisión de información.

La única alternativa al uso del puerto Ethernet como medio de comunicación es la UART (Universal Asynchronous Receiver-Transmitter) disponible en la placa de desarrollo. Gracias a ella es posible establecer comunicación con la interfaz de captura de datos mediante el puerto serie del ordenador.

La principal ventaja de esta solución es que el sistema Nios II permite acceso a la UART directamente a través de funciones de la librería C estándar incluida en la capa de abstracción de hardware HAL (ver figura 7-2), lo que simplifica enormemente el proceso de desarrollo.

La contrapartida a esta ventaja es que el hecho de modificar el mecanismo de transmisión de información supone también modificar por completo el código de la interfaz de captura de datos. A pesar de este inconveniente, finalmente se optó por

esta solución al considerarse más sencillo rehacer la interfaz de captura de datos que modificar la arquitectura de la UC creada inicialmente.

La UART de la FPGA tiene capacidad para trabajar con velocidades de transferencia que van desde los 2400 hasta los 115200 baudios (o símbolos) por segundo. Puesto que cada símbolo tiene una longitud de 8 bits, la velocidad mínima de la UART es de 19200 bits por segundo. Teniendo en cuenta el tiempo de integración del sistema y todo el proceso de captura, procesado y promediado de los datos, incluso la velocidad mínima de la UART está por encima de los requerimientos del sistema. Por tanto, la velocidad de transmisión no supone un inconveniente a la hora de adoptar la UART como medio de transmisión de información.

7.4 SOLUCIÓN IMPLEMENTADA

7.4.1 Nueva Unidad de Control

7.4.1.1 Descripción

La implementación de la UC para el procesador Nios II realiza las mismas funciones que se describieron en el capítulo 4 para la versión anterior. Se trata por tanto de un conjunto de subrutinas que tiene por misión actuar como intermediario entre la UC y la interfaz de captura de datos.

La nueva UC conserva el mismo diagrama de estados que la anterior así como las mismas secuencias de ejecución y de validación. En este apartado únicamente se detallarán los cambios más significativos con respecto a la UC original. Para una descripción detallada de la UC, consultar el capítulo 4.

7.4.1.2 Cambios en la arquitectura

Uno de los requisitos que dan lugar a esta implementación es el de mantener constante la arquitectura original de la UC con el fin de evitar cambios en el resto de componentes del sistema, especialmente la ALU. Por tanto la nueva UC comparte la misma arquitectura descrita en el capítulo 4.

Las diferencias más significativas se encuentran en la codificación en sí, que se adapta a los requerimientos del nuevo entorno de desarrollo, en el mecanismo de

sincronización con la UC y algunos cambios en la rutina principal derivados del nuevo mecanismo de transmisión.

Los cambios en la rutina principal de la UC (que en esta versión se encuentra en el archivo *PAU.RAD.Control.Unit.c*) afectan al bucle que controla la transición entre estados. La figura 7-3 muestra un esquema que ilustra el funcionamiento de dicho bucle:

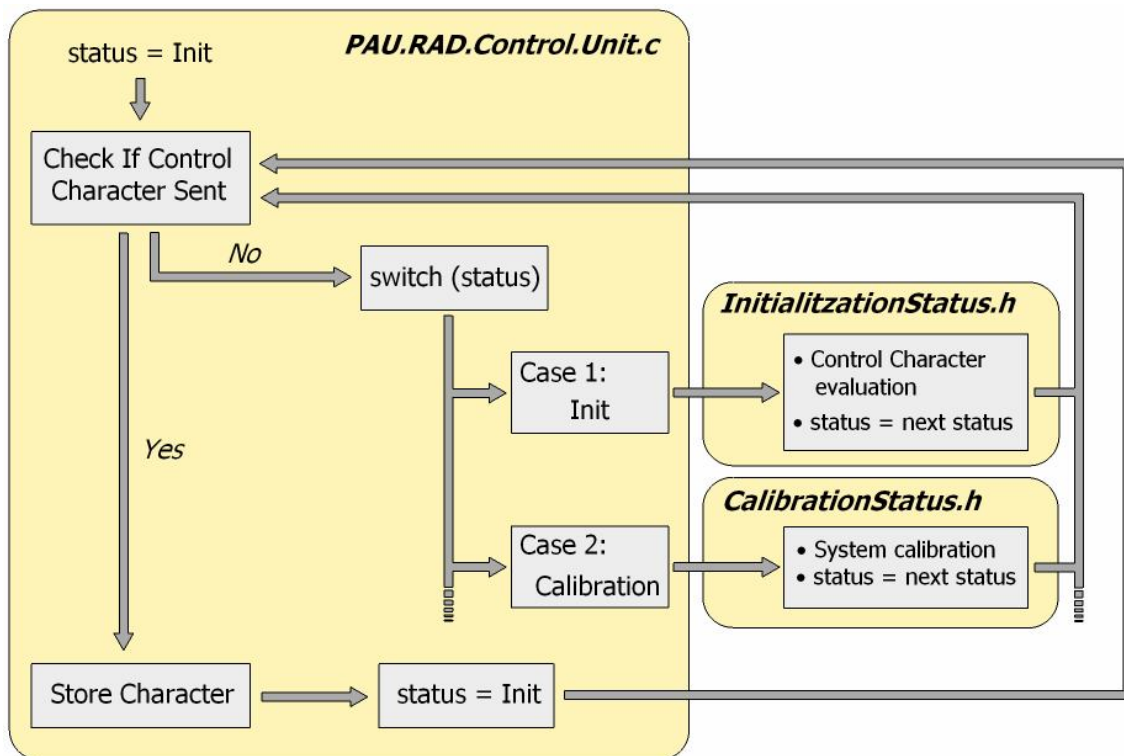


Figura 7-4: Bucle de control de estados de la rutina principal de la UC.

En esta nueva versión de la UC es necesario comprobar a cada iteración del bucle principal si la interfaz de captura de datos ha enviado un carácter de control, antes de evaluar el siguiente estado de la secuencia de ejecución.

Este cambio en la arquitectura se debe a un cambio en la gestión de las rutinas de atención de interrupción. La UART dispone de una rutina de atención a la interrupción predefinida, que puede ser reemplazada por una propia creada ex profeso para esta aplicación. No obstante, el procesador Nios II no admite la utilización de las librerías C estándar dentro de una RSI. Por tanto no es posible efectuar una lectura o escritura en la UART en el mismo instante en que se produce una interrupción tal y como se hacía en la UC anterior.

Para solucionar este inconveniente, en lugar de crear una RSI propia, se optó por emplear la RSI por defecto. Esto hace que sea necesario realizar una comprobación periódica del buffer de entrada de la UART para detectar si se ha enviado un carácter

de control desde la interfaz. Así pues, antes de evaluar la variable de estado, se lleva a cabo una lectura del buffer de entrada de la UART. En caso de que se haya recibido un carácter, este es almacenado y la variable de estado actualizada de forma que el siguiente estado en ejecutarse será InIt, estado en el que se evalúa el valor del carácter y se llevan a cabo las acciones correspondientes.

El segundo cambio importante en la arquitectura afecta al mecanismo de comunicación entre la UC y la interfaz de captura de datos. Debido a la utilización del puerto serie como medio de transmisión, se hace necesario el uso de un protocolo de que permita separar las tramas de datos en función del tipo de información que transportan y a que canal corresponde la información. De este proceso se encarga la subrutina *SendSingleChannel()*, que se encuentra dentro de la librería *CommunicationFunctions()*.

La subrutina *SendSingleChannel()* requiere tres parámetros de entrada: un entero que identifica el canal al cual pertenecen los datos, un carácter que define el tipo de información que se transite y un entero que representa el valor a enviar. Con estos tres parámetros se conforma una trama que presenta el siguiente aspecto:

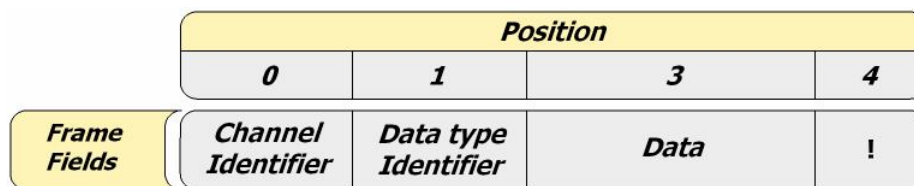


Figura 7-5: Formato de la trama empleada en la comunicación entre la UC y la interfaz de captura de datos.

El identificador de canal es un carácter que puede adoptar los valores: A, B, C, D en función del canal de origen de los datos. El identificador de tipo de datos puede adoptar los valores: F, P, O, B, L, C, S según el estado en el que se encuentre el sistema. Finalmente, la última posición de la trama contiene el carácter: !, que se emplea como identificador de final de trama. La figura 7-5 resume los posibles valores de los diferentes identificadores:

<i>Channel Identifier Value</i>		<i>Data Type Identifier Value</i>	
Channel 1	A	Calibration Phase	F
Channel 2	B	Calibration Power	P
Channel 3	C	Calibration Offset	O
Channel 4	D	Phase Sweep	B
		Open Loop	L
		Closed Loop	C
		Stokes Parameters	S

Figura 7-6: Valores que pueden adoptar los caracteres de identificación de canal y tipo de dato.

El resto de cambios que la nueva UC presenta con respecto a la anterior, son únicamente de codificación y no afectan al funcionamiento global de la UC ya descrito en el capítulo 4. Entre estos cambios cabe destacar el agrupamiento de las acciones más comunes en forma de funciones (como por ejemplo las acciones de comunicación, inicialización o adquisición de datos del radiómetro), lo que añade claridad y modularidad al código facilitando futuras expansiones.

7.4.2 Nueva Interfaz de captura de datos

Tal y como se ha mencionado, el hecho de sustituir la conexión TCP/IP por el puerto serie del ordenador como medio de comunicación implica realizar profundas modificaciones en la interfaz de captura de datos.

Debido a la complejidad que supone la tarea de modificar el código de la interfaz de captura de datos existente, se optó por la creación de una nueva interfaz. Inicialmente se consideró seguir empleando Matlab como plataforma de desarrollo para la nueva interfaz. No obstante, debido a los múltiples inconvenientes que presenta la gestión del puerto serie mediante Matlab (fundamentalmente frecuentes bloqueos de la aplicación), finalmente se desestimó esta opción en favor del uso de Visual Studio como plataforma de desarrollo y C# como lenguaje de programación.

7.4.2.1 Descripción

La nueva interfaz de captura de datos ejerce exactamente las mismas funciones que la anterior: actuar de intermediario entre el radiómetro y el usuario y recoger y presentar de forma gráfica la información procedente del mismo.

Esta nueva implementación se ha desarrollado en lenguaje C# y utiliza las herramientas de que dispone Visual Studio para la creación de aplicaciones Windows.

Se trata por tanto de una aplicación propia, independiente de cualquier otro software y cuyo único requerimiento es un sistema operativo Windows que disponga de la versión 2.0 de Microsoft Framework (de descarga gratuita en la web de Microsoft y presente en todo Windows actualizado). Esta nueva aproximación presenta las siguientes ventajas:

- No depende de una aplicación externa. A diferencia de la versión anterior que requería una copia registrada de Matlab, la nueva interfaz de captura es una aplicación independiente, cuyo único requisito es un sistema operativo Windows.
- Consume menos recursos del sistema. Al no depender de otras aplicaciones el consumo de recursos es menor. De esta forma se eliminan los bloqueos ocasionales que sufría la versión anterior y se mejora la velocidad en sistemas con pocos recursos.
- Presenta funcionalidades añadidas de gran utilidad. Puesto que accede directamente a los recursos del sistema operativo, proporciona más opciones que Matlab. Algunas de estas funcionalidades son: posibilidad de hacer zoom sobre una zona concreta de los gráficos para una mejor inspección, capacidad para guardar una captura de pantalla o bien imprimirla directamente, u obtener una lectura de los valores numéricos colocando el cursor encima de un punto. Todo esto sin necesidad de detener el proceso de captura de datos y sin perder muestras.

Estas nuevas funcionalidades están destinadas a mejorar la interacción con el usuario y facilitar la interpretación de los datos obtenidos.

La figura 7-6 muestra el aspecto que presenta la nueva interfaz de captura de datos:

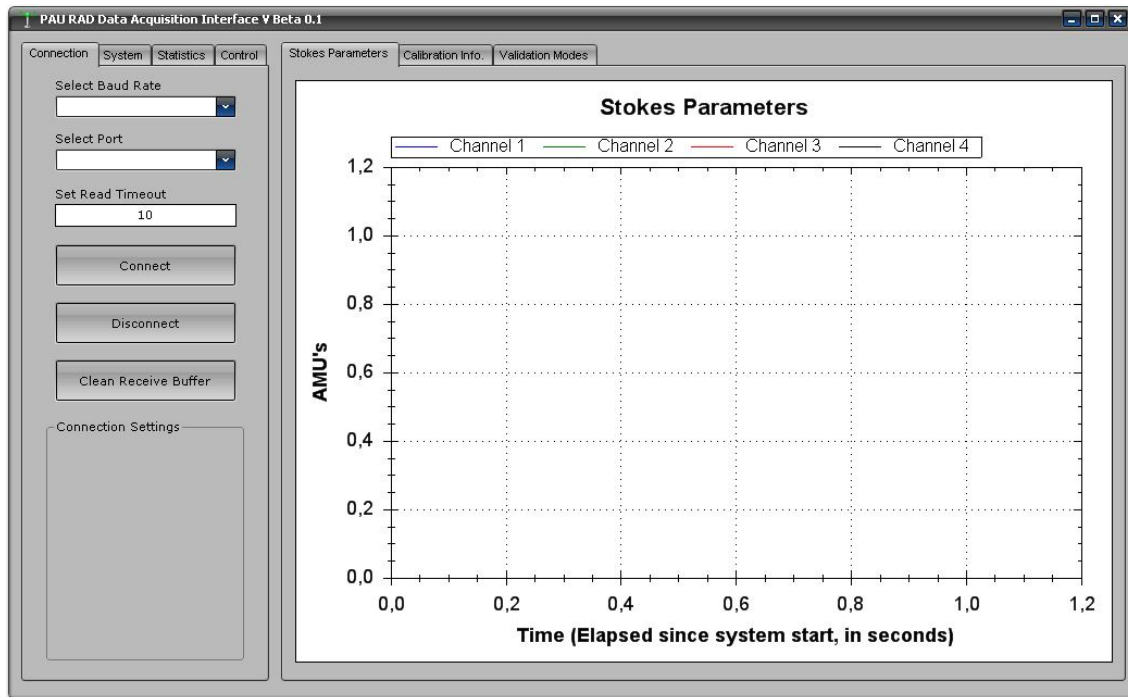


Figura 7-7: Pantalla inicial de la interfaz de captura de datos, antes de establecer comunicación.

La aplicación de la nueva interfaz se divide en dos zonas: la zona de control (a la izquierda de la imagen) y la zona de representación. La zona de control se compone de cuatro pestañas diferentes: *Connection*, *System*, *Statistics* y *Control*.

La pestaña *Connection* agrupa los elementos relacionados con el establecimiento de la conexión con la UC. Dispone de dos listas desplegables: la primera permite seleccionar la velocidad de transmisión, y la segunda presenta una relación de todos los puertos serie disponibles en la máquina para que el usuario elija a través de cual quiere establecer la comunicación. Contiene también un cuadro de texto que permite establecer el tiempo de espera de las operaciones de lectura, así como tres botones: uno para establecer la conexión, otro para la desconexión y un tercero que elimina los valores que pueda haber almacenados en el buffer de entrada.

Una vez establecida la conexión, muestra los principales parámetros de la misma en el cuadro inferior.

La segunda pestaña, *System*, contiene controles para realizar pruebas de envío y recepción de datos, un cuadro texto que muestra los mensajes del sistema (avisa del establecimiento de una conexión, fallo de la recepción...) y un botón que permite reiniciar el sistema a su estado inicial, es decir: eliminar el contenido de todos los vectores de datos, eliminar las trazas dibujadas en los paneles gráficos y reiniciar el contador de tiempo del sistema.

La figura 7-7 muestra en detalle los componentes de estas dos pestañas de la zona de control.

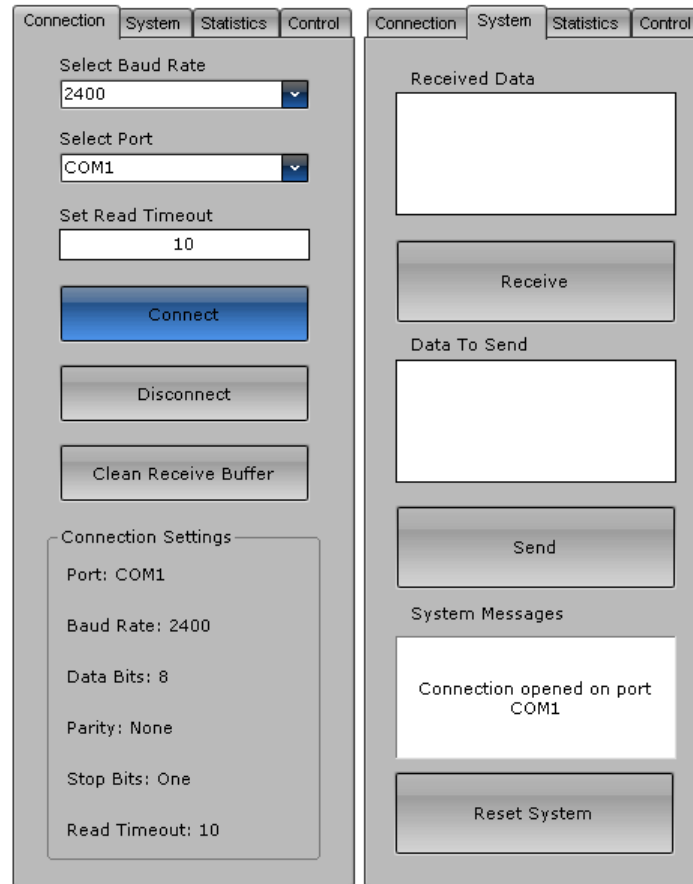


Figura 7-8: Detalle de las pestañas *Connection* y *System* de la zona de control de la interfaz.

La siguiente pestaña, *Statistics*, proporciona al usuario, en tiempo real, la media y la desviación típica de cada uno de los cuatro canales para los parámetros de Stokes y la fase y potencia de calibración. La información se refresca cada 20 muestras, no obstante se emplean todas las muestras disponibles hasta el momento para calcular los valores.

Por último, la pestaña *Control*, dispone de cuatro cuadros de entrada de texto que permiten al usuario modificar los cuatro parámetros dinámicos principales: tiempo de integración, tiempo de intra-calibración y ángulos de apuntamiento del haz. Asimismo incluye tres botones para activar las secuencias de validación: *Open Loop*, *Phase Sweep* y *Closed Loop*.

La figura 7-8 muestra las pestañas *Statistics* y *Control* con mayor detalle:

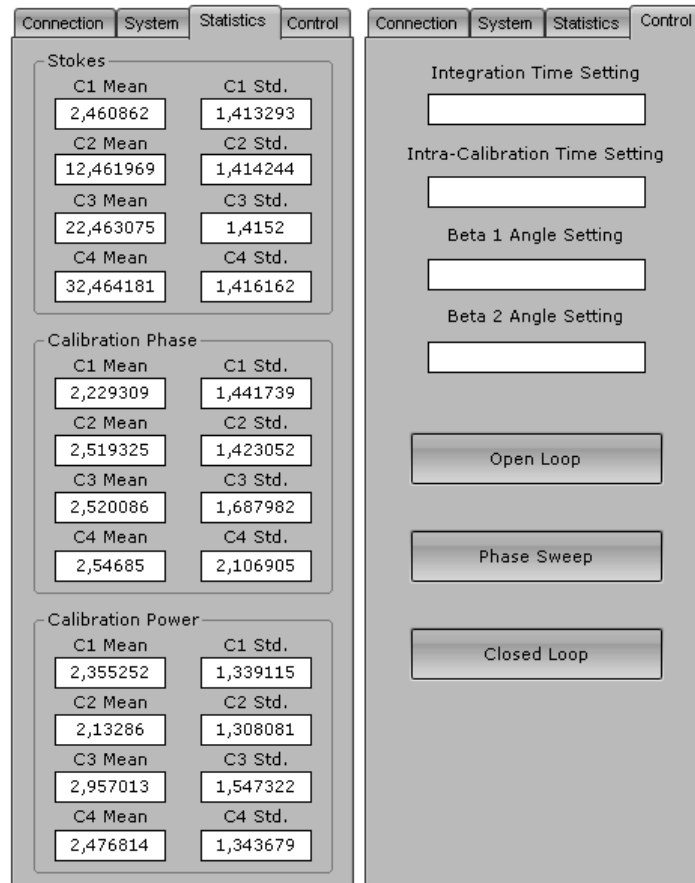


Figura 7-9: Detalle de las pestañas *Statistics* y *Control* de la zona de control de la interfaz.

Por otra parte, la zona de representación, se compone de tres pestañas que dan acceso a la visualización de: los parámetros de Stokes, la fase y la potencia de calibración y el resultado de las secuencias de validación. Más adelante se incluyen capturas que muestran con mayor detalle la zona de representación.

El eje horizontal de los cuatro paneles gráficos que componen la zona de representación marca el tiempo transcurrido en segundos desde el inicio de la conexión. Esto permite referenciar posibles acontecimientos observados con instantes concretos de tiempo.

7.4.2.2 Arquitectura

La arquitectura de la nueva interfaz de captura de datos se basa en la ocurrencia de eventos a los que se les asignan funciones determinadas. Así pues, el hecho de presionar alguno de los botones genera un evento registrado que ejecuta la función correspondiente. De la misma forma, la llegada de información a través del puerto serie provoca un evento que lanza la función encargada de capturar, procesar y representar la información.

La figura 7-9 muestra un esquema con las acciones que se llevan a cabo como respuesta a algunos de los principales eventos de la aplicación:

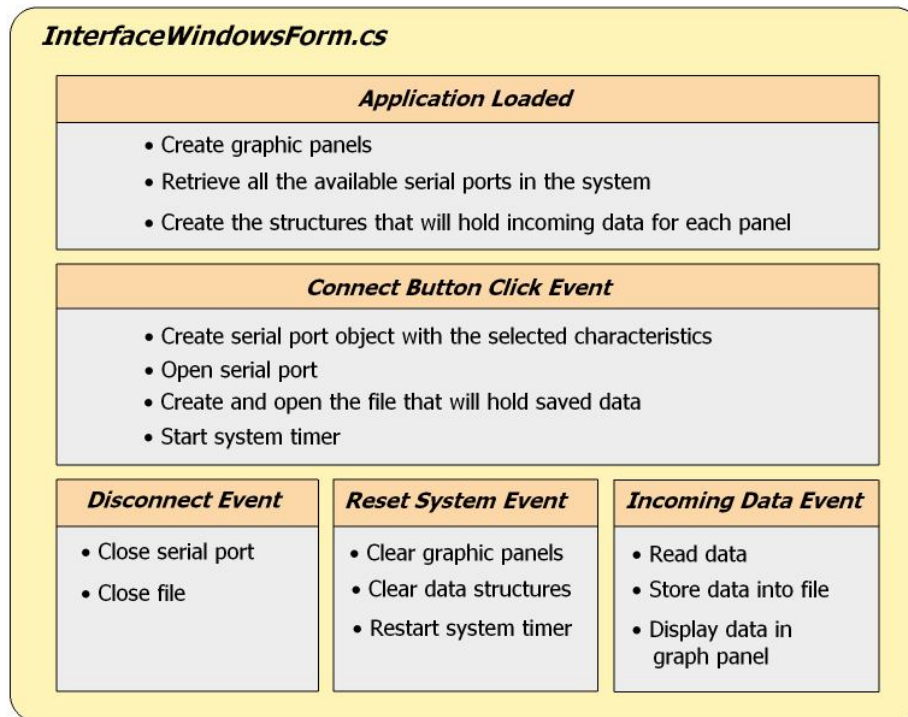


Figura 7-10: Diagrama de eventos de la interfaz de captura de datos. Se muestran únicamente tres de los múltiples eventos posibles una vez arrancada la aplicación y establecida la comunicación.

Al arrancar la aplicación se crean los paneles gráficos de las tres pestañas de la zona de representación así como los vectores que contendrán las muestras recibidas desde el radiómetro. Asimismo se obtiene la lista de todos los puertos serie disponibles en el sistema, que se mostrará en el panel de selección de puerto.

A partir de este punto, la aplicación permanece en espera hasta la llegada de un evento de conexión. Cuando el usuario acciona el botón de conexión, se abre el puerto serie con los parámetros seleccionados, se crea el fichero que almacenará los datos y se inicia el reloj del sistema que marcará la escala horizontal de todos los paneles gráficos. En caso de que el usuario pulse cualquiera de los botones de la interfaz antes del establecimiento de la conexión, el evento generará un mensaje de advertencia en la pestaña *System* de la zona de control.

Una vez establecida la conexión las acciones del usuario (pulsar un botón, modificar un valor, seleccionar una pestaña...) generan eventos que son recogidos y tratados según su naturaleza. En la figura 7-9 se ilustran tres de los múltiples eventos posibles.

Por razones obvias, es de especial importancia el evento de llegada de información al puerto serie. Este evento está programado para dispararse de forma

automática cada vez que en el buffer de entrada haya como mínimo 8 bytes (los tres bytes de señalización más 5 de datos). Cuando se produce este evento se leen caracteres del buffer de entrada hasta encontrar un carácter de fin de trama (en este caso el símbolo de exclamación: !). En caso de que no se haya completado la transmisión de la trama desde la UC se espera la llegada del carácter de fin de trama durante el tiempo establecido por el usuario. Si el evento lo provoca la llegada de múltiples tramas (es decir, el buffer de entrada contiene más de 8 bytes en el momento de realizar la lectura), estas se procesarán por orden de llegada.

La figura 7-10 ilustra, de manera simplificada, el sistema de procesamiento de las tramas:

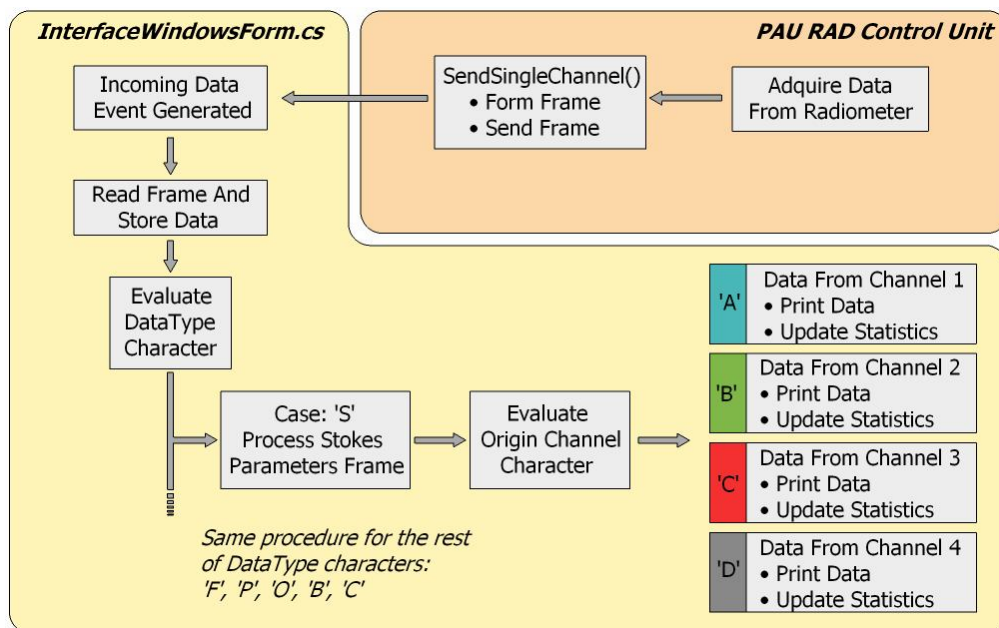


Figura 7-11: Esquema que ilustra el mecanismo de procesamiento de las tramas procedentes de la Unidad de Control del radiómetro. Se muestra únicamente el caso de recepción de una trama de parámetros de Stokes, el procedimiento es el mismo para el resto de tramas.

Los caracteres de identificación de tipo de dato permiten decidir a cuál de los cuatro paneles gráficos de la zona de representación va destinada la información contenida en la trama. Por su parte, el carácter de identificación de canal de origen evita posibles filtraciones de muestras de un canal a otro, y permite ubicar las muestras en su traza correspondiente del panel gráfico. De esta forma, toda la información queda almacenada en 16 vectores (4 vectores por panel, uno para cada canal) de 1500 muestras de longitud cada uno.

Una vez actualizada la información gráfica en los paneles, se lleva a cabo el cálculo de las estadísticas que se mostrarán en la pestaña *Statistics*. Para facilitar la lectura de los resultados (un refresco de la información demasiado frecuente hace que la variación de los decimales resulte desconcertante), este cálculo se lleva a cabo

únicamente cada 20 muestras. No obstante, para proporcionar medidas precisas, se emplean todos los valores contenidos en el vector de muestras. Por tanto, los valores presentados se corresponden exactamente con lo que el usuario visualiza en la zona de representación.

7.4.2.3 Aplicación de pruebas de transmisión

Con el fin de comprobar el correcto funcionamiento de la nueva interfaz, se desarrolló una aplicación capaz de simular la transmisión de los diferentes tipos de tramas. La figura 7-11 muestra una captura de esta aplicación:

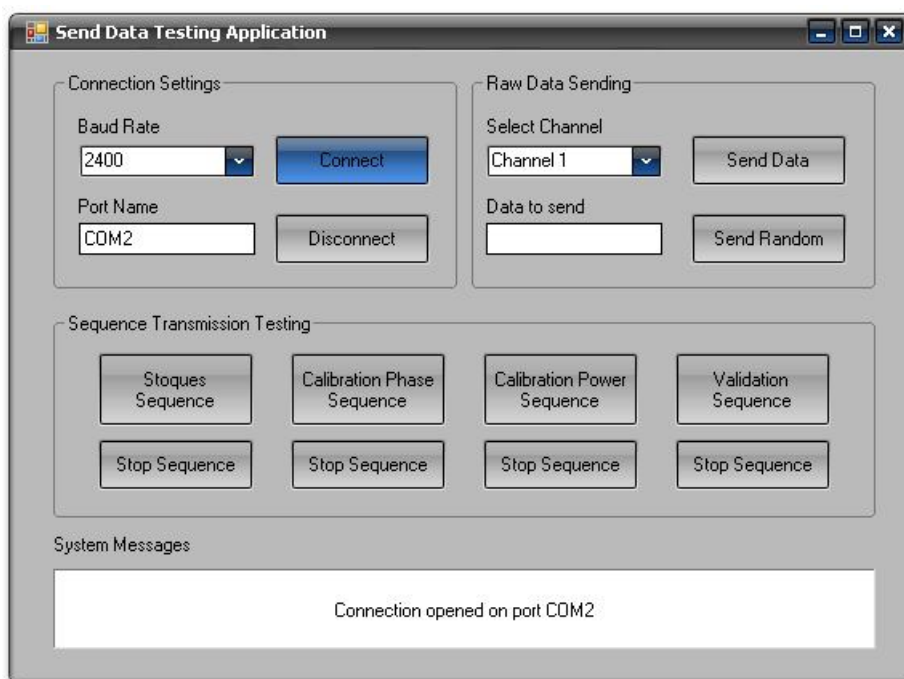


Figura 7-12: Captura de la aplicación de pruebas de transmisión.

Utilizando un emulador de puertos (Virtual Serial Port Driver), se creó una pareja de puertos en la misma máquina, unidos mediante una conexión null-modem virtual. Gracias a esta configuración es posible testear y depurar, en una única máquina, la nueva interfaz de captura de datos sin que para ello sea necesario disponer de un puerto serie físico, ni del radiómetro.

La aplicación de pruebas de transmisión, permite establecer una conexión con cualquiera de los puertos serie disponibles en la máquina. En la figura 7-12 se puede apreciar que la conexión está establecida en el puerto COM2, que, de forma virtual, está conectado al puerto COM1, puerto que en ese momento empleaba la interfaz de captura de datos.

Establecida la conexión, es posible iniciar la transmisión de cuatro tipos de tramas diferentes: parámetros de Stokes, fase de calibración, potencia de calibración, y secuencia de validación (que en esta versión implementa únicamente la secuencia de validación *Open Loop*). La información contenida en las tramas se genera de forma aleatoria y la periodicidad es de un segundo (se envía una trama cada segundo). Una vez iniciada la transmisión de tramas, no se detiene hasta que el usuario pulsa el botón de stop correspondiente.

También es posible componer una trama de forma manual para verificar que los valores que se representan en la interfaz coinciden con los enviados. Para ello es necesario seleccionar el canal a través del cual se quieren enviar los datos, introducir el valor en el cuadro de texto (incluyendo el carácter de identificación de tipo de dato) y pulsar el botón *Send Data*.

7.4.2.4 Funcionamiento de la interfaz

A continuación se incluyen varias capturas que muestran con mayor detalle el comportamiento de los diferentes paneles gráficos de la interfaz durante la recepción de tramas procedentes de la aplicación de pruebas de transmisión.

- *Stokes Parameters*

La figura 7-12 muestra el funcionamiento de la interfaz durante la transmisión de los parámetros de Stokes:

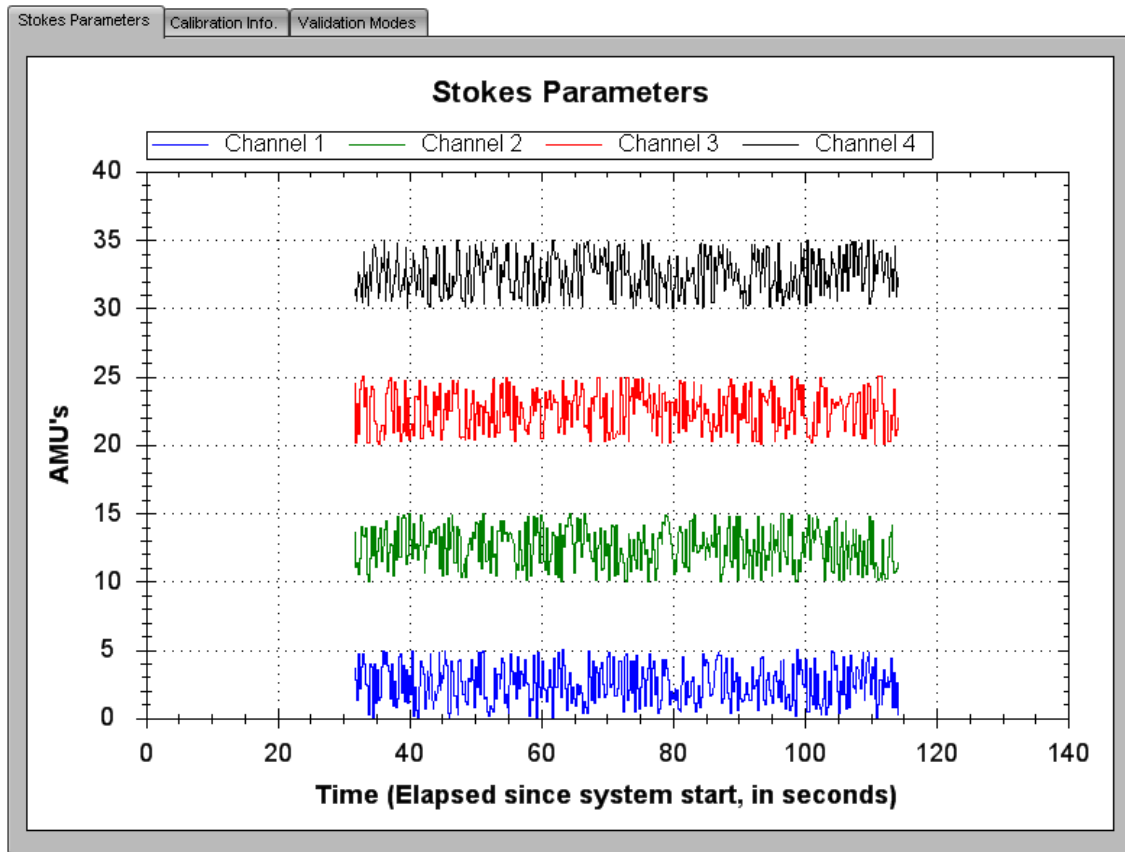


Figura 7-13: Detalle de la pestaña Stokes Parameters, durante las pruebas de transmisión.

Los valores enviados corresponden a una variable aleatoria uniformemente distribuida entre 0 y 5, con medias diferentes para cada uno de los cuatro canales.

La figura 7-13 muestra los valores de media y desviación típica obtenidos para los vectores de muestras representados en la figura 7-12. Como se puede observar tanto la media como la desviación típica calculadas coinciden con los valores calculados.

Stokes	
C1 Mean	C1 Std.
2,489047	1,412997
C2 Mean	C2 Std.
12,520321	1,450438
C3 Mean	C3 Std.
22,530023	1,446498
C4 Mean	C4 Std.
32,523355	1,510486

Figura 7-14: Valores de las estadísticas correspondientes a la captura de la figura 7-12.

o *Calibration Info.*

En la figura 7-14 se puede apreciar el aspecto que presenta la pestaña *Calibration Info.* Como se puede observar, esta pestaña dispone de dos paneles gráficos: el superior muestra la fase de calibración para los cuatro canales, mientras que en el inferior se representan las potencias de calibración de los cuatro canales.

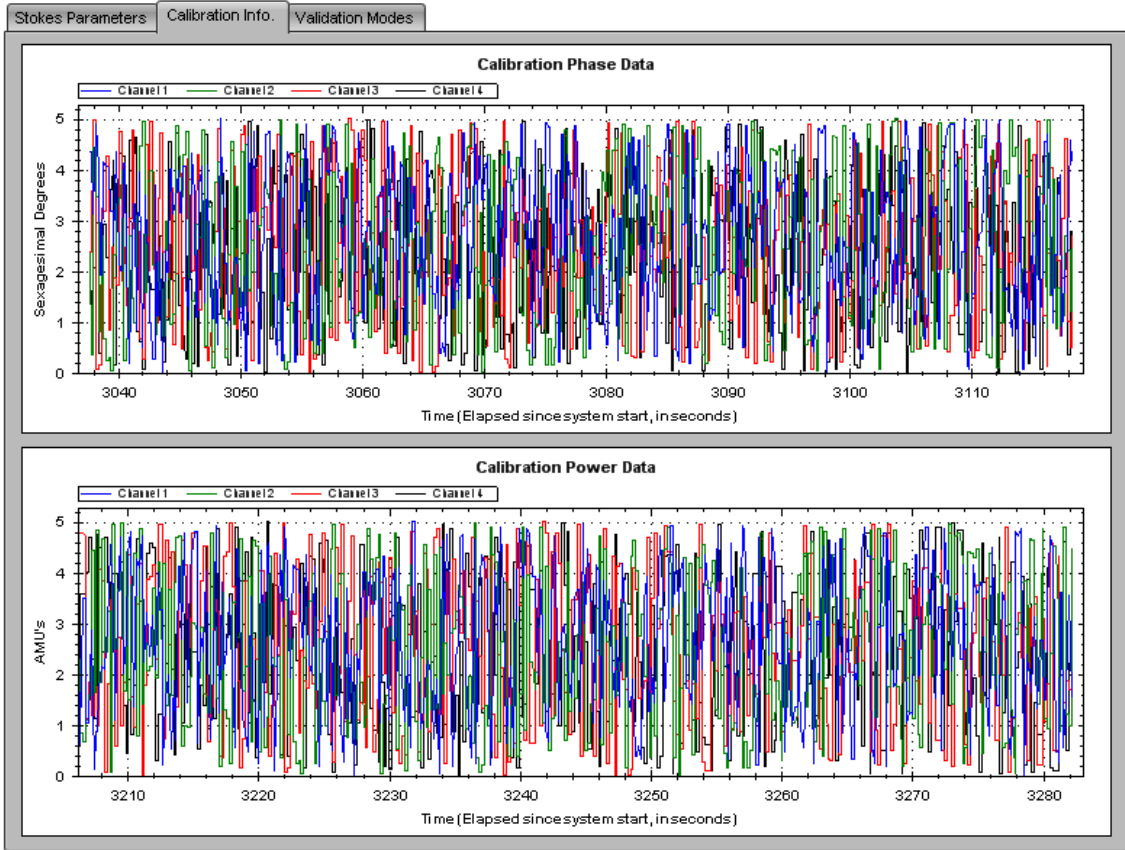


Figura 7-15: Detalle de la pestaña *Calibration Info*, durante las pruebas de transmisión.

Los datos que generan estas trazas de nuevo proceden de variables aleatorias uniformemente distribuidas entre 0 y 5, pero en esta ocasión la media es la misma para los cuatro canales.

Calibration Phase		Calibration Power	
C1 Mean	C1 Std.	C1 Mean	C1 Std.
2,685184	1,433309	2,676848	1,423526
C2 Mean	C2 Std.	C2 Mean	C2 Std.
2,551389	1,4959	2,501398	1,470238
C3 Mean	C3 Std.	C3 Mean	C3 Std.
2,474375	1,454523	2,521369	1,477798
C4 Mean	C4 Std.	C4 Mean	C4 Std.
2,4108	1,450571	2,49581	1,459801

Figura 7-16: Valores de las estadísticas correspondientes a la captura de la figura 7-14.

En la figura 7-15 se puede comprobar que los valores obtenidos tanto para la media como para la desviación típica, coinciden con los que se pueden observar en la figura 7-14.

- *Validation Modes*

Las secuencias de validación se representan en la pestaña *Validation Modes*. En la siguiente figura se muestra una captura de dicha pestaña durante la transmisión de una secuencia *Open Loop* generada en la aplicación de pruebas.

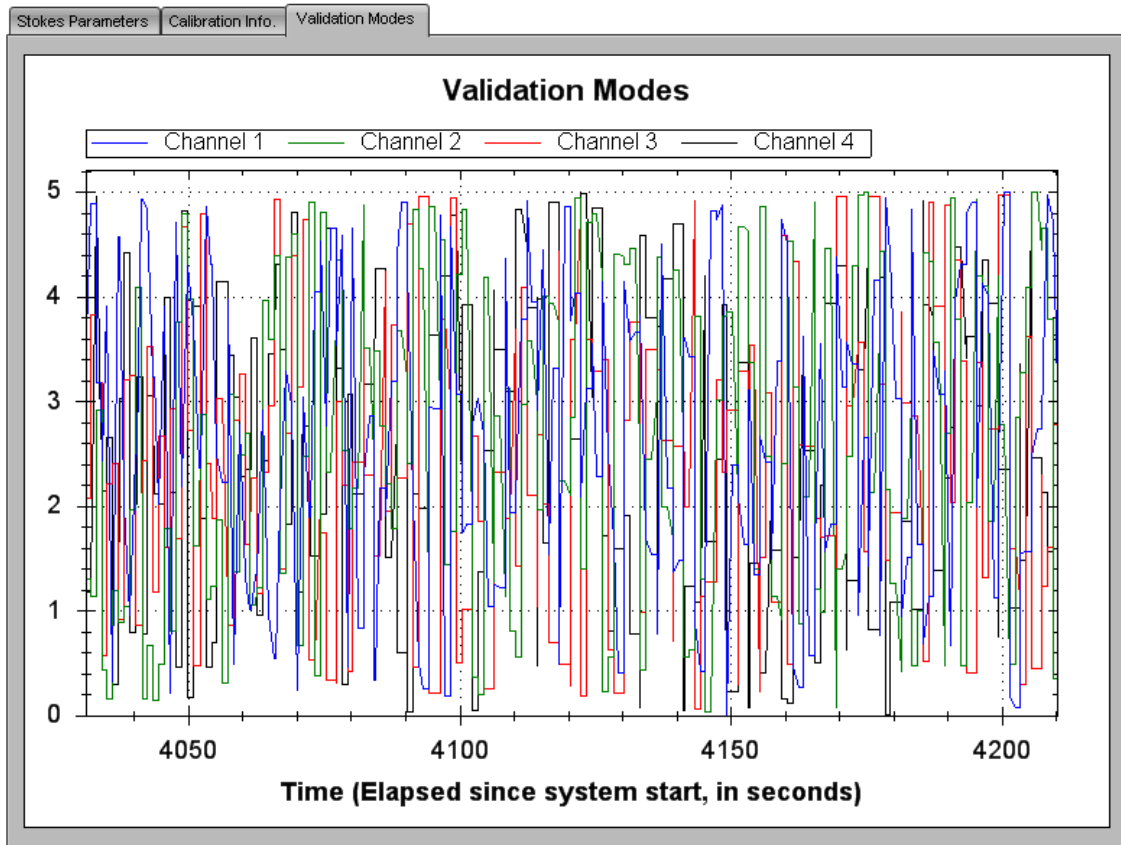


Figura 7-17: Detalle de la pestaña *Validation Modes*, durante las pruebas de transmisión.

- *Funcionalidades añadidas*

Por último, destacar algunas de las funcionalidades añadidas de las que dispone la nueva interfaz. En la siguiente figura se muestra un zoom de una de las trazas. Para hacer zoom sobre una zona en concreto basta con arrastrar el cursor con el botón izquierdo del ratón pulsado y seleccionar la zona. También es posible aumentar y reducir el zoom utilizando la rueda del ratón.

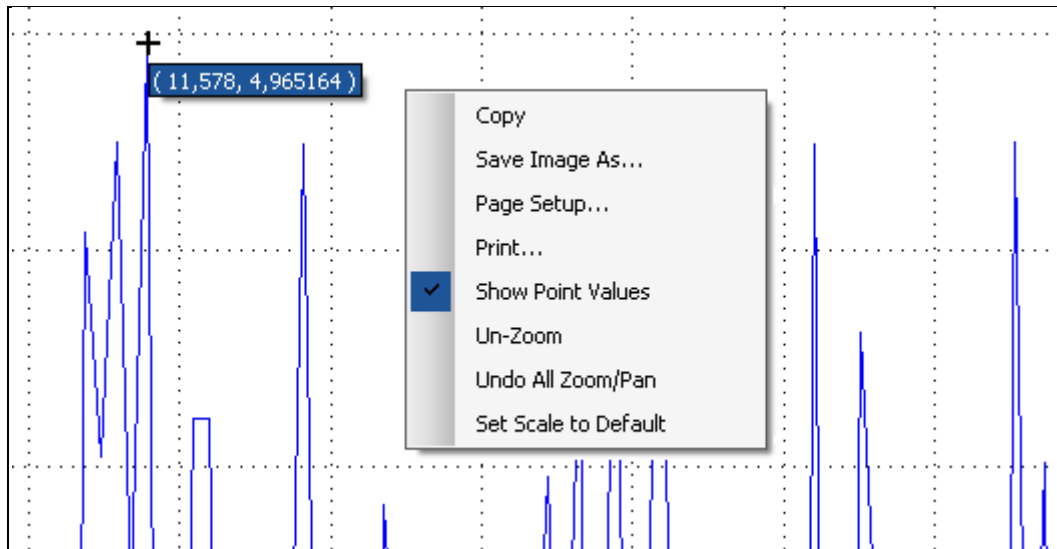


Figura 7-18: Zoom de una de las trazas y menú de opciones.

El menú de opciones que se puede ver en la figura, se muestra al hacer click con el botón derecho del ratón sobre cualquiera de los paneles gráficos. En él se puede acceder a opciones tales como: guardar la pantalla actual como una imagen, imprimirla directamente, mostrar los valores numéricos de los puntos que conforman la traza (tal y como se puede apreciar en la imagen), o restaurar el zoom a su valor por defecto.

8 CONCLUSIONES Y LÍNEAS DE TRABAJO FUTURAS

Como es lógico, los inicios en un proyecto de las dimensiones y complejidad de PAU-RAD suponen un reto para cualquiera. Es necesario tiempo para llegar a asumir los conceptos teóricos que constituyen la base fundamental de todo el sistema, comprender el funcionamiento de cada uno de los elementos que lo componen y entender cómo estos elementos interactúan entre sí. Una vez asumida toda esta información, se necesita más tiempo para asimilar el trabajo realizado por otros compañeros en proyectos anteriores, trabajo que constituye la piedra angular sobre la que se asienta este proyecto. Por tanto, se necesita mucho tiempo hasta estar preparado para contribuir al proyecto con una aportación significativa.

A lo largo del transcurso del proyecto se ha trabajado en múltiples campos: en el ámbito teórico durante el desarrollo de las ecuaciones del sistema; desarrollando software tanto para la Unidad de Control como para la Interfaz de Usuario; a nivel de hardware, ya sea durante el desarrollo del hardware de calibración o durante las múltiples pruebas efectuadas con los receptores; y, por supuesto, a nivel experimental, desarrollando hipótesis que justificasen los resultados obtenidos e ideando pruebas de laboratorio para validar o refutar dichas hipótesis. La completitud del trabajo realizado y la experiencia obtenida en todos estos campos compensa, con creces, el tiempo invertido en el proyecto.

Se han mejorado los mecanismos de presentación de la información al usuario, lo que a su vez ha permitido una interpretación más sencilla, rápida y fiable de los resultados obtenidos.

Se han diseñado, implementado, testeado y validado diferentes tipos de algoritmos de calibración así como nuevos métodos de validación del hardware del sistema. Al término de este proyecto, se dispone de un gran abanico de posibilidades en lo que a calibración se refiere. Esta variedad de algoritmos dotan al sistema de una gran flexibilidad, y de la capacidad de adaptarse a múltiples situaciones de medida.

Incorporando técnicas de filtrado adaptativo conjuntamente con los procesos de calibración, se ha mejorado la respuesta del sistema, aumentando la precisión y fiabilidad del mismo.

Al inicio de este proyecto se planteó como objetivo principal la extensión del sistema a su versión final de 16 receptores. Debido al trabajo realizado en otras áreas del sistema no fue posible asumir este objetivo. No obstante, al término del proyecto, se dispone de una versión preliminar que opera con 4 receptores y que constituye una prueba de viabilidad del concepto original de PAU-RAD.

Finalmente, se ha iniciado el necesario proceso de migración del sistema a una nueva FPGA. Este proceso ha requerido la creación de una nueva Unidad de Control así como de una nueva Interfaz de Usuario, lo que ha supuesto un nuevo reto, al tener que lidiar con tecnologías todavía en proceso de maduración (como es el caso de los mecanismos de comunicación Ethernet de la placa Stratix II).

No obstante, todo este trabajo no constituye más que una pequeña aportación al proyecto. Ninguno de los resultados obtenidos puede darse por definitivo, y todavía queda mucho por hacer. Es necesario continuar trabajando con el mismo espíritu y entrega para mejorar en todos los aspectos y llevar a buen término el proyecto. En este sentido, se han identificado las siguientes líneas de trabajo que será necesario seguir en un futuro cercano:

- En el ámbito de los algoritmos de calibración es necesario seguir trabajando en identificar y corregir los efectos no deseados que el hardware introduce en el sistema, especialmente en lo referente a acoplamientos de señal entre canales. A tal efecto se están barajando dos posibilidades. La primera consistiría en eliminar dos de los cuatro canales de cada receptor para emular un receptor PAU2. Esta solución presenta el inconveniente de eliminar una de las dos polarizaciones, hecho que se podría solventar añadiendo un conmutador a la entrada o utilizando un receptor para cada polarización. La segunda posibilidad pasa por el uso de materiales capaces de apantallar el campo magnético generado por las bobinas de los filtros de la etapa de FI, máximos responsables de las interferencias co-canal. Estos materiales son los conocidos como *μ-metals*. Esta alternativa presenta el inconveniente de que, si bien, a priori, solventa el problema, es posible que surjan otros problemas ocultos hasta el momento, pero presentes en el sistema, como por ejemplo: bucles de masa.
- A pesar de que es perfectamente viable, no se ha extendido el filtrado de los resultados a los algoritmos de calibración de ganancia. Puesto que toda la arquitectura ya está desarrollada y no requiere un gran esfuerzo, sería conveniente incluir en el sistema la opción de filtrado del error de ganancia.
- Extender el sistema a su versión final de 16 elementos y verificar el correcto funcionamiento del mismo. No obstante, para llevar a cabo esta tarea, es necesario previamente completar la migración del sistema a la nueva FPGA.

- Completar la migración del sistema completo a la FPGA Stratix II. Será necesario por tanto completar la nueva Unidad de Control implementando todos los estados de la Unidad de Control original así como las secuencias de funcionamiento. También será necesario continuar trabajando en la Interfaz de Usuario para permitir al usuario un mayor nivel de control sobre los procesos de calibración y sobre el funcionamiento del sistema en general.

9 BIBLIOGRAFIA

- [1] [Adriano Camps, 2004] Camps, A.; "Passive Advanced Unit (PAU): A Hybrid L-band Radiometer, GNSS-Reflectometer and IR-radiometer for Passive Remote Sensing of the Ocean", European Young Investigator Awards 2004.
- [2] [Bosch, 2005] Bosch, X.; "Disseny i implementació en FPGA d'un radiòmetre polarimètric de pseudo-correlació amb digital beamforming i autocalibració", PFC ETSETB, 2005.
- [3] [Ramos, 2005] Ramos, I.; "Diseño e implementación de un receptor para un radiómetro de pseudocorrelación polarimétrico en banda L", PFC ETSETB, 2005.
- [4] [Banqué, 2006] Banqué, X.; "Disseny i implementació de la unitat de control, calibració i comunicació d'un radiòmetre amb beamforming digital"
- [5] [Izquierdo, 2006] Izquierdo, B.; "Disseny i integració d'un convertidor A/D per un radiòmetre polarimètric de pseudocorrelació amb beamforming digital", PFC ETSETB, 2006.
- [6] [Hagen, Farley, 1973] Hagen, J.; Farley, D.; " Digital correlation techniques in radio science" Radio Sci., vol. 8, no. 8, 1973.
- [7] [Vericat, 2009] Vericat, M.; "Especificació i prestacions de radiòmetres digitals", PFC ETSETB, 2009
- [8] [Broquetas i Camps, 2000] Broquetas, A.; Camps, A.; "Teledetecció", transparències de classe, 2000.
- [9] [Camps, 1996] Camps, A.; "Application of interferometric radiometry to Earth observation", PhD Thesis, 1996.
- [10] [Ulaby et al., 1981] Ulaby, F.T.; Moore, R.K.; Fung, A. K. "Microwave remote sensing Active and Passive, Vol.1. Fundamentals of Radiometry", Addison-Wesley, 1981.
- [11] [Skou et al., 1999] Skou, N.; Lauren, B.; SØbjærg, S.; "Polarimetric Accuracy Configurations: Potential Accuracy and Sensitivity", IEEE Transactions on Geoscience and Remote Sensing, Vol. 37, no. 5, September 1999.
- [12] [Camps et al., 2007] Camps, A.; Bosch-Lluis, X.; Ramos-Perez, I.; Marchan-Hernandez, J.F.; Izquierdo, B.; Rodriguez-Alvarez, N.; "New Instrument Concepts for Ocean Sensing: Analysis of the PAU-Radiometer", IEEE Transactions on Geoscience and Remote Sensing, 2007.
- [13] [Lim, et al.. 2006] Lim, B.H.; Ruf S.C.; "Calibration of a Fully Polarimetric Microwave Radiometer Using a Digital Polarimetric Noise Source" IEEE Transactions on Geoscience and Remote Sensing, 2006.