



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Modelado del procesamiento de un
sistema de radar pulsado e
instrumentación del módulo de
compresión de pulso en un FPGA**

TESIS

Que para obtener el título de

Ingeniero Eléctrico Electrónico

P R E S E N T A

Luis Eduardo Rojas Pozos

DIRECTOR DE TESIS

Dr. Oleksandr Martynyuk



Ciudad Universitaria, Cd. Mx., 2025



**PROTESTA UNIVERSITARIA DE INTEGRIDAD Y
HONESTIDAD ACADÉMICA Y PROFESIONAL
(Titulación con trabajo escrito)**



De conformidad con lo dispuesto en los artículos 87, fracción V, del Estatuto General, 68, primer párrafo, del Reglamento General de Estudios Universitarios y 26, fracción I, y 35 del Reglamento General de Exámenes, me comprometo en todo tiempo a honrar a la institución y a cumplir con los principios establecidos en el Código de Ética de la Universidad Nacional Autónoma de México, especialmente con los de integridad y honestidad académica.

De acuerdo con lo anterior, manifiesto que el trabajo escrito titulado MODELADO DEL PROCESAMIENTO DE UN SISTEMA DE RADAR PULSADO E INSTRUMENTACION DEL MODULO DE COMPRESION DE PULSO EN UN FPGA que presenté para obtener el título de INGENIERO ELÉCTRICO ELECTRÓNICO es original, de mi autoría y lo realicé con el rigor metodológico exigido por mi Entidad Académica, citando las fuentes de ideas, textos, imágenes, gráficos u otro tipo de obras empleadas para su desarrollo.

En consecuencia, acepto que la falta de cumplimiento de las disposiciones reglamentarias y normativas de la Universidad, en particular las ya referidas en el Código de Ética, llevará a la nulidad de los actos de carácter académico administrativo del proceso de titulación.

LUIS EDUARDO ROJAS POZOS
Número de cuenta: 419051015

Agradecimientos

A mis padres, Luis Miguel y Rosa María, por su amor y apoyo incondicional. Sin ustedes, no sería la persona que soy al día de hoy. Siempre estaré agradecido por su cariño, su paciencia y todo lo que me han inculcado. Las palabras se quedan cortas para poder expresar lo que significan para mí.

A mi hermana, Marian, por su cariño y compañía. Gracias por tu consideración y, aunque no te lo digo mucho, eres alguien a quien admiro por su perseverancia, entre otras tantas cosas. Le agradezco a la vida por permitirme verte crecer y alcanzar cada una de tus metas.

A Baxter, por haberme acompañado mientras realizaba este trabajo escrito. Aunque no seas capaz de leer esta dedicatoria, tu lealtad y amor fueron muy importantes a lo largo de este proceso.

A mi asesor, el doctor Oleksandr Martynyuk, por su apoyo y enorme paciencia. Siempre estaré agradecido por su orientación y consideración hacia mi persona. Su gran vocación por la docencia y la investigación siempre será un verdadero ejemplo a seguir.

Al doctor Saúl de la Rosa Nieves, por haber confiado en mí y en mi trabajo. Le agradezco las oportunidades y enseñanzas que me ha brindado hasta el día de hoy.

Al doctor Jorge Rodríguez Cuevas, por su sabiduría, su honestidad, su comprensión y los consejos que me ha brindado. Gracias por ayudarme a encontrar mi camino y expandir mis horizontes.

Al resto de mi familia: a mi abuelita, mis tíos y mis primos. Les agradezco encarecidamente su confianza, paciencia y cariño. Espero poder celebrar con ustedes este logro en un futuro próximo.

A Neyci, por tantos años de amistad a pesar de la distancia, siempre serás alguien a quien admiro mucho y espero sepas lo mucho que te aprecio.

A Armando Ugalde, ya que sin ti no estaría aquí. Siempre estaré en deuda contigo, me alegra poder verte triunfar ya que te lo mereces.

A Erick, Fernando, Luis Gerardo y Rodrigo, por haberme acompañado desde la secundaria. Gracias por su amistad, sus consejos y por todas las cosas que hemos vivido juntos. Espero poder contar con ustedes por el resto de mi vida.

A Rafael, Serrato y Sebastián, por su amistad, lealtad y perspectiva. Les agradezco por ser siempre encontrarse dispuestos a escucharme y ayudarme. Me alegra mucho poder verlos triunfar en sus respectivos rubros.

A Juan Pablo, por tu consideración, tu honestidad y tus consejos. Espero siempre poder contar con tu amistad y con tu consejo en materia legal.

A mis amigos de primer semestre: Edgar, José, Alfonso, Diana e Iván. Les agradezco de todo corazón haberme acompañado a lo largo de esta etapa, su amistad fue fundamental a lo largo de la carrera.

A mis amigos Moisés y José Daniel, por las risas, discusiones y el enorme apoyo que siempre me han brindado.

A Guillermo, por acompañarme en el proceso y hacerlo un poco más llevadero. Fue un camino arduo y agotador, pero me alegra mucho verte concluir tu tesis de maestría.

A mis compañeros y amigos del LIESE, en especial a Aldair, Israel, Isai, Derian, Rebeca, Eduardo, David, Badillo y muchos más. Gracias por su confianza, sus consejos y todo lo que me han enseñado durante mi estancia en el laboratorio.

A mis profesores, por los conocimientos que me han transmitido y por inculcarme una pasión ferviente hacia mi carrera, con una dedicatoria especial hacia: Dr. Michael Rojas, Dr. Jose María Gómez, M.I. Ricardo Mota y M.I. Larry Escobar.

A aquellas personas que ya no forman parte de mi vida o ya no se encuentran aquí, por su cariño, dedicación y todas las experiencias que compartimos en su momento. En donde sea que se encuentren, les deseo lo mejor.

Índice general

1. Introducción	10
1.1. Perspectiva histórica	10
1.2. Planteamiento de la propuesta a desarrollar	21
1.3. Objetivos	22
1.3.1. Objetivo general	22
1.3.2. Objetivos específicos	22
1.4. Conclusiones	23
2. Marco teórico	24
2.1. Sistemas de radar	24
2.1.1. Clasificación los sistemas de radar	25
2.1.2. Sistema de coordenadas para radar	26
2.1.3. Estructura general del sistema de radar monoestático	27
2.1.4. Arquitecturas de recepción	31
2.2. Antenas	33
2.2.1. Antenas en arreglo de fase	35
2.2.2. Arreglos lineales uniformes	36
2.3. Procesamiento de señales para radar	40
2.3.1. Formadores de haz	40
2.3.2. Compresión de pulso	42
2.3.3. Efecto Doppler	43
2.4. Técnicas de procesamiento digital de señales	44
2.4.1. Filtrado de señales	44
2.4.2. Transformada discreta de Fourier (DFT)	45
2.5. Conceptos relevantes de diseño digital	48
2.5.1. Formato de punto fijo	48
2.5.2. Arreglos de compuertas programables en campo (FPGA)	49

2.5.3. Lenguajes de descripción de <i>hardware</i>	49
2.6. Conclusiones	50
3. Diseño	52
3.1. Metodología de diseño	52
3.2. Modelado del procesamiento del bloque de recepción del sistema de radar pulsado monoestático	53
3.2.1. Planeación del sistema	53
3.2.2. Desarrollo de concepto	54
3.2.3. Diseño a nivel sistema	56
3.2.4. Diseño a detalle	61
3.3. Instrumentación del módulo de compresión de pulso en un FPGA	80
3.3.1. Planeación del sistema	80
3.3.2. Desarrollo de concepto	81
3.3.3. Diseño a nivel sistema	85
3.3.4. Diseño a detalle	96
3.4. Conclusiones	116
4. Resultados	119
4.1. Caracterización del modelo de procesamiento	119
4.1.1. Comportamiento del modelo ante un objeto estático	120
4.1.2. Comportamiento del modelo ante un objeto en movimiento	136
4.2. Resultados de la instrumentación del módulo de compresión de pulso en un FPGA	142
4.2.1. Condiciones de prueba para el núcleo de procesamiento	142
4.2.2. Filtro en concordancia	142
4.2.3. Estimador de distancia	144
4.2.4. Recursos requeridos para la implementación del núcleo de procesamiento	146
4.3. Conclusiones	147
5. Conclusiones y trabajo a futuro	149
Anexos	152
Referencias	153

Índice de figuras

1.1. Módulo lógico intercambiable para la computadora AN/FSQ-7 [6].	11
1.2. Compuertas de alcance para un radar pulsado [7].	12
1.3. Plano horizontal de la antena giratoria.	12
1.4. Funcionamiento del detector de ventana corrediza [1].	13
1.5. Estructura del detector del ventana corrediza [8].	13
1.6. Estructura del elemento aritmético del FDP [1].	14
1.7. Espectro del filtro usado por MTI [9].	15
1.8. Diagrama a bloques de la primera versión del procesador MTD [10].	16
1.9. Diagrama a bloques de la segunda versión del procesador MTD [11].	17
1.10. Comparativa de filtros usados en MTI y MTD [12].	17
1.11. Arquitectura del DCS [1].	18
1.12. Implementación convencional de un DBF [13].	19
1.13. Diagrama a bloques del sistema demostrativo desarrollado [16].	20
1.14. Arquitectura de SDR con muestreo digital en frecuencia intermedia [14].	21
2.1. Bandas de frecuencia de acuerdo al estándar IEEE 521-2019 [19].	25
2.2. Sistema de coordenadas de radar.	26
2.3. Diagrama a bloques de un sistema monoestático.	27
2.4. Mezclador de conversión.	28
2.5. Mezcladores de conversión [24].	29
2.6. Demodulación de señales en cuadratura.	30
2.7. Receptor homodino o de conversión directa [26].	31
2.8. Conversión a IF de la señal de interés con su frecuencia imagen [26].	32
2.9. Filtro de rechazo de imagen [26].	32
2.10. Frente de onda plano [29].	33
2.11. Propagación de las ondas para una antena isotrópica [30].	34
2.12. Lóbulos del patrón de radiación [31].	34
2.13. Patrón de interferencia del arreglo [13].	36

2.14. Patrón de interferencia al aplicar diferencia de fase [13].	36
2.15. Arreglo lineal uniforme.	37
2.16. Compensación de la diferencia de fase en un formador de haz [13].	42
2.17. Frentes de onda para una fuente en movimiento y una fuente estática [37].	44
2.18. Estructura básica de procesamiento de la FFT <i>radix-2</i> [41].	47
2.19. Estructura básica de procesamiento de la FFT <i>radix-4</i> [41].	48
3.1. Etapas principales del sistema.	54
3.2. Modelo simplificado de la recepción de señales.	55
3.3. Diagrama ilustrativo del procesamiento digital de señales.	56
3.4. Diagrama a bloques de la primera etapa.	58
3.5. Diagrama a bloques de la segunda etapa.	60
3.6. Propuesta para el desarrollo de la generación de señales.	61
3.7. Generación de señales recibidas de RF en AWR Microwave Office.	64
3.8. Propuesta para el desarrollo del acondicionamiento de señales.	65
3.9. Modelo del acondicionamiento de señales.	67
3.10. Mezclador MM1-0212LS incorporado en el sistema.	67
3.11. Propuesta para el desarrollo del procesamiento digital de señales.	69
3.12. Bloque correspondiente a la demodulación en cuadratura.	74
3.13. Bloque correspondiente al cambio de representación rectangular a polar.	75
3.14. Bloque correspondiente a la aplicación de pesos y el cambio de representación.	76
3.15. Suma de componentes reales e imaginarias para todos los elementos del arreglo.	76
3.16. Obtención de la magnitud de los haces y exportación de datos.	77
3.17. Desplazamiento de ángulos obtenidos mediante <i>atan2</i>	79
3.18. Etapas principales del módulo de compresión de pulso.	81
3.19. Implementación del filtro mediante la convolución.	82
3.20. Implementación del filtro mediante la multiplicación compleja.	82
3.21. Método propuesto para la estimación de distancia.	83
3.22. Diseño a nivel sistema propuesto.	92
3.23. Intercambio de datos mediante AXI4-Stream [62].	97
3.24. Esquemático provisto para el bloque IP [62].	98
3.25. Arquitectura del filtro en concordancia.	100
3.26. Arquitectura del controlador asociado al filtro en concordancia.	105
3.27. Diagrama de estados del controlador asociado al filtro en concordancia.	107
3.28. Arquitectura del estimador de distancia.	109
3.29. Diseño elaborado para la suma de cuadrados.	111

3.30. Arquitectura del sumador completo elemental original [63].	112
3.31. Arquitectura de la unidad de comparación.	113
3.32. Arquitectura del circuito restador modificado.	113
3.33. Diseño elaborado por el sintetizador para la unidad de cálculo de distancia. .	114
3.34. Arquitectura final del sistema.	115
4.1. Pulso rectangular generado para simular el objeto estático.	121
4.2. Generación de canales con la diferencia de fase requerida.	121
4.3. Magnitud de la respuesta en frecuencia del filtro paso banda empleado. . . .	122
4.4. Retardo de grupo del filtro paso banda empleado.	123
4.5. Comparación del caso real y caso ideal de la señal de IF obtenida posterior al filtro.	123
4.6. Espectro de la señal de IF obtenida posterior al filtrado.	124
4.7. Señal de IF amplificada y cuantizada.	125
4.8. Señal de IF digitalizada con ajuste de formato numérico.	125
4.9. Retardo de grupo del filtro paso bajas empleado.	126
4.10. Componentes obtenidas del demodulador en cuadratura en el primer canal. .	127
4.11. Forma polar de las componentes obtenidas en el primer canal.	128
4.12. Fase de la respuesta en frecuencia del filtro empleado durante la demodulación en cuadratura.	128
4.13. Patrones de radiación correspondientes a los ángulos de elevación escaneados.	129
4.14. Aplicación de pesos del formador de haces en el canal 16 del arreglo.	130
4.15. Representación polar de las salidas del formador de haces para el caso base. .	131
4.16. Magnitud de las salidas del formador de haces para un objeto situado en $\theta=56.25^{\circ}$	132
4.17. Magnitud de las salidas del formador de haces para un objeto situado en $\theta=72.5^{\circ}$	133
4.18. Compresión de pulso en los ocho haces formados para el objeto estático. . . .	134
4.19. Compresión de pulso en los ocho haces formados para un objeto situado a 2037 (izq.) y 4251 metros (der.).	135
4.20. Señal de IF obtenida posterior al filtro para el primer objeto en movimiento.	137
4.21. Señal de IF obtenida posterior al filtro para el segundo objeto en movimiento.	138
4.22. Representación polar de la señal obtenida en el primer canal para el primer objeto en movimiento.	138
4.23. Representación polar de la señal obtenida en el primer canal para el segundo objeto en movimiento.	139

4.24. Salida del formador de haces para el primer objeto en movimiento.	139
4.25. Salida del formador de haces para el segundo objeto en movimiento.	140
4.26. Compresión de pulso en los ocho haces formados para el objeto en movimiento.140	
4.27. Condiciones de prueba para el núcleo de procesamiento.	142
4.28. Simulación funcional del filtro en concordancia.	143
4.29. Transición de estados dentro del controlador asociado al filtro en concordancia.143	
4.30. Transición al estado de espera del controlador.	144
4.31. Propagación de datos a lo largo de las etapas de encauzamiento.	144
4.32. Estimación de distancia para la primera compuerta de alcance.	145
4.33. Estimación de distancia para la segunda compuerta de alcance.	145
4.34. Estimación de distancia para la tercera compuerta de alcance.	145
4.35. Recursos empleados para la implementación del sistema.	146
4.36. Desglose de recursos empleados para la implementación del sistema.	146

Índice de tablas

3.1. Estimación de latencia y recursos requeridos para 65,536 muestras.	87
3.2. Latencias aproximadas del núcleo de procesamiento para diferentes tamaños de compuerta.	89
3.3. Tiempo de llenado para dos búferes de datos considerando distintos tamaños de compuerta.	89
3.4. Determinación de los tamaños para los que se cumple el tiempo real.	90
3.5. Recursos requeridos para la implementación de Radix-4 para diferentes tamaños de compuerta.	90
3.6. Puertos de entrada y salida del filtro en concordancia	101
3.7. Tabla de verdad para las salidas del controlador asociado al filtro en concordancia.	107
3.8. Funciones lógicas reducidas para el estado siguiente.	107
3.9. Puertos de entrada y salida del estimador de distancia.	109
4.1. Resultados no compensados para la compresión de pulso en los ocho haces formados durante el caso base.	134
4.2. Resultados no compensados para la compresión de pulso en los casos de prueba pertinentes.	135
4.3. Resultados obtenidos para la estimación de velocidad radial del caso base. . .	136
4.4. Resultados no compensados para la compresión de pulso de los objetos en movimiento.	141
4.5. Resultados obtenidos para la estimación de velocidad radial de los objetos en movimiento.	141

Capítulo 1

Introducción

Durante la mayor parte del siglo pasado, los sistemas de radar desempeñaban sus funciones a través del cómputo realizado a partir de señales analógicas. Lo anterior presentaba diversas limitaciones entre las que figura la incapacidad de almacenar la información obtenida, lo que obligaba a procesar las señales de manera síncrona en el tiempo [1]. En caso de perder la señal de interés, el procesamiento no podía llevarse a cabo de forma retroactiva. Otra limitación importante fue la imposibilidad de modificar la configuración del sistema posterior a su construcción. A partir de dichas problemáticas y de la creciente disponibilidad de la electrónica digital, surge la oportunidad de migrar hacia sistemas digitales que permitan una mayor flexibilidad y adaptabilidad.

1.1. Perspectiva histórica

Los primeros esfuerzos en llevar a cabo la migración hacia sistemas digitales comenzaron en la década de los años cincuenta. La Guerra Fría estaba en pleno auge, haciendo menester que países como los Estados Unidos y la Unión Soviética contaran con sistemas de radar capaces de monitorear su espacio aéreo en todo momento. Aunado a ello, se requería procesar una cantidad enorme de información para identificar los objetivos detectados y tomar acciones según fuese necesario. En aras de automatizar el proceso, se incorporaron las primeras computadoras digitales de propósito general con el objetivo de llevar a cabo el gran volumen de operaciones matemáticas asociadas. Dichas computadoras aún no contaban con *hardware* dedicado al procesamiento digital de señales y se programaban mediante tarjetas perforadas o tableros extraíbles.

Durante la década de los años cincuenta surgieron desarrollos contemporáneos que marcaron el inicio del uso de la computación digital en aplicaciones militares. Uno de los primeros

fue el proyecto Computadora Digital Transistorizada (TRADIC, por sus siglas en inglés), iniciado en 1951 bajo la dirección de J. H. Felker en *Bell Telephone Laboratories* [2]. El proyecto TRADIC resultó en la primera computadora construida a partir de transistores [3] y se desarrolló con el objetivo de asistir en la navegación de los bombarderos de la fuerza aérea de los Estados Unidos. De manera paralela, otro antecedente fundamental fue el sistema Entorno Terrestre Semi-Automático (SAGE, por sus siglas en inglés), desarrollado por el *Lincoln Laboratory* del Instituto Tecnológico de Massachusetts (MIT, por sus siglas en inglés). Aunque con un propósito distinto al de TRADIC, el primero orientado a la navegación aérea y el segundo al control del espacio aéreo continental, ambos representan los primeros intentos de integrar computadoras digitales en sistemas militares estratégicos.

El sistema SAGE hacía uso de computadoras AN/FSQ-7 de redundancia doble para el procesamiento de las señales detectadas por el radar [4]. Dichas computadoras fueron desarrolladas por el MIT en conjunto con *International Business Machines Corporation* (IBM, por sus siglas en inglés) a partir de la computadora experimental *Whirlwind*, incorporando memoria de núcleo magnético y tubos de vacío [5]. La inclusión de un medio de almacenamiento digital permitió gestionar la inserción de datos provenientes de la red de radares asíncronos.

Sin embargo, la computadora AN/FSQ-7 seguía siendo, hasta cierto punto, una computadora de propósito general que no contaba con tecnología dedicada al procesamiento digital de señales. Si bien la computadora permitía procesar conjuntos considerables de datos, no existían instrucciones especializadas que permitieran mejorar el rendimiento de dicho procesamiento.

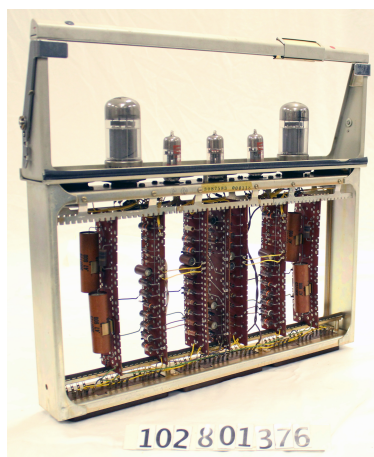


Figura 1.1: Módulo lógico intercambiable para la computadora AN/FSQ-7 [6].

Los retos a los que se enfrentaba el sistema SAGE, junto con las limitaciones de las

computadoras digitales disponibles en esa época, impulsaron innovaciones relevantes en el desarrollo de *hardware* especializado para el procesamiento digital de señales. Entre estos retos, uno de los más significativos fue la transmisión de datos correspondientes a los objetivos detectados a través de las líneas telefónicas de banda estrecha que conectaban los centros de dirección. Para abordar esta problemática, se desarrolló el detector de ventana corrediza, donde la expresión ventana corrediza designa el intervalo breve de tiempo durante el cual una antena giratoria permanecía orientada hacia una dirección específica en el plano horizontal. La antena en cuestión transmitía una serie de pulsos que correspondían a cada una de las direcciones específicas establecidas por la ventana corrediza. Aunado a esto, el detector empleaba compuertas de alcance, las cuales representan un intervalo de muestreo que se traduce en el recorrido ida y vuelta del pulso transmitido hacia un blanco a una distancia determinada.

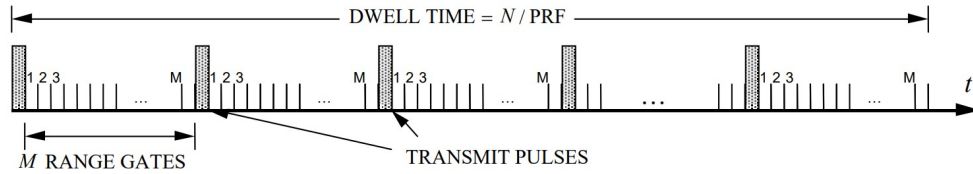


Figura 1.2: Compuertas de alcance para un radar pulsado [7].

En el diagrama de la figura 1.3, los sectores circulares representan las direcciones específicas del plano horizontal hacia las cuales se transmite cada pulso. A su vez, los anillos corresponden a las compuertas de alcance, cada una formada por múltiples pulsos que en conjunto cubren un giro completo de la antena.

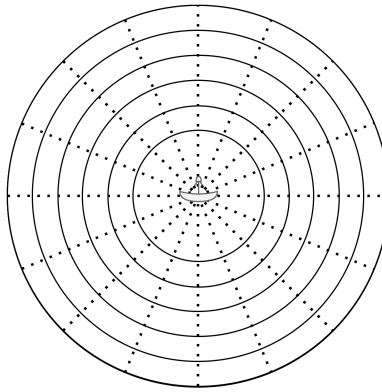


Figura 1.3: Plano horizontal de la antena giratoria.

El detector de ventana corrediza establecía un primer umbral en cada compuerta de alcance, de modo que solo se registraban los pulsos que lo superaban, como se observa en las figuras 1.4 (a) y (b). A continuación, la cantidad de pulsos registrados se acumulaba mediante un sumador, y el resultado se comparaba con un segundo umbral, denotado como μ en la figura 1.4 (b). Cuando dejaban de recibirse pulsos por encima del primer umbral, el valor del acumulador comenzaba a decrecer hasta llegar a cero.

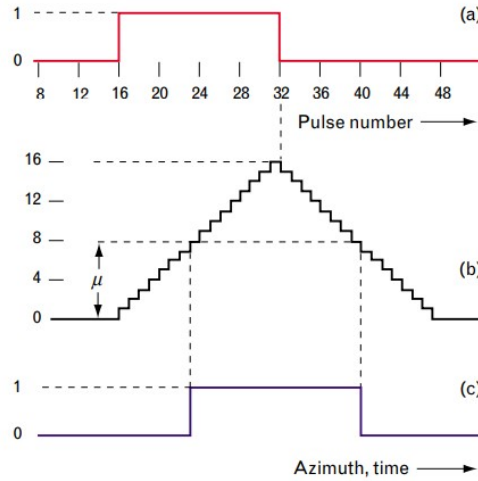


Figura 1.4: Funcionamiento del detector de ventana corrediza [1].

En caso de que la cantidad de pulsos registrados se encontrase por encima del umbral correspondiente, se declaraba un blanco, situando su coordenada angular con respecto al plano horizontal en el punto medio del intervalo durante el cual se excedió el segundo umbral.

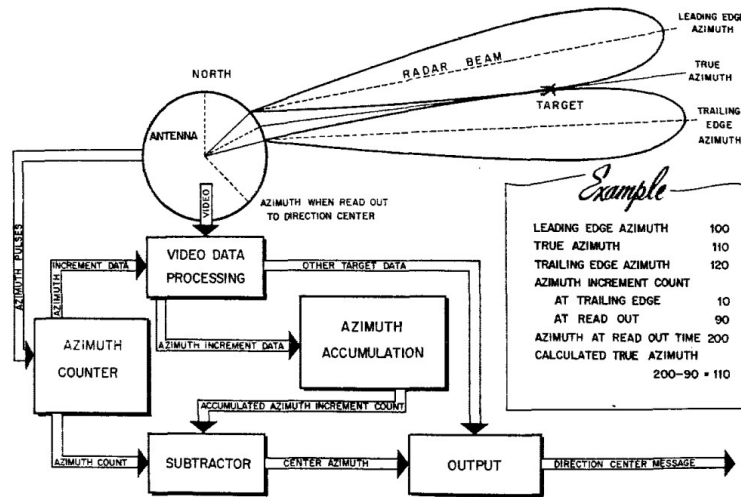


Figura 1.5: Estructura del detector del ventana corrediza [8].

Mientras que el detector de ventana corrediza resolvía el reto asociado a la transmisión de datos, otra innovación surgió de la necesidad de verificar de manera práctica el funcionamiento de los distintos elementos del sistema SAGE. Para este fin se diseñó y construyó el generador de patrones de prueba TS-923/FSQ, un dispositivo que permitía comprobar la operación de los sistemas de entrada de datos de largo alcance, relleno de huecos y transferencia de información entre centros de dirección de combate. Dicho generador tenía la capacidad de emitir señales digitales que representaban objetivos a diferentes distancias y azimuts, siguiendo diversos patrones variables. Este hecho marcó uno de los primeros métodos para evaluar sistemas de radar mediante estímulos simulados, sin requerir detecciones físicas reales.

En la década de los años sesenta comenzaron a desarrollarse las primeras arquitecturas orientadas específicamente al procesamiento digital de señales. En 1967, un grupo de investigadores del *Lincoln Laboratory* diseñó la arquitectura y el conjunto de instrucciones del Procesador Digital Rápido (FDP, por sus siglas en inglés) [1]. Para ello se empleó la computadora de propósito general Univac 1219, a la que se le incorporó una unidad aritmética lógica especializada. Este módulo incluía un sumador restador de 18 bits en complemento a dos, capaz de procesar funciones booleanas, además de incorporar conexiones que facilitaban los cálculos de precisión extendida. La arquitectura también optimizaba las operaciones de acumulación y multiplicación, así como la multiplicación compleja, indispensable para la implementación de la Transformada Rápida de Fourier (FFT, por sus siglas en inglés). De esta manera, la computadora resultante integró una arquitectura mejor adaptada al procesamiento digital de señales, lo que se tradujo en un incremento considerable en el rendimiento de los algoritmos utilizados.

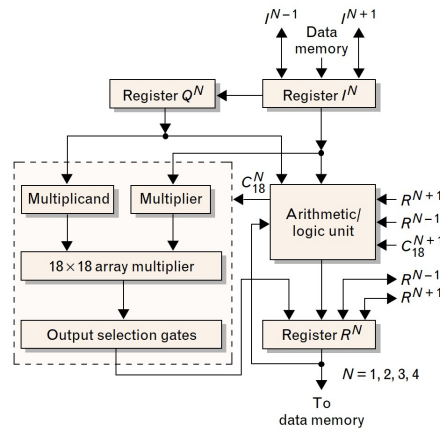


Figura 1.6: Estructura del elemento aritmético del FDP [1].

El FDP resultó de gran utilidad en el procesamiento de señales para radar, ya que, con ciertas modificaciones en su *hardware*, podía cumplir con restricciones de tiempo real. Gracias a ello, se empleó en el desarrollo y la evaluación de técnicas de procesamiento Doppler aplicadas al mapeo de *clutter*. De este trabajo derivó el Indicador de Objetos en Movimiento (MTI, por sus siglas en inglés), un sistema que permitía diferenciar los objetivos móviles del terreno adyacente.

El MTI utilizaba un filtro de supresión de banda para atenuar señales asociadas a objetivos poco relevantes, como el terreno, la lluvia o parvadas de aves. Adicionalmente, este sistema restaba dos pulsos consecutivos, lo que contribuía a la reducción de las detecciones vinculadas a objetos estáticos. En sus primeras versiones, el sistema operaba con una frecuencia de repetición de pulso (PRF, por sus siglas en inglés) fija, lo cual generaba velocidades no detectables en los múltiplos de dicha frecuencia. Para mitigar este problema, se introdujo el uso de una PRF escalonada, que disminuía la cantidad de velocidades ciegas. No obstante, el MTI presentaba limitaciones importantes: producía un número elevado de falsas alarmas en presencia de lluvia y mostraba dificultades para detectar objetivos de baja velocidad.

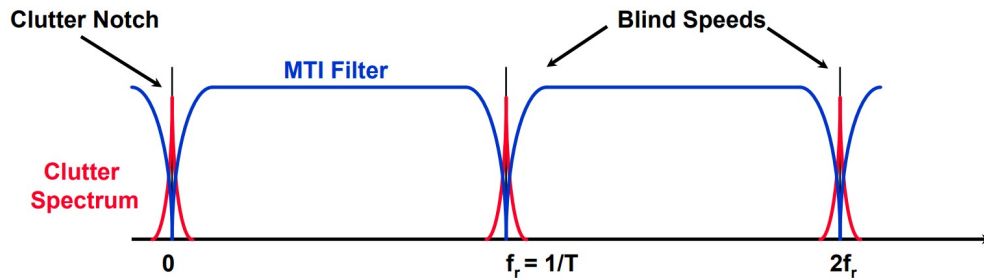


Figura 1.7: Espectro del filtro usado por MTI [9].

En la década de los años setenta, se comenzaría a esparcir el uso de sistemas digitales fuera del ámbito militar. En los Estados Unidos, los avances se empezarían a incorporar en un ámbito civil a petición de la Administración Federal de Aviación (FAA, por sus siglas en inglés). Se buscaba modernizar los sistemas de control aéreo debido a que se presentaban un número significativo de falsas alarmas con la tecnología empleada. En respuesta a ello, el *Lincoln Laboratory* desarrollaría el Detector de Objetos en Movimiento (MTD, por sus siglas en inglés), realizando un prototipo a partir del FDP.

El MTD empleaba múltiples PRF en conjunto con bancos de filtros Doppler en cada compuerta de alcance para cada una de dichas frecuencias. La primera versión del sistema procesaba señales en cuadratura muestreadas a una frecuencia de 2.6 MHz, con una longitud de palabra de 10 bits [10]. Para ello, se almacenaban las muestras correspondientes a diez

pulsos consecutivos en ambos canales, abarcando un total de 760 compuertas de alcance que cubrían 47.5 millas. Posteriormente, se aplicaba un cancelador MTI de tres pulsos, cuya salida era procesada mediante una Transformada Discreta de Fourier (DFT, por sus siglas en inglés) de ocho puntos.

De manera complementaria, se incorporaba un Filtro de Velocidad Cero (ZVF, por sus siglas en inglés) con el fin de mejorar la detección de objetivos lentos. A diferencia del cancelador MTI, este filtro permitía el paso tanto del *clutter* como de objetivos con baja velocidad radial, lo que hacía necesaria la incorporación de un filtro adicional para suprimir el *clutter* proveniente del terreno adyacente. En conjunto, estas características muestran que la primera versión del MTD integraba *hardware* fijo sumamente especializado para el procesamiento digital de señales, lo cual requería personal capacitado en caso de presentar fallas.

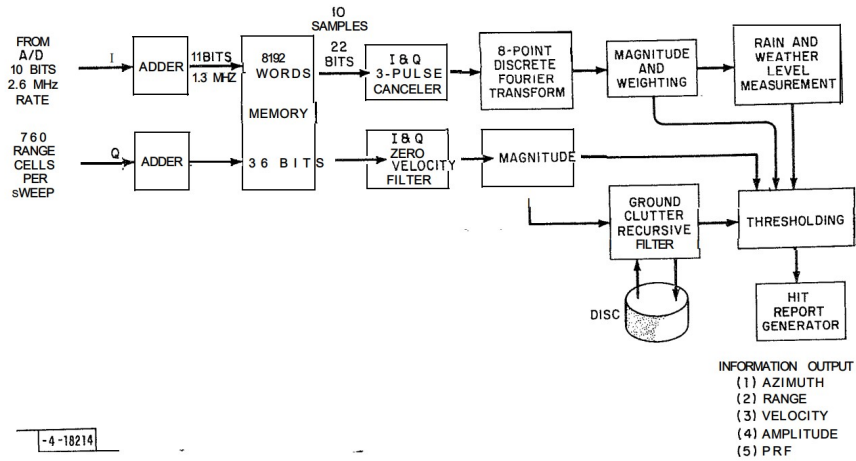


Figura 1.8: Diagrama a bloques de la primera versión del procesador MTD [10].

Por otra parte, se introdujeron modificaciones relevantes en la arquitectura, entre las que destaca la incorporación de la segunda versión del Procesador Paralelo Microprogramado (PMP-2, por sus siglas en inglés). Así mismo, se añadieron algoritmos de umbralización, como la Tasa Constante de Falsas Alarmas (CFAR, por sus siglas en inglés), los cuales permitieron disminuir la incidencia de falsas alarmas sin sacrificar la resolución de las celdas de detección necesarias para la localización de aeronaves.

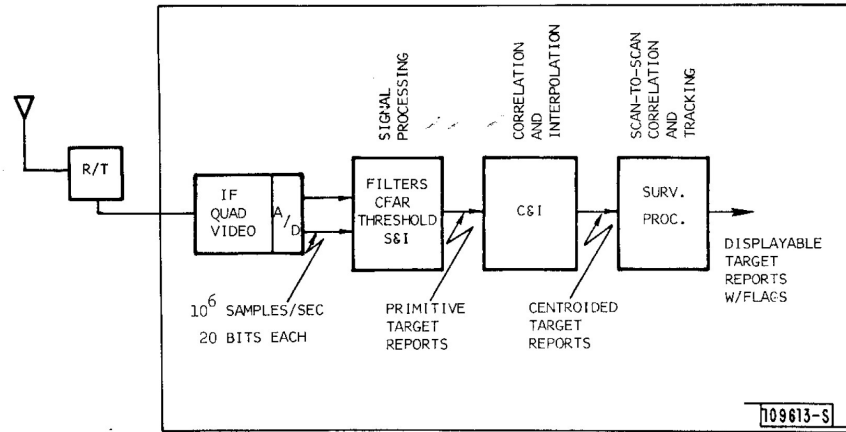


Figura 1.9: Diagrama a bloques de la segunda versión del procesador MTD [11].

El MTD permitió mejorar en gran medida las capacidades de supresión de *clutter* en comparación con el filtrado únicamente mediante MTI. En concreto, los filtros analógicos que se usaban como parte del MTI eran capaces de suprimir el *clutter* hasta en 20 decibelios, mientras que los filtros digitales incorporados en el MTD alcanzaban entre 50 y 60 decibelios de supresión [12].

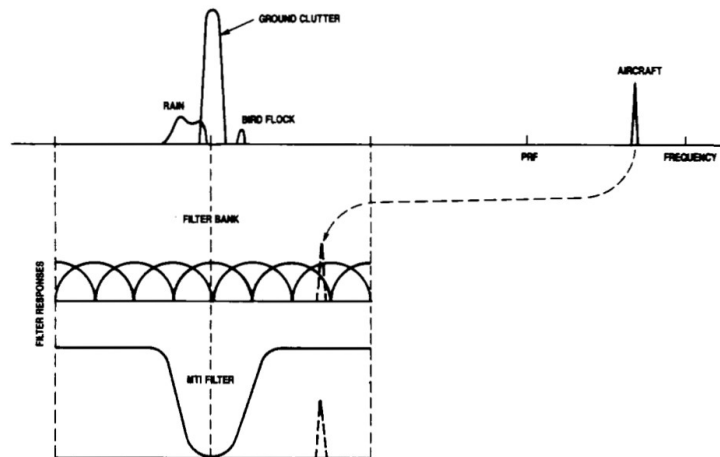


Figura 1.10: Comparativa de filtros usados en MTI y MTD [12].

En 1972, el ejército de los Estados Unidos trabajaría en un radar de estado sólido conceptual en banda L. La naturaleza de dicho radar hacía necesario tener un repertorio amplio de formas de onda para la detección y el seguimiento de objetivos, imposibilitando el uso de filtros analógicos. El procesamiento digital de señales volvió a demostrar su flexibilidad al posicionarse como la solución ideal, eliminando la necesidad de utilizar una gran cantidad

filtros fijos.

A partir de ello, nace el Sistema de Convolución Digital (DCS, por sus siglas en inglés), el cual tenía como propósito realizar filtrado en concordancia con restricciones de tiempo real para un ancho de banda amplio. El DCS incorporaba nuevas técnicas que permitían realizar la convolución de manera más rápida, así como una forma más sofisticada de FFT que implementaba encauzamiento reconfigurable. Adicionalmente, el sistema empleaba un formato de punto flotante híbrido, al igual que una computadora digital de rotación de coordenadas (CORDIC, por sus siglas en inglés) como parte de la FFT. Lo anterior permitió aumentar la eficiencia del procesamiento Doppler de forma considerable, al mismo tiempo que disminuyó el *hardware* requerido.

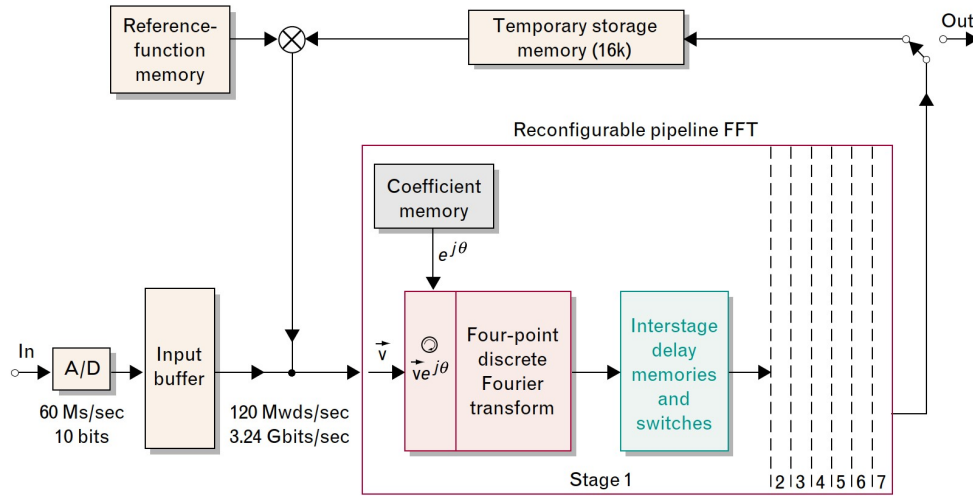


Figura 1.11: Arquitectura del DCS [1].

En la década de los años ochenta, el *Lincoln Laboratory* se encontraba estudiando técnicas de anulación adaptativa para antenas en arreglo de fase que permitieran aminorar los efectos de la interferencia proveniente de los lóbulos laterales de dichas antenas. Dicho fenómeno se puede atenuar por medio de la modificación de los pesos asignados a cada antena que comprende al arreglo. El cómputo de dichos pesos implica un gran número de recursos, dependiendo del tamaño del arreglo utilizado. En 1986, se desarrolló la computadora Experimento Sistólico de Actualización de Matrices (MUSE, por sus siglas en inglés) capaz de llevar a cabo el cálculo adaptativo al reducir el cómputo a una serie de rotaciones simples por medio de CORDIC.

Otro avance importante que se llevó a cabo en esta década fue el desarrollo de sistemas

que incorporasen formadores de haz digital (DBF, por sus siglas en inglés). En general, los formadores de haz aprovechan la interferencia que se genera entre cada elemento de un arreglo de antenas en fase para controlar el patrón de radiación y formar un haz principal. A partir de ello, se puede dirigir el haz principal en una dirección deseada al variar los pesos que se le dan a cada elemento del arreglo, variando su amplitud y fase.

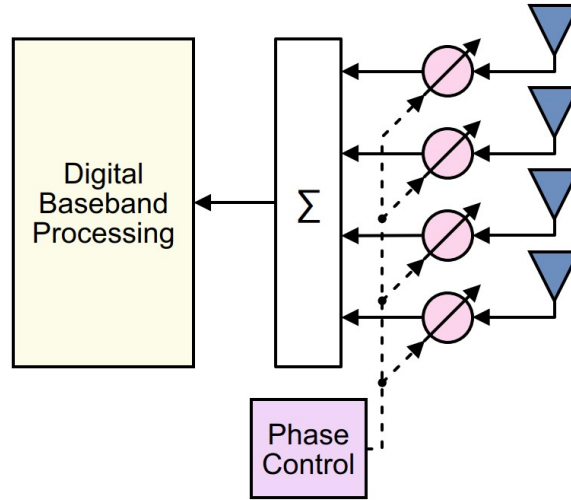


Figura 1.12: Implementación convencional de un DBF [13].

En conjunto con las técnicas de DBF, surge el concepto del Radio Definido por Software (SDR, por sus siglas en inglés) que implica reemplazar el *hardware* dedicado en sistemas de radio por algoritmos de procesamiento digital de señales implementados por medio de *software* o *hardware* programable [14]. Las primeras arquitecturas de SDR comenzaron a desarrollarse en los años noventa con la aparición de Convertidores Analógico/Digital (ADC, por sus siglas en inglés) con una mayor tasa de muestreo y una mayor longitud de palabra. Estos primeros sistemas hacían uso de múltiples etapas de conversión analógica, las cuales permitían modificar la frecuencia de las señales entre radiofrecuencia (RF, por sus siglas en inglés), frecuencia intermedia (IF, por sus siglas en inglés) y banda base.

Durante esta misma década se comenzó a incorporar la tecnología de estado sólido en los sistemas de radar. El uso de componentes de estado sólido permitió desarrollar sistemas más livianos y compactos, al igual que aumentar su fiabilidad y abaratar su costo [15]. Esta tecnología hizo posible la transmisión de pulsos con mayor duración, a partir de lo cual se logra una mayor potencia promedio a pesar de tener una potencia de pico menor. Sin embargo, un incremento en el ancho de pulso resulta en una menor resolución de alcance. Por consiguiente, se requiere modificar las características del pulso emitido, de tal forma que se

evite la pérdida excesiva de resolución. La manera más común de compensar esta limitación es a través de la modulación en frecuencia o fase del pulso transmitido. Posteriormente, con el fin de detectar la presencia de la señal reflejada, se aplica un filtro que correlaciona la señal suministrada con la señal transmitida. En caso de que exista correlación, el filtro entrega a la salida un pulso con menor duración y mayor amplitud, por lo que se suele referir a esta técnica como compresión de pulso. Lo anterior hace posible distinguir objetivos cercanos entre sí, aumentando efectivamente la resolución de alcance.

Entre 1992 y 1993, se desarrolló un sistema demostrativo a partir de la novena versión del Radar de Vigilancia de Aeropuertos (ASR-9, por sus siglas en inglés) que incorporó un transmisor de estado sólido en conjunto con técnicas de compresión digital de pulso. Originalmente, el ASR-9 operaba en banda S y su sistema de transmisión empleaba un klístron a partir de lo cual se generaba un pulso con una duración de un microsegundo [16]. El sistema demostrativo adicionó un pulso más largo de 75 microsegundos con su propia etapa dedicada de compresión, implementada mediante un filtro en concordancia [17]. Este sistema logró mostrar el potencial de la tecnología de estado sólido en conjunto con las técnicas digitales de compresión de pulso al lograr una disminución considerable de los lóbulos laterales. Sin embargo, las pruebas que involucraban la resolución de alcance no fueron del todo satisfactorias. Esto se atribuyó, en parte, al *hardware* usado por el procesador del ASR-9 al manejar una longitud de palabra de 12 bits, mientras que el sistema de demostración operaba con 18 bits.

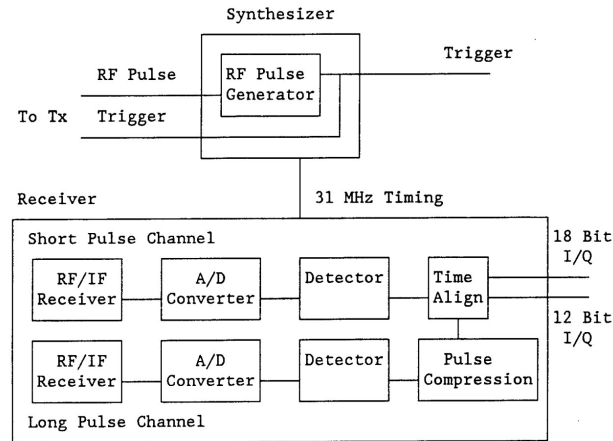


Figura 1.13: Diagrama a bloques del sistema demostrativo desarrollado [16].

Durante este mismo periodo, se comenzó a popularizar el uso de los arreglos de compuertas programables en campo (FPGA, por sus siglas en inglés), lo cual tuvo un impacto

muy significativo en el procesamiento de señales para radar. Los FPGA permiten describir sistemas digitales a la medida que pueden cambiar su configuración de manera parcial o total. Adicionalmente, los FPGA cuentan con *hardware* dedicado que permite realizar las operaciones asociadas al procesamiento digital de señales de forma más eficiente.

A principios del siglo XXI, los avances en la tecnología digital hicieron posible implementar las etapas de conversión entre frecuencia intermedia y banda base en sistemas digitales reconfigurables, como los FPGA. Este tipo de arquitecturas introdujo la capacidad de contar con etapas de sintonización adaptables, lo que permitió que los sistemas de radar modernos incrementaran su capacidad de respuesta al poder ajustarse en tiempo real a las condiciones específicas de operación.

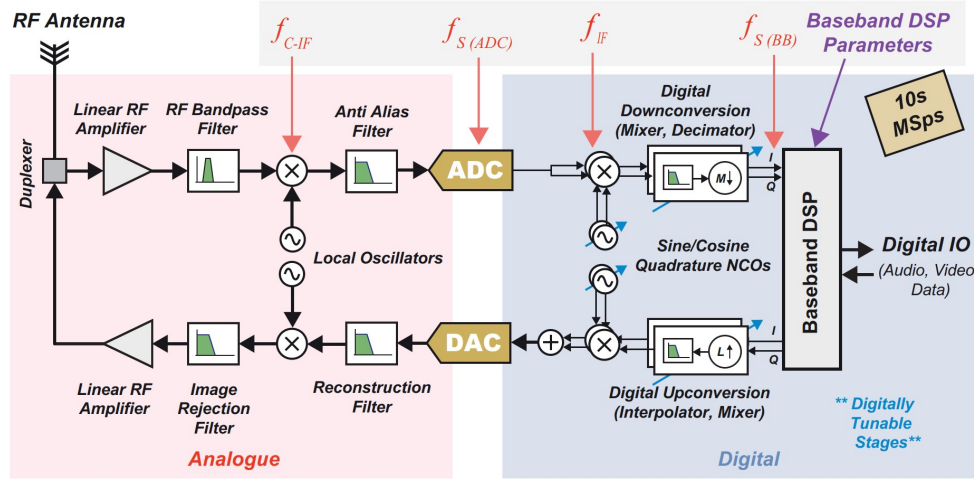


Figura 1.14: Arquitectura de SDR con muestreo digital en frecuencia intermedia [14].

1.2. Planteamiento de la propuesta a desarrollar

La migración de los sistemas de radar hacia la tecnología digital ha permitido el desarrollo de arquitecturas cada vez más robustas. No obstante, este avance ha traído consigo un incremento significativo en la complejidad de los mismos. A partir de ello, ha surgido un interés particular en establecer metodologías de diseño que posibiliten el análisis del comportamiento de dichos sistemas sin necesidad de disponer de su implementación física. En [18], Waqar et al. presentan un sistema orientado a la generación de datos para el reconocimiento de actividad humana (HAR, por sus siglas en inglés), basado en un radar de múltiples entradas y múltiples salidas (MIMO, por sus siglas en inglés), utilizando herramientas de *software* especializado. Este trabajo demuestra la viabilidad de estudiar la respuesta de un sistema

de radar frente a una amplia variedad de estímulos y condiciones mediante simulaciones. En consecuencia, existe un interés creciente en el desarrollo de modelos que faciliten el análisis y la evaluación de distintas arquitecturas de radar de manera práctica y flexible.

En este contexto, surge la necesidad particular de establecer un marco metodológico que facilite las tareas de modelado y análisis del procesamiento asociado al bloque de recepción de los sistemas de radar pulsado que operan mediante arreglos de antenas en fase. En respuesta a esta necesidad, el presente trabajo pretende desarrollar modelos que permitan generar, acondicionar y procesar las señales recibidas por un sistema de radar pulsado a través de un arreglo de antenas en fase. Este trabajo hace énfasis en el procesamiento digital de señales, delimitando las técnicas empleadas a la formación de haces, la compresión de pulso y la estimación de velocidad radial. Finalmente, se busca instrumentar el módulo de compresión de pulso en un FPGA.

1.3. Objetivos

1.3.1. Objetivo general

Desarrollar el modelado del procesamiento asociado al bloque de recepción de un sistema de radar pulsado mediante la creación de módulos que simulen la generación, el acondicionamiento y el procesamiento digital de las señales recibidas por un arreglo lineal uniforme de antenas en fase. Dicho procesamiento incluye la formación de haces, la compresión de pulso y la estimación de velocidad radial. Además, se plantea instrumentar el módulo de compresión de pulso en un FPGA.

1.3.2. Objetivos específicos

- Desarrollo de un modelo que permita generar las señales recibidas por un arreglo lineal de antenas en fase, al igual que el acondicionamiento pertinente previo a su digitalización.
- Desarrollo de un modelo del procesamiento digital de señales requerido para la formación de haces, la compresión de pulso y la estimación de velocidad radial.
- Desarrollo de la instrumentación del módulo de compresión de pulso para dicho sistema en un FPGA.

1.4. Conclusiones

La migración de los sistemas de radar hacia el ámbito digital ha expandido significativamente sus capacidades. Esta transición ha permitido superar limitaciones propias de los sistemas analógicos, como la carencia de medios de almacenamiento para las muestras. Al mismo tiempo, la adopción de tecnología digital ha aportado una gran flexibilidad a técnicas ya establecidas, como lo son la formación de haces, la compresión de pulso y la estimación de velocidad radial.

Los primeros sistemas digitales emplearon computadoras de propósito general que, si bien fueron de gran utilidad para manejar el enorme volumen de datos en sistemas como SAGE, carecían de circuitos digitales especializados para agilizar el procesamiento. Con el incremento en la complejidad de los sistemas de radar, surgió la necesidad de desarrollar las arquitecturas de procesamiento especializadas que optimizaran el cómputo de datos. Esto desembocó en la aparición de procesadores como el FDP y los MTD, que incorporaban diseños cada vez más sofisticados para el procesamiento digital de señales. De igual forma, el surgimiento de sistemas reconfigurables, como los FPGA, ha introducido un alto grado de adaptabilidad, permitiendo que conceptos como SDR florezcan y ofrezcan soluciones dinámicas y altamente responsivas en ámbitos como las telecomunicaciones.

Ante el crecimiento exponencial en la complejidad de los sistemas digitales de radar, aparece la necesidad de desarrollar modelos que permitan analizar el procesamiento asociado. En este contexto, el presente trabajo tiene como objetivo establecer un modelo de primera aproximación para el procesamiento correspondiente a la etapa de recepción de un sistema de radar pulsado. Dicho modelo abarca la formación de haces, la compresión de pulso y la estimación de velocidad radial. A partir de dicho modelo, se busca desarrollar la instrumentación del módulo de compresión de pulso en un FPGA.

Capítulo 2

Marco teórico

En este capítulo se presentan los conceptos fundamentales de sistemas de radar, arreglos de antenas en fase y técnicas relevantes de procesamiento digital de señales que serán mencionados con frecuencia. Lo anterior pretende ofrecer una base teórica que permita al lector comprender los aspectos importantes del modelado y de la instrumentación subsecuente.

2.1. Sistemas de radar

El término radar proviene del acrónimo en inglés de *Radio Detection and Ranging* que se traduce al español como detección y localización por radio. En este contexto, el término radio hace referencia a las ondas del espectro electromagnético cuya frecuencia se encuentra entre 3 kHz y 300 GHz. El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, por sus siglas en inglés) divide dicho intervalo de frecuencias en bandas para radar a partir de 3 MHz como se muestra en la figura 2.1. Así mismo, existen otros organismos como la Unión Internacional de Telecomunicaciones (ITU, por sus siglas en inglés) y la Organización del Tratado del Atlántico Norte (OTAN) que cuentan con su propia nomenclatura para las bandas de frecuencia. A lo largo de este trabajo, se hará uso de la nomenclatura IEEE según lo estipulado en su estándar 521-2019.

Radar nomenclature		ITU nomenclature			
Radar letter designation	Frequency range	Frequency range	Band No.	Adjectival band designation	Corresponding metric designation
HF	3 MHz to 30 MHz	3 MHz to 30 MHz	7	High frequency (HF)	Decametric waves
VHF	30 MHz to 300 MHz	30 MHz to 300 MHz	8	Very high frequency (VHF)	Metric waves
UHF	300 MHz to 1000 MHz	0.3 GHz to 3 GHz	9	Ultra high frequency (UHF)	Decimetric waves
L	1 GHz to 2 GHz				
S	2 GHz to 4 GHz				
C	4 GHz to 8 GHz	3 GHz to 30 GHz	10	Super high frequency (SHF)	Centimetric waves
X	8 GHz to 12 GHz				
K _u	12 GHz to 18 GHz				
K	18 GHz to 27 GHz	30 GHz to 300 GHz	11	Extremely high frequency (EHF)	Millimetric waves
K _a	27 GHz to 40 GHz				
V	40 GHz to 75 GHz				
W	75 GHz to 110 GHz				
mm	110 GHz to 300 GHz	300 GHz to 3000 GHz	12	None	Decimillimetric

Figura 2.1: Bandas de frecuencia de acuerdo al estándar IEEE 521-2019 [19].

Los sistemas de radar hacen uso de señales electromagnéticas dentro de una banda de frecuencia determinada con el fin de detectar y localizar de objetos en el espacio. La transmisión se realiza por medio de antenas que convierten señales eléctricas en ondas electromagnéticas que se propagan en el espacio libre. Al incidir sobre un objeto, las ondas se reflejan y, posteriormente, son captadas por una antena receptora, a partir de lo cual se puede determinar diferentes parámetros del objeto como su dirección, rango y velocidad radial.

2.1.1. Clasificación los sistemas de radar

De acuerdo con [20], existen tres funciones principales a partir de las cuales los sistemas de radar se pueden categorizar: búsqueda, seguimiento y formación de imágenes. Los radares de búsqueda se encargan de detectar la presencia de objetos en un espacio determinado. Posteriormente, los radares de seguimiento rastrean el objeto detectado a partir de su dirección, rango y velocidad, estimando su posición en función del tiempo. Por último, los radares de formación de imágenes permiten obtener características muy puntuales del objeto o pueden ser usados para generar imágenes de un área determinada.

Así mismo, se pueden clasificar a los sistemas de radar según la emisión continua o pulsada de ondas electromagnéticas. Los sistemas de radar de Onda Continua (CW, por sus siglas en inglés) suelen emplear una configuración biestática, lo cual implica el uso de antenas

diferentes para la transmisión y la recepción. Los sistemas de radar pulsado pueden incorporar una configuración monoestática, en donde se hace uso de una misma antena para transmisión y recepción, o biestática.

Se contempla modelar un sistema de radar pulsado por lo que es menester abordar sus características más relevantes. Por su naturaleza, los sistemas de radar pulsado cuentan con una Frecuencia de Repetición del Pulso (PRF) que indica el número de pulsos transmitidos por unidad de tiempo. Otra característica importante es el ciclo de trabajo del pulso que se encuentra definido como la relación entre la duración o ancho de pulso y el intervalo de tiempo entre cada pulso.

2.1.2. Sistema de coordenadas para radar

En este trabajo se hará énfasis en un sistema de radar pulsado monoestático por lo que es necesario ahondar en su funcionamiento. Este tipo de radar hace uso de coordenadas esféricas para describir la posición de un objeto, situando a la antena como el origen del sistema de coordenadas como se aprecia en la figura 2.2.

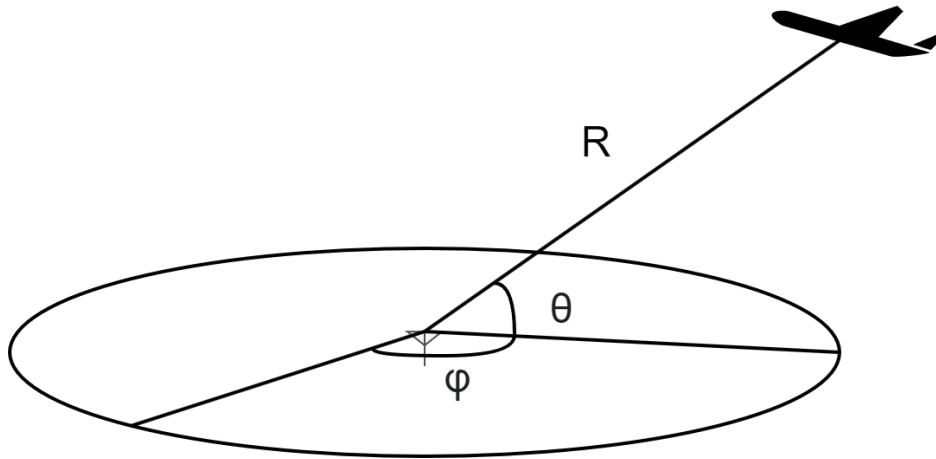


Figura 2.2: Sistema de coordenadas de radar.

En este sistema de coordenadas, R representa la distancia al objetivo, la cual se puede obtener a partir del tiempo que le toma a una onda transmitida en ser recibida por la misma antena posterior a que se refleja del objeto. Así mismo, las dos coordenadas restantes representan la dirección del objetivo a través de los ángulos ϕ y θ . Al ángulo θ se le conoce como ángulo de elevación y permite ubicar un objeto a lo largo del plano xz . Por otro lado, al ángulo ϕ se le conoce como ángulo de azimut y permite ubicar un objeto a lo largo del plano xy .

Como primera aproximación, la distancia al objetivo puede ser obtenida si se conoce la velocidad de propagación de la onda, así como la diferencia de tiempo que existe entre su transmisión y su recepción. Al tratarse de ondas electromagnéticas, se asume que la velocidad de propagación será la velocidad de la luz. Aunado a esto, la onda realiza un viaje de ida y regreso al objetivo por lo que recorre el doble de la distancia en la diferencia de tiempo registrada. A partir de ello, se obtiene la siguiente expresión:

$$R = \frac{c \cdot t_R}{2} \quad (2.1)$$

Dónde R es la distancia al objetivo, c es la velocidad de la luz y t_R es la diferencia de tiempo entre transmisión y recepción de la onda.

2.1.3. Estructura general del sistema de radar monoestático

De forma general, los sistemas de radar monoestáticos se encuentran compuestos conforme al esquema mostrado en la figura 2.3. En este tipo de sistemas, se cuenta con un circulador que permite hacer uso de una misma antena para la transmisión y recepción.

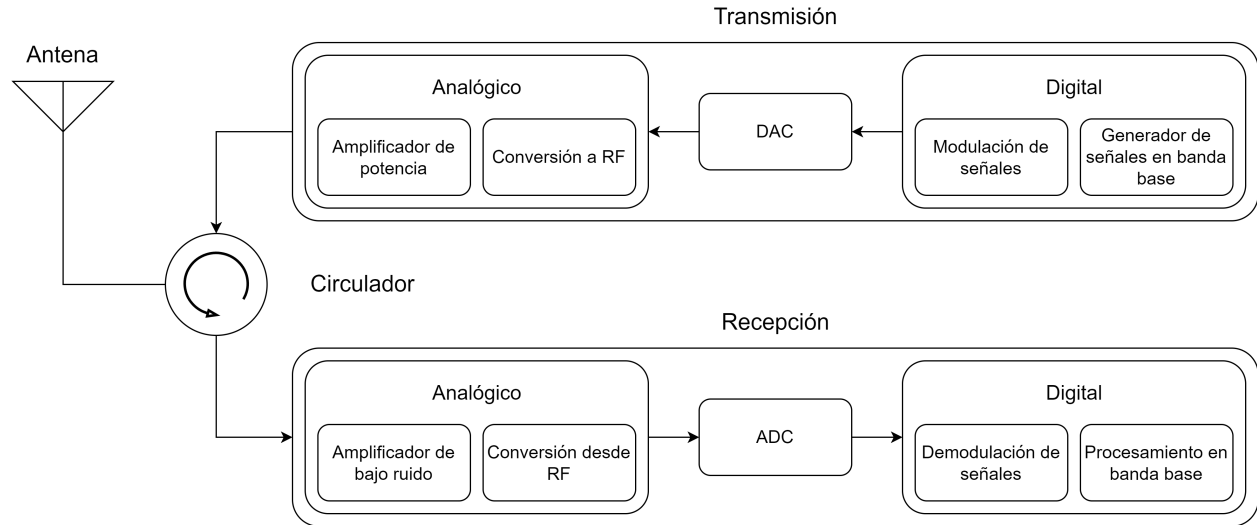


Figura 2.3: Diagrama a bloques de un sistema monoestático.

Por un lado, el sistema de transmisión cuenta con una etapa de generación y modulación digital de señales. Esta primera etapa se encarga de producir la señal a emitir con las características deseadas; esto puede incluir diferentes esquemas de modulación en amplitud, frecuencia o fase.

A continuación, se hace pasar la señal por un convertidor digital-analógico (DAC, por sus

siglas en inglés) previo a la etapa de conversión a RF. Este dispositivo permite generar una señal analógica a partir de la frecuencia de muestreo y la cuantización de una señal digital [22].

A partir de la señal analógica obtenida, se necesita trasladar la frecuencia de la señal a transmitir hasta RF. Dicho aumento en frecuencia permite reducir las dimensiones de la antena, dado que estas se encuentran determinadas por la longitud de onda de la señal emitida.

Para llevar a cabo este proceso, es necesario hacer uso de circuitos especializados denominados mezcladores, los cuales permiten obtener la suma y la diferencia de las frecuencias correspondientes a las señales suministradas. Aunado a esto, se requiere contar con un oscilador local, circuito electrónico que permite generar una señal sinusoidal en la banda de RF hacia donde se trasladará la señal a transmitir. Los mezcladores multiplican una señal de entrada $v_i(t)$ por la señal sinusoidal generada por el oscilador local $v_{LO}(t)$, dando como resultado una señal sinusoidal $v_o(t)$ en donde aparecen los denominados términos mezcla que contienen la suma y la diferencia de las frecuencias de las señales de entrada.

$$v_i(t) = A_i \cos(\omega_i t) \quad v_{LO}(t) = A_{LO} \cos(\omega_{LO} t)$$

$$v_o(t) = \frac{A_{LO} A_i}{2} \{ \cos([\omega_{LO} - \omega_i]t) + \cos([\omega_{LO} + \omega_i]t) \} \quad (2.2)$$

Donde A_{LO} y A_i son las magnitudes de la señal del oscilador local (LO, por sus siglas en inglés) y la señal de entrada, respectivamente. Por otro lado, ω_{LO} y ω_i representan las frecuencias angulares de cada una de las señales antes mencionadas. En el contexto de la transmisión, solo resulta de interés el término correspondiente a la suma, por lo que se usa un filtro para eliminar el término no deseado.

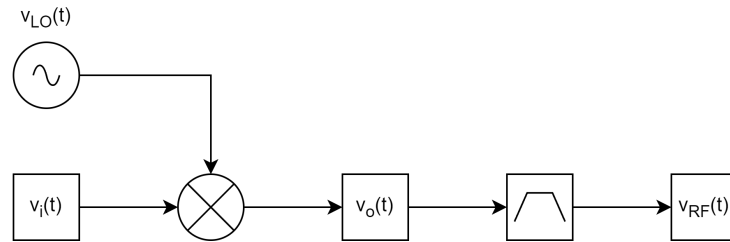


Figura 2.4: Mezclador de conversión.

Por último, solo resta amplificar la señal mediante un amplificador de potencia, previo a su transmisión por la antena. Estos dispositivos permiten incrementar el nivel de potencia

de la señal de RF hasta el requerido, en función de la aplicación y de las características de la antena [23]. La señal resultante se conduce al circulador, el cual la dirige hacia la antena, donde es irradiada en forma de ondas electromagnéticas.

Las ondas electromagnéticas generadas se propagan por el espacio y, al incidir sobre un objeto, son reflejadas de regreso hacia el sistema de radar. La antena del sistema recibe dichas ondas y las convierte en señales eléctricas, que son conducidas nuevamente a través del circulador y posteriormente dirigidas hacia el sistema de recepción.

Este sistema cuenta con una etapa de acondicionamiento que incorpora un amplificador de bajo ruido, el cual permite incrementar el nivel de la señal. Dicha amplificación es esencial ya que las antenas suelen captar señales muy atenuadas, por lo que es necesario suministrar una ganancia determinada, teniendo especial cuidado en no introducir ruido que pudiese afectar el resto del procesamiento.

Después, es necesario convertir la señal amplificada de RF a la frecuencia requerida por medio de un mezclador. El principio de funcionamiento es exactamente el mismo al que se describió en la transmisión a excepción de la señal de interés; por lo general, se desea obtener únicamente el término con la diferencia de frecuencias al que se le denomina como frecuencia intermedia (IF). Con base en dichos requerimientos, se utiliza un filtro paso banda a la salida del mezclador, contando con un ancho de banda que permita obtener únicamente la señal de interés.

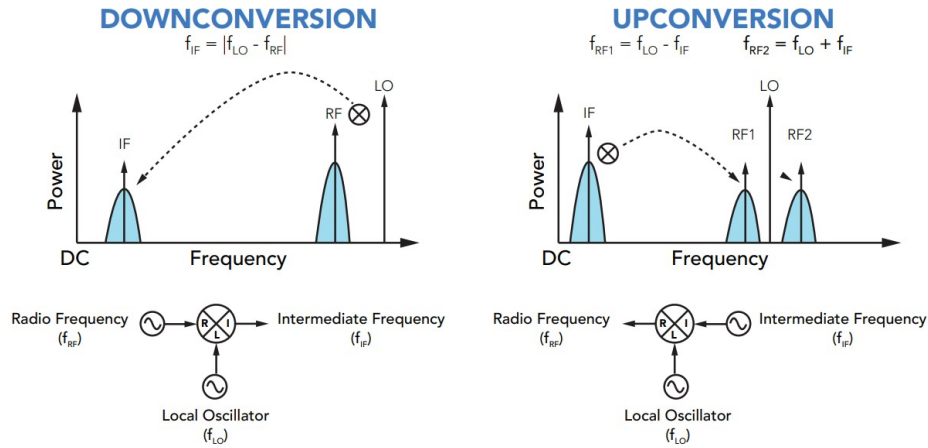


Figura 2.5: Mezcladores de conversión [24].

El siguiente componente del sistema de recepción es el ADC, encargado de digitalizar la señal analógica. Estos dispositivos poseen una longitud de palabra definida como el número de *bits* disponibles para representar una cantidad en sistema binario. A partir de este parámetro

y de un voltaje de referencia se determina la resolución del ADC, la cual establece el valor mínimo de voltaje asociado al *bit* menos significativo (LSB, por sus siglas en inglés).

A continuación, se lleva a cabo la demodulación de las señales digitales recibidas. Uno de los esquemas más utilizados es la demodulación de señales en cuadratura. Dicho esquema es equivalente a multiplicar la señal por un exponencial complejo, empleando dos osciladores controlados numéricamente (NCO, por sus siglas en inglés), los cuales cuentan con la misma frecuencia de operación f_{LO} y una diferencia de fase entre sí de 90° [25]. Dado que la señal es digital, se requiere normalizar su frecuencia tal y como se muestra en 2.3, tomando como referencia la frecuencia de muestreo del sistema digital.

$$\Omega_{LO} = \frac{2\pi f_{LO}}{f_s} \quad (2.3)$$

Donde Ω_{LO} es la frecuencia angular discreta y f_s es la frecuencia de muestreo del sistema digital.

En primera instancia, se multiplica de forma paralela la señal real digitalizada recibida en cada elemento del arreglo, denominada $r_n[n]$, con los NCO en cuestión, tal y como se muestra en la figura 2.6. A la señal resultante de la multiplicación de $r_n[n]$ con una señal coseno de frecuencia Ω_{LO} , se le conoce como la componente en fase. Por otro lado, a la señal resultante de la multiplicación de $r_n[n]$ con una señal seno de frecuencia Ω_{LO} , se le conoce como la componente en cuadratura.

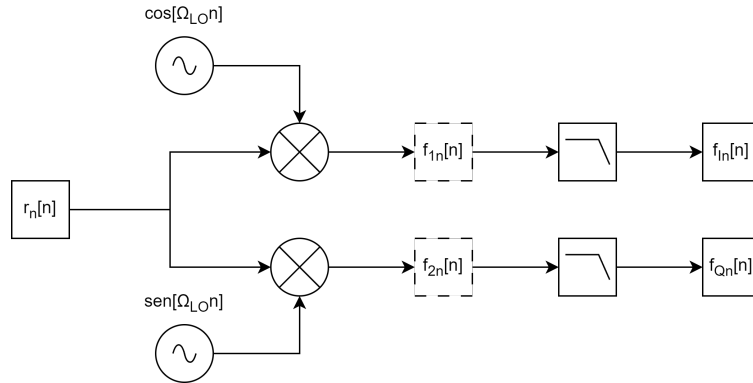


Figura 2.6: Demodulación de señales en cuadratura.

En este tipo de demoduladores, la frecuencia del NCO debe ser la misma que la frecuencia de la señal de IF. En consecuencia, las señales resultantes $f_{1n}[n]$ y $f_{2n}[n]$ contienen las componentes en banda base de interés, al igual que otros términos con el doble de frecuencia Ω_{LO} . La aparición de términos no deseados hace necesario la incorporación de un filtro paso

bajas que permita recuperar las componentes $f_{In}[n]$ y $f_{Qn}[n]$ correspondientes.

A partir de las señales obtenidas en banda base se puede llevar a cabo, mediante un procesamiento digital, la identificación del ángulo de elevación a partir de la formación de haz de las señales reflejadas, así como la estimación de la distancia al objeto mediante la compresión de pulso y de su velocidad radial a través del efecto Doppler.

2.1.4. Arquitecturas de recepción

Con base en la descripción general de los sistemas de radar monoestáticos, resulta pertinente profundizar en el bloque de recepción. Para ello, es necesario examinar las principales arquitecturas de recepción que deben considerarse para el modelado del sistema.

Por un lado, se cuenta con los receptores homodinos, los cuales convierten directamente la señal recibida desde RF hasta banda base. Este tipo de arquitecturas se componen de un filtro paso banda en la entrada, seguido de un amplificador de bajo ruido y, finalmente, una única etapa de demodulación en cuadratura en donde la frecuencia angular de los osciladores locales ω_{LO} debe ser igual a la frecuencia angular de la señal recibida ω_{in} .

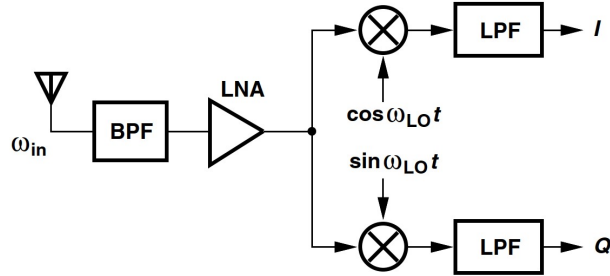


Figura 2.7: Receptor homodino o de conversión directa [26].

En principio, los receptores homodinos parecen ofrecer una solución idónea; no obstante, la ausencia de una frecuencia intermedia exige disponer de un sistema digital capaz de operar directamente en RF. Estos sistemas suelen implicar un costo elevado y requieren metodologías de diseño adaptadas a los fenómenos presentes en el rango de frecuencias correspondiente. Como se puede apreciar en [27], una arquitectura de esta naturaleza requiere, entre otras cosas, un ADC con una tasa de muestreo en el orden de miles de millones de muestras por segundo.

Por otro lado, se cuenta con los receptores heterodinos, los cuales incorporan etapas de conversión desde RF hacia IF y, posteriormente, una etapa de conversión a banda base. El

uso de una frecuencia intermedia facilita el diseño de circuitos con mayor selectividad; sin embargo, esta elección conlleva una consideración sumamente relevante que debe ser tomada en cuenta, la aparición de la denominada frecuencia imagen.

Durante el proceso de conversión hacia la frecuencia intermedia, aparece una componente espectral cuya separación respecto al oscilador local es idéntica a la existente con la frecuencia de interés. A esta componente se le conoce como la frecuencia imagen de la frecuencia intermedia. En consecuencia, al pasar por un mezclador, la frecuencia imagen se traslada a la misma frecuencia intermedia que la señal de interés de RF, lo que conduce a que ambas resulten indistinguibles.

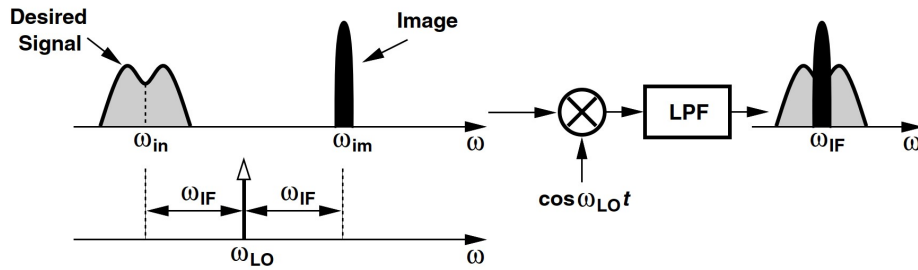


Figura 2.8: Conversión a IF de la señal de interés con su frecuencia imagen [26].

Esto resulta problemático, ya que la frecuencia imagen no contiene información útil y únicamente genera interferencia. Para mitigar este fenómeno, es necesario colocar un filtro paso banda a la entrada de cada etapa de conversión con el fin de suprimir la frecuencia imagen. De igual forma, a la salida se dispone de un amplificador sintonizado a la frecuencia intermedia seleccionada, encargado de proporcionar la ganancia requerida.

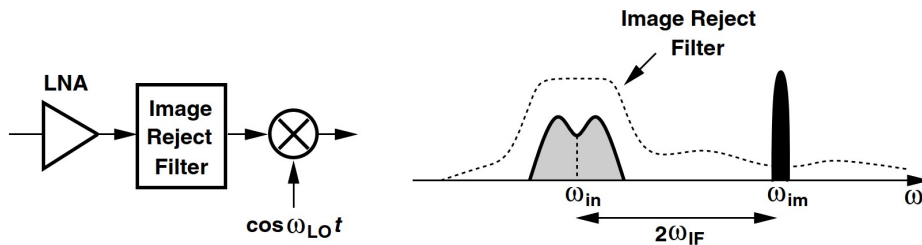


Figura 2.9: Filtro de rechazo de imagen [26].

Las arquitecturas para receptores heterodinos suelen emplear dos o tres etapas de conversión, lo que permite encontrar un balance entre selectividad y rechazo de la frecuencia imagen. La primer etapa de conversión suele contar con una frecuencia intermedia alta; esto permite rechazar de forma más eficiente la frecuencia imagen al existir una mayor diferencia

con respecto a la componente de interés. En consecuencia, el filtro paso banda situado a la entrada puede tener un ancho de banda más amplio, lo que simplifica su diseño. Las etapas siguientes de conversión suelen emplear frecuencias intermedias bajas, facilitando el diseño de circuitos más selectivos.

2.2. Antenas

Una vez que se han expuesto las generalidades de los sistemas de radar monoestáticos y se han examinado las principales arquitecturas de recepción, es necesario ahondar en aquellos elementos que permiten la recepción de señales en primer lugar.

Las antenas son dispositivos que permiten convertir señales eléctricas en ondas electromagnéticas, las cuales serán emitidas en el espacio libre y viceversa, es decir, pueden recibir ondas electromagnéticas y convertirlas en señales eléctricas. La propagación de ondas electromagnéticas se encuentra descrita por las ecuaciones de Maxwell según se muestra en [28]. A partir de las características propias de la propagación, se forma un conjunto de puntos en el espacio para los cuales la onda cuenta con una misma fase; a este conjunto de puntos se le conoce como frente de onda.

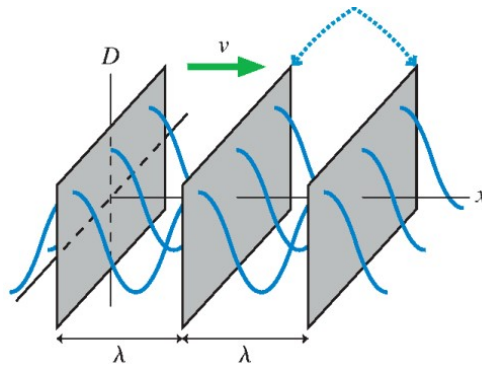


Figura 2.10: Frente de onda plano [29].

Idealmente, las ondas se propagan de forma uniforme en todas direcciones, dando lugar al radiador isotrópico. Sin embargo, en la práctica esto no es posible, aunque se usa como referencia para definir ciertos parámetros de las antenas.

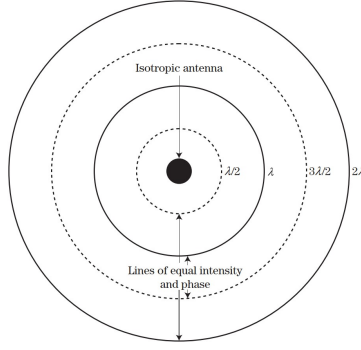


Figura 2.11: Propagación de las ondas para una antena isotrópica [30].

El primer parámetro a mencionar es el patrón de radiación, que se define como la distribución espacial de la potencia radiada por la antena en el espacio. Esta distribución angular del campo radiado se analiza generalmente en la región lejana, definida como la zona del espacio donde dicha distribución permanece constante a pesar de la distancia. En dicha región, las ondas propagadas pierden su fase y se consideran como ondas planas.

El patrón de radiación cuenta con regiones espaciales denominadas lóbulos, las cuales se caracterizan por contar con una mayor intensidad de potencia radiada. A la región que contiene la dirección con mayor radiación se le conoce como lóbulo principal. Al resto de regiones con menor potencia radiada se les conoce como lóbulos laterales.

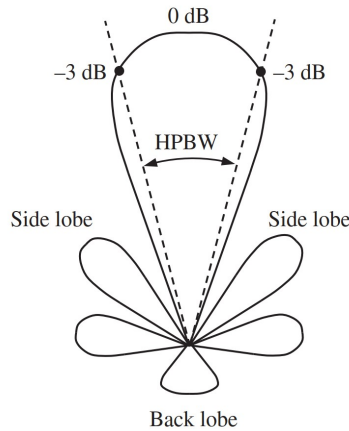


Figura 2.12: Lóbulos del patrón de radiación [31].

El ancho del lóbulo principal se define a partir del sector angular en el cual la potencia radiada ha caído 3 dB [32]. Otra característica a resaltar dentro del patrón de radiación es el nivel de los lóbulos laterales, el cual se procura mantener lo más reducido posible.

Existen otros parámetros fundamentales de las antenas entre los que se encuentra la

directividad y la ganancia. La directividad se puede definir como el número de veces que necesita aumentarse la potencia radiada al sustituir una antena direccional por una antena ideal isotrópica con el fin de mantener la magnitud del vector de Poynting constante en el punto de observación.

$$D = \frac{P_{\Sigma isotr}}{P_{\Sigma}} \quad (2.4)$$

Donde D representa la directividad de la antena, $P_{\Sigma isotr}$ es la potencia radiada por la antena ideal isotrópica y P_{Σ} es la potencia radiada por la antena direccional.

La ganancia se define de forma similar pero se usa la potencia suministrada a la antena, por lo que se toman en cuenta sus pérdidas, tal y como se muestra en la expresión (2.5).

$$G = \frac{P_{\Sigma isotr}}{P_{\Sigma} + P_P} \quad (2.5)$$

Donde G representa la ganancia de la antena y P_P es la potencia de pérdidas óhmicas en la antena.

En este contexto, se puede definir el rendimiento de la antena direccional como la relación entre potencia radiada y potencia suministrada. A partir de ello, se puede obtener una expresión equivalente de la ganancia, definiendo este parámetro como el producto de la directividad por el rendimiento de la antena.

2.2.1. Antenas en arreglo de fase

En este contexto, es posible formar un arreglo a partir de un conjunto de antenas individuales, denominadas elementos, dispuestas a cierta distancia entre sí en un patrón geométrico determinado. Cada elemento del arreglo emite simultáneamente ondas electromagnéticas a la misma frecuencia, lo que provoca que existan regiones en el espacio en donde las fases de dichas ondas coinciden en mayor o menor medida. En el caso de que las fases coincidan, se obtiene una onda de mayor amplitud, por lo que se habla de interferencia constructiva. En caso contrario, la diferencia de fases provoca una reducción significativa en la amplitud de la onda, por lo que se le conoce como interferencia destructiva.

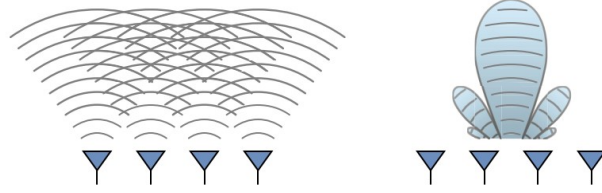


Figura 2.13: Patrón de interferencia del arreglo [13].

La distribución espacial del arreglo no es la única forma de manipular el patrón de interferencia resultante. Al suministrar señales con cierta diferencia de fase entre sí a cada elemento del arreglo, se puede dirigir de forma electrónica el haz o lóbulo principal en una dirección determinada. La diferencia de fase entre elementos se define en función de la distancia entre cada elemento y de la longitud de onda.

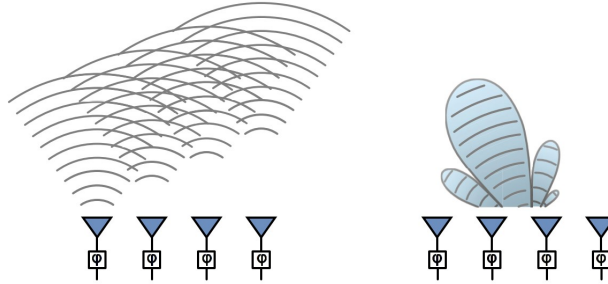


Figura 2.14: Patrón de interferencia al aplicar diferencia de fase [13].

2.2.2. Arreglos lineales uniformes

La configuración espacial más simple para un arreglo de antenas en fase es la correspondiente al arreglo lineal uniforme, la cual se caracteriza por contar con elementos equidistantes a lo largo de un solo eje. De forma ilustrativa, se puede plantear un arreglo lineal, también conocido como sistema radiador lineal discreto, compuesto por N elementos situados entre sí a una distancia d igual a $\frac{\lambda}{2}$, y una longitud total L , a lo largo del eje x de acuerdo a lo que se muestra en la figura 2.15. Así mismo, se sitúa un punto P en el plano xz , trazando el recorrido que realizan las ondas transmitidas por los elementos del arreglo para llegar a él. Se considera que P se encuentra en la región lejana del arreglo por lo que las ondas transmitidas son paralelas entre sí.

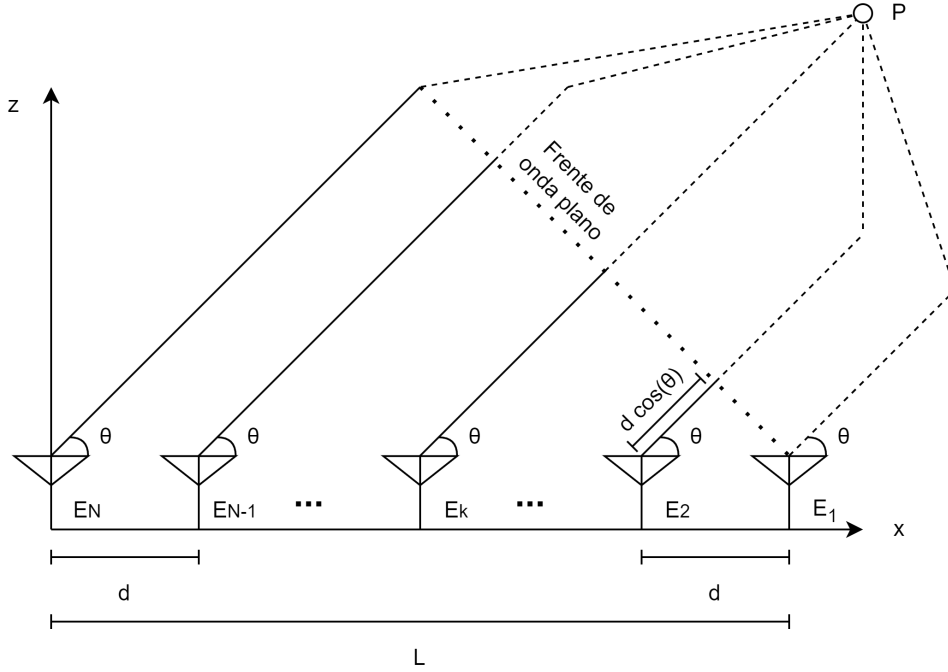


Figura 2.15: Arreglo lineal uniforme.

Según lo planteado por [33], al formarse un ángulo θ entre P y la posición de los elementos del arreglo, existirá una diferencia de recorrido en las ondas emitidas por dichos elementos. Esta diferencia de recorrido toma un tiempo τ en ser cubierta, dependiendo de la distancia entre elementos, del ángulo θ y de la velocidad de propagación de la señal.

$$\tau = \frac{d \cdot \cos(\theta)}{c} \quad (2.6)$$

Donde τ es la diferencia de tiempo, d es la distancia entre elementos, θ es el ángulo formado y c es la velocidad de la luz.

Al tratarse de una señal sinusoidal, se sabe que la distancia cubierta por dicha señal en un ciclo corresponde a la longitud de onda λ . Dado que cada ciclo corresponde a 2π radianes, se puede definir una variable que cuantifique un ciclo de la señal con respecto a la distancia que recorre. A esta variable se le conoce como el número de onda, y se le representa con la letra k , obteniéndose mediante la siguiente expresión:

$$k = \frac{2\pi}{\lambda} \quad (2.7)$$

Con base en lo anterior, se puede determinar la diferencia de fase que existe entre las ondas transmitidas por elementos contiguos del arreglo, debido a la diferencia que existe entre sus

recorridos. Para ello, se multiplica la diferencia de recorrido $d \cdot \cos(\theta)$ por el número de onda, según se muestra a continuación:

$$\Delta\phi = kd \cdot \cos(\theta) \quad (2.8)$$

La expresión (2.8) resulta de mucha utilidad ya que permite relacionar la diferencia de fase que existe entre cada elemento del arreglo con la dirección en la que se transmiten las ondas. Lo anterior permite modelar la forma en la que se propagan las ondas en el espacio debido a la distribución espacial del arreglo. Sin embargo, también es necesario tomar en cuenta la corriente de excitación I_n que se le suministra a cada elemento del arreglo. A partir de ambas características, se puede obtener el factor de direccionalidad f_N que permite describir la forma en la que se distribuye la potencia radiada por el arreglo en función del ángulo θ .

Dado que se cuenta con N elementos en el arreglo, se debe sumar la potencia radiada por cada elemento si se desea obtener el factor de direccionalidad para un ángulo determinado. Según [33], la expresión del factor de direccionalidad para el arreglo planteado es la siguiente:

$$f_N(\theta) = \sum_{n=1}^N I_n e^{j(n-1)(kd \cdot \cos(\theta))} \quad (2.9)$$

Donde N es el número total de elementos en el arreglo e I_n es la corriente de excitación para el n -ésimo elemento del arreglo. En esta expresión, se discretiza la posición a lo largo del eje x para los N elementos, tomando en cuenta que el primer elemento del arreglo se usa como referencia para las diferencias de recorrido en los elementos subsecuentes como se muestra en la figura 2.15.

Suponiendo que el punto P se encuentra en dirección normal al arreglo, y que todos los elementos tienen la misma fase, se observa que aparece un máximo en la dirección $\theta = 90^\circ$ como se aprecia en la figura 2.13. Esto se debe a que todas las ondas transmitidas recorren la misma distancia para dicho ángulo, por lo que se obtiene un resultado puramente real de f_N . Conforme el ángulo θ se modifica para el punto P , los recorridos empiezan a diferir y aparece un cierto grado de interferencia destructiva que desemboca en una menor magnitud para f_N .

Si se desea dirigir el haz en otra dirección, es necesario introducir una diferencia de fase lineal mediante la corriente de excitación I_n . Lo anterior permite modificar el ángulo en donde aparece el valor máximo para f_N al compensar la diferencia de recorridos que aparece como se muestra en la figura 2.14. Para ello, se puede definir a I_n de la siguiente forma:

$$I_n = I_0 e^{-j(n-1)\Delta\phi_{elem}}, \quad n = 1, 2, \dots, N \quad (2.10)$$

Donde I_0 representa una magnitud constante para la corriente de excitación, n es el índice del elemento dentro del arreglo y $\Delta\phi_{elem}$ es la diferencia de fase lineal entre elementos colindantes. La representación polar de las señales correspondientes a cada elemento del arreglo suele facilitar el análisis que se abordará a continuación.

Al sustituir (2.10) en (2.9), se puede observar que el valor máximo aparece cuando la fase se anula, quedando únicamente la magnitud de la corriente de excitación.

$$f_N(\theta) = I_0 \sum_{n=1}^N e^{j(n-1)(kd \cdot \cos(\theta) - \Delta\phi_{elem})} \quad (2.11)$$

De la misma forma, se puede describir la diferencia de fase entre los elementos vecinos mediante el coeficiente de desaceleración ξ , definido de la siguiente forma:

$$\xi = \frac{\Delta\phi_{elem}}{kd} \quad (2.12)$$

Si se sustituye (2.12) en (2.11) y se introduce una variable q igual a $e^{jkd(\cos(\theta) - \xi)}$, se obtiene la siguiente expresión:

$$f_N(\theta) = I_0 \sum_{n=1}^N q^{n-1} \quad (2.13)$$

A partir del desarrollo subsecuente de (2.13) expuesto en [33], el factor de direccionalidad del arreglo se encuentra descrito por la expresión (2.14), en donde Ψ es igual a $\frac{Nkd}{2}(\cos(\theta) - \xi)$.

$$f_N(\theta) = \left| \frac{\text{sen}(\Psi)}{N \text{sen}(\frac{\Psi}{N})} \right| \quad (2.14)$$

La expresión (2.14) describe el patrón de radiación del arreglo, el cual se asemeja a una función sinc con la peculiaridad de que aparecen lóbulos principales secundarios en los múltiplos de $\pm N\pi$. Por otra parte, como también es posible observar en las figuras 2.13 y 2.14, aparecen algunos máximos locales no deseados de la función f_N , a los que ya se había definido anteriormente como lóbulos laterales. A partir de ello, se define el nivel de los lóbulos laterales, el cual corresponde al valor normalizado de f_N de los lóbulos adyacentes al lóbulo principal.

Si se cuenta con una distribución de magnitud uniforme a lo largo del arreglo, el nivel de los lóbulos laterales será igual a -13.2 dB, lo que se considera alto para aplicaciones

reales. En este sentido, se puede usar una distribución no uniforme, asegurando que los elementos centrales del arreglo tendrán una mayor contribución hacia f_N que aquellos en los extremos. Esto trae como consecuencia que se disminuya la interferencia constructiva que pudiese ocurrir a los extremos, lo que reduce el nivel de los lóbulos laterales, al igual que el ensanchamiento del haz debido a una menor interferencia destructiva hacia el centro del arreglo. En este trabajo, se considera la distribución de magnitud no uniforme del coseno sobre pedestal, la cual se encuentra definida de la siguiente forma:

$$|I_n| = 1 + \Delta_{cos} \cdot \cos\left(\frac{2\pi z_{cos}}{L}\right) \quad (2.15)$$

Donde Δ_{cos} determina el aumento en magnitud de los elementos centrales del arreglo a comparación de aquellos en sus extremos y z_{cos} es la posición en el eje z con respecto al centro del arreglo. Por ende, z_{cos} se encuentra definido desde $-\frac{L}{2}$ hasta $\frac{L}{2}$ con una distancia d entre cada elemento del arreglo de tamaño N .

Es posible combinar ambos esquemas para lograr dirigir el haz, al mismo tiempo que se disminuye el nivel de los lóbulos laterales. Sustituir (2.15) en (2.13), tomando en cuenta que ya no se trata de una distribución uniforme, da como resultado la siguiente expresión:

$$f_N(\theta) = \sum_{n=1}^N |I_n| \cdot q^{n-1} \quad (2.16)$$

2.3. Procesamiento de señales para radar

A partir de la descripción realizada de las antenas en arreglo de fase, particularmente de los arreglos lineales uniformes (ULA, por sus siglas en inglés), se pueden desarrollar tres técnicas de procesamiento que permiten la estimación de los parámetros de un objeto detectado en el espacio. La primera de estas técnicas corresponde a la formación de haz, la cual permite obtener las coordenadas angulares del objeto. Por otro lado, se cuenta con la compresión de pulso, la cual permite obtener la distancia de un objeto al sistema de radar. Por último, se tiene la estimación de la velocidad radial mediante el efecto Doppler. A continuación, se explora de forma general cada una de estas técnicas, haciendo énfasis en su aplicación para un ULA.

2.3.1. Formadores de haz

En caso de que las señales recibidas por un sistema de radar correspondan a las señales reflejadas por un objeto, existirá una diferencia de fase entre los elementos del arreglo según

lo estipulado en la expresión (2.8), producto de la diferencia de recorridos. Lo anterior puede ser usado para determinar las coordenadas angulares del objeto en cuestión, mediante la formación de pesos que permitan compensar la diferencia de fase para la dirección del objeto. Al tratarse de un ULA, se puede determinar únicamente una de las dos coordenadas angulares del objeto mediante la formación de haz. En el caso del arreglo mostrado en la figura 2.15, la formación de haz permite obtener el ángulo de elevación. Sin embargo, al rotar el arreglo sobre el eje z , se puede realizar un barrido que permita determinar el ángulo azimutal.

Las señales recibidas en cada elemento del arreglo con una cantidad de elementos N pueden ser representadas utilizando su forma polar, como se muestra a continuación:

$$s_n = A_n e^{j\phi_n} \quad (2.17)$$

En donde A_n representa la magnitud de la señal recibida en cada elemento, mientras que ϕ_n representa su fase. Suponiendo que s_n corresponda a cada una de las señales reflejadas por el objeto, ϕ_n se encontrará dada en función de la posición del elemento en el arreglo, al igual que el ángulo de elevación de dicho objeto.

$$\phi_n(\theta) = (n - 1)(kd \cdot \cos(\theta)), \quad n = 1, 2, \dots, N \quad (2.18)$$

Los pesos, a los que se referirá como w_n de aquí en adelante, se encuentran definidos de la siguiente forma:

$$w_n = A_{w_n} e^{j\phi_{w_n}} \quad (2.19)$$

Si se desea compensar la fase para s_n , ϕ_{w_n} deberá ser igual a $-\phi_n$. Esto da como resultado que, al aplicarse estos pesos mediante una multiplicación compleja, las señales resultantes en todos los elementos del arreglo se encontrarán en fase. En este sentido, la aplicación de pesos es equivalente a contar con desplazadores de fase, con la enorme ventaja que se pueden modificar si se cuenta con un sistema reconfigurable.

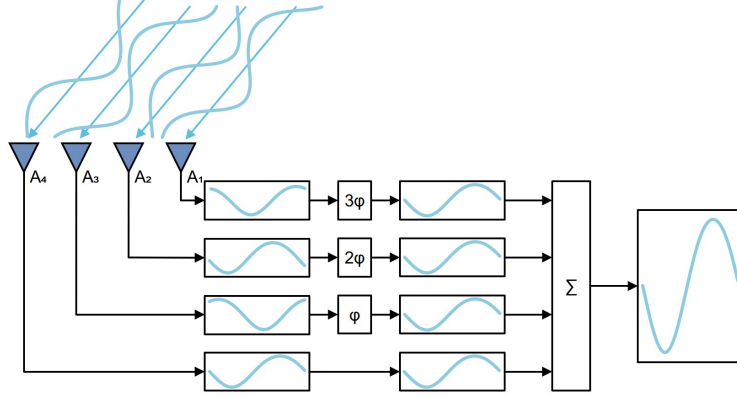


Figura 2.16: Compensación de la diferencia de fase en un formador de haz [13].

En principio, el sistema de recepción no conoce en que dirección podría encontrarse un objeto, a raíz de ello se generan múltiples conjuntos de pesos, los cuales representan haces cuyos lóbulos principales apuntan hacia diferentes ángulos de elevación. En caso de que los pesos para la formación del haz en una dirección determinada compensen la diferencia de fase existente entre las señales recibidas en cada elemento del arreglo, se obtendrá una señal de mayor magnitud al sumarse todas las salidas correspondientes. En dichas circunstancias, existe interferencia constructiva entre los elementos del arreglo, lo que permite determinar en cuál de los haces formados se encuentra la señal reflejada por el objeto.

Así mismo, las características del haz pueden modificarse aplicando una distribución no uniforme de la magnitud, como la que se muestra en la expresión (2.15). Esto permite escanear el espacio de forma más minuciosa, logrando detectar objetos con mayor exactitud al manipular el ancho del haz, al igual que el nivel de los lóbulos laterales.

2.3.2. Compresión de pulso

Existen ocasiones en donde el ruido u otros factores dificultan la detección de un objeto. Considerando lo anterior, se puede incorporar un filtro cuya respuesta en frecuencia sea igual en magnitud al espectro del pulso generado en banda base, con la peculiaridad de que su fase sea de signo contrario. A este tipo de filtros se les denomina filtros en concordancia, siendo posible implementarlos en el dominio del tiempo mediante la convolución, tal y como se muestra en la expresión (2.20).

$$y_{mf}[n] = \sum_{k=1}^N x[k]h[n-k] \quad (2.20)$$

Donde $x[n]$ es la señal de entrada, $y_{mf}[n]$ es la salida del filtro y $h[n]$ es la respuesta al impulso del filtro.

A este proceso se le denomina compresión de pulso, ya que se obtiene un pulso de menor duración y de mayor amplitud en caso de eliminarse la fase de la señal recibida. A raíz de ello, la compresión de pulso resulta sumamente útil para determinar la distancia a un objeto.

La forma de onda del pulso transmitido desempeña un papel fundamental en la compresión de pulso y, por consiguiente, en la estimación de distancia a un objeto. Como se demuestra en [35], la resolución de distancia es inversamente proporcional al ancho de banda del pulso. En este ámbito, suele emplearse la definición de Rayleigh para el ancho de pulso, la cual lo establece como el inverso de la duración del pulso. Lo anterior implica que, para un pulso rectangular simple, la resolución de distancia alcanzable será inversamente proporcional a su duración.

En este contexto, el pulso rectangular constituye el caso más simple empleado comúnmente. Sin embargo, existen otras formas de onda que permiten obtener mejores resultados, como lo es el caso de *chirp*. Este tipo de pulsos permite alcanzar una mayor resolución de distancia por medio de la modulación en frecuencia del pulso transmitido, a partir de lo cual se obtiene una mayor compresión.

Además de la compresión de pulso, los sistemas de radar modernos incorporan técnicas específicas que permiten distinguir la detección potencial de un objeto del ruido inherente al sistema u otras fuentes de interferencia [36]. Dichas técnicas comparan las mediciones realizadas con un umbral para confirmar la presencia de un objeto y, de esta manera, evitar falsas alarmas. Uno de los algoritmos que destacan dentro de esta categoría es CFAR, la cual emplea un umbral dinámico que se ajusta a los cambios en la interferencia.

2.3.3. Efecto Doppler

Posterior a la detección de un objeto, se puede determinar si se encuentra en movimiento o no a partir de la existencia de un cambio de frecuencia en la señal recibida. A este fenómeno se le conoce como el efecto Doppler y es sumamente útil para determinar la velocidad radial del objeto.

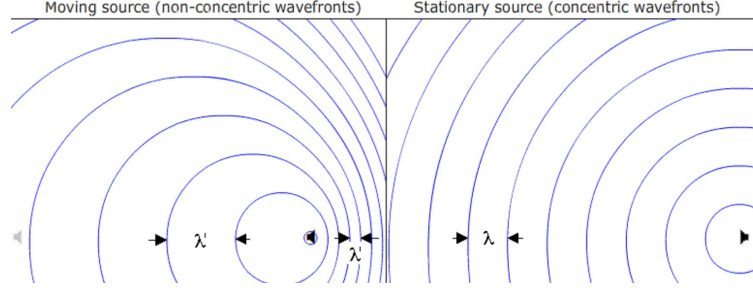


Figura 2.17: Frentes de onda para una fuente en movimiento y una fuente estática [37].

Si un objeto se mueve hacia el arreglo de antenas, las ondas se reflejan con mayor frecuencia, provocando que disminuya la distancia entre frentes de onda subsecuentes. Esto desemboca en un aumento de la frecuencia de llegada de las ondas recibidas por los elementos del arreglo. Por el contrario, si un objeto se aleja, aumenta la distancia entre frentes de onda y, por consiguiente, disminuye la frecuencia de llegada. Con base en lo anterior, este fenómeno se puede describir mediante la siguiente expresión:

$$f_{obs} = \frac{c + v_r}{c + v_s} f_{tx} \quad (2.21)$$

Donde f_{obs} es la frecuencia observada, c es la velocidad de propagación de la onda, v_r es la velocidad del receptor, v_s es la velocidad del transmisor y f_{tx} es la frecuencia de la onda transmitida originalmente. Por consiguiente, si se detecta un cambio en la frecuencia observada de la señal reflejada por un objeto, se puede determinar su velocidad radial.

2.4. Técnicas de procesamiento digital de señales

Existen ciertas técnicas de procesamiento digital de señales que suelen usarse con frecuencia en sistemas digitales de radar. En consecuencia, se explicarán de forma muy concisa dos técnicas de filtrado de señales, al igual que un algoritmo particular para aplicar la transformada discreta de Fourier.

2.4.1. Filtrado de señales

Existen dos tipos principales de filtros digitales, diferenciados entre sí a partir de la duración de su respuesta al impulso [38]. En primer lugar, se tiene a los filtros con respuesta finita al impulso (FIR, por sus siglas en inglés), descritos mediante la convolución discreta entre la señal de entrada $x[n]$ y los coeficientes del filtro b_k [39].

$$y[n] = \sum_{k=0}^M b_k x[n-k] \quad (2.22)$$

Donde M es el orden del filtro y $y[n]$ es la señal obtenida en la salida del filtro. Este tipo de filtros suelen ser más estables y fáciles de implementar, ya que la salida depende únicamente de la entrada y los coeficientes del filtro.

Por otro lado, se tiene aquellos filtros que cuentan con una respuesta infinita al impulso (IIR, por sus siglas en inglés) debido a la recursividad del mismo. Este tipo de filtros toman en cuenta valores previos de la salida para obtener el valor actual de esta misma, descritos mediante la siguiente expresión:

$$y[n] = \sum_{k=1}^N a_k y[n-k] + \sum_{k=0}^M b_k x[n-k] \quad (2.23)$$

Donde N es el orden del filtro, $y[n]$ es la señal obtenida en la salida del filtro, $x[n]$ es la señal de entrada y a_k en conjunto con b_k son los coeficientes del filtro. La realimentación que ocurre en los filtros IIR permite que se obtengan mejores resultados con un menor orden pero también hace que sean más propensos a volverse inestables.

Los filtros FIR suelen preferirse si se desea asegurar la estabilidad del sistema, requiriendo de un mayor número de coeficientes para lograr un mejor filtrado de la señal. Dado que no cuentan con realimentación, su implementación suele ser más fácil. Por otro lado, los filtros IIR suelen preferirse si se desea disminuir el tiempo de procesamiento del sistema, ya que necesitan un menor número de etapas para lograr el mismo resultado que un filtro FIR. Este menor número de etapas permite ahorrar una cantidad importante de recursos, sacrificando estabilidad por eficiencia.

2.4.2. Transformada discreta de Fourier (DFT)

La transformada de Fourier es un recurso fundamental en el procesamiento de señales ya que permite analizar las componentes en frecuencia de una señal [40]. En el dominio discreto, la transformada de Fourier se describe mediante la siguiente expresión:

$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n} \quad (2.24)$$

Donde $X(e^{j\omega})$ es la transformada de Fourier de la señal discreta $x[n]$. Si la expresión anterior se realiza sobre un número finito de muestras N , se obtiene la transformada discreta

de Fourier (DFT, por sus siglas en inglés). De acuerdo con [41], la DFT de N puntos se encuentra descrita de la siguiente forma:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-jk(\frac{2\pi}{N})n}, \quad k = 0, 1, \dots, N-1 \quad (2.25)$$

Donde $X[k]$ es la DFT de N puntos de la señal discreta $x[n]$. De la misma forma, se puede introducir una variable W_N igual a $e^{-j\frac{2\pi}{N}}$ en la expresión anterior:

$$X[k] = \sum_{n=0}^{N-1} x[n] W_N^{nk}, \quad k = 0, 1, \dots, N-1 \quad (2.26)$$

Como se muestra en [42], se puede demostrar que el cálculo directo de la DFT requiere N^2 operaciones, donde cada operación se define como una multiplicación compleja seguida de una suma compleja, lo que implica un consumo considerable de recursos computacionales. A partir de dicha problemática, surge una familia de algoritmos conocida como la transformada rápida de Fourier (FFT, por sus siglas en inglés), la cual permite reducir significativamente el número de cálculos requeridos.

Existen diversos algoritmos que permiten realizar la FFT, siendo uno de los más conocidos aquel propuesto por Cooley y Tukey en 1965 [42]. Este algoritmo reduce sustancialmente el número de operaciones involucradas al separar la DFT original en dos, agrupando las muestras pares e impares como se muestra en la expresión (2.27). Debido a la naturaleza de dicho agrupamiento, este algoritmo se clasifica como un algoritmo de decimación con un factor de dos en cada etapa.

$$X[k] = \sum_{n=0}^{(N/2)-1} x[2n] W_N^{2nk} + \sum_{n=0}^{(N/2)-1} x[2n+1] W_N^{k(2n+1)}, \quad k = 0, 1, \dots, N-1 \quad (2.27)$$

Dado que $W_N^2 = W_{N/2}$, se puede reescribir la expresión (2.27) de la siguiente forma:

$$X[k] = \sum_{n=0}^{(N/2)-1} x[2n] W_{N/2}^{nk} + W_N^k \sum_{n=0}^{(N/2)-1} x[2n+1] W_{N/2}^{nk}, \quad k = 0, 1, \dots, N-1 \quad (2.28)$$

Según lo planteado en [41], se pueden definir $F_1[k]$ y $F_2[k]$ como las DFT de $N/2$ puntos a partir de la expresión anterior:

$$X[k] = F_1[k] + W_N^k F_2[k], \quad k = 0, 1, \dots, N-1 \quad (2.29)$$

$F_1[k]$ y $F_2[k]$ son funciones periódicas para $N/2$ en donde $F[k + N/2] = F[k]$, mientras que para el factor W_N^k se cumple que $W_N^{k+N/2} = -W_N^k$. A partir de ello, se puede describir la expresión (2.29) de la siguiente forma:

$$X[k] = F_1[k] + W_N^k F_2[k], \quad k = 0, 1, \dots, \frac{N}{2} - 1 \quad (2.30)$$

$$X[k + \frac{N}{2}] = F_1[k] - W_N^k F_2[k], \quad k = 0, 1, \dots, \frac{N}{2} - 1 \quad (2.31)$$

De igual forma, se puede repetir este procedimiento, separando $F_1[k]$ y $F_2[k]$ en dos DFT de $N/4$ puntos, respectivamente:

$$F_1[k] = G_1[k] + W_{N/2}^k G_2[k], \quad k = 0, 1, \dots, \frac{N}{4} - 1 \quad (2.32)$$

$$F_1[k + \frac{N}{4}] = G_1[k] - W_{N/2}^k G_2[k], \quad k = 0, 1, \dots, \frac{N}{4} - 1 \quad (2.33)$$

$$F_2[k] = G_3[k] + W_{N/2}^k G_4[k], \quad k = 0, 1, \dots, \frac{N}{4} - 1 \quad (2.34)$$

$$F_2[k + \frac{N}{4}] = G_3[k] - W_{N/2}^k G_4[k], \quad k = 0, 1, \dots, \frac{N}{4} - 1 \quad (2.35)$$

El proceso anterior se realiza de forma recursiva hasta llegar al caso base, en donde se cuenta únicamente con dos secuencias de una muestra. Con base en las expresiones anteriores, se puede observar un patrón a partir del cual se forma la estructura básica de procesamiento de la FFT, tal y como se muestra a continuación:

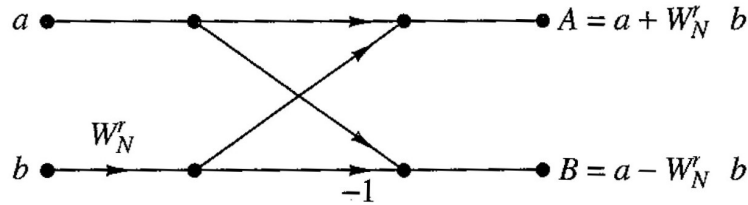


Figura 2.18: Estructura básica de procesamiento de la FFT *radix-2* [41].

Se suele referir a la representación gráfica de dicha estructura básica de procesamiento, mostrada en la figura 2.18, como el diagrama de mariposa. Dado que existe un factor de decimación de dos en cada etapa, se le denomina como *radix-2*. Existen variaciones de este algoritmo que involucran factores de decimación más altos, como lo es el caso de *radix-4*, así

como una combinación de dos factores de decimación (*split-radix*).

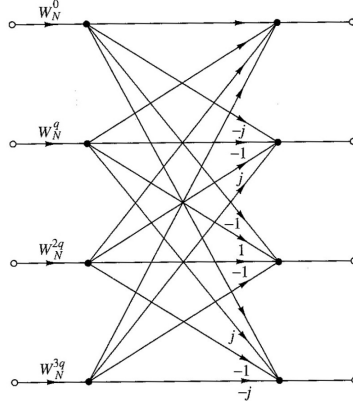


Figura 2.19: Estructura básica de procesamiento de la FFT *radix-4* [41].

Según se puede observar en [41], esta familia de algoritmos permite reducir el orden de operaciones realizadas desde N^2 hasta $N \log_2 N$. El número de etapas dependerá del factor de decimación utilizado, siendo dos y cuatro los más comunes.

2.5. Conceptos relevantes de diseño digital

Finalmente, solo resta desarrollar brevemente ciertos fundamentos teóricos de diseño digital que serán bastante significativos a lo largo de este trabajo. Para ello, se abordará de forma sucinta el formato de punto fijo, los FPGA y, en consecuencia, los lenguajes de descripción de *hardware*.

2.5.1. Formato de punto fijo

El formato de punto fijo permite representar números con punto decimal, destinando un bit para el signo, así como una cantidad fija de bits para la parte entera y otra para la parte decimal [43]. En este contexto, se suele emplear la notación Q para describir la cantidad de bits destinada a cada parte del número en cuestión. Dicha notación tiene el formato $Qm.n$, en donde m representa la cantidad de bits destinados a la parte entera, mientras n representa la cantidad de bits destinados a la parte fraccionaria [44]. Por lo tanto, cada bit de la parte entera representa una potencia positiva del número dos, i.e. 2^n , donde n es la posición del bit correspondiente. Por otro lado, cada bit de la parte fraccionaria representa una potencia negativa del número dos, i.e. 2^{-n} donde n es la posición del bit correspondiente.

Aunado a esto, se suele utilizar el complemento a dos para representar números negativos

de forma binaria. Dicho complemento se obtiene al invertir los bits del número binario de interés y sumar uno al valor invertido. Para esta representación, es esencial reservar un bit para el signo, lo que limita el intervalo de valores a representar.

En contraste con el formato de punto flotante, las operaciones resultan más fáciles de realizar al contar una resolución fija determinada por el número de *bits* destinados a la parte fraccionaria. En consecuencia, el *hardware* requerido para operaciones de punto fijo suele ser más simple, con la desventaja de no poder alterar la resolución.

2.5.2. Arreglos de compuertas programables en campo (FPGA)

Los FPGA son circuitos integrados digitales que permiten cambiar su estructura en campo según se requiera. La unidad fundamental de los FPGA son los bloques lógicos basados en tablas de búsqueda (LUT, por sus siglas en inglés), los cuales se caracterizan por contar con una tabla de verdad reconfigurable. Aunado a esto, dichos bloques permiten incorporar elementos como *flip-flops*, permitiendo generar lógica secuencial.

Estos bloques, en conjunto con interconexiones programables, permiten describir el *hardware* necesario para una aplicación específica, sin necesidad de contar con un circuito integrado dedicado para ello. Sin embargo, los FPGA también suelen incorporar algunos bloques de *hardware* dedicado que permite realizar las operaciones asociadas al procesamiento de forma más eficiente.

2.5.3. Lenguajes de descripción de *hardware*

El desarrollo de arquitecturas digitales en los FPGA se realiza mediante los lenguajes de descripción de *hardware* (HDL, por sus siglas en inglés), los cuales permiten describir los circuitos digitales requeridos para un diseño determinado. Esto agiliza el desarrollo de sistemas digitales al permitir la generación o síntesis de *hardware* específico de forma bastante práctica, simplificando el proceso de diseño.

Uno de los HDL más predominantes es el Lenguaje de Descripción de *Hardware* de Circuitos Integrados de Muy Alta Velocidad (VHDL, por sus siglas en inglés), creado en la década de 1980 en un esfuerzo del Departamento de Defensa de los Estados Unidos por estandarizar la descripción de *hardware* [45]. VHDL fue tan exitoso que se adoptó como un estándar de la IEEE (IEEE Std. 1076) en 1987 y sigue siendo actualizado hasta el día de hoy.

Según [46], en la terminología propia de VHDL, cada bloque de *hardware* es una entidad con puertos de entrada y salida descrita mediante una arquitectura que define su funciona-

miento. Así mismo, las entidades suelen hacer uso de bibliotecas en las que se definen tipos de datos, operaciones lógicas, etc.

Existen dos estilos para describir la arquitectura de una entidad: conductual y estructural. El primer estilo permite describir el comportamiento del *hardware* con un alto grado de abstracción, a partir de lo cual se sintetiza el circuito requerido [47]. Por otro lado, el estilo estructural permite describir los elementos que componen a la entidad como lo pueden ser compuertas lógicas, registros u otros circuitos digitales dedicados.

Una herramienta sumamente relevante dentro del contexto de los HDL, son los núcleos o bloques de propiedad intelectual (IP, por sus siglas en inglés). Estos bloques son componentes HDL reusables propietarios [48] que permiten agilizar el flujo de desarrollo al ofrecer arquitecturas flexibles y optimizadas. Dichos componentes pueden abarcar técnicas de procesamiento específicas como la FFT hasta núcleos completos de procesamiento de propósito general.

2.6. Conclusiones

En conclusión, los conceptos teóricos presentados a lo largo de este capítulo permitirán desarrollar el modelado del procesamiento asociado al bloque de recepción de un sistema de radar pulsado, al igual que la instrumentación del módulo de compresión de pulso del mismo en un FPGA. En primer lugar, se estableció la estructura fundamental de los sistemas de radar pulsados monoestáticos, haciendo énfasis en el bloque de recepción, a partir de lo cual será posible definir la arquitectura del sistema a modelar. A continuación, se presentaron ciertos aspectos esenciales de las antenas como su patrón de radiación. Así mismo, se ahondó en el comportamiento de las antenas en arreglo de fase, haciendo hincapié en los arreglos lineales uniformes. En este contexto, se pudo observar el efecto que tiene la diferencia de fases entre elementos contiguos del arreglo y la distribución no uniforme de magnitud sobre el factor de direccionalidad del arreglo.

Aunado a esto, se explicaron brevemente las tres técnicas de procesamiento de señales para radar que serán empleadas en este trabajo: los formadores de haz, la compresión de pulso y la estimación de velocidad radial mediante el efecto Doppler. Los formadores de haz permiten determinar el ángulo de elevación mediante la aplicación de pesos a los elementos del arreglo. Por otro lado, la compresión de pulso posibilita la estimación de la distancia entre el objeto y el sistema de recepción a través del uso de un filtro en concordancia. Por último, se presentó el efecto Doppler y su utilidad para la estimación de la velocidad radial de un objeto.

La siguiente sección se enfocó en las técnicas de procesamiento digital de señales requeridas a lo largo de este trabajo. Primero, se explicó de forma concisa los filtros FIR e IIR, describiendo sus expresiones fundamentales, al igual que sus ventajas y desventajas. Después, se desarrolló el concepto de la transformada discreta de Fourier, presentando de forma general el algoritmo básico requerido para el desarrollo de la transformada rápida de Fourier.

La última sección de este capítulo abordó los conceptos relevantes de diseño digital necesarios tanto para el modelado como para la instrumentación. En primer lugar, se presentó el formato de representación numérica a utilizar en este trabajo: el formato de punto fijo. Dicho formato permitirá representar números con punto decimal empleando una resolución fija determinada por el número de bits alocados a la parte entera y a la parte fraccionaria. Por otro lado, se introdujo brevemente los FPGA como herramientas de diseño de *hardware* a la medida, los cuales permitirán llevar a cabo la instrumentación requerida. En consecuencia, se presentaron los HDL, los cuales permitirán describir los circuitos digitales necesarios para el módulo de compresión de pulso dentro de un FPGA.

Capítulo 3

Diseño

A partir del establecimiento de la motivación del presente trabajo y de los conceptos fundamentales necesarios para comprender su contexto, este capítulo tiene por objetivo describir el proceso de diseño empleado. En primer lugar, se presenta brevemente la metodología de diseño utilizada, a partir de la cual se desarrolla el modelo de procesamiento correspondiente al bloque de recepción del sistema de radar pulsado. Este modelo constituye el punto de partida para la instrumentación del módulo de compresión de pulso.

3.1. Metodología de diseño

Se hizo uso de la metodología de diseño de producto presentada por Ulrich y Eppinger en [49] para desarrollar la propuesta ya que ofrece un marco de trabajo robusto. A partir de ello, se definen cinco etapas principales que conforman al proceso de diseño, las cuales se presentan brevemente a continuación.

En primer lugar, se tiene la etapa de planeación, la cual tiene como objetivo la identificación del problema a resolver. Con base en lo anterior, se define el objetivo del sistema a desarrollar, en conjunto con sus alcances. A continuación, se tiene la etapa de desarrollo de concepto, la cual se enfoca en la delimitación de requerimientos con base en el objetivo planteado. A partir de dichos requerimientos, se proponen soluciones, eligiendo aquella que permita satisfacer las necesidades de la mejor manera.

La siguiente etapa es la correspondiente al diseño a nivel sistema, en la cual se define la arquitectura general de la solución, desglosándose en subsistemas y componentes. Es aquí en donde se realizan los diagramas a bloques pertinentes, especificando las funcionalidades de cada subsistema. A partir de la etapa anterior, surge el diseño a detalle, que consiste en el

desarrollo de cada subsistema, enfatizando en la selección de componentes y el desempeño robusto del sistema. De igual forma, se elabora la documentación completa del sistema con todos los elementos necesarios para cada subsistema.

Por último, la etapa de pruebas y refinamiento conlleva la construcción de la solución expuesta, al igual que su evaluación con respecto a las especificaciones estipuladas en un principio. Cabe recalcar que en [49], se habla de una etapa adicional que no se toma en cuenta en este trabajo, al tratarse de un sistema que no está diseñado para su producción en masa.

3.2. Modelado del procesamiento del bloque de recepción del sistema de radar pulsado monoestático

3.2.1. Planeación del sistema

Con el fin de incursionar en los sistemas de radar modernos, el grupo de investigación especializado en electrónica de alta frecuencia y microondas de la Facultad de Ingeniería de la UNAM identificó la necesidad de migrar hacia sistemas digitales de radar. En este contexto, se solicitó desarrollar el modelado del procesamiento asociado al bloque de recepción de un sistema de radar pulsado monoestático que recibe señales por medio de un arreglo lineal uniforme de antenas en fase.

Características del sistema de radar y criterios de procesamiento digital

Tomando en cuenta el objetivo planteado, resulta menester delimitar las características del sistema de radar a modelar entre las que destacan su frecuencia de operación, el tipo de antenas que utiliza, así como el tipo de onda que transmite y recibe. Se requiere modelar el bloque de recepción de un sistema de radar monoestático pulsado en banda S que hace uso de un arreglo lineal uniforme de antenas ranuradas en fase. Dicho arreglo se encuentra compuesto por 32 elementos, con una separación de $\frac{\lambda}{2}$ entre sí, operando a una frecuencia de 2.5 GHz. Aunado a esto, el pulso transmitido para la detección inicial de objetivos cuenta con un ancho de 12.5 μ s y una frecuencia de repetición de 250 Hz.

De igual forma, se acota el alcance del procesamiento digital a la estimación del ángulo de elevación mediante un formador de haces, al igual que la obtención de ciertos parámetros básicos, como lo son la distancia al objeto y la velocidad radial. Se recomienda llevar a cabo el procesamiento digital de señales a partir de una señal de IF, centrada en 10 MHz, con

una amplitud mínima de 1 V. Así mismo, se sugiere emplear la demodulación en cuadratura como punto de partida para el procesamiento digital de señales, considerando una frecuencia de muestreo de 100 MHz.

3.2.2. Desarrollo de concepto

Con base en las características del sistema de radar a modelar, se considera apropiado elegir una arquitectura de recepción heterodina como punto de partida para el modelado del sistema. Este tipo de arquitectura requiere contar con una sección de conversión a IF, a partir de la cual se realizará la conversión analógica-digital de las señales recibidas. De forma subsecuente, se contempla una conversión a banda base en donde se llevará a cabo el procesamiento requerido. Por consiguiente, se puede proponer el diagrama a bloques general del sistema, tomando en cuenta la recepción de las señales en cada elemento del arreglo, así como su acondicionamiento previo al procesamiento digital de señales. Se identificaron dos etapas principales que componen al sistema, según se aprecia en la figura 3.1.

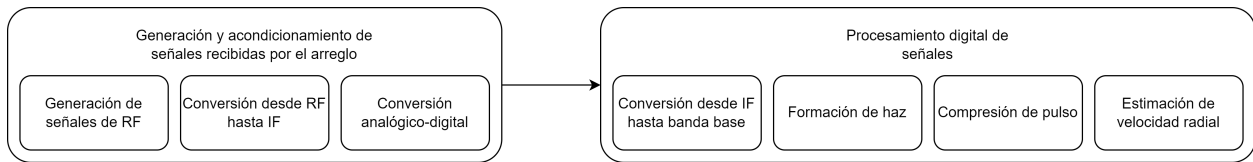


Figura 3.1: Etapas principales del sistema.

Etapas de generación y acondicionamiento de señales recibidas por el arreglo

La primer etapa involucra la generación y el acondicionamiento de las señales recibidas en cada elemento del ULA en la banda de frecuencia seleccionada. Durante esta etapa, es necesario generar las 32 señales correspondientes a cada elemento del arreglo, tomando en cuenta que, en caso de recibir las ondas reflejadas por algún objeto, los pulsos radiados tendrán una duración de $12.5 \mu s$ y contendrán una señal sinusoidal con frecuencia de 2.5 GHz. Al tratarse de un modelo de primera aproximación, se propone considerar la recepción de pulsos rectangulares, ya que constituyen la forma de onda más simple en los sistemas de radar.

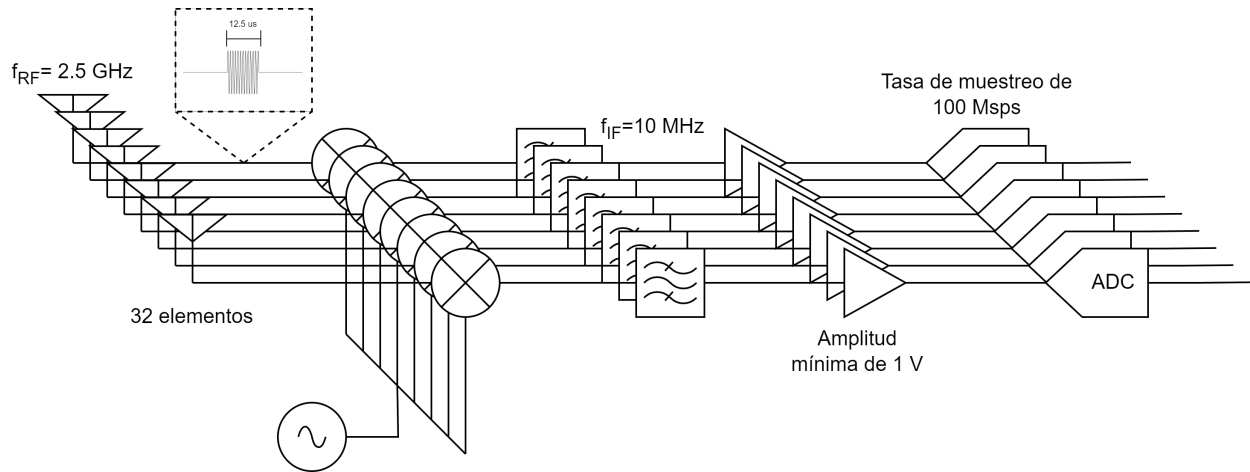


Figura 3.2: Modelo simplificado de la recepción de señales.

Las señales generadas de RF requieren un proceso de acondicionamiento, el cual abarca la conversión hacia la frecuencia intermedia seleccionada, así como su respectiva conversión analógica-digital. Tomando en cuenta lo anterior, se sugirió que las señales resultantes tuviesen una frecuencia centrada en 10 MHz, al igual que una amplitud mínima de 1 V. Lo anterior implica contar con al menos una etapa de conversión, capaz de bajar la frecuencia de las señales recibidas desde 2.5 GHz hasta 10 MHz, con su respectiva sección de filtrado y amplificación. Considerando las condiciones antes descritas, el modelo propuesto en la figura 3.2 representa una simplificación del sistema real, ya que la conversión directa desde 2.5 GHz hasta 10 MHz resulta poco factible debido al problema de la frecuencia imagen. Por último, dada la frecuencia de muestreo recomendada, es necesario incorporar un convertidor analógico-digital que tenga una tasa de muestreo mínima de 100 Mps.

Etapas de procesamiento digital de señales

La segunda etapa corresponde al procesamiento digital de señales, en donde se lleva a cabo la demodulación en cuadratura, la formación de haz, la compresión de pulso y la estimación de la velocidad radial. Para esta etapa, se sugiere comenzar con la demodulación en cuadratura de las señales digitales de IF previamente acondicionadas, estableciendo la frecuencia de operación del sistema en 100 MHz. A partir de las componentes obtenidas, es necesario llevar a cabo la formación de haces para la estimación del ángulo de elevación, lo cual requiere aplicar pesos determinados a cada uno de los 32 elementos del arreglo. Dados los alcances de este trabajo, el sistema contempla únicamente la estimación del ángulo de elevación. Así mismo, se requiere incorporar un filtro en concordancia que permita llevar a cabo la compresión de pulso. Por último, es necesario determinar la velocidad radial del

objeto a partir de la detección del desplazamiento en frecuencia ocasionado por el efecto Doppler.

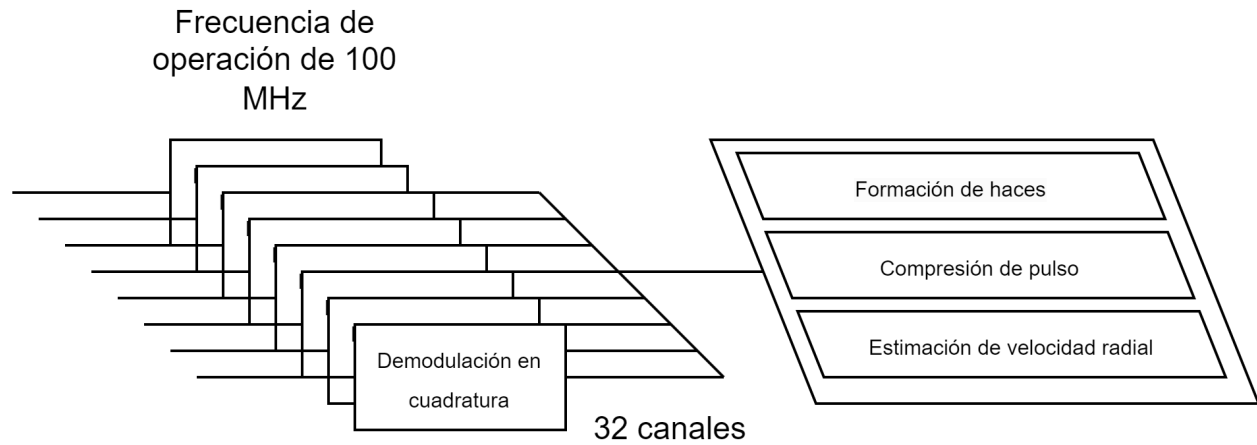


Figura 3.3: Diagrama ilustrativo del procesamiento digital de señales.

3.2.3. Diseño a nivel sistema

El desglose anterior resulta un buen punto de partida para profundizar en las especificaciones con las que deberá contar cada etapa.

Consideraciones para la generación de señales recibidas por el arreglo

En primer lugar, la generación de señales recibidas por cada elemento del arreglo en la banda de frecuencia seleccionada requiere estipular los fenómenos que se tomarán en cuenta, al igual que las características propias de los pulsos recibidos.

En este sentido, es necesario considerar que las señales de RF captadas por el arreglo suelen encontrarse atenuadas a un nivel considerable. En caso de recibirse las señales reflejadas por algún objeto, dicha atenuación puede ser descrita a partir de la distancia que recorren, al igual que las características propias del objeto y de las antenas desde las que se transmitieron las señales [50]. Sin embargo, dados los alcances estipulados para este trabajo escrito, se opta por establecer un nivel de atenuación fijo y uniforme en todos los elementos, independiente a las variables antes mencionadas, en aras de simplificar el proceso de generación de señales. Tomando en cuenta lo anterior, se propone representar la distancia al objeto únicamente mediante el retardo que existe en la recepción de las señales reflejadas.

Por otro lado, la generación de señales debe tomar en cuenta que las ondas reflejadas por el objeto, al ser captadas por el arreglo, presentan una diferencia de fase entre sí en función

de diversos parámetros. En particular, se consideran el ángulo de elevación del objeto, la distancia entre elementos, la frecuencia de operación del arreglo y la posición relativa de cada elemento receptor dentro del arreglo. Así mismo, si el objeto se encuentra en movimiento, la frecuencia recibida se verá afectada por el efecto Doppler, experimentando un desplazamiento conforme a lo descrito en la expresión (2.21).

Diseño de la etapa de generación y acondicionamiento de señales recibidas

Con base en las consideraciones enumeradas, se propone dividir esta etapa en dos secciones, las cuales contarán con dos subsistemas cada una. La primera sección corresponde a la generación de señales recibidas, mientras que la segunda sección corresponde al acondicionamiento requerido de dichas señales previo a su procesamiento digital.

El primer subsistema de la primera sección permitirá generar un pulso rectangular único con un retardo acorde a la distancia del objeto simulado, una duración de $12.5 \mu s$ y un nivel fijo de atenuación. De igual forma, este pulso se encontrará relleno de una señal sinusoidal con frecuencia base de 2.5 GHz, la cual puede encontrarse desplazada en caso de simular un objeto en movimiento. El pulso rectangular generado será suministrado al siguiente subsistema, encargado de formar los 32 canales correspondientes a las antenas ranuradas que componen el arreglo. Este segundo subsistema desplazará la fase de las señales en función de los parámetros mencionados. Para fines prácticos, se toma el caso ideal en donde los canales se encuentran totalmente aislados uno del otro.

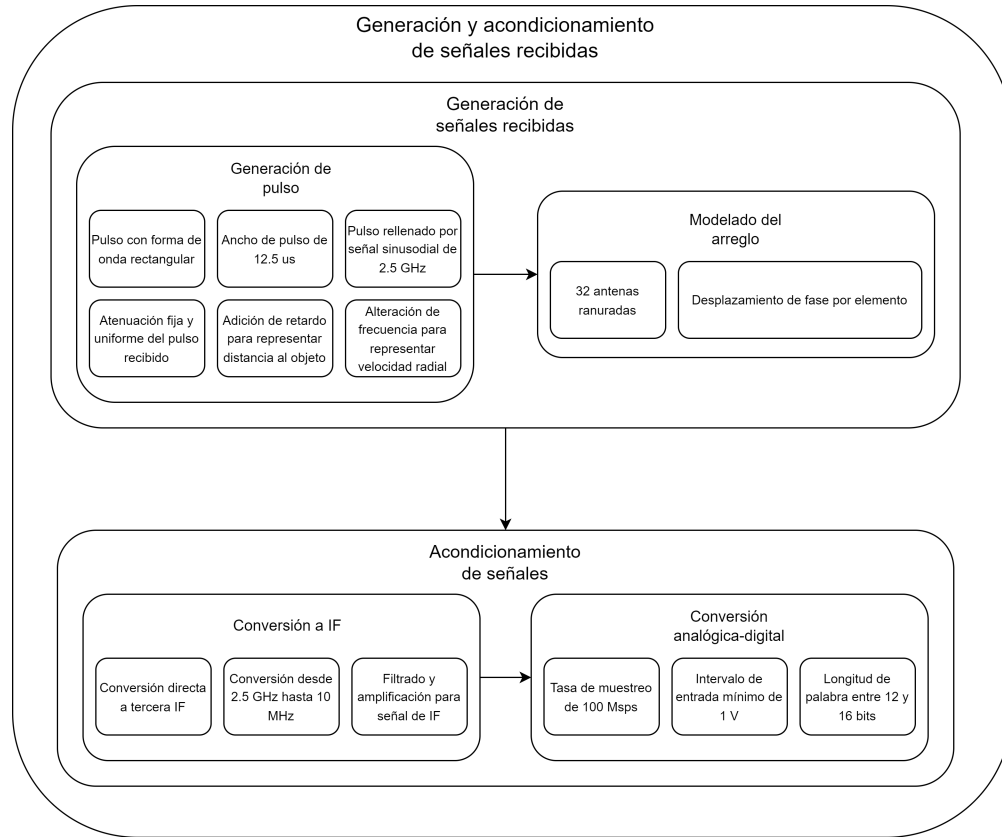


Figura 3.4: Diagrama a bloques de la primera etapa.

A continuación, se cuenta con la sección correspondiente al acondicionamiento de señales recibidas en donde se contempla la conversión a frecuencia intermedia, al igual que la conversión analógica-digital de las mismas. En esta sección, resulta fundamental proponer el número de etapas que conforman la sección de conversión a IF, en conjunto con sus respectivas frecuencias, al igual que delimitar las características deseadas de las señales digitales resultantes.

Como ya se mencionó anteriormente, se suele emplear dos o tres etapas de conversión a IF, cada una acompañada de su correspondiente sección de filtrado y amplificación. El uso de múltiples etapas permite eliminar componentes espectrales no deseadas que puedan generarse durante el proceso de conversión, además de aumentar el nivel de la señal recibida hasta hacerlo adecuado para las etapas de procesamiento posteriores. Sin embargo, el modelado minucioso de múltiples etapas se encuentra fuera de los alcances de este trabajo. En consecuencia, se considerará una única etapa de conversión desde RF hasta IF, eligiendo no tomar en cuenta la aparición de la frecuencia imagen. En un sistema real, la frecuencia intermedia elegida constituiría la frecuencia resultante de la tercera etapa de conversión des-

cedente. Por lo tanto, el desarrollo de un modelo que permita simular las tres etapas de conversión queda como trabajo a futuro. Tomando en cuenta esta limitación, la conversión analógica-digital se llevará a cabo a la salida de dicha etapa única de conversión.

De tal forma, el primer subsistema de la segunda sección llevará a cabo la conversión desde RF hasta IF, en conjunto con su respectivo filtrado y amplificación, de los 32 canales generados por la primera sección. Con este fin, es necesario contar con un mezclador capaz de realizar la conversión directa desde 2.5 GHz hasta 10 MHz, asegurando que las señales resultantes tengan una amplitud mínima de 1 V. El segundo subsistema se encargará de incorporar la conversión analógica-digital subsecuente. Para ello, se requiere un ADC con una tasa de muestreo de 100 Msps, al igual que un intervalo de entrada mínimo de 1 V. Se propone que la longitud de palabra del ADC se encuentre entre los 12 y los 16 bits a partir de los modelos recomendados por los fabricantes Analog Devices, NXP y Texas Instruments para este tipo de aplicaciones [51] [52] [53].

Consideraciones respecto al procesamiento digital de señales

Partiendo de las especificaciones de la etapa anterior, se puede abordar el desarrollo de la etapa de procesamiento digital de señales. Para ello, se delimitan las características del sistema digital en función de las señales que recibirá. En este rubro, se establece una frecuencia de operación de 100 MHz, con una longitud de palabra de 16 bits y un formato de punto fijo.

Así mismo, es relevante establecer los requerimientos para llevar a cabo las técnicas de procesamiento requeridas (demodulación en cuadratura, formación de haz, compresión de pulso y estimación de velocidad radial). Como se mencionó previamente, las señales digitales de IF obtenidas en la etapa de acondicionamiento se someten, en primera instancia, a la demodulación en cuadratura, a través de la cual se lleva a cabo la conversión desde la IF seleccionada hasta banda base. A partir de las componentes en banda base resultantes, se requiere aplicar de manera simultánea ocho conjuntos de pesos sobre cada uno de los 32 canales, con el propósito de formar haces que permitan barrer el plano xz . Estos pesos presentan una distribución no uniforme en magnitud, diseñada para reducir el nivel de los lóbulos laterales. Así mismo, considerando la definición del ángulo θ mostrada en la figura 2.15, se establecen los siguientes ángulos de elevación para la formación de haces: 45° , 52.5° , 60° , 67.5° , 75° , 82.5° y 90° . Estos valores fueron elegidos con el fin de evitar la degradación del lóbulo principal que ocurre en ángulos de elevación más bajos.

Por otro lado, se requiere realizar la compresión de pulso mediante un filtro en concordancia con el propósito de obtener la distancia al objeto. Considerando la asignación de

un valor fijo de potencia recibida, el alcance máximo del sistema queda determinado por el intervalo de repetición de pulso (PRI, por sus siglas en inglés). Con base en la expresión 2.1 y un PRI de 4 ms, el sistema alcanza un valor teórico cercano a 600 km. Sin embargo, en un radar real la estimación del alcance máximo es un proceso mucho más complejo, por lo que se encuentra fuera del alcance del presente trabajo.

Finalmente, se requiere estimar de velocidad radial mediante la obtención del desplazamiento en frecuencia ocasionado por el efecto Doppler. En este ámbito, se considera que el sistema de recepción se encuentra estático, por lo que el objeto se convierte en la única fuente posible de movimiento.

Diseño de la etapa de procesamiento digital de señales

La segunda etapa se encuentra compuesta por dos secciones, tal y como se aprecia en el esquema de la figura 3.5. La primer sección incorpora a la demodulación en cuadratura de las señales digitales, desplazando su frecuencia desde 10 MHz hasta banda base. La segunda sección incluye tres subsistemas, correspondientes a la formación de haz, la compresión de pulso y la estimación de velocidad radial con los requerimientos delimitados anteriormente.

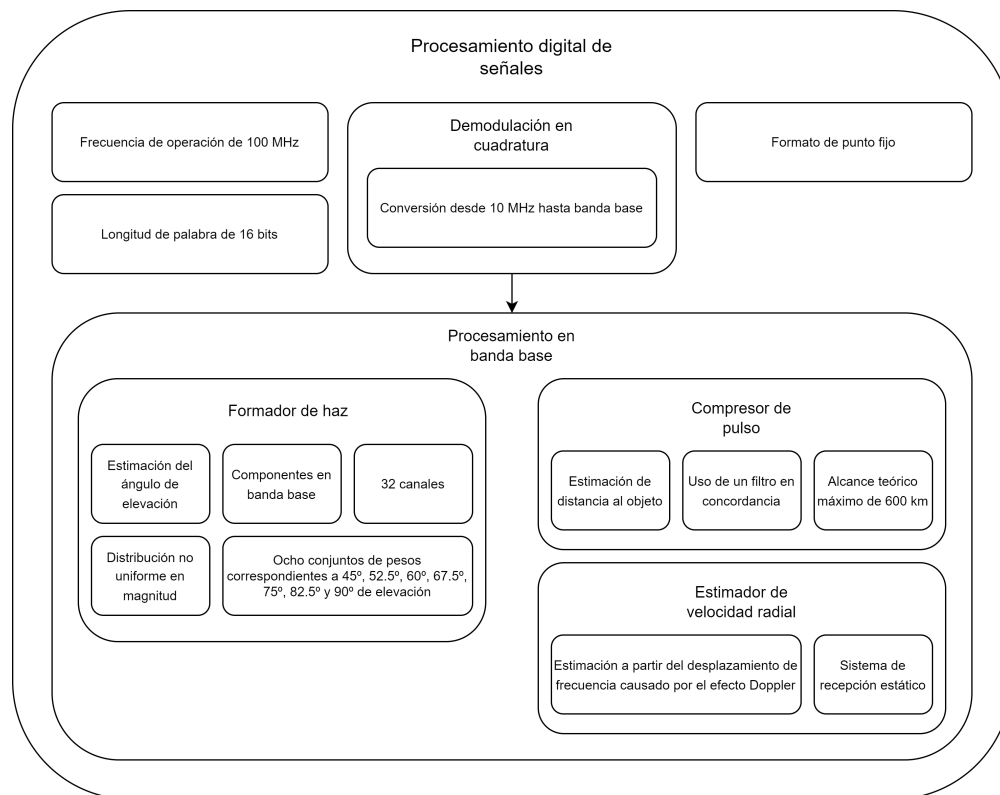


Figura 3.5: Diagrama a bloques de la segunda etapa.

3.2.4. Diseño a detalle

Una vez que se ha establecido la arquitectura general del sistema mediante el desglose en etapas, secciones y subsistemas, se puede proceder a diseñar cada uno de sus componentes.

Modelado de la generación de señales recibidas por el arreglo

A continuación, se presenta la propuesta para la sección de generación de señales recibidas por el arreglo.

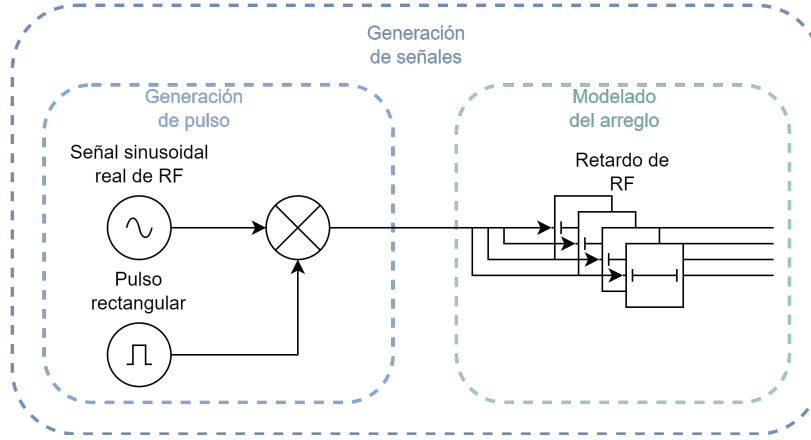


Figura 3.6: Propuesta para el desarrollo de la generación de señales.

El modelo mostrado en la figura 3.6 permite simular las señales recibidas por un ULA conformado por 32 antenas ranuradas que operan a 2.5 GHz. El arreglo en cuestión tiene una longitud total de 16λ (1.86 metros), con sus elementos distribuidos a lo largo del eje x y espaciados a una distancia de $\frac{\lambda}{2}$ (0.06 metros) entre sí.

En primer lugar, se cuenta con la generación de un pulso rectangular a partir del cual se crean los 32 canales correspondientes a los elementos del arreglo. Este pulso se forma mediante la multiplicación de una fuente sinusoidal con un pulso rectangular, configurados de tal forma que se satisfagan los requerimientos mostrados en la figura 3.4 para el subsistema correspondiente. La fuente permite configurar la atenuación de las señales reales recibidas, al igual que la frecuencia de las mismas. De ordinario, la frecuencia de la señal sinusoidal será de 2.5 GHz si el objeto simulado se encuentra estático. Por el contrario, si se desea simular un objeto en movimiento, el cambio de frecuencia puede ser obtenido al sustituir los parámetros necesarios en (2.21), como se muestra en la expresión (3.1).

$$f_{obs} = \frac{3 \times 10^8 \left[\frac{m}{s} \right]}{3 \times 10^8 \left[\frac{m}{s} \right] + v_s} 2.5 [GHz] \quad (3.1)$$

Donde f_{obs} es la frecuencia observada de la señal sinusoidal, mientras que v_s es la velocidad radial del objeto. Si se multiplica la frecuencia observada por 2π , se obtiene la frecuencia angular ω_{obs} correspondiente.

Por otro lado, el pulso rectangular permite establecer la distancia al objeto mediante el retardo que se le asigna conforme a lo descrito en (2.1). En este punto, la señal generada se encuentra descrita mediante la expresión (3.2).

$$a(t) = \cos(\omega_{obs}t) \cdot \text{rect}\left(\frac{t - (t_d + \frac{t_p}{2})}{t_p}\right) \quad (3.2)$$

Donde ω_{obs} es la frecuencia angular observada, t_d es el retardo en función de la distancia al objeto y t_p es el ancho del pulso.

El segundo subsistema se encarga de modelar la diferencia de fase que aparece entre elementos al captar la señal reflejada por un objeto con cierto ángulo de elevación. En esta sección, se generan los 32 canales que representan cada uno de los elementos del arreglo.

La diferencia de fase se puede representar mediante un retardo que aparece producto de la diferencia de recorrido mostrada en la figura 2.15. A partir de (2.6) y las características del arreglo, se puede obtener dicho retardo en función del ángulo de elevación como se muestra en (3.3).

$$\tau = \frac{\frac{\lambda}{2} \cdot \cos(\theta)}{c} \quad (3.3)$$

Dado que λ es igual al cociente de la velocidad de propagación y la frecuencia de la señal, (3.3) se reduce a la expresión (3.4).

$$\tau = \frac{\cos(\theta)}{2f_{RF}} \quad (3.4)$$

Este retardo aparece en función de la posición del elemento m dentro del arreglo conformado por N antenas. Por consiguiente, se puede actualizar (3.4) para que refleje esta relación.

$$\tau_m = \left(\frac{\cos(\theta)}{2f_{RF}}\right) (m - 1), \quad m = 1, 2, \dots, N \quad (3.5)$$

Si se multiplica este retardo por la frecuencia angular correspondiente, se puede obtener la diferencia de fase que dicho retardo representa. En este modelo, se asume que el desplazamiento de la frecuencia ω_o no afectará en gran medida al retardo τ , por lo que se toma ω_{RF} como la frecuencia angular asociada al retardo. Por lo tanto, las señales generadas para

cada PRI pueden ser descritas mediante la expresión (3.6).

$$a_m(t) = \cos(\omega_{obs}t - \omega_{RF} \cdot \tau_m) \cdot \text{rect}\left(\frac{t - (t_d + \frac{t_p}{2})}{t_p}\right), \quad m = 1, 2, \dots, N \quad (3.6)$$

Donde a_m es la señal generada para el elemento m del arreglo y τ_m corresponde al retardo descrito en (3.5). El término $\frac{t_p}{2}$ permite asegurarse que la función rectangular empiece en el instante $t = 0$.

De igual forma, se puede sustituir τ_m por la diferencia de fase ϕ_m equivalente descrita en (3.7).

$$\phi_m = \omega_{RF} \left(\frac{\cos(\theta)}{2 \cdot f_{RF}} \right) (m - 1), \quad m = 1, 2, \dots, N \quad (3.7)$$

Reduciendo términos en (3.7) se obtiene la expresión (3.8).

$$\phi_m = [\pi \cdot \cos(\theta)](m - 1), \quad m = 1, 2, \dots, N \quad (3.8)$$

Al sustituir ϕ_m en (3.6), se obtiene la expresión equivalente (3.9).

$$a_m(t) = \cos(\omega_{obs}t - \phi_m) \cdot \text{rect}\left(\frac{t - (t_d + \frac{t_p}{2})}{t_p}\right), \quad m = 1, 2, \dots, N \quad (3.9)$$

Implementación del modelo de la generación de señales recibidas por el arreglo

Las expresiones (3.6) y (3.9) conforman la base del modelo matemático de primera aproximación para las señales recibidas por el arreglo. A partir de ambas, se desarrolló en AWR Microwave Office el diagrama de sistema mostrado en la figura 3.7 empleando los bloques ofrecidos en dicha plataforma. El diagrama de sistema correspondiente permite generar las señales descritas por las expresiones mencionadas anteriormente.

En primera instancia, se utiliza el bloque *TONE* para generar, como su nombre lo indica, un tono real puro con los parámetros de frecuencia y potencia requeridos. La frecuencia de dicho tono se encuentra dada por la suma de dos variables globales dentro del sistema, las cuales corresponden a la frecuencia de operación del arreglo (f_{tx}) y al incremento o decremento de frecuencia causado por el efecto Doppler (f_D). De igual forma, se propone situar la potencia de las señales recibidas en -30 dBm, lo que equivale a $1 \mu W$.

Por otro lado, el bloque *PPULSE* permite establecer el retardo empleado para representar la distancia al objeto mediante la variable t_d , al igual que la PRF del pulso con una variable del mismo nombre. Se indica un ciclo de trabajo de 0.3125 % para obtener el ancho de pulso

deseado de $12.5 \mu s$ a través de la variable global PDC. Las señales generadas por ambos bloques se multiplican para obtener un pulso rectangular único a partir del cual se forman los 32 canales. Finalmente, se tiene al bloque *RFDELAY*, el cual permite añadir la diferencia de fase mediante la definición de la variable tau, tal y como se describe en la expresión (3.4). Al tratarse de un modelo de primera aproximación, se considera un acoplamiento ideal de impedancias en todos los elementos que conforman al sistema, lo que asegura máxima transferencia de potencia.

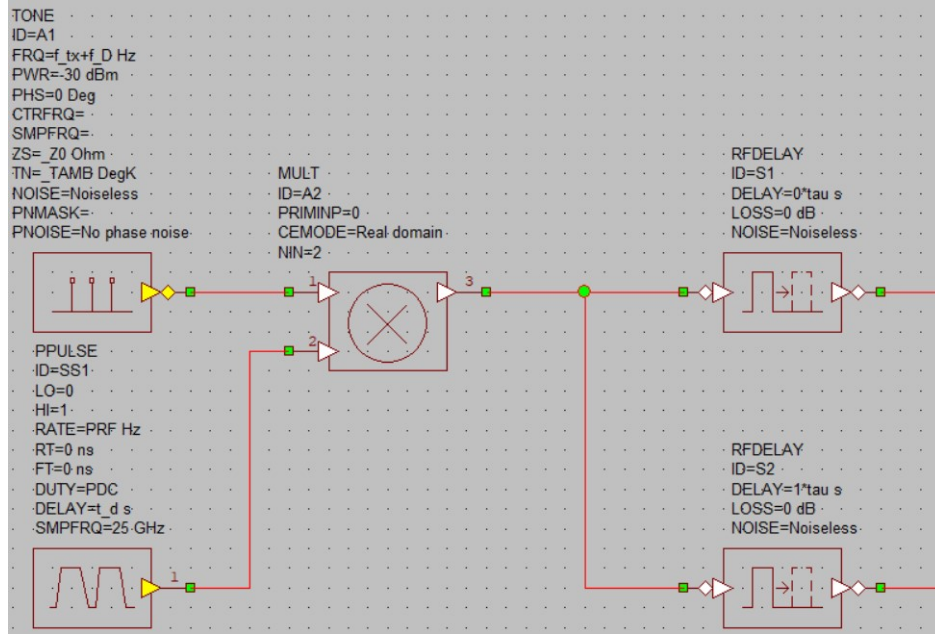


Figura 3.7: Generación de señales recibidas de RF en AWR Microwave Office.

Modelado del acondicionamiento de señales

Posteriormente, se cuenta con la sección correspondiente al acondicionamiento de señales, el cual se propone llevar a cabo conforme al esquema mostrado en la figura 3.8. En primer lugar, se cuenta con la conversión a IF que permite trasladar las señales generadas en RF hasta la frecuencia seleccionada de 10 MHz. Las señales resultantes se filtran y se amplifican, previo a su paso por el ADC correspondiente.

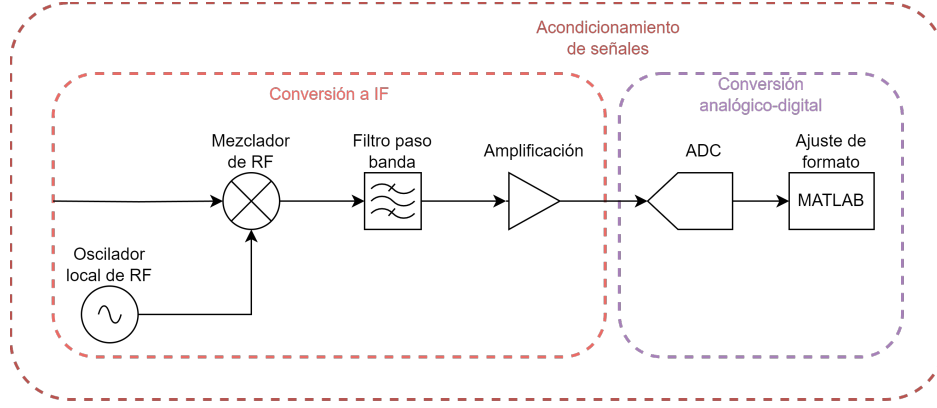


Figura 3.8: Propuesta para el desarrollo del acondicionamiento de señales.

Conforme a lo estipulado anteriormente, se cuenta con una etapa única de conversión hacia IF. Dicha etapa se implementó mediante inyección del lado superior, lo que implica que la frecuencia (f_{LO_1}) del oscilador local es superior a la de las señales suministradas (f_o). La señal proveniente de dicho oscilador local se puede expresar conforme a lo descrito en (3.10).

$$s_{LO_1}(t) = \cos(\omega_{LO_1} t) \quad (3.10)$$

Al multiplicar la señal s_{LO_1} generada por el oscilador local con las señales obtenidas en la etapa anterior, se busca recuperar el término mezcla que contiene la diferencia de frecuencias. De esta forma, la salida del mezclador correspondiente a cada canal se encuentra descrita por la expresión (3.11).

$$b_m(t) = a_m(t) \cdot s_{LO_1}(t) \quad (3.11)$$

Para fines prácticos, de aquí en adelante se analizará únicamente el intervalo de tiempo en donde se recibe un pulso. Desarrollando la expresión (3.11) se obtiene lo siguiente:

$$b_m(t) = 0.5 (\cos([\omega_{LO_1} + \omega_{obs}]t - \phi_m) + \cos([\omega_{LO_1} - \omega_{obs}]t + \phi_m)) \quad (3.12)$$

Como ya se ha mencionado anteriormente, solo resulta de interés el término que contiene la diferencia de frecuencias. Por lo tanto, es necesario incorporar un filtro paso banda que permita suprimir la suma de frecuencias. La ancho de banda de este filtro no necesita ser tan angosto ya que existe una separación bastante amplia entre las frecuencias de los términos mezcla. Sin embargo, se requiere tener especial cuidado en que el filtro no distorsione la señal.

De forma ideal, se obtiene lo siguiente a la salida de la sección de conversión a IF:

$$c_m(t) = A_{c_m} (\cos(\omega_{IF_{obs}} t + \phi_m)) \quad (3.13)$$

Dado que ω_{obs} se encuentra desplazada debido al movimiento del objeto detectado, la frecuencia intermedia resultante no será exactamente la IF esperada. Por lo tanto, $\omega_{IF_{obs}}$ arrastrará este desplazamiento en frecuencia. Dependiendo de la aplicación, esto puede resultar crítico; sin embargo, no existe una afectación crucial en este caso.

En este punto, se requiere suministrar una ganancia G a la señal c_m que permita alcanzar la amplitud requerida.

$$d_m(t) = G \cdot A_{c_m} (\cos(\omega_{IF_{obs}} t + \phi_m)) \quad (3.14)$$

Para este modelo, se asume una amplificación ideal, sin la incorporación de ruido adicional proveniente del elemento encargado de proporcionar ganancia a la salida.

Finalmente, se suministra la señal d_m al ADC, obteniendo (3.15).

$$d_m[n] = A_{d_m} (\cos[\Omega_{IF_{obs}} n + \phi_m]) \quad (3.15)$$

Al pasar del dominio continuo al dominio discreto, ω_{IF} se convierte en Ω_{IF} según lo descrito en la expresión (2.3).

Implementación del modelo del acondicionamiento de señales

Una vez que se han asentado las expresiones que permiten describir el modelo matemático de la segunda sección, se procedió a desarrollar el diagrama de sistema correspondiente en AWR Microwave Office, mostrado en la figura 3.9.

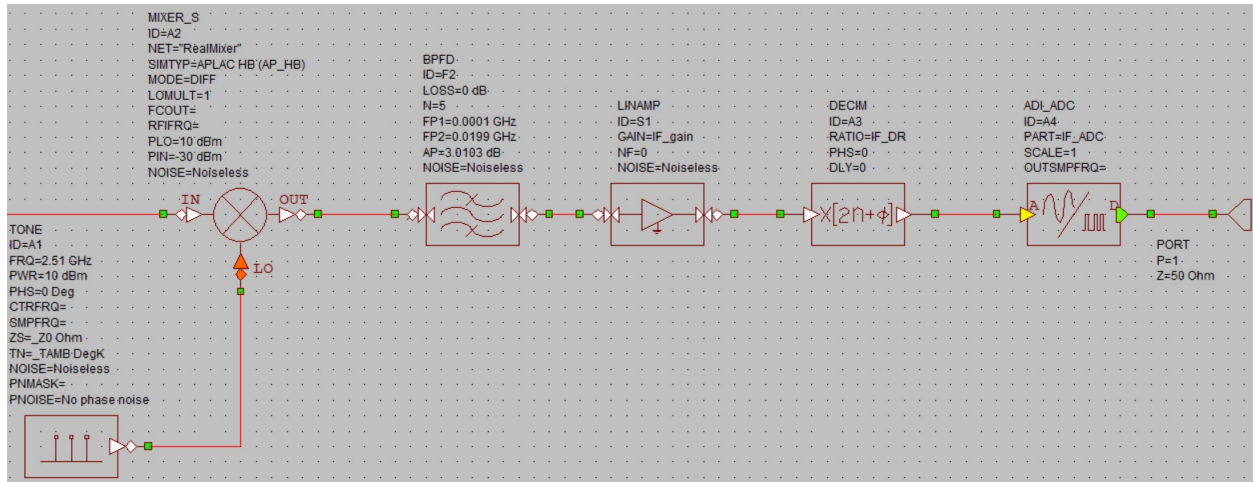


Figura 3.9: Modelo del acondicionamiento de señales.

Se seleccionó el mezclador MM1-0212LS de la empresa Marki Microwave para realizar la conversión hacia IF, a partir de las características que ofrece y la disponibilidad del modelo correspondiente en AWR Microwave Office. Este mezclador permite trasladar la señal de 2.5 GHz hasta 10 MHz, requiriendo de una potencia de entre 1 y 15 dBm para su funcionamiento [54]. El bloque *MIXER_S* permite simular las características de dicho mezclador a partir del modelo que ofrece la empresa para la plataforma de desarrollo. Se selecciona el modo diferencia, indicando la potencia de la señal de entrada y eligiendo 10 dBm como la potencia del oscilador local. Por consiguiente, el bloque correspondiente al oscilador local, se configura a una frecuencia de 2.51 GHz y una potencia de 10 dBm. De igual forma, se indica el modelo del mezclador referenciado como se aprecia en la figura 3.10.

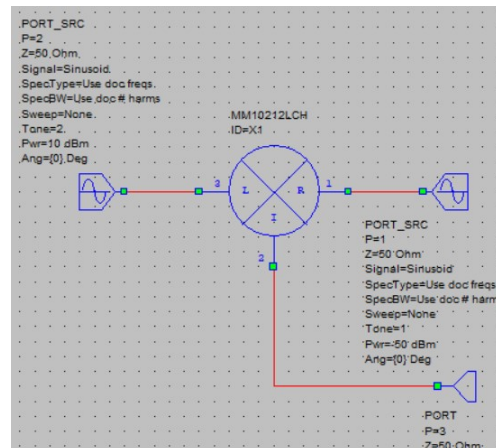


Figura 3.10: Mezclador MM1-0212LS incorporado en el sistema.

A continuación, se coloca un filtro paso banda tipo Bessel de quinto orden con frecuencia central igual a 10 MHz y un ancho de banda igual a 19.8 MHz, considerando un factor de calidad de 100. Se opta por utilizar un filtro tipo Bessel ya que ofrece una respuesta de fase lineal, evitando que se distorsione la señal. La separación entre los términos correspondientes a la suma y diferencia de frecuencias permite prescindir de un filtro de orden mayor, tomando en cuenta las condiciones ideales del sistema hasta el momento. Así mismo, se cuenta con un bloque de amplificación lineal, cuya ganancia se indica a través de la variable `IF_gain`, la cual debe ser suficiente para situarse por encima de la amplitud mínima requerida.

Por otro lado, la etapa de conversión analógica-digital requiere un bloque de decimación que permite indicar al diagrama de sistema que existe un cambio en la frecuencia de muestreo. La plataforma de desarrollo toma 25 GHz como frecuencia de muestreo, por lo que se le indica que se debe reducir dicha frecuencia 250 veces para obtener una frecuencia de 100 MHz. Por otro lado, dadas las características estipuladas para el subsistema correspondiente, se eligió al AD9268-105 de la empresa Analog Devices para llevar a cabo la conversión analógico-digital. Este ADC cuenta con una longitud de palabra de 16 bits, un intervalo de entrada de 2 V y una tasa de muestreo de hasta 105 Msps [55]. El bloque correspondiente permite simular dichas características, incorporando una latencia de trece muestras. A la salida del convertidor, se obtiene la representación binaria desplazada de las señales, por lo que será necesario ajustar el formato.

Los datos generados en AWR Microwave Office por el diagrama de sistema correspondiente se exportan hacia MATLAB mediante un *script*. Para ello, se propone designar una ventana de simulación con una duración de 81.92 μ s, eligiendo despreciar el periodo de latencia del ADC. Por lo tanto, se generan 8192 muestras a partir de las cuales se realizará el procesamiento digital de señales.

La conexión entre AWR Microwave Office y MATLAB se realiza aprovechando el modelo de objetos componentes (COM, por sus siglas en inglés) creado por Microsoft. A través de COM, MATLAB extrae las mediciones que corresponden a la salida del ADC en cada canal. De igual forma, se empleó esta misma técnica para crear el modelo completo en AWR Microwave Office de forma automática, ya que el proceso de generación manual para los 32 canales resulta poco práctico.

Posteriormente, se realiza una conversión del formato numérico, pasando de la representación binaria desplazada a la representación en complemento a dos. Esto se logra al aplicar una *XOR* al bit más significativo (MSB, por sus siglas en inglés) del número en cuestión.

Así mismo, se realiza una segunda conversión en donde se establece un formato de punto fijo $Q0.15$.

Estructura general del procesamiento digital de señales

A partir de las señales digitales de IF obtenidas, se puede llevar a cabo el procesamiento digital de señales. Con este fin, se propone seguir el esquema presentado en la figura (3.11), empleando una frecuencia de operación de 100 MHz y un formato de punto fijo $Q0.15$ que se irá ajustando según se requiera.

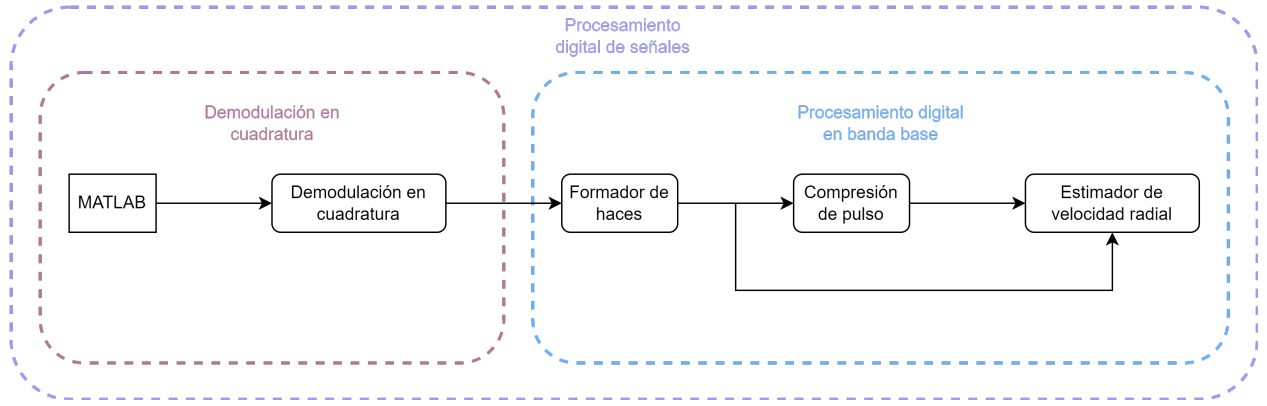


Figura 3.11: Propuesta para el desarrollo del procesamiento digital de señales.

Modelado de la demodulación en cuadratura

Como ya se mencionó en repetidas ocasiones, la primer sección de esta etapa estará encargada de realizar la demodulación en cuadratura en cada canal. Para llevar a cabo esta técnica, se utiliza el esquema planteado en la figura 2.6, suministrando la señal d_n correspondiente. La frecuencia angular discreta de los NCO (Ω_{LO_2}) debe ser igual a la frecuencia angular discreta correspondiente a la IF seleccionada. Si esto se cumple, se obtiene las señales descritas en (3.16) y (3.17) a la salida de los multiplicadores.

$$g_{m_1}[n] = A_{g_{m_1}} (\cos[\Omega_D n + \phi_m] + \cos[(\Omega_{LO_2} + \Omega_{IF_{obs}})n + \phi_m]) \quad (3.16)$$

$$g_{m_2}[n] = A_{g_{m_2}} (-\sin[\Omega_D n + \phi_m] - \sin[(\Omega_{LO_2} + \Omega_{IF_{obs}})n + \phi_m]) \quad (3.17)$$

Al igual que en la etapa de conversión hacia IF, solo resultan de interés aquellos términos que contienen la diferencia de las frecuencias. Por lo tanto, es necesario incorporar un filtro paso bajas que permita suprimir el término con la suma de las frecuencias. La frecuencia de corte de dicho filtro se encontrará en función del intervalo de frecuencias esperadas debido

al efecto Doppler. Adicionalmente, el filtro colocado no debe distorsionar la señal ya que, de lo contrario, no será posible estimar los parámetros del objeto. Por consiguiente, se obtienen las señales descritas en (3.18) y (3.19) a la salida de los filtros.

$$g_{m_I}[n] = A_{g_{m_I}} (\cos[\Omega_D n + \phi_m]) \quad (3.18)$$

$$g_{m_Q}[n] = A_{g_{m_Q}} (-\sin[\Omega_D n + \phi_m]) \quad (3.19)$$

Modelado del procesamiento digital en banda base

En este punto, se puede apreciar de forma mucho más evidente que las señales obtenidas contienen los parámetros a estimar del objeto. Sin embargo, pueden identificarse dos problemáticas esenciales: la definición del orden en la estimación de los parámetros y el desarrollo práctico de los componentes que integran el segundo subsistema.

Para resolver la primer problemática, se propone realizar primero la formación de haces, de tal forma que la compresión de pulso sea aplicada únicamente sobre los ocho haces formados en lugar de los 32 elementos. En [56], se puede apreciar una arquitectura similar en donde se sitúa la formación de haz previo a la compresión de pulso. Posteriormente, se puede emplear el resultado de la compresión de pulso para llevar a cabo la estimación de la velocidad radial.

Con el fin de realizar la formación de haces, se pueden emplear las componentes resultantes para obtener la representación polar de la señal. Dicha representación resulta particularmente útil ya que la aplicación de los pesos se reduce a la suma de las fases y la multiplicación de las magnitudes. Por consiguiente, es necesario obtener la magnitud y la fase a partir de las señales descritas en las expresiones (3.18) y (3.19). La magnitud se obtiene elevando al cuadrado cada componente, sumando ambas cantidades y calculando su raíz cuadrada:

$$A_{g_m}[n] = \sqrt{g_{m_I}[n]^2 + g_{m_Q}[n]^2} \quad (3.20)$$

Por otro lado, la fase se determina mediante la función *atan2* que recibe como parámetros la componente en cuadratura y la componente en fase.

$$\Phi_{g_m}[n] = \text{atan2}(g_{m_Q}[n], g_{m_I}[n]) \quad (3.21)$$

Las señales resultantes se encuentran descritas por la siguiente expresión:

$$h_m[n] = A_{g_m}[n] e^{j\Phi_{g_m}[n]} \quad (3.22)$$

Ignorando Ω_D en (3.18) y (3.19), se puede observar que Φ_{h_m} es igual a $-\phi_m$ al evaluar

la expresión (3.21). Esto resulta de mucha utilidad para la formación de haces ya que se necesita del complejo conjugado de h_m para compensar la fase. Por lo tanto, la fase de los pesos generados ϕ_{w_n} deberá ser igual a ϕ_m , obteniendo la diferencia de fase para cada elemento por medio de (3.8). Por otro lado, la magnitud de los pesos para la formación de haces se obtiene a partir de la expresión (2.15).

En este punto solo resta aplicar los pesos, sumando las fases y multiplicando las magnitudes.

$$p_m[n] = (A_{g_m}[n] \cdot A_{w_n})e^{j(\Phi_{g_m}[n] + \phi_{w_n})} \quad (3.23)$$

$$p_m[n] = A_{p_m}[n]e^{j\phi_{p_m}[n]} \quad (3.24)$$

El paso final en la formación de haces consiste en la suma de las 32 señales obtenidas tras la aplicación de los pesos correspondientes a cada haz. Para realizar dicha suma, es necesario obtener las componentes real e imaginaria de p_m . Con este fin, se multiplica la magnitud obtenida por la función coseno evaluada en la fase correspondiente, obteniendo la componente real de la señal p_m .

$$q_m[n] = A_{p_m}[n] \cdot \cos(\phi_{p_m}[n]) \quad (3.25)$$

Por otro lado, la componente imaginaria de la señal p_m se obtiene al multiplicar la magnitud por la función seno evaluada en la fase correspondiente.

$$r_m[n] = A_{p_m}[n] \cdot \sin(\phi_{p_m}[n]) \quad (3.26)$$

Por último, solo resta sumar todas las componentes reales e imaginarias para cada haz.

$$q[n] = \sum_{k=1}^N q_m[n] \quad (3.27)$$

$$r[n] = \sum_{k=1}^N r_m[n] \quad (3.28)$$

Por medio de $q[n]$ y $r[n]$, se puede obtener la magnitud de cada haz, a partir de lo cual se puede estimar el ángulo de elevación al identificar el haz con mayor magnitud.

A partir de estas señales, se realiza la compresión de pulso en cada haz mediante un filtro en concordancia. Este filtro puede implementarse en el dominio del tiempo a través de la con-

volución descrita en la expresión (2.20). Sin embargo, el filtro también puede implementarse en el dominio de la frecuencia mediante el uso de la FFT. En este método, la convolución se sustituye por la multiplicación de las transformadas de Fourier de la señal de entrada y la respuesta del filtro. Como se aprecia en [57], este enfoque suele preferirse a partir de la mejora que ofrece en términos de eficiencia computacional.

Para la obtención del modelo correspondiente a la compresión de pulso, se puede partir del caso idealizado en donde el objeto se encuentra estático, el primer elemento del arreglo no cuenta con fase adicional añadida y la formación de haces anuló la diferencia de fase en los demás elementos. En este caso, se obtiene un pulso rectangular puramente real de ancho igual a t_p y un retraso $t_d + \frac{t_p}{2}$, lo que implica dividir ambos entre el periodo de muestreo t_s para contar con el ancho y el retraso en términos de muestras (n_p y $n_d + \frac{n_p}{2}$, respectivamente). La transformada de Fourier en tiempo discreto para este pulso rectangular corresponde a lo descrito en la expresión (3.29).

$$Q(e^{j\omega}) = n_p \cdot \text{sinc}\left(n_p \frac{\omega}{2}\right) \cdot e^{-j\omega(n_d + \frac{n_p}{2})} \quad (3.29)$$

Por consiguiente, el filtro en concordancia deberá tener una respuesta en frecuencia según lo estipulado por la expresión (3.30).

$$H(e^{j\omega}) = n_p \cdot \text{sinc}\left(n_p \frac{\omega}{2}\right) \cdot e^{j\omega \frac{n_p}{2}} \quad (3.30)$$

Al aplicar el filtro en concordancia, se obtiene lo denotado en (3.31).

$$Y_{mf}(e^{j\omega}) = n_p^2 \text{sinc}^2\left(n_p \frac{\omega}{2}\right) \cdot e^{-j\omega n_d} \quad (3.31)$$

El último paso en la compresión de pulso consta en la obtención de la transformada inversa de (3.31), dando como resultado una función triangular cuyo punto máximo se encuentra en n_d .

$$y_{mf}[n] = A_{y_{mf}} \text{tri}[n - n_d] \quad (3.32)$$

En consecuencia, si se detecta la muestra en donde ocurre el máximo de la función triangular, será posible obtener la distancia al objeto. La muestra n_d deberá ser multiplicada por t_s , obteniendo el instante en que se recibió el pulso, a partir de lo cual se puede evaluar la expresión (2.1).

En caso de contarse con un objeto en movimiento, el resultado de la compresión de pulso corresponderá a una señal sinusoidal con una envolvente triangular. Si la velocidad radial

del objeto es baja, el rendimiento del filtro en concordancia no se verá afectado de forma sustancial. Sin embargo, conforme aumenta la frecuencia derivada del efecto Doppler, la señal se distorsionará y será necesario compensar dicha frecuencia. Para el caso particular del sistema que se modela a lo largo de este trabajo, se asume que la afectación causada por la aparición del efecto Doppler es mínima y, por lo tanto, despreciable.

Al conocer n_d con certeza, se puede realizar la estimación de la velocidad radial mediante la diferenciación de la fase de cada haz. Este procedimiento permite obtener la pendiente de la fase durante la recepción del pulso, a partir de lo cual se calcula Ω_D . Con este fin, se obtiene la diferencia de fase existente entre dos instantes del pulso recibido, tal y como se muestra en (3.33).

$$\Omega_D = \frac{\Delta\phi_h}{\Delta n} = \frac{\phi_{h_B} - \phi_{h_A}}{n_B - n_A} \quad (3.33)$$

La frecuencia obtenida se encuentra expresada en radianes sobre muestra, por lo que se divide entre el tiempo de muestreo multiplicado por 2π para estimar la frecuencia en Hertz.

$$f_D = \frac{\Delta\phi_h}{\Delta n} \cdot \frac{1}{2\pi t_s} \quad (3.34)$$

A continuación, se puede sustituir f_o en (2.21) por la suma de la frecuencia transmitida originalmente (f_{tx}) más f_D , tomando en cuenta que el sistema de recepción se encuentra estático.

$$f_{tx} + f_D = \frac{c}{c + v_s} f_{tx} \quad (3.35)$$

Despejando v_s en (3.35) se obtiene la expresión (3.36), a partir de lo cual se puede completar la estimación de la velocidad radial.

$$v_s = \frac{f_{tx} \cdot c}{f_{tx} + f_D} - c \quad (3.36)$$

En caso de que v_s sea positivo, el objeto se estará alejando del sistema de recepción, lo que provoca una disminución en la frecuencia observada. Por el contrario, si se obtiene un resultado negativo, el objeto se estará acercando al sistema de recepción, por lo que existirá un aumento en la frecuencia observada.

Implementación del modelo de la demodulación en cuadratura

A partir de las expresiones presentadas para la segunda etapa, se generó el modelo correspondiente en MATLAB y Simulink que permite realizar el procesamiento digital de señales. En primer lugar, se generó un bloque de subsistema en Simulink que permitiese llevar a cabo la demodulación en cuadratura, mostrado en la figura 3.12. Dicho bloque toma como entrada la señal digital de IF acondicionada y la multiplica por los NCO requeridos para llevar a cabo el esquema de demodulación. Posteriormente, se cuenta con un filtro paso bajas que permite suprimir los términos no deseados.

Los bloques encargados de prestar la función de NCO utilizan tablas de búsqueda de 10 elementos que permiten generar señales sinusoidales de 10 MHz a partir de una frecuencia de muestreo de 100 MHz. Se utilizó el mismo formato de punto fijo de la entrada para los NCO, dado que su salida puede expresarse en dicho formato al considerarse una amplitud unitaria. Por lo tanto, se aprecia que al realizar la multiplicación de la señal de entrada con aquella generada por el NCO, la señal resultante puede mantener un formato $Q0.15$ al no existir la posibilidad de sobreflujo.

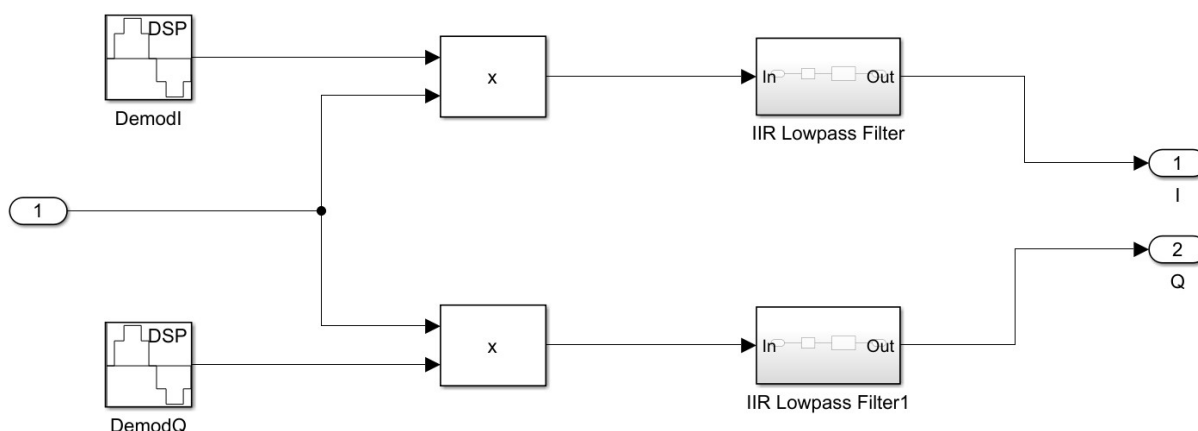


Figura 3.12: Bloque correspondiente a la demodulación en cuadratura.

A continuación, se cuenta con un filtro Butterworth paso bajas IIR de cuarto orden con frecuencia de corte de 5 MHz para cada componente. El filtro se diseñó empleando la herramienta *Filter Designer* de MATLAB, indicando que se requiere mantener el formato de punto fijo $Q0.15$ a la salida y se desea contar con precisión completa en las secciones intermedias del filtro.

Implementación del modelo del procesamiento digital en banda base

Las componentes en banda base recuperadas se someten al bloque de subsistema mostrado en la figura 3.13, en donde se convierte la representación rectangular de la señal en su forma polar. Con el fin de estimar la magnitud, se cuenta con dos bloques encargados de elevar cada componente al cuadrado, sumando las señales resultantes y obteniendo su raíz cuadrada. Dado que las componentes están representadas en el formato numérico signado $Q0.15$, al elevarlas al cuadrado es posible ajustar la representación a un formato no signado $Q0.16$. Una vez sumados los cuadrados de ambas componentes, se requiere realizar un ajuste al formato no signado $Q1.15$, bajo el supuesto de que se emplea la totalidad del rango dinámico disponible. Del mismo modo, se puede mantener el formato numérico para el resultado de la raíz cuadrada.

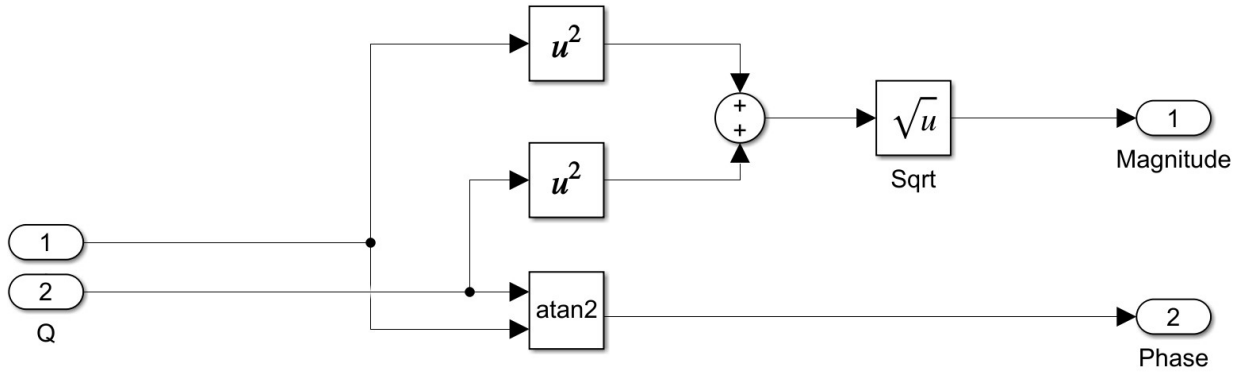


Figura 3.13: Bloque correspondiente al cambio de representación rectangular a polar.

La fase se puede obtener mediante la función $atan2$, la cual permite estimar la tangente inversa en los cuatro cuadrantes del plano cartesiano. El bloque correspondiente implementa $atan2$ a partir de una tabla de búsqueda de 16 elementos con un formato signado $Q2.13$. En caso de que el resultado deseado no se encuentre entre estos elementos, el bloque es capaz de interpolar los resultados al trazar una línea entre los puntos adyacentes y devolver el punto de esa línea correspondiente a la entrada [58] [59].

Una vez que se cuenta con la representación polar de la señal, se procede a la aplicación de los pesos requeridos para la formación de haces. Para cada elemento del arreglo, los pesos se calcularon mediante un *script* en MATLAB que determina la magnitud y la fase correspondientes a partir de las expresiones (2.15) y (3.8). Se utiliza un formato no signado $Q1.15$ para la magnitud ya que la distribución elegida de coseno sobre pedestal arroja valores en un intervalo entre 0 y 2. Por otro lado, se utiliza un formato no signado $Q3.13$ para la fase al producir valores situados en un intervalo de 0 a 2π . La aplicación de pesos se lleva a

cabo mediante la multiplicación de las magnitudes y la suma de las fases. La multiplicación resulta en un formato no signado $Q2.14$, mientras que la suma de fases resulta en un formato signado $Q4.11$.

El siguiente paso consta en regresar las señales a su representación rectangular con el fin de llevar a cabo la suma de los resultados correspondientes a la aplicación de pesos en todos los elementos del arreglo. La fase se ajusta mediante una función de MATLAB que sitúa el resultado en un intervalo de $-\pi$ a π , lo que permite suministrar dicha fase a los bloques encargados de estimar las funciones trigonométricas requeridas por las expresiones (3.25) y (3.26). Dichos bloques utilizan tablas de búsqueda de 16 elementos al igual que *atan2*, empleando un formato signado $Q0.15$ para sus salidas. La multiplicación de los términos de magnitud y frecuencia conduce al ajuste del formato numérico a $Q2.13$ signado.

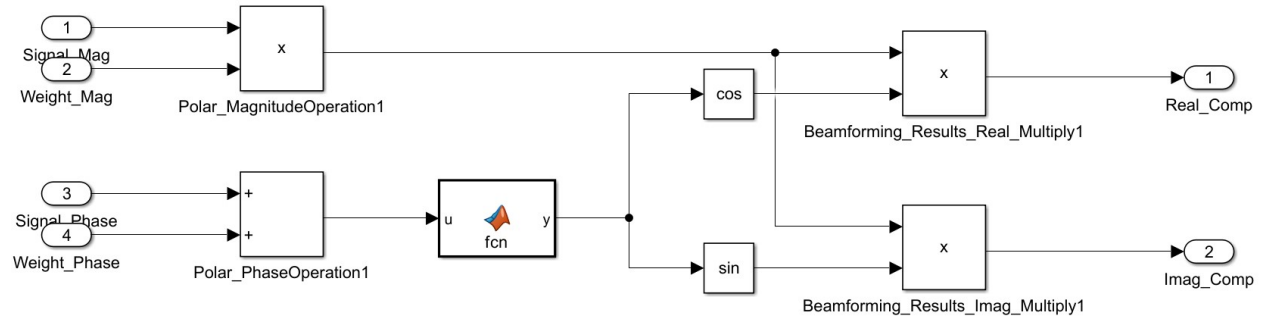


Figura 3.14: Bloque correspondiente a la aplicación de pesos y el cambio de representación.

Se efectúa la suma de las componentes correspondientes a la representación rectangular de todos los elementos del arreglo para cada haz formado. Tomando en cuenta el formato de las componentes y el número de elementos particular del arreglo, se ajusta el formato numérico de las salidas a formato signado $Q7.8$.

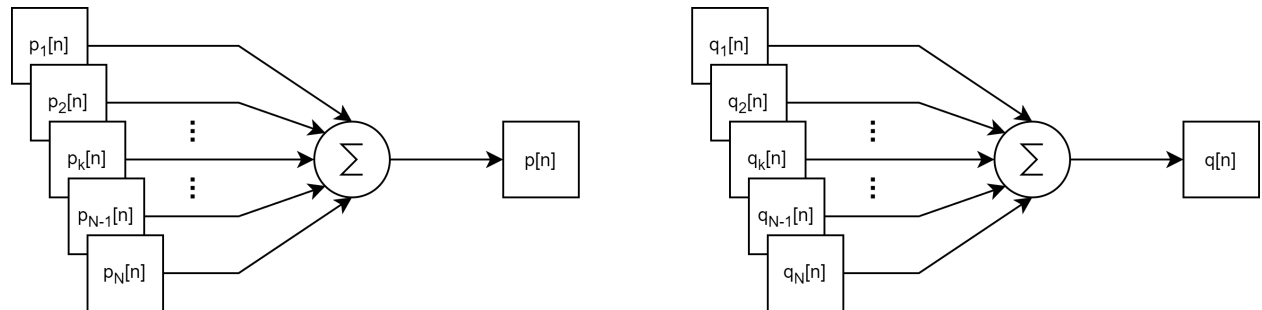


Figura 3.15: Suma de componentes reales e imaginarias para todos los elementos del arreglo.

Realizando pruebas preliminares para este subsistema, se dilucidó la posibilidad de opti-

mizar el formato numérico en dos instancias. Las pruebas realizadas emplearon una versión modificada del caso de prueba que no incluye la velocidad radial del objeto y provee mayor ganancia en la etapa de conversión hacia IF, de tal forma que se abarca el intervalo de entrada completo del ADC.

La primera instancia corresponde a la multiplicación de los términos de magnitud y frecuencia al realizar el cambio de representación polar a rectangular. En este caso, se puede ajustar el formato numérico a $Q0.15$ signado. Esto repercute en la suma subsecuente de las señales correspondientes a todos los elementos del arreglo para cada haz formado. Las pruebas mostraron la posibilidad de ajustar el formato numérico a $Q4.11$ signado, permitiendo una mayor resolución.

De esta forma, se obtienen ocho haces, compuestos por sus respectivas componentes, a partir de las cuales se puede determinar el ángulo de elevación. Con este fin, basta obtener la magnitud a partir de sus componentes, eligiendo el ángulo de elevación que da origen al haz con la mayor magnitud como la estimación de dicho parámetro.

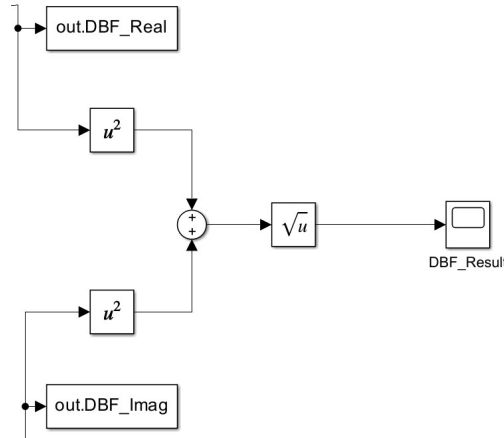


Figura 3.16: Obtención de la magnitud de los haces y exportación de datos.

El siguiente subsistema a desarrollar a partir de las expresiones obtenidas corresponde a la compresión de pulso. Para este subsistema, se optó por emplear un *script* en lugar de un sistema en Simulink, lo que permite realizar las operaciones sobre vectores completos de información. Las transformadas se implementan mediante los objetos de sistema *dsp.FFT* y *dsp.IFFT* ofrecidos por MATLAB, definiendo una longitud de 2048 muestras para ambas.

Las componentes reales e imaginarias de los ocho haces son el punto de partida para la compresión de pulso. En primer lugar, se exportan las señales procesadas por el modelo de Simulink como se aprecia en la figura 3.16 y se aplica una decimación por un factor de 4 para

obtener el número de muestras deseado. Posteriormente, se obtiene su FFT, designando un formato signado $Q12.3$ para el espectro resultante.

Por otro lado, se genera el pulso rectangular que servirá de referencia para el filtro en concordancia. Dicho pulso cuenta con la misma duración pero se encuentra invertido en el tiempo, lo que permite obtener la respuesta en frecuencia descrita por (3.30). Al igual que con la señal de entrada, se obtiene su FFT, estableciendo un formato signado $Q9.6$ para el espectro resultante. Ambos espectros complejos se encuentran dados en su representación rectangular.

A continuación, se realiza la multiplicación compleja de ambos espectros, expandiendo la longitud de palabra a 24 bits y asentando el formato de la salida a $Q21.2$. Este formato se escoge con el fin de representar completamente la parte entera del producto de ambos espectros. Posteriormente, se aplica la IFFT a la señal resultante, optando por mantener el mismo formato numérico y concluyendo la aplicación del filtro en concordancia.

Una vez se ha efectuado la aplicación del filtro en concordancia, se procede a identificar el punto máximo de la salida del mismo. Para este fin, se elevan al cuadrado las componentes real e imaginaria de la señal y se suman, en un procedimiento análogo al cálculo de la magnitud, pero omitiendo la raíz cuadrada al no ser necesaria para identificar el punto máximo. El resultado de tales operaciones hará necesario expandir la longitud de palabra a 42 bits y ajustar el formato numérico a $Q42.3$ no signado.

La localización del máximo se realiza mediante la función *max*, la cual permite identificar la posición del elemento con mayor magnitud en el vector correspondiente a la señal, al igual que el propio valor máximo. Dado que se cuenta con 2048 muestras en la ventana de muestreo seleccionada, se requiere de una variable entera no signada de al menos 11 bits para almacenar la posición del valor máximo. Así mismo, se puede obtener un factor de conversión a partir de (2.1) al multiplicar la velocidad de propagación de las ondas por el tiempo al que equivale cada muestra, tomando en cuenta la decimación aplicada.

$$\alpha_R = \frac{c \cdot \alpha_d \cdot t_s}{2} \quad (3.37)$$

Donde α_R es el factor de conversión que permite convertir el número de muestra en distancia al objeto, α_d es el factor de decimación y t_s es el tiempo de muestreo inicial. En este caso, se obtiene que cada muestra representa una distancia de 6 metros, necesitando de 3 bits para representar dicha cantidad no signada. En consecuencia, la distancia al objeto se puede representar mediante una variable entera no signada de al menos 14 bits.

El último subsistema a desarrollar corresponde al estimador de velocidad radial, el cual

recibe como entrada las señales resultantes de la formación de haces y de la decimación previa a la compresión de pulso. El *script* que representa este estimador toma como referencia la muestra asociada al valor máximo de la compresión de pulso. A partir de este punto, se define una ventana temporal de $12 \mu\text{s}$ que inicia cinco muestras después del valor máximo detectado con el fin de mitigar posibles errores en la estimación inicial. Dentro de esta ventana se calcula la fase de las señales decimadas mediante la función *atan2* ofrecida por MATLAB para valores de punto fijo, dando como resultado un formato numérico $Q2.13$ signado de 16 bits.

La función *atan2* devuelve valores en el intervalo $[-\pi, \pi]$, generando un salto abrupto cuando la componente imaginaria cambia de signo. Dado que no es posible conocer *a priori* si esta componente en banda base variará de signo dentro de la ventana temporal considerada, fue necesario incorporar en el código una sección que desplaza el ángulo obtenido en $\pm 2\pi$. Con este ajuste, equivalente al funcionamiento de la función *unwrap*, se eliminan posibles discontinuidades, lo que permite calcular la diferencia de fase entre los extremos de la ventana temporal. Como consecuencia, se requiere expandir la longitud de palabra a 17 bits para preservar la resolución de la parte fraccionaria.

Tomando en cuenta que la estimación de la velocidad radial se obtiene a partir del desplazamiento en frecuencia inducido por el efecto Doppler, este modelo asume que la magnitud de dicho desplazamiento es tal que, como máximo, puede producir una única discontinuidad.

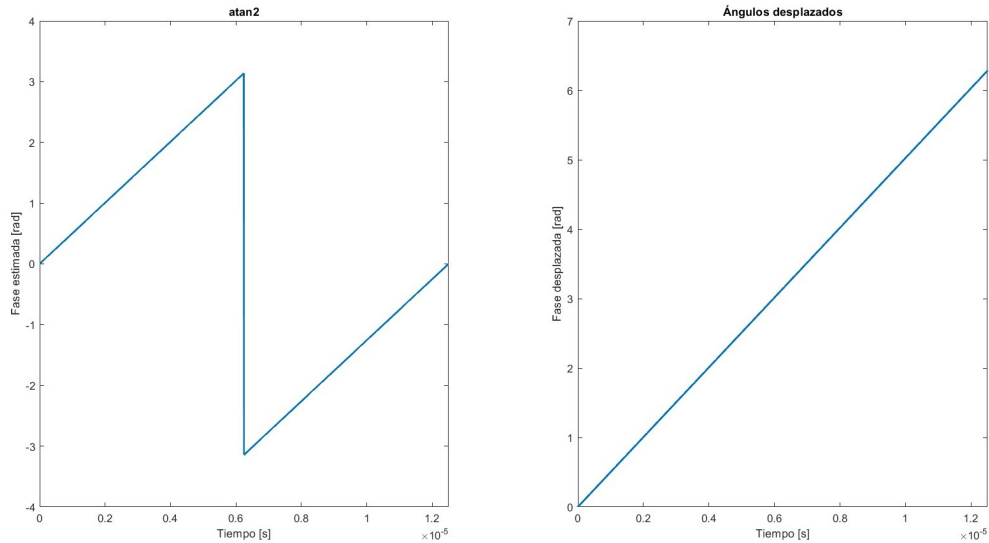


Figura 3.17: Desplazamiento de ángulos obtenidos mediante *atan2*.

A continuación, la diferencia de fase obtenida se multiplica por el factor de conversión mostrado en la expresión (3.38).

$$\alpha_{f_D} = \frac{1}{n_s \cdot 2\pi \cdot t_{ds}} \quad (3.38)$$

Donde n_s es el número de muestras que abarca la ventana temporal y t_{ds} es el tiempo de muestreo correspondiente a las señales decimadas. Para las condiciones descritas a lo largo de este modelo, el factor de conversión adquiere un valor de 13262.912 por lo que requiere una longitud de palabra de 24 bits y un formato no signado Q14.10.

Recordando las expresiones (3.18) y (3.19), se aprecia que la frecuencia obtenida hasta el momento contará con el signo contrario a la frecuencia Doppler real. Por lo tanto, es necesario multiplicar por menos uno para obtener el signo correcto, de modo que el resultado final de la conversión se expresa mediante un formato signado Q13.11.

Por último, solo resta sustituir la frecuencia estimada en la expresión (3.36), junto con los parámetros del sistema, para obtener la estimación de la velocidad radial. Debido a que existe una diferencia sustancial en los órdenes de magnitud de los parámetros involucrados, se opta por realizar la última operación con un formato de punto flotante.

3.3. Instrumentación del módulo de compresión de pulso en un FPGA

3.3.1. Planeación del sistema

Como parte de los objetivos planteados al inicio de este trabajo, se solicitó desarrollar la instrumentación requerida para el módulo de compresión de pulso modelado en la primera sección de este capítulo. Dicho módulo tiene como propósito estimar la distancia a la que se encuentra un objeto respecto al sistema de recepción mediante un filtro en concordancia.

El módulo se sitúa posterior a la formación de haces, tal y como se describió durante el modelado del procesamiento del bloque de recepción. En principio, se requiere desarrollar un sistema capaz de realizar simultáneamente la compresión de pulso en los ocho haces resultantes. Con este fin, se propone diseñar un núcleo de procesamiento que pueda ser fácilmente replicado, permitiendo llevar a cabo la compresión de pulso de todos los haces de forma paralela. La instrumentación que se desarrollará contempla un ángulo de azimuth fijo, suponiendo un alcance teórico máximo de 600 km y una frecuencia de operación mínima de 100 MHz.

3.3.2. Desarrollo de concepto

A grandes rasgos, el módulo de compresión de pulso propuesto consta de dos etapas principales: la primera comprende el desarrollo de un filtro en concordancia y la segunda corresponde a la estimación de la distancia al objeto.

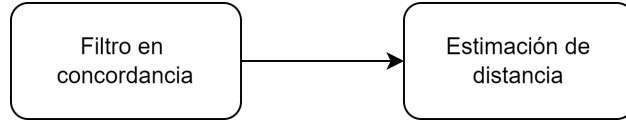


Figura 3.18: Etapas principales del módulo de compresión de pulso.

Etapas de filtro en concordancia

Como se mencionó en la sección anterior, existen dos alternativas para implementar el filtro en concordancia. La primera consiste en su realización en el dominio del tiempo mediante la convolución, mientras que la segunda corresponde a su implementación en el dominio de la frecuencia a través de la multiplicación compleja de los espectros de la señal de entrada y la respuesta del filtro.

Con el propósito de seleccionar un método para la implementación del filtro en concordancia, se establecen dos criterios principales para determinar la alternativa más adecuada. El primer criterio corresponde al número de etapas de procesamiento requeridas por cada método, ya que, en general, un mayor número de etapas implica un sistema de control más complejo. El segundo criterio considera el impacto que tiene la longitud del vector de muestras sobre el tiempo aproximado de ejecución para cada método. En este sentido, como primera aproximación, puede recurrirse a la notación Big O, la cual permite describir el comportamiento asintótico de los algoritmos, es decir, la tasa con la que crece su costo computacional conforme aumenta el tamaño de la entrada de manera considerable [60].

La implementación en el dominio del tiempo representa una alternativa más sencilla en términos de control, ya que requiere únicamente de una máquina encargada de realizar la operación de convolución correspondiente. Dicha máquina efectúa la convolución entre la señal de entrada de N muestras y un pulso de referencia con las mismas características que el pulso radiado en banda base. La convolución requiere de N^2 multiplicaciones y $(N - 1)^2$ sumas, por lo que su complejidad es del orden de $O(N^2)$ al considerar únicamente las operaciones que tienen mayor impacto en el comportamiento asintótico. Esto la convierte en una alternativa poco eficiente para vectores de muestras de tamaño considerable.

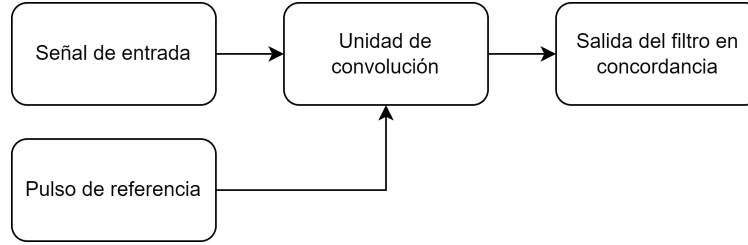


Figura 3.19: Implementación del filtro mediante la convolución.

Por otra parte, la implementación en el dominio de la frecuencia mediante multiplicación compleja exige un sistema de control más elaborado, ya que requiere principalmente de tres máquinas en secuencia: la FFT, el multiplicador complejo y la IFFT. La señal de entrada se suministra a la FFT, cuyos resultados se entregan al multiplicador complejo junto con el espectro del pulso de referencia. El producto espectral resultante se le suministra a la IFFT, obteniendo a la salida el pulso comprimido en el dominio del tiempo. No obstante, los algoritmos de cada etapa presentan una complejidad inferior a aquella de la convolución directa. En particular, la FFT y la IFFT son las operaciones más costosas, con una complejidad de $O(N \cdot \log(N))$ cada una, como se mencionó en el marco teórico, lo que en conjunto se traduce en una mayor eficiencia global, tal como se señala en [57].

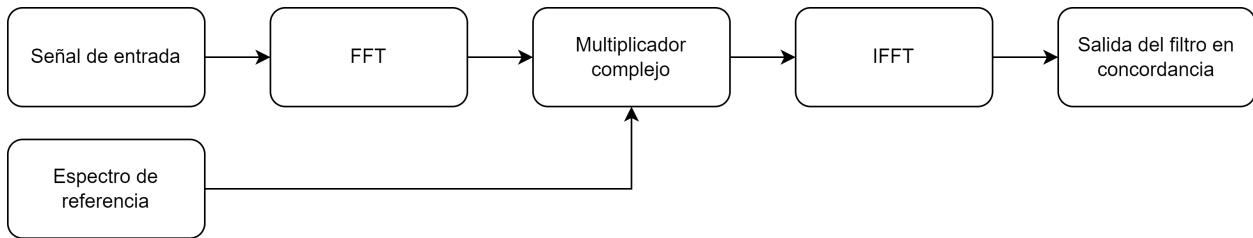


Figura 3.20: Implementación del filtro mediante la multiplicación compleja.

Con base en las ventajas y desventajas que ofrece cada método, se opta por implementar el filtro en concordancia en el dominio de la frecuencia. El principal motivo para elegir esta técnica fue la mejora en el rendimiento que ofrece, pues constituye también la base de los algoritmos de convolución rápida [61].

Etapas de estimación de distancia

La estimación de distancia sigue un esquema similar al propuesto durante el desarrollo del modelo para la compresión de pulso. Este método consiste en obtener la suma de los cuadrados de las componentes obtenidas a la salida del filtro en concordancia, con el fin de determinar la muestra que presenta el valor máximo.

La identificación de dicha muestra puede instrumentarse mediante una estructura compuesta por cuatro registros y un restador. En dicha estructura, se emplea un registro para almacenar el valor de la muestra actual y otro para conservar su índice, además de dos registros adicionales destinados al almacenamiento del valor máximo actual y su posición dentro del vector de muestras. Los valores de los registros correspondientes a la muestra actual y al valor máximo se suministran al restador, el cual permite determinar si el valor actual es mayor que el máximo registrado. En tal caso, se actualizan los registros asociados, entregando la posición del valor máximo a la entidad encargada de calcular la distancia, obteniendo así la estimación correspondiente.

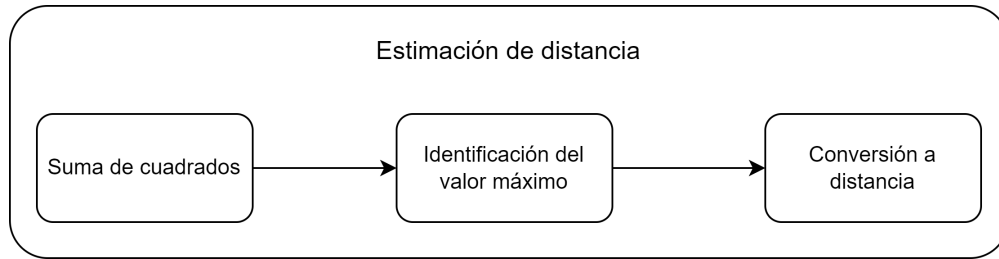


Figura 3.21: Método propuesto para la estimación de distancia.

Refinamiento del concepto planteado

La estructura descrita previamente para ambas etapas conforma el núcleo de procesamiento básico para la instrumentación del módulo de compresión de pulso. Sin embargo, surge una problemática muy importante con respecto a la cantidad de muestras a procesar. Realizar la compresión de pulso para un intervalo completo de repetición de pulso en un solo núcleo de procesamiento implica contar con transformadas rápidas de Fourier y búferes de datos de tamaño significativo. Tomando en cuenta un PRI de 4 ms y una frecuencia de muestreo de 100 MHz, un solo núcleo de procesamiento tendría que computar cuatrocientos mil paquetes de datos, cada uno conformado por dos muestras correspondientes a las componentes real e imaginaria. Esto resulta poco práctico considerando el tiempo que tardarían las transformadas rápidas de Fourier en procesar una ventana de datos de semejante longitud.

En este trabajo, se propone una solución que utiliza dos métodos principales para aliviar la problemática anterior. El primer método consiste en la decimación de los datos suministrados al módulo de compresión de pulso, lo que permite reducir la cantidad de información a procesar. Sin embargo, un incremento en el factor de decimación conlleva una pérdida proporcional en la resolución de distancia, como puede demostrarse fácilmente al aplicarlo en la expresión (3.37). Dado que el primer método resulta insuficiente para aliviar la problemática planteada, se propone particionar los datos decimados en lotes de menor tamaño. Este méto-

do permite aplicar la estrategia de dividir y conquistar, mediante la cual un problema de complejidad elevada se descompone en problemas de menor escala y, en consecuencia, más sencillos de abordar. De este modo, los lotes de datos de menor tamaño pueden procesarse en menos tiempo y con un menor consumo de recursos. La división del conjunto original de paquetes de datos equivale a fraccionar el alcance máximo del sistema en intervalos de distancia más pequeños, razón por la cual dichos intervalos se denominan compuertas de alcance.

La formación de compuertas permite reducir la carga de trabajo asignada a un solo núcleo de procesamiento, a partir de lo cual surgen tres alternativas: contar con un núcleo de procesamiento dedicado para cada compuerta, disponer de múltiples núcleos que se turnen para procesar todas las compuertas o emplear un solo núcleo capaz de procesar el vector de muestras correspondiente a la compuerta de alcance actual antes de la llegada del siguiente vector, es decir, un núcleo capaz de operar en tiempo real. Las dos primeras opciones ofrecen la posibilidad de paralelizar el procesamiento, mientras que la tercera reduce de forma significativa la utilización de recursos aunque requiere una latencia considerablemente menor para cumplir con el criterio de tiempo real. No obstante, contar con un núcleo de procesamiento dedicado para cada compuerta implica que, una vez concluido el procesamiento asignado, los núcleos permanecerán inactivos durante largos periodos, lo que representa un desperdicio significativo de recursos. Bajo estas circunstancias, resulta más conveniente plantear una arquitectura con un número limitado de núcleos de procesamiento que se alternen para procesar el intervalo completo de muestras. En este sentido, se busca, como objetivo de diseño ideal, que un solo núcleo de procesamiento sea capaz de barrer el intervalo completo.

Por consiguiente, de forma conceptual, el sistema se compone de dos bloques principales: el primero engloba a las compuertas de alcance, mientras que el segundo corresponde al núcleo de procesamiento. Las compuertas de alcance permiten dividir el conjunto original de paquetes de datos en lotes de menor tamaño, lo que hace que su procesamiento sea más manejable. La incorporación de compuertas de alcance requiere una etapa de control dedicada responsable de la formación de lotes y de la gestión de dichas compuertas. Por su parte, el núcleo consta de dos etapas principales: un filtro en concordancia implementado en el dominio de la frecuencia y un estimador de distancia. Este núcleo procesará cada compuerta en tiempo real, lo que permitirá reducir de manera significativa el consumo de recursos.

3.3.3. Diseño a nivel sistema

Con base en el concepto planteado, se define la arquitectura general del sistema partiendo de las consideraciones de diseño para las compuertas de alcance y la implementación del filtro en concordancia. A partir de esta arquitectura, se estiman los recursos requeridos y se lleva a cabo el desglose en subsistemas y elementos.

Consideraciones de diseño para las compuertas de alcance

Las compuertas de alcance, junto con el núcleo de procesamiento, constituyen las unidades fundamentales del módulo de compresión de pulso. Para cada compuerta, el núcleo debe ser capaz de estimar la distancia al objeto a partir del intervalo de muestras que recibe como entrada. No obstante, la partición del PRI en intervalos de muestras de menor tamaño implica la existencia de ciertas compuertas en donde no aparece ningún pulso recibido que corresponda a las señales reflejadas por algún objeto en el espacio. En un sistema real, esta problemática se soluciona mediante la incorporación de algoritmos de umbralización como CFAR. Estos algoritmos permiten tomar en cuenta el nivel de ruido actual dentro del sistema, a partir de lo cual se ajusta dinámicamente un umbral de detección. Adicionalmente, cada compuerta se analiza en conjunto con las compuertas circundantes, a partir de lo cual se obtiene una probabilidad de detección con el fin de reducir falsas alarmas. Sin embargo, debido a la complejidad que implica el desarrollo de dichos algoritmos, su implementación en la arquitectura propuesta queda fuera del alcance de este trabajo.

Otro reto importante de este enfoque consiste en determinar el tamaño adecuado para las compuertas de alcance. Esta decisión es crucial para el sistema completo, ya que define el número total de compuertas necesarias para cubrir el PRI estipulado e impone el criterio de latencia necesario para cumplir con la restricción de tiempo real. En primer lugar, se propone utilizar un factor de decimación de muestras igual a cuatro con el fin de mantener la resolución de distancia en seis metros. Por lo tanto, la frecuencia de muestreo se reduce a 25 MHz. Considerando dicha decimación, se cuenta con una agrupación de cien mil paquetes de datos que deben ser fraccionados en agrupaciones de menor tamaño. A medida que disminuya la cantidad de paquetes de datos que conforman a cada compuerta de alcance, se requiere aumentar proporcionalmente el número total de compuertas con el fin de seguir cubriendo la cantidad total de paquetes por procesar. Aunado a esto, reducir el tamaño de las compuertas permite disminuir los recursos requeridos por cada una y el tiempo de procesamiento asociado. Sin embargo, un tamaño reducido implica una restricción de tiempo real más estricta. Por consiguiente, es necesario asentar primero la implementación del filtro en

concordancia previo a la determinación del tamaño de las compuertas de alcance.

Implementación del filtro en concordancia

El filtro en concordancia constituye la técnica central mediante la cual el núcleo de procesamiento lleva a cabo su función; por consiguiente, su implementación tiene un impacto significativo sobre el resto de la arquitectura. Dentro de este esquema, la FFT y la IFFT son los bloques que demandan más recursos y presentan mayor complejidad de desarrollo, afectando de manera más directa la solución global. Por lo tanto, es necesario determinar su implementación previo al desglose y diseño de las etapas del sistema.

En aras de agilizar el proceso de diseño, se recurre a la implementación de la FFT y la IFFT mediante el bloque IP LogiCORE proporcionado por Xilinx. Este bloque ofrece una solución flexible y eficiente para llevar a cabo la técnica de procesamiento requerida, permitiendo configurar múltiples parámetros entre los que se encuentran el tamaño del vector de muestras suministrado y la arquitectura de procesamiento empleada.

El bloque IP en cuestión es capaz de obtener la transformada correspondiente a partir de un vector cuya longitud oscila entre 8 y 65536 muestras. Dicho bloque ofrece la posibilidad de contar con una transformada de longitud variable; sin embargo, se opta por mantenerla fija en aras de simplificar el diseño. Por otra parte, este bloque ofrece cuatro arquitecturas de procesamiento: Pipelined, Radix-4, Radix-2 y Radix-2 Lite, según lo estipulado en [62]. Estas arquitecturas pueden ser divididas en dos categorías principales: procesamiento continuo (Pipelined) y procesamiento en ráfaga (Radix-4, Radix-2 y Radix-2 Lite). La primera categoría corresponde a una arquitectura basada en múltiples etapas de procesamiento Radix-2 encauzadas, que permiten una carga y descarga continua de datos. Este esquema reduce la latencia a costa de una mayor utilización de recursos. En contraste, las arquitecturas de procesamiento en ráfaga requieren ciclos dedicados para la carga y descarga de datos, además de aquellos necesarios para la obtención de la transformada correspondiente. Sin embargo, emplean una cantidad de recursos mucho menor en comparación con la arquitectura de procesamiento continuo. A su vez, las arquitecturas de procesamiento en ráfaga se clasifican según el método empleado en su unidad básica de procesamiento (Radix-4 o Radix-2), lo que influye tanto en el rendimiento como en los recursos requeridos.

Con base en las configuraciones disponibles para implementar los bloques más demandantes del filtro en concordancia, es posible determinar el tamaño de las compuertas de alcance y seleccionar la arquitectura adecuada a partir de un análisis costo-beneficio que considere tanto la latencia como la utilización de recursos, el cual se desarrolla a continuación.

Determinación del tamaño de las compuertas de alcance y selección de arquitectura

El análisis costo-beneficio orientado a determinar el tamaño de las compuertas de alcance y seleccionar una arquitectura para las transformadas gira en torno a dos métricas esenciales: la latencia de cada arquitectura y su utilización de recursos correspondiente. La latencia se entiende como el tiempo requerido por cada arquitectura para computar la transformada a partir de un vector de muestras de tamaño específico. La utilización de recursos se mide en función de la cantidad de bloques de memoria y procesamiento dedicados que requiere su implementación. En los FPGA de Xilinx, los bloques de memoria embebida requeridos por el bloque IP de la transformada se implementan a través de los denominados Block RAM (BRAM, por sus siglas en inglés), los cuales permiten alocar memoria en bloques de 18 y 36 Kb. Por otra parte, los bloques de procesamiento embebido se implementan mediante los denominados DSP Slices, los cuales permiten llevar a cabo operaciones matemáticas a través de circuitos altamente optimizados. Es necesario alcanzar un balance entre ambas métricas de manera que se satisfaga la meta de diseño establecida respecto al criterio de tiempo real. Por el momento, se asume que el resto de la instrumentación tendrá un impacto mucho menor en los rubros antes mencionados por lo que, en principio, el análisis se centra principalmente en la implementación de las transformadas.

Como primera aproximación, se pueden analizar las métricas establecidas para la selección de arquitectura considerando la longitud máxima que ofrece el bloque IP. Para ello, se usa la estimación de latencia y utilización de recursos que ofrece la herramienta de configuración para el bloque IP correspondiente. La estimación contempla una frecuencia de operación de 100 MHz, un formato de punto fijo, una longitud de palabra de 16 bits y un esquema de escalamiento automático. Así mismo, se considera que las multiplicaciones complejas realizadas dentro del bloque IP se implementan mediante cuatro multiplicadores y que la aritmética de la mariposa dentro de la transformada se desarrolla a partir de LUT.

Arquitecturas	DSP Slices	Block RAM	Latencia [us]
Pipelined	40	389	1967.580
Radix-4	12	164	2622.870
Radix-2	4	132	6555.810
Radix-2 Lite	2	144	11797.42

Tabla 3.1: Estimación de latencia y recursos requeridos para 65,536 muestras.

En la tabla 3.1, se muestra la latencia y los recursos requeridos para cada arquitectura en el caso de una transformada con longitud de 65536 muestras. A partir de la información

recopilada, se determina que la arquitectura más apropiada para implementar el filtro en concordancia es Radix-4. Esta decisión radica en la disminución significativa en la latencia que ofrece dicha arquitectura respecto a Radix-2 y Radix-2 Lite empleando menos de la mitad de los recursos requeridos por la arquitectura Pipelined.

Una vez que se ha seleccionado una arquitectura, se puede proseguir con la determinación de las compuertas de alcance. Para ello, es necesario considerar ciertas particularidades que presenta el bloque IP al implementar la arquitectura elegida. En principio, Radix-4 requiere que el vector de muestras suministrado tenga un tamaño equivalente a una potencia de cuatro. No obstante, el bloque IP permite utilizar tamaños que no cumplan con esta condición mediante una arquitectura híbrida en la cual la última etapa puede sustituirse por una etapa única de Radix-2. Por otro lado, el bloque IP acepta únicamente tamaños equivalentes a potencias de dos y requiere contar con un tamaño mínimo de 64 muestras para poder emplear Radix-4.

De igual forma, es necesario considerar que se requieren dos transformadas para implementar el filtro en concordancia. Si bien el bloque IP elegido permite realizar ambas operaciones en un mismo núcleo, este método resulta poco eficiente y provocaría un cuello de botella importante. Por consiguiente, se propone considerar una arquitectura en donde se cuenta con dos bloques IP en secuencia, lo que permitiría procesar dos vectores de muestras simultáneamente. Por el momento, se omite el multiplicador complejo ya que su impacto es mucho menor al de las transformadas. Se asume que la identificación del valor máximo es directamente proporcional al tamaño del vector de muestras, requiriendo de un barrido completo realizado en $N \cdot t_s$ segundos, en donde N es el tamaño del vector de muestras y t_s es el periodo de muestreo. Por lo tanto, la latencia total aproximada del sistema se encontrará dada por la expresión (3.39).

$$Latencia_T = 2 \cdot Latencia_{FFT} + N \cdot t_s \quad (3.39)$$

El caso de prueba utilizado para la determinación de las compuertas de alcance consiste en un núcleo de procesamiento compuesto por dos transformadas en secuencia y una máquina encargada de encontrar el valor máximo dentro del vector de muestras. Dicho núcleo procesa las muestras almacenadas en los búferes de datos asociados a las compuertas de alcance, las cuales requieren cierto tiempo para llenarse en función de su tamaño y el periodo de muestreo de 40 ns considerando la decimación aplicada. En la tabla 3.2, se puede apreciar la estimación de latencia aproximada correspondiente al núcleo de procesamiento para distintos tamaños de compuerta de alcance. La estimación divide al núcleo en tres etapas que corresponden

a la FFT, la IFFT y la identificación del valor máximo, respectivamente. Por otro lado, la tabla 3.3 muestra el tiempo aproximado que tardarían en llenarse dos búferes de datos, correspondientes a dos compuertas de alcance, a partir de los cuales se realizará la compresión de pulso.

Número de muestras	Etapas 1 [us]	Etapas 2 [us]	Etapas 3 [us]
64	2.39	4.78	5.42
128	4.63	9.26	10.54
256	8.47	16.94	19.50
512	17.59	35.18	40.30
1024	34.23	68.46	78.70
2048	72.79	145.58	166.06
4096	144.47	288.94	329.90
8192	308.47	616.94	698.86
16384	615.67	1231.34	1395.18
32768	1312.15	2624.30	2951.98
65536	2622.87	5245.74	5901.10

Tabla 3.2: Latencias aproximadas del núcleo de procesamiento para diferentes tamaños de compuerta.

Número de muestras	Búfer 1 [us]	Búfer 2 [us]
64	2.56	5.12
128	5.12	10.24
256	10.24	20.48
512	20.48	40.96
1024	40.96	81.92
2048	81.92	163.84
4096	163.84	327.68
8192	327.68	655.36
16,384	655.36	1310.72
32,768	1310.72	2621.44
65,536	2621.44	5242.88

Tabla 3.3: Tiempo de llenado para dos búferes de datos considerando distintos tamaños de compuerta.

Se busca que el núcleo de procesamiento sea capaz llevar a cabo la primera transformada antes de que el siguiente búfer de datos se llene. Por lo tanto, es necesario determinar los tamaños para los cuales se cumple esta condición. Con este fin, se asume que la primera transformada empieza a llevarse a cabo en el instante en que el primer búfer de datos se llena, sumando el tiempo de llenado del primer búfer y la latencia de la primera etapa de la

compuerta de alcance. El periodo de tiempo obtenido debe ser menor al tiempo de llenado del segundo búfer de datos para que se cumpla la restricción de tiempo real.

Número de muestras	Búfer 1 + Etapa 1 [us]	Búfer 2 [us]	Diferencia [us]
64	4.95	5.12	-0.17
128	9.75	10.24	-0.49
256	18.71	20.48	-1.77
512	38.07	40.96	-2.89
1024	75.19	81.92	-6.73
2048	154.71	163.84	-9.13
4096	308.31	327.68	-19.37
8192	636.15	655.36	-19.21
16384	1271.03	1310.72	-39.69
32768	2622.87	2621.44	1.43
65536	5244.31	5242.88	1.43

Tabla 3.4: Determinación de los tamaños para los que se cumple el tiempo real.

Como se puede apreciar en la tabla 3.4, los últimos dos tamaños no cumplen con la restricción de tiempo real al existir una diferencia positiva entre el periodo requerido para realizar la primera transformada y el tiempo de llenado del segundo búfer. En consecuencia, no se toman en cuenta al realizar la estimación de los recursos requeridos para la implementación de la arquitectura particular del bloque IP para diferentes tamaños de compuerta.

Número de muestras	DSP Slices	Block RAM (18K)
64	12	7
128	12	7
256	12	7
512	12	7
1024	12	7
2048	12	7
4096	12	11
8192	12	22
16384	12	44

Tabla 3.5: Recursos requeridos para la implementación de Radix-4 para diferentes tamaños de compuerta.

En la tabla 3.5, se puede apreciar que las transformadas con tamaño entre 64 y 2048 muestras usan la misma cantidad de recursos, por lo tanto, se propone considerar únicamente el mayor tamaño de dicho intervalo con el fin de aprovechar los recursos de la mejor forma. En consecuencia, la elección del tamaño del vector de muestras reduce a 2048, 4096, 8192 y

16384.

Dado que el análisis realizado es de primera aproximación, no se sabe a ciencia cierta la cantidad de recursos que el diseño final requerirá. Por consiguiente, se propone asentar el tamaño de las compuertas de alcance a 2048 muestras, lo que corresponde a intervalos de distancia de 12288 metros, en aras de escoger una opción conservadora en cuanto a recursos, lo que permitirá contar con un núcleo de procesamiento más compacto.

Determinación del número de compuertas de alcance

A partir del tamaño del vector de muestras elegido, se determina que se requieren 49 compuertas de alcance para poder realizar el barrido del intervalo completo de muestras. Dicha cantidad de compuertas contempla que los intervalos de muestreo de dos compuertas contiguas sean mutuamente excluyentes. Esto puede resultar problemático en caso de que las muestras del pulso recibido se encuentren distribuidas entre ambos intervalos, ocasionando ambigüedad al obtenerse dos estimaciones de distancia degradadas para el mismo objeto. Por lo tanto, se requiere que los vectores de muestras correspondientes a compuertas de alcance contiguas tengan un traslape equivalente a la longitud del pulso radiado con el fin de evitar que existan estimaciones ambiguas. Dicho pulso tiene una longitud de $12.5 \mu s$, lo que representa un intervalo de 313 muestras al tomar en consideración la frecuencia a la que se realiza el muestreo de datos para la compresión de pulso. Considerando lo anterior, el tiempo de llenado del primer búfer de datos se mantiene igual, sin embargo, el segundo búfer de datos empieza a llenarse cuando el primer búfer se encuentra en la muestra número 1735. En consecuencia, es necesario aumentar el número de compuertas a 58 para cubrir el alcance teórico máximo. Sin embargo, el intervalo de muestreo de la última compuerta de alcance termina en 4.03884 ms, lo que resulta mayor al PRI. Esto representa un problema ya que, de detectarse un objeto en la primera compuerta del siguiente PRI, aparecerían dos estimaciones de distancia completamente diferentes que corresponden al mismo objeto. Por lo tanto, se elige no tomarla en cuenta, lo que reduce el alcance máximo del sistema a 595.248 kilómetros.

Verificación del cumplimiento de la restricción de tiempo real

El traslape de muestras en las compuertas de alcance implica contar con un periodo de tiempo menor para llevar a cabo la primera transformada, por lo que es necesario reevaluar la arquitectura preliminar con el fin de verificar si se sigue cumpliendo la restricción de tiempo real. Para ello, se realiza nuevamente la estimación del tiempo de llenado para dos búferes de datos correspondientes a dos compuertas consecutivas considerando el traslape antes

mencionado. Esto da como resultado que el segundo búfer termina de llenarse en $151.32 \mu s$, antes de que la primera transformada termine de procesar el búfer de datos anterior. Por consiguiente, será necesario ajustar el diseño para que el núcleo de procesamiento sea capaz de operar en tiempo real. En este sentido, surgen tres alternativas para subsanar el déficit observado: modificar el tamaño de las compuertas de alcance, cambiar la arquitectura seleccionada para las transformadas o aumentar la frecuencia de operación del núcleo. Las dos primeras opciones permiten reducir la latencia de las transformadas a costa de una mayor utilización de recursos, mientras que la tercera mantiene, en principio, la misma demanda de recursos. Se opta por esta última alternativa, proponiendo incrementar la frecuencia de operación del módulo a 110 MHz , lo que permite reducir el tiempo requerido en llevar a cabo la primera transformada en $148.09 \mu s$, tomando en cuenta el llenado del primer búfer. De esta forma, se cuenta con un margen de $3.227 \mu s$ para seguir cumpliendo con la restricción de tiempo real. No obstante, es importante considerar que el incremento en la frecuencia de operación implica que los tiempos de propagación presentes en los circuitos digitales tendrán un impacto más significativo sobre la instrumentación del módulo.

Diseño de la unidad de procesamiento

Con base en las especificaciones del sistema, se propone la unidad de procesamiento mostrada en la figura 3.22. A grandes rasgos, esta unidad se encuentra conformada por dos subsistemas principales: el subsistema de compuertas de alcance del haz (SCAH) y el núcleo de procesamiento. En principio, este diseño contempla únicamente la instrumentación de un solo haz con el fin de generar un producto mínimo viable a partir del cual se pueda expandir el diseño en un futuro.

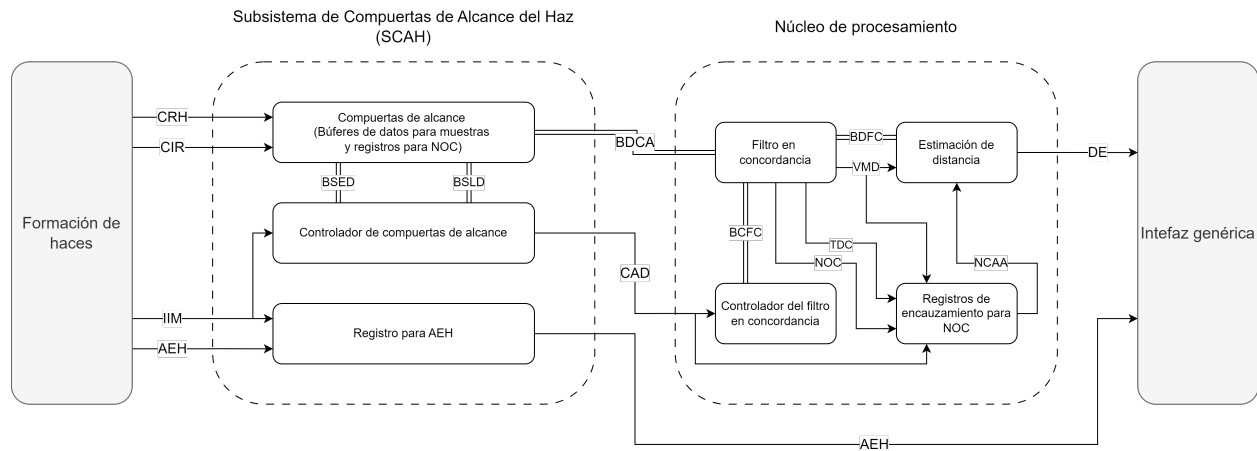


Figura 3.22: Diseño a nivel sistema propuesto.

En el diseño a nivel sistema propuesto, el SCAH se encarga de la formación y gestión de las compuertas de alcance, suministrando las señales necesarias para que el núcleo de procesamiento ejecute la compresión de pulso. El SCAH incorpora cuatro puertos de entrada conectados a la salida del formador de haces. Los tres primeros permiten la recepción de las señales de datos correspondientes al ángulo de elevación del haz (AEH) y a las componentes real e imaginarias de dicho haz (CRH y CIH, respectivamente), mientras que el cuarto puerto recibe la señal de control que indica el inicio del intervalo de muestras (IIM).

La señal AEH contiene el ángulo de elevación del haz al que pertenecen las muestras, requiriendo una longitud de siete bits para representar un intervalo de valores entre 0 y 90. Por su parte, las señales CRH y CIH corresponden a las muestras de las componentes del haz y utilizan un formato numérico signado de punto fijo con una longitud de palabra de 16 bits, conforme al modelo teórico del formador de haces planteado en la sección anterior. Finalmente, la señal de control IIM posee una longitud de un bit y tiene la función de indicar el comienzo de un conjunto completo de muestras que abarca un PRI entero.

El SCAH está conformado por tres elementos principales: un registro destinado al almacenamiento del valor de la señal AEH, un conjunto de compuertas de alcance y un controlador asociado. El registro encargado de la señal AEH tiene una longitud de palabra de siete bits y actualiza su contenido de manera síncrona al recibirse la señal IIM. Cada compuerta de alcance, a su vez, se compone de dos entidades: la primera corresponde a dos búferes de datos de 2048 localidades con una longitud de palabra de 16 bits, y la segunda consiste en un registro con una longitud de palabra de seis bits.

Las especificaciones de los búferes de datos se establecen considerando que almacenan las muestras de las señales CRH y CIH, cuyas características ya han sido mencionadas anteriormente. En cuanto al registro, este almacena el número ordinal de la compuerta (NOC), el cual permite ubicarla dentro del intervalo de distancia que le corresponde. Dado que el sistema contempla un total de 57 compuertas, cada registro asociado al NOC requiere una longitud de palabra de seis bits. En consecuencia, para almacenar el intervalo completo de muestras que abarcan todas las compuertas, se requieren 114 búferes de datos, equivalentes a un tamaño total de 3.735 Mb, al igual que 57 registros, con un tamaño acumulado de 342 bits.

El controlador asociado al SCAH tiene la función de gestionar las compuertas de alcance, lo cual comprende tres tareas esenciales: realizar la escritura de datos en los búferes asignados a cada compuerta, indicar al núcleo de procesamiento cuando una compuerta se encuentra lista para ser procesada y suministrar los datos de la compuerta disponible a dicho núcleo.

Durante la fase inicial de operación del SCAH, el controlador permanece en estado de

espera hasta la recepción de la señal IIM, la cual indica la disponibilidad de datos válidos a la salida del formador de haces. Una vez recibida dicha señal, el controlador inicia el llenado de los búferes de datos correspondientes a la primer compuerta de alcance a partir de las proporcionadas por el formador de haces. En la arquitectura propuesta, se asume que la asignación del NOC se define en tiempo de compilación, por lo que el controlador no modifica el contenido del registro correspondiente durante la ejecución.

La operación de escritura de datos se realiza a una frecuencia de 25 MHz. Una vez que los búferes de una compuerta se encuentran completamente llenos, el controlador genera la señal Compuerta de Alcance Disponible (CAD) para notificar al núcleo de procesamiento que dicha compuerta está lista para ser procesada. A continuación, el controlador prosigue con el llenado de los búferes correspondientes a la siguiente compuerta, mientras de forma simultánea transmite los datos de la compuerta disponible al núcleo y el valor de la señal AEH a la interfaz genérica.

Para llevar a cabo esta operación, el controlador requiere un multiplexor que le permita seleccionar la compuerta de alcance desde la cual se enviarán los datos. El intercambio de información se efectúa a través del bus de datos de la compuerta de alcance (BDCA), compuesto por tres líneas: la primera línea está destinada a transmitir el valor del NOC, mientras que las dos restantes se utilizan para enviar las muestras de las componentes real e imaginaria de la compuerta (CRCA y CICA, respectivamente), obtenidas mediante la lectura de los búferes correspondientes. Considerando la frecuencia de operación establecida para el núcleo de procesamiento, la lectura de los búferes se realiza a una frecuencia de 110 MHz.

Dado que las operaciones de lectura y escritura deben ejecutarse de manera independiente, el diseño del SCAH contempla el uso de dos buses independientes que contienen las señales necesarias para tales operaciones, denominados Bus de Señales para la Lectura de Datos (BSLD) y Bus de Señales para la Escritura de Datos (BSED), respectivamente.

Por consiguiente, el SCAH requiere de tres puertos de salida: el primero destinado a la señal AEH con longitud de siete bits, el segundo correspondiente al bus de datos BDCA, compuesto por una línea de seis bits y dos líneas adicionales de 16 bits, y el tercero para la señal CAD, cuya longitud es de un bit.

El núcleo de procesamiento incorpora dos puertos de entrada que permiten recibir las señales contenidas dentro del bus BDCA, así como la señal CAD, y se encuentra conformado por tres elementos principales: un filtro en concordancia, un controlador asociado y un estimador de distancia. Así mismo, se incluye un elemento auxiliar conformado por una serie de registros destinados a almacenar los valores de la señal NOC. Este elemento auxiliar requiere tres registros con una longitud de palabra de seis bits, dispuestos secuencialmente a modo

de encauzamiento.

Durante la fase inicial de operación, el controlador asociado al filtro en concordancia configura los bloques IP de dicha entidad que así lo requieran mediante el Bus de Datos del Filtro en Concordancia (BCFC) y permanece a la espera de la señal CAD. Una vez recibida dicha señal, el controlador utiliza el BCFC para habilitar el filtro y el búfer de datos almacena el valor de la señal NOC en su primera localidad. A partir de su habilitación, el filtro inicia el procesamiento de las señales CRCA y CICA, indicando cuando la primer transformada termine de procesar el vector de muestras suministrado a partir de la señal Transformada Directa Concluida (TDC). Esta señal se emplea para trasladar el valor almacenado en el primer registro de encauzamiento de la señal NOC hacia el segundo.

El vector de muestras suministrado continúa atravesando las etapas subsecuentes del procesamiento dentro del filtro, mientras este permanece a la espera de los datos correspondientes a la siguiente compuerta. Una vez concluido el procesamiento completo de la primera compuerta de alcance, el filtro genera la señal Vector de Muestras Disponible (VMD), la cual indica al estimador de distancia que las muestras del pulso comprimido están listas. De manera simultánea, las muestras resultantes se transfieren junto con sus índices a través del Bus de Datos del Filtro en Concordancia (BDFC).

El BDFC está compuesto por tres líneas: las dos primeras corresponden a las componentes real e imaginaria resultantes (CRR y CIR, respectivamente), mientras que la tercera transmite el índice de las componentes resultantes (ICR). Tentativamente, las señales CRR y CIR conservan la misma longitud de palabra que sus equivalentes en el BDCA; sin embargo, podría ser necesario incrementar dicho parámetro dependiendo de la implementación final del filtro. Dado que cada compuerta de alcance contiene 2048 muestras, la señal ICR requiere una longitud de palabra de 11 bits.

La señal VMD también se utiliza para trasladar el valor de la señal NOC de la primera compuerta hacia el tercer y último registro de encauzamiento. El valor contenido en dicho registro corresponde al número de compuertas de alcance acumuladas (NCAA) y permite ubicar la distancia estimada dentro del intervalo de distancia asociado a la compuerta de alcance correspondiente.

Tras la recepción de la señal VMD, el estimador de distancia inicia el barrido del vector de muestras con el propósito de identificar el índice asociado al valor máximo. Durante dicho barrido, se multiplica el valor del índice del máximo actual (IMA) por un factor que convierte el número de muestra representado por el IMA en distancia. Para las condiciones específicas del sistema, este factor es igual a seis, lo que implica que cada muestra equivale a seis metros dentro del intervalo de distancia cubierto por la compuerta de alcance, y requiere de tres

bits para su representación. Posteriormente, el resultado se suma con el producto de la señal NCAA y un factor equivalente a 10410, valor que representa la distancia acumulada correspondiente a las compuertas de alcance anteriores, tomando en cuenta el traslape existe entre compuertas contiguas y cuya representación requiere 14 bits. La expresión (3.40) describe el cálculo realizado para la estimación de distancia.

$$DE = IMA \cdot \alpha_{DPM} + NCAA \cdot \alpha_{NCAA} \quad (3.40)$$

Donde DE representa la distancia estimada, α_{DPM} el factor de conversión de distancia por muestra y α_{NCAA} el factor de conversión asociado al número de compuertas de alcance acumuladas. La señal DE posee una longitud de 20 bits, valor suficiente para representar el alcance máximo del sistema (595.248 km). Por consiguiente, el núcleo de procesamiento requiere un solo puerto de salida que corresponda a dicha señal y que posea la misma longitud.

Finalmente, las señales DE y AEH se entregan a una interfaz genérica, concluyendo así el funcionamiento global del sistema propuesto. Dado que ambos subsistemas presentan una complejidad considerable, se propone limitar la instrumentación al núcleo de procesamiento. En consecuencia, es necesario realizar el diseño a detalle de cada uno de sus elementos a partir de las especificaciones expuestas en esta sección.

3.3.4. Diseño a detalle

A partir del desglose en subsistemas y elementos desarrollado en la sección anterior, es necesario realizar el diseño a detalle de cada uno de los elementos que conforman al núcleo de procesamiento. En este sentido, se requiere desarrollar la instrumentación destinada a implementar el filtro en concordancia, su controlador asociado y el estimador de distancia. Posteriormente, dichos elementos se integran entre sí, incorporando registros de encauzamiento para el NOC de cada compuerta que está siendo procesada de forma simultánea dentro del núcleo.

No obstante, antes de abordar el diseño de estos elementos, resulta conveniente mencionar ciertas consideraciones previas, en particular aquellas relacionadas con el filtro en concordancia. En este elemento, tanto la FFT como la IFFT se instrumentan mediante el bloque IP LogiCORE Fast Fourier Transform, por lo que resulta necesario explicar brevemente su funcionamiento antes de proceder con el diseño del filtro. A su vez, este bloque IP emplea la interfaz de datos AXI4-Stream, por lo que se considera pertinente explicar primero los aspectos fundamentales de dicha interfaz, incluyendo su estructura y el protocolo básico requerido para realizar una transacción de datos.

Interfaz de datos AXI4-Stream

La interfaz de datos AXI4-Stream permite realizar la transferencia unidireccional de información entre un transmisor y un receptor a través de un bus compuesto por cuatro líneas esenciales: `aclk`, `tvalid`, `tready` y `data`. La señal `aclk` corresponde a la señal de reloj que establece la pauta de la transferencia de información, mientras que las señales `tvalid` y `tready` permiten controlar el flujo de datos entre transmisor y receptor. La señal `tvalid` indica que el transmisor cuenta con datos listos para ser enviados, mientras que `tready` señala que el receptor está preparado para recibirlos. La transferencia ocurre únicamente cuando ambas señales se encuentran en un nivel lógico alto de forma simultánea. Por último, la línea `tdata` contiene los datos a transferir, los cuales se transmiten de forma paralela. En caso de requerir el envío de una secuencia de datos, se transmite una muestra de dicha secuencia en cada ciclo de reloj. El protocolo básico descrito para la transferencia de datos se ilustra en la figura 3.23.

De manera adicional, el bus puede incorporar las líneas `tlast` y `tuser`. La primera permite al transmisor indicar cuando se haya transmitido la última muestra de una secuencia de datos, mientras que la segunda contiene información adicional asociada a los datos enviados, ya sea de cada dato o de la secuencia completa.

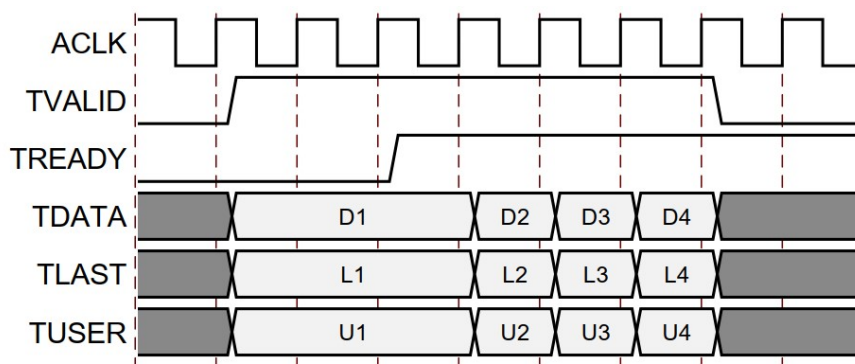


Figura 3.23: Intercambio de datos mediante AXI4-Stream [62].

Generalidades del funcionamiento del bloque IP Fast Fourier Transform

Una vez que se ha descrito brevemente la interfaz de datos AXI4-Stream, se puede proceder a explicar las generalidades del funcionamiento del bloque IP Fast Fourier Transform. Los puertos de entrada de este bloque incluyen dos interfaces AXI4-Stream mediante las cuales se suministra la configuración de operación del bloque (`s_axis_config`), al igual que los datos a procesar (`s_axis_data`). Los tres puertos de entrada restantes corresponden a las señales de reloj, reinicio y habilitación, respectivamente.

En primer lugar, la interfaz correspondiente al puerto de configuración del bloque permite modificar ciertas características de la transformada que incluyen su tamaño, su dirección (directa o inversa), la inserción de un prefijo cíclico y la cantidad de escalamiento para cada etapa. En el caso específico del filtro en concordancia a desarrollar, únicamente se requiere configurar la dirección de cada bloque ya que se cuenta con un solo canal de tamaño fijo, sin inserción de prefijo cíclico y que hace uso de un esquema de escalamiento automático. Bajo estas condiciones, la línea `s_axis_config_tdata` tiene una longitud total de ocho bits, a través de la cual se indica la dirección de la transformada mediante el bit menos significativo, rellenando el resto de bits con ceros lógicos. Dada la naturaleza del puerto correspondiente, la interfaz no incorpora las líneas `tlast` y `tuser`.

Por otro lado, la interfaz correspondiente al puerto de suministro de datos requiere que las muestras de la señal a procesar sean complejas y se encuentren en su representación rectangular. En consecuencia, se suministran las muestras de las componentes real e imaginaria de la señal a procesar mediante la línea `s_axis_data_tdata`, situando la parte real en los bits menos significativos, seguida de la parte imaginaria. Dicha interfaz cuenta con una línea `tlast` para indicar que se ha transmitido la última muestra del vector de datos a procesar; sin embargo, no tiene injerencia en el funcionamiento propio de la transformada. Cabe recalcar que esta interfaz no cuenta con una línea `tuser`.

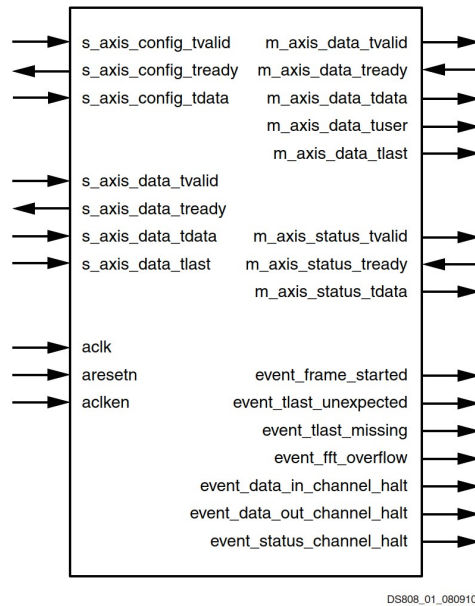


Figura 3.24: Esquemático provisto para el bloque IP [62].

Por otra parte, los puertos de salida del bloque incluyen dos interfaces AXI4-Stream mediante las cuales se entregan los datos procesados (`m_axis_data`), al igual que se monito-

rean ciertos parámetros del estado interno de la transformada (`m_axis_status`). Los puertos restantes corresponden a las líneas empleadas para el monitoreo de eventos específicos del funcionamiento propio del bloque.

En la arquitectura propuesta, únicamente se utiliza la interfaz de datos de salida, por lo que se ahondará exclusivamente en ella. El puerto de salida de datos cuenta con todas las señales básicas del protocolo AXI4-Stream mostradas en la figura 3.23. Dicho puerto entrega las componentes resultantes de la transformada aplicada en el mismo formato y orden que el puerto de entrada equivalente mediante la línea `m_axis_data.tdata`. Así mismo, el puerto utiliza la línea `m_axis_data.tuser` para indicar la cantidad de escalamiento aplicado a la salida de datos con el fin de conservar la misma longitud de palabra que en la entrada, tomando en cuenta el esquema de escalamiento automático seleccionado, y, si así se requiere, el índice de cada par de componentes dentro del vector de muestras entregado a la salida. Si se incorpora dicho índice, este se ubica en los bits menos significativos de la línea correspondiente, mientras que la cantidad de escalamiento se coloca en los bits más significativos.

Diseño del filtro en concordancia

Una vez se conoce la estructura de los bloques IP mediante los cuales se implementarán las transformadas, se puede proceder con el diseño del filtro en concordancia. Este elemento representa al corazón del núcleo de procesamiento, ya que es la entidad que permite llevar a cabo la compresión de pulso. Su implementación se delimitó a grandes rasgos durante el diseño a nivel sistema, a partir de lo cual se propone la arquitectura mostrada en la figura 3.25.

El diseño final del filtro en concordancia es síncrono, incorporando los puertos de entrada y salida descritos en la tabla 3.6. La arquitectura del filtro contempla dos bloques IP encargados de realizar la FFT y la IFFT. Así mismo, el diseño incluye una serie de registros de encauzamiento en la salida de la FFT, un bloque IP dedicado a realizar la multiplicación compleja, y un módulo de memoria que almacena los coeficientes espectrales de referencia. La lectura de estos coeficientes desde la memoria se gestiona mediante un contador de 11 bits, el cual genera las direcciones de memoria necesarias para dicho proceso.

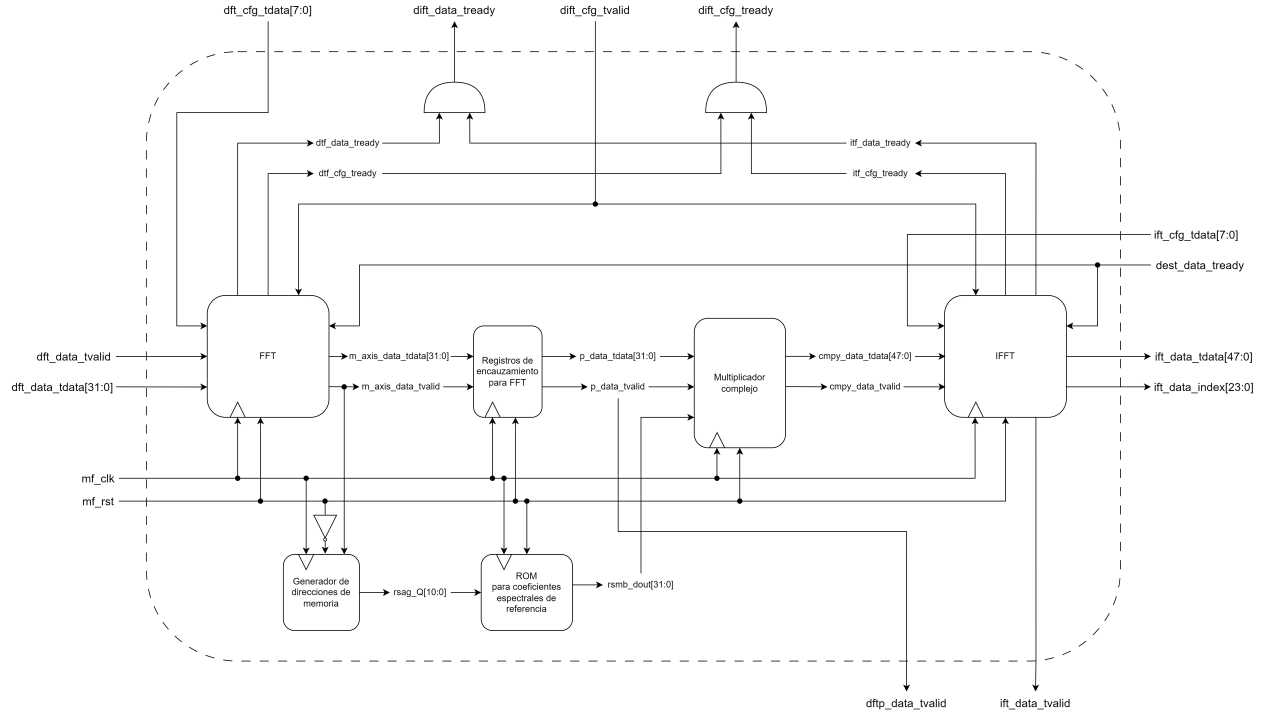


Figura 3.25: Arquitectura del filtro en concordancia.

Puerto	Dirección	Longitud	Descripción
dft_cfg_tready	Salida	1 bit	Puerto que indica si ambos bloques IP están listos para recibir sus parámetros de configuración
dft_cfg_tvalid	Entrada	1 bit	Puerto que notifica si el controlador asociado está listo para enviar parámetros de configuración
dft_cfg_tdata	Entrada	8 bits	Puerto que recibe los parámetros de configuración del bloque asociado a la FFT
ift_cfg_tdata	Entrada	1 bit	Puerto que recibe los parámetros de configuración del bloque asociado a la IFFT
dft_data_tready	Salida	1 bit	Puerto que indica si ambos bloques IP están listos para recibir datos de entrada

Puerto	Dirección	Longitud	Descripción
dft_data_tvalid	Entrada	1 bit	Puerto que notifica si existen datos de entrada disponibles para el bloque asociado a la FFT
dft_data_tdata	Entrada	32 bits	Puerto que recibe los datos de entrada para el bloque asociado a la FFT
dest_data_tready	Entrada	1 bit	Puerto que notifica si el siguiente elemento está listo para recibir datos
dftp_data_tvalid	Salida	1 bit	Puerto que indica si existen datos disponibles a la salida de los registros de encauzamiento asociados a la FFT
ift_data_tvalid	Salida	1 bit	Puerto que indica si existen datos disponibles a la salida del bloque asociado a la FFT
ift_data_tdata	Salida	48 bits	Puerto que entrega los datos de salida para el bloque asociado a la IFFT
ift_data_index	Salida	24 bits	Puerto que entrega la cantidad de escalamiento aplicado y el índice de los datos de salida para el bloque asociado a la IFFT
mf_clk	Entrada	1 bit	Puerto que recibe la señal de reloj
mf_rst	Entrada	1 bit	Puerto que recibe la señal de reinicio

Tabla 3.6: Puertos de entrada y salida del filtro en concordancia

Este elemento requiere que los bloques IP de las transformadas sean configurados por el controlador asociado antes de iniciar el procesamiento de muestras. Para ello, ambos bloques deben indicar al controlador que están listos para recibir los parámetros de configuración a través de sus líneas `s_axis_config_tready`, denominadas `dft_cfg_tready` e `ift_cfg_tready` en el esquema mostrado en la figura 3.25. Estas líneas se conectan una compuerta lógica AND, cuya salida corresponde al puerto `dift_cfg_tready`.

Cuando este puerto presenta un nivel lógico alto, el controlador responde activando el puerto `dift_cfg_valid`, el cual se conecta directamente a la línea `s_axis_config_tvalid` de cada bloque. De esta manera, se indica que los parámetros de configuración están listos para ser transferidos. Dichos parámetros se envían de forma simultánea a través de los puertos

dft_cfg_tdata e ift_cfg_tdata, los cuales se conectan a la líneas s_axis_config_tdata de cada transformada correspondiente.

Una vez configurados ambos bloques, el controlador debe verificar que estén listos para recibir datos, lo cual se indica mediante un nivel lógico alto en el puerto dft_data_tready. Este puerto se conecta a la línea s_axis_data_tready de cada bloque. A partir de este evento, el controlador notifica a los bloques que los elementos subsecuentes en la cadena de procesamiento también están listos para recibir datos mediante la activación del puerto dest_data_tready, el cual se conecta a las líneas m_axis_data_tready correspondientes.

Finalmente, el elemento queda a la espera de que la fuente de datos active el puerto dft_data_tvalid, conectado a la línea s_axis_data_tvalid del bloque destinado a realizar la FFT, lo cual ocurre cuando una compuerta de alcance se ha llenado por completo y cuenta con datos disponibles para su procesamiento. La recepción de un nivel lógico alto en dicho puerto viene acompañada por la recepción del vector de muestras complejas, correspondiente a la compuerta de alcance disponible, a través del puerto dft_data_tdata. Este puerto se conecta directamente a la línea s_axis_data_tdata del bloque encargado de realizar la FFT, la cual tiene una longitud de 32 bits. En esta línea, los 16 bits más significativos contienen la componente real de cada muestra, mientras que los 16 bits menos significativos corresponden a la componente imaginaria.

El bloque responsable de ejecutar la FFT procesa las muestras complejas de entrada e indica la finalización de dicho procesamiento mediante su línea m_axis_data_tvalid, momento en el cual las muestras complejas del espectro de la señal están disponibles en la línea m_axis_data_tdata. Estas muestras mantienen un formato de 32 bits, conservando la misma distribución de componentes real e imaginaria que s_axis_data_tdata.

A partir de la aparición de un nivel lógico alto en la línea m_axis_data_tvalid del bloque encargado a realizar la FFT, se habilita el generador de direcciones de memoria, el cual se instrumenta mediante un contador de 11 bits implementado a través del bloque IP Binary Counter. Este contador genera las direcciones necesarias para leer los coeficientes espectrales del pulso de referencia almacenados en una memoria de solo lectura (ROM, por sus siglas en inglés), a través de la línea rsag_Q. Dicha memoria cuenta con 2048 localidades de 32 bits y se implementa utilizando el bloque IP Block Memory Generator, seleccionándose una ROM de un solo puerto sin controlador BRAM dedicado y aplicando un algoritmo de minimización de área. Cada localidad almacena una muestra con el mismo formato utilizado por m_axis_data_tdata. Los datos de la localidad direccionada se entregan mediante la línea rsmb_dout. La memoria se inicializa con un archivo de coeficientes espectrales de referencia,

generado mediante un script de MATLAB que modela un pulso real de amplitud unitaria y duración de $12.5 \mu\text{s}$.

Inicialmente se consideró conectar directamente las muestras de los coeficientes espectrales obtenidas a la salida de la FFT al multiplicador complejo. No obstante, dado que el bloque de memoria introduce una latencia de dos ciclos de reloj, se incorporaron registros de encauzamiento en la salida de la FFT para las señales `m_axis_data_tvalid` y `m_axis_data_tdata`. Este componente consta de dos registros en cascada para cada señal, lo cual compensa la latencia y garantiza la llegada simultánea al multiplicador complejo de las muestras del espectro de la señal de entrada y los coeficientes de referencia correspondientes.

A continuación, se incorpora el multiplicador complejo encargado de aplicar el filtro en el dominio de la frecuencia. Este elemento se implementa mediante el bloque IP LogiCORE Complex Multiplier, configurado para emplear cuatro multiplicadores implementados basados en LUT, con el propósito de maximizar el rendimiento del procesamiento.

El multiplicador cuenta con dos interfaces AXI4-Stream reducidas a la entrada, las cuales cuentan únicamente con las líneas `s_axis_tdata` y `s_axis_tvalid`. En la primer interfaz, se conectan las salidas del registro de encauzamiento de la FFT mediante las líneas `p_data_tdata` y `p_data_tvalid`. En la segunda interfaz se conecta la salida de datos de la memoria (`rsmb_out`) a la línea de datos correspondiente. Dado que la memoria carece de una señal `tvalid` propia, se utiliza la línea `p_data_tvalid` para ambas interfaces, garantizando así la sincronización temporal de los operandos. Esta misma línea también se conecta al puerto de salida `dftp_data_tvalid`, el cual indica el inicio efectivo de la multiplicación compleja una vez finalizado el procesamiento de la FFT.

Con el fin de mantener la sincronización de datos dentro del multiplicador complejo, se utiliza la configuración de latencia automática sugerida por el bloque, la cual incorpora un encauzamiento interno de cinco etapas. A la salida del multiplicador, se dispone de una sola interfaz AXI4-Stream reducida, con las mismas líneas que las de entrada, pero con una línea de datos expandida a 48 bits, utilizando 24 bits por componente. Esta longitud de palabra implica un truncamiento del resultado de la multiplicación compleja, ya que se requerirían 33 bits por componente para conservar la precisión completa. La decisión de truncar busca equilibrar la pérdida de precisión con los recursos adicionales que demandaría el bloque IP de la IFFT al aumentar la longitud de la señal resultante. Finalmente, las salidas se le entregan a dicho bloque mediante las líneas `cmpy_data_tdata` y `cmpy_data_tvalid`.

Por último, el bloque IP correspondiente a la IFFT recibe el producto espectral resultante y se encarga de obtener la transformada inversa del mismo. El inicio de su procesamiento

está determinado por la línea `cmpy_data_tvalid` proveniente del multiplicador complejo. Al activarse esta señal en el puerto correspondiente, el bloque comienza a procesar el vector de muestras proporcionado a través de su línea de datos. A la salida, el bloque entrega a los puertos `ift_data_tvalid`, `ift_data_tdata` e `ift_data_tuser`, las señales `m_axis_data_tvalid`, `m_axis_data_tdata`, `m_axis_data_tuser` del mismo. Estas señales indican la disponibilidad de datos procesados por la IFFT, el conjunto de muestras resultantes y la cantidad de escalamiento aplicada para conservar la misma longitud de palabra junto con el índice de cada muestra, respectivamente.

Este subsistema se implementa mediante bloques IP con el fin de facilitar la interconexión entre elementos. Esto se aprecia al aprovechar las señales `tvalid` para sincronizar el flujo de datos dentro del filtro, garantizando una comunicación consistente entre etapas. Así mismo, esta metodología disminuye considerablemente los tiempos de desarrollo, pero queda restringida el uso a los bloques IP incluidos en la licencia disponible.

Diseño del controlador asociado al filtro en concordancia

El filtro en concordancia requiere de un controlador que permita gestionar la configuración de los bloques IP encargados de realizar la FFT y la IFFT, al igual que regular el suministro de datos proveniente de los búferes de datos de la compuerta de alcance en turno. Con esta finalidad, el controlador cuenta con 11 puertos de entrada y salida, tal y como se aprecia en la figura 3.26, de los cuales la mayoría se abarcan en la tabla 3.6 a excepción de `mfc_clk`, `mfc_rst`, `rgsb_full` y `rgsb_last`. Los primeros dos puertos permiten proporcionar las señales de reloj y reinicio al controlador, respectivamente. Por su parte, los dos puertos restantes indican cuando una compuerta de alcance está lista para ser procesada (`rgsb_full`) y cuando el último dato correspondiente a dicha compuerta ha sido enviado (`rgsb_last`).

Como se puede apreciar en la figura 3.26, el controlador se instrumenta mediante una máquina de estados tipo Moore, ya que sus salidas dependen únicamente del estado actual de la máquina. Esto permite tener una mejor estabilidad y mantiene al filtro en concordancia bajo un régimen de control síncrono en todo momento.

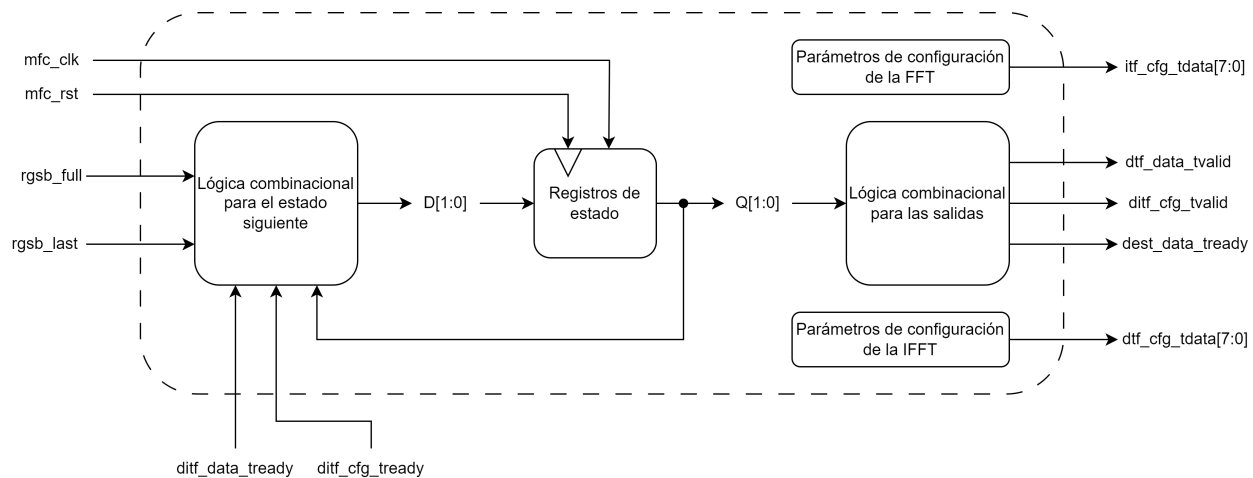


Figura 3.26: Arquitectura del controlador asociado al filtro en concordancia.

A partir de lo descrito durante el diseño del filtro en concordancia, se determina que el controlador requiere de cuatro estados para gestionar su funcionamiento, los cuales se describen a continuación:

Estado 1: Inicio El primer estado corresponde al estado inicial del filtro, al cual se entra tras un reinicio del sistema. Todas las salidas de la máquina de estados se mantienen en un nivel lógico bajo y el controlador espera a que el filtro notifique que está listo para recibir los parámetros de configuración. La transición al siguiente estado ocurre exclusivamente cuando se recibe un nivel lógico alto en el puerto `dift_cfg_tready`.

Estado 2: Configuración El segundo estado corresponde al estado de configuración, en el cual el controlador le indica al filtro que está listo para enviar los parámetros de configuración a través de las líneas `dtf_cfg_tdata` e `itf_cfg_tdata` mediante un nivel lógico alto en el puerto `dift_cfg_tvalid`. Aunque dichas líneas siempre cuenten con los parámetros de configuración requeridos, solo son considerados por el filtro cuando `dift_cfg_tvalid` tenga un nivel lógico alto. El primer bloque IP dentro del filtro se configura para llevar a cabo la transformada directa mediante el valor hexadecimal x“01” en su línea de datos, mientras que el segundo bloque se configura como transformada inversa asignando el valor hexadecimal x“00”. La transición al siguiente estado ocurre en el siguiente ciclo de reloj, independientemente de las entradas.

Cabe destacar que, en los dos primeros estados, los puertos `rgsb_full` y `rgsb_last` no pueden contar con un nivel lógico alto ya que el SCAH aún no está en operación. Por lo tanto, en

la tabla de verdad del estado siguiente, las combinaciones donde alguno de estos puertos presente un nivel lógico alto pueden tratarse como *don't-care*, lo que permite optimizar la lógica combinacional.

Estado 3: Espera El tercer estado corresponde al estado de espera, en el cual se notifica al filtro que los elementos subsecuentes están listos para recibir datos, mediante el puerto `dest_data_tready`. El controlador permanece en este estado mientras no se cuente con un nivel lógico alto en el puerto `rgsb_full`, lo que indica que se ha llenado el búfer de datos de una compuerta de alcance y se encuentra disponible para su procesamiento o se cuente con un nivel lógico bajo en el puerto `dif_data_tready`. La transición al siguiente estado ocurre únicamente cuando se recibe un nivel lógico alto en ambos puertos.

Estado 4: Carga El cuarto y último estado corresponde al estado de carga, en el cual el SCAH notifica al núcleo de procesamiento que se ha llenado una compuerta de alcance y comienza a transferir las muestras almacenadas dentro de su búfer correspondiente. Es en este estado donde el filtro inicia efectivamente el procesamiento de las muestras al enviar un nivel lógico alto a través del puerto `dtf_data_tvalid`. Lo anterior indica al bloque IP encargado de realizar la FFT dentro del filtro que la compuerta de alcance está lista para enviar las muestras a través de su línea de datos correspondiente.

El controlador permanece en dicho estado mientras no reciba un nivel lógico alto a través del puerto `rgsb_last`, lo que indica que ha ocurrido la transmisión de la última muestra del búfer. Así mismo, el controlador requiere que se mantenga un nivel lógico alto en el puerto `dif_data_tready`. Si se recibe un nivel lógico alto en `rgsb_last`, el controlador regresa al estado de espera hasta que se notifique la disponibilidad de una nueva compuerta de alcance. De igual forma, si `dif_data_tready` pasa a nivel bajo de manera inesperada, el controlador retornará al estado de espera.

Al igual que en los estados iniciales, es posible tratar como *don't-care* las combinaciones donde `rgsb_full` y `rgsb_last` cuentan con un nivel lógico alto de manera simultánea. Esta consideración se fundamenta en que la velocidad de lectura del búfer de la compuerta de alcance actual supera a la velocidad de escritura en el búfer de la siguiente compuerta, lo que imposibilita que ambos puertos cuenten con un nivel lógico alto al mismo tiempo.

En la figura 3.27 se muestra el diagrama de estados que caracteriza la máquina de estados descrita anteriormente. En la tabla 3.7, se muestran la tabla de verdad que caracteriza la lógica combinacional de las salidas con base en el estado actual de la máquina. Se opta

por implementar los registros de estado mediante *flip-flops* D al ser el tipo de *flip-flop* comúnmente utilizado por los FPGA, además que simplifica el diseño de la máquina de estados. Por último, en la tabla 3.8, se muestran las funciones lógicas reducidas para la lógica combinacional del estado siguiente.

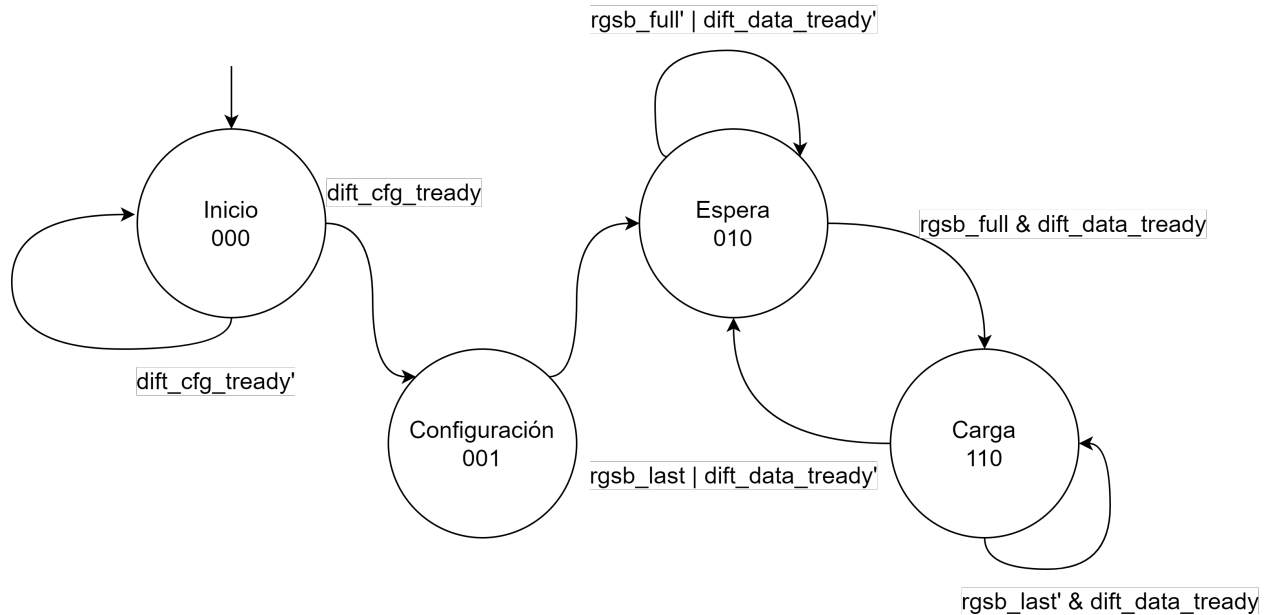


Figura 3.27: Diagrama de estados del controlador asociado al filtro en concordancia.

Estado actual		Salidas		
Q1	Q0	dft_data_tvalid	dest_data_tready	dift_cfg_tvalid
0	0	0	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	1	0

Tabla 3.7: Tabla de verdad para las salidas del controlador asociado al filtro en concordancia.

Señal	Función lógica reducida
D0	$Q1' Q0' \text{dift_cfg_tready} + \text{rgsb_full} \text{dift_data_tready} + Q1 Q0 \text{rgsb_last}' \text{dift_data_tready}$
D1	$Q0 + Q1$

Tabla 3.8: Funciones lógicas reducidas para el estado siguiente.

Diseño del estimador de distancia

El último elemento principal del núcleo de procesamiento que resta por diseñar es el estimador de distancia. La arquitectura propuesta para dicho elemento, mostrada en la figura 3.28, consta de tres etapas de procesamiento secuencial: la suma de cuadrados, la identificación del valor máximo y el cálculo de distancia. El filtro en concordancia suministra al estimador de distancia las muestras procesadas, junto con su índice y la señal *tvalid* que indica su disponibilidad, a partir de las cuales se obtendrá la estimación. Adicionalmente, se proporciona el NCAA de la compuerta de alcance cuyas muestras están siendo procesadas, permitiendo asociar el resultado al intervalo de distancia correcto. Esta arquitectura contempla ocho puertos de entrada y salida, cuyas características se detallan en la tabla 3.9, necesarios para cumplir con la funcionalidad asignada al estimador de distancia.

Cada etapa del estimador incorpora un registro de encauzamiento a la salida de su circuito combinacional correspondiente. Esta estructura garantiza un tiempo suficiente para la propagación de las señales a través de la lógica y mejora el rendimiento al permitir el procesamiento simultáneo de tres muestras. Las dos primeras etapas de encauzamiento, ubicadas tras el sumador de cuadrados y el comparador mediante el cual se identifica el valor máximo respectivamente, almacenan tanto el resultado de la suma de cuadrados como el índice de la muestra asociada. Sin embargo, la segunda etapa actualiza estos valores solo cuando la suma de cuadrados calculada en la primera etapa supera el valor máximo almacenado previamente, lo que se determina a través del comparador. Como resultado, la segunda etapa entrega exclusivamente el índice correspondiente al máximo actual a la unidad de cálculo de distancia, cuyo resultado es almacenado en la tercera y última etapa de encauzamiento.

En cada etapa se incluyen registros adicionales para almacenar el estado de la señal *tvalid* asociada. Esto permite verificar la disponibilidad de datos válidos en cada fase, habilitar condicionalmente los registros de datos e índices, y determinar cuándo finaliza el barrido completo del vector de muestras.

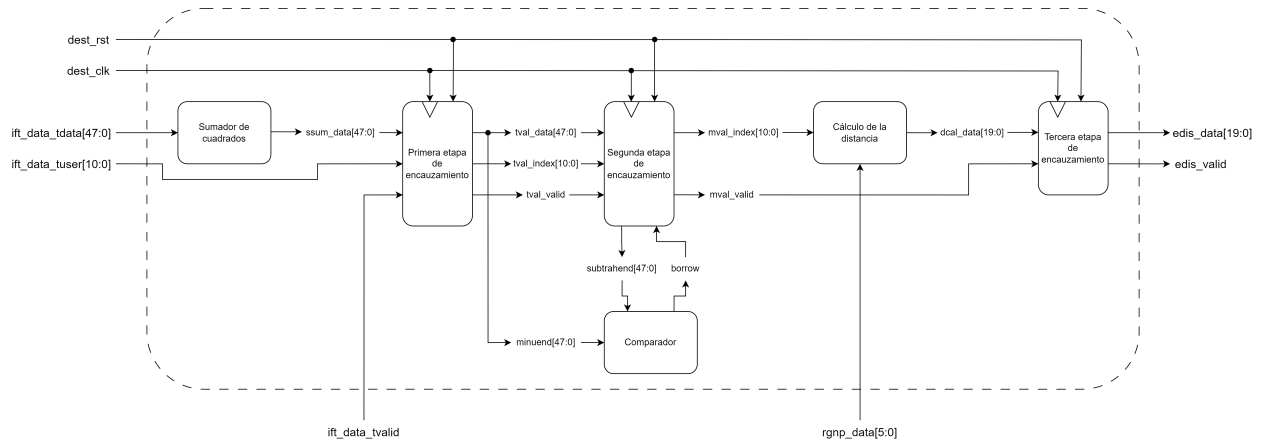


Figura 3.28: Arquitectura del estimador de distancia.

Puerto	Dirección	Longitud	Descripción
dest_rst	Entrada	1 bit	Puerto que recibe la señal de reloj
dest_clk	Entrada	1 bit	Puerto que reciba la señal de reinicio
ift_data_tdata	Entrada	48 bits	Puerto que recibe las muestras procesadas por el filtro en concordancia
ift_data_tuser	Entrada	24 bits	Puerto que recibe la cantidad de escalamiento aplicado por la IFFT y el índice de las muestras entregadas por el filtro en concordancia
ift_data_tvalid	Entrada	1 bit	Puerto que indica si existen datos disponibles a la salida del filtro en concordancia
rgnp_data	Entrada	6 bits	Puerto que recibe el NCAA de la compuerta de alcance que acaba de ser procesada completamente por el filtro en concordancia
edis_valid	Salida	1 bit	Puerto que indica si el estimador de distancia está listo para enviar datos a la interfaz genérica
edis_data	Salida	20 bits	Puerto que entrega los resultados de la estimación de distancia a la interfaz genérica

Tabla 3.9: Puertos de entrada y salida del estimador de distancia.

Como se ha mencionado previamente, la primera etapa del procesamiento tiene como objetivo calcular la suma de los cuadrados de las componentes real e imaginaria correspondientes a cada muestra compleja proporcionada por el filtro en concordancia. Para llevar a

cabo dicha operación, se requieren dos multiplicadores destinados a elevar al cuadrado cada componente, así como un sumador encargado de realizar la adición de los productos obtenidos. Estas operaciones pueden ser instrumentadas mediante diseños estructurales dedicados, contruidos a partir de sumadores elementales y cierta lógica adicional, cuyo rendimiento dependerá del algoritmo empleado y de las optimizaciones aplicadas.

No obstante, este enfoque no constituye la única alternativa disponible para implementar las operaciones requeridas. En este sentido, también es posible utilizar una descripción funcional, en la cual el sintetizador infiere automáticamente los circuitos necesarios a partir de los operadores empleados. Esta alternativa delega la implementación al sintetizador, quien determina si las operaciones se instrumentan mediante bloques lógicos de propósito general o a través de bloques de procesamiento embebidos, como los DSP Slices. Para este caso en particular, el sintetizador optaría por utilizar dichos bloques de procesamiento dedicado al contar con circuitos especializados para realizar las operaciones matemáticas requeridas. Por consiguiente, un diseño estructural resulta poco práctico al requerir de muchas optimizaciones para competir con el desempeño que ya ofrecen los DSP Slices.

Además de las dos alternativas mencionadas, existía la posibilidad de implementar las operaciones mediante bloques IP. No obstante, esta opción presentaba un rendimiento y una utilización de recursos muy similares a los obtenidos con la descripción funcional, al mismo tiempo que reducía la portabilidad del diseño sin aportar beneficios significativos que justifiquen su uso. Esta situación contrasta con el caso del filtro en concordancia, donde el uso de bloques IP para las transformadas tenía ventajas significativas. Por tales motivos, se descartó el uso de bloques IP para esta etapa específica.

La suma de cuadrados se instrumenta mediante un circuito compuesto por dos multiplicadores de 24 bits con formato signado y un sumador de 48 bits con formato no signado. El primer multiplicador recibe en ambas entradas los 24 bits más significativos del puerto `ift_data_tdata`, correspondientes a la componente real, mientras que el segundo multiplicador utiliza los 24 bits menos significativos, correspondientes a la componente imaginaria. Dado que se eleva al cuadrado, las salidas de ambos multiplicadores generan valores no signados de 48 bits, los cuales se conectan directamente a las entradas del sumador. Así mismo, considerando que el valor máximo de cada producto requiere 47 bits para su representación, la suma de ambos se mantiene dentro de un rango de 48 bits. Por lo tanto, se especifica dicha longitud para la salida del sumador, ya que no existe posibilidad de desbordamiento.

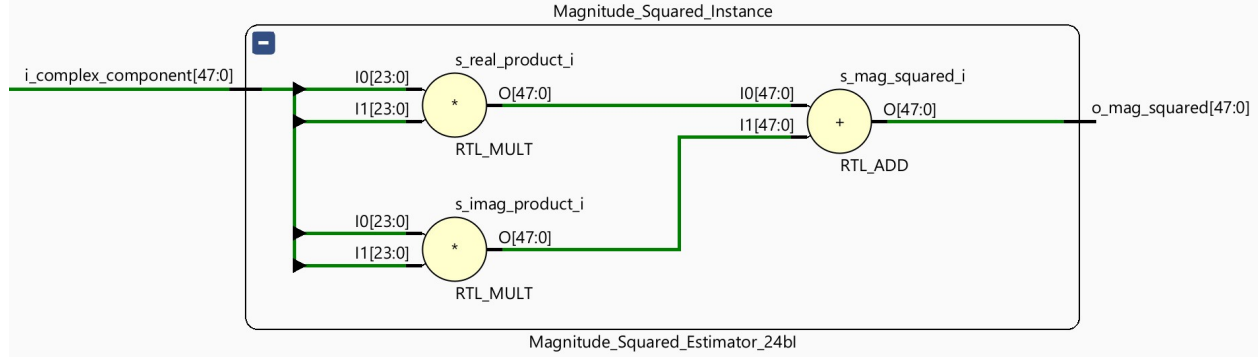


Figura 3.29: Diseño elaborado para la suma de cuadrados.

La salida del sumador de cuadrados se conecta a la entrada de la primera etapa de encauzamiento, específicamente al registro de datos correspondiente. Como se mencionó anteriormente, a esta etapa también se le suministra el índice de la muestra procesada por el sumador de cuadrados, junto con la señal `tvalid` asociada a dicha muestra. Los registros de datos y del índice se habilitan únicamente si se ha recibido un nivel lógico alto a través del puerto `ift_data_tvalid`.

La segunda etapa de procesamiento corresponde a la identificación del valor máximo, la cual se implementa mediante un comparador que evalúa si el valor de la muestra actual, proporcionada por la primera etapa de encauzamiento, es superior al valor máximo almacenado en la segunda etapa de encauzamiento. El registro encargado de almacenar el valor máximo en dicha etapa se inicializa en cero tras un reinicio del sistema o al concluir el barrido completo del vector de muestras.

La comparación se instrumenta mediante un circuito restador modificado, cuyo acarreo de salida (`borrow`) determina si el valor de la muestra actual es igual o mayor que el valor máximo almacenado. Cuando este acarreo está activo, se habilita la actualización de los registros de datos y del índice de la segunda etapa de encauzamiento, sustituyendo el valor máximo anterior por la nueva muestra y su índice correspondiente. Este restador se implementa tomando como base el circuito requerido para sumar el valor actual y el complemento a dos del valor máximo almacenado. En este sentido, se utiliza como punto de partida el diseño de un sumador de 48 bits compuesto por sumadores completos elementales, como el mostrado en la figura 3.30, introduciendo modificaciones específicas para adaptar su funcionalidad a los requisitos de la aplicación.

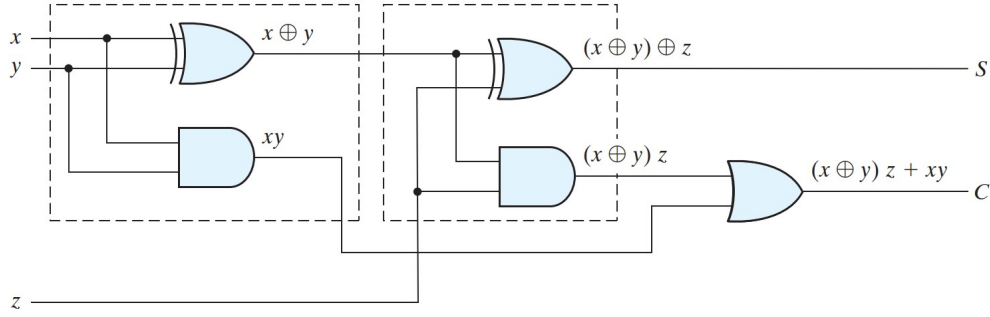


Figura 3.30: Arquitectura del sumador completo elemental original [63].

Originalmente, estos sumadores completos llevan a cabo la adición entre dos bits de entrada (x , y), considerando un acarreo de entrada (z) proveniente de una etapa anterior. En la implementación actual, se elimina la compuerta XOR necesaria para generar el bit de resultado (S), se reemplaza la compuerta XOR restante por XNOR y se invierte uno de los bits de entrada. Tal y como se puede apreciar en la figura 3.31, se encadenan dos sumadores modificados para formar la unidad básica de comparación.

La decisión de encadenar dos sumadores se fundamenta en la implementación de lógica mediante LUT propia de los FPGA. En particular, los FPGA de Xilinx emplean LUT con capacidad de hasta seis entradas [64], lo que permite describir dos sumadores completos modificados en cascada utilizando una única LUT. Esta optimización reduce a la mitad el número de etapas requeridas para implementar el circuito restador, lo que permite un mejor aprovechamiento de los recursos lógicos del sistema.

Por otro lado, a pesar de que el diseño contempla un número considerable de unidades de comparación en cascada, la latencia resultante no representa un inconveniente si el retardo de propagación en cada LUT es lo suficientemente pequeño. Con base en este requerimiento, al igual que los recursos lógicos y de procesamiento embebido requeridos hasta ahora, se propone emplear la tarjeta de desarrollo Nexys Video para la implementación del sistema. Esta tarjeta incorpora un FPGA Artix-7 XC7A200T-1SBG484C, el cual ofrece 33650 celdas lógicas, 740 DSP Slices y 13 Mb de BRAM, además de permitir una frecuencia de reloj de hasta 450 MHz [65]. Para dicho FPGA, el retardo de propagación a partir de que la LUT recibe sus entradas es, como máximo, 0.13 ns si la lógica es puramente combinacional y no se requiere de un multiplexor adicional [66]. En consecuencia, se estima que el acarreo requeriría aproximadamente 3.12 ns para propagarse a través de todas las unidades de comparación, lo que se encuentra dentro de un margen aceptable incluso al considerar retardos adicionales en la propagación de la señal.

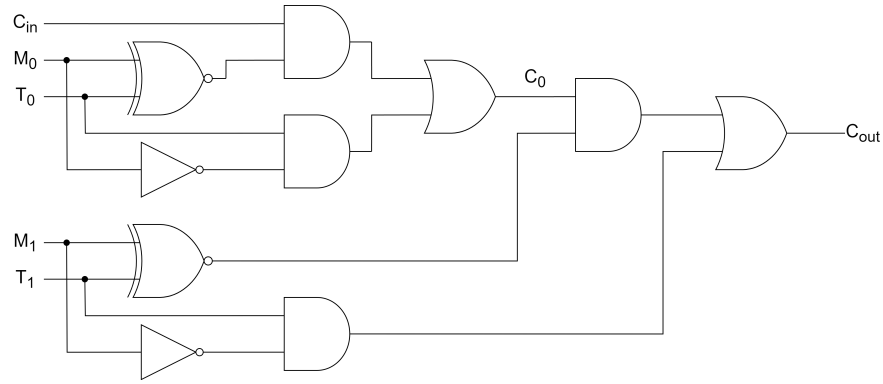


Figura 3.31: Arquitectura de la unidad de comparación.

En la figura 3.32 se muestra el diseño del circuito restador modificado que implementa la comparación. Cada unidad de comparación (UC) procesa dos bits del valor actual (VA) junto con dos bits del valor máximo almacenado (VM). Las unidades de comparación se conectan en cascada, de modo que cada una recibe como entrada de acarreo la salida de la etapa anterior, conformando así la estructura completa del comparador. Para generar el complemento a dos del valor máximo almacenado, basta con proporcionar un nivel lógico alto en el acarreo de entrada de la primera etapa, ya que la inversión de bits requerida para dicha operación está incorporada en el diseño interno de cada unidad de comparación.

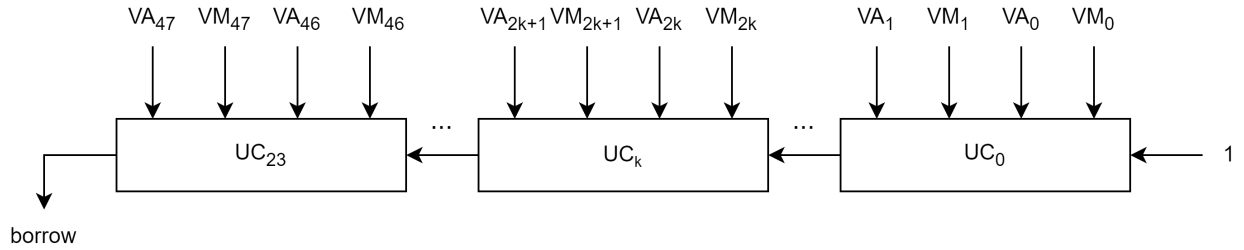


Figura 3.32: Arquitectura del circuito restador modificado.

Cuando el acarreo de salida de la última unidad de comparación (borrow) adquiere un nivel lógico alto, se habilita la actualización de los registros correspondientes en la segunda etapa de encauzamiento. En tal caso, el valor y el índice del máximo anterior se sustituyen por el valor y el índice de la muestra actual, siempre que la señal tvalid proveniente de la primera etapa de encauzamiento también se encuentre en nivel lógico alto. Esta condición garantiza que la actualización solo ocurra cuando existan datos disponibles provenientes de la primer etapa del procesamiento.

La unidad de cálculo de distancia se implementa mediante descripción funcional, fun-

damentada en las mismas consideraciones expuestas para el sumador de cuadrados. Este módulo requiere de dos sumadores y dos multiplicadores, de acuerdo con la expresión (3.40). El primer multiplicador se encarga de calcular el producto entre el índice del valor máximo, almacenado en la segunda etapa de encauzamiento y cuya longitud es de 11 bits, y el factor de conversión DPM, cuyo valor fijo es seis y requiere tres bits para su representación. El resultado de esta operación requiere una longitud de 14 bits para ser representado. La segunda multiplicación se realiza entre el valor del puerto rgnp_data, cuya longitud es de seis bits, y el factor de conversión NCAA, cuyo valor fijo es 10410 y requiere de 14 bits para su representación, produciendo una salida de 20 bits. Considerando el alcance máximo definido para el sistema y las longitudes de los productos intermedios, la salida final del sumador puede mantener una longitud de 20 bits.

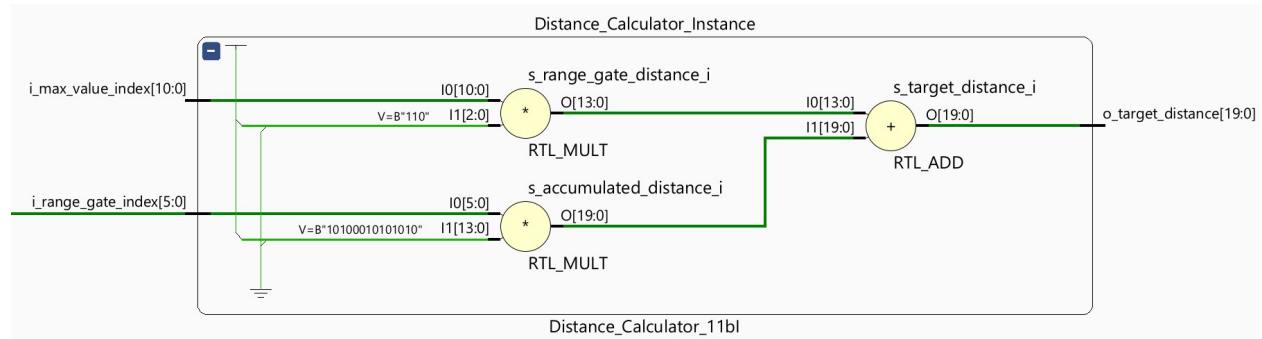


Figura 3.33: Diseño elaborado por el sintetizador para la unidad de cálculo de distancia.

La salida de la unidad de cálculo corresponde a la estimación de distancia actual y se almacena en el registro de datos de la tercera etapa de encauzamiento. Este registro se habilita únicamente cuando se cuenta con un nivel lógico alto para la señal tvalid proveniente de la segunda etapa de encauzamiento. La salida del registro de datos de la tercera etapa de encauzamiento se conecta al puerto edis_data. De igual forma, se dispone de un registro destinado a la señal tvalid, cuya salida se conecta al puerto edis_valid. Cabe destacar que las señales tvalid de cada etapa de encauzamiento se propagan en cada ciclo de reloj, manteniendo habilitados sus registros de forma continua.

Integración de elementos y arquitectura final del sistema

Una vez que se han diseñado los tres elementos principales del sistema, se procede a su integración conforme a lo mostrado en la figura 3.34. Durante esta etapa, se incorporaron los registros de encauzamiento encargados de almacenar los valores del NOC. Dichos registros tienen una longitud de 6 bits y se organizan en una secuencia de tres etapas que permiten

conservar el valor del NOC correspondiente a las compuertas de alcance que están siendo procesadas por el núcleo. Cada registro cuenta con una señal de habilitación que controla su actualización únicamente al alcanzarse hitos específicos dentro del procesamiento.

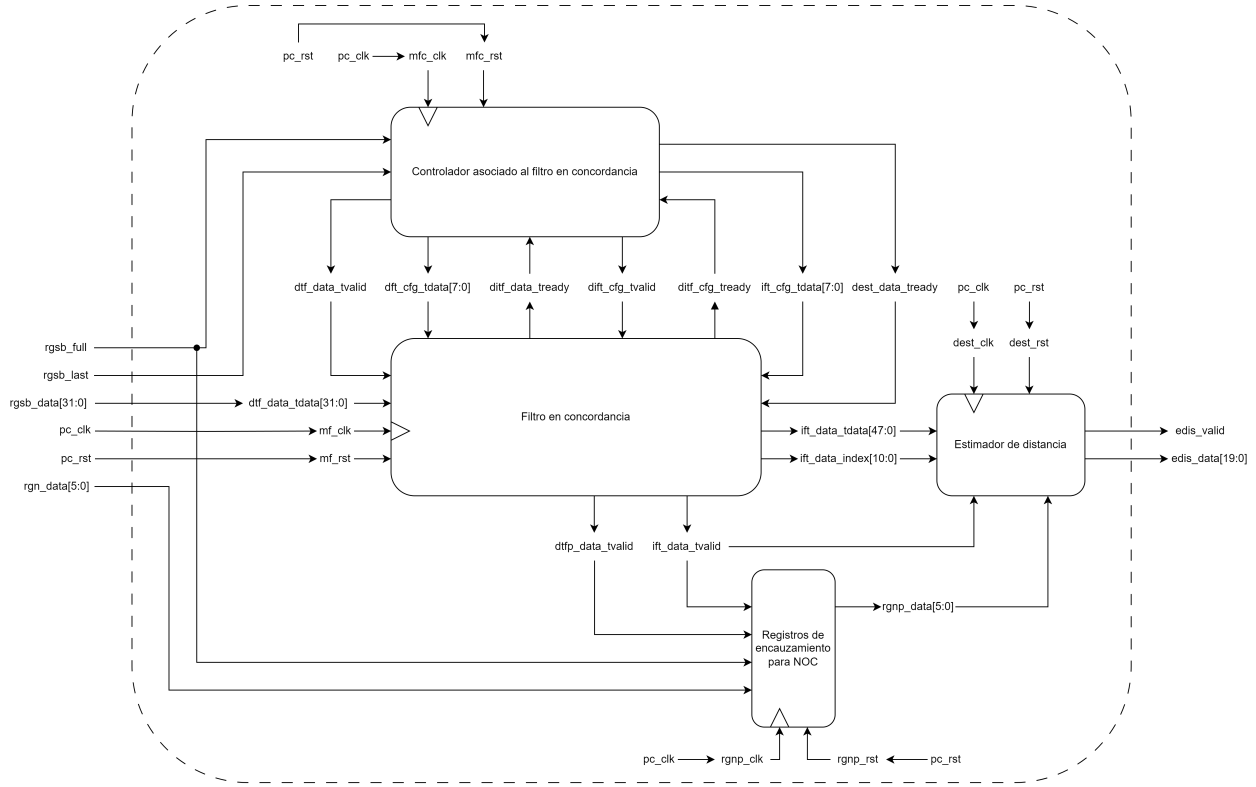


Figura 3.34: Arquitectura final del sistema.

La primera señal de habilitación está determinada por el nivel lógico del puerto `rgsb_full` del sistema, el cual indica la disponibilidad de una compuerta de alcance lista para ser procesada. Un nivel lógico alto en este puerto marca el inicio del suministro de muestras al filtro en concordancia, iniciando así la secuencia de procesamiento dentro del núcleo. La segunda señal de habilitación se encuentra dada por el nivel lógico del puerto `dtf_data_tvalid` del filtro en concordancia, el cual indica el inicio de la transferencia de los resultados de la FFT hacia el multiplicador complejo dentro de la cadena de procesamiento del filtro. Por último, la tercera señal de habilitación está determinada por el nivel lógico del puerto `ift_data_tvalid` del filtro en concordancia, el cual señala que la IFFT ha comenzado a suministrar muestras al estimador de distancia. La salida del tercer registro corresponde al NCAA, y se le entrega al puerto `rgnp_data` del estimador de distancia, donde se enruta hacia la unidad de cálculo correspondiente.

3.4. Conclusiones

El modelo de procesamiento desarrollado simula las señales recibidas por un ULA de 32 elementos que opera a una frecuencia de 2.5 GHz mediante un diagrama de sistema en AWR Microwave Office. Estas señales corresponden a los reflejos provenientes de un objeto situado a una distancia de hasta 12.288 km dentro del intervalo de simulación y, en caso de ampliarse dicho intervalo, hasta 600 km con base en el PRI del pulso radiado. El objeto puede encontrarse en un ángulo de elevación entre 0 y 90 grados y, si así se requiere, contar con una velocidad radial especificada. Para ello, se genera un pulso rectangular con un ancho de $12.5 \mu\text{s}$, el cual se rellena con una señal sinusoidal cuya frecuencia es, en principio, igual a la frecuencia de operación del arreglo. La generación de dicho pulso asume que la recepción de señales en los elementos del arreglo inicia inmediatamente tras la radiación del pulso por el propio arreglo. Por consiguiente, la distancia al objeto se modela mediante la introducción de un retardo temporal en el pulso generado, mientras que la velocidad radial se representa a través de un desplazamiento en la frecuencia del mismo, proporcional al corrimiento inducido por el efecto Doppler. A partir de este pulso, se generan 32 canales que corresponden a cada elemento del arreglo. El ángulo de elevación se incorpora agregando una cantidad de fase determinada a cada elemento mediante otro retardo, calculado en función de la posición de cada antenna dentro del arreglo y la dirección de llegada de la onda.

De igual forma, este modelo permite simular el acondicionamiento de señales que ocurre previo al procesamiento digital, contemplando una arquitectura heterodina con una etapa única de conversión a frecuencia intermedia de 10 MHz, seguida de su respectiva digitalización. La etapa de conversión hacia IF contempla incorpora un mezclador capaz de realizar la conversión directa desde RF hasta la IF especificada, seguido de un filtro paso bajas que elimina las componentes que no son de interés y un amplificador lineal que garantiza el aprovechamiento del rango dinámico completo del ADC seleccionado. El uso de componentes reales para el mezclador y el ADC permite contar con un modelo más apegado a una implementación real del bloque de recepción.

A continuación, se trasladan los datos generados AWR Microwave Office hacia MATLAB, en donde se realiza el procesamiento digital de señales. En primer lugar, se efectúa la demodulación en cuadratura en cada canal con el fin de obtener las componentes en banda base. Una vez que se han obtenido dichas componentes, se aplican los pesos correspondientes a la formación de ocho haces, permitiendo estimar el ángulo de elevación del objeto detectado. Los pesos para la formación de haz utilizan una distribución de magnitud no uniforme con el fin de disminuir el nivel de los lóbulos laterales.

Posteriormente, se efectúa la compresión de pulso y la estimación de velocidad radial en cada uno de los haces formados. La compresión de pulso se modela mediante la aplicación de un filtro en concordancia en el dominio de la frecuencia, lo que conlleva la obtención del espectro de la señal de cada haz mediante la FFT, su multiplicación por el espectro de un pulso de referencia y la obtención de una señal en el dominio del tiempo a partir de la IFFT. En seguida, se identifica el valor máximo dentro vector de muestras de la señal resultante, a partir del cual se estima la distancia a la que se encuentra el objeto respecto al arreglo. Por su parte, la estimación de velocidad radial se modela a través del cálculo de la pendiente de fase presente en la señal correspondiente a cada haz. Con base en el resultado obtenido, se determina el desplazamiento en frecuencia ocasionado por el efecto Doppler, a partir del cual se realiza la estimación de la velocidad radial del objeto.

En este contexto, resulta de particular interés instrumentar el módulo de compresión de pulso en un FPGA. La implementación de dicho módulo se realiza empleando la misma técnica utilizada en el modelo de procesamiento, adaptando ciertas secciones al entorno de los FPGA. La arquitectura completa del módulo gira en torno a la formación, la gestión y el procesamiento de un número determinado de intervalos de muestras más pequeños, denominados compuertas de alcance. Dichas compuertas permiten dividir el conjunto total de muestras, correspondiente a un intervalo de repetición de pulso, en lotes de menor tamaño, lo que facilita su manejo y procesamiento dentro del FPGA.

El sistema completo se integra por dos etapas principales: un subsistema de compuertas de alcance del haz (SCAH) y un núcleo de procesamiento, capaz de operar en tiempo real. El SCAH se encarga de la formación y gestión de las compuertas de alcance, lo que conlleva la carga de búferes de datos con las muestras provenientes de la etapa de formación de haces y, posteriormente, su entrega al núcleo de procesamiento una vez completados. Por su parte, el núcleo de procesamiento implementa un filtro en concordancia en el dominio de la frecuencia para comprimir el pulso. A partir de las muestras obtenidas a la salida de dicho filtro, el núcleo realiza la estimación de la distancia mediante la identificación del valor máximo.

Dados los alcances establecidos en cuanto a los tiempos de desarrollo del sistema, se opta por instrumentar únicamente el núcleo de procesamiento con el fin de obtener un producto mínimo viable. Dicho núcleo está compuesto por el filtro en concordancia, su controlador asociado, el estimador de distancia y los registros de encauzamiento para los números ordinales de las compuertas de alcance que se encuentran siendo procesadas.

El filtro en concordancia ejecuta la compresión de pulso procesando los datos almacenados en los búferes de 2048 muestras complejas que conforman cada compuerta de alcance. Con

este propósito, se utilizan bloques IP especializados que efectúan las transformadas mediante una arquitectura Radix-4, realizan la multiplicación compleja y gestionan el almacenamiento de los coeficientes espectrales del pulso de referencia. A partir de su salida, se estima la distancia a la que se encuentra el objeto respecto al sistema de recepción. El controlador asociado se implementa mediante una máquina de estados tipo Moore, permitiendo gestionar al filtro con base en los cuatro estados establecidos (inicio, configuración, espera y carga).

Por su parte, el estimador de distancia se implementa mediante descripción funcional y diseños estructurales dedicados, obteniendo la magnitud al cuadrado de cada muestra del pulso comprimido, identificando el valor máximo mediante un comparador y calculando la distancia a través del índice del valor máximo y el número ordinal de la compuerta que se encuentra siendo procesada. Con el fin de mantener una sincronización temporal en la estimación de distancia, se adicionan registros de encauzamiento posterior a cada etapa del procesamiento. A la salida del estimador, se indica si se cuenta con datos disponibles concernientes al cálculo de distancia y entregando la estimación correspondiente. Finalmente, con el objetivo de integrar los elementos que conforman al núcleo de procesamiento, se incorporan tres registros en secuencia que permiten almacenar los números ordinales de las tres compuertas de alcance que pueden ser procesadas de forma simultánea.

A partir de las necesidades específicas de la arquitectura, particularmente el uso intensivo de recursos del filtro en concordancia y la latencia requerida por el estimador de distancia, se propone implementarla en una tarjeta de desarrollo Nexys Video.

Capítulo 4

Resultados

En el siguiente capítulo se presentan los resultados obtenidos a partir de diversos casos de prueba diseñados para validar el funcionamiento del modelo de procesamiento desarrollado y la instrumentación correspondiente. En primer lugar, se establecen múltiples casos de prueba mediante los cuales se verifica el funcionamiento del modelo de procesamiento desarrollado, comparando los resultados obtenidos mediante simulaciones en *software* especializado con el comportamiento teórico esperado. Posteriormente, se presentan las pruebas realizadas para comprobar el funcionamiento del núcleo de procesamiento a través del cual se instrumenta el módulo de compresión de pulso, junto con los recursos lógicos empleados por su implementación en una tarjeta de desarrollo Nexys Video.

4.1. Caracterización del modelo de procesamiento

Con el propósito de validar el funcionamiento del modelo de procesamiento desarrollado, se propone un caso de prueba base mediante el cual se verifica la etapa de generación y acondicionamiento de las señales recibidas por el arreglo. Dicho caso consiste en considerar la recepción de las señales reflejadas por un objeto estático situado a 8313 metros del arreglo y cuyo ángulo de elevación es igual a 45° . Dado que el objeto se encuentra estático, la frecuencia de las señales recibidas por el arreglo es igual a su frecuencia de operación.

A partir de dicho caso base, se derivan tres tipos de casos diferentes, los cuales permiten evaluar cada una de las técnicas de procesamiento en banda base. El primer tipo de pruebas consiste en la variación del ángulo de elevación del objeto estático, manteniendo constante la distancia a la que se encuentra el objeto respecto al arreglo. Para dichas pruebas, se consideran ángulos de elevación de 56.25° y 72.5° ya que permiten analizar los casos en los que el objeto se encuentra exactamente entre dos haces formados o se acerca más a uno

de dichos haces sin ubicarse exactamente en el ángulo correspondiente, respectivamente. El segundo tipo de pruebas consiste en la variación de la distancia a la que se sitúa el objeto respecto al arreglo, manteniendo constante el ángulo de elevación en el que se ubica. Durante dichas pruebas, se sitúa el objeto a 2037 y 4251 metros del arreglo. Finalmente, el tercer tipo de pruebas consiste en la introducción de una velocidad radial determinada, manteniendo la distancia y el ángulo de elevación original del objeto. Para estas pruebas, se consideran las velocidades de crucero de los aviones de entrenamiento DA20i Katana ($226 \frac{km}{h}$, equivalente a $67.78 \frac{m}{s}$) y PC-7 MKX ($441 \frac{km}{h}$, equivalente a $122.5 \frac{m}{s}$) [67] [68]. En el primer caso, el objeto se aleja del sistema de recepción a $67.78 \frac{m}{s}$, lo que conlleva un decremento de 523 Hz en la frecuencia observada, mientras que en el segundo, se acerca con una velocidad de $122.5 \frac{m}{s}$, lo que implica un incremento de 1021 Hz en la frecuencia.

4.1.1. Comportamiento del modelo ante un objeto estático

Generación de señales recibidas por un objeto estático

Para implementar el caso de prueba base expuesto al inicio de esta sección, se requiere asignar el valor correspondiente a la frecuencia de operación del arreglo (2.5 GHz) a la variable f_{tx} , mientras que f_D se iguala a cero. Por otra parte, en lo que respecta al pulso rectangular generado, la variable PRF asociada a la frecuencia de repetición de pulso toma un valor de 250 Hz, ajustando el ancho del pulso mediante la variable PDC, cuyo valor es igual a 0.3125. Este valor corresponde al ciclo de trabajo necesario para obtener un pulso de $12.5 \mu s$. Aunado a esto, el retardo del pulso rectangular que permite representar la distancia al objeto deseada se establece asignando un valor de $55.42 \mu s$ a la variable t_d . Finalmente, se agrega la diferencia de fase requerida en cada canal para situar al objeto en el ángulo de elevación especificado. Esto se logra aplicando retardos equivalentes a múltiplos de la variable tau, cuyo valor aproximado es de 141.42 ps, determinados según la posición que ocupa cada elemento dentro del arreglo.

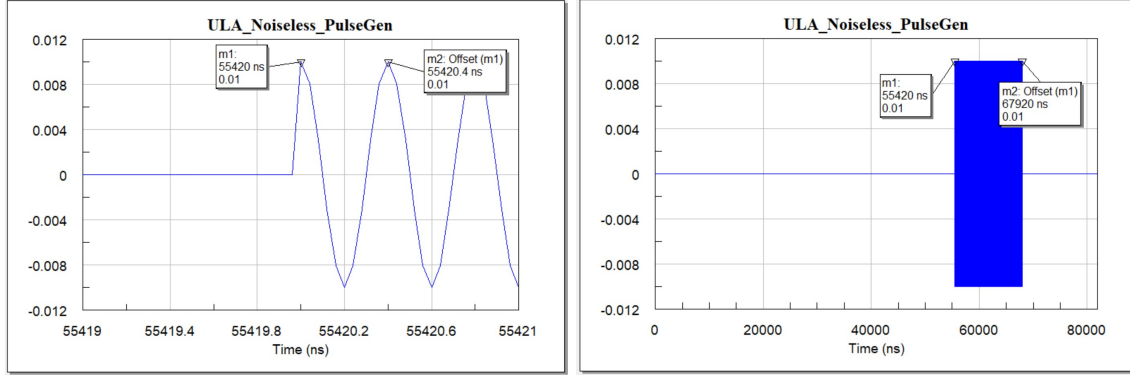


Figura 4.1: Pulso rectangular generado para simular el objeto estático.

En la figura 4.1, se aprecia el pulso rectangular generado para simular las señales reflejadas por el objeto estático para el caso base. Este pulso tiene el ancho especificado de $12.5 \mu\text{s}$, al igual que cuenta con el retardo temporal de $55.42 \mu\text{s}$ que permite situarlo a una distancia de 8313 metros. Así mismo, el pulso se encuentra relleno con una señal sinusoidal cuyo periodo (40 ps) corresponde al de la señal de RF requerida.

Por otro lado, en la figura 4.2, se muestran tres de los 32 canales generados para representar cada uno de los elementos del arreglo. Estos canales cuentan con un retardo de 141.42 ps, el cual permite generar la diferencia de fase resultante a partir del ángulo de elevación del objeto en cuestión. Las señales de los canales generados no son totalmente lisas dado que la plataforma cuenta con una frecuencia de muestreo que, si bien es lo suficientemente alta para generar la información necesaria, es finita.

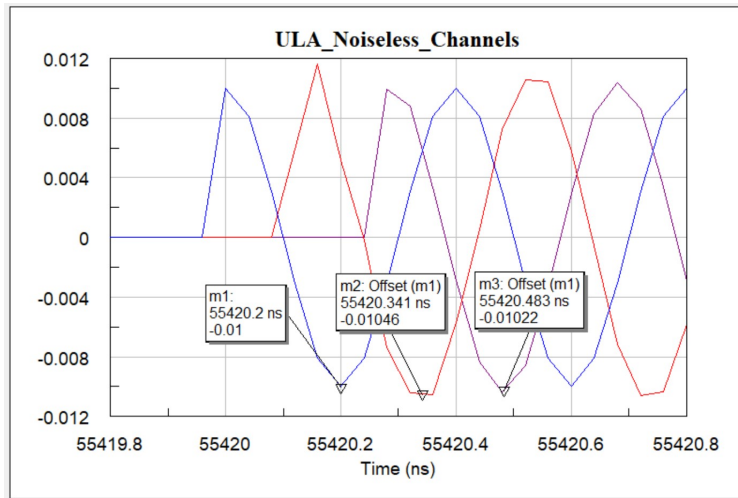


Figura 4.2: Generación de canales con la diferencia de fase requerida.

Acondicionamiento de señales recibidas

A continuación, las señales de RF generadas comienzan su acondicionamiento en la etapa de conversión hacia IF, pasando por el mezclador MM1-0212LS seleccionado y el filtro paso banda tipo Bessel correspondiente. En la figura 4.3, se aprecian los parámetros de diseño utilizados para desarrollar el filtro, al igual que la técnica empleada para su implementación y la magnitud de su respuesta en frecuencia.

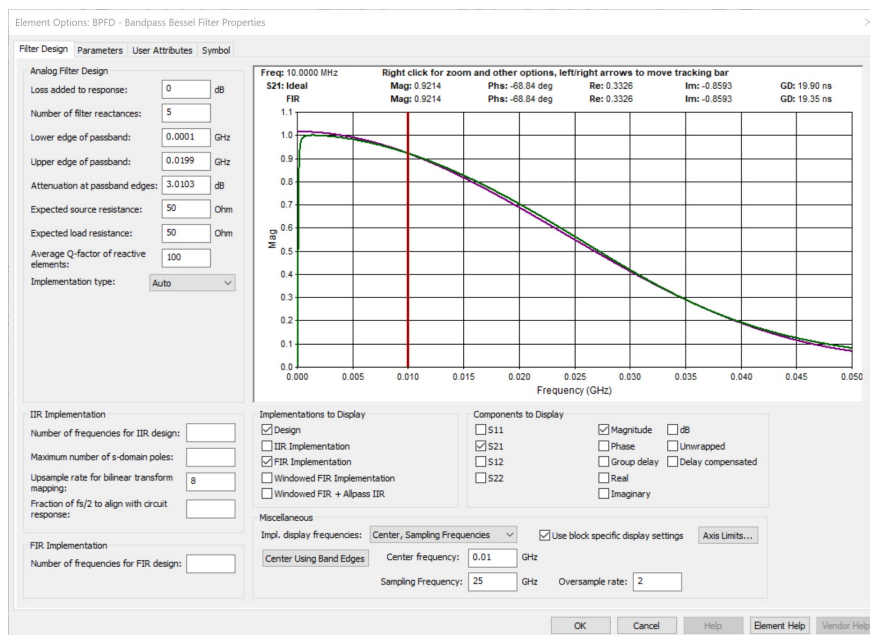


Figura 4.3: Magnitud de la respuesta en frecuencia del filtro paso banda empleado.

Las características de este filtro permiten alcanzar un equilibrio entre selectividad y linealidad, al mismo tiempo que su implementación se apega bastante al comportamiento esperado del filtro. Si bien el diseño del filtro no suprime sustancialmente la segunda armónica y atenúa en cierta medida la frecuencia de interés, este logra mantener la linealidad de la señal de salida al presentar un retardo de grupo prácticamente constante alrededor de 10 MHz, como se aprecia en la figura 4.4. En este contexto, se puede apreciar cierta discrepancia entre las características esperadas y la implementación real del filtro con respecto al retardo de grupo. Por consiguiente, se espera que las señales obtenidas a la salida del filtro para el caso base cuenten con un retardo de 19.35 ns, ligeramente menor al inyectado por el filtro especificado.

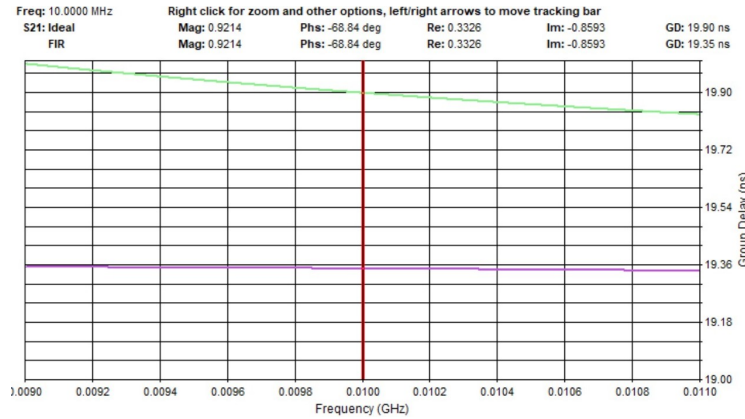


Figura 4.4: Retardo de grupo del filtro paso banda empleado.

En la figura 4.5, se aprecian las señales obtenidas en el primer canal posterior a su paso por el mezclador y por el filtro paso banda para el caso real e ideal. La traza roja corresponde al caso ideal en donde el proceso de conversión hacia IF es ideal, mientras que la traza azul corresponde al modelo de procesamiento desarrollado. En este punto, es posible hacer dos observaciones importantes acerca de las discrepancias entre ambas señales. En primer lugar, se puede apreciar la introducción del retardo de grupo esperado, correspondiente a 19.35 ns, al igual que cierto grado de distorsión debido a que se cuenta con una señal asimétrica. Esta última observación surge al comparar las amplitudes de los picos y de los valles, donde se evidencia una ligera diferencia entre ambos valores.

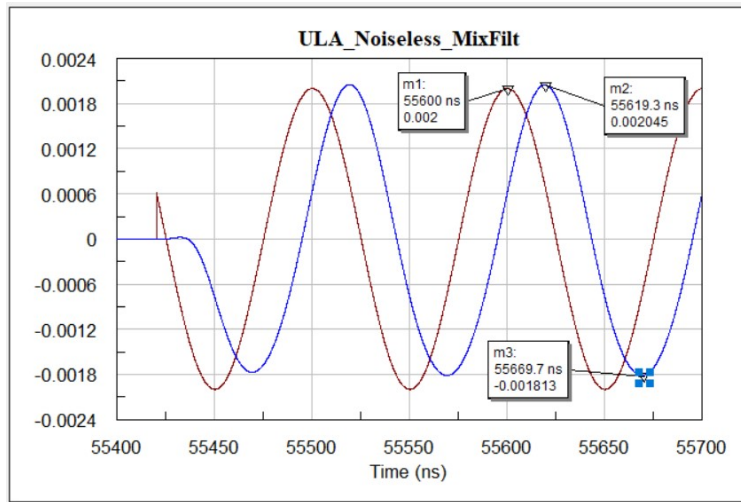


Figura 4.5: Comparación del caso real y caso ideal de la señal de IF obtenida posterior al filtro.

Con el fin de verificar dicho comportamiento, se obtiene el espectro de la señal de IF filtra-

da, el cual se muestra en la figura 4.6. Dicho espectro confirma la presencia de componentes armónicas no deseadas, lo que explica la distorsión observada. En aras de cuantificar este fenómeno, se puede obtener la distorsión armónica total (THD, por sus siglas en inglés) de la señal según lo estipulado en [69], dando como resultado 5.7 %. Lo anterior permite resaltar el beneficio ofrecido por el modelo de procesamiento, a través del cual se puede evaluar rápidamente el rendimiento de la arquitectura empleada y tomar decisiones con base en ello.

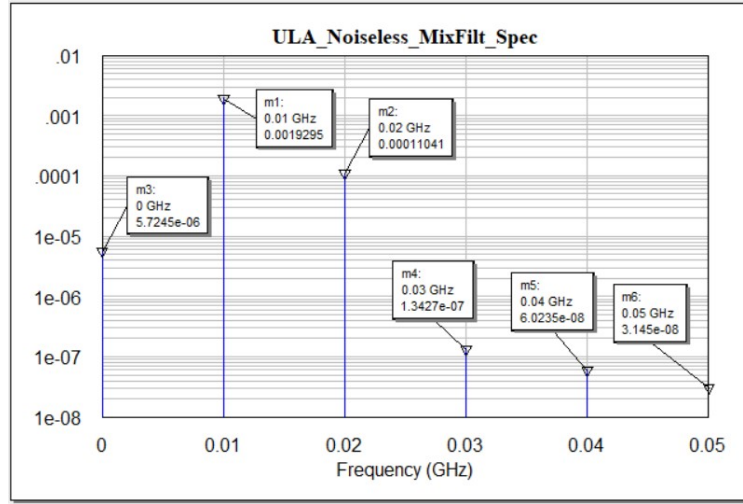


Figura 4.6: Espectro de la señal de IF obtenida posterior al filtrado.

A partir de la amplitud obtenida para la componente de interés, se designa que el bloque de amplificación lineal suministrará 53.5 dB de ganancia, equivalente a un factor de amplificación de 473.15 respecto a la señal de entrada. Finalmente, se aplica la decimación de la señal obtenida, requerida por la plataforma, y se pasa por el bloque correspondiente al ADC. En la figura 4.7, se muestra la señal cuantizada obtenida para el primer canal, la cual presenta un retardo de trece muestras y ocupa la mayor parte del rango dinámico del ADC, equivalente a 2 V pico a pico.

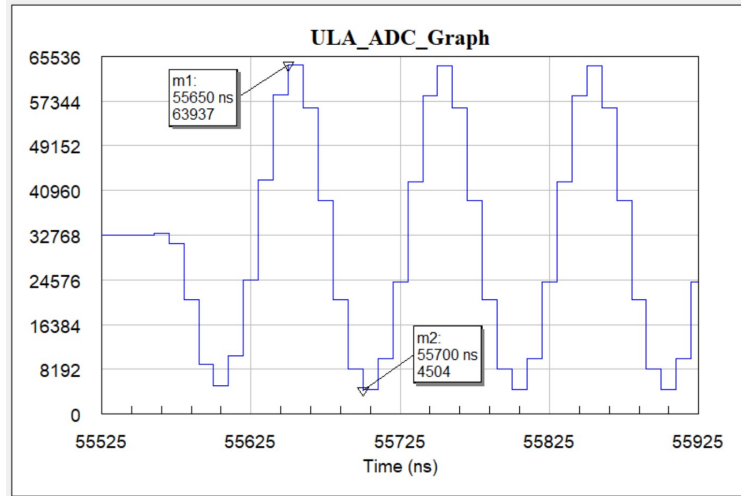


Figura 4.7: Señal de IF amplificada y cuantizada.

Las señales obtenidas se exportan desde AWR Microwave Office hacia MATLAB mediante el *script* correspondiente, a partir de lo cual se cuenta con una interacción ideal entre el ADC y el sistema de procesamiento digital, lo que impide la aparición de una latencia correspondiente al sistema real. Una vez que se han recibido los datos del ADC, se ajusta su representación numérica al formato de punto fijo especificado. En la figura 4.8, se aprecia la señal correspondiente al primer canal, cargada al modelo de procesamiento en Simulink.

Demodulación en cuadratura

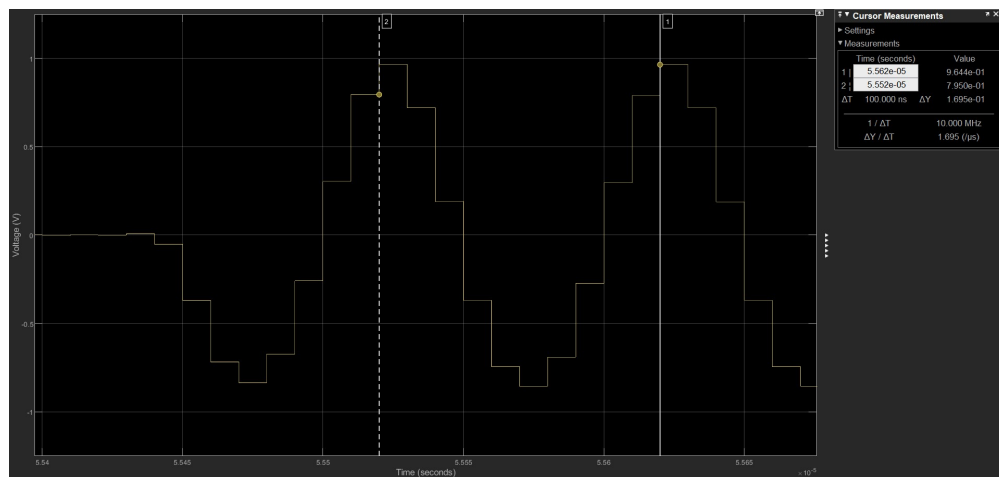


Figura 4.8: Señal de IF digitalizada con ajuste de formato numérico.

Al igual que durante la etapa de conversión hacia IF, la demodulación en cuadratura requiere que su filtro correspondiente no distorsione la señal de salida. Al analizar la respuesta del filtro utilizado, concretamente el retardo de grupo mostrado en la figura 4.9, se observa que el filtro adiciona un retardo de aproximadamente 82.5 ns para todas las frecuencias menores a 100 kHz. Esto asegura que, en caso de recibir múltiples pulsos correspondientes a los reflejos de diferentes objetos en movimiento la señal no se distorsionará, siempre y cuando el desplazamiento en la frecuencia debido al efecto Doppler sea menor a 100 kHz. En una implementación real, esto puede resultar excesivo ya que un desplazamiento en frecuencia de 100 kHz correspondería a un objeto con una velocidad radial absoluta de $12000 \frac{m}{s}$, lo cual es imposible para la mayoría de las aplicaciones. Por consiguiente, se puede ajustar el diseño del filtro de tal forma que el intervalo de frecuencias para el que se mantiene el mismo retardo de grupo vaya acorde a las necesidades del sistema de radar en cuestión. Una vez más el modelo de procesamiento demuestra las ventajas que ofrece el desarrollo de herramientas en *software* especializado para el diseño de arquitecturas en este ámbito.

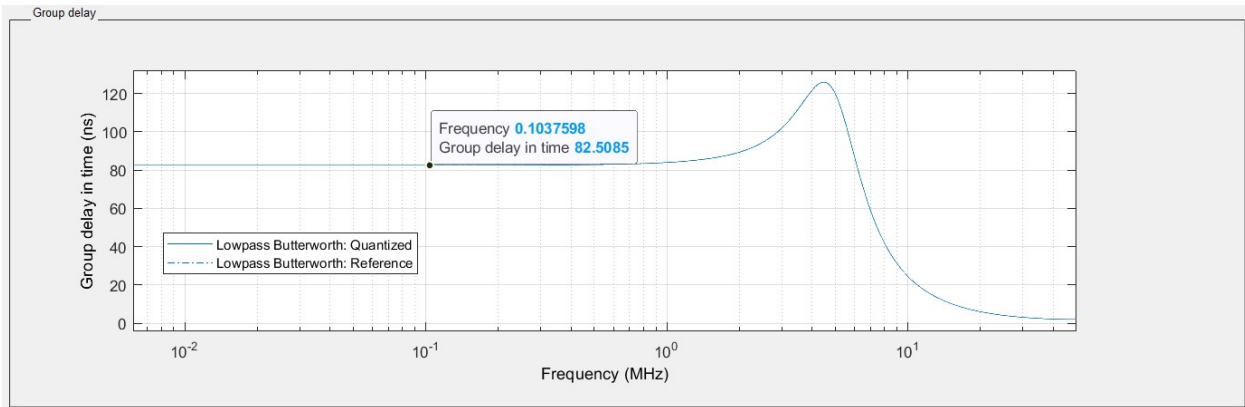


Figura 4.9: Retardo de grupo del filtro paso bajas empleado.

En la figura 4.10, se aprecian las componentes obtenidas a la salida de la demodulación en cuadratura del primer canal. En dicha gráfica, la traza amarilla corresponde a la componente en fase, mientras que la traza azul corresponde a la componente en cuadratura. Las formas de onda para ambas componentes coinciden con las descritas en el modelo teórico para un objeto estático, al ser constantes a lo largo del tiempo para el periodo de recepción del pulso.

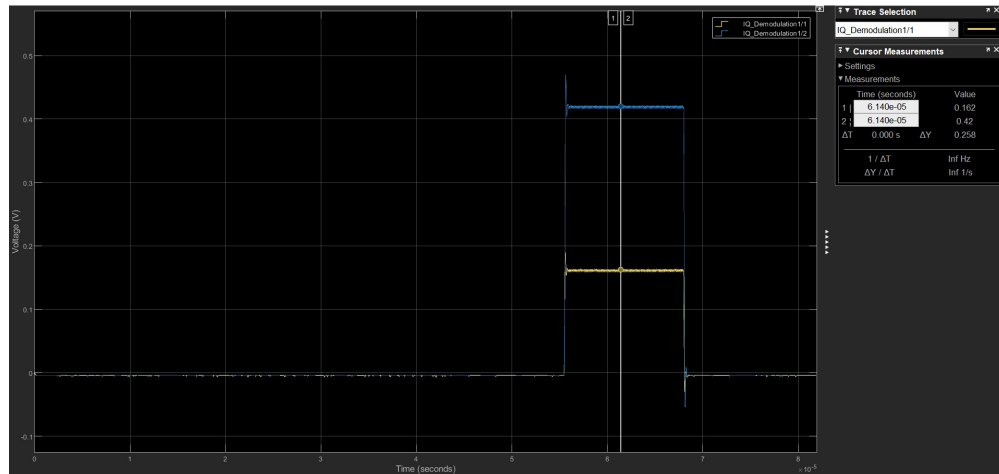


Figura 4.10: Componentes obtenidas del demodulador en cuadratura en el primer canal.

Idealmente, para el caso base evaluado, la componente en cuadratura tomaría un valor cercano a cero, obteniendo únicamente la componente en fase que corresponde a la parte real de la señal en banda base. Sin embargo, este no es el caso, por lo que es posible inferir que se le adiciona cierta fase a la señal en algún punto del procesamiento. Esto podrá medirse en la siguiente sección dado que se pasará la señal a su forma polar, lo que permite conocer con exactitud su fase.

Procesamiento digital en banda base

En la figura 4.11, se aprecia la magnitud (gráfica superior) y la fase (gráfica inferior) de la señal en banda base obtenida. Es posible observar que, en efecto, se añade una fase de 1.2 radianes a la señal, la cual puede ser atribuida al filtro paso banda utilizado durante la etapa de conversión hacia IF. Al examinar la respuesta en frecuencia de dicho filtro, se aprecia que el desfase debería ser igual a -1.2 radianes; sin embargo, ocurre un cambio de signo durante la demodulación en cuadratura al multiplicar la señal de IF digitalizada por los NCO empleados.

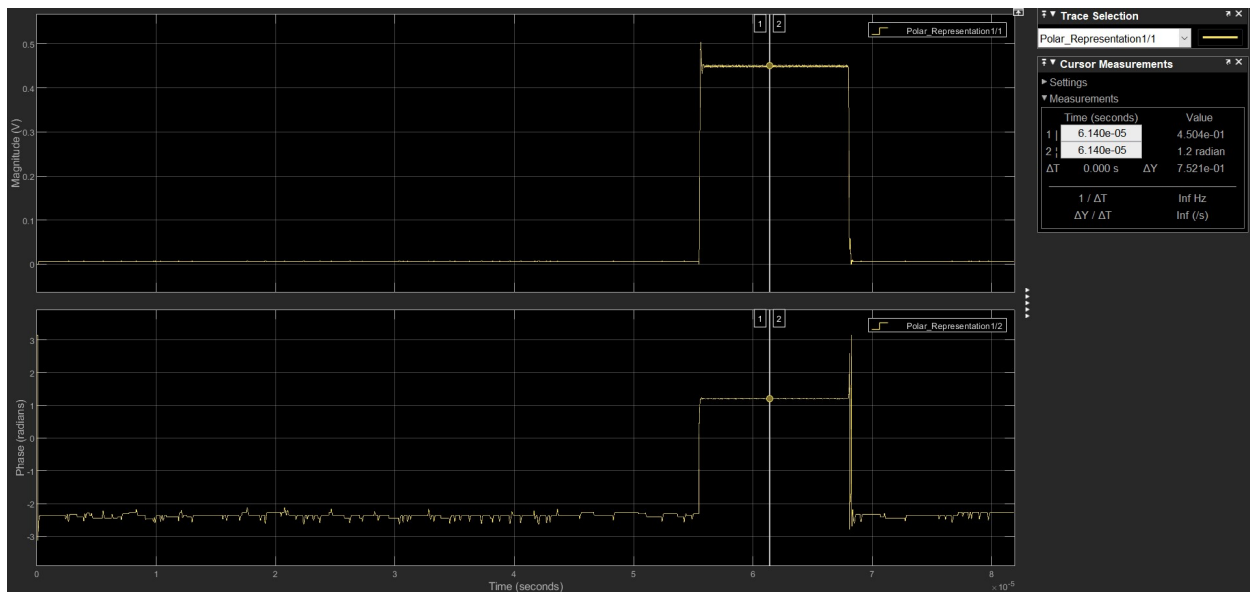


Figura 4.11: Forma polar de las componentes obtenidas en el primer canal.

Por su parte, el filtro paso bajas empleado en la demodulación en cuadratura adiciona una cantidad mínima de fase para frecuencias menores a 100 kHz, tal y como se aprecia en la figura 4.12, por lo que su contribución en este contexto es despreciable.

La fase añadida no generará inconvenientes para el procesamiento en banda base si el desfase introducido es el mismo para todos los canales. Esto se debe a que la formación de haces se encarga de compensar únicamente la diferencia de fase entre los elementos del arreglo que aparece debido al ángulo de elevación del objeto, de tal forma que las señales de todos los canales se encuentren en fase. Por consiguiente, aunque se cuente con cierta cantidad de fase añadida, todos los canales contarán con el mismo desfase y, al sumarlos, la interferencia constructiva resultante da lugar a una señal de mayor magnitud.

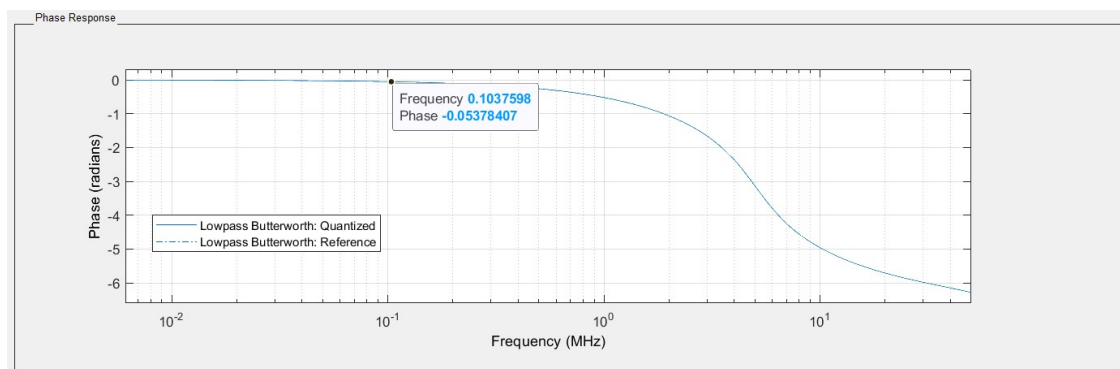


Figura 4.12: Fase de la respuesta en frecuencia del filtro empleado durante la demodulación en cuadratura.

Formación de haces

A partir de la representación polar obtenida de las señales complejas en banda base, se lleva a cabo la formación de ocho haces con el fin de estimar el ángulo de elevación para el caso base. Para ello, se aplican los ocho pesos correspondientes de magnitud y fase en cada canal del arreglo, lo que permite escanear la región del plano xz designada.

En la figura 4.13, se aprecian los patrones de radiación correspondientes a los ángulos de elevación escaneados por el formador de haces. Se puede apreciar que la distribución no uniforme de magnitud empleada permite suprimir los lóbulos laterales a costa de un ensanchamiento del lóbulo principal.

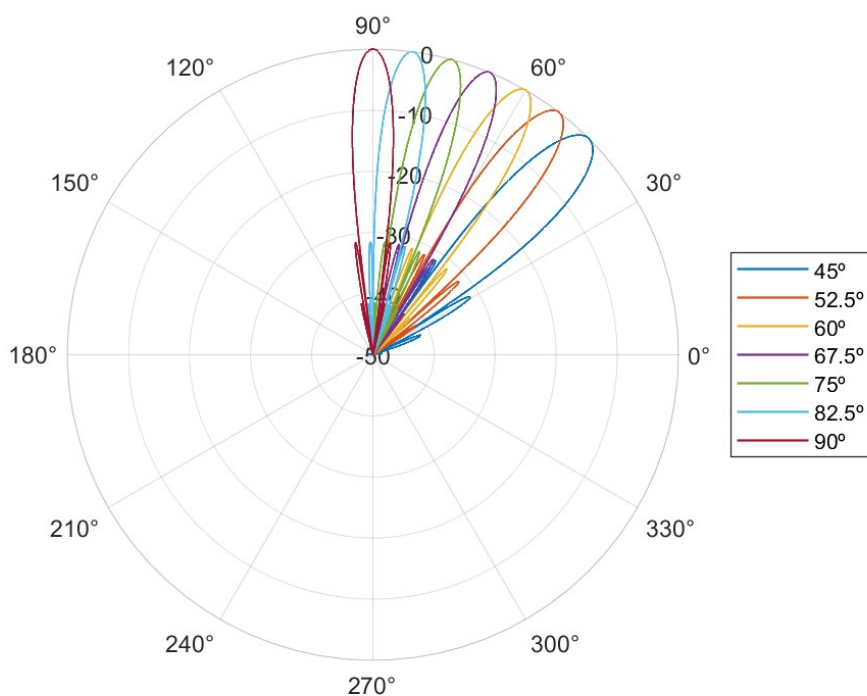


Figura 4.13: Patrones de radiación correspondientes a los ángulos de elevación escaneados.

En la figura 4.14, se aprecia el resultado de la aplicación de los pesos del formador de haces en el canal 16 del arreglo para el caso base. Se opta por mostrar los resultados de dicho canal debido a que la distribución de magnitud empleada (coseno sobre pedestal) suprime el primer canal en su totalidad. Esta distribución no uniforme de magnitud permite reducir la detección de objetos que se encuentren fuera del lóbulo principal a costa de un ensanchamiento del mismo, lo que implica la existencia de un mayor traslape con los haces contiguos.

Al analizar los resultados obtenidos, es posible observar que el primer haz, correspondiente a un ángulo de elevación de 45° y representado mediante una traza amarilla, sitúa la fase del canal en 1.18 radianes durante el periodo de tiempo en el que se recibe el pulso reflejado. A partir de lo mencionado con respecto a la representación polar del primer canal posterior a la demodulación en cuadratura, se puede apreciar que dicho haz permitirá generar la interferencia constructiva deseada con el resto de los canales. Esto es posible al contar aproximadamente con el mismo desfase que el primer canal en el resto de los canales. No obstante, dado que no se cuenta exactamente con la misma fase, el haz resultante se encontrará ligeramente atenuado.

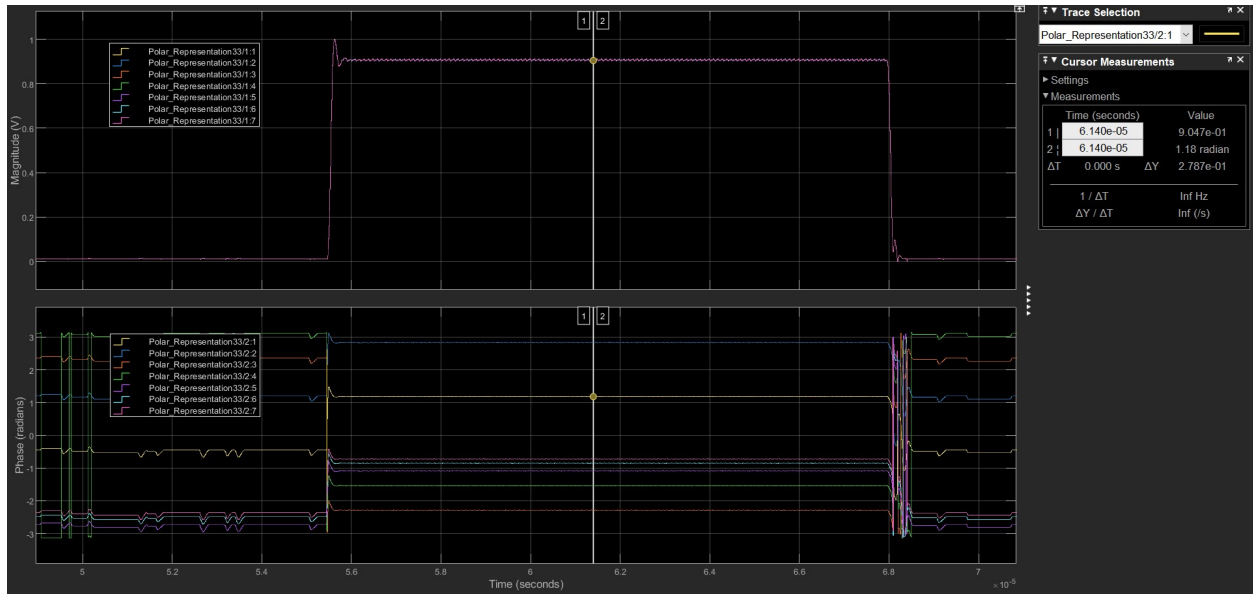


Figura 4.14: Aplicación de pesos del formador de haces en el canal 16 del arreglo.

En la figura 4.15, se aprecian las señales obtenidas a la salida del formador de haces para el caso de prueba que considera un objeto estático situado en 45° . Dichas señales se obtienen a partir de la suma de todos los canales una vez que se les han aplicado los pesos correspondientes a los ocho haces formados. Como se esperaba, el haz con la mayor magnitud corresponde al ángulo de elevación de 45° , seguido del correspondiente a 52.5° . Por consiguiente, se puede inferir que se cuenta con un objeto en dicho ángulo, lo que coincide con el caso de prueba utilizado. Cabe recalcar que el primer haz cuenta con una fase ligeramente mayor a la esperada, lo que indica que los canales no se encuentran completamente en fase y, en consecuencia, existe cierta atenuación.

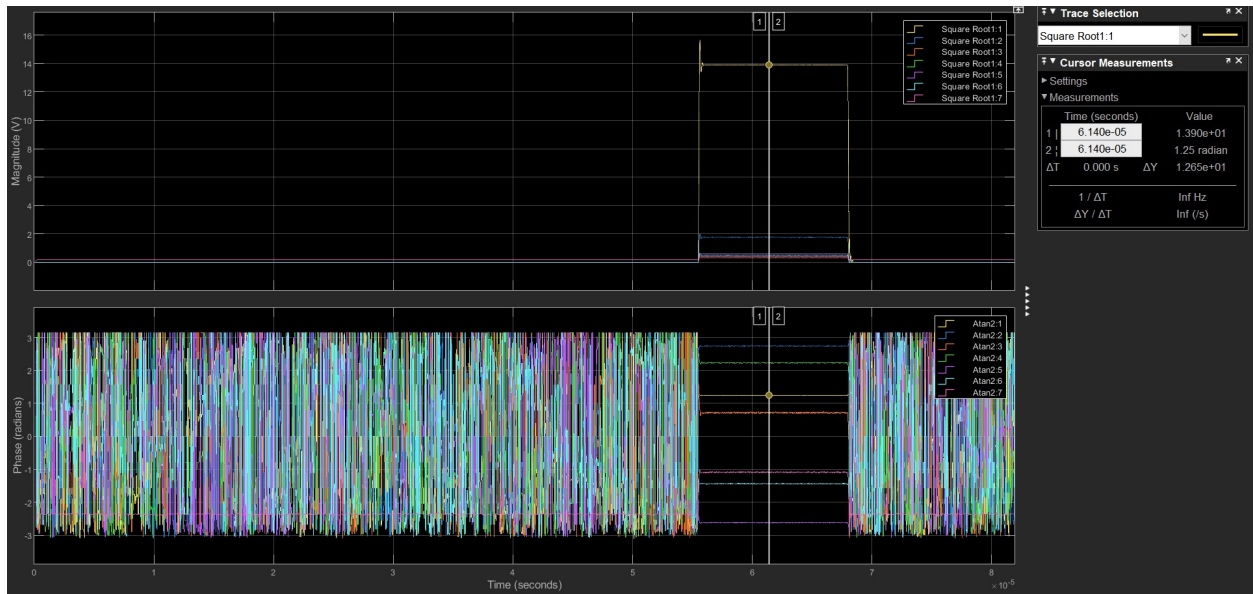


Figura 4.15: Representación polar de las salidas del formador de haces para el caso base.

Este procedimiento se repite para el primer tipo de casos de prueba, en donde se varía el ángulo de elevación en el que se ubica el objeto estático. En la figura 4.16, se aprecia la magnitud de las salidas del formador de haces para el caso en donde el objeto se sitúa en un ángulo de elevación igual a 56.25° . En dicha prueba, se cuenta con dos haces cuyas magnitudes son significativamente mayores al resto; dichos haces corresponden a 52.5° y 60° , en donde el primero tiene la mayor magnitud por un margen pequeño. Este comportamiento es el esperado ya que el objeto se sitúa entre ambos haces, por lo que ambos compensan la mayor parte de la diferencia de fase ocurrida en los canales. Es importante señalar que, en un sistema real, se tendría que ajustar los pesos del formador de haces para escanear el plano correspondiente de forma más minuciosa y determinar con certeza el ángulo de elevación del objeto.

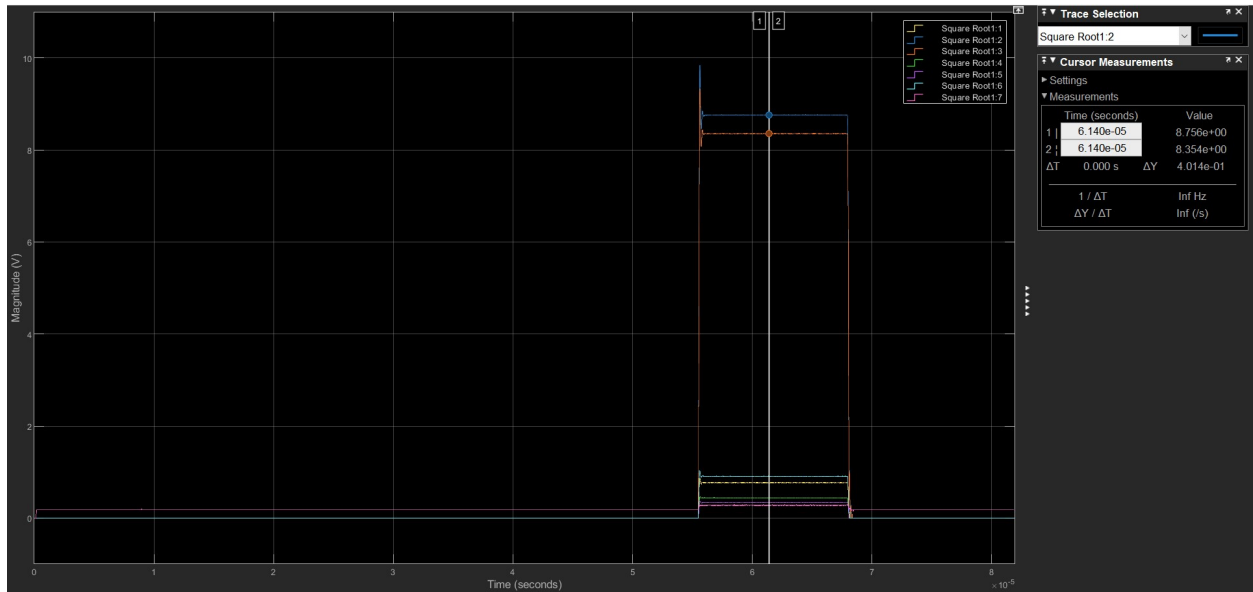


Figura 4.16: Magnitud de las salidas del formador de haces para un objeto situado en $\theta=56.25^\circ$.

De igual forma, en la figura 4.17 se aprecia la magnitud de las salidas del formador de haces para el caso en donde el objeto se sitúa en un ángulo de elevación igual a 72.5° . Para tal prueba, se vuelven obtener dos haces significativamente mayores al resto que corresponden a 67.5° y 75° . Sin embargo, en este caso se puede discernir con mayor certeza el ángulo de elevación del objeto a partir de los haces formados al obtenerse una magnitud considerablemente mayor en 75° . En vista de los resultados obtenidos para las estimaciones del ángulo de elevación, el modelo desarrollado cumple exitosamente el objetivo de diseño correspondiente.

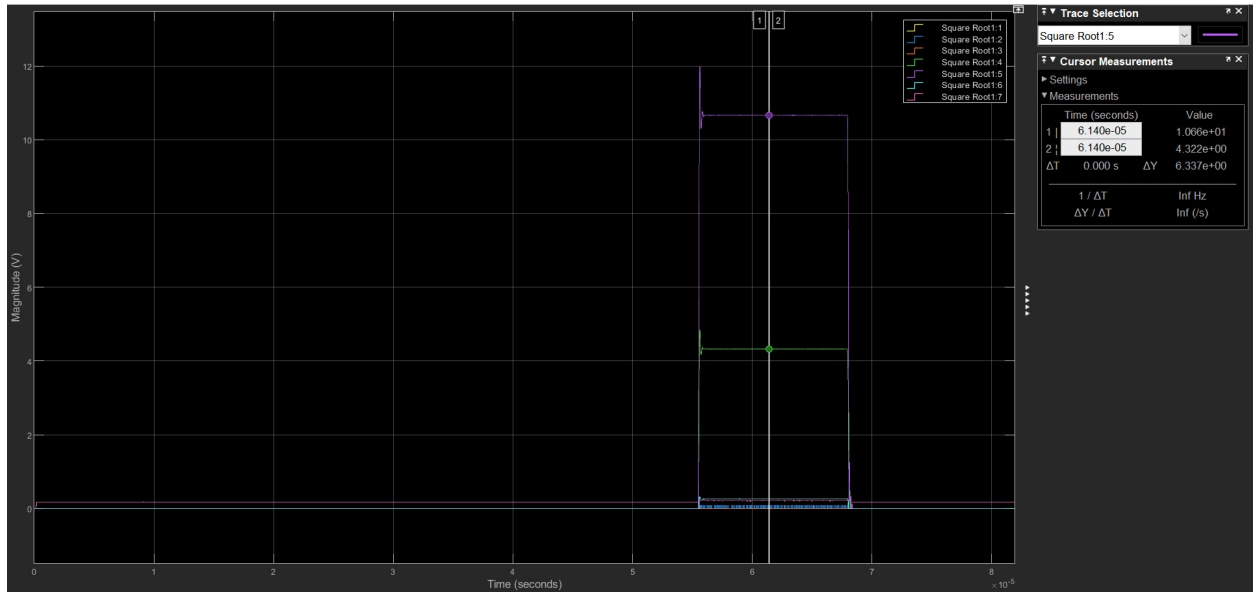


Figura 4.17: Magnitud de las salidas del formador de haces para un objeto situado en $\theta=72.5^{\circ}$.

Compresión de pulso

Las componentes obtenidas a la salida del formador de haces son exportadas a MATLAB nuevamente, en donde se lleva a cabo la compresión de pulso mediante la aplicación del filtro en concordancia. Para ello, se retoma el caso base en donde se sitúa el objeto en un ángulo de 45° y a una distancia de 8313 metros. En la figura 4.18, se aprecian las señales obtenidas a la salida del compresor de pulso para los ocho haces formados, considerando dicho caso de prueba. Las formas de onda resultantes coinciden con aquella predicha en el modelo teórico, ya que se cuenta con señales triangulares en cada uno de los haces.

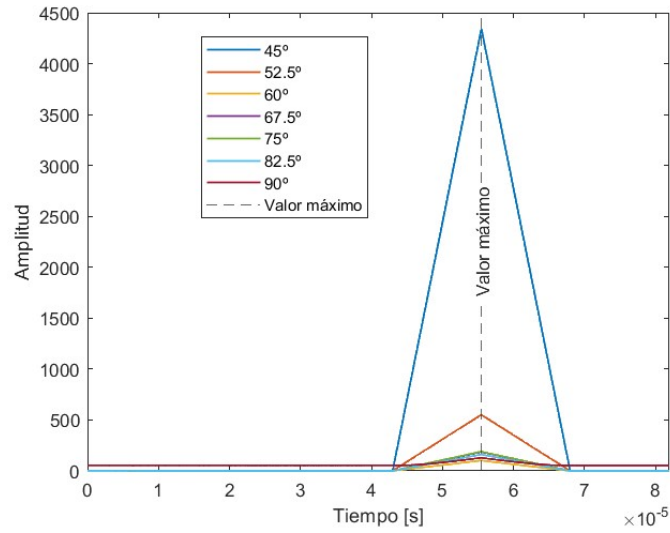


Figura 4.18: Compresión de pulso en los ocho haces formados para el objeto estático.

Como se esperaba, el primer haz resulta en el pico más pronunciado, a partir de lo cual se obtiene una estimación de distancia de 8328 metros. La discrepancia observada entre la distancia estimada y el valor real se debe al retardo de grupo total introducido por los filtros empleados, el cual equivale aproximadamente a diez muestras previo a la decimación aplicada. Dicho error se compensa hasta al restar quince metros a la estimación resultante, lo que equivale a restar diez muestras previo a la decimación. Así mismo, a pesar de haberse obtenido una señales significativamente más atenuadas en el resto de los haces, la compresión de pulso permite conseguir una estimación de distancia cuyo valor se acerca mucho al valor real, según se aprecia en la tabla 4.1.

Ángulo de elevación [°]	Caso base	
	Distancia estimada [m]	Error porcentual
45	8328	0.18 %
52.5	8328	0.18 %
60	8334	0.25 %
67.5	8328	0.18 %
75	8328	0.18 %
82.5	8328	0.18 %
90	8328	0.18 %

Tabla 4.1: Resultados no compensados para la compresión de pulso en los ocho haces formados durante el caso base.

Al igual que con la estimación del ángulo de elevación, se emplean dos casos de prueba adicionales en donde se mantiene el ángulo de elevación, variando la distancia a la que se encuentra el objeto a 2037 y 4251 metros. Las señales obtenidas para ambos casos se muestran en la figura 4.19, cuyas formas de onda corresponden a señales triangulares.

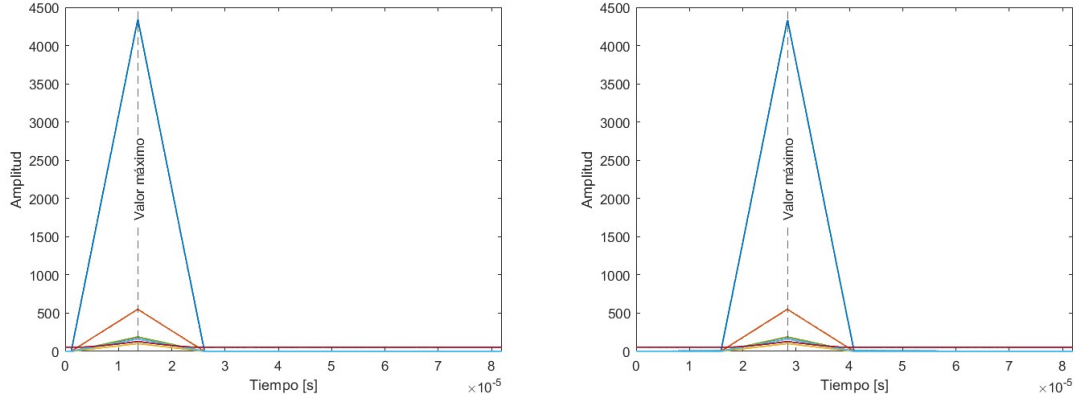


Figura 4.19: Compresión de pulso en los ocho haces formados para un objeto situado a 2037 (izq.) y 4251 metros (der.).

Las estimaciones obtenidas en ambos casos para cada haz se muestran en la tabla 4.2, en donde se aprecia que se consiguen de nueva cuenta estimaciones muy cercanas al valor real. Del mismo modo que con el caso base, se cuenta con un retardo de grupo que desplaza la estimación quince metros, siendo posible compensarlo fácilmente como se mencionó anteriormente.

Ángulo de elevación [°]	2037 m		4251 m	
	Dist. est. [m]	Err.	Dist. est. [m]	Err.
45	2052	0.74 %	4266	0.35 %
52.5	2052	0.74 %	4266	0.35 %
60	2052	0.74 %	4260	0.21 %
67.5	2052	0.74 %	4266	0.35 %
75	2052	0.74 %	4266	0.35 %
82.5	2052	0.74 %	4266	0.35 %
90	2052	0.74 %	4260	0.21 %

Tabla 4.2: Resultados no compensados para la compresión de pulso en los casos de prueba pertinentes.

Estimación de velocidad radial

Para el caso de prueba base en donde se cuenta con un objeto estático, solo resta analizar los resultados obtenidos para la estimación de la velocidad radial. Para ello, se utiliza el índice del valor máximo obtenido durante la compresión de pulso para determinar el intervalo de la señal obtenida a la salida del formador de haces que será utilizado para la estimación correspondiente.

En la tabla 4.3, se muestran los resultados obtenidos para la estimación de la velocidad radial en los ocho haces formados del caso base. En este contexto, se puede observar que no se obtiene una velocidad radial igual a cero en ninguno de los haces formados. El haz correspondiente a 45° es aquel que cuenta con la velocidad radial más cercana a cero, obteniéndose una estimación igual a $-4.857 \frac{m}{s}$. Estos resultados evidencian un área de oportunidad dentro del modelo actual, ya que su rendimiento podría optimizarse mediante la incorporación de un mecanismo adicional capaz de distinguir entre objetos estáticos y aquellos con velocidades muy cercanas a cero.

Ángulo de elevación [$^{\circ}$]	Caso base
	Velocidad radial [m/s]
45	-4.8570
52.5	-8.9369
60	21.1766
67.5	16.5139
75	-41.7704
82.5	-9.5198
90	19.4281

Tabla 4.3: Resultados obtenidos para la estimación de velocidad radial del caso base.

4.1.2. Comportamiento del modelo ante un objeto en movimiento

Una vez que se ha analizado el comportamiento del modelo de procesamiento ante objetos estáticos, se procede con los dos casos de prueba propuestos para objetos en movimiento. Al igual que con los objetos estáticos, se verifica el funcionamiento de la etapa de generación y acondicionamiento de señales recibidas por el arreglo.

Generación y acondicionamiento de señales recibidas por un objeto en movimiento

Para los casos de prueba empleados, la etapa de generación y acondicionamiento de señales recibidas por el arreglo produce señales cuyas IF son diferentes a 10 MHz. En el caso del objeto que se aleja del sistema de recepción con una velocidad radial de $62.78 \frac{m}{s}$, se obtiene una señal cuya IF es igual a 10.000523 MHz. El movimiento del objeto produce una disminución en la frecuencia de las señales de RF generadas que, al pasar por la etapa de conversión hacia IF, desemboca en un ligero aumento de la IF. Esto se puede comprobar al medir el semiperíodo de la señal de IF, mostrado en la figura 4.20, a partir de lo cual se obtiene la frecuencia de la señal. De igual forma, en dicha figura se puede notar que el filtro paso banda empleado introduce la misma cantidad de retardo que para el caso estático, lo que reitera la linealidad del filtro.

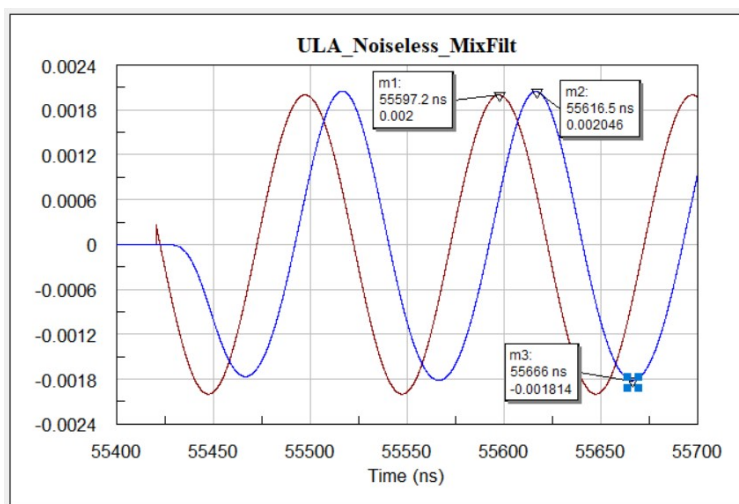


Figura 4.20: Señal de IF obtenida posterior al filtro para el primer objeto en movimiento.

Por otro lado, para el otro caso en donde el objeto se acerca al sistema de recepción con una velocidad radial de $122.5 \frac{m}{s}$, se obtiene una señal cuya IF es igual a 9.998979 MHz. En este caso, la dirección del movimiento del objeto tiene el efecto contrario al caso anterior, ya que se obtiene una disminución en la IF. Al igual que en el primer caso, el filtro paso banda introduce una vez más el mismo retardo de grupo a la señal, tal y como se aprecia en la figura 4.21, por lo que dicho filtro cumple con sus requerimientos.

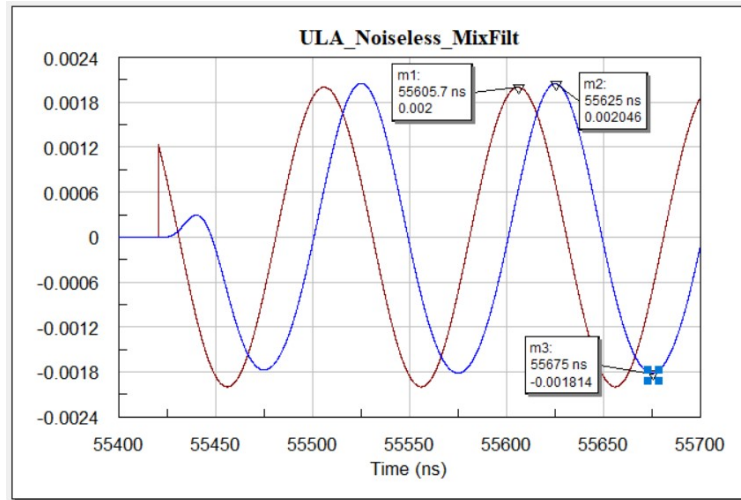


Figura 4.21: Señal de IF obtenida posterior al filtro para el segundo objeto en movimiento.

Procesamiento digital en banda base

En ambos casos, al pasar la señal de IF por el demodulador en cuadratura y al obtener su representación polar, se percibe que la fase recuperada ya no es constante a lo largo del tiempo, tal y como se muestra en las figuras 4.22 y 4.23. De nueva cuenta, las componentes del primer canal adquieren cierto desfase respecto al caso ideal debido a la fase introducida por el filtro paso banda tipo Bessel.

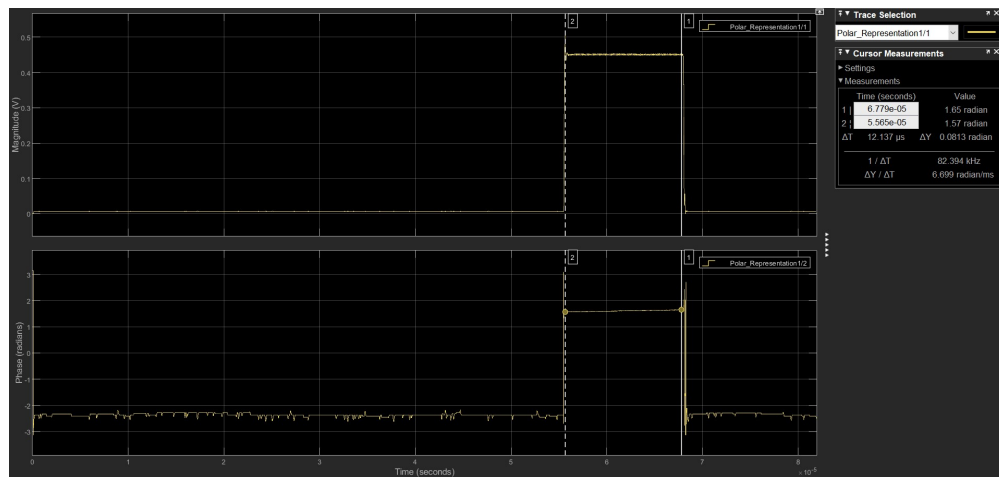


Figura 4.22: Representación polar de la señal obtenida en el primer canal para el primer objeto en movimiento.

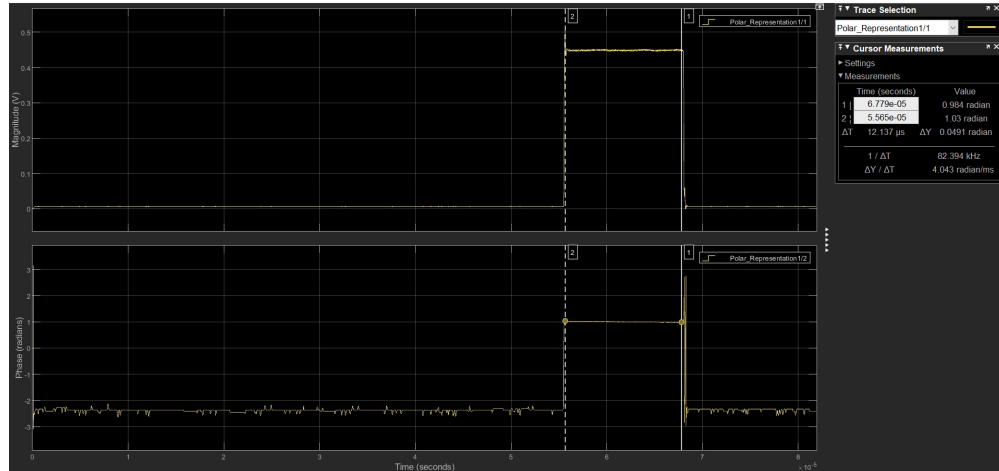


Figura 4.23: Representación polar de la señal obtenida en el primer canal para el segundo objeto en movimiento.

Formación de haces

Las salidas del formador de haces para ambos casos de prueba se muestran en las figuras 4.24 y 4.25. Para ambos objetos, se aprecia la estimación correcta del ángulo de elevación mediante una mayor magnitud en el haz correspondiente a 45° . Por consiguiente, el modelo es capaz de estimar el ángulo de elevación del objeto a pesar de que este se encuentre en movimiento. Cabe recalcar que dicha aseveración aplica, por el momento, únicamente para velocidades radiales menores a $122.5 \frac{m}{s}$. Esto se debe a que no se han realizado pruebas exhaustivas de los límites de funcionamiento del modelo.

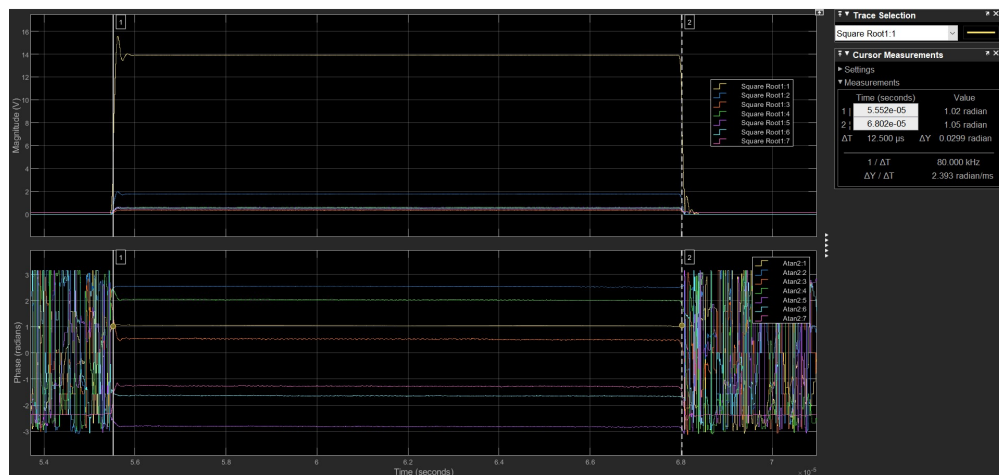


Figura 4.24: Salida del formador de haces para el primer objeto en movimiento.

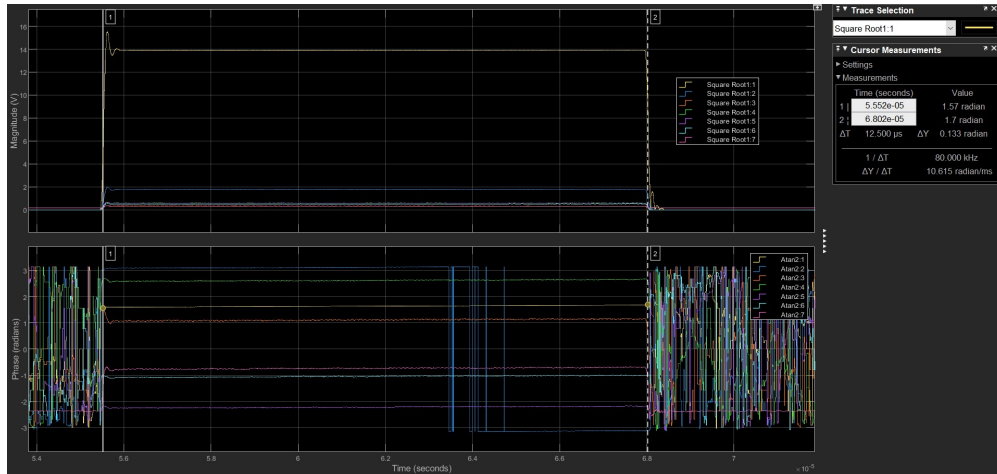


Figura 4.25: Salida del formador de haces para el segundo objeto en movimiento.

Compresión de pulso

Del mismo modo, el desplazamiento en la frecuencia observada tiene un efecto mínimo en la compresión de pulso, como se aprecia en la figura 4.26. Esto se debe a que los desplazamientos en la frecuencia causado por el movimiento de los objetos son relativamente pequeños, causando únicamente un ligero decremento en la magnitud del pico obtenido. En este contexto, se puede volver a observar como el retardo de grupo total introducido por los filtros es prácticamente el mismo respecto al caso del objeto estático. Esto se puede dilucidar al obtener la misma discrepancia en la estimación de distancia para la mayoría de los haces formados en ambos casos de prueba, tal y como se aprecia en la tabla 4.4.

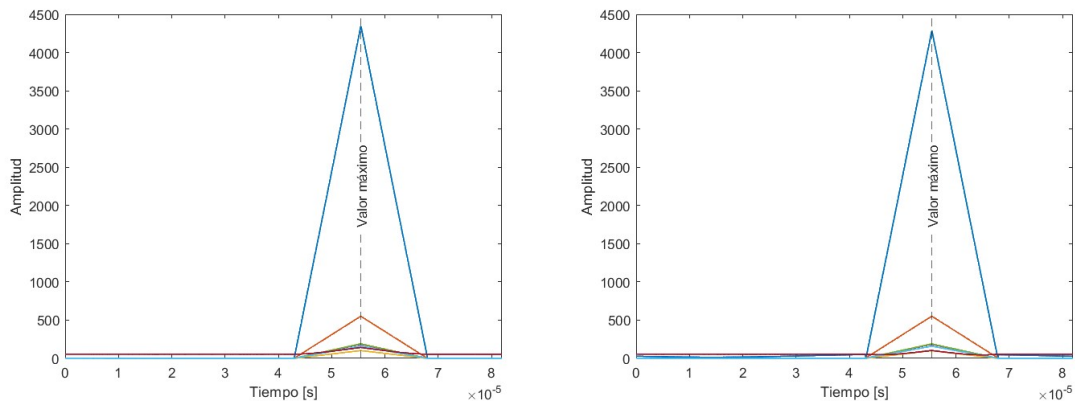


Figura 4.26: Compresión de pulso en los ocho haces formados para el objeto en movimiento.

Ángulo de elevación [°]	62.78 m/s		-122.5 m/s	
	Dist. est. [m]	Err.	Dist. est. [m]	Err.
45	8328	0.18 %	8328	0.18 %
52.5	8328	0.18 %	8328	0.18 %
60	8328	0.18 %	8322	0.11 %
67.5	8328	0.18 %	8322	0.11 %
75	8328	0.18 %	8328	0.18 %
82.5	8328	0.18 %	8322	0.11 %
90	8322	0.11 %	8322	0.11 %

Tabla 4.4: Resultados no compensados para la compresión de pulso de los objetos en movimiento.

Estimación de la velocidad radial

Por último, se realiza la estimación de velocidad radial en cada uno de los haces formados para ambos objetos en movimiento, cuyos resultados se muestran en la figura 4.5. En este punto, se puede observar que solo se obtienen estimaciones precisas para los haces correspondientes a 45° y 82.5° en el primer caso y 45° , 52.5° y 75° en el segundo. En ambos casos, los ángulos en donde se recibe la señal reflejada tienen errores porcentuales bajos, mientras que en el resto no se existe un objeto móvil, lo que provoca un aumento significativo en el error de la estimación. Por consiguiente, este método requiere de algún mecanismo que le permita discriminar aquellos haces en donde es poco probable que se encuentre el objeto, evitando que se realice la estimación o ignorando su resultado completamente.

Ángulo de elevación [°]	62.78 m/s		-122.5 m/s	
	Vel. rad. [m/s]	Err.	Vel. rad. [m/s]	Err.
45	60.0328	-4.37 %	-124.7283	1.82 %
52.5	54.0101	-13.97 %	-127.254	3.88 %
60	110.1573	75.47 %	-300.3581	145.19 %
67.5	54.7872	-12.73 %	-144.9335	18.31 %
75	46.0446	-26.65 %	-115.4028	-5.79 %
82.5	64.1127	2.13 %	-161.2531	31.64 %
90	51.873	-17.37 %	-193.3094	57.80 %

Tabla 4.5: Resultados obtenidos para la estimación de velocidad radial de los objetos en movimiento.

4.2. Resultados de la instrumentación del módulo de compresión de pulso en un FPGA

4.2.1. Condiciones de prueba para el núcleo de procesamiento

A partir de los resultados obtenidos a la salida del formador de haces, se crea un *script* en MATLAB con el propósito de generar un banco de pruebas con los datos producidos por el modelo. Para dicho banco, se consideraron únicamente los datos correspondientes al primer haz durante las pruebas asociadas al objeto estático. Los datos dentro de la ventana de simulación representan una compuerta de alcance, por lo que se deciman con el factor correspondiente y adquieren el formato requerido por el núcleo de procesamiento. En este formato, cada muestra tiene una longitud de 32 bits, donde los 16 bits menos significativos corresponden a la componente real y los 16 bits restantes a la componente imaginaria.

Con el fin de verificar la capacidad de operación en tiempo real del núcleo de procesamiento, el banco de pruebas simula la recepción de los datos correspondientes a tres compuertas de alcance contiguas, simulando los tiempos en que las compuertas se llenan y envían su información. La primer compuerta contienen los datos decimados generados para un objeto situado a 8313 metros, mientras que la segunda y la tercera compuerta contienen los datos generados para objetos situados a 4251 y 6375 metros, respectivamente. Lo anterior es equivalente a contar con tres objetos situados a 8313, 14661 y 27198 metros del sistema de recepción. Para todas las estimaciones, se debe efectuar la aplicación del filtro en concordancia y el barrido completo del vector de muestras dentro de los tiempos requeridos.

Las condiciones de prueba se muestran en la figura 4.27. En dicha figura, se designa una señal de reloj con frecuencia de 110 MHz, se aplica un reinicio global del sistema al inicio de la simulación y se simula la recepción de las señales *rgsb_full* y *rgsb_last*. El diseño del núcleo de procesamiento se evalúa utilizando la simulación funcional pos-implementación ofrecida por Vivado dados los alcances del presente trabajo.

```
s_pc_clk <= not s_pc_clk after 4.5ns;
s_pc_rst <= '1' after 9ns;
s_rgsb_full <= '1' after 82129.5ns, '0' after 82138.5ns, '1' after 151528.5ns, '0' after 151537.5ns, '1' after 220927.5ns, '0' after 220936.5ns;
s_rgsb_last <= '1' after 100561.5ns, '0' after 100570.5ns, '1' after 169960.5ns, '0' after 169969.5ns, '1' after 239359.5ns, '0' after 239368.5ns;
s_rgn_data <= "000000" after 82129.5ns, "000001" after 151528.5ns, "000010" after 220927.5ns;
```

Figura 4.27: Condiciones de prueba para el núcleo de procesamiento.

4.2.2. Filtro en concordancia

En la figura 4.28, se aprecia el funcionamiento del filtro en concordancia para las condiciones de prueba antes descritas. En dicha simulación, se observa el procesamiento en cascada

del filtro, siendo capaz de procesar las compuertas de alcance en los tiempos requeridos. En primer lugar, se aprecia que la FFT termina de entregar resultados antes de recibir un nuevo lote de muestras de la siguiente compuerta de alcance. Lo mismo acontece con la IFFT, por lo que el filtro en concordancia es capaz de operar en tiempo real, bajo las condiciones descritas anteriormente.

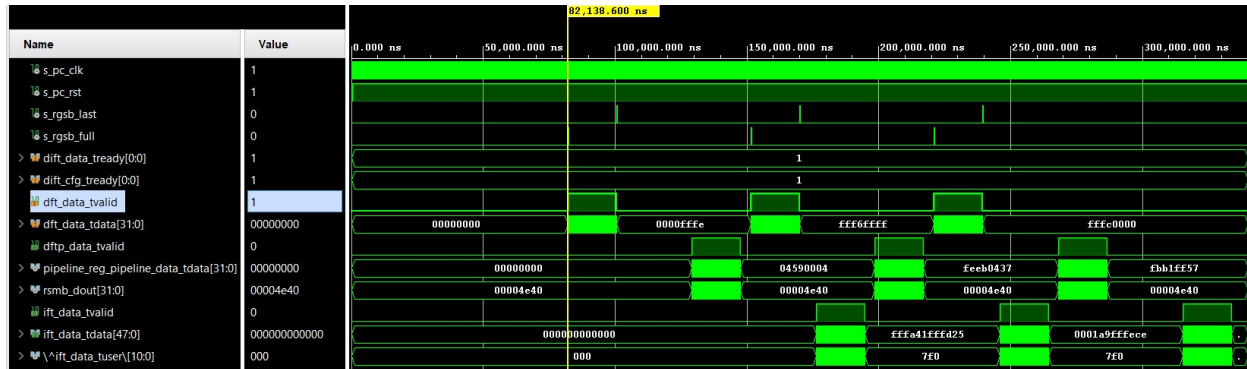


Figura 4.28: Simulación funcional del filtro en concordancia.

Por otro lado, en la figura 4.29, se aprecia el funcionamiento apropiado del controlador asociado al filtro en concordancia. En la simulación, el controlador realiza las transiciones de estado únicamente bajo las condiciones especificadas. El controlador permanece en su estado inicial hasta que recibe la señal dift_cfg_tready, pasando al estado de configuración y enviando la señal correspondiente. Así mismo, se pasa al estado de espera al siguiente ciclo de reloj, manteniéndose en dicho estado hasta que se recibe la señal rgsb_full y el filtro en concordancia esté listo para recibir datos.

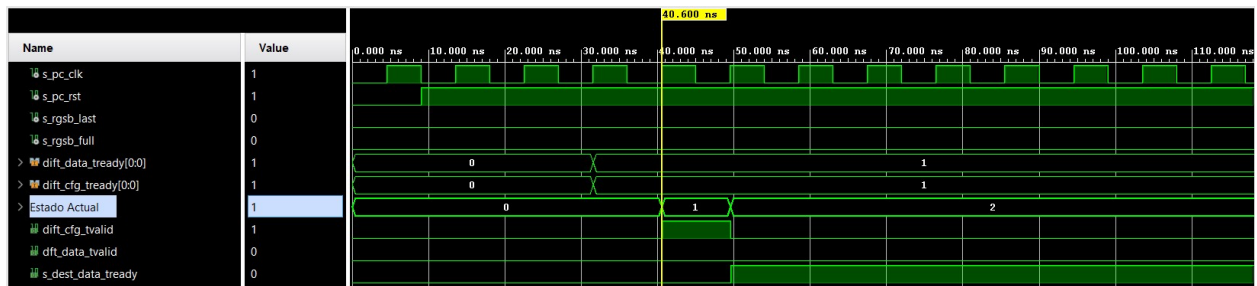


Figura 4.29: Transición de estados dentro del controlador asociado al filtro en concordancia.

Al recibir la señal rgsb_full, el controlador pasa al estado de carga y se mantiene en dicho estado hasta que recibe la señal rgsb_last, que indica el envío de la última muestra de la compuerta. En la figura 4.30, se aprecia la transición de regreso al estado de espera, en donde se mantiene hasta que vuelve a recibir un nivel lógico alto en rgsb_full.

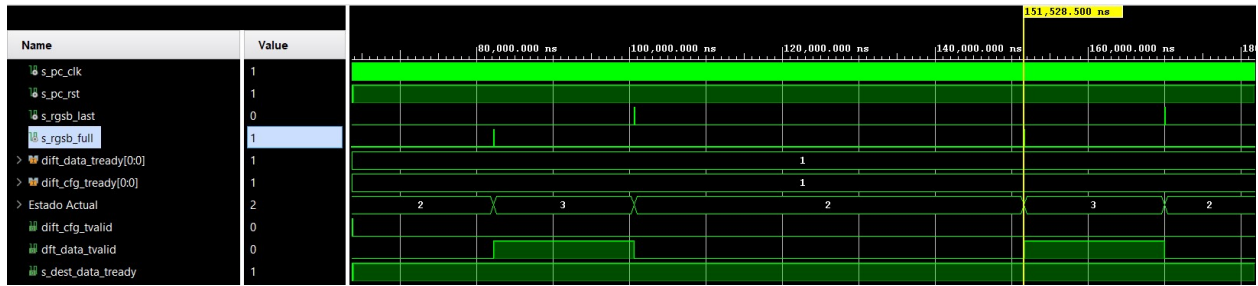


Figura 4.30: Transición al estado de espera del controlador.

4.2.3. Estimador de distancia

Los resultados del filtro en concordancia son recibidos por el estimador de distancia, el cual comienza su operación al contar con datos disponibles de dicha entidad. En la figura 4.29, se aprecia el funcionamiento apropiado de las etapas de encauzamiento del estimador de distancia. Es posible observar que se actualiza el registro de datos de la primera etapa de encauzamiento, correspondiente a la suma de cuadrados, únicamente cuando se cuentan con datos disponibles de la IFFT del filtro en concordancia. A partir de la señales tvalid, se propaga correctamente la habilitación de cada etapa de encauzamiento. Aunado a esto, se aprecia la actualización del valor máximo únicamente cuando se cuenta con un nivel lógico alto en el acarreo de salida del comparador.

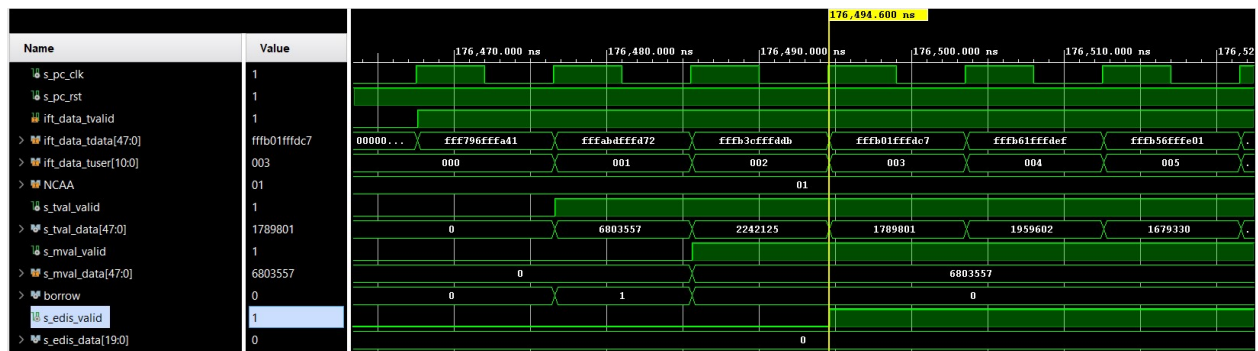


Figura 4.31: Propagación de datos a lo largo de las etapas de encauzamiento.

Por otro lado, en la figura 4.29, se aprecia el resultado de la estimación de distancia a lo largo del barrido del vector de muestras proporcionado. La estimación final obtenida es igual a 8316 metros, lo que se acerca mucho al valor real y coincide con el error de tres metros esperado debido al factor de decimación empleado. De igual forma, es posible notar que se levanta la señal edis_valid durante el periodo indicado y que el subsistema limpia sus salidas al concluir el barrido del vector de muestras.



Figura 4.32: Estimación de distancia para la primera compuerta de alcance.

En la figura 4.33, se aprecia que la estimación de distancia para la segunda compuerta es igual a 14664 metros, lo que se encuentra muy cerca del valor real. Así mismo, la estimación obtenida para la tercera compuerta es igual 27198 metros, tal y como se aprecia en la figura 4.34, lo cual se acerca en igual medida al valor real que las dos compuertas anteriores. Para todas las compuertas, se obtiene un error de tres metros que se atribuye al factor de decimación empleado.



Figura 4.33: Estimación de distancia para la segunda compuerta de alcance.

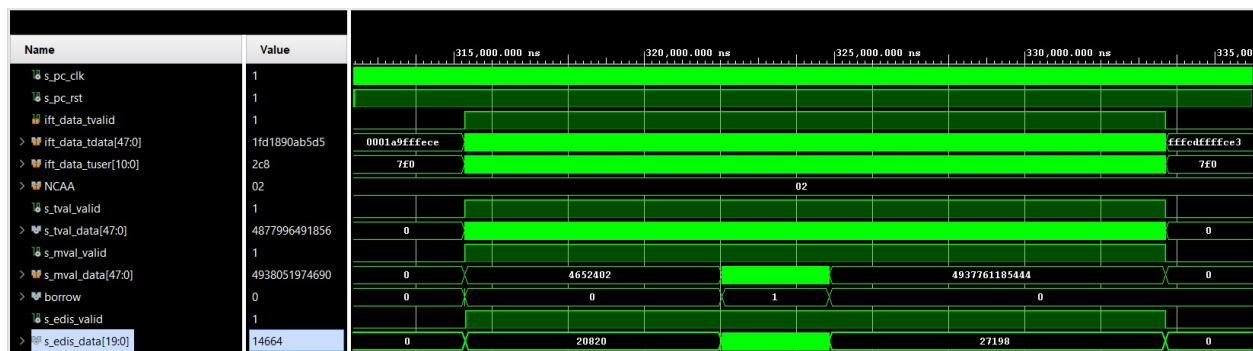


Figura 4.34: Estimación de distancia para la tercera compuerta de alcance.

4.2.4. Recursos requeridos para la implementación del núcleo de procesamiento

Por último, en la figura 4.35, se aprecia la cantidad total de recursos empleados por la implementación final del sistema. La arquitectura desarrollada para el núcleo de procesamiento permite obtener un diseño compacto y fácilmente replicable. A partir de ello, sería posible procesar los ocho haces formados en una sola tarjeta Nexys Video, suponiendo que no se requiera de una cantidad significativa de hardware adicional.

Resource	Utilization	Available	Utilization %
LUT	4386	134600	3.26
LUTRAM	278	46200	0.60
FF	6303	269200	2.34
BRAM	12.50	365	3.42
DSP	42	740	5.68
IO	63	285	22.11

Figura 4.35: Recursos empleados para la implementación del sistema.

En la figura 4.36, se aprecia el desglose de los recursos lógicos empleados por cada elemento del núcleo de procesamiento. Como se esperaba, el filtro en concordancia acapara la mayor cantidad de recursos, en especial los bloques correspondientes a las transformadas y al multiplicador complejo. El resto de los elementos que componen al núcleo utilizan una cantidad mínima de recursos, por lo que si se desea contar con un núcleo más compacto, los esfuerzos deben centrarse en generar un diseño más liviano para el filtro en concordancia.

Name	Slice LUTs (134600)	Slice Registers (269200)	F7 Muxes (67300)	Slice (33650)	LUT as Logic (134600)	LUT as Memory (46200)	Block RAM Tile (365)	DSPs (740)	Bonded IOB (285)	BUFGCTRL (32)
pcore_syst	4386	6303	16	1588	4108	278	12.5	42	63	1
> dest_ssys_inst (dest_ssys)	73	99	0	31	73	0	0	6	0	0
> mfilt_ctrl_inst (mfilt_ctrl)	4	2	0	1	4	0	0	0	0	0
mfilt_ssys_inst (mfilt_ssys_bd_wrapper)	4309	6196	16	1558	4031	278	12.5	36	0	0
> mfilt_ssys_bd_i (mfilt_ssys_bd)	4305	6196	16	1557	4027	278	12.5	36	0	0
> cmpy (mfilt_ssys_bd_cmpy_core_0)	1003	960	0	282	1003	0	0	0	0	0
> dft (mfilt_ssys_bd_fft_core_0)	1412	2108	8	499	1295	117	3.5	12	0	0
> dft_p_data (mfilt_ssys_bd_pipeline_reg_0)	1	66	0	13	1	0	0	0	0	0
> ift (mfilt_ssys_bd_ifft_core_0)	1883	3049	8	768	1722	161	7	24	0	0
> mem_addr_gen (mfilt_ssys_bd_mem_addr_gen_0)	1	11	0	3	1	0	0	0	0	0
> ref_spec_mem (mfilt_ssys_bd_ref_spec_mem_0)	2	2	0	1	2	0	2	0	0	0
rgate_gnum_pipeline_reg_inst (rgate_gnum_pipeline_reg)	1	6	0	1	1	0	0	0	0	0

Figura 4.36: Desglose de recursos empleados para la implementación del sistema.

4.3. Conclusiones

Mediante múltiples casos de prueba, se evalúa el funcionamiento de las etapas de generación, acondicionamiento y procesamiento digital de señales del modelo de procesamiento desarrollado. El caso de prueba base permite corroborar que el proceso de generación de señales recibidas por el arreglo ocurre de forma apropiada. Al acondicionar dichas señales, se aprecia que el filtro paso bandas empleado introduce cierta cantidad de retardo en las señales y que el balance entre selectividad y linealidad involucra ciertos sacrificios como la aparición de un porcentaje dado de THD. Así mismo, el filtro paso bandas presente en la etapa de acondicionamiento introduce una cantidad de fase determinada que se puede vislumbrar posterior a la obtención de la representación polar de las señales demoduladas.

Los casos de prueba empleados para evaluar la formación de haces dieron los resultados esperados y demuestran que la fase añadida por los filtros empleados durante la cadena de procesamiento no influyen con la estimación del ángulo de elevación si el desfase introducido es el mismo en todos los canales. Por su parte, las pruebas realizadas para la compresión de pulso demostraron que la técnica empleada es sumamente robusta, al obtenerse resultados cercanos al valor real aún en aquellos haces que se encontraban sumamente atenuados. Aunado a esto, la discrepancia existente entre las estimaciones y los valores reales se compensa fácilmente mediante una simple resta. Por último, se puede apreciar que la técnica empleada para la estimación de velocidad radial requiere cierto refinamiento mediante la incorporación de algún método de umbralización. Lo anterior permitiría dilucidar en qué haces se cuenta con un objeto, a partir de lo cual se obtiene una estimación de velocidad radial mucho más exacta.

Por su parte, la instrumentación del módulo de compresión de pulso se verifica mediante un banco de pruebas que utiliza las salidas del formador de haces para tres casos en donde se cuenta con un objeto estático. A través de dicho banco, se demuestra la capacidad del núcleo de procesamiento de operar en tiempo real. De forma concreta, se aprecia que el filtro en concordancia cumple con las restricciones de tiempo real, ya que cada transformada finaliza su procesamiento antes de la llegada de un nuevo lote de datos. Esto permite procesar dos compuertas de alcance de forma secuencial dentro del filtro, mejorando el rendimiento del mismo. El controlador asociado al filtro es capaz de gestionar la configuración y la carga de datos mediante los cuatro estados propuestos. De igual forma, se aprecia que el estimador de distancia cumple su función de forma correcta. Las etapas de encauzamiento de dicho elemento permiten procesar cada muestra de forma apropiada, propagándose en las condiciones adecuadas. Así mismo, el estimador es capaz de identificar correctamente el

valor máximo, almacenando su índice a partir del cual se realiza el cálculo de la distancia.

Finalmente, se aprecia que el núcleo es capaz de procesar tres compuertas de alcance de forma simultánea, brindando estimaciones muy cercanas al valor real, las cuales son equiparables con los resultados del modelo del procesamiento. Aunado a esto, los recursos utilizados por el núcleo permiten contar con una arquitectura escalable. En este sentido, el diseño puede hacerse aún más liviano si se logra reducir la utilización de recursos del filtro en concordancia.

Capítulo 5

Conclusiones y trabajo a futuro

La migración de los sistemas de radar hacia el ámbito digital ha permitido que el desarrollo de arquitecturas sumamente sofisticadas. Esto ha traído consigo un incremento exponencial en la complejidad de dichos sistemas a medida que se vuelven cada vez más responsivos a su entorno. A partir de este panorama, surge la necesidad de contar con herramientas que permitan modelar este tipo de sistemas, sin la necesidad de contar con toda la infraestructura asociada.

En respuesta a dicha necesidad, en el presente trabajo se desarrolló un modelo del procesamiento asociado al bloque de recepción de un sistema de radar pulsado. Para ello, se generó un modelo teórico de dicho procesamiento a partir del cual se construyeron los modelos correspondientes en *software* especializado que permitiesen simular la cadena completa de procesamiento. Se consideró que el sistema de radar en cuestión utiliza un arreglo lineal uniforme de antenas en fase compuesto por 32 elementos operando a una frecuencia de 2.5 GHz. Los pulsos radiados por dicho arreglo tienen una duración de $12.5 \mu\text{s}$ y una frecuencia de repetición de 250 Hz.

El modelo de procesamiento se encargó de generar las señales recibidas en dicho arreglo correspondientes a los pulsos reflejados por un objeto situado a cierta distancia del arreglo, con un ángulo de elevación determinado y, si así se requería, con una velocidad radial dada. Dicho modelo contempla una arquitectura heterodina con una etapa única de conversión hacia IF centrada en 10 MHz, seguida de una etapa de conversión analógico-digital cuya frecuencia de muestreo es de 100 MHz. Las señales digitalizadas de IF obtenidas pasan por un demodulador en cuadratura, a partir del cual se obtienen las componentes en banda base de la señal. Dichas componentes permiten realizar la formación de ocho haces mediante los cuales se estimó correctamente el ángulo de elevación del objeto detectado. Posteriormente, se lleva a cabo la compresión de pulso a través de un filtro en concordancia implementado

en el dominio de la frecuencia. Dicha técnica de procesamiento permitió estimar la distancia entre el objeto y el sistema de recepción con bastante exactitud, requiriendo hacer un ajuste mínimo debido al retardo introducido por los filtros empleados en la cadena de procesamiento. Finalmente, se realiza la estimación de velocidad radial mediante la obtención de la diferencia de fase presente en las salidas del formador de haces. Esta técnica brindó estimaciones de la velocidad radial cercanas al valor real para aquellos haces que no se encontrasen muy atenuados. En este contexto, se resalta la importancia de los algoritmos de umbralización dado que permiten al sistema discernir si se cuenta con un objeto en cierta dirección o si se cuenta únicamente con ruido.

A partir de este modelo, se desarrolló un núcleo de procesamiento que permitiese instrumentar el módulo de compresión de pulso en un FPGA. Dada la gran cantidad de muestras generadas para cada intervalo de repetición de pulso, la arquitectura planteada contempla la partición de los datos en lotes de muestras de menor tamaño, denominados compuertas de alcance, los cuales representan intervalos de distancia respecto al sistema de recepción. Los datos que conforman a las compuertas de alcance son procesados por el núcleo de procesamiento, el cual es capaz de operar en tiempo real.

Este núcleo se encarga de realizar la estimación de distancia mediante la aplicación de un filtro en concordancia y la identificación del valor máximo a la salida de dicho filtro, cuyo índice permite obtener la distancia al objeto. La instrumentación consta de cuatro elementos principales: un filtro en concordancia, su controlador asociado, un estimador de distancia y una serie de registros que permiten almacenar el número ordinal de las compuertas procesadas por el núcleo. Estos elementos se implementan mediante una combinación de bloques IP, descripciones funcionales y diseños estructurales dedicados, priorizando un balance entre latencia y utilización de recursos.

La implementación de dicha instrumentación se realizó en una tarjeta de desarrollo Nexys Video y permitió estimar la distancia al objeto con resultados equiparables a los obtenidos por el modelo de procesamiento. Aunado a esto, la cantidad de recursos utilizados por el núcleo se mantuvo dentro de un margen razonable, siendo el filtro en concordancia el elemento más demandante en cuanto a recursos. Por consiguiente, la arquitectura propuesta cumple con los objetivos planteados al brindar un núcleo de procesamiento replicable y escalable que permite llevar a cabo la compresión de pulso y la subsecuente estimación de distancia.

No obstante, existen diversas áreas de oportunidad dentro de los modelos y la instrumentación desarrollada a partir de las cuales se puede plantear el trabajo a futuro. En primer lugar, la sección del modelo encargada de generar las señales de RF recibidas por el arreglo

no toma en cuenta factores como acoplamiento mutuo que ocurre en elementos contiguos del arreglo o la aparición de ruido. Aunado a esto, la sección de acondicionamiento de señales utiliza una sola etapa de conversión hacia IF, lo cual es poco factible en una arquitectura real. Por consiguiente, como trabajo a futuro, se puede actualizar dicho modelo para que se tome en cuenta los factores antes mencionados, al igual que reemplazar la etapa única de conversión por un modelo que permita simular dos o más etapas de conversión hacia IF.

De igual forma, tanto el modelo como la instrumentación se beneficiarían de incorporar algún mecanismo que permita evaluar si se cuenta realmente con un objeto dentro de cierto haz. Como se mencionó anteriormente, esto puede lograrse mediante la inclusión de algoritmos de umbralización como CFAR, los cuales permiten determinar de forma dinámica la probabilidad de detección de un objeto.

Por otro lado, en cuanto a la instrumentación, se puede desarrollar la implementación completa del subsistema de compuertas de alcance del haz, cuyo diseño se planteó únicamente a nivel sistema. Del mismo modo, se puede explorar la posibilidad de reemplazar los bloques IP utilizados en el filtro en concordancia por diseño propios con una funcionalidad más a la medida. Finalmente, se puede plantear un análisis más minucioso de la latencia para verificar si realmente es posible cumplir con las restricciones de tiempo real mediante el diseño propuesto.

Anexos

Los *scripts* de MATLAB empleados a lo largo de este trabajo escrito pueden ser consultados en https://github.com/luis-r18/radar_dsp_model.git.

Referencias

- [1] R. J. Purdy, P. E. Blankenship, C. E. Muehe, C. M. Rader, E. Stern y R. C. Williamson, «Radar Signal Processing,» *Lincoln Laboratory Journal*, vol. 12, n.º 10, págs. 297-320, 2000. dirección: https://archive.ll.mit.edu/publications/journal/pdf/vol12_no2/12_2radarsignalprocessing.pdf.
- [2] W. Higgins, B. Holbrook y J. Emling, «Electrical Computers for Fire Control,» *Annals of the History of Computing*, vol. 4, n.º 3, págs. 218-244, 1982. DOI: 10.1109/MAHC.1982.10026.
- [3] L. Brown, «Flyable TRADIC: the first airborne transistorized digital computer,» *IEEE Annals of the History of Computing*, vol. 21, n.º 4, págs. 55-61, 1999. DOI: 10.1109/85.801533.
- [4] «SAGE: Semi-Automatic Ground Environment Air Defense System,» Massachusetts Institute of Technology, Lincoln Laboratory. dirección: <https://www.ll.mit.edu/about/history/sage-semi-automatic-ground-environment-air-defense-system#>.
- [5] «SAGE,» International Business Machines Corporation. dirección: <https://www.ibm.com/history/sage>.
- [6] «SAGE/AN-FSQ-7 plug-in logic module,» Computer History Museum. dirección: <https://www.computerhistory.org/collections/catalog/102801376>.
- [7] D. Jenn, *Radar Fundamentals*, Naval Postgraduate School. dirección: <https://faculty.nps.edu/jenn/Seminars/RadarFundamentals.pdf>.
- [8] W. A. Ogletree, H. W. Taylor, E. W. Veitch y J. Wylen, «AN/FST-2 radar-processing equipment for SAGE,» en *Papers and Discussions Presented at the December 9-13, 1957, Eastern Joint Computer Conference: Computers with Deadlines to Meet*, ép. IRE-ACM-AIEE '57 (Eastern), Washington, D.C.: Association for Computing Machinery, 1957, págs. 156-160. DOI: 10.1145/1457720.1457748.
- [9] R. M. O'Donnell, *Clutter Rejection: MTI and Pulse Doppler Processing*, Massachusetts Institute of Technology, Lincoln Laboratory, jun. de 2002. dirección: <https://www.ll.mit.edu/sites/default/files/outreach/doc/2018-07/lecture%208.pdf>.
- [10] L. Cartledge y R. M. O'Donnell, «Description and Performance Evaluation of the Moving Target Detector,» inf. téc. FAA-RD-76-190, mar. de 1977. dirección: https://www.ll.mit.edu/sites/default/files/publication/doc/2018-12/Cartledge_1977_ATC-69_WW-15318.pdf.
- [11] D. Karp y J. R. Anderson, «Moving Target Detector (Mod II) Summary Report,» inf. téc. FAA-RD-80-77, nov. de 1981. dirección: https://www.ll.mit.edu/sites/default/files/publication/doc/2018-12/Karp_1981_ATC-95_WW-15318.pdf.

- [12] S. Rabinowitz, C. Gager, E. Brookner, C. Muehe y C. Johnson, «Applications of digital technology to radar,» *Proceedings of the IEEE*, vol. 73, n.º 2, págs. 325-339, 1985. DOI: 10.1109/PROC.1985.13143.
- [13] J. Craig y B. McTaggart, «MIMO and Beamforming,» en *Software Defined Radio with Zynq® UltraScale+ RFSoc*, L. Crockett, D. Northcote y R. W. Stewart, eds., 1.ª ed. Glasgow, Scotland, United Kingdom: Strathclyde Academic Media, 2023, cap. 18, págs. 593-627.
- [14] L. Crockett, «Software Defined Radio,» en *Software Defined Radio with Zynq® UltraScale+ RFSoc*, L. Crockett, D. Northcote y R. W. Stewart, eds., 1.ª ed. Glasgow, Scotland, United Kingdom: Strathclyde Academic Media, 2023, cap. 2, págs. 13-181.
- [15] A. J. Fenn, D. H. Temme, W. P. Delaney y W. E. Courtney, «The Development of Phased-Array Radar Technology,» *Lincoln Laboratory Journal*, vol. 12, n.º 2, págs. 321-340, 2000. dirección: <https://www.ll.mit.edu/sites/default/files/publication/doc/development-phased-array-radar-technology-fenn-ja-7838.pdf>.
- [16] T. A. Healy y R. L. Ferranti, «Airport Surveillance Radar (ASR) Solid-State Transmitter Demonstration Final Report,» inf. téc. DOT/FAA/CT-TN94/11, sep. de 1994. dirección: <https://apps.dtic.mil/sti/pdfs/ADA289261.pdf>.
- [17] R. L. Ferranti, «Solid State Radar Demonstration Test Results at the FAA Technical Center,» inf. téc. ATC-221, jun. de 1994. dirección: https://www.ll.mit.edu/sites/default/files/publication/doc/2018-12/Ferranti_1994_ATC-221_WW-15318.pdf.
- [18] S. Waqar, M. Muaaz, S. Sigg y M. Pätzold, «A Paradigm Shift From an Experimental-Based to a Simulation-Based Framework Using Motion-Capture Driven MIMO Radar Data Synthesis,» *IEEE Sensors Journal*, vol. 24, n.º 10, págs. 16 614-16 628, 2024. DOI: 10.1109/JSEN.2024.3386221.
- [19] «IEEE Standard Letter Designations for Radar-Frequency Bands,» *IEEE Std 521-2019 (Revision of IEEE Std 521-2002)*, págs. 1-15, 2020. DOI: 10.1109/IEEESTD.2020.8999849.
- [20] J. A. Scheer y W. A. Holm, «Introduction and Radar Overview,» en *Principles of Modern Radar, Basic Principles*, M. A. Richards, J. A. Scheer y W. A. Holm, eds., 1.ª ed., 3 vols. USA: SciTech Publishing, 2010, vol. 1, cap. 1, págs. 18-54.
- [21] *Nike Radars and Computer, Lesson 4. Track Radar Systems*, Missile and Munitions Center and School, U.S. Army, nov. de 1973. dirección: <https://ed-thelen.org/MMS-150-Ch04.pdf>.
- [22] C. Pearson, «High Speed, Digital to Analog Converters Basics,» inf. téc. SLAA523A, oct. de 2012. dirección: <https://www.ti.com/lit/an/slaa523a/slaa523a.pdf>.
- [23] «RF Power Amplifiers: An Overview,» Cadence PCB Solutions. dirección: <https://resources.pcb.cadence.com/blog/2023-rf-power-amplifiers-an-overview>.
- [24] F. Marki y C. Marki, *Mixer Basics Primer*, Marki Microwave, 2010. dirección: https://markimicrowave.com/assets/c2c4688b-15c7-4421-a703-254cb238f9fb/Mixer_Basics_Primer.pdf.
- [25] *RF-to-Bits Solution Offers Precise Phase and Magnitude Data to 6 GHz*, CN-0374, Analog Devices, Norwood, MA, USA, 2015, 7 págs. dirección: <http://analog.com/media/en/reference-design-documentation/reference-designs/cn0374.pdf>.

- [26] B. Razavi, «Transceiver Architectures,» en *RF Microelectronics*, 2.^a ed. USA: Prentice Hall, 2012, cap. 4, págs. 160-226.
- [27] A. Srivastav, P. Nguyen, M. McConnell, K. A. Loparo y S. Mandal, «A Highly Digital Multiantenna Ground-Penetrating Radar (GPR) System,» *IEEE Transactions on Instrumentation and Measurement*, vol. 69, n.º 10, págs. 7422-7436, 2020. DOI: 10.1109/TIM.2020.2984415.
- [28] G. T. Markov y D. M. Sazonov, «Fundamentos electrodinámicos de la teoría de las antenas,» en *Antenas*, A. Grdian, ed. Moscú, Rusia: URSS, 1994, cap. 1, págs. 17-26, Traducido del ruso al español.
- [29] R. Knight, «Traveling Waves,» en *Physics for Scientists and Engineers*. Pearson Education, Limited, 2012, cap. 16, págs. 441-442.
- [30] C. D. Bailey, «Radar Antennas,» en *Principles of Modern Radar, Basic Principles*, M. A. Richards, J. A. Scheer y W. A. Holm, eds., 1.^a ed., 3 vols. USA: SciTech Publishing, 2010, vol. 1, cap. 9, págs. 310-314.
- [31] W. Geyi, «Radiation in Free Space (I),» en *Foundations of Antenna Radiation Theory*. John Wiley & Sons, Ltd, 2023, cap. 4, págs. 151-242. DOI: <https://doi.org/10.1002/9781394170951.ch4>. dirección: <https://onlinelibrary.wiley.com/doi/abs/10.1002/9781394170951.ch4>.
- [32] C. Balanis, *Antenna Theory: Analysis and Design*. Wiley, 2015. dirección: <https://books.google.com.mx/books?id=u-xbCwAAQBAJ>.
- [33] G. T. Markov y D. M. Sazonov, «Análisis de los sistemas radiadores lineales,» en *Antenas*, A. Grdian, ed. Moscú, Rusia: URSS, 1994, cap. 6, págs. 178-228, Traducido del ruso al español.
- [34] R. M. O'Donnell, *Lecture 11: Waveforms and Pulse Compression*, IEEE New Hampshire Section, IEEE AES Society, ene. de 2010. dirección: http://www.radar-course.org/Radar%202010%20PDFs/Radar%202009%20A_11%20Waveforms%20and%20Pulse%20Compression.pdf.
- [35] B. M. Keel, «Fundamentals of Pulse Compression Waveforms,» en *Principles of Modern Radar, Basic Principles*, M. A. Richards, J. A. Scheer y W. A. Holm, eds., 1.^a ed., 3 vols. USA: SciTech Publishing, 2010, vol. 1, cap. 20, págs. 773-808.
- [36] B. M. Keel, «Constant False Alarm Rate Detectors,» en *Principles of Modern Radar, Basic Principles*, M. A. Richards, J. A. Scheer y W. A. Holm, eds., 1.^a ed., 3 vols. USA: SciTech Publishing, 2010, vol. 1, cap. 16, págs. 589-620.
- [37] «The Doppler Effect,» University of New South Wales, School of Physics. dirección: <https://www.animations.physics.unsw.edu.au/jw/doppler.htm>.
- [38] «FAST FOURIER TRANSFORMS,» en *Mixed Signal and DSP Design Techniques*, W. Kester, ed. Newnes/Elsevier, 2002, cap. 5. dirección: https://www.analog.com/media/en/training-seminars/design-handbooks/MixedSignal_Sect5.pdf.
- [39] A. V. Oppenheim y R. W. Schaffer, *Discrete-Time Signal Processing*, 3.^a ed. USA: Prentice Hall Press, 2009.
- [40] A. Oppenheim, A. Willsky y S. Nawab, *Signals & Systems*, 2.^a ed. USA: Prentice-Hall, Inc., 1997.
- [41] J. Proakis y D. Manolakis, «Efficient Computation of the DFT: FFT Algorithms,» en *Digital Signal Processing*, 4.^a ed. USA: Prentice-Hall, Inc., 2006, cap. 8, págs. 511-537.

- [42] J. W. Cooley y J. W. Tukey, «An Algorithm for the Machine Calculation of Complex Fourier Series,» *American Mathematical Society*, vol. 19, n.º 90, págs. 297-301, 1965.
- [43] M. Doggett, *Fixed-point Mathematics*, Lund University of Technology, Faculty of Engineering, 2009. dirección: https://fileadmin.cs.lth.se/cs/Education/EDA075/notes/mgh_appA_fixed.pdf.
- [44] «TMS320C64x DSP Library Programmer's Reference,» inf. téc. SPRU565B, oct. de 2003. dirección: <https://www.ti.com/lit/ug/spru565b/spru565b.pdf>.
- [45] K. Røed, *VHDL – Historical view*, University of Oslo, 2021. dirección: https://www.uio.no/studier/emner/matnat/fys/FYS4220/h21/lecture-slides/vhdl_history.pdf.
- [46] N. Weste y D. Harris, *CMOS VLSI Design: A Circuits and Systems Perspective*, 4.ª ed. USA: Addison-Wesley Publishing Company, 2010.
- [47] P. J. Ashenden, *VHDL Tutorial*, Ashenden Designs Pty. Ltd., 2004. dirección: https://www.eecs.umich.edu/courses/doing_dsp/handout/vhdl-tutorial.pdf.
- [48] «What Is an IP Core?» MathWorks. dirección: <https://mathworks.com/discovery/ip-core.html>.
- [49] K. T. Ulrich y S. D. Eppinger, *Product Design and Development*, 6.ª ed. New York, NY, USA: McGraw-Hill, 2016.
- [50] R. M. O'Donnell, *Introduction to Radar Systems: The Radar Equation*, Massachusetts Institute of Technology, Lincoln Laboratory, jun. de 2002. dirección: <https://www.ll.mit.edu/sites/default/files/outreach/doc/2018-07/lecture%202.pdf>.
- [51] W. Kester, «Which ADC Architecture Is Right for Your Application?» *Analog Dialogue*, vol. 39, n.º 2, págs. 1-8, 2005. dirección: <https://www.analog.com/media/en/analog-dialogue/volume-39/number-2/articles/the-right-adc-architecture.pdf>.
- [52] M. Rives, *Intermediate Frequency (IF) Sampling Receiver Concepts*, Texas Instruments, 2011. dirección: <https://www.ti.com/lit/wp/snaa107/snaa107.pdf>.
- [53] *High performance aerospace and defense solutions*, NXP Semiconductors, 2010. dirección: <https://www.mouser.com/ds/2/302/75017008-99829.pdf>.
- [54] *MM1-0212LS Connectorized Module*, Marki Microwave, 2018, 14 págs. dirección: <https://markimicrowave.com/assets/f09ab33c-fbfa-4749-aa16-3a5271db8631/MM1-0212LS-Connectorized%20Module.pdf>.
- [55] *AD9268: 16-Bit, 80 MSPS/105 MSPS/125 MSPS, 1.8 V Dual Analog-to-Digital Converter (ADC) Data Sheet (Rev. A)*, Analog Devices, Norwood, MA, USA, 2009, 44 págs. dirección: <https://www.analog.com/media/en/technical-documentation/data-sheets/AD9268.pdf>.
- [56] M. Yang, J. Yang, Y. Hou y C. Jin, «Implementation architecture of signal processing in pulse Doppler radar system based on FPGA,» *The Journal of Engineering*, vol. 2019, págs. 7335-7338, 21 2019. DOI: 10.1049/joe.2019.0644. dirección: <https://digital-library.theiet.org/doi/abs/10.1049/joe.2019.0644>.
- [57] M. J. Bonato, «A Comparison of Two Computational Technologies for Digital Pulse Compression,» en *High Performance Embedded Computing Conference 2002*, Lexington, MA, USA. dirección: <https://apps.dtic.mil/sti/pdfs/ADA419798.pdf>.

- [58] *About Lookup Table Blocks*, The MathWorks, Inc., 2025. dirección: <https://www.mathworks.com/help/simulink/ug/about-lookup-table-blocks.html>.
- [59] *Methods for Approximating Function Values*, The MathWorks, Inc., 2025. dirección: <https://www.mathworks.com/help/simulink/ug/methods-for-estimating-missing-points.html>.
- [60] N. Bowman y K. Jue, *Big-O Notation and Algorithmic Analysis*, Stanford University, 2021. dirección: <https://web.stanford.edu/class/archive/cs/cs106b/cs106b.1218/lectures/07-bigo/Lecture7Slides.pdf>.
- [61] I. Selesnick, *Fast Convolution Algorithms*, New York University, Tandon School of Engineering, 2025. dirección: <https://eeweb.engineering.nyu.edu/iselesni/EL713/zoom/overlap>.
- [62] *Fast Fourier Transform v9.1 LogiCORE IP Product Guide*, 9.^a ed., Xilinx, 2022. dirección: <https://docs.amd.com/r/en-US/pg109-xfft>.
- [63] M. M. Mano y M. D. Ciletti, «Combinational Logic,» en *Digital Design: With an Introduction to the Verilog HDL*, 5.^a ed. USA: Prentice Hall, 2013, cap. 4, pág. 137.
- [64] *7 Series FPGAs Configurable Logic Block*, Xilinx, 2025, 81 págs. dirección: https://docs.amd.com/r/en-US/ug474_7Series_CLB/.
- [65] *Nexys Video™ FPGA Board Reference Manual*, Digilent, 2015, 28 págs. dirección: https://digilent.com/reference/_media/nexys-video:nexys_video_rm.pdf.
- [66] *Artix-7 FPGAs Data Sheet: DC and AC Switching Characteristics*, Xilinx, 2018, 64 págs. dirección: <https://www.mouser.com/pdfDocs/DCandACSwitchingCharacteristics.pdf>.
- [67] *DA20i KATANA FACTS AND SPECIFICATIONS*, DIAMOND AIRCRAFT INDUSTRIES GMBH, 2025. dirección: https://www.diamondaircraft.com/fileadmin/diamondaircraft/documents/da20/DA20i_Katana_Factsheet_SCREEN.pdf.
- [68] *PC-7 MKX Factsheet*, Pilatus Aircraft Ltd, 2025. dirección: <https://www.pilatus-aircraft.com/assets/files/Brochures/PC-7-MKX/PC-7-MKX-Factsheet.pdf>.
- [69] *Total Harmonic Distortion Measurements*, Anritsu, 2023. dirección: <https://dl.cdn-anritsu.com/en-us/test-measurement/files/Technical-Notes/White-Paper/11410-02911A.pdf>.