



**Universidad**  
Zaragoza

## Trabajo Fin de Grado

Diseño de un bloque IP (AXI-4 compatible) en VHDL aplicado a la interpolación de señal de corriente en una etapa de potencia

VHDL IP block design (AXI-4 compatible) applied to power stage current signal interpolation

Autor

Alberto Mur López

Directores

Isidro Urriza Parroqué





# DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D<sup>a</sup>. ALBERTO MUR LOPEZ,

con nº de DNI 18046260T en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster)  
Grado \_\_\_\_\_, (Título del Trabajo)

Diseño de un bloque IP (AXI-4 compatible) en VHDL aplicado a la  
interpolación de señal de corriente en una etapa de potencia

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 31 de Enero de 2019

Fdo: ALBERTO MUR LOPEZ





# AGRADECIMIENTOS

Agradezco a todos los que me apoyaron para embarcarme en esta aventura. A mi familia y amigos que siempre han velado porque saliera adelante y no han permitido que me rindiera. A todos los que han invertido algo de su tiempo en mí, a todos los que me han enseñado a ser un mejor profesional y mejor persona. A mis compañeros con los que he atravesado este duro camino. En especial a Isidro, por sus consejos, tiempo y experiencia para sacar adelante este proyecto. Gracias a todos.



# **Diseño de un bloque IP (AXI-4 compatible) en VHDL aplicado a la interpolación de señal de corriente en una etapa de potencia**

## **RESUMEN**

En este trabajo se lleva a cabo la caracterización e interpolación de una señal de intensidad en la carga de una etapa de potencia con configuración de inversor resonante serie en semi-puente. Aunque el objetivo específico de este trabajo es emplear la señal para distintos procesos de control y seguridad de un sistema de inducción doméstico, el proceso desarrollado es equivalente para cualquier aplicación en la que se requiera aumentar la resolución de una señal. Se han establecido los parámetros de diseño tras analizar las señales de muestra extraídas directamente de una plataforma experimental. Además se imponen unas medidas de calidad mínimas en determinados parámetros que debe cumplir la señal reconstruida. La interpolación de la señal se realiza mediante sobremuestreo y filtrado paso bajo. Durante la etapa de diseño se valoran diferentes configuraciones y tipos de filtro, comparando la calidad de reconstrucción y coste computacional de los mismos. Finalmente se propone una implementación en VHDL sintetizable del diseño seleccionado.

# **VHDL IP block design (AXI-4 compatible) applied to power stage current signal interpolation**

## **ABSTRACT**

A characterization and interpolation applied to a power stage current signal with half-bridge resonant inverter configuration are performed. Although this work aims to use the signal for different control and security mechanisms of a domestic induction system, this method is equivalent for any application requiring better signal resolution. Design specifications are established after analyzing sample signals extracted from an experimental platform. Also, minimum quality measures on specific parameters are imposed on the reconstructed signal. Signal interpolation is achieved by upsampling and low pass filtering. Through the design section, a reconstruction quality and computational cost comparison between different filters and configuration is carried out. Finally, a synthesizable VHDL implementation of the selected design is proposed.

# Índice

<b>1. Introducción</b>	<b>1</b>
1.1. Interpolación . . . . .	2
1.2. Etapa de potencia . . . . .	3
1.3. Planteamiento del trabajo . . . . .	5
1.4. Alcance y objetivos . . . . .	6
1.5. Estructura de la memoria . . . . .	7
<b>2. Caracterización de la señal</b>	<b>8</b>
<b>3. Diseño de la etapa de interpolación</b>	<b>12</b>
3.1. Escenario primero . . . . .	13
3.1.1. Polinomio de Lagrange . . . . .	14
3.1.2. Filtro FIR óptimo . . . . .	17
3.2. Escenario segundo . . . . .	18
3.2.1. Polinomio de Lagrange . . . . .	19
3.2.2. Filtro FIR óptimo . . . . .	22
3.2.3. Filtro <i>Half-band</i> . . . . .	23
3.3. Escenario tercero . . . . .	27

3.3.1.	Polinomio de Lagrange . . . . .	28
3.3.2.	Filtro FIR óptimo . . . . .	31
3.3.3.	Filtro <i>Half-band</i> . . . . .	33
3.4.	Resultados . . . . .	36
<b>4.</b>	<b>Diseño del sistema digital, implementación en VHDL</b>	<b>38</b>
4.1.	Especificaciones . . . . .	38
4.2.	Arquitectura del sistema . . . . .	38
4.3.	Protocolo de comunicación . . . . .	40
4.4.	Implementación del bloque de filtrado . . . . .	41
<b>5.</b>	<b>Simulación y validación</b>	<b>51</b>
5.1.	Implementación en MATLAB . . . . .	51
5.1.1.	Validación de la interpolación . . . . .	51
5.2.	Implementación en VHDL . . . . .	52
5.2.1.	Simulación . . . . .	52
5.2.2.	Validación del protocolo de comunicación . . . . .	52
5.2.3.	Validación de la interpolación . . . . .	54
5.2.4.	Validación de la implementación . . . . .	55
<b>6.</b>	<b>Conclusiones</b>	<b>56</b>
6.1.	Lineas futuras . . . . .	57
	<b>Bibliografía</b>	<b>58</b>

---

<b>Lista de Figuras</b>	<b>60</b>
<b>Lista de Tablas</b>	<b>63</b>
<b>Glosario</b>	<b>65</b>
<b>Lista de símbolos</b>	<b>66</b>
<b>Anexos</b>	<b>67</b>
<b>A. Cálculo de parámetros temporales</b>	<b>68</b>
<b>B. Filtros de interpolación</b>	<b>73</b>
B.1. Filtros FIR . . . . .	73
B.1.1. Lagrange . . . . .	73
B.1.2. Filtros FIR óptimos . . . . .	74
B.1.3. Filtros Half-Band . . . . .	75
<b>C. Modificación del sistema digital</b>	<b>77</b>
C.1. Archivo de configuración . . . . .	77
C.1.1. Constantes . . . . .	77
C.1.2. Tipos de dato . . . . .	77
C.1.3. Configuración de los filtros . . . . .	78
C.2. Modificación del número de etapas . . . . .	78
C.3. Incremento del número de coeficientes en filtros <i>Half-band</i> . . . . .	80

# Capítulo 1

## Introducción

Existen muchas situaciones en las que se observan procesos o fenómenos a ciertos intervalos temporales ya que puede ser difícil medirlos y registrarlos de manera continua. Sin embargo, para determinados objetivos puede requerirse conocer el estado o valor del proceso en un instante determinado que se encuentra entre el intervalo de observación. Sir Edmund Whittaker, un matemático profesor de la Universidad de Edinburgo, opinaba que “la forma más común de interpolación ocurre cuando buscamos datos de una tabla que no tiene los valores exactos que necesitamos”.

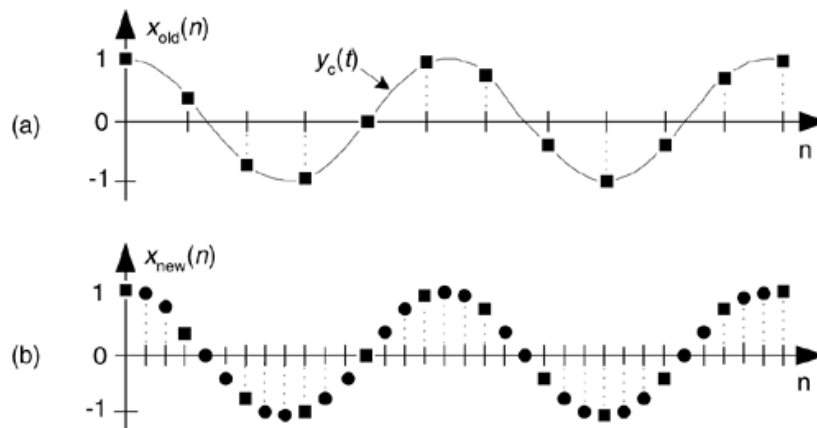
El problema de interpolación se conoce desde la antigua Babilonia pero no es hasta la era de la información y la comunicación donde se aborda este problema desde una perspectiva del procesado de señal [1]. Es en 1973 cuando Schafer y Rabiner estudian el proceso de interpolación como un problema de filtrado digital, esto es, la interpolación basada en convolución[2]. En dicho estudio, los autores determinan que los filtros FIR (respuesta finita al impulso) de fase lineal tienen características muy atractivas para la interpolación discreta y muestran como se pueden utilizar de forma eficiente.

A continuación se introducirá el concepto de interpolación y el contexto del origen de la señal que se quiere interpolar.



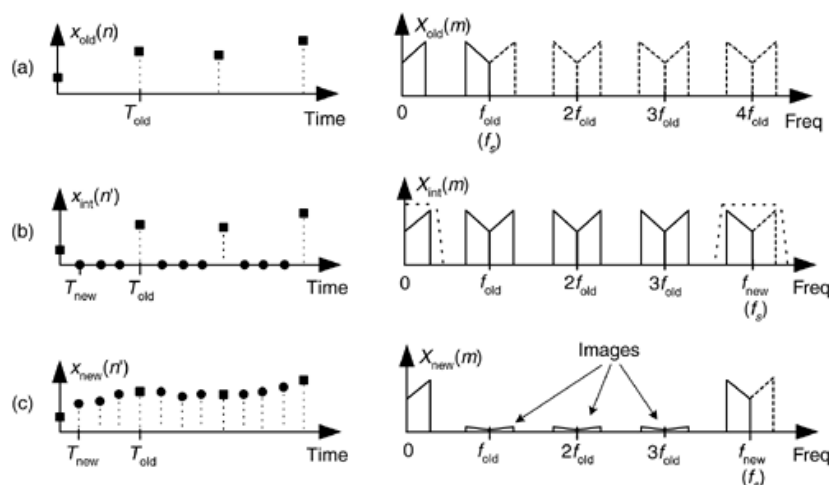
## 1.1. Interpolación

La interpolación se refiere a una clase de métodos o algoritmos para la reconstrucción de una forma de onda dados unos puntos. En este caso, se precisa aumentar la resolución temporal de la señal y, por lo tanto, generar muestras de la señal que no están disponibles (Figura 1.1.1).



**Figura 1.1.1:** Interpolación: (a) secuencia original; (b) secuencia interpolada por  $L = 3$ ; [3, p. 581].

El sobremuestreo es el proceso de insertar muestras de valor cero entre las muestras disponibles para aumentar la frecuencia de muestreo. Esta operación añade a la señal original réplicas espectrales no deseadas centradas en múltiplos de la frecuencia de muestreo original (Figura 1.1.2.b).



**Figura 1.1.2:** Interpolación por cuatro: (a) secuencia original muestreada y su espectro; (b) ceros insertados en la secuencia original y su espectro; (c) secuencia resultado del filtro interpolador y espectro resultante [3, p. 582].

La interpolación, en el sentido del procesamiento digital de señal, es el proceso de sobremuestrear seguido de un filtrado (Figura 1.1.3). Este filtrado elimina las réplicas

espectrales no deseadas (Figura 1.1.2.c). Como proceso lineal, el sentido del procesado digital de señal es algo diferente al sentido matemático de la interpolación, pero el resultado, conceptualmente, es similar: crear las muestras intermedias a partir de las muestras originales. El resultado es cómo si se hubiera muestreado la señal a una frecuencia de muestreo superior.

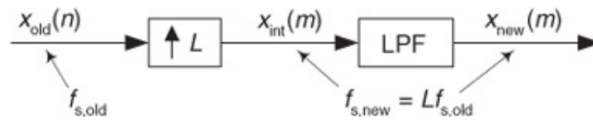


Figura 1.1.3: Notación funcional de la interpolación [3, p. 581].

## 1.2. Etapa de potencia

Los convertidores de potencia resonantes son esenciales en muchas aplicaciones: la conversión DC-DC de alta eficiencia y densidad de potencia, los inversores de alta frecuencia, el calentamiento por inducción, o los sistemas transferencia de energía sin contacto, ya que proveen una excitación sinusoidal con un elevada eficiencia de conversión en potencia. De las topologías disponibles, la topología resonante serie en semipunte es una de las mas utilizadas debido a su balance entre coste, rendimiento y complejidad de control [4].

La figura 1.2.1 representa un inversor resonante serie en semipunte. En este modelo, el conjunto inductor-recipiente, se modela mediante un circuito  $RL$  serie siendo  $C_r$  el condensador de resonancia.

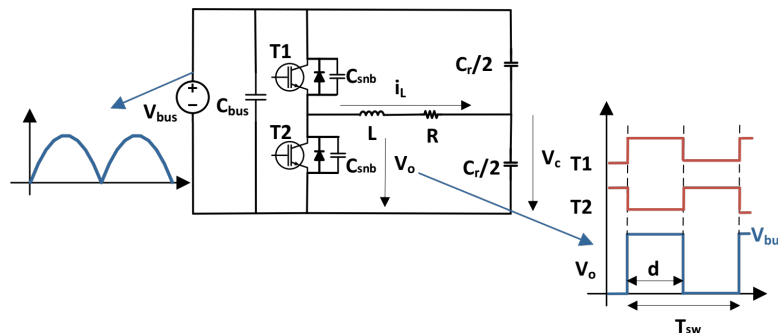
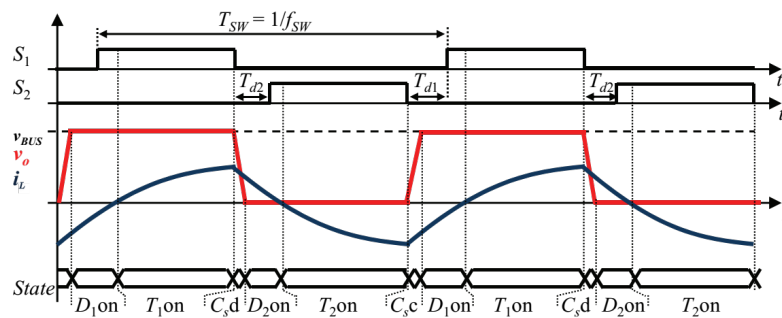


Figura 1.2.1: Inversor resonante serie en semipunte resonante con dos condensadores de resonancia [5, p. 14].

En este tipo de tecnología la variable a controlar es la potencia entregada a la carga. En las cocinas de inducción, el control se realiza variando la frecuencia de conmutación de los dispositivos ( $f_{sw}$ ) entre 35 y 75 kHz aproximadamente, con lo que se proporcionan

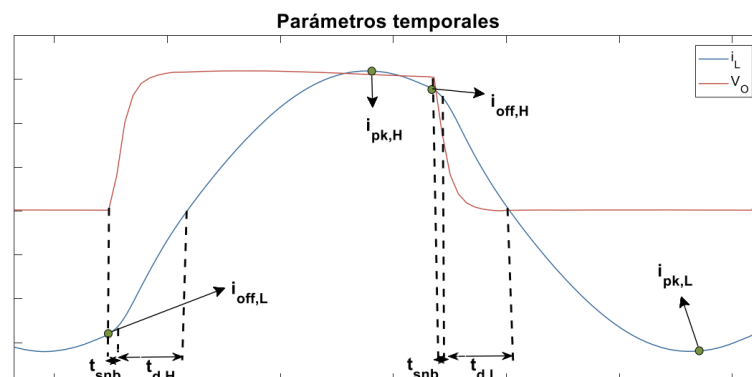
potencias ( $P_{avg}$ ) en el rango de 50 y 3700 W [5, p. 14]. Pero hay otros parámetros que se pueden modificar, como el ciclo de servicio ( $D$ ), para controlar la potencia [6].

Con el objetivo de conseguir alta eficiencia y garantizar la seguridad de los dispositivos de conmutación, el inversor opera bajo condición ZVS (*Zero Voltage Switching*). Esto implica que  $f_{sw}$  tiene que ser mayor que la frecuencia de resonancia de la etapa ( $f_0$ ) [7, p. 48].



**Figura 1.2.2:** Formas de onda del inversor resonante serie en semipunto operando bajo condición ZVS [7, p. 244].

Un parámetro muy relevante de esta etapa es la corriente que circula por la carga ( $i_L$ ). A partir de ella se extraen algunos parámetros de interés de las cocinas de inducción, como los tiempos de conducción de los diodos ( $t_D$ ), los tiempos de snubber ( $t_{snb}$ ) o el valor de las corrientes de pico ( $i_{Lpeak}$ ) entre otros. En la figura 1.2.3 se muestran dichos parámetros para un ciclo de conmutación. Para su cálculo, será necesario conocer también la tensión de salida de la etapa ( $v_o$ ) y la tensión de bus rectificadora ( $v_{bus}$ ) las cuales están representadas en la figura 1.2.2. El cálculo de estos parámetros es importante para el control la etapa, lo cual hace necesario conocer con precisión los valores que toma la señal en cada ventana de  $v_{bus}$  de 10 ms de duración (Ver anexo A).



**Figura 1.2.3:** Parámetros temporales relevantes para las cocinas de inducción relativos a la corriente [5, p. 26].

### 1.3. Planteamiento del trabajo

En el apartado anterior se ha introducido el origen de las señales con las que se va a trabajar. De la plataforma experimental de laboratorio fruto de un trabajo fin de máster anterior [8] se obtienen las señales de intensidad en la carga ( $i_L$ ), tensión de salida ( $v_o$ ) y tensión de bus ( $v_{bus}$ ) con un osciloscopio a una frecuencia de muestreo de 50 MHz. Estas medidas corresponden a un recipiente de 21 cm en el que se ha variado  $f_{sw}$  entre 35 y 75 kHz y se han registradas en diferentes ficheros de MATLAB.

El sistema cuenta con un ADC (conversor analógico digital) de tipo LTC2315 de Linear de 12 bits y frecuencia de muestreo máxima ( $f_{s,ADC,max}$ ) igual a 5 MHz[9], configurado a  $f_{s,ADC,1}$  igual a 2.77 MHz [8], que muestrea dicha señal [5, p. 31].

Se pide diseñar e implementar un bloque IP (bloque de propiedad intelectual) que interpole la señal a una frecuencia de muestreo ( $f_{s,dst}$ ) de 16.67 MHz, esto es, con un factor de interpolación (L) igual a 6, y sea compatible con el bloque IP instalado en la actualidad. Para ello, se estudian diversos procedimientos de procesado digital de señal y otros escenarios más económicos.

La primera tarea es caracterizar la señal de intensidad en la carga obteniendo su espectro y extrayendo unos determinados parámetros de calidad, para lo que se desarrolla un banco de pruebas en MATLAB que facilita la verificación y comparación de los métodos evaluados. El proceso de interpolación se realiza mediante inserción de ceros y eliminación de las réplicas espectrales producidas por el sobremuestreo mediante filtros FIR (respuesta finita al impulso). Este tipo de filtros se prefiere debido a su estabilidad y, sobre todo, por su respuesta lineal en fase en la banda de paso. Se estudia el rendimiento de filtros interpoladores con los coeficientes del polinomio de Lagrange, filtros FIR óptimos y filtros de media banda o *Half-band*.

Como se ha comentado, además de una interpolación con factor L igual a 6, se van a estudiar dos casos adicionales en los que se reduce la frecuencia de muestreo del ADC (conversor analógico digital). El segundo caso con  $f_{s,ADC,2}$  igual a 2.08 MHz ( $\frac{100}{48}$  MHz) y L igual a 8. El tercero y último con  $f_{s,ADC,3}$  igual a 1.04 MHz ( $\frac{100}{96}$  MHz) y L igual a 16.

Para la obtención de una señal de referencia con la que comparar la señal interpolada, se realiza un diezmado de la señal de osciloscopio muestreada a 50 MHz hasta  $f_{s,dst}$  aplicando filtros *antialiasing*. Esta señal de referencia se vuelve a diezmar

hasta la frecuencia de muestreo ( $f_{s,ADC}$ ) que tendrá la señal a interpolar en el escenario correspondiente. Se valorará la reconstrucción en la frecuencia de conmutación más alta y se verificará que se mantienen o mejoran esos parámetros en la más baja. De esta manera quedará acotado el comportamiento en las frecuencias intermedias.

Como medidas de calidad se utilizan los parámetros temporales definidos en la sección anterior y además se tendrán en cuenta factores intrínsecos de los sistemas estudiados como son el retardo y el coste computacional. De los parámetros extraídos se presta especial atención a los tiempos de snubber ( $t_{snb}$ ) y de conducción de diodo ( $t_D$ ). Controlando estos tiempos de forma adecuada se consigue que la etapa trabaje en condición ZVS. Tras mantener una reunión con personas afines al proyecto se obtiene una cota superior de 300 ns de error para los parámetros temporales y un error del 5 % en el resto de medidas.

Se cuenta con un amplio margen de tiempo para poder obtener los parámetros de calidad. Por lo tanto, uno de los factores más decisivos a la hora de elegir el filtro será el coste computacional. Los multiplicadores son un recurso escaso y caro en silicio, por lo que el empleo de un número reducido de multiplicadores será determinante.

A continuación se implementará el procedimiento seleccionado en *hardware*. Se utilizará la herramienta *Vivado Design Suite - HLL Editions 2018.2.2* para el desarrollo y simulación y se proporcionará un diseño sintetizable en VHDL (*VHSIC Hardware Description Language*) para sustituir el bloque IP actual. Para ello se realizará una implementación en coma fija del filtro y se verificarán los resultados obtenidos en simulación.

Todo el código generado para la elaboración del trabajo y de esta memoria está disponible públicamente en <https://github.com/almu10/MultiStageHalfBandInterpolator> [10].

## 1.4. Alcance y objetivos

En este trabajo se pretende mostrar el proceso de diseño e implementación en *hardware* de un sistema de interpolación de una señal de la que se conocen sus parámetros de calidad. El objetivo es aportar una solución para la interpolación de una señal de corriente en una etapa de potencia y que esté alineada con las líneas de futuro planteadas en trabajos anteriores [5]. En este sentido se tratará de buscar una solución

fácilmente escalable o reconfigurable. Se propone, además, el estudio de otros escenarios posibles que reduzcan la frecuencia de muestreo del sistema y se pueda conseguir un abaratamiento de los equipos. Por último se pide realizar una implementación en VHDL sintetizable que sea compatible con el bloque IP actual.

## 1.5. Estructura de la memoria

En esta memoria se recogen todos los procedimientos realizados durante el desarrollo del trabajo. En el capítulo 2 se introduce la caracterización de la señal y se extraen los parámetros de calidad requeridos. En el capítulo 3 se realiza un estudio exhaustivo de diferentes métodos de interpolación en los diferentes escenarios y se comparan sus prestaciones. En el capítulo 4 se lleva a cabo la descripción e implementación de del sistema en VHDL. En el capítulo 5 se valida y se verifica la implementación del sistema diseñado. Finalmente se elabora la conclusión del trabajo.

## Capítulo 2

### Caracterización de la señal

La señal a interpolar es la corriente medida en la carga de una etapa resonante serie en semi-puente. Esta medida se ha obtenido de una plataforma experimental de laboratorio [8] mediante un osciloscopio con frecuencia de muestreo 50 MHz. Además, también se disponen de las medidas  $v_o$  y  $v_{bus}$ , todas ellas para diferentes frecuencias de conmutación.

En la figura 2.0.1 se muestran las señales de  $i_L$  y  $v_{bus}$  durante una ventana de 10 ms duración.

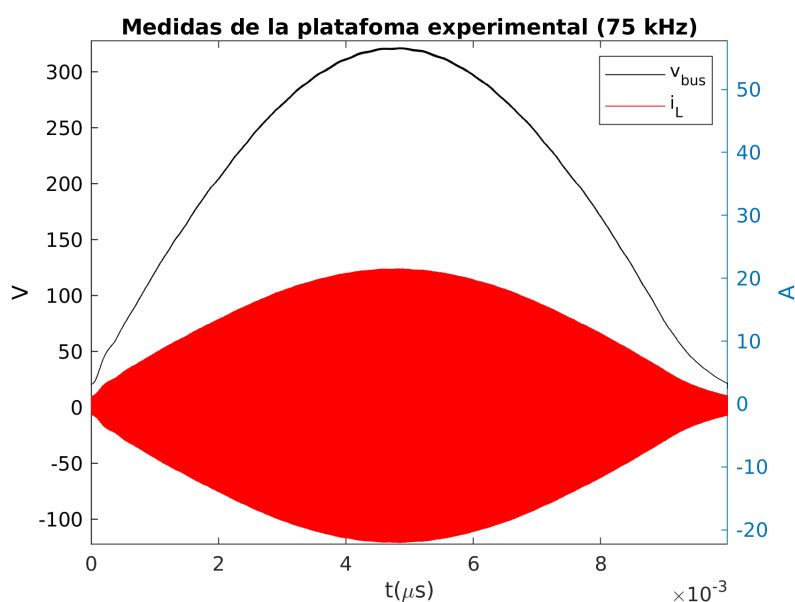
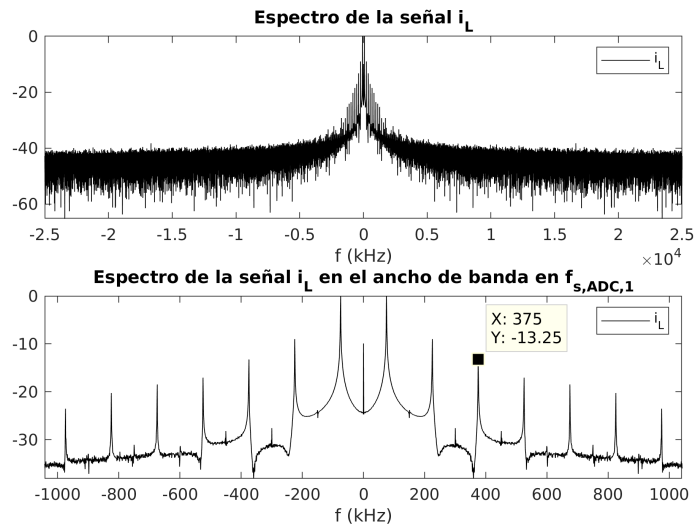


Figura 2.0.1: Señales de  $v_{bus}$  e  $i_L$  tomadas con el osciloscopio.

Se analiza el espectro frecuencial de la señal sabiendo que se utiliza información que incluye hasta el tercer armónico impar. Puesto que la máxima frecuencia de conmutación son 75 kHz, el tercer armónico impar se sitúa en 375 kHz (Figura 2.0.2).

Así, se limita el ancho de banda de señal en 750 kHz, y queda definida la mínima banda de paso.



**Figura 2.0.2:** Espectro de la señal  $i_L$  del osciloscopio.

Para obtener las medidas con las que se han trabajado, se han diezmado las señales obtenidas con el osciloscopio de 50 MHz a  $f_{s,dst}$  igual a  $\frac{100}{6}$  MHz. Para ello, primero se ha calculado el ratio de conversión  $\frac{P}{Q}$  con la función `rat`. Luego se ha hecho uso de la función `resample` de MATLAB, que aplica un filtrado *antialiasing*, con el ratio de conversión sobre  $i_L$ . Se han obtenido los coeficientes del filtro *antialiasing* para utilizarlos en las señales  $v_o$  y  $v_{bus}$ . Una vez obtenida la señal de referencia, se ha aplicado este mismo proceso de diezmado a dicha señal con frecuencia objetivo, la de muestreo del ADC que viene dada por el factor de interpolación establecido ( $L$ ) igual a 6, 8 o 16. Estos factores de interpolación hacen que las frecuencias de muestreo del ADC en los diferentes escenarios sean  $\frac{100}{36}$ ,  $\frac{100}{48}$  y  $\frac{100}{96}$  MHz respectivamente. Es necesario que la frecuencia de destino sea una número tal, que el divisor de 100 sea un número entero. Esto se debe a que el reloj de trabajo del sistema digital donde se va a implementar el diseño opera a 100 MHz. En la tabla 2.0.1 se muestra un resumen de los escenarios propuestos y sus especificaciones.

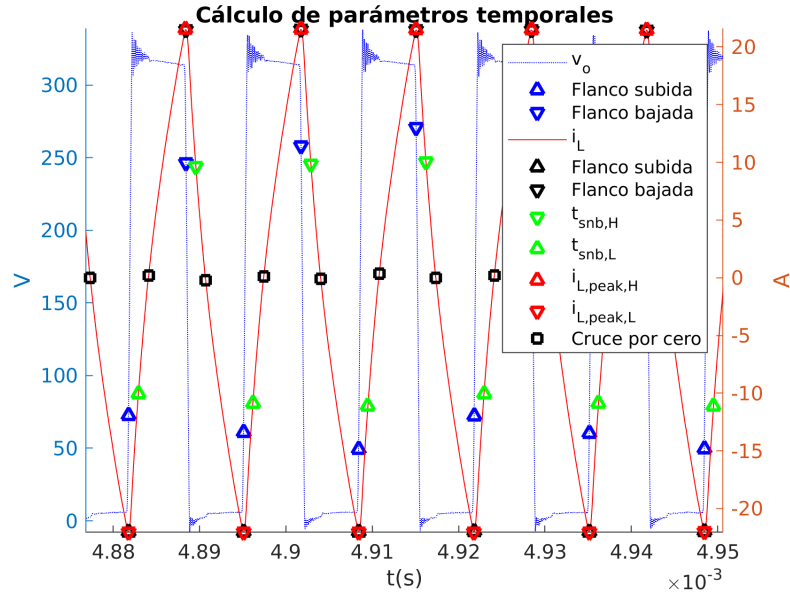
	$f_{s,ADC}$ (MHz)	$f_{s,dst}$ (MHz)	Factor de interpolación ( $L$ )
Escenario 1	$\frac{100}{36} = 2.78$	$\frac{100}{6} = 16.67$	6
Escenario 2	$\frac{100}{48} = 2.08$		$8 = 2^3$
Escenario 3	$\frac{100}{96} = 1.04$		$16 = 2^4$

**Tabla 2.0.1:** Características de los escenarios propuestos.

Para el cálculo de los parámetros temporales definidos la sección 1.2, se han



desarrollado unas funciones en MATLAB que extraen los tiempos y características relevantes. En la figura 2.0.3 se observan estos parámetros extraídos directamente de la señal con la que se ha trabajado (Ver anexo A).



**Figura 2.0.3:** Parámetros temporales extraídos de la señal de destino.

Con estas medidas se pueden obtener los errores cometidos en los parámetros para cada uno de los escenarios propuestos en el caso de no realizar la interpolación (Tabla 2.0.2). El objetivo es no superar los 300 ns de error en las medidas temporales y el 5% de error en el resto.

	Escenario 1	Escenario 2	Escenario 3	Escenario 1	Escenario 2	Escenario 3	Escenario 1	Escenario 2	Escenario 3	Escenario 1	Escenario 2	Escenario 3
	Media			$\sigma$			mín.			máx.		
$e_{it,rms}$ (%)	0.075	0.104	0.026	-	-	-	-	-	-	-	-	-
$e_{purp}$ (%)	0.158	0.207	0.108	-	-	-	-	-	-	-	-	-
$i_{pk,H}$ (%)	0.699	1.400	5.324	0.678	1.093	2.544	0.018	0.102	2.233	2.217	3.690	10.246
$i_{pk,L}$ (%)	0.794	1.348	5.158	0.663	1.031	2.477	0.017	0.019	2.160	2.049	3.474	9.660
$i_{off,H}$ (%)	1.717	2.640	10.074	1.637	2.338	6.602	0.018	0.102	2.194	4.801	7.521	23.594
$i_{off,L}$ (%)	1.500	2.789	9.828	1.370	2.368	6.399	0.017	0.019	2.160	4.465	7.388	22.809
$t_{snb,H}$ (ns)	-421.875	-586.875	-571.875	10.607	178.207	481.446	-480.000	-720.000	-1,200.000	-420.000	-180.000	-180.000
$t_{snb,L}$ (ns)	-380.625	-620.625	-680.625	28.954	28.954	487.826	-420.000	-660.000	-1,140.000	-360.000	-600.000	-120.000
$t_{d,H}$ (ns)	744.375	1,029.375	1,674.375	159.008	372.714	235.782	600.000	240.000	780.000	1,020.000	1,320.000	1,800.000
$t_{d,L}$ (ns)	675.000	1,080.000	1,800.000	180.322	169.020	379.167	480.000	600.000	1,560.000	960.000	1,200.000	2,640.000

**Tabla 2.0.2:** Comparación de los errores cometidos en los parámetros de calidad de los distintos escenarios sin realizar interpolación.

De la tabla 2.0.2 se pueden extraer varias conclusiones. El tercer escenario no cumple con ninguno de los requisitos exigidos. En los dos primeros escenarios los valores con error porcentual están dentro del objetivo, aunque el segundo escenario tiene uno de los valores con 1 décima por encima del umbral. Por otro lado, las medidas temporales están, en todos los casos, por encima de los 300 ns exigidos.

Este hecho justifica que sea necesario realizar un proceso de interpolación para tratar de minimizar el error de estos resultados.

## Capítulo 3

# Diseño de la etapa de interpolación

Se han seleccionado diferentes técnicas de filtrado para el diseño de filtros FIR. Se ha escogido el uso de filtros FIR frente a los IIR (respuesta infinita al impulso), por su la facilidad para obtener una respuesta lineal en fase en la banda de paso, y porque el error de interpolación debido a la distorsión en amplitud se puede hacer pequeña. Los IIR, por el contrario, aunque son computacionalmente más eficientes, sus características en amplitud son muy buenas, pero pueden llegar a ser inestables porque son muy sensibles a los errores de cuantización. Además, su respuesta en fase no es lineal y siempre existirá error en la interpolación. [2, p. 695].

La primera técnica que se ha empleado ha sido la interpolación mediante filtros con los coeficientes del polinomio de Lagrange de orden  $n$ , que no deja de ser interpolación polinómica. Para muestras equiespaciadas, la interpolación de Lagrange converge a la función *sinc* (filtro ideal paso bajo) cuando el número de coeficientes tiende a infinito. Si el número de coeficientes es finito, la interpolación de Lagrange es equivalente a una *sinc* enventanada utilizando una ventana binomial [11]. Para el diseño de estos filtros basta con determinar el factor de interpolación y el grado del polinomio (Ver anexo B.1.1). Se ha decidido realizar la comparación con polinomios de grado 5 y 19, por tener una idea del comportamiento de estos filtros con la señal en función del orden.

A continuación se han diseñado filtros FIR por medio del algoritmo de Parks-McClellan (Ver anexo B.1.2). Se trata de un método para el diseño e implementación de filtros FIR óptimos y eficientes. El objetivo del algoritmo es minimizar el error de las bandas de paso y rechazo utilizando una aproximación de Chebyshev. Se deben especificar las bandas de paso y rechazo y el rizado de las mismas. Se han seleccionado filtros cuyo número de coeficientes es múltiplo del

factor de sobremuestreo, para que al realizar una implementación polifase, no sea necesario rellenar con ceros los coeficientes extra que se necesitarían para completar dicha implementación.

También se ha explorado el rendimiento de los filtros CIC (Cascaded Integrator-Comb). Éste tipo de filtros son muy atractivos en el procesado digital de señal para la interpolación o diezmado de señales debido a que sus bloques únicamente realizan sumas y restas. Sin embargo, debido a la atenuación en la banda de paso que presentan este tipo de filtros, se hacía necesario el diseño de un filtro compensador previo de coste computacional similar al de los filtros FIR óptimos. Por ello, se ha desestimado el estudio de este tipo de solución.

En los dos últimos escenarios, se ha planteado un proceso de interpolación multietapa con filtros *Half-band* (Ver anexo B.1.3). Estos filtros tienen la característica de reducir el ancho de banda de la señal muestreada por un factor 2 aproximadamente, de ahí su nombre. La región de transición de estos filtros se sitúa en  $\frac{f_s}{4}$ , donde  $f_s$  es la frecuencia de muestreo. Cuando el filtro tiene un número impar de coeficientes, esta simetría tiene la propiedad de que en el dominio del tiempo, los coeficientes impares son 0, salvo el central. Además por ser un filtro de fase lineal, sus coeficientes tienen simetría respecto al centro. Estas propiedades son utilizadas para realizar una implementación muy eficiente.

La elección del diseño se ha basado en un compromiso entre el coste computacional, el retardo introducido por el filtro, el número de coeficientes y el error obtenido en los parámetros de calidad. En el coste computacional se han tenido en cuenta las MPS (multiplicaciones por muestra) y las APS (sumas por muestra).

En las secciones siguientes se exponen los resultados obtenidos en cada escenario así como una representación de la respuesta impulsional de los filtros durante el estudio.

### 3.1. Escenario primero

En el escenario principal se ha trabajado con una  $f_{s,ADC,1}$  igual a  $\frac{100}{36}$  MHz (2.78 MHz), esto es, un factor de interpolación igual a 6. Para esta frecuencia de muestreo, el espectro de la señal de  $i_L$  que se ha obtenido del ADC corresponde con el que se muestra en la figura 3.1.1. En esta figura se representan los espectros de la señal del osciloscopio, la señal diezmada a  $f_{s,dst}$  (destino) y la señal diezmada a la frecuencia del

escenario 1 (fuente), en el intervalo de  $-\frac{f_{s,ADC,1}}{2}$  a  $\frac{f_{s,ADC,1}}{2}$ .

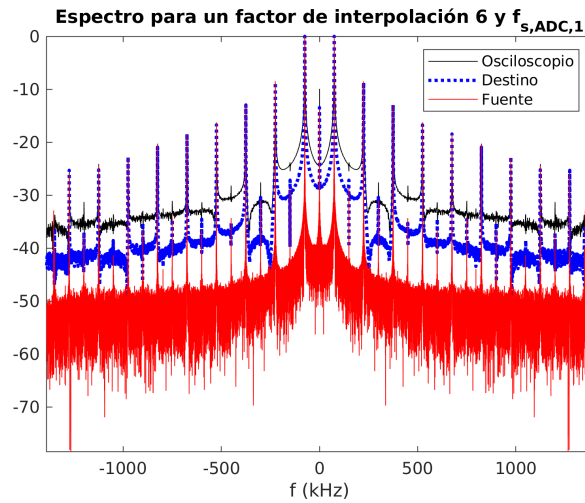


Figura 3.1.1: Espectro de la corriente para  $f_{s,ADC,1}$

En la tabla 3.1.1 se muestran los parámetros de calidad de la señal sin realizar la interpolación para poder realizar una valoración objetiva de los resultados obtenidos.

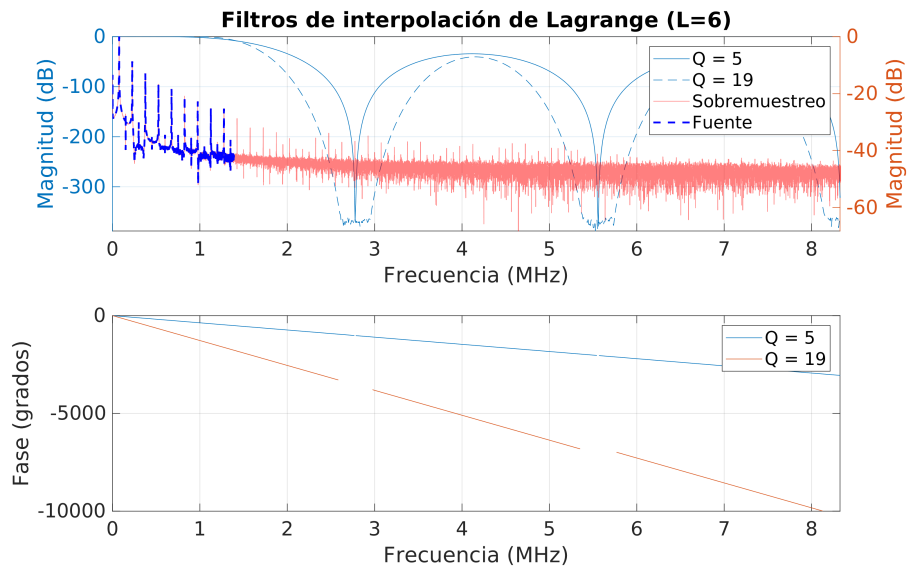
	Media		$\sigma$		min.		máx.	
	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz
$e_{i_{rms}}$	0.020	0.075	-	-	-	-	-	-
$e_{p_{avg}}$	0.044	0.158	-	-	-	-	-	-
$i_{l_{pk,H}}$	0.108	0.699	0.062	0.678	0.015	0.018	0.277	2.217
$i_{l_{pk,L}}$	0.093	0.794	0.049	0.663	0.013	0.017	0.211	2.049
$i_{off,H}$	2.306	1.717	1.144	1.637	0.057	0.018	4.055	4.801
$i_{off,L}$	2.644	1.500	1.080	1.370	0.817	0.017	4.498	4.465
$t_{snb,H}(ns)$	-315.000	-421.875	120.964	10.607	-360.000	-480.000	0.000	-420.000
$t_{snb,L}(ns)$	-339.375	-380.625	28.954	28.954	-360.000	-420.000	-300.000	-360.000
$t_{d,H}(ns)$	661.875	744.375	228.338	159.008	240.000	600.000	1020.000	1020.000
$t_{d,L}(ns)$	680.625	675.000	162.440	180.322	420.000	480.000	840.000	960.000

Tabla 3.1.1: Error sin a interpolar para  $f_{s,ADC,1}$  del recipiente Zen12 (21 cm)

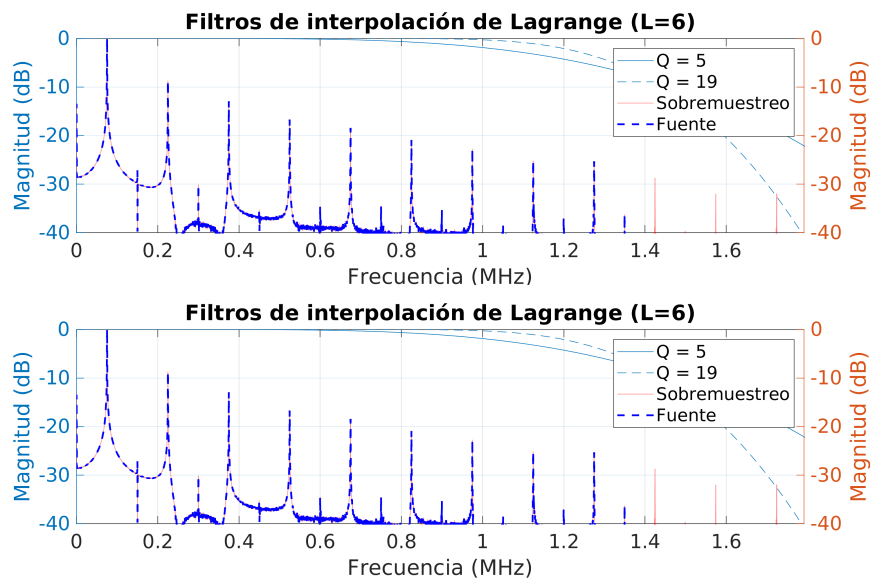
### 3.1.1. Polinomio de Lagrange

A continuación se muestra la respuesta en frecuencia de la amplitud y fase de los filtros interpoladores de Lagrange de órdenes 5 y 19. En la figura 3.1.2a se pueden observar los espectros de la señal del ADC y su correspondiente sobremuestreo con su magnitud en el eje derecho. En el eje izquierdo se representa la respuesta impulsional de ambos filtros. Se puede verificar fácilmente la fase lineal del filtro en la banda de paso. Además, en la figura 3.1.2b se puede apreciar como al aumentar el orden del filtro la banda de transición se hace más abrupta y se observa como en la banda de

paso no hay rizado. Sin embargo para el filtro de orden 5, el tercer armónico sufre cierta atenuación, esto se traduce en un mayor error en las medidas de calidad de la reconstrucción.



(a) Respuesta en fase y frecuencia de los filtros interpoladores de Lagrange del escenario 1.



(b) Detalle de respuesta en frecuencia de los filtros interpoladores de Lagrange del escenario 1.

**Figura 3.1.2:** Filtros de Lagrange del escenario 1

En la tabla 3.1.2 se detallan las características y el coste de este tipo de filtros.

Orden	Retardo	Simetría	Coefficientes	Ceros	MPS	APS
5	18 (1080ns)	Sí	35	4	5	30
19	60 (3600ns)	Sí	119	18	17	100

**Tabla 3.1.2:** Características de los filtros de Lagrange del escenario 1

## Resultados

En las tablas 3.1.3 y 3.1.4 se observa como el error cometido esta por debajo del límite establecido en todos los casos. Un punto apreciable a destacar es que al aumentar el orden, el porcentaje de error cometido en los valores de corrientes de pico y de paso a “off” tiende a disminuir. Esto puede deberse a la pequeña atenuación que sufre el tercer armónico en el caso del filtro de orden 5.

	Media		$\sigma$		min.		máx.	
	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz
$e_{i_{rms}}$	0.021	0.073	-	-	-	-	-	-
$e_{p_{avg}}$	0.021	0.068	-	-	-	-	-	-
$i_{pk,H}$	0.090	0.186	0.058	0.147	0.001	0.003	0.227	0.517
$i_{pk,L}$	0.073	0.216	0.049	0.136	0.003	0.017	0.202	0.464
$i_{off,H}$	0.373	0.225	0.280	0.176	0.031	0.011	0.865	0.586
$i_{off,L}$	0.294	0.262	0.172	0.161	0.026	0.029	0.584	0.582
$t_{snb,H}(ns)$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$t_{snb,L}(ns)$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$t_{d,H}(ns)$	-5.625	-13.125	17.769	25.201	-60.000	-60.000	0.000	0.000
$t_{d,L}(ns)$	18.750	0.000	28.256	0.000	0.000	0.000	60.000	0.000

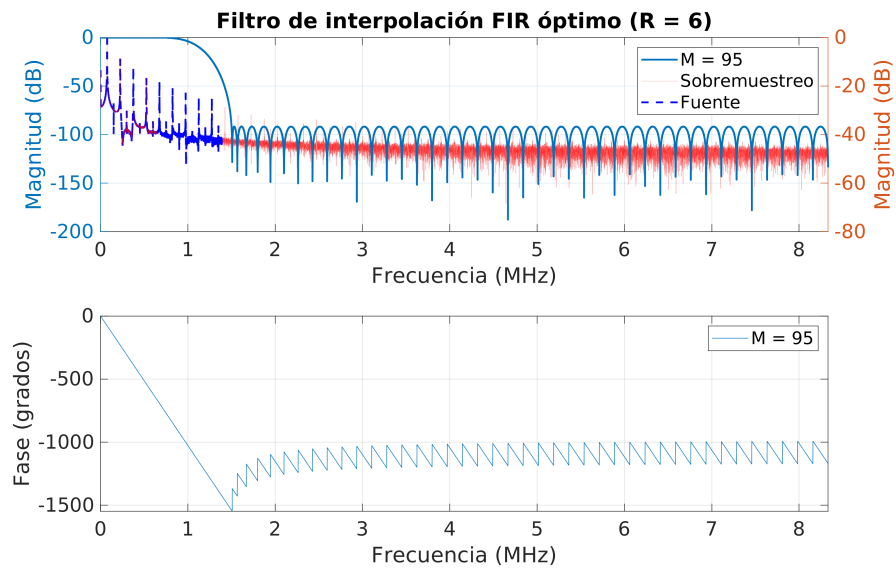
**Tabla 3.1.3:** Errores con filtro de Lagrange de orden 5 en el escenario 1

	Media		$\sigma$		min.		máx.	
	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz
$e_{i_{rms}}$	0.076	0.076	-	-	-	-	-	-
$e_{p_{avg}}$	0.077	0.077	-	-	-	-	-	-
$i_{pk,H}$	0.085	0.085	0.058	0.058	0.000	0.000	0.210	0.210
$i_{pk,L}$	0.086	0.086	0.057	0.057	0.002	0.002	0.193	0.193
$i_{off,H}$	0.094	0.094	0.067	0.067	0.006	0.006	0.250	0.250
$i_{off,L}$	0.092	0.092	0.067	0.067	0.002	0.002	0.243	0.243
$t_{snb,H}(ns)$	1.875	1.875	10.607	10.607	0.000	0.000	60.000	60.000
$t_{snb,L}(ns)$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$t_{d,H}(ns)$	-15.000	-15.000	26.396	26.396	-60.000	-60.000	0.000	0.000
$t_{d,L}(ns)$	5.625	5.625	17.769	17.769	0.000	0.000	60.000	60.000

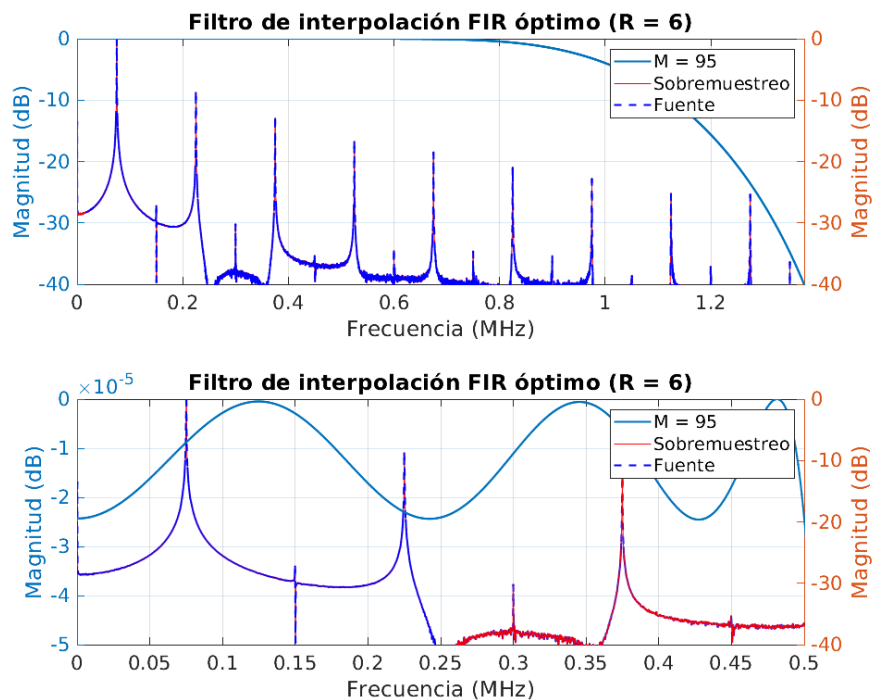
**Tabla 3.1.4:** Errores con filtro de Lagrange de orden 19 en el escenario 1

### 3.1.2. Filtro FIR óptimo

Tras un estudio con distintas configuraciones se ha decidido utilizar un filtro definido por una frecuencia de paso de 500 kHz (0.03 ciclos/muestra) y una frecuencia de parada de 1.5 MHz (0.09 ciclos/muestra). El rizado de las bandas de paso y atenuadas será de 0.002 dB y 40 dB respectivamente. En la figura 3.1.3a se puede verificar la respuesta lineal en fase y en el detalle de la figura 3.1.3b el rizado constante en la banda de paso.



(a) Respuesta en fase y frecuencia del filtro óptimo del escenario 1.



(b) Detalle de respuesta frecuencia del filtro óptimo del escenario 1.

**Figura 3.1.3:** Filtro óptimo del escenario 1



En la tabla 3.1.5 se detallan las características del filtro diseñado.

Orden	Retardo	Simetría	Coefficientes	Ceros	MPS	APS
95	48 (2880ns)	Sí	96	0	16	95

**Tabla 3.1.5:** Características del filtro óptimo del escenario 1

## Resultados

Los errores obtenidos con este diseño (Tabla 3.1.6) son cuantitativamente más altos que los obtenidos con los filtros de Lagrange. Sin embargo es mas económico en términos de memoria y operaciones.

	Media		$\sigma$		min.		máx.	
	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz
$e_{il_{rms}}$	0.021	0.074						
$e_{p_{avg}}$	0.771	3.936						
$il_{pk,H}$	0.099	0.250	0.058	0.065	0.011	0.130	0.240	0.340
$il_{pk,L}$	0.070	0.218	0.050	0.063	0.002	0.081	0.208	0.315
$i_{off,H}$	1.192	0.538	0.321	0.153	0.620	0.273	1.750	0.787
$i_{off,L}$	1.133	0.520	0.148	0.160	0.799	0.256	1.437	0.784
$t_{snb,H}(ns)$	0.000	-50.625	0.000	43.399	0.000	-120.000	0.000	0.000
$t_{snb,L}(ns)$	-20.625	-39.375	28.954	28.954	-60.000	-60.000	0.000	0.000
$t_{d,H}(ns)$	22.500	78.750	29.512	61.788	0.000	0.000	60.000	180.000
$t_{d,L}(ns)$	60.000	80.625	21.553	28.954	0.000	60.000	120.000	120.000

**Tabla 3.1.6:** Errores con filtro óptimo de orden 95 en el escenario 1

## 3.2. Escenario segundo

En este escenario se ha trabajado con una  $f_{s,ADC,2}$  igual a  $\frac{100}{48}$  MHz. Por lo tanto el factor de interpolación es 8. Como el factor de interpolación es una potencia de 2, se ha incluido el estudio de un diseño multietapa basado en filtros de *Half-band*.

En la figura 3.2.1 se se representa el espectro de señal obtenido a dicha frecuencia de muestreo.

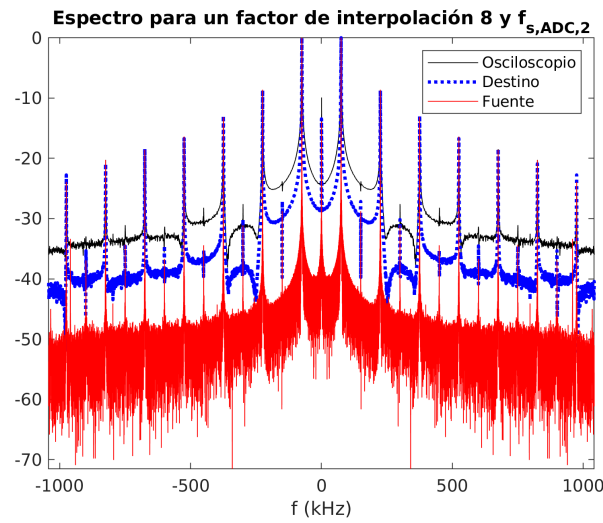


Figura 3.2.1: Espectro de la corriente para  $f_{s,ADC,2}$

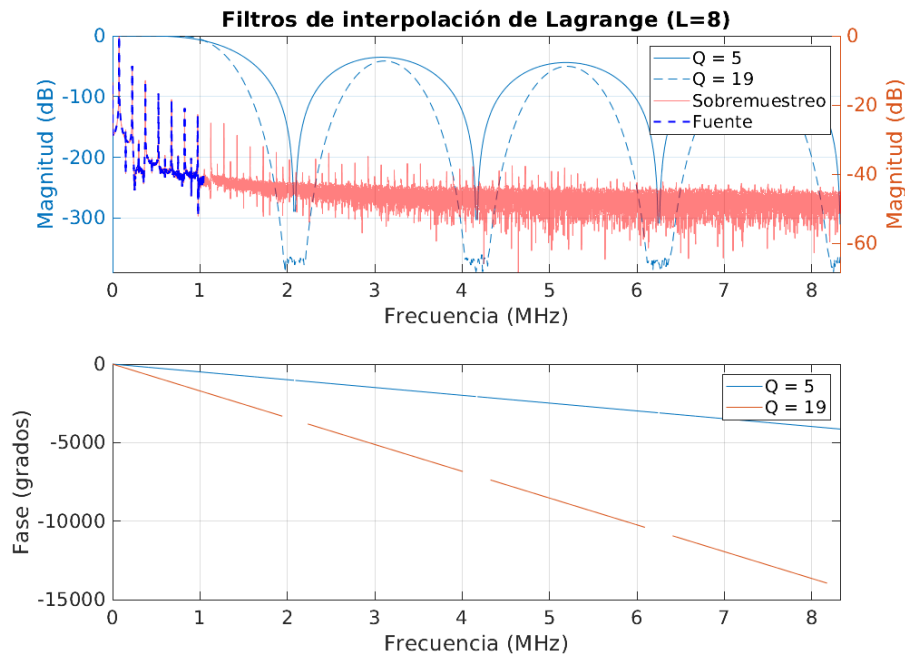
Como en el apartado anterior, en la tabla 3.2.1 se muestran los parámetros de calidad de la señal sin interpolar para realizar una valoración objetiva de los resultados obtenidos.

	Media		$\sigma$		min.		máx.	
	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz
$e_{i_{rms}}$	0.031	0.104	-	-	-	-	-	-
$e_{p_{avg}}$	0.065	0.207	-	-	-	-	-	-
$i_{pk,H}$	0.107	1.400	0.060	1.093	0.014	0.102	0.237	3.690
$i_{pk,L}$	0.074	1.348	0.057	1.031	0.002	0.019	0.196	3.474
$i_{off,H}$	3.302	2.640	1.407	2.338	0.666	0.102	5.853	7.521
$i_{off,L}$	3.859	2.789	1.313	2.368	0.686	0.019	6.115	7.388
$t_{snb,H}(ns)$	-360.000	-586.875	0.000	178.207	-360.000	-720.000	-360.000	-180.000
$t_{snb,L}(ns)$	-339.375	-620.625	28.954	28.954	-360.000	-660.000	-300.000	-600.000
$t_{d,H}(ns)$	845.625	1029.375	213.011	372.714	720.000	240.000	1260.000	1320.000
$t_{d,L}(ns)$	856.875	1080.000	237.683	169.020	480.000	600.000	1080.000	1200.000

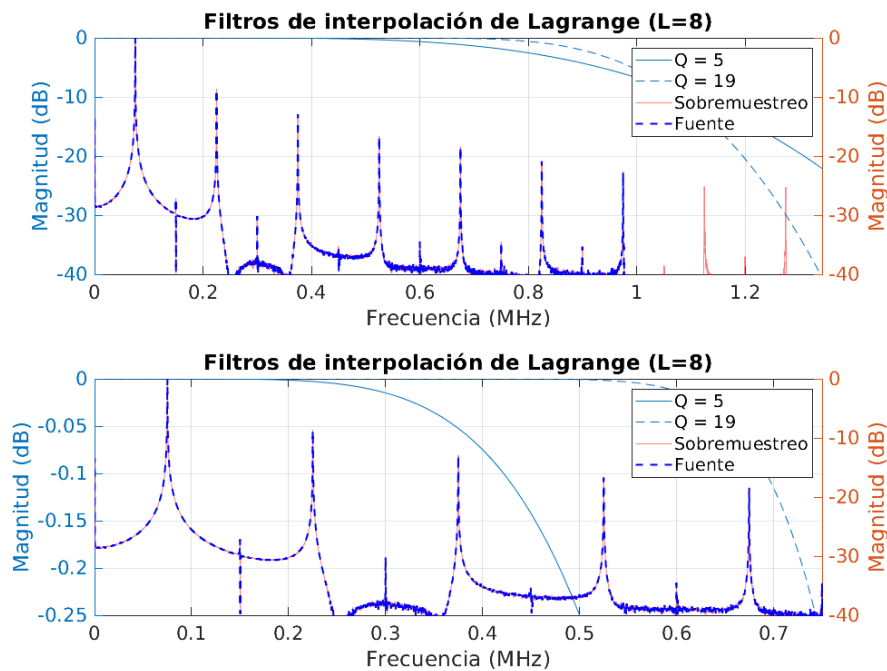
Tabla 3.2.1: Error sin interpolar para  $f_{s,ADC,2}$  del recipiente Zen12 (21 cm)

### 3.2.1. Polinomio de Lagrange

En la figura 3.2.2 se observa el comportamiento en frecuencia de los filtros de Lagrange de órdenes 5 y 19 para un factor de interpolación 8. En la figura 3.2.2b se puede apreciar que el tercer armónico sufre una atenuación algo mayor que en el escenario 1 (Figura 3.1.2b).



(a) Respuesta en fase y frecuencia de los filtros interpoladores de Lagrange del escenario 2.



(b) Detalle de respuesta frecuencia de los filtros interpoladores de Lagrange del escenario 2.

**Figura 3.2.2:** Filtros de Lagrange del escenario 2

Al aumentar el factor de interpolación, los filtros de Lagrange para el mismo orden son más exigentes. Para este escenario los filtros de Lagrange de órdenes 5 y 19 presentan las características mostradas en la tabla 3.2.2. Se observa como aumentan en número todas sus propiedades.

Orden	Retardo	Simetría	Coefficientes	Ceros	MPS	APS
5	24 (1440ns)	Sí	47	4	6	42
19	80 (4800ns)	Sí	159	18	18	140

**Tabla 3.2.2:** Características de los filtros de Lagrange del escenario 2

## Resultados

Por lo general los errores respecto al primer escenario han aumentado pero siguen estando muy por debajo del error permitido. Se observa la misma tendencia en el error relativo de las corrientes de pico y “off” que en el caso anterior.

	Media		$\sigma$		min.		máx.	
	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz
$e_{i_{rms}}$	0.035	0.101	-	-	-	-	-	-
$e_{p_{avg}}$	0.033	0.074	-	-	-	-	-	-
$i_{l_{pk,H}}$	0.076	0.582	0.057	0.294	0.005	0.100	0.225	1.024
$i_{l_{pk,L}}$	0.046	0.547	0.048	0.318	0.003	0.019	0.185	1.025
$i_{off,H}$	0.775	0.813	0.355	0.347	0.160	0.102	1.419	1.320
$i_{off,L}$	0.692	0.782	0.206	0.367	0.377	0.070	1.142	1.403
$t_{snb,H}(ns)$	0.000	-9.375	0.000	22.134	0.000	-60.000	0.000	0.000
$t_{snb,L}(ns)$	-3.750	-7.500	14.756	20.161	-60.000	-60.000	0.000	0.000
$t_{d,H}(ns)$	-5.625	-3.750	17.769	37.135	-60.000	-60.000	0.000	60.000
$t_{d,L}(ns)$	22.500	7.500	33.215	20.161	0.000	0.000	120.000	60.000

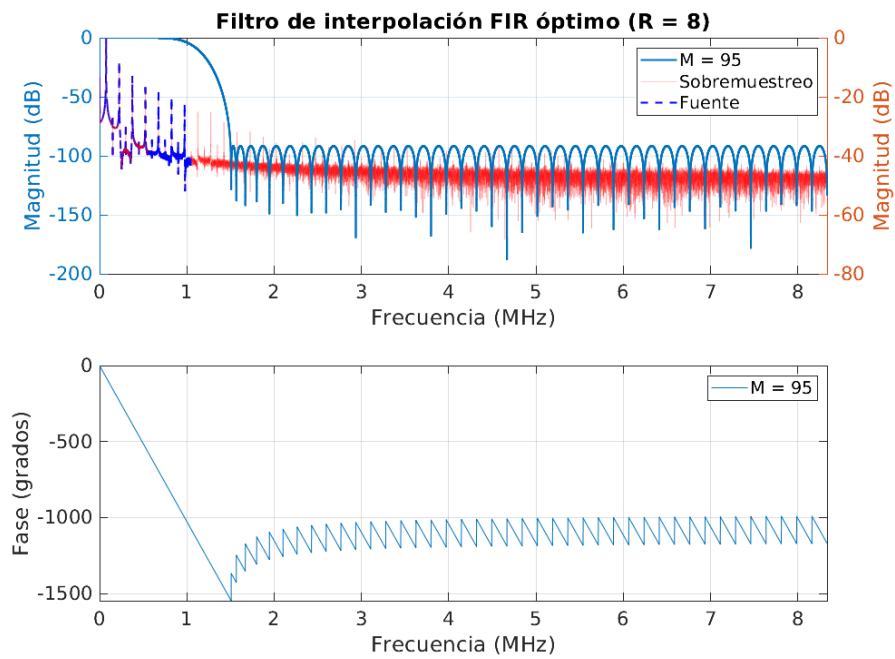
**Tabla 3.2.3:** Errores con filtro de Lagrange de orden 5 en el escenario 2

	Media		$\sigma$		min.		máx.	
	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz
$e_{i_{rms}}$	0.035	0.107	-	-	-	-	-	-
$e_{p_{avg}}$	0.034	0.095	-	-	-	-	-	-
$i_{l_{pk,H}}$	0.074	0.324	0.057	0.139	0.001	0.082	0.222	0.555
$i_{l_{pk,L}}$	0.044	0.286	0.047	0.161	0.002	0.006	0.194	0.541
$i_{off,H}$	0.588	0.467	0.312	0.172	0.007	0.102	1.140	0.753
$i_{off,L}$	0.520	0.440	0.149	0.193	0.241	0.070	0.862	0.827
$t_{snb,H}(ns)$	0.000	-9.375	0.000	22.134	0.000	-60.000	0.000	0.000
$t_{snb,L}(ns)$	0.000	-9.375	0.000	22.134	0.000	-60.000	0.000	0.000
$t_{d,H}(ns)$	-5.625	-3.750	17.769	37.135	-60.000	-60.000	0.000	60.000
$t_{d,L}(ns)$	18.750	11.250	28.256	23.793	0.000	0.000	60.000	60.000

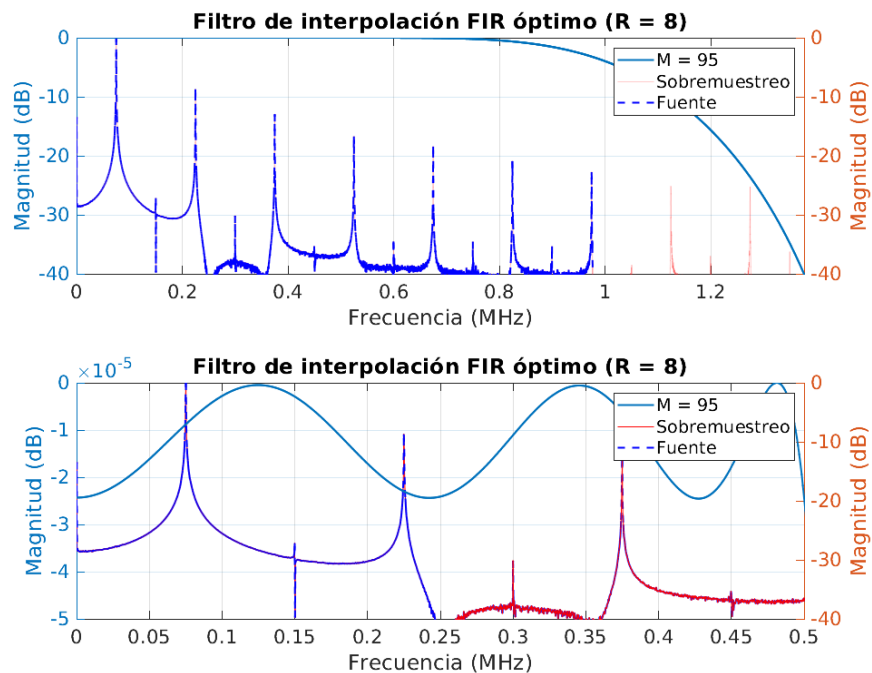
**Tabla 3.2.4:** Errores con filtro de Lagrange de orden 19 en el escenario 2

### 3.2.2. Filtro FIR óptimo

En este escenario las especificaciones del filtro son las mismas que las del filtro óptimo del escenario 1 (Sección 3.1.1). En la figura 3.2.3 se pueden visualizar la amplitud y fase de la respuesta en frecuencia de este filtro



(a) Respuesta en fase y frecuencia del filtro interpolador óptimo del escenario 2.



(b) Detalle de respuesta frecuencia del filtro interpolador óptimo del escenario 2.

**Figura 3.2.3:** Filtro óptimo del escenario 2

Las propiedades del filtro se mantienen con respecto al anterior, salvo el número de multiplicaciones que se reduce por haber aumentado el factor de sobremuestreo (Tabla 3.2.5).

Orden	Retardo	Simetría	Coeficientes	Ceros	MPS	APS
95	48 (2880ns)	Sí	96	0	12	95

**Tabla 3.2.5:** Características del filtro óptimo del escenario 2

## Resultados

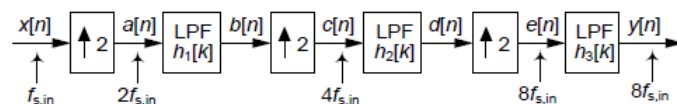
Respecto al escenario anterior la diferencia es mínima. Los errores del filtro óptimo para este escenario (Tabla 3.2.6) son muy similares a los obtenidos en el escenario 1 (Tabla 3.1.6). Como en el caso anterior, los errores respecto al filtro de Lagrange son peores, pero por debajo de la cota establecida.

	Media		$\sigma$		min.		máx.	
	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz
$e_{i_{rms}}$	0.035	0.108	-	-	-	-	-	-
$e_{p_{avg}}$	0.785	3.963	-	-	-	-	-	-
$i_{pk,H}$	0.072	0.301	0.057	0.195	0.003	0.001	0.220	0.581
$i_{pk,L}$	0.043	0.286	0.047	0.193	0.004	0.024	0.198	0.601
$i_{off,H}$	1.284	0.668	0.345	0.259	0.591	0.191	1.916	1.127
$i_{off,L}$	1.214	0.642	0.167	0.280	0.885	0.183	1.637	1.194
$t_{snb,H}(ns)$	0.000	-52.500	0.000	45.081	0.000	-120.000	0.000	0.000
$t_{snb,L}(ns)$	-20.625	-39.375	28.954	28.954	-60.000	-60.000	0.000	0.000
$t_{d,H}(ns)$	20.625	76.875	28.954	59.484	0.000	0.000	60.000	180.000
$t_{d,L}(ns)$	61.875	80.625	24.021	28.954	0.000	60.000	120.000	120.000

**Tabla 3.2.6:** Errores con filtro de óptimo en el escenario 2

### 3.2.3. Filtro *Half-band*

En este escenario se introduce una nueva propuesta de diseño basada en filtros *Half-band* en cascada (Figura 3.2.4).



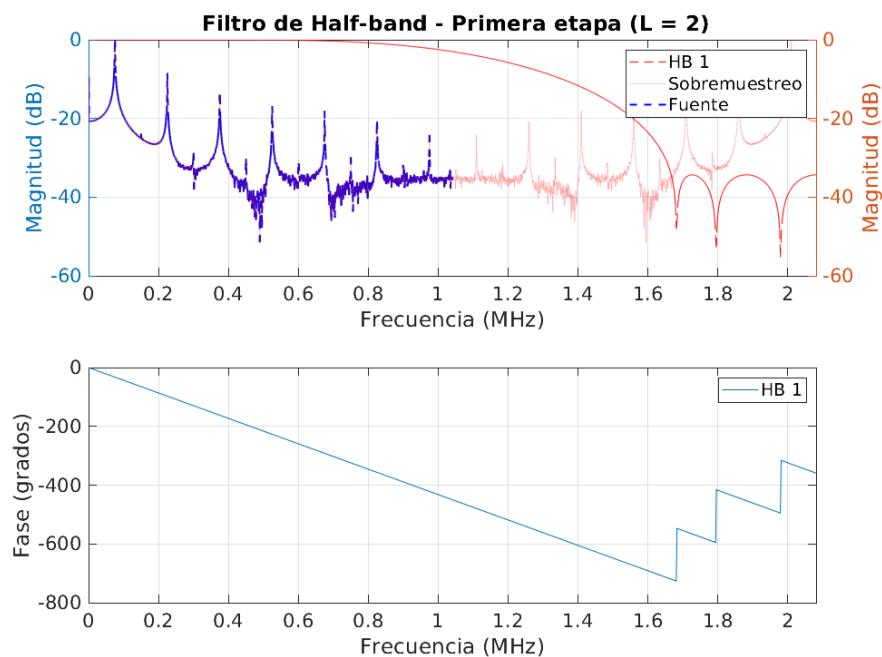
**Figura 3.2.4:** Esquema funcional de la interpolación multietapa basada en filtros *Half-band* [12].

Se ha optado por una implementación con filtros de distinto número de coeficientes con las configuraciones mostradas en la tabla 3.2.7[12].

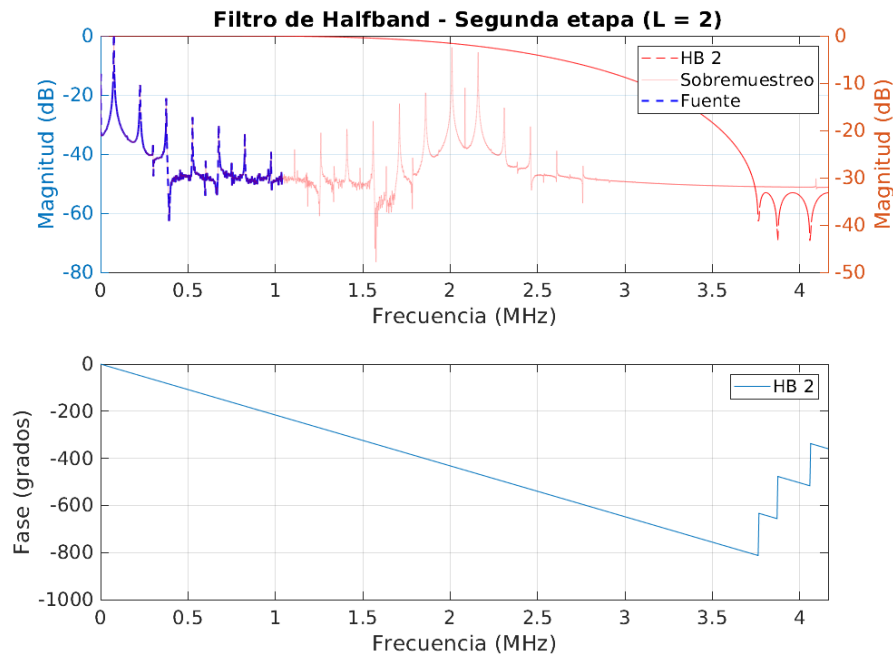
Orden	$f_0$	$A_p$ (dB)	Retardo	Simetría	Coefficientes	Ceros	MPS	APS
10	416.68 kHz (0.2 ciclos/muestra)	0.005	6 (1440ns)	Sí	11	4	3	6
10	416.68 kHz (0.1 ciclos/muestra)	0.0025	6 (720ns)	Sí	11	4	3	6
6	416.68 kHz (0.05 ciclos/muestra)	0.002	3 (180 ns)	Sí	7	2	2	4

**Tabla 3.2.7:** Características de las etapas de filtrado *Half-band* del escenario 2

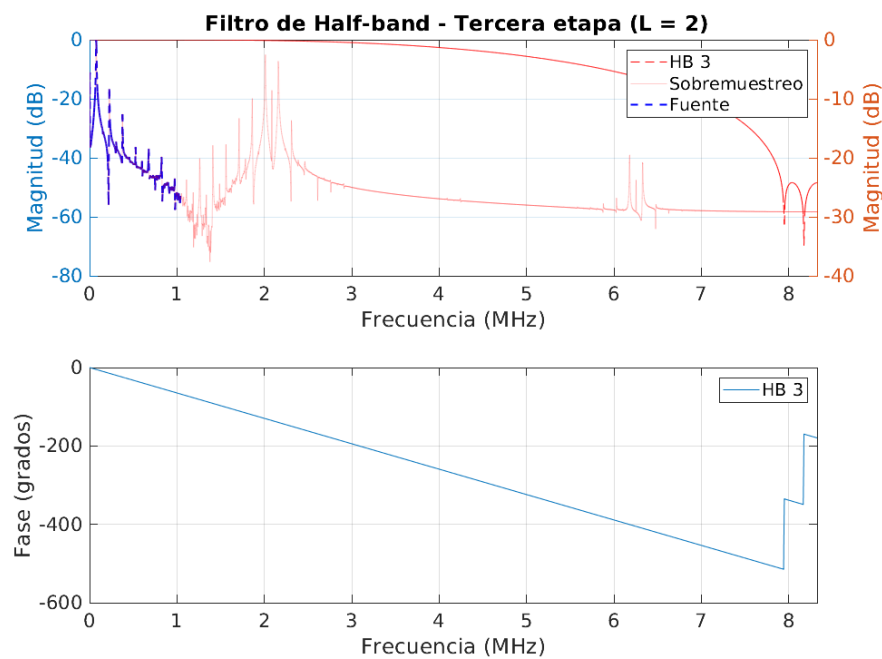
En figura 3.2.5 se observa como se aplican los filtros de forma sucesiva a la señal. Se representa por un lado la respuesta impulsional del filtro de cada etapa y la señal sobre la que se aplica dicho filtro. Cabe destacar que en este caso los filtros operan a distinta velocidad. Esto quiere decir que la primera etapa trabaja al doble de  $f_{s,ADC,2}$ , la segunda al cuádruple y la tercera a velocidad  $f_{s,dst}$ . Dado que se ha elegido una implementación en la que el número de coeficientes disminuye conforme se aumenta la frecuencia de trabajo, hace que los filtros con mayor número de operaciones por muestra trabajen a una velocidad reducida.



(a) Respuesta en fase y frecuencia de la primera etapa de filtrado.



(b) Respuesta en fase y frecuencia de la segunda etapa de filtrado.

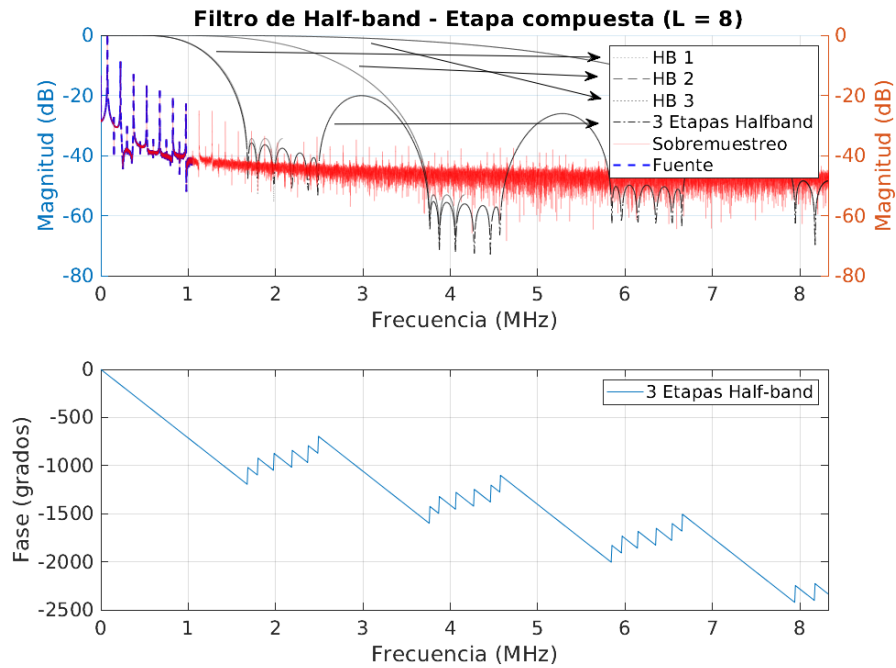


(c) Respuesta en fase y frecuencia de la tercera etapa de filtrado.

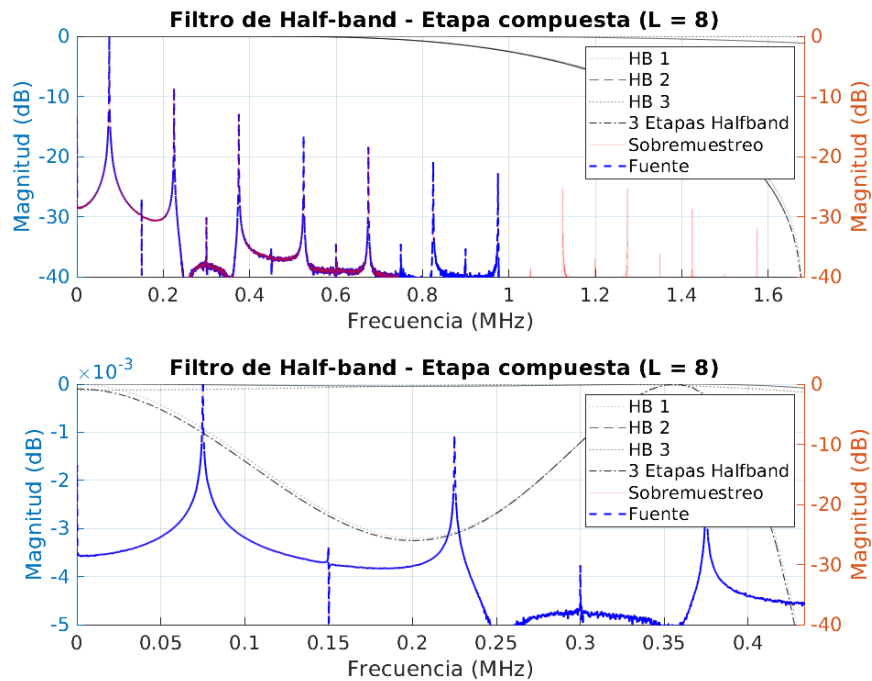
**Figura 3.2.5:** Etapas de filtrado *Half-band* del escenario 2

La respuesta impulsional resultado de las tres etapas en cascada se muestra en la figura 3.2.6. Se puede apreciar su semejanza en la banda de paso con el filtro óptimo de este mismo escenario. A su vez se puede comprobar que su respuesta en fase es lineal y que el rizado es constante en la banda de paso.





(a) Respuesta en fase y frecuencia del filtro *Half-band* resultante del escenario 2.



(b) Detalle de respuesta en frecuencia del filtro *Half-band* resultante del escenario 2.

**Figura 3.2.6:** Filtro *Half-band* del escenario 2

## Resultados

Los resultados de la tabla 3.2.8 exponen que el error cometido es, en general, menor respecto al filtro óptimo y similar a los errores obtenidos con un filtro de Lagrange de

orden 19 (Tabla 3.2.4). La ventaja de esta configuración es el número de operaciones por muestra y la frecuencia de trabajo a la que se realizan las operaciones. Mientras que en las configuraciones anteriores los filtros trabajan a velocidad  $L \cdot f_{s,ADC}$ , en este caso, la velocidad es distinta tal y como se a comentado anteriormente. Además se puede observar que el número de operaciones a realizar es muy reducido. En total, por muestra, se realizan 10 multiplicaciones y 14 sumas. El número de sumas es mucho menor que el número de sumas que hay que realizar con el filtro óptimo (95 sumas) o el de Lagrange (140 sumas). Por otro lado el número de multiplicaciones se reduce en 2 y 8 respectivamente. También es importante observar que el retardo introducido total es de 2340 ns, que es menor que cualquiera de los dos filtros anteriores. Por último, en términos de memoria, las ventajas de implementación de estos filtros hace que tan sólo se tengan que almacenar en total 8 coeficientes, mientras que en el filtro óptimo serían 48 y en el de Lagrange 141.

	Media		$\sigma$		min.		máx.	
	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz
$e_{i_{rms}}$	0.066	0.119	-	-	-	-	-	-
$e_{p_{avg}}$	0.063	0.098	-	-	-	-	-	-
$i_{l_{pk,H}}$	0.062	0.445	0.044	0.266	0.003	0.003	0.202	0.829
$i_{l_{pk,L}}$	0.046	0.416	0.030	0.279	0.001	0.010	0.143	0.855
$i_{off,H}$	0.661	0.638	0.335	0.297	0.019	0.102	1.261	1.103
$i_{off,L}$	0.586	0.607	0.176	0.321	0.297	0.070	1.002	1.187
$t_{snb,H}(\text{ns})$	0.000	-9.375	0.000	22.134	0.000	-60.000	0.000	0.000
$t_{snb,L}(\text{ns})$	0.000	-7.500	0.000	20.161	0.000	-60.000	0.000	0.000
$t_{d,H}(\text{ns})$	-5.625	-3.750	17.769	37.135	-60.000	-60.000	0.000	60.000
$t_{d,L}(\text{ns})$	18.750	7.500	28.256	20.161	0.000	0.000	60.000	60.000

**Tabla 3.2.8:** Errores con filtros *Half-band* en cascada en el escenario 2

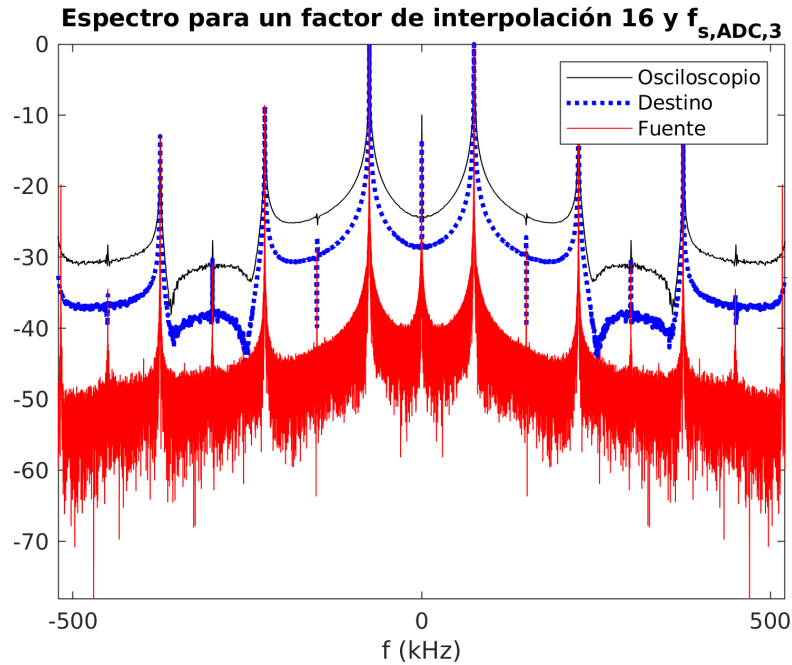
### 3.3. Escenario tercero

El último escenario es de especial interés al trabajar a una frecuencia cercana a 1 Mhz como es  $f_{s,ADC,3}$  igual a  $\frac{100}{96}$  MHz. El factor de interpolación en este caso es 16. Al igual que en el escenario anterior este factor es potencia de 2, por lo que también se estudia un diseño multietapa basado en filtros *Half-band*.

En el espectro representado en la figura 3.3.1, se observa que el tercer armónico se sitúa muy cerca del límite de Nyquist ( $f_s/2$ ). Por lo que la banda de transición de los filtros deberá ser muy abrupta. Esto implica un aumento significativo en el número de

coeficientes.

**Figura 3.3.1:** Espectro de la corriente para  $f_{s,ADC,3}$



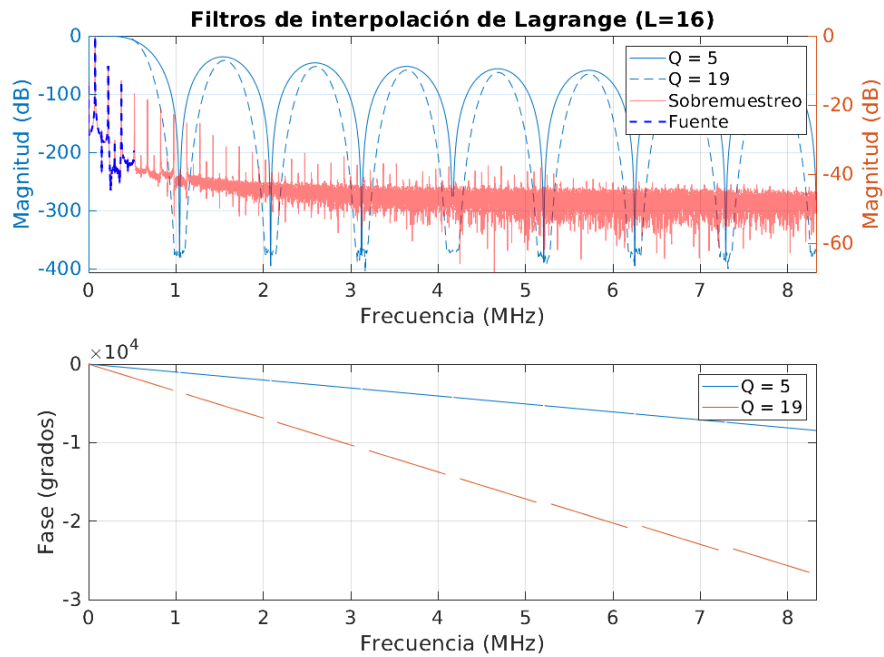
En este caso los errores medidos sobre los parámetros de interés para la señal sin interpolar son los peores que se han obtenido (Tabla 3.3.1).

	Media		$\sigma$		min.		máx.	
	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz
$e_{i_{rms}}$	0.096	0.026	-	-	-	-	-	-
$e_{p_{avg}}$	0.178	0.108	-	-	-	-	-	-
$i_{l_{pk,H}}$	0.234	5.324	0.094	2.544	0.068	2.233	0.481	10.246
$i_{l_{pk,L}}$	0.203	5.158	0.104	2.477	0.037	2.160	0.482	9.660
$i_{off,H}$	6.977	10.074	2.294	6.602	1.989	2.194	10.007	23.594
$i_{off,L}$	7.304	9.828	2.230	6.399	3.012	2.160	10.380	22.809
$t_{snb,H}(ns)$	-360.000	-571.875	0.000	481.446	-360.000	-1200.000	-360.000	-180.000
$t_{snb,L}(ns)$	-339.375	-680.625	28.954	487.826	-360.000	-1140.000	-300.000	-120.000
$t_{d,H}(ns)$	1445.625	1674.375	423.815	235.782	1200.000	780.000	2220.000	1800.000
$t_{d,L}(ns)$	1471.875	1800.000	485.290	379.167	960.000	1560.000	1980.000	2640.000

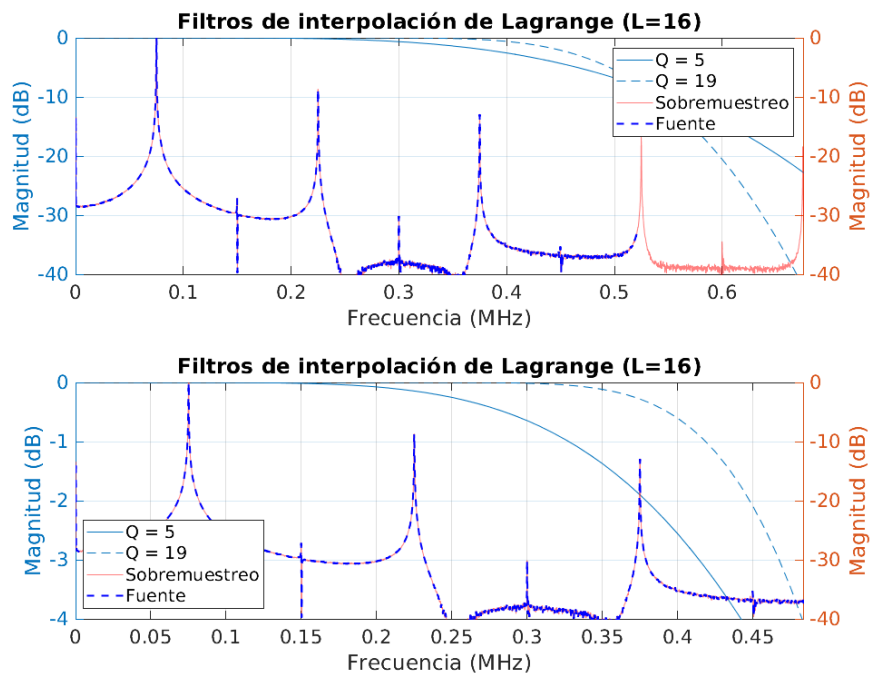
**Tabla 3.3.1:** Error sin interpolar para  $f_{s,ADC,3}$  del recipiente Zen12 (21 cm)

### 3.3.1. Polinomio de Lagrange

En la figura 3.3.2 se observa que el orden del filtro empleado en los escenarios anteriores empieza a ser crítico, ya que el tercer armónico sufre una atenuación considerable en el primer caso, casi 2 dB, y 0.375 dB en el segundo. Esto se puede resolver fácilmente aumentando el orden del filtro.



(a) Respuesta en fase y frecuencia de los filtros interpoladores de Lagrange del escenario 3.



(b) Detalle de respuesta frecuencia de los filtros interpoladores de Lagrange del escenario 3.

**Figura 3.3.2:** Filtros de Lagrange del escenario 3

Para un factor de interpolación 16 los filtros de Lagrange de órdenes 5 y 19 tienen las características mostradas en la tabla. Se observa un aumento significativo en el número de coeficientes, dado el aumento de factor de interpolación.

Orden	Retardo	Simetría	Coefficientes	Ceros	MPS	APS
5	48 (2880ns)	Sí	95	4	6	90
19	160 (9600ns)	Sí	319	18	19	300

## Resultados

En las tablas 3.3.2 y 3.3.3 se puede observar que los errores cometidos han aumentado significativamente respecto a los dos escenarios anteriores. Aun así en el segundo caso se estarían cumpliendo todos los objetivos, aunque de forma crítica en las medidas relativas. Por ello para asegurar un correcto cumplimiento de los objetivos sería conveniente aumentar el orden del filtro. Se debe tener en cuenta que el crecimiento del número de coeficientes es lineal con el factor de interpolación y el orden del filtro.

	Media		$\sigma$		min.		máx.	
	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz
$e_{ilrms}$	0.098	0.036	-	-	-	-	-	-
$e_{pavg}$	0.075	0.299	-	-	-	-	-	-
$i_{pk,H}$	0.096	3.291	0.061	0.738	0.016	2.185	0.274	4.365
$i_{pk,L}$	0.073	3.207	0.050	0.730	0.007	2.114	0.177	4.231
$i_{off,H}$	2.699	<b>4.811</b>	0.473	0.855	1.740	3.036	3.591	<b>6.158</b>
$i_{off,L}$	2.522	<b>4.718</b>	0.364	0.831	1.882	2.998	3.135	<b>6.116</b>
$t_{snb,H}(ns)$	0.000	-11.250	0.000	23.793	0.000	-60.000	0.000	0.000
$t_{snb,L}(ns)$	-20.625	-16.875	28.954	27.408	-60.000	-60.000	0.000	0.000
$t_{d,H}(ns)$	-7.500	11.250	20.161	35.537	-60.000	-60.000	0.000	60.000
$t_{d,L}(ns)$	39.375	30.000	32.719	34.078	0.000	0.000	120.000	120.000

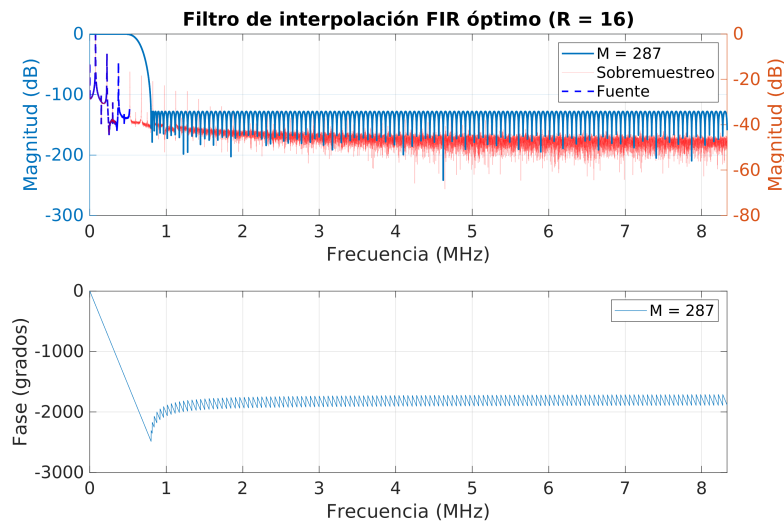
**Tabla 3.3.2:** Errores con filtro de Lagrange de orden 5 en el escenario 3

	Media		$\sigma$		min.		máx.	
	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz
$e_{ilrms}$	0.102	0.023	-	-	-	-	-	-
$e_{pavg}$	0.082	0.184	-	-	-	-	-	-
$i_{pk,H}$	0.092	2.654	0.066	0.393	0.000	2.014	0.226	3.245
$i_{pk,L}$	0.074	2.592	0.052	0.383	0.007	1.980	0.169	3.178
$i_{off,H}$	2.318	4.005	0.365	0.484	1.557	3.036	3.033	4.871
$i_{off,L}$	2.166	3.946	0.203	0.459	1.814	2.998	2.552	4.847
$t_{snb,H}(ns)$	0.000	-3.750	0.000	14.756	0.000	-60.000	0.000	0.000
$t_{snb,L}(ns)$	-20.625	-9.375	28.954	22.134	-60.000	-60.000	0.000	0.000
$t_{d,H}(ns)$	-16.875	0.000	27.408	30.480	-60.000	-60.000	0.000	60.000
$t_{d,L}(ns)$	39.375	18.750	32.719	28.256	0.000	0.000	120.000	60.000

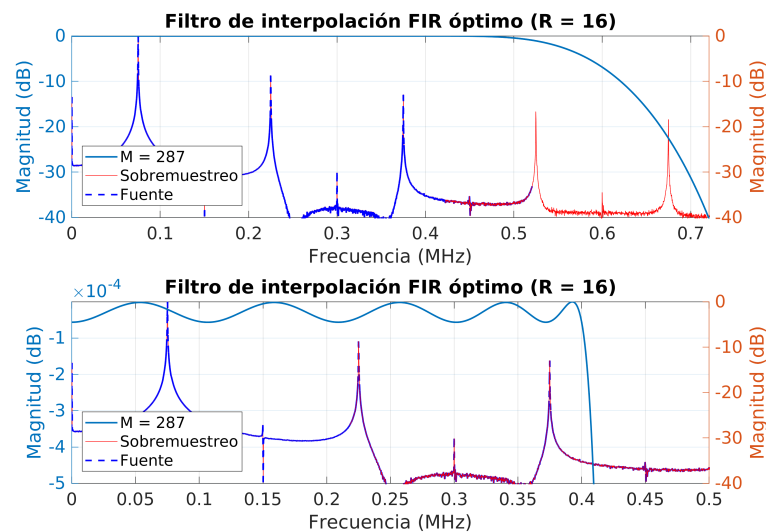
**Tabla 3.3.3:** Errores con filtro de Lagrange de orden 19 en el escenario 3

### 3.3.2. Filtro FIR óptimo

Se ha diseñado un filtro FIR óptimo que minimiza la atenuación de los armónicos de interés. Como se observa en el espectro de la señal de origen de este escenario (Figura 3.3.1), el tercer armónico queda bastante cerca del límite de Nyquist. Esto va a requerir un filtro óptimo con una banda de transición muy abrupta y una banda de paso muy estrecha. El filtro se ha especificado con una frecuencia de paso de 400 kHz (0.0240 ciclos/muestra) y una frecuencia de parada de 800 kHz (0.0480 ciclos/muestra). El rizado de las bandas de paso y atenuadas es de 0.002 dB y 40 dB respectivamente.



(a) Respuesta en fase y frecuencia del filtro interpolador óptimo del escenario 3.



(b) Detalle de respuesta frecuencia del filtro interpolador óptimo del escenario 3.

**Figura 3.3.3:** Filtro óptimo del escenario 3

El orden de este filtro es elevado dadas sus restricciones. Esto incurre en mayores necesidades de memoria y un mayor número de operaciones. Las características del filtro diseñado se muestran en la siguiente tabla.

Orden	Retardo	Simetría	Coeficientes	Ceros	MPS	APS
287	144 (8640ns)	Sí	288	0	18	287

**Tabla 3.3.4:** Características del filtro óptimo del escenario 3

## Resultados

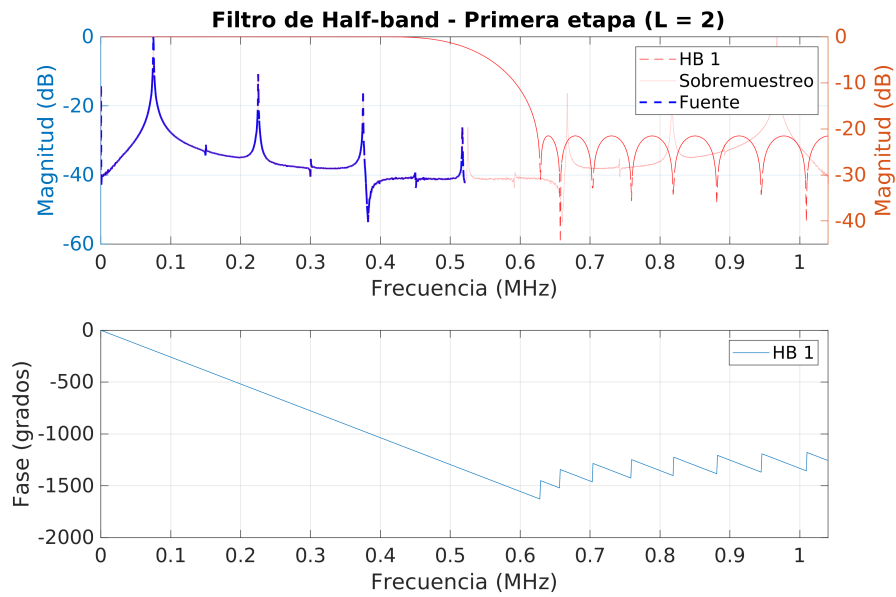
Se observa en los resultados (Tabla 3.3.5) que los errores con respecto al filtro de Lagrange son algo mayores. Cabe señalar que aunque en media los valores están dentro del margen permitido, para el caso de 75 kHz el error máximo supera en algún caso el 5%.

	Media		$\sigma$		min.		máx.	
	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz
$e_{i_{rms}}$	0.103	0.039	-	-	-	-	-	-
$e_{p_{avg}}$	0.819	3.616	-	-	-	-	-	-
$i_{l_{pk,H}}$	0.154	2.273	0.094	1.005	0.002	0.810	0.307	3.709
$i_{l_{pk,L}}$	0.128	2.192	0.083	0.974	0.009	0.799	0.332	3.704
$i_{off,H}$	2.944	3.899	0.597	1.064	1.709	2.207	3.963	5.692
$i_{off,L}$	2.773	3.823	0.477	1.025	1.987	2.190	3.645	5.613
$t_{snb,H}(ns)$	0.000	-37.500	0.000	39.595	0.000	-120.000	0.000	0.000
$t_{snb,L}(ns)$	20.625	-33.750	28.954	30.241	0.000	-60.000	60.000	0.000
$t_{d,H}(ns)$	-24.375	52.500	29.939	64.207	-60.000	-60.000	0.000	180.000
$t_{d,L}(ns)$	-69.375	63.750	30.894	48.042	-120.000	0.000	0.000	120.000

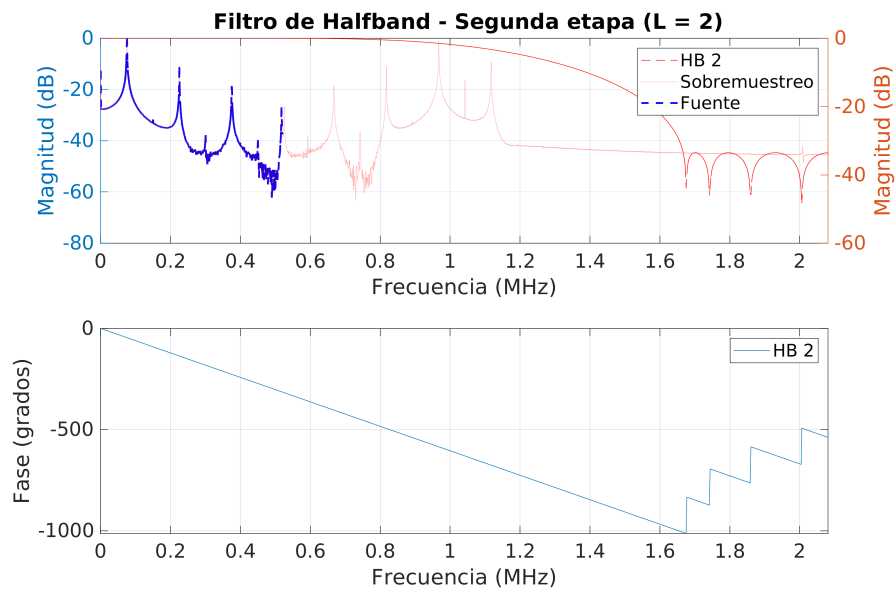
**Tabla 3.3.5:** Errores con filtro óptimo en el escenario 3

### 3.3.3. Filtro *Half-band*

Al igual que en el escenario anterior, se procede a diseñar una implementación multietapa. En este caso son cuatro etapas con distinto número de coeficientes. El detalle de cada una de las etapas se muestra en la figura 3.3.4.

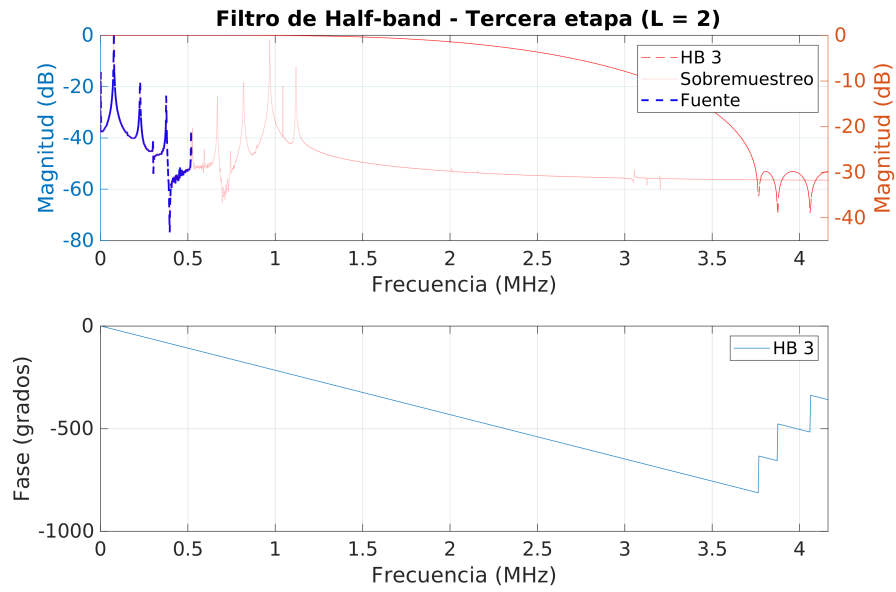


(a) Respuesta en fase y frecuencia de la primera etapa de filtrado.

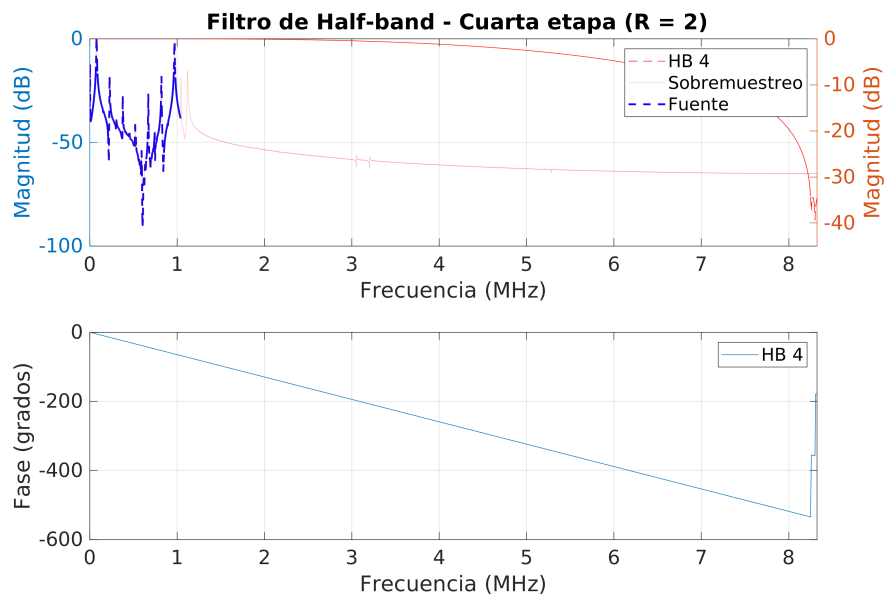


(b) Respuesta en fase y frecuencia de la segunda etapa de filtrado.





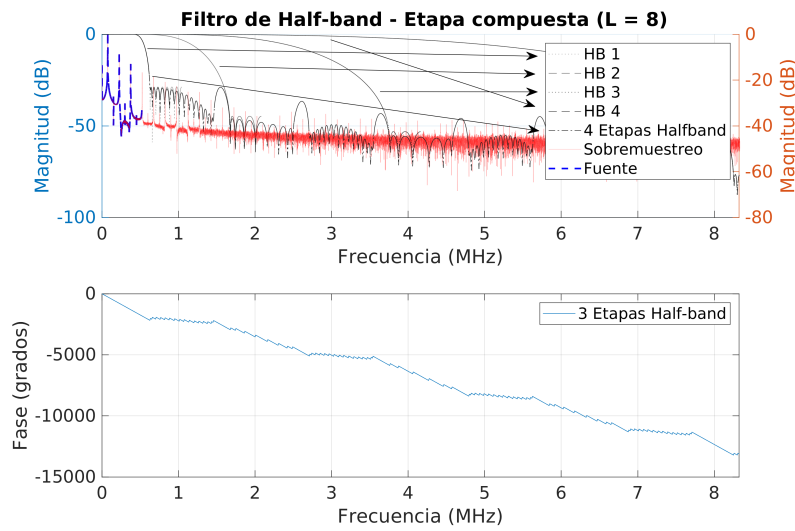
(c) Respuesta en fase y frecuencia de la tercera etapa de filtrado.



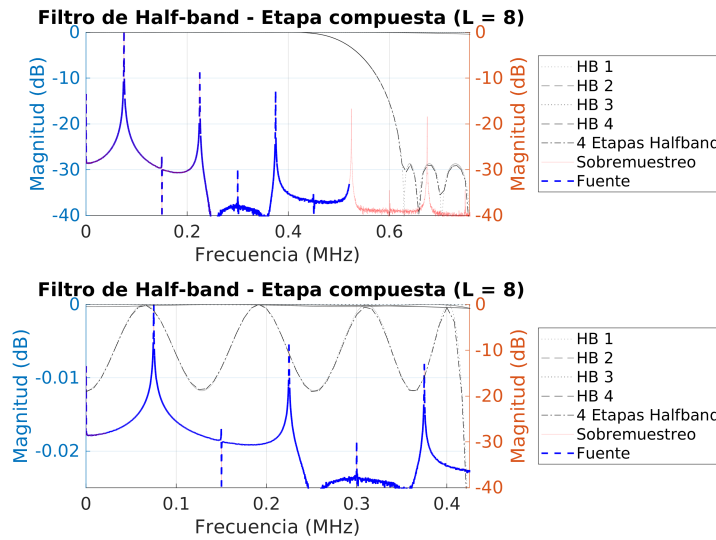
(d) Respuesta en fase y frecuencia de la cuarta etapa de filtrado.

**Figura 3.3.4:** Etapas de filtrado *Half-band* del escenario 3

El filtro resultante de aplicar las cuatro etapas en cascada se muestra en la figura 3.3.5 junto con el espectro de la señal de entrada. Se observa que el rizado en la banda de paso se mantiene constante debido a que las restricciones sobre la banda de paso de las últimas etapas es más baja que en las anteriores.



(a) Respuesta en fase y frecuencia del filtro *Half-band* resultante del escenario 3.



(b) Detalle de respuesta en frecuencia del filtro *Half-band* resultante del escenario 3.

**Figura 3.3.5:** Filtro *Half-band* del escenario 3

A continuación se detallan las especificaciones de cada una de las etapas. En la tabla 3.3.6 se ve como el número de coeficientes decrece en cada etapa. Si bien es cierto que es la configuración con el mayor retardo, es muy económico en cuanto a sumas (tan solo 34), multiplicaciones y necesidades de memoria.

Orden	$f_0$	$A_p$ (dB)	Retardo	Simetría	Coefficientes	Ceros	MPS	APS
30	416.68 kHz (0.4 ciclos/muestra)	0.025	16 (7680ns)	Sí	31	14	8	16
14	416.68 kHz (0.2 ciclos/muestra)	0.0025	8 (1920ns)	Sí	15	6	4	8
10	416.68 kHz (0.1 ciclos/muestra)	0.0005	6 (720ns)	Sí	11	4	3	6
6	83.33 kHz (0.01 ciclos/muestra)	0.0005	3 (180 ns)	Sí	7	2	2	4

**Tabla 3.3.6:** Características de las etapas de filtrado *Half-band* del escenario 3

## Resultados

De los resultados de la tabla 3.3.7 se observa que los errores obtenidos son mas similares al caso del filtro de Lagrange, incluso menores. Sin embargo se ha conseguido reducir el número de operaciones por debajo de cualquier otra implementación, 17 multiplicaciones y 34 sumas. El retardo introducido por este sistema es el mayor de todos (10500ns). Por otro lado los requerimientos de memoria para los coeficientes y la linea de retardo son los más bajos de este escenario.

	Media		$\sigma$		min.		máx.	
	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz
$e_{i_{rms}}$	0.118	0.144	-	-	-	-	-	-
$e_{p_{avg}}$	0.105	0.040	-	-	-	-	-	-
$i_{l_{pk,H}}$	0.094	2.516	0.064	0.366	0.003	1.889	0.226	3.025
$i_{l_{pk,L}}$	0.084	2.463	0.048	0.346	0.002	1.865	0.199	2.991
$i_{off,H}$	2.247	3.933	0.360	0.431	1.526	3.131	2.932	4.739
$i_{off,L}$	2.101	3.885	0.173	0.403	1.753	3.091	2.424	4.717
$t_{snb,H}$ (ns)	0.000	3.750	0.000	14.756	0.000	0.000	0.000	60.000
$t_{snb,L}$ (ns)	20.625	5.625	28.954	17.769	0.000	0.000	60.000	60.000
$t_{d,H}$ (ns)	16.875	1.875	27.408	28.448	0.000	-60.000	60.000	60.000
$t_{d,L}$ (ns)	-39.375	-11.250	32.719	28.256	-120.000	-60.000	0.000	60.000

**Tabla 3.3.7:** Errores con filtros *Half-band* en cascada en el escenario 3

## 3.4. Resultados

En los apartados anteriores se ha observado como a medida que se incrementa el factor de interpolación, los parámetros de calidad empeoran y la banda de transición de los filtros se hace más estrecha. Esto se debe principalmente a que los parámetros de diseño se mantienen pero la frecuencia de muestreo se reduce y las replicas espectrales al realizar el sobre-muestreo están más juntas.

De los escenarios estudiados se puede decir que el último, y más extremo, requiere de un mayor coste computacional y las medidas de calidad se sitúan en el extremo superior. Existe la posibilidad de una implementación con estas características que se deja para una línea de trabajo futuro.

El escenario intermedio proporciona un caso interesante en términos de computo e implementación. Por un lado se reduce la frecuencia de muestreo a 2,08 MHz, un 25 %

menos que el primer escenario, algo muy positivo en términos de consumo eléctrico, ruido y resolución del ADC. Además se podría sustituir por un modelo más económico como el LTC2313-12[13] cuya  $f_{s,ADC,max}$  son 2.5 Msps. Por otro lado se puede valorar el empleo de una configuración multietapa de filtros *Half-band* al interpolar por un factor potencia de 2. Esto es importante por dos motivos: los requisitos computacionales son bajos y permite la implementación de un diseño modular altamente escalable y reconfigurable.

El primero de los escenarios había sido abordado este mismo año eligiendo un filtro óptimo de 108 coeficientes [5, p. 28]. Sin embargo, existiendo la posibilidad de modificar la frecuencia de muestreo y habiendo obtenido unos errores, en general, menores que los obtenidos en dicho trabajo [5, Anexo E] se decide implementar esta solución. Además una de las líneas futuras del trabajo de Barrutia A. era el estudio de la viabilidad de implementar un sistema de filtrado para reconstruir la corriente que se adapte a la frecuencia de trabajo de los dispositivos; de modo que, conforme  $f_{sw}$  crezca, la reconstrucción de corriente contemple armónicos más altos y se persiga una reconstrucción de mayor calidad. Y por otra parte, la variación de la frecuencia de muestreo en función de la frecuencia de trabajo, con el fin de disponer de la misma información por período de conmutación independientemente de las condiciones de trabajo.

En esta línea, la solución elegida abre el camino a un sistema reconfigurable de filtrado multietapa que pueda combinar los bloques en función de la  $f_{sw}$  [5, p. 52].

# Capítulo 4

## Diseño del sistema digital, implementación en VHDL

### 4.1. Especificaciones

En la sección 3.4 se ha elegido el diseño multi-etapa con filtros *Half-band* del segundo escenario. Por lo tanto, el sistema a implementar cuenta con las siguientes especificaciones:

- Frecuencia de muestreo de entrada  $f_{s,ADC,2}$  igual a 2,08 MHz ( $\frac{100}{48}$  MHz).
- Factor de interpolación 8 (Tres etapas de interpolación por un factor 2)
- Frecuencia de muestreo de salida  $f_{s,dst}$  igual a 16,67 MHz ( $\frac{100}{6}$  MHz)

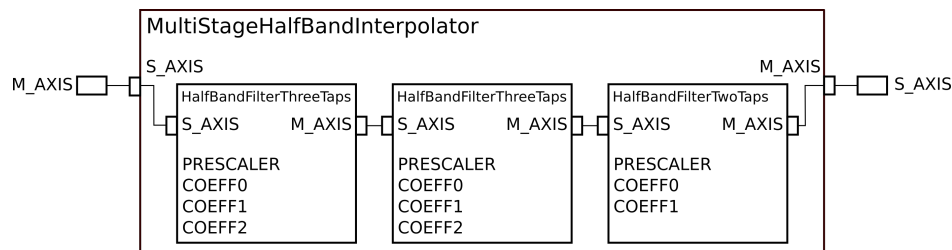
Para que el bloque implementado pueda ser integrado en el sistema sistema actual, los buses de entrada y salida deben ser compatibles con la interfaz AXI4-Stream definida en la especificación AMBA 4 (*Advanced Microcontroller Bus Architecture*). La especificación AMBA 4 define un estándar de comunicaciones *on-chip* para el diseño de sistemas embebidos de alto rendimiento [14].

### 4.2. Arquitectura del sistema

El objetivo es diseñar un bloque IP que interpole una señal muestreada a 2.08 MHz por un factor 8. Para realizar esta tarea se ha decidido utilizar un sistema

de tres etapas en el que cada etapa interpola por un factor 2 con filtros *Half-band*. Además, este sistema debe implementar un bus de comunicación compatible con la interfaz AXI4-Stream. El protocolo de comunicación, como se detalla en la sección 4.3, define señales para los dos extremos de comunicación. Estas señales están agrupadas en la figura 4.2.1 bajo el nombre *M\_AXIS* y *S\_AXIS*, para el emisor y el receptor respectivamente. Como se desea que el sistema sea escalable y fácilmente reconfigurable en el futuro, se ha optado por que cada una de las etapas se comunique mediante el mismo estándar de comunicación. En la figura 4.2.1 se detalla el conexionado de los bloques y los parámetros configurables de cada componente.

El sistema `MultiStageHalfBandInterpolator` estará constituido por 3 componentes: 2 instancias de `HalfBandFilterThreeTaps`, etapas de 11 coeficientes; y 1 instancia de `HalfBandFilterTwoTaps`, etapa de 7 coeficientes.



**Figura 4.2.1:** Diseño de bloques de los componentes

Para el diseño en VHDL se ha seguido una guía de estilo proporcionada por BSH[15] que ha resultado muy útil para escribir el código de manera limpia, funcional y comprensible. La guía de estilo define una nomenclatura unificada para las distintas señales según tipo y función y, por otro lado, establece la existencia de un archivo donde recoger todos los parámetros de configuración del componente (Ver Anexo C.1). En la tabla 4.2.1 se resumen las constantes del sistema que se pueden modificar.

Constante	Descripción	Tipo de dato	Rango
<code>C_S_AXI_DATA_WIDTH</code>	Tamaño de palabra del bus AXI en bits	Entero	Según especificación, en este caso 32
<code>C_S_SAMPLE_DATA_WIDTH</code>	Tamaño de palabra de los valores del ADC	Entero	Según codificación empleada
<code>C_S_COEFF_DATA_WIDTH</code>	Tamaño de palabra de los coeficientes en bits	Entero	Según codificación empleada
<code>PrescalerHB<math>n</math></code>	Frecuencia de trabajo del componente $n$ . Cada cuantos ciclos de reloj pone un dato válido el componente	Entero	Mínimo 1
<code>C_HB<math>n</math>_i</code>	Coficiente $i$ -ésimo codificado en coma fija del componente $n$	<code>signed(C_S_COEFF_DATA_WIDTH-1 downto 0)</code>	Definido por <code>C_S_COEFF_DATA_WIDTH</code>

**Tabla 4.2.1:** Parámetros modificables en el archivo de configuración

### 4.3. Protocolo de comunicación

Como se ha comentado, el bloque IP a diseñar debe ser compatible con el protocolo de comunicación AXI4-Stream. La transferencia de datos sobre un canal AXI4-Stream (Figura 4.3.1) se implementa mediante máquinas de estado en el esclavo (Figura 4.3.2b) y el maestro (Figura 4.3.2a). La señal `tvalid` la gestiona la fuente (maestro), y la señal `tready` la gestiona el receptor (esclavo). `tvalid` indica que los valores de datos de los campos (`tdata`, `tuser` y `tlast`) son válidos. En el diseño sólo se ha definido la señal `tdata`, ya que `tuser` y `tlast` son opcionales. `tready` indica que el receptor está preparado para recibir datos. Cuando ambas señales (`tvalid` y `tready`) están activas en un ciclo de reloj, tiene lugar una transferencia de datos. Las señales de reloj son `m_axis_aclk` para el maestro y `s_axis_aclk` para el receptor. Además, también se dispone de la señal `aresetn`, activa en bajo, para reiniciar de forma asíncrona el componente cuando está activa durante un ciclo de reloj.

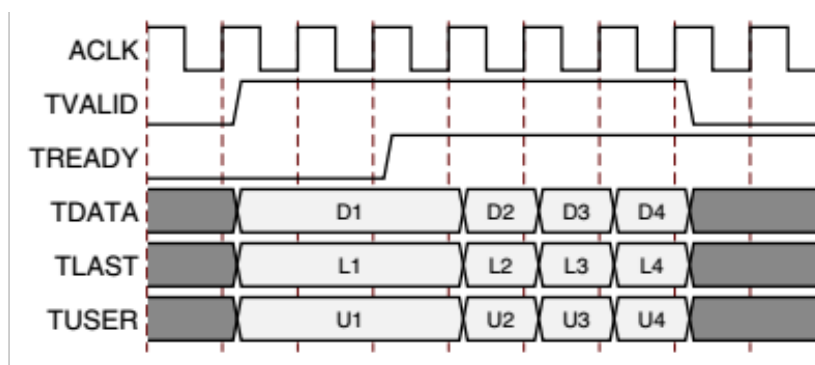
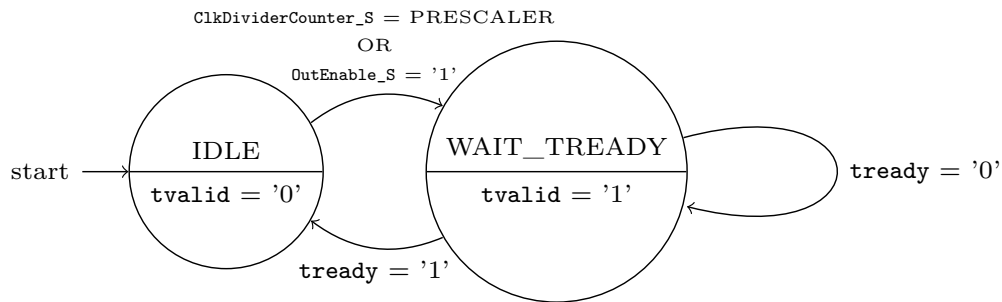


Figura 4.3.1: Transferencia de datos en un canal AXI4-Stream

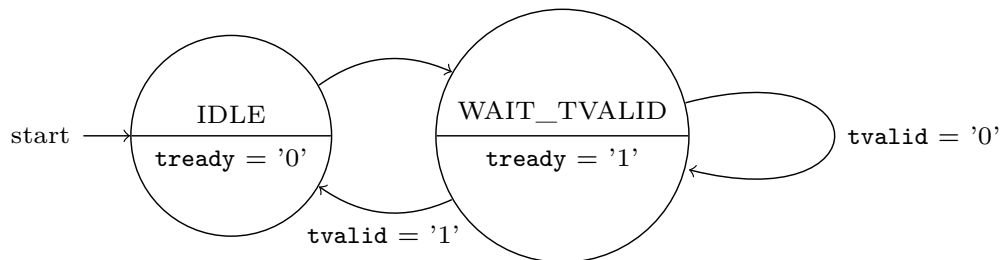
Del diagrama de tiempos de la figura 4.3.1 se define el funcionamiento de las máquinas de estado necesarias para cumplir con el protocolo de comunicación.

La parte emisora (Figura 4.3.2a) permanece en reposo hasta que: se habilita la señal `OutEnable`, que indica que ya se han realizado todas las operaciones y por tanto los datos están disponibles; o han transcurrido tantos ciclos de reloj como indique `PRESCALER` (frecuencia de trabajo del componente). Estas dos señales están sincronizadas de tal manera que el contador `ClkDividerCounter_S` se inicializa cuando se habilita la señal `OutEnable`. El maestro permanece en el estado `WAIT_TREADY` hasta que el esclavo habilita la señal `tready`.

En el extremo receptor del componente (Figura 4.3.2b) sólo se permanece un ciclo en el estado de reposo `IDLE`. El estado siguiente es esperar un dato válido (`WAIT_TVALID`), esto es, que el maestro habilite la señal `tvalid`.



(a) Máquina de estados MASTERAXI\_DECODE



(b) Máquina de estados SLAVEAXI\_DECODE

**Figura 4.3.2:** Máquinas de estado para la comunicación AXI4-Stream del emisor y el receptor

## 4.4. Implementación del bloque de filtrado

### Estructura del bloque de filtrado

El diseño seleccionado consta de tres etapas de filtrado, dos de ellas con el mismo número de coeficientes. Los filtros *Half-band* tienen una respuesta en frecuencia simétrica respecto de  $f_s/4$ . Esto implica que la suma de las bandas de paso y rechazo sumen  $f_s/2$ . Cuando el filtro tiene un número impar de coeficientes, esta simetría tiene la propiedad de que en el dominio del tiempo, los coeficientes impares son 0, salvo el central. Si el filtro tiene ganancia unidad, el valor del coeficiente central es 0.5. El diseño propuesto tiene dos filtros de 11 coeficientes y uno de 7 coeficientes. Además se trata de un filtro de fase lineal por lo que sus coeficientes son, además, simétricos respecto al centro. De forma general, los valores de un filtro *Half-band* de 11 coeficientes con ganancia unidad se distribuirán de la siguiente forma:

$$h[1] = h[3] = h[7] = h[9] = 0$$

$$h[5] = 0,5$$

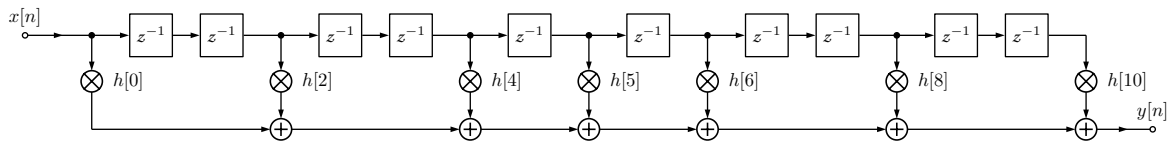
$$h[0] = h[10]$$

$$h[4] = h[8]$$

$$h[2] = h[6]$$

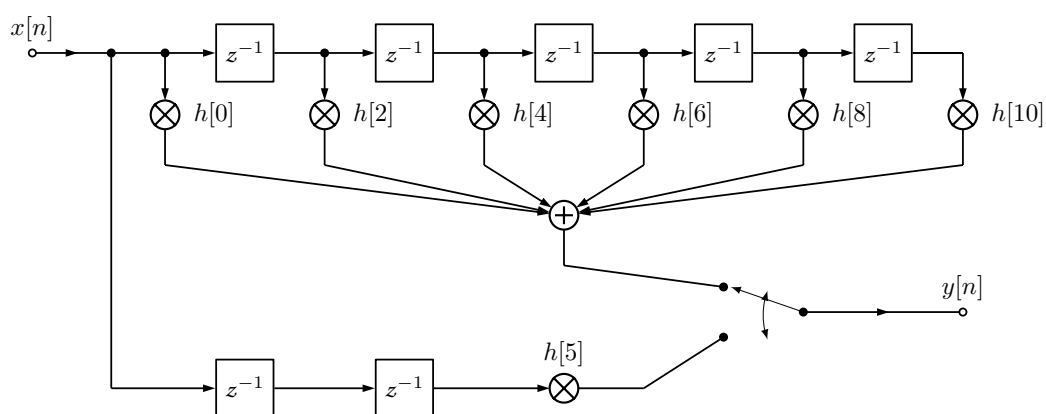


La implementación de este filtro en forma directa queda de la siguiente manera:



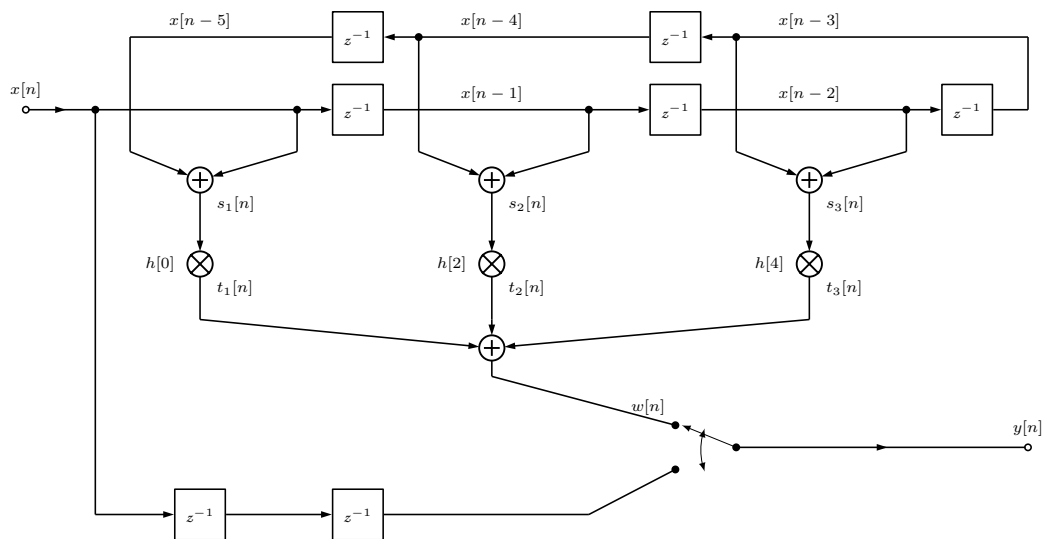
**Figura 4.4.1:** Estructura directa para el filtro *Half-band* de 11 coeficientes

Una implementación computacionalmente eficiente de un interpolador se puede conseguir realizando una descomposición polifase. Para el caso de una interpolación por un factor dos, se obtendrán dos ramas. Se observa en la figura 4.4.2 que debido a que los coeficientes son cero de forma alterna, excluyendo el coeficiente central, la segunda rama sólo tiene que realizar una multiplicación. Aun así, dicho coeficiente es de valor 0.5 si el filtro tiene ganancia unidad. Esto significa que en la implementación digital se puede sustituir por un desplazamiento. Si, como es el caso, el factor de interpolación es 2, se puede, o bien multiplicar la salida por 2 o bien, por tratarse de un sistema lineal, multiplicar los coeficientes por 2. Esto hace que ese coeficiente tenga valor 1, por lo que no sería necesaria ninguna operación en la rama inferior.



**Figura 4.4.2:** Implementación polifase para el filtro *Half-band* de 11 coeficientes

Adicionalmente, cuando una estructura en forma directa presenta simetría en los coeficientes, como es el caso, se puede simplificar su estructura (Figura 4.4.3) para reducir el número de operaciones.



**Figura 4.4.3:** Estructura simplificada de la implementación polifase para el filtro *Half-band* de 11 coeficientes

Con esta estructura se consiguen explotar dos propiedades: la simetría de coeficientes y el uso de mínima memoria. En la tabla 4.4.1 se resumen los requisitos de cada filtro.

Número de coeficientes	Coefficientes en memoria	Línea de retardo	Multiplicaciones	Sumas
11	3	5	3	5
7	2	3	2	3

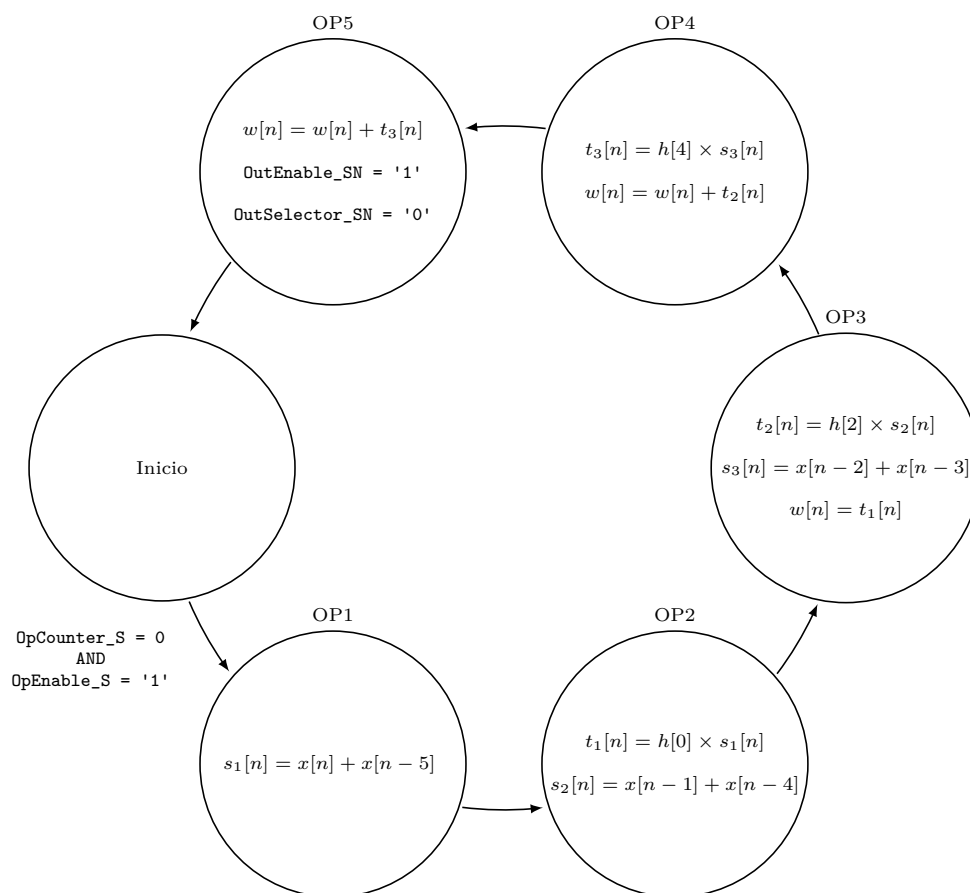
**Tabla 4.4.1:** Resumen de las necesidades de filtros *Half-band* para su implementación

## Planificador de operaciones

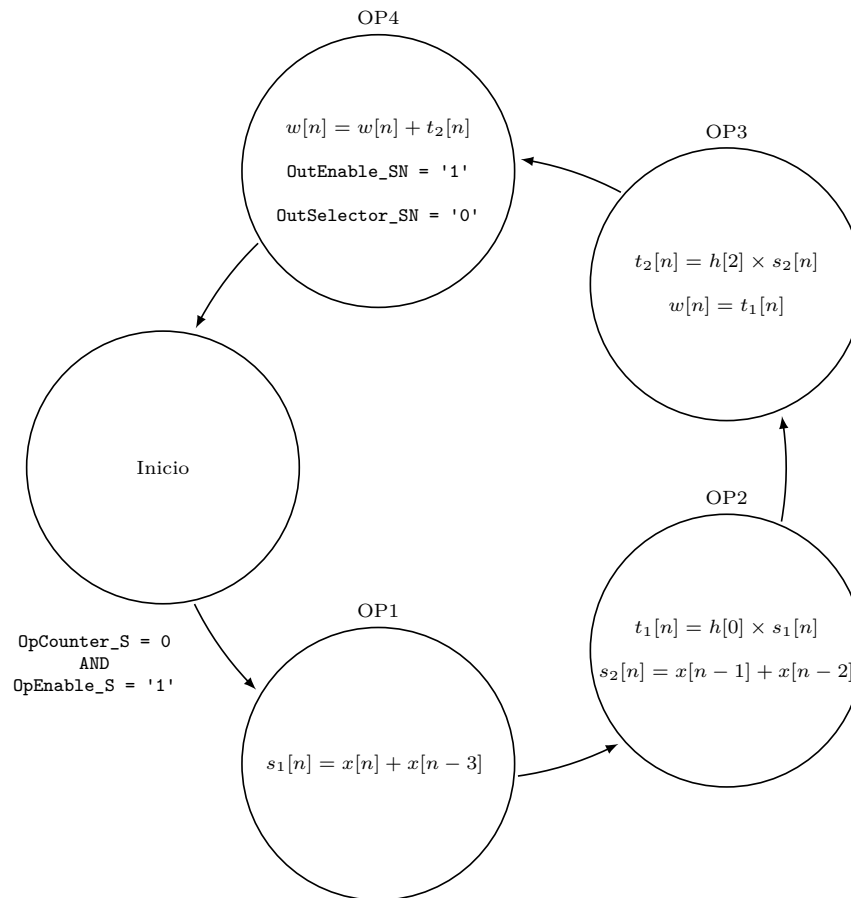
Para la implementación se ha diseñado un planificador para que en el momento de realizar las operaciones sobre las muestras de entrada sólo sea necesario utilizar un multiplicador y dos acumuladores por componente. Esto es posible porque se disponen de ciclos de reloj suficientes para realizar todos los cálculos por separado. Con esta planificación se consigue que el componente en conjunto sólo emplee 3 multiplicadores. Este planificador se modela con un secuenciador de operaciones (`OpCounter_S`) y una señal de activación (`OpEnable_S`). A su vez, cuando ha terminado de realizar las operaciones, se habilita la señal `OutEnable_S` empleada por la máquina de estados `MASTERAXI_DECODE` (Figura 4.3.2a) durante un ciclo de reloj y cambia el selector de salida con la señal `OutSelector_S`. El selector de salida cambiará de valor nuevamente cuando el divisor de reloj `ClkDividerCounter_S` alcance el valor definidor por `PRESCALER`.

En la figura 4.4.4 se representan los ciclos de operaciones de los secuenciadores para filtros de 11 y 7 coeficientes respectivamente. Los ciclos representados en la figura 4.4.4a para el filtro de 11 coeficientes se apoyan en las señales definidas en las ramas de la figura 4.4.3. En cada uno de los círculos se agrupa un conjunto de operaciones por ciclo de reloj. En cada ciclo de reloj se calculan valores que son usados en el ciclo siguiente hasta que se completan todas las operaciones. Para el planificador del filtro de 7 coeficientes (Figura 4.4.4b), el desarrollo es análogo eliminando el ciclo OP3 del planificador de 11 coeficientes.

El diseño de estos planificadores ha resultado muy versátil puesto que aumentar o reducir el número de coeficientes tan sólo exige modificar el diseño del componente de manera localizada (Ver Anexo C.3).



(a) Planificador de operaciones para un filtro *Half-band* de 11 coeficientes

(b) Planificador de operaciones para un filtro *Half-band* de 7 coeficientes**Figura 4.4.4:** Planificadores de operaciones de los filtros de 11 y 7 coeficientes

### Codificación de la señal

La señal se obtiene de un ADC LTC2315 de Linear de 12 bits[9] con un rango dinámico de 0 a 3,3 V. La precisión de la codificación no podía ser, en ningún caso, superior a 12 bits.

La codificación de un número real en coma fija se definirá mediante la notación *Qm.n*. Esta notación hace referencia a un número entero en complemento a dos, por lo que un bit de signo se asume de forma implícita. *m* representa el número de bits de la parte entera y *n* el número de bits de la parte fraccionaria.

La codificación de un número real  $U_R$  en formato  $Qm.n$ , con un tamaño de palabra total  $w$ , error de cuantificación  $\epsilon$  y cuyo valor como número entero es  $\hat{U}$ , se realiza de la siguiente manera:

$$m = \text{ceil}(\log_2(\text{abs}(U_R))) \quad (4.4.1)$$

$$n = w - 1 - m \quad (4.4.2)$$

$$\hat{U} = \text{round}(U_R \times 2^n) \quad (4.4.3)$$

$$\epsilon = \hat{U} \times 2^{-n} - U_R \quad (4.4.4)$$

Si se quiere representar la señal en coma fija con signo con toda la precisión, esta se debe codificar en formato Q0.11. Se sabe que la ganancia del filtro de medida de la corriente,  $G_{filt,iL}$  es de 0,116 [5, p. 69]. Las medidas obtenidas directamente del osciloscopio tienen unidades del S.I. (sistema internacional), por lo tanto, para codificarlas en formato Q0.11 se realiza la siguiente operación, donde  $V_{ADC}$  es igual a 3,3V:

$$i_L = \text{round}(i_L[A] \cdot 2^{11} \cdot \frac{G_{filt,iL}}{V_{ADC}}) \quad (4.4.5)$$

### Codificación de los coeficientes

Para realizar una implementación digital del filtro diseñado se precisa codificar los coeficientes en un formato de coma fija a partir de los obtenidos en coma flotante. Por motivos de simplicidad se ha optado por codificar todos los coeficientes con el formato Q0.15. Esto son 16 bits de palabra repartidos en 1 bit de signo y 15 bits decimales. Una vez validada la implementación con esta codificación se puede ajustar el número de bits empleados para optimizar la síntesis del diseño si fuera necesario.

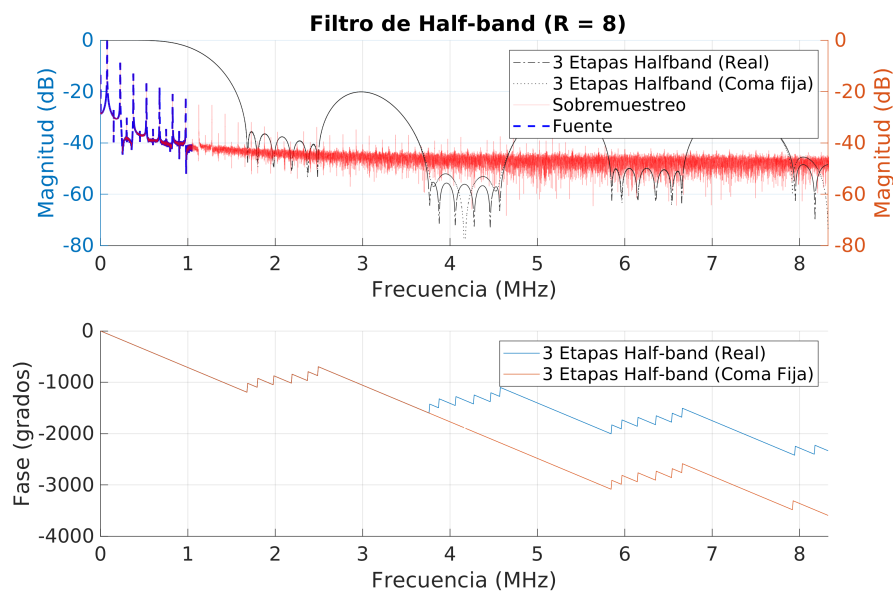
En un proceso de interpolación, el filtro debe tener la ganancia del factor de interpolación en unidades lineales, en este caso 2. Esto es debido a que el proceso de interpolación atenúa la señal de entrada en ese mismo factor si el filtro tiene ganancia unidad. Al tratarse de un sistema LTI (lineal e invariante en el tiempo), lo que se hará es escalar el valor de los coeficientes por un factor 2 como se ha comentado al principio de esta sección.

En la tabla 4.4.2 se muestran los coeficientes del filtro y su codificación en formato Q0.15. El máximo error cometido en la codificación propuesta es de  $1.4960e-05$ .

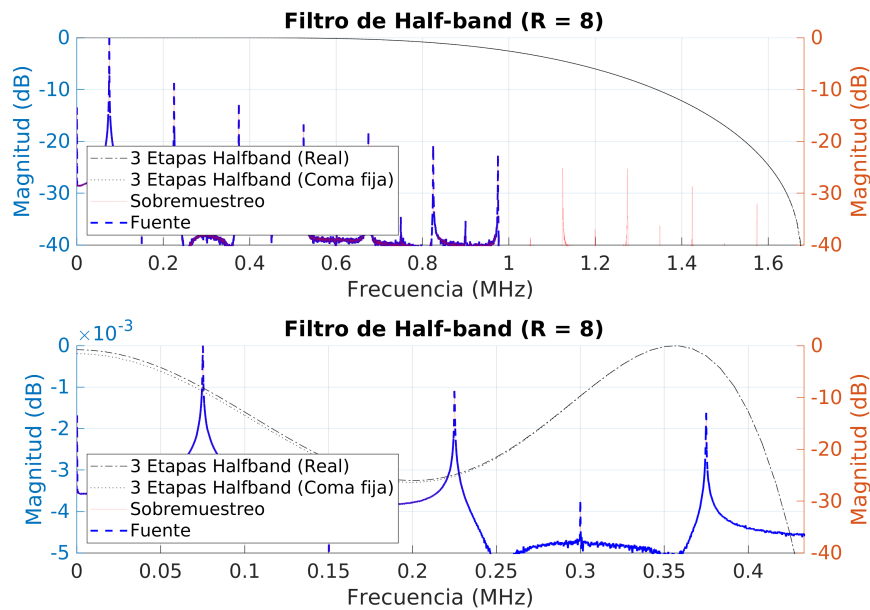
		Coeficiente real	Coeficiente en coma fija	Q0.15	$\epsilon$
HB1	h(0)	0.019424139930736	0.019409179687500	636	$1.4960e-05$
	h(2)	-0.116242132913476	-0.116241455078125	-3809	$6.7784e-07$
	h(4)	0.597186948348207	0.597198486328125	19569	$1.1538e-05$
HB2	h(0)	0.013266313966274	0.013275146484375	435	$8.8325e-06$
	h(2)	-0.102070145774946	-0.102081298828125	-3345	$1.1153e-05$
	h(4)	0.588808754948819	0.588806152343750	19294	$2.6026e-06$
HB3	h(0)	-0.063669188505992	-0.063659667968750	-2086	$9.5205e-06$
	h(2)	0.563654819319767	0.563659667968750	18470	$4.8486e-06$

**Tabla 4.4.2:** Codificación de los coeficientes de los filtros diseñados en coma fija

En figura 4.4.5 se muestra la comparación de la respuesta en frecuencia y fase del filtro resultante de las tres etapas en cascada con coeficientes reales y con los coeficientes codificados en coma fija. Se observa que el error cometido es muy pequeño y las respuestas al impulso prácticamente idénticas.



**(a)** Comparación de la respuesta en fase y frecuencia del filtro *Half-band* del escenario 2 con coeficientes reales y en codificados en coma fija



(b) Detalle de la comparación de la respuesta en fase y frecuencia del filtro *Half-band* del escenario 2 con coeficientes reales y codificados en coma fija

**Figura 4.4.5:** Filtro *Half-band* del escenario 2

Uno de los problemas que puede aparecer en las implementaciones digitales en coma fija es el desbordamiento. Para evitar el desbordamiento en el diseño, se debe calcular el tamaño de palabra a utilizar en cada señal que sea resultado de una operación, para asegurar que el resultado se puede representar con ese número de bits. Se ha comentado que los filtros tienen ganancia 2 para que la amplitud de la señal a la salida sea la misma que a la entrada. También se sabe que el ADC tiene una resolución de 12 bits sin signo y que por ello se va a codificar la señal de entrada y salida como un número con signo de 11 bits, esto es, con formato Q0.11. Esta misma codificación se utilizará para los elementos de la línea de retardo. Además se ha especificado que los coeficientes del filtro estarán codificados en formato Q0.15.

En las figuras 4.4.6 y 4.4.6 se pueden ver cuantos bits se necesitan para no tener desbordamiento en cada una de las operaciones para los filtros de 11 y 7 coeficientes respectivamente. De la imagen, cabe resaltar que el resultado de las multiplicaciones duplica el bit de signo por lo que son suficientes 28 bits para almacenar el resultado. Por otro lado el sumatorio que se guarda en  $w[n]$  dependerá del número de ramas a sumar según:

$$\max(\text{número\_bits}(t_i[n])) + \text{ceil}(\log_2(\text{número\_sumas}))$$

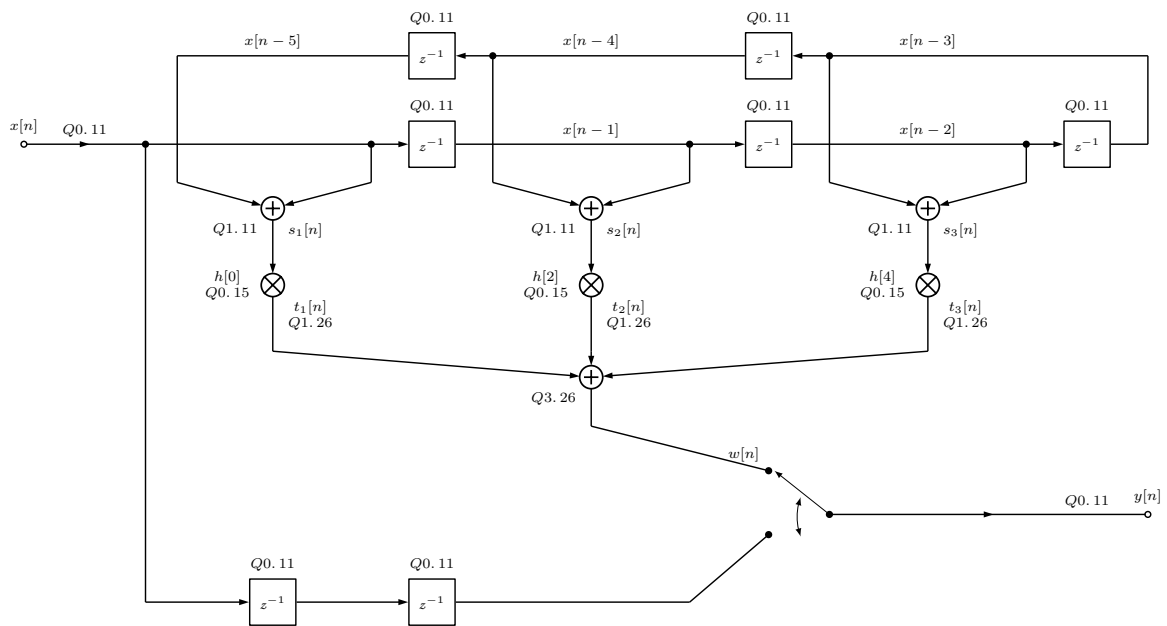


Figura 4.4.6: Codificación en coma fija de los valores del filtro de 11 coeficientes

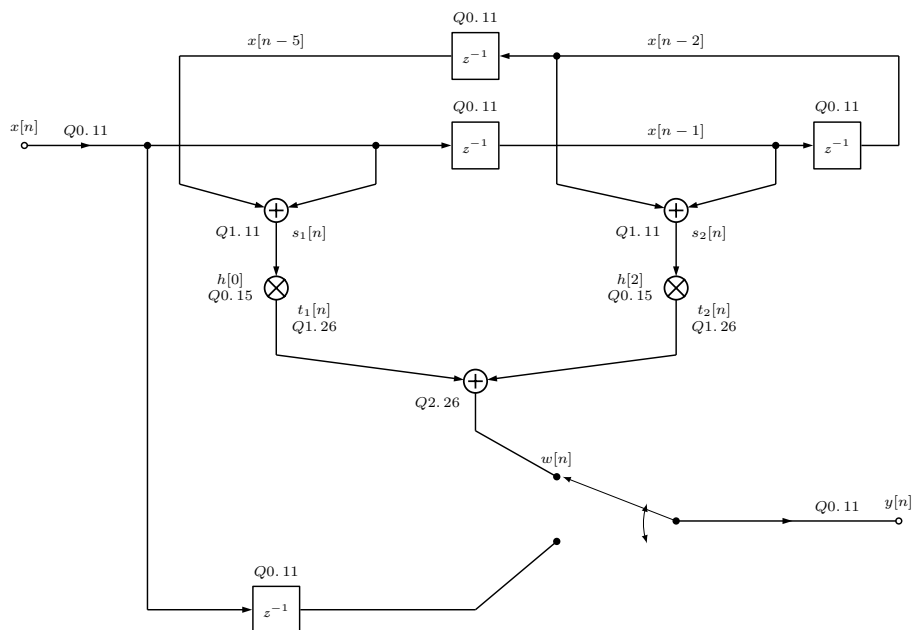


Figura 4.4.7: Codificación en coma fija de los valores del filtro de 7 coeficientes

Aún así, esto no garantiza la ausencia de desbordamiento pero existen diferentes maneras de mitigarlo o minimizarlo. Una de ellas es hacer que el filtro sature al máximo valor posible manteniendo el signo. Esto reduce el error comparado con devolver el valor desbordado al rango útil usando aritmética de módulo pero requiere más hardware o instrucciones[16]. Otra opción es realizar un escalado de los coeficientes del filtro, reduciendo su ganancia; o escalar la señal. El primer caso no es conveniente puesto



que la ventaja que proporciona que el coeficiente central sea 1 se perdería. Otra opción sería ajustar la ganancia del filtro ganancia del filtro de medida de la corriente para obtener menor amplitud de señal.

Un método general de obtener los factores de escala que permiten prevenir el desbordamiento son una clase de medidas vectoriales llamadas normas. Una norma es únicamente una medida de longitud vectorial [16, p. 96]. Se define la norma  $L_p$  para una secuencia de longitud N como

$$\|h[n]\|_p = \left( \sum_{n=-\infty}^{\infty} |h[n]|^p \right)^{1/p}$$

En la tabla 4.4.3 se indican los factores de escalado según las normas  $L_2$ ,  $L_{\infty\omega}$  y  $L_{1t}$ , de la más restrictiva a la menos restrictiva.

	Menos restrictiva	$\leq$ ... $\leq$	Más restrictiva
	$L_2$	$L_{\infty\omega}$	$L_{1t}$
	$s = \frac{1}{\max \ h[n]\ _2}$	$s = \frac{1}{\max  H(e^{j\omega}) }$	$s = \frac{1}{\max \sum_{n=-\infty}^{\infty}  h[n] }$
HB1	1.515730268368723	0.999607808889516	0.811129263824942
HB2	1.527394575907516	1	0.830452633179583
HB3	1.560059768783289	0.999971778483766	0.887060097455333

**Tabla 4.4.3:** Factores de escalado de las distintas etapas para la prevención de desbordamiento [16, p. 96]

- $L_2$  Proporciona la mejor SNR (relación señal a ruido) de las tres si no hay desbordamiento.
- $L_{\infty\omega}$  Sinusoides a escala completa no sufren desbordamiento nunca, pero otras señales puede que sí.
- $L_{1t}$  Se mitiga por completo el desbordamiento, pero produce la peor SNR de las tres escalas.

# Capítulo 5

## Simulación y validación

### 5.1. Implementación en MATLAB

Se ha realizado una primera implementación en MATLAB con la señal y los coeficientes convertidos en coma flotante después de haberlos codificado en coma fija.

#### 5.1.1. Validación de la interpolación

En la tabla 5.1.1 se muestran los resultados obtenidos de la interpolación. Se puede comprobar que los objetivos se cumplen y se valida así la implementación.

	Media		$\sigma$		min.		máx.	
	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz	35 kHz	75 kHz
$e_{ilrms}$	0.102	0.153	-	-	-	-	-	-
$e_{pavg}$	0.100	0.137	-	-	-	-	-	-
$i\hat{l}_{pk,H}$	0.075	0.448	0.047	0.261	0.001	0.003	0.168	0.915
$i\hat{l}_{pk,L}$	0.078	0.419	0.049	0.256	0.004	0.065	0.175	1.009
$i\hat{o}_{ff,H}$	0.624	0.601	0.328	0.321	0.073	0.029	1.188	1.331
$i\hat{o}_{ff,L}$	0.549	0.575	0.182	0.332	0.235	0.044	1.039	1.286
$t_{snb,H}$ (ns)	0.000	-5.625	0.000	17.769	0.000	-60.000	0.000	0.000
$t_{snb,L}$ (ns)	0.000	-11.250	0.000	23.793	0.000	-60.000	0.000	0.000
$t_{d,H}$ (ns)	-5.625	-1.875	17.769	32.273	-60.000	-60.000	0.000	60.000
$t_{d,L}$ (ns)	16.875	18.750	27.408	28.256	0.000	0.000	60.000	60.000

**Tabla 5.1.1:** Errores con filtros *Half-band* en cascada de la implementación en coma fija con MATLAB

## 5.2. Implementación en VHDL

Se ha realizado una implementación en VHDL de la arquitectura especificada en el capítulo anterior. Con el fin de validarla se ha diseñado un banco de pruebas, mediante el cual se pretenden simular y validar los procesos de comunicación e interpolación.

### 5.2.1. Simulación

Para la simulación se han declarado tres procesos:

#### Proceso de reloj (`clk_proc`)

Este proceso genera una señal de reloj de 100 MHz

#### Proceso de lectura (`adc_proc`)

Se simula el muestreo realizado por el ADC mediante la lectura de un fichero generado con MATLAB que contiene la señal de entrada codificada en formato Q0.11, con periodo `ADC_PERIOD` igual a 480ns, es decir, a una tasa de  $\frac{100}{48}$  Mhz. Estos datos se envían al componente `MultiStageHalfBandInterpolator` mediante el protocolo AXI4-Stream.

#### Proceso de escritura (`read_proc`)

Este proceso está conectado a la salida del componente `MultiStageHalfBandInterpolator` y actúa como un esclavo del protocolo AXI4-Stream. Lee los datos a la tasa marcada por el `PRESCALER` de la última etapa y los escribe en un fichero en formato Q0.11 para su posterior procesado. En este caso la tasa de recepción serán  $\frac{100}{6}$  Mhz.

### 5.2.2. Validación del protocolo de comunicación

En las figuras 5.2.1, 5.2.2, 5.2.3 y 5.2.4 se visualiza el cumplimiento de la especificación AXI4-Stream en todo el componente. En las figuras se visualizan las señales de entrada y salida del extremo receptor de la comunicación (esclavo) de las etapas internas y de la salida del componente. En las imágenes se pueden localizar fácilmente todas las señales involucradas en el protocolo AXI4-Stream introducidas en la sección 4.3 (`aclk`, `aresetn`, `tdata`, `tvalid` y `tready`) y se puede validar el correcto funcionamiento del protocolo de comunicación. Adicionalmente, se puede medir el

tiempo transcurrido entre dos activaciones de `tvalid` para calcular la tasa de entrada de datos a cada una de las etapas y por lo tanto validar que el sistema esta trabajando según lo especificado. Se observa que los datos entran a la primera etapa con un periodo de 480 ns ( $\frac{100}{48}$  MHz) igual a  $f_{s,ADC,2}$ . Salen de la misma y entran a la segunda etapa con un periodo de 240 ns ( $\frac{100}{24}$  MHz). De esta etapa entran en la primera con un periodo de 120 ns ( $\frac{100}{12}$  MHz). Por último la salida del componente envía muestras con una tasa de 60 ns ( $\frac{100}{6}$  MHz) que es la frecuencia de destino esperada  $f_{s,dst}$ .

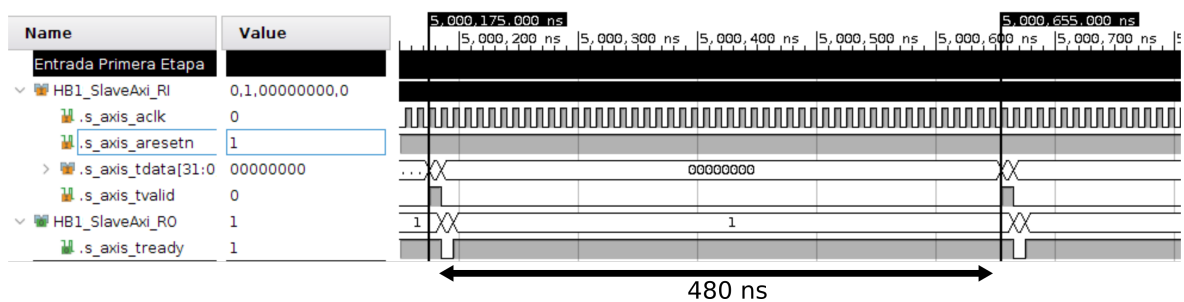


Figura 5.2.1: Verificación del protocolo de comunicación en la primera etapa

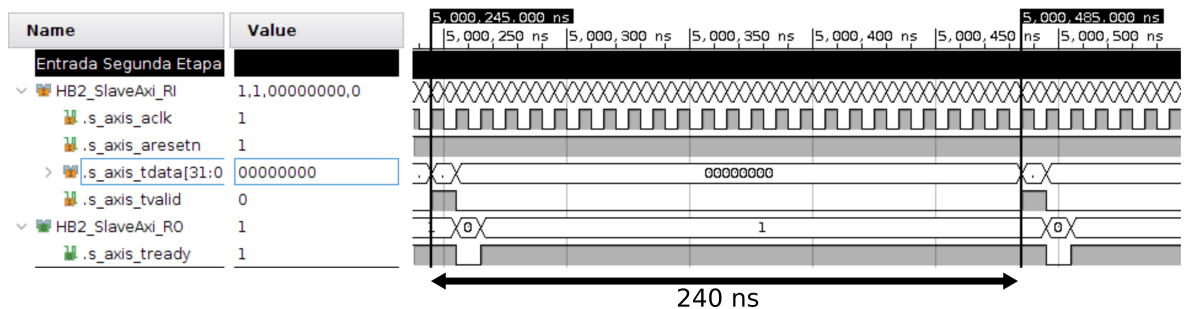


Figura 5.2.2: Verificación del protocolo de comunicación en la segunda etapa

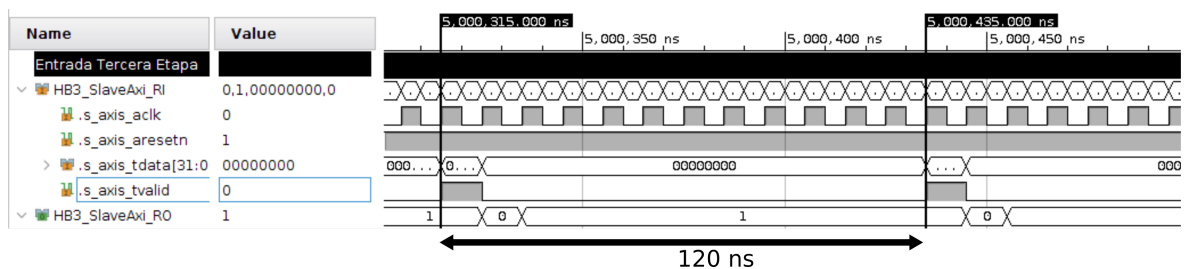


Figura 5.2.3: Verificación del protocolo de comunicación en la tercera etapa

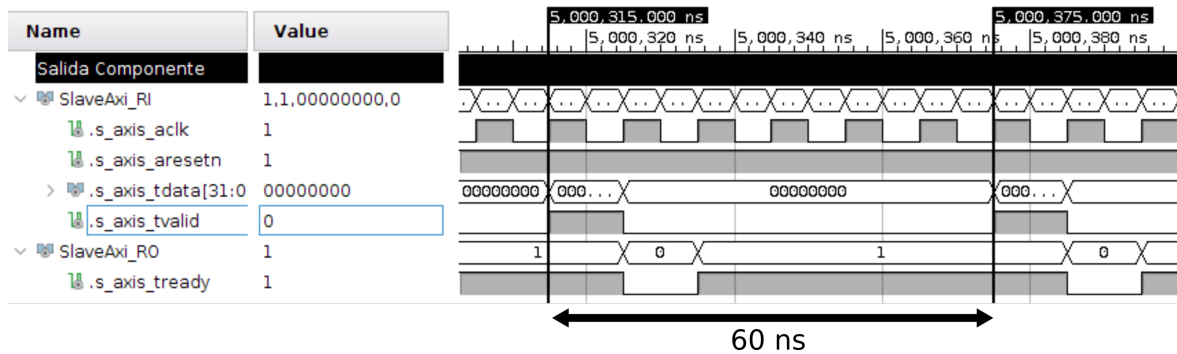


Figura 5.2.4: Verificación del protocolo de comunicación en la salida del componente

### 5.2.3. Validación de la interpolación

Como se ha comentado al inicio de esta sección, se ha realizado la lectura de los valores de la señal codificados en formato Q0.11 y se han escrito los resultados en otro fichero con el mismo formato.

Desde MATLAB se ha procedido a la lectura de los mismos y se han convertido al S.I. para poder realizar una comparación con las medidas del osciloscopio. Los resultados obtenidos se muestran en las tablas 5.2.1 y 5.2.2, para 35 KHz y para 75 KHz respectivamente. En dichas tablas se comparan los resultados obtenidos con el filtro del diseño y su implementación en VHDL. Como era de esperar los errores han aumentado pero, en general, el resultado es bueno. Por otro lado, el retraso calculado entre la señal de entrada y la generada por el sistema de filtros es de 34 muestras (2040 ns) que es aproximadamente el calculado de forma teórica (2340 ns).

	Media		$\sigma$		min.		máx.	
	MATLAB	VHDL	MATLAB	VHDL	MATLAB	VHDL	MATLAB	VHDL
$e_{ilrms}$	0.066	0.097	-	-	-	-	-	-
$e_{pavg}$	0.063	0.364	-	-	-	-	-	-
$i_{lpk,H}$	0.062	0.103	0.044	0.083	0.003	0.006	0.202	0.317
$i_{lpk,L}$	0.046	0.305	0.030	0.092	0.001	0.147	0.143	0.474
$i_{off,H}$	0.661	0.862	0.335	0.390	0.019	0.016	1.261	1.532
$i_{off,L}$	0.586	0.389	0.176	0.190	0.297	0.008	1.002	0.851
$t_{snb,H}(ns)$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$t_{snb,L}(ns)$	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$t_{d,H}(ns)$	-5.625	-3.750	17.769	21.213	-60.000	-60.000	0.000	60.000
$t_{d,L}(ns)$	18.750	-0.000	28.256	21.553	0.000	-60.000	60.000	60.000

Tabla 5.2.1: Comparación de la señal de 35 kHz entre las simulaciones en MATLAB y VHDL

	Media		$\sigma$		min.		máx.	
	MATLAB	VHDL	MATLAB	VHDL	MATLAB	VHDL	MATLAB	VHDL
$e_{i_{rms}}$	0.119	0.174	-	-	-	-	-	-
$e_{p_{avg}}$	0.098	2.344	-	-	-	-	-	-
$i_{lpk,H}$	0.445	0.738	0.266	0.381	0.003	0.079	0.829	1.493
$i_{lpk,L}$	0.416	0.284	0.279	0.180	0.010	0.025	0.855	0.713
$i_{off,H}$	0.638	1.192	0.297	0.520	0.102	0.100	1.103	2.101
$i_{off,L}$	0.607	0.236	0.321	0.187	0.070	0.019	1.187	0.862
$t_{snb,H}(\text{ns})$	-9.375	-18.750	22.134	28.256	-60.000	-60.000	0.000	0.000
$t_{snb,L}(\text{ns})$	-7.500	3.750	20.161	14.756	-60.000	0.000	0.000	60.000
$t_{d,H}(\text{ns})$	-3.750	16.875	37.135	40.991	-60.000	-60.000	60.000	60.000
$t_{d,L}(\text{ns})$	7.500	-15.000	20.161	26.396	0.000	-60.000	60.000	0.000

**Tabla 5.2.2:** Comparación de la señal de 75 kHz entre las simulaciones en MATLAB y VHDL

#### 5.2.4. Validación de la implementación

La plataforma donde se va a probar esta implementación cuenta con una unidad de procesamiento y control integrada en System on Module TE0720 de Trenz Electronic. Este modelo está basado en un System on Chip Zynq-7000, que incluye la lógica programable y 2 procesadores ARM Cortex A9, para el software de alto nivel. Por lo que el proyecto se ha configurado para un dispositivo del tipo XC7Z020.

Una vez validado el diseño se procede a su síntesis e implementación. La una restricción temporal que se ha creado es para un reloj de 100 MHz. El peor tiempo negativo de *slack* (WNS, del inglés *Worst Negative Slack*) obtenido es de 3.085 ns y el requisito de tiempo es de 10 ns (1/100 Mhz). Un valor positivo del WNS quiere decir que se cumplen las restricciones temporales. A continuación se muestra un extracto del informe de implementación generado en Vivado con el que se comprueba que no se han generado *latches* y que solo se emplea un multiplicador y por tanto un solo DSP48 por componente, lo que valida la arquitectura del planificador.

```

+-----+-----+-----+-----+
| Site Type | Used | Fixed | Available | Util% |
+-----+-----+-----+-----+
| Slice LUTs | 475 | 0 | 53200 | 0.89 |
| LUT as Logic | 475 | 0 | 53200 | 0.89 |
| LUT as Memory | 0 | 0 | 17400 | 0.00 |
| Slice Registers | 432 | 0 | 106400 | 0.41 |
| Register as Flip Flop | 432 | 0 | 106400 | 0.41 |
| Register as Latch | 0 | 0 | 106400 | 0.00 |
+-----+-----+-----+-----+

+-----+-----+-----+-----+
| Site Type | Used | Fixed | Available | Util% |
+-----+-----+-----+-----+
| DSPs | 3 | 0 | 220 | 1.36 |
| DSP48E1 only | 3 | | | |
+-----+-----+-----+-----+

```

# Capítulo 6

## Conclusiones

Durante el proceso de diseño se ha experimentado con diferentes sistemas de filtrado FIR. Además se han propuesto dos escenarios adicionales cuyo objetivo era reducir la frecuencia de muestreo del ADC. Tras validar los resultados obtenidos en la fase de diseño se ha podido comprobar como los errores obtenidos por filtros interpoladores con coeficientes del polinomio de Lagrange situaban un umbral inferior para el caso de estudio en cuestión. Igual que en trabajos anteriores[5], se han diseñado filtros FIR óptimos. Se ha conseguido poca mejora en este aspecto respecto al trabajo citado. Por último se han estudiado filtros *Half-band* para los dos escenarios adicionales. Dichos escenarios tienen la ventaja de que el factor de interpolación requerido en los dos casos son potencias de dos, 8 y 16. Esto ha hecho posible la utilización de filtros *Half-band* en una implementación multietapa. Este tipo de configuración es muy versátil y es utilizada en multitud de aplicaciones. Con un reducido número de coeficientes se consiguen prestaciones similares a la interpolación con filtros FIR óptimos. Además, el número de operaciones es muy reducido y los filtros realizan las operaciones a diferentes frecuencias.

Se ha decidido realizar la implementación multietapa del segundo escenario con un factor de interpolación de 8. Esto reduce la frecuencia de muestreo del ADC un 25 % lo que puede conllevar un abaratamiento del componente. Además se ha optado por una implementación desacoplada permitiendo la recombinación de componentes de hasta 12 coeficientes, y no sería difícil crear bloques que soporten más coeficientes si fuera necesario. La etapa propuesta consta de tres componentes conectados en cascada. Esto puede permitir en el futuro un componente reconfigurable en función de la frecuencia de conmutación del sistema, reutilizando etapas o extraer muestras a distintas frecuencias de muestreo.

## 6.1. Lineas futuras

Se podría seguir por la misma línea creando un bloque reconfigurable con bancos de filtros *Half-band* en función de la frecuencia de muestreo. También se puede trabajar más a fondo en el tercer escenario ya que se ha demostrado su viabilidad.

En función de la disponibilidad de recursos o espacio se podría realizar una implementación con aritmética distribuida para eliminar los multiplicadores, o realizar un componente conjunto con un sistema de planificación de operaciones más complejo.

Se ha pensado que dada la tipología de la señal podría ser viable utilizar métodos de predicción para poder actuar en tiempo real, puesto que es una señal de la que se conocen sus parámetros estadísticos.



# Bibliografía

- [1] E. Meijering, “A chronology of interpolation: from ancient astronomy to modern signal and image processing,” *Proceedings of the IEEE*, vol. 90, pp. 319–342, March 2002.
- [2] R. W. Schafer and L. R. Rabiner, “A digital signal processing approach to interpolation,” *Proceedings of the IEEE*, vol. 61, pp. 692–702, June 1973.
- [3] R. Lyons, *Understanding Digital Signal Processing*. Prentice Hall, 2011.
- [4] H. Sarnago, O. Lucía, A. Mediano, and J. Burdío, “Analytical model of the half-bridge series resonant inverter for improved power conversion efficiency and performance,” *IEEE Transactions on Power Electronics*, vol. 30, pp. 4128–4143, Aug 2015.
- [5] A. Barrutia Adrián, “Current estimation in an induction cooktop by means of digital processing of the resonance capacitor voltage,” trabajo fin de máster, Universidad de Zaragoza, 2018.
- [6] O. Lucía, J. M. Burdío, I. Millan, J. Acero, and S. Llorente, “Efficiency optimization of half-bridge series resonant inverter with asymmetrical duty cycle control for domestic induction heating,” *2009 13th European Conference on Power Electronics and Applications*, pp. 1–6, 2009.
- [7] Ó. Jiménez Navascués, “Measurement and system identification tools for power converters applied to domestic induction heating systems,” trabajo fin de máster, Universidad de Zaragoza, 2013.
- [8] J. Villa, “Plataforma electrónica para la investigación de estrategias de adquisición, procesamiento y control de potencia en cocinas de inducción,” trabajo fin de máster, Universidad de Zaragoza, 2016.
- [9] L. T. Corporation, “LTC2315-12: 12-Bit, 5MSPS Serial Sampling ADC in TSOT,” 2013.

- 
- [10] A. M. López, “Código fuente de MultiStageHalfBandInterpolator (MATLAB y VHDL).” <https://github.com/almul0/MultiStageHalfBandInterpolator>, 2018.
- [11] J. O. Smith, *Physical Audio Signal Processing: for Virtual Musical Instruments and Audio Effects*. W3K Publishing, 2010.
- [12] R. Lyons G., “Optimizing the Half-band Filters in Multistage Decimation and Interpolation.” "<https://www.dsprelated.com/showarticle/903.php>", December 16, 2018.
- [13] L. T. Corporation, “LTC2313-12: 12-Bit, 2.5MSPS Serial Sampling ADC in TSOT,” 2015.
- [14] Xilinx, “AXI Reference Guide,” 2017.
- [15] BSH ELECTRODOMÉSTICOS ESPAÑA S.A., “VHDL Guidelines for IP Design.” Comunicación privada, 2018.
- [16] W. T. Padgett and D. V. Anderson, *Fixed-Point Signal Processing*. Synthesis Lectures on Signal Processing, Morgan & Claypool Publishers, 2009.
- [17] S. Mitra, *Digital Signal Processing*. McGraw-Hill Higher Education, 2001.
- [18] H. G. Gockler, “Most efficient digital filter structures: The potential of halfband filters in digital signal processing,” in *Applications of Digital Signal Processing*, InTech, 2011.

# Lista de Figuras

1.1.1. Interpolación: (a) secuencia original; (b) secuencia interpolada por $L = 3$ ; [3, p. 581]. . . . .	2
1.1.2. Interpolación por cuatro: (a) secuencia original muestreada y su espectro; (b) ceros insertados en la secuencia original y su espectro; (c) secuencia resultado del filtro interpolador y espectro resultante [3, p. 582].	2
1.1.3. Notación funcional de la interpolación [3, p. 581]. . . . .	3
1.2.1. Inversor resonante serie en semipunto resonante con dos condensadores de resonancia [5, p. 14]. . . . .	3
1.2.2. Formas de onda del inversor resonante serie en semipunto operando bajo condición ZVS [7, p. 244]. . . . .	4
1.2.3. Parámetros temporales relevantes para las cocinas de inducción relativos a la corriente [5, p. 26]. . . . .	4
2.0.1. Señales de $v_{bus}$ e $i_L$ tomadas con el osciloscopio. . . . .	8
2.0.2. Espectro de la señal $i_L$ del osciloscopio. . . . .	9
2.0.3. Parámetros temporales extraídos de la señal de destino. . . . .	10
3.1.1. Espectro de la corriente para $f_{s,ADC,1}$ . . . . .	14
3.1.2. Filtros de Lagrange del escenario 1 . . . . .	15
3.1.3. Filtro óptimo del escenario 1 . . . . .	17

3.2.1. Espectro de la corriente para $f_{s,ADC,2}$ . . . . .	19
3.2.2. Filtros de Lagrange del escenario 2 . . . . .	20
3.2.3. Filtro óptimo del escenario 2 . . . . .	22
3.2.4. Esquema funcional de la interpolación multietapa basada en filtros <i>Half-band</i> [12]. . . . .	23
3.2.5. Etapas de filtrado <i>Half-band</i> del escenario 2 . . . . .	25
3.2.6. Filtro <i>Half-band</i> del escenario 2 . . . . .	26
3.3.1. Espectro de la corriente para $f_{s,ADC,3}$ . . . . .	28
3.3.2. Filtros de Lagrange del escenario 3 . . . . .	29
3.3.3. Filtro óptimo del escenario 3 . . . . .	31
3.3.4. Etapas de filtrado <i>Half-band</i> del escenario 3 . . . . .	34
3.3.5. Filtro <i>Half-band</i> del escenario 3 . . . . .	35
4.2.1. Diseño de bloques de los componentes . . . . .	39
4.3.1. Transferencia de datos en un canal AXI4-Stream . . . . .	40
4.3.2. Máquinas de estado para la comunicación AXI4-Stream del emisor y el receptor . . . . .	41
4.4.1. Estructura directa para el filtro <i>Half-band</i> de 11 coeficientes . . . . .	42
4.4.2. Implementación polifase para el filtro <i>Half-band</i> de 11 coeficientes . . . . .	42
4.4.3. Estructura simplificada de la implementación polifase para el filtro <i>Half-band</i> de 11 coeficientes . . . . .	43
4.4.4. Planificadores de operaciones de los filtros de 11 y 7 coeficientes . . . . .	45
4.4.5. Filtro <i>Half-band</i> del escenario 2 . . . . .	48
4.4.6. Codificación en coma fija de los valores del filtro de 11 coeficientes . . . . .	49

---

4.4.7.Codificación en coma fija de los valores del filtro de 7 coeficientes . . .	49
5.2.1.Verificación del protocolo de comunicación en la primera etapa . . . . .	53
5.2.2.Verificación del protocolo de comunicación en la segunda etapa . . . . .	53
5.2.3.Verificación del protocolo de comunicación en la tercera etapa . . . . .	53
5.2.4.Verificación del protocolo de comunicación en la salida del componente	54

# Lista de Tablas

2.0.1. Características de los escenarios propuestos. . . . .	9
2.0.2. Comparación de los errores cometidos en los parámetros de calidad de los distintos escenarios sin realizar interpolación. . . . .	10
3.1.1. Error sin a interpolar para $f_{s,ADC,1}$ del recipiente Zen12 (21 cm) . . . . .	14
3.1.2. Características de los filtros de Lagrange del escenario 1 . . . . .	15
3.1.3. Errores con filtro de Lagrange de orden 5 en el escenario 1 . . . . .	16
3.1.4. Errores con filtro de Lagrange de orden 19 en el escenario 1 . . . . .	16
3.1.5. Características del filtro óptimo del escenario 1 . . . . .	18
3.1.6. Errores con filtro óptimo de orden 95 en el escenario 1 . . . . .	18
3.2.1. Error sin interpolar para $f_{s,ADC,2}$ del recipiente Zen12 (21 cm) . . . . .	19
3.2.2. Características de los filtros de Lagrange del escenario 2 . . . . .	21
3.2.3. Errores con filtro de Lagrange de orden 5 en el escenario 2 . . . . .	21
3.2.4. Errores con filtro de Lagrange de orden 19 en el escenario 2 . . . . .	21
3.2.5. Características del filtro óptimo del escenario 2 . . . . .	23
3.2.6. Errores con filtro de óptimo en el escenario 2 . . . . .	23
3.2.7. Características de las etapas de filtrado <i>Half-band</i> del escenario 2 . . . . .	24
3.2.8. Errores con filtros <i>Half-band</i> en cascada en el escenario 2 . . . . .	27

---

3.3.1. Error sin interpolar para $f_{s,ADC,3}$ del recipiente Zen12 (21 cm) . . . . .	28
3.3.2. Errores con filtro de Lagrange de orden 5 en el escenario 3 . . . . .	30
3.3.3. Errores con filtro de Lagrange de orden 19 en el escenario 3 . . . . .	30
3.3.4. Características del filtro óptimo del escenario 3 . . . . .	32
3.3.5. Errores con filtro óptimo en el escenario 3 . . . . .	32
3.3.6. Características de las etapas de filtrado <i>Half-band</i> del escenario 3 . . . . .	35
3.3.7. Errores con filtros <i>Half-band</i> en cascada en el escenario 3 . . . . .	36
4.2.1. Parámetros modificables en el archivo de configuración . . . . .	39
4.4.1. Resumen de las necesidades de filtros <i>Half-band</i> para su implementación . . . . .	43
4.4.2. Codificación de los coeficientes de los filtros diseñados en coma fija . . . . .	47
4.4.3. Factores de escalado de las distintas etapas para la prevención de desbordamiento [16, p. 96] . . . . .	50
5.1.1. Errores con filtros <i>Half-band</i> en cascada de la implementación en coma fija con MATLAB . . . . .	51
5.2.1. Comparación de la señal de 35 kHz entre las simulaciones en MATLAB y VHDL . . . . .	54
5.2.2. Comparación de la señal de 75 kHz entre las simulaciones en MATLAB y VHDL . . . . .	55

# Glosario

**ADC** conversor analógico digital

**AMBA 4** *Advanced Microcontroller Bus Architecture*

**APS** sumas por muestra

**bloque IP** bloque de propiedad intelectual

**FIR** respuesta finita al impulso

**IGBT** *Insulated Gate Bipolar Transistor*

**IIR** respuesta infinita al impulso

**LTI** lineal e invariante en el tiempo

**MPS** multiplicaciones por muestra

**S.I.** sistema internacional

**SNR** relación señal a ruido

**VHDL** *VHSIC Hardware Description Language*

**VHSIC** *Very High Speed Integrated Circuit*

**ZVS** *Zero Voltage Switching*



# Lista de símbolos

$D$  Ciclo de servicio o duty cycle

$f_0$  Frecuencia de resonancia

$f_{s,ADC}$  Frecuencia de muestreo del ADC

$f_{s,ADC,1}$  Frecuencia de muestreo del ADC (Escenario 1)

$f_{s,ADC,3}$  Frecuencia de muestreo del ADC (Escenario 3)

$f_{s,ADC,2}$  Frecuencia de muestreo del ADC (Escenario 2)

$f_{s,ADC,max}$  Frecuencia de muestreo máxima del ADC

$f_{s,dst}$  Frecuencia de muestreo de destino

$f_{sw}$  Frecuencia de conmutación

$i_L$  Corriente en la carga

$i_{Lpeak}$  Corriente de pico

$P_{avg}$  Potencia media

$t_D$  Tiempo de conducción del diodo

$t_{snb}$  Tiempo de snubber

$v_{bus}$  Tensión de bus

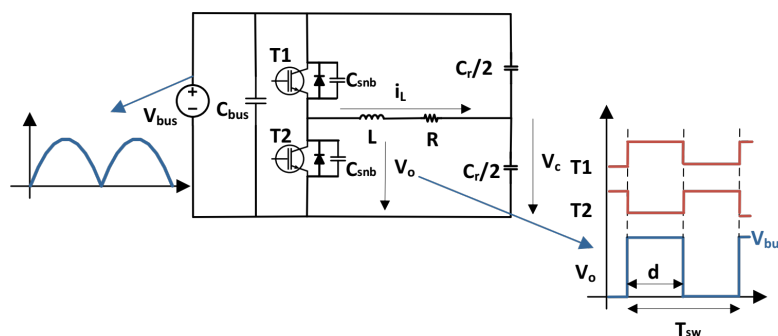
$v_o$  Tensión de salida

# Anexos

## Anexos A

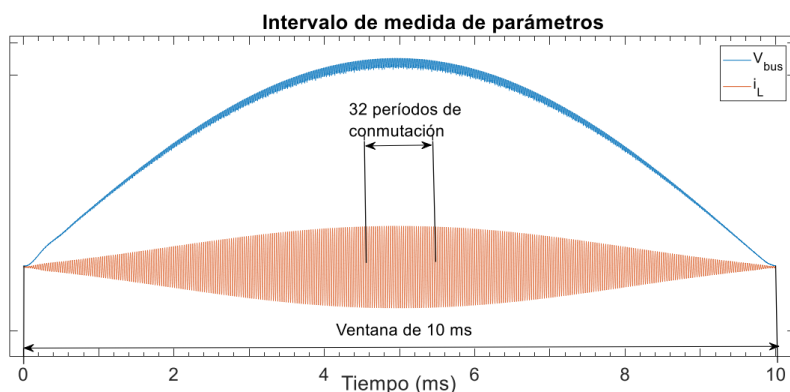
# Cálculo de parámetros temporales

El cálculo de los parámetros temporales de interés de una cocina de inducción se ha realizado en base a la información obtenida del Trabajo Fin de Máster de *A. Barrutia Adrián* [5]. Para su cálculo es necesario conocer, además de la señal  $i_L$ , los valores de las señales  $v_o$  y  $v_{bus}$  (Figura A.0.1) de la etapa de potencia.



**Figura A.0.1:** Inversor resonante serie en semipunto resonante con dos condensadores de resonancia [5, p. 14].

De cada ventana de bus (10 ms) se obtiene un valor de cada parámetro. De ellos, los valores medios y eficaces se obtienen teniendo en cuenta toda la ventana, mientras que para el resto se realiza la media de los 32 ciclos de conmutación de la cresta de la ventana, allí es donde los parámetros adquieren mayor importancia (Figura A.0.2).



**Figura A.0.2:** Intervalo de medida para la extracción de los parámetros temporales [5, p. 29].

Se han implementado en MATLAB funciones para el cálculo de los parámetros[10] en base a las señales sincronizadas obtenidas del osciloscopio.

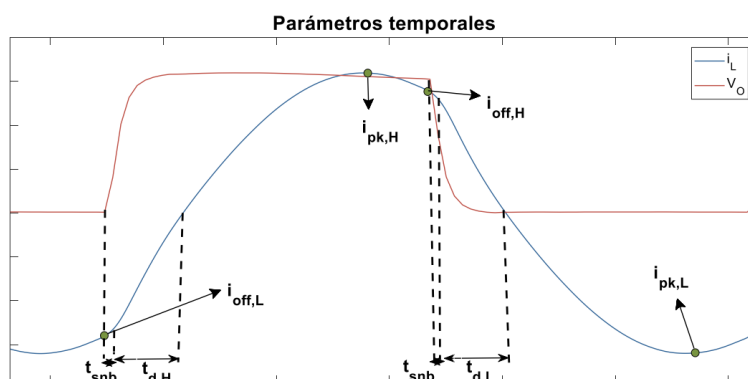
La primera tarea es detectar los flancos de subida y bajada de la tensión de salida, correspondientes a la conmutación de los transistores IGBT (*Insulated Gate Bipolar Transistor*), y necesarios para calcular algunos de los parámetros. Estos flancos se detectan a partir de las siguientes expresiones respectivamente:

$$v_o(j-1) < \frac{v_{bus}(j-1)}{4} \quad y \quad v_o(j) \geq \frac{v_{bus}(j)}{4} \quad (\text{A.0.1})$$

$$v_o(j-1) \geq \frac{3}{4} \cdot v_{bus}(j-1) \quad y \quad v_o(j) < \frac{3}{4} \cdot v_{bus}(j) \quad (\text{A.0.2})$$

donde  $v_o(j)$  son las muestras de la tensión de salida,  $v_{bus}(j)$  la tensión de bus, y  $j$  es el número de muestra.

La función que se encarga de realizar la detección en MATLAB es `igbt_edge_detector`. Esta función toma como parámetros los rangos de 32 ciclos de las señales  $v_o$  y  $v_{bus}$  y devuelve los flancos de subida y bajada en dos vectores.



**Figura A.0.3:** Parámetros temporales relevantes para las cocinas de inducción relativos a la corriente [5, p. 26].

En la figura A.0.3 se han representado los parámetros que se van extraer. Su utilidad, método y función en MATLAB se detallan a continuación:

### Corriente eficaz ( $i_{L,rms}$ )

“Se utiliza para temas de protección de los componentes electrónicos, y para tener una idea de la corriente que soportan. Su cálculo se basa en aplicar la ecuación A.0.3:

$$i_{L,rms} = \sqrt{\frac{1}{n} \cdot \sum_{j=1}^n i_L(j)^2} \quad (\text{A.0.3})$$

donde  $i_L(j)$  son las muestras de corriente,  $j$  es el índice de la muestra, y  $n$  es el número de muestras del intervalo de medida considerado”[5].

Se aplica la función de MATLAB `rms` sobre la señal de  $i_L$  completa.

### Potencia media de salida ( $P_{avg}$ )

Es la variable a controlar de la etapa. Se calcula según la ecuación A.0.4:

$$P_{avg} = \frac{1}{n} \cdot \sum_{j=1}^n i_L(j) \cdot v_o(j) \quad (\text{A.0.4})$$

donde  $i_L(j)$  son las muestras de corriente,  $v_o(j)$  son las muestras de tensión de salida,  $j$  es el índice de la muestra, y  $n$  es el número de muestras del intervalo de medida considerado.

Su cálculo en MATLAB es trivial si se disponen de las señales  $i_L$  y  $v_o$ .

### Corriente de pico positivo ( $i_{Lpk,H}$ ) y negativo ( $i_{Lpk,L}$ )

”Se emplea para temas de protección de la electrónica y del usuario.”[5]

Se obtienen las posiciones de los picos en dos vectores utilizando la función `peak_cicle_detector(il, fs, fsw)`. Toma como parámetros 32 ciclos de conmutación de la señal  $i_L$ , la frecuencia de muestreo y la frecuencia de conmutación  $f_{sw}$  de dicha señal.

### Corrientes de paso a OFF del IGBT superior ( $i_{off,H}$ ) e inferior ( $i_{off,L}$ )

“Son necesarias para calcular los tiempos de conducción de los condensadores de snubber, según un método alternativo que usa la empresa. Se obtiene como el valor de la muestra de corriente correspondiente a la detección de un nuevo flanco de subida  $i_{off,L}$  o bajada  $i_{off,H}$  de la tensión de salida.”[5]

La función de `off_transition_current(il, rlocs, flocs)` devuelve los valores de dichas corrientes en base a las posiciones obtenidos por la función `igbt_edge_detector`.

### Tiempo de conducción de los condensadores de snubber superior ( $t_{snb,H}$ ) e inferior ( $t_{snb,L}$ )

“Este parámetro se emplea para obtener los tiempos de conducción de los diodos. Para su cálculo se debe detectar el paso a OFF del IGBT correspondiente (ecs. A.0.1 y A.0.2). A partir de ese instante se integran las muestras de corriente hasta que se cumpla la ec.A.0.5:

$$v_{bus} < \left| \frac{T_{s,int} \cdot \sum i_L(j)}{2 \cdot C_{snb}} \right| \quad (\text{A.0.5})$$

donde  $C_{snb}$  es el valor del condensador de snubber.

Esto indica que el condensador ya está descargado. Multiplicando el número de muestras transcurridas por el período de muestreo  $T_{s,int}$  se obtienen los tiempos de snubber”[5].

La función de MATLAB que calcula las posiciones donde se cumple la ecuación A.0.5 es `snbcap_discharge_time(i1, vbus, fs, rlocs, flocs)`. Para su cálculo son necesarios los 32 ciclos de la cresta de la señal  $i_L$ , la señal de  $v_{bus}$  en ese mismo intervalo, la frecuencia de muestreo y las posiciones de los flancos de subida y bajada obtenidos mediante `igbt_edge_detector`. Para obtener los tiempos se realiza la diferencia entre las posiciones obtenidas por la función `snbcap_discharge_time` y las obtenidas por la función `igbt_edge_detector` y se multiplican por el período de muestreo como se muestra en las ecuaciones A.0.6 y A.0.7:

$$t_{snb,H}[k] = \frac{1}{f_s} \cdot (locs_{t_{snb,H}}[k] - locs_{flancos\_bajada}[k]) \quad (\text{A.0.6})$$

$$t_{snb,L}[k] = \frac{1}{f_s} \cdot (locs_{t_{snb,L}}[k] - locs_{flancos\_subida}[k]) \quad (\text{A.0.7})$$

### Tiempos de conducción de los diodos superior ( $t_{d,H}$ ) e inferior ( $t_{d,L}$ )

“Sirven para asegurar que la conmutación de los IGBT es ZVS. Se calculan como la diferencia entre el instante en el que la corriente pasa por cero y el instante en el que se ha detectado la descarga del condensador de snubber”[5].

Para el cálculo de este parámetros primero es necesario calcular los cruces por cero de la señal  $i_L$ . Esto se consigue mediante la función `zero_crossing(z, mzd, approximate)`, que toma como parámetros los 32 ciclos de conmutación de la cresta de la señal  $i_L$ , la distancia mínima entre dos ceros consecutivos en muestras y si se quiere una aproximación lineal de la posición del cero.

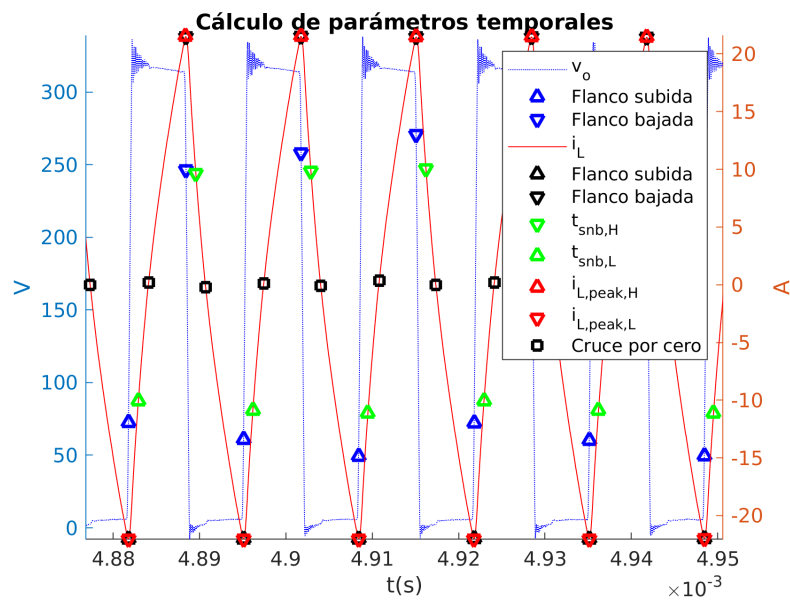
Después la función que obtiene las localizaciones de los tiempos de conducción de los diodos es `diode_driving_time(zc1, tsnbh1, tsnbl1)`. Esta función

necesita las posiciones donde la señal  $i_L$  cruza por cero y las posiciones de los tiempos de conducción de los condensadores de snubber. El tiempo se puede calcular mediante la diferencia entre las localizaciones de obtenidas mediante esta función y las localizaciones obtenidas mediante `snbcap_discharge_time` multiplicadas por el período de muestreo como se indica en las ecuaciones ecuaciones A.0.8 y A.0.9:

$$t_{d,H}[k] = \frac{1}{f_s} \cdot (locs_{t_{d,H}}[k] - locs_{t_{snb,H}}[k]) \quad (\text{A.0.8})$$

$$t_{d,L}[k] = \frac{1}{f_s} \cdot (locs_{t_{d,L}}[k] - locs_{t_{snb,L}}[k]) \quad (\text{A.0.9})$$

En la figura A.0.4 se representan los parámetros extraídos directamente de una señal medida con un osciloscopio haciendo uso de las funciones definidas.



**Figura A.0.4:** Parámetros temporales extraídos de la señal de destino.

# Anexos B

## Filtros de interpolación

### B.1. Filtros FIR

Existe una amplia abanico de literatura en lo que se refiere a filtros FIR. Por lo que en este anexo se detallarán los aspectos más relevantes de los mismos. Para generalizar un filtro de respuesta finita al impulso es un tipo de filtro compuesto solo por ceros, esto significa que no tiene realimentación, por lo tanto su salida estará acotada siempre que su entrada esté acotada.

#### B.1.1. Lagrange

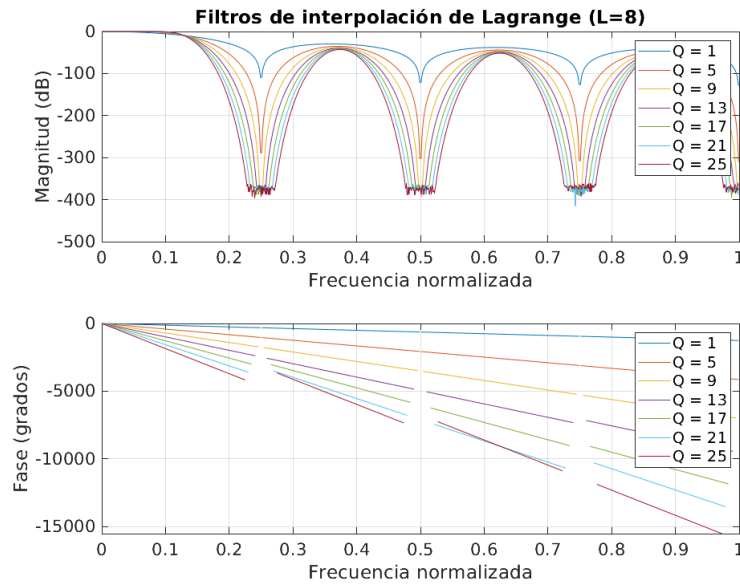
La interpolación de Lagrange es básicamente interpolación polinómica. Un polinomio de orden  $N$  interpola  $N+1$  puntos. Si se trata de un polinomio de primer orden se estaría ante un caso de interpolación lineal.

Se ha utilizado la función `intfilt(L,N,'Lagrange')` de MATLAB para generar los coeficientes del filtro.  $L$  es el factor de interpolación y  $N$  es el orden del filtro. Se prefiere utilizar polinomios de orden impar para asegurar una respuesta en fase en lineal como indica la documentación. Esta función devuelve un filtro de  $(N+1)*L-1$  coeficientes para un orden impar.

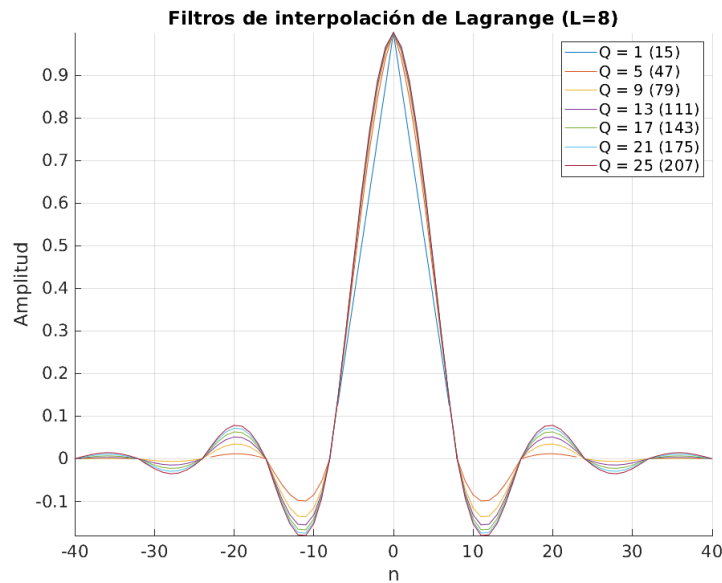
En la figura B.1.1a se representa la respuesta impulsional en frecuencia para distintos ordenes de filtros de Lagrange con un factor de interpolación 8. Respecto a la respuesta en amplitud, puede observarse que el filtro tiene los ceros equiespaciados en frecuencia cada  $2\pi/L$  radianes y que en la banda de paso no presenta rizado. También



se puede ver fácilmente que la fase de todos ellos es lineal. En el dominio temporal (Figura B.1.1b) puede verse que la forma de onda se aproxima a una *sinc* a medida que se aumenta el orden del filtro.



(a) Respuesta impulsional en fase y frecuencia



(b) Respuesta impulsional en el dominio del tiempo

Figura B.1.1: Filtros de Lagrange

## B.1.2. Filtros FIR óptimos

Para los filtros FIR óptimos se han empleado las funciones `firpmord(F,A,DEV,Fs)` y `firpm(N,Fo,Ao,W)`. La primera encuentra el orden aproximado (N), las bandas

de frecuencia normalizadas ( $F_0$ ), la magnitud de las bandas de frecuencia ( $A_0$ ) y los pesos ( $W$ ) para su uso en la función `firpm`. La función `firpmord` toma como parámetros las frecuencias de corte ( $F$ ), la amplitud ( $A$ ) de las bandas definidas en  $F$ , la desviación del rizado en las bandas de paso en unidades lineales ( $DEV$ ) y la frecuencia de muestreo ( $F_s$ ). En la figura B.1.2 se detallan los parámetros que definen un filtro FIR óptimo. La función `firpm` diseña un filtro FIR de fase lineal utilizando el algoritmo de Parks-McClellan. El filtro es óptimo en el sentido de que el error máximo entre la respuesta en frecuencia deseada y la respuesta en frecuencia actual es minimizada. Los filtros diseñados mediante este algoritmo exhiben un equirizado en sus bandas de paso y parada. El número de coeficientes de estos filtros es de  $N + 1$ .

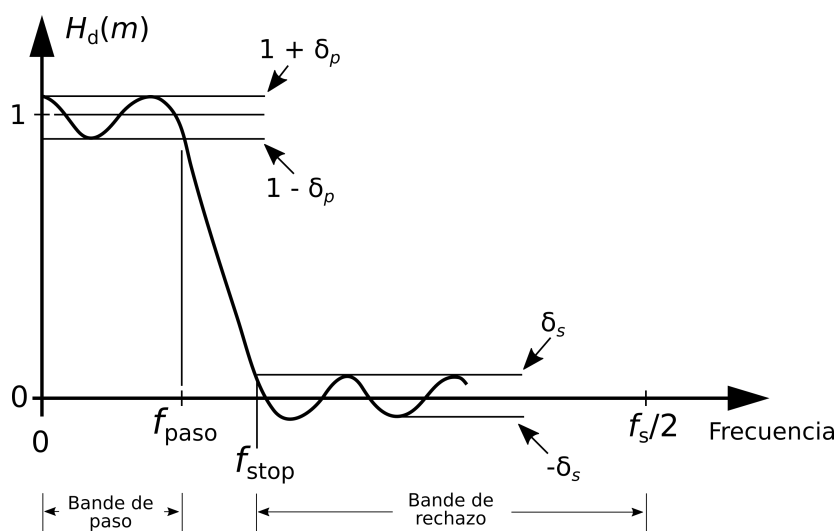


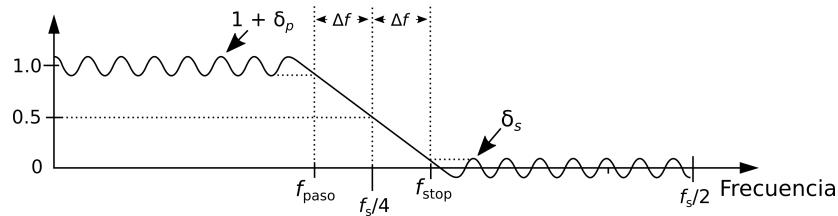
Figura B.1.2: Parámetros de un filtro FIR óptimo

En el trabajo realizado primero se calcula el orden aproximado según las especificaciones. Después se itera sobre la función `firpm` con esos parámetros aumentando el orden hasta que se cumple el error definido en la banda de paso y el número de coeficientes es múltiplo del factor de interpolación, esto se hace así para que en la implementación polifase del filtro se utilicen todos los coeficientes.

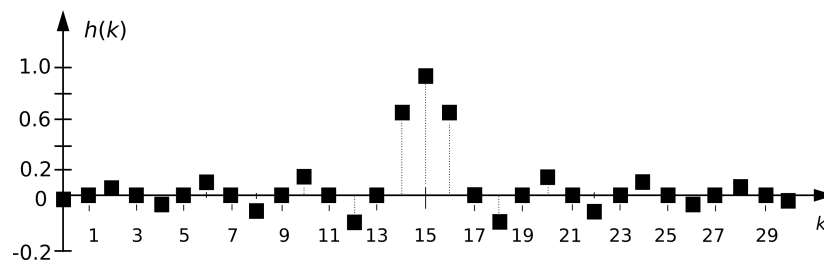
### B.1.3. Filtros Half-Band

Los filtros *Half-band* son filtros FIR paso bajo con frecuencia de corte  $f_s/4$  y simetría impar entorno a  $f_s/4$  [17, p. 701-702]. En la figura B.1.3a se representan las especificaciones que definen al filtro. Se puede ver que el ancho de las bandas de paso y rechazo son iguales. En la figura B.1.3b se observa que en la respuesta impulsional casi la mitad de sus coeficientes son cero. Esto hace que se conviertan en unas etapas de filtrado computacionalmente eficientes.

Para su cálculo se ha utilizado la función de matlab `firhalfband('minorder',fp,dev)` donde se han especificado las restricciones en la banda de paso ( $f_p$ ) y la desviación del rizado en la misma ( $dev$ ).



(a) Especificaciones en dominio frecuencial



(b) Respuesta impulsional en el dominio del tiempo

**Figura B.1.3:** Respuesta en dominio frecuencial y temporal de un filtro *Half-band*

# Anexos C

## Modificación del sistema digital

El sistema desarrollado se ha pensado para que sea simple su recomposición, configuración y modificación.

### C.1. Archivo de configuración

En el archivo `MultiStageHalfBandInterpolator_param.vhd` se encuentran parámetros que definen tipos de datos y constantes que se utilizan en la implementación.

#### C.1.1. Constantes

Las constantes que configuran parte del sistema implementado se detallan en la tabla C.1.1.

Constante	Descripción	Tipo de dato	Rango
<code>C_S_AXI_DATA_WIDTH</code>	Tamaño de palabra del bus AXI en bits	Entero	Según especificación, en este caso 32
<code>PERIPHERAL_NUMBER</code>	Número de periféricos instanciados	Entero	-
<code>CLOCK_FREQUENCY_MHz</code>	Velocidad del reloj	Entero	-
<code>C_S_SAMPLE_DATA_WIDTH</code>	Tamaño de palabra de las muestras en bits	Entero	Según codificación empleada
<code>C_S_COEFF_DATA_WIDTH</code>	Tamaño de palabra de los coeficientes en bits	Entero	Según codificación empleada

**Tabla C.1.1:** Constantes modificables en el archivo de configuración

#### C.1.2. Tipos de dato

Los tipos de datos y registros definidos se encuentran en la tabla C.1.2.

Tipo de dato	Descripción
Coeff	Subtipo de <code>signed</code> de tamaño <code>C_S_COEFF_DATA_WIDTH</code>
CoeffsArray	Vector de enteros de tipo <code>Coeff</code>
GLOBAL2SAXIS	Registro que incluye las señales de entrada del receptor del bus AXI
SAXIS2GLOBAL	Registro que incluye las señales de salida del receptor del bus AXI
MAXIS2GLOBAL	Registro que incluye las señales de salida del emisor del bus AXI
GLOBAL2MAXIS	Registro que incluye las señales de entrada del emisor del bus AXI

**Tabla C.1.2:** Tipos de dato definidos en el archivo de configuración

### C.1.3. Configuración de los filtros

Los componentes implementados requieren algunos parámetros para su correcto funcionamiento. Según el tipo de componente se requerirán más o menos coeficientes.

Constante	Tipo de dato	Descripción
PrescalerHBn	Entero	Frecuencia de trabajo del componente $n$ . Cada cuantos ciclos de reloj pone un dato válido el componente
C_HBn_i	Coeff	Coefficiente $i$ -ésimo codificado en coma fija del componente $n$

**Tabla C.1.3:** Parámetros configurables de los bloques de filtrado

## C.2. Modificación del número de etapas

Para añadir o quitar etapas en `MultiStageHalfBandInterpolator` bastaría con declarar el tipo de componente si no está declarado en la arquitectura.

En el caso de un componente filtro de 11 coeficientes:

```

COMPONENT HalfBandFilterThreeTaps
  Generic(
    PRESCALER: integer;
    COEFF0: Coeff;
    COEFF1: Coeff;
    COEFF2: Coeff
  );
  PORT(
    SlaveAxi_RI : in GLOBAL2SAXIS;
    SlaveAxi_RO : out SAXIS2GLOBAL;

    MasterAxi_RI : in GLOBAL2MAXIS;
    MasterAxi_RO : out MAXIS2GLOBAL
  );
END COMPONENT;

```

Y para un comonente tipo filtro de 7 coeficientes:

```

COMPONENT HalfBandFilterTwoTaps
  Generic(
    PRESCALER: integer;

```

```

    COEFF0: Coeff;
    COEFF1: Coeff
);
PORT(
    SlaveAxi_RI : in GLOBAL2SAXIS;
    SlaveAxi_RO : out SAXIS2GLOBAL;

    MasterAxi_RI : in GLOBAL2MAXIS;
    MasterAxi_RO : out MAXIS2GLOBAL
);
END COMPONENT;

```

Además serán necesarias  $N - 1$  señales auxiliares de tipo GLOBAL2MAXIS y MAXIS2GLOBAL. Siendo  $N$  el número de etapas y  $n$  el número de etapa, la nomenclatura utilizada ha sido Master0*n*AXI\_RI y Master0*n*AXI\_RO respectivamente.

```

signal Master0nAxi_RI : GLOBAL2MAXIS;
signal Master0nAxi_RO : MAXIS2GLOBAL;

```

Después hay que instanciar el componente y conectar sus puertos de entrada salida. Pueden darse tres casos:

### Primera etapa

Este es el caso más sencillo:

```

HB1: HalfBandFilterXTaps
  GENERIC MAP(
    PRESCALER => PrescalerHB1,
    COEFFi => C_HBn_i,
    ...
  )
  PORT MAP(
    SlaveAxi_RI => SlaveAxi_RI,
    SlaveAxi_RO => SlaveAxi_RO,

    MasterAxi_RO => Master0nAxi_RO,
    MasterAxi_RI => Master0nAxi_RI
  );

```

### Etapas intermedia

En este caso hay que realizar una conexión de los puertos esclavo uno a uno con los puertos maestro de la etapa anterior, si se trata de la etapa  $n$ -ésima:

```

HBn: HalfBandFilterXTaps
  GENERIC MAP(
    PRESCALER => PrescalerHBn,
    COEFFi => C_HBn_i,
    ...
  )
  PORT MAP(
    SlaveAxi_RI.s_axis_aclk => Master0(n-1)Axi_RO.m_axis_aclk,
    SlaveAxi_RI.s_axis_aresetn => Master0(n-1)Axi_RO.m_axis_aresetn,
    SlaveAxi_RI.s_axis_tdata => Master0(n-1)Axi_RO.m_axis_tdata,
    SlaveAxi_RI.s_axis_tvalid => Master0(n-1)Axi_RO.m_axis_tvalid,
    SlaveAxi_RO.s_axis_tready => Master0(n-1)Axi_RI.m_axis_tready,

    MasterAxi_RO => Master0nAxi_RO,
    MasterAxi_RI => Master0nAxi_RI
  );

```

### Última etapa

Esta tapa se conecta directamente con la salida de la entidad `MultiStageHalfBandInterpolator`:

```
HBN: HalfBandFilterXTaps
GENERIC MAP(
  PRESCALER => PrescalerHBN,
  COEFFi => C_HBN_i,
  ...
)
PORT MAP(
  SlaveAxi_RI.s_axis_aclk      => Master0(N-1)Axi_RO.m_axis_aclk,
  SlaveAxi_RI.s_axis_aresetn  => Master0(N-1)Axi_RO.m_axis_aresetn,
  SlaveAxi_RI.s_axis_tdata    => Master0(N-1)Axi_RO.m_axis_tdata,
  SlaveAxi_RI.s_axis_tvalid   => Master0(N-1)Axi_RO.m_axis_tvalid,

  SlaveAxi_RO.s_axis_tready   => Master0(N-1)Axi_RI.m_axis_tready,

  MasterAxi_RO => MasterAxi_RO,
  MasterAxi_RI => MasterAxi_RI
);
```

## C.3. Incremento del número de coeficientes en filtros *Half-band*

Si se quisiera crear un nuevo filtro que soporte mayor número de coeficientes solo es necesario copiar el componente `HalfBandFilterThreeTaps` en un nuevo archivo y renombrarlo con el nombre que se desee. Los pasos a seguir a continuación son los siguientes:

1. Buscar la constante que define el número de coeficientes (`N_COEFF`) e igualarla a lo que corresponda.
2. Buscar la constante que define el número de operaciones del planificador (`N_OP`) e igualarla a lo que corresponda (Ver sección 4.4).
3. Buscar el proceso `FILTER_OPS` y localizar el planificador. Añadir tantos bloques de operaciones como sea necesario dentro de la estructura tal y como se detalla a continuación:

```
case to_integer(OpCounter_S) is
  ...
  -- Inicio bloque de operación
when NUMERO_DE_OPERACION_i =>
  Out2_DN <= Out2_D + MultResult_S;
  MultA_S <= COEFF(i-1);
  MultB_S <= SumResult_S;
  SumResult_SN <= resize( DelayLine( to_integer(OpCounter_S)-1),
                          SumResult_SN'length ) +
                    DelayLine( DelayLine'LENGTH - 1 -
                                to_integer(OpCounter_S) );
  -- Fin bloque de operación
```

```
...
when N_OP =>
  Out2_DN <= Out2_D + MultResult_S;
  OutEnable_SN <= '1';
  OutSelector_SN <= '0';
  ClkDividerCounter_SN <= (others => '0');
```

Con estos sencillos pasos se pueden construir componentes de filtrado *Half-band* de tantos coeficientes como se requiera. La única limitación es que el número de operaciones tiene que poder realizarse dentro del tiempo marcado por la mitad del período con el que llegan las muestras de entrada.