



**UNIVERSIDAD AUTÓNOMA DE QUERÉTARO**  
**FACULTAD DE INGENIERÍA**

**Sistemas digitales con lógica reconfigurable II**

Proyecto parcial 2 | Cosimulación de un filtro digital de Cauer pasa bajas.

Profesor: Dr. Miguel Ángel Martínez Prado

Alumno: Lorena Fuentes Flores

Expediente: 263769

Santiago de Querétaro, Qro., 13 de abril de 2021.

## Contenido

Objetivo.....	2
Marco teórico .....	2
Metodología .....	4
Resultados .....	7
Conclusiones .....	10
Anexos.....	10

## Objetivo

Diseño e implementación de un filtro digital de Cauer pasa bajas en cosimulación con los softwares ModelSim y Simulink de Matlab.

## Marco teórico

### Filtro digital

Un filtro digital, es un filtro que opera sobre señales digitales. Es una operación matemática que toma una secuencia de números (la señal de entrada) y la modifica produciendo otra secuencia de números (la señal de salida) con el objetivo de resaltar o atenuar ciertas características.

### Aplicaciones

- Separación de señales que fueron combinadas desafortunadamente (ruido, interferencias provenientes de otros sistemas)
- Recuperación de señales distorsionadas de alguna forma (por ejemplo, al ser transmitidas)
- Síntesis de sonido: creación o modificación de señales para moldear espectros o formas de onda y lograr el efecto auditivo buscado.
- Efectos de audio: chorus, flanger, phaser, reverb

### Filtro digital IIR

IIR es una sigla en inglés para Infinite Impulse Response o Respuesta infinita al impulso. Se trata de un tipo de filtros digitales en el que, como su nombre indica, si la entrada es una señal impulso, la salida tendrá un número infinito de términos no nulos, es decir, nunca vuelve al reposo.

### Expresión matemática de los filtros IIR

La salida de los filtros IIR depende de las entradas actuales y pasadas, y además de las salidas en instantes anteriores. Esto se consigue mediante el uso de realimentación de la salida.

$$y_n = b_0x_n + b_1x_{n-1} + \dots + b_Nx_{n-N} - a_1y_{n-1} - a_2y_{n-2} - \dots - a_My_{n-M}$$

Donde los **a** y **b** son los coeficientes del filtro. El orden es el máximo entre los valores de **M** y **N**, que son los términos que determinan la cantidad de polos y ceros en la función de transferencia.

Aplicando la transformada Z a la expresión anterior:

$$H(z) = \frac{\sum_{k=0}^N b_k z^{-k}}{1 + \sum_{k=1}^M a_k z^{-k}}$$

La diferencia entre los filtros IIR y FIR es que los filtros FIR son de respuesta impulsiva finita y los IIR de respuesta impulsiva infinita. Podemos agregar que los filtros FIR no son recurrentes y los IIR sí lo son.

### Estructura

Hay numerosas formas de implementar los filtros IIR. La estructura afecta a las características finales que presentará el filtro como la estabilidad. Otros parámetros a tener en cuenta a la hora de elegir una estructura es el gasto computacional que presenta.

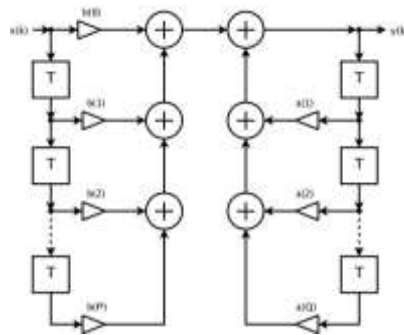


Figura 1. Estructura IIR.

### Filtro de Cauer

Un filtro elíptico o filtro de Cauer diseñado de manera que consiguen estrechar la zona de transición entre bandas y, además, acotando el rizado en esas bandas. La diferencia con el filtro de Chebyshev es que este sólo lo hace en una de las bandas.

Estos filtros suelen ser más eficientes debido a que al minimizar la zona de transición, ante unas mismas restricciones consiguen un menor orden.

Por el contrario, son los que presentan una fase menos lineal.

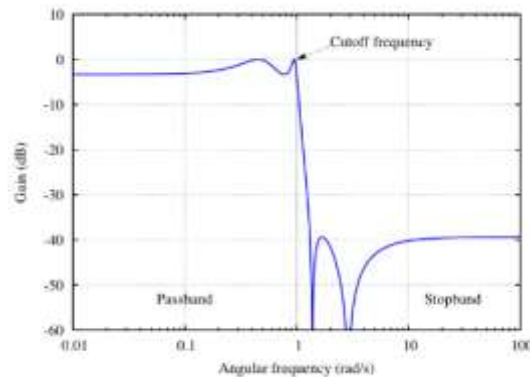


Figura 2. Respuesta de un filtro de Cauer.

## Metodología

### Diseño de filtro IIR-Elíptico

Para el diseño del filtro IIR- elíptico pasa bajo, se desarrolló con ayuda de la herramienta Filter Designer en Matlab.

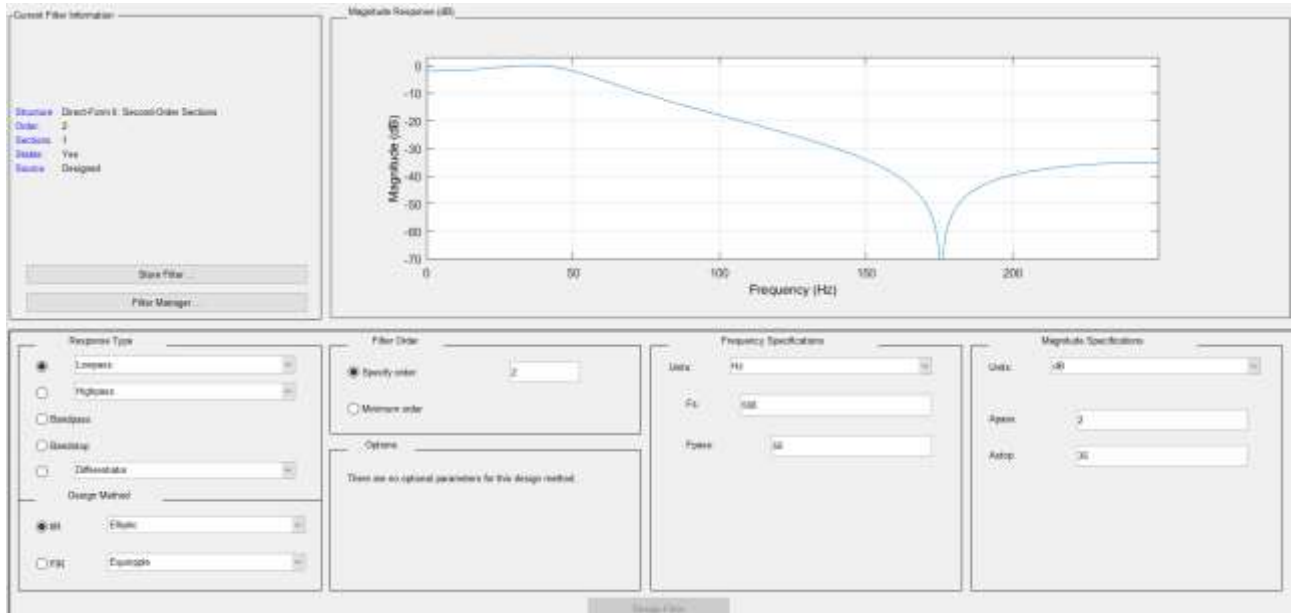


Figura 3. Captura de pantalla del diseño de filtro pasa bajas elíptico.

Las especificaciones del filtro diseñado fueron las siguientes:

- IIR-elíptico de 2° orden
- Frecuencia de muestreo:  $F_s = 500 \text{ Hz}$
- Frecuencia de corte:  $F_{pass} = 50 \text{ Hz}$
- Ondulación de la banda de paso:  $A_{pass} = 2 \text{ dB}$
- Atenuación de la banda de rechazo:  $A_{stop} = 35 \text{ dB}$
- $Passband \text{ Edge } W_n = 0.2$

En Matlab, teniendo las características del filtro, se implementó el siguiente código para obtener los coeficientes y las gráficas correspondientes al filtro.

```
Fs= 500; %Frecuencia de muestreo
fnyq = Fs/2;%frecuencia de Nyquist
Fpass = 50;%frecuencia de corte
Wn = Fpass/fnyq;%frecuencia de corte normalizada
Ap= 02; %Ondulacion de la banda de paso
As= 35; %Atenuación de la banda de rechazo
[b,a] = ellip(2,2,35,Wn);%La función ellip ayuda al diseño de un filtro elíptico
fvtool(b,a);%función para graficar
```

De acuerdo con los resultados del software Matlab, los coeficientes  $a$  y  $b$  del filtro son las siguientes:

Coeficiente b	$b \cdot 2^{16}$	Trunc	HEXADECIMAL
0.065167935	4270.845766	4270	10AE
0.077499126	5078.98272	5078	13D6
0.065167935	4270.845766	4270	10AE

Coeficiente a	$a \cdot 2^{16}$	Trunc	HEXADECIMAL
-1.354789836	-88787.50667	-88787	FFFFFEA52D
0.616438593	40398.91961	40398	9DCE

Las gráficas de respuesta en fase y magnitud del filtro son las siguientes:

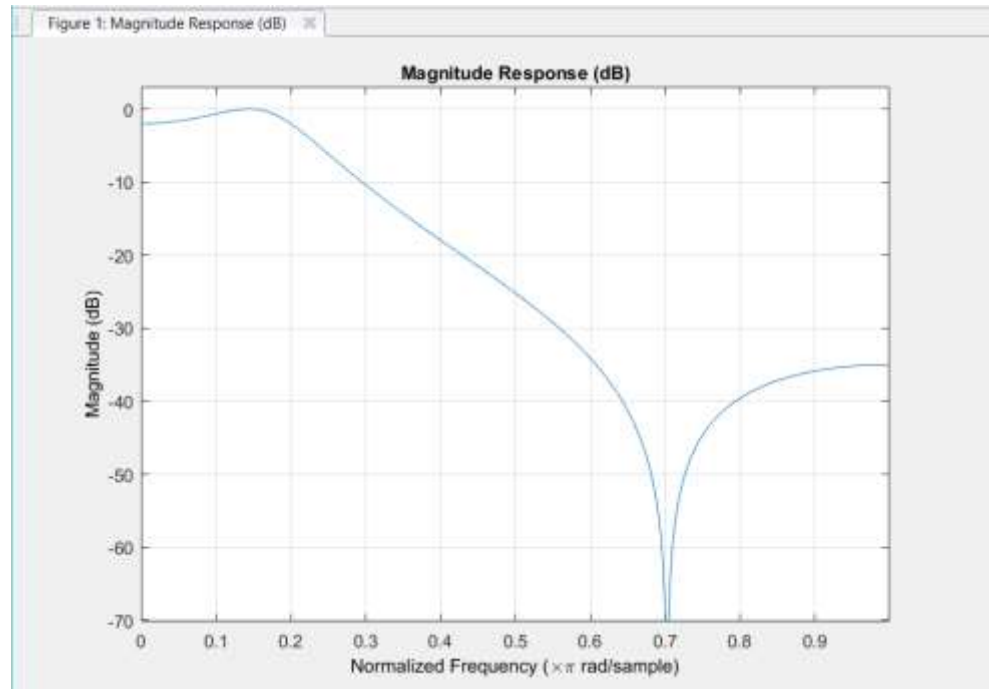


Figura 4. Respuesta en magnitud de filtro diseñado.

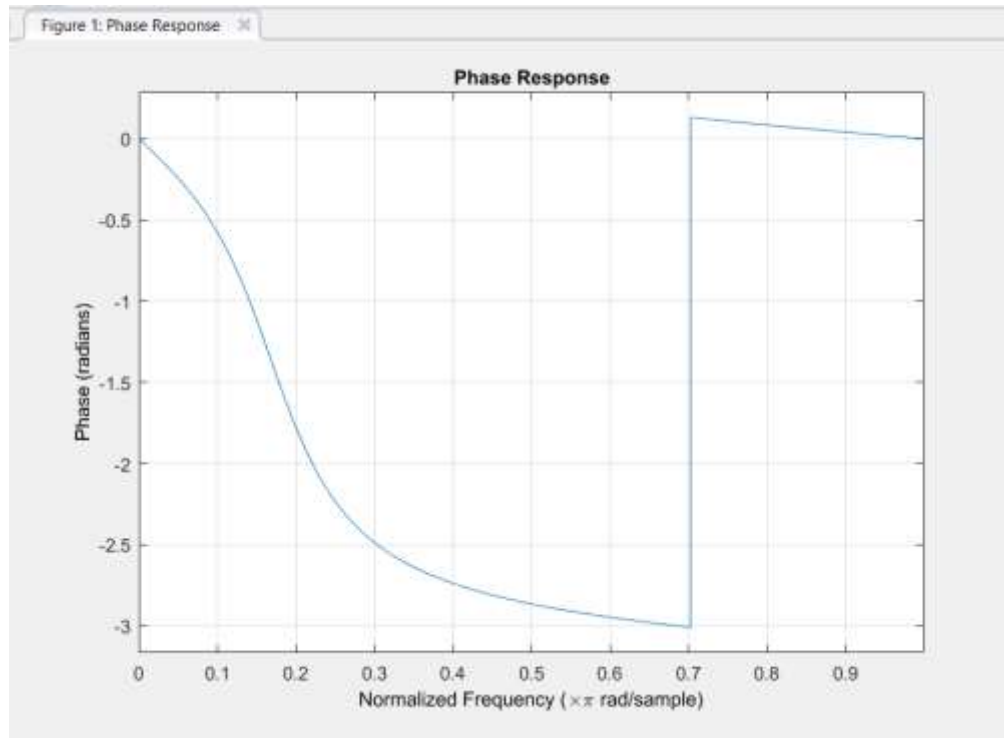


Figura 5. Respuesta en fase de filtro diseñado.

Siguiendo el diagrama mostrado en la figura 1 e implementando lo visto en clase, se siguió el diagrama de bloques de la figura 6 para realizar el código en Active-HDL.

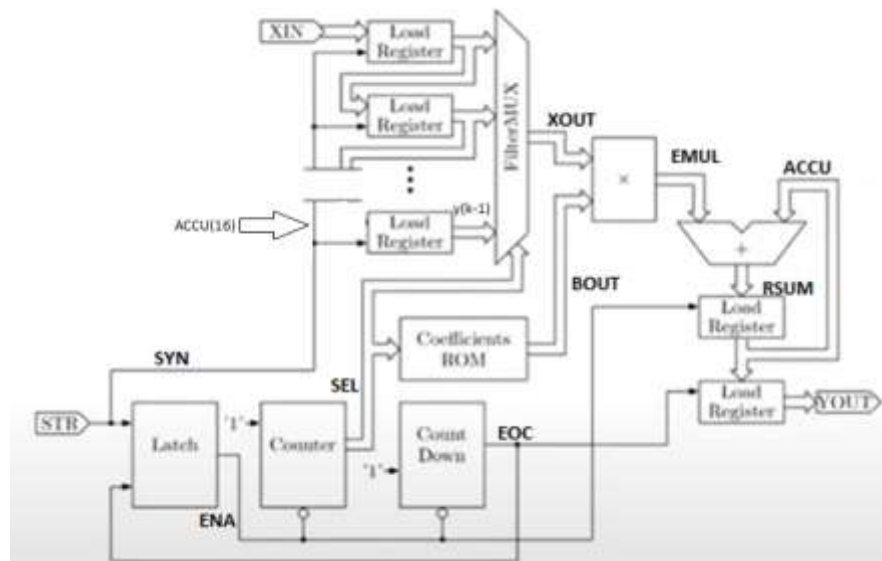


Figura 6. Diagrama a bloques para la implementación de un filtro IIR.

Como se mencionó anteriormente, el código del filtro diseñado fue implementado en cosimulación con los softwares ModelSim y Simulink de Matlab.

El diseño de la simulación en Simulink quedó de la siguiente manera:

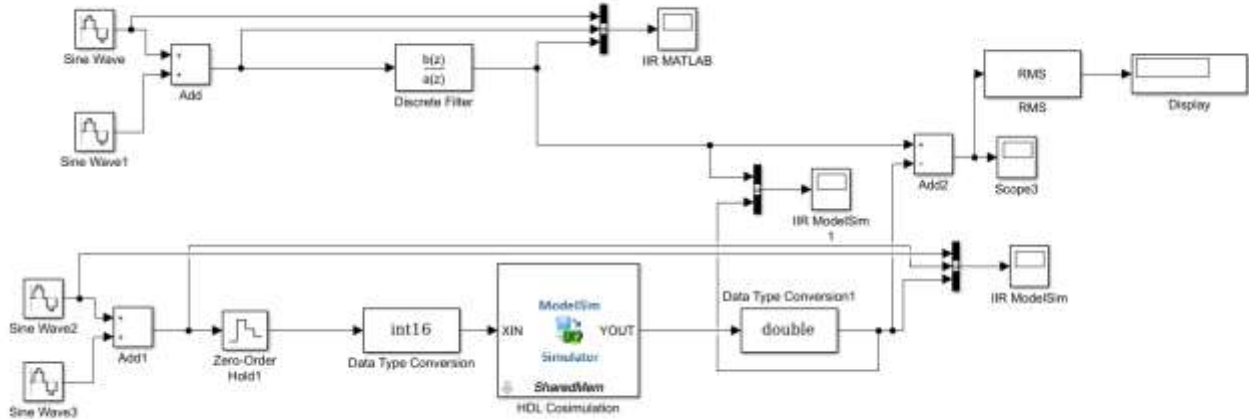


Figura 7. Diseño de simulación en Simulink para implementación de filtro digital.

Como se observa en la figura 7, se realizó la simulación para comparar el desempeño del filtro digital realizado en HDL y el filtro de Matlab.

## Resultados

Para visualizar el desempeño del filtro se tuvieron varias señales de entrada, a continuación, se muestran las respuestas que tuvo el filtro ante ellas.

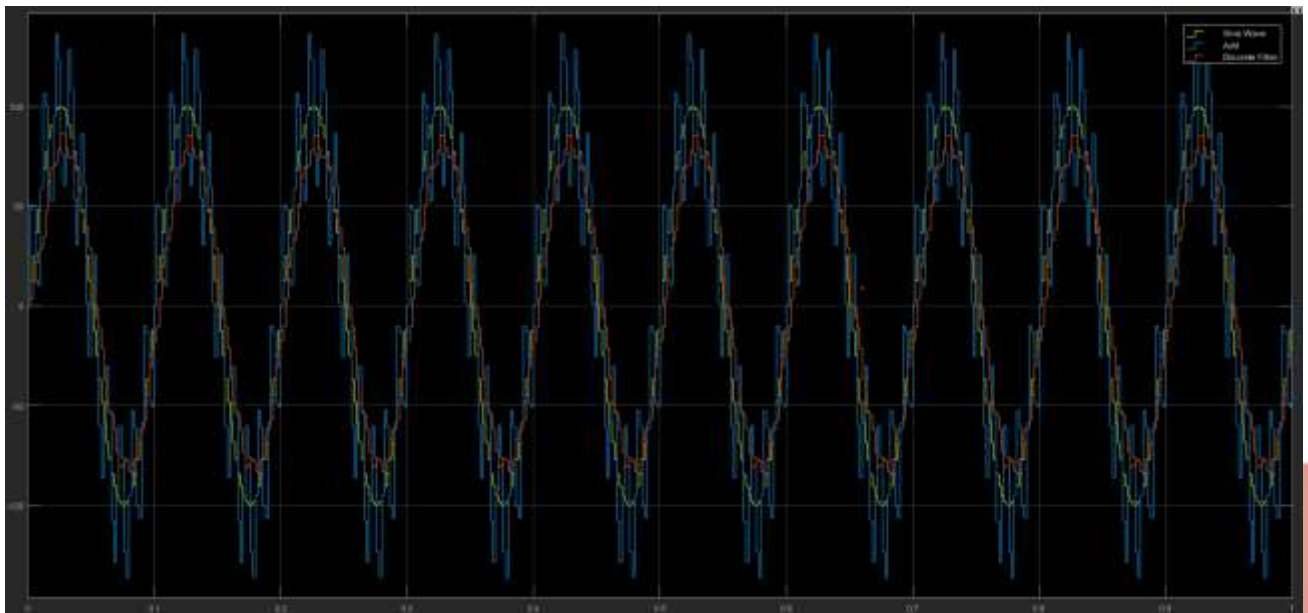


Figura 8. Respuesta de filtro de Matlab con una señal de 10 Hz y una amplitud de 100. Y una señal de 100 Hz con una amplitud de 40.



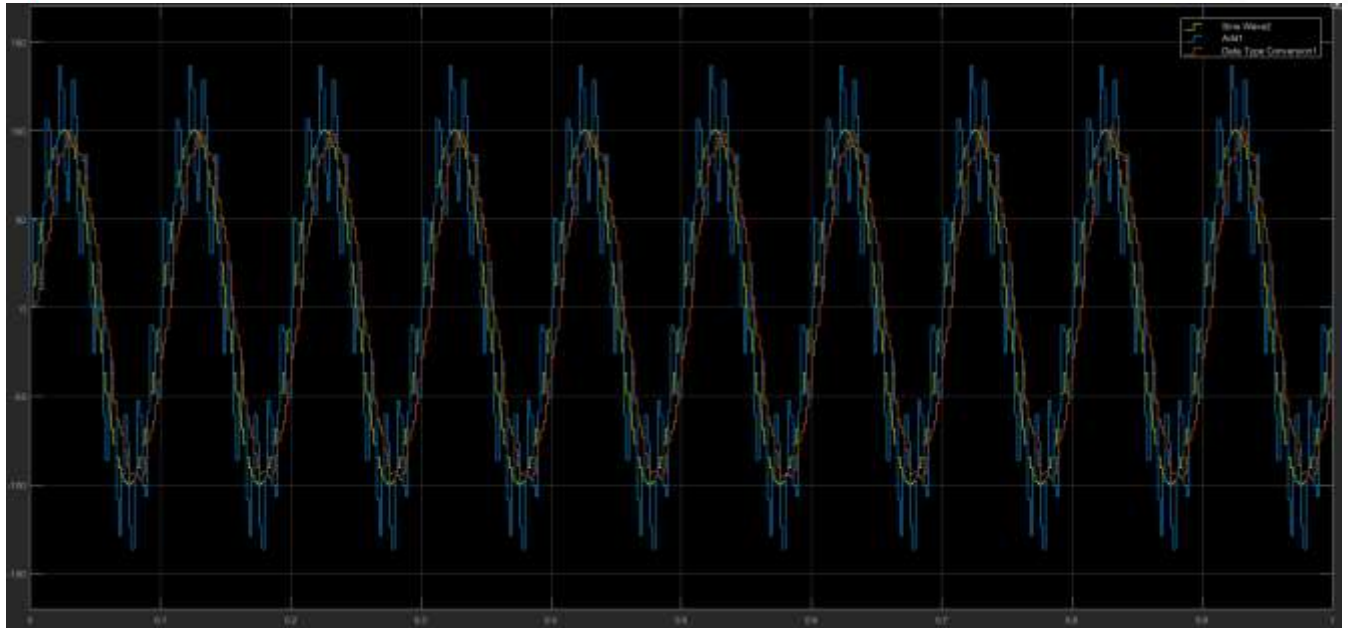


Figura 9. Respuesta de filtro de Active-HDL con una señal de 10 Hz y una amplitud de 100. Y una señal de 100 Hz con una amplitud de 40.

Como puede observarse, el filtro diseñado en Active a comparación con el de Matlab tiene una atenuación mucho menor. Mientras que el desfase es similar.

Obteniendo los datos de las graficas mostradas en las figuras 4 y 5 se obtiene lo siguiente:

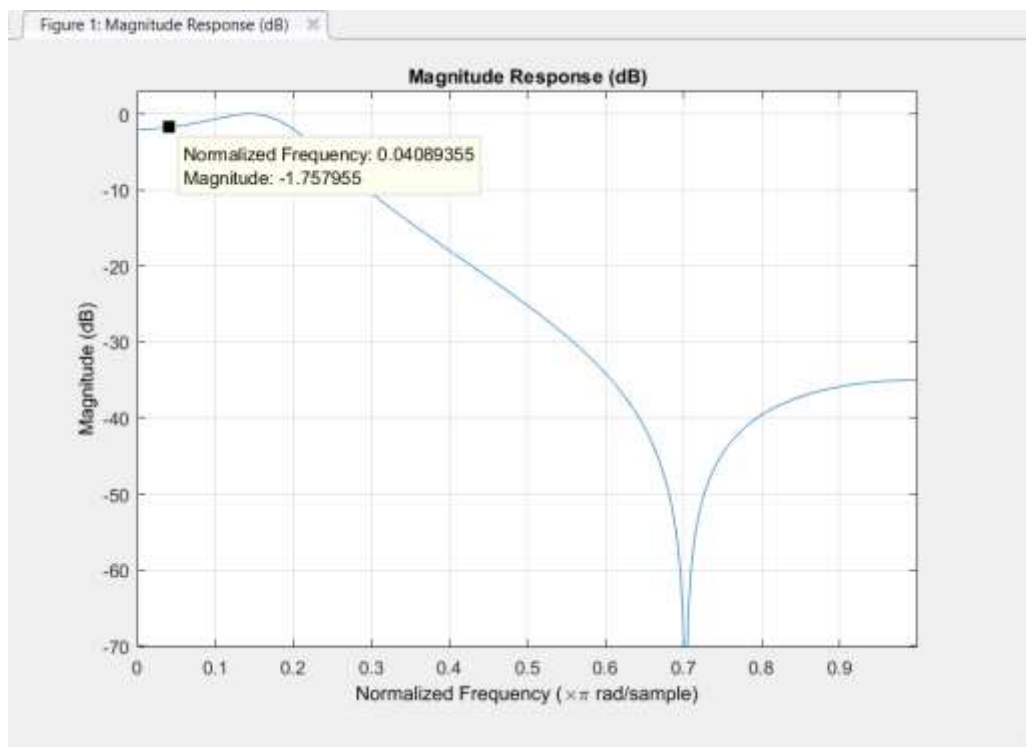


Figura 10. Gráfica donde muestra cuanto es aproximadamente la atenuación que sufre una señal de 10 Hz.



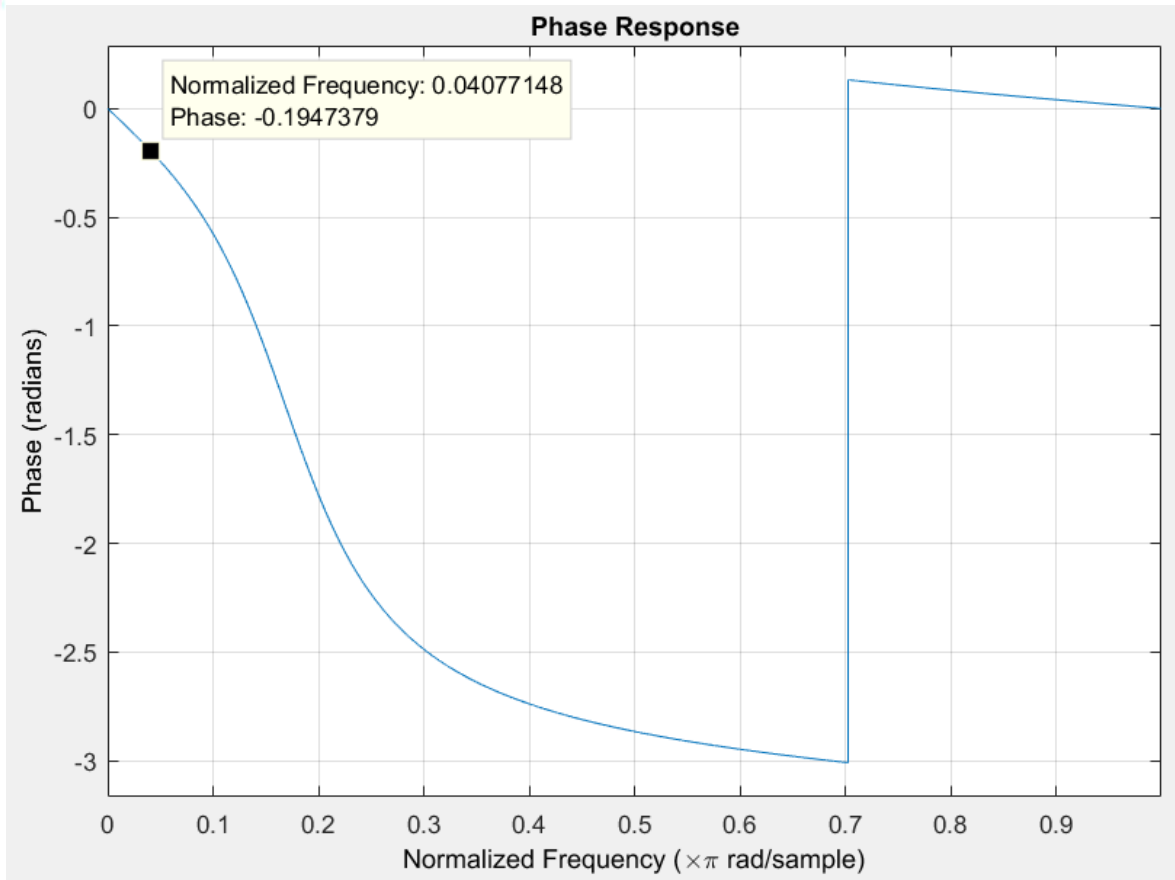


Figura 11. Gráfica donde muestra cuanto es aproximadamente el desfase que sufre una señal de 10 Hz.

Realizando las operaciones adecuadas se tiene que la atenuación de la señal de 10 Hz debería ser de -1.75 dB aproximadamente y el filtro diseñado tiene -0.03 dB.

En cuanto el desfase se muestra en la figura 12.

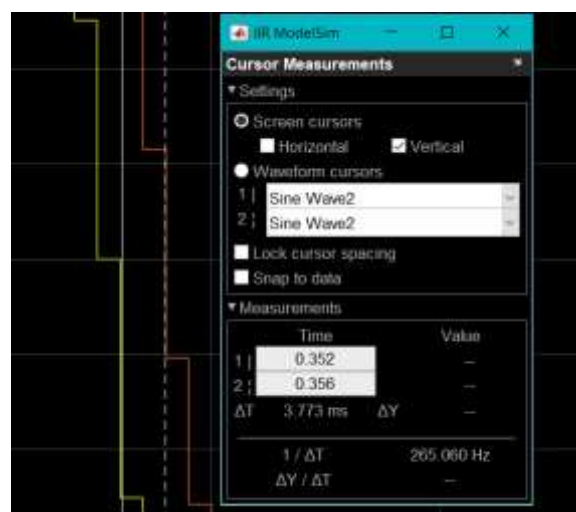
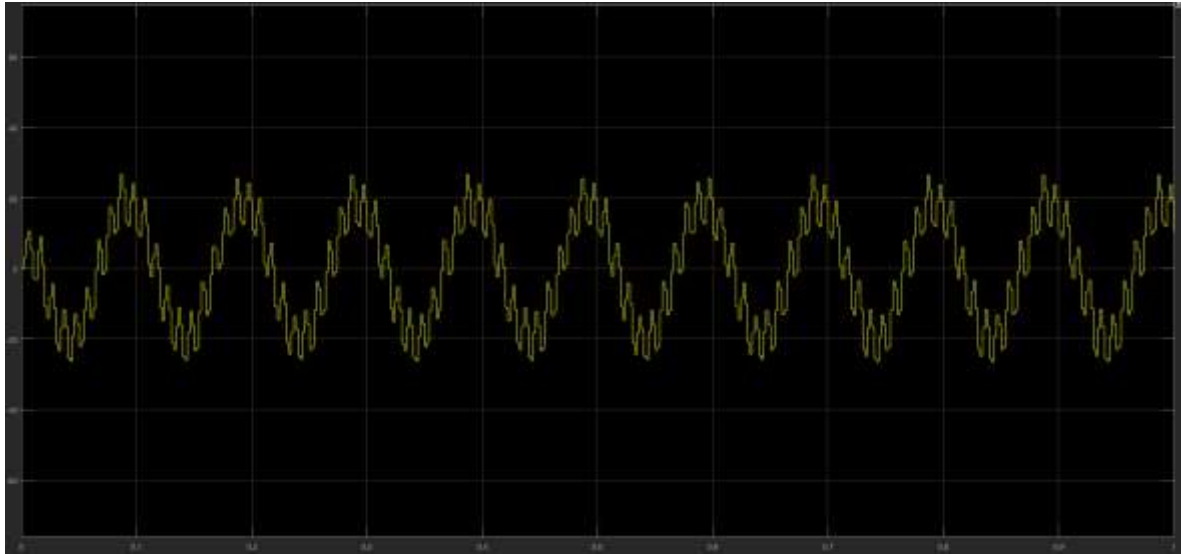


Figura 12. Diferencia de tiempo entre la señal de paso (10 Hz) y la señal que se obtuvo como respuesta del filtro digital en HDL.

De acuerdo con los datos, la señal de respuesta del filtro implementado tiene un desfase de 0.23706 radianes mientras que en la grafica de la figura 11 muestra que debería de ser de 0.194 radianes.

La diferencia entre ambos filtros fue graficada, mostrada en la figura 13, en esta grafica se puede ver que el error es de aproximadamente de una amplitud de 25.



*Figura 13. Error graficado del filtro implementado.*

## Conclusiones

El implementar un filtro digital en simulación nos ayuda a observar el posible comportamiento que se tendría al implementarlo en una situación real, ahorrando costos y tiempo.

El primer objetivo de este proyecto era implementar un filtro elíptico pasa bajas de 5° orden, pero se tuvieron dificultades a la hora de implementarlo, puesto que el diseño realizado en el código de HDL en el momento de realizar la retroalimentación, este valor era 0.

Buscando posibles errores por el que diera este valor, creyendo que el código no hacía su funcionalidad realmente como se mostraba en la simulación de Active, se determinó que los coeficientes correspondientes al filtro de este orden eran demasiado pequeños y al realizarles la conversión a punto fijo, los bits que fueron usados para realizar la retroalimentación no contenían alguna información.

Debido a ello, la respuesta siempre se mostraba diferente a lo esperado y o inestable cuando los bits de retroalimentación se recorrían hacia los menos significativos. Fue por ello que se decidió por un filtro elíptico de un orden menor implementando un filtro elíptico pasa bajas de 2° orden.

## Anexos

Repositorio donde se encuentran los archivos que componen el proyecto:

<https://github.com/LoreFuentes/IIRECauer.git>