

# Cálculo y ensayo de un filtro digital FIR pasa altos

## Filtro Digital FIR pasa altos

Coletto, Danilo Gallego

UTN - FRBB

L.U. 19249

Bahía Blanca, Argentina

[daniloagust2008@gmail.com](mailto:daniloagust2008@gmail.com)

Joel Ermantraut

UTN - FRBB

L.U. 19251

Bahía Blanca, Argentina

[joelermantraut@gmail.com](mailto:joelermantraut@gmail.com)

**Resumen**—Este trabajo de laboratorio consiste en calcular, simular y ensayar un filtro digital pasa altos tipo FIR de orden 10. Se calcularán los coeficientes utilizando MatLab y la placa Discovery STM32F407VG para ensayar el filtro digital mediante un código proporcionado por la cátedra. La finalidad de este laboratorio no será en si el cálculo y desarrollo del filtro, ya que se utilizara la herramienta ‘FilterDesigner’ de MatLab para el mismo, sino la posibilidad de probar, ensayar el filtro y observar su funcionamiento.

**Palabras claves**—filtro digital, ventanas, muestreo, filtro pasa alto, FilterDesigner, ensayo filtro digital, filtro FIR.

### I. INTRODUCCIÓN

La experiencia de este laboratorio consiste en sintetizar un circuito filtro digital, de tipo FIR mediante la placa de pruebas ya nombrada, el cuál contará con una frecuencia de corte de 1.750 Hz y se utilizará una frecuencia de muestreo de 10.000 Hz. De esta manera podremos obtener un pequeño rango de frecuencias en donde la señal de salida obtenida podrá ser todavía una señal reconocible.

Se mostrarán cómo será la configuración de los parámetros y como se obtendrán los coeficientes, además veremos también la respuesta en frecuencia del filtro ideal dada por MatLab y luego se procederá a levantar la curva correspondiente del filtro ensayado para poder compararla con la ideal.

### II. CONSIDERACIONES SOBRE FILTROS

#### A. Descripción de filtros.

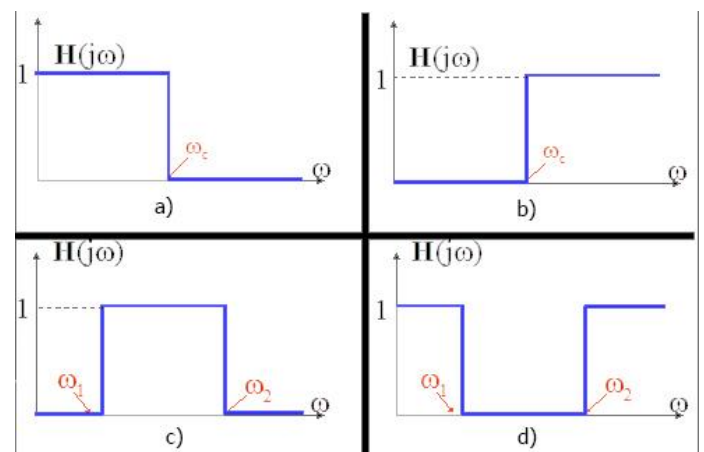
Los filtros de cierto tipo son esenciales para la mayoría de los circuitos electrónicos. Por lo que es de notorio interés para quienes están relacionados con esta rama de la ingeniería tener conocimientos sobre estos.

En la teoría de circuitos, un filtro es una red eléctrica o, en el caso de los digitales, una implementación matemática (cálculo numérico) que se aplica sobre una señal previamente digitalizada, que altera la amplitud y/o fase de una señal, con respecto a la frecuencia. Idealmente, un filtro no añadirá nuevas frecuencias a la señal de entrada, pero si cambiará la amplitud

relativa a varias componentes frecuenciales o su fase correspondiente. Los filtros son usualmente utilizados para

enfatar señales en ciertos rangos de frecuencias y atenuarlas en otros rangos distintos. Así es como un filtro tendrá un valor de ganancia, el cual es dependiente de la frecuencia. Con respecto a su clasificación, se diferencian según las frecuencias que modifican y la utilización de componentes que requieren para la síntesis del circuito. Esto es, si el filtro está compuesto únicamente por elementos pasivos, como resistencias, capacitores e inductores, se denomina pasivo. En otro caso, si emplea componentes activos, se denomina activo. Por otro lado, respecto a las frecuencias que permite (o discrimina, desde el punto de vista opuesto), se clasifican en:

- Pasa – bajos (Figura 1.a): Si permite el paso de frecuencia menores a cierta frecuencia, llamada frecuencia de corte ( $\omega_c$ ), y atenúa las frecuencias mayores a la de corte.
- Pasa – altos (Figura 1.b): Opuesto al anterior, permite las frecuencias por encima de la frecuencia de corte ( $\omega_c$ ), y atenúa las demás.
- Pasa – bandas (Figura 1.c): Permite el paso de frecuencias ubicadas entre dos frecuencias de corte ( $\omega_1$  y  $\omega_2$ ), y atenúa las que no se encuentran en este rango.
- Elimina – bandas (Figura 1.d): Complementario al



anterior, elimina las frecuencias entre las de corte, y permite las demás.

Figura 1. Espectro en frecuencia para cada tipo de filtro.

### B. Especificaciones del filtro

El proceso de diseño de un filtro empieza con que el usuario del mismo especifique las curvas características requeridas del filtro. Esta especificación no puede ser de la forma que se muestra en la figura 1 por que los circuitos físicos no pueden realizar estas curvas idealizadas. En la figura 2 se ilustran especificaciones realistas para las curvas de un filtro de paso bajo. Observemos que como un circuito físico no puede dar ganancia constante a todas las frecuencias de banda pasante, las especificaciones toman en cuenta la desviación de la banda pasante desde el ideal de 0 dB, pero pone un límite superior,  $A_{\max}$  (dB), en esta desviación.

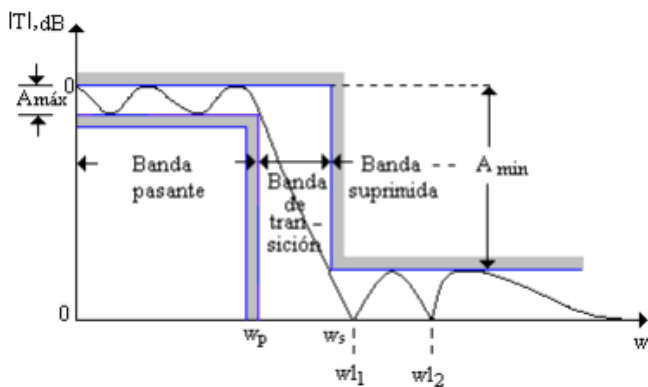


Figura 2. Respuesta de un filtro activo bajo ciertas especificaciones.

Dependiendo de la aplicación,  $A_{\max}$  oscila típicamente de 0.05 a 3 dB. Del mismo modo, como un circuito físico no puede anular todas las frecuencias de la banda suprimida, las especificaciones de la figura 2 toman en cuenta alguna transmisión sobre la banda suprimida, pero las especificaciones requieren que las señales de banda suprimida sean atenuadas en por lo menos  $A_{\min}$  (dB) con respecto a las señales de banda pasante. Dependiendo de la aplicación del filtro,  $A_{\min}$  puede variar de 20 a 100 dB.

Como la transmisión de un circuito físico no puede cambiar abruptamente en el borde de la banda pasante, las especificaciones de la figura 3 dan una banda de frecuencias sobre las cuales la atenuación aumenta de cerca de 0 dB a  $A_{\min}$ . Para resumir, el funcionamiento de un filtro se especifica por cuatro parámetros:

1. El límite inferior de la banda pasante,  $\omega_p$ .
2. La máxima variación permitida en transmisión de banda pasante,  $A_{\min}$ .
3. El borde de banda suprimida,  $\omega_s$ .
4. La atenuación mínima de banda suprimida requerida,  $A_{\min}$ .

Cuanto más estrecha sean las especificaciones de un filtro, es decir, menor  $A_{\max}$ , más alta  $A_{\min}$  y/o una razón de selectividad  $\omega_s/\omega_p$  más cerca de la unidad, la respuesta del filtro resultante será más cercana a la ideal. El circuito resultante, sin embargo,

debe ser del orden más alto y por lo tanto más complejo y costoso.

### C. Sobre Filtros Digitales

Al igual que lo visto para los otros tipos de filtros, es decir, filtros no digitales o analógicos, estos no pueden responder de forma ideal a las curvas que se desearían en un filtro como se mostró en la introducción. De esta manera este cálculo numérico que se aplicará sobre una función previamente digitalizada responderá con ciertas curvas características que estarán lejos de ser las curvas abruptas de cortes ideales de un filtro.

De esta manera estos sistemas discretos que serán nuestros filtros digitales responderán de acuerdo a distintas aproximaciones matemáticas y según la forma de dicha aproximación o transformación. Además, tendremos dos grandes tipos de filtros digitales que responderán a la forma que tendrá cada sistema para resolver su cálculo:

- Filtro IIR (Respuesta Infinita al Impulso): Este filtro basará su funcionamiento bajo el concepto que indica su nombre, si la entrada es una señal impulso, la salida tendrá infinita cantidad de términos no nulos. También se lo conoce como "Filtro Recurrente". Los filtros recurrentes podrán adaptarse a las distintas aproximaciones matemáticas ya vistas para filtros pasivos y activos (Butterworth, Chebyshev, Cauer, etc).
- Filtros FIR (Respuesta Finita al Impulso): Al igual que en el anterior este filtro basa su funcionamiento en que la respuesta a una entrada impulso será una salida con un número finito de términos no nulos. Esto no nos permitirá realizar filtros tipo FIR con las aproximaciones dadas, ya que al ser finito se obtiene una serie de inconvenientes a la hora de su aplicación debido al Fenómeno de Gibbs que nos obliga a optar por una solución conocida como el método de Funciones Ventanas. Por lo tanto, tendremos así filtros FIR con distintos tipos de ventanas (Chebyshev, Rectangular, Hamming, Blackman, etc).

Esta será nuestra principal diferencia en los filtros digitales, teniendo en cuenta además que los filtros IIR implican un cálculo más complejo que los filtros tipo FIR.

### III. DESARROLLO

Como se mencionó previamente, se obtuvo como consigna la implementación de un filtro digital pasa altos de orden 10, donde:

$$F_C = 0,35 \cdot F_{Máx}$$

La frecuencia máxima mencionada en la ecuación resulta la frecuencia máxima de Nyquist que se da para una cierta frecuencia de muestreo:

$$F_S = 2 \cdot F_{Máx}$$

De esta manera la frecuencia de muestreo que se utilizará será impuesta definiendo así las otras dos frecuencias:

$$F_S = 10 \text{ kHz}$$

$$F_{Máx} = 5 \text{ kHz}$$

$$F_C = 0,35 \cdot F_{Máx} = 1,75 \text{ kHz}$$

#### A. Diseño de Filtro Digital mediante MatLab

Como la finalidad del laboratorio 7 será poder ensayar los filtros digitales y tomar muestra de sus curvas de respuesta, la parte matemática del desarrollo de los coeficientes se hará utilizando el software de diseño de filtros que ofrece MatLab (Filter Designer), entonces a continuación veremos cómo establecieron los parámetros.

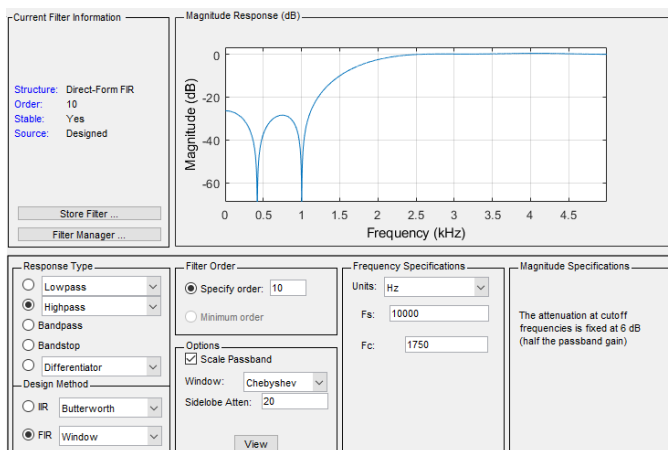


Figura 3. Selección de parámetros para un filtro FIR con ventana Chebyshev

Para este caso, se optará por una ventana tipo Chebyshev, la atenuación buscada del filtro en la frecuencia de corte será como lo especifica en la imagen de -6 dB.

Además, se varió la atenuación mínima de los lóbulos laterales sobre la banda prohibida del filtro, la cuál no debe ser mayor a los -20 dB.

De esta manera una vez definidos los parámetros del filtro se procede al cálculo y obtenemos la siguiente respuesta en fase y amplitud del mismo:

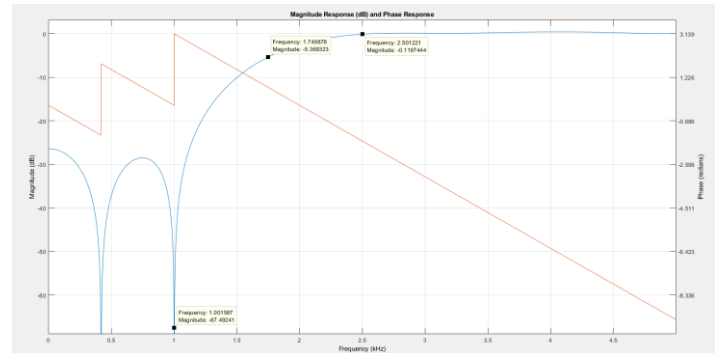


Figura 4. Respuesta en Fase y Amplitud del filtro FIR con ventana Chebyshev

En esta situación no se especifico el tipo de ventana requerido para al filtro, de manera que se probaron varias buscando aquella que tuviera el corte más abrupto y el menor error posible sobre la frecuencia de corte, optando así finalmente por la ventana Chebyshev.

Sin embargo, no habría ningún problema en utilizar otro tipo de ventana para el desarrollo del filtro. De hecho, se realizaron las pruebas con varias ventanas, a continuación, se mostrará la respuesta en fase y atenuación de otras ventanas con la finalidad de comparar y ver diferencias:

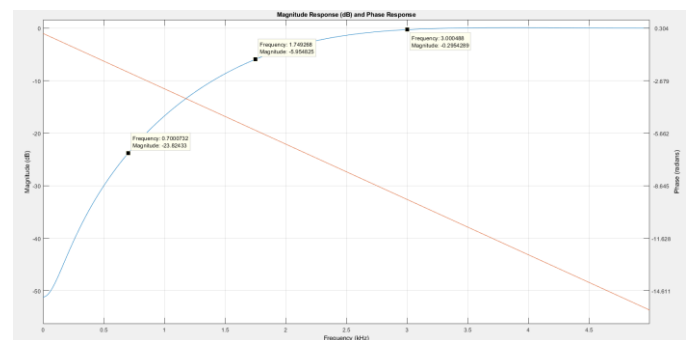


Figura 5. Respuesta en Fase y Amplitud del filtro FIR con ventana Hamming.

Como podemos ver, la ventana Hamming, nos ofrece un filtro sin rebotes en su respuesta en amplitud y con una variación de fase lineal, pero con una variación en atenuación menos abrupta que el filtro con ventana Chebyshev.

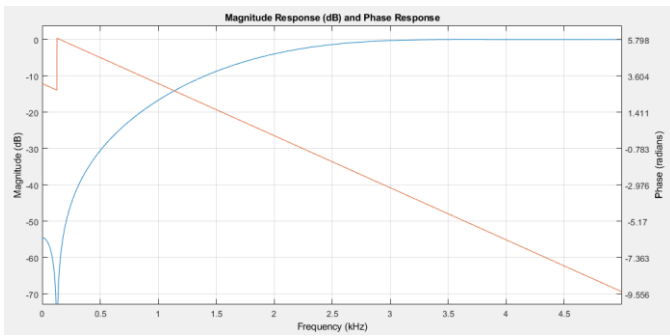


Figura 6. Respuesta en Fase y Amplitud del filtro FIR con ventana Kaiser

#### B. Cálculo de los Coeficientes

Como se menciona, esta herramienta nos ofrece los coeficientes calculados ya para nuestro filtro diseñado, estos coeficientes serán incorporados en el código dado por la catedra de la siguiente manera:

```
float Coeff_f32[N_COEFF]={
0.030984705522381150027966256743638950866f,
0.044019563198970892980987912324053468183f,
0.012509332954367671439377396325198787963f,
-0.115534779713970581216386790401884354651f,
-0.281534894478530473005406520314863882959f,
0.666948721026436097680800685338908806443f,
-0.281534894478530473005406520314863882959f,
-0.115534779713970581216386790401884354651f,
0.012509332954367671439377396325198787963f,
0.044019563198970892980987912324053468183f,
0.030984705522381150027966256743638950866f};
```

*// Coeficientes Ventana Chebyshev*

En este caso como es un filtro de orden 10 obtendremos 11 coeficientes, por lo cual la variable N\_COEFF será igual a 11.

## IV. MEDICIONES

Pasamos, entonces, a realizar las mediciones para comprobar el funcionamiento de los filtros. Cada autor realizó sus propias mediciones.

#### A. Mediciones realizadas por Danilo A. Coletto Gallego

Para ensayar este circuito se utilizó el PIN PC2 de la placa Discovery como entrada al ADC para muestrear y el pin PA5 como salida del DAC para la reconstrucción de la señal. Se utilizó un osciloscopio Hantek DSO4102C (Figura 7) para realizar la medición con ambos canales, midiendo tanto la entrada como la salida y las correspondientes puntas de prueba que vienen con el equipo (modelo PP-150) en X10.



Figura 7. Osciloscopio utilizado.

Para la señal generada no se utilizó ningún circuito de adaptación a la entrada del ADC, ya que la señal de entrada se le impuso un offset de 1.5 V y una amplitud pico a pico máxima de 1V, siendo así la idea que la señal oscile entre los márgenes de 0.5 V y 2.5 V.

Durante las mediciones no se presentaron mayores inconvenientes, se pudo desarrollar correctamente el ensayo del filtro y se pudieron visualizar, además, por mera curiosidad, el rendimiento y funcionamiento de las distintas ventanas presentadas.

TABLA DE MEDICIONES CHEBYSHEV

Filtro Digital FIR con Ventana Chebyshev Medido			
Frecuencia [Hz]	Vin [Vpp]	Vout [mVpp]	Vout [dB]
100	2.24	200	-20.984
150	2.24	170	-22.396
250	2.20	150	-23.327
300	2.20	100	-26.848
400	2.20	20	-40.828
450	2.20	10	-46.848
500	2.20	40	-34.807
600	2.20	65	-30.590
700	2.20	80	-28.787
800	2.20	77.5	-29.062
900	2.20	55	-32.041
1000	2.20	10	-46.848
1200	2.20	180	-21.743
1300	2.16	320	-16.586
1400	2.16	440	-13.820
1500	2.20	600	-11.285
1600	2.16	800	-8.627
1750	2.16	1100	-5.861
1850	2.16	1200	-5.105
2000	2.16	1500	-3.167
2200	2.16	1800	-1.584
2300	2.16	1900	-1.114
2400	2.16	2000	-0.668
2500	2.16	2000	-0.668

# CURVA LEVANTADA DE MEDICIONES

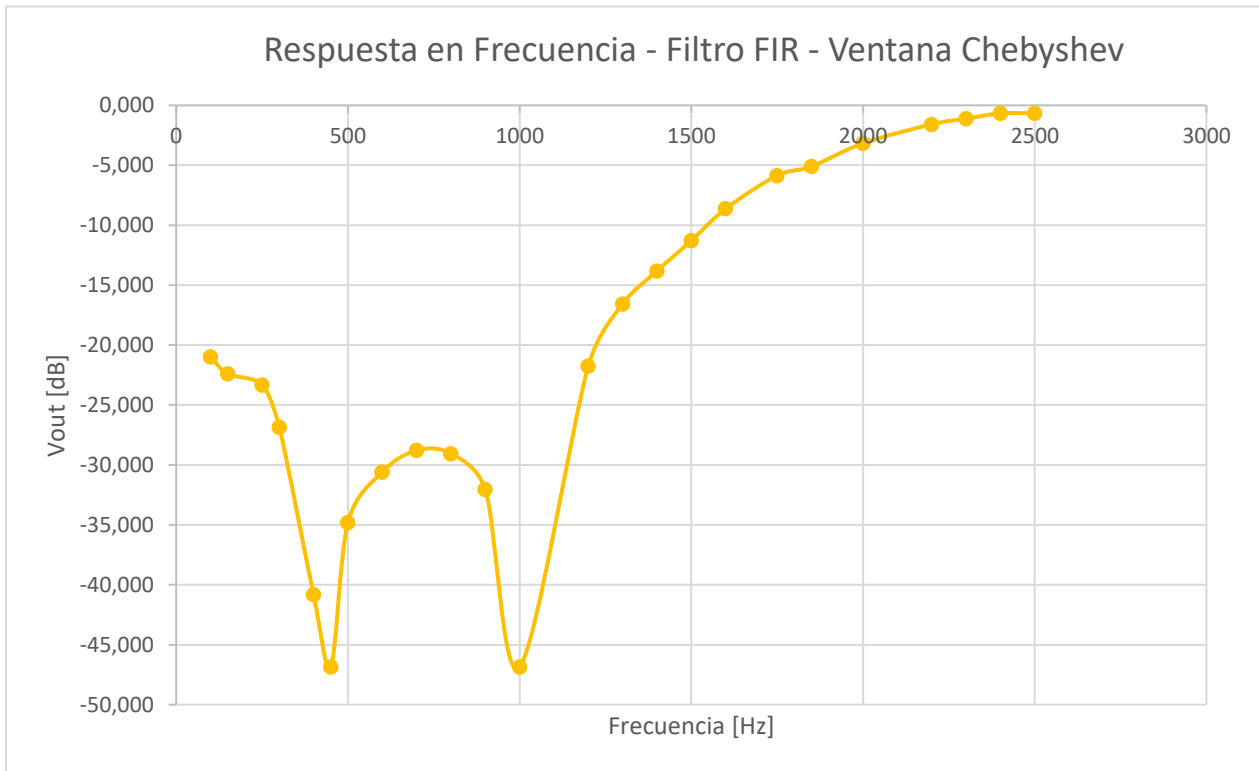


Figura 8. Curva obtenida de las mediciones.

## B. Mediciones realizadas por Joel Ermantraut

En este caso se empleó una placa de audio USB como la que se puede ver en la Figura 9.



Figura 9. Placa de audio USB.

La misma es una placa de audio genérica, con salida estero y entrada mono. Por lo tanto, al usar esta placa, se podían obtener dos señales en el generador, pero solo se podía ver una en el osciloscopio.

A diferencia del otro autor, en este caso se utilizó el circuito presentado en la Figura 10 para adaptar la señal obtenida del generador a niveles de tensión que no dañen el ADC del

microcontrolador y que además le permitieran tomar todas las muestras, dado que el mismo no puede realizar muestreo de tensiones negativas.

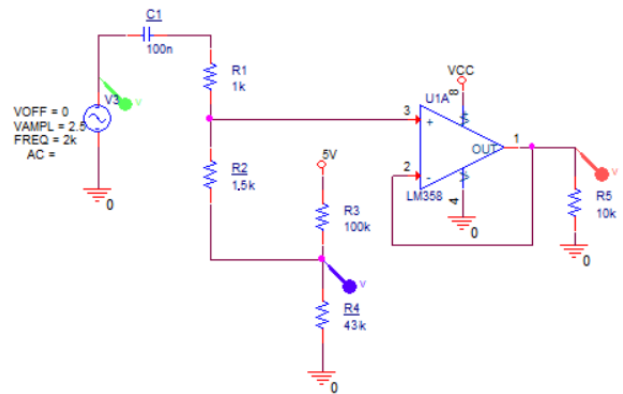


Figura 10. Circuito de adaptación de tensión.

Nota: Se emplearon dos resistencias en serie, de  $33\text{k}\Omega$  y  $10\text{k}\Omega$  para obtener la resistencia de  $43\text{k}\Omega$  que se puede ver en el circuito.

Para este caso se empleó una ventana Hamming como se mostró en los ejemplos de simulación.

Las mediciones obtenidas se presentan a continuación:

TABLA DE MEDICIONES HAMMING

Filtro Digital FIR con Ventana Chebyshev Medido			
Frecuencia [Hz]	Vin [Vpp]	Vout [mVpp]	Vout [dB]
100	0,68	50	-22,671
150	0,68	50	-22,671
250	0,68	100	-16,650
300	0,68	115	-15,436
400	0,68	190	-11,075
450	0,68	280	-7,707
500	0,68	400	-4,609
600	0,68	640	-0,527
700	0,68	650	-0,392
800	0,68	840	1,835
900	0,68	900	2,435
1000	0,68	950	2,904
1200	0,68	960	2,995
1300	0,68	980	3,174
1400	0,68	1000	3,350
1500	0,68	1000	3,350
1600	0,68	1000	3,350
1750	0,68	1000	3,350
1850	0,68	1000	3,350
2000	0,68	1000	3,350
2200	0,68	1000	3,350
2300	0,68	1000	3,350
2400	0,68	1000	3,350
2500	0,68	1000	3,350



## CURVA LEVANTADA DE MEDICIONES

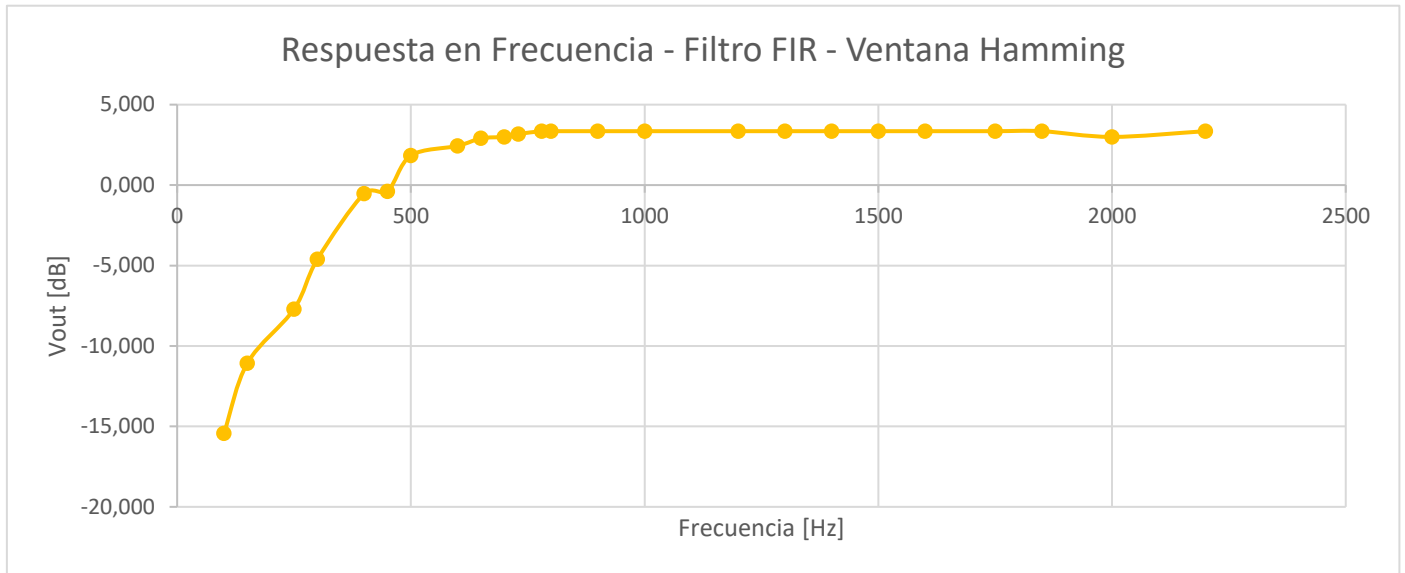


Figura 11. Curva obtenida de mediciones

### CONCLUSIONES

Como conclusión podemos ver que el filtro FIR con ventana Chebyshev utilizado tiene en pocos Hertz de diferencia una gran caída en amplitud y aunque posee lóbulos de “rebote”, la atenuación mínima de estos “rebotes” es de al menos -25 dB asegurando cierta atenuación, mientras que las caídas en los pozos del filtro tienen una atenuación muy grande que prácticamente casi no tenemos señal, aunque es apenas perceptible. Durante el ensayo no hubo problemas y el filtro respondía muy bien de acuerdo a lo esperado, de hecho, se puede apreciar que la curva obtenida es una curva muy prolija y que respeta en gran medida los valores y forma esperados. Con un mayor número de muestras hubiéramos logrado un mayor acercamiento a la curva ideal.

Los mayores inconvenientes se presentaban cuando subíamos en frecuencia y nos acercábamos cada vez más a la frecuencia de Nyquist, donde la señal comenzaba a deformarse completamente, sin embargo, esto no es un problema propio del filtro en sí, sino del muestreo. Aunque realizando pruebas extra y permitiendo una frecuencia de muestreo mayor recalculando el filtro, no se podía subir indefinidamente este parámetro ya que la respuesta del filtro comenzaba a ser distinta, comenzaba

a parecerse cada vez más a un filtro con otro tipo de ventana, esto resulta solamente como una anécdota más que parece interesante mencionar.

En cuanto al ensayo con la ventana Hamming, podemos ver que se logra también una muy buena forma respuesta que emula muy bien la curva esperada en el filtro vista en MatLab pero con atenuaciones muy menores a las esperadas y no estaría teniendo una frecuencia de corte en 1750 Hz ya que recién aproximadamente a los 700 Hz se produce una caída de 3 dB. Si bien respeta la forma de la curva, no estaría respetando el comportamiento esperado del filtro, aunque al contrastar los ensayos con el uso de un osciloscopio y los ensayos que se hicieron con la Placa de Audio USB, podríamos decir que el filtro está funcionando correctamente ya que no se presentaron inconvenientes con el ensayo con osciloscopio (DSO4102C) mientras que sí se presentaron este tipo de inconvenientes con los ensayos con la placa de audio.

A modo de resumen, cabe destacar que, si bien no se logra una exacta precisión, y una gran atenuación en líneas generales, el filtro FIR presenta un comportamiento real más que satisfactorio y obviamente esto dependerá además de la aplicación en la que sea requerido.

### REFERENCIAS

- Apuntes de Cátedra, Teoría de Circuitos II, UTN, FRBB.

# Filtros Digitales

## Filtros FIR

Agustín Emanuel Allende

4° año – Ingeniería Electrónica – UTN

Bahía Blanca, Buenos Aires, Argentina

[aguxx96@hotmail.com](mailto:aguxx96@hotmail.com)

Juan José Miguel Aguirre

4° año – Ingeniería Electrónica – UTN

Bahía Blanca, Buenos Aires, Argentina

[lighmyfire8@hotmail.com](mailto:lighmyfire8@hotmail.com)

**Resumen**—En el presente trabajo de laboratorio se detalla el ensayo sobre la placa STM32F407VG Discovery de un filtro FIR pasa-bajo de orden 10. Mediante la herramienta “Filter Design and Analysis” de MATLAB se verificaron los coeficientes del filtro FIR hallados de forma teórica y, mediante el software brindado por la cátedra, se procedió con la implementación de dicho filtro digital.

**Palabras Clave**—Filtro Digital, Filtro FIR, Filtro No Recurrente, Función Ventana, Función Ventaneada.

### I. INTRODUCCIÓN

#### A. Filtro Digital

Llamamos filtro digital a un sistema discreto cuyo objetivo es realizar sobre una señal, usualmente analógica, una operación de selección de bandas de frecuencias similar a las ya conocidas para soluciones analógicas mediante filtros pasivos o activos.

Los pasos usuales en este proceso son, básicamente, los siguientes:

1. Muestreo y conversión de la señal analógica de entrada en una sucesión numérica. Este paso es realizado por un *Convertor Analógico – Digital*, mejor conocido como ADC.
2. Procesado aritmético de la sucesión obtenida en el paso anterior. Esto representa la verdadera superación de filtrado.
3. Reconstrucción de la señal analógica de salida a partir de la sucesión procesada. Este paso es realizado por un *Convertor Digital – Analógico*, mejor conocido como DAC.

El proceso completo se puede esquematizar con un diagrama en bloques (Figura 1), como se observa a continuación:



Figura 1 – Diagrama en bloques del proceso de filtrado digital.

#### B. Filtro FIR

El sistema FIR (*Finite Impulsional Response*), mejor conocido como *sistema no recurrente*, es aquel en el que la secuencia de salida está determinada por los valores actual y anteriores de la secuencia de entrada, pero no depende de los valores previos de la secuencia de salida. En otras palabras, para determinar el valor actual de la secuencia de salida, el sistema “no recurre” a los valores anteriores de la misma secuencia de salida. Para un sistema no recurrente, la función transferencia se reduce a:

$$H(z) = \sum_{k=0}^M a_k \cdot z^{-k}$$

siendo  $a_k = h(k)$ . En otras palabras, los coeficientes de la función transferencia son los términos de la respuesta impulsional.

El problema de diseño consiste en seleccionar los coeficientes  $h(k)$ , de forma que  $H(e^{j\lambda})$  se aproxime a la especificación de partida dentro de las tolerancias también especificadas. Recordemos que la *frecuencia digital*  $\lambda$  varía entre  $-\pi \leq \lambda \leq \pi$ .

La frecuencia digital  $\lambda$  está relacionada con la frecuencia analógica a través de la relación  $\lambda = \omega \cdot T$ , donde  $T$  es el periodo de la señal muestreo.

Las especificaciones se darán en el campo analógico a través de  $H(j\omega)$  dando el comportamiento deseado de modulo y fase en función de  $\omega$ , o de parte real e imaginaria en función de  $\omega$ .

El mismo comportamiento deberá ser satisfecho por  $H(e^{j\lambda})$  con  $\lambda = \omega \cdot T$ . Llamaremos  $R(\lambda)$  e  $I(\lambda)$  a las partes real e imaginaria de  $H(e^{j\lambda})$ :

$$H(e^{j\lambda}) = R(\lambda) + jI(\lambda)$$

dónde:

$$\begin{cases} R(\lambda) = \frac{1}{2} \cdot [H(e^{j\lambda}) + H(e^{-j\lambda})] & \text{función par de } \lambda \\ I(\lambda) = \frac{1}{2} \cdot [H(e^{j\lambda}) - H(e^{-j\lambda})] & \text{función impar de } \lambda \end{cases}$$

siendo  $R(\lambda)$  e  $I(\lambda)$  funciones par e impar de  $\lambda$  respectivamente y periódicas de periodo  $2\pi$ . Las mismas pueden ser desarrolladas en Serie de Fourier, la primera en términos de coseno y la segunda en términos de seno, es decir:

$$\begin{cases} R(\lambda) = a_0 + \sum_{n=1}^{\infty} a_n \cdot \cos(n\lambda) \\ I(\lambda) = \sum_{n=1}^{\infty} b_n \cdot \sin(n\lambda) \end{cases}$$

Operando, llegamos a:

$$H(z) = a_0 + \sum_{n=1}^{\infty} \left[ z^{-n} \cdot \frac{a_n - b_n}{2} + z^n \cdot \frac{a_n + b_n}{2} \right]$$

Podemos observar que dicha función no es realizable por tener orden infinito. Sin embargo, se podría utilizar si se acepta trabajar con una aproximación. Para ello se realiza un truncado de la serie en un valor cualquiera  $n = N$ . Entonces:

$$H(z) = a_0 + \sum_{n=1}^N \left[ z^{-n} \cdot \frac{a_n - b_n}{2} + z^n \cdot \frac{a_n + b_n}{2} \right]$$

Esta función si bien es finita es no causal dado que existen potencias positivas de  $z$ . Si operamos sobre esta expresión y sacamos factor común  $z^N$  tenemos que:

$$H(z) = \frac{a_N + b_N}{2} + z^{-1} \cdot \frac{a_{N-1} + b_{N-1}}{2} + \dots + z^{-N} \cdot a_0 + z^{-N-1} \cdot \frac{a_1 - b_1}{2} + \dots + z^{-2N} \cdot \frac{a_N - b_N}{2}$$

Esta función se aproxima más a la deseada cuanto mayor sea el orden  $N$  de la truncación. Resulta evidente también que ahora el orden del sistema discreto es  $2N$ , es decir, el doble del de la truncación.

Lo que se hace habitualmente en la práctica es tomar  $I(\lambda) = 0$ . Esto implica rotación de fase cero a toda frecuencia y, por tanto, distorsión de fase nula. Además, considerar  $I(\lambda) = 0$  significa hacer  $b_k = 0$ . Por lo tanto, los coeficientes de  $H(z)$  resultan simétricos, es decir, la respuesta impulsional es simétrica.

Entonces, la función transferencia queda:

$$H(z) = \frac{a_N}{2} + z^{-1} \cdot \frac{a_{N-1}}{2} + \dots + z^{-N+1} \cdot \frac{a_1}{2} + z^{-N} \cdot a_0 + z^{-N-1} \cdot \frac{a_1}{2} + \dots + z^{-2N} \cdot \frac{a_N}{2}$$

Al hacer  $I(\lambda) = 0$  resulta:

$$|H(e^{j\lambda})| = R(\lambda)$$

El desarrollo en Serie de Fourier de  $R(\lambda)$  será:

$$R(\lambda) = a_0 + \sum_{n=1}^{\infty} a_n \cdot \cos(n\lambda)$$

dónde:

$$\begin{cases} a_0 = \frac{1}{\pi} \int_0^{\pi} R(\lambda) d\lambda \\ a_n = \frac{2}{\pi} \int_0^{\pi} R(\lambda) \cdot \cos(n\lambda) d\lambda \end{cases}$$

Al momento de graficar  $|H(e^{j\lambda})|$  puede apreciarse una importante ondulación alrededor de la frecuencia de corte, la cual se suaviza a medida que se aleja de la misma. Este efecto se presenta tanto en la banda pasante como en la banda de rechazo. Dicho fenómeno es conocido como *Fenómeno de Gibbs* en honor al físico estadounidense Josiah Willard Gibbs, el cual fue el primero que dio importancia sobre el efecto mencionado. Este fenómeno está originado en el corte abrupto que presenta la característica que busca aproximar en  $\lambda_c$ .

La amplitud de la primera oscilación es independiente del orden  $N$  de la aproximación. Al aumentar el orden  $N$  de truncamiento, solo se consigue achicar la banda de frecuencia donde se produce la oscilación, es decir, se minimiza la amplitud de las ondulaciones lejanas a la frecuencia de corte.

Para disminuir la amplitud de la ondulación de Gibbs, y considerando que el origen de estas ondulaciones es el corte abrupto, la mejor solución está basada en el método de *Funciones Ventana*.

Una función ventana es una secuencia numérica  $w(n)$  cuya transformada  $z$ , evaluada en la circunferencia unitaria, tiene la forma de un pulso angosto cuando se la representa en función de  $\lambda$ . Esta función debe ser construida con igual número de términos que  $h(n)$ , respuesta impulsional de la

función  $H(z)$  que se quiere mejorar, y con el mismo tipo de simetría que  $h(n)$ , es decir:

$$w(k) = w(2N - k)$$

Los términos de esta sucesión se multiplican por los correspondientes términos de  $h(n)$  y los productos se toman como los coeficientes de la nueva función, que llamaremos *Función Ventaneada*. Resultará, entonces:

$$H_w(z) = \sum_{k=0}^{2N} h(k) \cdot w(k) \cdot z^{-k}$$

Cuando esta nueva función es aproximada por Serie de Fourier, la ondulación Gibbs se ve sensiblemente disminuida, aunque la transición resulta ahora, menos abrupta. La conveniencia del uso de funciones ventana radica en la sencillez del cálculo de los coeficientes del algoritmo del filtro que son los mismos de la función ventaneada. Los coeficientes de la sucesión ventana vienen dados por expresiones algebraicas sencillas y el cálculo de los coeficientes de Fourier de la función original también resulta sencillo puesto que en ella  $R(\lambda)$  vale 0 o 1.

Existe una amplia variedad de funciones ventanas  $W(z)$ , por ejemplo:

- Hann

$$w(n) = 0.5 - 0.5 \cos\left(\frac{n \cdot \pi}{N}\right) \quad \text{para } n \text{ de } 0 \text{ a } 2N$$

- Hamming

$$w(n) = 0.54 - 0.46 \cos\left(\frac{n \cdot \pi}{N}\right) \quad \text{para } n \text{ de } 0 \text{ a } 2N$$

- Blackman

$$w(n) = 0.42 - 0.5 \cos\left(\frac{n \cdot \pi}{N}\right) + 0.08 \cos\left(\frac{n \cdot \pi}{N}\right) \quad \text{para } n \text{ de } 0 \text{ a } 2N$$

## II. DESARROLLO

### A. Enunciado del problema

La experiencia en el laboratorio consiste en ensayar sobre la placa STM32F407VG Discovery un filtro FIR que cumpla con los siguientes requerimientos:

- Filtro pasa-bajo de orden 10.
- La frecuencia de corte debe ser 0.35 de la frecuencia máxima.

Se eligió una frecuencia de muestreo  $f_s$  de 24 kHz. Por lo tanto, la frecuencia máxima es  $f_m = 12$  kHz y, por consiguiente, la frecuencia de corte es  $f_c = 4.2$  kHz.

### B. Obtención de los coeficientes

Los datos que tenemos son:

- Frecuencia de muestreo  $f_s = 24$  kHz.
- Frecuencia de corte  $f_c = 4.2$  kHz.
- Orden del filtro  $n = 10$ .

Entonces:

$$T_s = \frac{1}{f_s} = \frac{1}{24 \text{ kHz}} = \frac{1}{24} \cdot 10^{-3} \text{ s}$$

La frecuencia digital se calcula como:

$$\lambda_c = 2\pi \cdot f_c \cdot T_s = 0.35 \cdot \pi$$

Como el orden del filtro es  $2N = 10$ , la truncación de la Serie de Fourier es  $N = 5$ .

El cálculo de los coeficientes de la Serie de Fourier es:

$$a_0 = 0.35$$

$$a_n = \frac{2}{n \cdot \pi} \sin(n \cdot \lambda_c) \quad \text{para } n = 1, 2, 3, 4 \text{ y } 5.$$

La función ventana elegida es la de Hamming, por ende:

$$w(n) = 0.54 - 0.46 \cos\left(\frac{n \cdot \pi}{N}\right) \quad \text{para } n = 0, 1, 2, 3, 4 \text{ y } 5$$

No será necesario extender el cálculo de los coeficientes por encima de 5 por la ya citada simetría que presenta esta sucesión, es decir:

$$W_0 = W_{10}$$

$$W_1 = W_9$$

$$W_2 = W_8$$

$$W_3 = W_7$$

$$W_4 = W_6$$

Resolviendo, los coeficientes del filtro FIR son:

$$\frac{a_5}{2} \cdot W_0 = -0.0036012653$$

$$\frac{a_4}{2} \cdot W_1 = -0.0127035018$$

$$\frac{a_3}{2} \cdot W_2 = -0.00660363450$$

$$\frac{a_2}{2} \cdot W_3 = 0.0878327075$$

$$\frac{a_1}{2} \cdot W_4 = 0.2586998844$$

$$a_0 \cdot W_5 = 0.35$$

### III. SIMULACIÓN EN MATLAB

Para la simulación del filtro FIR se utilizó la herramienta *Filter Design and Analysis* que el software MATLAB

posee. Dicha herramienta es muy útil ya que devuelve los coeficientes del filtro para su inmediata implementación.

Lo primero que se debe hacer es ingresar las especificaciones que queremos que el filtro FIR a diseñar cumpla. Para este caso, le ingresamos los parámetros que pueden observarse (Figura 2) a continuación:

Figura 2 – Especificaciones ingresadas para el filtro FIR.

- *Response Type:* Lowpass.
- *Design Method:* FIR Window.
- *Filter Order:* 10.
- *Window:* Hamming.
- $F_s$  = frecuencia de muestreo = 24 kHz.
- $F_c$  = frecuencia de corte = 4.2 kHz.

Una vez hecho esto, se deberá clicar en *Design Filter* y dicha herramienta de diseño de filtros nos dará la respuesta en frecuencia desnormalizada (Figura 3) y la respuesta de fase desnormalizada (Figura 4) del filtro FIR a diseñar en base a las especificaciones ingresadas previamente.

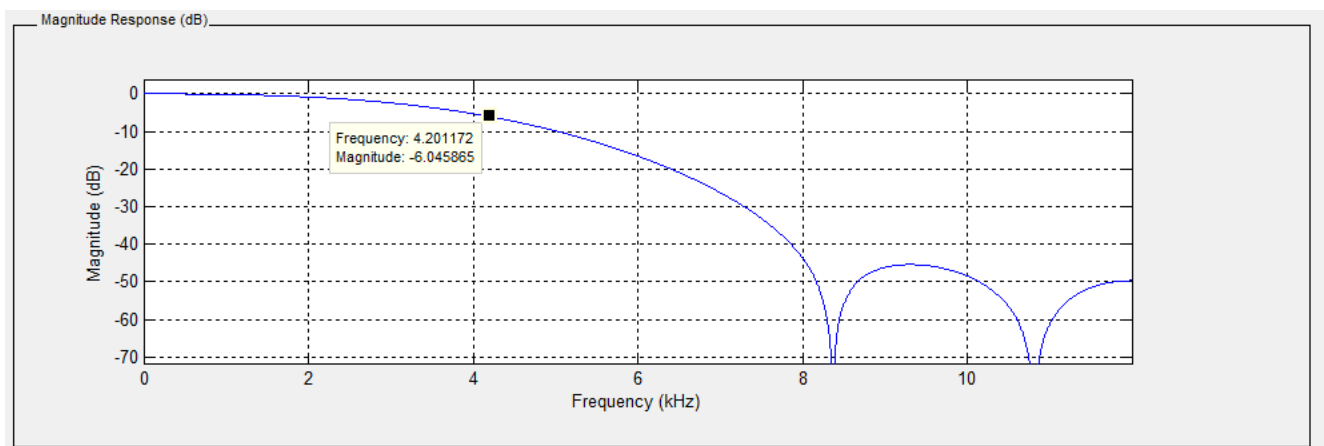


Figura 3 – Respuesta en frecuencia desnormalizada del filtro FIR diseñado.

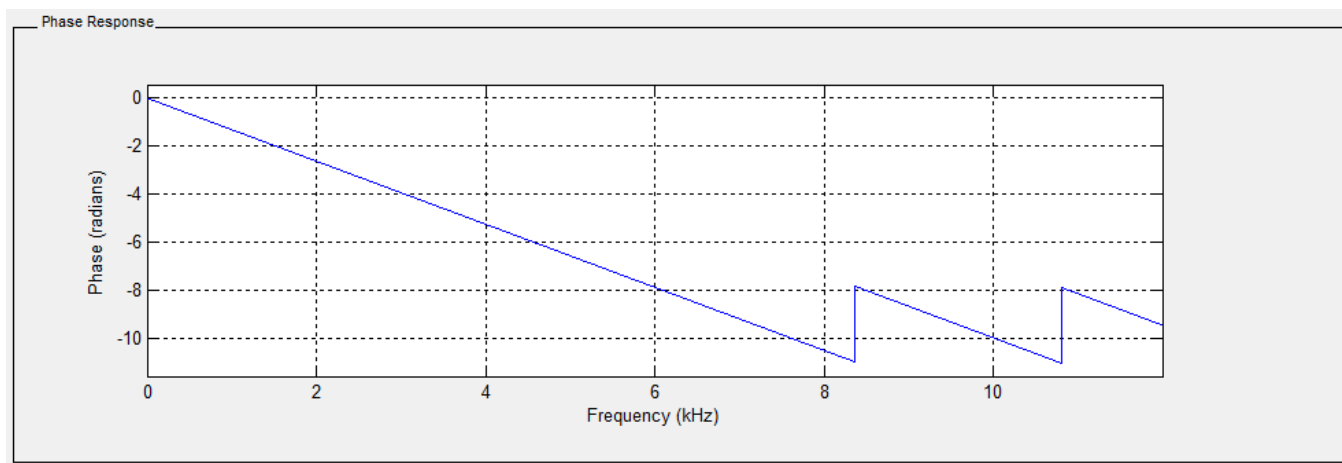


Figura 4 – Respuesta de fase desnormalizada del filtro FIR diseñado.

Puede observarse que la frecuencia de corte del filtro está dada para 6 dB. Esto se debe a que, por defecto, la herramienta *Filter Design and Analysis* lo realiza de esta manera. De todas formas, solo se impusieron como parámetros el tipo de filtro, el orden y la frecuencia de corte del mismo. Los demás parámetros están a libre elección del diseñador.

Dicha herramienta también devuelve los coeficientes del filtro FIR diseñado (Figura 5). Puede observarse que los coeficientes arrojados por dicha herramienta son los mismos que se calcularon previamente de forma teórica.

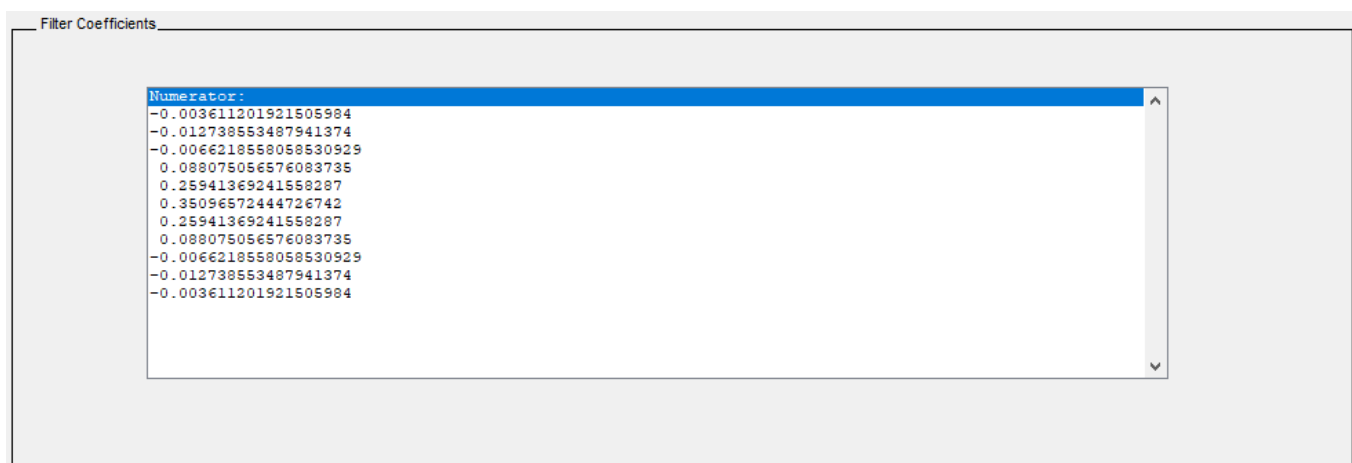


Figura 5 – Coeficientes del filtro FIR diseñado.

Es importante destacar que mediante la herramienta *Filter Design and Analysis* puede generarse un archivo .h para, posteriormente, ser utilizado en la STM32F407VG Discovery. Dicho archivo .h contiene los coeficientes del filtro FIR diseñado en forma de vector fila.

#### IV. MEDICIONES EN EL LABORATORIO

Mediante el código brindado por la cátedra, se implementó el filtro FIR con los requerimientos preestablecidos. Para esto, se cargaron los coeficientes calculados, y verificados mediante la herramienta *Filter Design and Analysis*, de dicho filtro en un arreglo.

**NOTA:** recordar que la cantidad de decimales que se toman en los coeficientes debe ser igual al orden del filtro requerido.

Luego, se generó una onda sinusoidal de  $0.5 V_p$  con 1V de offset para poder ingresarla a la entrada del ADC de la STM32F407VG Discovery y, finalmente, se precedió con el barrido en frecuencia para verificar el correcto funcionamiento del filtro FIR.

Se confeccionó la siguiente tabla con el fin de levantar una curva aproximada de la respuesta en frecuencia del filtro FIR diseñado (Figura 6). Para ello, se utilizó el software *Graph*.

$V_{IN} [V_{PP}]$	$V_{OUT} [V_{PP}]$	[dB]	Frecuencia [Hz]
1	1	0	24
1	1	0	50
1	1	0	130
1	1	0	200
1	0.98	-0.175	300
1	0.98	-0.175	500
1	0.96	-0.355	600
1	0.94	-0.537	800
1	0.91	-0.812	1000
1	0.91	-0.812	1200
1	0.91	-0.812	1300
1	0.90	-0.915	1520
1	0.88	-1.11	1715
1	0.86	-1.31	1915
1	0.85	-1.412	2120
1	0.82	-1.724	2307
1	0.79	-2.047	2512
1	0.75	-2.499	2695
1	0.73	-2.734	2831
1	0.69	-3.223	3048
1	0.65	-3.742	3223
1	0.62	-4.152	3400

1	0.57	-4.883	3504
1	0.55	-5.193	3611
1	0.50	-6.0206	3809
1	0.47	-6.560	4023
1	0.45	-6.936	4093
1	0.43	-7.331	4147
1	0.41	-7.744	4200
1	0.39	-8.179	4252
1	0.38	-8.404	4300
1	0.34	-9.370	4425
1	0.32	-9.897	4490
1	0.30	-10.458	4700
1	0.28	-11.057	4910
1	0.23	-12.765	5111
1	0.20	-13.980	5322
1	0.18	-14.895	5440
1	0.16	-15.918	5530
1	0.14	-17.077	5600
1	0.12	-18.416	5740
1	0.10	-20	5920
1	0.09	-20.915	6200
1	0.07	-23.01	6420
1	0.04	-27.960	7000

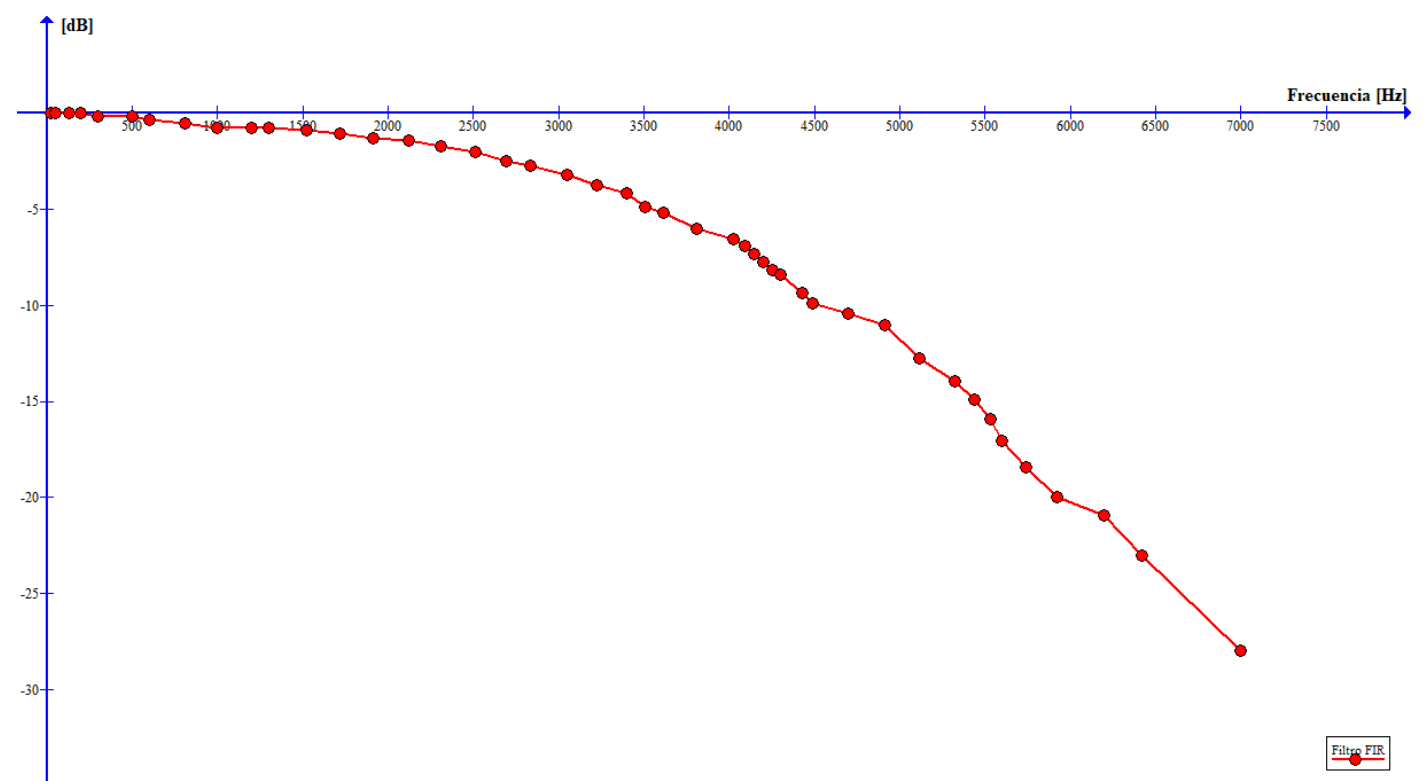


Figura 6 – Curva aproximada de respuesta en frecuencia del filtro FIR diseñado.

## V. INSTRUMENTAL UTILIZADO

Para realizar las mediciones correspondientes se utilizó el siguiente instrumental de laboratorio:

- Osciloscopio digital *Siglent SDS 1102 CNL+* (Figura 7).
- Generador de señales *GW GFG-8050* (Figura 8).
- *STM32F407VG Discovery* (Figura 9).



Figura 7 – Osciloscopio digital Siglent SDS 1102 CNL+.



Figura 8 – Generador de señales GW GFG-8050.



Figura 9 – STM32F407VG Discovery.

## VI. CONCLUSIONES

Mediante el ensayo del presente trabajo de laboratorio pudimos verificar el correcto funcionamiento del filtro FIR diseñado. Se pudo observar una discrepancia entre la respuesta en frecuencia dada por la herramienta *Filter Design and Analysis* de MATLAB y la aproximación de la respuesta en frecuencia hecha de forma experimental. La discrepancia mencionada es la atenuación a la frecuencia de corte  $f_c$ , la cual es de 7.75 dB y debería ser de, aproximadamente, 6 dB. De todas formas, el filtro FIR diseñado cumple con las expectativas.

Por otra parte, fue muy enriquecedor poder instruirse en el manejo de la herramienta *Filter Design and Analysis* que el software MATLAB posee. Es una herramienta muy útil y completa para el diseño de filtros, ya que permite migrar a un determinado software los coeficientes del filtro diseñado para su inmediata implementación.

Respecto a los filtros FIR, se pudo observar que un alto grado de selectividad requerirá de un orden elevado en la función transferencia, lo que traerá aparejado, en la práctica, un aumento en el costo del filtro y una disminución en su velocidad de operación. Por lo tanto, para aplicaciones de alta selectividad, donde la característica de fase es de importancia secundaria, los filtros FIR no serán la elección más conveniente. En cambio, para aplicaciones donde es fundamental mantener la linealidad de la característica de fase (retardo constante), como lo es en la transmisión de datos, el diseño no recurrente es la opción más apropiada. El diseño no recurrente resulta adecuado también para situaciones especiales no abarcadas por las aproximaciones convencionales, como es el caso de los filtros multibanda.

## REFERENCIAS

- [1] *Sistemas Discretos* – Apunte brindado por la cátedra de Teoría de Circuitos II.
- [2] *Filtros Digitales* – Apunte brindado por la cátedra de Teoría de Circuitos II.



# Laboratorio Real N°7: Filtro FIR

Luciano Balestra

Teoría de Circuitos 2-UTN-FRBB  
Bahía Blanca, Argentina  
lucianobalestra2233@gmail.com

Tomás De Caso

Teoría de Circuitos 2-UTN-FRBB  
Bahía Blanca, Argentina  
tomydecaso@hotmail.com

**Resumen**—En el presente informe, se detalla el funcionamiento de un filtro FIR, utilizando diferentes frecuencias, y observando que sucede.

**Palabras clave:** Aliasing – Alias – frecuencia de muestreo – frecuencia de entrada – Nyquist – ADC – DAC

## I. INTRODUCCIÓN

Se diseñará un filtro FIR pasa bajos según los requerimientos exigidos por la cátedra. En este caso particular se exige, como único requerimiento, que la frecuencia de corte sea de 0.5 de la frecuencia máxima. Los demás parámetros fueron elegidos por la comisión. Se eligió una frecuencia de muestreo de 4000Hz, por ende, la frecuencia máxima es de 2000Hz.

## II. CONSIDERACIONES GENERALES

Llamamos filtro digital a un sistema discreto cuyo objetivo es realizar sobre una Señal, usualmente analógica, una operación de selección de bandas de frecuencia similar a las estudiadas en las soluciones analógicas mediante circuitos pasivos o activos.

Como ya dijimos al comenzar el estudio de los sistemas discretos, los pasos usuales de este proceso son básicamente tres:

1. Muestreo y conversión de la señal analógica de entrada en una sucesión numérica.
2. Procesado aritmético de la sucesión obtenida en el paso anterior. Esto representa la verdadera superación de filtrado.
3. Reconstrucción de la señal analógica de salida a partir de la sucesión procesada.

El procesamiento completo se puede esquematizar con un diagrama en bloques:



### Filtros FIR (filtros no recurrentes)

Para un sistema no recurrente, la función de transferencia se reduce a:

$$H(z) = \sum_{k=0}^M a_k \cdot z^{-k}$$

siendo  $a_k = h(k)$ . Es decir, los coeficientes de la función de transferencia, son los términos de la respuesta impulsional.

El problema de diseño consiste en seleccionar los coeficientes  $h(k)$ , de forma que  $H(e^{j\lambda})$  (donde  $\lambda$  varía entre  $-\pi \leq \lambda \leq \pi$ ) se aproxime a la especificación de partida dentro de tolerancias también especificadas.

La frecuencia digital  $\lambda$  está relacionada con la frecuencia analógica a través de la relación ya vista  $\lambda = \omega \cdot T$ , siendo  $T$  el período de la señal de muestreo.

Las especificaciones se darán en el campo analógico a través de  $H(j\omega)$  dando el comportamiento deseado de módulo y fase en función de  $\omega$ , o de parte real e imaginaria en función de  $\omega$ .

El mismo comportamiento deberá ser satisfecho por con  $\lambda = \omega \cdot T$

$$H(e^{j\lambda}) = R(\lambda) + jI(\lambda)$$

$$R(\lambda) = \frac{1}{2} \cdot [H(e^{j\lambda}) + H(e^{-j\lambda})] \text{ función par de } \lambda$$

$$I(\lambda) = \frac{1}{2} [H(e^{j\lambda}) - H(e^{-j\lambda})] \text{ función impar de } \lambda$$

Siendo  $R(\lambda)$  e  $I(\lambda)$  funciones par e impar de  $\lambda$  respectivamente y periódicas de período  $2\pi$ , se pueden desarrollar en serie de Fourier. La primera en términos coseno y la segunda en términos seno:

$$R(\lambda) = a_0 + \sum_{n=1}^{\infty} a_n \cdot \cos(n\lambda)$$

$$I(\lambda) = \sum_{n=1}^{\infty} b_n \cdot \sin(n\lambda)$$

Recordando que:

$$\cos(n\lambda) = \frac{e^{jn\lambda} + e^{-jn\lambda}}{2} \text{ y } \sin(n\lambda) = \frac{e^{jn\lambda} - e^{-jn\lambda}}{2j}$$

Resultará, entonces:

$$H(e^{j\lambda}) = R(\lambda) + jI(\lambda) = a_0 + \sum_{n=1}^{\infty} a_n \cdot \frac{e^{jn\lambda} + e^{-jn\lambda}}{2} + b_n \cdot \frac{e^{jn\lambda} - e^{-jn\lambda}}{2j}$$

Si en esta expresión hacemos el reemplazo  $e^{j\lambda} = z$ , tendremos la expresión buscada de  $H(z)$ .

$$H(z) = a_0 + \sum_{n=1}^{\infty} \left[ z^{-n} \cdot \frac{a_n - b_n}{2} + z^n \cdot \frac{a_n + b_n}{2} \right]$$

Como se puede observar, esta función no es realizable por tener orden infinito ( $1 \leq n < \infty$ ). Sin embargo, se podría usar esta expresión si aceptamos trabajar con una aproximación (con el consecuente error) y para ello realizaremos un truncado (corte) de la serie (sumatoria) en un valor cualquiera  $n = N$ .

$$H_N(z) = a_0 + \sum_{n=1}^N \left[ z^{-n} \cdot \frac{a_n - b_n}{2} + z^n \cdot \frac{a_n + b_n}{2} \right]$$

Esta función, si bien es finita ( $1 \leq n < N$ ), es no causal dado que existen potencias positivas de  $z$ .

Vamos a operar sobre esta expresión, sacando  $z^N$  factor común:

$$H_N(z) = z^N \left( \frac{a_N + b_N}{2} + z^{-1} \cdot \frac{a_{N-1} + b_{N-1}}{2} + \dots + z^{-N} \cdot a_0 + z^{-N-1} \cdot \frac{a_1 - b_1}{2} + \dots + z^{-2N} \cdot \frac{a_N - b_N}{2} \right)$$

Ahora podemos tomar como  $H(z)$  el polinomio en  $z^{-n}$  que quedó entre paréntesis, ya que  $z^N$  representa un simple avance de la sucesión de salida de  $N$  unidades de tiempo discreto.

$H(z)$  resulta entonces:

$$H_N(z) = \frac{a_N + b_N}{2} + z^{-1} \cdot \frac{a_{N-1} + b_{N-1}}{2} + \dots + z^{-N} \cdot a_0 + z^{-N-1} \cdot \frac{a_1 - b_1}{2} + \dots + z^{-2N} \cdot \frac{a_N - b_N}{2}$$

Esta función se aproximará tanto más a la deseada, cuanto mayor sea el orden  $N$  de la truncación.

Resulta también evidente que el orden del sistema discreto resultante será ahora  $2N$ , es decir, el doble del de la truncación. El problema ha quedado reducido ahora a encontrar los  $2N+1$  coeficientes  $a_0; a_1; \dots; a_N$  y  $b_1; b_2; \dots; b_N$ , correspondientes a los desarrollos en serie de  $R(\lambda)$  e  $I(\lambda)$ .

Los datos de partida podrían haber sido el módulo y el argumento de  $H(e^{j\lambda})$ , con los que se puede obtener  $R(\lambda)$  e  $I(\lambda)$ . Lo que se hace habitualmente en la práctica, es tomar  $I(\lambda) = 0$ , lo que implica rotación de fase cero a toda frecuencia y por tanto, distorsión de fase nula.

El hecho de considerar  $I(\lambda) = 0$ , significa que estamos haciendo  $b_k = 0$  y como puede observarse en la expresión de  $H(z)$ , sus coeficientes resultan simétricos, es decir, la respuesta impulsional es simétrica:  $h(k) = h(2N-k)$ .

El sistema que se obtiene en la práctica presenta un retardo de  $N$  unidades de tiempo discreto que proviene de haber ignorado el factor  $z^N$ , que hacía no causal la función  $H(z)$ .

Este retardo, por ser constante, implica una rotación de fase perfectamente lineal, que no produce distorsión alguna.

La función de transferencia queda:

$$H(z) = \frac{a_N}{2} + \frac{a_{N-1}}{2} \cdot z^{-1} + \dots + \frac{a_1}{2} \cdot z^{-N+1} + a_0 \cdot z^{-N} + \frac{a_1}{2} \cdot z^{-N-1} + \dots + \frac{a_N}{2} \cdot z^{-2N}$$

Solo nos resta ahora calcular los  $N+1$  coeficientes  $a_0; a_1; \dots; a_N$ .

Al hacer  $I(\lambda) = 0$ , resulta:

$$|H(e^{j\lambda})| = R(\lambda)$$

El desarrollo en Serie de Fourier de  $R(\lambda)$  será, entonces:

$$R(\lambda) \cong a_0 + \sum_{n=1}^N a_n \cdot \cos(n \cdot \lambda) \quad \text{Donde} \quad \begin{cases} a_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} R(\lambda) d\lambda \\ a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} R(\lambda) \cdot \cos(n \cdot \lambda) d\lambda \end{cases}$$

Sabiendo que tanto  $R(\lambda)$  como  $\cos(n \cdot \lambda)$  son funciones pares, las expresiones quedaran finalmente:

$$a_0 = \frac{1}{\pi} \int_{-\pi}^{\pi} R(\lambda) d\lambda$$

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} R(\lambda) \cdot \cos(n \cdot \lambda) d\lambda$$

Para que la aplicación de un filtro de estas características sea más sencilla se prefiere el uso de Funciones Ventanas.

Una **Función Ventana** es una secuencia numérica que llamaremos  $w(n)$  y cuya transformada  $z$ , evaluada en la circunferencia unitaria, tiene la forma de un pulso angosto cuando se la representa en función de  $\lambda$ .

### III. DESARROLLO

Se implementó sobre una protoboard, el circuito dado por la cátedra para la adaptación de señal de entrada al ADC de la placa Discovery (Figura 1), para limitar la entrada de 0 a 3V, como se ve en la señal de salida de ejemplo, en la Figura 2. Luego de verificar que el circuito adaptador funcionara correctamente, se procedió a trabajar sobre la placa de desarrollo Discovery.

Se utilizó el programa brindado por la cátedra y se lo adaptó a la plataforma Atollic TrueStudio. Se realizaron pruebas para comprobar la frecuencia de muestreo y se verificó el buen funcionamiento del programa.

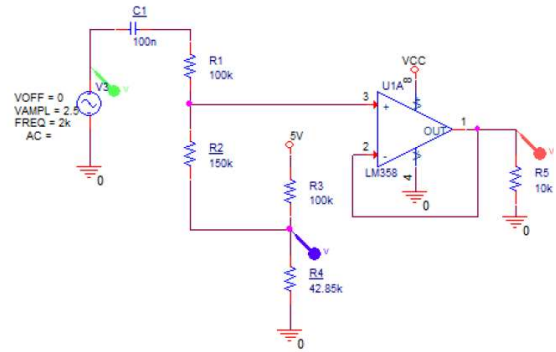


Figura 1: Circuito de adaptación

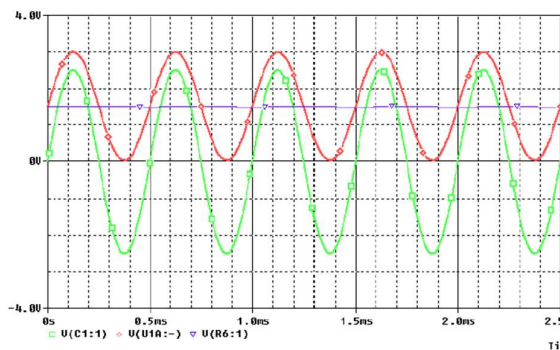


Figura 2: Señal de salida de prueba (adaptador)

#### A. Cálculo de un filtro FIR por medio del software MATLAB

Se utilizó la herramienta para modelado de filtros de MATLAB "Filter design and analysis tool", donde se seleccionó según el criterio de la comisión, el tipo de filtro pasa bajos a utilizar. En esta herramienta se seleccionó un filtro pasa bajos FIR como se muestra en la Figura 3

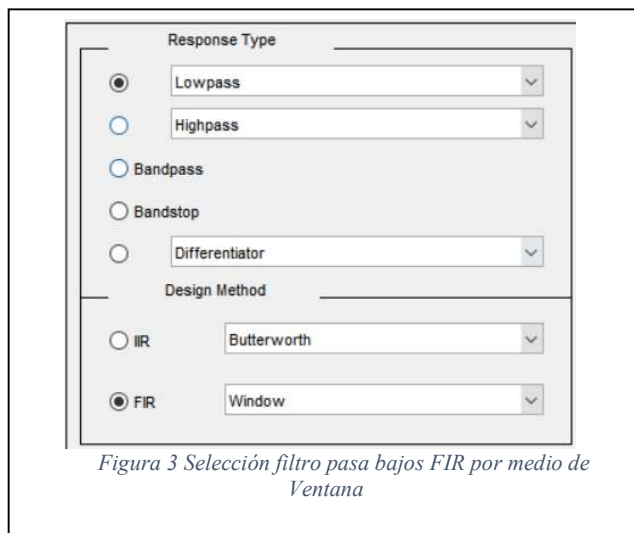


Figura 3 Selección filtro pasa bajos FIR por medio de Ventana

Luego de seleccionado el método de ventana, se selecciona esta y el orden del filtro (Figura 4).

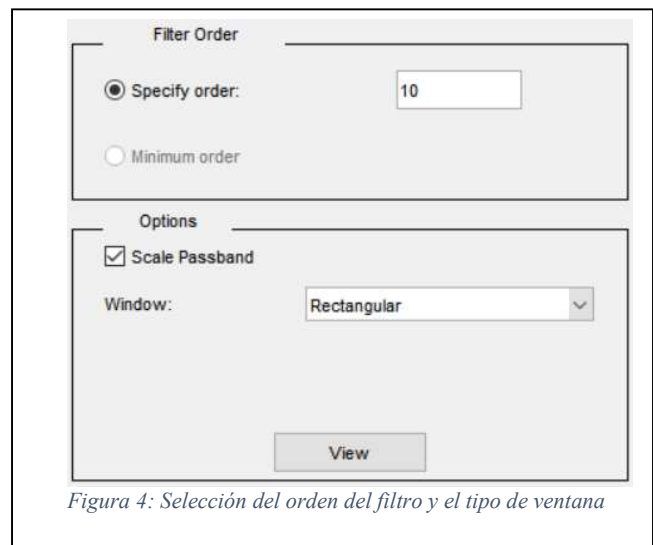


Figura 4: Selección del orden del filtro y el tipo de ventana

Como se ve en la imagen anterior, se seleccionó para que se calcule un filtro de orden 10 y se decidió utilizar una ventana rectangular por su respuesta en frecuencia.

Una vez seleccionados estos dos valores se elige la frecuencia de muestreo  $f_s$ , que, en este caso, fue de 4000Hz y la frecuencia de corte que debía ser de 0.5 de la frecuencia máxima, por ende, 1000Hz.

La respuesta del filtro diseñado se muestra en la Figura 5.

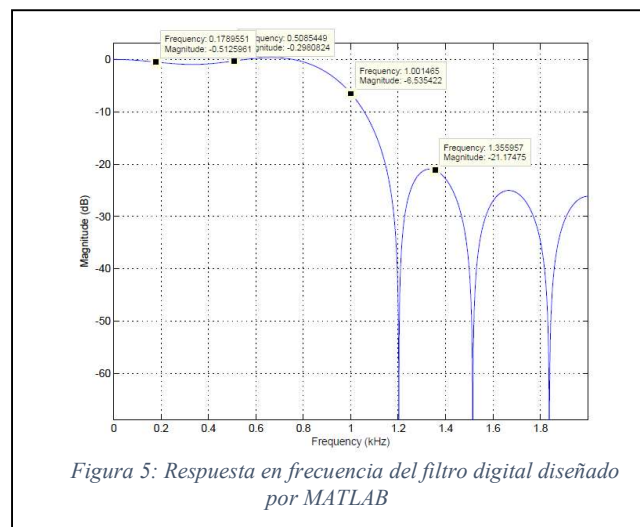


Figura 5: Respuesta en frecuencia del filtro digital diseñado por MATLAB

Para poder obtener los coeficientes del filtro, para luego declararlos en el código de atollc, se debe seleccionar en la pestaña "Targets", y ahí elegir la opción "Generate C header". Esto devolverá un archivo de texto ".txt" con los coeficientes a utilizar, de la siguiente manera.

```
const int BL = 11;
const real64_T B[11] = {
    0.06053031223727, -1.853206328319e-17, -0.1008838537288, 1.853206328319e-17,
    0.3026515611864, 0.4754039606103, 0.3026515611864, 1.853206328319e-17,
    -0.1008838537288, -1.853206328319e-17, 0.06053031223727
}
```

Figura 6: Coeficientes obtenidos en el archivo .txt

Con estos coeficientes se debe modificar el código dado por la catedra.

Una vez modificado el software, se procede a ensayar el funcionamiento del filtro FIR.

### Ensayos y mediciones

Luego de realizada la simulación de nuestro filtro FIR en el software Matlab, se procede a realizar el ensayo y las mediciones del filtro utilizando la plataforma atollc para su implementación, y utilizando una placa de audio usb con puntas atenuadoras para la medición, en el osciloscopio de la computadora. Cabe aclarar, que para las mediciones, ambos alumnos (Balestra, Luciano y De Caso, Tomás), trabajaron en conjunto, por lo que las mediciones realizadas se harán solo una vez.

Utilizando el código brindado por la cátedra para la implementación del filtro FIR, y luego de su modificación, se procede a cargar el programa dentro del microcontrolador de la Discovery, para verificar su funcionamiento.

A continuación, en las Figuras 7 y 8, se observan las señales de salida obtenidas del DAC de la Discovery, a las frecuencias de 500Hz y 700Hz, respectivamente.



Figura 7: Señal de salida a 500Hz



Figura 8: Señal de salida a 700Hz

En las Figuras anteriores, se puede observar claramente, que el filtro se asemeja al simulado en el software MATLAB,

A continuación, en la Figura 9, se observa el circuito utilizado para adaptar la entrada al ADC y eliminar el nivel de continua a la salida del DAC, utilizando un capacitor.

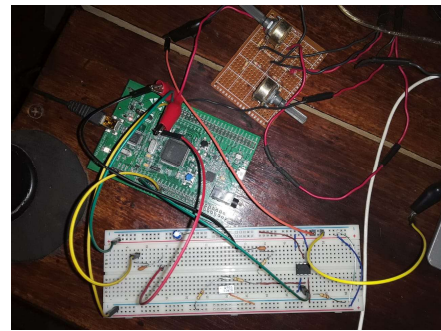


Figura 9: Circuito de adaptación implementado

Luego de realizar las mediciones del filtro digital implementado, haciendo un barrido en frecuencia, se procede a realizar una interpolación de los puntos medidos, para obtener la curva de respuesta en frecuencia real.

A continuación, en las Figuras 10 y 11, se pueden observar, la tabla de puntos interpolados y la curva de respuesta en frecuencia obtenida de dicha tabla, respectivamente.

Frecuencia [Hz]	Vin [mV]	Vout [mV]	Amplitud[dB]
100	780	312	-7,958800173
200	780	470	-4,399934895
300	780	560	-2,878131514
400	780	632	-1,827550488
500	780	702	-0,915149811
600	780	778	-0,022300114
700	780	807	0,295578641
800	780	760	-0,225620208
900	780	619	-2,008079073
1000	780	412	-5,543947733
1100	780	212	-11,31517484
1200	780	50	-23,86249197
1300	780	97	-18,10645737
1400	780	98	-18,01737054
1500	780	52	-23,52182518
1600	780	67	-21,320396
1700	780	83	-19,46033021
1800	780	64	-21,71829257
1900	780	52	-23,52182518
2000	780	73	-20,57543485

Figura 10: Tabla de puntos interpolados

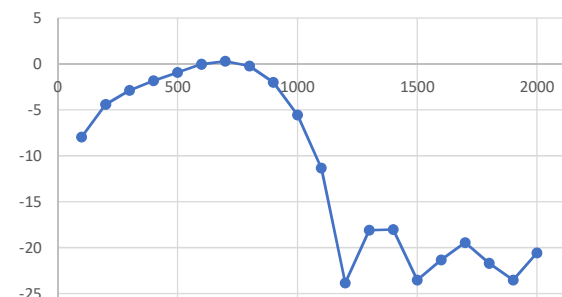


Figura 11: Respuesta en frecuencia obtenida en la realidad

#### IV. INSTRUMENTAL UTILIZADO PARA LAS MEDICIONES

Para las mediciones de ambos alumnos, se utilizó una placa de audio genérica (Figura 12), la cual no se puede caracterizar por falta de información de la misma, junto con una punta de prueba construida en base al esquemático de la Figura 13. Esta misma placa de audio, se utilizó de generador de señales y osciloscopio.



Figura 12: Placa de audio genérica

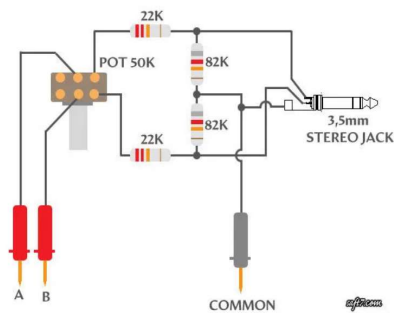


Figura 13: Esquemático punta de prueba para placa de audio

#### V. CONCLUSION

Luego de realizada la experiencia, se puede concluir que el filtro digital FIR, como era de esperar, cumple con las expectativas. Se ve que, aunque tenga algunas diferencias, la respuesta real del filtro sigue de manera relativamente fiel a la gráfica obtenida en MATLAB, las atenuaciones en los valores de frecuencia bajos se los adjudicamos a la frecuencia de corte que impone el amplificador operacional utilizado para la adaptación de la señal para utilizar el ADC.

#### REFERENCIAS

- [1] Apuntes de la cátedra Teoría de Circuitos II, UTN – FRBB.
- [2] Pueyo y Marco, Análisis de modelos circuitales. Tomo I y II.

# Laboratorio real N°7

## Filtros digitales FIR

Cantallops Facundo Nehuen  
Teoría de circuitos II  
UTN - FRBB  
Punta Alta, Argentina  
cantallopsfacundonehuen@gmail.com

Zuccarini Marcos  
Teoría de circuitos II  
UTN - FRBB  
Bahía Blanca, Argentina  
zuccarinimarcos@gmail.com

**Resumen**— El presente informe se plasmarán los resultados de la experiencia consiste en ensayar sobre la placa STM32F407VG un filtro digital tipo FIR en conjunto con el software Matlab con el que se efectuará el cálculo de los coeficientes los cuales se utilizarán en el algoritmo cargado al procesador.

**Palabras claves**— Filtros digital; FIR; Discovery;

### I. INTRODUCCION

La experiencia en el laboratorio consiste en ensayar un filtro FIR pasa bajo de orden 10, con una frecuencia de corte de:

$$F_c = 0.2F_{Nyquist}/2$$

El cálculo de coeficientes de la función transferencia se podrá realizar a mano o mediante funciones proveídas por Matlab. Estos coeficientes se ingresarán al código C que realizará las cuentas necesarias para ejecutar un filtro digital en la placa de desarrollo STM32 nombrada.

Se procede a ingresar al ADC una señal de medida amplitud, montada sobre 1.5V. Una vez muestreada se realiza el filtrado digital utilizando los coeficientes dados y se produce una reconstrucción de la señal filtrada en el DAC.

La señal se monta en 1.5V y se tiene cuidado con su amplitud ya que el ADC de la placa soporta tensiones desde 0V hasta 3V.

### II. CONSIDERACIONES GENERALES

Se llama filtro digital a cualquier sistema discreto que realice una operación de selección de bandas de frecuencia sobre una señal, generalmente analógica. Las operaciones son similares estudiadas en los filtros analógicos pasivos o activos.

Se puede dividir este proceso en tres partes:

1. Muestreo y conversión de la señal analógica de entrada en una sucesión numérica.
2. Procesado aritmético de la sucesión obtenida en el paso anterior, Esto representa la verdadera acción de filtrado.
3. Reconstrucción de la señal analógica de salida a partir de la sucesión procesada.

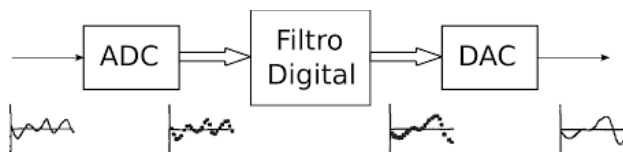


Figura 1: Diagrama en bloques de un filtro digital

Para un sistema no recurrente, la función transferencia se reduce a:

$$H(z) = \sum_{k=0}^{N-1} h(k)z^{-k}$$

O, en función de la frecuencia digital  $\lambda$ :

$$H(e^{-j\omega T}) = R(\lambda) + jI(\lambda)$$

Se toma parte imaginaria nula, por lo que, no hay distorsión de fase. Esto significa que, al hacer pasar una señal por un filtro FIR de estas características, el retardo es el mismo para todos los armónicos que componen la señal y ésta, por ende, no resulta distorsionada. La función transferencia queda:

$$H(z) = \frac{a_N}{2} + \frac{a_{N-1}}{2}z^{-1} + \dots + \frac{a_N}{2}z^{-2N}$$

Se puede llegar a una serie de Fourier recordando la igualdad del coseno en función de exponenciales complejas, resultando un filtro, en este caso pasa bajo, con la siguiente respuesta:

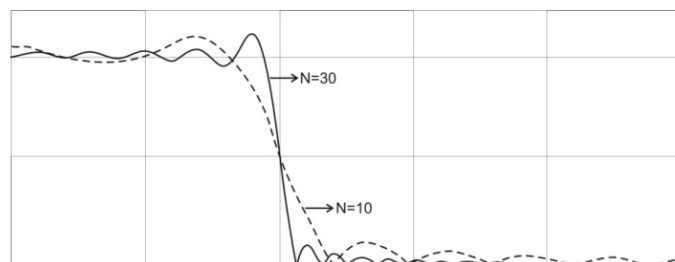


Figura 2: Fenomeno de GIBBS

Se puede apreciar una importante ondulación de gran altura, independiente del orden. Esto se llama fenómeno de GIBBS y para solucionarlo se multiplica a la función transferencia por una *función ventana*  $w(k)$  resultando el producto en una *función ventaneada*.

$$H(z) = \sum_{k=0}^{2N} h(k)w(k)z^{-k}$$

Existen varias funciones ventana aplicables, en este caso se utilizó la ventana de Hamming:

$$w(n) = 0.54 - 0.46 \cos\left(\frac{n\pi}{N}\right)$$



Ventana	Secuencia temporal $0 \leq n \leq N-1$	Anchura aproximada del lóbulo principal	Pico secundario (dB)
Rectangular	1	$4\pi/N$	-13
Bartlett	$\begin{cases} \frac{2n}{N-1}, & 0 \leq n \leq \frac{N-1}{2} \\ 2 - \frac{2n}{N-1}, & \frac{N-1}{2} \leq n \leq N-1 \end{cases}$	$8\pi/N$	-27
Hanning	$\frac{1}{2} \left( 1 - \cos \left( \frac{2\pi n}{N-1} \right) \right)$	$8\pi/N$	-32
Hamming	$0.54 - 0.46 \cos \left( \frac{2\pi n}{N-1} \right)$	$8\pi/N$	-43
Blackman	$0.54 - 0.5 \cos \left( \frac{2\pi n}{N-1} \right) + 0.08 \cos \left( \frac{4\pi n}{N-1} \right)$	$12\pi/N$	-58

Figura 3: Posibles funciones ventana

#### A. Orden de un filtro FIR

Representa el número de muestras anteriores a la actual que se utilizan en un filtro para generar una muestra de salida corresponde al orden del filtro. Un filtro de primer orden utiliza una sola muestra precedente. De esta forma, un filtro no recursivo (FIR) de segundo orden se expresaría con la ecuación siguiente:

$$y[n] = a_0 \cdot x[n] + a_1 \cdot x[n-1] + a_2 \cdot x[n-2]$$

### III. DESARROLLO

#### A. Cálculo de los coeficientes:

El cálculo de los coeficientes se realizó de dos maneras distintas. Una es la utilización de una función propia de Matlab, *fir1()*, como se muestra a continuación.

```
syms h3;
h3 = fir1(10,0.2,'LOW');
freqz(h3, 1,1000); %% Respuesta
frecuencial del filtro
```

Los coeficientes resultantes:

```
h3 =
0.0000 0.0093 0.0476 0.1224 0.2022 0.2370
0.2022 0.1224 0.0476 0.0093 0.0000
```

También se obtuvieron los coeficientes siguiendo los pasos teóricos para el cálculo de los mismos brindados por la cátedra, también utilizando Matlab. Este procedimiento se podrá ver en el anexo I. Los resultados fueron los siguientes:

```
Hz =
0.0000 0.0079 0.0401 0.1033 0.1707 0.2000
0.1707 0.1033 0.0401 0.0079 0.0000
```

Se puede apreciar una pequeña diferencia y esto es porque la función *fir1* no es del todo exacta para la implementación de filtros de bajo orden y baja frecuencia de corte (10 y 0.2Fmax, respectivamente).

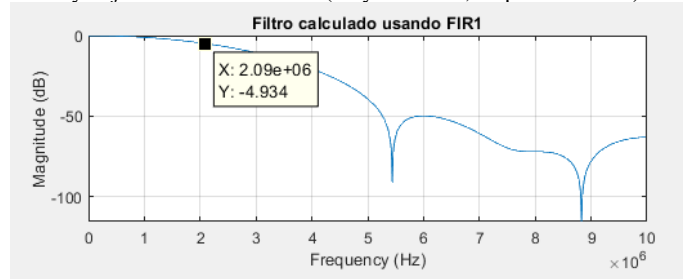


Figura 4: Respuesta del filtro usando FIR –  $F_s = 20\text{kHz}$

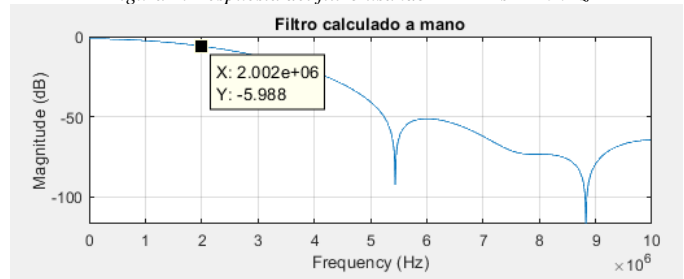


Figura 5: Respuesta del filtro usando los cálculos teóricos –  $F_s = 20\text{kHz}$

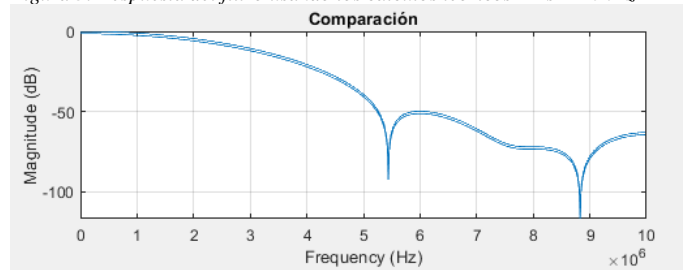


Figura 6: Comparación –  $F_s = 20\text{kHz}$

Una vez que se obtienen los coeficientes, se ingresan en el código para la implementación práctica del filtro.

```
float firState_f32[N_COEFF + 1];
/* Desclaracion del arreglo de la
Señal de Estado. */
float Coeff_f32[N_COEFF] = {0.0000,
0.0079, 0.0401, 0.1033, 0.1707, 0.2000,
0.1707, 0.1033, 0.0401, 0.0079, 0.0000};
```

Código ingresado en Atollic. Anexo II.

### IV. MEDICIONES

La situación extraordinaria obligó a los autores a realizar las mediciones por separado por lo tanto se presentarán en cada apartado dos instrumentos y dos mediciones diferenciadas.

#### A. Instrumental utilizado

Placa de desarrollo Stm32f407-Discovery

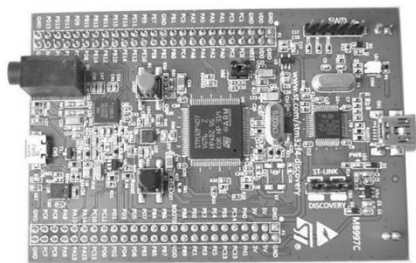


Figura 6: STM32F407VG

Cantalops utilizó, para ver las señales y realizar las mediciones, el osciloscopio UTD2052CL y realizó el barrido de frecuencia con una señal senoidal proveída por el generador UTG9005C (Figuras 7 y 8 respectivamente).

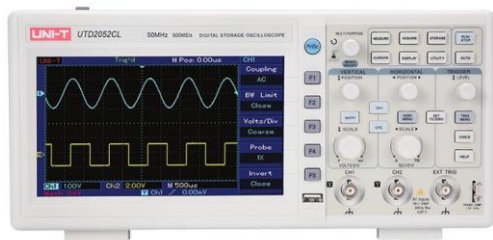


Figura 7: Osciloscopio UTD2052CL



Figura 8: Generador de funciones UTG9005C

Zuccarini, en cambio, utilizó un osciloscopio Hantek 2d42 con generador de señales incluido



Figura 9: Osciloscopio de mano HANTEK 2D4

## B. Resultados obtenidos

Se inyectó una señal senoidal de 1Vpp montada sobre una componente de continua de 1,5V en la entrada de la placa de desarrollo y con el osciloscopio se midieron la entrada (ADC) y la salida (DAC). El resultado de un barrido frecuencial se puede ver en las siguientes tablas:

Vin [V]	Vout [V]	Frec [Hz]	At [dB]
1,19	0,823	358,69	-3,20294
1,23	0,839	400,57	-3,32286
1,27	0,839	449,07	-3,60084
1,27	0,823	556,77	-3,76808
1,19	0,815	648,45	-3,28779
1,27	0,799	757,48	-4,02514
1,27	0,807	850,05	-3,9386
1,27	0,792	948,28	-4,10157
1,31	0,76	1051	-4,72915
1,27	0,752	1150	-4,55172
1,31	0,744	1250	-4,91397
1,27	0,72	1355	-4,92942
1,27	0,72	1454	-4,92942
1,27	0,696	1551	-5,22389
1,27	0,657	1654	-5,72477
1,27	0,649	1755	-5,83118
1,27	0,617	1848	-6,27037
1,23	0,564	1949	-6,77252
1,23	0,594	2001	-6,32237
1,31	0,57	2108	-7,22793
1,23	0,546	2201	-7,05425
1,19	0,522	2302	-7,15753
1,23	0,475	2503	-8,26423
1,23	0,443	2700	-8,87003
1,19	0,348	3000	-10,6794
1,23	0,285	3503	-12,7012
1,23	0,253	4034	-13,7357
1,15	0,308	5060	-11,4429
1,19	0,356	6060	-10,4819
1,15	0,198	10060	-15,2807

Tabla 1- Barrido en frecuencia -Cantalops

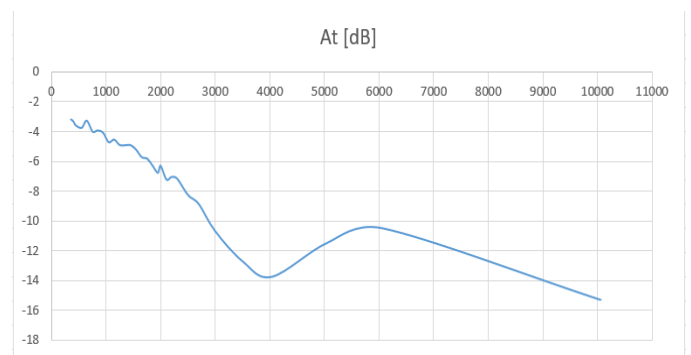


Figura 10 - Gráfica de At en dB vs Frecuencia en Hz -Cantalops

Vin [V]	Vout [V]	Frec [Hz]	At [dB]
1,18	1,02	100	-1,26564



1,1	0,9	150	-1,743
1,1	0,9	200	-1,743
1,1	0,88	250	-1,9382
1,06	0,86	300	-1,81615
1,06	0,86	350	-1,81615
1,06	0,86	400	-1,81615
1,06	0,84	450	-2,02053
1,06	0,84	500	-2,02053
1,04	0,82	550	-2,06439
1,04	0,8	600	-2,27887
1,04	0,8	650	-2,27887
1,06	0,8	700	-2,44432
1,02	0,78	750	-2,33011
1,06	0,78	800	-2,66423
1	0,74	850	-2,61537
1,06	0,76	900	-2,88985
1,06	0,8	950	-2,44432
1,06	0,7	1000	-3,60416
1,04	0,64	1200	-4,21707
1,06	0,52	1400	-6,18605
1,08	0,34	2000	-10,0389
1,1	0,192	2500	-15,1618
1,08	0,2	3000	-14,6479
1,04	0,32	3500	-10,2377
1,02	0,33	4000	-9,80172
1,02	0,3	4500	-10,6296
1,02	0,23	5500	-12,9374
1,02	0,2	6000	-14,1514
1	0,2	8000	-13,9794

Tabla 2- Barrido en frecuencia -Zuccarini

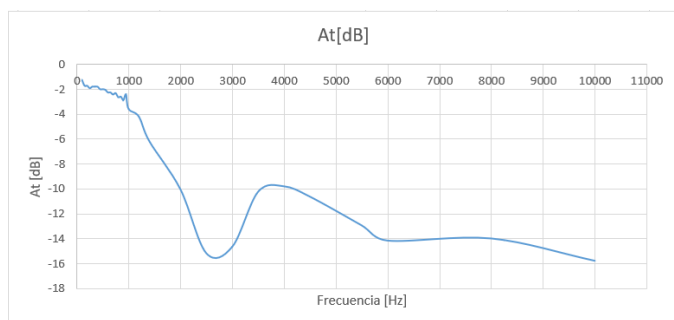


Figura 11 - Gráfica de At en dB vs Frecuencia en Hz - Zuccarini

Se puede notar que la frecuencia de corte de la primera tabla está en 2000Hz y la de la segundo está en 1000Hz, esto se debe a que se utilizaron distintas frecuencias de muestreo 20kHz y 10kHz.

No obstante la discrepancia, esta frecuencia de corte se encuentra a 0.2 de la frecuencia máxima, por lo que el filtro cumple con los requisitos impuestos. Es interesante notar que los coeficientes obtenidos funcionaron para los dos muestreos ya que están normalizados.

## V. CONCLUSION

Luego de la experiencia se puede concluir que es posible realizar un filtro digital sin mayor dificultad gracias a las herramientas disponibles actualmente, siempre teniendo en cuenta el error de aproximaciones que se generan al utilizar distintos métodos. Este error puede resultar significativo o no dependiendo de la exactitud que se requiera.

También se pudo ver la conveniencia de trabajar con coeficientes normalizados ya que uno puede llevarlo a distintas frecuencias de muestreo con tan solo cambiar un número. Ahora, si se requiere cambiar la frecuencia de corte se deberá realizar todo el procedimiento desde cero. Ni hablar si se quiere cambiar el tipo de filtro.

Resulta interesante volver a remarcar algo que parece obvio, tanto que puede ser olvidado. Matlab, como tantos otros softwares de cálculo o simulación trabajan mediante aproximaciones y algoritmos. Estos algoritmos varían y pueden resultar mejor o peor para distintas aplicaciones. Es importante tener en cuenta estas cuestiones a la hora de llevar a cabo un proyecto o experimento, ya sea para calcular los errores cometidos o para resolver algo que no está funcionando correctamente. El estudiante de ingeniería va aprendiendo a lo largo de la carrera que todo tiene su ventaja y su desventaja, más cuando se empieza a hilar fino. A veces, lo barato puede salir caro.

## VI. BIBLIOGRAFIA

- [1] Apuntes de la catedra TdCH – Filtros digitales I y II
- [2] Ayuda de Matlab.

*ANEXO I. CODIGO MATLAB PARA EL CALCULO DE  
COEFICIENTES POR METODO TEORICO*

```
%% FILTRO FIR
Fm = 20e3;
Fc = 0.2* Fm/2;
N = 10/2;
n = linspace(1,N,N);

%% Frecuencia digital
lambda = 2*pi*Fc/Fm;

%% Calculo de coeficientes
k = 2./(n*pi);

a0 = lambda / pi;
an = k.*sin(n*lambda);

%% Ventana de Hamming
n = linspace(0,N-1,N);
w_0a4 = 0.54 - 0.46.*cos(n.*pi/N);
w_6aX = flip(w_0a4);

wn = horzcat(w_0a4, 1, w_6aX);

%% Función ventaneada
Hz1 = (an/2).*w_6aX;
Hz2 = flip(Hz1);
Hz = horzcat(Hz2, 0.2, Hz1)

H3 = fir1(10,0.2)

%% Respuesta frecuencial del filtro
figure(1)
freqz(Hz, 1, 1000);
title('Filtro calculado a mano');
figure(2)
freqz(H3, 1,1000)
title('Filtro calculado usando FIR1');
```

```

/* Includes */
#include "stm32f4xx.h"
#include "stm32_tim2.h"
#include "stm32_adc_single.h"
#include "stm32_dac_single.h"

//-----
void filter_fir_f32(float* pSrc, float* pDst, uint32_t N_COEF, float* pCoeff);

//-----
#define FS 20000 /* Frecuencia de muestreo */
#define N_COEFF 11 /* Numero de coeficientes */

float firState_f32[N_COEFF + 1]; /* Declaracion del arreglo de la Señal de Estado. */
float Coeff_f32[N_COEFF] = {0.0000, 0.0079, 0.0401, 0.1033, 0.1707, 0.2000, 0.1707, 0.1033, 0.0401, 0.0079, 0.0000}; /* Coeficientes del filtro */
//
uint8_t ADC_ready = 0;

```

```

/* ----- */
void TIMER2_ISR_Callback(void)
{
    ADC_ready = 1;
}
//-----
/*
 * Y(n) = A0*X(n)
 */
//-----
void filter_fir_f32(float* pSrc, float* pDst, uint32_t N_COEF, float* pCoeff)
{
    float ACUM;
    uint16_t k;

    ACUM = *pSrc; /* Señal de Entra */

    firState_f32[N_COEF] = ACUM; /* X[n] = Ultimo elemento del buffer */
    ACUM = 0.0f; /* Blanquear Acumulador */

    for(k = 0; k < N_COEF; k++) /* Ciclo de K = 0 a NUM_COEF */
    {
        ACUM = ACUM + pCoeff[k] * firState_f32[k];
        /* "Convolución" */
        firState_f32[k] = firState_f32[k+1];
        /* Corrimiento de Datos */
    }

    *pDst = ACUM; /* Señal de Salida */
}

```

```

int main(void)
{
    float      FIR_IN = 0.0f, FIR_OUT = 0.0f;
    int16_t     signal_IN = 0, signal_OUT = 0;
    uint32_t    period = 0;
    SystemInit();
    ADC_SINGLE_Init();
    DAC_SINGLE_Init();
    ADC_ready = 0;          /* Inicializacion de la bandera del conversor AD. */

    /* F_TIM = 84 MHz / (prescaler+1) / (periode+1) */
    period = 84000000/2/FS - 1;    /* Calculo del periodo para obtener una aproximacion
de la Fs deseada. */

    TIMER2_Init(1,period);/* Inicializacion del timer para que interrumpa a una
Ffrecuencia de Fs. */
    TIMER2_Start();    /* Habilitacion del timer. */

    while(1)
    {
        if(ADC_ready)          /* Bandera habilitada por el timer. */
        {
            ADC_ready    = 0;    /* Clareo de la bandera del conversor. */

            signal_IN = ((int16_t)ADC_SINGLE_Read(ADC_PC2)) - 2048;          /*
Conversion del dato del AD. */
            FIR_IN = ((float)signal_IN) / 4096;          /* Normalizado 0.0 a 1.0.
*/          /* -0.5 a 0.5 */

            filter_fir_f32(&FIR_IN, &FIR_OUT, N_COEFF, Coeff_f32);/* Llamado a la
función de proceso FIR. */

            signal_OUT  = (FIR_OUT * 4096) + 2048; /* Desnormalizado 0 a 4096. */
            DAC_SetDAC2((uint16_t)signal_OUT);    /* Conversion del dato del
DA. */
        }
    }
}

```