



Instituto Tecnológico
de Buenos Aires

INSTITUTO TECNOLÓGICO
DE BUENOS AIRES

22.49 LABORATORIO DE DSP-FPGA (2022)

TRABAJO PRÁCTICO N°3

2 de diciembre de 2022

GRUPO 2	
Legajo	Apellido y nombre
58750	Bertachini, Germán Carlos
60278	Tripoli, Matías
60503	Gonzalez Leiro, Pablo
60557	Demeco, Tobías

Índice

1. Ejercicio a	2
1.1. Diseño	2
1.1.1. Filtro Notch	2
1.1.2. Resultados	3
1.1.3. Resultados	4
2. Ejercicio b	5
2.1. Diseño	5
2.1.1. Filtro Notch	5
2.2. Diseño	6
2.2.1. Filtro Notch	6
2.2.2. Resultados	8
2.2.3. Resultados	8
3. Ejercicio c	9
4. Ejercicio d	9
5. Ejercicio e	10

1. Ejercicio a

Esta sección será dedicada a la implementación de un filtro notch de respuesta impulsiva infinita (IIR), tanto a su diseño como el código que los ejecuta.

1.1. Diseño

El diseño del se realizó en MATLAB utilizando el ejemplo provisto por la cátedra como base y realizando modificaciones sobre este.

1.1.1. Filtro Notch

El filtro IIR diseñado tiene las siguientes especificaciones:

- $f_s = 48kHz$
- $f_c = 4kHz$
- $Q = 1$

La función transferencia, en el dominio de Laplace, de un filtro notch de segundo orden es la siguiente:

$$H(s) = \frac{s^2 + w_0^2}{s^2 + \frac{w_0}{Q}s + w_0^2} \quad (1.1)$$

Para poder operar con más facilidad con la misma dentro del DSP se la convierte mediante una transformación bilineal en $H(z)$.

$$H(z) = \frac{\alpha(1 - 2\cos(\theta_0)z^{-1} + z^{-2})}{\frac{1}{2} - \gamma z^{-1} + \beta z^{-2}} \quad (1.2)$$

Basándonos en la nota de aplicación *Implementing IIR/FIR Filters* para Motorola's DSP56000/SPS/DSP56001, se implementa el siguiente esquema:

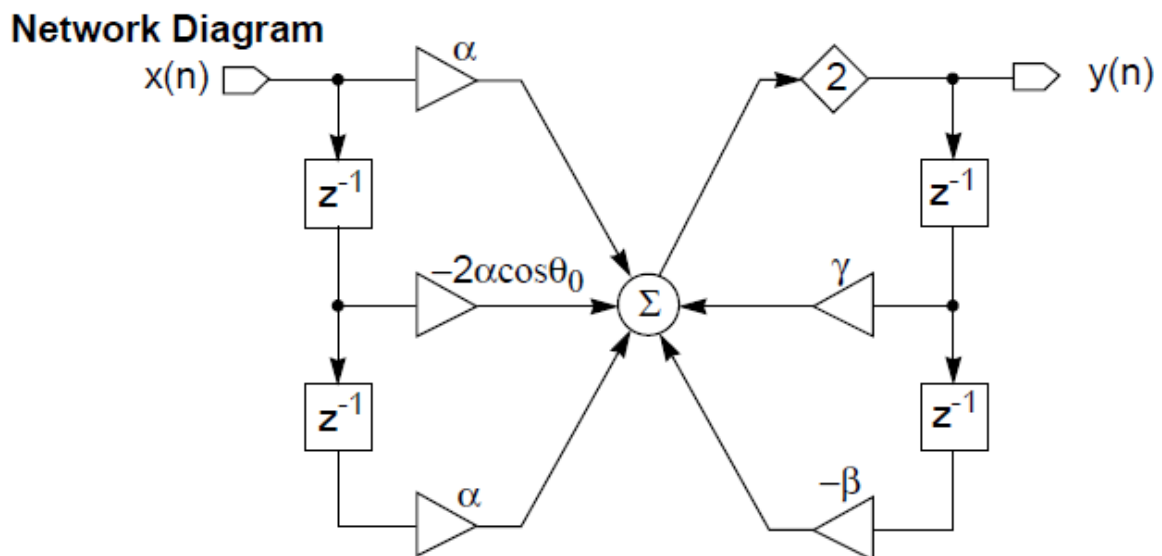


Figura 1.1: Esquema Notch IIR

Se creo un script para calcular los parámetros del filtro bajo el nombre notch_IIR.m. A continuación, se presentan los valores hallados para el filtro rechaza-banda.

Parámetro	Valor
α	0.395709884
β	0.291419769
γ	0.672549180
$\cos(\theta_0)$	0.849800834
σ	1.000000000

Tabla 1.1: Parámetros del IIR

Se simula su diagrama de Bode teórico.

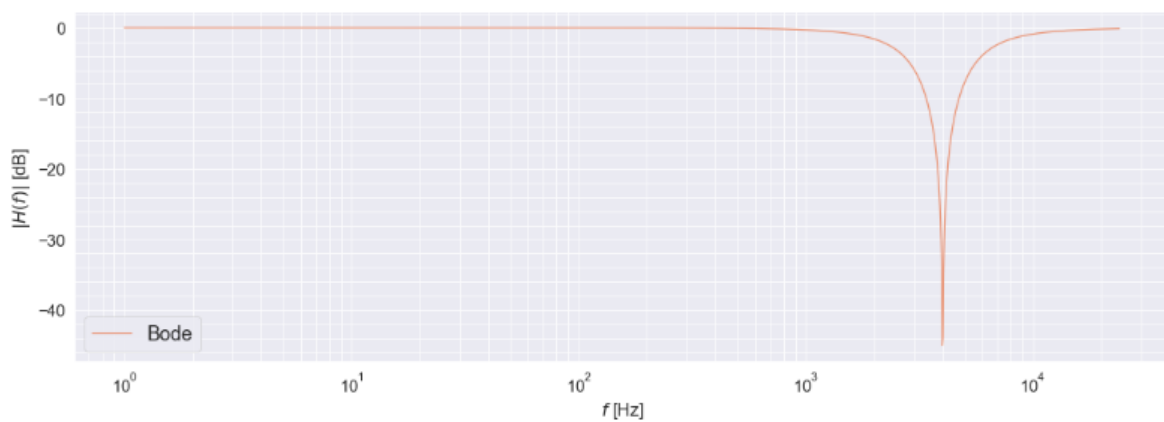


Figura 1.2: Bode medido sobre el DSP

1.1.2. Resultados

Para su implementación, se escribe la siguiente rutina de interrupción para realizar el filtrado IIR deseado.

```

;=====
;  interrupt routine
;=====
RCV EQU *

    ORI  #$08,MR

    MOVE    X:M_RX,Y1

    MPY X0,Y1,A      X:(R0)+,X0  Y:(R4)+,Y0      ;A=ax(n)
    MAC X0,Y0,A      X:(R0)+,X0  Y:(R4),Y0       ;A=A-2acosq0x(n-1)
    MAC X0,Y0,A      X:(R0)+,X0  Y:(R5)+,Y0       ;A=A+alpha*x(n-2)
    MAC X0,Y0,A      X:(R0)+,X0  Y:(R5),Y0        ;A=A+gama*y(n-1)
    MAC X0,Y0,A      X:(R0)+,X0  Y1,Y:(R4)        ;A=A-beta*y(n-2)
    MOVE    A,X1      A,Y:(R5)                    ;y(n)=2A

    MOVEP   X1,X:M_TX

    RTI

```

Figura 1.3: Rutina de interrupción para el filtrado IIR

1.1.3. Resultados

Para validar nuestro diseño, se compila, linkea y corre el programa sobre la DSP56002PV80 de Motorola, usando como código fuente **fl.asm**, que utiliza la rutina de interrupción ya mostrada. Se conecta la entrada de audio de la placa de desarrollo del DSP al generador de señales, y su respectiva salida, al osciloscopio. Se realizan mediciones en un espectro de frecuencia entre los 20Hz y los 20KHz . La frecuencia de muestreo de la placa f_s es 48KHz . En la siguiente tabla, se presenta la diferencia entre las especificaciones de diseño y las mediciones prácticas.

Parámetro	Especificación	Medición	Error porcentual
f_c	4kHz	3.933KHz	1.68%
Δf	4kHz	3.897KHz	2.57%
Q	1	0.9987	1.3%

A continuación, se presenta el diagrama de Bode medido sobre la salida del DSP56002PV80.

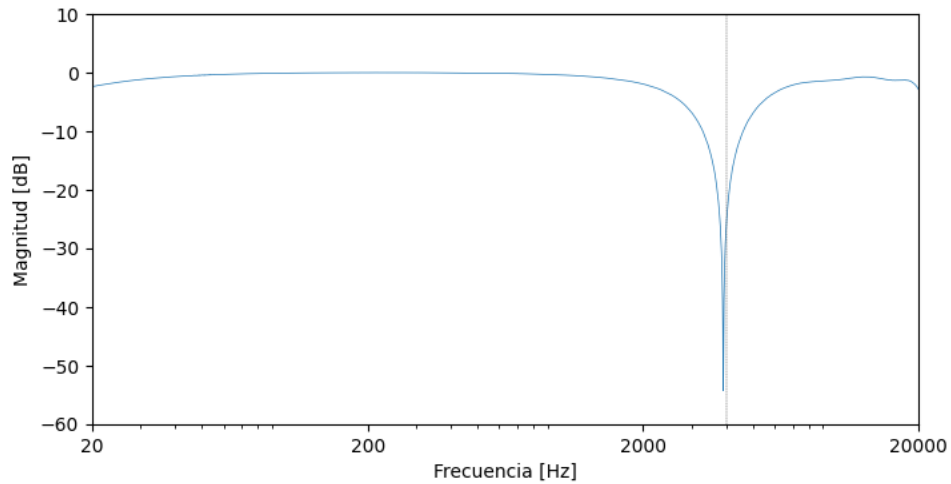


Figura 1.4: Bode medido sobre el DSP

2. Ejercicio b

Esta sección será dedicada a la implementación de un filtro notch de respuesta impulsiva infinita (IIR) de orden 6, tanto a su diseño como el código que los ejecuta.

2.1. Diseño

El diseño del se realizó en MATLAB mediante la herramienta *designfilt* de MATLAB.

2.1.1. Filtro Notch

El filtro IIR diseñado tiene las siguientes especificaciones:

- $f_s = 48kHz$
- $f_c = 4kHz$
- $Q = 1$
- $order = 6$
- $f_{3db1} = 3.6KHz$
- $f_{3db2} = 4.4KHz$

La respuesta en frecuencia teórica es la siguiente:

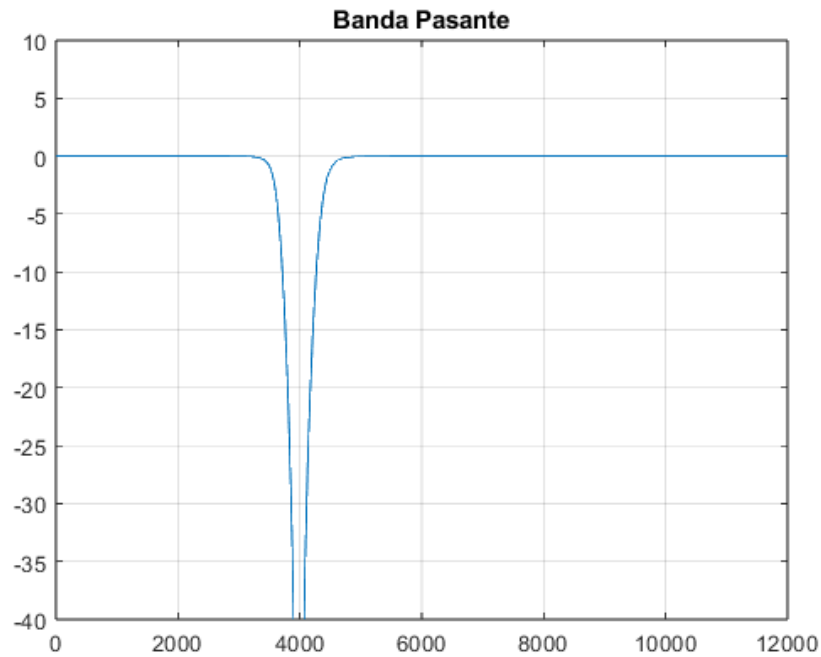


Figura 2.1: Respuesta en frecuencia teórica

2.2. Diseño

El diseño del se realizó en MATLAB utilizando el ejemplo provisto por la cátedra como base y realizando modificaciones sobre este.

2.2.1. Filtro Notch

La función transferencial, en el dominio de Laplace, de cada sección del notches la siguiente:

$$H(s) = \frac{s^2 + w_0^2}{s^2 + \frac{w_0}{Q}s + w_0^2} \quad (2.1)$$

Se realiza el mismo procedimiento que para el punto anterior, para poder operar con más facilidad con la misma dentro del DSP se la convierte mediante una transformación bilineal en $H(z)$.

$$H(z) = \frac{\alpha(1 - 2\cos(\theta_0)z^{-1} + z^{-2})}{\frac{1}{2} - \gamma z^{-1} + \beta z^{-2}} \quad (2.2)$$

Basándonos en las notas de clase, se implementa el siguiente esquema:

Diagrama de red

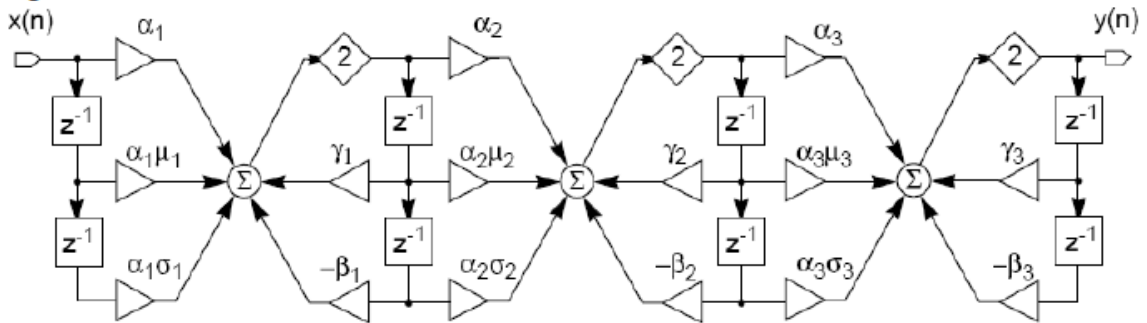


Figura 2.2: Esquema Notch IIR

Los coeficientes calculados mediante la herramienta están presentes en el archivo *coef.fcf*. A partir de los mismos, se calculan los siguientes parámetros:

Parámetro	Valor
α_1	0.493414523
β_1	-0.500017564
γ_1	1.000017564
$\cos(\theta_0)_1$	-2.000036095
σ_1	1
α_2	0.493414523
β_2	-0.500018049
γ_2	1.000018048
$\cos(\theta_0)_2$	-2.000036095
σ_2	1
α_3	0.487241142
β_3	-0.500017588
γ_3	1.000017588
$\cos(\theta_0)_3$	-2.000036095
σ_3	1

Se simula su diagrama de Bode teórico.

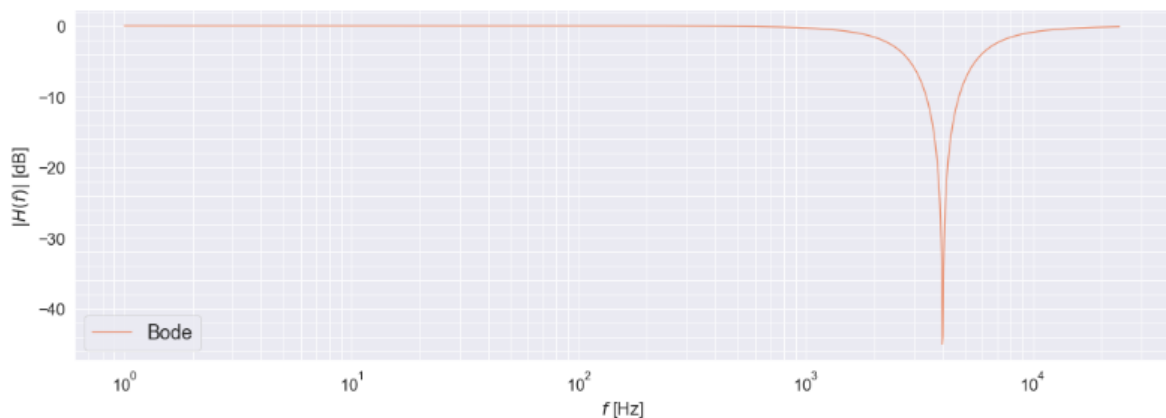


Figura 2.3: Bode medido sobre el DSP

2.2.2. Resultados

Para su implementación, se escribe la siguiente rutina de interrupción para realizar el filtrado IIR deseado.

```

108 ;=====
109 ; interrupt routine
110 ;=====
111 RCV=EQU*
112
113 —→ORI#<math>\$08</math>,MR
114
115 DO X: nsec, Sectn
116 MPY X0, Y1, A X:(R0)+, X0 Y:(R4)+, Y0 ; A = i xi(n)
117 MAC X0, Y0, A X:(R0)+, X0 Y:(R4)+N4, Y0 ; A = A + i  $\sigma$ i xi(n-2)
118 MAC X0, Y0, A X:(R0)+, X0 Y:(R4)+, Y0 ; A = A + i  $\mu$ i xi(n-1)
119 MAC X0, Y0, A Y:(R4)-N4, Y0 ; A = A + i yi(n-1)
120 MAC -X1, Y0, A X:(R0)+, X1 Y1, Y:(R4)+N4 ; A = A -  $\beta$ i yi(n-2) save x(n)
121 MOVE A, Y1 X:(R0)+, X0 ; yi(n) = 2 A (scaling mode is set)
122 Sectn ; X1=  $\beta$ i+1 X0= i+1
123 ; Output: y(n) = Y1
124 MOVE xbuf_len-1, M4 ; Filter order + 1
125 NOP
126 MOVE Y1, Y:(R4)+N4 ; Save y(n)
127 MOVE #1,M4
128 JMP NEXT_SAMPLE
129 —→
130
131 —→RTI
132

```

Figura 2.4: Rutina de interrupción para el filtrado IIR en cascada

2.2.3. Resultados

Para validar nuestro diseño, se compila, linkea y corre el programa sobre la DSP56002PV80 de Motorola, usando como código fuente **f1ej2.asm**, que utiliza la rutina de interrupción ya mostrada. Se conecta de igual manera que para el primer ejercicio. En la siguiente tabla, se presenta la diferencia entre las especificaciones de diseño y las mediciones prácticas.

Parámetro	Especificación	Medición	Error porcentual
f_{3db1}	3.6 KHz	3.521KHz	2.19%
f_{3db2}	4.4 KHz	4.483KHz	1.88%
Q	1	0.9977	2.3%

A continuación, se presenta el diagrama de Bode medido sobre la salida del DSP56002PV80.

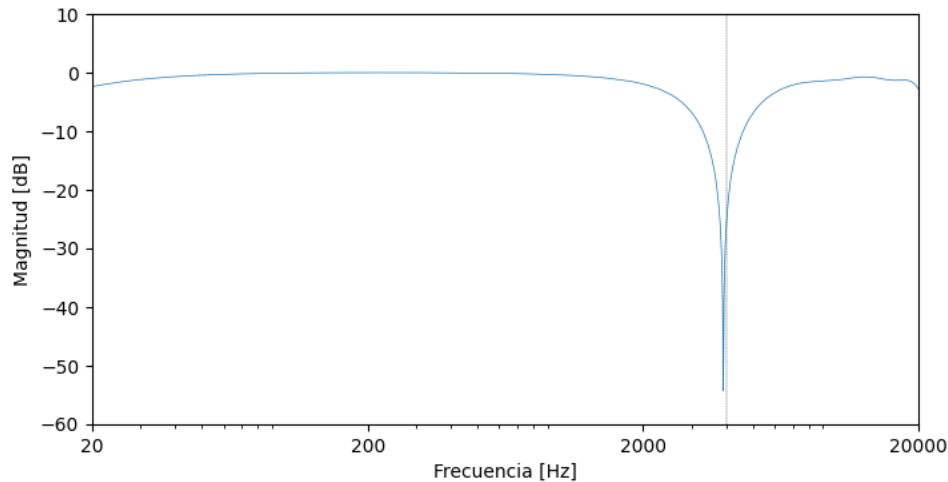


Figura 2.5: Bode medido sobre el DSP

3. Ejercicio c

La definición de rango dinámico es la siguiente:

$$RD[dB] = 20 \log\left(\frac{V_{i\max}}{V_{i\min}}\right) \quad (3.1)$$

Al tratarse de un filtro digital, la limitación en el procesamiento estará atada a la cantidad finita de bits que vayamos a utilizar con su cuantización asociada. Sin considerar los problemas propios del procesamiento digital, como en el caso de un DAC/ADC, el principal inconveniente está asociado con el redondeo (o truncado) de operaciones matemáticas debido a los registros usados, con su debida precisión asociada.

Este truncamiento se lo puede considerar como una fuente de ruido, *round-off noise*, presente en todas las secciones del filtro donde se hacen operaciones matemáticas, sumas y multiplicaciones, limitando el rango dinámico final del filtro. Debido a esto, $V_{i\max}$ estará limitado para evitar casos de *overflow* en secciones internas del filtro. Por otro lado, también se limita $V_{i\min}$ tal que se pueda utilizar el sistema sin verse sumergido por el piso de ruido debido en parte a este *round-off noise*, es indispensable considerar el SNR deseado.

Este es el motivo detrás del cual los acumuladores A y B vistos en clase tienen tantos bits de precisión, que a priori parece excesivo, siendo el objetivo disminuir lo más posible dicho ruido en etapas de cálculo.

Tanto la ganancia como el Q de etapas individuales se ven afectadas por este problema, siendo mandatorio evitar valores de Q muy altos así como también una distribución muy despereja de ganancias. Este criterio de diseño se comparte con los filtros analógicos.

4. Ejercicio d

La cuantización de los coeficientes de los filtros digitales IIR trae consigo un posible problema de estabilidad ya que al cuantizar, se pueden ver afectadas las posiciones de las singularidades del filtro, corriendo el riesgo de obtener polos en el semiplano derecho. Por otro lado, también se ve afectada la ganancia del filtro.

5. Ejercicio e

Para implementar el filtro presentado en el punto b de este trabajo práctico en formato FIR en vez de IIR se volverá indispensable utilizar un filtro de mayor orden para obtener las mismas especificaciones de diseño. De esta manera, habrá un mayor requerimiento de memoria para la respuesta impulsiva y una necesidad de una mayor cantidad de muestras.

Adicionalmente, se tendrá un costo computacional mayor debido al orden del filtro, requiriendo un hardware más rápido para mantener la misma velocidad que se puede tener en un IIR.