

Teoría de los circuitos II

Trabajo final

Filipuzzi, Fernando Rafael

25 de octubre de 2022

Índice general

1	Introducción	7
2	Filtros por funciones de aproximación	9
2.1	Introducción	9
2.2	Filtro de Butterworth	11
2.3	Filtro de Chebyshev	14
2.4	Filtro de Bessel	18
2.5	Comentarios sobre la bibliografía	21
3	Transformación de frecuencia	23
3.1	Filtro pasa alto	23
3.2	Filtro pasa medio	24
3.3	Comentarios sobre la bibliografía	25
4	Diseño de Filtros digitales	27
4.1	Introducción	27
4.2	Filtros IIR	28
4.2.1	Transformación por Método de Euler	29
4.2.2	Transformación invariante al impulso	31
4.2.3	Transformación por mapeo bilineal	34
4.3	Filtros FIR	36
4.4	Comentarios sobre la bibliografía	39
5	Realización de filtros Digitales	41
5.1	Realización	41
5.2	Comentarios sobre la bibliografía	42
6	Procesamiento digital-construcción de filtros digitales	43
6.1	Realización	43
6.2	Comentarios sobre la bibliografía	46
7	Especificaciones de las implementaciones de los filtros realizados	47
7.1	Especificaciones generales del hardware utilizado	47
7.2	Especificaciones de los filtros IIR	53
7.2.1	Filtros pasa bajo	53
7.2.2	Filtros pasa banda	59
7.2.3	Filtros pasa alto	61
7.3	Especificaciones de los filtros FIR	63
7.3.1	Filtro pasa bajo	63
7.3.2	Filtro pasa alto	64

Índice de figuras

2.1	Filtro ideal	9
2.2	Especificaciones de un filtro.	9
2.3	Respuesta en frecuencia del filtro de Butterworth	11
2.4	Diagrama de polos y ceros del filtro de Butterworth	13
2.5	Polinomos de Chebyshev	14
2.6	Función respuesta en frecuencia del filtro de Chebyshev	14
2.7	Ubicación de los polos en el plano imaginario.	16
2.8	Diagrama de polos y ceros del filtro de Chebyshev	18
3.1	Procedimiento de diseño de un filtro no normalizado.	23
3.2	Funciones de atenuación. A la izquierda filtro pasa alto. A la derecha filtro pasa bajo	24
3.3	Relaciones entre las frecuencias de un filtro pasa alto y un filtro pasa bajo.	24
3.4	Funciones de atenuación. A la izquierda filtro pasa banda. A la derecha filtro pasa bajo normalizado .	25
3.5	Relaciones entre las frecuencias de un filtro pasa banda y un filtro pasa bajo normalizado	25
4.1	Pasos para el diseño de filtros digitales	28
4.2	Mapeo del eje $j\omega$, en el plano s, en el circulo unitario, en el plano z	28
4.3	Mapeo del semiplano izquierdo del plano s dentro del circulo unitario en el plano z	29
4.4	Mapeo del eje $j\omega$ en el plano z	30
4.5	Mapeo resultante en el plano z	32
4.6	Mapeo resultante en el plano z	34
4.7	Mapeo resultante en el plano z	35
4.8	Mapeo resultante en el plano z	35
4.9	Representación gráfica de $H(j\omega)$ y de su desarrollo serie de Fourier en $j\omega$	37
4.10	Representación gráfica de la respuesta al impulso $h_d(n)$	37
4.11	Representación gráfica de $h_d(n)$ y $w(n)$ (trazado en rojo)	37
4.12	Representación de $h_d(n)$ y de $h(n)$ para N impar.	38
4.13	Representación de $h_d(n)$ y de $h(n)$ para N par.	38
4.14	Respuesta en frecuencia.	39
5.1	Representación en bloques de elementos computacionales básicos	41
5.2	Forma directa I y II	42
6.1	Diagrama de bloques, procesamiento digital de señales	43
6.2	Representación del algoritmo de un filtro FIR, rutinas en c correspondientes al proceso, asociado al diagrama general de procesamiento digital, 6.1	44
6.3	Implementación con FPGA del bloque de retraso	45
6.4	Implementación del bloque sumador y representación del bloque DAC	45
6.5	Representación del bloque ADC	46
7.1	Montaje del sistema embebido con microcontrolador	47
7.2	Representación en bloques de la arquitectura desarrollada para los filtros digitales	48
7.3	Representación del algoritmo de un filtro FIR	49
7.4	Montaje del sistema embebido con FPGA	50
7.5	Circuito de entrada analógica unipolar incluido en la placa embebida utilizada	50
7.6	Circuito R2R sugerido por la empresa que manufactura la placa embebida	51
7.7	Mapeo resultante en el plano z	52
7.8	Filtro analógico	53
7.9	Filtro digital	53
7.10	Filtro analógico	54
7.11	Filtro digital	54
7.12	Filtro analógico	55

7.13 Filtro digital	55
7.14 Filtro Analógico	56
7.15 Filtro Digital	56
7.16 Filtro analógico	57
7.17 Filtro digital	57
7.18 Filtro Analógico	58
7.19 Filtro Digital	58
7.20 Filtro analógico	59
7.21 Filtro digital	59
7.22 Filtro Analógico	60
7.23 Filtro Digital	60
7.24 Filtro analógico	61
7.25 Filtro digital	61
7.26 Filtro Analógico	62
7.27 Filtro Digital	62

1 Introducción

El objetivo de este trabajo llevar los aspectos teóricos de diseño y realización de filtros digitales a la práctica. En las primeras secciones se busca reunir las definiciones de tal forma que sirvan para la construcción de los script de diseño de los filtros en Matlab.

En la secciones finales se describen el hardware utilizados y las especificaciones de los filtros realizados.

2 Filtros por funciones de aproximación

2.1. Introducción

Filtro ideal

Un filtro ideal es una red circuital que transmite libremente las señales con frecuencias dentro de su banda de transmisión libre, y rechaza a las señales fuera de esa banda. Un filtro ideal no introduce distorsión y debería mantener la relación de fase. La señal se puede retrasar en el filtro, pero todas las componentes de frecuencia deberán retrasarse igualmente.

En la banda de rechazo(atenuación) los filtros no absorben potencia, es decir que rehusan admitir potencia en sus terminales de entrada. Siendo respuesta en frecuencia ideal está dado por la función magnitud como

$$\begin{aligned} |H(s)|^2 &= A \quad \text{para } \omega_1 \leq \omega \leq \omega_2 \\ &= 0 \quad \text{de otro modo} \end{aligned}$$

donde $\omega_1 \leq \omega \leq \omega_2$, este intervalo de frecuencia se conoce como la banda libre o banda de paso y el desplazamiento en dicha banda es cero o 180° , o es lo mismo que decir que la función de fase es proporcional a la frecuencia, Fig. 2.1.

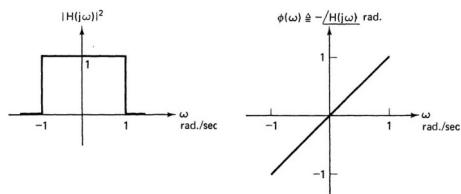


Figura 2.1: Filtro ideal

Entonces, el problema de diseño es que ningún circuito lineal de elementos concentrados puede producir tal función de transferencia exacta. En resumen tenemos que:

- 1- Un filtro de elementos pasivos tienen una función racional de respuesta en frecuencia.
- 2- Una función racional no puede tener una valor constante en ninguna banda.

El problema de la aproximación.

En la práctica se permiten ciertas tolerancias, Fig 2.2, y esto nos lleva a que además de las bandas libres y las bandas de rechazo va a ver bandas de transición.

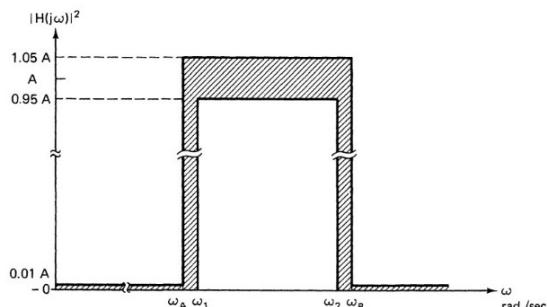


Figura 2.2: Especificaciones de un filtro.

El problema se reduce en encontrar una función $f_a(x, \alpha_1, \dots, \alpha_n)$ que aproxime a otra función $f(x)$ de un filtro ideal en un intervalo $x_1 \leq x \leq x_2$. Donde los parámetros $\alpha_1, \dots, \alpha_n$ en la función aproximación son encontrados por medio

de algún criterio que contemple el error que surge entre la diferencia entre ambas funciones

$$error = f(x) - f_a(x; \alpha_1, \dots, \alpha_n)$$

La selección de la función de aproximación se pueden dividir en los siguientes criterios.

1. Mínimos cuadrados. El valor de $I(\alpha_1, \dots, \alpha_2)$

$$I(\alpha_1, \dots, \alpha_2) = \int_{x_1}^{x_2} |\epsilon|^2 w(x) dx$$

es minimizado, donde $w(x)$ es una función de peso la cual atenúa el error en cierto subintervalos.

2. Máxima planicidad. Las primeras $n - 1$ derivadas del error, ϵ se hacen cero en $x = x_0$.

3- Chebyshev. El valor de μ (constante de riple) es minimizado en el intervalo $x_1 \leq x \leq x_2$ donde $\mu = |\epsilon|_{max}$.

4. Interpolación. Se busca que el valor del error, ϵ , se reduzca para un conjunto de n puntos en el intervalo $x_1 \leq x \leq x_2$.

Una vez que se a tomado alguno de estos criterios se debe determinar la forma de la función aproximación, ya sea en el dominio del tiempo o en el dominio de la frecuencia.

Selección de la función de aproximación en el dominio del tiempo.

Consiste en aproximar una respuesta al impulso $h(t)$ desde el dominio del tiempo con una función $h^*(t)$ tal que el error sea mínimo.

$$\epsilon = \int_0^\infty (h(t) - h^*(t))^2 dt$$

La función va a tener una forma como:

$$f_a(x; \alpha_1, \dots, \alpha_n) = \alpha_1 e^{\alpha_2 x} + \alpha_3 e^{\alpha_4 x} + \dots$$

Un procedimiento efectivo es usar en el dominio del tiempo funciones ortogonales, $\phi(t)$, donde la función $h^*(t)$ toma la forma

$$h^*(t) = \sum_{k=1}^n \alpha_k \phi_k(t)$$

así el error queda expresado como

$$\epsilon = \int_0^\infty \sum_{k=1}^n \alpha_k \phi_k(t)^2 dt$$

y es minimizado cuando

$$h^*(t) = \sum_{k=1}^n \alpha_k \varphi_k(t)$$

Si se eligen funciones ortogonales como una suma de exponentiales $e^{\alpha_k t}$, la respuesta será $h^*(t) = \sum_{k=1}^n \alpha_k \phi_k(t)$ y finalmente tendrá la transformada

$$h^*(t) = \sum_{k=1}^n \frac{\alpha_k}{s - s_k}$$

Selección de la función de aproximación en el dominio de la frecuencia.

El problema trata en que hay que encontrar una función racional $H(s)$ que tenga alguna característica como máxima planicidad o igual riple. Estas características se pueden buscar en la función de magnitud, o en la función de fase. La forma de la función en general va ser como

$$f_a(x; \alpha_2 + \alpha_4 x + \alpha_6 x^2 + \dots)$$

El filtro ideal no es realizable porque la respuesta en el impulso no es cero para $t < 0$. Por ejemplo, en el caso de buscar la máxima planicidad, la función de aproximación va a ser una función racional, y donde se asume que los ceros están en infinito (filtro pasa bajos) llegando a algo como:

$$M(\omega) = \frac{K_0}{[1 + f(\omega^2)]^{1/2}}$$

Donde K_0 es la constante de DC y $f(\omega^2)$ es el polinomio a ser seleccionado.

Y por último, esta función elegida debe cumplir con los criterios de que sea realizable con elementos pasivos o elementos activos en el caso de filtros analógicos(o elementos computacionales para el caso de los filtros digitales).

2.2. Filtro de Butterworth

El parámetro característico es el orden, dependiente de las atenuaciones elegida para cada banda.

Función de aproximación

La función de aproximación queda dada por

$$B_n(\omega) = \omega^n$$

Respuesta en frecuencia

$$|H(j\omega)| = \frac{1}{1 + \omega^{2n}}$$

Propiedades:

1- Valores característicos:

$$|H(j0)|^2 = 1, |H(j1)|^2 = \frac{1}{2}, (3dB), |H(j\infty)|^2 = 0$$

2- Monótona decreciente

3- Magnitud máxima plana, significa que la derivada $(2n-1)$ -ésima es cero.

4- La caída es de $20ndB/dec$ para orden n .

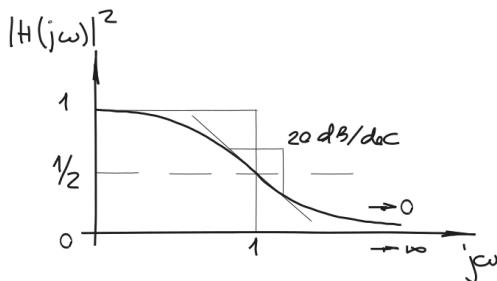


Figura 2.3: Respuesta en frecuencia del filtro de Butterworth

Orden del filtro y atenuación

En base a las características enunciadas anteriormente se llega a que el orden en función de la atenuación requerida según la banda de interés queda como sigue:

$$n = \frac{\log_{10}(10^{A_{db}/10} - 1)}{2\log_{10}\omega}$$

Desarrollo

$$\begin{aligned} |H(j\omega)|_{db}^2 &= 10\log_{10}\frac{1}{1 + \omega^{2n}} = -A_{db} \\ \omega^{2n} &= 10^{A_{db}/10} - 1 \\ 2n\log_{10}\omega &= \log_{10}(10^{A_{db}/10} - 1) \longrightarrow n = \frac{\log_{10}(10^{A_{db}/10} - 1)}{2\log_{10}\omega} \end{aligned}$$

Función de transferencia y polos de la función de transferencia

La función de transferencia tiene la forma:

$$\begin{aligned} |H(s))|^2 &= \begin{cases} \frac{1}{1+s^{2n}} & , \text{ si } n \text{ es par} \\ \frac{1}{1-s^{2n}} & , \text{ si } n \text{ es impar} \end{cases} \\ H(s) &= \prod_{k=1}^n \frac{1}{s - s_k} = \begin{cases} \frac{\prod_{k=1}^{n/2} \frac{1}{s^2 - 2\sin\theta_k + 1}}{s+1} & , \text{ si } n \text{ es par} \\ \frac{1}{s+1} \prod_{k=1}^{n/2} \frac{1}{s^2 - 2\sin\theta_k + 1} & , \text{ si } n \text{ es impar} \end{cases} \end{aligned}$$

La expresión para determinar los polos queda dada por

$$\hat{s}_k = -\sin\left(\frac{2k-1}{2n}\pi\right)\pi + j\cos\left(\frac{2k-1}{2n}\pi\right)$$

donde k y n es un número entero, dado entre $k = 1, 2, \dots, n$.

Desarrollo

Para el caso de n es par, se tiene:

$$1 + s^{\frac{2n}{2}} = 0$$

$$s^{\frac{2k}{2}} = -1 = e^{j\left(\frac{\pi - 2k\pi}{2n}\right)} \quad k=1, \dots, 2n$$

$$s_k = e^{j\left(\frac{2k-1}{2n}\pi\right)} \Rightarrow \theta_k = \frac{2k-1}{2n}\pi \Rightarrow \hat{s}_k = \cos\left(\frac{2k-1}{2n}\pi\right) + j \sin\left(\frac{2k-1}{2n}\pi\right)$$

$$\cos \theta = -\sin\left(\theta - \frac{\pi}{2}\right)$$

$$\left(\frac{2\left(k+\frac{n}{2}\right)-1}{2n}\right)\pi = \pi\left(\frac{2k+n-1}{2n}\right) = \pi\left(\frac{2k-1}{2n}\right) - \frac{\pi n}{2n} = \pi\left(\frac{2k-1}{2n}\right) - \frac{\pi}{2}$$

$$\Rightarrow \hat{s}_k = -\sin\left(\frac{2k-1}{2n}\pi\right) + j \cos\left(\frac{2k-1}{2n}\pi\right)$$

Y en el otro caso, donde n se impar el desarrollo se da en:

$$1 - s^{\frac{2n}{2}} = 0$$

$$s^{\frac{2k}{2}} = 1 = e^{j\left(\frac{\pi - 2k\pi}{2n}\right)} = -j\left(\frac{1-k}{n}\pi\right) \quad k=1, \dots, (2n-1)$$

$$\Rightarrow \theta_k = \left(\frac{k-1}{n}\right)\pi \Rightarrow s_k = -\sin\left(\frac{2k-1}{2n}\pi\right) + j \cos\left(\frac{2k-1}{2n}\pi\right)$$

$$\cos \theta = -\sin\left(\theta - \frac{\pi}{2}\right)$$

$$\frac{k-1}{n}\pi = \frac{k+n-1}{n}\pi = \frac{2k+n-1}{2n}\pi = \frac{2k-1}{2n}\pi + \frac{\pi}{2}$$

Finalmente, los polos conjugados se pueden desarrollar la siguiente expresión:

$$(s - s_k)(s - \bar{s}_k) = (s - (\alpha + j\beta))(s - (\alpha - j\beta)) = 0$$

$$s^2 - (\alpha + j\beta)s - (\alpha - j\beta)s + (\alpha^2 + \beta^2) = 0$$

$$s^2 - 2\alpha s + (\alpha^2 + \beta^2) = 0$$

$$s^2 + 2 \sin(\theta_k)s + 1 = 0$$

Ejemplo con Matlab 1: Determinando polos y ceros para orden 3, Fig. ??.

En este ejemplo se utiliza la expresión dada para los polos conjugados.

```

1 clear all; clc; close all;
2
3 n=3 % orden del filtro
4
5 syms k integer;
6 theta(k)=(2*k-1)/(2*n) *pi;
7 m=floor(n/2);
8
9 syms s;
10 H(s)=1/prod( s^2 + 2*sin(theta(1:m))*s + 1 ) * 1/(mod(n,2)*s+1);
11 H(s)=collect(H(s))
12
13 [num, den]= numden(H(s));
14 B=sym2poly(num); A=sym2poly(den);
15
16 fig=figure;
17 zplane( [zeros(1, n-length(B)+1) B], [zeros(1, n-length(A)+1) A] );
18 grid on; title('Filtro Butterworth');
19 saveas(fig, './butterworth_pz_ej1', 'jpg');

```

Matlab con 2: Determinando polos y ceros para orden 3, Fig. ??.

A diferencia del ejemplo anterior, se determinan la función de transferencia calculando los polos individuales en el semiplano-izquierdo.

```

1 clear all; clc; close all;
2
3 n=3 % orden del filtro
4
5 syms k integer;
6 theta(k)=(2*k-1)/(2*n)*pi;
7
8 sk(k)=-sin(theta(k)) + j*cos(theta(k));
9
10 syms s;
11 H(s)=collect( prod( 1./(s-sk(1:n)) ) );
12
13 [num, den]= numden(1/H(0)*H(s));
14 B=sym2poly(num); A=sym2poly(den);
15
16 fig=figure;
17 zplane( [zeros(1, n-length(B)+1) B], [zeros(1, n-length(A)+1) A] );
18 grid on; title('Filtro Butterworth');
19 saveas(fig,'./butterworth_pz_ej2','jpg');

```

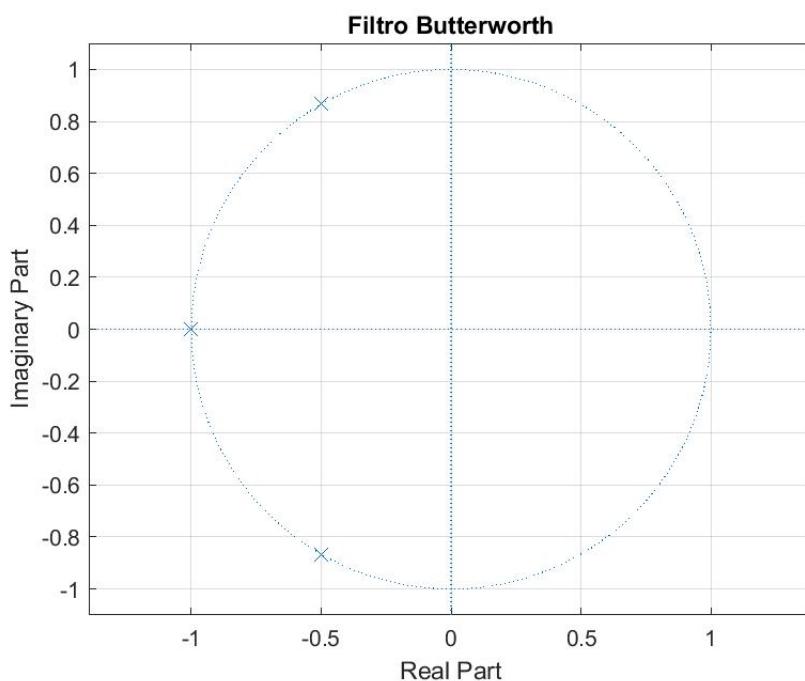


Figura 2.4: Diagrama de polos y ceros del filtro de Butterworth

2.3. Filtro de Chebyshev

La característica principal es la de tener una buena aproximación uniforme en toda la banda de paso. Los parámetros característicos son la constante de riple y el orden, que dependen de los requerimientos de atenuación. Otra característica importante es que la frecuencia de corte es mayor a uno(frecuencia normalizada).

Función de aproximación:

Queda definida por

$$C(\omega) = \begin{cases} \cos(n \cdot \cos^{-1}(\omega)) & , \text{ si } |\omega| \leq 1 \\ \cosh(n \cdot \cosh^{-1}(\omega)) & , \text{ si } |\omega| > 1 \end{cases}$$

y la expresión dada en forma recursiva sería:

$$C(\omega) = \begin{cases} 1 & , \text{ si } n = 0 \\ \omega & , \text{ si } n = 1 \\ 2\omega C_{n-1} - C_{n-2}(\omega) & , \text{ para otro valor} \end{cases}$$

Propiedades:

1- Los valores característicos son:

$$C_n(1) = 1, C_n(0) = \begin{cases} 0 & , \text{ si } n \text{ es impar} \\ 1 & , \text{ si } n \text{ es par} \end{cases}$$

2- Los ceros están en el intervalo $|\omega| < 1$.

3- Dentro del intervalo, $|\omega| \leq 1$, no excede la unidad.

4- Fuera del intervalo, el valor absoluto de la función se incrementa rápidamente.

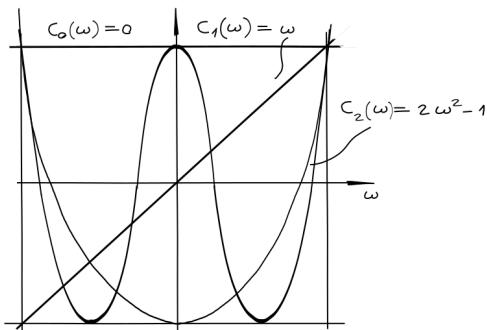


Figura 2.5: Polinomos de Chebyshev

Función de respuesta en frecuencia

$$|H(j\omega)| = \frac{1}{1 + \epsilon^2 C_n^2(\omega)}$$

donde ϵ es la constante de riple, y $C_n(\omega)$ es la función de chebyshev de orden n . En el intervalo $|\omega| \leq 1$ oscilará desde un máximo en 1 y un mínimo en $\frac{1}{1+\epsilon^2}$, y fuera de este intervalo se aproxima a cero rápidamente.

Propiedades:

1- Los valores característicos son:

$$|H(j1)|^2 = \frac{1}{1 + \epsilon^2} \text{ para todo } n$$

$$|H(j0)|^2 = \begin{cases} 1 & \text{para } n \text{ impar} \\ \frac{1}{1+\epsilon^2} & \text{para } n \text{ par} \end{cases}$$

2- Dentro de la banda de paso, $0 \leq \omega \leq 1$, $H(j\omega)$ oscilará entre los valores de 1 a $\frac{1}{\sqrt{1+\epsilon^2}}$ y entonces la amplitud del riple queda definida por:

$$\text{Ripple} = 1 - \frac{1}{\sqrt{1 + \epsilon^2}}$$

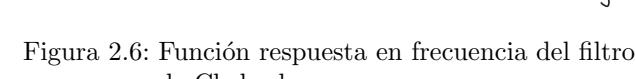


Figura 2.6: Función respuesta en frecuencia del filtro de Chebyshev

Función del filtro y atenuación:

Teoría de los circuitos II

La atenuación esta dada por:

$$\alpha_n = -10 \log_{10} |H_n(j\omega)|^2 = 10 \log |1 + \epsilon^2 C_n^2(\omega)| \text{ dB}$$

En la banda de rechazo, $\omega > \omega_p$, la función de aproximación de Chebyshev es:

$$C_n(\omega) = \cosh(n \cdot \cosh^{-1}\omega))$$

y por lo tanto la atenuación en esta banda es:

$$\alpha_n = 10 \log |1 + \epsilon^2 \cosh^2(n \cdot \cosh^{-1}\omega)| \text{ dB} \quad (2.1)$$

donde es dependiente de la constante de riple ϵ y del orden n .

Además, el orden n_s requerido para la mínima atenuación deseada en la banda de rechazo se despeja evaluando la ecuación 2.1 con ω_s :

$$n_s = \frac{\cosh^{-1}\left(\frac{\sqrt{10^{A_{dB}/10}-1}}{\epsilon}\right)}{\cosh^{-1}(\omega_s)}$$

De igual forma, el orden n_{3db} requerido en la frecuencia de corte (atenuación 3dB) es: ω_c

$$n_{3dB} = \frac{\cosh^{-1}\left(\frac{\sqrt{10^{3dB/10}-1}}{\epsilon}\right)}{\cosh^{-1}(\omega_c)}$$

El orden del filtro se estima como el máximo orden que cumpla los requerimientos de atenuación dados.

$$n = \max(n_s, n_{3dB})$$

El riple se despeja de la atenuación máxima deseable en la banda de rechazo, que en $H(j1)$, y es:

$$\alpha_{max} = 10 \log_{10}(1 + \epsilon^2)$$

quedando que el riple es:

$$\epsilon = \sqrt{10^{\alpha_{max}/10} - 1}$$

Desarrollo:

$$\begin{aligned} 10 \log \left| \frac{1}{1 + \epsilon^2 \cosh^2(n \cdot \cosh^{-1}\omega)} \right| &= -A_{dB} \quad \cancel{\text{if } \cosh^2(n \cdot \cosh^{-1}\omega_s) = \frac{\sqrt{10^{A_{dB}/10} - 1}}{\epsilon^2}} \\ n &= \boxed{\frac{\cosh^{-1}\left[\sqrt{\frac{10^{A_{dB}/10}}{\epsilon^2} - 1}\right]}{\cosh^{-1}\omega_s}} \end{aligned}$$

Función de transferencia y polos de la función de transferencia

La función de transferencia va a ser una función racional en la variable compleja s y va a tener la forma

$$H(s) = \prod_{k=1}^n \frac{1}{s - s_k}$$

donde s_k son polos en semiplano izquierdo, k son valores enteros definidos desde $k = 1, \dots, n$, donde n es el orden del filtro y $s_k = (\sigma'_k + j\omega'_k)\cosh(\beta_k)$ son las raíces del denominador, para la cual se tiene que:

$$\sigma'_k = \tanh(\beta_k) \sin\left(\frac{2k-1}{2n}\pi\right), \omega'_k = \cos\left(\frac{2k-1}{2n}\pi\right), \beta_k = \frac{\sinh^{-1}(1/\epsilon)}{n}$$

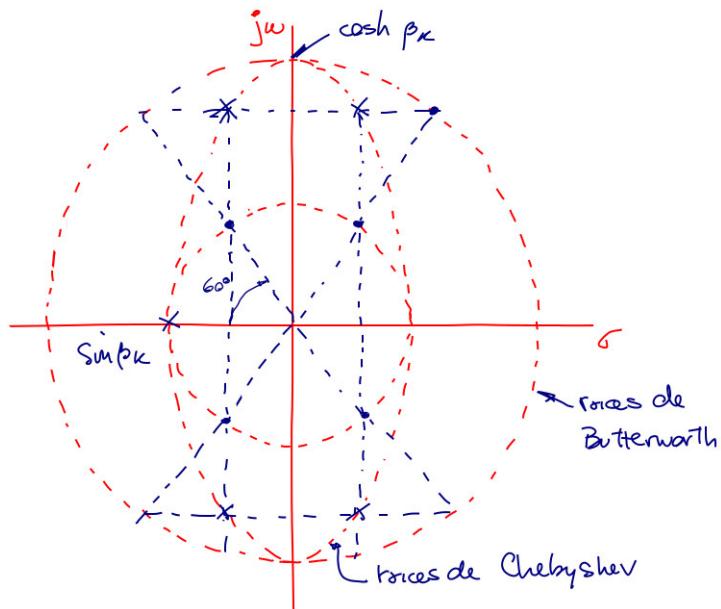


Figura 2.7: Ubicación de los polos en el plano imaginario.

Los polos en el filtro Chebyshev se ubicaran sobre una elipse, con el eje mayor sobre el eje imaginario, Fig.2.7. Así que cuanto más delgada sea la elipse más influencia tendrán los polos sobre el eje imaginario, y mayor riple tendrá.

Desarrollo de la expresión de las raíces:

Para encontrar los polos, se busca los ceros del denominador, así que primero hay que desarrollar el denominador como sigue:

$$\begin{aligned}
 1 + [C_n(\omega)]^2 &= 0 \\
 [1 + j C_n(\omega)][1 - j C_n(\omega)] &= 0 \Rightarrow C_n(\omega) = \pm \frac{1}{\epsilon} j \\
 \cos n \underbrace{\cos^{-1}\left(\frac{s}{\epsilon}\right)}_{\alpha_k + j \beta_k} &= \pm \frac{1}{\epsilon} j \\
 \cos(n\alpha_k + j n \beta_k) &= \underbrace{\sin n \alpha_k \cdot \cosh n \beta_k}_{=0} - j \underbrace{\cos n \alpha_k \cdot \sin n \beta_k}_{\pm 1/\epsilon} = \pm \frac{1}{\epsilon} j
 \end{aligned}$$

dando como soluciones

$$\begin{aligned}
 \bullet \quad & \cosh n \alpha_k \cdot \cosh n \beta_k = 0 \Rightarrow \alpha_k = \frac{\pi(2k-1)}{2n} \Rightarrow \alpha_k = \frac{\pi(2k-1)}{2n} = \frac{\pi}{2n}, \frac{3}{4n}\pi, \dots \\
 \bullet \quad & \sin n \alpha_k \cdot \sinh n \beta_k = \pm \frac{1}{\epsilon} \\
 & \Rightarrow \beta_k = \frac{\sinh^{-1}\left(\frac{(-1)^k}{\epsilon}\right)}{n} \\
 & k = 1, 2, 3, \dots
 \end{aligned}$$

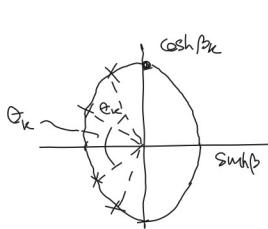
Despejando s_k se tiene:

$$\cos^{-1} \frac{s_k}{j} = \alpha_k + j\beta_k \Rightarrow s_k = j \cos(\alpha_k + j\beta_k)$$

$$s_k = j [\cos \alpha_k \cosh \beta_k - j \sin \alpha_k \sinh \beta_k]$$

$$s_k = \underbrace{\sin \alpha_k \sinh \beta_k}_{\sigma_k} + j \underbrace{\cos \alpha_k \cosh \beta_k}_{\omega_k}$$

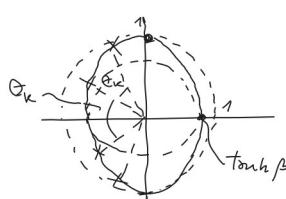
y por identidad trigonométrica, la ubicación geométrica de los polos queda dada en una elipse, tal como se muestra en Fig.2.7



$$\sigma_k = \sin \left(\frac{\pi(2k-1)}{2n} \right) \cdot \sinh \left[\frac{\sinh^{-1} \left(\frac{(\pm 1)^k}{e} \right)}{n} \right] = |A| \sin \theta_k \quad \Rightarrow \quad \frac{\sigma_k^2}{|\sinh \beta_k|^2} + \frac{\omega_k^2}{|\cosh \beta_k|^2} = 1$$

$$\omega_k = \cos \left(\frac{\pi(2k-1)}{2n} \right) \cdot \cosh \left[\frac{\sinh^{-1} \left(\frac{(\pm 1)^k}{e} \right)}{n} \right] = |B| \sin \theta_k$$

La normalizando de las raíces queda como sigue:



$$\sigma'_k = \frac{s_k}{\cosh \beta_k} = \frac{\sigma_k}{\cosh \beta_k} + j \frac{\omega_k}{\cosh \beta_k} = \sigma'_k + j \omega'_k$$

$$\sigma'_k = \frac{\sinh \beta_k}{\cosh \beta_k} \cdot \sin \left(\frac{2k-1}{2n} \pi \right) \Rightarrow \boxed{\sigma'_k = \tanh \beta_k \cdot \sin \left(\frac{2k-1}{2n} \pi \right)}$$

$$\omega'_k = \frac{\cosh \beta_k}{\sinh \beta_k} \cdot \cos \left(\frac{2k-1}{2n} \pi \right) \Rightarrow \boxed{\omega'_k = \cos \left(\frac{2k-1}{2n} \pi \right)}$$

$$\boxed{\beta_k = \frac{\sinh^{-1} \left(\frac{(\pm 1)^k}{e} \right)}{n}}$$

$$\boxed{s_k = (\sigma'_k + j \omega'_k) \cosh \beta_k}$$

Ejemplo con Matlab 1: Determinando polos y ceros para orden 3, Fig. ??.

```

1 clear all; clc; close all;
2
3 n=3           % orden del filtro
4 e=0.2;        % constante del triple
5
6 syms k integer;
7 theta(k)=(2*k-1) / (2*n) *pi;
8 beta(k)=asinh((1)^k/e)/(n);
9
10 sk(k)=tanh(beta(k)).*(-1).*sin(theta(k)) + j*cos(theta(k));
11
12 syms s;
13 H(s)=collect( prod( 1./(s-sk(1:n)) ) );
14
15 [num, den]= numden(1/H(0)*H(s));
16 B=sym2poly(num); A=sym2poly(den);
17
18 fig=figure;
19 zplane( [zeros(1, n-length(B)+1) B], [zeros(1, n-length(A)+1) A] );
20 grid on; title('Filtro Chebyshev');
21 saveas(fig,'./chebyshev_pz_ej1','jpg');

```

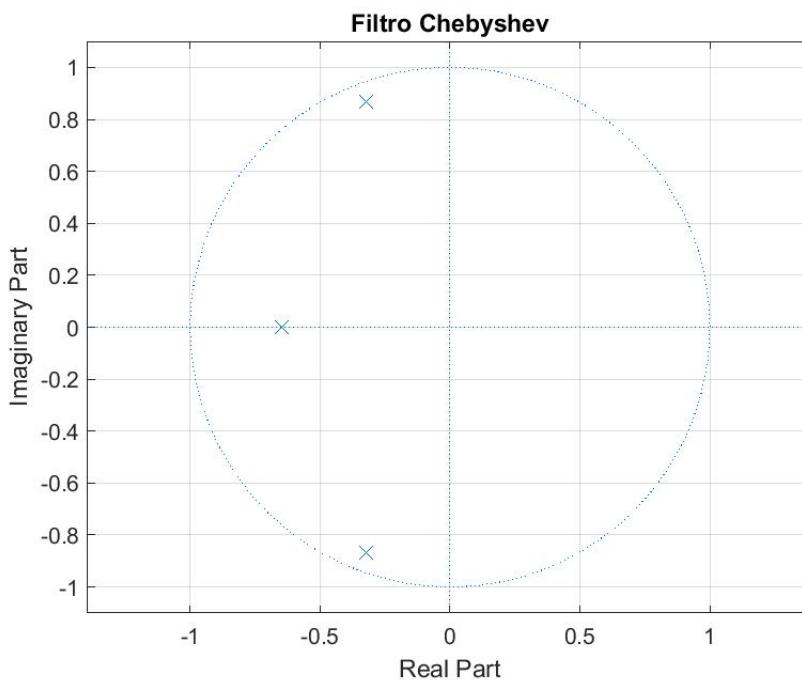


Figura 2.8: Diagrama de polos y ceros del filtro de Chebyshev

2.4. Filtro de Bessel

Este tipo de filtro consiste en relacionar las especificaciones de retardo de fase lineal o de grupo constante del filtro ideal con los polinomios de la función de transferencia.

El retardo de grupo es de particular importancia cuando es relevante las formas de onda de la señal. Por ejemplo para señales de audio, no es perceptible los cambio de fase, pero para señales de vídeo es importante conservar la forma de onda. Las funciones obtenidas de esta manera tienen un retraso plano máximo en el origen.

Función de aproximación:

El polinomio de Bessel queda definido en forma cerrada por:

$$B_n(\tau_0 s) = s^n + b_{n-1}s^{n-1} + \dots + b_1s + b_0$$

donde es $b_k = \frac{(2n-k)!}{2^{n-k}k!(n-k)!}$ y n es el orden del polinomio. Por otro lado, la forma recursiva esta dada por:

$$B_n(s) = \begin{cases} 1 & , \text{ para } n = 0 \\ s + 1 & , \text{ para } n = 1 \\ (2n - 1)B_{n-1}(s) - B_{n-2}(s) & , \text{ otro valor} \end{cases}$$

Función de transferencia

$$H(s) = \frac{k}{e^{\tau_0 s}} = \frac{K}{s^n + b_{n-1}s^{n-1} + \dots + b_1s + b_0}$$

donde $K = b_0$, y los coeficientes del polinomio se pueden obtener de dos formas:

- 1- Igualación de parámetros, o
- 2- Desarrollo de polinomios de Bessel.

1- Método por igualación de parámetros

Planteando la función de transferencia en s tal como:

$$H(s) = \frac{K}{e^{\tau_0 s}} = \frac{K}{\cosh(s) + \sinh(s)} = \frac{K}{s^n + b_{n-1}s^{n-1} + \dots + b_1s + b_0} = \frac{K}{M(s) + N(s)} = K \left(\frac{M(s) - N(s)}{M^2(s) - N^2(s)} \right)$$

donde el par transformado de $H(s)$ viene dado por

$$h(t) = K\delta(t - \tau_0) \xrightarrow{\mathcal{L}} H(s) = Ke^{-\tau_0 t}$$

y $H(s)$ se relaciona con la función de transferencia a partir de los polinomios en s , donde la fase quedada dada por

$$\varphi(s) = \operatorname{atanh} \left(-\frac{N(s)}{M(s)} \right)$$

El grupo de retardo, $\tau(s)$, es la derivada de la fase, así que siendo

$$\frac{d}{dx} \operatorname{atan}(u) = \frac{du/dx}{1 - u^2}$$

se tiene entonces que

$$\tau(s) = \frac{\partial \varphi(s)}{\partial s} = \frac{u'}{1 - u^2} = \frac{-1}{1 - \frac{N^2(s)}{M^2(s)}} = \left(\frac{N'(s)M(s) - N(s)M'(s)}{M^2(s)} \right) = \frac{\frac{N(s)M'(s) - N'(s)M(s)}{M^2(s)}}{\frac{M^2(s) - N^2(s)(s)}{M^2(s)}}$$

Finalmente,

$$\tau(s) = \frac{N(s)M'(s) - N'(s)M(s)}{M^2(s) - N^2(s)}$$

Por ejemplo, para un orden 2, la función de transferencia,

$$H(s) = \frac{K_0}{s^2 + b_1s + b_0} = \frac{K_0}{(s^2 + b_0) + (b_1s)}$$

en donde $M(s) = s^2 + b_0$, $\frac{dM(s)}{ds} = 2s$, $N(s) = b_1s$, $\frac{dN(s)}{ds} = b_1$ y reemplazando estas relaciones en $\tau(s)$ se llega a:

$$\tau(s) = \frac{b_1s^2 + b_1b'}{s^2(2b_0 - b_1^2)s^2 + b_0^2} = \frac{-2(s)(b_1s) + (s^2 + b_0)b_1}{(s^2 + b_0)^2 - (b_1s)^2} = \frac{-2b_1s^2 + b_1s^2 + b_1b_0}{s^4 + 2b_0s^2 + b_0^2 - b_1^2s^2} = \frac{(-2b_1 + b_1)s^2 + b_1b_0}{s^4 + (2b_0 - b_1^2)s^2 + b_0^2}$$

Para despejar las constantes, se tienen en cuenta las siguientes características del filtro.

1- De la planicidad del retardo surge: $\tau(0) = \frac{b_1b_0}{b_0^2}$, por lo tanto $b_1 = b_0$.

2- Siendo constante el retardo de fase se tiene que $b_1 = 3$.

quedando que:

$$\tau(s) = \frac{(-b_1s^2 + b_1b_0)}{s^4 + (2b_0 - b_1^2)s^2 + b_0^2} = 1$$

se despeja

$$\begin{aligned} -b_1s^2 + b_1b_0 - s^4 - (2b_0 - b_1^2)s^2 - b_1b_0 &= 0 \\ s^4 + (b_1 + 2b_0 - b_1^2)s^2 &= 0 \end{aligned}$$

Finalmente,

$$\begin{cases} b_1^2 + (-2b_1 - 2b_0) = 0 \\ b_1 = b_0 \end{cases} \xrightarrow{\quad} \begin{cases} -b_1^2 - 3b_1 = 0 \\ -b_1 = 3 \end{cases}$$

2- Desarrollo de polinomios de Bessel

Partiendo de que

$$H(s) = \frac{K}{e^{\tau_0 s}} = \frac{K}{\cosh(s) + \sinh(s)}$$

la fase se podría plantear mediante el cociente de $\cosh(s)$ y de $\sinh(s)$, y luego con los desarrollos de serie de Taylor se puede llegar a

$$\varphi_2(s) = \frac{\cosh(s)}{\sinh(s)} = \frac{1 - \frac{s^2}{2!} + \frac{s^4}{4!} - \frac{s^6}{6!} + \dots}{1 - \frac{s^3}{3!} + \frac{s^5}{5!} - \frac{s^7}{7!} + \dots} = \frac{1}{s} + \frac{1}{\frac{3}{s} + \frac{\frac{1}{5}}{\frac{7}{s} + \dots}}$$

para el caso de un orden 2, se puede asociar el numerador de la fase a la parte par del denominador del polinomio de la función de transferencia, y de igual forma para con el denominador de la fase, quedando que

$$\varphi_2(s) = \frac{1}{s} + \frac{1}{\frac{1}{s}} = \frac{3 + s^2}{3s} = \frac{M(s)}{N(s)}$$

Entonces la función de transferencia queda como;

$$H(s) = \frac{K}{M(s) + N(s)} = \frac{K}{1+s^2+3s} \text{ donde } k = 3$$

El desarrollo de la división continuada queda como sigue:

$$\begin{array}{r}
 \overbrace{s + \frac{s^3}{3!} + \frac{s^5}{5!} + \frac{s^7}{7!} + \dots}^{(1)} \quad \overbrace{1 + \frac{s^2}{2!} + \frac{s^4}{4!} + \frac{s^6}{6!} + \dots}^{(2)} \\
 \hline
 \overbrace{1 + \frac{s^2}{3!} + \frac{s^4}{5!} + \frac{s^6}{7!} + \dots}^{(3)} \quad \overbrace{\frac{3}{s}}^{(4)} \\
 \hline
 \overbrace{0 + \frac{s^2}{3} + \frac{s^4}{3!5} + \frac{s^6}{3!5!7} + \dots}^{(5)} \quad \overbrace{s + \frac{s^3}{3!} + \frac{s^5}{5!} + \frac{s^7}{7!} + \dots}^{(6)} \\
 \hline
 \overbrace{s + \frac{s^3}{2!5} + \frac{s^5}{2!(4!)^2} + \dots}^{(7)} \quad \overbrace{\frac{5}{s}}^{(8)} \\
 \hline
 \overbrace{0 + \frac{s^2}{3!5} + \frac{s^4}{3!5!7} + \dots}^{(9)} \quad \overbrace{\frac{7}{s}}^{(10)} \\
 \hline
 \overbrace{0 + \frac{s^4}{3!5!7} + \dots}^{(11)} \quad \overbrace{\frac{3}{s} + \frac{s^5}{3!5} + \dots}^{(12)} \\
 \hline
 \overbrace{\frac{3}{s} + \dots}^{(13)} \\
 \hline
 0 + \dots
 \end{array}$$

2.5. Comentarios sobre la bibliografía

Lam Harry. Analog and digital filters design and realization.
Desarrollo general.

Pactitis S. A. Active Filters: Theory and design
Desarrollo del Filtro de Chebyshev.

Weinberg. Network analysis and synthesis
Tiene mejor desarrollado la introducción de funciones de aproximación (pag.365).
Las expresiones para calcular los polos del filtro de Chebyshev.

Kendal Su. Analog Filters.
Mejor desarrollo de los filtros de Bessel.

Van Valkenburg, Mac Elwyn. Introduction to modern network synthesis.
Sobre los polos del filtro de Bessel

Daryanani G. Principles of active network synthesis and design.
Ejemplo de filtros de Bessel

Arthur Williams, Fred J. Taylor. Electronic Filter Design Handbook, Fourth Edition

3 Transformación de frecuencia

Un procedimiento para el diseño de filtros en otras estructuras diferentes a un filtro pasa bajo normalizado se consigue al trasladar los requerimientos de algunos estructuras de filtros no normalizadas a un filtro pasa bajo normalizado, para luego aplicar el procedimiento de diseño de un filtro pasa bajo normalizado, y finalmente, por medio de alguna función de transformación, se transforma la función de transferencia del filtro obtenido a la función de transferencia del filtro no normalizado de interés. Este procedimiento se resumen en la Fig.3.1

3.1. Filtro pasa alto

Función de transformación

La función de transformación a un filtro pasa alto (HP) no normalizado, $T_{HP}(s)$, desde un filtro pasa bajo normalizado (LP), $T_{LP}(s)$, es:

$$S = \frac{\omega_p}{s} \quad (3.1)$$

o bien, siendo $S = j\Omega$ y $s = j\omega$, con la siguiente función de transformación

$$\Omega = \frac{-\omega_p}{\omega} \quad (3.2)$$

Función de resolución de filtros paso alto.

1- Al conjunto de requerimientos dados de un filtro pasa alto:

$$(A_{max}, \omega_p, A_{min}, \omega_s)$$

donde,

A_{max} , Atenuación máxima en la frecuencia dada en la banda de paso, ω_p

ω_p , Frecuencia en la banda de paso dada.

A_{min} , Atenuación mínima en la frecuencia dada en la banda de paso de rechazo, ω_s .

ω_s , Frecuencia en la banda de rechazo dada.

se transforman con la expresión dada en Ec. 3.1 a los requerimientos equivalentes del filtro paso bajo normalizado

$$(A_{max}, 1, A_{min}, \Omega_s = \omega_p/\omega_s)$$

2- Seguido, se determina la función que aproxime los requerimientos pasa bajos, según la aplicación requerida por el filtro (Butterworth, Chebyshev o Bessel).

3- Por último, la función $T_{LP}(S)$ se transforma a la función $T_{HP}(s)$ con la Ec. Ec. 3.1, quedando.

$$T_{HP}(s) = T_{LP}(S)|_{S=\omega_p/s}$$

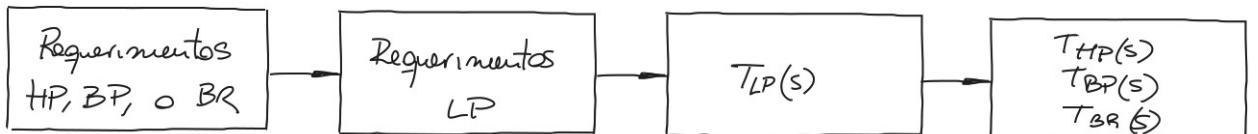


Figura 3.1: Procedimiento de diseño de un filtro no normalizado.

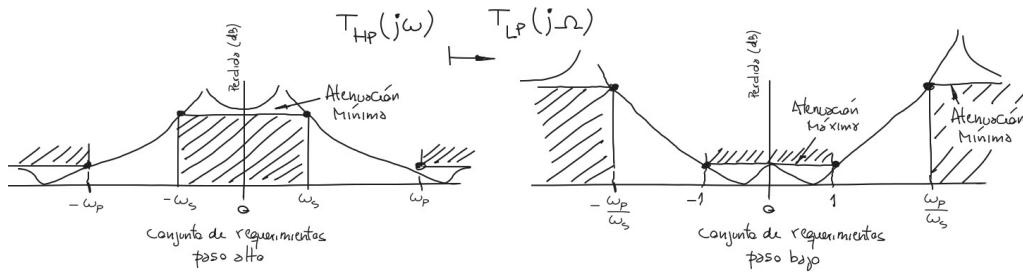


Figura 3.2: Funciones de atenuación. A la izquierda filtro pasa alto. A la derecha filtro pasa bajo.

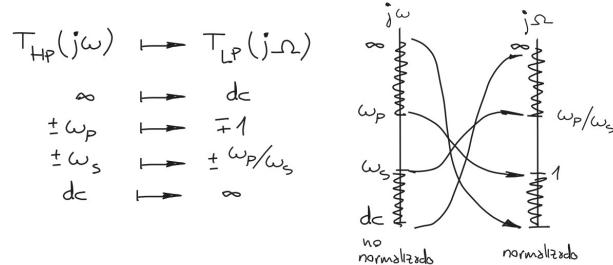


Figura 3.3: Relaciones entre las frecuencias de un filtro pasa alto y un filtro pasa bajo.

3.2. Filtro pasa medio

Función de transformación

La función de transformación de a filtro pasa banda (BP) no normalizado, $T_{BP}(s)$, desde un filtro pasa bajo (LP) normalizado, $T_{LP}(S)$, es:

$$S = \frac{s^2 + \omega_0^2}{B \cdot s} \quad (3.3)$$

o bien

$$\Omega = \frac{\omega_p^2 - \omega_0^2}{B \cdot \omega} \quad (3.4)$$

donde,

$B = |\omega_1 - \omega_2|$, es el ancho de banda de paso del filtro.

$\omega_0^2 = \omega_1 \cdot \omega_2 = \omega_3 \cdot \omega_4$, es la frecuencia central (media geométrica) del filtro pasa banda.

Procedimiento de resolución de filtros paso alto.

1-Al conjunto de requerimientos de un filtro pasa alto:

$$(A_{max}, \omega_1, \omega_2, A_{min}, \omega_3, \omega_4)$$

se transforman con la función dada en la Ec. 3.4 a los requerimientos equivalentes del filtro pasa bajo normalizado (LP), quedando que

$$(A_{max}, \Omega_p = 1, A_{min}, \Omega_s = \omega_p/\omega_s)$$

2- Seguido, se determina la función que aproxime Los requerimientos pasa bajos según la aplicación (Butterworth, Chebyshev o Bessel).

3- Por último, la función $T_{LP}(S)$ se transforma a la función $T_{BP}(s)$ talque

$$T_{BP}(s) = T_{LP}(S)|_{S=(s^2+\omega_0^2)/Bs}$$

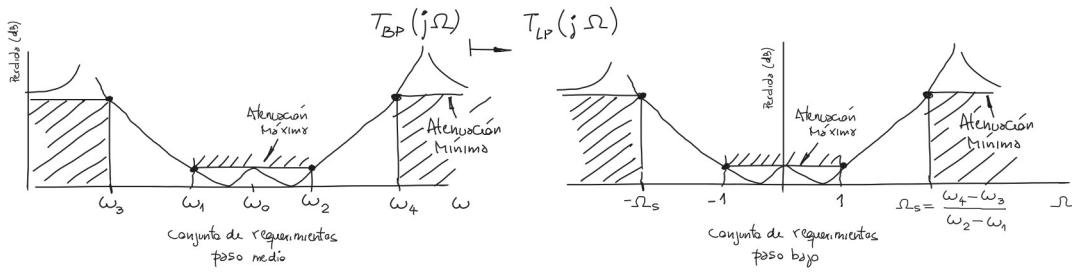


Figura 3.4: Funciones de atenuación. A la izquierda filtro pasa banda. A la derecha filtro pasa bajo normalizado

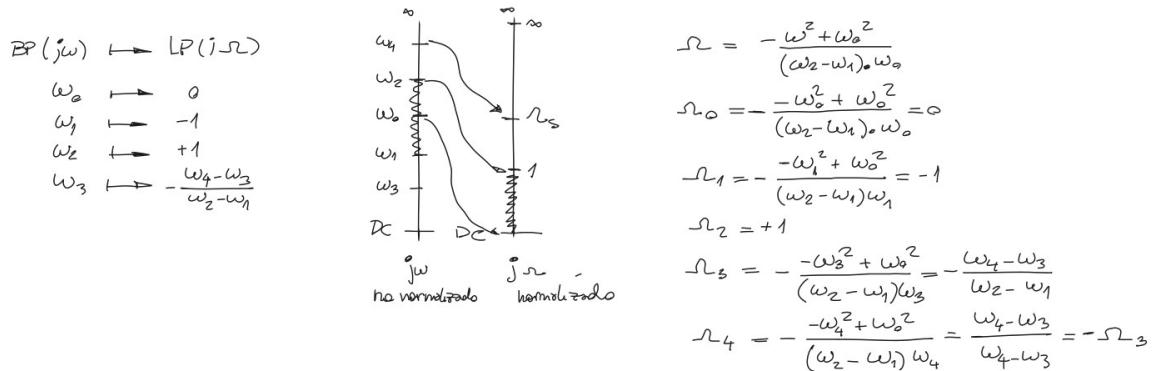


Figura 3.5: Relaciones entre las frecuencias de un filtro pasa banda y un filtro pasa bajo normalizado

3.3. Comentarios sobre la bibliografía

Daryanani. Principles of active network synthesis and design.
Desarrollo de las transformaciones de frecuencia.

4 Diseño de Filtros digitales

4.1. Introducción

Una forma de caracterizar un filtro digital en tiempo discreto es por medio de una ecuación de diferencias o por una función racional en la variable compleja z^{-1} :

$$H(z) = \frac{\alpha \prod_{i=1}^M (1 - z_i z^{-1})}{\prod_{k=1}^N (1 - p_k z^{-1})} = \frac{\prod_{i=0}^M a_i z^{-i}}{1 + \prod_{i=1}^N b_i z^{-i}} \quad (4.1)$$

donde $N > M$ y

α_{max} ,

z_i , son los ceros del numerador

p_k , son los ceros de polinomio del denominador

Similar a los filtros analógicos, las funciones realizables tendrán esta forma y el problema que se presenta es el mismo que se encuentra en los filtros analógicos, el cual es que en este tipo de funciones no presentan un valor constante en ninguna banda. Y por ende se busca una función que aproxime algunos aspectos de las características deseadas.

Teniendo en cuenta la expresión $H(z)$ se pueden presentar dos enfoques de diseños de filtros.

- Filtros de Respuesta al Impulso infinita, IIR. Cuando $b_k \neq 0$, la respuesta al impulso en tiempo discreto consiste de una serie infinita de términos.

- Filtros de respuesta al impulso finita, FIR: en este caso, la Ec. 4.1, con $b_k = 0$, tomará la forma:

$$H(z) = \alpha \prod_{i=1}^M (1 - z_i z^{-1})$$

Aquí, M es una cantidad finita y de ahí surge su nombre de respuesta al impulso finito, ya que la respuesta al impulso $h(n)$ consiste de una serie finita de valores. También se desprende que sus características principales son la estabilidad y la causalidad, es decir: $\sum_{n=-\infty}^{\infty} |h(n)| < \infty$ y que todas las ubicaciones de los polos del filtro digital están dentro del círculo unitario en el plano z.

4.2. Filtros IIR

Por la similitud que hay entre la función de transferencia realizable de un filtro digital y la función de transferencia realizable de un filtro analógico (funciones racionales, la primera en z^{-1} , y la segunda en s), los filtros IIR son versiones digitales de los diseños de filtros analógico.

Aprovechando esta similitud, una técnica de diseño consiste en mapear la función racional en s a otra función racional en z^{-1} .



Figura 4.1: Pasos para el diseño de filtros digitales

Por lo tanto el procedimiento, ilustrado en la Fig. 4.1, se puede resumir en que las especificaciones del filtro se suelen dar en la frecuencia analógica para el tipo de filtro deseado, luego se sigue el procedimiento de diseño para un filtro analógico y por último, se mapea la función de transferencia en del plano s al plano z^{-1} , esto es siempre que se cumplan dos condiciones siguientes.

1- Condición 1: Preservar las características en frecuencia de los filtros analógicos. Significa que el eje $j\omega$ se mapea en el círculo unitario en el plano z , Fig. 4.2.

$$\{s = j\omega | -\infty < \omega < \infty\} \mapsto \{z = e^{j\theta} | -\pi < \theta \leq \pi\}$$

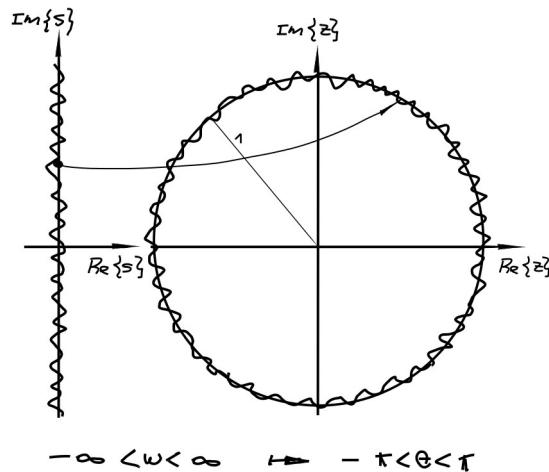


Figura 4.2: Mapeo del eje $j\omega$, en el plano s , en el círculo unitario, en el plano z

2- Condición 2: Preservar las propiedades de estabilidad. Para un sistema estable en s , no debería haber polos en el semiplano derecho, de forma equivalente para un sistema digital, los polos no deberían estar fuera del círculo unitario, Fig. 4.3

$$\{s / \operatorname{Re}[s] < 0\} \mapsto \{z / |z| < 1\}$$

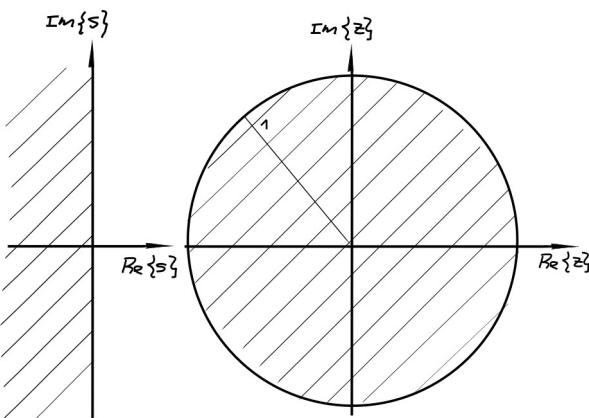


Figura 4.3: Mapeo del semiplano izquierdo del plano s dentro del circulo unitario en el plano z

4.2.1. Transformación por Método de Euler

Este método de mapeo es uno de los métodos por integración numérica, surge de aproximar la derivada en tiempo continuo mediante algunas diferencias finitas. El efecto resultante es reemplazar una ecuación diferencial (caracterización de un filtro analógico) con una ecuación de diferencias (caracterización de un filtro digital).

Función de mapeo

Se obtendrán distintos mapeos dependiendo del método de integración numérica utilizado, aquí se trata el método de aproximación de Euler, que consiste en aproximar la derivada en tiempo continuo con en diferencia de primer orden

$$\left. \frac{dy(t)}{dt} \right|_{t=nT} = \frac{y(n) - y(n-1)}{T}$$

dando como resultado de función de mapeo, $s = f(z)$, del plano s al plano z lo siguiente:

$$s = \frac{1 - z^{-1}}{T} := f(z)$$

donde T es el periodo de muestreo, y en el caso de la relación inversa sería

$$z = \frac{1}{1 - sT}$$

Desarrollo:

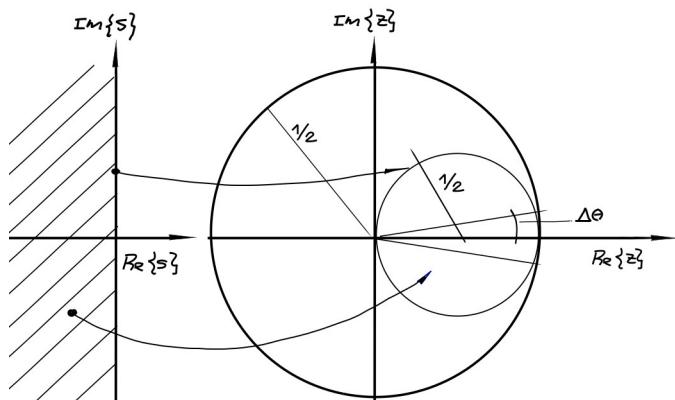
$$\begin{aligned} \left. \frac{dy(\hat{t})}{dt} \right|_{\hat{t}=nT} &= \frac{y(n) - y(n-1)}{T} \rightarrow sY(s) = \frac{Y(z^{-1}) - z^{-1}Y(z^{-1})}{T} \\ y(k) = \hat{y}|_{\hat{t}=nT} &\rightarrow sY(s) = \frac{1 - z^{-1}}{T} y(z^{-1}) \end{aligned}$$

Verificación de las condiciones de mapeo:

1- Condición 1: conservación de las características de frecuencia. Mapear el eje $j\omega$ en el plano z resulta la siguiente expresión

$$(z - 1/2) = 1/2 \cdot e^{j2 \cdot \text{atan}(\omega T)}$$

Como se ve en la Fig. 4.4 no se cumple la condición, porque no queda circunscrito en el circulo unitario. Pero también se observa que para un $\Delta\theta = \omega T$ pequeño el mapeo resulta en una buena aproximación de la condición 1.


 Figura 4.4: Mapeo del eje $j\omega$ en el plano z

2- La condición 2: para preservar la características de estabilidad implica que el semiplano izquierdo $\text{Real}[s] \leq 0$ quede mapeado dentro del circulo unitario, $|z| < 1$. Esto se verifica mediante la siguiente expresión:

$$|z| = \left| \frac{1}{1 - sT} \right|_{s=\sigma+j\omega} = \frac{1}{\sqrt{(1 + T\sigma)^2 + (T\omega)^2}} < 1$$

4.2.2. Transformación invariante al impulso

Este procedimiento asegura que la respuesta al impulso $h(n)$ del filtro digital resultante sea la versión muestreada de la respuesta al impulso $\hat{h}(t)$ de su correspondiente filtro analógico

$$h(n) := \hat{h}(t) \Big|_{t=nT}$$

donde T es el periodo de muestreo.

Función de mapeo

La función de mapeo queda establecida como

$$z = e^{sT} \rightarrow \theta = \omega T$$

donde θ es la frecuencia digital, y esta comprendida entre $|\theta| \leq \pi$.

La función de mapeo surge de comparar ambas respuestas en frecuencia

$$\sum \frac{\hat{\xi}}{s + \hat{p}_i} \rightarrow \underbrace{\sum \frac{\hat{\xi}}{1 - e^{\hat{p}_i T} z^{-1}}}_{p_i = e^{\hat{p}_i T}} = \sum \frac{\hat{\xi}}{1 - p_i z^{-1}}$$

resultando que el polo en el plano s, p_i , queda mapeado en el plano z como $p_i = e^{\hat{p}_i T}$.

Desarrollo:

Para un par transformado de $h(t)$ de la forma

$$\hat{h}(t) = \sum_{i=1}^N \hat{\xi}_i e^{\hat{p}_i t} \hat{u}(t) \xrightarrow{\mathcal{L}} \hat{H}(s) = \sum_{i=1}^N \frac{\hat{\xi}_i}{s + \hat{p}_i} \quad (4.2)$$

muestreando la $h(t)$ a intervalos regulares T se tiene que

$$\hat{h}(t) = |\hat{h}(t)|_{nT} = \sum_{i=1}^N \hat{\xi}_i e^{\hat{p}_i t} \hat{u}(t) \xrightarrow{\mathcal{Z}} \hat{H}(z) = \sum_{i=1}^N \frac{\hat{\xi}_i}{1 - p_i z^{-1}} \quad (4.3)$$

Comparando ambos resultados de 4.2 y de 4.3 se llega a

$$p_i = e^{\hat{p}_i T}$$

donde p_i es el polo i en el plano z correspondiente al polo equivalente \hat{p}_i en el plano s.

Verificación de las condiciones de mapeo:

La característica de frecuencia de la secuencia muestreada, $x(n)$, es una suma escalada de un número infinito de copias desplazadas en frecuencia de la característica de frecuencia de la correspondiente señal de tiempo continuo $x(t)$

$$\mathcal{X}(e^{j\omega}) = \frac{1}{T} \sum_{k=-\infty}^{\infty} \hat{X}\left[j\left(\frac{\omega}{T} + \frac{2\pi k}{T}\right)\right] = \frac{1}{T} \sum_{k=-\infty}^{\infty} \hat{X}\left[j\left(\omega + \frac{2\pi k}{T}\right)\right]$$

$\omega T = \Theta$ $\omega_0 = \frac{2\pi}{T}$

Gráficamente se puede ver en la Fig. 4.5 que el mapeo del plano s en el plano z resulta en tiras del plano s dentro del círculo unitario, quedando estas en una misma región en z. El mapeo no es lineal, es decir hay mas de un punto que se mapea en un punto.

Por lo tanto solo es aplicable a señales de banda limitada que cumplen la condición

$$|\hat{H}(j\omega)| \simeq 0 \text{ para } |\omega| > \omega_B$$

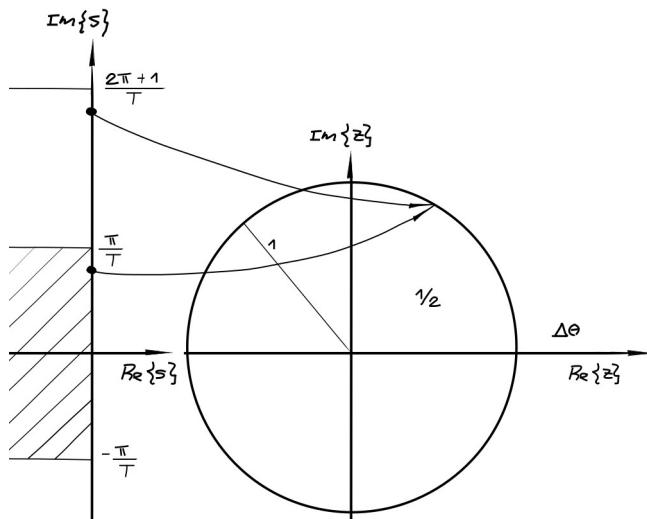


Figura 4.5: Mapeo resultante en el plano z

Si no se cumple esa condición ocurre un solapamiento (o aliasing) de las regiones y por ende de las señales en tiempo.

Desarrollo:

Primeramente, para ver la relación entre la respuesta en la frecuencia digital, $X(e^{j\theta})$ y la respuesta en frecuencia analógica, $\hat{X}(j\omega)$ se plantea la transformada de Fourier de la señal muestreada $\hat{x}(nT)$. Así que primero se presenta la señal muestreada como un tren de impulsos x espaciados y escalado por $x(nT)$ y luego a esta se le aplica la transformada de Fourier.

$$\begin{aligned}
 x(n) &= \hat{x}(nT) = \hat{x}(t) * \sum_n \delta(t - nT) \xrightarrow{\text{SF}} \hat{x}(j\omega) * \frac{1}{T} \sum_n \delta(\omega - \omega_0 n) \\
 &\quad \xrightarrow{\text{TF}} \hat{x}(t) * \frac{1}{T} \sum_n e^{j\omega_0 n t} \quad \left. \begin{array}{l} \text{señal de tiempo} \\ \text{Transformada de Fourier} \end{array} \right\} \omega_0 = 1/T \\
 &\quad \xrightarrow{\text{TF}} \hat{x}(j\omega) * \frac{1}{T} \sum_n \delta(\omega - \omega_0 n) \\
 &\quad \xrightarrow{\text{transformada inversa de Fourier}} \frac{1}{T} \sum_n \hat{x}(j\omega) * \delta(\omega - \omega_0 n) = \boxed{\frac{1}{T} \sum_{n=-\infty}^{+\infty} \hat{x}(j\omega - \omega_0 n)} \\
 &\quad \xrightarrow{\text{transformada inversa de Fourier}} \hat{x}(nT) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \frac{1}{T} \sum_{n=-\infty}^{+\infty} \hat{x}(j(\omega - \omega_0 n)) d\omega = \boxed{\frac{1}{T} \hat{x}(j(\omega - \omega_0 n)) = x(e^{j\theta})} \\
 &\quad x(n) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} x(e^{j\theta}) e^{-j\theta n} d\theta
 \end{aligned}$$

y así, se llega a que la respuesta en frecuencia digital es la sumatoria de las respuesta analógica escalada en $1/T$ y desplazada en múltiplos enteros de ω_0

$$\frac{1}{T} \hat{X}(j(\omega - \omega_0 n)) = X(e^{j\theta}) \text{ para } |\theta| \leq \pi$$

Segundo, si se parte de la respuesta analógica se debería llegar a la misma relación. Esto es,

$$x(t)|_{t=nT} = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \hat{X}(j\omega) e^{j\omega t} dt \Big|_{t=nT}$$

y luego desarrollando esta expresión como

$$\begin{aligned}
 x(\omega) &= \sum_{k=-\infty}^{\infty} \frac{1}{2\pi} \left(\int_{-\frac{2\pi k}{T}}^{\frac{2k+1}{T}\pi} \hat{X}(j\omega) e^{j\omega_n T} d\omega \right) \Big|_{\omega=\omega_0 + \frac{2\pi k}{T}} = \sum_{k=-\infty}^{\infty} \frac{1}{2\pi} \int_{-\frac{\pi}{T}}^{\frac{\pi}{T}} \hat{X}(j(\omega_0 + \frac{2\pi k}{T})) e^{j\omega' n T} d\omega' \\
 \omega &= \omega_0 + \frac{2\pi k}{T} \\
 \omega_2 &= \frac{2\pi k + \pi}{T} = \omega_0 + \frac{2\pi k}{T} \\
 \frac{2\pi k + \pi}{T} &= \omega_2' + \frac{2\pi k}{T} \rightarrow \omega_2' = \frac{\pi}{T} \\
 \omega_1 &= -\frac{2\pi k - \pi}{T} = \omega_1' - \frac{2\pi k}{T} \rightarrow \omega_1' = -\frac{\pi}{T}
 \end{aligned}$$

mediante un cambio de variable queda que

$$\begin{aligned}
 x(n) &= \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \int_{k=\frac{2k-1}{T}\pi}^{\frac{2k+1}{T}\pi} \hat{X}(j(\omega + \frac{2\pi k}{T})) \Big|_{\omega T = \theta} \\
 &\quad \omega_1 T = \theta_2 \\
 &\quad -\omega_1 T = \theta_1 \\
 &= \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \int_{-\pi}^{\pi} \hat{X}(j(\frac{\theta}{T} + \frac{2\pi k}{T})) e^{jn\theta} \frac{d\theta}{T}
 \end{aligned}$$

y finalmente se pueden relacionar ambas características en frecuencia como:

$$\begin{aligned}
 X(\omega) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \underbrace{\frac{1}{T} \sum_{k=-\infty}^{+\infty} \hat{X}(j(\frac{\theta}{T} + \frac{2\pi k}{T}))}_{\chi(e^{j\theta})} e^{j\omega\theta} d\theta \\
 X(\omega) &= \overbrace{\frac{1}{2\pi} \int_{-\pi}^{\pi} \chi(e^{j\theta}) e^{j\omega\theta} d\theta}^{\chi(e^{j\omega})}
 \end{aligned}$$

4.2.3. Transformación por mapeo bilineal

Función de mapeo:

Se define a partir de la siguiente función de mapeo

$$s = f(z) = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}$$

y la relación inversa

$$z = \frac{2+sT}{2-sT} \text{ o de forma equivalente } z^{-1} = \frac{2-sT}{2+sT}$$

$$\text{y la relación entre la frecuencia digital y analógica } \theta(\omega) = \operatorname{atan}\left(\frac{4\omega T}{4-(\omega T)^2}\right)$$

Verificación de las condiciones de mapeo:

De la condición 1, el eje $j\omega$ queda mapeado en el círculo unitario, así que cumple dicha condición de conservar las características en frecuencia. Y de la condición 2, la región del semiplano izquierdo del plano s queda dentro del círculo unitario en el plano z, Fig. 4.8

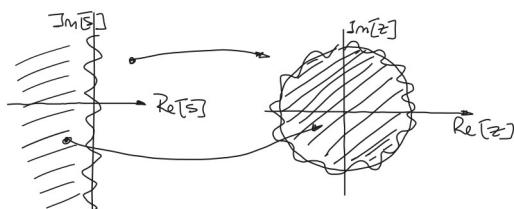


Figura 4.6: Mapeo resultante en el plano z

Desarrollo:

1- La condición de que se conservan las características de frecuencia sale de reemplazar $s = j\omega$ en

$$z = \frac{2 + j\omega T}{2 - j\omega T} = e^{j\operatorname{atan}\left(\frac{4\omega T}{4 - (\omega T)^2}\right)} = e^{j\theta(\omega)}$$

quedando finalmente

$$\theta(\omega) = \operatorname{atan}\left(\frac{4\omega T}{4 - (\omega T)^2}\right)$$

2- De igual manera, pero ahora con $\sigma \neq 0$ se tiene que $s = \sigma + j\omega$ y por lo tanto

$$z = \frac{2 + j\omega T}{2 - j\omega T} \Big|_{s=\sigma+j\omega} = \frac{2 + \sigma T + j\omega T}{2 - \sigma T - j\omega T}$$

si $\operatorname{Real}(s) = \sigma < 0$ entonces $|z| < 1$, de esta manera se concluye que también cumple la condición 2. Quedando representado el mapeo en z como se muestra en la Fig. 4.8.

Función de transferencia y Respuesta en frecuencia

Como se vio en el desarrollo de las condiciones para preservar las características de frecuencia y estabilidad este mapeo bilineal cumple con ambas propiedades, quedando relacionadas de la siguiente manera ambas respuestas en frecuencia.

$$H(z) = \hat{H}(s) \Big|_{s=(2/T)(1-z^{-1})/(1+z^{-1})}$$

donde T es el período de muestreo.

Pero la relación entre las frecuencia digital θ y la frecuencia analógica no es uniforme ω no es uniforme, esto es

$$s = \sigma + j\omega = \frac{2}{T} \frac{1 - e^{-j\theta}}{1 + e^{-j\theta}} = j \frac{2}{T} \tan \frac{\theta}{2} \begin{cases} \omega = \frac{2}{T} \tan \frac{\theta}{2} \\ \sigma = 0 \end{cases}$$

Predistorsión – prewarping.

Consiste en encontrar una constante o una frecuencia de muestreo equivalente $\omega_s = \frac{2\pi}{T}$ tal que dada una frecuencia

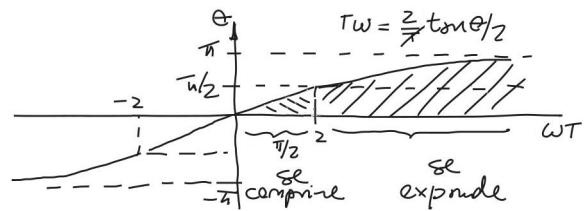


Figura 4.7: Mapeo resultante en el plano z

analógica coincida con el mapeo inverso de una frecuencia digital relacionada linealmente con esta frecuencia analógica dada.

$$\omega_x T' / 2 = \tan \frac{\theta}{2} \Big|_{\theta = \omega_x T / 2}$$

Ya que para la frecuencia de muestreo utilizada, $\omega T < \pi/2$, las frecuencias analógica se comprimen, y $\omega T > \pi/2$ las frecuencias expanden respecto a la frecuencia analógica.

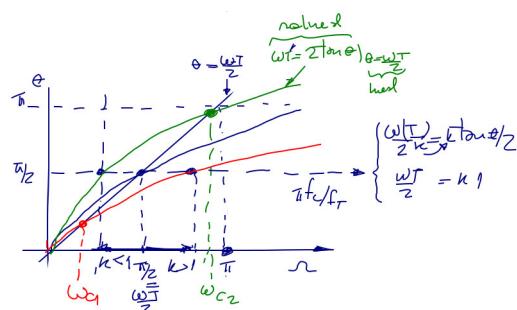


Figura 4.8: Mapeo resultante en el plano z

Entonces se busca un punto donde coincida la relación lineal con la relación binomial. A continuación se busca una frecuencia equivalente de muestreo tal que se relacionen esta dos relaciones lineal y no lineal (intersección de los dos puntos.)

$$\omega_x T' = 2 \tan \frac{\omega_x T}{2}$$

$$\omega_x T' = 2 \tan \frac{\omega_x T}{2} = \frac{\omega_x^2}{\omega_s^2} \omega_s T$$

$\boxed{\omega_x' = \frac{\omega_x T}{2}}$

la frecuencia de muestreo equivalente
para que se cumpla que

$$\omega_x' = \theta_x$$

4.3. Filtros FIR

Definición

La función de transferencia de un filtro FIR tiene la forma:

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

$h(n)$ es una secuencia que representa la respuesta al impulso en tiempo discreto, de longitud finita y causal.

Hay varias formas de diseñar un filtro FIR, una es por truncamiento de la respuesta al impulso, y la otra forma puede ser por muestreo de frecuencia.

Aquí se desarrolla solo el primer método. En este método se especifica la respuesta en frecuencia, $H(j\omega)$, y luego se plantea la serie de Fourier de ésta (en $j\omega$), la secuencia resultante representa la respuesta al impulso en tiempo discreto, $h_d(n)$. Como esta secuencia tiene duración infinita y es no causal, se multiplica por una ventana tal que trunque la secuencia fuera de los límites de dicha ventana, y finalmente se desplaza de tal modo que quede causal.

En base a lo dicho al párrafo anterior, $h(n)$ va ser una secuencia de tiempo de discreto causal y de longitud N finita, y esto hace que se den cuatro tipos o casos de simetrías.

- Tipo I: Simetría respecto a π con una cantidad N impar de muestras.
- Tipo II: Simetría respecto a π con una cantidad N par de muestras.
- Tipo III: Simetría respecto a 0 y a 2π con una cantidad N impar de muestras.
- Tipo IV: Simetría respecto a 0 y a 2π con una cantidad N par de muestras.

Para el tipo II y IV solo es aplicable a filtros pasa bajos y pasa medios, y los otros tipos no tienen restricciones. Para que el filtro fuese de fase lineal $h(n)$ debe cumplir la condición:

$$h(n) = h(N - 1 - n)$$

y esto asegura también que sea causal.

Otra característica de la aproximación por truncamiento de la respuesta del impulso ideal es que se da el fenómeno de convergencia no uniforme de Gibbs en la discontinuidades. En este sentido se utilizan diferentes ventanas para atenuar dichos sobre-impulsos.

Desarrollo

Para comenzar con la aproximación por truncamiento de la respuesta del impulso se especifica el filtro ideal en el dominio de la frecuencia,

$$H(j\omega T = j\Omega) = \begin{cases} 1 & |\omega T| < \omega_c T = \Omega_c \\ 0 & \text{para otro valor} \end{cases}$$

donde $\Omega = \omega T$ es la frecuencia normalizada de ω con respecto a la frecuencia de muestreo $1/T$, entonces Ω va variar entre $-\pi$ y π ; Ω_c es la frecuencia de corte normalizada, y ω_c es la frecuencia de corte no normalizada.

Se supone que $H(j\Omega)$ es periódica de período 2π , quedando el desarrollo en serie de Fourier de la respuesta en frecuencia $H(j\Omega)$, representada en la Fig. 4.9, tal como

$$H(j\omega T = j\Omega) = \sum_{n=-\infty}^{n=\infty} b_n e^{-j \cdot 1 \cdot n}$$

donde la frecuencia fundamental es 1, entonces $\omega_0 = \frac{2\pi}{2\pi} = 1$. En este caso, b_n se asocia directamente con la secuencia

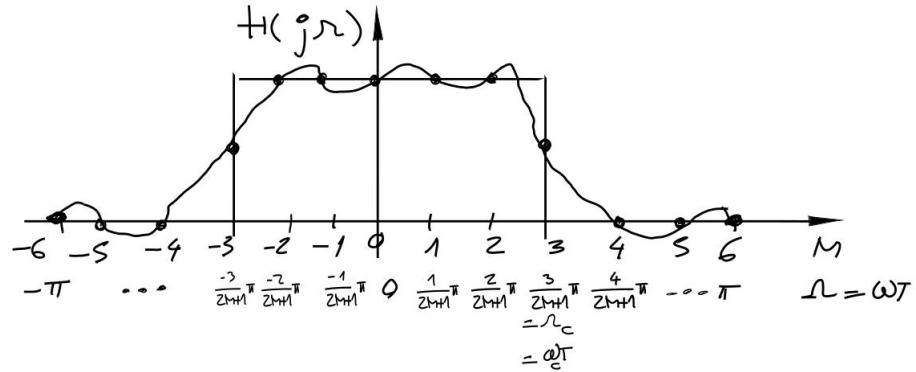


Figura 4.9: Representación gráfica de $H(j\omega)$ y de su desarrollo serie de Fourier en $j\omega$

en tiempo discreto $h_d(n)$, Fig. 4.9, y que corresponde a los coeficientes de $H(j\omega)$.

$$h_d(n) = b_k = \frac{1}{2\pi} \int_{-\Omega_c}^{\Omega_c} 1 \cdot e^{-j1 \cdot n\Omega} = \frac{\sin(\Omega_c n)}{\pi n}$$

Esta secuencia, $h(n)$ tiene una cantidad infinita de términos y además no es causal. Una forma de resolver la causalidad

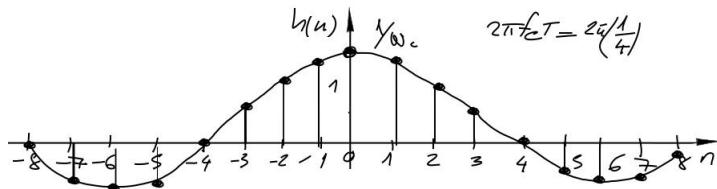


Figura 4.10: Representación gráfica de la respuesta al impulso $h_d(n)$.

es multiplicar esta secuencia por una ventana que trunque la secuencia a una cantidad de términos diferentes de cero y luego desplazar esta secuencia de tal forma que quede causal. En primera instancia la ventana $w(n)$ puede ser una ventana rectangular de longitud N , tal como se muestra trazada en rojo en la Fig. 4.11.

$$h(n) = h_d(n) \cdot w(n) \quad (4.4)$$

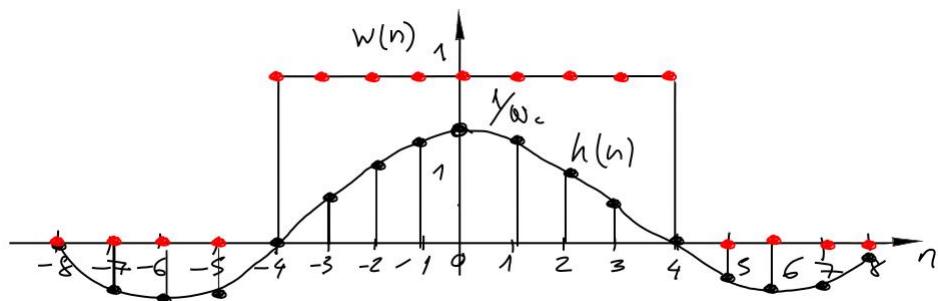


Figura 4.11: Representación gráfica de $h_d(n)$ y $w(n)$ (trazado en rojo)

Finalmente, a partir de la Ec. 4.4, $h(n)$ será una secuencia de longitud N finita y causal, lo que para un N par la expresión quedaría definida como:

$$h(n) = \begin{cases} \frac{\sin(\Omega_c \pi (n - \frac{N-1}{2}))}{\pi(n - \frac{N-1}{2})} & \text{para } n! = \frac{N-1}{2} \\ \Omega_c & \text{para } n = \frac{N-1}{2} \end{cases} \quad (4.5)$$

En la Fig. 4.12 representa el caso de longitud impar dado en la Ec. 4.5.

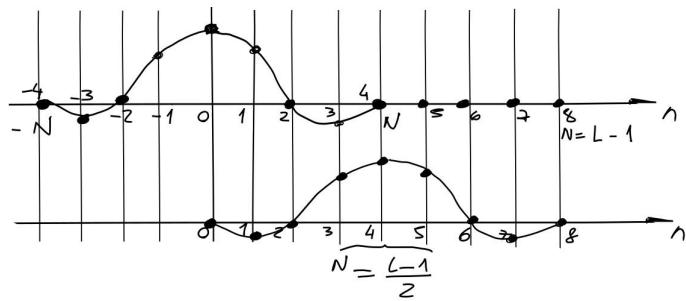


Figura 4.12: Representación de $h_d(n)$ y de $h(n)$ para N impar.

Y para una secuencia $h(n)$ de N par se tiene:

$$h(n) = \frac{\sin(\Omega_c \pi (n - \frac{N-1}{2}))}{\pi (n - \frac{N-1}{2})} \quad (4.6)$$

En la Fig. 4.13 representa el caso de longitud par dado en la Ec. 4.6 .

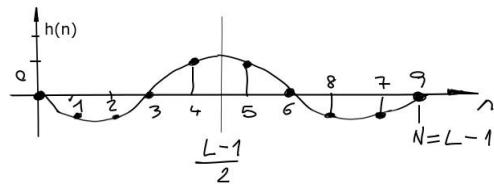


Figura 4.13: Representación de $h_d(n)$ y de $h(n)$ para N par.

Ejemplo con Matlab 1: Filtro pasa bajos FIR, Fig. 4.14.

```

1 clear all; clc; close all;
2
3 N=15; % longitud
4 Wc=0.3; % frecuencia de corte normalizada wct
5 esLP=1; % es filtro pasa bajos
6
7 hd0=Wc;
8 if esLP==0; Wc=2-0.3; hd0=1-hd0; end;
9
10 syms n;
11 m=(N-1)/2 ;
12 h= @(n) sin(Wc*pi*( n - m ) )./(pi*( n - m));
13
14
15 if(mod(N,2)==1)
16     hd= [h(0:(N-3)/2) hd0 h((N+1)/2:(N-1)) ];
17 else
18     hd=[h(0:N-1) ];
19 end
20
21 fig=figure;
22 freqz(hd,1); title('Filtro FIR');
23 saveas(fig,'./fir1_ej1','jpg');
```

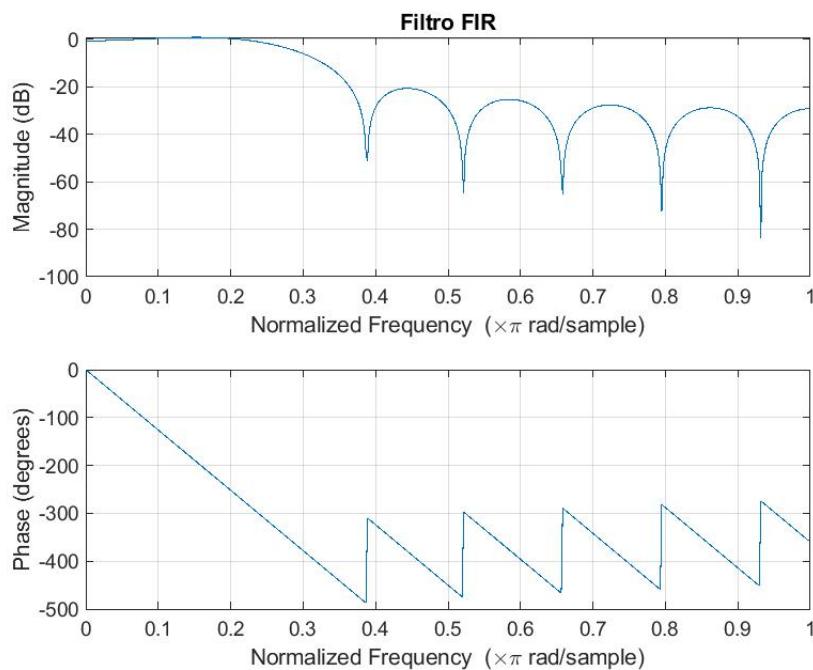


Figura 4.14: Respuesta en frecuencia.

4.4. Comentarios sobre la bibliografía

Lam Harry. Analog and digital filters design and realization.
Desarrollo general.

Oppenheim y Schafer. Digital Signal Processing. 1975.
Desarrollo de los filtros FIR.

Schlichthärle, Dietrich. Digital Filters: Basics and Design. Second Edition
Las expresiones de las Ventanas para los filtros FIR

5 Realización de filtros Digitales

5.1. Realización

Los filtros digitales que son realizables mediante elementos computacionales estarán dados por una función racional en z^{-1}

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

o de forma equivalente

$$Y(z) = \sum_{k=1}^N a_k Y(z) cdot z^{-k} + \sum_{k=0}^M b_k X(z) \cdot z^{-k}$$

y en la caracterización correspondiente en tiempo discreto estará dada por una ecuación de diferencias.

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

La realización del filtro digital se expresa mediante un algoritmo, que puede a su vez puede ser construido en hardware dedicado o hardware de propósito general mediante software, mediante elementos computacionales sencillos, tales como, sumas (Fig. 5.1a), productos de constantes (Fig. 5.1b), y retardos temporales (Fig. 5.1c).

Estos algoritmos se dan en diferentes estructuras o tipologías que representan en diagramas de bloques, diagramas de flujo o notación matricial. Entre estas estructuras se encuentran:

1- Formas directas: la $H(z)$ se realiza en una sola pieza. Se clasifican en Formas directas I y II, estructura de escalera, estructura de celosías, extracción de multiplicadores, y formas modulares de filtros de ondas.

2- Formas indirectas: la $H(z)$ se descompone en varias secciones de primer y segundo orden. Mediante las formas directas se consiguen estructuras de primer y segundo orden, y luego se combinan de algún modo.

Forma directa I y II

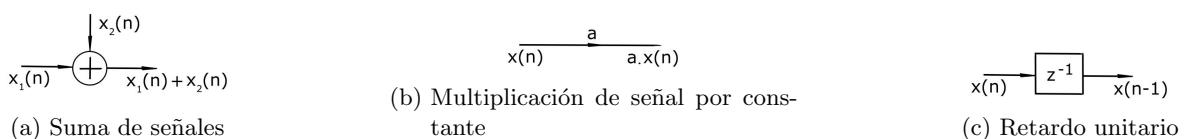


Figura 5.1: Representación en bloques de elementos computacionales básicos



Figura 5.2: Forma directa I y II

5.2. Comentarios sobre la bibliografía

Rabiner, L. R., and Gold, B. Theory and Application of Digital Signal Processing. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1975.

6 Procesamiento digital-construcción de filtros digitales

6.1. Realización

En general, un sistema de procesamiento de señales digitales se puede representar como se muestra en el diagrama, Fig. 6.1. El bloque de conversión a digital, ADC, muestrea y cuantifica la señal digital $\hat{x}(t)$ en una secuencia de tiempo discreto $x(n)$, seguido se trata la señal digital mediante algún algoritmo de procesamiento digital, DSP, y finalmente se reconstruye la secuencia discreta resultante $y(n)$ en una señal de analógica de tiempo continuo $\hat{y}(t)$. Adicionalmente se incluyen filtros limitadores de frecuencia en cada uno de los extremos, uno en la entrada con el fin de rechazar el ingreso de frecuencias indeseadas, que van a estar por encima de la mitad de la frecuencia muestreo del sistema, y otro filtro limitador en la salida para descartar las componentes de frecuencia superiores y así suavizar la salida.

El algoritmo computacional de un filtro digital puede ser construido mediante un hardware de propósito general o un hardware de dedicado. En un hardware dedicado, los elementos básicos computacionales, como las sumas, multiplicación por constantes y retardos se implementan mediante compuertas y unidades de memorias, ya sea con componentes discretos o integrados en un microcircuito. Otra forma de representar estos algoritmos es por medio de software, ejecutado en algún hardware de propósito general, como puede ser un microcontrolador o una computadora personal.

Construcción mediante microcontrolador

La implementación del algoritmo de un filtro digital puede haber ligeras variaciones entre una y otra, dependiendo del lenguaje de programación, la técnica de bufferización que ofrezca el hardware, etc. Por ejemplo, los microcontroladores avanzados traen instrucciones especiales de procesamientos de señales, conocidos como DSP. Estas instrucciones pueden ser como, por ejemplo, la convolución de señales.

Una representación general del sistema en bloques se muestra en la Fig. 6.2. Los bloques ADC, y DAC es hardware específico, que en la mayoría de los casos se incluye junto el microcontrolador en el mismo microcircuito. El bloque de procesamiento es implementado mediante rutinas de software.

En la Fig. 6.2, se muestra de forma simplificada las rutinas asociadas al bloque DSP de un filtro FIR, la lista de coeficientes y superpuesto al diagrama se asocia otro diagrama de bloques que presenta el algoritmo completo del filtro.

Construcción mediante hardware específico

En la construir un filtro con componentes discretos o un microcircuito se pueden presentar dificultades prácticas, esto es porque para el primero se necesita cablear una enorme cantidad de componentes y para el segundo, las dificultades pueden ser simplemente por la dificultad de acceso a la tecnología necesaria. Una alternativa viable es la utilización de una FPGA (o tecnología similar).

Las FPGAs y similares consisten en un microcircuito con una enorme cantidad de celdas lógicas configurables. Así se pueden construir mediante algún lenguaje de especificación de hardware directamente o indirectamente (por medio de un esquemático) circuitos digitales de diversas complejidades.



Figura 6.1: Diagrama de bloques, procesamiento digital de señales

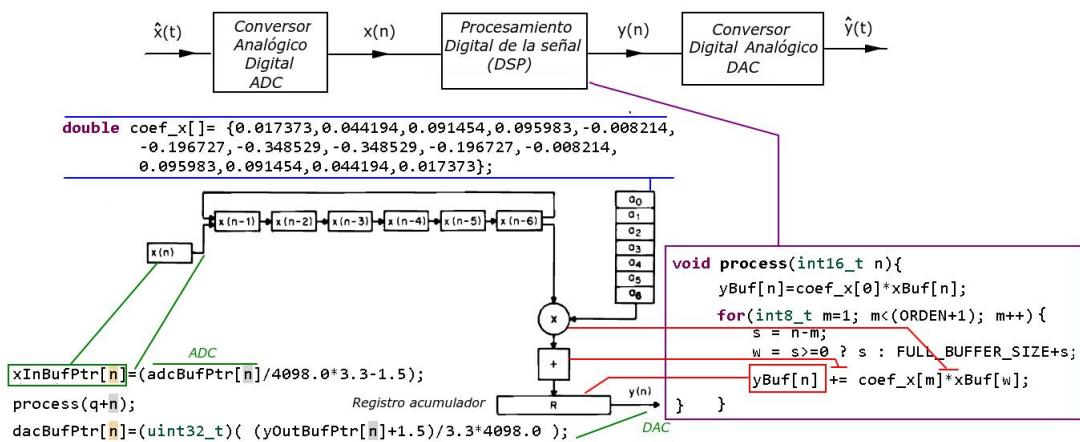


Figura 6.2: Representación del algoritmo de un filtro FIR, rutinas en c correspondientes al proceso, asociado al diagrama general de procesamiento digital, 6.1

Dependiendo el entorno de desarrollo que se utilice se puede ir definiendo los bloques fundamentales del filtro y luego ir interconectandolos hasta lograr el filtro de las características deseadas.

Bloque de retardo y multiplicación

Bloque de suma

Bloque de adc

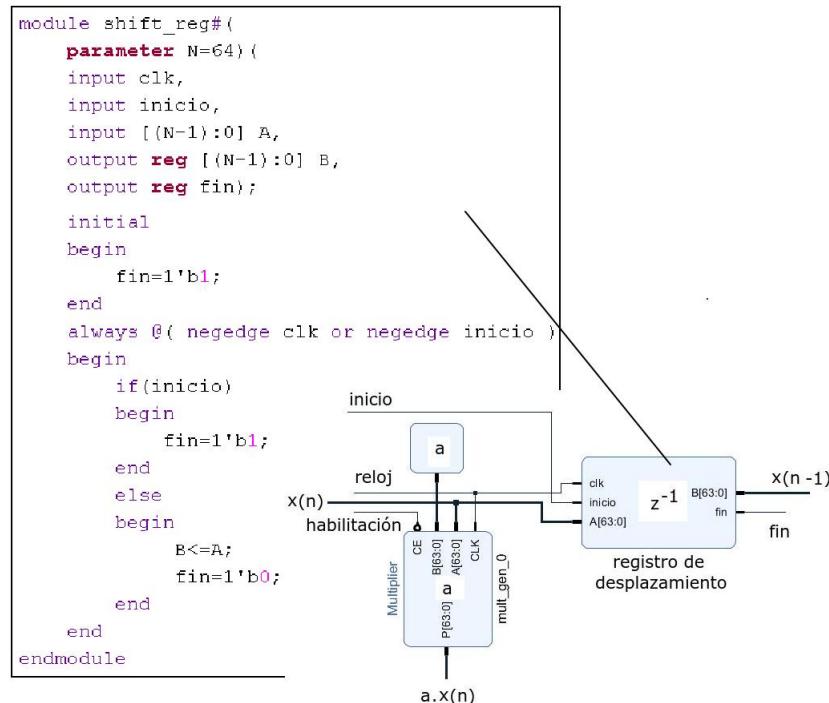


Figura 6.3: Implementación con FPGA del bloque de retraso

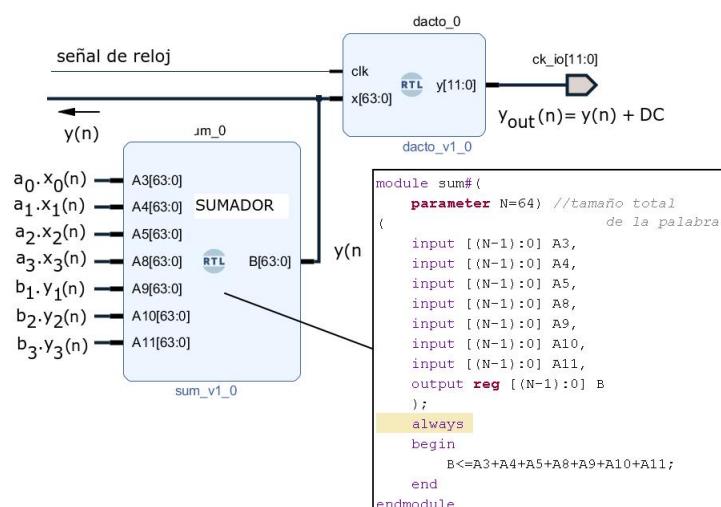


Figura 6.4: Implementación del bloque sumador y representación del bloque DAC

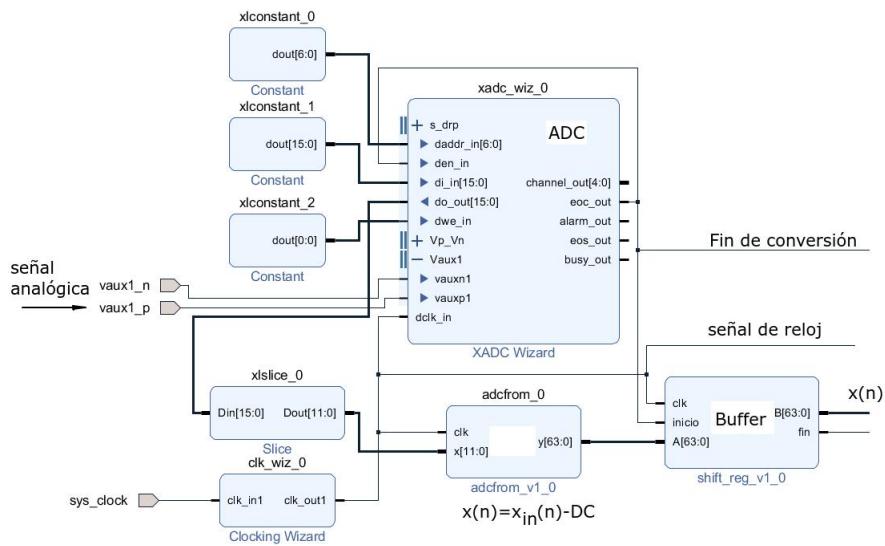


Figura 6.5: Representación del bloque ADC

6.2. Comentarios sobre la bibliografía

Rabiner, L. R., and Gold, B. Theory and Application of Digital Signal Processing. Englewood Cliffs, N.J.: Prentice-Hall, Inc., 1975.

7 Especificaciones de las implementaciones de los filtros realizados

7.1. Especificaciones generales del hardware utilizado

Implementación realizada utilizando Microcontrolador

El montaje completo consiste de una placa embebida de bajo costo, una interfaz para la grabación y depuración del código en el microcontrolador, un circuito para proteger las entrada analógica, y un circuito de salida que actúa como carga, Fig. 7.1.

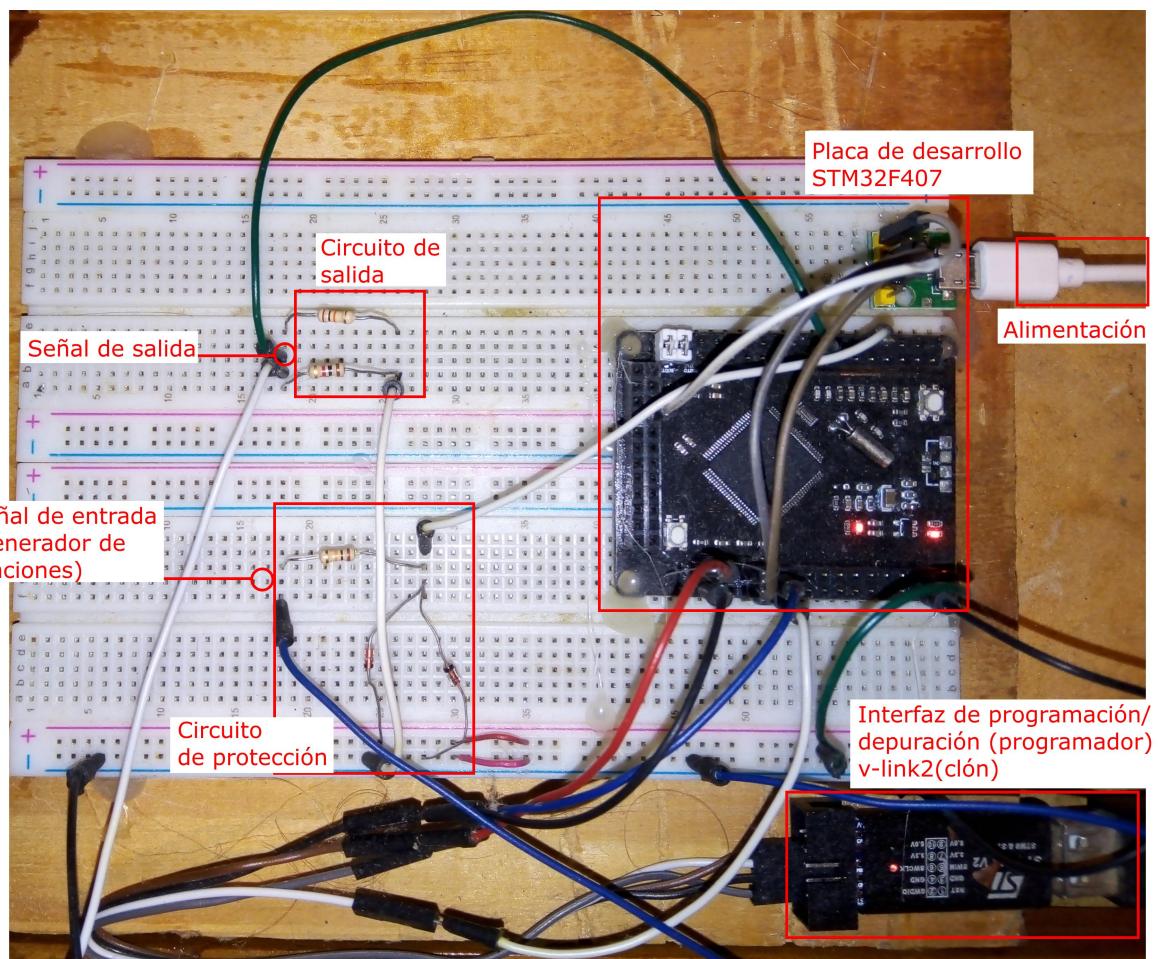


Figura 7.1: Montaje del sistema embebido con microcontrolador

Las características generales del microcontrolador de interés que posee la placa embebida para esta aplicación son: - La denominación comercial del microcontrolador es STM32F407VGT, - La CPU es de arquitectura Arm, Corte-M4-32 bits, con una señal de reloj de sistema máxima de 168 MHz. - El rango de voltaje de alimentación es de 1.8 de 3.3V (valor típico). - El módulo de conversión analógica-digital, ADC, incluido: tiene 3 unidades de 12-bit, con un voltaje máximo de entrada menor o igual al voltaje de alimentación, y la tasa de muestreo ronda entre 2 a 6 MSPS, dependiente de la alimentación y su configuración. La impedancia de entrada es de 50 kΩ, el valor de resistencia de muestreo es de 6 kΩ y el capacitor de muestreo es aproximadamente de 4 pF. - El módulo de conversión digital-analógica, DAC, incluido: tiene dos unidades, tiene una resistencia de carga de 5KΩ si tiene el buffer de salida activado, impedancia de

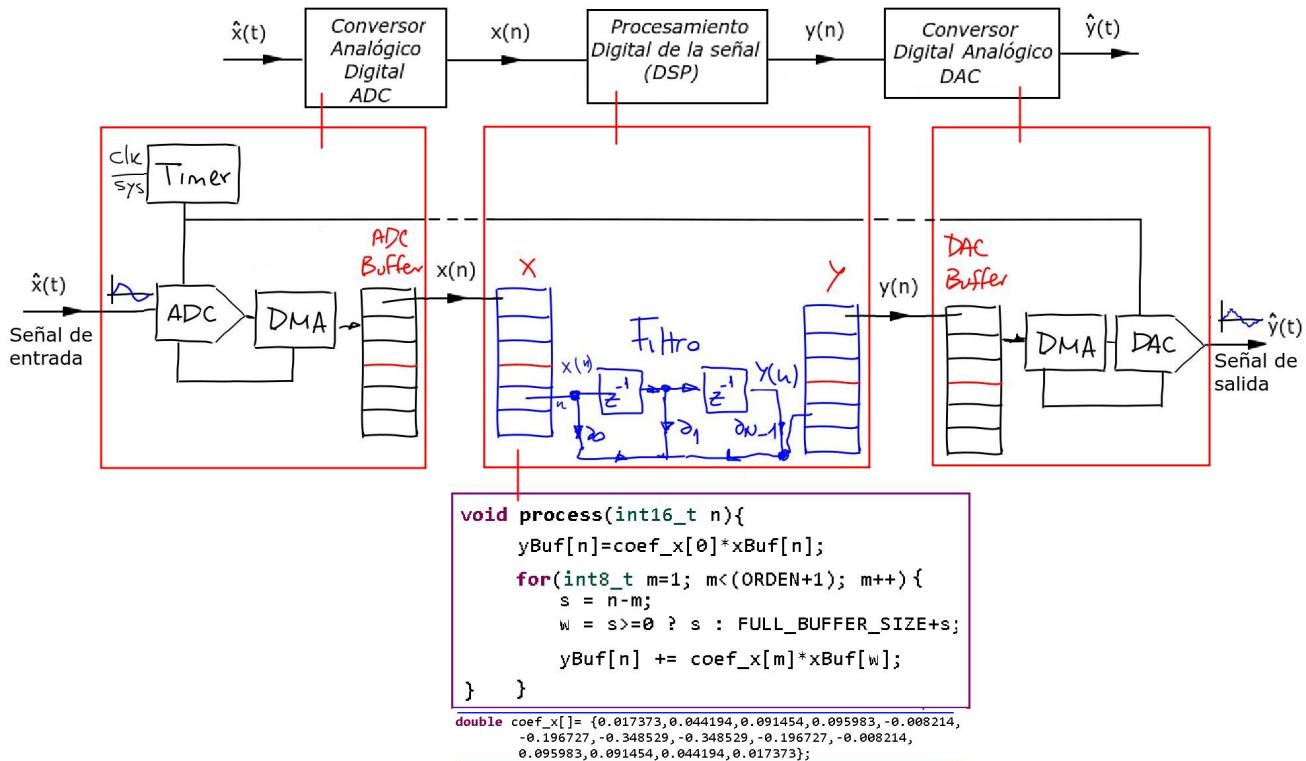


Figura 7.2: Representación en bloques de la arquitectura desarrollada para los filtros digitales

salida $15 \text{ k}\Omega$ y una capacitancia parásita de 50 pF . - El módulo de temporización incluido, Timer: tiene 17 unidades, hay de 16 bits y de 32 bits, con frecuencia de operación por máxima de 168 MHz.

La arquitectura de la implementación del sistema embebido para el desarrollo de los filtros digitales se puede resumir en la Fig. 7.2. El Timer es simplemente un contador que toma la señal de reloj del bus interno del microcontrolador al que está conectado, y a partir del valor de la cuenta cargado en el registro de comparación genera señales de menor frecuencia. El principio de funcionamiento consiste en iniciar un registro de comparación y cuando el contador desborda o llega al valor de comparación precargado genera una señal de activación y luego se reinicia para generar otro ciclo de cuenta. En este caso, se genera una señal para disparar los inicio de conversión del módulo ADC y del módulo DAC. La frecuencia de esta señal de control se considera la frecuencia de muestreo.

El cálculo del valor que configurará el registro de comparación del Timer va a determinar la frecuencia de muestreo utilizada. Éste valor de registro va a depender de la frecuencia de la señal de reloj del bus que alimenta al Timer en cuestión (en este caso $\text{freqbus} = 84 \text{ MHz}$), a esta frecuencia se la divide por la cantidad de cuentas que va a realizar menos uno ($\text{reg_contador} = 1252 \text{ cuentas}$) y también, se divide por el valor del registro divisor de frecuencia mas uno (el valor del divisor es $\text{prescaler} = 2$). Con todo esto la frecuencia de muestreo queda determinada por:

$$F_m = \frac{\text{freqbus}}{(\text{reg_contador} - 1) * (1 + \text{prescaler})} = 168M / (1252 - 1) * (2 + 1) = 22418 \text{ kHz}$$

Ambos módulos para las conversiones de analógico a digital y viceversa son configurados para que trabajen junto con el bloque de Acceso Directo a Memoria, DMA. Este último bloque gestiona la transferencia de datos entre los módulos de conversión y los buffers de memoria de entrada y salidas correspondientes. El uso del módulo DMA libera a la CPU de las rutinas de copiado de los valores de los registros de entradas y salidas hacia, o desde, los buffers, aumentando las posibilidades de procesamiento del CPU.

Tanto el buffer de entrada, como el de salida, se dividen cada uno en dos mitades. El objetivo de esto es que mientras, por ejemplo, el módulo ADC llena una de las mitades del buffer, el bloque de procesamiento de señal (DSP) consuma los datos de la otra mitad del buffer, producidos por el ADC en instantes anteriores, evitando interferencias entre el productor de los datos (ADC) y el consumidor de los datos (DSP).

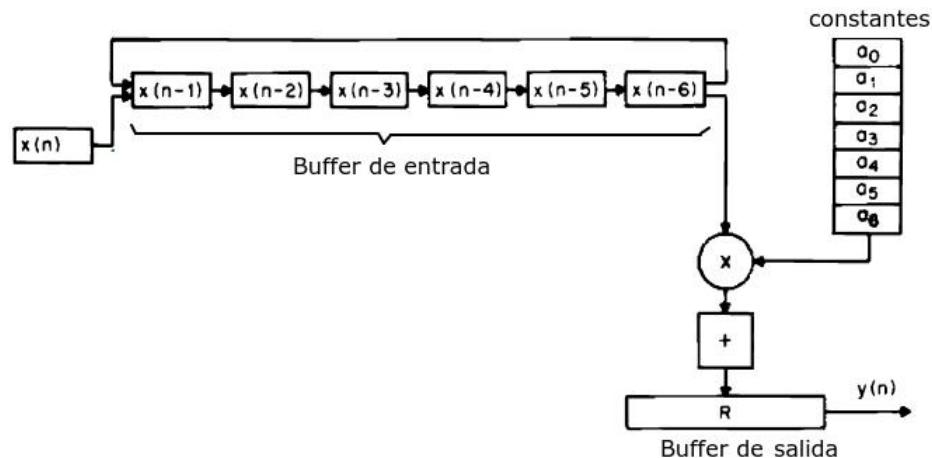


Figura 7.3: Representación del algoritmo de un filtro FIR

El bloque de procesamiento de señales digitales, DSP, está implementado totalmente por software, básicamente hace las operaciones computables de sumar, multiplicar por constantes, y retardos mencionadas en la sección anterior. Estas combinadas conforman el algoritmo del filtro desarrollado.

Respecto a la operación de retardo hay dos formas de implementarla:

1- Teniendo registros para cada una de las muestras de la secuencia a almacenar. Todos estos juntos conforman el buffer de entrada y salida del bloque de proceso de señales. Para poder crear el efecto de desplazamiento temporal se sobre-escribe el registro que representa el valor de la señal en un tiempo particular con el valor del registro en el instante siguiente, finalizando con el primer registro al que se copia la lectura actual del ADC.

$$\begin{aligned}x(2) &< -x(1) \\x(1) &< -x(0) \\x(0) &< -ADC \quad \text{valor actual}\end{aligned}$$

Esta técnica se utiliza frecuentemente en los desarrollos con placas Arduino. De realizar esto con componentes discretos se utilizarían registros de desplazamientos.

2- Del mismo modo que el primero, va a haber una serie de registros concatenados formando dos buffers, pero conectados en anillo (el primero interconectado con el último). En esta técnica no se van sobre-escribiendo los valores de los registros con los valores del registro anterior, sino que se trabaja con un índice que incrementa el puntero inicial y final de la cola de secuencias a procesar. Con esto se consigue mejor eficiencia en el proceso, ya que no se tienen que actualizar una cantidad determinada de registros, sino que se actualiza solo el registro que indica en qué posición del anillo está actualmente apuntando.

Ambos métodos se pueden representar con la Fig. 7.3.

Si bien el módulo ADC tiene como características de soportar casi 2 MSPS, intervienen otros factores que hacen que se reduzcan la tasa de muestreo. El factor preponderante es la cantidad de ciclos de reloj que lleva el procesamiento del algoritmo del filtro, compuestos por multiplicaciones, acumulaciones, actualización de registros; este tiempo se incrementa proporcional a la cantidad de términos que tenga el filtro a procesar. Cuanto más sea el orden del filtro, mayor cantidad de términos tendrá y mayor cantidad de operaciones computacionales tendrá que ejecutar.

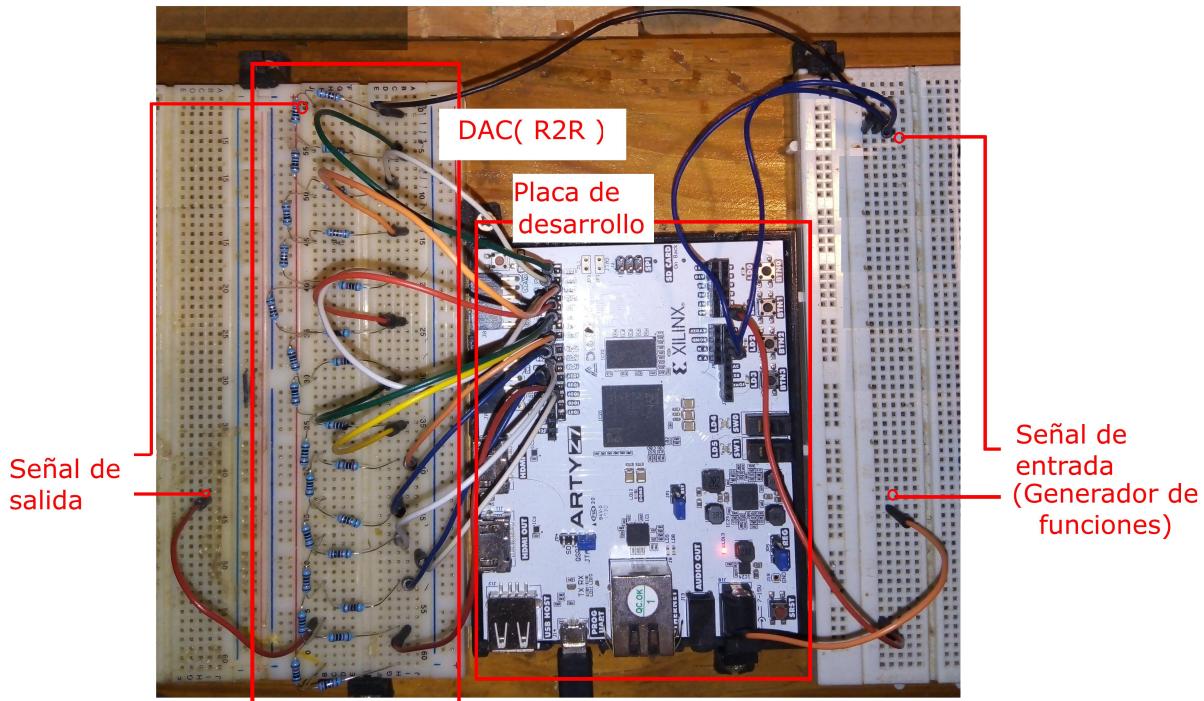


Figura 7.4: Montaje del sistema embebido con FPGA

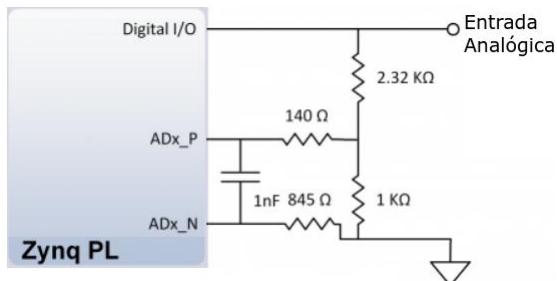


Figura 7.5: Circuito de entrada analógica unipolar incluido en la placa embebida utilizada

Implementación realizada utilizando matriz de puertas lógicas programables, FPGA

El montaje completo consiste en la utilización de una placa de desarrollo FPGA y un circuito R2R, Fig. 7.4. La placa de desarrollo utilizada pertenece a la serie Arty Z7, fabricada por Digilent, contiene un chip programable de la linea Zynq-7000 de Xilinx denominado Arty Z7-20 XCZ020-1CLG400C. Este chip trae las siguientes capacidades:

1- Un ADC integrado de 12bits y de 1MSPS, tiene entrada diferenciales. La placa deja accesible 6 puertos analógicos de forma diferencial y 4 mediante una adaptación que convierte la entrada a unipolar. EL voltaje máximo de entrada en modo diferencial es de 1Vpp, y para el modo unipolar es de 0 de 3.3V.

2 - El integrado trae otros módulos, como el generador de señal de reloj, microprocesadores, etc. Entre estos el integrado contiene una FPGA compuesta de 53200 LUTs y 106400 Flip-Flops.

En cuanto a la entrada analógica utilizada, se usó las entradas unipolares, 7.5. Ésta tiene una frecuencia máxima de entrada de 100 MHz.

Esta placa de desarrollo, no trae el modulo de conversión digital a analógico, para compensar, la empresa que manufatura la placa sugiere la utilización de un R2R, Fig. 7.6.

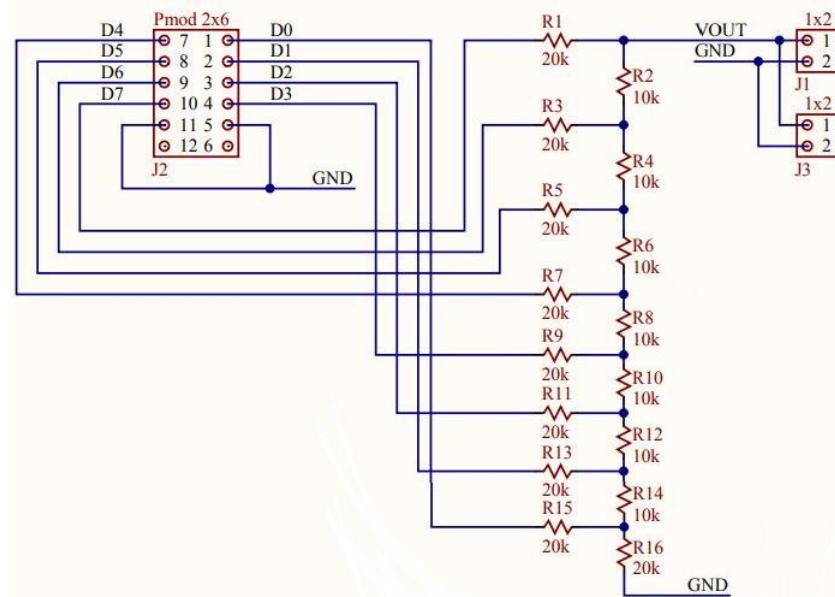


Figura 7.6: Circuito R2R sugerido por la empresa que manufactura la placa embebida

Al igual que la implementación realizada con microcontrolador, la arquitectura del filtro se puede resumir en la siguiente figura 7.7. En la Fig 7.7 se señalan las abstracciones de las operaciones básicas necesarias para el filtro bajo bloques representativos.

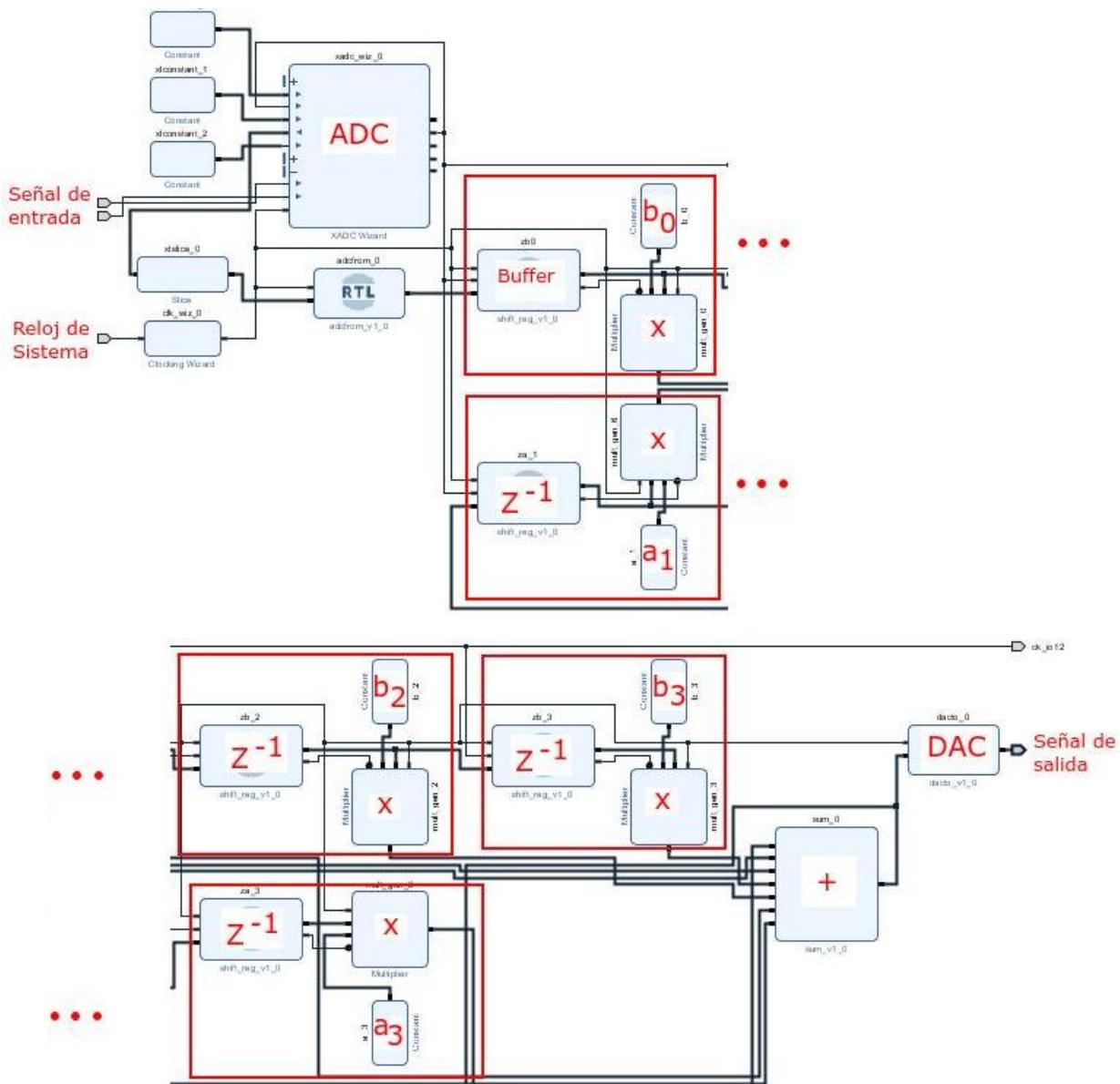


Figura 7.7: Mapeo resultante en el plano z

7.2. Especificaciones de los filtros IIR

7.2.1. Filtros pasa bajo

7.2.1.1. Filtro de Butterworth (microcontrolador)

Datos del filtro

Tipo Hardware	pasa bajo Microcontrolador	Butterworth STM32F407	
fm	22418	Hz	Frecuencia de muestreo
fc	1000	Hz	Frecuencia de muestreo
fp	750	Hz	Frecuencia en la banda de paso
Ap	0.3	dB	Atenuación en la banda de paso
fs	1250	Hz	Frecuencia en la banda de rechazo
As	15	dB	Atenuación en la banda de rechazo
N	8		Orden del filtro necesario

Función de transferencia del filtro analógico

$$H(s) = \frac{1,3354e + 42}{5,4976e + 11 s^8 + 1,7706e + 16 s^7 + 2,8514e + 20 s^6 + 2,9794e + 24 s^5 + 2,2013e + 28 s^4 + 1,1763e + 32 s^3 + 4,4445e + 35 s^2 + 1,0896e + 39 s + 1,3357e + 42}$$

Función de transferencia del filtro digital

$$H(z) = \frac{7,3926e - 08 z^8 + 5,9141e - 07 z^7 + 2,0699e - 06 z^6 + 4,1399e - 06 z^5 + 5,1748e - 06 z^4 + 4,1399e - 06 z^3 + 2,0699e - 06 z^2 + 5,9141e - 07 z + 7,3926e - 08}{z^8 - 6,5697 z^7 + 18,9914 z^6 - 31,5369 z^5 + 32,8915 z^4 - 22,0550 z^3 + 9,2825 z^2 - 2,2414 z + 0,2377}$$

Respuesta en frecuencia

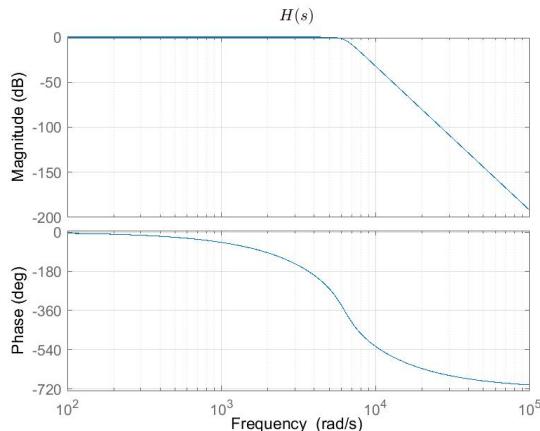


Figura 7.8: Filtro analógico

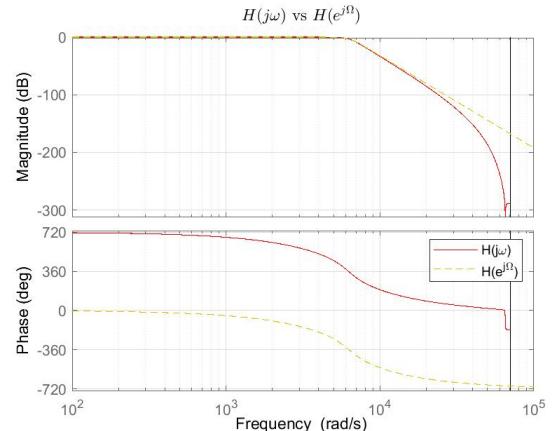


Figura 7.9: Filtro digital

Diagrama de Polos y ceros

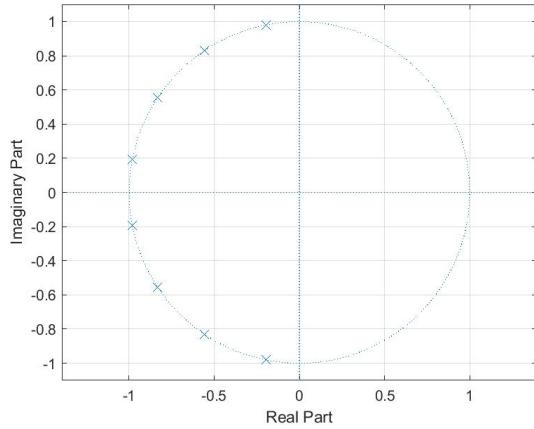


Figura 7.10: Filtro analógico

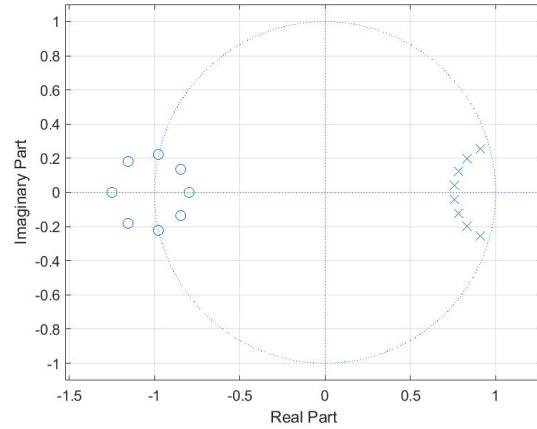


Figura 7.11: Filtro digital

Coeficientes preparados para el microcontrolador

```

1 //Filtro IIR - Butterworth - Pasa Bajo - bilineal(matlab)
2 //Orden: 8
3 //Frecuencia de Muestreo: 22364 (Hz)
4 #define ORDEN 8
5 double coef_x []= {0.00000007,0.00000059,0.00000207,0.00000414,0.00000517,
6     0.00000414,0.00000207,0.00000059,0.00000007};
7 double coef_y []= {-1.00000000,6.56970420,-18.99137075,31.53686888,-32.89149047,
8     22.05499865,-9.28246019,2.24141601,-0.23768525};
```

7.2.1.2. Filtro de Chebyshev (microcontrolador)

Datos del filtro

Tipo Hardware	pasa bajo	Chebyshev STM32F407	
fm	12760	Hz	Frecuencia de muestreo
fc	1000	Hz	Frecuencia de muestreo
fp	950	Hz	Frecuencia en la banda de paso
Ap	0.1	dB	Atenuación en la banda de paso
fs	1150	Hz	Frecuencia en la banda de rechazo
As	15	dB	Atenuación en la banda de rechazo
N	8		Orden del filtro necesario

Función de transferencia del filtro analógico

$$H(s) = \frac{5,5458e + 28}{s^8 + 9,5337e + 03 s^7 + 1,0979e + 08 s^6 + 6,5039e + 11 s^5 + 3,5376e + 15 s^4 + 1,2674e + 19 s^3 + 3,5510e + 22 s^2 + 6,1638e + 25 s + 5,5458e + 28}$$

Función de transferencia del filtro digital

$$H(z) = \frac{2,7126e - 09 z^8 + 2,1701e - 08 z^7 + 7,5954e - 08 z^6 + 1,5191e - 07 z^5 + 1,8989e - 07 z^4 + 1,5191e - 07 z^3 + 7,5954e - 08 z^2 + 2,1701e - 08 z + 2,7127e - 09}{z^8 - 7,4537 z^7 + 24,4351 z^6 - 46,0107 z^5 + 54,4221 z^4 - 41,4021 z^3 + 19,7818 z^2 - 5,4270 z + 0,6545}$$

Respuesta en frecuencia

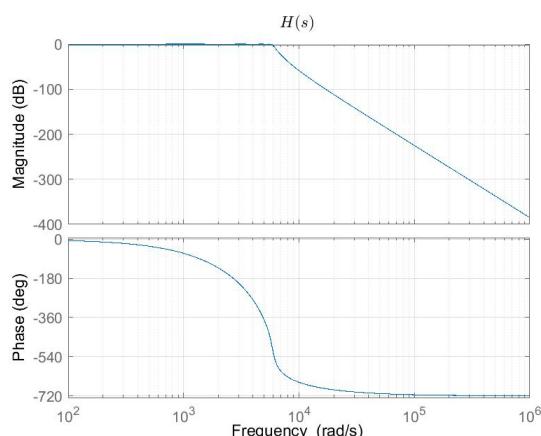


Figura 7.12: Filtro analógico

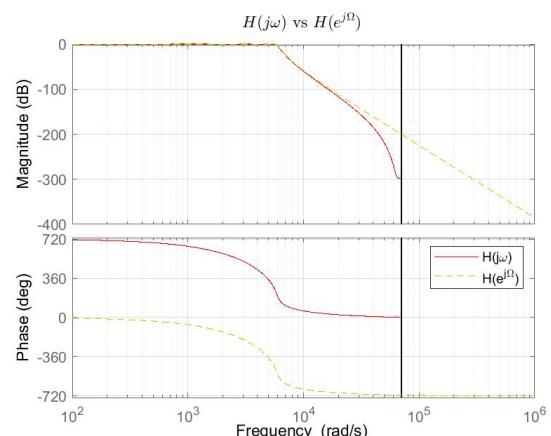


Figura 7.13: Filtro digital

Diagrama de Polos y ceros

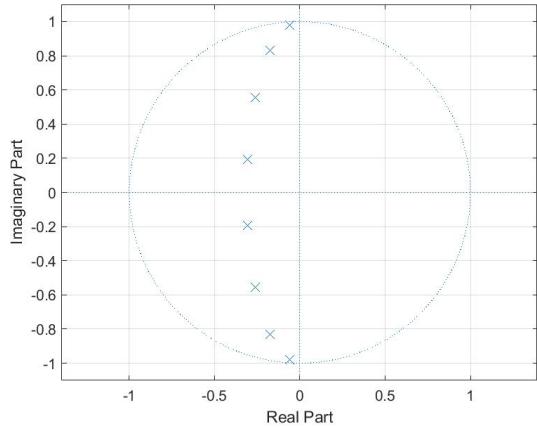


Figura 7.14: Filtro Analógico

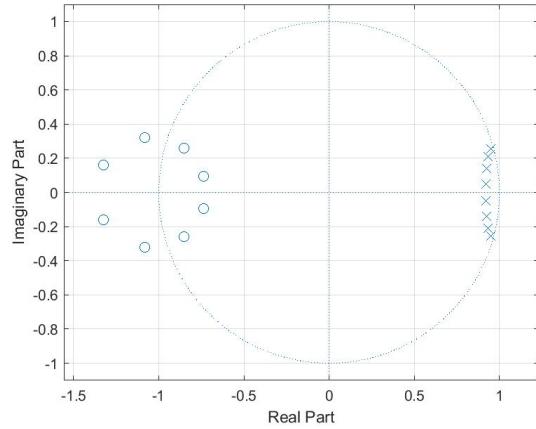


Figura 7.15: Filtro Digital

Coeficientes preparados para el microcontrolador

```

1 //Filtro IIR - Chebyshev - Pasa Bajo - bilineal(matlab)
2 //Orden: 8
3 //Frecuencia de Muestreo: 22364 (Hz)
4 #define ORDEN 8
5 double coef_x []= {0.00000000,0.00000002,0.00000008,0.00000015,0.00000019,
6     0.00000015,0.00000008,0.00000002,0.00000000};
7 double coef_y []= {-1.00000000,7.45366912,-24.43508422,46.01069869,-54.42208055,
8     41.40206273,-19.78176771,5.42697740,-0.65447616};

```

7.2.1.3. Filtro de Butterworth (FPGA)

Datos del filtro

Tipo Hardware	pasa bajo	Butterworth	
fm	FPGA 150000	Hz	Frecuencia de muestreo
fc	1000	Hz	Frecuencia de muestreo
fp	750	Hz	Frecuencia en la banda de paso
Ap	0.5	dB	Atenuación en la banda de paso
fs	6380	Hz	Frecuencia en la banda de rechazo
As	10	dB	Atenuación en la banda de rechazo
N	8	dB	Orden del filtro necesario

Función de transferencia del filtro analógico

$$H(s) = \frac{1,1332e + 27}{1,7592e + 13 s^3 + 1,4104e + 18 s^2 + 5,6538e + 22 s + 1,1332e + 27}$$

Función de transferencia del filtro digital

$$H(z) = \frac{0,0018 z^3 + 0,0055 z^2 + 0,0055 z + 0,0018}{z^3 - 2,4701 z^2 + 2,0717 z - 0,5869}$$

Respuesta en frecuencia

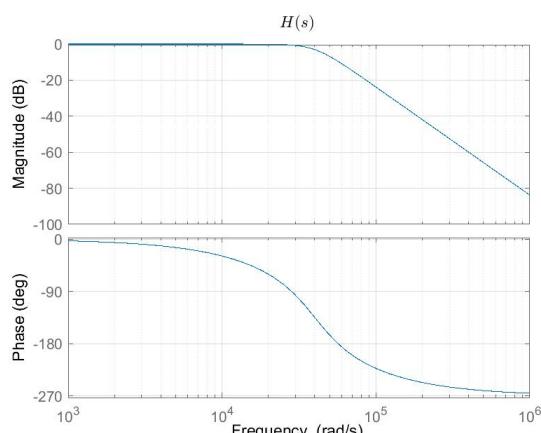


Figura 7.16: Filtro analógico

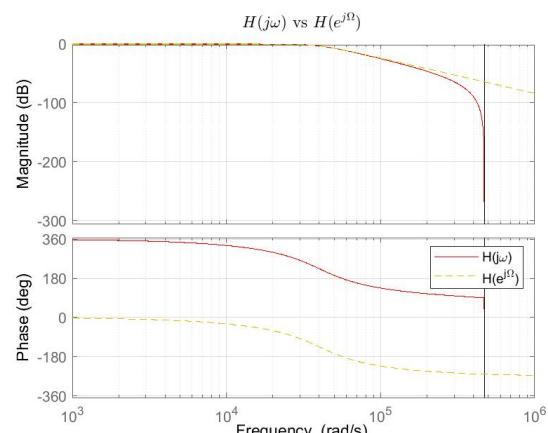


Figura 7.17: Filtro digital

Diagrama de Polos y ceros

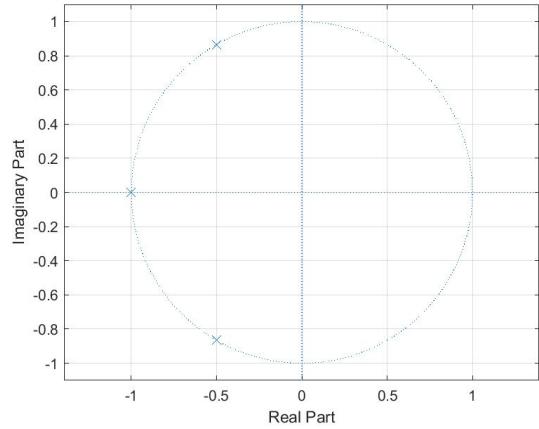


Figura 7.18: Filtro Analógico

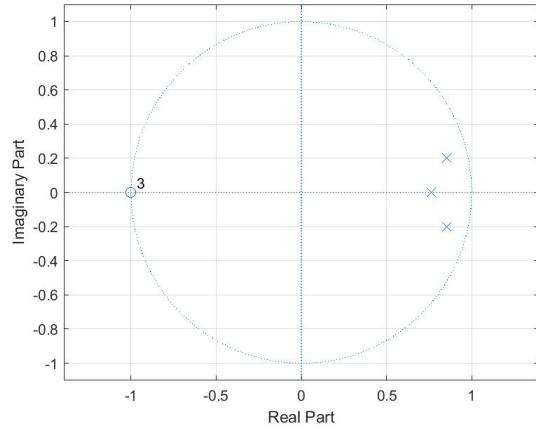


Figura 7.19: Filtro Digital

Coeficientes preparados para la FPGA

```

1 //Filtro IIR - Pasa Bajo - bilineal(matlab)
2 //Orden: 3
3 //Frecuencia de Muestreo: 150000 (Hz)
4 assign dx0=64'0x000000000077C7B3; assign dx1=64'0x000000001675719; assign dx2=64'0x000000001675719
5 ; assign dx3=64'0x000000000077C7B3;
6 assign dy1=64'0x00000002785B8840; assign dy2=64'0xFFFFFFFDEDA83EAA; assign dy3=64'0x00000000963DFB80
7 ;

```

7.2.2. Filtros pasa banda

7.2.2.1. Filtro de Butterworth (microcontrolador)

Datos del filtro

Tipo Hardware	pasa banda Microcontrolador	Butterworth STM32F407	
fm	22418	Hz	Frecuencia de muestreo
f0	1326	Hz	Frecuencia central
B1	865	Hz	Ancho de banda de paso (frecuencia de borde)
Bp	625	dB	Ancho de banda de paso
Ap	0.3	Hz	Atenuación en la banda de paso
Bs	2652	dB	Ancho de banda suprimida
As	10	dB	Atenuación
N	5		Orden del filtro necesario

Función de transferencia del filtro analógico

$$H(s) = \frac{1,6137e + 59 s^5}{3,4028e + 40 s^{10} + 5,9847e + 44 s^9 + 1,7073e + 49 s^8 + 1,9477e + 53 s^7 + 2,8316e + 57 s^6 + 2,1434e + 61 s^5 + 1,9655e + 65 s^4 + 9,3848e + 68 s^3 + 5,7102e + 72 s^2 + 1,3894e}$$

Función de transferencia del filtro digital

$$H(z) = \frac{-1,6072e - 05 z^{10} + 5,3291e - 15 z^9 + 8,0361e - 05 z^8 + 5,6843e - 14 z^7 - 1,6072e - 04 z^6 + 2,5580e - 13 z^5 + 1,6072e - 04 z^4 + 1,2790e - 13 z^3 - 8,0361e - 05 z^2 + 1}{z^{10} - 8,6022 z^9 + 33,8786 z^8 - 80,3886 z^7 + 127,2170 z^6 - 140,2646 z^5 + 109,1107 z^4 - 59,1356 z^3 + 21,3765 z^2 - 4,6561 z + 0,4644}$$

Respuesta en frecuencia

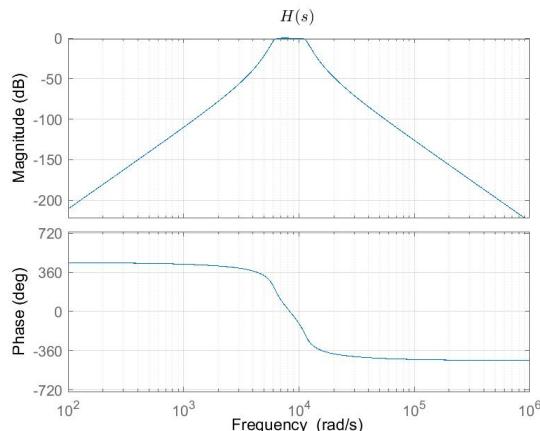


Figura 7.20: Filtro analógico

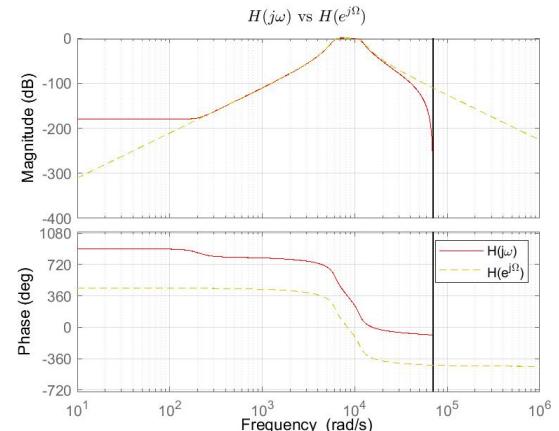


Figura 7.21: Filtro digital

Diagrama de Polos y ceros

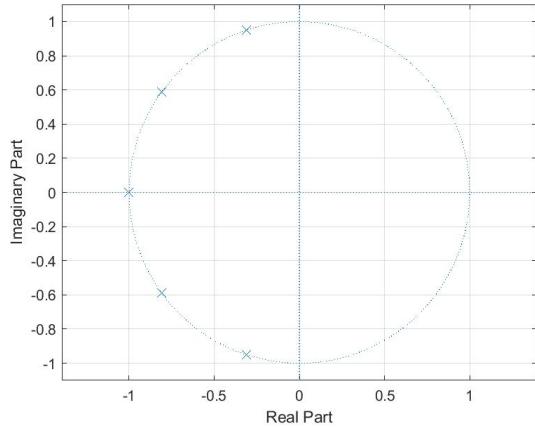


Figura 7.22: Filtro Analógico

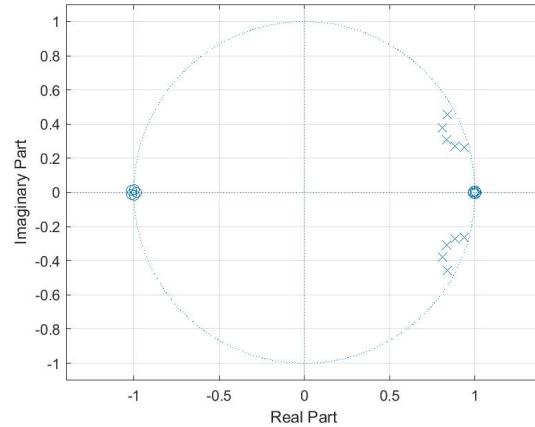


Figura 7.23: Filtro Digital

Coeficientes preparados para el microcontrolador

```

1 //Filtro IIR - Butterworth - Pasa Banda - bilineal(matlab)
2 //Orden: 5
3 //Frecuencia de Muestreo: 22364 (Hz)
4 #define ORDEN 10
5 double coef_x []= {0.00001607,-0.00000000,-0.00008036,-0.00000000,0.00016072,
6 -0.00000000,-0.00016072,-0.00000000,0.00008036,-0.00000000,
7 -0.00001607};
8 double coef_y []= {-1.00000000,8.60220996,-33.87859132,80.38862918,-127.21702273,
9 140.26459652,-109.11066741,59.13558735,-21.37648854,4.65612400,
10 -0.46441013};
```

7.2.3. Filtros pasa alto

7.2.3.1. Filtro de Butterworth (microcontrolador)

Datos del filtro

Tipo Hardware	pasa banda Microcontrolador	Butterworth STM32F407	
fm	22418	Hz	Frecuencia de muestreo
fc	1000	Hz	Frecuencia de muestreo
fp	750	Hz	Frecuencia en la banda de paso
Ap	0.3	dB	Atenuación en la banda de paso
fs	1250	Hz	Frecuencia en la banda de rechazo
As	15	dB	Atenuación en la banda de rechazo
N	8		Orden del filtro necesario

Función de transferencia del filtro analógico

$$H(s) = \frac{4,5036e + 15 s^8}{4,5046e + 15 s^8 + 1,4508e + 20 s^7 + 2,3361e + 24 s^6 + 2,4408e + 28 s^5 + 1,8033e + 32 s^4 + 9,6355e + 35 s^3 + 3,6405e + 39 s^2 + 8,9248e + 42 s + 1,0940e + 46}$$

Función de transferencia del filtro digital

$$H(z) = \frac{0,4874 z^8 - 3,8994 z^7 + 13,6479 z^6 - 27,2957 z^5 + 34,1197 z^4 - 27,2957 z^3 + 13,6479 z^2 - 3,8994 z + 0,4874}{z^8 - 6,5698 z^7 + 18,9917 z^6 - 31,5376 z^5 + 32,8922 z^4 - 22,0554 z^3 + 9,2825 z^2 - 2,2414 z + 0,2377}$$

Respuesta en frecuencia

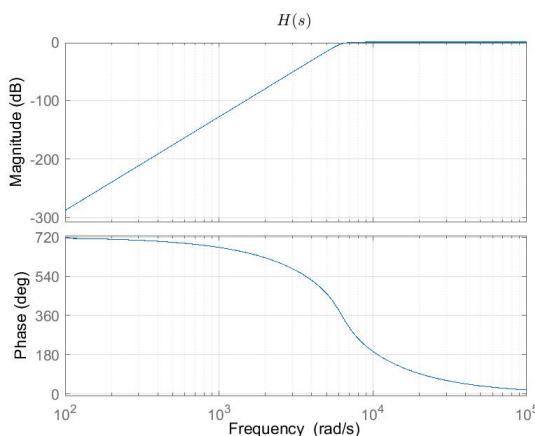


Figura 7.24: Filtro analógico

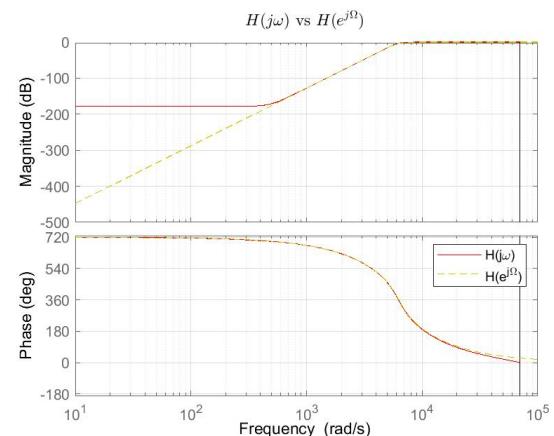


Figura 7.25: Filtro digital

Diagrama de Polos y ceros

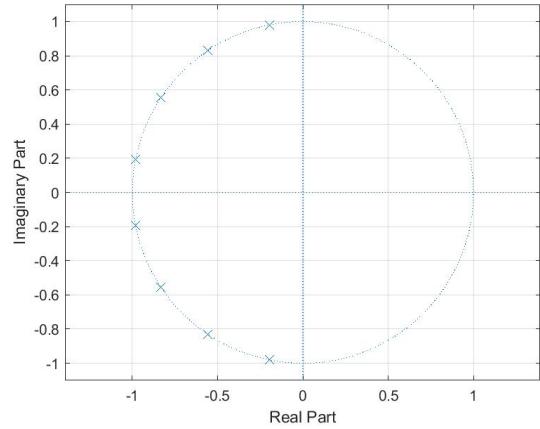


Figura 7.26: Filtro Analógico

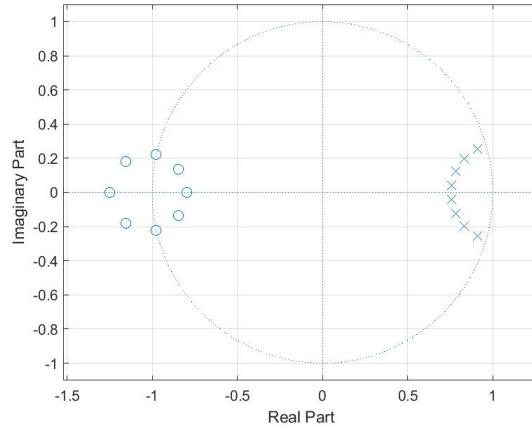


Figura 7.27: Filtro Digital

Coeficientes preparados para el microcontrolador

```

1 //Filtro IIR - Butterworth - Pasa Alto - bilineal(matlab)
2 //Orden: 8
3 //Frecuencia de Muestreo: 22364 (Hz)
4 #define ORDEN 8
5 double coef_x []= {0.48742393,-3.89939144,13.64787002,-27.29574005,34.11967506,
6 -27.29574005,13.64787002,-3.89939144,0.48742393};
7 double coef_y []= {-1.00000000,6.56977298,-18.99171389,31.53756909,-32.89221913,
8 22.05537544,-9.28251651,2.24138942,-0.23767630};
```

7.3. Especificaciones de los filtros FIR

7.3.1. Filtro pasa bajo

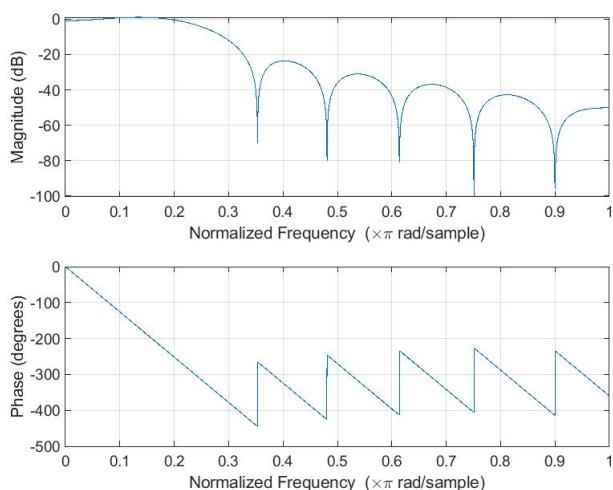
Datos del filtro:

Tipo	pasa bajo		
Hardware	Microcontrolador	STM32F407	
fm	12760	Hz	Frecuencia de muestreo
fc	1500	Hz	Frecuencia de muestreo
N	8	dB	Orden del filtro necesario

Función de transferencia del filtro digital

$$H(z) = -0,0170 z^{14} - 0,0499 z^{13} - 0,0559 z^{12} - 0,0181 z^{11} + 0,0610 z^{10} + 0,1581 z^9 + 0,2376 z^8 + 0,2683 z^7 + 0,2376 z^6 + 0,1581 z^5 + 0,0610 z^4 - 0,0181 z^3 - 0,0559 z^2 - 0,0499 z - 0,0170$$

Respuesta en frecuencia



Coeficientes preparados para el microcontrolador

```

1 //Filtro FIR -pasa_bajo -
2 //Orden: 15
3 //Frecuencia de Muestreo: 22364 (Hz)
4 #define ORDEN 14
5 double coef_x []= {-0.01700257, -0.04993050, -0.05592574, -0.01812776, 0.06104677,
6     0.15810535, 0.23763237, 0.26828832, 0.23763237, 0.15810535,
7     0.06104677, -0.01812776, -0.05592574, -0.04993050, -0.01700257
8     };
9 double coef_y []= {-1.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,
10    0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,
11    0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000
12    };

```

7.3.2. Filtro pasa alto

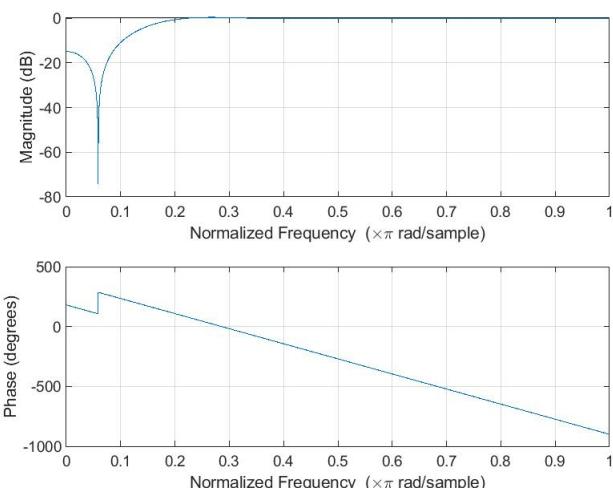
Datos del filtro

Tipo Hardware	pasa alto		
fm	Microcontrolador	STM32F407	
fc	12760	Hz	Frecuencia de muestreo
N	1500	Hz	Frecuencia de muestreo
	8	dB	Orden del filtro necesario

Función de transferencia del filtro digital

$$H(z) = -0,0087 z^{14} - 0,0305 z^{13} - 0,0547 z^{12} - 0,0791 z^{11} - 0,1012 z^{10} - 0,1188 z^9 - 0,1302 z^8 + 0,8659 z^7 - 0,1302 z^6 - 0,1188 z^5 - 0,1012 z^4 - 0,0791 z^3 - 0,0547 z^2 - 0,0305 z - 0,0087$$

Respuesta en frecuencia



Coeficientes preparados para el microcontrolador

```

1 //Filtro FIR -pasa_alto -
2 //Orden: 15
3 //Frecuencia de Muestreo: 22364 (Hz)
4 #define ORDEN 14
5 double coef_x []= {-0.00865977, -0.03052339, -0.05472302, -0.07905268, -0.10115784,
6   -0.11881618, -0.13020860, 0.86585584, -0.13020860, -0.11881618,
7   -0.10115784, -0.07905268, -0.05472302, -0.03052339, -0.00865977
8 };
9 double coef_y []= {-1.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,
10  0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000,
11  0.00000000, 0.00000000, 0.00000000, 0.00000000, 0.00000000
12 };

```