

# FIR-Tools

## HERRAMIENTAS PARA EL DISEÑO DE FILTROS FIR CON LA CALCULADORA HP48G/G+/GX

Alexander López Parrado  
Universidad del Quindío  
Ingeniería Electrónica

### Resumen

Este documento se presenta como manual de referencia de la librería **FIR-Tools** para la calculadora HP48G/G+/GX, la cual posee herramientas para el diseño y análisis de filtros FIR en el dominio del tiempo y de la frecuencia.

## 1 Introducción

FIR-Tools es una librería para la HP48G/G+/GX (Checksum: #61D2h Bytes: 3300) escrita enteramente en SYSRPL[4], con la cual se puede realizar el diseño de filtros FIR por el método de las ventanas y por muestreo en frecuencia. FIR-Tools cuenta con comandos para calcular la respuesta en frecuencia del filtro así como representarla gráficamente.

El diseño de los filtros por el método de las ventanas parte con la respuesta al impulso truncada de un filtro pasa bajo ideal, con la cual se pueden derivar los filtros pasa alto, pasa banda y rechaza banda; adicionalmente se permite multiplicar la respuesta al impulso resultante por alguna de las ventanas (Hanning, Hanning, Blackman, Barlett y Kaiser).

El método por muestreo en frecuencia parte de la especificación de un par de vectores de frecuencias de borde y respuesta deseada (real), con lo cual dependiendo de la cantidad de coeficientes<sup>1</sup> se obtiene la

respuesta al impulso deseada.

Se dispone de comandos para calcular la respuesta en frecuencia de un filtro, magnitud (lineal o decibels) y fase, así como para graficar un vector de datos o una matriz  $N \times 2$ .

## 2 Diagrama de Pila de los Comandos

En lo siguiente se mostrará el tipo de argumentos recibidos por los comandos disponibles en FIR-Tools y sus respectivos resultados.

### 2.1 ideal

Genera la respuesta al impulso truncada a  $N$  coeficientes de un filtro pasa bajo ideal con frecuencia de corte normalizada  $\Omega$ .

#### Entrada

2: Frecuencia de corte normalizada  
1: Cantidad de coeficientes

#### Salida

1: Respuesta al impulso truncada

<sup>1</sup>Actualmente solo es posible realizar el diseño con una longitud del filtro potencia de dos, ya que el método se soporta

en una IFFT

## 2.2 barlett

Genera una ventana de Barlett de  $N$  puntos.

### Entrada

1: Cantidad de coeficientes

### Salida

1: Ventana de Barlett

## 2.3 blackman

Genera una ventana de Blackman de  $N$  puntos.

### Entrada

1: Cantidad de coeficientes

### Salida

1: Ventana de Blackman

## 2.4 hamming

Genera una ventana de Hamming de  $N$  puntos.

### Entrada

1: Cantidad de coeficientes

### Salida

1: Ventana de Hamming

## 2.5 hanning

Genera una ventana de Hanning de  $N$  puntos.

### Entrada

1: Cantidad de coeficientes

### Salida

1: Ventana de Hanning

## 2.6 kaiser

Genera una ventana de Kaiser de  $N$  puntos a partir del parámetro  $\alpha$ . Este comando puede ser muy lento debido al cálculo de la función de Bessel modificada de primera clase y orden cero.

### Entrada

2: alpha

1: Cantidad de coeficientes

### Salida

1: Ventana de Kaiser

## 2.7 setkaiser

Genera el parámetro  $\alpha$  y la cantidad de coeficientes  $N$  para una ventana de Kaiser a partir del rizado y del ancho de la banda de transición como fracción de la frecuencia de muestreo.

### Entrada

2: Rizado

1: Ancho de la banda de transicion

### Salida

2: alpha

1: Cantidad de Coeficientes

## 2.8 sums

Dados dos vectores de datos en la pila, realiza la suma de éstos punto a punto. Útil en el diseño de filtros rechaza banda los cuales se pueden modelar como la

suma de la respuesta al impulso de un filtro pasa bajo con la de uno pasa alto.

#### Entrada

2: Vector1  
1: Vector2

#### Salida

1: Vector1.+Vector2

## 2.9 muls

Dados dos vectores de datos en la pila, realiza la multiplicación de éstos punto a punto. Útil cuando se requiere aplicar una ventana a una respuesta al impulso.

#### Entrada

2: Vector1  
1: Vector2

#### Salida

1: Vector1.\*Vector2

## 2.10 lp2hp

Realiza la conversión de un filtro pasa bajo a un filtro pasa alto:

$$h_{HP}[n] = (-1)^n h_{LP}[n]$$

#### Entrada

1: Respuesta al impulso LP

#### Salida

1: Respuesta al impulso HP

## 2.11 lp2bp

Realiza la conversión de un filtro pasa bajo a un filtro pasa banda:

$$h_{BP}[n] = 2h_{LP}[n] \cos[\Omega_c n]$$

Donde  $\Omega_c$  es la frecuencia central (normalizada) del filtro.

#### Entrada

1: Respuesta al impulso LP

#### Salida

1: Respuesta al impulso BP

## 2.12 sampf

Diseña un filtro por el método de muestreo en frecuencia. El comando **sampf** Requiere: Un vector de longitud par con las frecuencias de borde de la respuesta deseada (**Wedge**), el primer elemento debe ser 0 y el último  $\pi$ ; un vector con la respuesta deseada en las frecuencias de borde (**Aedge**); y la cantidad de coeficientes N, la cual debe ser potencia de dos.

La respuesta en frecuencia deseada será el resultado de la unión mediante rectas de los puntos cuyas coordenadas son los elementos de los vectores **Wedge** y **Aedge**.

#### Entrada

3: Vector de frecuencias borde  
2: Respuesta deseada  
1: Cantidad de coeficientes

#### Salida

1: Respuesta al impulso

### 2.13 freqz

Calcula la respuesta en frecuencia a partir de la respuesta al impulso de un filtro. El comando `freqz` retorna dos matrices de  $256 \times 2$  para la respuesta en magnitud y fase respectivamente, la primera columna de cualquiera de las matrices es un vector de 256 frecuencias equiespaciadas desde 0 hasta  $\pi$ .

La matriz de magnitud puede entregar  $|H[2\pi k/N]|$  o  $20 \log(|H[2\pi k/N]|)$  dependiendo del valor de la bandera colocada en primer nivel de la pila, 0 y 1 respectivamente.

#### Entrada

2: Respuesta al impulso  
1: Bandera

#### Salida

2: Matriz con la respuesta en magnitud  
1: Matriz con la respuesta en fase

### 2.14 plot

Grafica un vector de datos o una matriz  $M \times 2$ . La gráfica se realiza como una serie de puntos.

En el caso de un vector, para cada elemento, se grafica el valor de éste en función del índice que ocupa el elemento en el vector desde 0 hasta  $N - 1$ , donde  $N$  es la longitud del vector.

En el caso de una matriz  $M \times 2$  se graficarán  $M$  puntos donde la columna 1 corresponde al eje vertical y la 2 al eje horizontal.

#### Entrada

1: Vector o matriz Nx2

#### Salida

Ninguna, se grafica en pantalla.

### 2.15 about

Muestra el nombre de la librería y del autor.

## 3 Ejemplos

Los ejemplos de esta sección fueron tomados de [2].

### 3.1

El transductor de cierto sistema de instrumentación entrega una señal cuya máxima componente de frecuencia es de  $500\text{Hz}$ . Esta señal está fuertemente contaminada por la red eléctrica. Diseñe un filtro digital de fase lineal que permita eliminar las componentes de frecuencia de la red.

**Solución:** De acuerdo a [2] se propone un filtro rechaza banda, el cual se obtiene mediante la suma de las respuestas al impulso de un filtro pasa bajo y pasa alto con frecuencias de corte en  $55\text{Hz}$  y  $65\text{Hz}$  respectivamente usando frecuencia de muestreo de  $f_c = 2 * 500\text{Hz} = 1000\text{Hz}$ . Con lo cual las frecuencias de corte normalizadas de los filtros pasa bajo y pasa alto son respectivamente  $\Omega_{LP} = 2\pi 55/1000 = 0.34557$  y  $\Omega_{HP} = 2\pi 65/1000 = 0.4084$ .

Recuerde que para diseñar el filtro pasa alto a partir del pasa bajo, este último debe tener una frecuencia de corte igual a  $\pi - \Omega_{HP} = 2.733186$ .

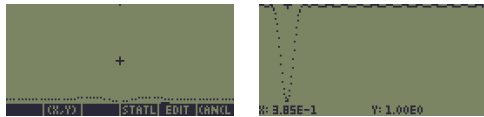
Inicialmente diseñaremos el filtro pasa bajo de 61 puntos con el diagrama de pila mostrado en la figura 1(a) usando el comando `ideal`. Para el filtro pasa alto, se diseña un filtro pasa bajo usando el diagrama de pila de la figura 1(b) y luego se usa el comando `lp2hp`, después se suman las respuestas al impulso con `sums`. Se calcula una ventana de Hamming de 61 puntos usando `hamming` para finalmente multiplicar con `muls`. Si deseamos graficar la respuesta al impulso resultante podemos usar `plot`, pero es recomendable duplicarla antes (`DUP`) con el fin de usar después `freqz`, la figura 1(c) muestra la gráfica de  $h(n)$ . Finalmente calcularemos la respuesta en frecuencia del filtro colocando un 0 en la pila y pulsando `freqz`, se pulsa `DROP`, pues por ahora no interesa la respuesta en fase, finalmente pulsamos `plot`, la figura 1(d) muestra la respuesta en frecuencia del filtro, donde se ve que la frecuencia central normalizada de la banda de rec-

hazo es 0.3851 que corresponde a una frecuencia real de  $0.3851 * 1000 / (2 * \pi) = 61.29 Hz$ .



(a) Cálculo de  $h_{LP}(n)$

(b) Cálculo de  $h_{HP}(n)$



(c) Gráfica del  $h(n)$  resultante

(d) Gráfica de  $|H(\Omega)|$

Figura 1: Resultados arrojados por la HP48 en el ejemplo 3.1

### 3.2

Diseñar un filtro digital pasabajo de longitud  $M = 31$ , tal que la banda de paso sea de 0 a  $f_p = 0.2$ , y la banda de atenuación de  $f_a = 0.3$  a 0.5. Las frecuencias están especificadas en fracciones de la frecuencia de muestreo.

**Solución:** Teniendo en cuenta que el rango de frecuencias solicitado se especificó como fracciones de la frecuencia de muestreo para usar **sampf** se deben multiplicar por  $2\pi$ . De esta forma el diagrama pila entregado a **sampf** es: en el nivel 3, [0 1.2566 1.8850 3.1416]; en el nivel 2, [1 1 0 0] y en el nivel 3, 32 ya que **sampf** solo funciona con longitudes de filtro potencia de dos; este procedimiento se muestra en la figura 2(a). En la figura 2(b) se muestra el resultado después de usar **freqz** y **plot**.



(a) Cálculo de  $h(n)$

(b) Gráfica de  $|H(\Omega)|$

Figura 2: Resultados arrojados por la HP48 en el ejemplo 3.2

## Referencias

- [1] Proakis, John G. y Manolakis, Dimitris G.. *Tratamiento Digital de Señales*, Prentice Hall, 1998.
- [2] Marín, Jorge Iván. *Procesamiento Digital de Señales -Notas de Clase-*, 2003.
- [3] Fliege, N. J.. *Multirate Digital Signal Processing*, John Wiley & Sons, 1994.
- [4] Kalinowski, Eduardo *Programming in System RPL*, 1998.