

## Trabajo práctico Filtrado digital FIR

### 1) Filtro Moving Average con señales senoidales en MATLAB


- a) Genere una señal senoidal con frecuencia fundamental  $f_n = 100$  Hz. Elija una frecuencia de muestreo adecuada.
- b) Agregue ruido gaussiano a la señal senoidal tal que la relación señal-ruido entre la señal senoidal y la señal con ruido sea de 15 dB.
- c) Calcule el valor máximo del orden del filtro ( $N_{\max}$ )  $f_{co} = 2 f_n$ .
- d) Aplique filtrado del tipo moving average a la señal con ruido para un filtro MA con dimensión igual  $N = N_{\max}$ . Utilice la función `filter()` (`help filter`).
- e) Grafique la respuesta en frecuencia y fase del filtro MA. Use la función `freqz()`.
- f) Grafique las señales en el dominio del tiempo sin ruido, con ruido y filtrada, y compare las tres.
- g) Grafique la respuesta en frecuencia de las señales original y filtrada y compare. Utilice la función provista `my_dft`.
- h) Repita los puntos d) a g) para  $N = N_{\max} / 2$  y  $N = N_{\max} * 10$ .

### 2) Filtro Moving Average con señales de audio en MATLAB

- a) Cargue el archivo de audio provisto llamado `Tchaikovsky.mat`. En el mismo encontrará dos variables, la matriz `signal` con dos canales (stereo) y la variable `Fs`. Elija 1 de los 2 canales disponibles.
- b) Agregue ruido gaussiano a esta señal tal que la relación señal-ruido entre la señal y la señal con ruido sea de 50 dB.
- c) Calcule el valor máximo de  $N$  ( $N_{\max}$ ), con las frecuencias  $f_s = F_s$  y  $f_{co} = 22.050$  Hz.
- d) Aplique filtrado del tipo moving average a la señal con ruido para un filtro MA con dimensión igual  $N = N_{\max}$ . Utilice la función `filter()`.
- e) Utilice la función `sound(signal_n, Fs)` para reproducir las señales sin ruido, con ruido y filtrada.
- f) Grafique la respuesta en frecuencia de las señales original y filtrada y compare. Utilice la función provista `my_dft`.
- h) Repita los puntos d) a g) para  $N = N_{\max} / 2$  y  $N = N_{\max} * 10$ .

### 3) Filtro Moving Average con oversampling

El objetivo de este ejercicio es el de cuantificar los efectos de elevar la frecuencia de muestreo en un caso de un filtro MA.

- a) Tome el ejercicio 1 ya resuelto.
- b) Eleve la frecuencia de muestreo,  $f_s$ , y el orden  $N$ , pero sin alterar la frecuencia de corte,  $f_{co}$ , del filtro MA. Tome los valores de  $f_s$  iguales a 2.000, 5.000 y 10.000 Hz.
- c) Cuantifique la diferencia entre la señal de entrada y la señal de salida utilizando la función `rmse()` (root mean squared error).
- d) Determine qué efecto tiene en el error entre ambas señales aumentar la frecuencia de muestreo. 

### 4) Filtrado por ventanas en MATLAB

- a) Use la herramienta `filterDesigner` para diseñar un filtro pasa-banda con frecuencias de corte de 300 Hz y 3.400 Hz (canal telefónico), con ventana Kaiser con  $\beta = 7.5$ , orden 10, con formato punto flotante, precisión doble.
- b) Aumente el orden del filtro a 100. ¿Se modifica la respuesta en frecuencia del filtro?.
- c) Exporte el diseño del filtro como una función de MATLAB: File > Generate MATLAB Code > Filter Design Function.
- d) Utilice como señal de entrada el archivo `Tchaikovsky.mat`.
- e) Aplique a la señal de interés el filtro diseñado en el punto b).
- f) Grafique los espectros de la señal original y filtrada con la función `my_dft()`.
- g) Examine ambas gráficas. ¿Qué diferencia observa entre ambas señales?
- h) Utilice la función `sound(signal_n, Fs)` para reproducir las señales sin filtrar y filtrada.

### 5) Filtrado por ventanas en C en formato punto flotante precisión simple

Se pretende ejecutar desde MATLAB una función descrita en C que implementa un filtro FIR (versión off-line) en el dominio de la frecuencia. Se propone el siguiente ejemplo.

Se cuenta con una señal compuesta por un tono de 300 Hz y otro de 600 Hz, que ha sido corrompida por una señal de 50 Hz (frecuencia de línea eléctrica). Se desea diseñar un filtro pasa-banda que rechace la señal de 50 Hz y que deje pasar los dos tonos.

Se diseña un filtro pasa-banda con `filterDesigner` con frecuencias de corte 200 Hz y 800 Hz tipo Kaiser con beta igual a 7.5 y de orden 100. La frecuencia de muestreo es de 10 kHz.

Los coeficientes del filtro FIR se exportan (Targets > Generate C Header) al archivo `fir_coefs.h` en formato punto flotante precisión simple.

Ejecute los siguientes pasos:

a) Compile en MATLAB las funciones `fir_filter.c` y `fir_matlab_wrapper.c` con el comando:

```
>> mex fir_matlab_wrapper.c fir_filter.c
```

`fir_matlab_wrapper.c` constituye una interfaz entre las variables del Workspace de MATLAB y los argumentos de entrada/salida de las funciones en C.

`fir_filter.c` contiene la función `fir_filter_float()`, la cual implementa la convolución off-line entre los coeficientes del filtro FIR y una señal de entrada, todo en formato punto flotante, precisión simple (float).

b) Analice el código de la función `fir_matlab_offline.m` y ejecútela. ¿Qué observa?

c) Filtre la señal con las funciones que provee MATLAB en formato punto flotante precisión doble

d) Grafique las señales en el dominio del tiempo sin filtrar, filtrada en C y filtrada en MATLAB.

e) Grafique los espectros de la señal sin filtrar y filtrada en C con la función `my_dft()`.

f) Examine las gráficas. ¿Qué diferencia observa entre ambas señales?

## 6) Filtrado por ventanas en C en formato punto fijo Q15

Use el ejemplo del ejercicio 5 para implementar la función `fir_filter_fixed()`, la cual debe ejecutar la misma función que `fir_filter_float()` pero en formato punto flotante Q15.

Recuerde descomentar la última línea del archivo `fir_matlab_wrapper.c`.