

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`.

```
> load Tchaikovsky.mat
```

Se cargarán dos variables, la matriz `signal` con dos canales (estéreo) 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 $f_{co} = 11025$ 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()` para reproducir las señales sin ruido, con ruido y filtrada.

```
> sound(signal_n, Fs)
```

f) Grafique la respuesta en frecuencia de las señales original y filtrada y compare. Utilice la función provista `my_dft()`.

```
> [f, mag_s] = my_dft(signal_f, Fs)
```

h) Repita los puntos d) a f) para $N = N_{\text{max}} / 2$ y $N = N_{\text{max}} * 10$.

3) Filtrado por ventanas en MATLAB

a) Use la herramienta `filterDesigner` para diseñar un filtro:

- Pasa-banda.
- Ventana Kaiser con $\beta = 7.5$, orden 10.
- Frecuencias de corte de 300 Hz y 3400 Hz (canal telefónico), con formato punto flotante, precisión doble (valor por defecto).
- Frecuencia de muestreo 44100 Hz.

b) Aumente el orden del filtro a 50. ¿Se modifica la respuesta en frecuencia del filtro?

c) Exporte el filtro del punto b) haciendo `File > Generate MATLAB Code > Filter Design Function`. Nombre la función como `fir_kaiser_300_3400.m`.

d) Utilice como señal de entrada el archivo `Tchaikovsky.mat`.

e) Aplique a la señal de interés el filtro diseñado en el punto b) haciendo:

```
Hd = fir_kaiser_300_3400;  
b = Hd.Numerator;  
a = 1;  
  
fir_output = filter(b, a, signal);
```

f) Grafique los espectros de la señal original (`signal`) y filtrada (`fir_output`) con la función `my_dft()`.

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

4) Filtrado por ventanas en C, formato punto flotante

Se pretende ejecutar desde MATLAB una función desarrollada en C que implementa un filtro FIR (versión online). Se propone el siguiente ejemplo.

Se cuenta con una señal de entrada compuesta por:

- Tono de 300 Hz, más tono de 600 Hz, más tono 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 con `filterDesigner`:

- Pasa banda.
- Ventana Blackman, orden 100.
- Frecuencias de corte 200 Hz y 800 Hz.
- Frecuencia de muestreo de 10 kHz.

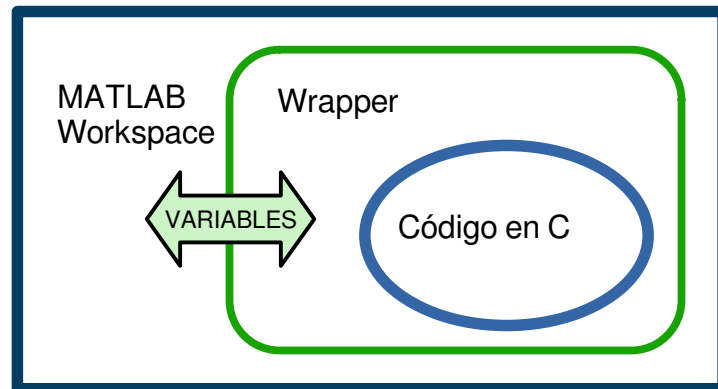
Los coeficientes del filtro FIR se exportan haciendo `Targets > Generate C Header`, al archivo `fdacoeffs.h` en formato punto flotante, precisión simple.

Ejecute los siguientes pasos:

a) Compile en consola las funciones `fir_wrapper.c` y `fir_filter.c`, en este específico orden, con el comando:

```
>> mex fir_wrapper.c fir_filter.c
```

`fir_wrapper.c` construye la 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_online_float()`, la cual implementa la convolución online entre los coeficientes del filtro FIR y una señal de entrada, todo en formato punto flotante, precisión simple (`float`). La función usa un buffer circular para implementar la convolución.

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

5) Filtrado por ventanas en C, formato fixed point

Use el ejemplo del ejercicio 4 para implementar la función `fir_online_fixed()`, la cual debe ejecutar la misma función que `fir_online_float()` pero en formato punto flotante Q15.

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

NOTA

Si se ejecuta MATLAB bajo Windows, se recomienda usar el compilador MinGW [1] y configurarlo cómo se indica en [2].

[1] <https://www.mingw-w64.org/>

[2] <https://www.mathworks.com/matlabcentral/answers/313298-i-already-have-mingw-on-my-computer-how-do-i-configure-it-to-work-with-matlab>