

# Control y Sistemas

## Trabajo práctico: Filtrado digital FIR

### 1) Filtro Moving Average en MATLAB:

- a) Genere una señal senoidal con frecuencia fundamental de 100Hz. Elija una frecuencia de muestreo adecuada.
- b) Agregue ruido a la señal senoidal tal que la relación señal-ruido entre la señal senoidal y la señal con ruido sea de 15 dB.
- c) Aplique filtrado del tipo *moving average* a la señal con ruido para filtros con dimensión  $M$  iguales a 10, 50 y 100. Utilice la función `filter`.
- d) Grafique la respuesta en frecuencia y fase del filtro MA. Use la función `freqz`.
- e) Grafique las señales sin ruido, con ruido y filtrada, y compare.
- f) Grafique la respuesta en frecuencia de las señales original y filtrada y compare. Utilice la función provista `my_dft`.

### 2) Filtro Moving Average 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 a esta señal tal que la relación señal-ruido entre la señal y la señal con ruido sea de 50 dB.
- c) Aplique filtrado del tipo *moving average* a la señal con ruido para filtros con dimensión  $M$  iguales a 10, 50 y 100. Utilice la función `filter`.
- d) Grafique la respuesta en frecuencia de las señales originales y filtradas y compare. Utilice la función provista `my_dft`.
- e) Utilice la función `sound(signal_n, Fs)` para reproducir las señales sin ruido, con ruido y filtrada.

### 3) Filtrado por ventanas en MATLAB:

- a) Use la herramienta `filterDesigner` para diseñar un filtro pasa-banda con frecuencias de corte de 300 Hz y 3400 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 50. ¿Qué observa?.
- c) Utilice como señal de entrada el archivo `Tchaikovsky.mat`.
- d) Aplique a la señal de interés el filtro diseñado en el punto a).
- f) Grafique los espectros de la señal original y filtrada.
- g) Examine ambas gráficas. ¿Qué diferencia observa entre ambas señales?

#### 4) Filtrado por ventanas en C.

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

Se cuenta con una señal compuesta por un tono de 200 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 tonos útiles.

Se diseña un filtro pasa-banda con `filterDesigner` con frecuencias de corte 100 Hz y 800 Hz tipo Kaiser con beta igual a 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` 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_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_to_C.m` y ejecútela. ¿Qué observa?

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

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