Señales y sistemas

Práctico 1
Implementación de filtros en tiempo discreto

Cada ejercicio comienza con un símbolo el cual indica su dificultad de acuerdo a la siguiente escala: ✦ básico, ★ medio, ✶ avanzado, y ✹ difícil.

El objetivo de este práctico es ensayar las diferentes implementaciones de un filtro FIR general en dos plataformas distintas: computadora de uso general y sistema embebido basado en microcrontrolador. Se asume que el filtro está previamiente diseñado y no se discutirán aspectos teóricos sobre el funcionamiento del mismo. Se utilizará una señal de largo N y un filtro de largo L. La generación y visualización de señales se realizará desde un lenguaje de prototipado rápido (se recomienda Python). Todas las implementaciones deberán ser genéricas en todos los parámetros existentes:

- Tamaños de las señales y los filtros.
- Potencias y frecuencias de las señales y ruido.

Para todas las implementaciones de los filtros suponga condiciones iniciales nulas.

```
10 #tamaño de la señal
#a continuación viene la señal, cada muestra separada por un espacio.
0.0 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
```

Figura 1: Ejemplo de formato de almacenamiento de señal.

Ejercicio 1 ✦ (Implementación de filtros en un lenguaje de alto nivel (Matlab o Python) en una computadora de alto nivel)

Para las pruebas iniciales se utilizarán dos señales de entrada: un escalón unitario discreto y una sinusoidal.

(a) Escriba un programa (generar.py) que genere un escalón unitario y lo guarde en un archivo de texto de acuerdo al formato descrito en la Figura 1.

(b) Extienda el código para que genere una señal sinusoidal de amplitud A y frecuencia f y la contamine con ruido aleatorio gaussiano de potencia σ y lo guarde con el mismo formato del escalón

(c) Escriba un programa (procesar.py) que implemente un filtro FIR de forma no causal y con retardo de grupo nulo. Verifique que la respuesta al escalón sea la correcta.

(d) Escriba un programa (visualizar.py) que cargue las señales generadas anteriormente y las grafique.

(e) Filtre la señal con una media móvil, comparando señal original con filtrada. Para una primera implementación y a modo de ejemplo utilice los siguientes parámetros: $A = 1$, $f = 100Hz$, $\sigma = 0{,}1$, $N = 1024$, $L = 5$ y $f_s = 5000Hz$.

(f) Verificar el correcto funcionamiento para diferentes valores del tamaño del filtro. Muestre que el filtro se comporta efectivamente como un pasabajos.

(g) Extienda el código del programa visualizar.py para que evalue cuantiativamente el desempeño del filtro. Se utilizará como medida de performance el SNR (signal to noise ratio). El ruido a la salida se estimará como la diferencia entre la señal original filtrada y la señal con ruido filtrada.

(h) La estructura del código realizado es parte de la evaluación. El sistema completo debe estar formado por los 3 programas mencionados en las partes anteriores. Las señales de entrada y de salida serán leidas y escritas en archivos de texto. Con el formato que se muestra en la Figura 1.

✦Ejercicio 2 **(Implementación de Filtro en C (no causal) en PC)**

El objetivo de este problema es realizar una implementación en lenguaje C del módulo de procesamiento realizado en el ejercicio anterior. Esto consiste en el primer paso para migrar esta implementación a un sistema embebido (Arduino). El filtro será realizado en una computadora de uso general de forma de facilitar su desarrollo y verificación.

(a) Escriba un programa en lenguaje C (procesar.c) que implemente el filtro de forma no causal, asumiendo que toda la señal de entrada está disponible.

(b) El programa deberá comportarse de la misma forma que su equivalente python, es decir, leer y escribir las señales en el mismo formato.

(c) Verificar nuevamente que el filtro implementado se comporta como se espera.