Instituto Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica
Profesor: Dr. Pablo Alvarado Moya
EL 5805 Procesamiento Digital de Señales
II Semestre 2023

Entrega: **5 de octubre, 2023**

Versión: 25 de septiembre de 2023

Tarea 3: Filtrado en el tiempo

En esta tarea realizaremos procesamiento digital de señales en el tiempo. Debe utilizar la siguiente invitación al Github Classroom para iniciar su repositorio. Note que allí está una base de código similar a la proyecto 1, pero con la diferencia de que permite ejecutar archivos `.wav`.

El código construye un ejecutable llamado `tarea3`, que puede utilizar con

```
> tarea3 -f audio1.wav audio2.wav audio3.wav
```

Una vez que se han ejecutado esos archivos, se reactiva el modo de captura de micrófono. Con la tecla `r` (de *replay*) puede volver a ejecutar los archivos indicados en la línea de comando.

Para el diseño de filtros, debe plantear primero el diseño en GNU/Octave. Primero, a partir de la expresión de la función de transferencia en el dominio de la transformada z, debe graficar la respuesta en frecuencia como diagrama de Bode (escalas de frecuencia y de ganancia logarítmicas). Además, debe derivar la correspondiente ecuación de diferencias. La implementación de los filtros deben hacerse en C++ dentro del marco de trabajo contenido en el repositorio del Github Classroom. Ninguno de los filtros debe amplificar ninguna frecuencia; debe cumplirse entonces $\max(|H(\omega)|) = 1$.

Obsérvese que la implementación puede ser compartida entre los filtros, pues solo cambian los coeficientes en cada caso. Estos coeficientes pueden manejarse atributos de clase o como parámetros de plantilla, entre otras posibilidades.

1. Con el proyecto 1, determine un rango de frecuencias para su voz natural.
2. Diseñe un filtro paso-bajas de segundo orden, que suprima completamente la máxima frecuencia representable $F_s/2$, y que tenga como frecuencia de corte (esto es, frecuencia de potencia mitad) 1,25 veces la máxima frecuencia del formante de su voz. Use la tecla '`l`', de *low-pass filter*.
3. Diseñe un filtro muesca, de segundo orden, que elimine la frecuecia media del intervalo determinado por usted en el punto 1. Use la tecla '`m`'. Seleccione un rango interesante de supresión de banda relacionado con su propia voz.
4. Diseñe un filtro paso-banda, de segundo orden que cubra el rango de voz encontrado en el punto 1. Use la tecla '`b`', de *band-pass filter*.
5. Diseñe un filtro paso-altas, que tenga como frecuencia de corte, la frecuencia mínima determinada en el punto 1. Use la tecla '`h`'.

Esta tarea es individual.

Entregables:

1. Código GNU/Octave con el diseño de los filtros.
2. Código C++ con el diseño de los filtros.
3. Archivo PDF con las gráficas de respuesta en frecuencia de los filtros diseñados.
4. El desarrollo debe quedar documentado paso a paso en Git.