UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
DEPARTAMENTO DE INGENIERÍA INFORMÁTICA

Laboratorio 2: Diseñar un filtro FIR

Nombre: Juan Arredondo
Alberto Rodríguez
Curso: Redes de Computadores
Sección 0-L-1
Profesor: Carlos González
Ayudante: Nicole Reyes

3 de Mayo de 2019

Tabla de contenidos

Índice de figuras

1. Introducción

En un sistema dado, una señal f(t) de entrada produce una señal de respuesta r(t) de manera característica del sistema. La función de densidad espectral de la señal de entrada es F(w), mientras que la función de densidad espectral de la respuesta es F(w)H(w). Por lo tanto, el sistema modifica la función de densidad espectral de la señal de entrada. Se puede analizar que el sistema funciona como un filtro de las diferentes componentes de frecuencia.

De esta manera, utilizando la teoría vista en clases, el laboratorio se pide algo similar a lo descrito anteriormente, se tiene una señal de audio en formato .wav la cual debe ser leída mediante la función read de scipy del lenguaje python. Ya con esta función se puede obtener la función de densidad espectral (F(w)), la cual se debe apreciarse en el espectrograma de la función original.

Se solicita aplicar un filtro (FIR) en la función para poder ver la densidad espectral de la señal de entrada en los distintos parámetros. Estos parámetros son para analizar cómo cambia la señal respectos a distintos filtros. Los filtros utilizados son de paso alto, paso bajo y de banda. A cada uno se le calcula la transformada de fourier inversa para poder comparar cada una con la señal original. Después se guardan los audios de los audios en los que se aplicaron los filtros, cada uno de estos se comparar con el original.

El informe está compuesto de esta introducción, un marco teórico donde se detalla cada uno de los filtros a utilizar, un desarrollo de la experiencia donde se detalla cómo se utiliza cada filtro y los gráficos obtenidos. Ya obtenido todos los gráficos y los audio de salida, se realiza un análisis de los resultados obtenidos y la comparación de cada uno de ellos. Finalmente se tiene una conclusión de la experiencia, los cuales detalla que es lo que se concluye de los objetivos y los análisis realizados.

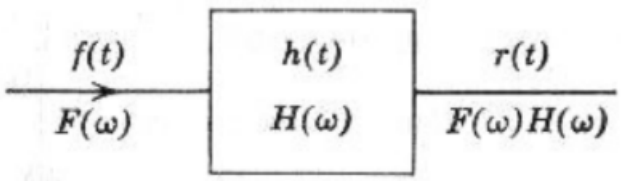

Figura 1: Ilustración de como se representa un filtro como sistema.

2. Marco teórico

En la elaboración del laboratorio número dos se utilizó una variedad de herramientas tanto teóricas como tecnológicas para el desarrollo de este, algunas de estas son:

2.1. Filtro FIR en la teoría:

Antes de definir el filtro FIR debemos entender qué es un filtro y para qué se utiliza. Estos principalmente son sistemas que se diseñan para eliminar ciertos componentes no deseados en una señal, los cuales generalmente se describen en función de su contenido en frecuencia. Hay varios tipos de filtros así como distintas clasificaciones para estos, los principales filtros son de paso alto, bajo y banda.

Antes de definir el filtro FIR debemos entender qué es un filtro y para qué se utiliza. Estos principalmente son sistemas que se diseñan para eliminar ciertos componentes no deseados en una señal, los cuales generalmente se describen en función de su contenido en frecuencia. Hay varios tipos de filtros así como distintas clasificaciones para estos, los principales filtros son de paso alto, bajo y banda.

Basándose en eso es que aparecen los filtros de acuerdo al tipo de respuesta ante un entrada unitaria, como lo es el filtro FIR. Los filtros FIR tienen la gran ventaja de que pueden diseñarse para ser de fase lineal, lo cual hace que presenten ciertas propiedades en la simetría de los coeficientes. Este tipo de filtros tiene especial interés en aplicaciones de audio. Además son siempre estables.

Por el contrario también tienen la desventaja de necesitar un orden mayor respecto a los filtros IIR para cumplir las mismas características. Esto se traduce en un mayor gasto computacional.

En cuanto a su forma matemática, para obtener la salida solo se basan en entradas actuales y anteriores. Por lo que su expresión en un filtro de longitud N es:

$$y_n = \sum_{k=0}^{N-1} b_k * n_{n-k} \tag{1}$$

Donde Bk son los coeficientes del filtro. Ante un estímulo impulsional, la respuesta es

finita lo que justifica su denominación. Por lo tanto la salida y(n) puede escribirse como la convolución de la entrada x(n) con la respuesta impulsional h(n).

$$y_n = \sum_{k=0}^{N-1} h_k * x_{n-k} \tag{2}$$

Aplicando la transformada Z a la expreción anterior:

$$H(z) = \sum_{k=0}^{N-1} h_0 + h_1 * z^{-1} + ... + h_{N-1} * z^{-(N-1)} \tag{3}$$

2.2. Filtro FIR en la práctica:

Para la implementación del filtro FIR en Python se debe hacer uso de la librería Scipy.signal y Pylab por la funcionalidades que nos entregan para una sencilla implementación del filtro FIR.

La lógica principal del filtro se desarrolla en la función "filtfilt()" y "butter()", las cuales permiten realizar el filtro directamente. Para el caso de .butter sus entradas son los vectores coeficientes numerador y denominador del filtro, además de un parámetro extra que da cuenta del tipo de filtro que se quiere aplicar. Por otra parte la función filtfilt aplica un filtro lineal dos veces, una hacia adelante y una hacia atrás. Antes de aplicar el filtro, la función puede rellenar los datos a lo largo del eje dado en una de tres formas: impar, par o constante. Las extensiones pares e impares tienen la simetría correspondiente sobre el punto final de los datos. La extensión constante extiende los datos con los valores en los puntos finales.

De es forma utiliza los resultados obtenidos en la función .butter para generar la función filtrada de audio, donde sus entradas son los vectores y el audio en que se está trabajando, luego encontrando el intervalo se grafica con facilidad el espectrograma.

3. Desarollo de la experiencia

3.1. Herramientas Utilizadas

Primeramente, antes de comenzar la experiencia debemos tener instaladas las herramientas básicas para un correcto funcionamiento del programa, estas herramientas son:

- **Python:** Es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multiparadigma, ya que soporta orientación a objetos, programación imperativa y, en menor medida, programación funcional. Es un lenguaje interpretado, usa tipado dinámico y es multiplataforma.

- **Numpy:** Es una extensión de Python, que le agrega mayor soporte para vectores y matrices, constituyendo una biblioteca de funciones matemáticas de alto nivel para operar con esos vectores o matrices.

- **Scipy:** Es una biblioteca open source de herramientas y algoritmos matemáticos para Python. Además contiene módulos para optimización, álgebra lineal, integración, interpolación, funciones especiales, FFT, procesamiento de señales y de imagen y otras tareas para la ciencia e ingeniería.

- **Matplotlib:** Es una biblioteca para la generación de gráficos a partir de datos contenidos en listas o arrays.

3.2. Importación y espectrograma del audio

En primer lugar al igual que en la experiencia pasada deberemos importar la señal del audio a trabajar, lo cual se realiza con facilidad utilizando la función read de Scipy, obteniendo parámetros como la frecuencia y un arreglo con las energías que la señal presenta.

Posteriormente, deberemos graficar el espectrograma de la señal obtenida al importar el audio lo cual se realiza fácilmente utilizando la función .specgram de matplotlib.pyplot. Dicha función utiliza como parámetros el arreglo con el audio y la frecuencia (variables obtenidas al importar el audio), además aplicamos .colorbar() para que nos muestre una pequeña barra

al costado derecho que indique qué niveles del espectrograma son más altos que otros y qué representa cada color. Esto se representa en las siguientes imágenes.

```
#Importando la señal de audio utilizando
#fs: Frencuencia del audio
#audio: arreglo con la informacion obtenida del audio wav
fs,audio = wavfile.read("handel.wav")

#Cantidad de intervalos del audio
intervalos= len(audio)

#Espectograma del audio original
plt.figure(1)
plt.specgram(audio, Fs=fs)
plt.colorbar()
plt.xlabel('Tiempo (s)')
plt.ylabel('Frecuencia (Hz)')
```

Figura 2: Código que muestra cómo leer y graficar el audio .wav.

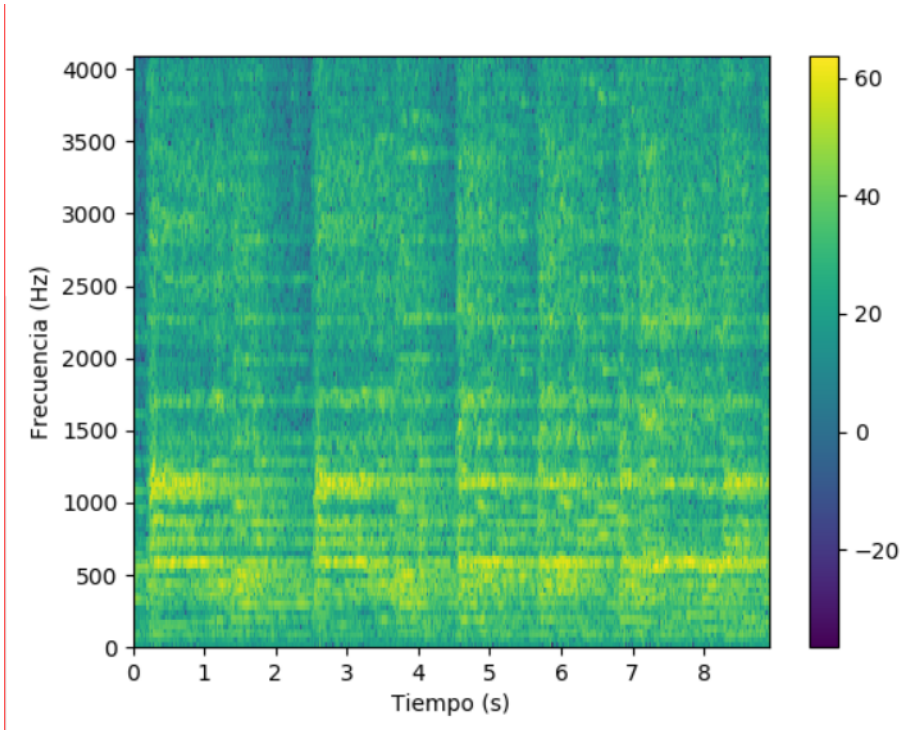

Figura 3: Espectrograma del audio wav

3.3. Aplicación de los filtros

Tal como se explicó en el marco teórico, existe una gran variedad de filtros que se pueden implementar utilizando un correcto uso de las matemáticas, por suerte muchas de estos filtros ya se encuentran programados y podemos hacer uso de sus funciones, una de esas funciones son .butter y .filtfilt, que de manera forman parte de la librería scipy.signal.

.butter permite ubicar el punto en que se quiere realizar aplicar el paso de frecuencias, también puede recibir intervalos de valores y en esos casos hace referencia a un paso de banda, finalmente se le puede añadir un parámetro extra (opcional) en que dependiendo lo que se escriba aplica aplica el filtro (lowpass, highpass o bandpass).

Por otra parte, .filtfilt se encarga de devolver la salida filtrada y lista para graficar, además se basa en el filtro FIR y ni IIR, siendo más eficiente porque ocupa un tipo de ventana Hamming.

Ejemplos de filtros veremos a continuación:

3.3.1. Pasa Bajo:

El filtro pasa bajo se caracteriza porque deja pasar las frecuencias bajas de la señal, si aplicamos este filtro entonces obtenemos la siguientes gráficas.

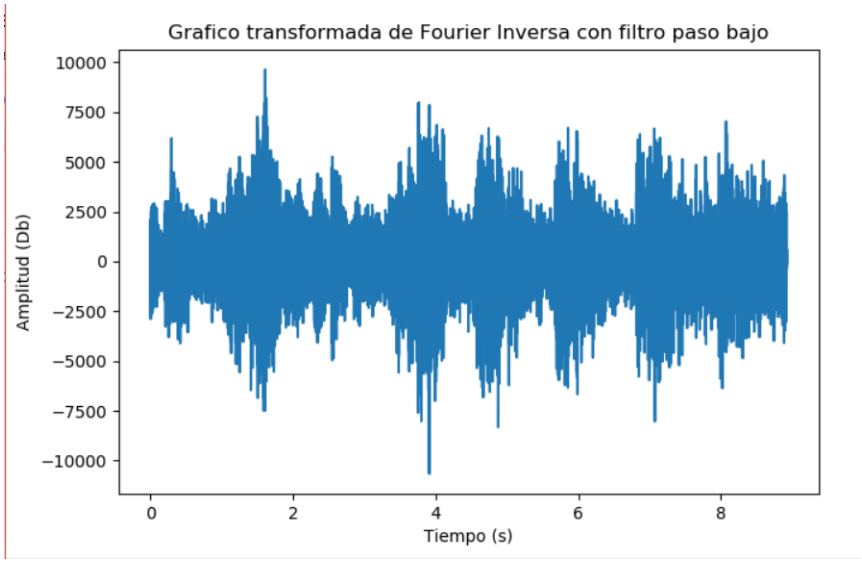

Figura 4: Gráfico de la transformada de Fourier Inversa con filtro paso bajo.

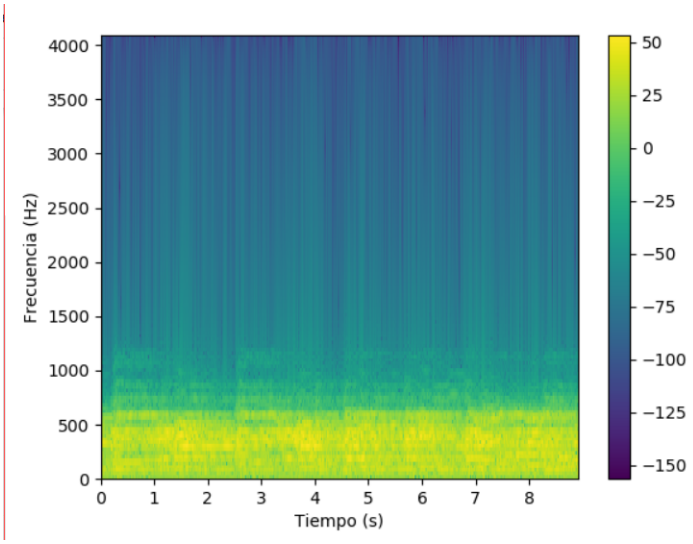

Figura 5: Espectrograma obtenido al aplicar un filtro al 10 % de la frecuencia total.

3.3.2. Paso Alto

Por otra parte, el filtro pasa alto deja pasar las frecuencias altas. Las siguientes imágenes son un ejemplo de su aplicación.

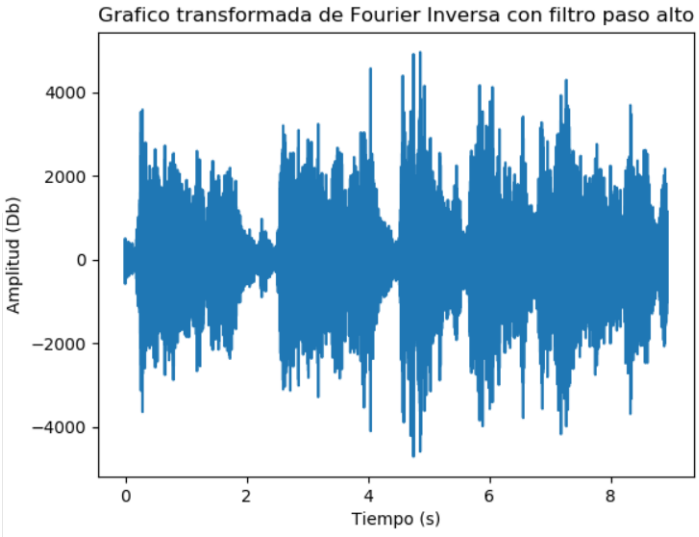

Figura 6: Gráfico de la transformada de Fourier Inversa con filtro paso alto.

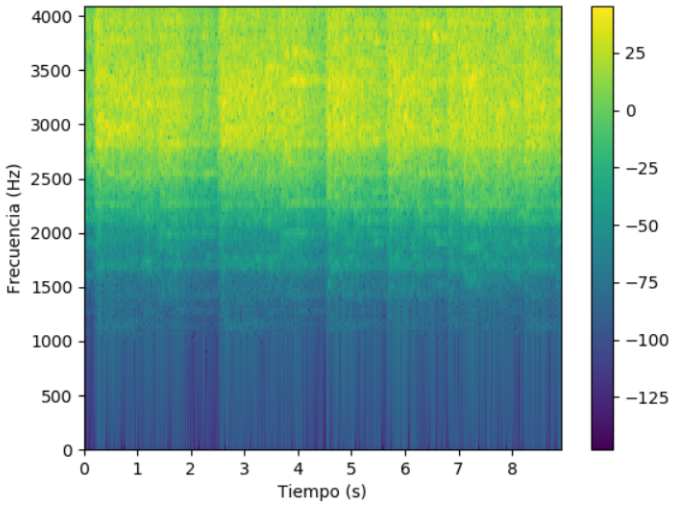

Figura 7: Espectrograma obtenido al aplicar un filtro al 70 % de la frecuencia total.

3.3.3. Paso Banda

Finalmente, el paso cumple la labor de dejar pasar las frecuencias que se asignen dentro del intervalo en la función .butter, un ejemplo aplicado entre frecuencias 30 % a 50 % es:

Figura 8: Gráfico de la transformada de Fourier Inversa con filtro paso banda.

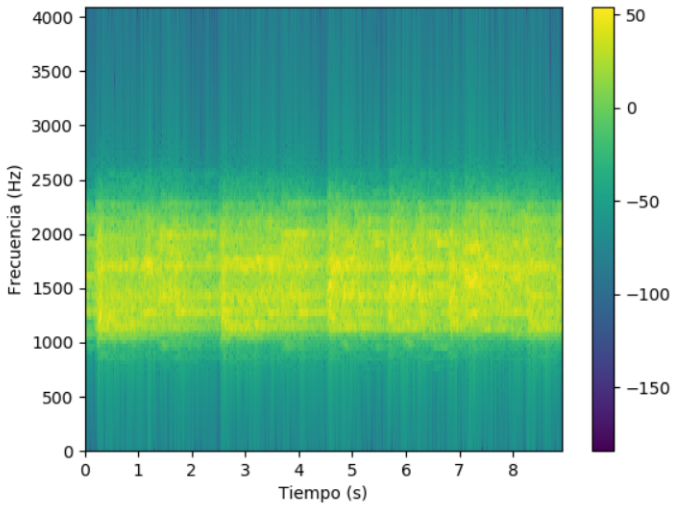

Figura 9: Espectrograma obtenido al aplicar un filtro entre 30 % y 70 % de la frecuencia total

4. Análisis de los resultados

La experiencia arrojó varios resultados que se logra visualizar de manera explícita a través de gráficas con funciones de tiempo y los espectrogramas, algunas cosas que se pueden deducir de estos resultados son:

- **Aplicación de los filtros FIR:** El cambio más importante que podemos percibir al realizar los filtros de paso bajo, paso alto y paso banda, son a través de cada espectrograma obtenido.
- **Aplicación de filtro paso bajo:** Al aplicar el filtro paso bajo podemos observar en el espectrograma como solo se deja pasar las frecuencias bajas menores a aproximadamente 500 Hz. En el audio obtenido llamado "AudioPasoBajo" se puede escuchar un sonido muy "profundo" y las voces se escuchan muy bajos y no se entiende con claridad lo que quieren decir.
- **Aplicación de filtro paso alto:** Al aplicar el filtro paso alto podemos observar en el espectrograma como solo se deja pasar a las frecuencias alta mayores a aproximadamente 2700 Hz. En el audio obtenido llamado "AudioPasoAlto" sólo se puede apreciar las voces altas pero son tan altas las frecuencias que no son perceptibles las palabras.
- **Aplicación de filtro paso banda:** Al aplicar el filtro paso banda podemos observar en el espectrograma como se deja pasar a las frecuencias que están en el rango desde 1000 hasta 2500 Hz. En el audio obtenido llamado "AudioPasoBanda" se puede apreciar que se escuchan las voces y se entienden lo que dice, pero es como escuchar un audio de mala calidad donde las señales están con mucho ruido.

Al realizar una comparación de todas las señales, podemos notar que todas son muy diferentes entre ellas y cómo al aplicar los filtros sus audios cambian notoriamente como es mencionado en los puntos anteriores. Una observación que se puede apreciar es que en todos los audios se pierde la información entonces se pierde la calidad del sonido. También se compara que la señal que queda con más ruido es la paso banda ya que no alcanza a tomar ninguno de los extremos de frecuencia, entonces se podría decir que no tiene ni un inicio bajo ni final alto.

5. Conclusiones

Luego de la experiencia realizada pudimos analizar y entender de mejor manera los conceptos relacionados con la transformada de Fourier en señales reales, utilizando nuevamente un archivo .wav pudimos realizar aplicaciones en esta componente de una forma más sofisticada con el uso de funciones que nos entrega la librería scipy.

Para esta oportunidad tuvimos que aplicar dichos conocimientos aprendidos en cátedra y laboratorio en la elaboración de un filtro FIR, cuyo proceso fue realizado exitosamente obteniendo las salidas deseadas y comprobando la teoría vista en clases.Gracias a la transformada de Fourier y la aplicación de los filtros, se puede ver con mayor facilidad en el espectrograma los picos más altos de las amplitudes y como estas se reducen al aplicar los filtros en sus diferentes "pasos".

Cuando se realiza un cutoff en el 10 % de las frecuencias se ve claramente en el espectrograma como se anulan las frecuencias mayores a dicho valor y la energía se concentra sólo en frecuencias bajas, también se logra verificar con mayor seguridad al extraer los audios de las señales, donde claramente muestran sus efectos al aplicar los filtros.

Una de las dificultades que tuvimos al momento de elaborar el laboratorio fue el desconocimiento de los filtros FIR, donde en un principio fue un impedimento para poder realizar la experiencia. Sin embargo, al investigar más a fondo y comprobar con los contenidos vistos en cátedra se simplificó el problema y se pudo crear el código fácilmente.

Finalmente, podemos decir que gracias al laboratorio número dos hemos reforzado nuestros conocimientos acerca de las señales eléctricas y de igual manera nuestros conocimientos en el área de la programación, donde se espera en un próximo trabajo de laboratorio seguir complementando los conocimientos vistos en este laboratorio con otras nuevas enseñanzas de las clases, y no solo aplicarlos en archivos de audio sino también en otras formas de capas físicas, como pueden ser las imagenes, videos, etc.

6. Referencias

- The Scipy Community (2014) *scipy.signal.butter*. EstadosUnidos: https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.signal.butter.html
- The Scipy Community (2014) *scipy.signal.firwin*. EstadosUnidos: https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.firwin.html
- The Scipy Community (2014) *scipy.signal.filtfilt*. EstadosUnidos: https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.signal.filtfilt.html
- Lathi B.(2001) *Introducción a la teoría y sistemas de comunicación*. Mexico: Limusa Noriega Editores.