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

Laboratorio 1: Señales Análogas y Digitales

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

19 de Abril de 2019

Tabla de contenidos

Índice de figuras

1. Introducción

Desde hace siglos los medios de comunicación han sido de vital importancia para mantenerse informado sobre los acontecimientos, mecanismos como señales de humo, código morse o el cartero ya no son eficientes en los tiempos actuales, es por ello que nacen las redes de computadores, como una necesidad de enviar información al receptor de la forma más rápida y eficiente. Para aquello, la información enviada debe seguir una serie de capas que están restringidas por protocolos en el modelo TCP/IP, estas capas son: Física, acceso a la red, internet, transporte y aplicación. En que para objetivos de este curso se analizan las dos primeras capas.

La primera de ellas tiene directa relación con las señales eléctricas que se envían la información, en ellas la Transformada de Fourier tiene importantes aplicaciones ya que se encarga de transformar una señal del dominio del tiempo, al dominio de la frecuencia, de donde se puede calcular su antitransformada y volver al dominio temporal. De esa manera algunas aplicaciones de Fourier son lograr determinar las amplitudes y fases de cada una de las componentes de frecuencia que tiene la señal.

De esa manera, utilizando la teoría vista en clases, el laboratorio nos propone el siguiente desafío: Utilizar el lenguaje de programación Python y sus librerías Numpy, Matplotlib y Scipy, que nos permitirán la facilidad de importar una señal de audio y experimentar con ella, calculando sus transformadas de Fourier e inversas.

Finalmente, analizaremos los resultados obtenidos por las gráficas de las transformadas y sacaremos conclusiones de ello comparándolo con la teoría.

2. Desarollo de la experiencia

2.1. 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.

2.2. Importación y gráfico del audio

Al comenzar con la experiencia lo primero que debemos hacer será importar el audio a trabajar, para aquello utilizamos un archivo con formato .wav que se encontraba en la plataforma moodle, de esa forma aplicamos la función read de scipy. Dicha función devuelve la frecuencia del muestreo y los datos de un archivo .wav, con ello podremos obtener la cantidad de intervalos, espaciado y el periodo del audio. Posteriormente se utilizan las funcionalidades de Matplotlib que nos permiten graficar, tal como se ve en la imágen.

```
#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)

#Espaciado entre intervalos
dt= 1/fs

#Periodo maximo segun la frecuencia T= intevalos/frecuencia
Tmax = float(intervalos)/float(fs)
#Arreglo Delta T, distancia entre ellos
T=np.linspace(0,Tmax,intervalos)

#Generando grafico
plt.figure(1)
plt.plot(T,audio)
plt.xlabel("T")
plt.ylabel("A")
plt.title("Grafico del audio Wav")
```

Figura 1: Código que muestra cómo leer y graficar el audio .wav.

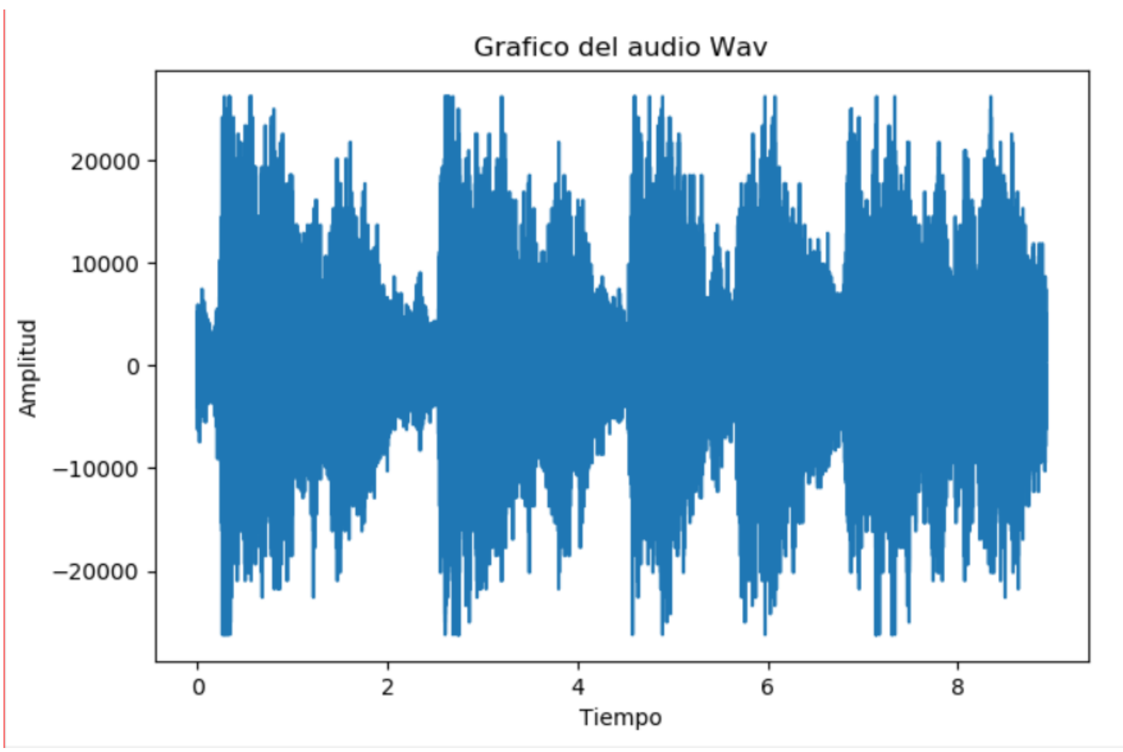

Figura 2: Gráfico del audio wav

2.3. Utilizando la transformada de Fourier

2.3.1. Gráfica de la señal en el dominio de la frecuencia

Para realizar la transformada de Fourier en la señal utilizamos una gran funcionalidad que entrega la librería scipy, dicha funcionalidad es "ftt", la cual realiza la transformada en el programa de forma discreta y siendo muy eficiente a la vez ya que su orden es O(nlog(n)).

Por otra parte, para obtener la frecuencia de la transformada utilizamos "fftreq", funcionalidad que funciona de manera similar a fft. Finalmente graficamos la función tal como lo hicimos anteriormente.

```
#ftt: devuelve la tranformada de fourier
yfourier = fft(audio)/intervalos #normalizada
#fftfreq: devuelve la frecuencia de cada punto de la transformada de fourier
xfourier= fftfreq(int(intervalos),dt)


#Transformada de Fourier
plt.figure(2)
plt.plot(xfourier,abs(yfourier))
plt.xlabel("Frecuencia")
plt.ylabel("Amplitud")
plt.title("Grafico transfromada de Fourier")
```

Figura 3: Código de obtención de la transformada de Fourier y como graficarlo.

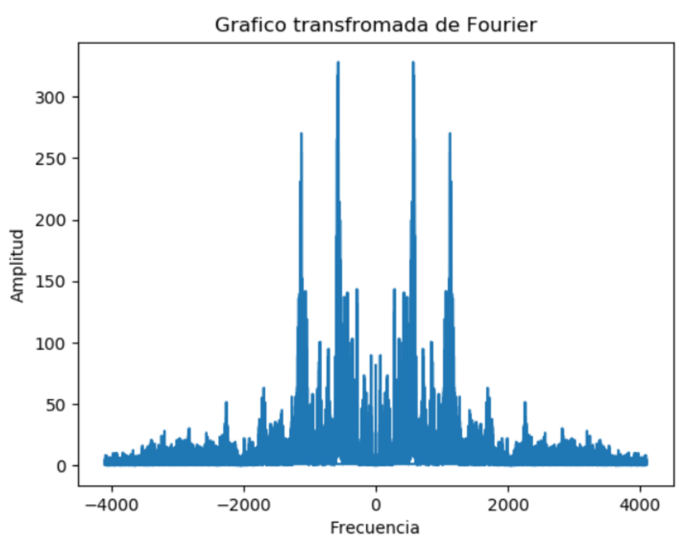

Figura 4: Gráfico de la tranformada de Fourier de la función inicial.

2.3.2. Transformada de Fourier inversa

En cuanto a la transformada inversa, realizamos un procedimiento similar, aplicando la funcionalidad “ifft” de scipy. La que realiza la operación inversa a “fft”, aplicando esta transformada inversa al vector de frecuencias obtenemos el mismo gráfico de la señal en el tiempo

```
#Transformada de fourier inversa

#Se transforma la transfromada de fourier con la funcion ifft
yinvf= ifft(yfourier)

#Se grafica el tiempo con la inversa de la tranformada
plt.figure(3)
plt.plot(T,yinvf)
plt.xlabel("Frecuencia")
plt.ylabel("Amplitud")
plt.title("Grafico transfromada de Fourier Inversa")
```

Figura 5: Código de obtención de la transformada de Fourier inversa y como graficarlo.

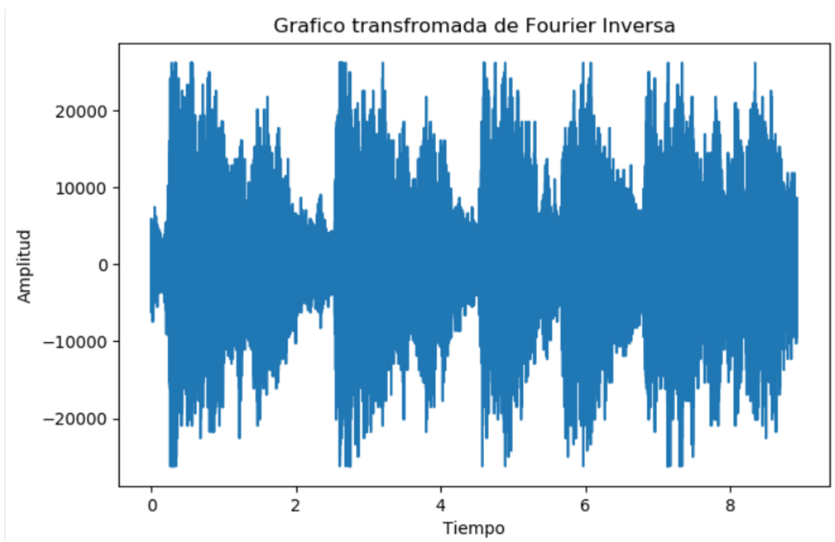

Figura 6: Gráfico de la tranformada de Fourier inversa.

2.4. En el dominio de la frecuencia

2.4.1. El espectro y sus componentes de mayor amplitud

Luego del gráfico obtenido por la transformada de Fourier se pueden apreciar que existen ciertos picos en la amplitudes del gráfico, estos se producen cuando los niveles de sonido se aumentan considerablemente dentro de este mismo, produciendo así niveles bajos que son más bien “ruido” dentro de la gráfica.

2.4.2. Espectro truncado en torno a la amplitud máxima con un margen del 15 %

Para reducir la amplitud máxima en un 15 % realizamos el siguiente método: Obtenemos el máximo de las amplitudes en el gráfico y lo dejamos con un factor de 0.15. Posteriormente, recorrimos el arreglo de amplitudes y las que son inferiores al 15 % del máximo los dejamos con valor cero por ser considerados como “ruido”, finalmente graficamos y obtenemos la siguiente señal.

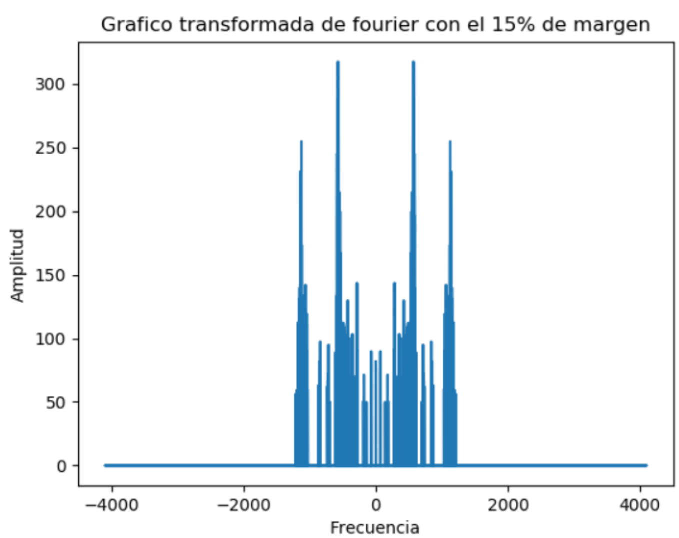

Figura 7: Gráfico de la tranformada de Fourier con el margen del 15 %.

2.4.3. Transformada de Fourier inversa con el margen de 15 %

Finalmente, realizamos nuevamente la transformada inversa de Fourier pero esta vez al gráfico con un margen del 15 %, obteniendo el siguiente gráfico.

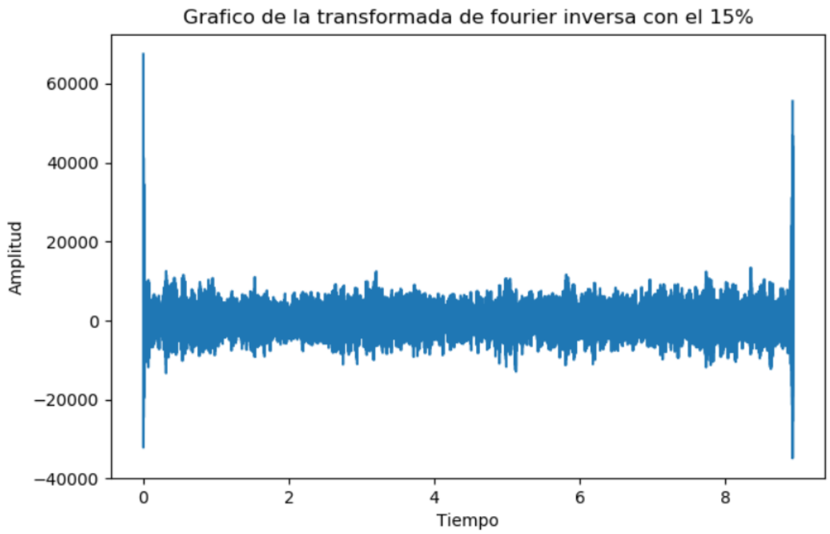

Figura 8: Gráfico de la tranformada de Fourier inversa con el margen del 15 %.

3. Análisis de los resultados

La experiencia arrojó varios resultados que se logran visualizar de manera explícita a través de gráficas con funciones del tiempo y la transformadas de Fourier, algunas cosas que se pueden deducir de estos resultados son:

- **Aplicación de transformada de fourier al gráfico del audio:** El cambio más importante que podemos percibir al realizar la transformada es que las energías se concentran en dos picos en los cuales se marca que las amplitudes son más altas, es decir la densidad de frecuencia es mayor que en las demás. También se puede apreciar que en al gráfico de la transformada, la función es simétrica respecto al eje y (el de la amplitud).

- **Transformada inversa:** Aquí claramente comprobamos la teoría vista en clases, donde desde una función del tiempo se puede transformar a frecuencias y así viceversa, lo que se demuestra obteniendo un gráfico exactamente igual al inicial. Además hicimos uso de funcionalidades que nos permitían transformar la gráfica obtenida a un archivo .wav, el cual al reproducirse genera el mismo sonido que la gráfica inicial. El nombre de este audio en el archivo es AudioInversaTransformada.wav.

- **Truncado de la imágen con margen del 15 %:** Luego de acotar la imágen con margen del 15 % obtenemos una transformada que nos deja en cero las frecuencias cuya concentración sea bajo el 15 % de la amplitud máxima en las frecuencias del sonido. De esta forma al realizar la transformada inversa obtenemos un gráfico en función del tiempo bastante particular, ya que sus niveles generales de amplitud se reducen significativamente, esto lo podemos justificar con el margen realizado a la transformada anteriormente, sin embargo esto no nos dice mucho. De esa forma generamos un archivo .wav (llamado AudioFiltrado.wav) que nos permitiera escuchar la gráfica obtenida y así analizar de mejor manera el resultado. El resultado fue que al escuchar el archivo .wav se escucha constante los sonidos más agudos del audio, mientras que el canto original se escucha muy bajo respecto al audio original.

4. Conclusiones

Como conclusiones generales se pudo analizar y entender los conceptos y cómo funciona la transformada de Fourier en señales reales, en este caso para un archivo .wav. Al realizar la pruebas se puedo ver como cambia la señal del dominio del tiempo al dominio de la frecuencia y viceversa cuando se aplicar la transformada inversa de Fourier, todo esto gracias a las librerías mencionadas anteriormente que nos entregan Python. Gracias a la transformada de Fourier y su transformación al dominio de la frecuencia, se puede ver con más facilidad los picos más alto de las amplitudes y resulta con más facilidad poder modificarlos.

Cuando se realiza el truncamiento de las amplitudes en un 15 %, se puede concluir que en el audio las voces más agudas son las que no se eliminaron en el truncado (por ser la de amplitud mayor) y por eso se siguen escuchando, mientras que el canto de bajo volumen que se escucha es porque el audio que quedó debe ser ahora el de más baja amplitud de audio filtrado.

Un análisis que no nos pudimos explicar con exactitud es porque en el gráfico final del dominio del tiempo (gráfico que se obtiene después de aplicarle la transformada inversa a la función de la frecuencia truncada) al inicio y al final de la función, los picos de amplitud son muy altos, se concluyó que se debe a que al cambiar tan drásticamente el audio, al convertir muchos elementos de la lista en cero, pudo haber deformado de alguna forma el audio y al aplicar la transformada inversa de fourier se obtuvo ese resultado.

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.

5. Referencias

- Cano J.(2012) *Transformada de Fourier discreta en Python con SciPy*. España: https://www.pybonacci.org/2012/09/29/transformada-de-fourier-discreta-en-python-con-scipy/

- The Scipy Community (2014) *scipy.io.wavfile.read*. EstadosUnidos: https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.io.wavfile.read.htmlscipy-io-wavfile-read

- Lathi B.(2001) *Introducción a la teoría y sistemas de comunicación*. Mexico: Limusa Noriega Editores.