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

Laboratorio N 1

Señales analógicas y digitales

Integrantes: Matías Coronado
Gustavo Hurtado

Curso: Redes de Computadores

Profesor(a): Carlos González Cortés

16 de Septiembre de 2019

Tabla de contenidos

1. Introducción

En el siguiente informe se mostrará el código desarrollado para este laboratorio y los resultados que entregó, incluyendo sus respectivos gráficos y las muestras de audio que quedarán adjuntas.

Los objetivos de este laboratorio son los de aplicar la teoría, considerando señales análogas y digitales, pero además, la de una herramienta matemática excepcional, las series y transformada de Fourier. Cabe mencionar que para realizar esto, se hará uso del lenguaje de programación Python, en conjunto con las librerías scipy, numpy y matplotlib, siendo la primera la que entrega herramientas para el procesamiento de archivos de audio (entre otras funciones que no se utilizarán para este laboratorio), mientras que numpy es la librería que permite hacer uso de la transformada de Fourier y otras funciones matemáticas como la norma. Finalmente matplotlib permite generar gráficos.

Entonces, lo que se realizará en este laboratorio es tomar un archivo de audio en formato '.wav' y analizarlo, pero además hacer transformaciones y truncaciones en él, para ver cómo se ve afectado por estos procedimientos y si al realizar estos cambios, el audio se mantiene intacto o se produce algún tipo de distorción en él.

2. Marco teórico

2.1. Señales analógicas

En primer lugar, se debe dar una definición de análogo, que según Maupe (2019) en su artículo "¿Qué son las señales analógicas?", es todo elemento al que se debe asegurar un uso contínuo, siendo desarrollado para garantizar esta continuidad. Por lo tanto, una señal análoga es la que transmite una información que se puede representar con una función matemática continua. Estas suelen ser adecuadas para transmitir elementos multimedia, tales como videos o audio, expandiéndose mediante ondas senoidales (Ver Figura 1), pudiendo ser solo leídas por dispositivos que estén diseñados para este fin específico, es decir, dispositivos análogos.

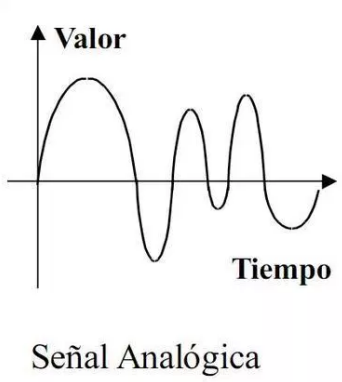

Figura 1: Ejemplo de señal analógica

La página EcuRed (2011), por su parte da la siguiente definición de señal analógica: 'Es una señal producida por un fenómeno electromagnético. Se representa por una función matemática continua; donde varía el período y la amplitud en función del tiempo. Generalmente la intensidad, la temperatura, la presión, la tensión, la mecánica y la potencia son portadoras de este tipo de señal.'.

2.2. Señales digitales

Según Definiciones.de (2016), se trata de la señal cuyos signos representan ciertos valores discretos que contienen información codificada. Los sistemas que emplean señales

digitales suelen apelar a la lógica binaria, de dos estados, los cuales son reemplazados por unos y ceros, que indican el estado alto o bajo del nivel de tensión eléctrica.

Lo que hace interesante a estas señales, es que pierden poca calidad y pueden reconstruirse mediante un procedimiento de restauración. Por otro lado, son de fácil procesamiento y no se ven afectadas en gran parte por los ruidos ambientales.

Definiciones.de (2016), indica lo siguiente acerca de las diferencias entre las señales analógicas y digitales: "Mientras que las señales analógicas son continuas, las señales digitales tienen la capacidad de pasar de un valor a otro sin atravesar los valores intermedios. Esto está vinculado a lo que comentábamos antes de la lógica binaria: cada señal digital sólo puede tomar dos estados, representados por unos y ceros".

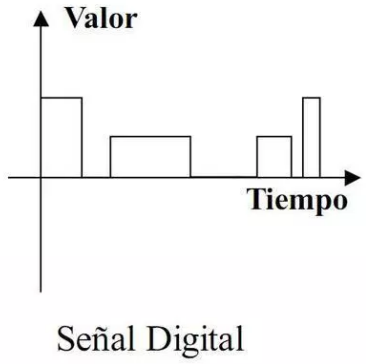

Figura 2: Ejemplo de señal digital

Por otro lado, Maupe (2019) en su artículo "Definición de señal digital", indica que los parámetros que definen una señal digital son los siguientes:

- Altura de pulso, directamente relacionada con el nivel eléctrico.
- Duración, que es el ancho del pulso.
- Frecuencia de repetición, que es la cantidad de veces que se repite el pulso por segundo.

Estos elementos se pueden apreciar fácilmente en la Figura 2.

2.3. Serie de Fourier

Según Domingo Alcaráz (2017) en sus apuntes sobre series de Fourier: "Las series de Fourier son series de términos coseno y seno y surgen en la tarea práctica de representar funciones periódicas generales. Como aplicación constituyen una herramienta muy importante en la solución de problemas en los que intervienen ecuaciones diferenciales ordinarias y parciales."(p.1) Por otro lado, Domingo Alcaráz (2017) indica que la introducción de las series de fourier fue uno de los mayores avances de la física/matemática y sus aplicaciones en el ámbito de la ingeniería, ya que es una potente herramienta para la resolución de problemas con valores en frontera (p. 1).

Básicamente lo que se hace en la serie de Fourier para señales, es descomponer una señal en una serie de otras señales (sen y coseno), que sumadas. dan la señal original.

Como se mencionó antes, las series de Fourier son en base a senos y cosenos, por lo que se pueden determinar las siguientes fórmulas para representar estas series:

$$S(t) = \frac{a_0}{2} + \sum_{n=1}^{+\infty} \left(a_n \cos(nt) + b_n \sin(nt)\right)$$

Figura 3: Serie de Fourier

Teniendo en cuenta que seno y coseno son funciones conocidas, lo que interesa es obtener los coeficientes de la Serie de Fourier, es decir a_n y b_n, lo que se puede hacer de la siguiente forma:

$$a_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \cos(nt)\, dt \text{ para todo } n \geq 0.$$

$$b_n = \frac{1}{\pi} \int_{-\pi}^{\pi} f(t) \sin(nt)\, dt \text{ para todo } n > 0.$$

Figura 4: Coeficientes Serie de Fourier

Siendo $f(t)$ la función que representa la señal a descomponer.

Por otro lado, las serie de Fourier se pueden escribir en su forma compleja, haciendo uso de las fórmulas de Euler. De esta manera, es comúnmente más fácil obtener el valor de las integrales relacionadas a los coeficientes de la señal.

$$f(t) = \sum_{n=-\infty}^{+\infty} c_n e^{i\left(\frac{n\pi t}{L}\right)}, \qquad c_n = \frac{1}{2L}\int_{-L}^{L} f(t)\, e^{-i\left(\frac{n\pi t}{L}\right)}\, dt.$$

Figura 5: Serie de Fourier Compleja

2.4. Transformada de Fourier

Si bien las series de Fourier son una gran herramienta, éstas se encuentra limitada sólo a señales periódicas, por lo que como respuesta a esto, existe la Transformada de Fourier, que vendría siendo una generalización de las series de Fourier.

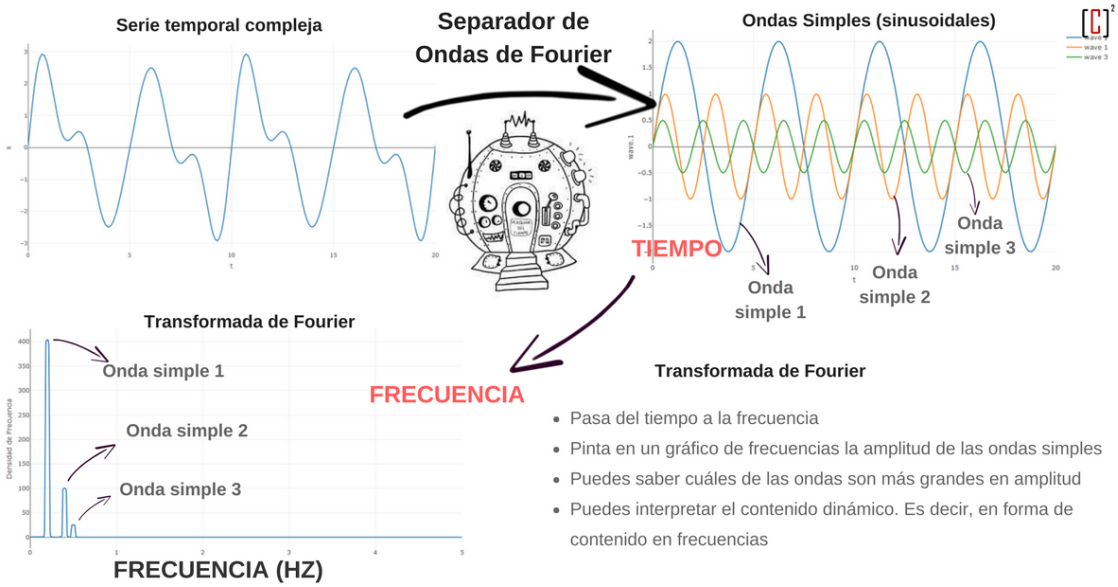

Figura 6: Serie de Fourier Compleja

En la Figura 6 se puede apreciar de manera gráfica el proceso de transformación. En primer lugar se recibe una señal cualquiera. Esta señal es separada en varias ondas simples (sinusoidales), para luego mostrar la frecuencia que llevan estas ondas.

Según Uniovi (1995): "La transformada Fourier de una señal unidimensional o función $f(x)$ continua es una transformación de dicha señal que nos permite calcular la contribución de cada valor de frecuencia a la formación de la señal."

Por lo tanto, en palabras simples, se puede decir que la Transformada de Fourier lo que hace es pasar del dominio del tiempo, al dominio de las frecuencias, para así, ver cuáles son las frecuencias más importantes en la señal original.

$$\mathbb{F}[f(t)] = F(\omega) = \int_{-\infty}^{\infty} f(t) \cdot e^{-j\omega t} \partial t$$

$$\mathbb{F}^{-1}[F(\omega)] = f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) \cdot e^{+j\omega t} \partial \omega$$

Figura 7: Formula Transformada de Fourier y su inversa

Pero así como se puede obtener la Transformada de Fourier, se puede obtener su inversa (Ver Figura 7), de esta manera, se puede pasar del dominio del tiempo al de la frecuencia y del de la frecuencia al del tiempo. Esto tiene una gran utilidad, ya que por ejemplo, al ver las frecuencias que son sólo ruido dentro de la señal, se pueden suprimir estas, para luego volver la señal al dominio del tiempo y entregarla libre de este ruido.

3. Desarrollo de la experiencia

La experiencia fue desarrollada en 4 etapas relevantes, por lo que se mostrará el código relavante a cada una de estas partes con su respectiva explicación y resultados.

3.1. Etapa 1

Leer una señal de audio (.wav) y determinar a qué corresponde cada uno de los valores entregados de la lectura

En esta etapa, mediante el uso de la biblioteca **Scipy** y sus funciones para archivos .wav, se realizó la lectura del archivo 'Handel.wav', que fue entregado dentro del enunciado del laboratorio, para su posterior análisis.

Por lo tanto, se realizó el siguiente código en Python para esta primera etapa:

```
1  import scipy.io.wavfile as sci
2
3  data = sci.read('handel.wav')
4  rate = data[0]
5  audio = data[1]
6
```

En la primera línea, se puede ver la importación de la biblioteca Scipy con sus dependencias para archivos wav, que posteriormente se renombró como "sci". Luego en la línea 3, se aprecia el uso de esta biblioteca con la función read, la cual entrega un arreglo de 2 elementos:

- **Rate**: Es la frecuencia de muestreo del archivo .wav. Se representa como un número, que en este caso es 8192.
- **Audio**: Este es un arreglo de dos elementos, el primero, es una lista de números que representan la amplitud del archivo .wav [0, -202, -2459, ..., 7452, 4930, 0], mientras que el segundo elemento, es el tipo de dato del archivo, que en este caso es int16

3.2. Etapa 2

Graficar la función de audio en el tiempo

En esta etapa lo que se realizó es hacer uso de la variable data y rate, para poder realizar el gráfico.

Dado que rate es la frecuencia de muestreo y $1/Frecuencia = Periodo$, entonces $1/Rate = Tiempo$, pero como queremos un arreglo de estos tiempos, entonces hacemos uso del siguiente código para realizar la transformación:

```
1   import matplotlib.pyplot as matp
2   import numpy
3
4   def graph(datax, datay, title, ylabel, xlabel):
5       matp.plot(datax, datay)
6       matp.title(title)
7       matp.ylabel(ylabel)
8       matp.xlabel(xlabel)
9       matp.show()
10
11  time = numpy.linspace(0,len(audio)/rate, num=len(audio))
12
13  graph(time,audio,"Grafico amplitud vs tiempo: Original", "Amplitud", "
    Tiempo (s)")
14
15
```

Como se puede ver en la página oficial de la biblioteca Numpy, específicamente de la función linspace, lo que hace esta función es un arreglo de números, que en este caso, parte desde 0, hasta el largo del arreglo de audio, pero que va aumentando en función del largo del arreglo del audio, entregando un arreglo del tipo [0.00000000e+00 1.22071982e-04 2.44143964e-04 ... 8.92468261e+00 8.92480469e+00 8.92492676e+00], que será almacenado en la variable time de la línea 11.

Luego para graficar se hace uso de la biblioteca Matplotlib y sus dependencias pyplot, que será renombrado como matp- Mediante la función graph de la línea 4, se solicitan

los datos del eje x, eje y, título, etiqueta del eje y y etiqueta del eje x. En este caso, los datos del eje x son el tiempo, es decir, la variable time. En el eje y es la amplitud del audio, que en este caso, es la variable audio. Con esto se obtiene el gráfico de la Figura 8.

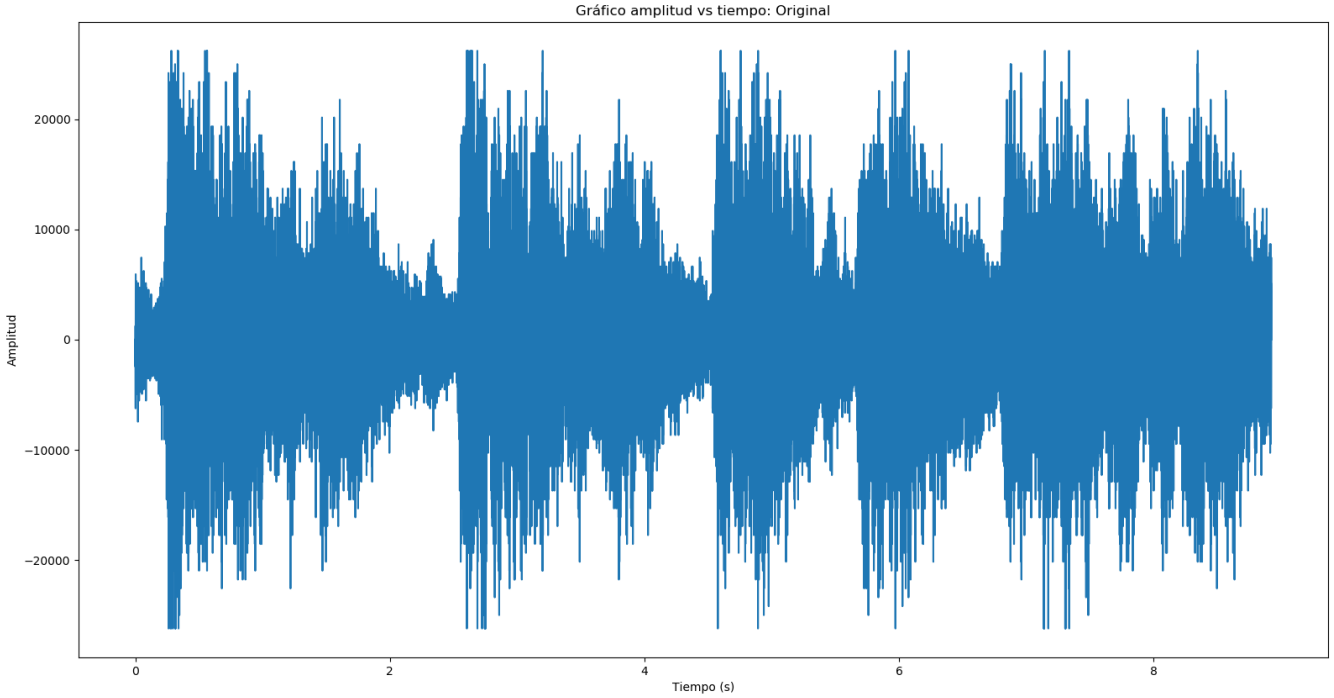

Figura 8: Resultado gráfico etapa 2

3.3. Etapa 3

Calcular la transformada de fourier de la señal de audio:

- **Graficar la señal en el dominio de la frecuencia.**
- **Al resultado del punto 3, calcular la transformada de Fourier inversa.**
- **Comparar con la señal leída en el punto 1.**

Para esta etapa, se hizo uso de varias funciones de la biblioteca Numpy, que son fft (Fast Fourier Transform), fftfreq (Fast Fourier Transform Frequencies), ifft (Inverse Fast Fourier Transform) y abs (Valor absoluto), que se pueden ver en el siguiente código:

```
1    import scipy.io.wavfile as sci
2    import matplotlib.pyplot as matp
3    import numpy
4
5    fourier = numpy.fft.fft(audio)
6    fourierFreq = numpy.fft.fftfreq(len(audio), 1/rate)
7
8    graph(fourierFreq,numpy.abs(fourier),"Grafico amplitud vs frecuencia:
     Transformada de Fourier", "Amplitud", "Frecuencia (Hz)")
9
10   fourierInv = numpy.fft.ifft(fourier)
11
12   graph(time, fourierInv,"Grafico amplitud vs tiempo: Transformado", "
     Amplitud", "Tiempo (s)")
13
14   sci.write('handelTransformed.wav', rate, numpy.int16(fourierInv))
15
```

En la línea 5 se realiza el cálculo de la transformada de Fourier, en la que se obtienen las amplitudes en el dominio de la frecuencia de la señal analizada en forma de un arreglo de este tipo [5.96734100e+06 +0.j 7.04949552e+04+30329.77067247j 5.79698388e+03+40276.67933462j ... -4.10766288e+04-10696.21829713j 5.79698388e+03-40276.67933462j 7.04949552e+04-30329.77067247j]

En la línea 6 se calculan todas las frecuencias de la señal, haciendo uso de el largo del arreglo de amplitudes en función del tiempo y de $1/rate$, que como se mencionó antes, vendría siendo el tiempo/período, por lo que se genera un arreglo de frecuencias de este tipo [0 0.11204574 0.22409147 ... -0.33613721 -0.22409147 -0.11204574]

Luego en la línea 8 se grafica utilizando como datos para el dominio, las frecuencias obtenidas en fourierFreq y para el eje y, las amplitudes en valor absoluto obtenidas de la variable fourier, obteniendo el gráfico de la Figura 9.

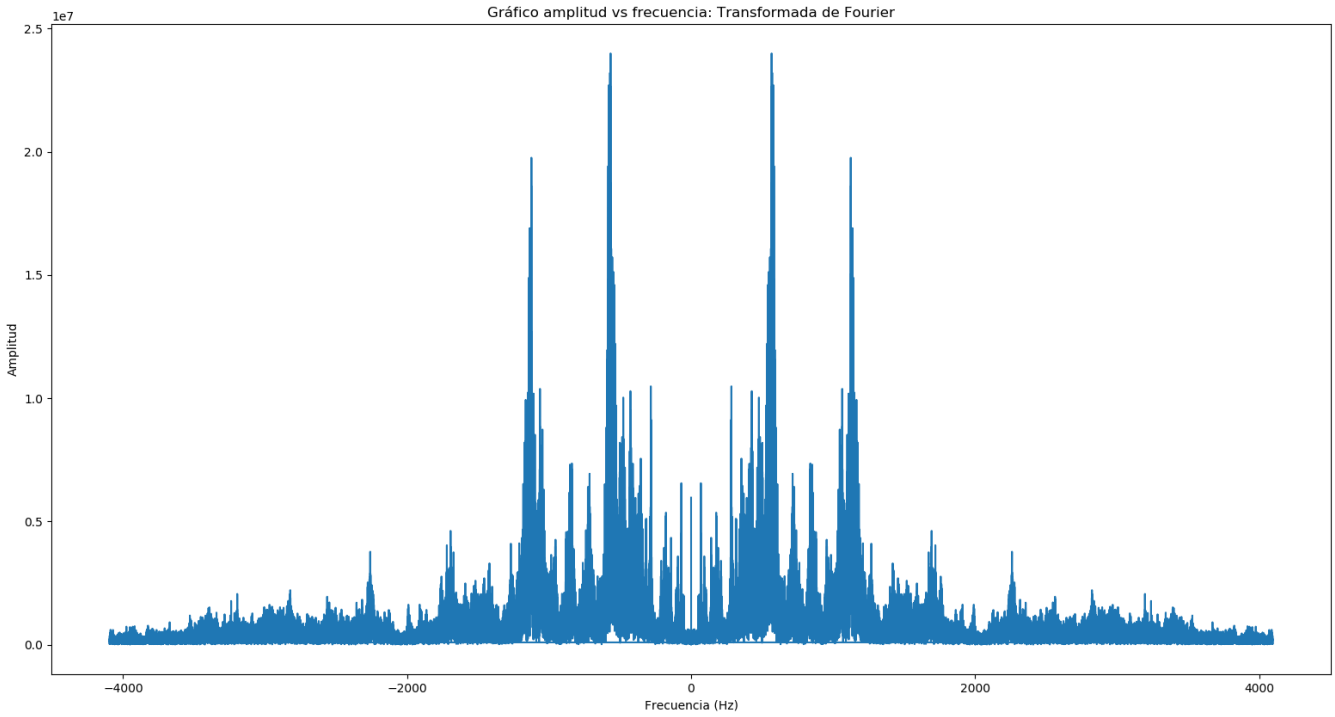

Figura 9: Primer resultado gráfico etapa 3

Posteriormente, en la línea 10 se realiza la inversa de la transformada de fourier, para así pasar del dominio de la frecuencia, al del tiempo nuevamente. Con esto se está "deshaciendo"la transformada inicial de la línea 5 y volver a realizar la gráfica que se aprecia en la Figura 10.

Finalmente en la línea 14 utilizando los datos de la inversa y del rate del audio original, se vuelve a escribir un archivo .wav de nombre "handelTransformed.wav".

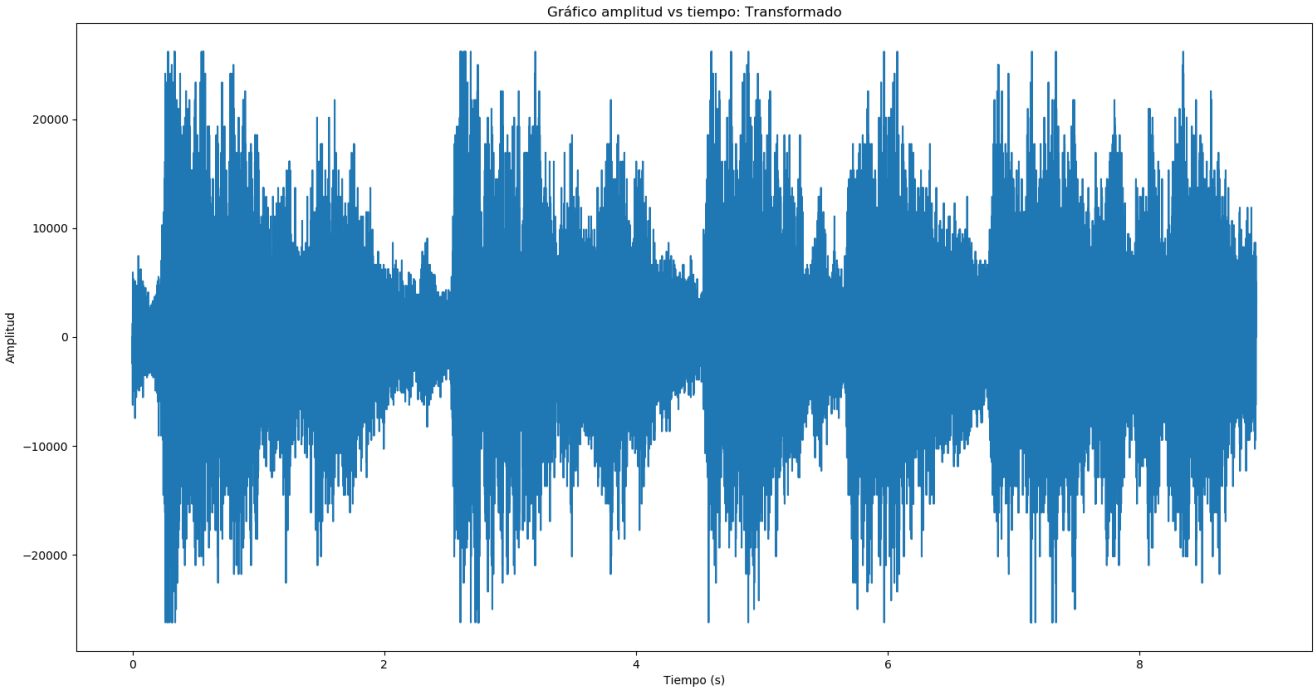

Figura 10: Segundo resultado gráfico etapa 3

3.4. Etapa 4

Analice el espectro de frecuencias obtenido en el punto 3:

- **Determine los rangos de frecuencia más importantes de la señal de audio.**
- **Genere un nuevo espectro de frecuencias truncando la transformada de fourier original al valor 0 fuera de los rangos de frecuencias principales.**
- **Calcule la transformada de fourier inversa del nuevo espectro truncado (4.b) y compare la señal la original (1).**

En esta última etapa se determinaron las frecuencias más importantes basándose en los resultados entregados en la Figura 9, dónde se estableció que desde la frecuencia 0 a aproximadamente 1250 $[hz]$ y de 0 a aproximadamente -1250 $[hz]$, se encontraba la porción de frecuencias más relevantes.

Posteriormente considerando estas frecuencias más importantes, se trunca la señal transformada mediante el proceso de transformada de fourier, llevando a 0 todos los valores

del arreglo entregado por la función fft del audio original que se encuentren más allá de la frecuencia 1250 [hz] y -1250 [hz], proceso que se puede apreciar en el siguiente código (en este caso no se agregan las importaciones de la biblioteca Numpy, Scip ni Matplotlip, asumiendo que ya se encuentran a principio del código fuente. Lo mismo con la función graph, que fue mostrado y explicado en la página 8):

```
1    fourierTrunc = fourier
2    for i in range(11156, len(fourierTrunc)):
3    if fourierFreq[i] < 0:
4        for j in range(i, 61957):
5            fourierTrunc[j] = 0
6        break
7    else:
8        fourierTrunc[i] = 0
9
10   fourierTruncInv = numpy.fft.ifft(fourierTrunc)
11
12   graph(time, fourierTruncInv,"Grafico amplitud vs tiempo: Truncado", "
     Amplitud", "Tiempo (s)")
13
14   graph(fourierFreq, numpy.abs(fourierTrunc),"Grafico amplitud vs
     frecuencia: Truncado", "Amplitud", "Frecuencia (hz)")
15
16   sci.write('handelTransformedTrunc.wav', rate, numpy.int16(
     fourierTruncInv))
17
18
```

Desde la línea 2 a 8 lo que se realiza es recorrer desde la posición 11156, que equivale a la frecuencia 1249.9822466592807 (aproximadamente 1250 [hz]), hasta el final del arreglo con las amplitudes del audio en términos de la frecuencia, llevando a 0 todos los elementos que estén en ese rango. Esto en un principio sólo se hace para los valores positivos de las frecuencias, pero cuando se llega a los elementos negativos, se vuelve a realizar un ciclo que lleva a 0 todos los elementos que se encuentran más allá de la frecuencia -1249.9822466592807 (aproximadamente -1250 [hz]). De esta manera, se logra truncar todo

el espectro de frecuencias que se consideró como no relevante, tanto en el lado positivo, como en el negativo de las frecuencias.

Posteriormente en la línea 10 se realiza la transformada inversa de fourier al arreglo de amplitudes en términos de la frecuencia que anteriormente fueron truncados. Es decir, primero se realizó la transformada de fourier al audio .wav, para luego truncar las frecuencias no fundamentales y posteriormente realizar su inversa para volverlo al dominio del tiempo. De esta parte lo que se obtuvo es el gráfico de las Figura 11, que se realizó en la línea 12.

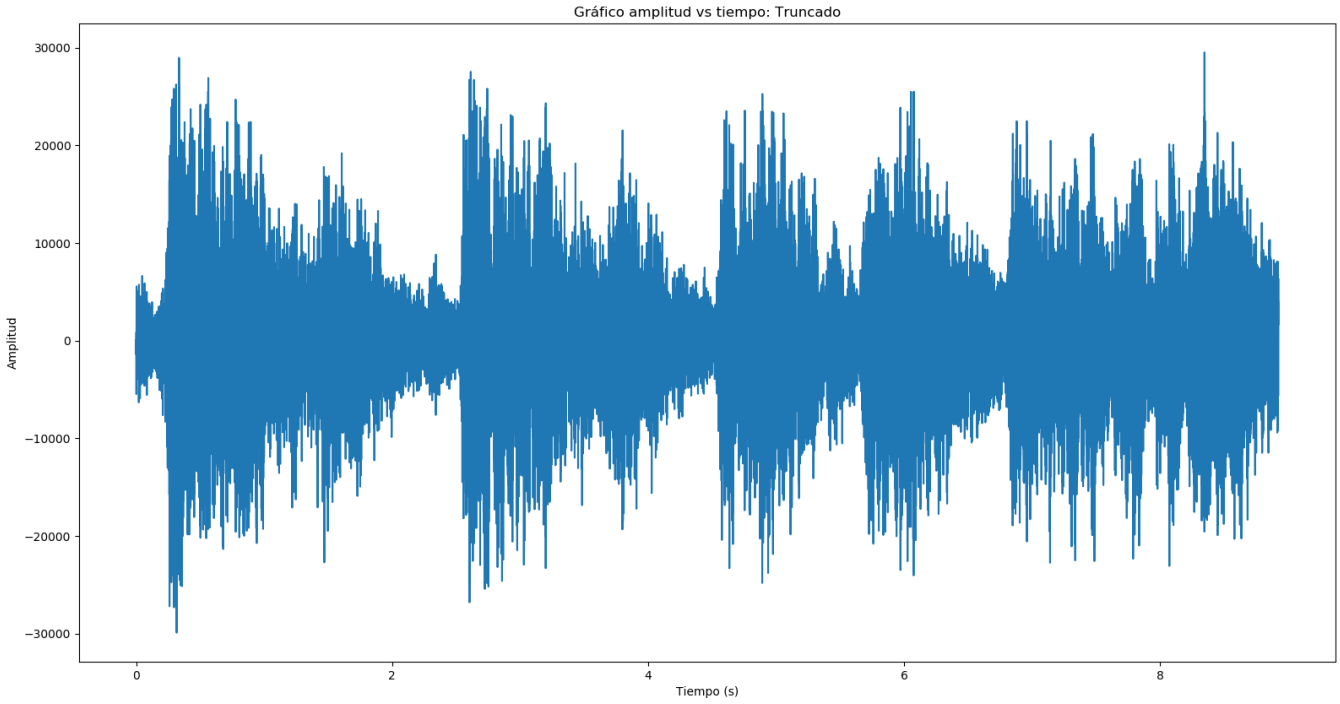

Figura 11: Primer resultado gráfico etapa 4

Luego en la línea 14 se realiza el gráfico de la amplitud en el dominio de la frecuencia, pero del audio ya truncado, el cual se puede apreciar en la Figura 12, para finalmente volver a utilizar los datos de la inversa del audio truncado y del rate del audio original (que es el mismo del truncado), se vuelve a escribir un archivo .wav de nombre "handelTransformedTrunc.wav".

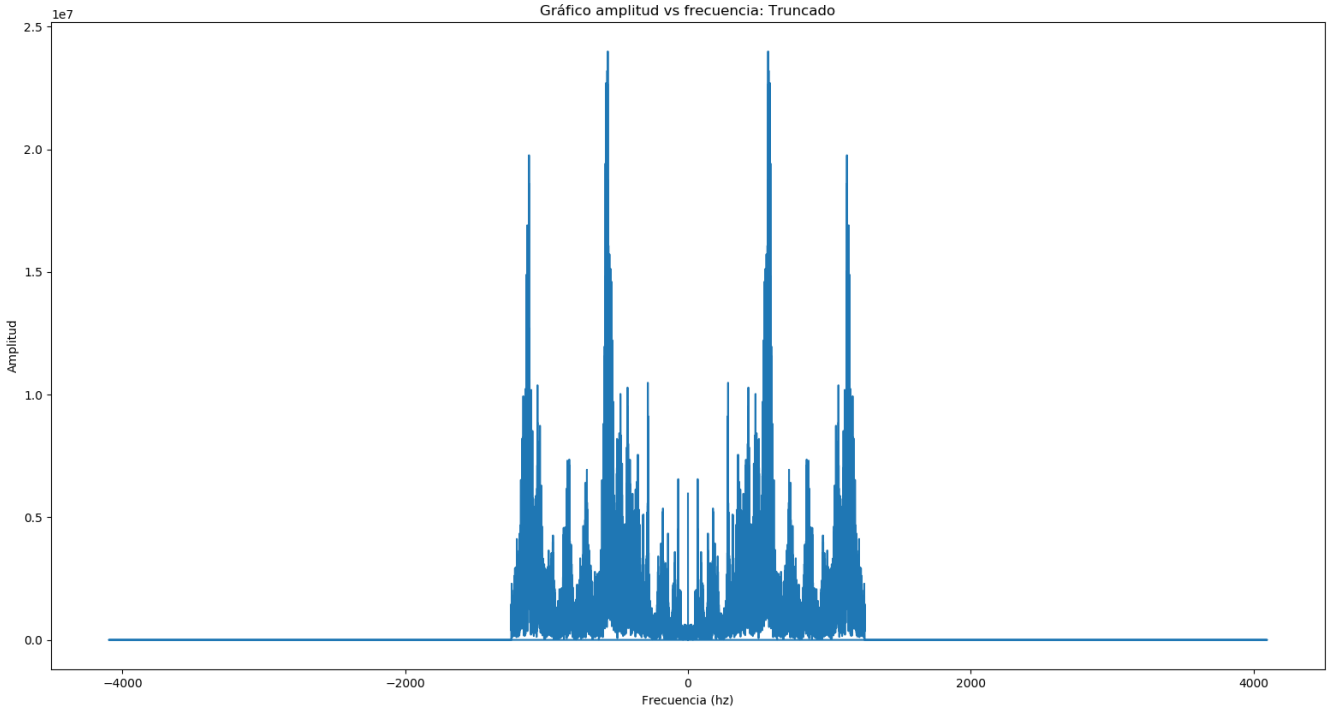

Figura 12: Segundo resultado gráfico etapa 4

4. Análisis de los resultados

1. Entre los resultados obtenidos, se aprecia que el tipo de dato del archivo creado por numpy es int16, este equivalente al formato WAV de 16-bit PCM, el cual posee variaciones de amplitudes que oscilan entre como mínimo -32768 y máximo 32768. Se puede ver que el rango de las amplitudes oscilan entre los valores 30,000 y $-30{,}000$, los cuales concuerdan con la jerarquía de rangos del formato del WAV.

   Es posible calcular la duración del fenómeno a partir de la división de la cantidad de datos de muestra (73113) y el rate obtenido (8192), dando como resultado 8,924 s, el cual se aproxima bastante a la duración real de 8,92 $[s]$. Se destaca que el rate es equivalente a la cantidad de muestras que se recopilaron en una unidad de tiempo, lo que quiere decir, que en 1 seg se recopilaron 8192 muestras del fenómeno.

2. Desde una perspectiva auditiva, se puede asegurar que el gráfico resultante de la etapa 2 (Figura 8) calza con las oscilaciones de entonación de la canción, por lo cual se presume que los datos recolectados están relacionados correctamente con los del audio.

   A partir de tales datos, es posible calcular la amplitud relativa del audio por medio de la formula: $dBrelativa = 20 * \log_{10}(amplitudMax/dtypeMax)$, en donde, al reemplazar los valores de la amplitud maxima del audio (26214) y dtype (32768), se obtiene que la amplitud relativa equivale a $-1{,}9380\ rdB$, valor que se encuentra entre el rango audible y meaningful en la tabla de dB relativos (Figura 13)

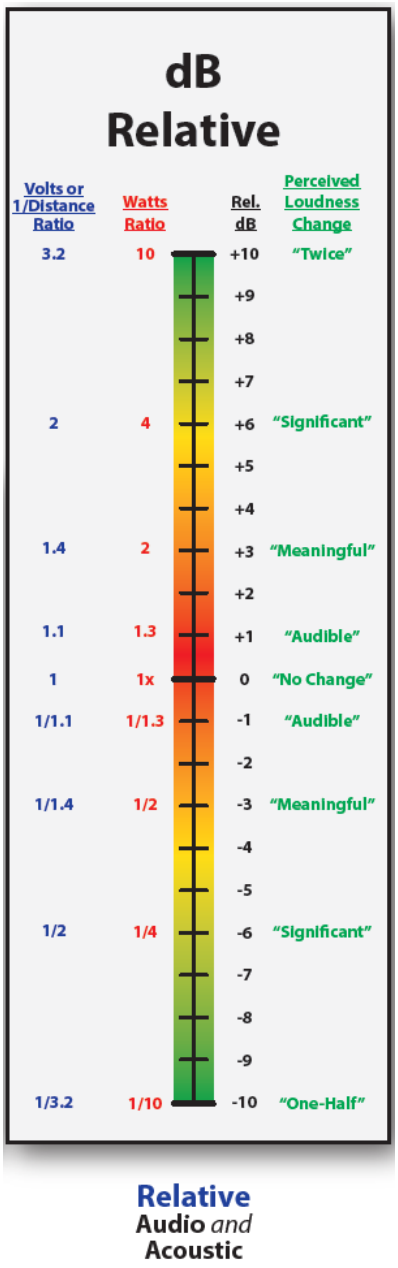

Figura 13: Tabla con rangos de amplitudes

3. En el gráfico con la transformada de fourier (Figura 9 de la etapa 3) se puede apreciar que las frecuencias mas relevantes oscilan entre los valores $200 - 1250\ Hz$, los cuales se posicionan en los tonos medios de la escala musical de las octavas $(250 - 1400\ Hz)$, rango que abarca a los armónicos de la mayoría de las fuentes sonoras. Por otro lado, las frecuencias con menor amplitud se posicionan en los tonos agudos de las octavas

$(2000 - 16000\ Hz)$.

Se aprecia que no existen amplitudes de frecuencias sobre los $4000Hz$, factor que tiene sentido en la escala de las octavas, ya que sobre ese valor correspondería a un sonido agudo y estridente. El gráfico de la transformada de fourier inversa (Figura 10 de la etapa 3) coincidió en todos los puntos con el de la señal de audio, lo cual incentivo a generar un nuevo archivo de audio con la inversa obtenida, el cual no presento diferencias auditivas en comparación con la original.

4. En el primer gráfico truncado (Figura 12 de la etapa 4), se puede apreciar que se mantuvieron las amplitudes relacionadas a las frecuencias mas significativas, pasando las restantes a amplitud 0, lo que trajo consigo una perdida de frecuencias agudas (superiores a 1250 Hz e inferiores a $-1250\ Hz$) que conllevo una no menor perdida de datos de la muestra.

   Aplicando la transformada de fourier inversa a lo obteniendo en el paso anterior, se obtuvo un espectro que al graficarlo (Figura 11 de la etapa 4) presento una similar distribución de datos que el del audio original (Figura 8 de la etapa 1). Esto se produjo debido a que se removieron las amplitudes de las frecuencias que tenían menos relevancia dentro del espectro.

   Al pasar a audio el espectro truncado, se aprecio la que era bastante idéntico al original, la única diferencia era la ausencia de ciertos todos agudos.

   Finalmente el audio generado mediante la inversa truncada, si bien mantiene un sonido similar al original, se puede notar que ya no es exactamente el mismo audio, debido a que se pierden muchos de los sonidos más agudos. Lo que tiene sentido, ya que como se mencionó, las frecuencias superiores a $1250[Hz]$ fueron llevadas a 0, siendo estas las que conllevan un sonido más agudo dentro de los rangos de frecuencias.

5. Conclusiones

Esta experiencia sirvió para profundizar el contenido de la transformada de fourier y su inversa, experimento que entrego variados gráficos que permitieron comprender a fondo su funcionamiento. En el caso de la transformada de fourier inversa, se obtuvo un gráfico y sonido completamente iguales al original, demostrando un claro cumplimiento de lo visto teóricamente. Para asimilar como el truncado afectó al espectro del fenómeno se decidió transformar a audio lo obtenido, en donde se pudo apreciar una ausencia de los tonos mas agudos encontrados originalmente en el audio, lo que confirmo que realmente se habían tornado a cero las frecuencias mas agudas (encontradas sobre los 1250 Hz).

El uso de audios dentro del experimento permitió un mejor manejo de los resultados de los procesos, esto por el hecho de ser un medio con bastante uso. Se presentaron problemas para saber qué representaba los sectores negativos al aplicar la transformada de fourier al audio, pero estos fueron resueltos al consultar lo ocurrido en laboratorio. Se resolvió, de igual forma, para determinar cuales tenían que ser los rangos mas relevantes dentro el espectro de frecuencias, estas determinadas por el grado de amplitud en comparación con las otras del fenómeno. El trabajo permitió analizar la transformada de fourier de una perspectiva mas practica, en cuanto a como funciona junto con las aplicaciones que tiene.

Por lo anterior mencionado, se puede decir que los objetivos que se plantearon al comienzo de este documento, fueron cubiertos a cabalidad, ya que se aplicó la teoría con respecto a señales y transformada de Fourier, pero además, se realizaron estos audios de prueba que permitieron de forma más natural, hacer la comparación entre las etapas del laboratorio.

Bibliografía

Brown, P. (2013). Quick db reference chart. [Online] https://www.prosoundtraining.com/2013/07/27/quick-db-reference-chart/.

Ecured (2011). Señales analógicas y digitales. [Online] https://www.ecured.cu/Se%C3%B1ales_anal%C3%B3gicas_y_digitales.

Elder, R. (2016). Loading wav files and showing frequency response. [Online] https://pythondsp.rob-elder.com/loading-wav-files-and-showing-frequency-response/.

Hernández, S. P. (2014). Series y transformadas de fourier. [Online] http://www.dmae.upct.es/~paredes/am_ti/apuntes/Tema%202.%20Series%20y%20transformadas%20de%20Fourier.pdf.

Maupe (2019). Señales analógicas. [Online] https://www.maupe.com/Empresa/que-son-las-senales-analogicas/.

Numpy (2019). Numpy v1.18.dev0 manual. [Online] https://docs.scipy.org/doc/scipy/reference/generated/scipy.io.wavfile.read.html.

Porto, J. P. (2014). Señal digital. [Online] https://definicion.de/senal-digital/.

SciPy (2019). Scipy manual. [Online] https://docs.scipy.org/doc/scipy/reference/generated/scipy.io.wavfile.read.html.

Usach (2019). Enunciado laboratorio 1. [Online] http://www.udesantiagovirtual.cl/moodle2/pluginfile.php?file=%2F257760%2Fmod_resource%2Fcontent%2F1%2FLaboratorio%201%20Redes.pdf.

Wikipedia (2017). Espectro audible. [Online] https://es.wikipedia.org/wiki/Espectro_audible.