Universidad de Buenos Aires
Facultad De Ingeniería
Año 2020 – 2.^{do} Cuatrimestre

Trabajo práctico.

86.05 Señales y Sistemas

Alumno	Padrón	Correo electrónico
Álvaro Scarramberg	103370	ascarramberg@fi.uba.ar

Índice

1. Introducción **1**

2. Objetivo **1**

3. Desarrollo **1**

3.1. Pre-procesamiento de la señal: 1
3.2. Análisis de ventanas: 3
3.3. Reducción de la tasa de muestreo: 4
3.4. Análisis del espectrograma: 8
3.5. Generación de la huella digital acústica: 11
3.6. Test del algoritmo: 13

4. Conclusiones **15**

1. Introducción

Para facilitar el reconocimiento de señales de audio se diseñaron una serie de métodos para la representación de estas señales mediante lo que se denomina una huella digital acústica. Estas tienen como objetivo representar en forma única y compacta a todas las señales dentro de una cierta base de datos.

Esto permite, por ejemplo, comparar la huella del segmento de una canción que se desea reconocer con todas las huellas de las canciones en una base de datos para determinar el nombre de la canción a la que pertenece el audio analizado.

En el presente escrito se detalla el proceso de creación y caracterización de un programa implementado en Matlab que permite generar un tipo de huella digital acústica para una señal de audio, basada en el análisis de la energía de las bandas de frecuencia en un espectrograma de la señal analizada.

2. Objetivo

En este trabajo se busca la incorporación de distintas de herramientas de diseño y análisis relacionadas al área de señales y sistemas con el fin de afianzar los temas vistos en la materia y poder además ponerlos en practica.

Se requiere crear un programa que pueda producir una representación fiel y compacta de una señal de audio en la forma de una huella digital acústica la cual será utilizada para la creación de una base de datos a partir del conjunto de archivos de audio proveído, para su posterior uso en el testeo del programa.

3. Desarrollo

3.1. Pre-procesamiento de la señal:

Antes que nada se debe aclarar que se asume como pre-condición que los archivos con los que va a trabajar el programa son audios en stereo grabados con una frecuencia de muestreo de $44{,}1kHz$.

En primer lugar, en el programa se cambia de stereo a mono al archivo de audio recibido ya que para los siguientes pasos del procesado se necesita disponer de una única señal. Para lograr esto simplemente se promedian ambos canales del audio. Utilizando el archivo proveído "Pink.ogg", se muestra como ejemplo un segmento de la señal resultante de realizar esta operación sobre dicho audio:

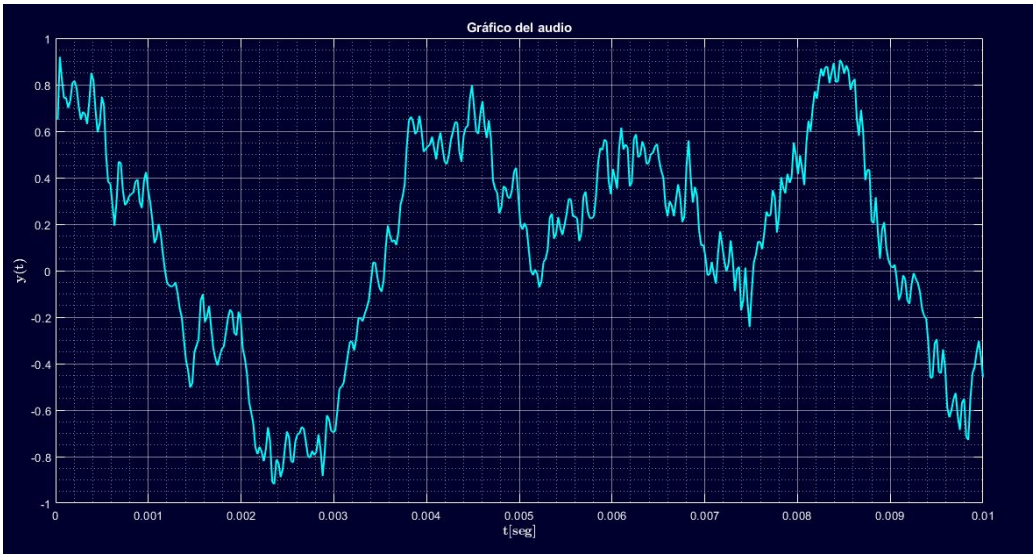

Figura 1: Gráfico resultante del promedio de los canales.

Antes de poder construir la huella correspondiente a la señal es necesario acondicionarla con el fin de reducir la cantidad de memoria a utilizar.

Se realizó un análisis de la potencia de la señal anterior en función de su frecuencia para obtener de forma cualitativa el rango de frecuencias donde se concentra la mayor parte de la energía de una señal de audio típica. Teniendo esta información se puede modificar la tasa de muestreo de las señales que se desea analizar para disminuir la cantidad de memoria necesaria para representarlas.

Para realizar dicho análisis se obtuvo el gráfico de la densidad espectral de potencia de la señal obtenida anteriormente:

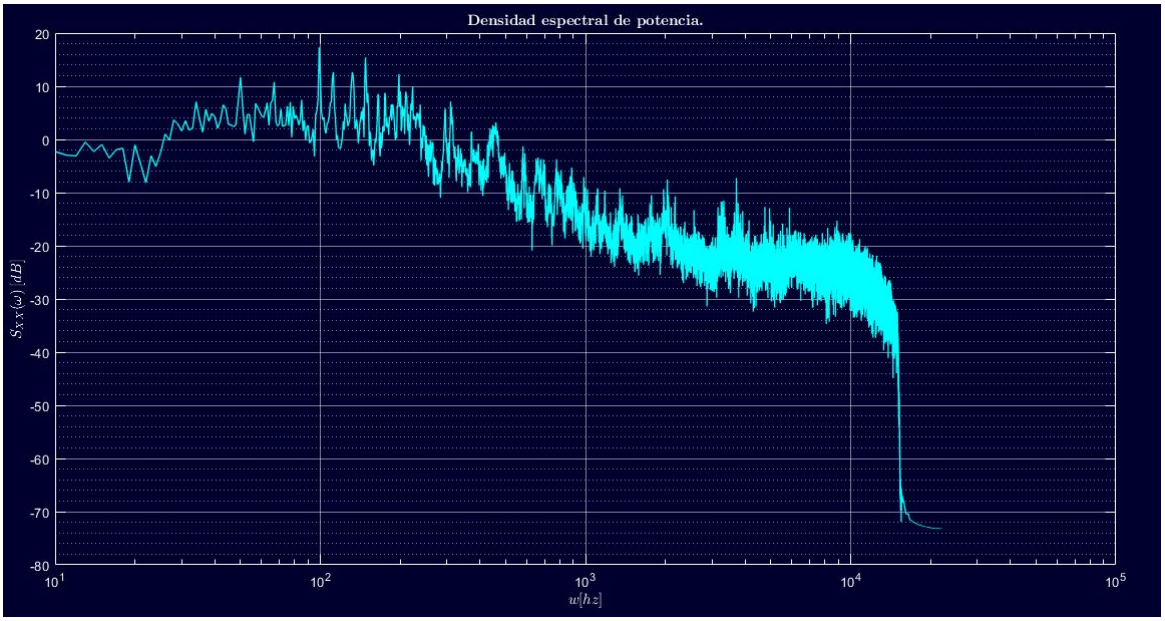

Figura 2: Densidad espectral de potencia de la señal analizada.

Se puede observar que la potencia esta concentrada en las frecuencias mas bajas, en particular para las frecuencias mayores a los $1kHz$ la atenuación es del orden de los $20dB$, por lo que como paso previo al procesado de la señal se decidió realizar un submuestreo de manera que se reduzca la frecuencia de muestreo a $5512{,}5Hz$.

3.2. Análisis de ventanas:

En las siguientes secciones resultó necesario el uso de ventanas adecuadas, ya sea para la implementación de un filtro pasa-bajos mediante el método del ventaneo o para realizar el espectrograma de la señal, por lo que se decidió realizar un análisis de las posibles ventanas a utilizar.

Entre ellas se consideraron tanto la ventana uniforme como la de Hamming. Para poder comparar estas ventanas de forma clara se graficó la superposición de la potencia de ambas en forma logarítmica con el eje de frecuencias normalizado:

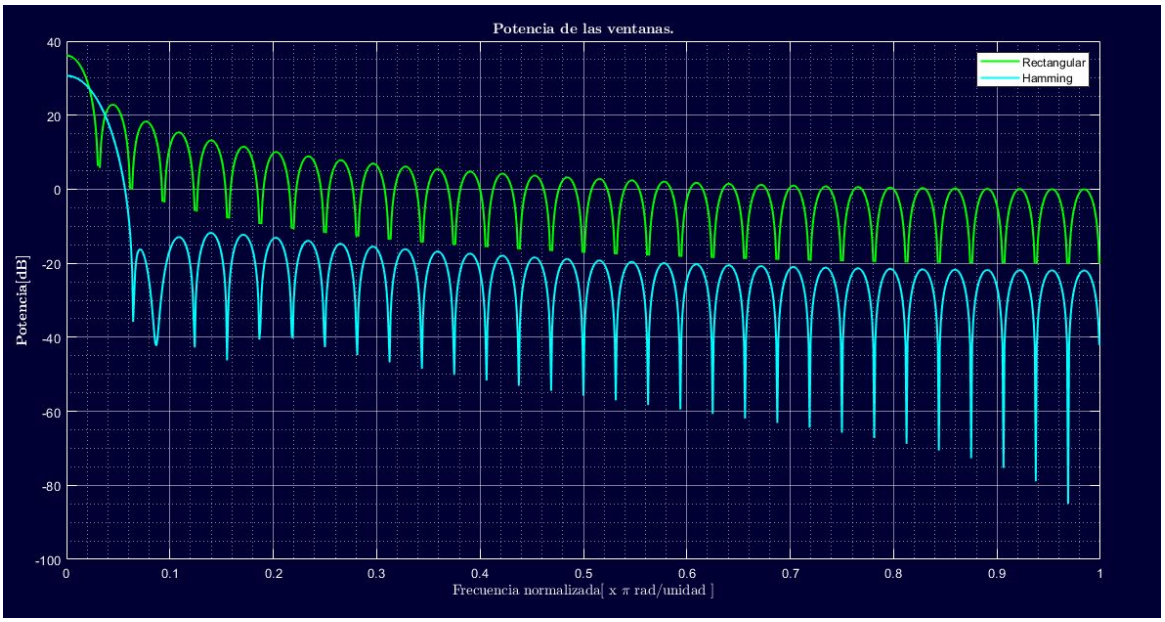

Figura 3: Comparación de la potencia de las ventanas en escala logarítmica.

Por definición la ventana uniforme es la que posee la menor concentración temporal por lo que era de esperar por el principio de incertidumbre que en el espacio frecuencial esta estuviese mas concentrada que la ventana de Hamming, lo cual se evidencia en el hecho de que el lóbulo principal correspondiente a la ventana uniforme es mucho mas angosto que el correspondiente a la ventana de Hamming.

Desde el gráfico, producido en Matlab, se pudo medir la resolución en frecuencia de la ventana uniforme que en este caso fue aproximadamente de $0{,}03\pi$ $rad/unidad$, en cambio la resolución de la ventana de Hamming resulto ser de $0{,}06\pi$ $rad/unidad$.

Se sabe que existe una relación inversa entre la resolución en frecuencia de la ventana y la atenuación de sus lóbulos secundarios, por lo que era de esperar que para la ventana uniforme la atenuación en sus lóbulos secundarios sea inferior a la que se tiene en el caso de la ventana de Hamming.

La selección de la ventana a utilizar depende del parámetro que se requiere priorizar. Si solo se necesita tener una buena resolución espectral la ventana uniforme es la mejor opción, sin embargo si se requiere tener una mayor atenuación en los lóbulos secundarios la ventana de Hamming es una mejor elección.

3.3. Reducción de la tasa de muestreo:

El esquema del sistema que se utilizó para modificar la frecuencia de muestreo es el que se muestra a continuación:

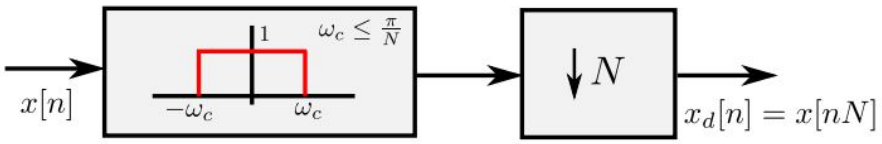

Figura 4: Diagrama de bloques del sistema utilizado.

Debido a que la frecuencia de muestreo original es 8 veces la frecuencia que se quiere obtener, el valor de N requerido para el sistema es igual a 8.

El filtro pasa-bajos es necesario para evitar el aliasing que puede obtenerse en el proceso de decimación de la señal. Su frecuencia de corte debe ser igual a:

$$\omega_c = \frac{\pi}{N} \Longrightarrow f_c = \frac{\omega_c \cdot Fs}{2\pi} = \frac{Fs}{2 \cdot N} = 2756{,}25Hz$$

Para implementar el filtro en cuestión se utilizó el método del ventaneo. Se lo implementó utilizando una ventana de Kaiser ya que esta permite una mayor flexibilidad en la selección de los parámetros del filtro resultante. Para asistir en la selección de las características de la ventana se utilizó la herramienta 'filterDesigner', llegando finalmente a la configuración que se decidió utilizar:

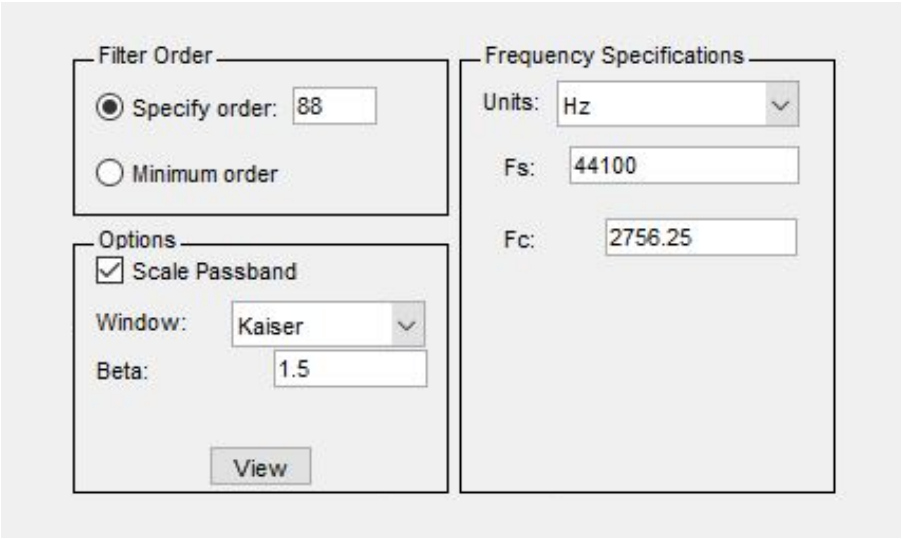

Figura 5: Parámetros seleccionados para el filtro diseñado.

El valor de beta de la ventana modula el ancho de la zona de transición entre la banda de pase y atenuación así como la atenuación presente en los lóbulos secundarios. Entre los aspectos anteriormente mencionados de la respuesta en frecuencia del filtro se tiene una relación inversa, por lo que se eligió el valor de beta de manera que se tenga un punto medio entre un ancho de banda de transición angosto y una atenuación adecuada en los lóbulos secundarios.

En cuanto al orden del filtro se seleccionó el valor máximo que garantiza que el tiempo de retardo sea menor a $1ms$. Como el retardo de grupo en muestras es igual a la mitad del orden del filtro se puede ver que se debe cumplir la siguiente condición:

$$\tau_r = \alpha/Fs \Longrightarrow \alpha = \tau_r \cdot Fs = \frac{Orden}{2} \Longrightarrow \tau_r = \frac{Orden}{2 \cdot Fs} < 1ms$$

$$Orden < 2 \cdot Fs \cdot 1ms = 88{,}2 \Longrightarrow max(Orden) = 88$$

Se utilizó la función 'fvtool' para generar los gráficos de todas las características relevantes del filtro, obteniéndose los siguientes resultados:

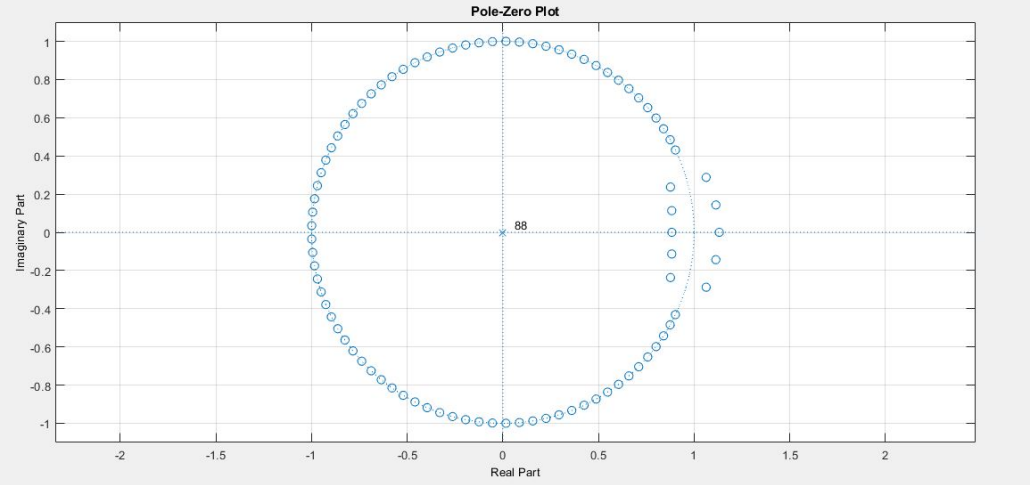

Figura 6: Diagrama de polos y ceros del filtro.

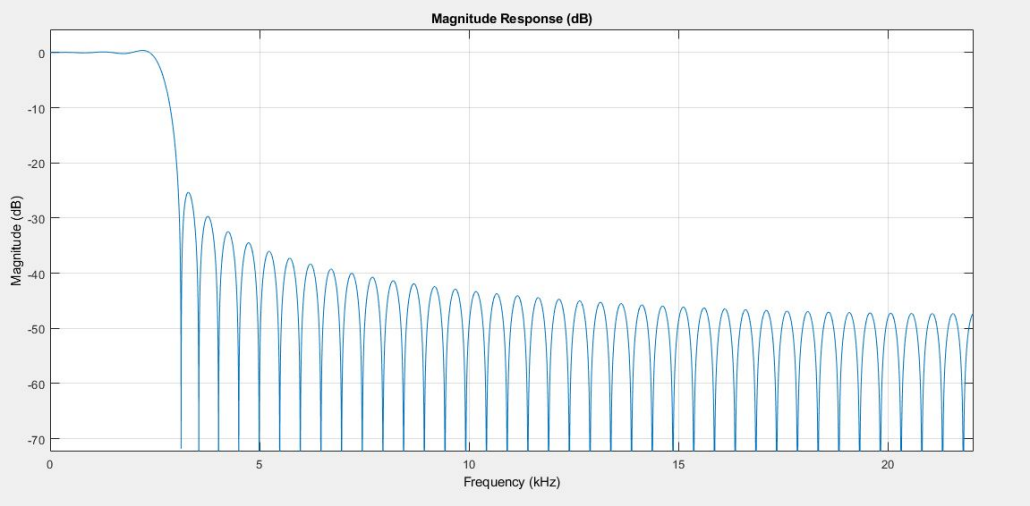

Figura 7: Magnitud de la respuesta en frecuencia del filtro.

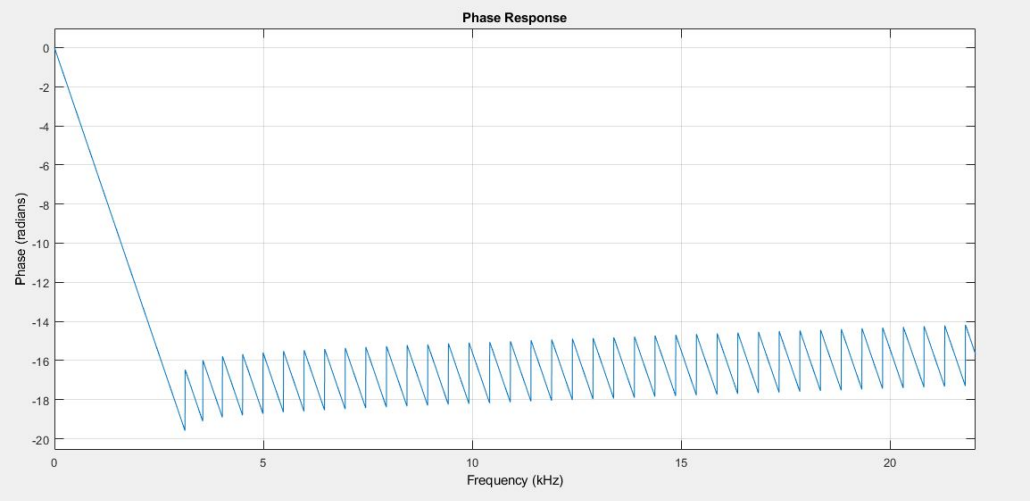

Figura 8: Fase de la respuesta en frecuencia del filtro.

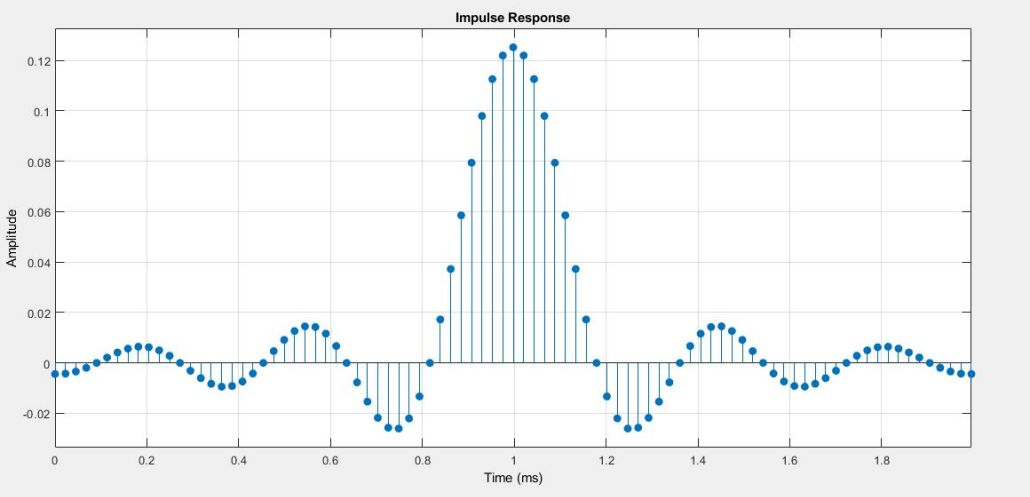

Figura 9: Respuesta al impulso del filtro.

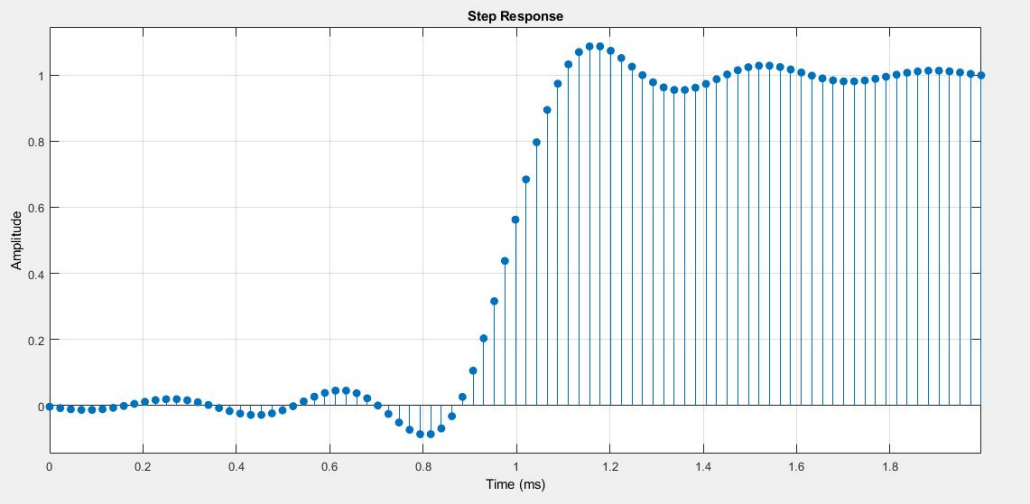

Figura 10: Respuesta al escalón del filtro.

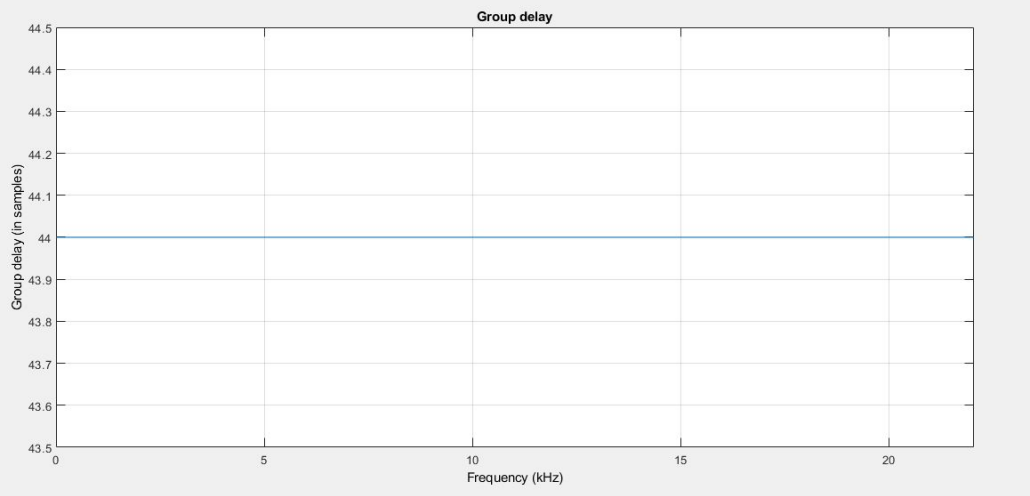

Figura 11: Retardo de grupo del filtro.

Los resultados conseguidos son los esperados para la ventana y parámetros seleccionados, y se ve que el filtro cumple en forma satisfactoria las condiciones deseadas para su funcionamiento.

Para comprobar el correcto funcionamiento del filtro se obtuvieron los espectrogramas correspondientes a la señal original y la filtrada. Se los generó utilizando una ventana de Hamming de largo de 1024 muestras, obteniéndose los siguientes resultados:

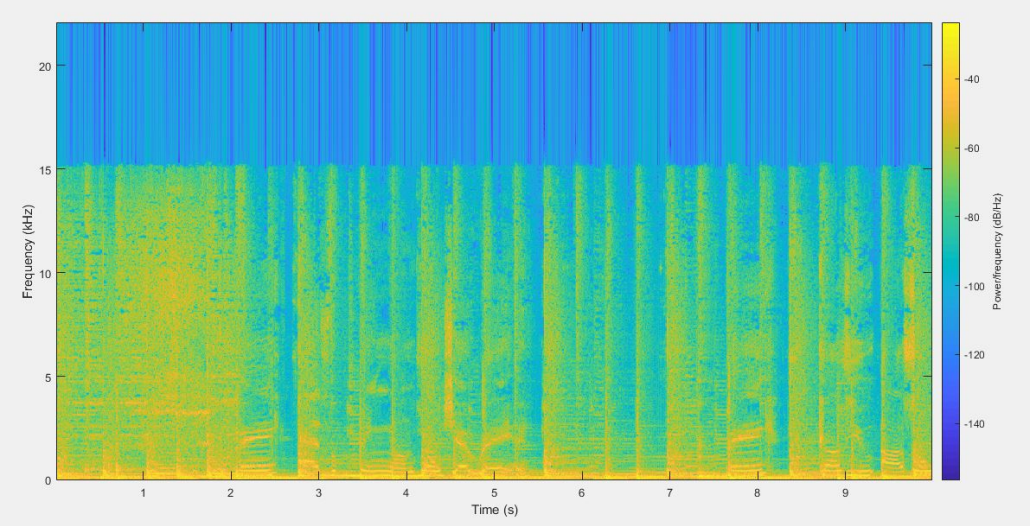

Figura 12: Espectrograma de la señal original.

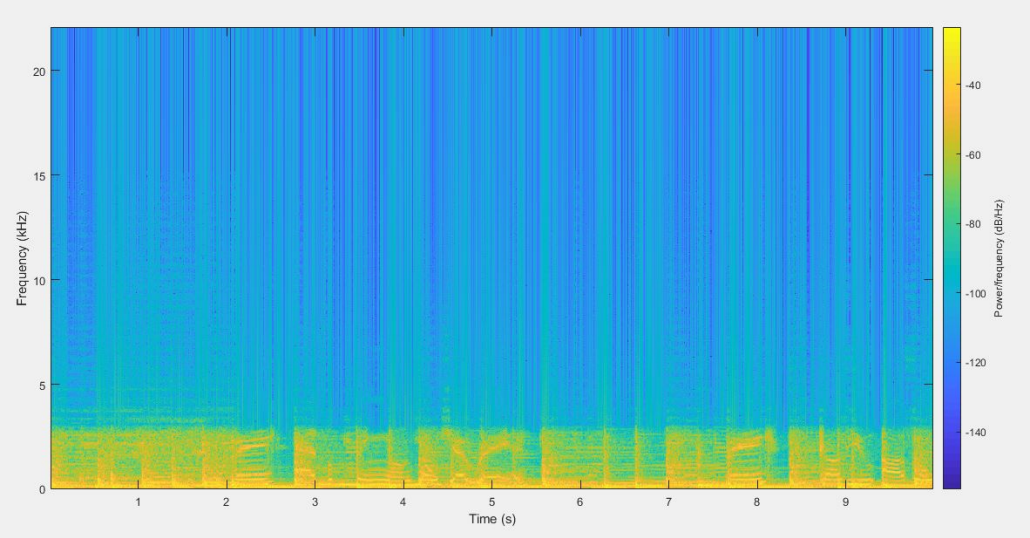

Figura 13: Espectrograma de la señal filtrada.

De la comparación de ambos se ve claramente que las frecuencias posteriores a la frecuencia de corte seleccionada se eliminaron manteniendo aparentemente intacta la banda de frecuencia de interés para la señal, lo cual indica que el filtro diseñado funciona correctamente.

Finalmente se utilizó el sistema construido para submuestrar la señal analizada y poder avanzar con la creación de su huella digital acústica.

3.4. Análisis del espectrograma:

En el resto del procesado de la señal se va a estar trabajando con su espectrograma por lo que es fundamental conocer que parámetros se deben seleccionar para obtener una representación de la señal que permita identificarla en forma casi unívoca minimizando el espacio en memoria de dicha representación.

Con este fin en mente se realizó el análisis del espectrograma de la señal utilizada en las secciones anteriores para distintos valores de largo de ventana en el caso de utilizar la ventana uniforme o la de Hamming, seleccionando por ahora un solapamiento nulo entre ventanas. Se realizaron los espectrogramas de la señal para ambas ventanas para los largos de ventana:

$$L1 = 128 \qquad L2 = 512 \qquad L3 = 2048$$

Sobre una región del espectrograma se realizaron gráficos para las distintas configuraciones con el fin de dar una idea de la resolución en frecuencia del espectrograma en cada caso. Los resultados obtenidos fueron los siguientes:

Espectrogramas correspondientes a la ventana uniforme:

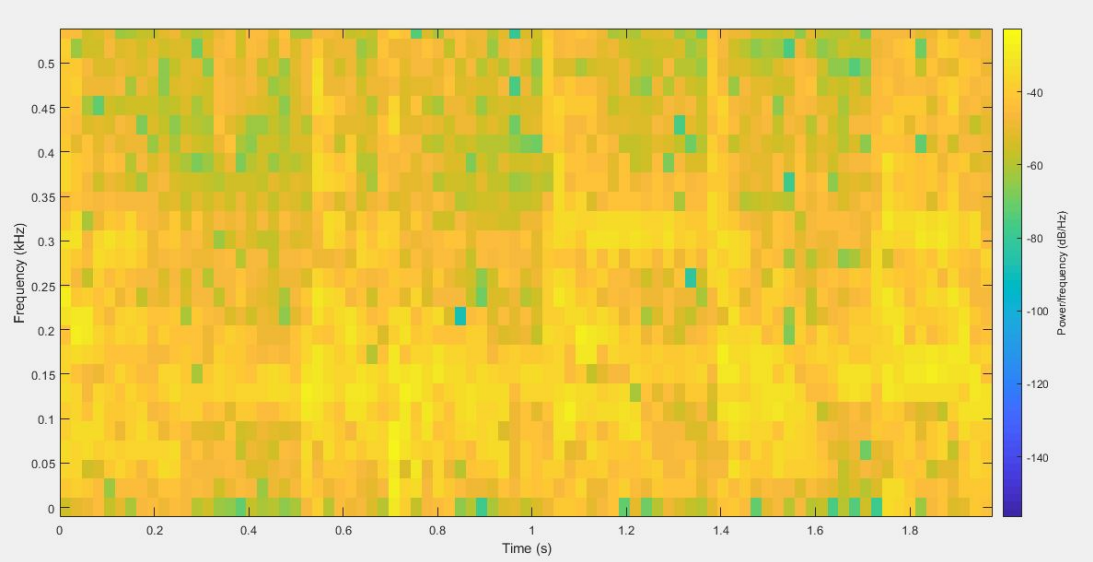

Figura 14: Espectrograma dado por una ventana uniforme de largo 128.

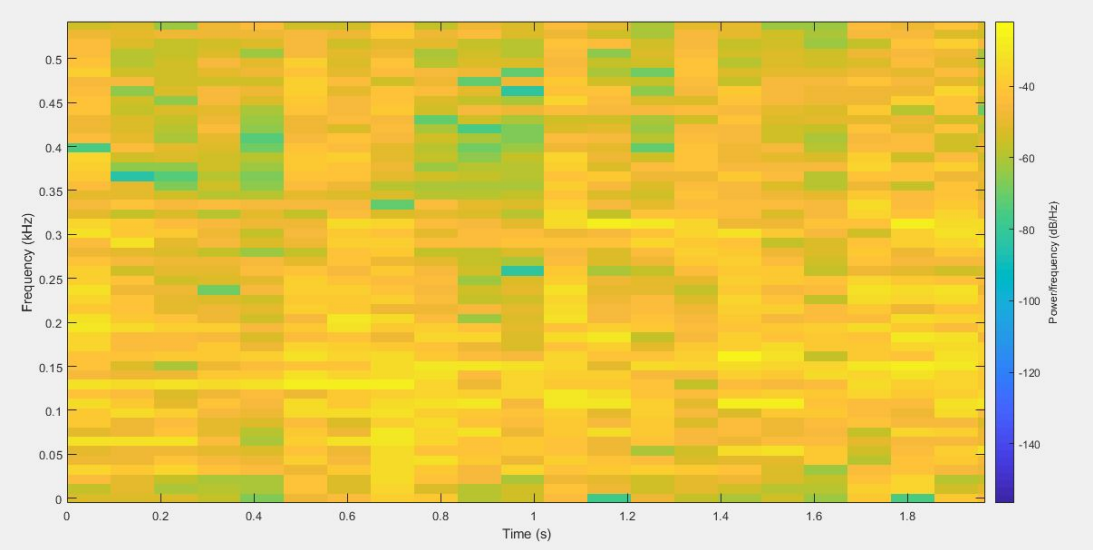

Figura 15: Espectrograma dado por una ventana uniforme de largo 512.

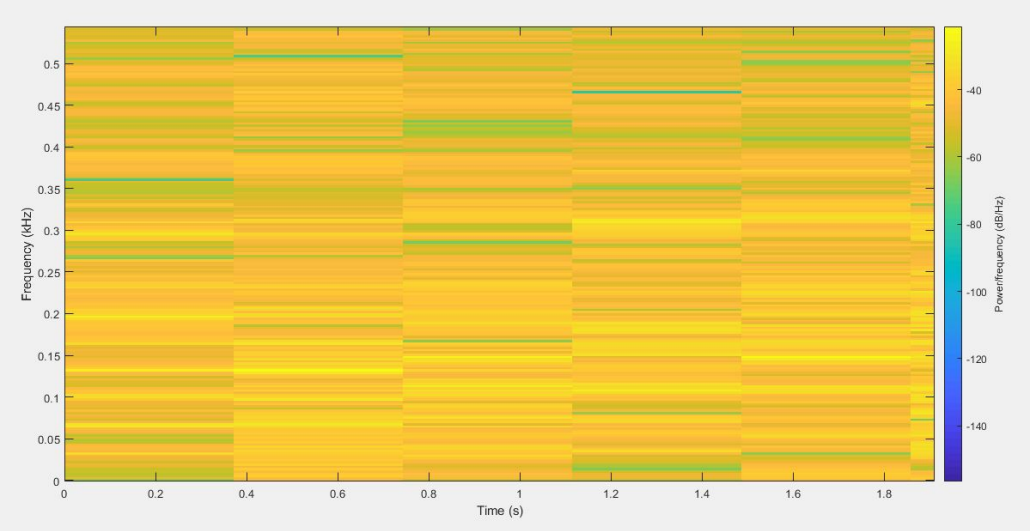

Figura 16: Espectrograma dado por una ventana uniforme de largo 2048.

Espectrogramas correspondientes a la ventana de Hamming:

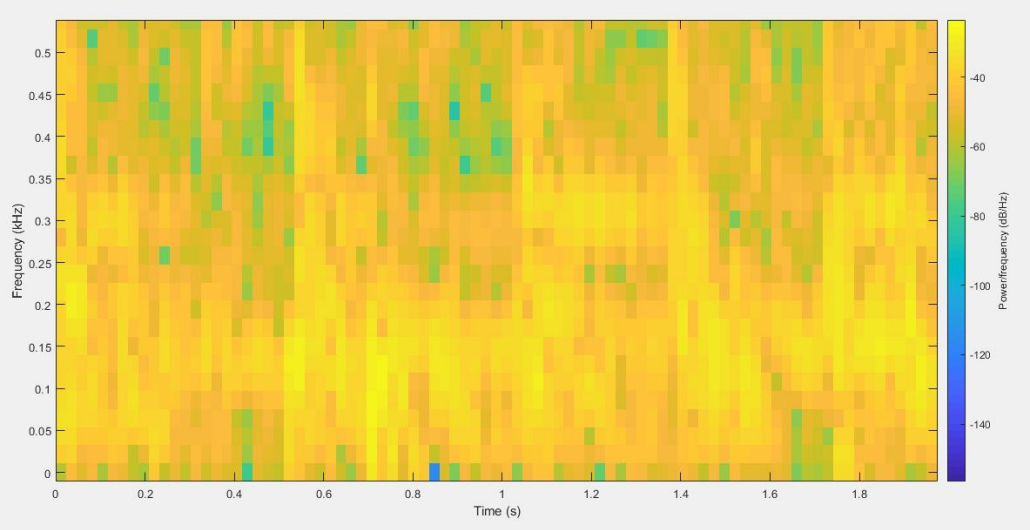

Figura 17: Espectrograma dado por una ventana de Hamming de largo 128.

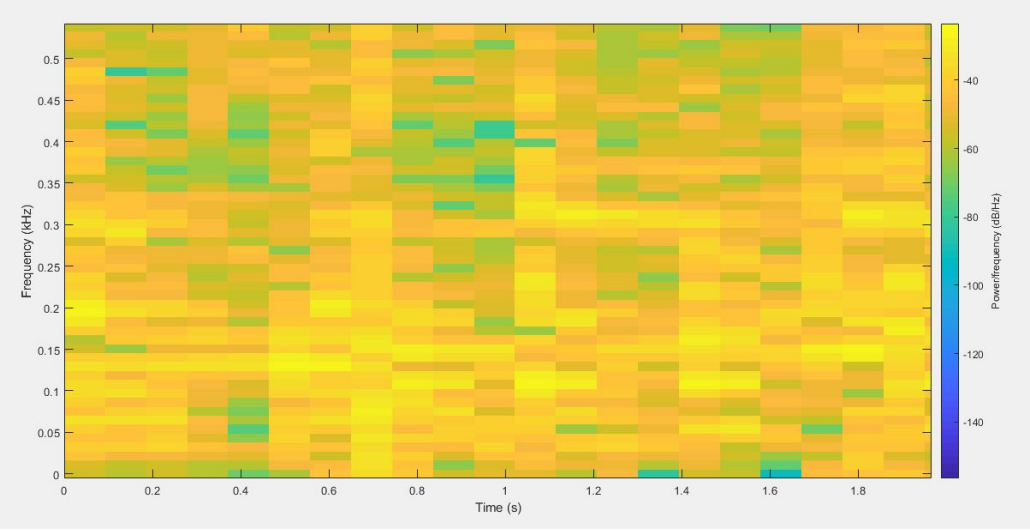

Figura 18: Espectrograma dado por una ventana de Hamming de largo 512.

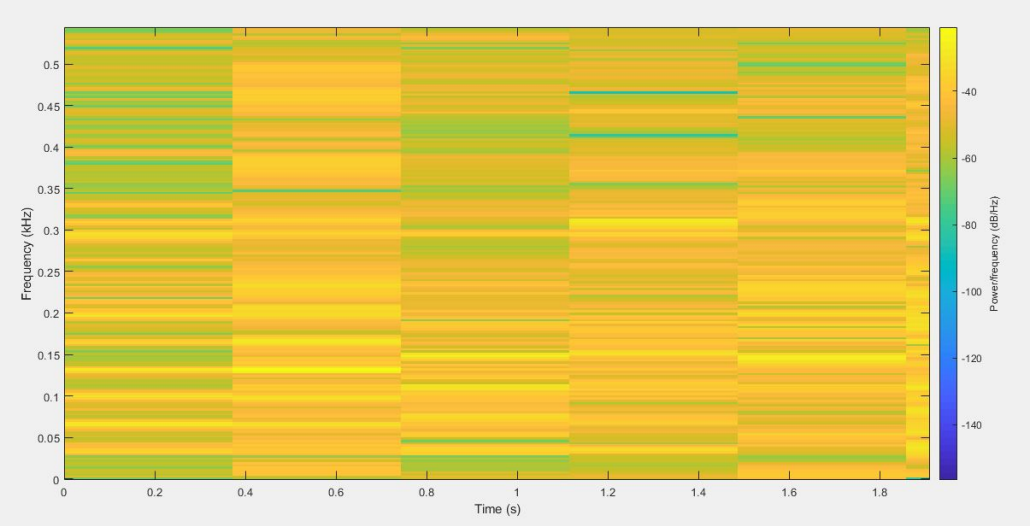

Figura 19: Espectrograma dado por una ventana de Hamming de largo 2048.

Se nota claramente que para ventanas de mayor largo la resolución en frecuencia del espectrograma es mejor, lo cual era esperable ya que en ese caso el número de puntos de las fft realizadas aumenta. Es evidente entonces que si lo que se necesita es generar una representación de la señal que permita diferenciarla de cualquier otra, es favorable tener un tamaño largo de ventana.

Además, al aumentar el tamaño de la ventana se necesitan menos ventaneos por espectrograma lo cual mas adelante se va traducir en un menor consumo de memoria.

Sin embargo, no es conveniente excederse demasiado con el tamaño de la ventana ya que esto implica una pérdida en resolución temporal.

Teniendo esto en cuenta, se decidió utilizar una ventana de Hamming con un largo de 2048 muestras para realizar los espectrogramas en el resto del programa. Para el largo seleccionado la ventana va a tener una duración temporal de 0,37s aproximadamente, si bien es un tamaño considerablemente grande, este va a permitir tener un porcentaje de solapamiento muy elevado para las condiciones en las que se necesita trabajar con el programa.

3.5. Generación de la huella digital acústica:

El método que se utiliza para representar las señales consiste en la caracterización de las mismas mediante la relación entre la energía de las bandas de frecuencia correspondientes a dos ventanas consecutivas en el espectrograma de la señal analizada.

Estas bandas de frecuencia son seleccionadas con un espaciado logarítmico en parte debido al carácter logarítmico que tiene la forma en la que percibe el sonido una persona y por otro lado debido a que la mayor parte de la energía de la señal se concentra en las bajas frecuencias por lo que estas son mas útiles para describirla y por lo tanto deben tener un mayor peso en su representación. Se crearon las bandas de frecuencia de tal forma que la frecuencia mínima de la primera banda sea $300Hz$ y que la frecuencia máxima de la ultima banda sea $2kHz$. Además se las construyó de manera que en total se tengan 21 bandas distintas.

El primer paso para obtener la huella es generar una matriz de la energía de las correspondientes bandas de frecuencia de cada ventana del espectrograma. A esta matriz se la llama $E(m,n)$ y va a tener entonces 21 filas(una por cada banda de frecuencia) y tantas columnas como ventanas se utilicen en el espectrograma.

Una vez obtenida la matriz E se puede generar finalmente la huella digital acústica de la señal la cual es una matriz H cuya expresión es la que se muestra a continuación:

$$H(m,n) = \begin{cases} 1, & si \;\; E(m+1,n) - E(m,n) > E(m+1,n-1) - E(m,n-1) \\ 0, & otro \;\; caso \end{cases}$$

De la expresión de H se ve que la matriz tiene una fila y una columna menos que la matriz E.

Teniendo todo este desarrollo solo falta determinar un parámetro para poder terminar el programa y ese es el solapamiento de ventanas utilizado al generar los espectrogramas. Se utilizó como criterio que el solapamiento sea tal que la cantidad de columnas de la matriz H por segundo de la señal tenga un valor próximo a 25, de manera que, debido a que H tiene 20 filas, el espacio necesaria para almacenarla sea de 500 bits por cada segundo de duración de la señal representada.

Para calcular el valor de solapamiento que cumple esto se debe tener en cuenta que el largo efectivo ocupado por cada ventana, osea las muestras que no son ocupadas por ninguna ventana anterior a ella, tiene un valor aproximado de:

$$L_{ef} = L_{Ventana} - L_{Overlap}$$

Y se requiere entonces que:

$$25 \cdot L_{ef} = 5512{,}5Hz \cdot 1s$$

Por lo que despejando al solapamiento se ve que debe valer:

$$25 \cdot L_{ef} = 5512{,}5 \Rightarrow L_{ef} = L_{Ventana} - L_{Overlap} = \frac{5512{,}5}{25}$$

$$L_{Overlap} = L_{ventana} - \frac{5512{,}5}{25} = 1827{,}5$$

Redondeando el valor obtenido se ve que para cumplir lo pedido de forma aproximada, la cantidad de muestras solapadas entre dos ventanas consecutivas del espectrograma deben ser 1828. Bajo estas condiciones el porcentaje de solapamiento es próximo al 90 %. Se mostrará en la sección siguiente que esto permite tener una tasa de aciertos muy elevada cuando se realicen las pruebas del algoritmo de reconocimiento.

Finalmente, se obtuvo la huella digital correspondiente a la señal de audio utilizada en las secciones anteriores luego de ser procesada. La imagen resultante es la siguiente:

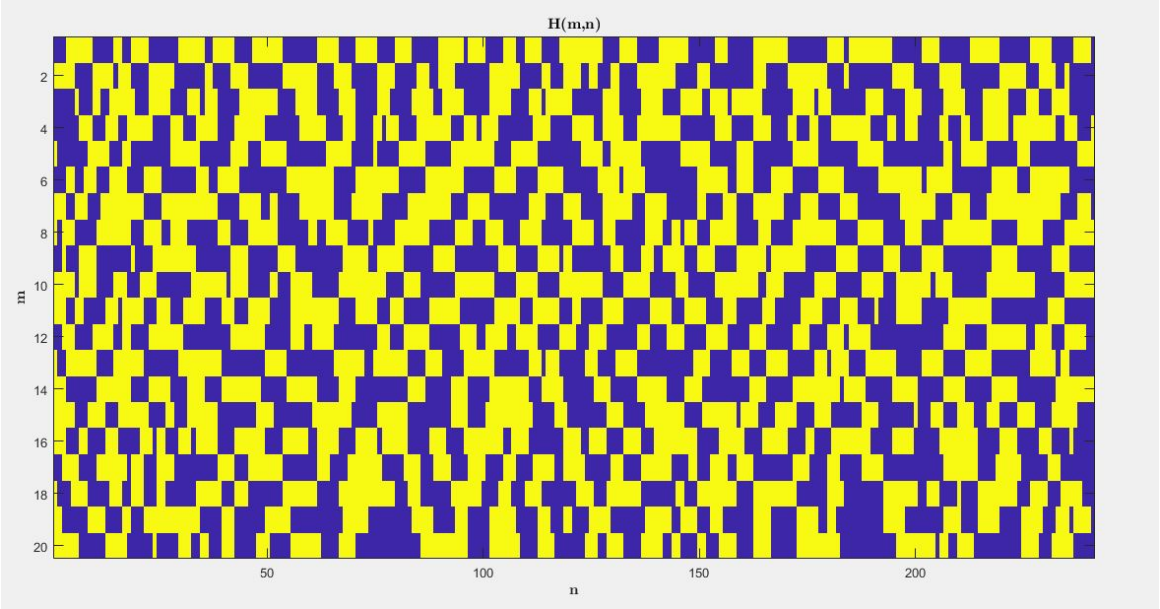

Figura 20: Huella digital acústica de la señal analizada.

Los cuadros amarillos en la imagen representan unos en la matriz y los azules representan a los ceros. Siendo que el audio utilizado tiene una duración de 10 segundos se ve que se cumple la condición de que se tengan aproximadamente 25 columnas por segundo.

3.6. Test del algoritmo:

Para comprobar la efectividad del método de representación de las señales de audio mediante el proceso descrito en las secciones anteriores se creo al script de Matlab generar_huella, que en conjunto con los scripts proveídos generar_DB, guardar_huella y query_DB permiten realizar un test de reconocimiento de señales de audio utilizando sus huellas digitales acústicas, generadas con las especificaciones relatadas anteriormente.

La función generar_DB() permite, haciendo uso de generar_huella y guardar_huella, crear una base de datos que contiene las huellas correspondientes a un conjunto de archivos de audio seleccionados. Una vez se tiene dicha base de datos se le puede entregar la huella de un segmento de audio que se quiera identificar a la función query_DB la cual va a devolver el nombre del archivo que tiene la mayor probabilidad de contener a dicho segmento. Utilizando un conjunto de 40 canciones distintas se generó dicha base de datos.

Teniendo todo esto en cuenta se creo un script para poder leer en forma iterativa segmentos aleatorios de duración especificada de los audios que conforman a la base de datos con el fin de utilizar query_DB para obtener el porcentaje de aciertos de dicha función para un cierto número de ensayos.

Para las especificaciones estipuladas para el programa, ya sean las características del filtro utilizado o las de los espectrogramas realizados, el resultado de la tasa de aciertos en 50 intentos que devolvió el test fue:

Tiempo del segmento	Tasa de aciertos en 50 intentos
20s	100 %
10s	100 %
5s	100 %

Tabla 1: Tasa de aciertos utilizando segmentos de la base de datos.

En estos ensayos no se pudo observar ninguna falla para el programa lo cual evidencia la eficacia que tiene esta configuración en el reconocimiento de las señales.

Con el fin de mostrar la importancia de tener un alto porcentaje de solapamiento, se realizó el mismo test que antes pero modificando el solapamiento a 50 %, obteniéndose los siguientes resultados:

Tiempo del segmento	Tasa de aciertos en 50 intentos
20s	76 %
10s	68 %
5s	56 %

Tabla 2: Tasa de aciertos para un solapamiento del 50 %.

Se observo una caída muy importante en la tasa de aciertos simplemente por haber disminuido este parámetro lo cual muestra la importancia de tener un porcentaje de solapamiento muy alto en los espectrogramas para el efectivo funcionamiento del programa.

Se desea realizar el mismo test pero ahora introduciendo ruido en los segmentos de audio. Para ello se hizo uso de la función de Matlab 'randn' que devuelve muestras correspondientes a una variable aleatoria de distribución normal estándar. Para este test se requiere poder seleccionar valores de SNR específicos, por lo que se debe determinar que operación se debe aplicar a las muestras de manera que su potencia media cumpla la siguiente condición:

$$SNR = 10 \cdot log_{10}\left(\frac{P_X}{P_N}\right)$$

Donde P_X es la potencia media de la señal y P_N es la potencia media del ruido. Teniendo en cuenta que tanto la media del ruido como la de la señal son al menos aproximadamente nulas se puede demostrar que sus potencias medias son iguales a sus varianzas:

$$Var(X) = E[X^2] - E[X]^2 \approx E[X^2] = P_X$$

$$Var(N) = E[N^2] - E[N]^2 = E[N^2] = P_N$$

Esto resulta bastante útil porque la varianza de la señal es fácil de obtener utilizando la función 'var' y además si se requiere generar a partir de 'randn' las muestras de una variable normal de varianza P_N, por propiedad de la varianza simplemente se debe multiplicar a esta función por la raíz cuadrada de P_N.

Finalmente, solo resta despejar a P_N en función de parámetros conocidos para determinar su valor y obtener muestras de ruido que cumplen la condición requerida:

$$P_N = \frac{PX}{10^{SNR/10}} = \frac{var(X)}{10^{SNR/10}} \Longrightarrow N = randn \cdot \sqrt{P_N} \sim N(0, P_N)$$

Teniendo esta información se pueden realizar las pruebas bajo las condiciones deseadas simplemente sumando las muestras aleatorias a los valores de la señal que se requiere reconocer. Los nuevos resultados en 50 intentos para la tasa de aciertos para las distintas combinaciones de ruido y largos de segmento seleccionados fueron los que se muestran en la siguiente tabla:

Duración del segmento	SNR		
	20dB	10dB	0dB
20s	100 %	100 %	100 %
10s	100 %	100 %	96 %
5s	100 %	100 %	92 %

Tabla 3: Tasa de aciertos incorporando ruido a la señal.

La función 'audiorecorder' de Matlab permite grabar audio utilizando un micrófono disponible, con una frecuencia de muestreo seleccionada. Se utilizó dicha función para grabar segmentos de audio de 10 segundos de duración de algunas de las canciones pertenecientes a la base de datos, las cuales fueron reproducidas con un dispositivo externo a la computadora que corría al programa.

Una vez se consiguieron dichas grabaciones se produjeron sus huellas digitales para comprobar si el programa era capaz de reconocerlas. En un ambiente con ruido externo moderado, se reprodujeron 7 canciones distintas y se consiguieron reconocer con un solo intento 6 canciones en total, lo cual indica que el sistema funciona satisfactoriamente también en estas condiciones.

4. Conclusiones

Los resultados obtenidos de las pruebas del programa indican altas tasas de aciertos en el reconocimiento de las canciones de la base de datos, incluso incorporando ruido en ellas, lo cual lleva a concluir que las huellas digitales acústicas generadas a través del programa creado pueden ser muy buenas representaciones para las señales de audio analizadas.

Se pudo incorporar a lo largo del trabajo una gran variedad de herramientas útiles en el análisis de señales que resultaron útiles para visualizar y comprender las dificultades enfrentadas en el transcurso de la creación del programa. Los problemas que se requirieron solucionar en este proceso permitieron poner en práctica y profundizar los conocimientos adquiridos a lo largo de la materia.