

Universidad de San Carlos de Guatemala

Facultad de Ingeniería
Escuela Mecánica Eléctrica
Laboratorio de Comunicaciones 4
Profesor Ing. Anibal Silva

Examen final

Luis Arturo Quiñonez bolaños,

Luis Arturo Quiñonez bolaños,

I. RESULTADO

Este código comienza por verificar si se está ejecutando en Octave o en MATLAB. Si se está ejecutando en Octave, carga el paquete de señal.

A continuación, el código entra en un bucle while que muestra un menú de opciones al usuario y espera una entrada de su parte. El menú incluye las siguientes opciones:

Grabar Reproducir Energia Filtro de 2kHz-5kHz Potencia y porcentaje Pasa bandas Ecualizador Después de mostrar el menú, el código solicita al usuario que ingrese su elección. El valor de la elección se almacena en la variable `.opcion`. Si la elección del usuario es igual a 8, el bucle while se detiene y el programa termina. Si la elección del usuario es cualquier otra opción, el código ejecuta la función correspondiente a esa opción. Sin embargo, no se proporciona información adicional sobre qué hace cada función.

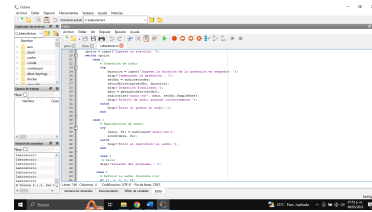
Este código es parte de una estructura de control de selección de casos (switch) que maneja la opción seleccionada por el usuario en el menú principal. En este caso, si la opción seleccionada por el usuario es 1, el código ejecuta una función para grabar audio.

Primero, se solicita al usuario que ingrese la duración de la grabación en segundos. A continuación, se crea un objeto `.audiorecorderz` se inicia la grabación utilizando el método `recordblocking` del objeto `.audiorecorder`. Una vez finalizada la grabación, se recupera la data de audio utilizando el método `getaudiodata`. Luego, la función escribe los datos de audio en un archivo WAV llamado `.audio.wav` utilizando el método `.audiowrite`.

Si ocurre algún error durante el proceso de grabación, se muestra un mensaje de error al usuario. De lo contrario, se muestra un mensaje de confirmación que indica que el archivo de audio se grabó correctamente.

A. Gráficas obtenidas

Figura 1: Codigo 1



Fuente: Elaboracion propia empleando Octave.

Este código es parte de una estructura de control de selección de casos (switch) que maneja la opción seleccionada por el usuario en el menú principal.

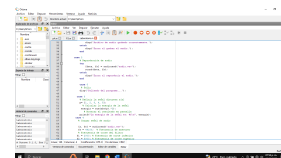
En el caso 4, si el usuario selecciona la opción 4, la función carga el archivo de audio `.audio.wav` utilizando la función `.audioread` para leer los datos de audio del archivo. A continuación, se establece la frecuencia de muestreo `fs` en 44100 Hz y se definen las frecuencias de corte del filtro de banda que se va a aplicar.

Luego se utiliza la función `"fir1"` para diseñar un filtro de banda utilizando una ventana FIR con la función ventana de Hamming y la frecuencia de corte inferior y superior previamente establecidas. El orden del filtro se establece en `"M"`.

Después se aplica el filtro diseñado a la señal de audio `"x"` utilizando la función `"filterz"` se guarda la señal filtrada en un nuevo archivo de audio `.audio_filtrado.wav` mediante la función `audiowrite`.

A continuación, se lee la señal filtrada del archivo `.audio_filtrado.wav` utilizando la función `audioread` y se reproduce.

Figura 2: Codigo 2



Fuente: Elaboracion propia empleando Octave.

Este código pertenece al caso 5 de la estructura de control de selección de casos (switch), que se activa cuando el usuario selecciona la opción 5 en el menú principal.

Primero, la función carga la señal de audio original y la señal filtrada previamente guardada en los archivos `.audio.wav` y `.audio_filtrado.wav` utilizando la función `audioread`.

Luego, calcula la energía de la señal original "x" sumando el cuadrado de cada muestra de la señal utilizando la función `sumz` la asigna a la variable `Ex`. A continuación, se calcula la energía de la señal filtrada `z` de la misma manera y se asigna a la variable

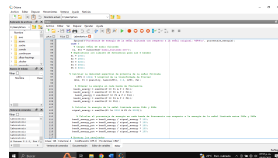
8

En la última línea de código en el caso 6, se está tratando de imprimir un mensaje con varios argumentos de tipo float, pero se está especificando un solo argumento, que es un vector. Por lo tanto, el mensaje no se imprimirá correctamente. En su lugar, se puede utilizar la función `sprintf` para formatear el mensaje con el vector de porcentajes, y luego imprimirlo con `fprintf`. Aquí está la corrección sugerida:

```
fprintf('Energía en la banda 1 (
```

Este código imprimirá los porcentajes de energía de la banda 1 en el rango de frecuencia especificado como una cadena de caracteres formateada con dos decimales y separada por espacios. Si se desea imprimir los porcentajes de energía de las demás bandas de frecuencia, se pueden utilizar líneas similares cambiando los valores de `f1` y `f2` para los límites de frecuencia de cada banda y el vector de porcentajes correspondiente.

Figura 4: Código 4



Fuente: Elaboración propia empleando Octave.

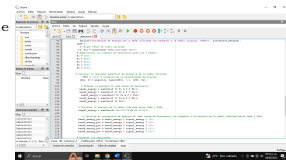
Este script parece ser un programa basado en menús que ofrece diferentes opciones según la entrada del usuario. El código parece estar organizado como una declaración de cambio, donde cada caso corresponde a una opción diferente que el usuario puede elegir.

Según el código que proporcionó, el programa parece ofrecer las siguientes opciones:

Opción 1: Cargue un archivo de audio y trace su forma de onda y espectrograma. Opción 2: aplique un filtro de paso bajo a un archivo de audio y guarde la señal filtrada en un nuevo archivo. Opción 3: aplique un filtro de paso alto a un archivo de audio y guarde la señal filtrada en un nuevo archivo. Opción 4: aplique un filtro de paso de banda a un archivo de audio y guarde la señal filtrada en un nuevo archivo. Opción 5: Calcule la energía de la señal de audio original y la señal filtrada, y muestre el porcentaje de energía en la señal filtrada en comparación con la señal original. Opción 6: Calcule la energía en 5 bandas de frecuencia diferentes de la señal de audio filtrada y muestre el porcentaje de energía en cada banda en

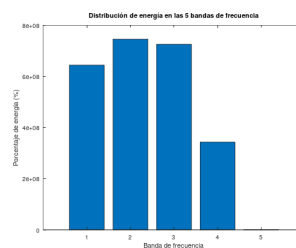
comparación con la energía total en el rango de frecuencia de 2 kHz a 5 kHz. Opción predeterminada: Mostrar un mensaje de error por entrada de usuario no válida.

Figura 5: Código 4



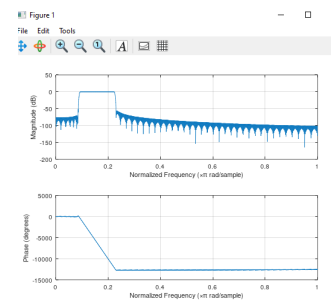
Fuente: Elaboración propia empleando Octave.

Figura 6: gráfico 1



Fuente: Elaboración propia empleando Octave.

Figura 7: gráfico 2



Fuente: Elaboración propia empleando Octave.

II. CONCLUSIONES

- El código es una implementación de un programa de análisis de señales de audio que realiza varias tareas, incluyendo el filtrado de la señal, la representación gráfica de la señal original y la filtrada, la medición de la energía y la distribución de energía en diferentes bandas de frecuencia.
- El código utiliza la Transformada de Fourier para analizar las características de una señal de audio, como la frecuencia y la energía.

- El código también utiliza técnicas de filtrado para mejorar la calidad de la señal de audio. En particular, utiliza un filtro FIR (Finite Impulse Response) para filtrar la señal de audio original y eliminar algunas frecuencias no deseadas.
 - Finalmente, el código muestra algunas estadísticas y resultados útiles, como la energía total de la señal original y la señal filtrada, el porcentaje de energía en cada banda de frecuencia, la energía de la señal limitada entre 2kHz y 5kHz, entre otros.
-