%---------------------------------------------------------

% Operaciones con señales digitales de audio

% Navil Pineda Rugerio

% Inteligencia Artificial

% 5to Semestre

%---------------------------------------------------------

clc

1. Ejecutar el comando load

load('handel.mat');

2. Abrir uno de los archivos.

disp(y);

3. Guardar en una variable con el mismo nombre

y = y;

5. Generar vector de tiempo

np = 1:length(y);

fprintf('%d', length(np));

6. Graficar cada una de las señales

np = y(1:round(length(y)/3));

stem(np,'b')

xlabel('Longitud de la señal (np)')

ylabel('Amplitud de la señal (y)')

title('Señal 1')

nd = y(round(length(y)/3):round(length(y)/3)\*2);

stem(nd,'b')

xlabel('Longitud de la señal (nd)')

ylabel('Amplitud de la señal (y)')

title('Señal 2')

nde = y(round(length(y)/3)\*2:round(length(y)/3)\*3);

stem(nde, 'b')

xlabel('Longitud de la señal (np)')

ylabel('Amplitud de la señal (y)')

title('Señal 3')

7. Reproducir la señal como sonido a una frecuencia de muestreo de 8000 hz.

fs\_new = 8000; % Frecuencia de muestreo

t\_original = (0:length(np) - 1) / Fs;

t\_new = (0:(length(np) - 1) \* (fs\_new / Fs)) / fs\_new;

% Interpolar la señal np para ajustarla a la nueva frecuencia de muestreo

np\_resampled = interp1(t\_original, np, t\_new);

% Reproducir sonido

%sound(np\_resampled, fs\_new);

stem(np\_resampled, 'b')

xlabel('np')

ylabel('y (resampleada)')

title('8000 Fs')

Para 4000 Fs.

fs\_new = 4000; % Frecuencia de muestreo

t\_original = (0:length(np) - 1) / Fs;

t\_new = (0:(length(np) - 1) \* (fs\_new / Fs)) / fs\_new;

np\_resampled1 = interp1(t\_original, np, t\_new);

%sound(np\_resampled1, 4000);

stem(np\_resampled1, 'k')

xlabel('np')

ylabel('y (resampleada)')

title('4000 Fs')

Para 5000 Fs.

fs\_new = 5000; % Frecuencia de muestreo

t\_original = (0:length(np) - 1) / Fs;

t\_new = (0:(length(np) - 1) \* (fs\_new / Fs)) / fs\_new;

np\_resampled2 = interp1(t\_original, np, t\_new);

%sound(np\_resampled2, 5000);

stem(np\_resampled2, 'k')

xlabel('np')

ylabel('y (resampleada)')

title('5000 Fs')

Para 8000 Fs.

fs\_new = 8000; % Frecuencia de muestreo

t\_original = (0:length(np) - 1) / Fs;

t\_new = (0:(length(np) - 1) \* (fs\_new / Fs)) / fs\_new;

np\_resampled3 = interp1(t\_original, np, t\_new);

%sound(np\_resampled3, 8000);

stem(np\_resampled3, 'k')

xlabel('np')

ylabel('y (resampleada)')

title('8000 Fs')

Para 20000 Fs.

fs\_new = 20000; % Frecuencia de muestreo

t\_original = (0:length(np) - 1) / Fs;

t\_new = (0:(length(np) - 1) \* (fs\_new / Fs)) / fs\_new;

np\_resampled4 = interp1(t\_original, np, t\_new);

%sound(np\_resampled4, 20000);

stem(np\_resampled4, 'k')

xlabel('np')

ylabel('y (resampleada)')

title('20000 Fs')

8. Explicación del cambio en la frecuencia de muestreo.

9. Obtener la suma de las cuatro señales.

Para ello se debe modificar la la longitud de las señales, pues en la suma todas las señales deben ser del mismo tamaño, por lo que se interpolan hasta la frecuencia más alta (20000 hz).

% Frecuencia de muestreo común (20,000 hz)

fs\_common = 20000;

t\_original = (0:length(np) - 1) / fs\_new;

t\_new = (0:(length(np) - 1) \* (fs\_common / fs\_new)) / fs\_common;

max\_length = max([length(np\_resampled1), length(np\_resampled2), length(np\_resampled3), length(np\_resampled4)]);

np\_resampled1 = [np\_resampled1, zeros(1, max\_length - length(np\_resampled1))];

np\_resampled2 = [np\_resampled2, zeros(1, max\_length - length(np\_resampled2))];

np\_resampled3 = [np\_resampled3, zeros(1, max\_length - length(np\_resampled3))];

np\_resampled4 = [np\_resampled4, zeros(1, max\_length - length(np\_resampled4))];

t\_original = (0:max\_length - 1) / fs\_new;

t\_new = (0:(max\_length - 1) \* (fs\_common / fs\_new)) / fs\_common;

np\_resampled1 = interp1(t\_original, np\_resampled1, t\_new);

np\_resampled2 = interp1(t\_original, np\_resampled2, t\_new);

np\_resampled3 = interp1(t\_original, np\_resampled3, t\_new);

np\_resampled4 = interp1(t\_original, np\_resampled4, t\_new);

% Suma

summed\_signal = np\_resampled1 + np\_resampled2 + np\_resampled3 + np\_resampled4;

10. Graficar la suma de señales y reproducirla como sonido.

%sound(summed\_signal, fs\_common);

stem(summed\_signal, 'g')

xlabel('np')

ylabel('y (sumada)')

title('Suma de señales')

11. Realizar las operaciones necesarias para concatenar las cuatro señales en el mismo orden dándoles un espaciamiento de 2000 muestras y guardarlas en la señal datosConcatenados. Graficar y reproducir.

space = zeros(1, 2000); % Espaciado

% Concatenar las señales con el espaciado

datosConcatenados = [np\_resampled1, space, np\_resampled2, space, np\_resampled3, space, np\_resampled4];

% Graficar y reproducir

%sound(datosConcatenados, 20000);

stem(datosConcatenados, 'r')

xlabel('Muestras concatenadas')

ylabel('Amplitud de la señal')

title('Señales Concatenadas')

12. Invertir la señal datosConcatenados y guardalos en invertida.

invertida = flip(datosConcatenados);

13. Graficar y reproducir la señal invertida.

sound(invertida, 20000);

stem(invertida, 'c')

xlabel('Muestras concatenadas')

ylabel('Amplitud de la señal')

title('Señal Invertida')