% Test code for song filtering in Lab3

%% Read the file

dir = ''; %directory of file

file = 'songOne\_original.wav';

song\_file = [dir file];

[y,Fs] = audioread(song\_file);

signal = y;

%% pre-processing and trimming

%-- This is only for making the files for lab from the original songs

%-- Do not run this on the songs included in the zip folder

% tstart = 1/Fs; %starting time

% istart = tstart\*Fs;

% tend = 10;

% iend = tend\*Fs;

% mono = sum(signal,2)/size(signal,2); %make mono channel

% signal = mono(istart:iend); %cut to time given

% n=2; %sets down sample

% Fs = Fs/n;

% signal = downsample(signal,n);

% audiowrite('songTwo\_original.wav',signal,Fs);

%% filter signal

% filtered = filter1('lp',signal,'fc',200,'fs',Fs); %low pass with fc = 200Hz;

filtered = filter1('hp',signal,'fc',2000,'fs',Fs); %high pass with fc = 2000Hz;

t = (1:1:size(signal,1)).\*(1/Fs);

% figure(1) %Plot original and filtered signals

% plot(t,signal)

% hold on

% plot(t,filtered,'r')

% hold off

%% Play the songs

% sound(signal,Fs)

% sound(filtered,Fs)

%%Use 'clear sound' in the command window to stop playing

%% Plot Freq Response

% T = 1/Fs; % Sampling period

% L = size(signal,1); % Length of signal

% t = (0:L-1)\*T; % Time vector

%

% Y = fft(signal);

% P2 = abs(Y/L);

% P1 = P2(1:L/2+1);

% P1(2:end-1) = 2\*P1(2:end-1);

% f = Fs\*(0:(L/2))/L;

% figure(2)

% plot(f,P1)

% title('Single-Sided Amplitude Spectrum of X(t)')

% xlabel('f (Hz)')

% ylabel('|P1(f)|')

% axis([0 5e3 0 max(P1)])

%

%

% Y = fft(filtered);

% P2 = abs(Y/L);

% P1 = P2(1:L/2+1);

% P1(2:end-1) = 2\*P1(2:end-1);

% f = Fs\*(0:(L/2))/L;

%

% hold on

% plot(f,P1,'r')