```
% 19ucc023
% Mohit Akhouri
% Experiment 9 - Observation 3

% This code will read an audio signal 'inputwithhissmadelaptop.wav'
 and
% sample it using 32 Khz. We will take the first 8 samples of this
 audio
% signal and try to locate the hissing frequencies ( approximately at
 3rd
% and 7th samples ).

% Next we will design a NOTCH FILTER and convolve it with distorted
 signal.
% What we finally get is a smoothened version of the audio signal with
% hissing removed and some distortion still left.

clc;
clear all;
close all;

[x,fs] = audioread('inputwithhissmadelaptop.wav'); % reading an audio
 file

samples_x = x(1:8); % Taking first 8 samples of audio signal x[n]
spectrum_sample = abs(fftshift(fft(samples_x))); % Finding the
 frequency spectrum of the first 8 samples for locating hissing
 frequencies

% Plot of the frequency spectrum of the first 8 samples
figure;
stem(spectrum_sample,'Linewidth',1.5);
xlabel('frequency (radians/sec) ->');
ylabel('X(\omega) ->');
title('19ucc023 - Mohit Akhouri','Frequency Spectrum of the first
 8 samples of input x[n] - hissing frequencies at 3^{rd} and 7^{th}
 samples');
grid on;

hn = [1 -2*cos(pi/2) 1]; % Filter impulse response for NOTCH FILTER
x = conv(x,hn); % convolution of distorted signal x[n] and impulse
 response of NOTCH FILTER h[n]

% Plots of input signal x[n] for first 8 samples , impulse response
 h[n]
% and convolved signal y[n].
figure;
subplot(3,1,1);
stem(samples_x,'Linewidth',1.5);
xlabel('samples(n) ->');
ylabel('x[n] ->');
title('Plot of the first 8 samples of input signal x[n]');
```

```
grid on;
subplot(3,1,2);
stem(hn,'Linewidth',1.5);
xlabel('samples(n) ->');
ylabel('h[n] ->');
title('impulse response h[n] of the NOTCH FILTER used for
 convolution');
grid on;
subplot(3,1,3);
stem(x(1:8),'Linewidth',1.5);
xlabel('samples(n) ->');
ylabel('x[n] ->');
title('Plot of the first 8 samples of input signal x[n] after
 convolution with NOTCH FILTER');
grid on;
sgtitle('19ucc023 - Mohit Akhouri');

% Plot of the smoothened version of audio signal after removing the
 hissing
% frequencies
figure;
plot(x);
xlabel('samples(n) ->');
ylabel('x[n] ->');
title('19ucc023 - Mohit Akhouri','smoothened version of the audio
 signal x[n] after passing through NOTCH FILTER');
grid on;

sound(x,fs); % listening to the smoothened version of the audio signal
 after passing through NOTCH FILTER
audiowrite('Exp9_obs_3_notch_filter_output.wav',x,fs); % Writing the
 final audio signal to an audio file
```

Published with MATLAB® R2020b