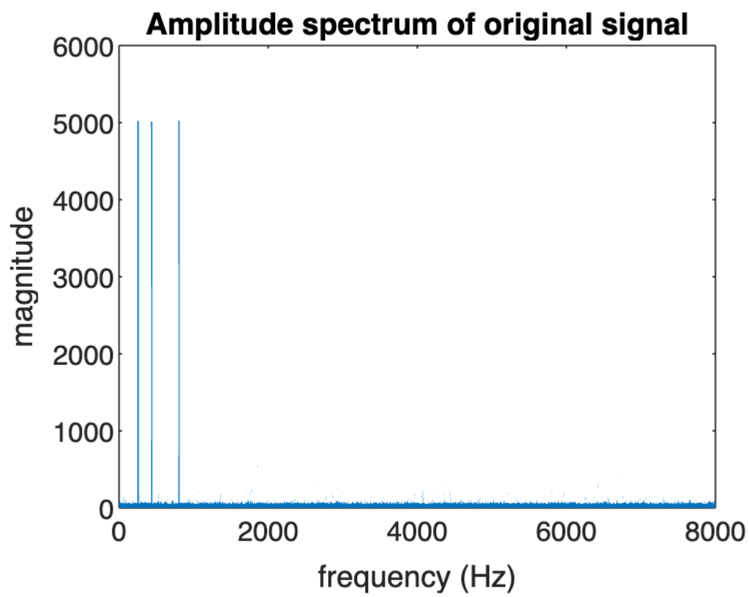
Item 1

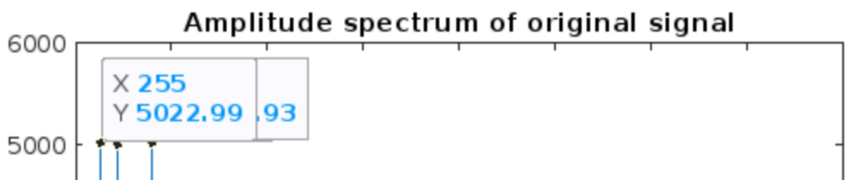
```
close all; clear all; clc;

%load and read the audio file
[sig, Fs] = audioread('mixSinusoid.wav');

% Compute the Fourier transform of the audio signal
N = length(sig); %length of the signal
Y = fft(sig); %Compute the Fourier transform
f = (0:N-1)*(Fs/N); %frequency axis

%plot the magnitude spectrum
plot(f(1:N/2), abs(Y(1:N/2)));
%In the plot, only half the length is used to avoid mirroring
%of the spikes. Only the positive frequencies of the
%Fourier transform are considered.
title('Amplitude spectrum of original signal');
xlabel('frequency (Hz)');
ylabel('magnitude');
```






```
%Using the cursor tool, the frequencies present are 255 Hz,
%437 Hz, and 803 Hz.
```

Low-pass filter

```
% Define the filter parameters
fc = 260;  % Cut-off frequency in Hz
buttorder = 6;  % Filter order

% Normalize the cut-off frequency
fnlow = fc / (Fs/2);

% Design the Butterworth filter coefficients
[b, a] = butter(buttorder, fnlow, 'low');

% Apply the filter to the audio signal
buttlowpass = filter(b, a, sig);
lowpasssig = fft(buttlowpass);

% Plot the magnitude spectrum
figure();
subplot(311);
plot(f(1:N/2), abs(lowpasssig(1:N/2)));
xlim([0 1000]); % limits the plot to its first 1000 points
title('Magnitude Spectrum of low-pass filtered signal');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
```

The cutoff frequency used to recover the 255 Hz components of the signal is 260 Hz. Combined with a filter order of 6, the component can be recovered with only a small attenuation of its magnitude and a satisfactory suppression of the nearby 437 Hz component.

Bandpass filter

```
% Define the filter parameters
fband = [400 650];  % Passband frequency range in Hz
buttorder = 4;  % Filter order

% Normalize the passband frequencies
fnband = fband / (Fs/2);

% Design the Butterworth filter
[b, a] = butter(buttorder, fnband, 'bandpass');

% Apply the filter to the audio signal
buttbandpass = filter(b, a, sig);
bandpasssig = fft(buttbandpass);

subplot(312);
plot(f(1:N/2), abs(bandpasssig(1:N/2)));
xlim([0 1000]); % limits the plot to its first 1000 points
title('Magnitude spectrum of bandpass filtered signal');
xlabel('Frequency (Hz)');
```

```
ylabel('Magnitude');
```

To recover the 437 Hz component, a lower cutoff frequency of 400 Hz and a higher cutoff frequency of 650 Hz were used in a bandpass filter. The filter order of 4 was sufficient to suppress the other components.

High-pass filter

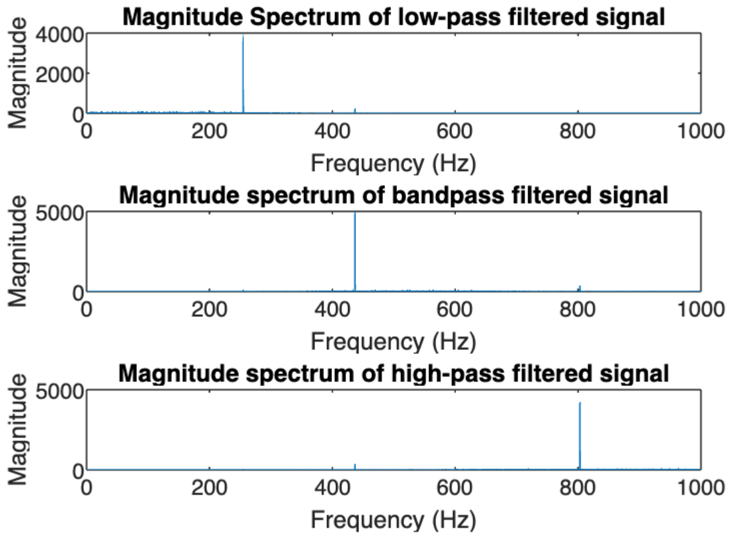
```
% Define the filter parameters
fc = 740;  % Cut-off frequency in Hz
buttorder = 5;  % Filter order

% Normalize the cut-off frequency
fnhigh = fc / (Fs/2);

% Design the Butterworth filter coefficients
[b, a] = butter(buttorder, fnhigh, 'high');

% Apply the filter to the audio signal
butthighpass = filter(b, a, sig);
highpasssig = fft(butthighpass);

% Plot the magnitude spectrum
subplot(313);
plot(f(1:N/2), abs(highpasssig(1:N/2)));
xlim([0 1000]); % limits the plot to its first 1000 points
title('Magnitude spectrum of high-pass filtered signal');
xlabel('Frequency (Hz)');
ylabel('Magnitude');
```



The 803 Hz component was recovered by using a cutoff frequency of 740 Hz and a filter order of 5. The lower components were successfully suppressed.