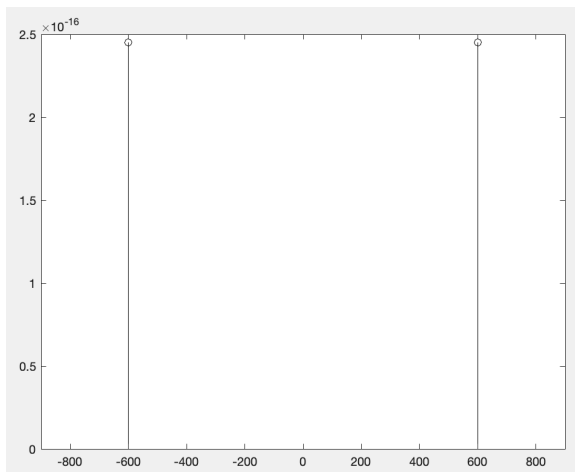


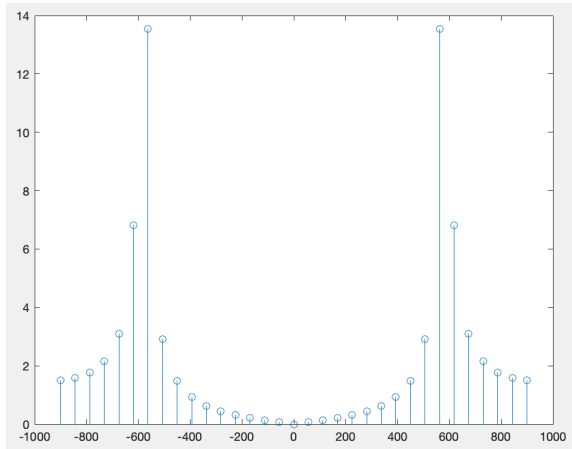
Roland Yin
ECE 130B
Software Exercise 3

```
clear; close all; clc;
%% Part 1.1
f0 = 1200; fs1 = 1800; t0 = 1/f0;
t1 = linspace(0,1/f0,ceil(t0/(1/fs1)));
x1 = sin(2*pi*f0*t1);
dft1 = fftshift(fft(x1));
aliasFreq1 = mod(f0,fs1);
if aliasFreq1 > fs1/2
    aliasFreq1 = fs1 - aliasFreq1;
end
figure(1);
ymax = max(abs(dft1));
stem(aliasFreq1,ymax,'k'); hold on;
stem(-aliasFreq1,ymax,'k'); hold off;
xlim([-fs1/2 fs1/2]);
% Sinc interpolation on the samples would recover the signal sin(2pi600t),
% because the frequency content of the sampled signal contains peaks at
% +-600 Hz with equal magnitude, but opposite complex parts. The peaks
% exist at +-600 Hz due to aliasing in the sampled signal. To satisfy the
% Nyquist rate, the sampling frequency must be 2*f0, or 2400 Hz.
```



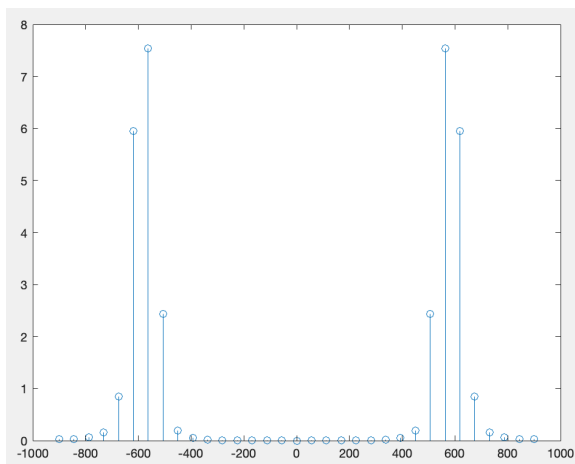
```
%% Part 1.2
L = 33; % L = 33 was chosen instead of 32 for symmetry purposes
t2 = linspace(0,L*1/fs1,L);
x2 = sin(2*pi*f0*t2);
dft2 = fftshift(fft(x2));
freq2 = linspace(-fs1/2,fs1/2,length(dft2));
figure(2); stem(freq2,abs(dft2));
% The plot closely resembles the analytical sketch from part 1.1. There are
% two clear peaks at around +-600 Hz, which is expected given that the
% sampling frequency is below the Nyquist rate. The peaks in the plot
```

```
% correspond to a sinusoid of frequency 600 Hz [sin(2pi(600)t)]. The main
% difference between this plot and the theoretically expected frequency
% domain characterization of the sampled signal from part 1.1 is that this
% plot contains non-zero magnitudes at frequencies other than +/-600 Hz.
% This is a result of the sharp, jagged, and distorted sampled signal,
% which only barely resembles a periodic sinusoidal function and thus
% contains frequencies outside of +/-600 Hz in its frequency domain.
```



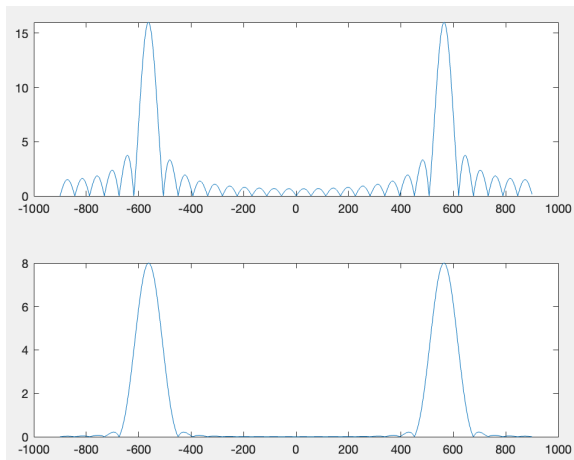
%% Part 1.3

```
n = 0:1:L-1;
wHann2 = 0.5*(1-cos(2*pi*n/(L-1)));
x3 = x2.*wHann2;
dft3 = fftshift(fft(x3));
freq3 = linspace(-fs1/2,fs1/2,length(dft3));
figure(3); stem(freq3,abs(dft3));
% The plot constructed using a Hann window contains the same fundamental
% structure as the plot constructed using a rectangular window but displays
% greater precision near the peaks. Unlike the plot from the rectangular
% window, the plot from the Hann window shows sharper edges near the peaks,
% more closely resembling the delta-like shape that is theoretically
% expected in the DFT.
```



%% Part 1.4

```
N = 2048;
x2ZeroPad = [x2,zeros(1,N-L)];
x3ZeroPad = [x3,zeros(1,N-L)];
dft2ZeroPad = fftshift(fft(x2ZeroPad));
dft3ZeroPad = fftshift(fft(x3ZeroPad));
freq2ZeroPad = linspace(-fs1/2,fs1/2,length(dft2ZeroPad));
freq3ZeroPad = linspace(-fs1/2,fs1/2,length(dft3ZeroPad));
figure(4);
subplot(2,1,1); plot(freq2ZeroPad,abs(dft2ZeroPad));
subplot(2,1,2); plot(freq3ZeroPad,abs(dft3ZeroPad));
% The frequency domain magnitude plots that are constructed using very long
% zero-padded sampled signals and windowed using rectangular and Hann
% windows both display distinct peaks near the expected locations at +/-600
% Hz, with greater accuracy than the plots from the non-zero-padded sampled
% signals. Regardless of the window used, the magnitude plot contained
% smaller oscillations around the peaks with the zero-padded signal.
% Nonetheless, comparison of the two differently windowed zero-padded plots
% resulted in significant differences as well. While the rectangular
% windowed signal produced sharper peaks in the DFT, the Hann windowed
% signal produced a much smoother plot. Ultimately, depending on the
% application and system that the signal is traveling through, we may
% choose to prioritize sharper peaks over a smoother plot, or vice versa.
```

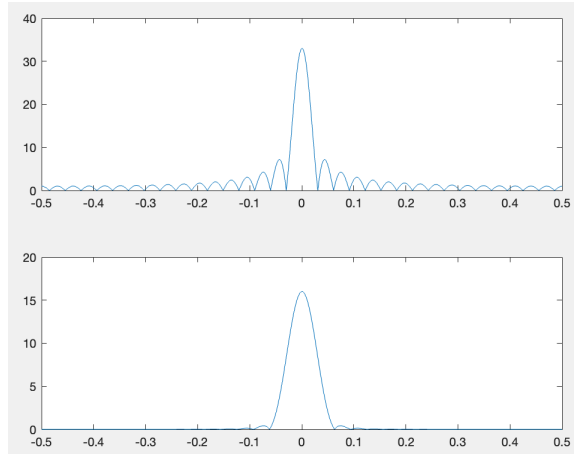


% Caption: top plot corresponds to rectangular window; bottom plot corresponds to Hann window.

%% Part 1.5

```
wRect = ones(1,L);
dftRect = fftshift(fft([wRect ,zeros(1,N-L)]));
dftHann = fftshift(fft([wHann2,zeros(1,N-L)]));
freqRect = linspace(-1/2,1/2,length(dftRect));
freqHann = linspace(-1/2,1/2,length(dftHann));
figure(5);
subplot(2,1,1); plot(freqRect,abs(dftRect));
subplot(2,1,2); plot(freqHann,abs(dftHann));
% The DTFT of the rectangular window and Hann window display contrasting
```

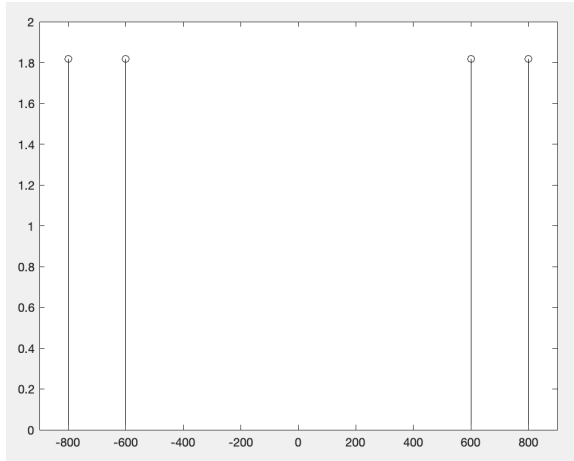
```
% structures. The rectangular window yields a sinc-type structure, while
% the Hann window produces a Gaussian-type structure. Although both plots
% exhibit similar windowed averaged shapes, a crucial difference lies in
% the small additional oscillating peaks in the DTFT of the rectangular
% window, which the DTFT of the Hann window do not contain.
```



```
% Caption: top plot corresponds to rectangular window; bottom plot corresponds
to Hann window.
```

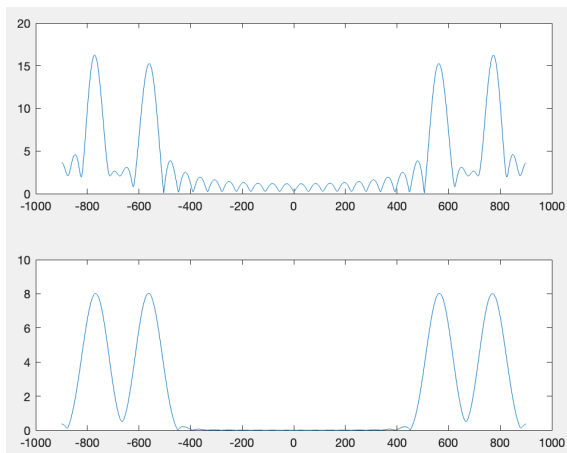
%% Part 1.6

```
f6a = 1200; f6b = 1000;
f6 = lcm(f6a,f6b); t6Period = 1/f6;
t6 = linspace(0,1/f6,ceil(t6Period/(1/fs1)));
x6 = sin(2*pi*f6a*t6) + sin(2*pi*f6b*t6);
dft6 = fftshift(fft(x6));
aliasFreq6a = mod(f6a,fs1);
if aliasFreq6a > fs1/2
    aliasFreq6a = fs1 - aliasFreq6a;
end
aliasFreq6b = mod(f6b,fs1);
if aliasFreq6b > fs1/2
    aliasFreq6b = fs1 - aliasFreq6b;
end
figure(6);
ymax = max(abs(dft6));
stem( aliasFreq6a,ymax,'k'); hold on;
stem(-aliasFreq6a,ymax,'k'); hold on;
stem( aliasFreq6b,ymax,'k'); hold on;
stem(-aliasFreq6b,ymax,'k'); hold off;
xlim([-fs1/2 fs1/2]);
% A sinc interpolation on the DTFT would result in the signal sin(2pi600t)
% + sin(2pi800t). This is easily verifiable by observing the DTFT, which
% exhibits peaks at +-600 Hz and +-800 Hz, as expected due to aliasing in
% the sampled signal. To satisfy the Nyquist rate, the sampling frequency
% must be 2*f6, or 12000 Hz.
```



%% Part 1.7

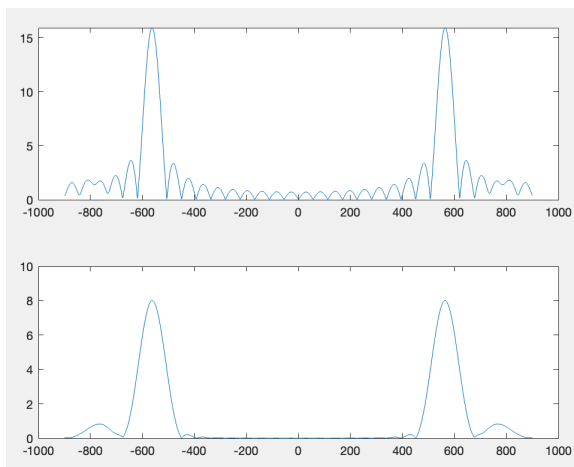
```
t7 = linspace(0,L*1/fs1,L);
x7 = sin(2*pi*f6a*t7) + sin(2*pi*f6b*t7);
x7Hann = x7.*wHann2;
x7ZeroPad = [x7, zeros(1,N-L)];
x7HannZeroPad = [x7Hann, zeros(1,N-L)];
dft7ZeroPad = fftshift(fft(x7ZeroPad));
dft7HannZeroPad = fftshift(fft(x7HannZeroPad));
freq7ZeroPad = linspace(-fs1/2,fs1/2,length(dft7ZeroPad));
freq7HannZeroPad = linspace(-fs1/2,fs1/2,length(dft7HannZeroPad));
figure(7);
subplot(2,1,1); plot(freq7ZeroPad,abs(dft7ZeroPad));
subplot(2,1,2); plot(freq7HannZeroPad,abs(dft7HannZeroPad));
% The peaks in the 1024 point DFT are distinguishable using both the
% rectangular window and the Hann window. In both plots, peaks exist near
% +/-600 Hz and +/-800 Hz. Additionally, the rectangular windowed sampled
% signal displays an oscillatory frequency domain characterization, whereas
% the Hann windowed sampled signal displays a clean and smooth frequency
% domain characterization with only slightly wider peaks. This is
% consistent with previous observations about the frequency domain
% characterizations of rectangular and Hann windows.
```



% Caption: top plot corresponds to rectangular window; bottom plot corresponds to Hann window.

%% Part 1.8

```
x8      = sin(2*pi*f6a*t7) + 1/10*sin(2*pi*f6b*t7);
x8Hann  = x8.*wHann2;
x8ZeroPad = [x8      ,zeros(1,N-L)];
x8HannZeroPad = [x8Hann,zeros(1,N-L)];
dft8ZeroPad = fftshift(fft(x8ZeroPad));
dft8HannZeroPad = fftshift(fft(x8HannZeroPad));
figure(8);
subplot(2,1,1); plot(freq7ZeroPad ,abs(dft8ZeroPad ));
subplot(2,1,2); plot(freq7HannZeroPad,abs(dft8HannZeroPad));
% The peaks in the DFT are still distinguishable, but the contribution that
% the 1000 Hz sinusoid makes to the frequency domain characterization of
% the mixed frequency function is only barely visible in the frequency
% spectrum, regardless of how the signal is windowed. In the frequency plot
% of the rectangular windowed signal, there is a small uneven oscillation
% at 800 Hz, while in the frequency plot of the Hann windowed signal, there
% is a noticeable bump near 800 Hz.
```



% Caption: top plot corresponds to rectangular window; bottom plot corresponds to Hann window.

%% Part 1.9

```
% Decreasing the number of samples taken to 17 caused the accuracy of the
% DFT to drop and the width of the peaks to increase. The plots from parts
% 1.5 and 1.8 changed most significantly; in both, the peaks in the DFT
% were much wider, regardless of how the sampled signal was windowed.
% Increasing the number of samples taken to 67 caused the accuracy of the
% DFT to increase and the height of the peaks to sharpen. The plots from
% 1.2, 1.3, 1.7, and 1.8 changed most significantly. In figure 2 and 3, the
% sharpness of the peaks increased significantly around +/-600 Hz; in figure
% 7, the peaks at +/-600 Hz and +/-800 Hz were very clear; and in figure 8,
% the peaks at +/-800 Hz which correlate to the 1000 Hz signal also became
% more clear, although they were still hard to notice compared to the peaks
% at +/-600 Hz which correspond to the 1200 Hz signal
```

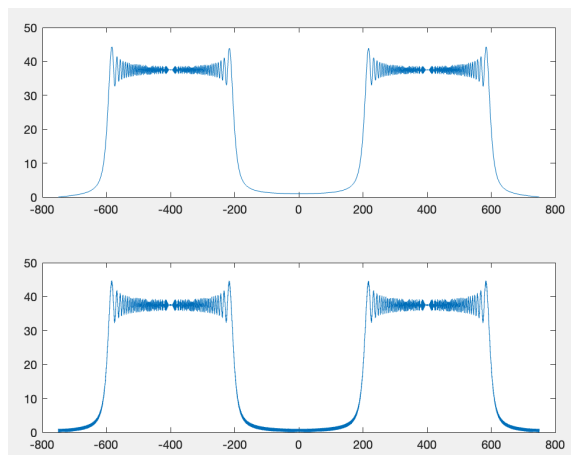
```
% Decreasing the amount of zero-padding to a vector length of 512 caused
% the accuracy of the DFT to drop. In all plots, there were fewer points,
% and as a result, the sharpness of peaks in the plots dulled. Other than
% duller peaks, there was no significant structural change in the frequency
% domain characterization of the sampled signals from 1.2 to 1.8.
% Increasing the amount of zero-padding to a vector length of 2048 caused
% the accuracy of the DFT to increase. In all plots, the number of points
% increased, and as a result, the sharpness of the peaks increased. Other
% than sharper peaks, there was no significant structural change in the
% frequency domain characterization of the sampled signals from 1.2 to 1.8.
```

%% Part 2.10

```
fs10 = 1500; numS = 1;
t = linspace(0,numS,fs10*numS);
audio10 = chirp(t,200,1,600);
% sound(audio,fs10);
```

%% Part 2.11

```
dtft11 = fftshift(fft(audio10));
dtft11L = fftshift(fft([audio10,zeros(1,10*fs10)]));
freq11 = linspace(-fs10/2,fs10/2,length(dtft11));
freq11L = linspace(-fs10/2,fs10/2,length(dtft11L));
figure(11);
subplot(2,1,1); plot(freq11 ,abs(dtft11 ));
subplot(2,1,2); plot(freq11L,abs(dtft11L));
% The frequency content contains a plateau of peaks from +-200 Hz to +-600
% Hz. This is consistent with what we expect for a chirp from 200 Hz to 600
% Hz, since a chirp is a signal whose frequency varies linearly over time.
% Although there are also frequencies from 0 Hz to 200 Hz and 600 Hz to 700
% Hz, the amplitude of the peaks makes these frequencies negligible.
```



```
% Caption: top plot corresponds to non-zero-padded signal; bottom plot
% corresponds to zero-padded signal.
```

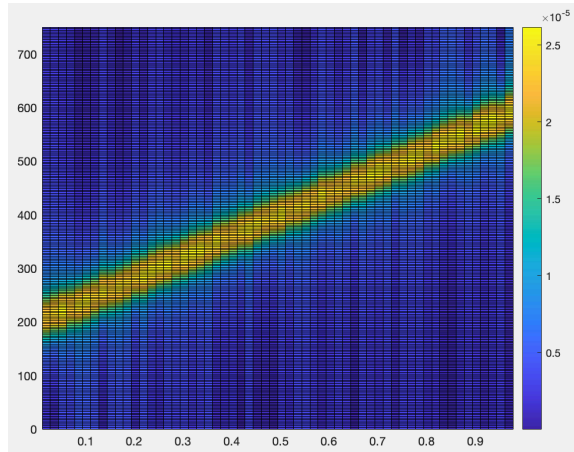
%% Part 2.12

```
t12 = linspace(0,(L-1)/fs10*numS,L);
wHann12 = 0.5*(1-cos(2*pi*t12/(L-1)));
```

```

overlap12 = round(L*0.25); nfft12 = 512;
[S12,F12,T12] = spectrogram(audio10,wHann12,overlap12,nfft12,fs10);
figure(12); pcolor(T12,F12,abs(S12)); colorbar;
% The plot distinctly shows the frequency linearly increasing from 200 Hz
% to 600 Hz in 1 second. There is a high-intensity +-25 Hz layer relative
% to the most intense frequency at each time step sandwiching the linear
% trend and a medium-intensity +-25 Hz layer extending beyond that.

```

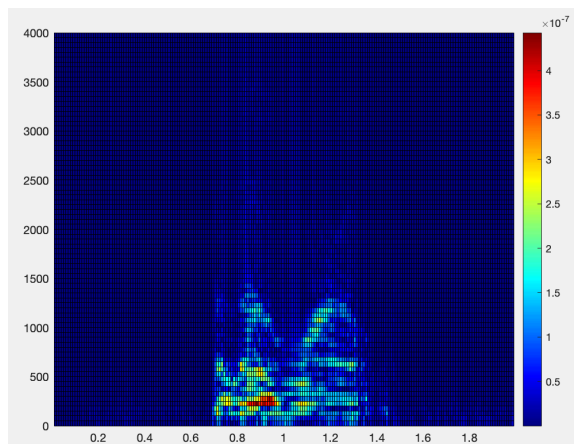


%% Part 2.13

```

load('audio8k.mat');
audio13 = getaudiodata(record8k);
fs13 = 8000; recordL = 2; period13 = 0.02;
tDiv = recordL/length(audio13); stftL = round(period13/tDiv);
t13 = linspace(0, (stftL-1)/fs13*recordL, stftL);
wHann13 = 0.5*(1-cos(2*pi*t13/(stftL-1)));
overlap13 = round(stftL*0.5);
nfft13 = round(1/100*length(audio13));
[S13,F13,T13] = spectrogram(audio13,wHann13,overlap13,nfft13,fs13);
figure(13); pcolor(T13,F13,abs(S13)); colormap jet; colorbar;

```



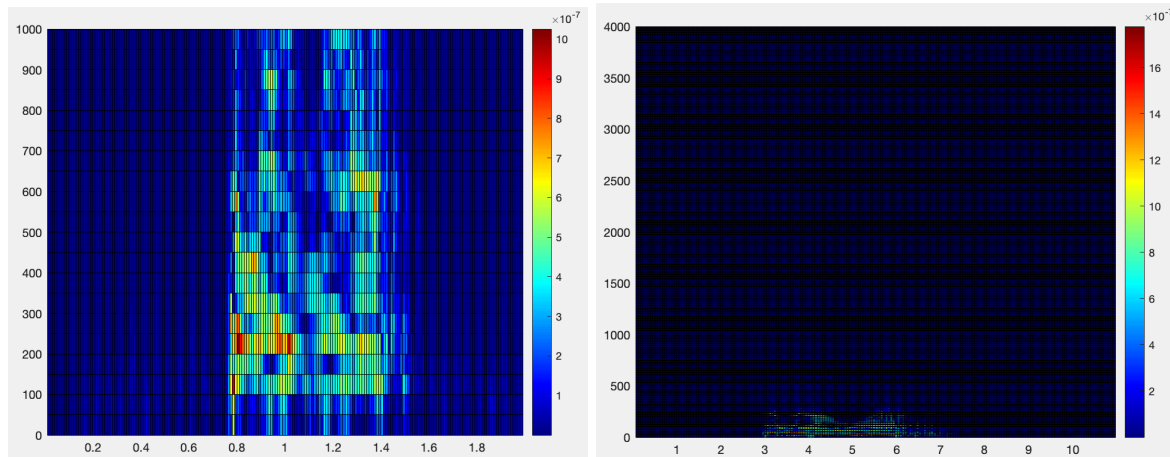
%% Part 2.14

```

% Drastically different speech segments result in visibly distinct patterns
% in the STFT plots. High-pitched sounds exhibit more high intensity
% high-frequency values, while lower-pitched sounds primarily show high
% intensity low-frequency values. In addition, muffled sounds present a

```


% spectrogram with significant intensity across the entire frequency
% spectrum, while clear sounds produce a pinpoint precise spectrogram with
% little frequency smearing across the plot.



% Caption: left plot corresponds to muffled sound; bottom plot corresponds to clear sound.