```
% reading in the audio file
% x: audio data
% fs: sampling frequency
[x, fs] = audioread('persevere_bad.wav');

% chunck data into frames with zero overlap
frames = [];
for i = 1:length(x)/fs
    l_bound = (i-1)*fs + 1;
    r_bound = i*fs;
    frames = [frames; x(l_bound:r_bound)'];
end

% get fft of each frame
fft_frames = [];
for i = 1:size(frames, 1)
    fft_frames = [fft_frames; fft(frames(i, :))];
end

% only keep the first half of the fft
fft_frames = fft_frames(:, 1:length(fft_frames(1, :))/2);

% find the average of the fft's energy
avg_fft = mean(abs(fft_frames).^2);

% finding the frequency of the maximum energy
% mag: magnitude of the maximum energy, actual maximum energy
% freq: frequency of the maximum energy, index of the maximum energy
[mag, freq]= max(abs(avg_fft(500:2000)));
freq = freq + 500; % adding in the offset
BW = 50;

notch_filter = designfilt('bandstopiir','FilterOrder',2, ...
    'HalfPowerFrequency1',freq-BW,'HalfPowerFrequency2',freq+BW, ...
    'DesignMethod','butter','SampleRate', fs);

% extract the poles and zeros of the notch filter
[b, a] = tf(notch_filter);
y = filter(b, a, x);

% if the length of the vectors are not equal, pad the shorter vector
with zeros
max_len = max(length(b), length(a));
if length(b) < max_len
    b = [b, zeros(1, max_len - length(b))];
elseif length(a) < max_len
    a = [a, zeros(1, max_len - length(a))];
end

z = roots(b); % zeros
```

```
p = roots(a); % poles

% printing deliverables
freq = freq

poleRad = angle(p)
zeroRad = angle(z)

poleDeg = rad2deg(poleRad)
zeroDeg = rad2deg(zeroRad)

length(avg_fft)

% everything below this point is for plotting
figure;

% plot the average energy of the combined frames
subplot(2, 1, 1);
plot(avg_fft); % this range was chosen by visually inspecting the plot
title('Average energy of the combined frames');

% sketch of pole zero plot that removes the tones
subplot(2, 1, 2);
title('Pole Zero Plot');
rectangle('Position', [-1, -1, 2, 2], 'Curvature', [1, 1]);
axis equal;
hold on;

% find the maximum value of the zeros and poles
max_val = max([max(abs(real(z))), max(abs(imag(z))),
max(abs(real(p))), max(abs(imag(p)))]) * 1.1;

% if the maximum value is less than 1, set it to 1
if max_val < 1.1
    max_val = 1.1;
end

% set the axis limits
axis([-max_val, max_val, -max_val, max_val]);
% plot the x & y axes through the origin
graph = plot([-max_val max_val],[0 0], 'black');
graph = plot([0 0], [-max_val max_val], 'black');

% plot zeros
scatter(real(z), imag(z), max_val * 125, 'o', 'blue'); % max_val * 125
in order to scale markers
% plot poles
scatter(real(p), imag(p), max_val * 125, 'x', 'red');
```