ECE 345: Linear Systems and Signals
Fall 2020
Lab #4 Report

Note: you can use the equation editor in MS Word or a tool such as LaTeXiT to generate formulas for questions which ask about formulas. Alternatively, you can write your derivation and put a photo into the box.

Mini-Lab 1 (54 points)		**Mini-Lab 2 (36 points)**			**TOTAL (90 points)**

Group members:

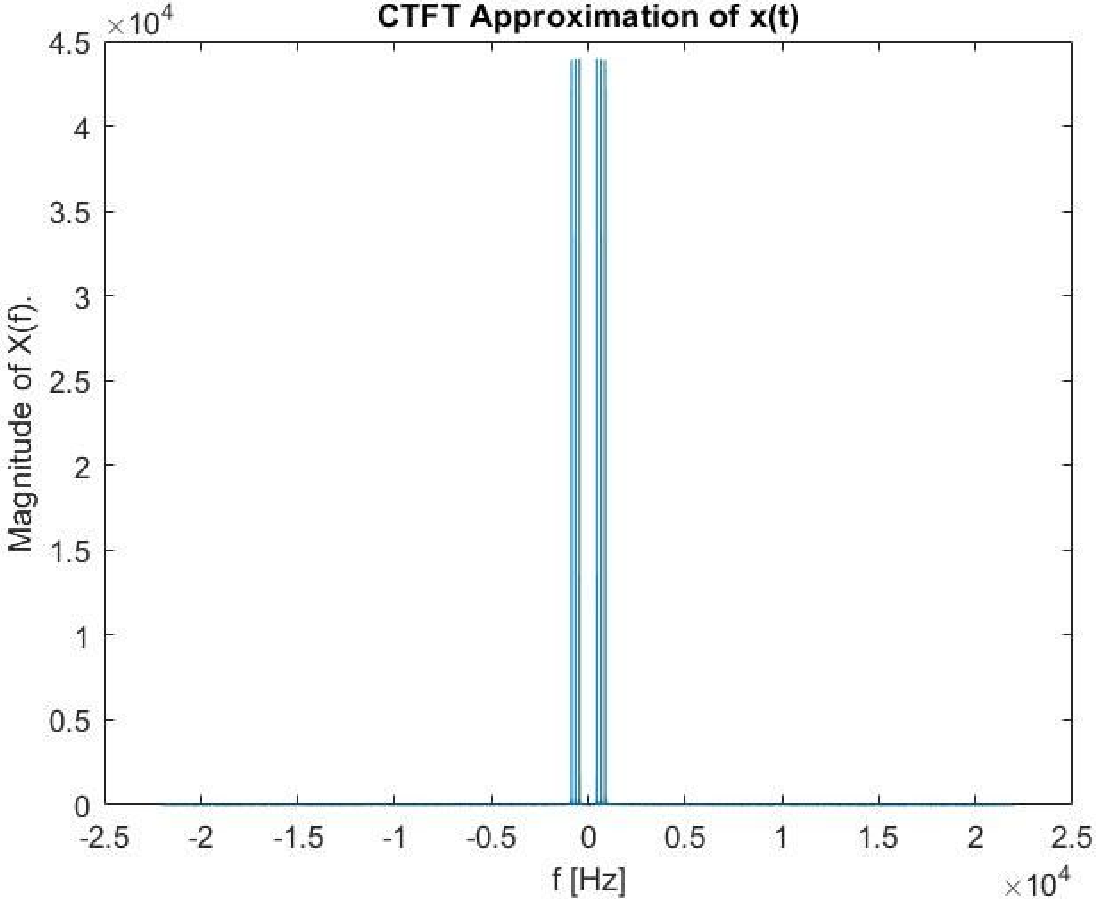
- Alan Chacko
- Guilherme Silva
- Hagan Pereira
- Matthew Avendano

Minilab 1: Sampling above the Nyquist rate (54 points)

(a) (2 points) What is the Nyquist rate for this signal?

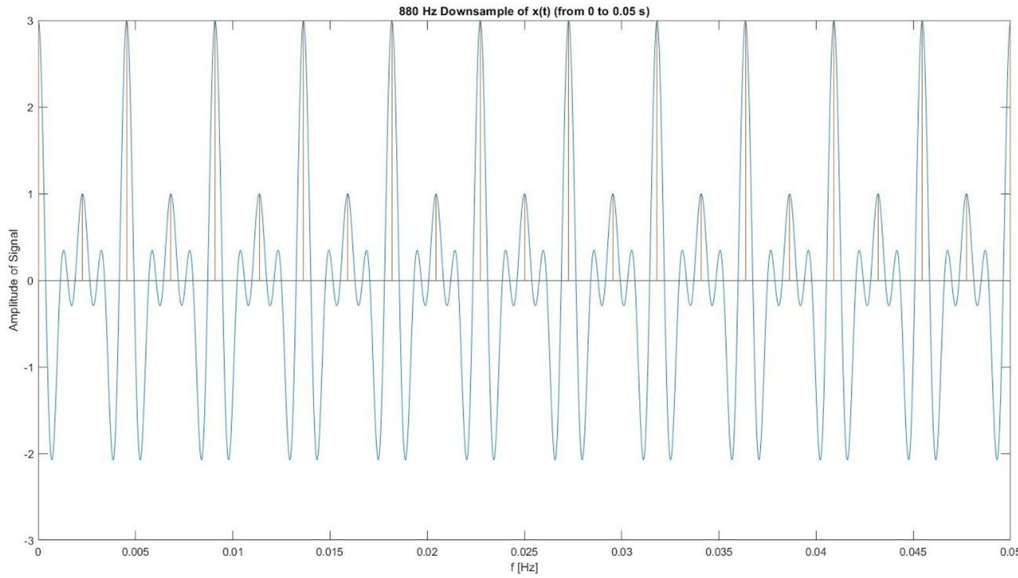
The largest frequency of the input signal was $\omega_M = 1760\pi$ so the Nyquist rate is $2\omega_M = 3520\pi$, or in terms of frequency $f_N = 1760$ Hz.

(b) (8 points) Use a $N = 2^{20}$ point FFT to approximate the CTFT of $x(t)$. Plot the magnitude as a function of the frequency f in Hz.

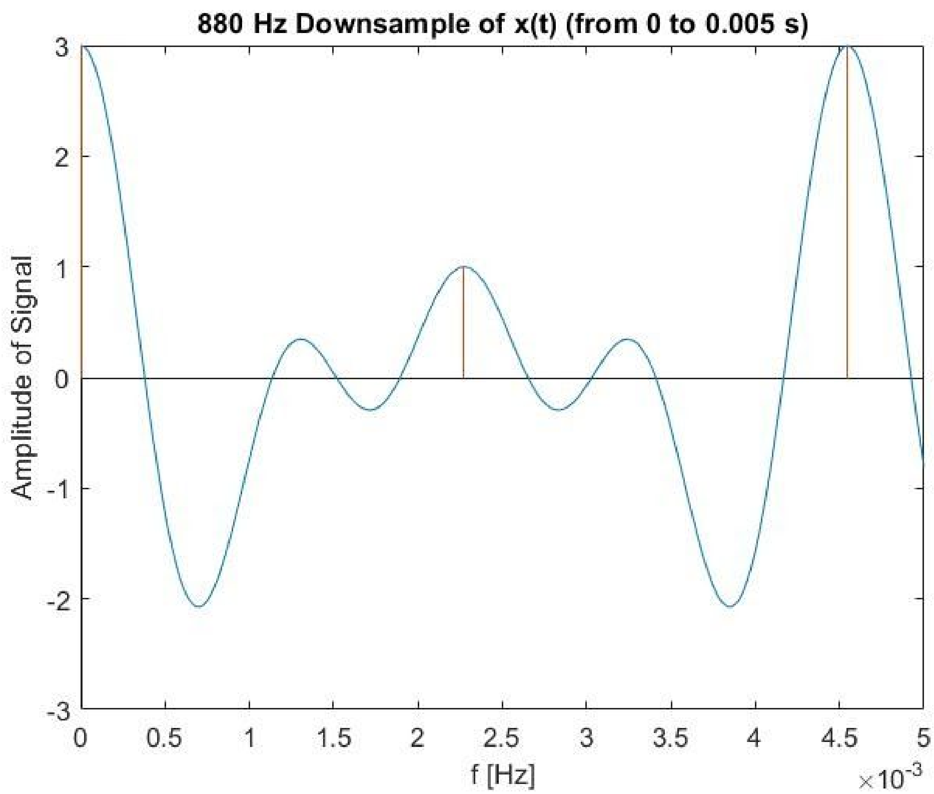


(c) (10 points) Sample the signal at half the Nyquist rate. Plot the original signal from $t = 0$ to $t = 0.05$ using plot and the sampled signal using stem on the same plot. Play your sampled signal using soundsc with the sampling rate f_0. Can you hear the original signal? How does the plot show that you should not expect to hear the original signal?

As the plots below show, at a frequency of half the Nyquist rate, only half of the signal is sampled, which is the top half. This isn't enough information to produce the original signal. Playing the signal leads to a distorted version (lots of buzzing) of the original signal at a much lower pitch.

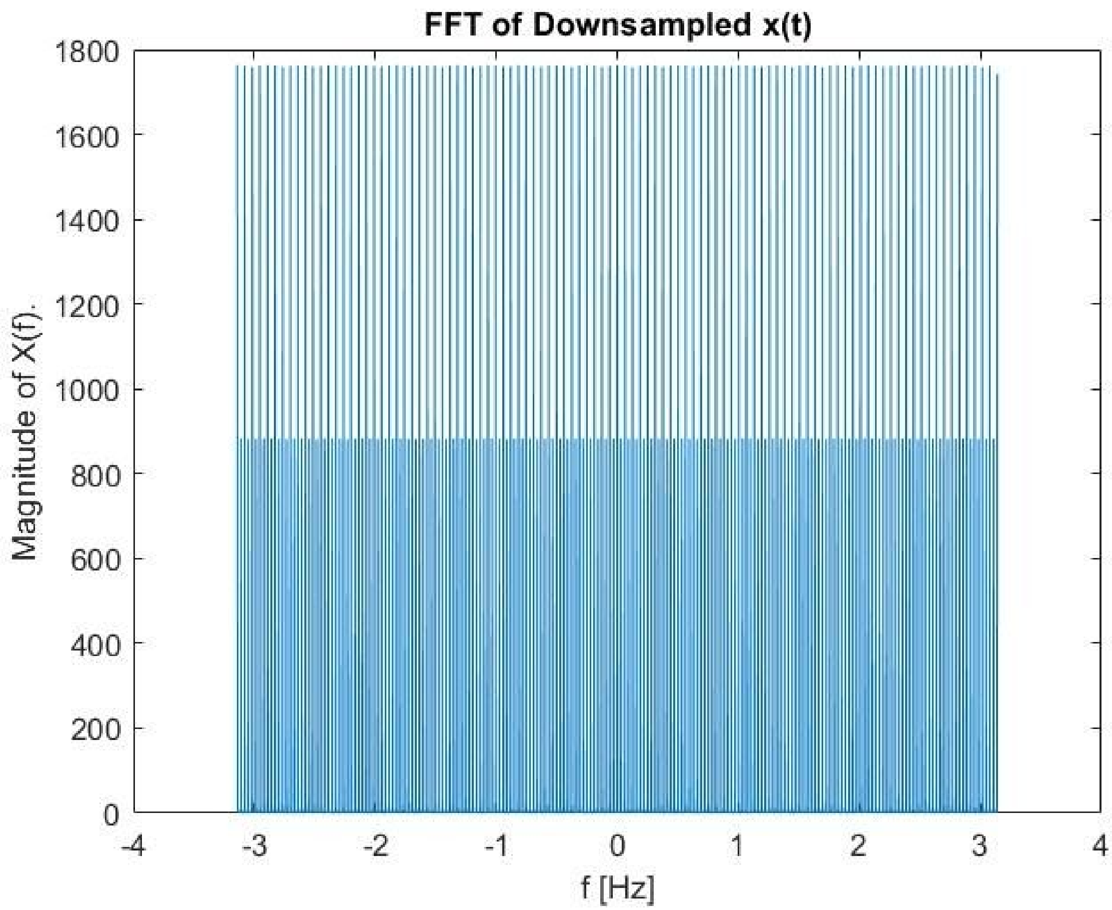


Below is a clearer image of the same signal at a smaller period.



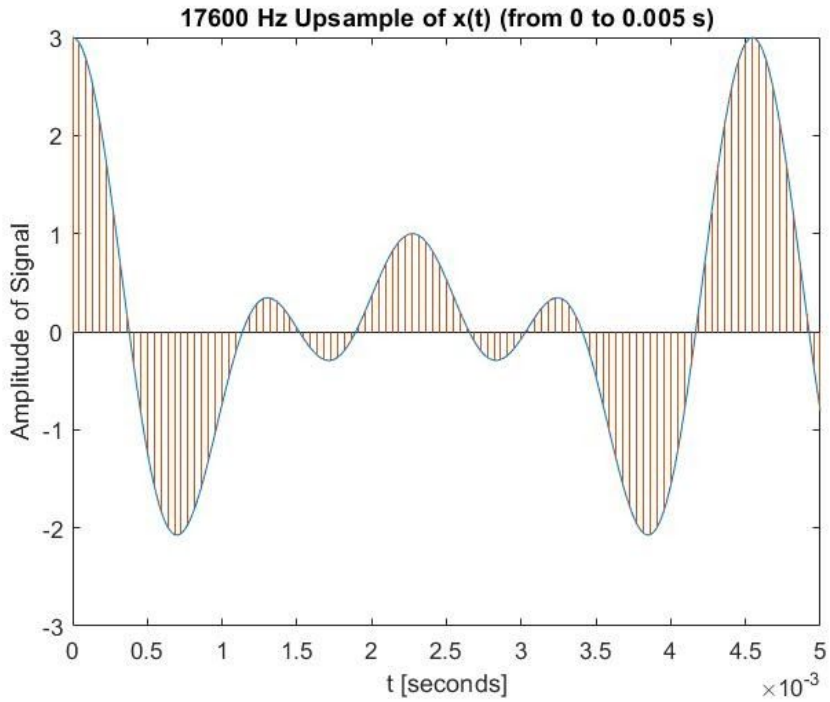
NOTE: Some of the sample points are where the amplitude is 0, one is between 1e-3 and 1.5e-3, the other is between 3e-3 and 3.5e-3. So the above picture DOES HAVE a sampling frequency of 880 Hz, NOT 440 Hz.

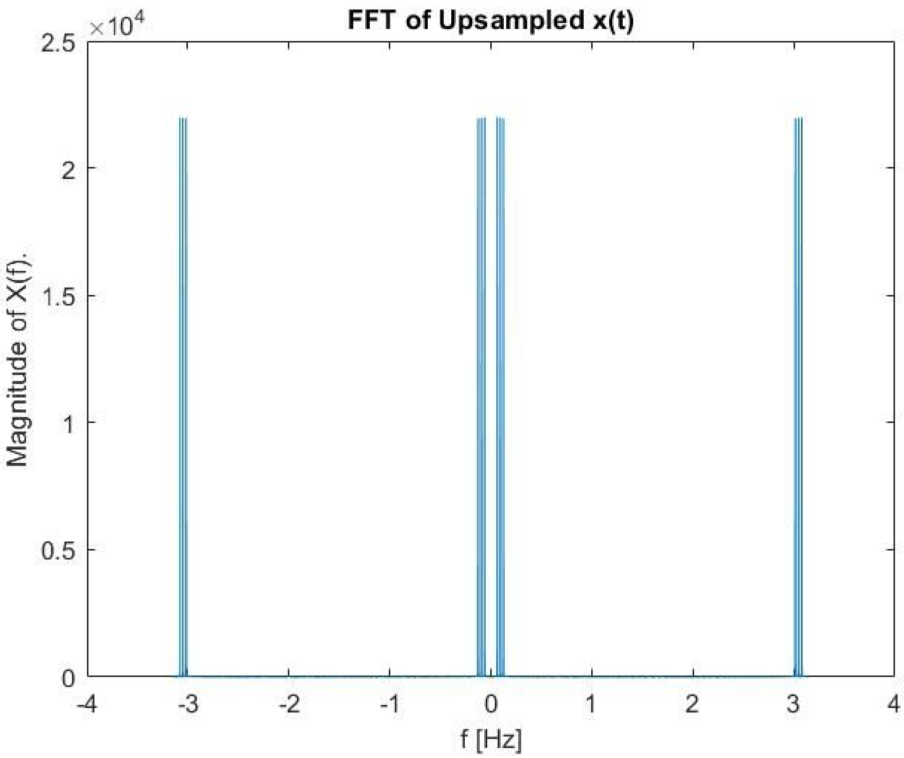
(d) (8 points) Plot the magnitude of the DTFT of the signal from (c) using fft and fftshift.



(e) (10 points) Now sample the signal at $K = 10$ times the Nyquist rate. Play your downsampled signal using soundsc with the sampling rate f_0. Can you hear the original signal? Plot the DTFT of the resulting downsampled signal. What do you see?

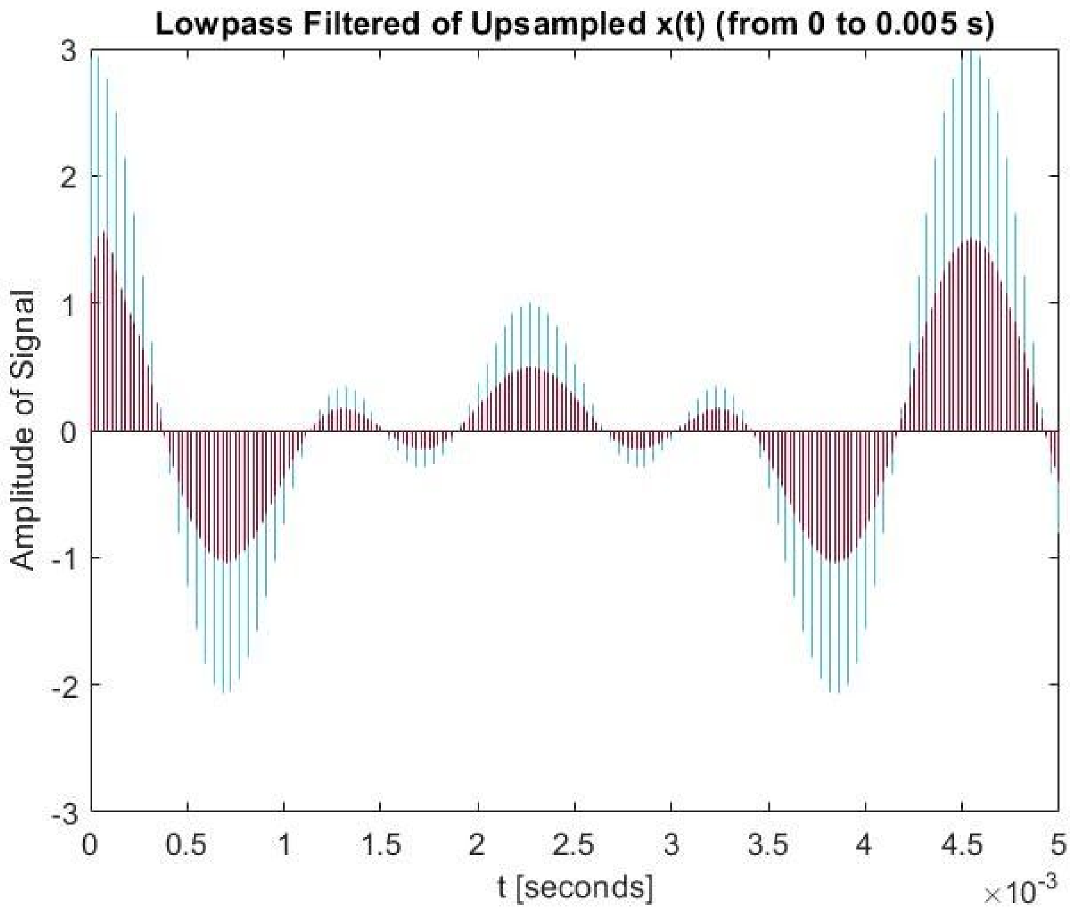
The downsampled signal at a higher sampling rate produces the sound of the original signal. Below is a plot of the sample as well as its DTFT. Note, it is impossible to produce a signal of 17 kHz since it requires us to downsample, then upsample by 2.5, which isn't an integer value. So the below graph is actually for 22 kHz for an upsample of 2 and a downsample of 2. The other option was an upsample of 3 and downsample of 3, which lead to a 14 kHz signal, which didn't have as clear of a sound.





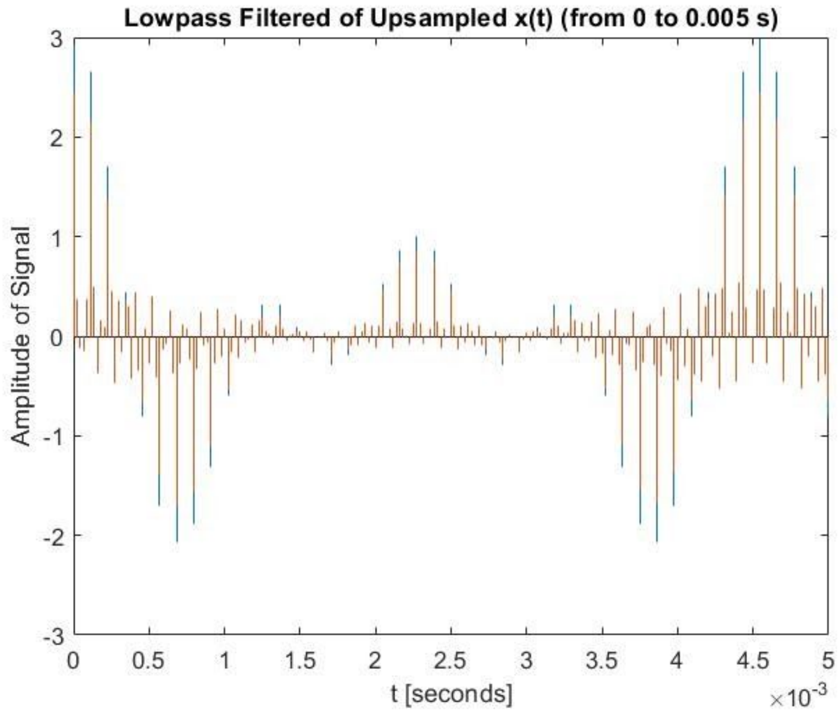
(f) (10 points) Use lowpass to filter the signal from the previous part with cutoff equal to twice the Nyquist rate. Play the resulting signal. How does it compare to the original signal from part (a)?

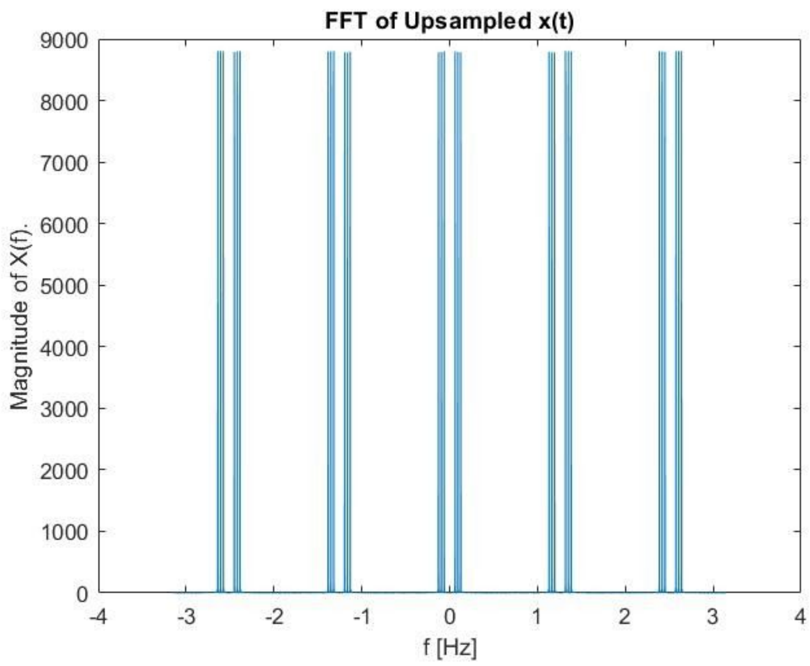
The signal sounds identical to the original, even though the low-pass filter (red) seemed to just reduce the amplitude of the previous signal.



(g) (6 points) Repeat the previous two parts with $K = 5$. What differences do you see and hear?

The filter (in yellow) has less of an effect on the amplitude of the signal, but due to the lower sampling rate, the output sound has more of a buzzing effect applied to it. It is notable that the FFT produces 5 pairs of the three adjacent frequencies.



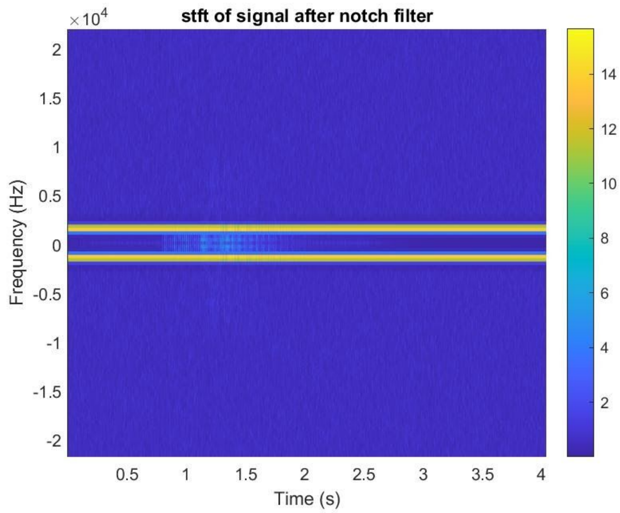


Mini-lab 2: Audio manipulation and filtering (36 points)

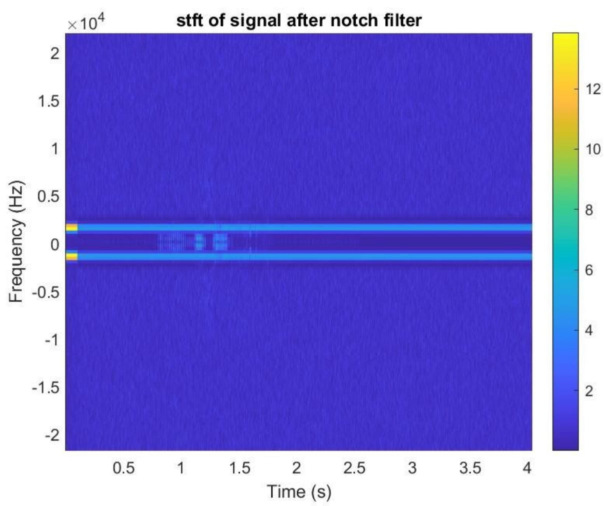
(a) (3 points) Read in the file mysteryclip.wav, extract the left channel, and listen to the clip. Can you tell what is being said?

It is not possible to tell what the message of the soundwave is. The distortion, reverb, and high pitch are too strong.

(b) (8 points) Compute the STFT and save the resulting image.

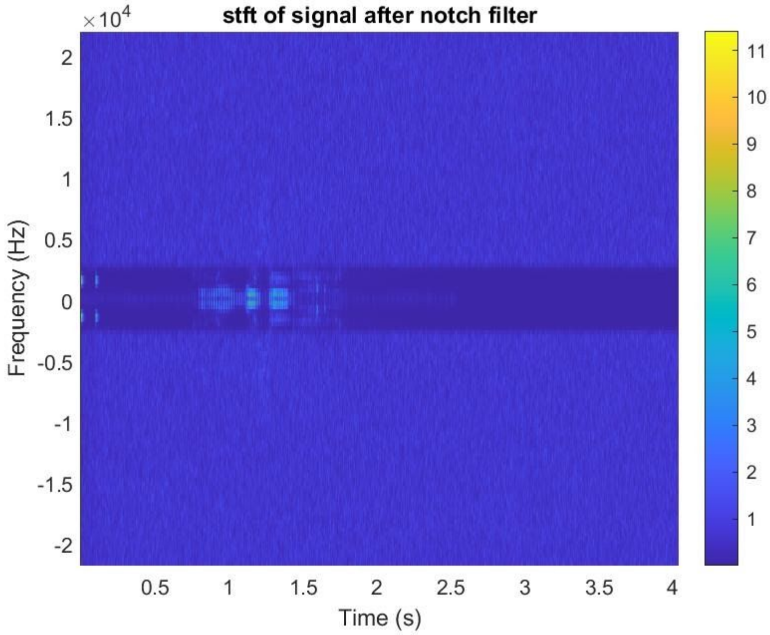


(c) (8 points) Design a DT filter to remove the reverb of the signal. Plot the STFT.



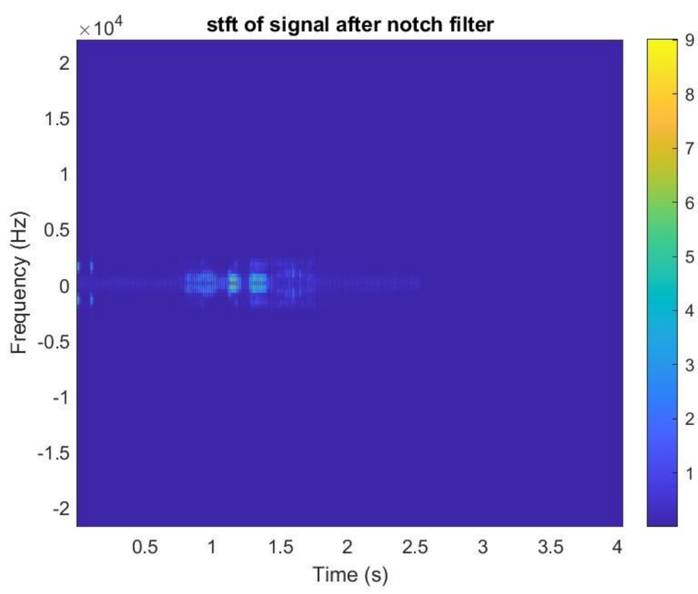
To remove the reverb we used the filter $G(z) = 1 - rz^{-D*f_s}$

(d) (8 points) Design a notch filter using iirnotch and use filter to remove the tone. Plot the STFT of the result.



As we can see the lines around the sound wave disappeared. The frequency of the cosine wave was 1500Hz

(e) (8 points) Use the lowpass function to filter out the highpass noise and listen to the result. Adjust your cutoff for the lowpass filter appropriately to make the audio signal understandable. Plot the STFT of the result.



The soundwave now looks mostly normal (other than the fact that it looks small on this plot). It is also possible to understand what the clip is saying. There wasn't much tuning needed for the lowpass filter. Changing frequencies did not improve the quality too much. As such we used 100Hz for this.

(f) (1 point) What is the person saying in the clip?
You are a wizard Harry