FFT/IFFT Audio Signal Processing - Noise canceling application

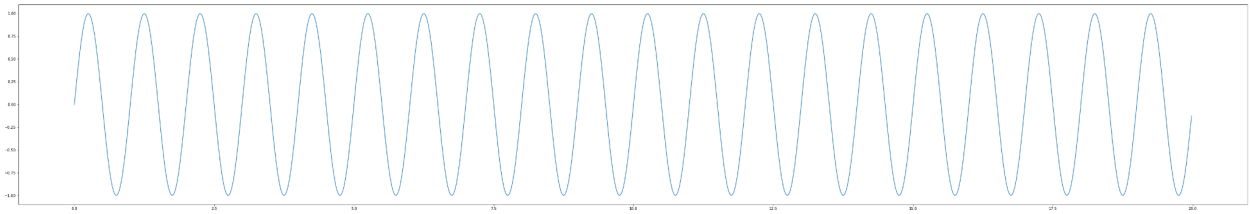
Before running this code be sure to include these libraries before running the code.

import numpy as np
from scipy import fftpack
from matplotlib import pyplot as plt
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline

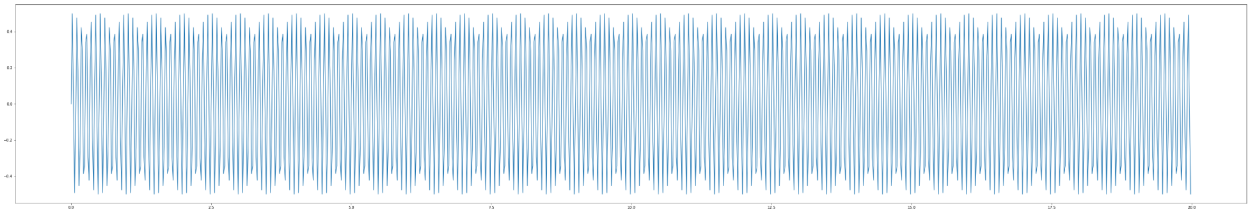
This code generates three wave signals of different frequencies and plots them in two different graphs. One is representing the time domain and the other is in the frequency domain. The frequencies being generated can be altered by changing the number in ‘freq1’,’freq2’, and ‘freq3’ to combine different frequencies.

```
freq1=1 #1Hz
time_step = 0.02
period1 = 1/freq1
time_vec = np.arange(0, 20, time_step)

sig1=(np.sin(2*np.pi/period1 * time_vec))
plt.figure(figsize=(60,10))
plt.plot(time_vec,sig1, label='Low Frequency')

freq2=12 #12Hz
period2 = 1/freq2
noise_amplitude = 0.5
sig2 = noise_amplitude*(np.sin(2*np.pi/period2*time_vec))
plt.figure(figsize=(60,10))
plt.plot(time_vec, sig2, label='High Frequency')

freq3=7 #7Hz
period2 = 1/freq2
noise_amplitude = 0.5
sig3 = noise_amplitude*(np.sin(2*np.pi/period2*time_vec))
plt.figure(figsize=(60,10))
plt.plot(time_vec, sig2, label='Mid Frequency')
```

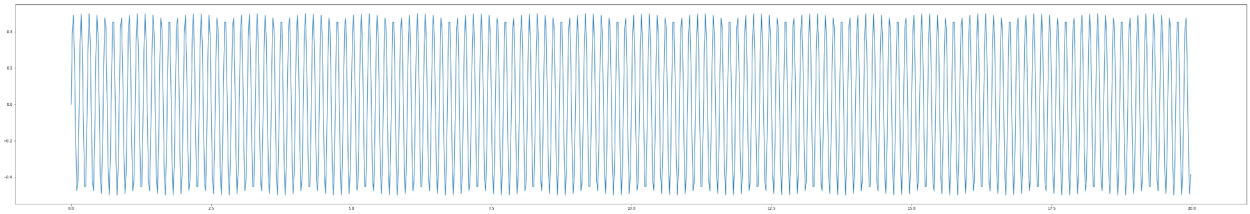
This will output 3 different frequencies to get a view of how each frequency looks before being combined.



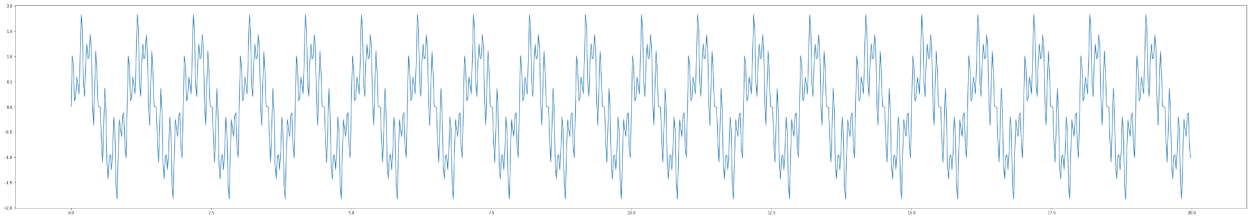
Freq1



Freq2



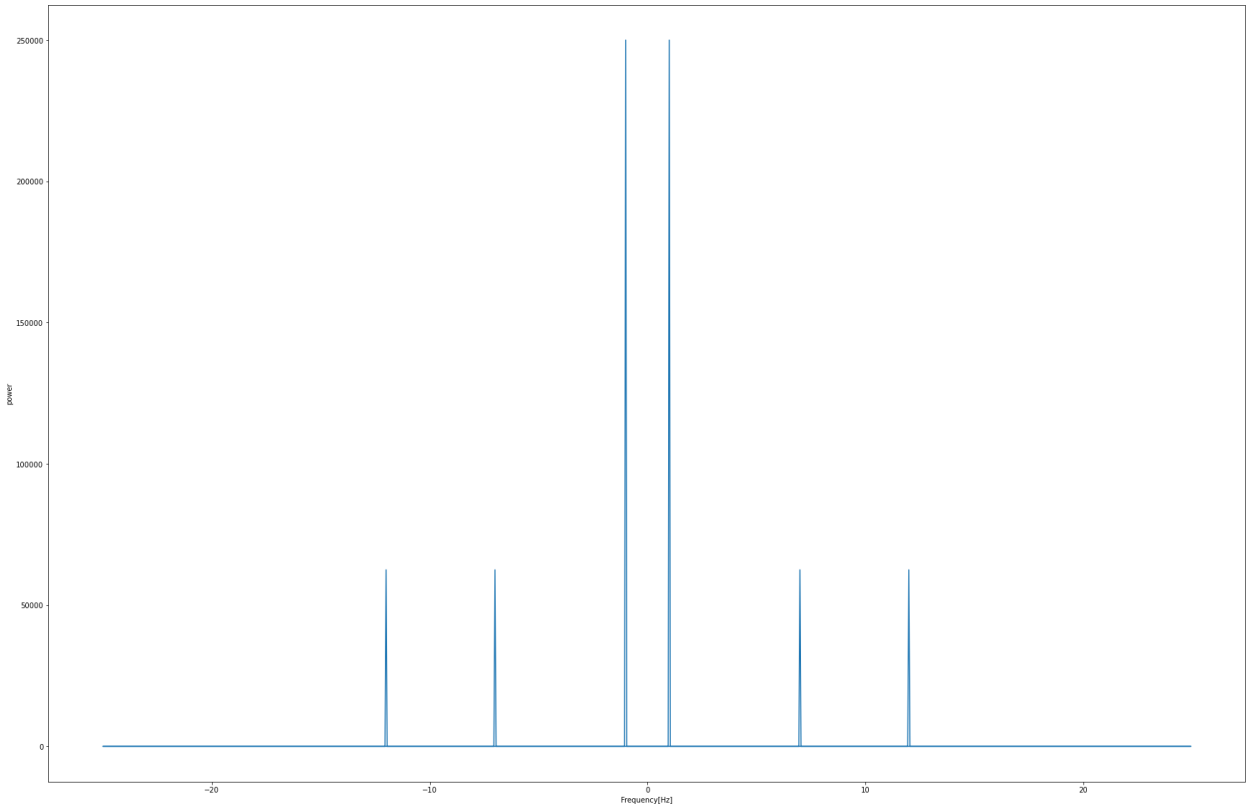
Freq3



Combined frequencies

After the signals are combined and plotted, the signal in the frequency spectrum will display the frequencies that were entered. This can give a general idea of what frequencies that can be

filtered.

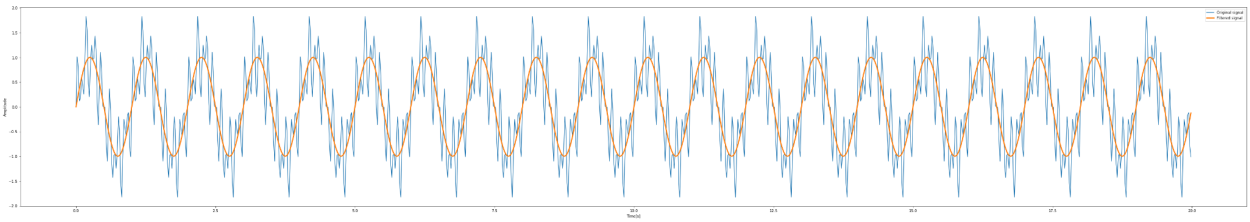


Signal in Frequency domain

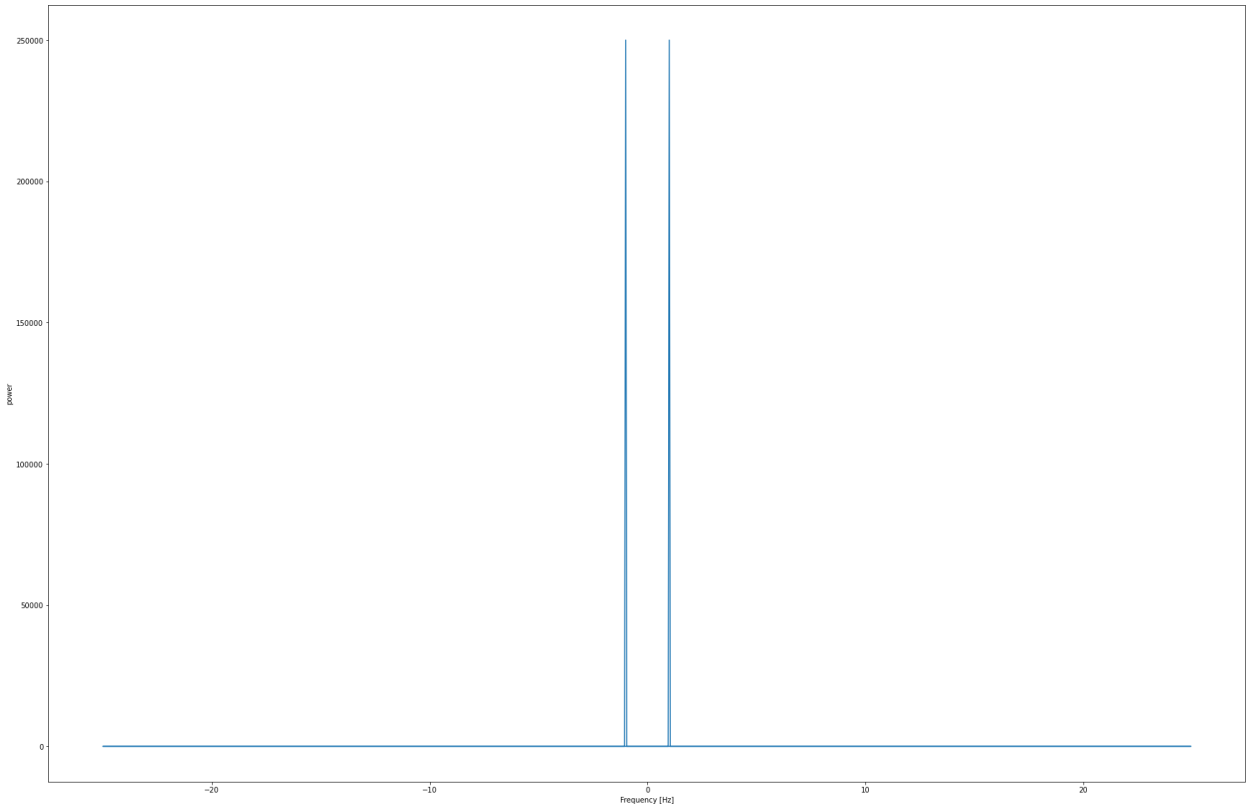
The frequency spectrum that you want to be filtered can be changed by changing the ‘peak_freq’ to the frequency to be filtered. Then change the range for the ‘high_fre_fft’ in line 67.

```
59  #Find the peak frequency
60  pos_mask = np.where(sample_freq > 0)
61  freqs = sample_freq[pos_mask]
62  peak_freq = freqs[power[pos_mask].argmax()]
63  peak_freq
64
65  #Remove Frequencies
66  high_freq_fft=sig_fft.copy()
67  high_freq_fft[np.abs(sample_freq) > peak_freq]=0
68  filtered_sig = fftpack.ifft(high_freq_fft)
69
```

After doing so, the frequency will be filtered and plotted again to show the filtered frequency and proof of the frequencies being filtered.



Original vs Filtered high frequency signal



Signal check for filtered high frequencies

Demonstration video:https://youtu.be/y0w8qaLFJpI

Heart Rate Analysis-Time Domain Measurements

Before running this code be sure to include these libraries before running the code.

import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import math

from scipy.interpolate import interp1d

This code does not need to be altered in anyway. If different data wants to observed, then change the csv data title for the ‘dataset’ value in line 142 .

Demonstration video:https://youtu.be/wsqJ1gwi_i4

Game Development - Red Alert

Before running this code be sure to include these libraries before running the code.

```
import pgzrun
import pygame
import pgzero
import random
from pgzero.builtins import Actor
from random import randint
```

Simple run the code and the game will begin. The objective is to click on the redstar before it reaches the bottom. The stars will shuffle and speed up as time goes on. There is a retry option for when you fail or succeed if the game wants to be tried again. To do so just press the space bar and the game will restart.

Demonstration video:https://youtu.be/SGJ20vcD0eQ