MileStone 1 - Project Report P-23

Mohamed Ayman Bahy 52-1096

Seif Basiouny 52-0314

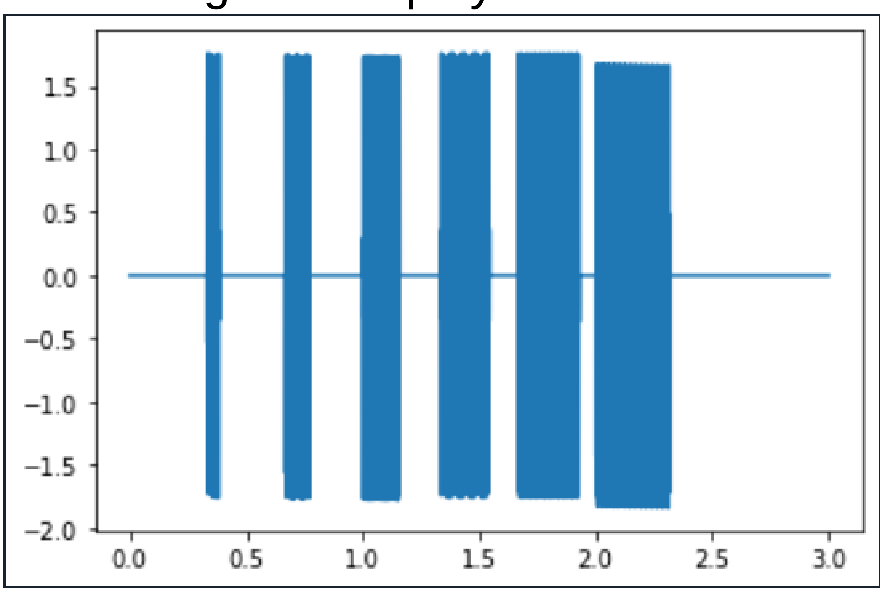
- Save the frequencies (F and f) in an array so they can easily be accessed by an index instead of having to write them each time .
- Make a for loop from range 0 to 7 (because we have 7 different frequencies) where i is the counter of the for loop
- Let ti (the pressing starting time) be = i/3
- Let Ti (how long you press) be = i/19
- The normal u[t] is 1 at t>=0 , so it makes sense that u[t-ti] is 1 at t>=ti , also u[t-ti-Ti] is 1 at t>=ti+Ti
- Generate the summation using :

  x1=np.where(t>=(i/3),1,0)

  x2=np.where(t>=((i/3)+(i/19)),1,0)

  x=x+((np.sin(2*p*F[i]*t)+np.sin(2*p*f[i]*t))*(x1-x2))
  where p is np.pi and t is np.linspace(0,3,12*1024)

- Plot the figure and play the sound



MileStone 2 - Project Report P-23

- Create a 6 figures sub plot
- Plot the original song of MileStone 1 in figure 1
- Convert the song to the frequency domain using the function fft from scipy.fftpack

    xf = fft(x)

    xf = 2/n * np.abs(xf[0:int(n/2)])

    where n is 3*1024

    and f is np.linspace(0,512,int(n/2))
- Plot the song in the frequency domain in figure 2
- Generate 2 noise frequencies using 2 random frequencies

    f1 = np.random.randint(0,512,2)

    noise = np.sin(2*f1[0]*p*t)+np.sin(2*f1[1]*p*t)
- Now add the noise to the original song (in the time domain)
- Plot the (song + noise) in time domain in figure 3
- Convert the (song + noise) into the frequency domain
- Plot the (song + noise) in frequency domain in figure 4
- Use a for loop and loop on the (song + noise) in the frequency domain and save the index of the maximum frequency in a variable
- Loop again to get the index of the second maximum frequency and also save it into a variable
- Get the 2 noise frequencies from f using the 2 saved indexes
- Round the 2 noise frequencies to the nearest integer
- Generate the noise using the 2 rounded up frequencies

    m = np.sin(2*int(f[ind1])*p*t) + np.sin(2*int(f[ind2])*p*t)
- Subtract the noise from the (song + noise) to get the filtered song
- Plot the filtered song in the time domain in figure 5
- Plot the filtered song in the frequency domain in figure 6

The final output figures will look like the following :