EE-679 Assignment 1B

Soham Naha
Roll : 193079003

September 27, 2020

Q. Use your previous synthesized vowel /u/ at two distinct pitches (F0 = 120 Hz, F0 = 220 Hz). Keep the bandwidths constant at 100 Hz for all formants.
Vowel F1, F2, F3 /u/ 300, 870, 2240
We would like to use the DFT computed with various window lengths and shapes to estimate the vowel's F0 and formant frequencies and study the obtained accuracies with reference to our 'ground truth' values. For the analysis, use a single waveform segment near the centre of your synthesized vowel. Plot the magnitude (dB) spectrum with rectangular and Hamming windows of lengths: 5 ms, 10 ms, 20 ms, 40 ms, each with a large zero-padded DFT.

- **Comment on the similarities and differences between the different computed spectra.**
- **Estimate the signal parameters from each of the magnitude spectra and report the error with respect to the ground-truth.**

Solution

```python
# initial package imports
import numpy as np
from scipy.signal import zpk2tf,freqz,sawtooth,square,impulse
from scipy.fft import fft,fftfreq
from math import pi
from numpy import exp,zeros_like,cos,sin,log10,angle,hamming
from numpy import convolve as conv

# to make the plots more TEX-like
import matplotlib
matplotlib.use('PS')
import pylab as plt
plt.switch_backend('PS')

plt.rcParams['text.usetex'] = True
plt.rcParams['text.latex.unicode']=True
plt.style.use(['bmh'])
plt.rcParams['font.size'] = 10
plt.rcParams['axes.labelsize'] = 10
#plt.rcParams['axes.labelweight'] = 'bold'
plt.rcParams['xtick.labelsize'] = 8
plt.rcParams['ytick.labelsize'] = 8
plt.rcParams['legend.fontsize'] = 10
plt.rcParams['figure.titlesize'] = 12
plt.rcParams.update({"axes.facecolor" : "white",
                     "axes.edgecolor":  "black"})


def generate_signal_response(t,sig,b,a):
    """Generates the excitation signal response from the signal and filter
        coefficients using the difference equation
    inputs: t (time-vector of the excitation signal)
            sig (excitation signal)
            b,a (filter numerator and denominator coefficients)
    output: returns the filter excitation response
    """
```

```
    y = zeros_like(sig)
    # difference equation
    for n in range(len(sig)):
        for k in range(len(b)):
            if (n-k)>=0:
                y[n] += b[k] * sig[n-k]
        for k in range(1,len(a)):
            if (n-k)>=0:
                y[n] -= a[k] * y[n-k]
    return y

def hamming_window(win_length,fs,output_signal,vowel,f0):
    """Calculates the rectangular window operation and DFT of windowed signal
    """
    window_size = int(win_length*fs/1000)
    window_signal = output_signal[:window_size] * hamming(window_size)
    dft = fft(window_signal, n=1024)
    freq = fftfreq(dft.shape[-1], 1/fs)
    s = "Hamming Window response of window length: {}ms  for
         vowel: /'{}'/ with signal freq: {}Hz"
    plt.figure()
    plt.title(s.format(win_length,vowel,f0) ,fontsize=12,weight="bold")
    plt.plot(abs(freq),20*log10(abs(dft)),'b',linewidth=0.7)
    plt.ylabel(r"$|H(\Omega|$",fontsize=10)
    plt.xlabel(r"$\Omega$")
    plt.xlim(xmin=0)
    plt.grid("True")
    plt.tight_layout()
    plt.savefig("../plots/Hamming_Window_Freq_resp_"+vowel+"_"+str(f0)+"_"+
                str(win_length)+".png",bbox_inches="tight",pad=-1,format="png")

def rectangular_window(win_length,fs,output_signal,vowel,f0):
    """Calculates the rectangular window operation and DFT of windowed signal
    """
    window_size = int(win_length*fs/1000)
    window_signal = output_signal[:window_size]
    dft = fft(window_signal, n=1024)
    freq = fftfreq(dft.shape[-1], 1/fs)
    s = "Rectangular Window response of window length: {}ms for
         vowel: /'{}'/ with signal freq: {}Hz"
    plt.figure()
    plt.title(s.format(win_length,vowel,f0) ,fontsize=12,weight="bold")
    plt.plot(abs(freq),20*log10(abs(dft)),'b',linewidth=0.7)
    plt.ylabel(r"$|H(\Omega|$",fontsize=10)
    plt.xlabel(r"$\Omega$")
    plt.xlim(xmin=0)
    plt.grid("True")
    plt.tight_layout()
    plt.savefig("../plots/Rect_Window_Freq_resp_"+vowel+"_"+str(f0)+"_"+
                str(win_length)+".png",bbox_inches="tight",pad=-1,format="png")

def vocal_tract(formant_frequencies,f_sampling):
    """Given the formant frequencies calculates the numerator and denominator
    coefficients by convolving between the different formant frequencies
    inputs: formant_frequencies (list of the formant frequencies)
    outputs: numerator and denominator coefficients
    """
    global bw
    r = []
    theta = []
    ts = 1/f_sampling
    for i in formant_frequencies:
        r.append(np.exp(-pi*bw*ts)) #radius in z-plane
        theta.append(2*pi*i*ts) #angle in z-plane

```

```
    denom_coeffs = []
    num_coeffs = []
    convolved_a = 1
    for radius,angle in zip(r,theta):
        poles = [radius*exp(1j*angle),radius*exp(-1j*angle)]
        zeros = zeros_like(poles)
        b,a = zpk2tf(zeros,poles,k=1)
        num_coeffs.append(b)
        denom_coeffs.append(a)
        convolved_a = conv(convolved_a,a)

    denom_coeffs = zeros_like(convolved_a)
    denom_coeffs[0] = 1

    return denom_coeffs,convolved_a

def generate_vowels(formant_frequencies,bandwidth,signal_frequency,vowel,time,f_sampling,
                    window,win_length):
    ts = 1/f_sampling # sampling time
    num_samples = int(f_sampling*time) # total number of signal samples
    b,a = vocal_tract(formant_frequencies,f_sampling)

    t = np.linspace(0,time,num_samples)
    # sawtooth approximation using square
    sig = square(2 * pi * signal_frequency* t, duty=0.01)+1

    response = generate_signal_response(t,sig,b,a)
    #plot_and_save_waveform(t,response,signal_frequency,f_sampling,vowel)
    if window=="hamming":
        hamming_window(win_length,f_sampling,response,vowel,signal_frequency)
    elif window=="rectangular":
        rectangular_window(win_length,f_sampling,response,vowel,signal_frequency)

f0 = [120,220]
formants = [300,870,2240]
vowel = "u"
duration = 0.5
fs = 16000
bw = 100

windows = ["hamming","rectangular"]
window_lengths = [5,10,20,40]

for sig_freq in f0:
    for window in windows:
        for win_len in window_lengths:
            generate_vowels(formants,bw,sig_freq,vowel,duration,fs,window,win_len)
```

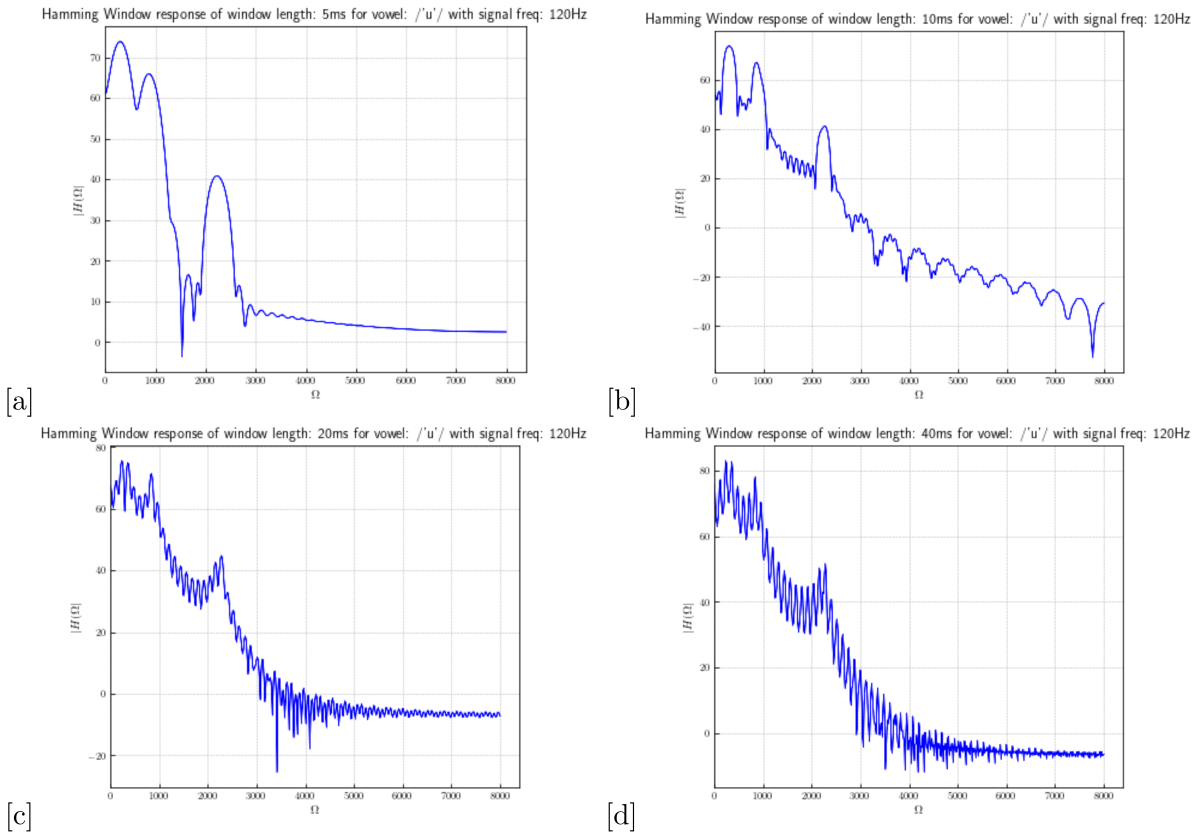

Figure 1: (a) Hamming Window Response for vowel /u/ with F_0 = 120Hz and Window Size=5ms (b) Hamming Window Response for vowel /u/ with F_0 = 120Hz and Window Size = 10ms (c) Hamming Window Response for vowel /u/ with F_0 = 120Hz and Window Size = 20ms (d) Hamming Window Response for vowel /u/ with F_0 = 120Hz and Window Size = 40ms

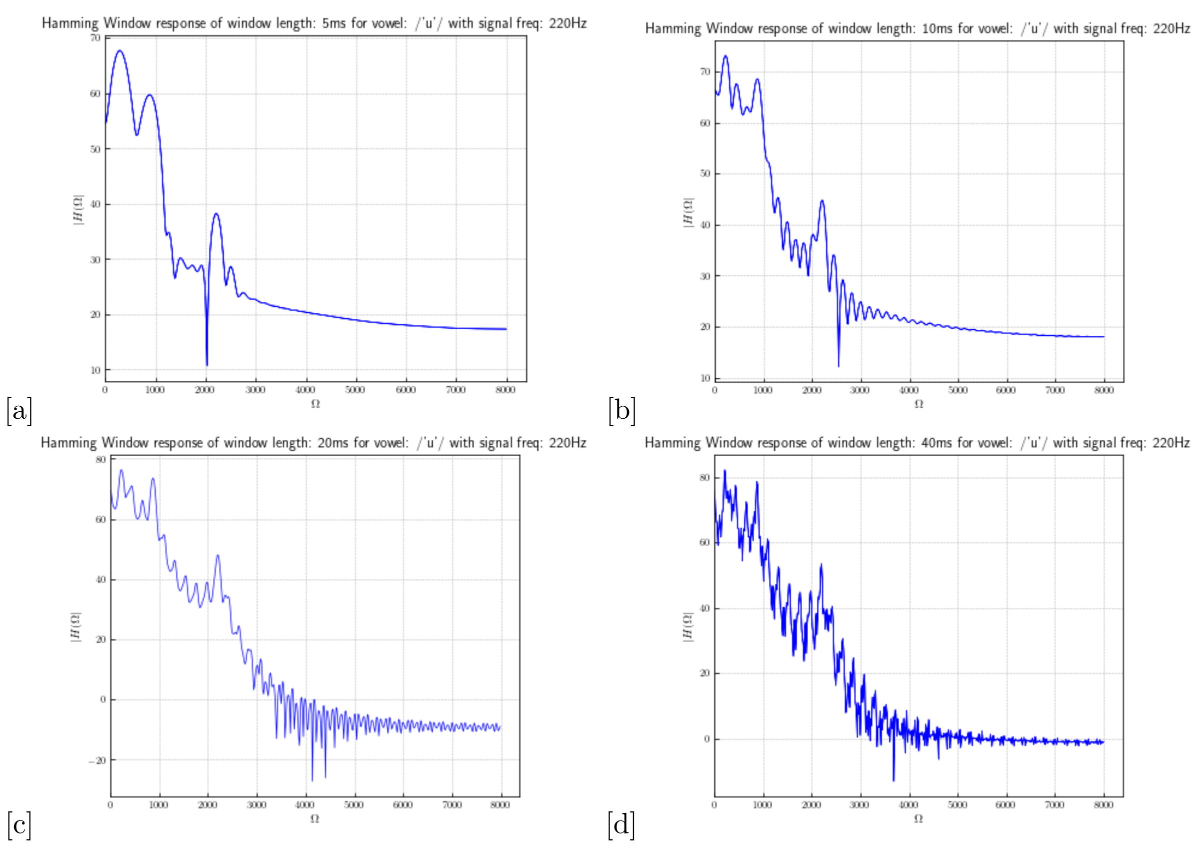

Figure 2: (a) Hamming Window Response for vowel /u/ with F_0 = 220Hz and Window Size=5ms (b) Hamming Window Response for vowel /u/ with F_0 = 220Hz and Window Size = 10ms (c) Hamming Window Response for vowel /u/ with F_0 = 220Hz and Window Size = 20ms (d) Hamming Window Response for vowel /u/ with F_0 = 220Hz and Window Size = 40ms

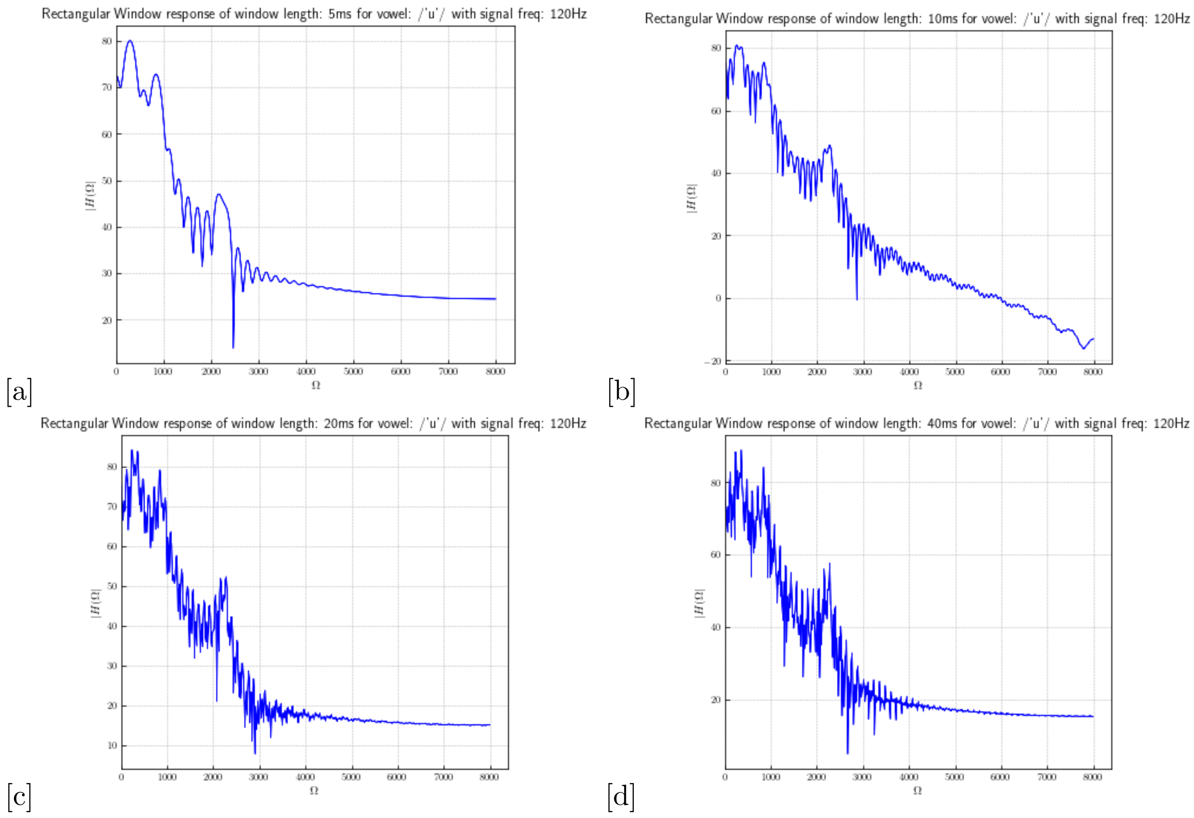

Figure 3: (a) Rectangular Window Response for vowel /u/ with $F_0 = 120$Hz and Window Size=5ms (b) Rectangular Window Response for vowel /u/ with $F_0 = 120$Hz and Window Size = 10ms (c) Rectangular Window Response for vowel /u/ with $F_0 = 120$Hz and Window Size = 20ms (d) Rectangular Window Response for vowel /u/ with $F_0 = 120$Hz and Window Size = 40ms

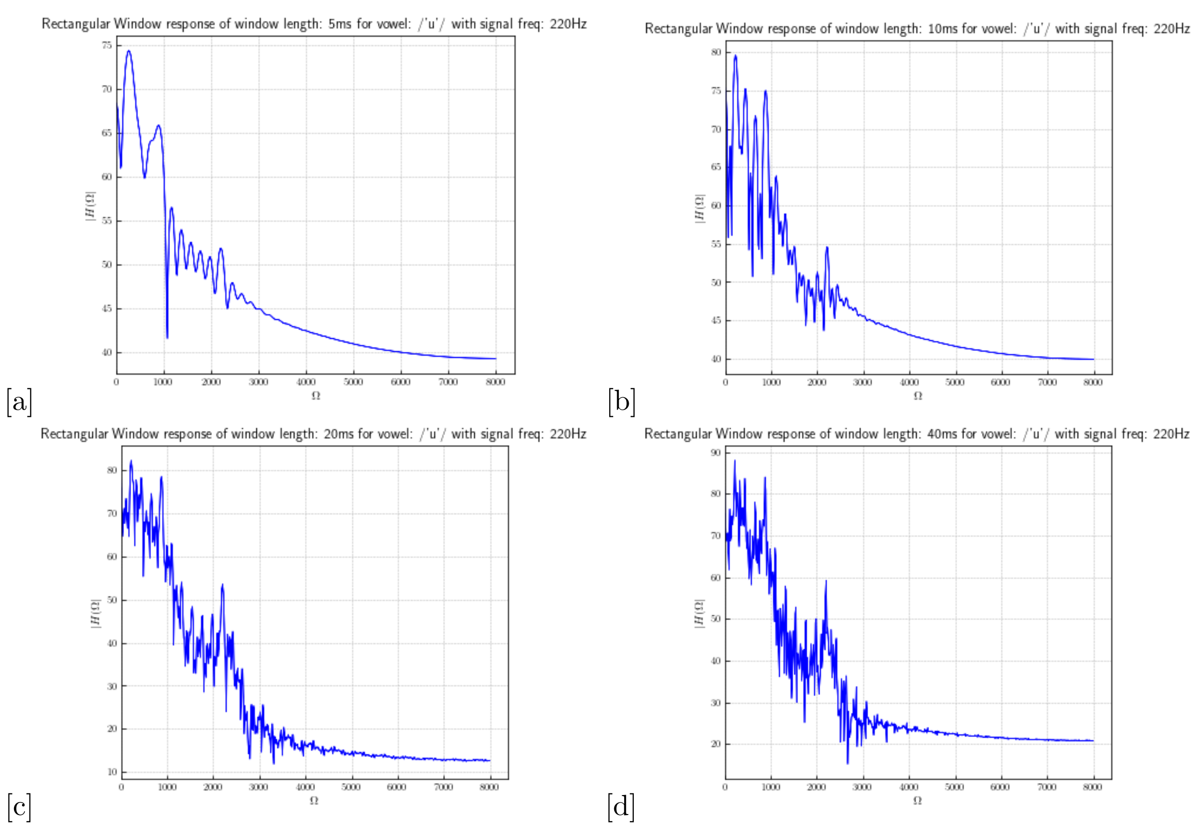

Figure 4: (a) Rectangular Window Response for vowel /u/ with $F_0 = 220$Hz and Window Size=5ms (b) Rectangular Window Response for vowel /u/ with $F_0 = 220$Hz and Window Size = 10ms (c) Rectangular Window Response for vowel /u/ with $F_0 = 220$Hz and Window Size = 20ms (d) Rectangular Window Response for vowel /u/ with $F_0 = 220$Hz and Window Size = 40ms

Ground Truth Frequency(Hz)		**Values Detected from Hamming Window Responses**							
		Window Lengths							
		5ms	**Difference**	**10ms**	**Difference**	**20ms**	**Difference**	**40ms**	**Difference**
F0 :	120 Hz	500 Hz	380 Hz	166 Hz	46 Hz	125 Hz	5 Hz	125 Hz	5 Hz
F1 :	300 Hz	300 Hz	0 Hz	295 Hz	5 Hz	246 Hz	54 Hz	239 Hz	61 Hz
F2 :	870 Hz	874 Hz	4 Hz	857 Hz	13 Hz	845 Hz	25 Hz	845 Hz	25 Hz
F3 :	2240 Hz	2231 Hz	9 Hz	2251 Hz	11 Hz	2281 Hz	41 Hz	2282 Hz	42 Hz
F0 :	220 Hz	500Hz	280 Hz	250 Hz	30 Hz	250 Hz	25 Hz	200 Hz	20 Hz
F1 :	300 Hz	300 Hz	0 Hz	234 Hz	66 Hz	231 Hz	69 Hz	222 Hz	78 Hz
F2 :	870 Hz	878 Hz	8 Hz	875 Hz	5 Hz	875 Hz	5 Hz	875 Hz	5 Hz
F3 :	2240 Hz	2222 Hz	18 Hz	2209 Hz	31 Hz	2202 Hz	38 Hz	2202 Hz	38 Hz

Table 1: Table for Comparison of Hamming Window

Ground Truth Frequency(Hz)		**Values Detected from Rectangular Window Responses**							
		Window Lengths							
		5ms	**Difference**	**10ms**	**Difference**	**20ms**	**Difference**	**40ms**	**Difference**
F0 :	120 Hz	125 Hz	5 Hz	125 Hz	5 Hz	125 Hz	5 Hz	125 Hz	5 Hz
F1 :	300 Hz	292 Hz	8 Hz	260 Hz	40 Hz	243 Hz	57 Hz	245 Hz	55 Hz
F2 :	870 Hz	851 Hz	19 Hz	843 Hz	27 Hz	846 Hz	24 Hz	849 Hz	21 Hz
F3 :	2240 Hz	2185 Hz	65 Hz	2270 Hz	30 Hz	2280 Hz	40 Hz	2285 Hz	45 Hz
F0 :	220 Hz	250 Hz	30 Hz	250 Hz	30 Hz	250 Hz	30 Hz	250 Hz	30 Hz
F1 :	300 Hz	288 Hz	12 Hz	231 Hz	69 Hz	224 Hz	76 Hz	224 Hz	76 Hz
F2 :	870 Hz	891 Hz	21 Hz	878 Hz	8 Hz	880 Hz	10 Hz	880 Hz	10 Hz
F3 :	2240 Hz	2205 Hz	35 Hz	2218 Hz	22 Hz	2206 Hz	34 Hz	2206 Hz	34 Hz

Table 2: Table for Comparison of Rectangular Window

General Observations

- If we increase the Window Size of the either the Hamming Window or the Rectangular Window, the clarity between the peak magnitudes increases, but identifying the formant frequencies (F_1, F_2 and F_3) becomes a bit challenging, due to peaks occurring in close proximity. (This portrays the fact that increasing the window size, means going from wide-band to narrow-band.)
- If we increase the Window Size of the either the Hamming Window or the Rectangular Window, calculating the fundamental frequency F_0 becomes a bit easy. It is calculated by dividing 1 KHz frequency by the number of high peaks occurring from 0 to 1 KHz.
- For small window sizes, the formant frequencies are calculated easily. Due to wideband nature of the frequency response, the peaks get accumulated to give the exact formant frequency.
- As the fundamental frequency F_0 increases, the resonating frequency of the vocal tract changes as well. This also affects the detection of the formant frequencies. For, F_0=220Hz, we see that F_2 and F_3 being near to the multiples of 220Hz are visualized properly, but F_1 not being near to the multiple of 220Hz, has a peak elsewhere. Similar for F_0=120Hz.
- As expected from theoretical concepts, the side lobes of the Hamming Windowed signal are quite lower than that in those where Rectangular window was applied. For a given window size, the peaks are more frequently occuring and narrowly spaced in Rectangular window setting.
- For the same Window Function, if the fundamental frequency F_0 is changed the peaks are more spaced out and the details of the waveform are more clear in the case of higher F_0.