Assignment 1B Speech Processing

Aditya Khanna 17D070036
Faculty mentor : Prof. Preeti Rao

28th September 2020

1 Question 1

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

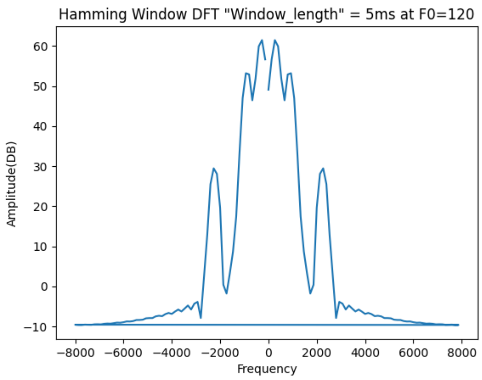


Figure 1: Flower one.

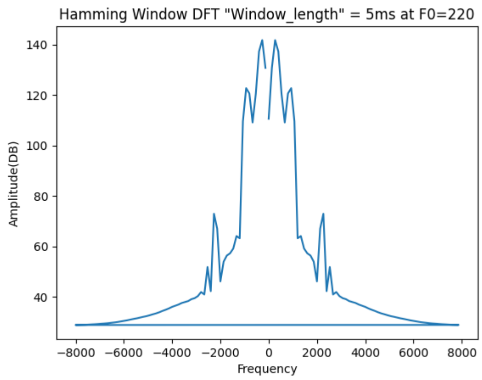
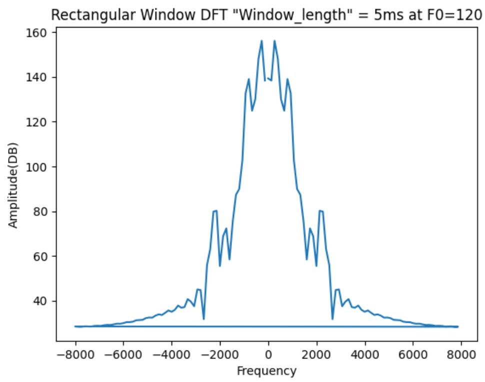
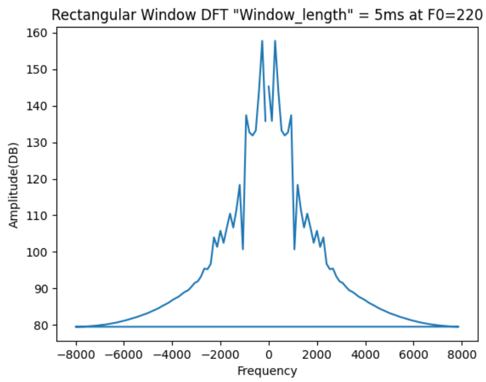
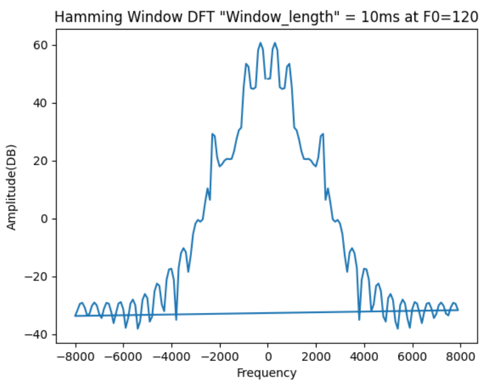
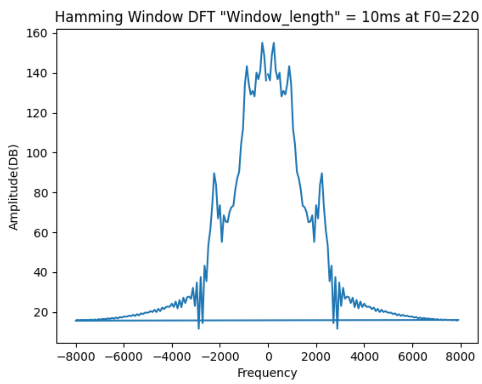


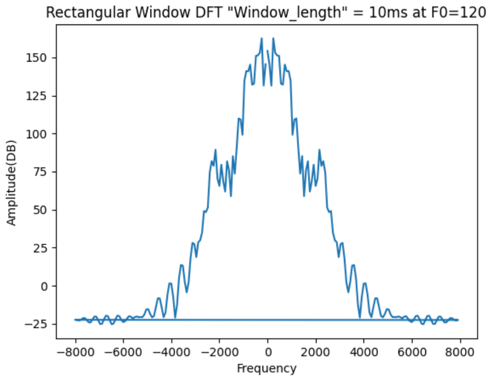
Figure 2: Flower two.

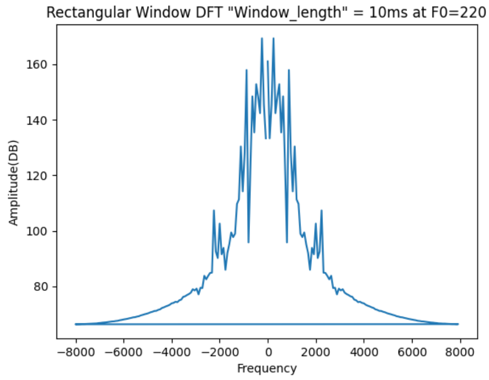


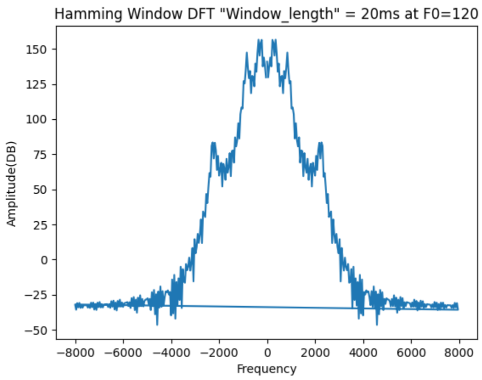


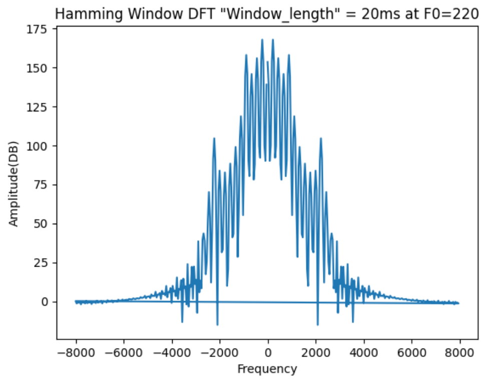


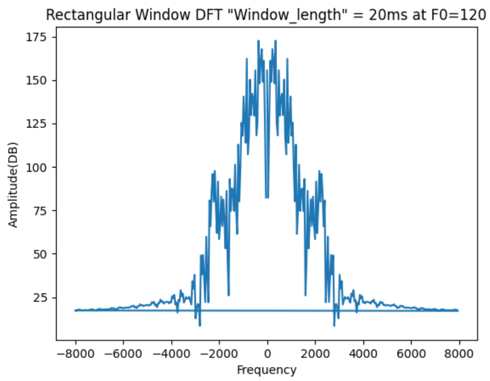


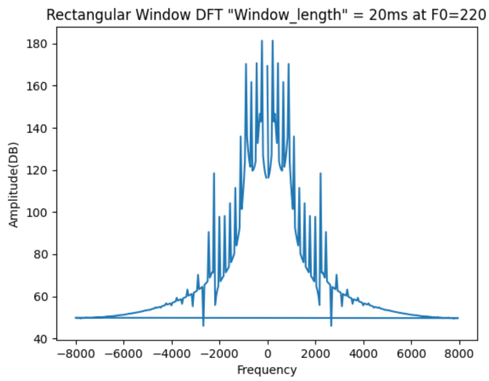

Rectangular Window DFT "Window_length" = 10ms at F0=120
Amplitude(DB)
Frequency

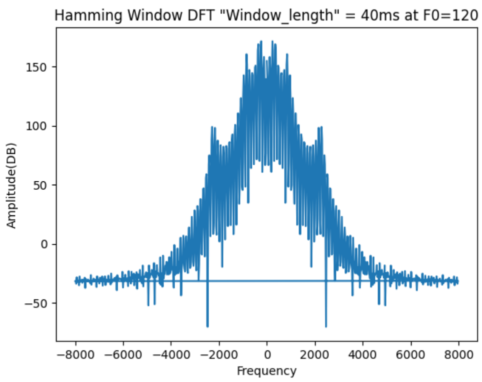

Rectangular Window DFT "Window_length" = 10ms at F0=220
Amplitude(DB)
Frequency

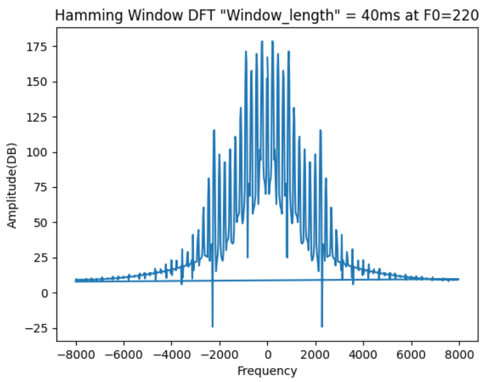

Hamming Window DFT "Window_length" = 20ms at F0=120
Amplitude(DB)
Frequency

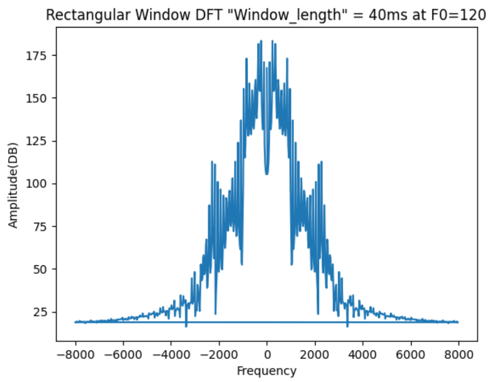

Hamming Window DFT "Window_length" = 20ms at F0=220
Amplitude(DB)
Frequency

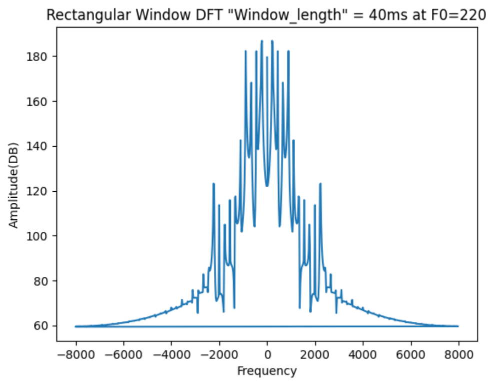

Rectangular Window DFT "Window_length" = 20ms at F0=120
Amplitude(DB)
Frequency


Rectangular Window DFT "Window_length" = 20ms at F0=220
Amplitude(DB)
Frequency


Hamming Window DFT "Window_length" = 40ms at F0=120
Amplitude(DB)
Frequency


Hamming Window DFT "Window_length" = 40ms at F0=220
Amplitude(DB)
Frequency





With the increase in window length, the spectrum becomes smoother.As the window length is more, the pitch of the original signal is preserved more and therefore we get more original signal.With the rectangular windows, we see a much rougher waveform as compared to the hamming windows. The Hamming windows are a better approximation to get the sounds. As the frequency increases, the spectrum also becomes more clearer. We can see that the spectra gets smoother with decreasing window length. The errors in F1, F2 and F3 are more in the rectangular Window from the measured and real values. As the Window size increases, it becomes more difficult to get the exact peaks since the value has a lot of deviations.

1.1 $\mathbf{F}_0 = 120Hz$

1.1.1 Rectangular

Window Length = 5; F1= 262 ; F2= 817 ; F3= 2178
Window Length = 10; F1= 252 ; F2=852 ; F3= 2230
Window Length = 20; F1=270 ; F2=851 ; F3=2232
Window Length = 40; F1= 295 ; F2= 863 ; F3= 2209
Original Values F1=300; F2=870; F3=2240

1.1.2 Hamming

Window Length = 5; F1= 310 ; F2= 833 ; F3= 2249
Window Length = 10; F1= 286 ; F2= 851 ; F3= 2235
Window Length = 20; F1= 289 ; F2= 859 ; F3= 2218
Window Length = 40; F1= 299 ; F2= 862 ; F3= 2259
Original Values F1=300; F2=870; F3=2240

1.2 $\mathbf{F}_0 = 220Hz$

1.2.1 Rectangular

Window Length = 5; F1= 270 ; F2= 942 ; F3= 2115
Window Length = 10; F1= 250 ; F2=886 ; F3= 2235
Window Length = 20; F1=245 ; F2=890 ; F3=2231
Window Length = 40; F1= 278 ; F2= 902 ; F3= 2216
Original Values F1=300; F2=870; F3=2240

1.2.2 Hamming

Window Length = 5; F1= 277 ; F2= 926; F3= 2303
Window Length = 10; F1= 287 ; F2= 902 ; F3= 2198
Window Length = 20; F1= 258 ; F2= 869 ; F3= 2308
Window Length = 40; F1= 289 ; F2= 903 ; F3= 2249
Original Values F1=300; F2=870; F3=2240

```python
import numpy as np
import math
import matplotlib.pyplot as plt
from scipy import signal
from scipy.io.wavfile import write


B_1=100
playingtime=0.5
F_0=120
F_s=16000
Window_length = 40/1000

x=[]
y=[]
y_1=[]
y_2=[]
time=[]
total_samples=int(F_s*playingtime)

for i in range(0,total_samples):
  x.append(0)
  y.append(0)
  y_1.append(0)
  y_2.append(0)
  time.append(i)


i=0
count=0

while i<total_samples and count < int(F_0*playingtime):
  x[i]= 1;
  i=i+int(F_s/F_0);
  count+=1

# Input Parameters
F_1=300
F_2=870
F_3=2240


def response(x,y,F):
  r_i = np.exp(-B_1*3.142/F_s)
  theta_i = 2*3.142*F/F_s
  denominator=[1,-2*r_i*math.cos(theta_i),r_i*r_i]
  y[0]=x[0] # impulse
  y[1]= x[1]-y[0]*denominator[1]
  for i in range(2,total_samples):
    y[i] = x[i]-y[i-1]*denominator[1]-y[i-2]*denominator[2];
  return y


y=response(x,y,F_1)
y_1=response(y,y_1,F_2)
y_2=response(y_1,y_2,F_3)

por=int(Window_length*F_s)
# final=y_2[:por]*np.hamming(por)
final=y_2[:por]*np.ones(por)
final = np.pad(final, (20, 20), 'constant')
ft = np.abs(np.fft.fft(final))
t = np.fft.fftfreq(ft.shape[0], 1/F_s)

plt.title('Rectangular Window DFT "Window_length" = 40ms at F0=120')
plt.plot(t, 20*np.log(ft))
plt.ylabel('Amplitude(DB)')
plt.xlabel('Frequency')
plt.show()
plt.savefig('Rectangular Window DFT "Window_length" = 40ms at F0=120.png')

# plt.title('Hamming Window DFT "Window_length" = 40ms at F0=120')
# plt.plot(t, 20*np.log(ft))
# plt.ylabel('Amplitude(DB)')
```

```
# plt.xlabel('Frequency')
# plt.savefig('Hamming Window DFT "Window_length" = 40ms at F0=120.png')

```