BRNO UNIVERSITY OF TECHNOLOGY
Faculty of information technology

ISS - Signals and Systems
2020 / 2021

Project protocol

Author: Elena Carasec
Login: xcaras00

Brno
4th January, 2021

1 Tones

Name	Length [samples]	Length [seconds]
maskon_tone.wav	26624	00:00:01.66
maskoff_tone.wav	26624	00:00:01.66

2 Sentences

Name	Length [samples]	Length [seconds]
maskon_sentence.wav	87040	00:00:05.44
maskoff_sentence.wav	79872	00:00:04.99

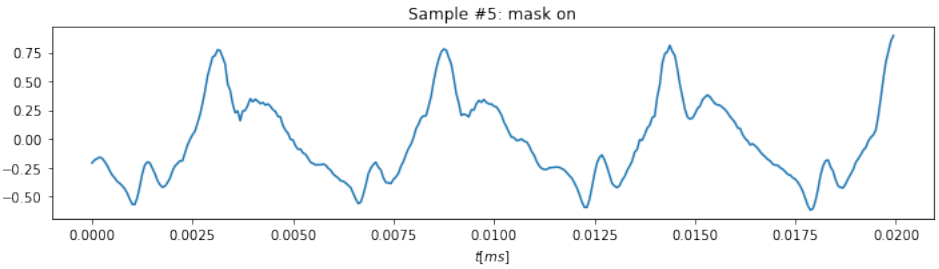
$F_s = 16000$ Hz

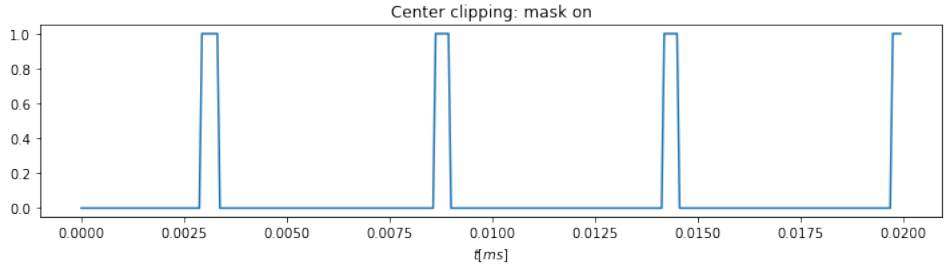
3 Segments' extraction

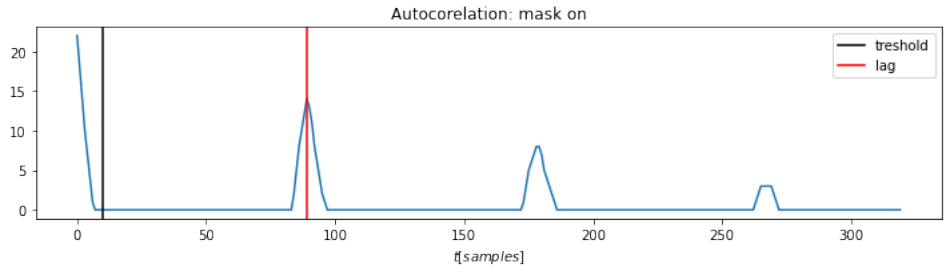
They have been chosen manually.

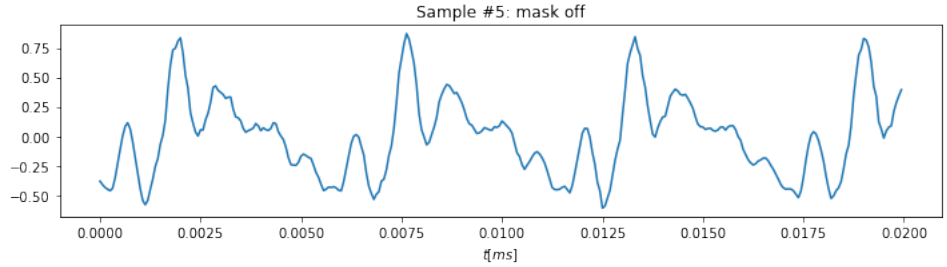
4 Main frequency

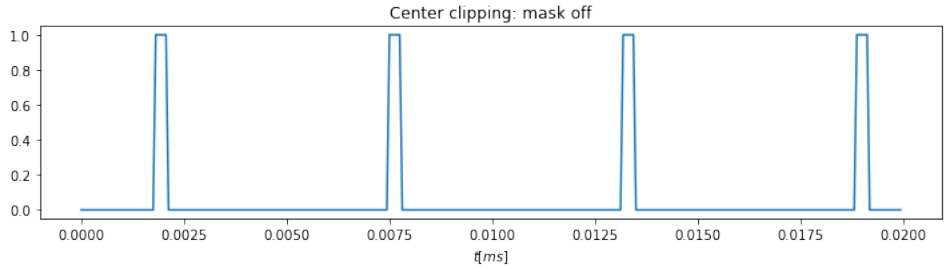
a)

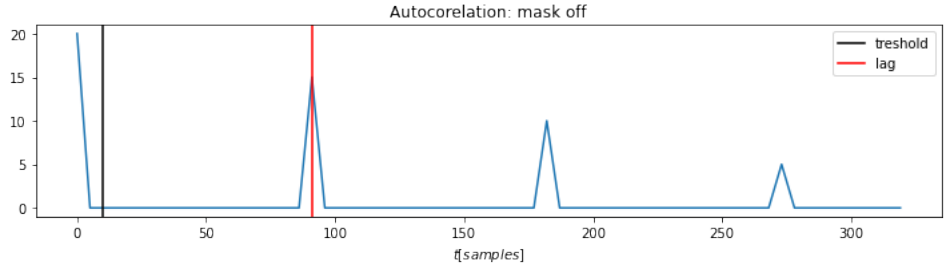


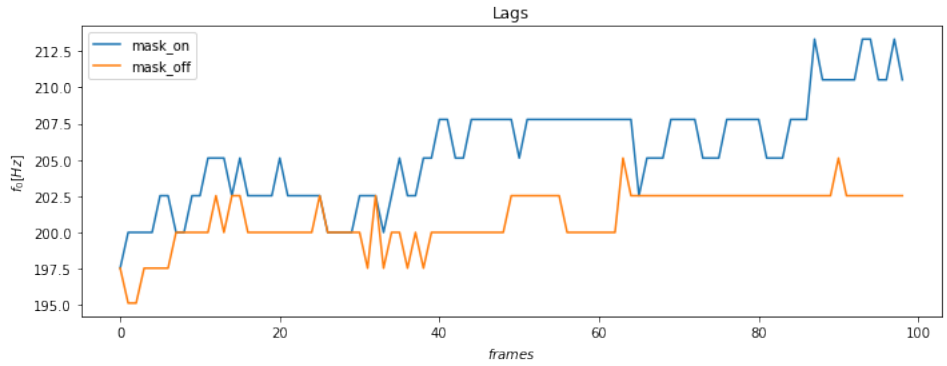

Center clipping: mask on
1.0
0.8
0.6
0.4
0.2
0.0
0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0.0150
0.0175
0.0200
t[ms]


Autocorelation: mask on
treshold
lag
20
15
10
5
0
0
50
100
150
200
250
300
t[samples]


Sample #5: mask off
0.75
0.50
0.25
0.00
−0.25
−0.50
0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0.0150
0.0175
0.0200
t[ms]


Center clipping: mask off
1.0
0.8
0.6
0.4
0.2
0.0
0.0000
0.0025
0.0050
0.0075
0.0100
0.0125
0.0150
0.0175
0.0200
t[ms]





b)

- Mask on:
  - Mean: 205.61351967987724
  - Variance: 12.21190170028492
- Mask off:
  - Mean: 200.9569069361186
  - Variance: 2.2409738588289705

c) To minimize the influence of error $\pm$ 1, we may choose another value of threshold or divide the initial signal into longer samples (for instance, 25 ms).

5 DFT

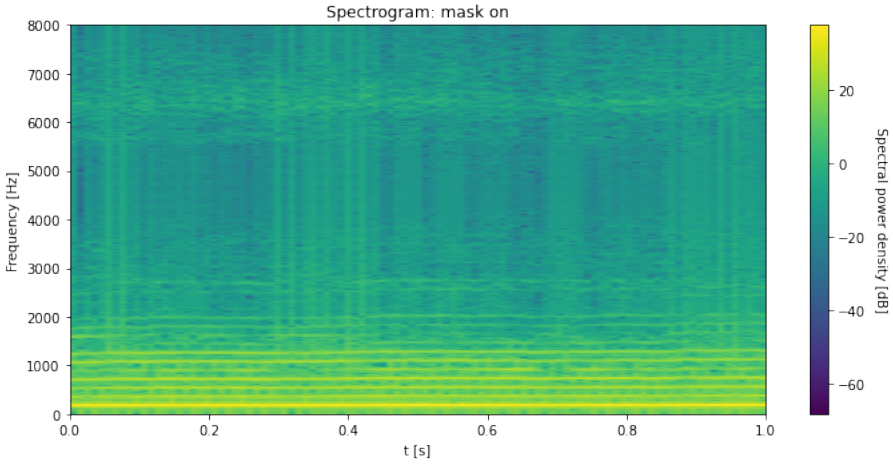
a)

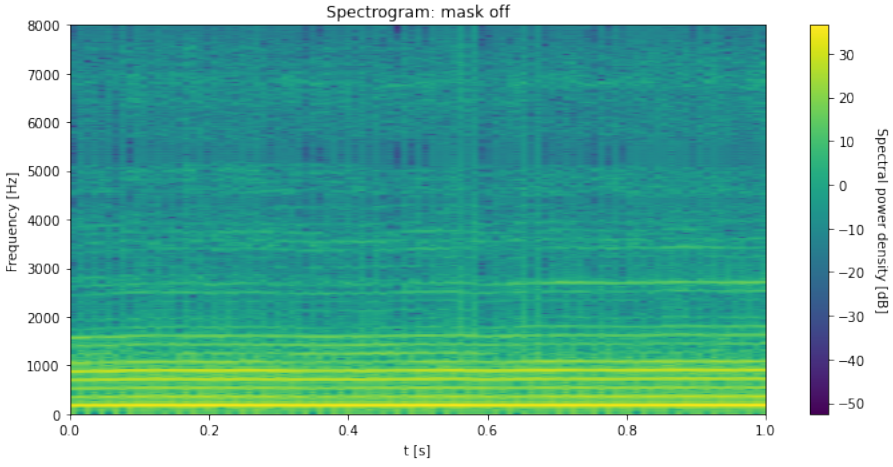
$$X_{DFT}[k] = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N} \qquad k = 0, 1, \cdots, N-1$$

```
import numpy as np
import cmath
```

```
def count_dft(tone):
    dft_arr = []
    for frame in tone:
        x_k = []
        for k in range(1024):
            x_dft = 0
            for n in range(320):
                x_dft += frame[n]*cmath.exp(-2j*cmath.pi*n*k/1024)
            x_k.append(x_dft)
        dft_arr.append(np.array(x_k))
    dft_arr = np.array(dft_arr)
    return dft_arr
```

b)



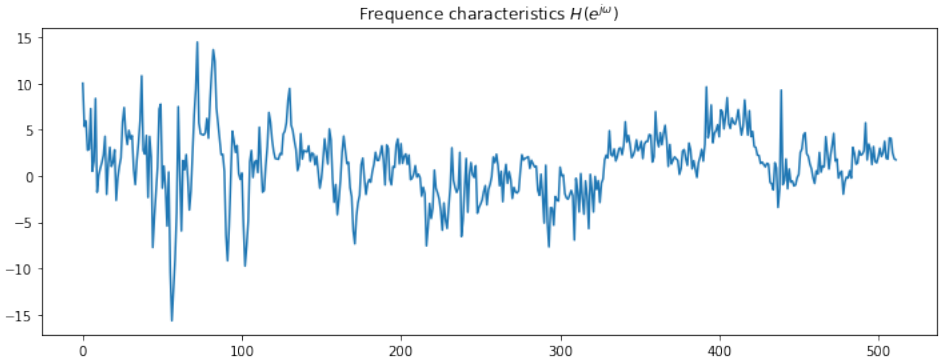


6 Frequency characteristics

a)

$$H(e^{j\omega}) = \frac{Y(e^{j\omega})}{X(e^{j\omega})} = \frac{DFT\_mask\_on}{DFT\_mask\_off}$$

b)



c) $N = 512$ was used for the filter, otherwise it simulates echo.

7 IDFT

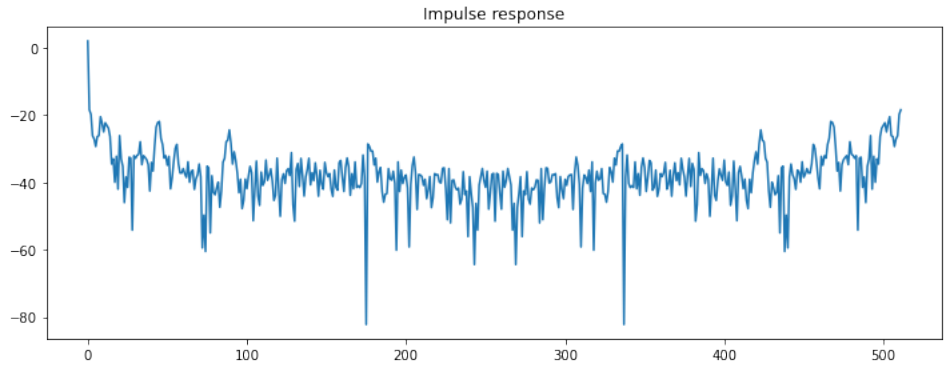
a)

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1} X_{DFT}[k]e^{j2\pi nk/N} \qquad n = 0,1,\cdots,N-1$$

```
import numpy as np
import cmath
```

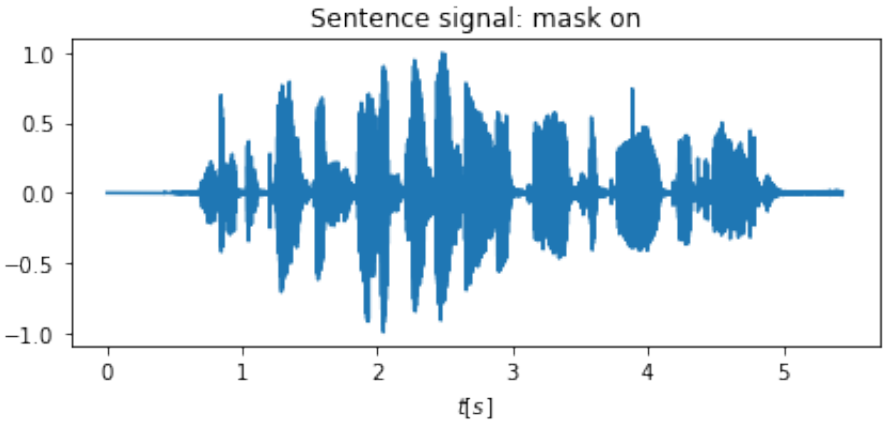
```
def count_idft(freq_medium):
    my_impulse_response = []
    for n in range(512):
        idft_sum = 0
        for k in range(512):
            idft_sum += freq_medium[k]*cmath.exp(2j*cmath.pi*n*k/512)/512
        my_impulse_response.append(idft_sum)
    my_impulse_response = np.array(my_impulse_response)
    return my_impulse_response
```

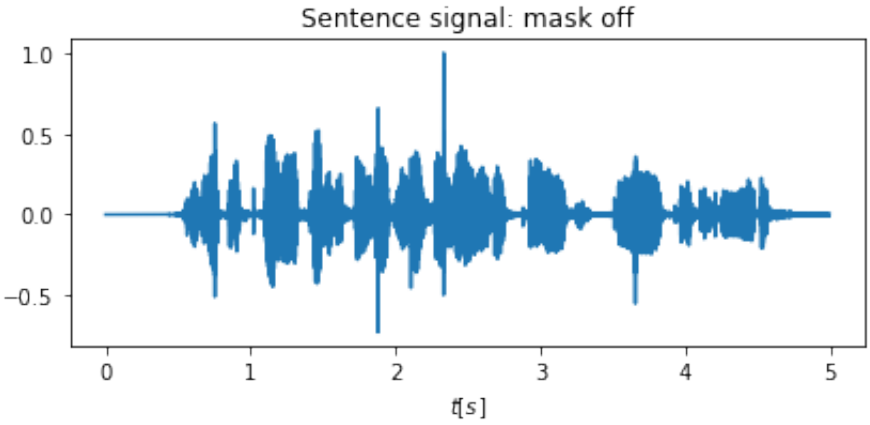
b)

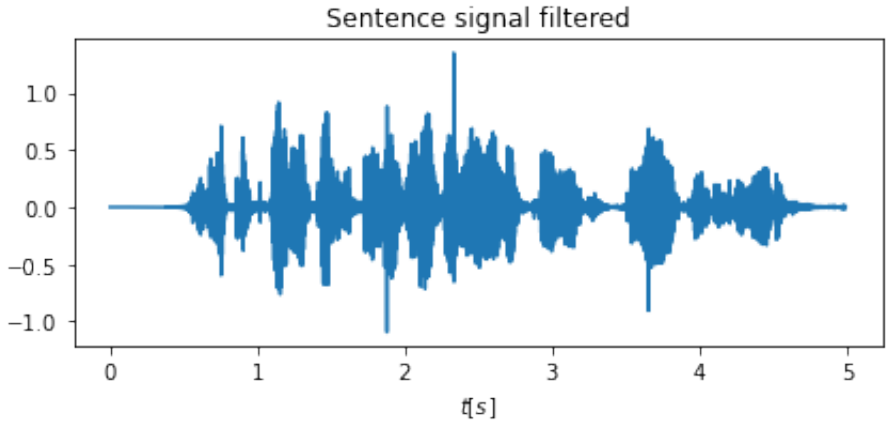


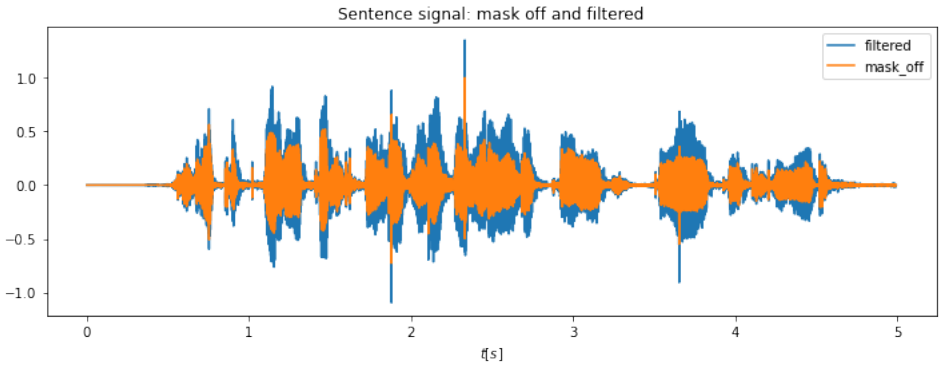
8 Mask simulation

a)


Sentence signal: mask on
1.0
0.5
0.0
−0.5
−1.0
0
1
2
3
4
5
t[s]


Sentence signal: mask off
1.0
0.5
0.0
−0.5
0
1
2
3
4
5
t[s]





b) Unlike the signal without mask, the signals with mask and with simulated mask sound like some noise was added, but the noise on the simulated signal is weaker.

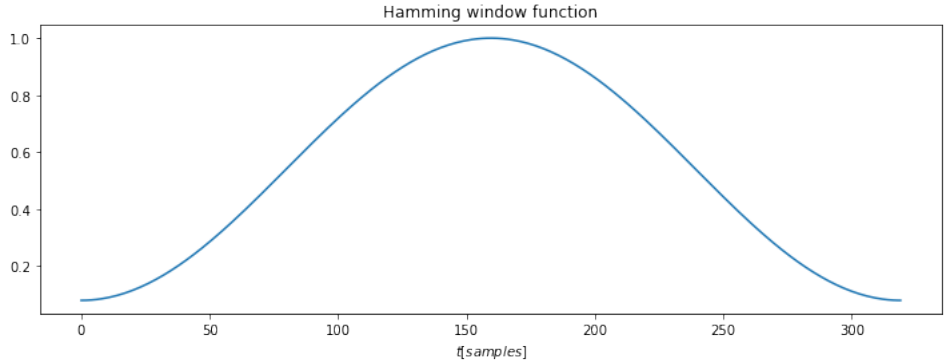
9 Conclusion

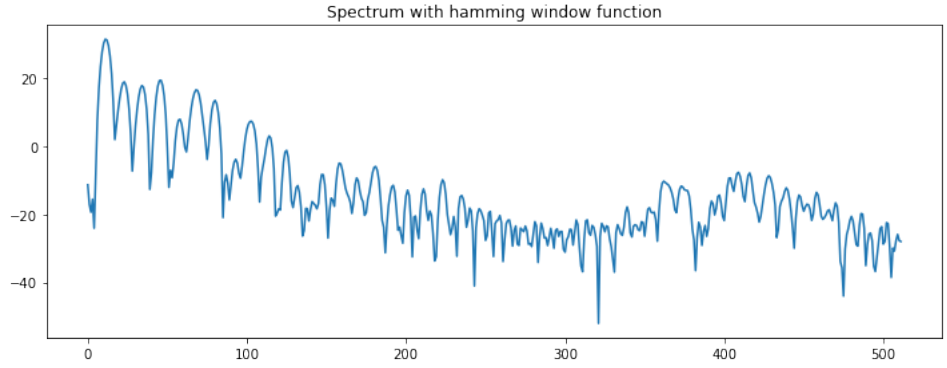
In conclusion I can say, that the filter is weaker than it should be, that is why the simulation does not work perfectly, nevertheless there is a difference between the sentence without mask and with simulated mask. It seems like a thin mask was simulated.

11 Window function

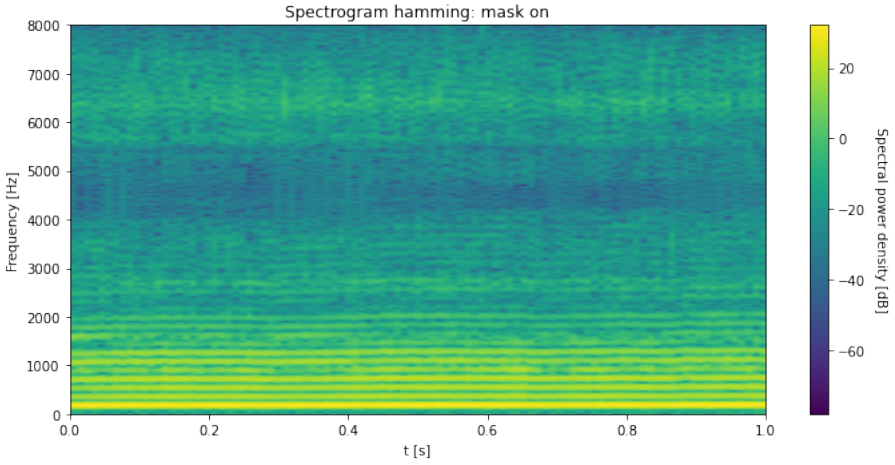
a) Hamming window function was used.

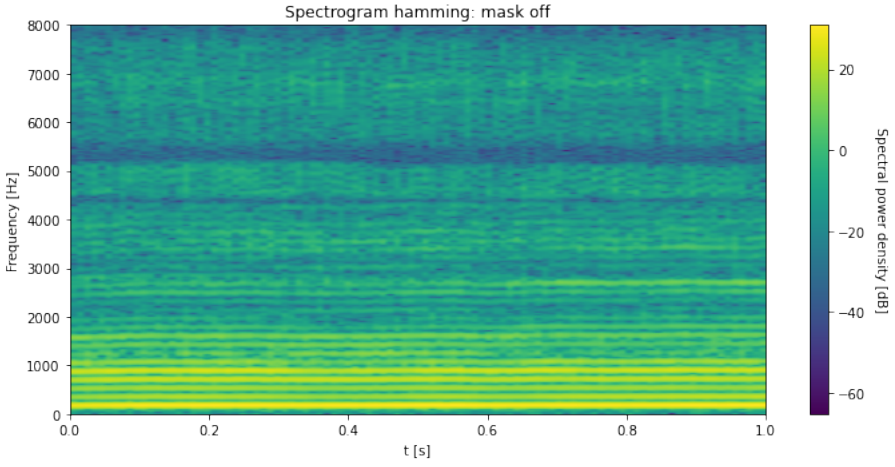
b)



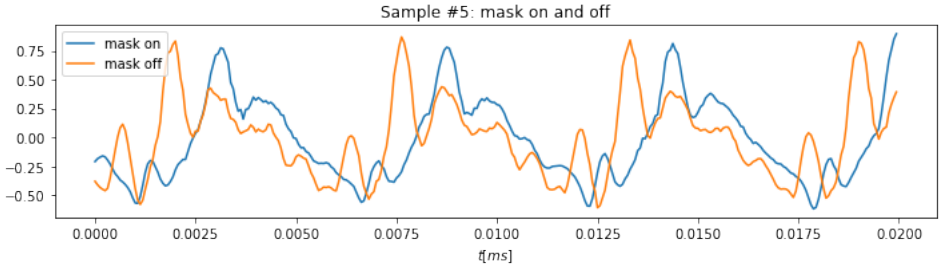


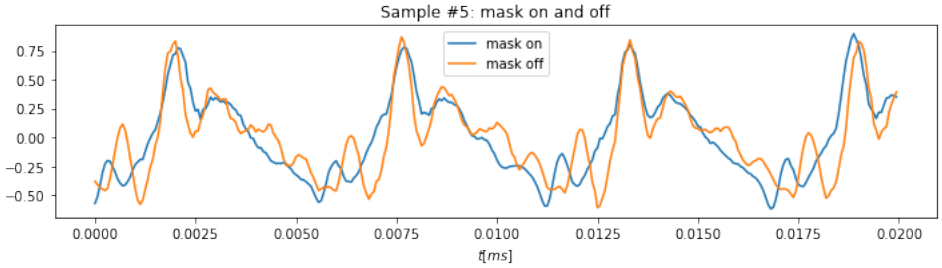
c) Hamming window function helps to improve signal clarity. Windowing reduces the amplitude of the discontinuities at the boundaries of each finite sequence acquired by the digitizer.

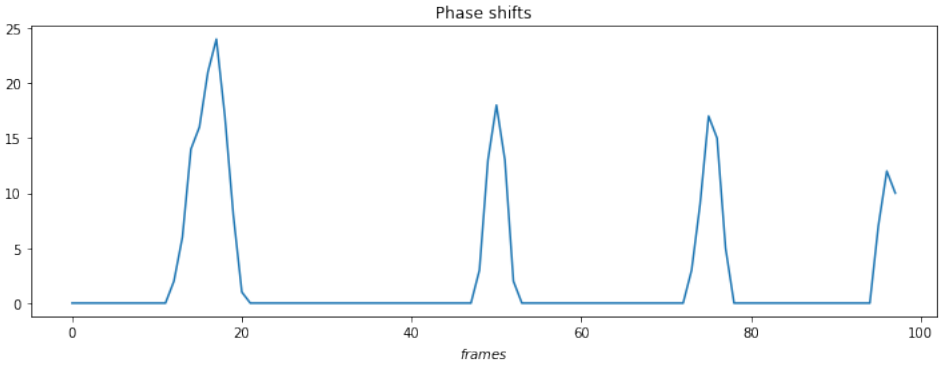

Spectrogram hamming: mask on
Frequency [Hz]
t [s]
Spectral power density [dB]
8000
7000
6000
5000
4000
3000
2000
1000
0
0.0
0.2
0.4
0.6
0.8
1.0
20
0
−20
−40
−60


Spectrogram hamming: mask off
Frequency [Hz]
t [s]
Spectral power density [dB]
8000
7000
6000
5000
4000
3000
2000
1000
0
0.0
0.2
0.4
0.6
0.8
1.0
20
0
−20
−40
−60

15 Phase shift







If we sum values of phase shifts for both sides, we are going to get a value similar to lag. Lets say, that start of the phase is the first signals' peak. Then the first value is "distance" between the first peaks of signals. And the

second value is "distance" between the first peak of the second signal and the second peak of the first signal. Thus, their sum is the "distance" between the first two peaks of the same signal, which is exactly a lag.

Reference

1. http://www.kiv.zcu.cz/∼mautner/Azs/Azs5_Fourierova_transformace.pdf

2. https://download.ni.com/evaluation/pxi/Understanding%20FFTs%20and%20Windowing.pdf