Experimental Assignment 2
Medical Imaging Techniques
(EECS4640/5640)

Professor Sadeghi-Naini

SeyedMostafa Ahmadi
(#219138924)

February 2023

Consideration

After getting professor Sadeghi's approval, I programmed this assignment using Python (3.6) instead of Matlab.
You can find all my codes in the directory "*codes*" for each question in the following format: *q_a.py* for the first question (or part a), *q_b.py* for the second question (or part b), etc.
To run the code for each item in a specific question, please uncomment it under the line :

```
if __name__ == "__main__":
```

in the corresponding file. For example, for running the code for ***question e item ii***, i.e. ***e-ii)***, please uncomment the following line in file ***q_e.py***:

```
# item_ii(rf=rf_file, fs=fs, lowcut=lowcut, highcut=highcut)
```

Question a

The calculation is simple. First, we have to calculate the sampling time. We know each line has 10,000 samples and the sampling frequency is 500 MHz. So,

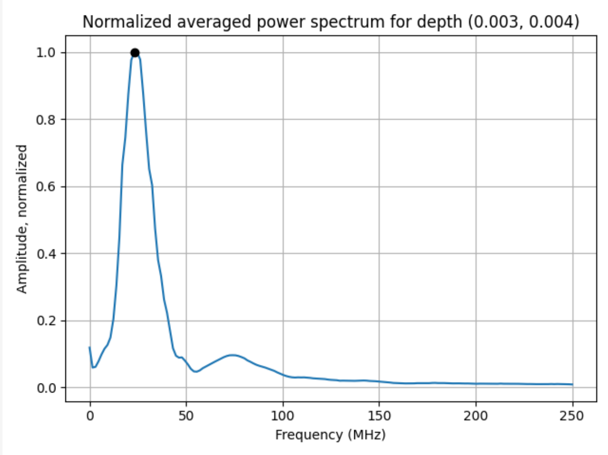
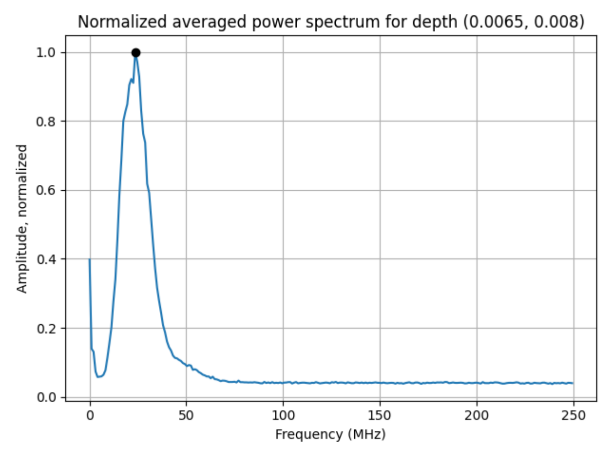
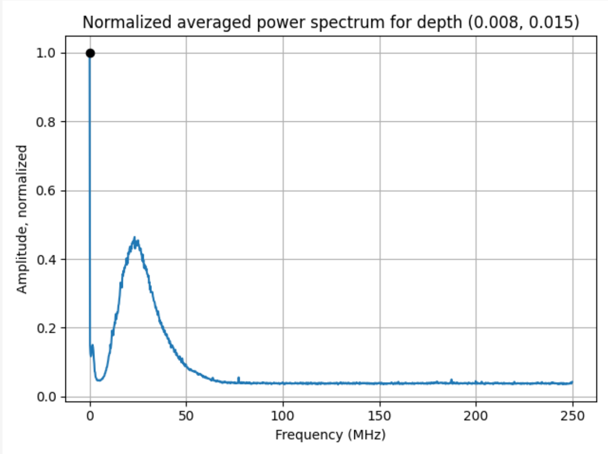
sampling time = $\dfrac{10{,}000}{(500 \times 10^6)} = 2 \times 10^-5$ sec

And using sampling time (above) and the speed of sound (≈ 1540 m/s), we have:

max depth = $(2 \times 10^-5) \times (1540) = 0.0308$ m ≈ 3 cm

(You can also find the calculations in python in *q_a.py* file.)

Question b

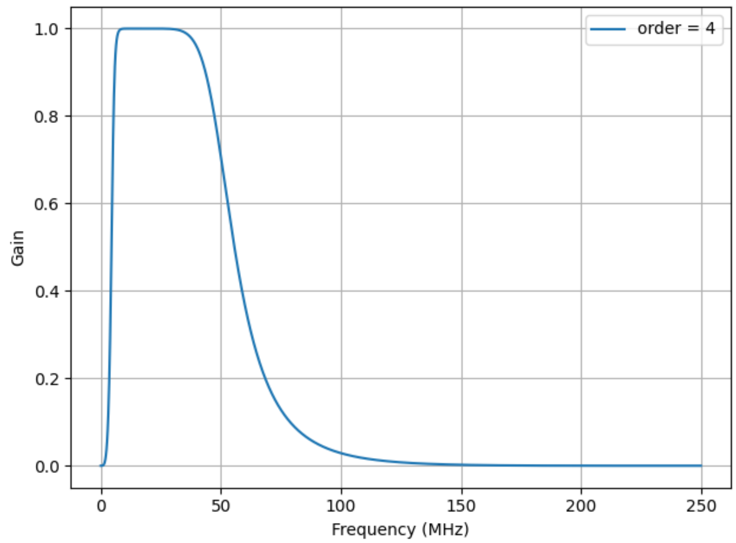
Depth	Center Frequency	Average Power Spectrum
3 mm to 4 mm	23.148 MHz	
6.5 mm to 8 mm	23.614 MHz	
8 mm to 15 mm	0.000 MHz	

Question c

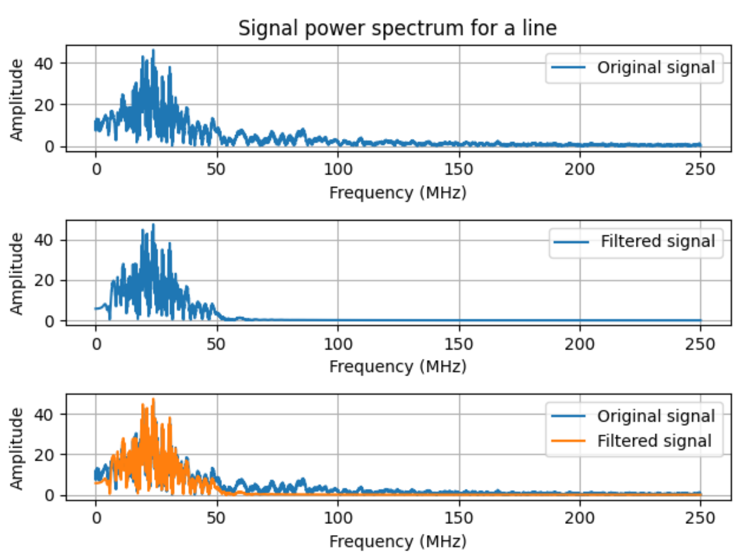
If you look at the previous diagrams, all of them have a rise in (0, 0). And going deeper, the point goes higher. Eventually, in the third diagram, it is higher than the peak around 23 MHz! It is a pretty interesting experiment since it proves the wave is losing energy and more specifically, **its details**. Traveling through tissue, sound waves encounter variations in the density, speed of sound, and attenuation of the tissue. As the sound waves penetrate deeper into the tissue, they encounter higher-density and higher-attenuation tissue that slows them and causes them to lose their energy.

Question d

i)
Order: 4
Lower cut: 5 MHz
Higher cut: 50 MHz



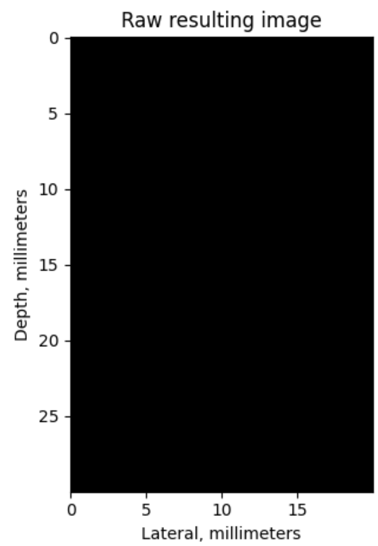
ii)
The first one is the original signal, and the second diagram is the filtered signal. The third diagram is nothing new but an all-in-one diagram! Obviously, high and low-frequency noises have been smoothed out.



iii)
I visualized it, but the resulting image had these properties:

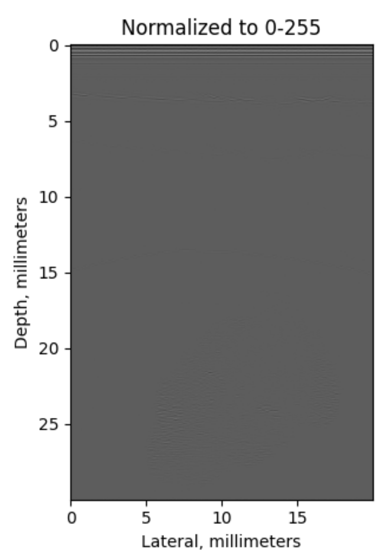
Min value: -0.24909939
Max value: 0.42828718
Mean: 0.00056988286

And it is not going to be anything suitable to visualize as you can see below:



So, I thought it might be better to normalize the matrix to get better in visualization. This time matrix was like (but the result is not much better):

Min value: 0
Max value: 255
Mean: 93.480325



Question e

i)

As professor Sadeghi concluded in slide 41-44 of Lecture 4-5_Ultrasound Imaging, to achieve $A_{RX}(t)$, we have to create the analytical signal using the Hilbert transform. Package ***scipy.signal*** has the Hilbert transform in it. So, using it we create:

$$Analytical(t) = y(t) + j \cdot z(t)$$

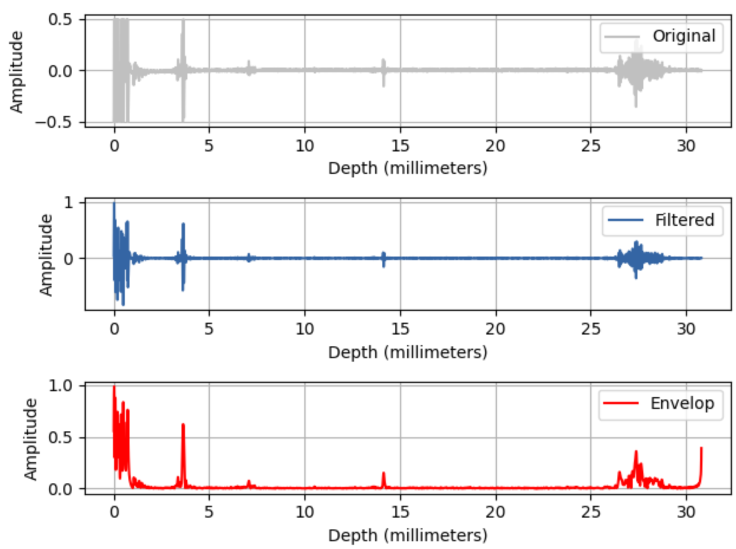
Where $z(t) = Hilbert(y(t))$ and $y(t)$ is the filtered signal.

Knowing that:

$$A_{RX}(t) = \sqrt{y_{RX}(t)^2 + z_{RX}(t)^2}$$

And this is easily achievable using Python built-in abs() function call when its input is the analytical signal!

ii)

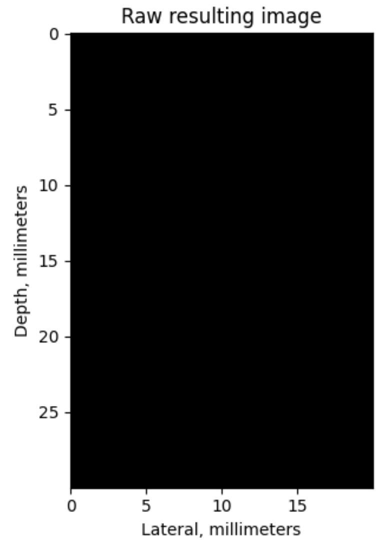


iii)

I visualized it, but the resulting image had these properties:

Min value: -0.00082732504
Max value: 0.92912143
Mean: 0.036636807

And it is not going to be anything suitable to visualize as you can see below:

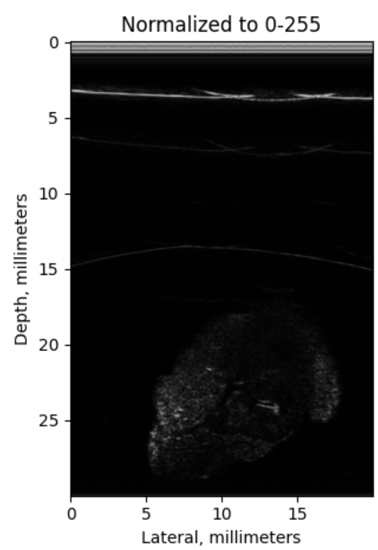


So, I normalized the values of the matrix using this transform:

$$img_{new} = \frac{(img - Min(img))}{Max(img) - Min(img)} \times 255$$

The new image has these properties:

Min value: 0
Max value: 255
Mean: 9.7760875



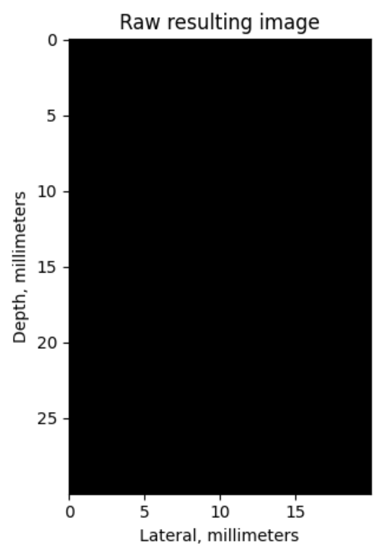
There are some visible stuff in the bottom of the image, but the contrast is too low to distinguish.

Question f

i)

Again, the properties of the resulting image matrix was not suitable to display with the best contrast possible.

Min value: -70.56744
Max value: -5.6456714
Mean: -43.993153

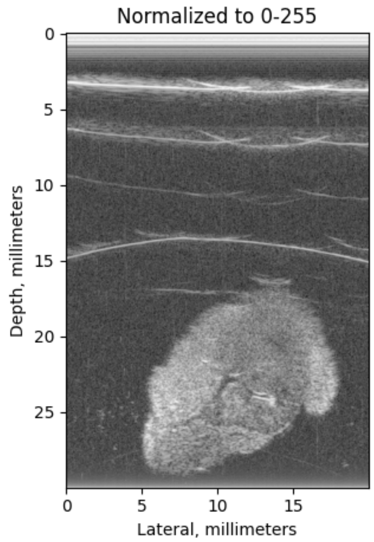


And again, to make the contrast better, I used this transform (we have learnt this transform in the previous experimental assignment):

$$img_{new} = \frac{(img - Min(img))}{Max(img) - Min(img)} \times 255$$

And the resulting image was like:

Min value: 0
Max value: 255
Mean: 103.87835416666667


Normalized to 0-255
Depth, millimeters
Lateral, millimeters
0
5
10
15
20
25
0
5
10
15