Digital Signal Processing
Home Assignment 1

Course Instructor: Prof. Dr. Nikolay Shilov
Teaching Assistant: Imre Delgado

Innopolis University
Spring 2022, Second Block

Groups:
B19-**DS**-01
B19-**AAI**-01
B19-**RO**-01

April 3, 2022

Instructions

- This Home Assignment consists in two tasks. They are all to be solved in `python` in a `jupyter` notebook (no `.py` files). The only libraries here to be used are `matplotlib`, `numpy` and `math` **and just those**.
- After solving the tasks a report is to be elaborated where the student provides a summary of the solution process, explaining the procedure, the parameters chosen, the results and displaying the visualizations for every task.
- The student will upload to the moodle the `jupyter` notebook (no links to the drive, no `.py` files), the report and any other files the student may had used. (Put it altogether in a zip folder if needed).
- The **deadline** is set to be April 10^{th} (included), the live grading session will start on the 17^{th} of April.

Tasks

Task 1: Noise Reduction

1. Consider the signals

$$y(t) = \sin(2\pi 50t) + \sin(2\pi 120t) \quad \text{and} \quad y_\epsilon(t) = y(t) + \epsilon$$

defined in the unit time domain $T = [0, 1]$. The ϵ represents some random noise added to the original signal. As it is shown in Figure 1, discretize the domain T in n evenly spaced subintervals and visualize both functions.

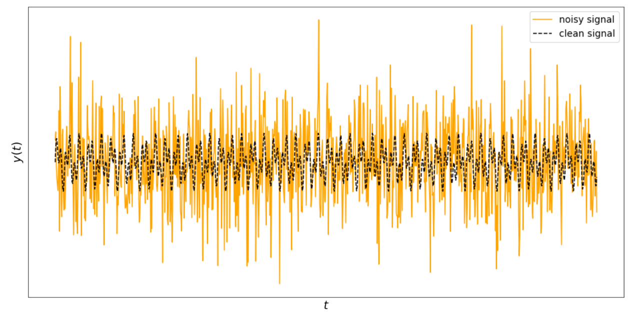

Figure 1: Noisy and Clean Signals

2. Compute the Fast Fourier Transform of the discretized function $y_\epsilon(t)$,

$$y_\epsilon(t) \xrightarrow{\mathcal{F}(\cdot)} \mathcal{F}\{y(t)_\epsilon\}(\omega) := z$$

3. Compute the power density spectrum (PSD) of z by multiplying it element-wise by its conjugate, and dividing it by the number of points n then visualize the PSD, as in Figure 2.

$$z \xrightarrow{\text{PSD}} \frac{z\bar{z}}{n}$$

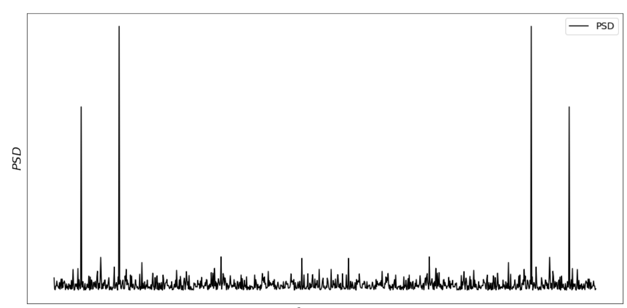

Figure 2: Power Density Spectrum

4. Based on your observations set a PDS-threshold τ to keep just the most representative frequencies, i.e. just keep the elements in the PDS that are less than threshold τ

$$\text{indices} = \{i \mid z[i] > \text{PSD-threshold}\}$$

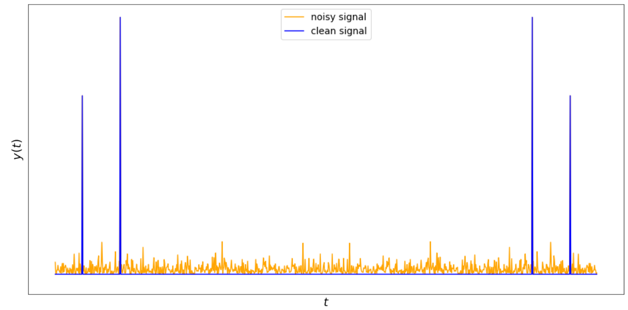

Figure 3: Signal Filtering

5. Return to the discretized function $\hat{y}_\epsilon(\omega)$, and using the indices as a filter set to zero the elements whose index is not in your set of indices, and compute the Inverse Fast Fourier Transform, which is your reconstructed function.

$$z[\text{indices}] \xrightarrow{\mathcal{F}^{-1}\{\cdot\}} \mathcal{F}^{-1}\{z[\text{indices}]\}$$

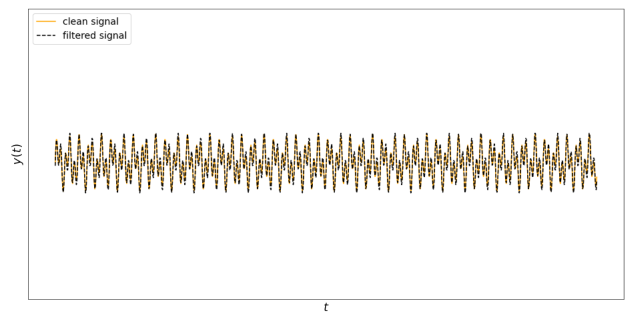

Figure 4: Noise Reduction

If done correct, then your signal with no noise should accurately resemble the original $y(t)$ function.

Task 2: Image Compression

Choose an image to play with. Figure 5 shows the picture for this toy example.

Figure 5: Lion Waiting In Namibia, taken from Wikipedia

1. Visualize your image, and convert it from RGB format to a grayscale format, i.e. the dimensions of your original picture (thinking of it as a three dimensional array) will be reduced one dimension

$$A_{w\times h\times c} \to A_{w\times h}$$

   where w, h, and c stand for weight, height and channel, respectively.

Figure 6: The Lion in Grayscale Format

2. Compute the Fast Fourier Transform in two dimensions (since you are working now with a two dimensional array) to get the spectrum of your image, then shift the zero-frequency to the center of that spectrum

$$A_{w\times h} \xrightarrow{\mathcal{F}^2(\cdot)} \mathcal{F}^2(A)_{w\times h}$$

3. Compute the magnitude (a.k.a. module, absolute value, etc) of the shifted spectrum to work with an homogeneous spectrum

$$\mathcal{F}^2(A)_{w\times h} \xrightarrow{|\,\cdot\,|} \left|\mathcal{F}^2(A)\right|_{w\times h}$$

4. Compute the (natural) logarithm adding 1 to the argument of your spectrum (to avoid values close to 0 that could make the operation to blow up)

$$\left|\mathcal{F}^2(A)\right|_{w\times h} \xrightarrow{\log(\,\cdot\,)} \log(\left|\mathcal{F}^2(A)\right| + 1)_{w\times h}$$

visualize the spectrum.

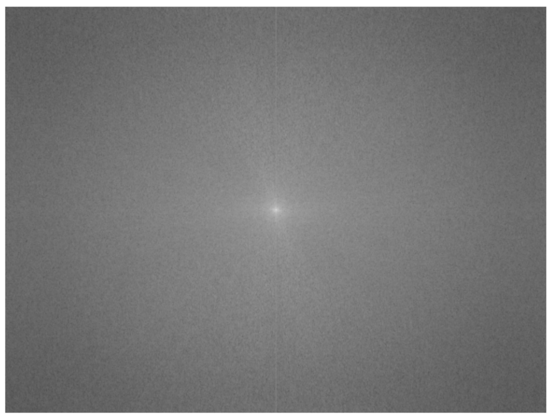

Figure 7: Image Spectrum

5. Reduce the dimension of your original shifted matrix in one array of one single dimension with the length being the product of the weight and the height of $\mathcal{F}^2(A)_{w\times h}$, i.e. reshape it

$$\mathcal{F}^2(A)_{w\times h} \xrightarrow{\text{reshape}} \mathcal{F}^2(A)_{wh\times 1}$$

6. Compute the magnitude of your reshaped array

$$\mathcal{F}^2(A)_{wh\times 1} \xrightarrow{|\,\cdot\,|} \left|\mathcal{F}^2(A)\right|_{wh\times 1} := A_f$$

7. Sort your array

$$\left|\mathcal{F}^2(A)\right|_{wh\times 1} \xrightarrow{\text{sort}} \text{sort}(\left|\mathcal{F}^2(A)\right|)_{wh\times 1} := a \in \mathbb{R}^{wh}$$

8. Set a value for τ, it should be sufficiently small, i.e. $\tau \ll 1$, multiply it by the length of your array to keep the values that are useful for the reconstruction, then floor the computation (i.e. approximate it to the smaller integer)

$$(1-\tau)wh \xrightarrow{\lfloor\,\cdot\,\rfloor} \lfloor(1-\tau)wh\rfloor := b \in \mathbb{N}$$

9. Define your threshold as that number whose index is given by b in your sorted array a,

$$\text{threshold} = a[b] := c \in \mathbb{R}$$

10. Zero out all small coefficients by keeping just the indices of the elements of $\left|\mathcal{F}^2(A)\right|_{wh\times 1}$ that are above the threshold

$$\text{indices} = \{(i,j) \mid A_f(i,j) > \text{threshold}\}$$

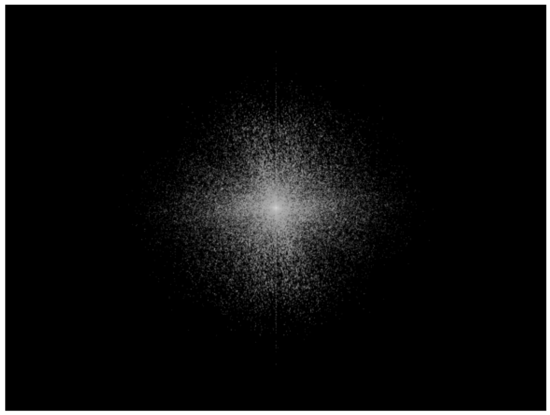

Figure 8: Filtered Spectrum

11. Finally return to your initial matrix $\mathcal{F}^2(A)_{w\times h}$ and select those indices and apply the shifting, absolute value and the logarithm as before and display it

$$\log(\left|\mathcal{F}^2(A)\right| + 1)_{w\times h}[\text{indices}]$$

if everything was done correctly, and the selected threshold was appropriate, then you should see your reconstructed image.

Figure 9: Reconstruction