Difital Signal Processing
Homework 2

Anton Buguev BS19-RO-01
a.buguev@innopolis.university

8 May 2022

Task 1: Moving Average Filter

In this task we will work with **Moving Average Filter** and we will consider the case of $M = 3$. Therefore we have

$$y[n] = \frac{1}{3}\sum_{k=0}^{2} x[n-k] = \frac{1}{3}x[n] + \frac{1}{3}x[n-1] + \frac{1}{3}x[n-2]$$

1. Compute system function H, plot its poles and zeros in the complex plane. First, let us apply **Z-transform** knowing that $Z\{x[n-k]\} = Z\{x[n]\}z^{-k} = \mathcal{X}(z)z^{-k}$ and we will get:

$$\mathcal{Y}(z) = \frac{1}{3}\mathcal{X}(z) + \frac{1}{3}\mathcal{X}(z)z^{-1} + \frac{1}{3}\mathcal{X}(z)z^{-2}$$

$$\mathcal{Y}(z) = \mathcal{X}(z)(\frac{1}{3} + \frac{1}{3}z^{-1} + \frac{1}{3}z^{-2})$$

And from here we can derive

$$H(z) = \frac{\mathcal{Y}(z)}{\mathcal{X}(z)} = \frac{1}{3} + \frac{1}{3z} + \frac{1}{3z^2} = \frac{z^2 + z + 1}{3z^2} - \text{system function}$$

To find zeros and poles we need to find when numerator and denominator become 0 respectively. So we have to solve system of equations:

$$\begin{cases} z^2 + z + 1 = 0, -\text{to find zeros} \\ 3z^2 = 0. - \text{to find poles} \end{cases} \tag{1}$$

From 1st equation we get that zeros of system function are $\frac{-1+j\sqrt{3}}{2}$ and $\frac{-1-j\sqrt{3}}{2}$. And from 2nd we find that poles of system function is 0.

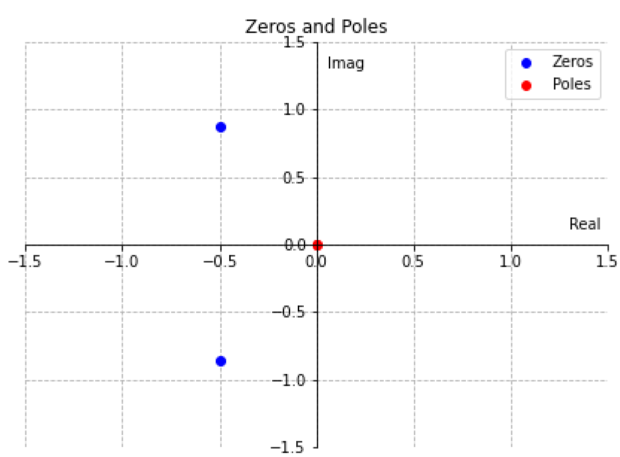

Figure 1: Poles and zeros of system function.

2. Compute the magnitude $|H(z)|$, the argument $arg(H(z))$.

   In order to find magnitude of system function let us first rewrite it in the general way:

$$H(z) = \sum_{n=0}^{N-1} z^{-n}$$

$$H(z) = \sum_{n=0}^{N-1} \frac{1}{z^n}$$

$$H(z) = \frac{1}{N} \cdot \frac{z - z^{1-N}}{z-1}$$

$$H(z) = \frac{1}{N} \cdot \frac{z \cdot (1 - z^{-N})}{z \cdot (1 - z^{-1})}$$

$$H(z) = \frac{1}{N} \cdot \frac{1 - z^{-N}}{1 - z^{-1}}$$

   Now let us substitute $z = e^{j\omega}$ in order to plot the magnitude and argument in frequency domain and we get:

$$H(\omega) = \frac{1}{N} \cdot \frac{1 - e^{-j\omega N}}{1 - e^{-j\omega}}$$

$$H(\omega) = \frac{1}{N} \cdot \frac{e^{-j\omega N/2} \cdot (e^{-j\omega N/2} - e^{j\omega N/2})}{e^{-j\omega/2} \cdot (e^{-j\omega/2} - e^{j\omega/2})}$$

$$H(\omega) = \frac{e^{-j\omega(N-1)/2}}{N} \cdot \frac{\cos\frac{\omega N}{2} - j\sin\frac{\omega N}{2} - \cos\frac{\omega N}{2} - j\sin\frac{\omega N}{2}}{\cos\frac{\omega}{2} - j\sin\frac{\omega}{2} - \cos\frac{\omega}{2} - j\sin\frac{\omega}{2}}$$

$$H(\omega) = \frac{e^{-j\omega(N-1)/2}}{N} \cdot \frac{(-2j\sin\frac{\omega N}{2})}{(-2j\sin\frac{\omega}{2})}$$

$$H(\omega) = \frac{e^{-j\omega(N-1)/2}}{N} \cdot \frac{\sin\frac{\omega N}{2}}{\sin\frac{\omega}{2}}$$

   Now let us substitute $N = 3$ to find $H(\omega)$:

$$H(\omega) = \frac{e^{-j\omega}}{3} \cdot \frac{\sin\frac{3\omega}{2}}{\sin\frac{\omega}{2}}$$

   Also knowing that $e^{-j\omega} = \cos\omega - j\sin\omega$ we get:

$$H(\omega) = \frac{\cos\omega - j\sin\omega}{3} \cdot \frac{\sin\frac{3\omega}{2}}{\sin\frac{\omega}{2}}$$

$$H(\omega) = \frac{\sin\frac{3\omega}{2}}{3\sin\frac{\omega}{2}}(\cos\omega - j\sin\omega)$$

   Therefore magnitude can be computed as:

$$|H(\omega)| = \sqrt{(Re(H(\omega))^2 + (Im(H(\omega))^2)}$$

   and argument is:

$$arg(H(\omega)) = \arctan\left(\frac{Im(H(\omega)}{Re(H(\omega)}\right)$$

After we calculated magnitude and argument of system function H using **Python**, we can plot them in frequency domain:

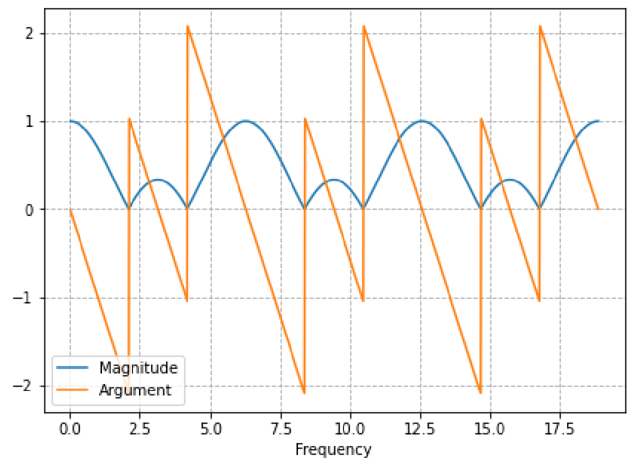

Figure 2: Magnitude and argument of system function.

3. Implement the moving average filter for any given M.

   In code we keep values of our function in list, which means in order to implement Moving Average Filter (MAF) for each element of input list (which consists of values of function) we can take piece (slice) of that list which will contain the current element and $(M-1)$ elements before (as a result slice will contain exactly M elements), sum up these elements and divide the result sum by M. This is the corresponding value of output list.

   However, first $(M-1)$ elements of input list do not contain enough elements before them, so let us start considering input list starting from M^{th} element and all inappropriate output elements set to 0.

```python
def MAF(x, M):
    y = np.zeros(x.shape) # output list, initially all elements zero
    for i in range(M - 1, len(x)): # start from M-th element
        y[i] = np.sum(x[i - (M - 1) : i + 1]) / M # take slice, find sum and didvide by M
    return y
```

Listing 1: Moving Average Filter

   Let us consider the function

$$x(t) = \cos\frac{\pi t}{2}$$

   For convenience let us divide the range $[0, 2\pi]$ into 30 equal segments. So the plot of the function will look the following way:

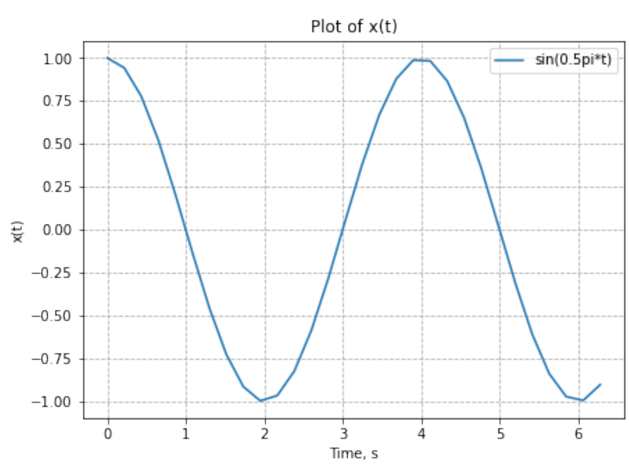

Figure 3: Plot of $x(t)$

After running MAF function for $M = 3$, $M = 6$ and $M = 9$ we get the following plot:

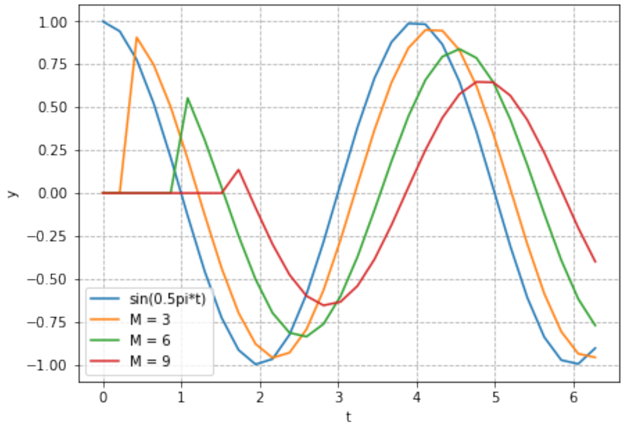

Figure 4: Plot of $x(t)$ and filtered values

As we can see, the higher value of M, the closer function to 0.

Task 2: The Window Method

- The Bartlett window. This window can be implemented using formula

$$\omega(n) = \frac{2}{M-1}\left(\frac{M-1}{2} - \left|n - \frac{M-1}{2}\right|\right), \ n \in [0; M-1]$$

Let us use $M = 51$ in this and all following examples. After computing window values we need to apply Fast Fourier Transform (FFT) to visualize frequency response as well. Also for better visualization let us shift the spectrum of FFT to the center and transform result into dB values, i.e. apply log_{10} and multiply by 20.

In order to check correctness of our implemented window, let us run function from *scipy* library to compute Bartlett window. And we laso apply FFT, shift and $20log_{10}$ in the same manner.

After these operations we can plot and compare results:

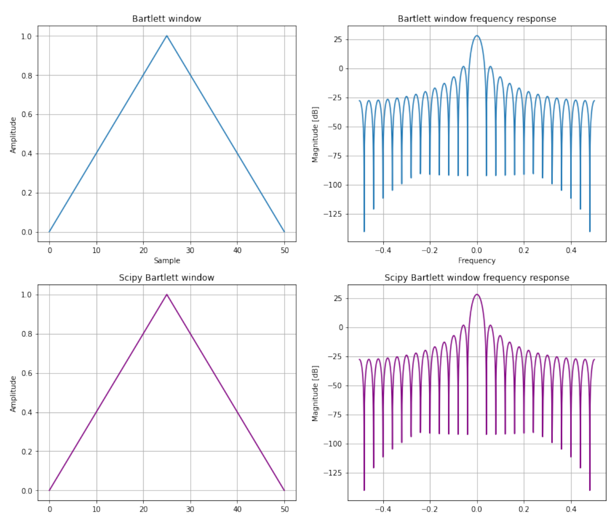

Figure 5: The Bartlett window

As we can see, our plots generated function and function from library look identical which means our implementation is correct.

- The Blackman Window.

  This is another window implementation which can be computed as

$$\omega(n) = 0.42 - 0.5\cos\frac{2\pi n}{M} + 0.08\cos\frac{4\pi n}{M},\ n \in [0; M-1]$$

  Let us apply the same procedure to compute window values and find frequency response of our function and function to compute the Blackman window from *scipy*. After that we can visualize and compare results:

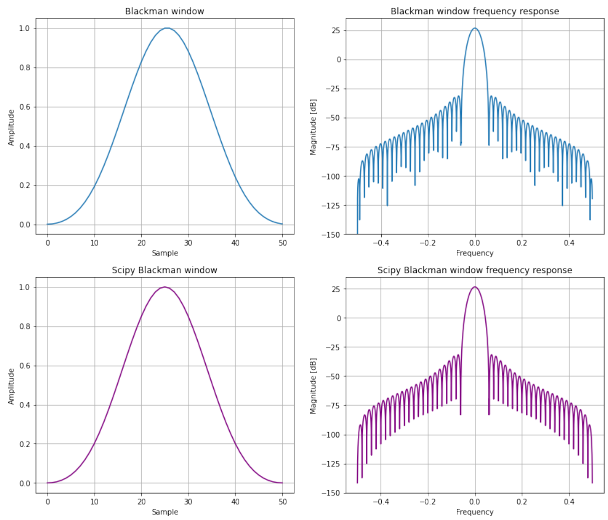

Figure 6: The Blackman window

  And again we see identical results which proves that our implementation is indeed correct.

- The Hamming window.

  Another window implementation using formula:

$$\omega(n) = 0.54 - 0.46\cos\frac{2\pi n}{M-1},\ n \in [0; M-1]$$

  Again we use the same steps for our and function for the Hamming window in *scipy* and compare results:

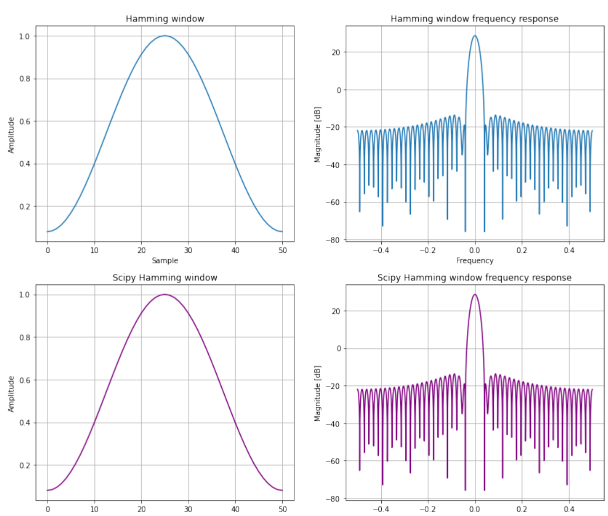

Figure 7: The Hamming window

  The result plots are identical so we can conclude that our function is correct.

- The Hann window.

  And finally the last window implementation can be computed using equation

$$\omega(n) = 0.5 - 0.5\cos\frac{2\pi n}{M-1},\ n \in [0; M-1]$$

  We use the same steps for our generated function and *scipy* function for the Hann window and results are:

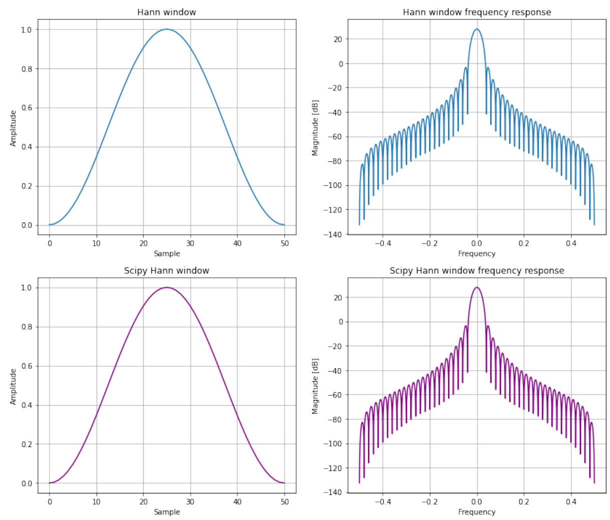

Figure 8: The Hann window

Results are identical as well, which means we successfully implemented these 4 window methods.