DSP Assignment 2

Parth Kalkar (p.kalkar@innopolis.university)

Task 1: Filter Implementation

Moving Average Filter

The general form of a Finite Impulse Response Filter (FIR) is described by,

$$y[n] = \sum_{k=0}^{M-1} h[k]\, x[n-k]$$

if the filter impulses (the h [k]'s) are set to constant coefficients divided by the tap number M, the equation becomes the well-known Moving Average Filter given by

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

for instance, we have
$M = 2 : y[n] = 1/2\, x[n] + 1/2\, x[n-1]$
$M = 3 : y[n] = 1/3\, x[n] + 1/3\, x[n-1] + 1/3\, x[n-2]$
$M = 4 : y[n] = 1/4\, x[n] + 1/4\, x[n-1] + 1/4\, x[n-2] + 1/4\, x[n-3]$ and so on.

In this task we will work with Moving Average Filter and we will consider the case of M = 3. Therefore we have $y[n] = 1/3\, x[n] + 1/3\, x[n-1] + 1/3\, x[n-2]$

1. Compute system function H, plot its poles and zeros in the complex plane. First, let us apply Z-transform knowing that $Z\{x[n-k]\} = Z\{x[n]\}z^{-k} = X(z)z^{-k}$ and we will get:
$Y(z) = 1/3\, X(z) + 1/3\, X(z)z^{-1} + 1/3\, X(z)^{-2}$
$Y(z) = X(z)(1/3 + 1/3\, z^{-2} + 1/3\, z^{-2})$

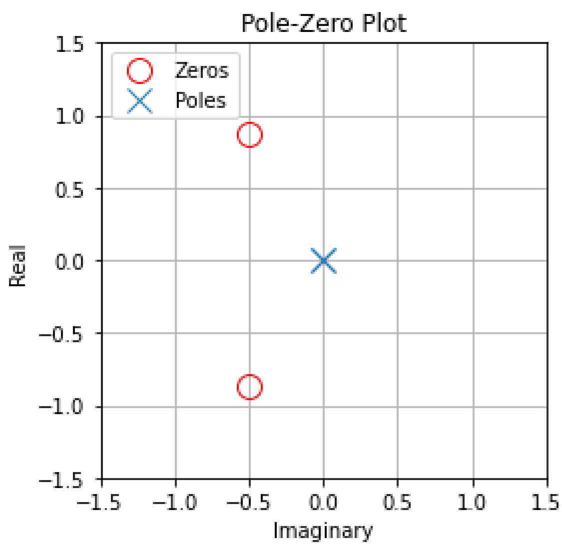
And from here we can derive system function as follows:
$H(z) = Y(z)/\, X(z) = 1/3 + 1/3z + 1/3z^{2} = (z^{2} + z + 1)/\, 3z^{2}$

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

To find zeros: $z^2 + z + 1 = 0$

To find poles: $3z^2 = 0$

We get that zeros of system function as $(-1 + j\sqrt{3})/2$ and $(-1 - j\sqrt{3})/2$ And from that poles of system function is 0 .



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:

$H(z) = 1/N * (z - z^{1-N})/(z - 1)$

$H(z) = 1/N * (1 - z^{-N})/(1 - z^{-1})$

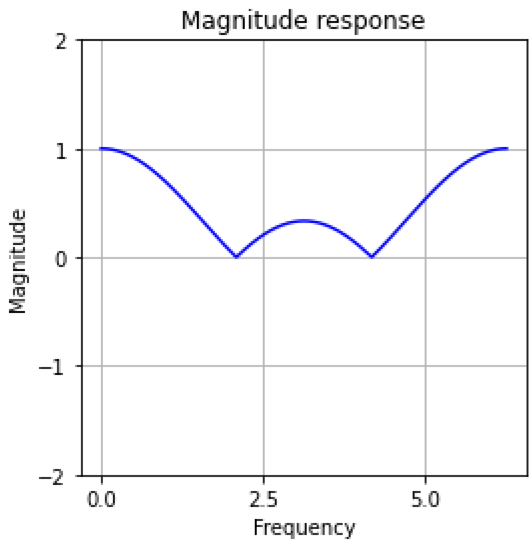
Now let us substitute $z = e^{j\omega} = \cos\omega + j\sin\omega$ in order to plot the magnitude and argument in frequency domain and we get:

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

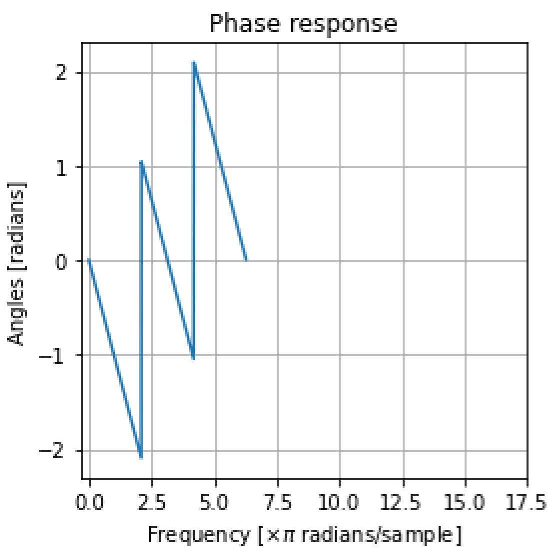
Simplifying, $H(\omega) = e^{-j\omega(N-1)/2}/N * (sin(\omega N/2)/sin(\omega/2))$

Now let us substitute N = 3 to find H(ω): $H(\omega) = e^{-j\omega}/3 * sin(3\omega/2)/sin(\omega/2)$

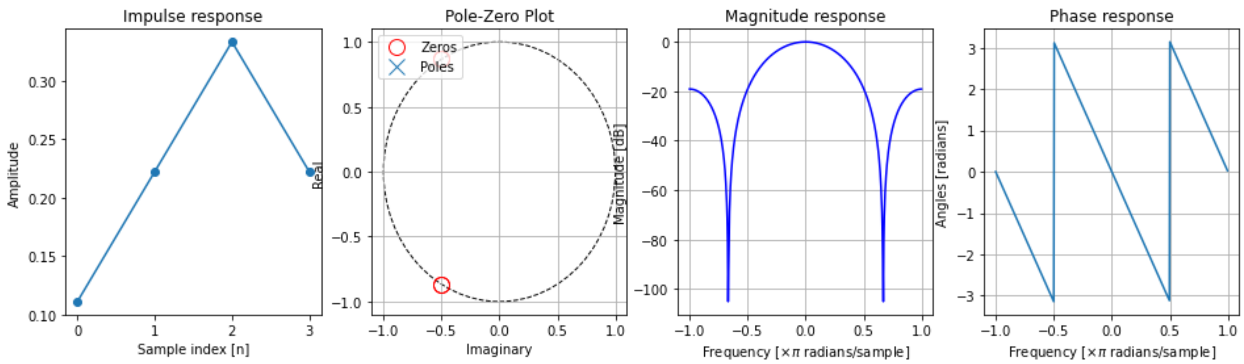
Therefore magnitude can be computed as: $|H(\omega)| = \sqrt{(Re(H(\omega))^2 + (Im(H(\omega))^2)}$



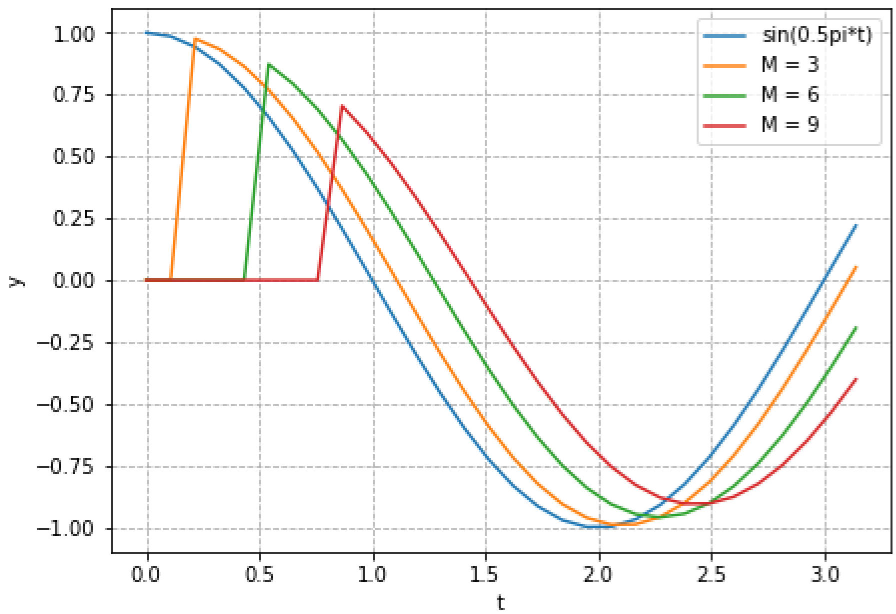
and argument is: $arg(H(\omega)) \;=\; arctan[Im(H(\omega)/Re(H(\omega)]$



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



3. Testing for M= 3, 6 and 9



The Window Method

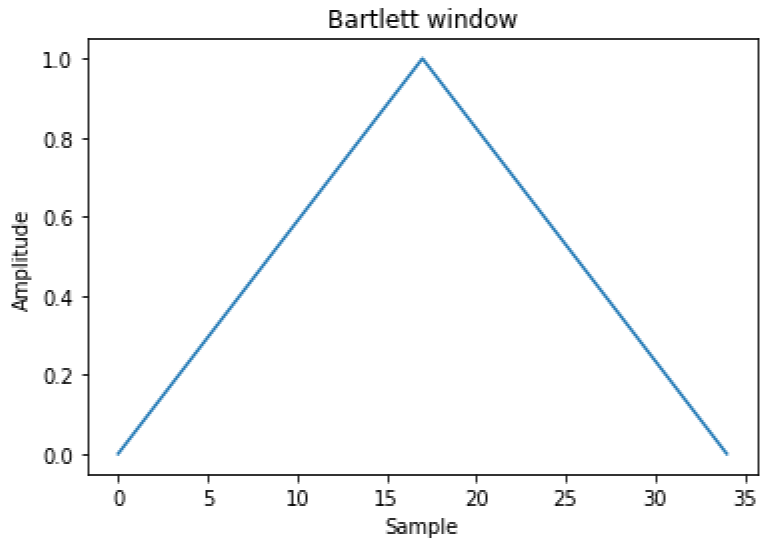
We recall that oscillations tend to appear when we abruptly truncate any signal. Windows help to considerably reduce these spurious oscillations providing a cleaner output after truncation. Implement the following window functions for any number of points M,

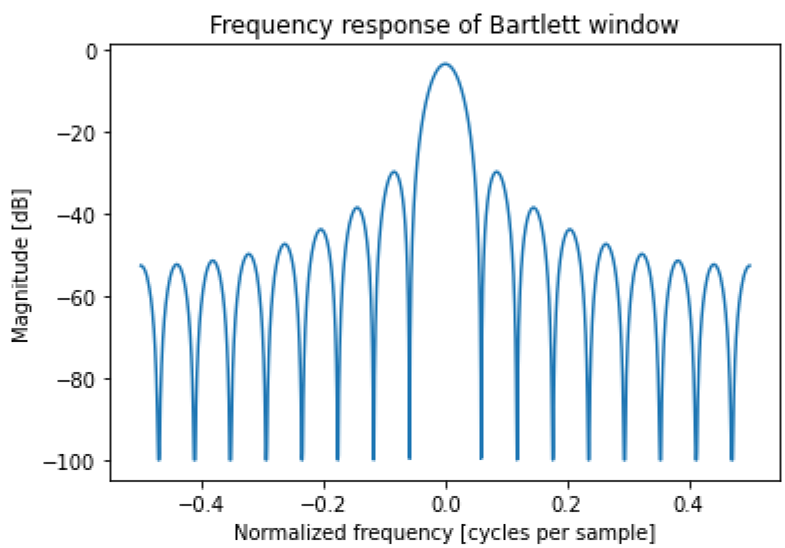
The Bartlett window

This window can be implemented using the formula:

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

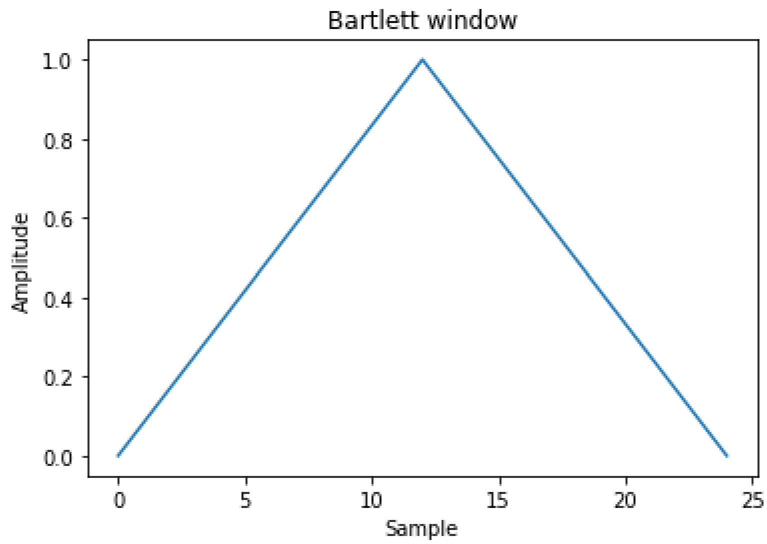
Let us use M = 35 in this and all the following examples. After computing window values we need to apply Fast Fourier Transform to visualize frequency response as well. Also for better visualization let us shift the spectrum of FFT to the center and transform the result into dB values.

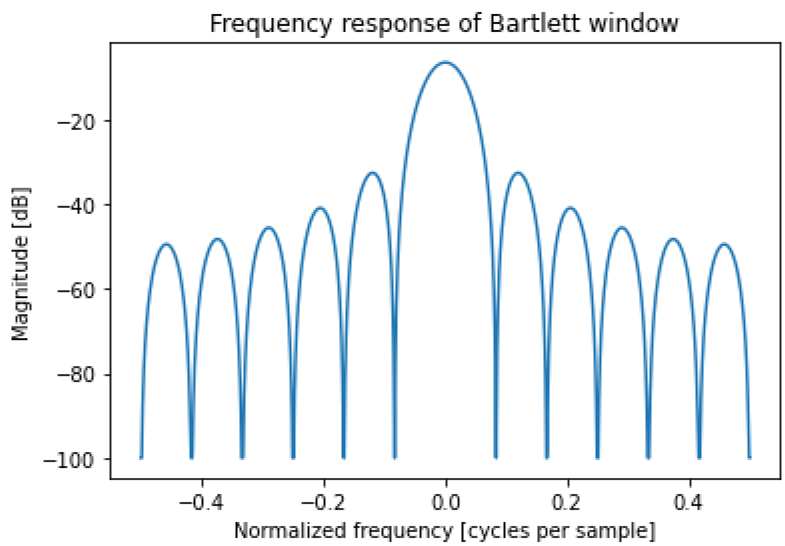




In order to check the correctness of our implemented window, let us run the function from scipy library to compute the Bartlett window. And we also apply FFT, shift, and in the same manner.

After these operations we can plot and compare the results:





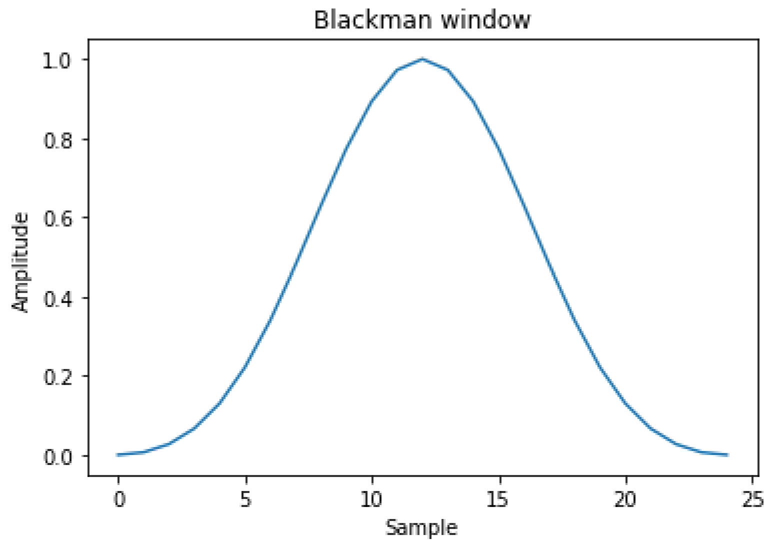
As we can see, our plots generated function and function from the library look identical which means our implementation is correct.

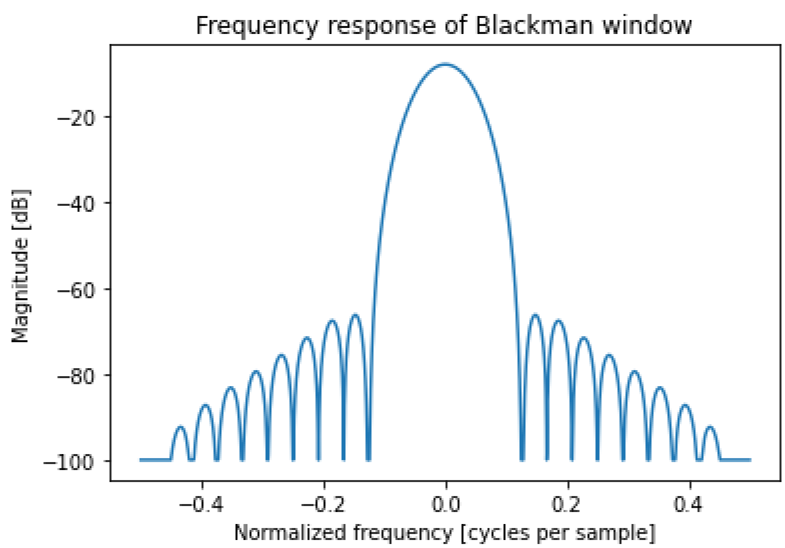
The Blackman window

This is another window implementation that can be computed as

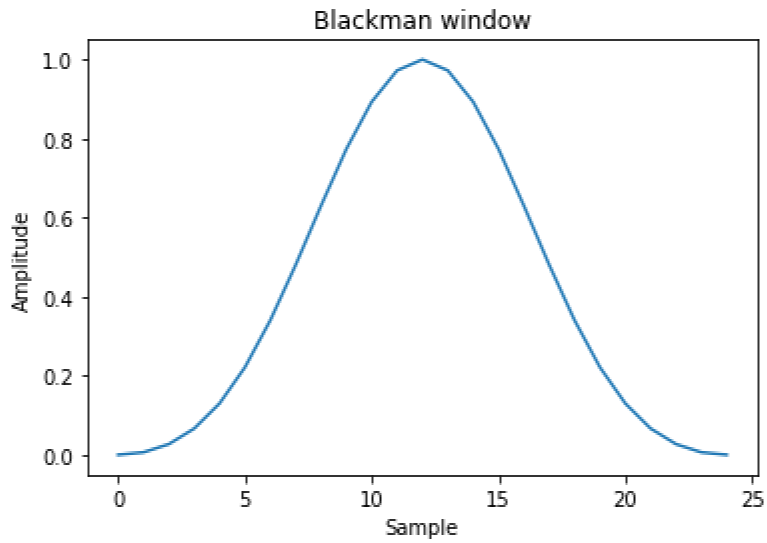
$$\omega(n) = 0.42 - 0.5\cos\frac{2\pi n}{M} + 0.08\cos\frac{4\pi n}{M},\ \ n \in [0; M-1]$$

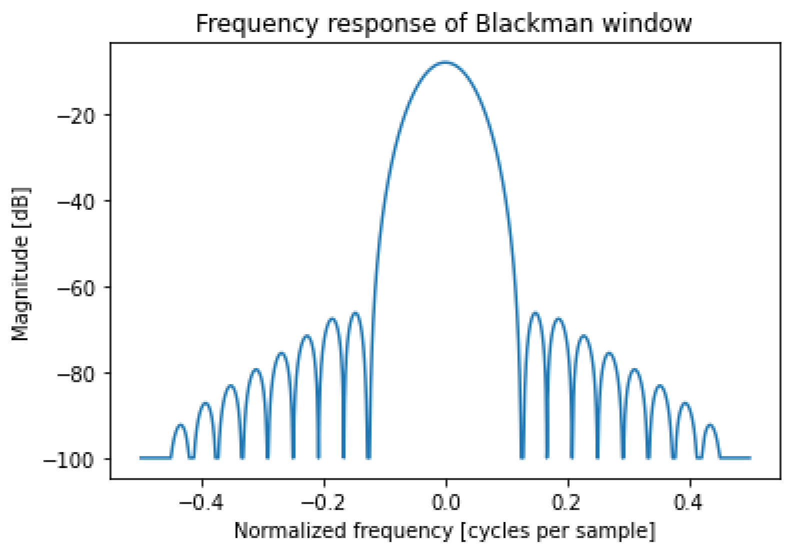
Let us apply the same procedure to compute window values and find the frequency response of our function and function to compute the Blackman window from scipy.





After that we can visualize and compare the results with scipy:





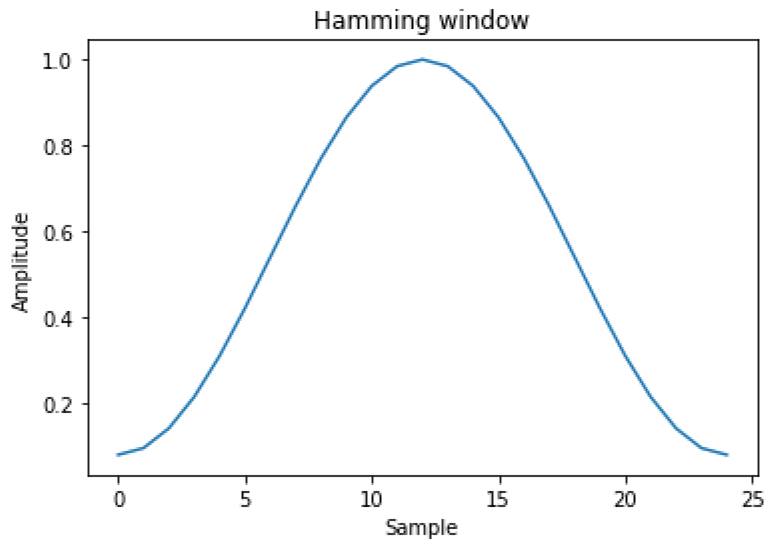
And again we see identical results which prove that our implementation is indeed correct.

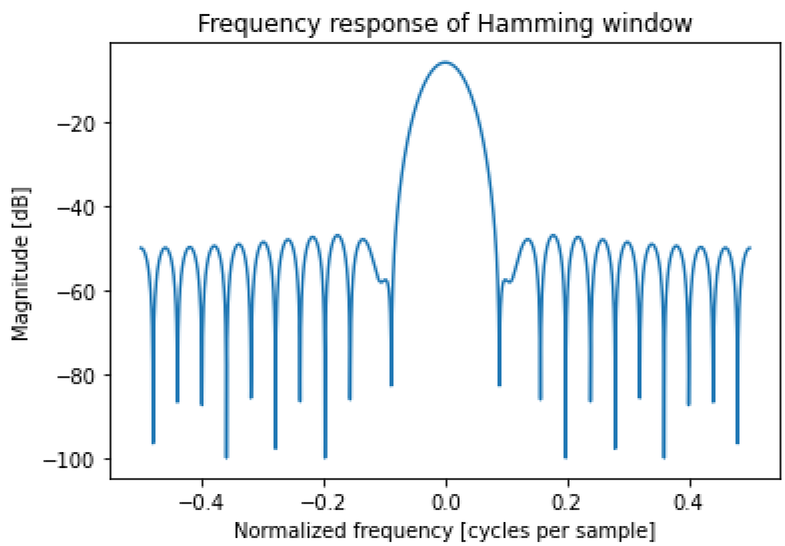
The Hamming window

Another window implementation uses the formula:

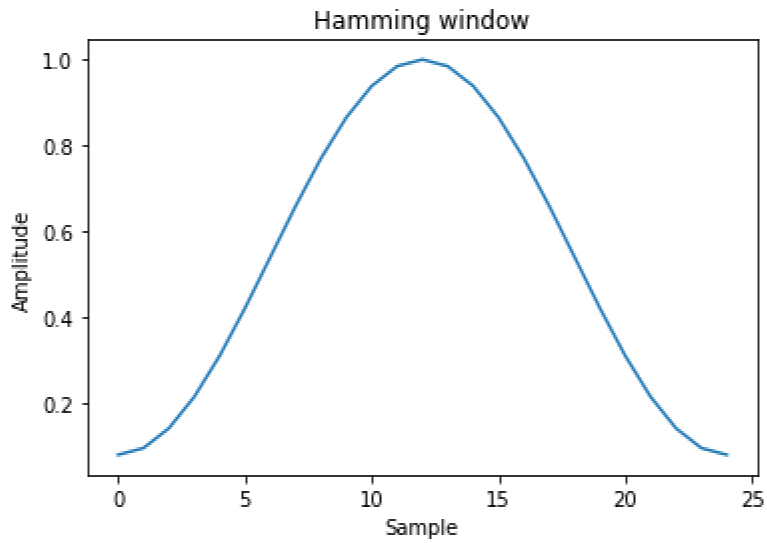
$$\omega(n) = 0.54 - 0.46\cos\frac{2\pi n}{M-1},\ \ n \in [0; M-1]$$

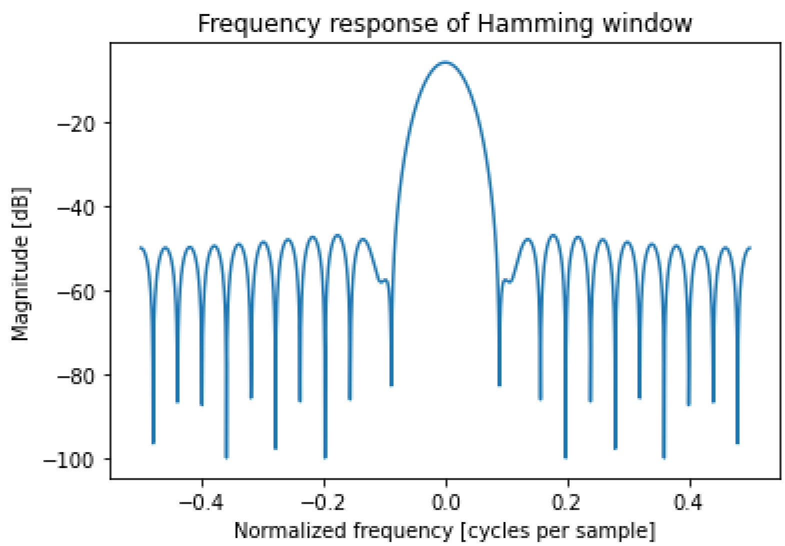
Let us apply the same procedure to compute window values and find the frequency response of our function and function to compute the Hamming window from scipy.





After that we can visualize and compare the results with scipy:





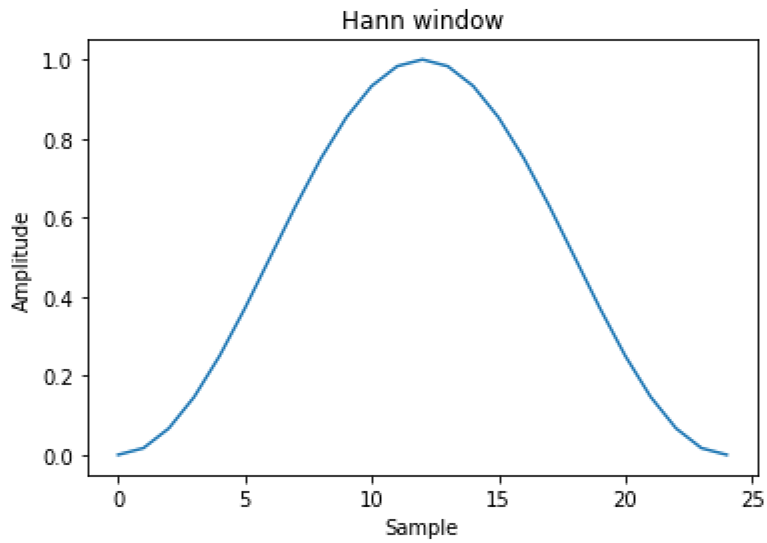
And again we see identical results which prove that our implementation is indeed correct.

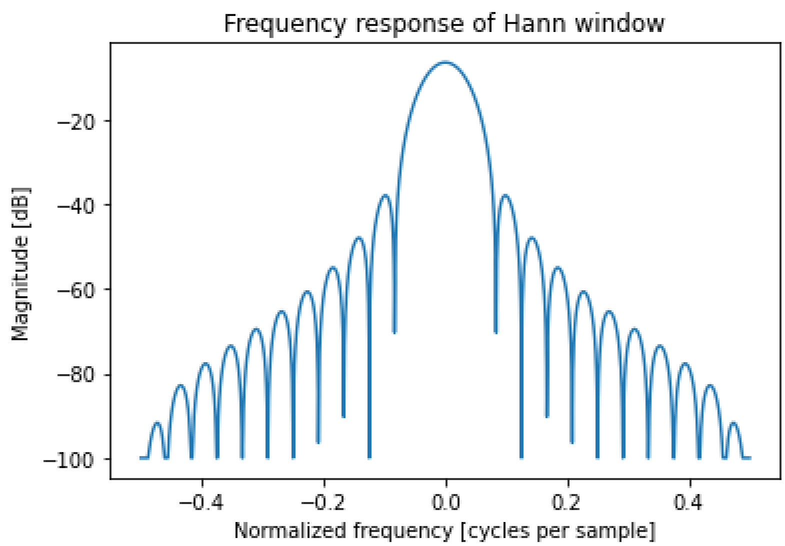
The Hann window

Finally, the last window implementation can be computed using the equation

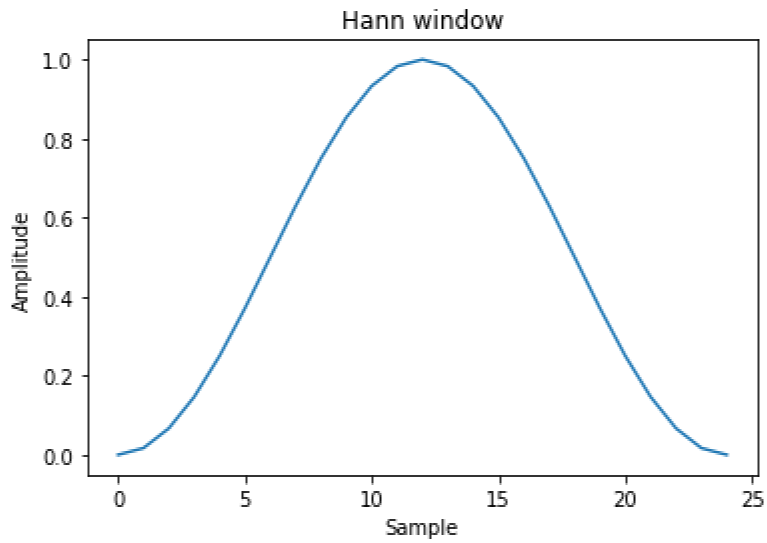
$$\omega(n) = 0.5 - 0.5\cos\frac{2\pi n}{M-1},\ \ n \in [0; M-1]$$

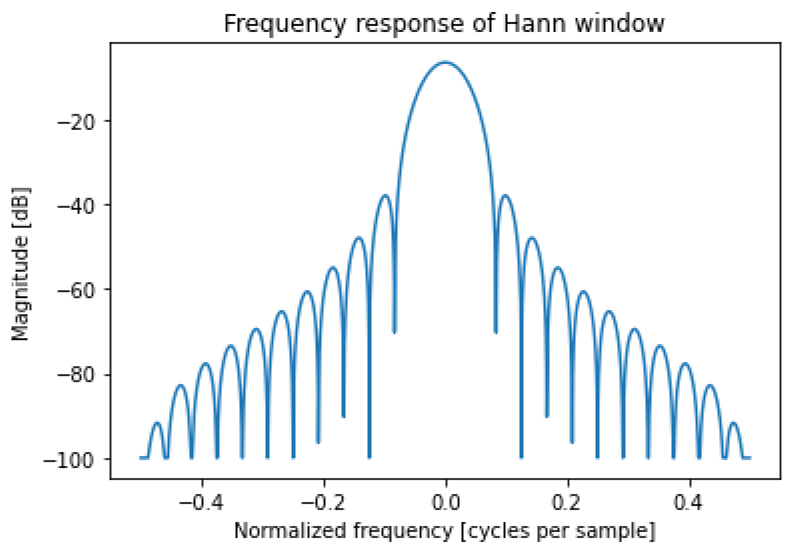
Let us apply the same procedure to compute window values and find the frequency response of our function and function to compute the Hann window from scipy.





After that we can visualize and compare the results with scipy:





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

Video Demonstration:

[Link]