GATE BM 23

EE23BTECH11018 - E.Mohana*

Question

A finite impulse response (FIR) filter has only two non-zero samples in its impulse response $h[n]$, namely $h[0] = h[1] = 1$. The Discrete Time Fourier Transform (DTFT) of $h[n]$ equals $H(e^{j\omega})$, as a function of the normalized angular frequency ω. For the range $|\omega| \leq \pi$, $\left|H(e^{j\omega})\right|$ is equal to

(A) $2\left|\cos(\omega)\right|$

(B) $2\left|\sin(\omega)\right|$

(C) $2\left|\cos(\frac{\omega}{2})\right|$

(D) $2\left|\sin(\frac{\omega}{2})\right|$

(GATE BM 2023)

Input Parameters Table

Parameter	Value	Description
$h[n]$	-	impulse response
$h[0]$	1	impulse response at $n = 0$
$h[1]$	1	impulse response at $n = 1$
ω	$-\pi \leq \omega \leq \pi$	normalized frequency
$H(e^{j\omega})$	$\sum_{n=0}^{M} h[n]e^{-jn\omega}$	frequency response

Table: Input Parameters Table

Solution

From Table 1,

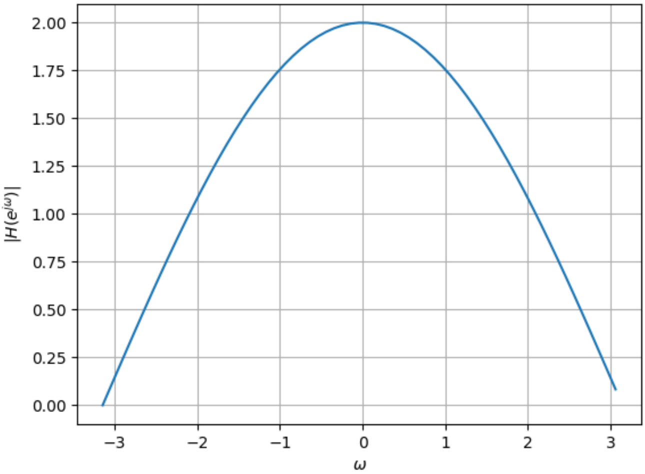
$$H(e^{j\omega}) = 1 + e^{-j\omega} \tag{1}$$

$$= e^{\frac{-j\omega}{2}}(e^{\frac{j\omega}{2}} + e^{\frac{-j\omega}{2}}) \tag{2}$$

$$= e^{\frac{-j\omega}{2}}(2cos\left(\frac{\omega}{2}\right)) \tag{3}$$

$$\left|H(e^{j\omega})\right| = 2\left|\cos\left(\frac{\omega}{2}\right)\right| \tag{4}$$

Plot



Python code

```python
import numpy as np
import matplotlib.pyplot as plt

# Load data from the "output.dat" file using numpy's loadtxt
data = np.loadtxt("output.dat")

# Extract n_values and y_values from the data
x_values = data[:, 0].astype(float)
y_values = data[:, 1].astype(float)

# Plot
plt.plot(x_values, y_values)
plt.xlabel(r'$\omega$')
plt.ylabel(r'$|H(e^{j\omega})|$')
plt.grid(True)
plt.savefig('../figs/fig1.png')
```

C code

```c
#include <stdio.h>
#include <math.h>
#include "mylib.h"
float f(float x){
return 2 * fabs(cos(x / 2));
}
int main() {
    // Define the range and step size
    float start = -M_PI;
    float stop = M_PI;
    float step = 0.1;

    // Calculate the number of values in the range
    int num_values = (stop - start) / step + 1;

    // Allocate arrays to store the generated values
    float x_values[num_values];
    float y_values[num_values];
        float(*func)(float);
        func = f;
    // Call the linspace function
    linspace(start, stop, step, x_values, y_values, num_values, func);
    //Save data to a file
     FILE* file = fopen("output.dat", "w");

    if (file != NULL) {
        for (int i = 0; i < num_values; ++i) {
            fprintf(file, "%f %f\n", x_values[i], y_values[i]);
        }

        fclose(file);
        printf("Data saved to 'output.dat'.\n");
    } else {
        printf("Error opening file for writing.\n");
    }

    return 0;
}
```

C-library code

```
#include "mylib.h"
#include <stdio.h>
#include <math.h>
void linspace(float start, float stop, float step, float* x_values, float* y_values, int num_values, float(*func)(
for(int i = 0; i<num_values; ++i){
 x_values[i] = start + i * step;
 y_values[i] = func(x_values[i]);
}}
```