



Digital Filtering

preparing for the exam - part 1, FIR filters

Antonin Novak

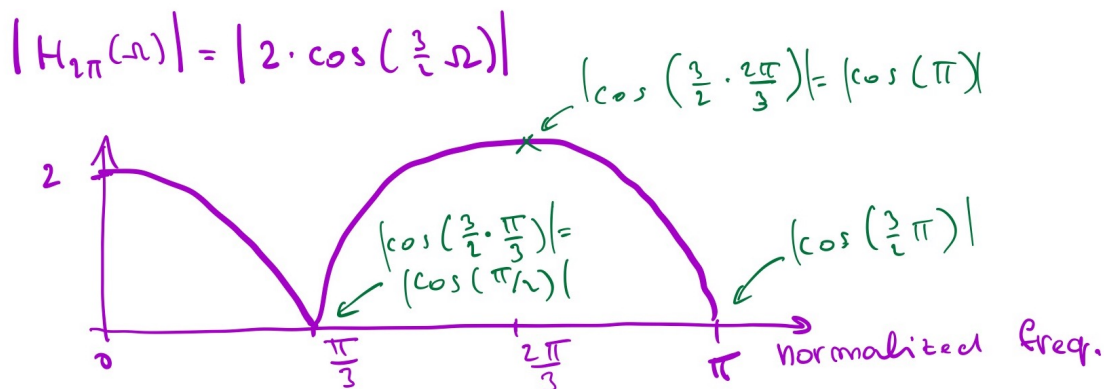


Theoretical Examples

1. Derive the FRF (Frequency Response Function) of a FIR (Finite Impulse Response) filter with filter coefficients $\mathbf{b} = [1, 0, 0, 1]$ and draw its absolute value as a function of normalized angular frequency between 0 and π .

.....

$$\begin{aligned}
 H_{2\pi}(\Omega) &= \sum_{n=0}^N b_n \cdot e^{-j\Omega n} = b_0 \cdot e^0 + b_1 \cdot e^{-j\Omega} + b_2 \cdot e^{-j2\Omega} + b_3 \cdot e^{-j3\Omega} = \\
 &= 1 \cdot e^0 + 1 \cdot e^{-j3\Omega} = 1 + e^{-j3\Omega} = \\
 &= (1 + e^{-j3\Omega}) \cdot \frac{e^{j\frac{3}{2}\Omega}}{e^{j\frac{3}{2}\Omega}} = \frac{e^{j\frac{3}{2}\Omega} + e^{-j\frac{3}{2}\Omega}}{e^{j\frac{3}{2}\Omega}} = \\
 &= 2 \cdot \frac{e^{j\frac{3}{2}\Omega} + e^{-j\frac{3}{2}\Omega}}{2} \cdot e^{-j\frac{3}{2}\Omega} = \\
 &= 2 \cdot \cos\left(\frac{3}{2} \cdot \Omega\right) \cdot e^{-j\frac{3}{2}\Omega}
 \end{aligned}$$



2. Find the amplitude of the output signal $y[n]$ of a FIR (Finite Impulse Response) filter with filter coefficients $b = [1, 0, 0, 0, 1]$ if the input signal is $x[n] = \cos(2\pi kn)$ for :

a) $k = 1/8$

b) $k = 1/4$.

hint: $\cos(a+b) = \cos(a)\cos(b) - \sin(a)\sin(b)$

.....

Solution: a) $k = \frac{1}{8}$
$$Y[n] = \sum_{m=0}^M b[m] \cdot x[n-m]$$

$$\begin{aligned} y[n] &= b_0 \cdot x[n] + b_1 \cdot x[n-1] + b_2 \cdot x[n-2] + b_3 \cdot x[n-3] + b_4 \cdot x[n-4] = \\ &= x[n] + x[n-4] = \cos(2\pi kn) + \cos(2\pi k(n-4)) = \\ &= \cos(2\pi kn) + \cos(2\pi kn - 8\pi k) = \\ &= \cos(2\pi kn) + \underbrace{\cos(2\pi kn)}_{\text{even function}} \cdot \underbrace{\cos(-8\pi k)}_{\text{even function}} - \underbrace{\sin(2\pi kn)}_{\text{odd function}} \cdot \underbrace{\sin(-8\pi k)}_{\text{odd function}} = \\ &= \cos(2\pi kn) + \cos(2\pi kn) \cdot \cos(8\pi k) + \sin(2\pi kn) \cdot \sin(8\pi k) = \\ &= \cos(2\pi kn) [1 + \cos(8\pi k)] + \sin(2\pi kn) \cdot \sin(8\pi k) \end{aligned}$$

a) $k = \frac{1}{8}$

$$Y[n] = \cos\left(\frac{\pi}{4}n\right) \underbrace{[1 + \cos(\pi)]}_0 + \sin\left(\frac{\pi}{4}n\right) \cdot \underbrace{\sin(\pi)}_0 = 0 \quad \text{AMPLITUDE} = 0$$

b) $k = \frac{1}{4}$

$$Y[n] = \cos\left(\frac{\pi}{2}n\right) \underbrace{[1 + \cos(2\pi)]}_2 + \sin\left(\frac{\pi}{2}n\right) \cdot \underbrace{\sin(2\pi)}_0 = 2 \cdot \cos\left(\frac{\pi}{2}n\right) \quad \text{AMPLITUDE} = 2$$

Practical Examples

3. Write a Python code that implements a FIR (Finite Impulse Response) filter. Use a circular buffer and sample-by-sample approach (no matrix or vector multiplication). Do not use the `lfilter` function. Continue the following code where `h` contains filter coefficients and `x` contains the input signal. Get the output signal `y`.

```
import numpy as np

h = ... # filter coefficients
x = ... # input signal

# prepare an empty array for the output signal
y = np.zeros_like(x)

# complete the code here to get the output signal y

.....

# prepare the buffer
circular_buffer = np.zeros(len(h))

# pointer to the circular buffer
pointer = 0

for n in range(len(x)):
    # write register
    circular_buffer[pointer] = x[n]

    # run the convolution equation
    temp = 0
    for k in range(len(h)):
        temp += h[k]*circular_buffer[(pointer-k) % len(h)]

    # write the output
    y[n] = temp

    # increment the pointer
    pointer = (pointer + 1) % len(h)
```

4. Implement a FIR (Finite Impulse Response) filter with filter coefficients $\mathbf{b} = [1, 0.3, 0.1]$ in Sigma Studio without using Filter block. You can use Multiply, Add, Gain, Delay, and T-connection.

.....

