Master 1: STLC

Practice #3: FIR filters design

1. Objectives:

- Design of Finite Impulse Response (FIR) filters using window and frequency sampling methods.
- Calculation of frequency responses of FIR filters.
- Analysis of simulation results using Matlab software.

2. Rappel théorique sur les filtres RIF

FIR filters have advantages with respect to IIR filters such as phase linearity and the resulting stability of the digital system. In general, FIR filters design is based on the window method and the frequency sampling method.

2. 1 Conception using a window method

A direct approach to obtaining a finite impulse response is to truncate an infinite impulse response. If we assume that $H_d\left(e^{j\omega}\right)$ is an ideal frequency response, so

$$H_d\left(e^{j\omega}\right) = \sum_{n=-\infty}^{+\infty} h_d(n) e^{-j\omega n} \tag{1}$$

where $h_d(n)$ is the corresponding impulse response, i.e.,

$$h_d(n) = \frac{1}{2\pi} \int_{-\omega_c}^{\omega_c} H_d\left(e^{j\omega}\right) e^{j\omega n} d\omega \tag{2}$$

Generally, $H_d\left(e^{j\omega}\right)$ is a constant response with discontinuities at the boundaries between bands. In this case, $h_d(n)$ is infinite. (2) is the FT of the periodic frequency response $H_d\left(e^{j\omega}\right)$ with $h_d(n)$ as the Fourier coefficients. If $h_d(n)$ is infinite, a way to get an impulse response $h(n)$ causal and finite is to truncate $h_d(n)$ as

$$h(n) = \begin{cases} h_d(n) & 0 \le n \le N-1 \\ 0 & otherwise \end{cases} \tag{3}$$

In general, one can represent $h(n)$ as the product of the desired impulse response $h_d(n)$ with a window $w(n)$ i.e.,

$$h(n) = h_d(n) w(n) \tag{4}$$

where

$$w(n) = \begin{cases} 1 & 0 \le n \le N-1 \\ 0 & otherwise \end{cases} \tag{5}$$

A causal linear phase low-pass filter is considered. The desired frequency response is given

$$\text{by } H_d\left(e^{j\omega}\right) = \begin{cases} e^{-j\omega\alpha} & |\omega| \le \omega_c \\ 0 & otherwxise \end{cases} \tag{6}$$

The corresponding impulse response is given by

$$h_d(n) = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} e^{j\omega(n-\alpha)}d\omega. = \frac{\sin(\omega_c(n-\alpha))}{\pi(n-\alpha)} \qquad n \ne \alpha \tag{7}$$

Clearly, $h_d(n)$ has an infinite duration. To create a linear-phase causal filter with finite N duration, we use (4) with $\alpha = \frac{N-1}{2}$ (linear phase condition). It is easy to verify that if $w(n)$ is symmetrical, the choice of α results in a sequence $h(n)$ that satisfies the equation $h(n) = h(N-1-n)$. Some examples of the most used windows for $0 \le n \le N-1$ (Fig. 1):

- Rectangulaire: $w(n) = 1$
- Bartlett: $w(n) = \begin{cases} \frac{2n}{N-1} \\ 2 - \frac{2n}{N-1} \end{cases}$
- Hanning: $w(n) = \frac{1}{2}\left[1 - \cos\left(\frac{2\pi n}{N-1}\right)\right]$
- Hamming: $w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right)$
- Blackman: $w(n) = 0.42 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right)$

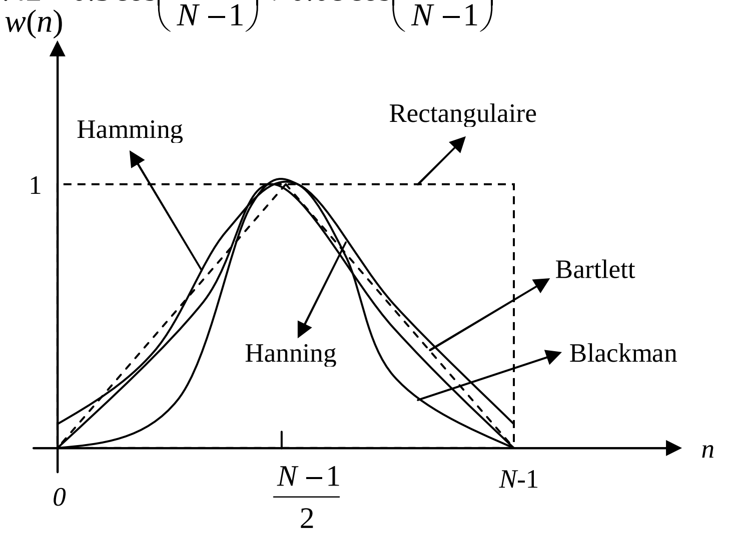


Fig. 1. *Windows usually used in FIR filters design.*

2. 2 FIR filter design by a frequency sampling method

We have already seen that a finite sequence can be represented by its discrete FT. FIR filters therefore have a representation in terms of frequency samples.

$$\tilde{H}(k) = H(z)\Big|_{z=e^{j\frac{2\pi}{N}k}} = \sum_{n=0}^{N-1} h(n) e^{-j\frac{2\pi}{N}kn}, \qquad k=0,\ 1,\ \ldots,\ N\text{-}1 \tag{8}$$

$H(z)$ may be represented by its samples according to the following expression

$$H(z) = \frac{1-z^{-N}}{N} \sum_{k=0}^{N-1} \frac{\tilde{H}(k)}{1-e^{j\frac{2\pi}{N}k} z^{-1}} \tag{9}$$

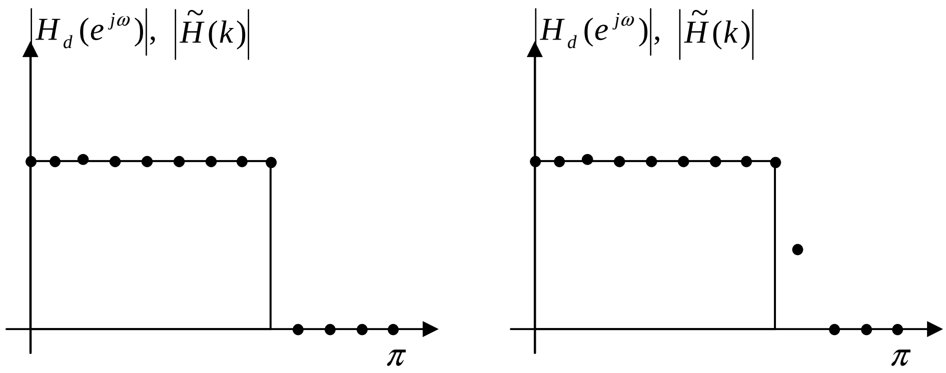
If, $z = e^{j\omega}$, the frequency response is obtained

$$\begin{aligned} H(e^{j\omega}) &= \frac{1-e^{-j\omega N}}{N} \sum_{k=0}^{N-1} \frac{\tilde{H}(k)}{1-e^{j\frac{2\pi}{N}k} e^{-j\omega}} \\ &= \frac{e^{-j\omega\frac{N-1}{2}}}{N} \sum_{k=0}^{N-1} \tilde{H}(k) e^{j\pi k\left(1-\frac{1}{N}\right)} \frac{\sin\left[\frac{N}{2}\left(\omega - \frac{2\pi}{N}k\right)\right]}{\sin\left[\frac{1}{2}\left(\omega - \frac{2\pi}{N}k\right)\right]} \end{aligned} \tag{10}$$

where

$$\tilde{H}(k) = H_d(e^{j\frac{2\pi}{N}k}),\ k=0,\ 1,\ \ldots,\ N\text{-}1 \tag{11}$$

The approach is therefore to specify the filter in terms of these samples from a single period of its desired response.



(a) **Fig. 2.** *Samples of the ideal low-pass filter frequency response*
(a) Sampling without transition
(b) Sampling with transition

The phase is assumed to be linear with duration of (N-1)/2 samples. The impulse response can be obtained using the inverse Fourier transform as

$$h(n) = \frac{1}{N}\sum_{k=0}^{N-1}\tilde{H}(k)e^{j\frac{2\pi}{N}kn} \quad , \quad n=0,\ 1,\ \ldots,\ N\text{-}1 \tag{12}$$

4. Manipulations

4. 1 Window method

In this experiment, we will use the Matlab «fir1» routine to perform the different types of FIR filters using the window approach.

Manip # 1 (low-pass FIR filter case):

We want to create a low-pass FIR filter with a cut-off frequency, $\omega_c = 0.25\pi$. For this purpose, if *N*=8 and *N*=48, calculate *h(n)* for *n*=0, …, *N*-1, $|H(e^{i\omega})|$ and $\arg(H(e^{j\omega}))$ using Hamming window (corresponding Matlab codes are given below) :

```
b = fir1(N-1,0.25);
freqz(b,1,512)
```

- Observe the two obtained responses and check the specifications in the pass-band.

Manip # 2 (high-pass FIR filter case) :

We want to create a high-pass FIR filter with a cut-off frequency, $\omega_c = 0.6\pi$. For this purpose, if *N*=8 and *N*=48, calculate *h(n)* for *n*=0, …, *N*-1, $|H(e^{i\omega})|$ and $\arg(H(e^{j\omega}))$ using the Hamming window (corresponding Matlab codes are given below) :

```
b = fir1(N-1,0.6,'high');
freqz(b,1,512)
```

- Observe the two obtained responses and check the specifications in the pass-band.

Manip # 3 (pass-band FIR filter case) :

We want to create a pass-band FIR filter with a cut-off frequencies, ω_1=0.2π and ω_2=0.6π. For this, if *N*=8 and *N*=48, calculate *h(n)* for *n*=0, …, *N*-1, $|H(e^{i\omega})|$ and $\arg(H(e^{j\omega}))$ using the Hamming window (corresponding Matlab codes are given below) :

```
b = fir1(N-1,[0.2 0.6]);
freqz(b,1,512)
```

- Observe the two obtained responses and check the specifications in the pass-band.

Manip # 4 (stop-band FIR filter case) :

We want to create a stop-band FIR filter with a cut-off frequencies, $\omega_1 = 0.3\pi$ and $\omega_2 = 0.6\pi$.

For *N*=8 and *N*=48, calculate *h(n)* for *n*=0, …, *N*-1, $\left|H(e^{i\omega})\right|$ and $\arg\left(H(e^{j\omega})\right)$ using the Hamming window (corresponding Matlab codes are given below) :

```
b = fir1(N-1,[0.2 0.7],'stop');
freqz(b,1,512)
```

- Observe the two obtained responses and check the specifications in the pass-band.

4. 2 Frequency sampling method

In this section, we will use the "fir2" Matlab routine to perform the different types of FIR filters using the frequency sampling approach.

Manip # 1 (low-pass FIR filter case):

Now, we want to create a low-pass cut-off frequency FIR filter, $\omega_c = 0.2\pi$ and *N*=30. Here, f represents the vector of the frequency points in the interval [0, 1] and m represents the vector containing the desired frequency response to the points specified in f. Execute the following Matlab codes.

```
f=[0 0.2 0.2 1]; m=[1 1 0 0];
b=fir2(30,f,m);
freqz(b,1,512)
```

- Observe the two obtained responses and check the specifications in the pass-band.

Manip # 2 (high-pass FIR filter case) :

We want to create a high-pass cut-off frequency FIR filter, $\omega_c = 0.6\pi$ and *N*=30. Execute the following Matlab codes.

```
f = [0 0.6 0.6 1]; m = [0 0 1 1];
b = fir2(30,f,m);
freqz(b,1,512)
```

- Observe the two obtained responses and check the specifications in the pass-band.

Manip #3 (pass-band FIR filter case):

- Perform a pass-band FIR filter for N=30, $\omega_1 = 0.3\pi$ and $\omega_2 = 0.6\pi$.

- Observe the two obtained responses and check the specifications in the pass-band.

Manip #4 (stop-band FIR filter case) :

- Perform a stop-band FIR filter for N=30, $\omega_1 = 0.2\pi$ and $\omega_2 = 0.7\pi$.

- Observe the two obtained responses and check the specifications in the pass-band.