FINITE IMPULSE RESPONSE FILTERS

A **finite impulse response (FIR)** filter is a type of a discrete-time filter. The impulse response, the filter's response to a Kronecker delta input, is *finite* because it settles to zero in a finite number of sample intervals. This is in contrast to infinite impulse response (IIR) filters, which have internal feedback and may continue to respond indefinitely. The impulse response of an Nth-order FIR filter lasts for N+1 samples, and then dies to zero. The FIR filter is a causal filter that requires a present input and a finite number of past input samples to compute the present output. The number of past samples required (N) is also known as the order of filter, and it signifies the amount of memory needed.

Thus, for FIR filter response we can write

$$h[n] = 0 \; n < 0 \; and \; n \geq N$$

The first condition is implied by the causality of the filter while the later condition indicates that the response of the filter needs only N past samples. An FIR filter of order N has (N + 1) terms in the RHS of its expression. The formula for FIR system using convolution can be written as:

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

where x(n) represents input to the system at time n, while y(n) is the output at the same time instant n.

The transfer function of the FIR filter is obtained by taking Z-transform of h[n].

An FIR filter has a number of useful properties which sometimes make it preferable to an infinite impulse response (IIR) filter. FIR filters:

- Are inherently stable. This is due to the fact that all the poles are located at the origin and thus are located within the unit circle.
- Require no feedback. This means that any rounding errors are not compounded by summed iterations. The same relative error occurs in each calculation. This also makes implementation simpler.
- They can easily be designed to be linear phase by making the coefficient sequence symmetric. Linear phase, or phase change proportional to frequency, corresponds to equal delay at all frequencies. This property is sometimes desired for phase-sensitive applications, for example crossover filters.

The main disadvantage of FIR filters is that considerably more computation power is required compared to an IIR filter with similar sharpness or selectivity, especially when low frequency (relative to the sample rate) cutoffs are needed.

The design of FIR filter is to determine the set of (N + 1) coefficients *h(n)*, n = 0 to N. Sometimes *h(0)* is set to unity and other coefficients are accordingly scaled, which reduces the the problem to the calculation of N values. Various techniques are used to design an FIR filter from the given specifications. The current experiment involves design of the FIR filter using window methods. The window techniques for FIR filter begin with specifications of the desired frequency response of the filter, usually denoted by $H_d(\omega)$. Using an inverse Fourier transform one can obtain a desired impulse response of the filter, $h_d(n)$ as:

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

However, usually an implementation of this response translates into an infinite impulse response (IIR) filter. In order to obtain an FIR filter with N coefficients, this response is truncated by multiplying $h_d(n)$ by a window function w(n) to obtain

$$h(n) = h_d(n)w(n)$$

where, *h(n)* is the impulse response of the desired FIR filter. Figure.1 illustrates the same. The choice of the window function greatly affects the performance of the filter.

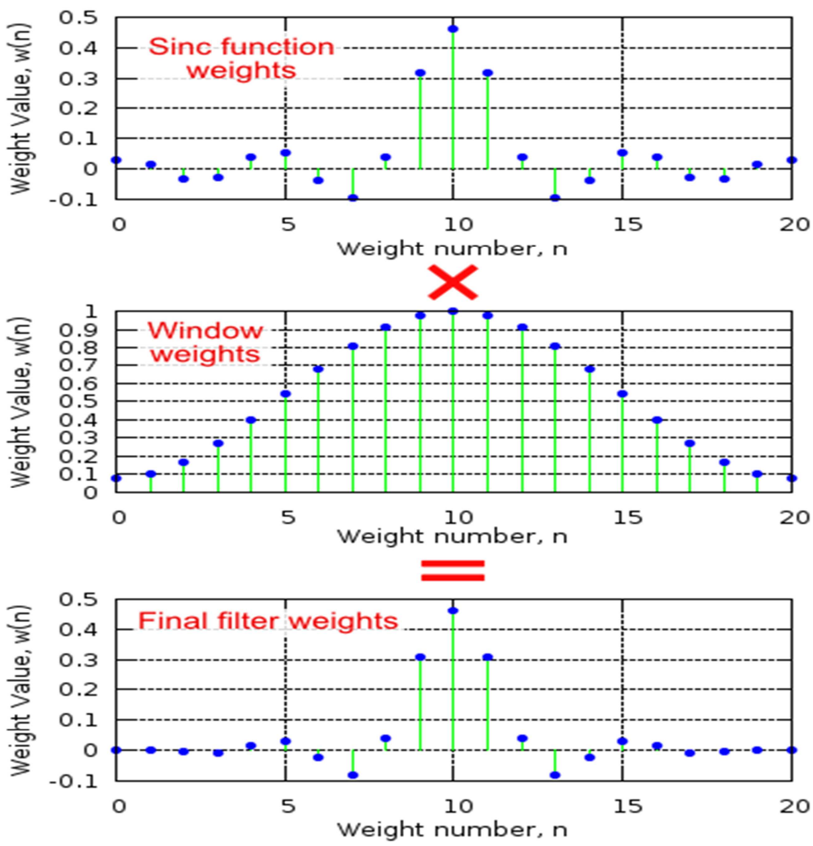

Figure 1. [4]

Determination of filter coefficients:

LPF:

The frequency response of LPF is

$$H_d\left(e^{j\omega}\right) = \begin{cases} 1, & |\omega| < 2\pi f_t \\ 0, & 2\pi f_t < |\omega| < \pi, \end{cases}$$

Taking inverse Fourier transform of this equation

$$h_d[n] = \frac{1}{2\pi}\int_{-2\pi f_t}^{2\pi f_t} e^{j\omega n}\, d\omega$$

$$= \frac{1}{2\pi jn}\left[e^{j\omega n}\right]_{-2\pi f_t}^{2\pi f_t}$$

$$= \frac{2f_t sin 2\pi f_t n}{2\pi f_t n} \quad , -\infty < n < \infty, n \neq 0$$

at $n = 0$ using L'Hospitals rule $= 2f_t$

Shifting by N/2 so for causality we get

LPF:

$$h_d(n) = \begin{cases} \dfrac{2f_t \sin\left(2\pi f_t\left(n - \frac{N}{2}\right)\right)}{2\pi f_t\left(n - \frac{N}{2}\right)} & , n \neq N/2 \\ 2f_t & , n = N/2 \end{cases}$$

Similarly all others come out to be,

HPF:

$$h_d(n) = \begin{cases} \dfrac{-2f_t \sin\left(2\pi f_t\left(n - \frac{N}{2}\right)\right)}{2\pi f_t\left(n - \frac{N}{2}\right)} & , n \neq N/2 \\ 1 - 2f_t & , n = N/2 \end{cases}$$

BPF:

$$h_d(n) = \begin{cases} \dfrac{2f_{t2}\sin\left(2\pi f_{t2}\left(n-\frac{N}{2}\right)\right) - 2f_{t1}\sin\left(2\pi f_{t1}\left(n-\frac{N}{2}\right)\right)}{\pi(f_{t1}+f_{t2})\left(n-\frac{N}{2}\right)} & ,n \neq N/2 \\ 2(f_{t2}-f_{t1}) & ,n = N/2 \end{cases}$$

BSF:

$$h_d(n) = \begin{cases} \dfrac{2f_{t1}\sin\left(2\pi f_{t2}\left(n-\frac{N}{2}\right)\right) - 2f_{t2}\sin\left(2\pi f_{t1}\left(n-\frac{N}{2}\right)\right)}{\pi(f_{t1}+f_{t2})\left(n-\frac{N}{2}\right)} & ,n \neq N/2 \\ -2(f_{t2}-f_{t1}) & ,n = N/2 \end{cases}$$

References:

- CC Studio examples.
- TMS320C6713 Datasheet, User Manual (and supporting documents).
- http://en.wikipedia.org/wiki/Finite_impulse_response.
- http://en.wikipedia.org/wiki/Kaiser_window.
- http://www.labbookpages.co.uk/audio/firWindowing.html.