Finite impulse response

Contents hide
(Top)
Definition
Properties
Frequency response
Filter design
Window design method
Least mean square error (MSE) method
Moving average example
See also
Notes
References

From Wikipedia, the free encyclopedia

In signal processing, a **finite impulse response** (**FIR**) **filter** is a filter whose impulse response (or response to any finite length input) is of finite duration, because it settles to zero in finite time. This is in contrast to infinite impulse response (IIR) filters, which may have internal feedback and may continue to respond indefinitely (usually decaying).[citation needed]

The impulse response (that is, the output in response to a Kronecker delta input) of an N^{th}-order discrete-time FIR filter lasts exactly $N+1$ samples (from first nonzero element through last nonzero element) before it then settles to zero.

FIR filters can be discrete-time or continuous-time, and digital or analog.

Definition [edit]

For a causal discrete-time FIR filter of order N, each value of the output sequence is a weighted sum of the most recent input values:

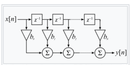
$$\begin{aligned} y[n] &= b_0 x[n] + b_1 x[n-1] + \cdots + b_N x[n-N] \\ &= \sum_{i=0}^{N} b_i \cdot x[n-i], \end{aligned}$$

where:

- $x[n]$ is the input signal,
- $y[n]$ is the output signal,
- N is the filter order; an N^{th}-order filter has $N+1$ terms on the right-hand side
- b_i is the value of the impulse response at the ith instant for $0 \le i \le N$ of an N^{th}-order FIR filter. If the filter is a direct form FIR filter then b_i is also a coefficient of the filter.

This computation is also known as discrete convolution.

The $x[n-i]$ in these terms are commonly referred to as *taps*, based on the structure of a tapped delay line that in many implementations or block diagrams provides the delayed inputs to the multiplication operations. One may speak of a *5th order/6-tap filter*, for instance.



A direct form discrete-time FIR filter of order 3. The top part is an *N*-stage delay line with *N* + 1 taps. Each unit delay is a z^{-1} operator in Z-transform notation.

A lattice-form discrete-time FIR filter of order 3. Each unit delay is a z^{-1} operator in Z-transform notation.

The impulse response of the filter as defined is nonzero over a finite duration. Including zeros, the impulse response is the infinite sequence:

$$h[n] = \sum_{i=0}^{N} b_i \cdot \delta[n-i] = \begin{cases} b_n & 0 \le n \le N \\ 0 & \text{otherwise.} \end{cases}$$

If an FIR filter is non-causal, the range of nonzero values in its impulse response can start before $n = 0$, with the defining formula appropriately generalized.

Properties [edit]

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

- 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.
- Are inherently stable, since the output is a sum of a finite number of finite multiples of the input values, so can be no greater than $\sum |b_i|$ times the largest value appearing in the input.
- Can easily be designed to be linear phase by making the coefficient sequence symmetric. This property is sometimes desired for phase-sensitive applications, for example data communications, seismology, crossover filters, and mastering.

The main disadvantage of FIR filters is that considerably more computation power in a general purpose processor is required compared to an IIR filter with similar sharpness or selectivity, especially when low frequency (relative to the sample rate) cutoffs are needed. However, many digital signal processors provide specialized hardware features to make FIR filters approximately as efficient as IIR for many applications.

Frequency response [edit]

The filter's effect on the sequence $x[n]$ is described in the frequency domain by the convolution theorem:

$$\underbrace{\mathcal{F}\{x * h\}}_{Y(\omega)} = \underbrace{\mathcal{F}\{x\}}_{X(\omega)} \cdot \underbrace{\mathcal{F}\{h\}}_{H(\omega)} \quad \text{and} \quad y[n] = x[n] * h[n] = \mathcal{F}^{-1}\{X(\omega) \cdot H(\omega)\},$$

where operators $\mathcal{F}$ and $\mathcal{F}^{-1}$ respectively denote the discrete-time Fourier transform (DTFT) and its inverse. Therefore, the complex-valued, multiplicative function $H(\omega)$ is the filter's frequency response. It is defined by a Fourier series:

$$H_{2\pi}(\omega) \triangleq \sum_{n=-\infty}^{\infty} h[n] \cdot \left(e^{i\omega}\right)^{-n} = \sum_{n=0}^{N} b_n \cdot \left(e^{i\omega}\right)^{-n},$$

where the added subscript denotes 2π-periodicity. Here ω represents frequency in normalized units (*radians/sample*). The substitution $\omega = 2\pi f_s$ favored by many filter design programs, changes the units of frequency (f) to *cycles/sample* and the periodicity to 1.[A] When the $x[n]$ sequence has a known sampling-rate, f_s *samples/second*, the substitution $\omega = 2\pi f / f_s$ changes the units of frequency (f) to *cycles/second* (hertz) and the periodicity to f_s. The value $\omega = \pi$ corresponds to a frequency of $f = \frac{f_s}{2}$ *Hz* = $\frac{1}{2}$ *cycles/sample*, which is the Nyquist frequency.

$H_{2\pi}(\omega)$ can also be expressed in terms of the Z-transform of the filter impulse response:

$$\begin{aligned} \widehat{H}(z) &\triangleq \sum_{n=-\infty}^{\infty} h[n] \cdot z^{-n}, \\ H_{2\pi}(\omega) &= \widehat{H}(z)\Big|_{z=e^{i\omega}} = \widehat{H}(e^{i\omega}). \end{aligned}$$

Filter design [edit]

FIR filters are designed by finding the coefficients and filter order that meet certain specifications, which can be in the time domain (e.g. a matched filter) or the frequency domain (most common). Matched filters perform a cross-correlation between the input signal and a known pulse shape. The FIR convolution is a cross-correlation between the input signal and a time-reversed copy of the impulse response. Therefore, the matched filter's impulse response is "designed" by sampling the known pulse-shape and using those samples in reverse order as the coefficients of the filter:[1]

When a particular frequency response is desired, several different design methods are common:

1. Window design method
2. Frequency sampling method
3. Least MSE (mean square error) method
4. Parks–McClellan method (also known as the equiripple, optimal, or minimax method). The Remez exchange algorithm is commonly used to find an optimal equiripple set of coefficients. Here the user specifies a desired frequency response, a weighting function for errors from this response, and a filter order N. The algorithm then finds the set of $N+1$ coefficients that minimize the maximum deviation from the ideal. Intuitively, this finds the filter that is as close as possible to the desired response given that only $N+1$ coefficients can be used. This method is particularly easy in practice since at least one text[2] includes a program that takes the desired filter and N, and returns the optimum coefficients.
5. Equiripple FIR filters can be designed using the DFT algorithms as well.[3] The algorithm is iterative in nature. The DFT of an initial filter design is computed using the FFT algorithm (if an initial estimate is not available, h[n]=delta[n] can be used). In the Fourier domain, or DFT domain, the frequency response is corrected according to the desired specs, and the inverse DFT is then computed. In the time-domain, only the first N coefficients are kept (the other coefficients are set to zero). The process is then repeated iteratively: the DFT is computed once again, correction applied in the frequency domain and so on.

Software packages such as MATLAB, GNU Octave, Scilab, and SciPy provide convenient ways to apply these different methods.

Window design method [edit]

In the window design method, one first designs an ideal IIR filter and then truncates the infinite impulse response by multiplying it with a finite length window function. The result is a finite impulse response filter whose frequency response is modified from that of the IIR filter. Multiplying the infinite impulse by the window function in the time domain results in the frequency response of the IIR being convolved with the Fourier transform (or DTFT) of the window function. If the window's main lobe is narrow, the composite frequency response remains close to that of the ideal IIR filter.

The ideal response is often rectangular, and the corresponding IIR is a sinc function. The result of the frequency domain convolution is that the edges of the rectangle are tapered, and ripples appear in the passband and stopband. Working backward, one can specify the slope (or width) of the tapered region (*transition band*) and the height of the ripples, and thereby derive the frequency-domain parameters of an appropriate window function. Continuing backward to an impulse response can be done by iterating a filter design program to find the minimum filter order. Another method is to restrict the solution set to the parametric family of Kaiser windows, which provides closed form relationships between the time-domain and frequency domain parameters. In general, that method will not achieve the minimum possible filter order, but it is particularly convenient for automated applications that require dynamic, on-the-fly, filter design.

The window design method is also advantageous for creating efficient half-band filters, because the corresponding sinc function is zero at every other sample point (except the center one). The product with the window function does not alter the zeros, so almost half of the coefficients of the final impulse response are zero. An appropriate implementation of the FIR calculations can exploit that property to double the filter's efficiency.

Least mean square error (MSE) method [edit]

Goal:

To design FIR filter in the MSE sense, we minimize the mean square error between the filter we obtained and the desired filter.

$\text{MSE} = f_s^{-1} \int_{-f_s/2}^{f_s/2} |H(f) - H_d(f)|^2 \, df$, where f_s is sampling frequency. $H(f)$ is the spectrum of the filter we obtained, and $H_d(f)$ is the spectrum of the desired filter.

Method:

Given an N-point FIR filter $h[n]$, and $r[n] = h[n+k], k = \frac{(N-1)}{2}$.

Step 1: Suppose $h[n]$even symmetric. Then, the discrete time Fourier transform of $\mathbf{r}[n]$ is defined as

$$R(F) = e^{2\pi i F k} H(F) = \sum_{n=0}^{k} s[n] \cos(2\pi n F)$$

Step 2: Calculate mean square error.

$$\text{MSE} = \int_{-1/2}^{1/2} |R(F) - H_d(F)|^2 \, dF$$

Therefore,

$$\text{MSE} = \int_{-1/2}^{1/2} \sum_{n=0}^{k} s[n] \cos(2\pi n F) \sum_{r=0}^{k} s[r] \cos(2\pi r F) \, dF - 2 \int_{-1/2}^{1/2} \sum_{n=0}^{k} s[n] \cos(2\pi n F) H_d \, dF + \int_{-1/2}^{1/2} H_d(F)^2 \, dF$$

Step 3: Minimize the mean square error by doing partial derivative of MSE with respect to $s[n]$

$$\frac{\partial \text{MSE}}{\partial s[n]} = 2 \sum_{r=0}^{k} s[r] \int_{-1/2}^{1/2} \cos(2\pi r F) \cos(2\pi n F) \, dF - 2 \int_{-1/2}^{1/2} H_d(F)^2 \cos(2\pi n F) \, dF = 0$$

After organization, we have

$$s[0] = \int_{-1/2}^{1/2} H_d(F) \, dF$$

$$s[n] = \int_{-1/2}^{1/2} \cos(2\pi n F) H_d(F) \, dF, \text{ for } n \neq 0$$

Step 4: Change $s[n]$ back to the presentation of $h[n]$

$h[k] = s[0], h[k+n] = s[n]/2, h[k-n] = s[n]/2,$ *for* $n = 1, 2, 3, \ldots, k,$ where $k = (N-1)/2$ and $h[n] = 0$ for $n < 0$ and $n \geq N$

In addition, we can treat the importance of passband and stopband differently according to our needs by adding a weighted function, $W(f)$ Then, the MSE error becomes

$$\text{MSE} = \int_{-1/2}^{1/2} W(F) |R(F) - H_d(F)|^2 \, dF$$

Moving average example [edit]

A moving average filter is a very simple FIR filter. It is sometimes called a boxcar filter, especially when followed by decimation, or a sinc-in-frequency. The filter coefficients, $b_0, \ldots, b_N$, are found via the following equation:

$$b_i = \frac{1}{N+1}$$

To provide a more specific example, we select the filter order:

$$N = 2$$

The impulse response of the resulting filter is:

$$h[n] = \frac{1}{3}\delta[n] + \frac{1}{3}\delta[n-1] + \frac{1}{3}\delta[n-2]$$

The block diagram on the right shows the second-order moving-average filter discussed below. The transfer function is:

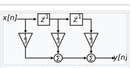
$$H(z) = \frac{1}{3} + \frac{1}{3}z^{-1} + \frac{1}{3}z^{-2} = \frac{1}{3}\frac{z^2 + z + 1}{z^2}.$$

The next figure shows the corresponding pole–zero diagram. Zero frequency (DC) corresponds to (1, 0), positive frequencies advancing counterclockwise around the circle to the Nyquist frequency at (−1, 0). Two poles are located at the origin, and two zeros are located at $z_1 = -\frac{1}{2} + j\frac{\sqrt{3}}{2}$, $z_2 = -\frac{1}{2} - j\frac{\sqrt{3}}{2}$.

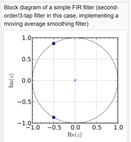
The frequency response, in terms of normalized frequency ω, is:

$$\begin{aligned} H\left(e^{i\omega}\right) &= \frac{1}{3} + \frac{1}{3}e^{-i\omega} + \frac{1}{3}e^{-2i\omega} \\ &= \frac{1}{3}e^{-i\omega}\left(1 + 2\cos(\omega)\right). \end{aligned}$$

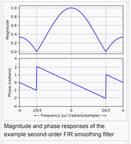
The magnitude and phase components of $H\left(e^{i\omega}\right)$ are plotted in the figure. But plots like these can also be generated by doing a discrete Fourier transform (DFT) of the impulse response.[B] And because of symmetry, filter design or viewing software often displays only the [0, π] region. The magnitude plot indicates that the moving-average filter passes low frequencies with a gain near 1 and attenuates high frequencies, and is thus a crude low-pass filter. The phase plot is linear except for discontinuities at the two frequencies where the magnitude goes to zero. The size of the discontinuities is π, representing a sign reversal. They do not affect the property of linear phase, as illustrated in the final figure.



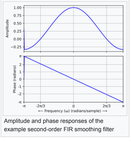
Block diagram of a simple FIR filter (second-order/3-tap filter in this case, implementing a moving average smoothing filter)



Pole–zero diagram of the example second-order FIR smoothing filter



Magnitude and phase responses of the example second-order FIR smoothing filter



Amplitude and phase responses of the example second-order FIR smoothing filter

See also [edit]

- Electronic filter
- Filter (signal processing)
- Infinite impulse response (IIR) filter
- Z-transform (specifically Linear constant-coefficient difference equation)
- FIR transfer function
- Filter design
- Cascaded integrator–comb filter
- Compact support

Notes [edit]

A. ^ An exception is MATLAB, which prefers units of *half-cycles/sample* = *cycles/2-samples*, because the Nyquist frequency in those units is 1, a convenient choice for plotting software that displays the interval from 0 to the Nyquist frequency.

B. ^ See § Sampling the DTFT.

References [edit]

1. ^ Oppenheim, Alan V., Willsky, Alan S., and Young, Ian T.,1983: Signals and Systems, p. 256 (Englewood Cliffs, New Jersey: Prentice-Hall, Inc.) ISBN 0-13-809731-3
2. ^ Rabiner, Lawrence R., and Gold, Bernard, 1975: Theory and Application of Digital Signal Processing (Englewood Cliffs, New Jersey: Prentice-Hall, Inc.) ISBN 0-13-914101-4
3. ^ A. E. Cetin, O.N. Gerek, Y. Yardimci, "Equiripple FIR filter design by the FFT algorithm," IEEE Signal Processing Magazine, pp. 60–64, March 1997.

Categories: Digital signal processing | Filter theory

Text is available under the Creative Commons Attribution-ShareAlike License 4.0; additional terms may apply. By using this site, you agree to the Terms of Use and Privacy Policy. Wikipedia® is a registered trademark of the Wikimedia Foundation, Inc., a non-profit organization.