

A non recursive filter



NON RECURSIVE FILTER

A non recursive filter is that in which the current output (y_n) is calculated solely from the current and previous input values ($x_n, x_{n-1}, x_{n-2}, \dots$). A recursive filter is one which in addition to input values also uses previous output values. These, like the previous input values, are stored in the processor's memory.

A digital filter that lacks feedback; that is, its output depends on present and past input values only and not on previous output values in non recursive filter

In non-recursive filters, the output y at the moment t is a function of only input values $x(t-z)$, $z > 1$ corresponding to the time moments $t-z$. A non-recursive filter is also known as an FIR (or Finite Impulse Response) filter.

A finite impulse response (FIR) filter is a type of a digital 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 N th-order FIR filter lasts for $N+1$ samples, and then dies to zero

Definition

The difference equation that defines the output of an FIR filter in terms of its input is:

where:

- $x[n]$ is the input signal,

- $y[n]$ is the output signal,
- b_i are the filter coefficients, and
- N is the filter order – an N th-order filter has $(N + 1)$ terms on the right-hand side; these are commonly referred to as taps.

This equation can also be expressed as a convolution of the coefficient sequence b_i with the input signal:

That is, the filter output is a weighted sum of the current and a finite number of previous values of the input.

Properties

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, and mastering.

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 frequencies (relative to the sample rate) cutoffs are needed.

Impulse response

The impulse response $h[n]$ can be calculated if we set in the above relation, where $d[n]$ is the Kronecker delta impulse. The impulse response for an FIR filter then becomes the set of coefficients b_n , as follows

FIR filters are clearly bounded-input bounded-output (BIBO) stable, since the output is a sum of a finite number of finite multiples of the input values, so can be no greater than times the largest value appearing in the input.

Filter design

To design a filter means to select the coefficients such that the system has specific characteristics. The required characteristics are stated in filter specifications. Most of the time filter specifications refer to the frequency response of the filter. There are different methods to find the coefficients from frequency specifications:

1. Window design method
2. Frequency Sampling method
3. Weighted least squares design
4. Minimax design
5. Equiripple design. 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 you can get to the desired response given that you can use only $(N + 1)$ coefficients. This method is particularly easy in practice since at least one text includes a program that takes the desired filter and N , and returns the optimum coefficients.

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

Some of the time, the filter specifications refer to the time-domain shape of the input signal the filter is expected to “recognize”. The optimum matched filter is to sample that shape and use those samples directly as the coefficients of the filter — giving the filter an impulse response that is the time-reverse of the expected input signal.

Window design method

In the Window Design Method, one designs an ideal IIR filter, and then applies a window function to it - in the time domain, multiplying the infinite impulse by the window function. This result in the frequency response of the IIR being convolved with the frequency response of the window function - thus the imperfections of the FIR filter (compared to the ideal IIR filter) can be understood in terms of the frequency response of the window function.

The ideal frequency response of a window is a Dirac delta function, as that results in the frequency response of the FIR filter being identical to that of the IIR filter, but this is not attainable for finite windows, and deviations from this yield differences between the FIR response and the IIR response.

Moving average example

Block diagram of a simple FIR filter (2nd-order/3-tap filter in this case, implementing a moving average)

A moving average filter is a very simple FIR filter. The filter coefficients are found via the following equation:

The following figure shows the absolute value of the frequency response. Clearly, the moving-average filter passes low frequencies with a gain near 1, and attenuates high frequencies. This is a typical low-pass filter characteristic. Frequencies above p are aliases of the frequencies below p , and are generally ignored or filtered out if reconstructing a continuous-time signal.