4. Digital filtering

4.1. Introduction to digital filters

Two main uses for filters:

- signal separation
  - signal is contaminated with interference, noise or other signals
- signal restoration
  - used when a signal has been distorted in some way

Analogue or digital filters may be used:

- analogue filters are cheap, fast, have large dynamic range
  - but performance is limited by electronics, e.g. accuracy and stability of resistors and capacitors
- digital filters can achieve much higher levels of performance
  - e.g. a low pass filter with a gain of 1 ± 0.0002 from DC to 1000 Hz, then less than 0.0002 for frequencies > 1001 Hz – impossible for electronic filter

4. Digital filtering

4.2 Filter parameters

Every linear filter has an **impulse response**, a **step response** and a **frequency response**.

- each contains complete information about the filter
- when one is specified, others are fixed and can be calculated

Digital filters can be implemented by convolution of signal with impulse response of filter or **filter kernel**.

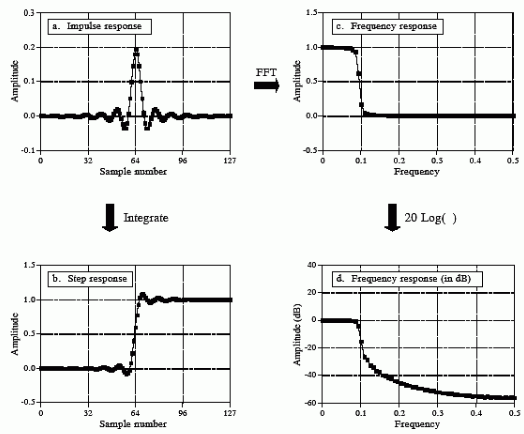


FIGURE 14-1
Filter parameters. Every linear filter has an impulse response, a step response, and a frequency response. The step response, (b), can be found by discrete integration of the impulse response, (a). The frequency response can be found from the impulse response by using the Fast Fourier Transform (FFT), and can be displayed either on a linear scale, (c), or in decibels, (d).

(figure : DSPGUIDE.COM)

4.1. Digital filtering

4.3. Decibels – a reminder (?)

The **bel** is named in honour of Alexander Graham Bell

- power is changed by factor of 10
- decibel (dB) is one tenth of a bel

So dB values of

–20 dB –10 dB 0 dB 10 dB 20 dB mean

power ratios of

0.01 0.1 1 10 100

$$dB = 10\log_{10}\frac{P_2}{P_1}$$

BUT we are usually dealing with signal amplitude and not power, proportional to square root of power, so

dB values of

–40 dB –20 dB 0 dB 20 dB 40 dB mean

amplitude ratios of

0.01 0.1 1 10 100

$$dB = 20\log_{10}\frac{A_2}{A_1}$$

dB used to compare **ratio** of two signals, but also for **absolute values**:

- dBV – signal is referenced to 1 V RMS signal (A_1)
- dBm – reference signal producing 1 mW power into 600 Ω load (P_1)

4. Digital filtering

4.4. Frequency domain responses

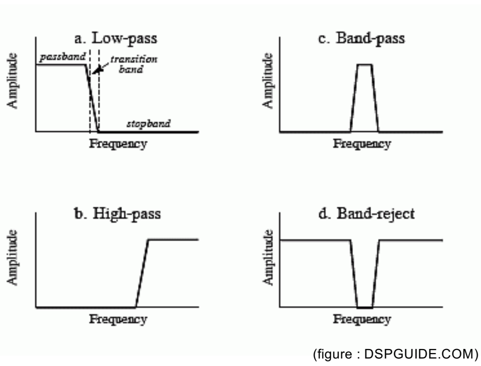
Four basic frequency responses:

- low pass, high pass, band pass, band reject / band stop

Passband: frequencies that are passed

Stopband: frequencies that are blocked

Transition band is between passband and stopband



(figure : DSPGUIDE.COM)

A **fast roll-off** means that the transition band is very narrow.

The division between the passband and transition band is called the **cutoff frequency.**

- for analogue filters often defined as where amplitiude reduced to 0.707 or –3 dB
- for digital filters many definitions eg 99%, 90%, 70.7%, 50% amplitude levels

4. Digital filtering

4.5 Frequency domain performance

To separate closely spaced frequencies a good filter will have

- a **fast roll-off**
- no **passband ripple**, as
- good **stopband attenuation**

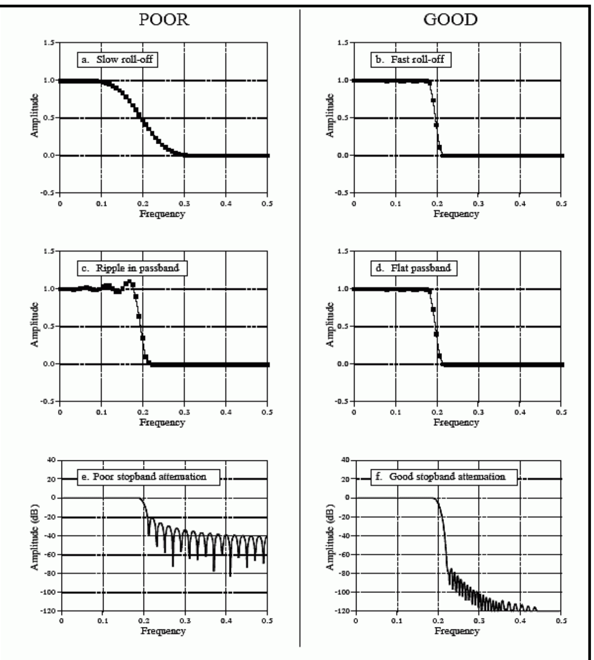


FIGURE 14-4
Parameters for evaluating *frequency domain* performance. The frequency responses shown are for low-pass filters. Three parameters are important: (1) roll-off sharpness, shown in (a) and (b), (2) passband ripple, shown in (c) and (d), and (3) stopband attenuation, shown in (e) and (f).

(figure : DSPGUIDE.COM)

4. Digital filtering

4.6 Time domain performance

Step response parameters that are important for good filter design

- a **fast rise-time**
- no **overshoot**, and
- minimal **phase distortion**

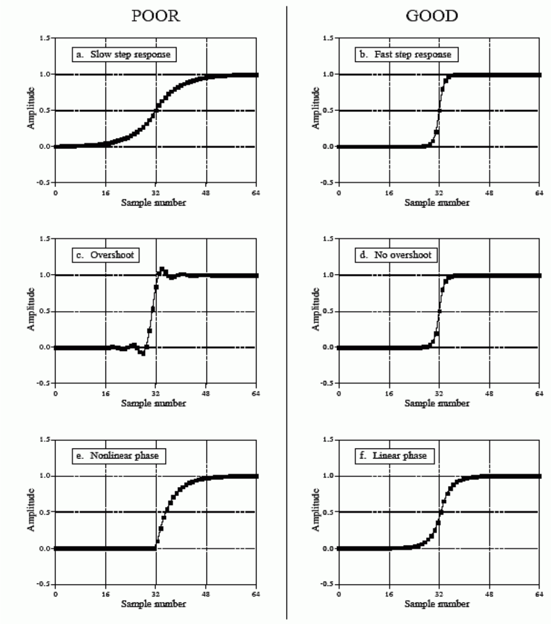


FIGURE 14-2
Parameters for evaluating *time domain* performance. The step response is used to measure how well a filter performs in the time domain. Three parameters are important: (1) transition speed (risetime), shown in (a) and (b), (2) overshoot, shown in (c) and (d), and (3) phase linearity (symmetry between the top and bottom halves of the step), shown in (e) and (f).

(figure : DSPGUIDE.COM)

4. Digital filtering

4.7 FIR and IIR filters

Two main types of filter:

- Finite Impulse Response (**FIR**) filters and
- Infinite Impulse Response (**IIR**) filters

4.7.1 Finite Impulse Response (FIR) filters

- implemented by convolution of signal with filter kernel
- filter kernel is of finite length
- example is windowed sinc filter, see TD3

4.7.1 Finite Impulse Response (FIR) filters (cont)

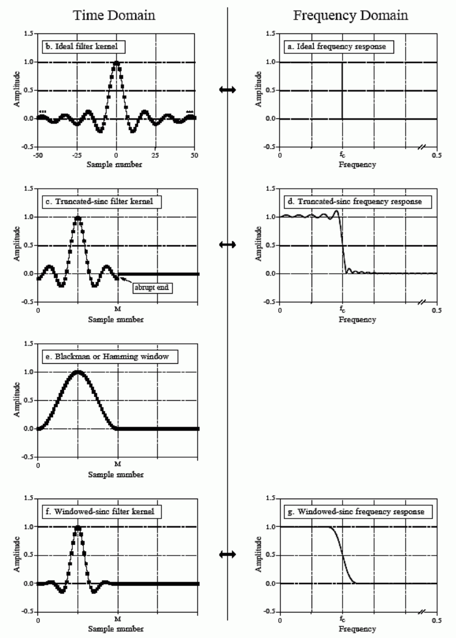


FIGURE 16-1

FIGURE 16-1 (facing page)
Derivation of the windowed-sinc filter kernel. The frequency response of the ideal low-pass filter is shown in (a), with the corresponding filter kernel in (b), a sinc function. Since the sinc is infinitely long, it must be truncated to be used in a computer, as shown in (c). However, this truncation results in undesirable changes in the frequency response, (d). The solution is to multiply the truncated-sinc with a smooth window, (e), resulting in the windowed-sinc filter kernel, (f). The frequency response of the windowed-sinc, (g), is smooth and well behaved. These figures are not to scale.

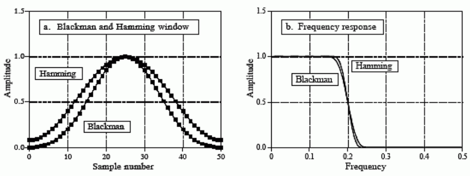
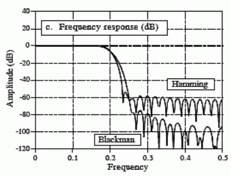


FIGURE 16-2
Characteristics of the Blackman and Hamming windows. The shapes of these two windows are shown in (a), and given by Eqs. 16-1 and 16-2. As shown in (b), the Hamming window results in about 20% faster roll-off than the Blackman window. However, the Blackman window has better stopband attenuation (Blackman: 0.02%, Hamming: 0.2%), and a lower passband ripple (Blackman: 0.02% Hamming: 0.2%).



(figures : DSPGUIDE.COM)

4. Digital filtering

4.7 FIR and IIR filters

4.7.2 Infinite Impulse Response (IIR) filters

- implemented by recursion (faster than convolution)

$$y[n] = a_0 x[n] + a_1 x[n-1] + a_2 x[n-2] + a_3 x[n-3] + \cdots$$
$$+ b_1 y[n-1] + b_2 y[n-2] + b_3 y[n-3] + \cdots$$

EQUATION 19-1
The recursion equation. In this equation, $x[\ ]$ is the input signal, $y[\ ]$ is the output signal, and the a's and b's are coefficients

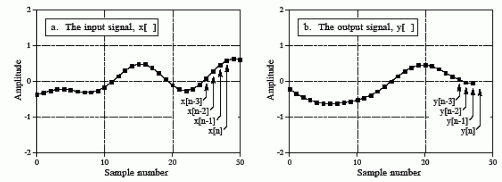


FIGURE 19-1
Recursive filter notation. The output sample being calculated, $y[n]$, is determined by the values from the input signal, $x[n]$, $x[n-1]$, $x[n-2]$, ..., as well as the *previously* calculated values in the output signal, $y[n-1]$, $y[n-2]$, $y[n-3]$, ... These figures are shown for $n = 28$.

- defined by a set of recursion coefficients
- impulse response is infinitely long
- example is Chebyshev filter (see dspguide.com)

4.7.2.1 Chebyshev filters

- used to separate one band of frequencies from another
- slightly lower performance than windowed-sinc

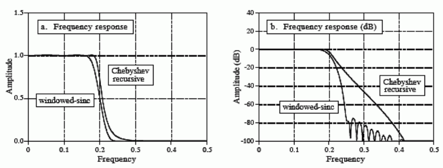


FIGURE 21-2
Windowed-sinc and Chebyshev frequency responses. Frequency responses are shown for a 51 point windowed-sinc filter and a 6 pole, 0.5% ripple Chebyshev recursive filter. The windowed-sinc has better stopband attenuation, but either will work in moderate performance applications. The cutoff frequency of both filters is 0.2, measured at an amplitude of 0.5 for the windowed-sinc, and 0.707 for the recursive.

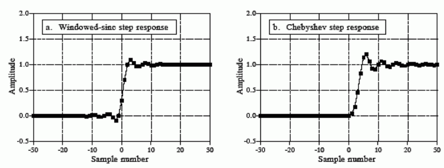
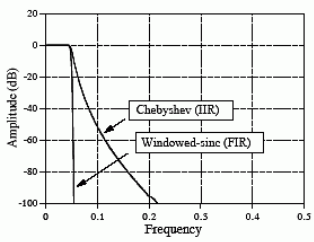


FIGURE 21-3
Windowed--sinc and Chebyshev step responses. The step responses are shown for a 51 point windowed-sinc filter and a 6 pole, 0.5% ripple Chebyshev recursive filter. Each of these filters has a cutoff frequency of 0.2. The windowed-sinc has a slightly better step response because it has less overshoot and a zero phase.

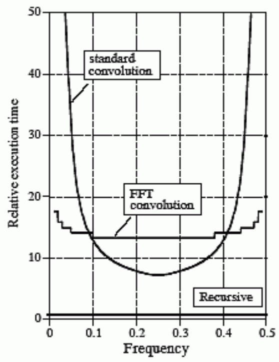
FIGURE 21-4
Maximum performance of FIR and IIR filters. The frequency response of the windowed-sinc can be virtually any shape needed, while the Chebyshev recursive filter is very limited. This graph compares the frequency response of a six pole Chebyshev recursive filter with a 1001 point windowed-sinc filter.



4.7.2.1 Chebyshev filters (cont)

- but much faster (typically an order of magnitude) as implemented by recursion rather than convolution

FIGURE 21-5
Comparing FIR and IIR execution speeds. These curves shows the relative execution times for a windowed-sinc filter compared with an equivalent six pole Chebyshev recursive filter. Curves are shown for implementing the FIR filter by both the standard and the FFT convolution algorithms. The windowed-sinc execution time rises at low and high frequencies because the filter kernel must be made longer to keep up with the greater performance of the recursive filter at these frequencies. In general, IIR filters are an order of magnitude faster than FIR filters of comparable performance.



- design is based on the z-transform ("digital Laplace transform")

4.7.2.1 Chebyshev filters (cont)

- Uses "poles" : what is a pole? Here are two answers. If you don't like one, maybe the other will help (DSPGUIDE.COM):

- Answer 1- The Laplace transform and z-transform are mathematical ways of breaking an impulse response into sinusoids and decaying exponentials. This is done by expressing the system's characteristics as one complex polynomial divided by another complex polynomial. The roots of the numerator are called zeros, while the roots of the denominator are called poles. Since poles and zeros can be complex numbers, it is common to say they have a "location" in the complex plane. Elaborate systems have more poles and zeros than simple ones. Recursive filters are designed by first selecting the location of the poles and zeros, and then finding the appropriate recursion coefficients (or analog components). For example, Butterworth filters have poles that lie on a circle in the complex plane, while in a Chebyshev filter they lie on an ellipse.

- Answer 2- Poles are containers filled with magic powder. The more poles in a filter, the better the filter works.