7B. FIR filter

1. Moving average filter
2. Windowed-sinc filter

[Smith, ch. 15, 16]

1. Moving Average Filter

The **moving average filter** is *optimal* for a common task:
reducing random noise while retaining a sharp step response.

This makes it appropriate to use for **time domain** encoded signals.
It is the *worst* filter for frequency domain encoded signals, with little ability to separate one band of frequencies from another.

Relatives of the moving average filter:

- **Gaussian, Blackman**, and **multiple-pass moving average**.

These have slightly better performance in the frequency domain, at the expense of increased computation time.

1.1 Implementation by Convolution

Averaging a number of points from the input signal to produce each point in the output signal:

$$y[i] = \frac{1}{M}\sum_{j=0}^{M-1} x[i+j]$$

where $x[\,]$ is the input signal, $y[\,]$ is the output signal, and M is the number of points used in the moving average.

This equation only uses points on **one side** of the output sample being calculated.

Example

- In a 5 point moving average filter, point 80 in the output signal is:

$$y[80] = \frac{1}{5}(x[80] + x[81] + x[82] + x[83] + x[84])$$

- Filtering 5000 samples with a 101 point moving average filter, results in 4900 samples of filtered data.

The group of points from the input signal can also be chosen **symmetrically** around the output point:

$$y[80] = \frac{1}{5}(x[78] + x[79] + x[80] + x[81] + x[82])$$

Symmetrical averaging requires that M be an *odd* number.

The moving average filter is a *convolution* using a very simple filter kernel.

Example

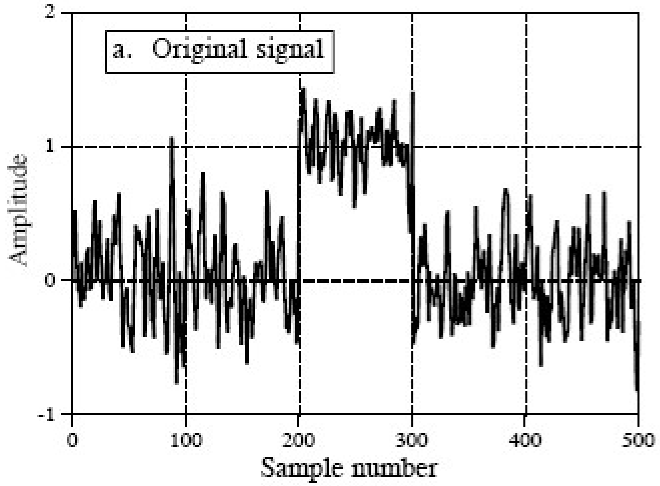
A 5 point filter has the filter kernel:

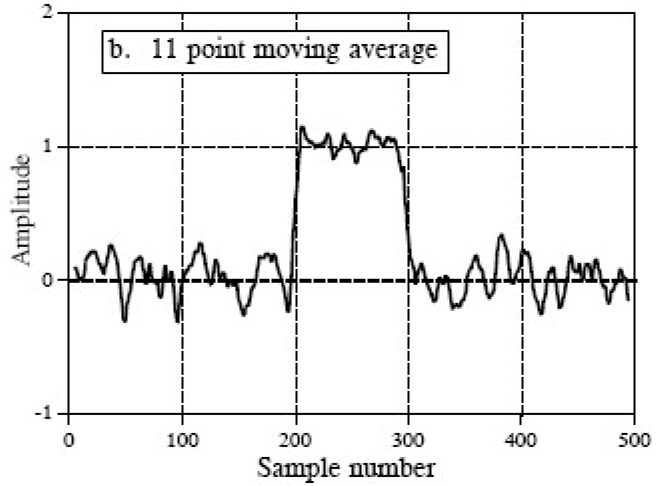
[1/5, 1/5, 1/5, 1/5, 1/5]

The **moving average** filter is a **convolution** of the input signal with a **rectangular pulse** having an area of **one**.

1.2 Noise Reduction vs. Step Response

The moving average filter is *optimal* for a common problem, reducing random white noise while keeping the sharpest step response.





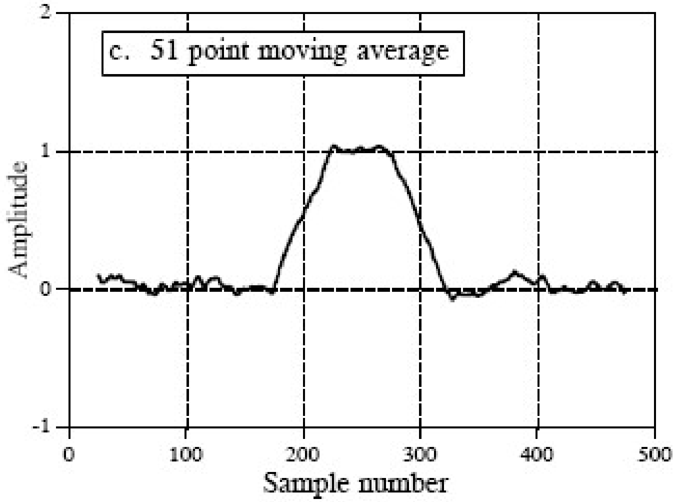


Fig. 10 Example of a moving average filter. In (a), a rectangular pulse is buried in random noise. In (b) and (c), this signal is filtered with 11 and 51 point moving average filters, respectively.

The amount of noise reduction is equal to the square-root of the number of points in the average.

Example

A 100 point moving average filter reduces the noise by a factor of 10.

Since the noise we are trying to reduce is random, it is useless to give preferential treatment to any one of the input points.

The lowest noise is obtained when all the input samples are treated equally, i.e., the moving average filter.

1.3 Frequency Response

The frequency response of the moving average filter is mathematically described by the Fourier transform of the rectangular pulse.

Frequency response of an M point moving average filter:

$$H[f] = \frac{\sin(\pi f M)}{M(\pi f)}$$

The frequency, f, runs between 0 and 0.5. For $f = 0$, use: $H[f] = 1$

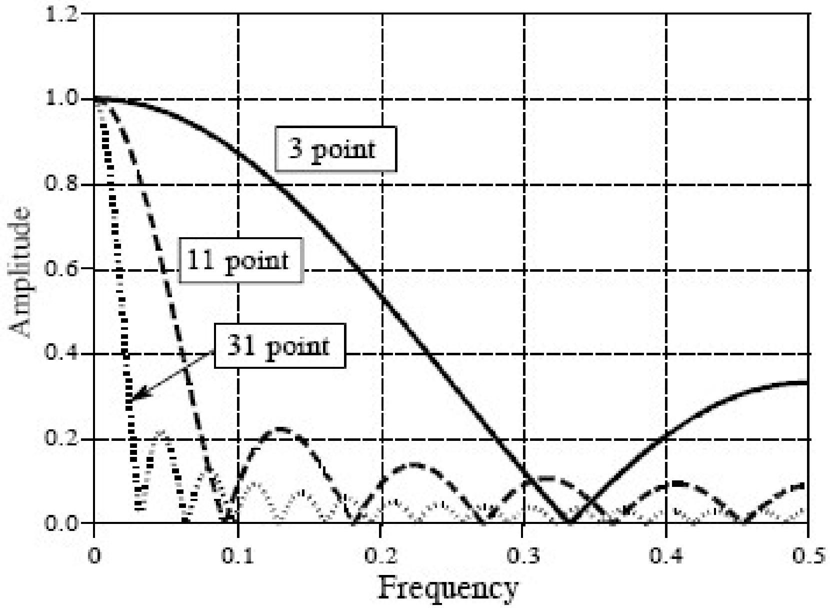


Fig. 11 Frequency response of the moving average filter.

The **moving average** is an exceptionally good *smoothing filter* (the action in the time domain), but an exceptionally bad *low-pass filter* (the action in the frequency domain) due to its slow roll-off and poor stopband attenuation.

The moving average filter cannot separate one band of frequencies from another.

1.4 Relatives of the Moving Average Filter

There are some applications where both domains (time and frequency) are simultaneously important. For instance, television signals:

- **Video information** is encoded in the time domain - the shape of the waveform corresponds to the patterns of brightness in the image.
- However, during transmission the video signal is treated according to its **frequency composition**, such as its total bandwidth, how the carrier waves for sound & color are added, elimination & restoration of the DC component, etc.

As another example, electromagnetic interference is best understood in the frequency domain, even if the signal's information is encoded in the time domain.

Relatives of the moving average filter have better frequency domain performance, and can be useful in these mixed domain applications.

Multiple-pass moving average filters involve passing the input signal through a moving average filter two or more times.

- The overall filter kernel resulting from two or three passes is like a *triangular* filter kernel (a rectangular filter kernel convolved with itself).
- After four or more passes, the equivalent filter kernel looks like a *Gaussian*.
- Multiple passes produce an "s" shaped step response, as compared to the straight line of the single pass.
- The frequency responses are given *multiplied* by itself for each pass. Each time domain convolution results in a multiplication of the frequency spectra.

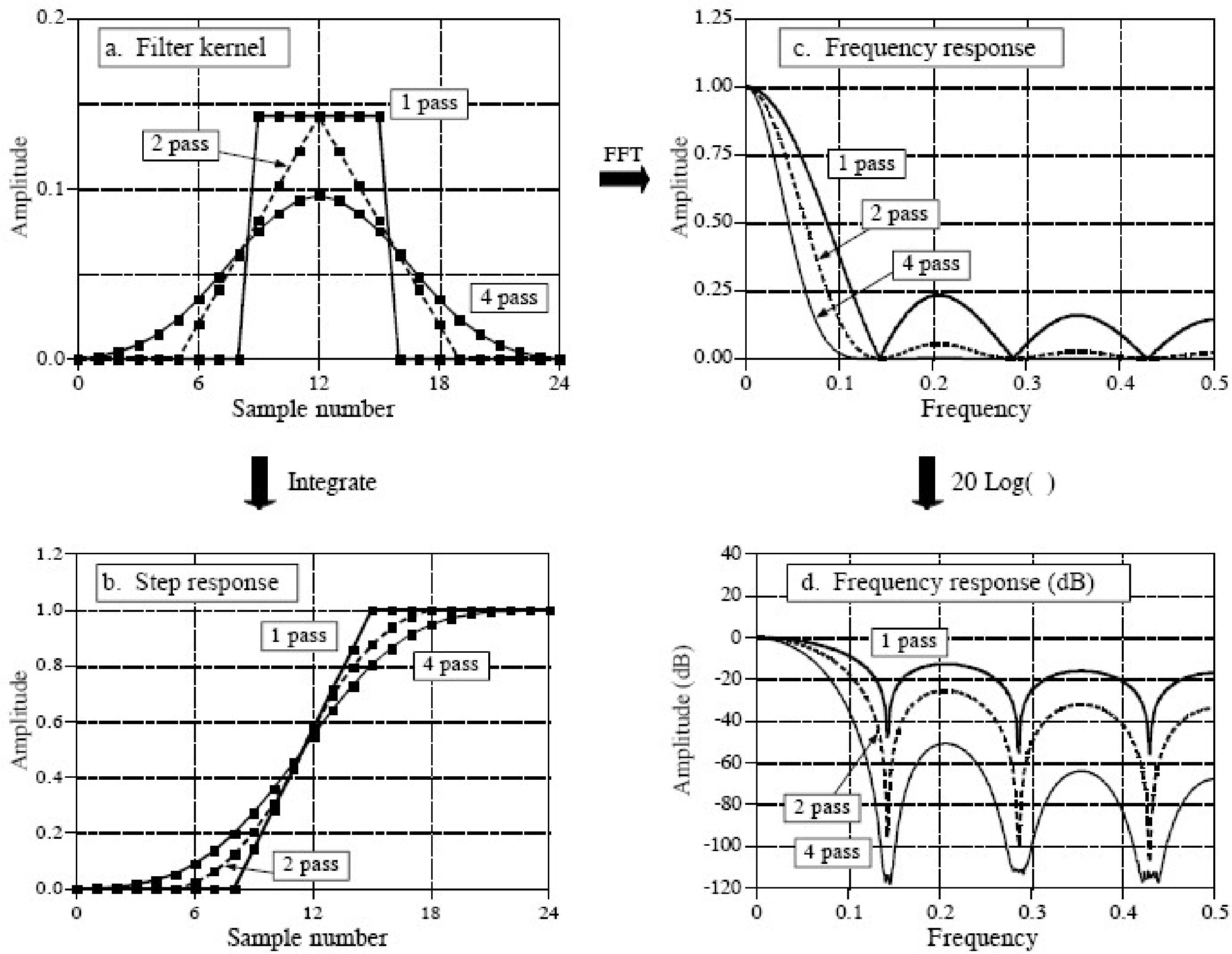


Fig. 12 Characteristics of **multiple-pass moving average** filters. Figure (a) shows the filter kernels resulting from passing a seven point moving average filter over the data once, twice and four times. Figure (b) shows the corresponding step responses, while (c) and (d) show the corresponding frequency responses.

Other relatives of the moving average filter:

- When a pure **Gaussian** is used as a filter kernel, the frequency response is also a Gaussian.
- The **Blackman window** is also used as a filter kernel.

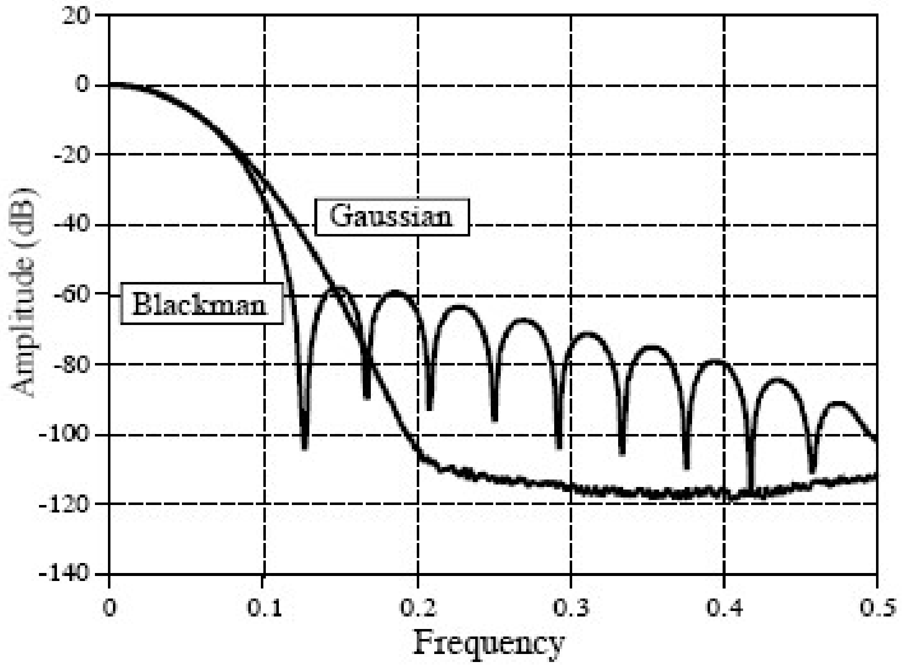


Fig. 13 Frequency response of the **Blackman** window and **Gaussian** filter kernels. Both these filters provide better stopband attenuation than the moving average filter. This has no advantage in removing random noise from time domain encoded signals, but it can be useful in mixed domain problems. The disadvantage of these filters is that they must use *convolution*, a slow algorithm.

How are these relatives better than the moving average filter itself:

1. These filters have better *stopband attenuation* than the moving average filter.
2. The filter kernels *taper* to a smaller amplitude near the ends, i.e. samples in the input signal that are farther away are given less weight than those close by.
3. The step responses are *smooth* curves, rather than the abrupt straight line of the moving average.

Risetime of filters

- If the risetime is measured from 0% to 100% of the step, the moving average filter is the best one.
- But measuring the risetime from 10% to 90% makes the Blackman window *better* than the moving average filter.

Execution speed

The biggest difference in these filters is *execution speed* :

1. Using a recursive algorithm, the moving average filter it is the *fastest* digital filter available.
2. Multiple passes of the moving average will be correspondingly slower, but still very quick.
3. The Gaussian and Blackman filters are slow, because they must use convolution – slower by a factor of ten times the number of points in the filter kernel (based on multiplication being about 10 times slower than addition).

For example, expect a 100 point Gaussian to be 1000 times slower than a moving average using recursion.

1.5 Recursive Implementation

The moving average filter can be implemented with a very fast algorithm. For example, imagine passing an input signal, x [ ], through a seven point moving average filter to form an output signal, y [ ]. E.g.

$$y[50] = x[47]+x[48]+x[49]+x[50]+x[51]+x[52]+x[53]$$
$$y[51] = x[48]+x[49]+x[50]+x[51]+x[52]+x[53]+x[54]$$

But if y [50] has already been calculated, the most *efficient* way to calculate y [51] is: $y[51] = y[50]+x[54]-x[47]$

After the first point is calculated in y [ ], all of **the other points** can be found with only **a single addition** and **subtraction per point**:

$$y[i] = y[i-1] + x[i+p] - x[i-q]$$

where $p = (M-1)/2, \; q = p+1$

This equation use two sources of data to calculate each point in the output:

- points from the input *and* previously calculated points from the output.

This is called a **recursive** equation, meaning that the result of one calculation is used in *future* calculations.

2. Windowed-Sinc Filter

Windowed-sinc filters are used to **separate one band** of frequencies **from another**:

- Their **exceptional frequency domain** characteristics are obtained at the expense of **poor performance in the time domain**.
- Carried out by standard convolution, windowed-sinc filters are easy to program, but slow to execute.
- The FFT can be used to improve the computational speed of these filters.

2.1 Strategy of the Windowed-Sinc

The *ideal* low-pass filter:

1. all frequencies below the cutoff frequency, f_C, are passed with unity amplitude, while all higher frequencies are blocked;
2. the passband is perfectly flat, the attenuation in the stopband is zero and the transition between the two is infinitesimally small.

The Inverse Fourier Transform of this ideal frequency response produces the ideal filter kernel (impulse response) - this curve is of the general form: $\sin(x)/x$, called the **sinc function**, given by:

$$h[i] = \frac{\sin(2\pi f_C i)}{i\pi}$$

Convolving an input signal with this filter kernel provides a *perfect* low-pass filter. The problem is, the sinc function **continues to both negative and positive infinity** without dropping to zero amplitude.

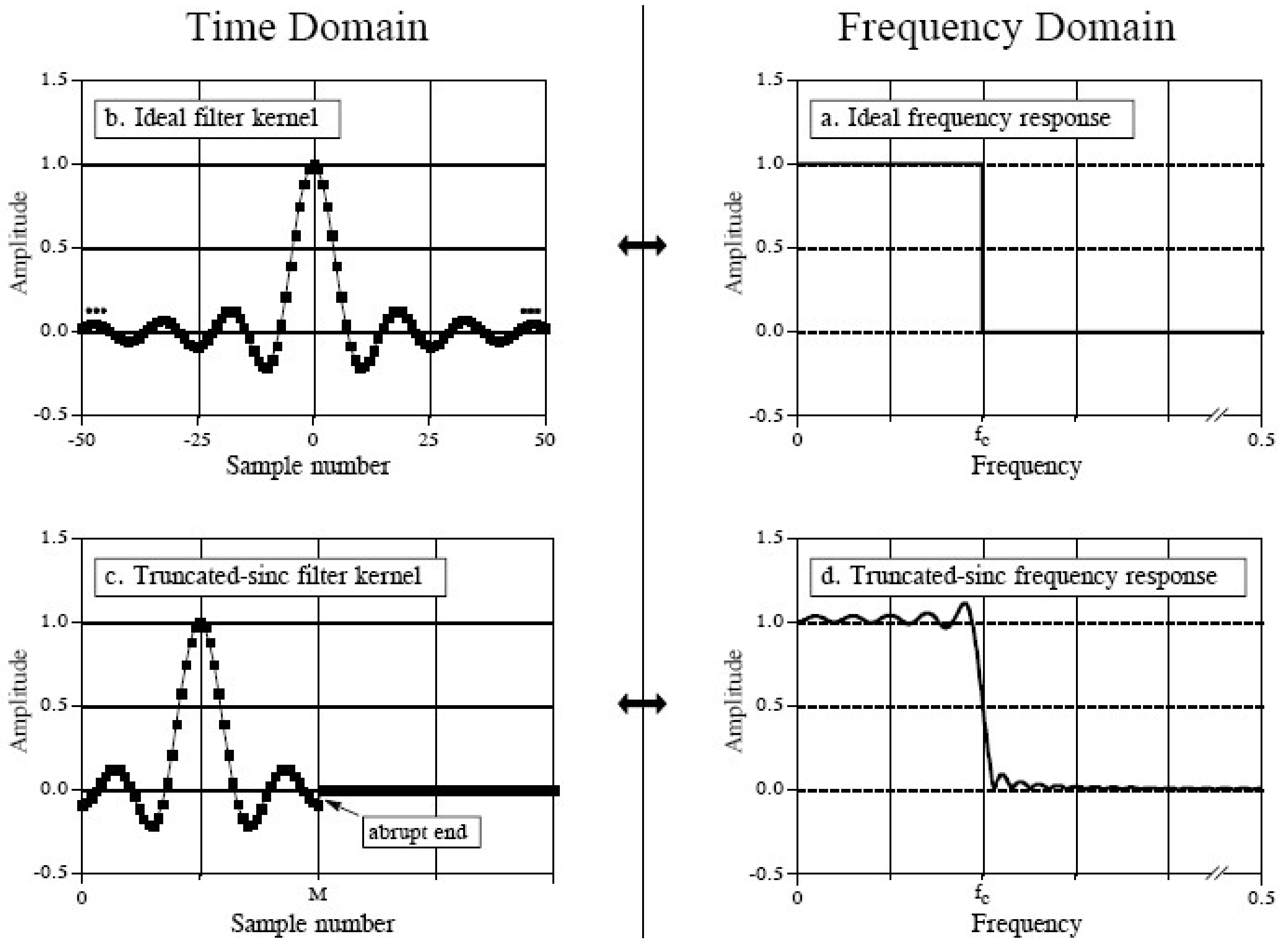


Fig. 14 Derivation of the windowed-sinc filter kernel. The frequency response of the ideal low-pass filter is shown in (a), with the corresponding filter kernel in (b), a sinc function. The sinc must be truncated to be used in a computer, (c). However, this truncation results in undesirable changes in the frequency response, (d).

We will make two modifications to the sinc function:

1. First, it is **truncated to *M*+1 points**, symmetrically chosen around the main lobe, where M is an even number.
2. Second, the entire sequence is **shifted to the right** so that it runs from 0 to M. This allows the filter kernel to be represented using only *positive* indexes.

The only one effect of this $M/2$ shift in the filter kernel is to shift the output signal by the same amount.

Windowing

Problems that result from the **abrupt discontinuity** at the ends of the **truncated sinc function** :

- in the frequency response there is **excessive ripple** in the passband and **poor attenuation** in the stopband (recall the Gibbs effect).

There is a simple method of **improving this situation**.

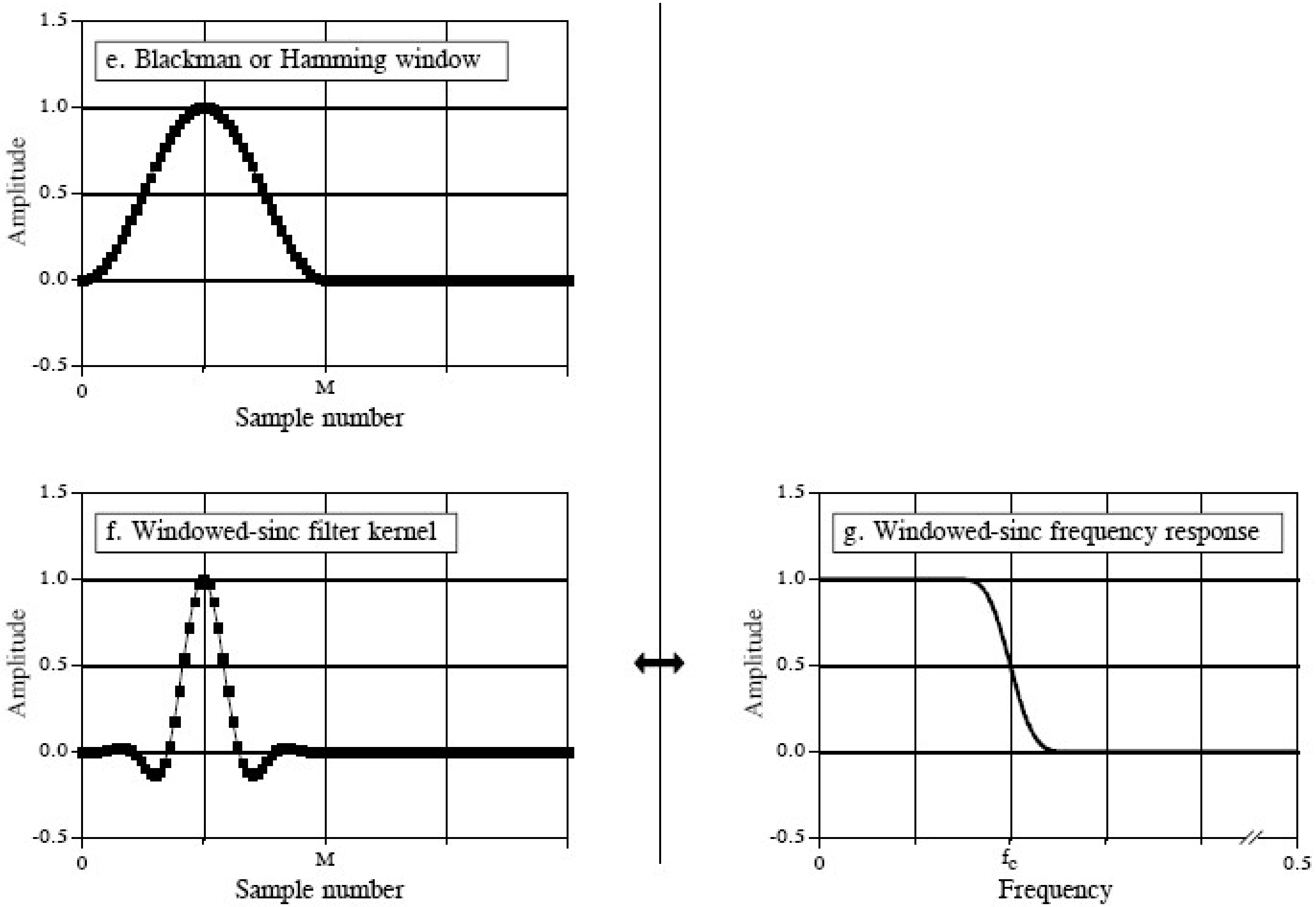


Fig. 14 (cont.) Derivation of the **windowed-sinc** filter kernel. The solution is to multiply the truncated-sinc with a smooth window, (e), resulting in the windowed-sinc filter kernel, (f). The frequency response of the windowed-sinc, (g), is smooth and well behaved.

Multiplying the **truncated-sinc** by the Blackman window results in the **windowed-sinc** filter kernel.
The idea is to reduce the abruptness of the truncated ends and thereby improve the frequency response.

Several different windows are available. Most important are: the **Hamming window** and the **Blackman window.**

The Hamming window

$$w[i] = 0.54 - 0.46 \cos(2\pi i / M)$$

The Blackman window

$$w[i] = 0.42 - 0.5 \cos(2\pi i / M) + 0.08 \cos(4\pi i / M)$$

These windows run from $i = 0$ to M, for a total of $M + 1$ points.

Which of these two windows should we use?

A trade-off between parameters appear:

1. The Hamming window has about a 20% faster *roll-off* than the Blackman.
2. However, the Blackman has a better *stopband attenuation* - the stopband attenuation for the Blackman is -74dB (-0.02%), while the Hamming is only -53dB (-0.2%).
3. The Blackman has a *passband ripple* of only about 0.02%, while the Hamming is typically 0.2%.

In general, the Blackman should be first choice:

- a slow roll-off is easier to handle than poor stopband attenuation.

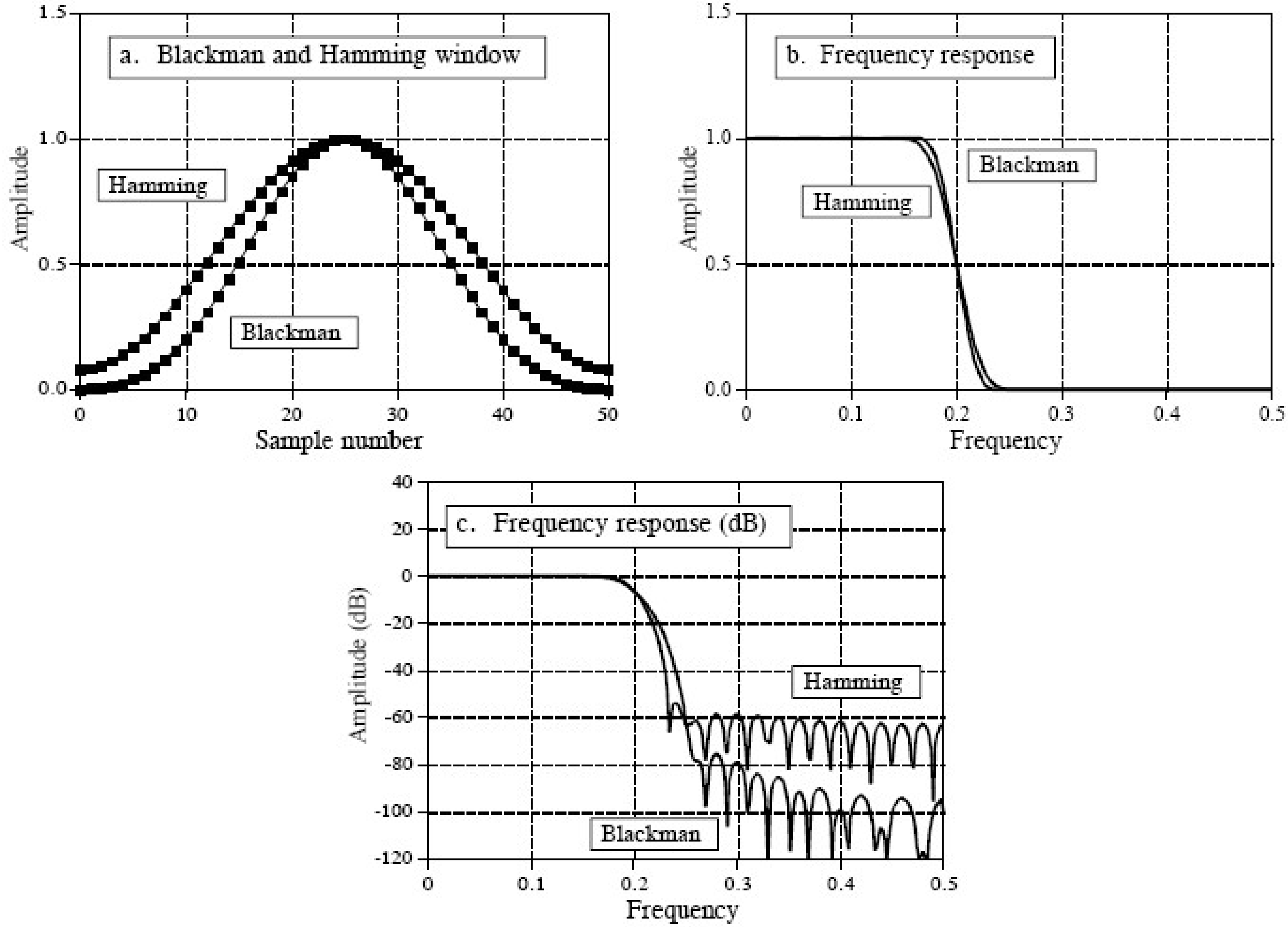


Fig. 15 Characteristics of the **Blackman and Hamming** windows. The Hamming window results in about 20% faster roll-off than the Blackman window. The Blackman window has better stopband attenuation (Blackman: 0.02%, Hamming: 0.2%), and a lower passband ripple (Blackman: 0.02% Hamming: 0.2%).

Other windows, but not better than the Blackman and Hamming:

- The **Bartlett window** is a triangle.
- The **Hanning window,** the **raised cosine window**, is given by:

$$w[i] = 0.5 - 0.5\cos(2\pi i / M).$$

These two windows have about the same roll-off speed as the Hamming, but worse stopband attenuation (Bartlett: -25dB or 5.6%, Hanning -44dB or 0.63%).

A **rectangular window** is the same as *no* window, just a truncation of the tails. While the roll-off is 2.5 times faster than the Blackman, the stopband attenuation is only -21dB (8.9%).

2.2 Designing the Filter

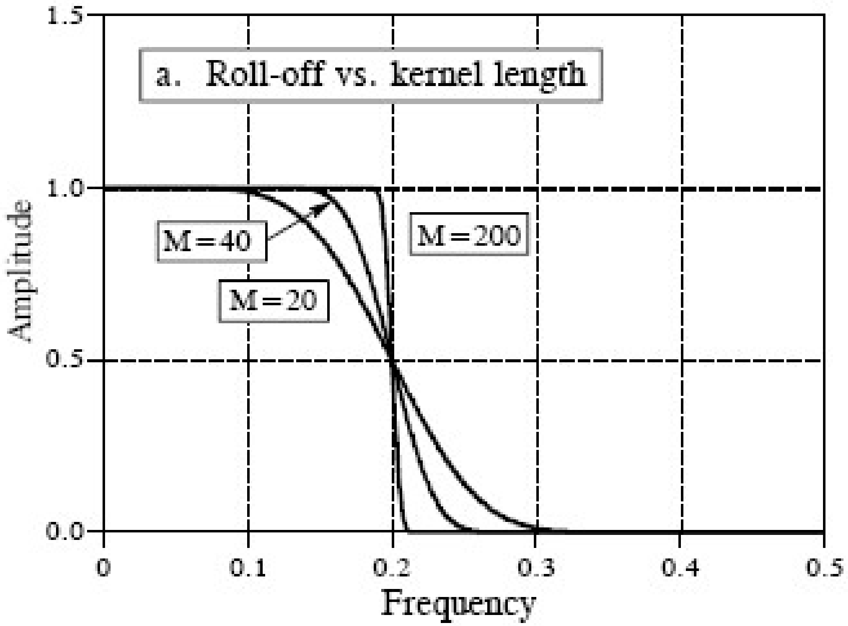
To design a **windowed-sinc, two parameters** must be selected:

1. the cutoff frequency, f_C, and
2. the length of the filter kernel, M.

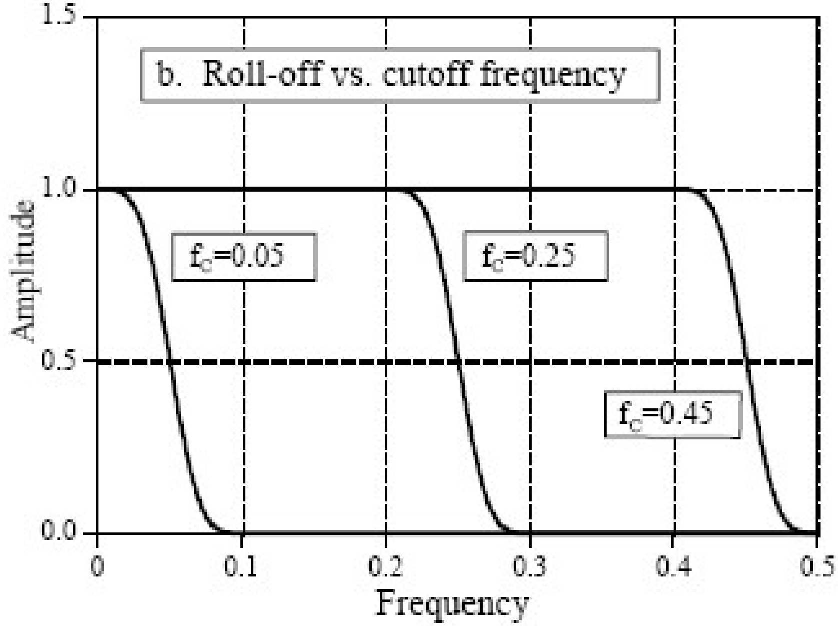
The **cutoff frequency** is expressed as a fraction of the sampling rate, and therefore must be between 0 and 0.5.
The **value for *M*** sets the *roll-off* according to the approximation:

$$M \approx \frac{4}{BW}$$

where BW is the width of the transition band, measured from where the curve just barely leaves one, to where it almost reaches zero (say, 99% to 1% of the curve).
The transition bandwidth is also expressed as a fraction of the sampling frequency, and must be between 0 and 0.5.



(a)



(b)

Fig. 16 Filter length vs. roll-off of the windowed-sinc filter:

- (a) for M = 20, 40, and 200, the transition bandwidths are BW = 0.2, 0.1, and 0.02 of the sampling rate, respectively.
- (b) the shape of the frequency response does not change with different cutoff frequencies. In (b), M = 60.

Trade-off between computation time and filter sharpeness.

A trade-off between *computation time* (depends on the value of *M*) and *filter sharpness* (the value of *BW*) is needed.

For instance, the 20% slower roll-off of the Blackman window (as compared with the Hamming) can be compensated for by using a filter kernel 20% longer. It could be said that the Blackman window is 20% slower to execute that an equivalent roll-off Hamming window.

The **cutoff frequency** (at 0.5 of the amplitude)

The cutoff frequency of the windowed-sinc filter is measured at the ***one-half amplitude*** point.

Why use 0.5 instead of the standard 0.707 (-3dB) for analog electronics and other digital filters?

- This is because the windowed-sinc's frequency response is **symmetrical** between the passband and the stopband.
- This symmetry makes the windowed-sinc ideal for *spectral inversion*.

The windowed-sinc **filter kernel**

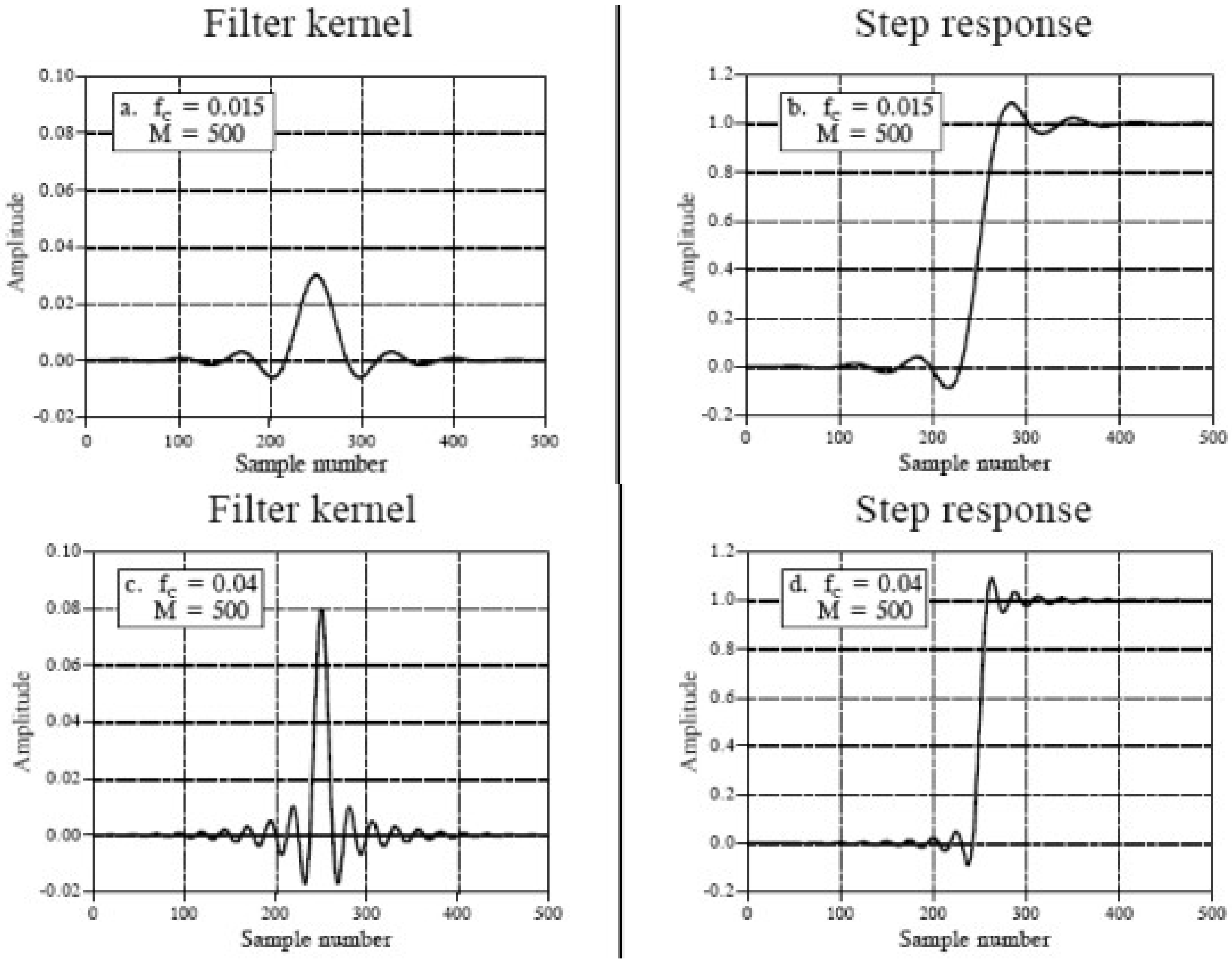
After f_C and M have been selected, the filter kernel is calculated from:

$$h[i] = K \frac{\sin(2\pi f_C (i - M/2))}{i - M/2} \left[0.42 - 0.5 \cos\left(\frac{2\pi i}{M} \right) + 0.08 \cos\left(\frac{4\pi i}{M} \right) \right]$$

Identify **three components**:

the *sinc function*, the *M/2 shift*, and the *Blackman window*.

- The cutoff frequency, f_C , is expressed as a fraction of the sampling rate, a value between 0 and 0.5.
- The length of the filter kernel is determined by M, which must be an even integer. The sample number i, is an integer that runs from 0 to M, resulting in M+1 total points in the filter kernel.
- The constant, K, is chosen to provide unity gain at zero frequency - the sum of all the samples is equal to one.
- To avoid a divide-by-zero error, for $i = M/2$, (at the center of the sinc) use $h[M/2] = 2\pi f_C K$.



Filter kernel
a. f_c = 0.015
M = 500
Amplitude
Sample number
0.10
0.08
0.06
0.04
0.02
0.00
-0.02
0
100
200
300
400
500
Step response
b. f_c = 0.015
M = 500
Amplitude
Sample number
1.2
1.0
0.8
0.6
0.4
0.2
0.0
-0.2
Filter kernel
c. f_c = 0.04
M = 500
Step response
d. f_c = 0.04
M = 500

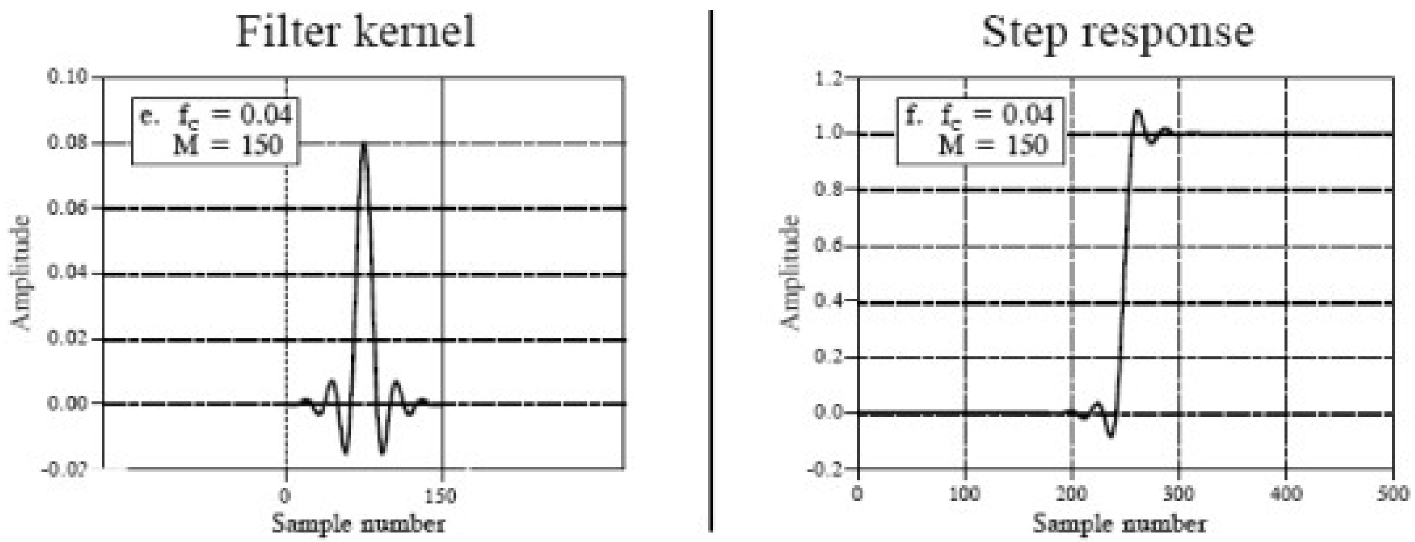


Fig. 17 Example **filter kernels** and the corresponding **step responses**.

The frequency of the sinusoidal oscillation is approximately equal to the cutoff frequency, f_C , while M determines the kernel length.

The windowed-sinc filter performs **badly** in the **time domain**: the step response has overshoot and ringing.
This is *not* a filter for signals with information encoded in the time domain.

Pushing it to the Limit

Extremely high performance of the windowed-sinc.

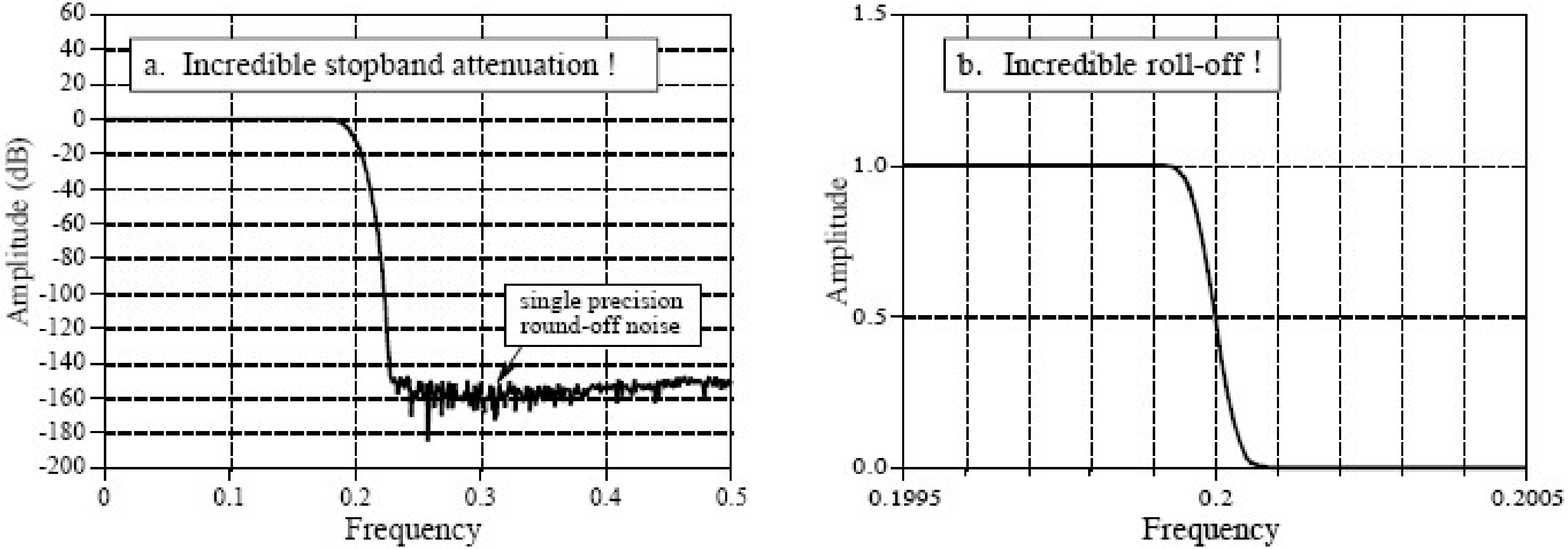


Fig. 18 The incredible performance of the windowed-sinc filter:
(a) increased stopband attenuation, -148dB, of a 201 point low-pass filter, formed by convolving a 101 point Blackman windowed-sinc kernel with itself;
(b) the very rapid roll-off a 32,001 point windowed-sinc filter; a roll-off of 0.000125 of the sampling rate.

Exercises 7

Task 7-1

An electroencephalogram, or EEG, is a measurement of the electrical activity of the brain (assume the sampling rate is 100 Hz). Different frequencies in the EEG can be identified with specific mental states:

- *alpha rhythm* - if you close your eyes and relax, the predominant EEG pattern will be a slow oscillation between about 7 and 12 hertz
- *beta rhythm* - opening your eyes and looking around causes the EEG to change to oscillation of 17 to 20 hertz.
- other frequencies.

A) Design a **digital low-pass** filter to separate the alpha from the beta rhythms in EEG signals (i.e. oscillation between about 7 and 12 Hertz from oscillation of 17 to 20 Hertz).

Determine the **type of filter kernel**, its **cutoff frequency**, the **transition bandwidth** (the length of filter kernel) and the **window type**.

B) Get the corresponding **high-pass filter** kernel.

Task 7-2

Design a ***band-pass filter*** to isolate a *signaling tone* in an audio signal, i.e. to isolate an 80 hertz band of frequencies centered on 2 kHz.
The sampling rate is 10.000 Hz.