WINDOWS

1. Ideal impulse response truncation (IRT)
2. IRT example: Ideal low-pass
3. IRT: Frequency domain effect
4. Gibbs phenomenon
5. FIR filter design by windows
6. Window parameters
7. Kinds of window
8. Windows in spectral estimation
9. The short-time Fourier transform (STFT)
10. Windows in the spectrogram

IDEAL IMPULSE RESPONSE TRUNCATION (IRT)

Let $d(n)$ be the impulse response of the ideal, or desired, frequency response $D^f(\omega)$.

$$d(n) = \text{IDTFT}\left\{D^f(\omega)\right\} = \frac{1}{2\pi}\int_{-\pi}^{\pi} D^f(\omega)\, e^{j\omega n} d\omega$$

For ideal filters,

1. $d(n)$ is of infinite duration
2. $d(n)$ is noncausal

The IRT method simply

1. truncates $d(n)$ to finite length, and
2. shifts $d(n)$ to obtain a causal filter.

IRT EXAMPLE: IDEAL LOWPASS

Let $D^f(\omega)$ be the response of an ideal lowpass filter with cutoff frequency ω_c.

$$D^f(\omega) = \begin{cases} 1 & |\omega| < \omega_c \\ 0 & \omega_c < |\omega| < \pi \end{cases}$$

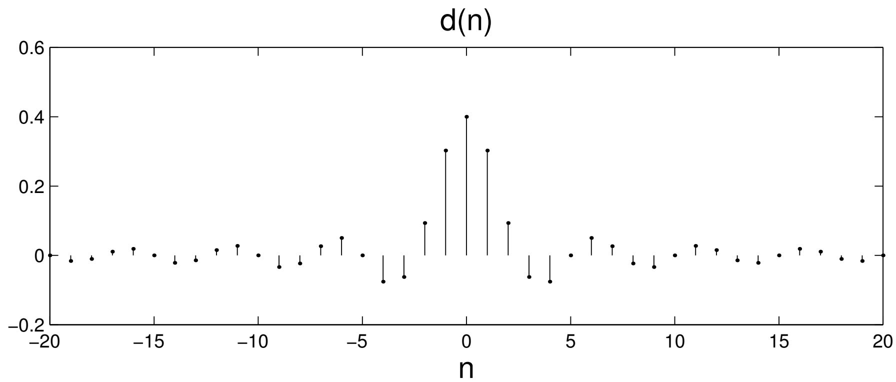
Then $d(n)$ is given by

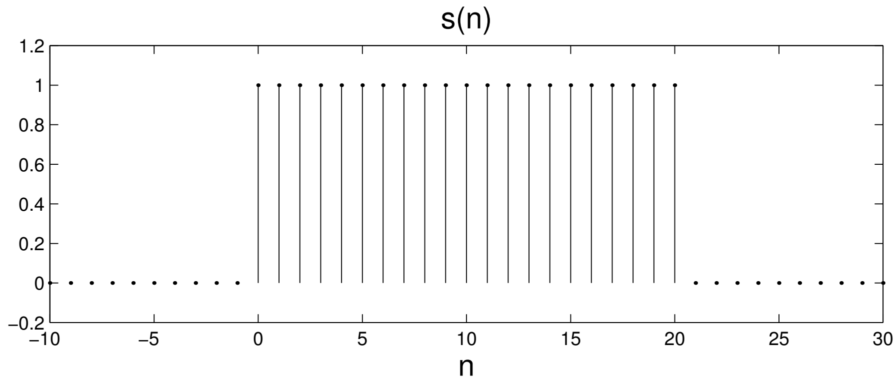
$$\boxed{d(n) = \text{IDTFT}\left\{D^f(\omega)\right\} = \frac{\omega_c}{\pi}\,\text{sinc}\left(\frac{\omega_c}{\pi} n\right)}$$

where the sinc function is defined as

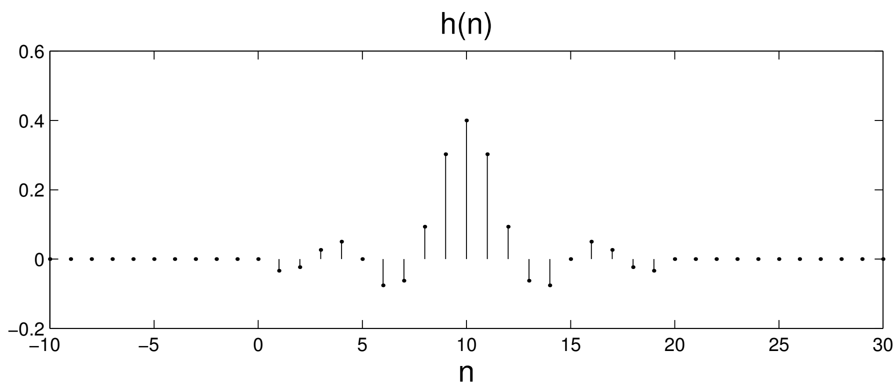
$$\text{sinc}(\theta) = \begin{cases} \frac{\sin(\pi\,\theta)}{\pi\,\theta} & n \neq 0 \\ 1 & n = 0 \end{cases} \tag{1}$$

IRT EXAMPLE





$$\Downarrow$$



IRT: FREQUENCY DOMAIN EFFECT

We can write the IRT procedure as

$$h(n) = d(n-M) \cdot s(n)$$

where M is the amount by which $d(n)$ is shifted and $s(n)$ is the rectangle function,

$$s(n) := \begin{cases} 1 & 0 \le n \le N-1 \\ 0 & \text{otherwise.} \end{cases}$$

Taking the DTFT of both sides and using the modulation property of the DTFT gives

$$H^f(\omega) = \frac{1}{2\pi} \left[D^f(\omega) \circledast S^f(\omega) \right] e^{-jM\omega}$$

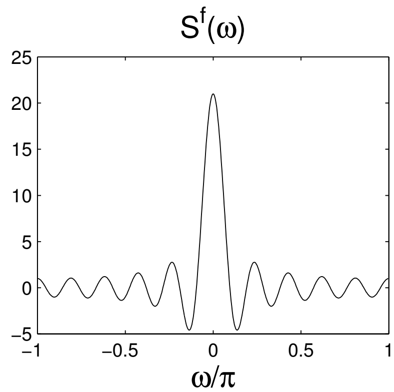
or

$$\boxed{A^f(\omega) = \frac{1}{2\pi} D^f(\omega) \circledast S^f(\omega).}$$

Recall that $S^f(\omega) = \text{DTFT}\,\{s(n)\}$ is the digital sinc function,

$$S^f(\omega) = \frac{\sin\left(\frac{N}{2}\,\omega\right)}{\sin\left(\frac{1}{2}\,\omega\right)}.$$

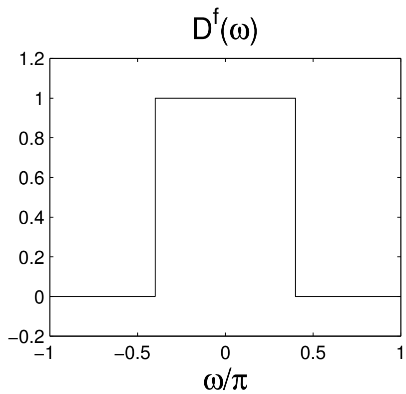
REMARKS ON THE DIGITAL SINC FUNCTION



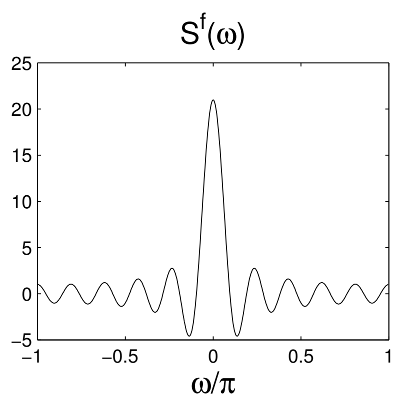
$$S^f(\omega) = \frac{\sin\left(\frac{N}{2}\,\omega\right)}{\sin\left(\frac{1}{2}\,\omega\right)}.$$

The digital sinc function $S^f(\omega)$ is also called the *Dirichlet kernel*.

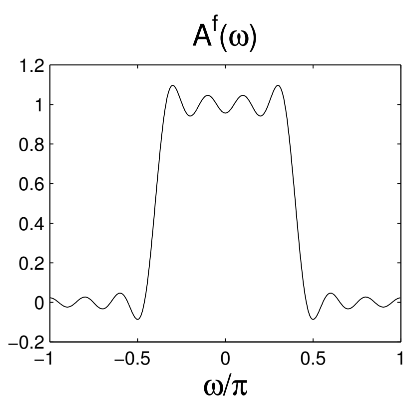
1. Its maximum values is N, occurring at $\omega = 0$.
2. The zeros closest to $\omega = 0$ are at $\omega = \pm\frac{2\pi}{N}$. The region between these two zeros is called the *main lobe*.
3. There are additional zeros at $\omega = \frac{2\pi}{N}k$ for $k = \pm 2, \ldots, N-1$. Between these zeros we have the *side lobes*.
4. The highest sidelobe (in absolute value) occurs at about $\omega = \pm\frac{3\pi}{N}$ and its values is about $2N/3\pi$. The ratio of the highest sidelobe to the main lobe is therefore $2/(3\pi)$ or about -13.5 dB.



$$\circledast$$

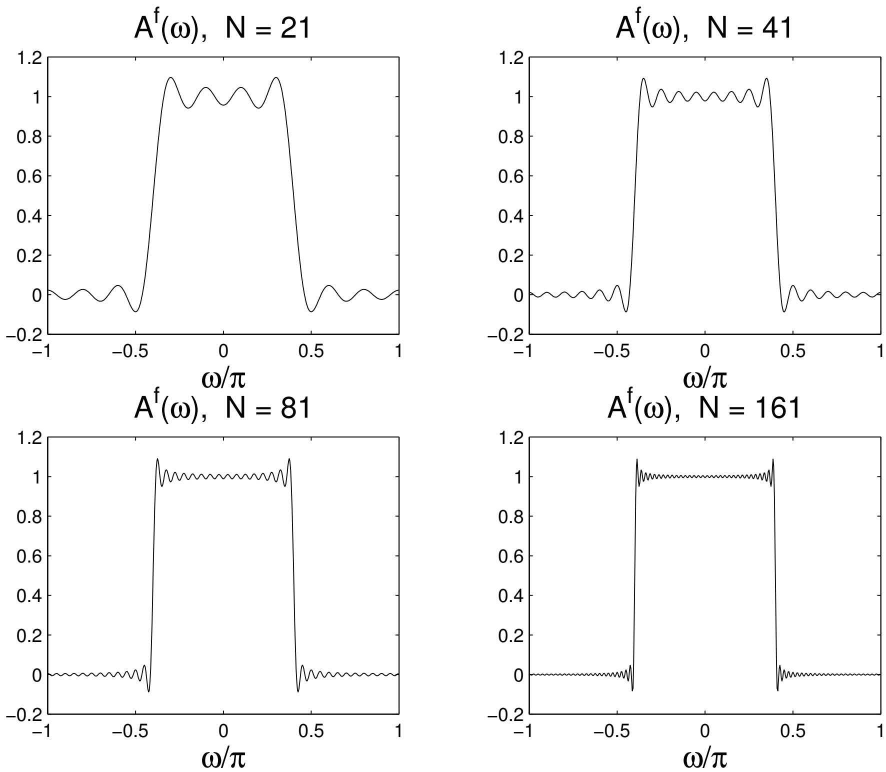


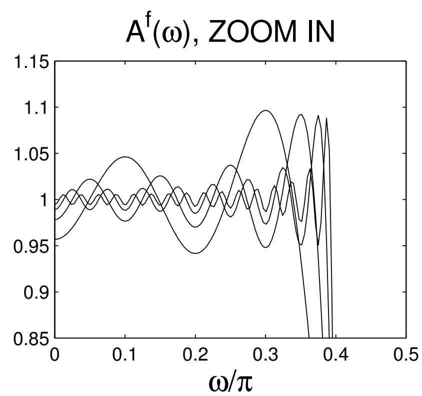
$$\Downarrow$$



There are two effects: (1) The cutoff of the lowpass filter is "smeared". (2) There are ripples in the pass-band and stop-band.

Sec 7.6.3
in Mitra





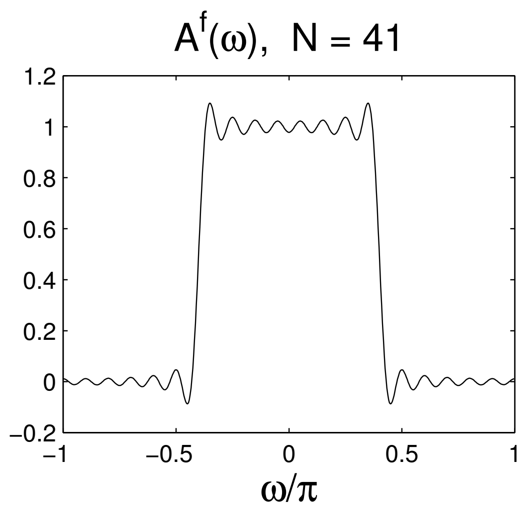
When obtaining a filter by truncating the ideal impulse response, the size of the peak error does not diminish to zero, even as the filter length is increased. This is known as *Gibb's phenomenon*. This reveals a problem with the IRT approach to filter design.

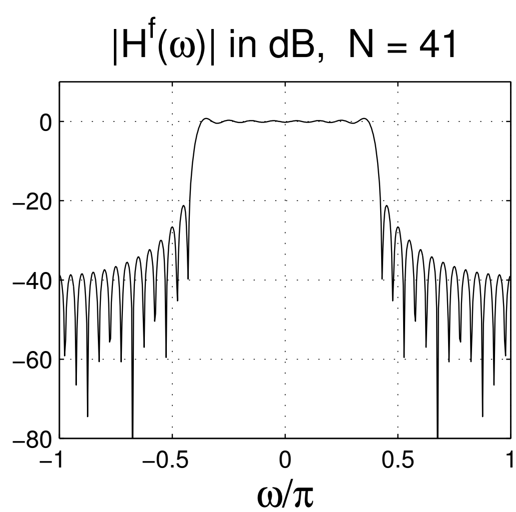
Displaying the frequency response of a filter in decibels (dB) magnifies the stop-band behavior of a filter. That is

$$20 \log_{10}(|H^f(\omega)|)$$

or equivalently,

$$10 \log_{10}(|H^f(\omega)|^2).$$





The maximum stop-band ripple is at about 21 dB, which is considered very poor.

FIR FILTER DESIGN BY WINDOWS

The windowing technique proposes that an FIR impulse response $h(n)$ be obtained from the infinitely supported ideal impulse response $d(n)$ by

1. truncating $d(n)$ to finite length,
2. multiplying by a function (a "window") that is tapered near its endpoints, and
3. shifting $d(n)$ so that it is causal.

$$h(n) = d(n - M) \cdot w(n).$$

WINDOWING: FREQUENCY DOMAIN EFFECT

Taking the DTFT of both sides and using the modulation property of the DTFT gives

$$H^f(\omega) = \frac{1}{2\pi} \left[D^f(\omega) \circledast W^f(\omega) \right] e^{-jM\omega}$$

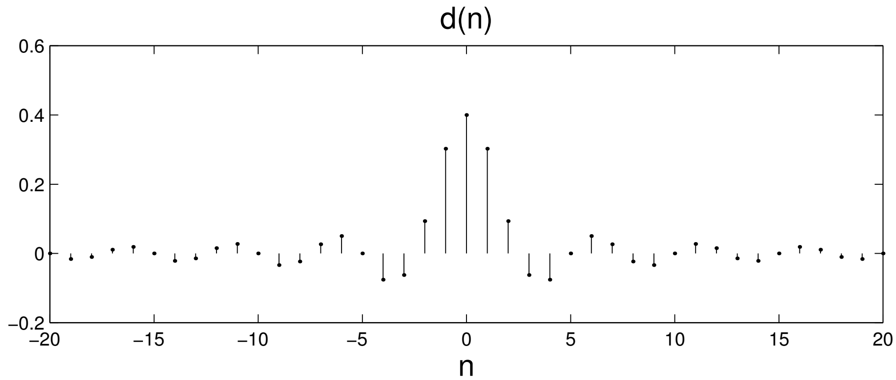
or

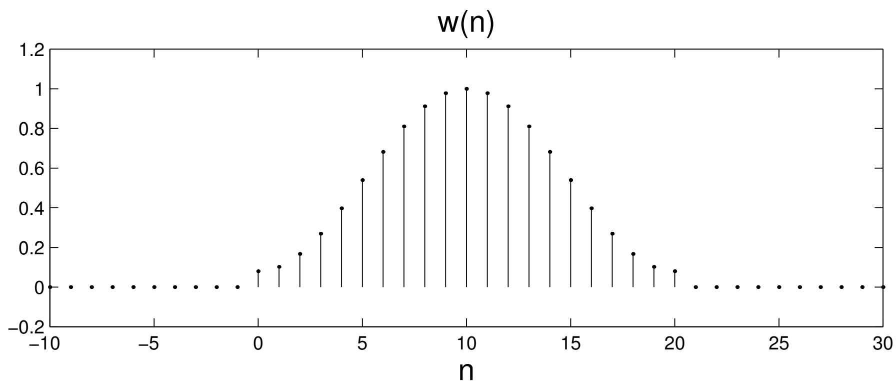
$$\boxed{A^f(\omega) = \frac{1}{2\pi} D^f(\omega) \circledast W^f(\omega)}$$

where

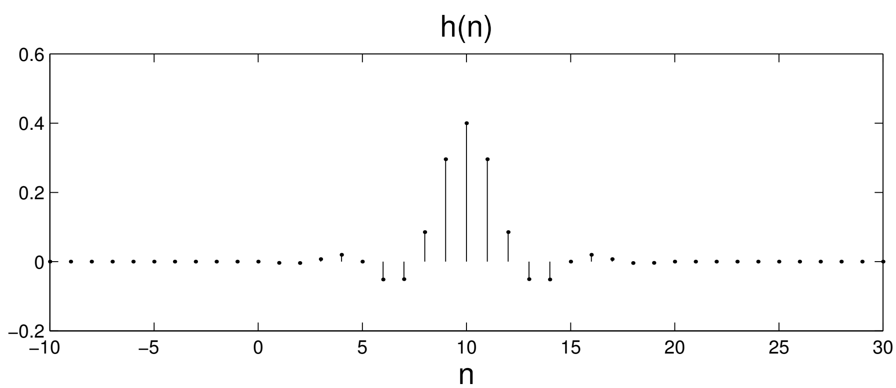
$$W^f(\omega) = \text{DTFT}\left\{ w(n) \right\}.$$

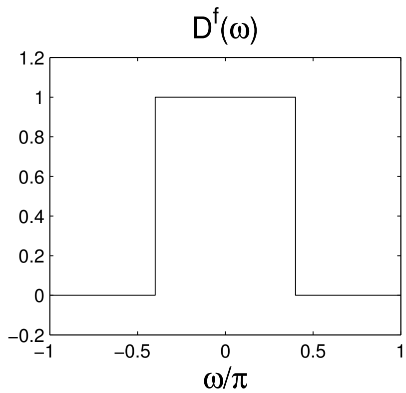
WINDOWING: EXAMPLE



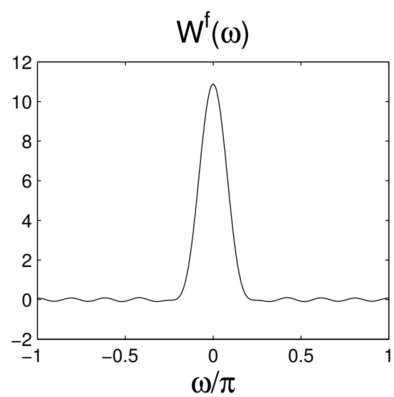


$\Downarrow$

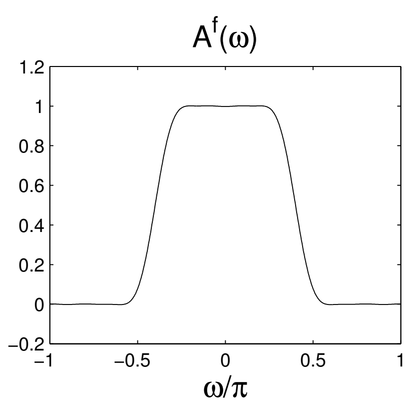




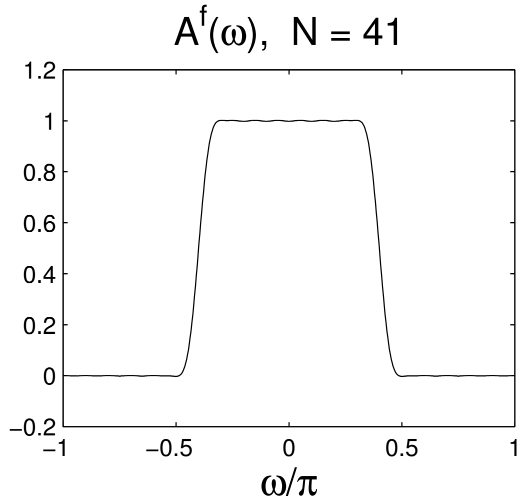
$\circledast$

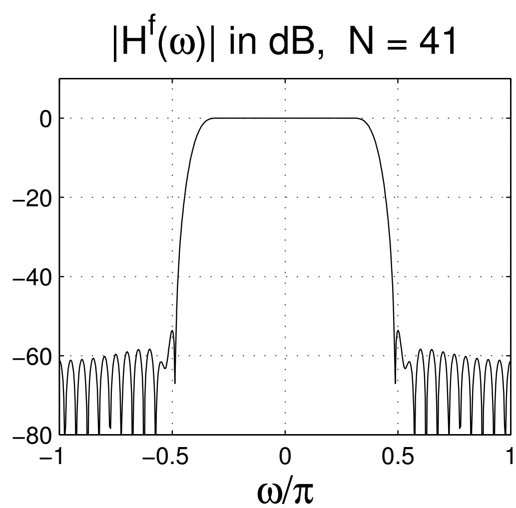


$\Downarrow$



Compare with figures on page 6. Wider transition (due to wider mainlobe) and smaller ripples (due to smaller side lobes).





Compare to the figures on page 8. Now the stop-band attenuation is down to 53 dB, but the transition between the pass-band and stop-band is much wider.

WINDOW PARAMETERS

Sec 7.6.4 in Mitra

Roughly,

1. the mainlobe width effects the width of the transition band of $H^f(\omega)$.
2. the relative sidelobe height effects the size of the ripples in $H^f(\omega)$.

The frequency response of a good window function has

1. a narrow mainlobe,
2. a small relative peak sidelobe height, and
3. good sidelobe roll-off.

For a good window, $W^f(\omega)$ should approximate a delta function: $W^f(\omega) = 2\pi\,\delta(\omega)$, for $|\omega| < \pi$. The mainlobe width is zero, and there are no sidelobes. Then it would not smear $D^f(\omega)$ so much. However, the window $w(n)$ corresponding to $W^f(\omega) = 2\pi\,\delta(\omega)$ is $w(n) = 1$ with infinite duration.

> For a fixed window length, there is a trade-off between mainlobe width and relative sidelobe height. They can not be made arbitrarily good at the same time.

The rectangular window $s(n)$, has a very narrow mainlobe, but very high sidelobes. Usually, it is desired to decrease the sidelobe height at the expense of a wider mainlobe.

Some windows provide a parameter that can be varied to control this trade-off (Kaiser window, Prolate spheroidal window, Dolph-Chebyshev window).

NON-ADJUSTABLE WINDOWS

Sec 7.6.4 in Mitra

Many different window functions have been introduced, each having somewhat different properties.

- Generalized cosine windows

$$w(n) = a - b\cos\left(\frac{2\pi(n+1)}{(N+1)}\right) + c\cos\left(\frac{4\pi(n+1)}{(N+1)}\right) \qquad (2)$$

for $n = 0, \dots, N-1$ where a, b and c are:

Window	a	b	c
Rectangular	1	0	0
Hann	0.5	0.5	0
Hamming	0.54	0.46	0
Blackman	0.42	0.5	0.08

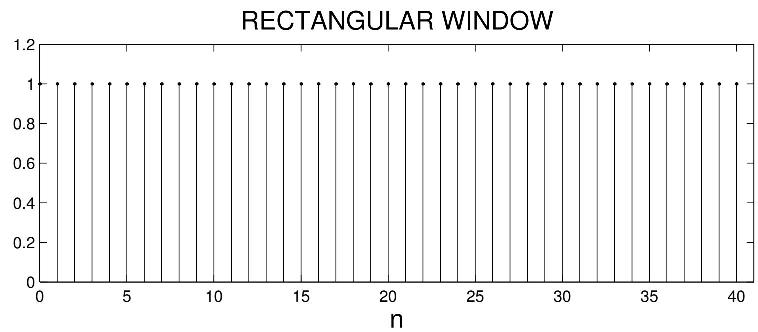
Although definitions vary slightly from text to text, the constants a, b, c are standard.

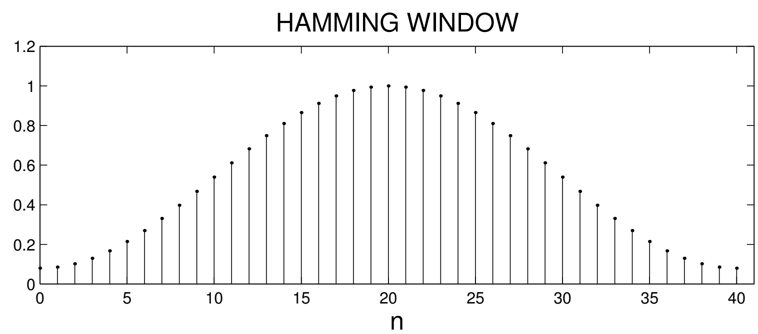
- Bartlett (triangular) window

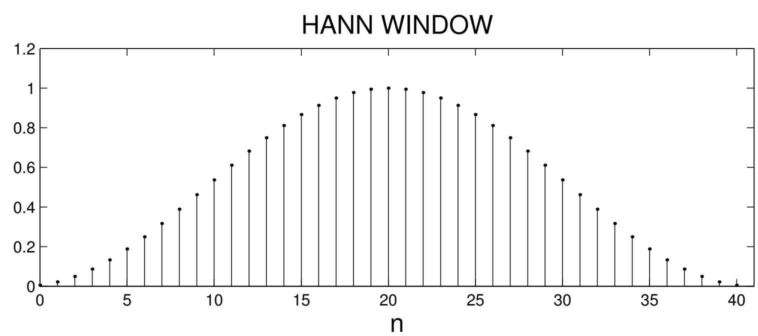
$$w(n) = \begin{cases} \frac{2(n+1)}{N-1} & 0 \le n \le \frac{N-1}{2} \\ 2 - \frac{2(n+1)}{N-1} & \frac{N-1}{2} < n \le N \end{cases} \qquad (3)$$

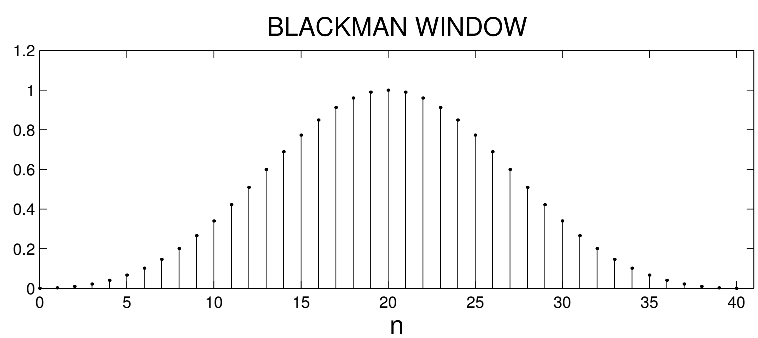
for $n = 0, \dots, N-1$.

NON-ADJUSTABLE WINDOWS

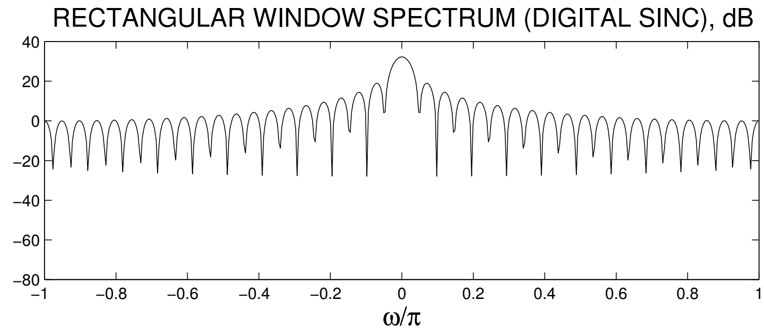


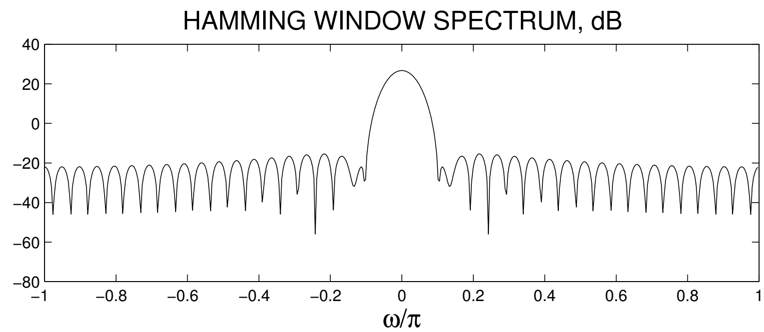


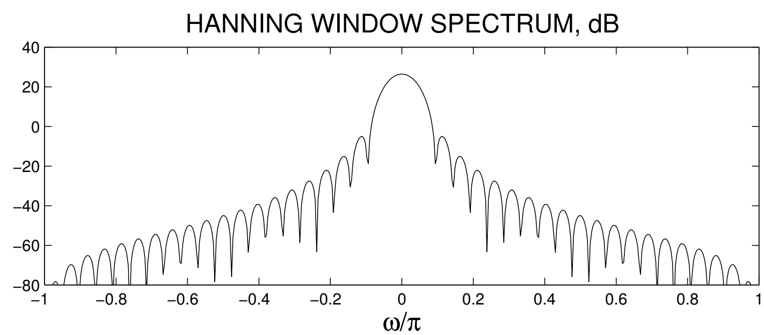


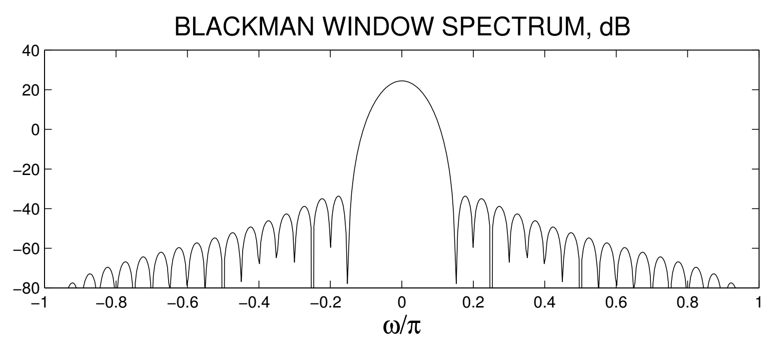


NON-ADJUSTABLE WINDOWS

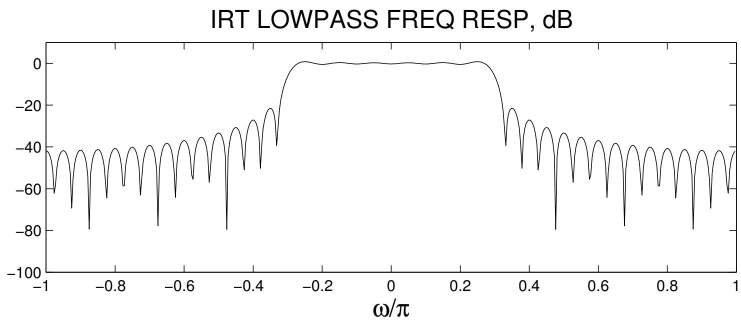


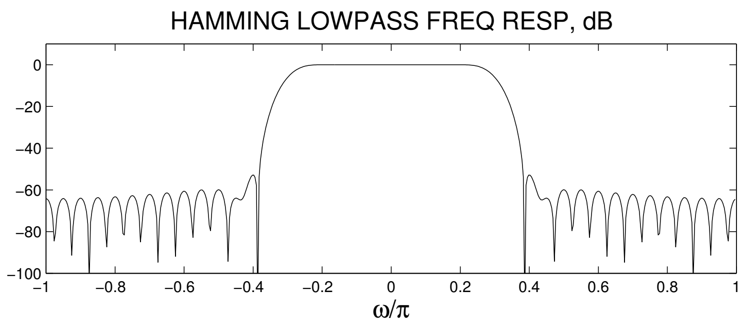


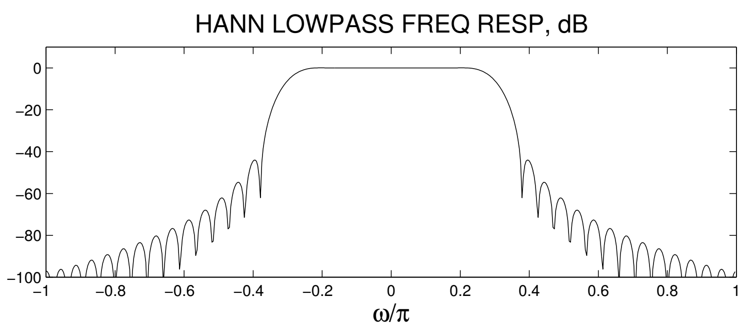


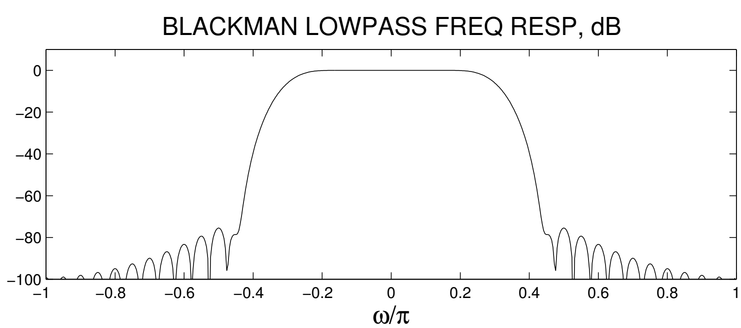


NON-ADJUSTABLE WINDOWS









Windowing $d(n)$ greatly reduces the ripple size, with a corresponding increase in transition width.

One approach to window design computes the window sequence that has most of its energy in a given frequency band, say $[-B, B]$. The question of simultaneous concentration in both time and frequency was addressed by Slepian [?]. Specifically, the problem is formulated as follows. Find $w(n)$ of specified finite support that maximizes

$$\lambda = \frac{\int_{-B}^{B} |W(\omega)|^2 d\omega}{\int_{-\pi}^{\pi} |W(\omega)|^2 d\omega} \tag{4}$$

where

$$W(\omega) = \text{DTFT}\left\{w(n)\right\}.$$

The solution is a particular discrete prolate spheroidal (DPS) sequence, that can be normalized so that $W(0) = 1$. The solution to this problem was traditionally found by finding the largest eigenvector[1] of a matrix whose entries are samples of the sinc function [?]. However, that eigenvalue problem is numerically ill conditioned — the eigenvalues cluster closely to 0 and 1. Recently, an alternative eigenvalue problem has become more widely known, that has exactly the same eigenvectors as the first eigenvalue problem (but different eigenvalues), and is numerically well conditioned [?, ?, ?]. The well conditioned eigenvalue problem is described by $\mathbf{A}\mathbf{v} = \theta\mathbf{v}$ where $\mathbf{A}$ is tridiagonal and has the following form:

$$\mathbf{A}_{i,j} = \begin{cases} \frac{1}{2} i(N-i) & j = i-1 \\ (\frac{N-1}{2} - i)^2 \cos B & j = i \\ \frac{1}{2}(i+1)(N-1-i) & j = i+1 \\ 0 & |j-i| > 1 \end{cases} \tag{5}$$

for $i, j = 0, \ldots, N-1$. Again, the eigenvector with the largest eigenvalue is the sought solution.

[1] The eigenvector with the largest eigenvalue.

The advantage of $\mathbf{A}$ in (5) over the first eigenvalue problem is twofold.

1. The eigenvalues of $\mathbf{A}$ in (5) are well spread (so that the computation of its eigenvectors is numerically well conditioned).
2. The matrix $\mathbf{A}$ in (5) is tridiagonal, facilitating the computation of the largest eigenvector via the power method.

By varying the bandwidth B, a family of DPS windows is obtained. By design, these windows are optimal in the sense of energy concentration. They have good mainlobe width and relative peak sidelobe height characteristics. However, it turns out that the sidelobe roll-off of the DPS windows is relatively poor [?].

The Kaiser [?] and Saramäki [?, ?] windows were designed to approximate this prolate spheroidal sequence. Computing the Kaiser and Saramäki windows does not require the solution to an eigenvalue problem — they were originally developed in order to avoid the numerically ill conditioning of the first matrix eigenvalue problem described above.

KAISER WINDOW

Sec 7.6.5 in Mitra

Kaiser's approximation to the prolate spheroidal window [?] is given by

$$w(n) = \frac{I_o\left(\beta\sqrt{1-\left(\frac{n}{M}\right)^2}\right)}{I_o(\beta)} \qquad -M \le n \le M \tag{6}$$

where β is an adjustable parameter and $I_o(x)$ is the modified zeroth-order Bessel function of the first kind. β controls the tradeoff between the mainlobe width and the peak sidelobe level — it is typically chosen to lie between 2 and 10. High values of β produce filters having high stopband attenuation, but wide transition widths. To attain a stopband attenuation of $A_s = -20\log_{10}\delta_s$, Kaiser suggests that β be chosen according to the empirically found formula

$$\beta = \begin{cases} 0.1102(A_s - 8.7) & A_s > 50 \\ 0.5842(A_s-21)^{0.4} + 0.07886(A_s - 21) & 21 < A_s < 50 \\ 0 & A_s < 21 \end{cases} \tag{7}$$

Note that the suggested value of β does not depend on the desired transition width. To attain a transition width of $\omega_p - \omega_s$ it is suggested that the filter length be chosen according to

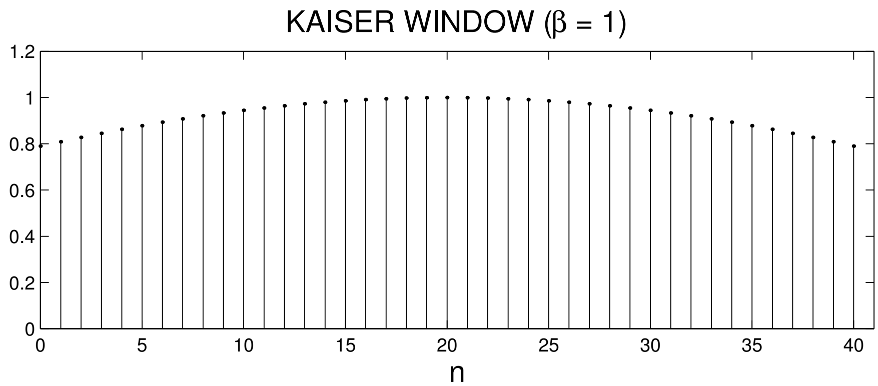
$$N = \frac{A_s - 7.95}{14.36\Delta F} + 1. \tag{8}$$

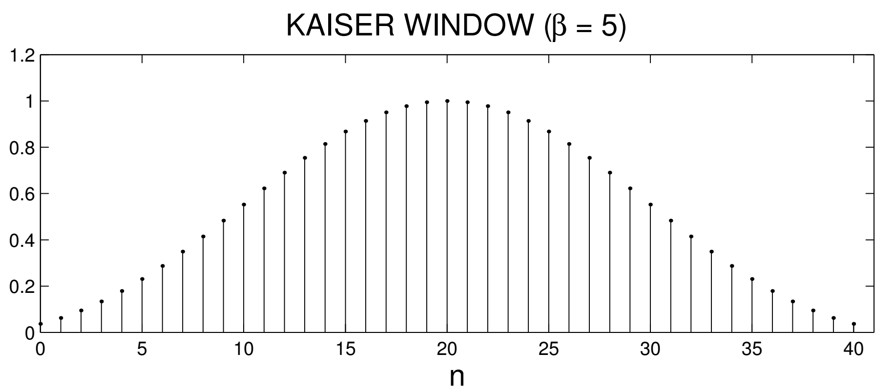
where $\Delta F = (\omega_s - \omega_p)/(2\pi)$. The cut-off frequency ω_c in (1) is $\omega_c = (\omega_p + \omega_s)/2$;
The power series of $I_o(x)$ is

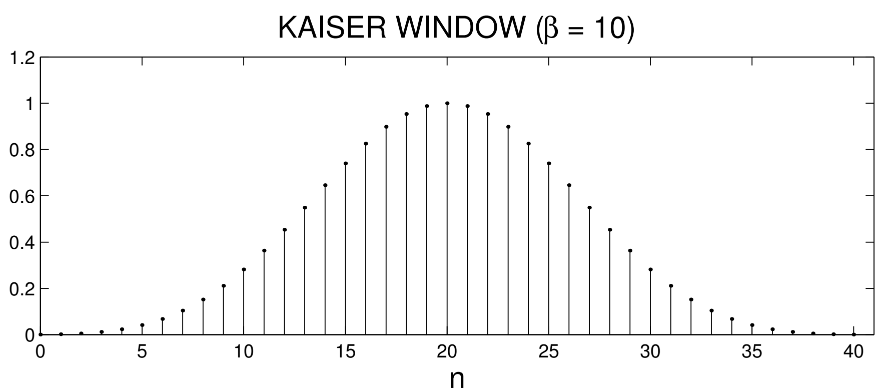
$$I_o(x) = 1 + \sum_{k=1}^{\infty}\left(\frac{(x/2)^k}{k!}\right)^2 \tag{9}$$

of which 20 terms are usually used.

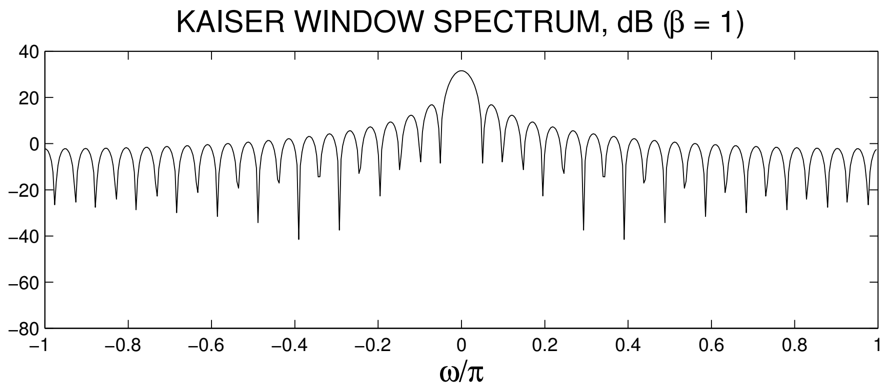
KAISER WINDOW

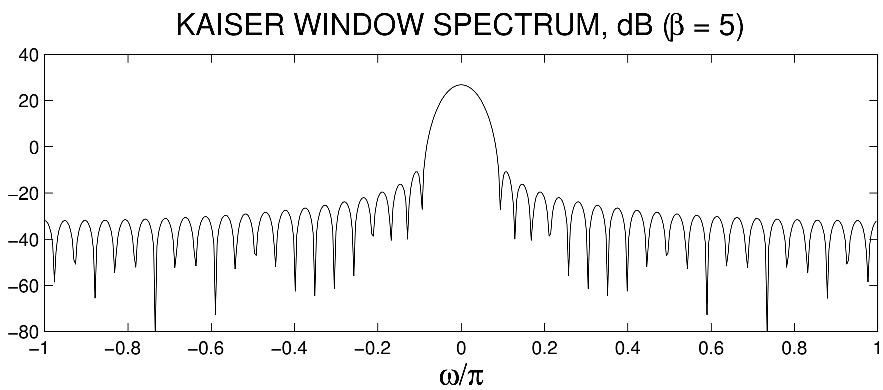


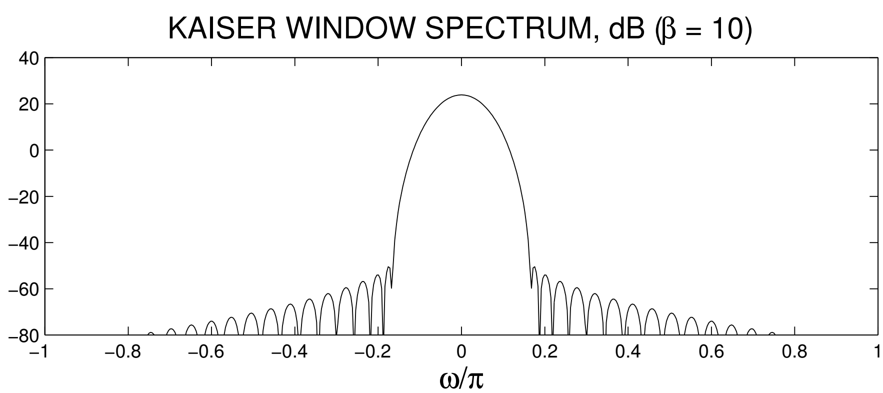




KAISER WINDOW







OTHER WINDOWS

A second approach to window design minimizes the relative peak sidelobe height. The solution is the Dolph-Chebyshev window [?, ?], all the sidelobes of which have equal height. Saramäki has described a family of transitional windows that combine the optimality properties of the DPS window and the Dolph-Chebyshev window. He has found that the transitional window yields better results than both the DPS window and the Dolph-Chebyshev window, in terms of attenuation vs. transition width [?].

An extensive list and analysis of windows is given in [?]. In addition, the use of nonsymmetric windows for the design of fractional delay filters has been discussed in [?, ?].

LPF DESIGN USING WINDOWS: EXAMPLE

Problem: Using the Hamming window, design a linear-phase FIR filter satisfying the following specifications.

1. The passband edge f_p is at 5 Hz
2. The stopband edge f_s is at 10 Hz
3. The maximum passband attenuation is $A_p = 0.05$ dB
4. The minimum stopband attenuation is $A_s = 50$ dB

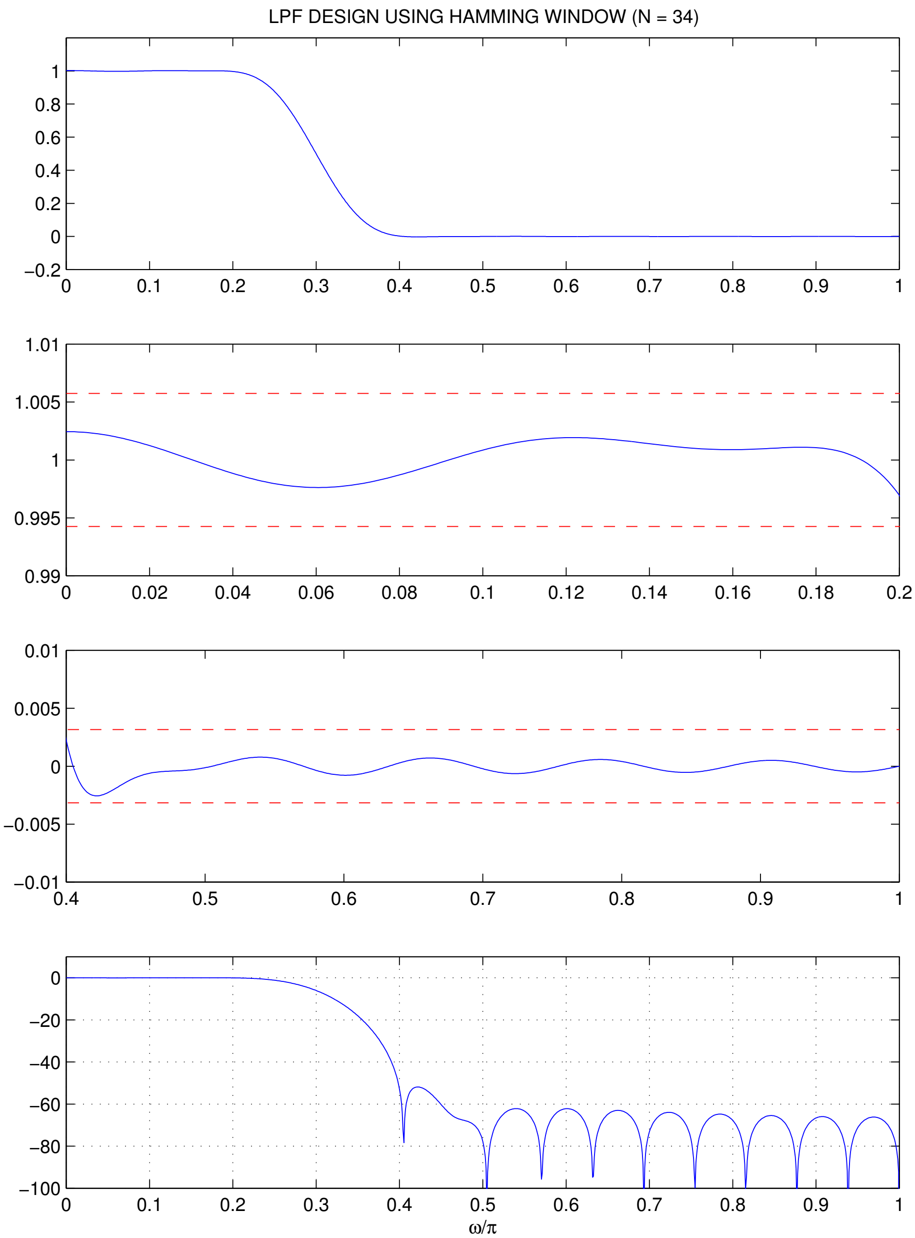
The digital filter will operate at a sampling frequency of 50 Hz.

What is the minimal length of $h(n)$ so that the specifications are satisfied?

We tried several different lengths, and found that the shortest impulse response is of length $N = 34$.

The frequency response is displayed on the following page and the Matlab code is also given.

LPF DESIGN USING WINDOWS: EXAMPLE



The 2nd and 3rd panels show the passband and stopband details.

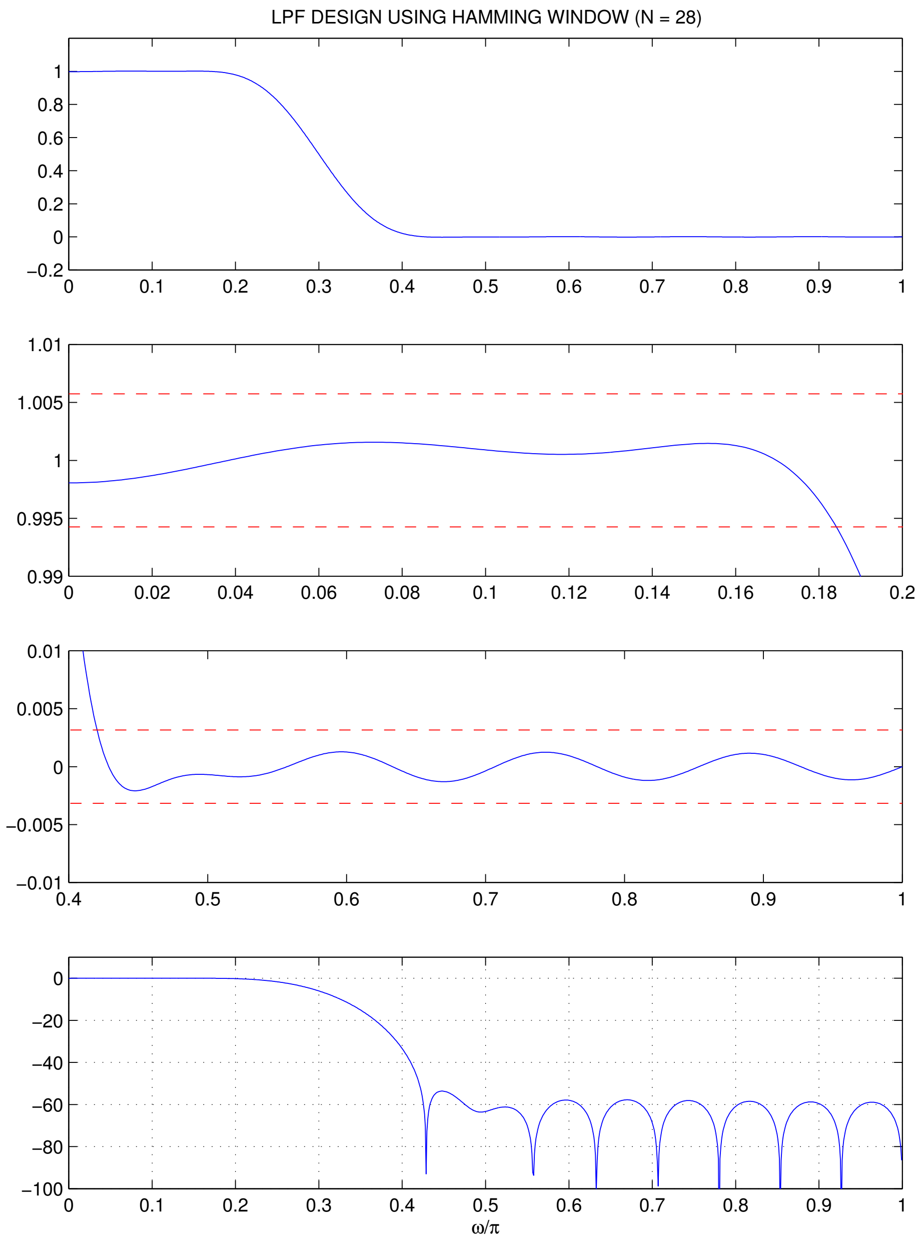
LPF DESIGN USING WINDOWS: EXAMPLE

```
% WINDOW-BASED FIR FILTER DESIGN: EXAMPLE

Fs = 50;          % sampling frequency (Hertz)
fp = 5;           % passband edge (Hertz)
fs = 10;          % stopband edge (Hertz)
wp = 2*pi*fp/Fs;  % passband edge (Rad/Sec)
ws = 2*pi*fs/Fs;  % passband edge (Rad/Sec)
wc = (wp+ws)/2;   % cut-off frequency (Rad/Sec)

Ap = 0.05;             % passband attenuation
As = 50;               % stopband attenuation
dp = 1 - 10^(-Ap/20);  % passband deviation
ds = 10^(-As/20);      % stopband deviation

N = 34;                          % length of h(n)
M = (N-1)/2;                     % midpoint
n = 0:N-1;                       % time index
d = wc/pi * sinc(wc/pi*(n-M));   % sinc function
h = d .* hamming(N)';            % impulse response
[A,w] = firamp(h,1);             % amplitude response

% DISPLAY FREQUENCY RESPONSE
figure(1), clf
subplot(4,1,1)
plot(w/pi,A)
title('LPF DESIGN USING HAMMING WINDOW')
axis([0 1 -0.2 1.2])
subplot(4,1,2)
plot(w/pi,A,[0 1],[1+dp 1+dp],'r--',[0 1],[1-dp 1-dp],'r--')
axis([0 wp/pi 0.99 1.01])
subplot(4,1,3)
plot(w/pi,A,[0 1],[ds ds],'r--',[0 1],[-ds -ds],'r--')
```

```
axis([ws/pi 1 -0.01 0.01])
subplot(4,1,4)
plot(w/pi,20*log10(abs(A)))
axis([0 1 -100 10])
grid
xlabel('\omega/\pi')
```

LPF DESIGN USING WINDOWS: EXAMPLE

If we use a shorter impulse response (length $N = 28$) again with the Hamming window we can see from the following figure that the specification is not satisfied.



FIR DESIGN BY WINDOWS: PROS AND CONS

Pros:

1. The technique is conceptually and computationally simple.

Cons:

1. Using the window method, it is not possible to weight the passband and stopband differently. The ripple sizes in each band will be approximately the same. But requirements are often more strict in the stopband.

2. It is difficult to specify the band edges and maximum ripple size precisely.

3. Not suitable for arbitrary desired responses.

4. The use of windows for filter design is generally considered suboptimal, because they do not solve a clear optimization problem.

FREQUENCY MEASUREMENT (AND WINDOWS)

Sec 11.2 in Mitra

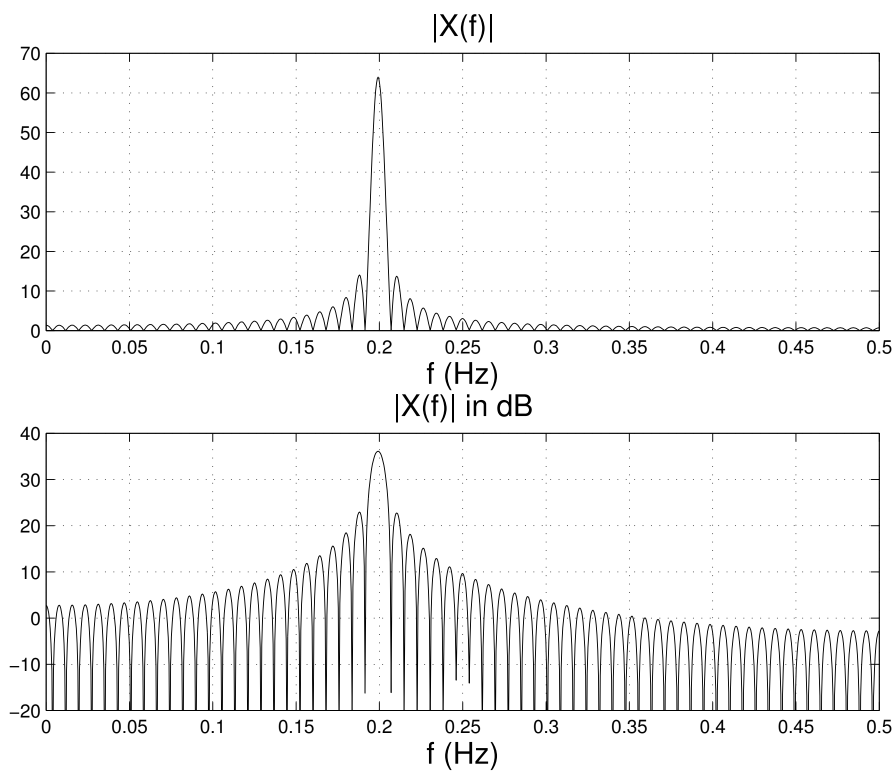
An important application of the DFT and FFT is the measurement of frequencies of periodic signals. Sinusoidal signals are prevalent in science and engineering and the need to measure the frequency of a sinusoidal signals arises in numerous applications.

In practice, signals are measured over a finite duration, and this introduces artifacts. For example, consider the $N = 128$ point signal,

$$x(n) = \sin(2\pi\, 0.1992\, n) + 0.005\, \sin(2\pi\, 0.25\, n) \quad 0 \le n \le 127,$$

$$f_1 = 0.1992, \qquad f_2 = 0.25.$$

The sampling period is $T_s = 1$ and so $F_s = 1$. In this case, the physical frequency is called *normalized frequency*. We wish to find the frequencies f_1, f_2 from the 128 point signal $x(n)$. Let us examine the DTFT of $x(n)$ obtained by zero padding and using the FFT.



The figures on the previous slide were generated with the following Matlab program

```
%  FREQUENCY MEASUREMENT
%  NO WINDOW (RECTANGULAR WINDOW)
f1 = 0.1992;
f2 = 0.25;
N = 128;
n = 0:N-1;
x = sin(2*pi*f1*n) + 0.005*sin(2*pi*f2*n);
L = 2^11;
X = fft(x,L);  % zero padding to length L
k = 0:L-1;
figure(1), clf
subplot(2,1,1)
plot(0:1/L:1/2,abs(X(1:L/2+1)))
grid
xlabel('f (Hz)')
title('|X(f)|')
subplot(2,1,2)
plot(0:1/L:1/2,20*log10(abs(X(1:L/2+1))))
grid
xlabel('f (Hz)')
title('|X(f)|, in dB')
```

Note that one of the sinusoidal components on page 30 is far weaker than the other. It is totally invisible in the plot of the spectrum. From the spectrum, one would conclude that the signal $x(n)$ contains only one sinusoidal component.

> The weaker sinusoid is invisible in the spectrum on page 30 because the side lobes of the rectangular window dominate it.

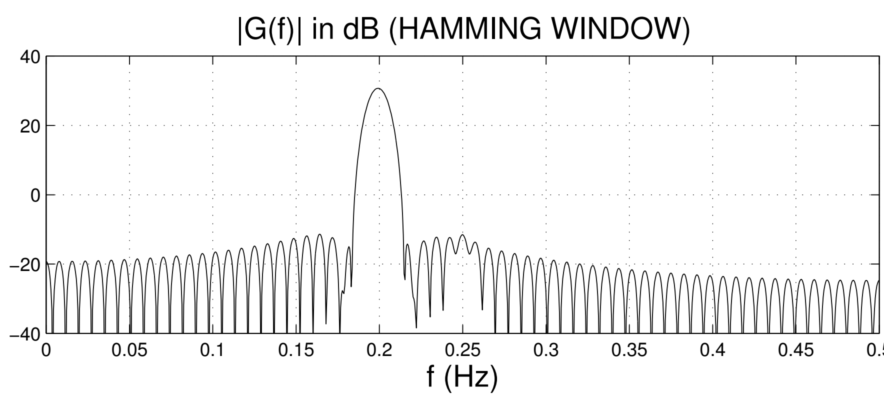
Solution: Window the 128 point data $x(n)$ using a window function with very low side lobes.

$$g(n) = x(n) \cdot w(n).$$

In Matlab, we just add the line

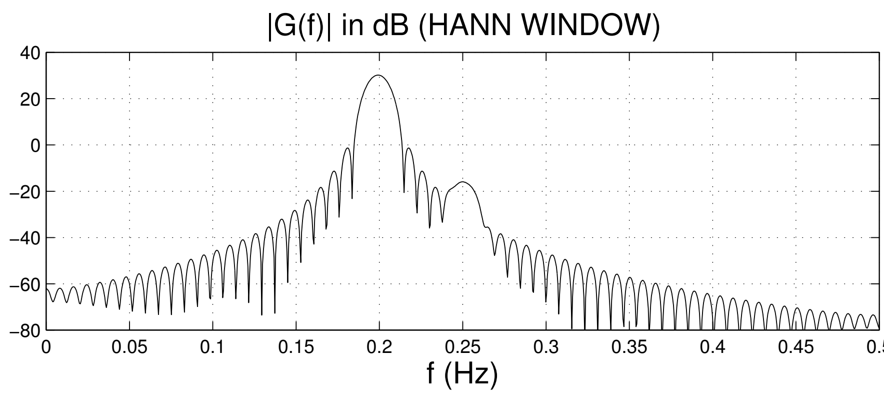
```
g = x .* hamming(N)';
```

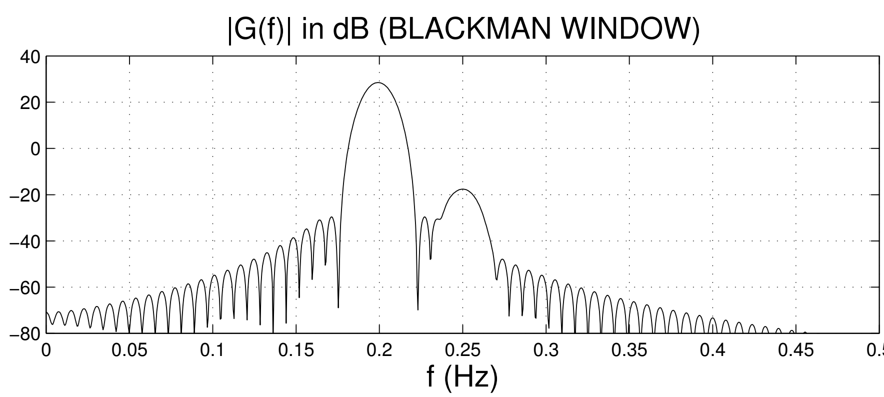
into the program. When we use the Hamming window function, we do not get much improvement.

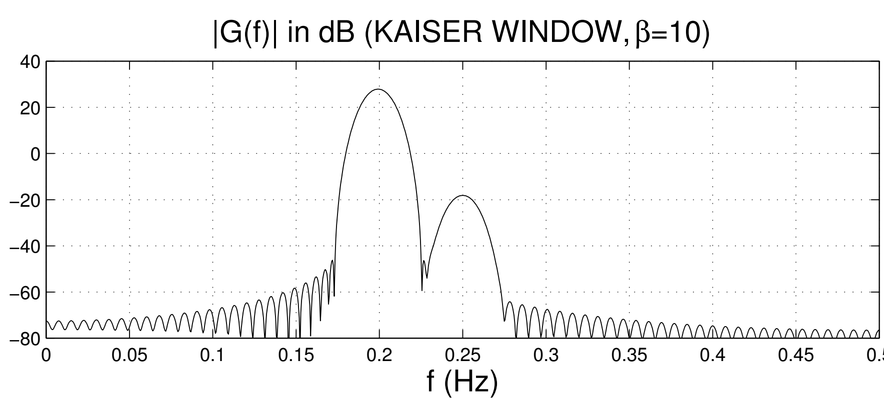


Let us try some other window functions . . .

Compare to the spectrum on page 30. With a suitably chosen window function, we can now detect the weaker sinusoid.







Let us take another example, where the two sinusoidal are much closer together but of not so disparate strengths.

$$x(n) = \sin(2\pi\, f_1\, n) + 0.5\, \sin(2\pi\, f_2\, n) \quad 0 \le n \le 127,$$

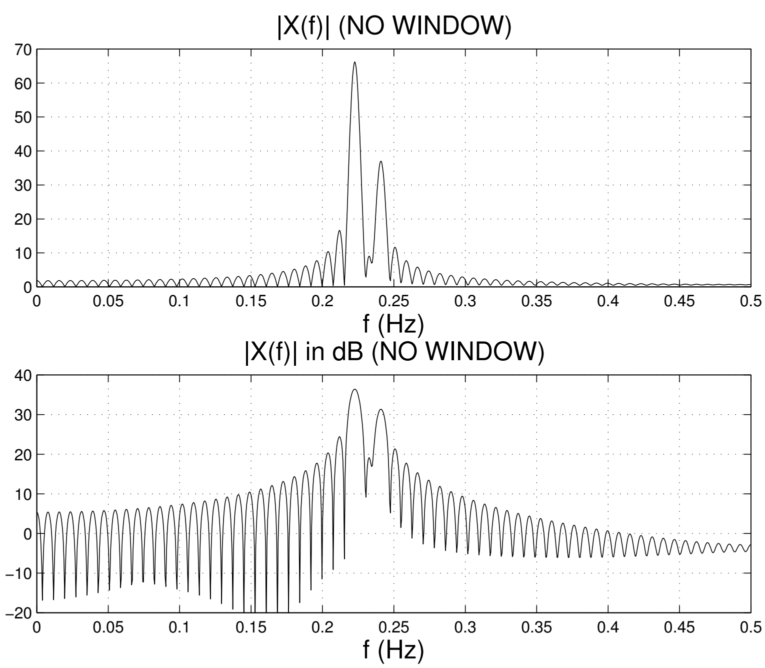
with

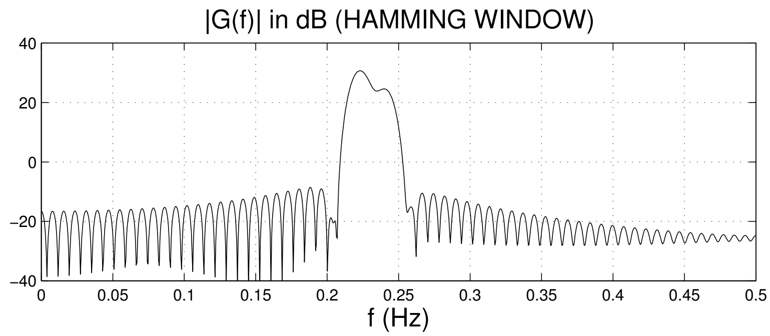
$$f_1 = 0.223, \qquad f_2 = 0.240$$

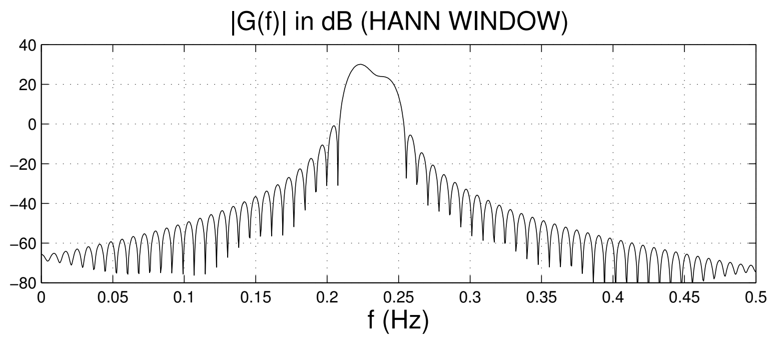
Again, we wish to find the frequencies f_1, f_2 from the 128 point signal $x(n)$.

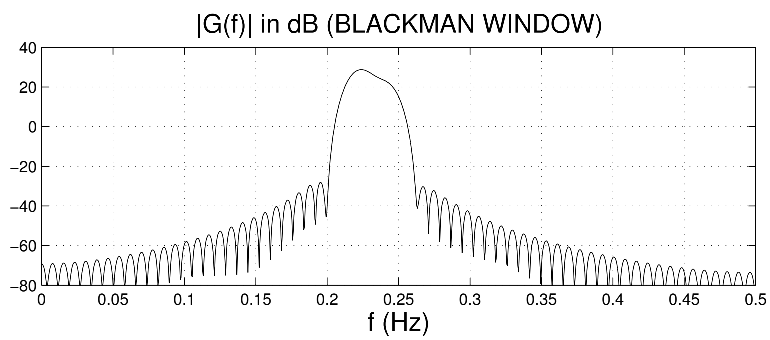
In this case the window that is most appropriate will have a different trade-off between main lobe width and relative side lobe height. A narrow main lobe will be important so as not to smear, or blur, the two frequencies together. Furthermore, the relative side lobe height is not so critical because the two sinusoidal components have more similar strengths.

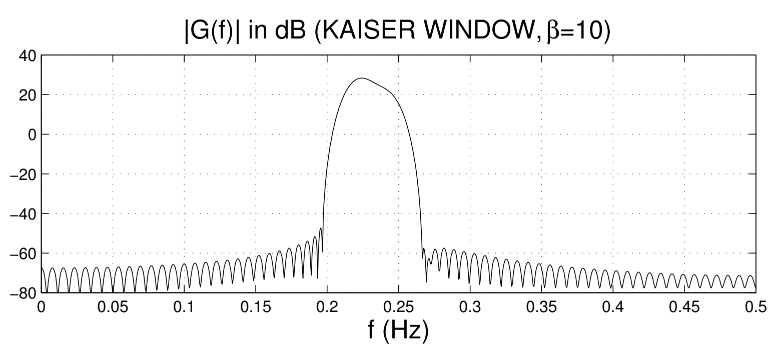
Let us look at the zero padded FFT of the 128 point signal, with and with out windowing, with various windows.




|G(f)| in dB (HAMMING WINDOW)
40
20
0
−20
−40
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
f (Hz)


|G(f)| in dB (HANN WINDOW)
40
20
0
−20
−40
−60
−80
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
f (Hz)


|G(f)| in dB (BLACKMAN WINDOW)
40
20
0
−20
−40
−60
−80
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
f (Hz)


|G(f)| in dB (KAISER WINDOW, β=10)
40
20
0
−20
−40
−60
−80
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0.4
0.45
0.5
f (Hz)

It can be seen that the un-windowed data is the best window in this example, for isolating the two frequencies. The Hamming window also works. However, when the Hann, Blackman, or Kaiser window (with $\beta = 10$) is used, the two components blur together (that is *leakage*) and it is not possible to identify the two different frequencies.

To reduce spectral leakage, a window with low side-lobes is required. (First example.)

To reduce frequency smearing, a window with a narrow main-lobe is required. (Second example.)

1. Note when the signal strengths are very different or when the two frequencies are very close together, if *lots of data is available* then it does not matter what window is used. That is because when the window length N is very long, all the windows have low side-lobes and narrow main-lobes.

2. We did not consider the noisy case here. When the signal $x(n)$ is noisy, then additional steps must be taken to perform frequency estimation.

SHORT TIME FOURIER TRANSFORM (STFT)

Sec 11.3 in Mitra

The Fourier transforms (FT, DTFT, DFT, etc) do not clearly indicate how the frequency content of a signal changes over time.

That information is hidden in the phase — it is not revealed by the plot of the magnitude of the spectrum.

> To see how the frequency content of a signal changes over time, we can cut the signal into blocks and compute the spectrum of each block.

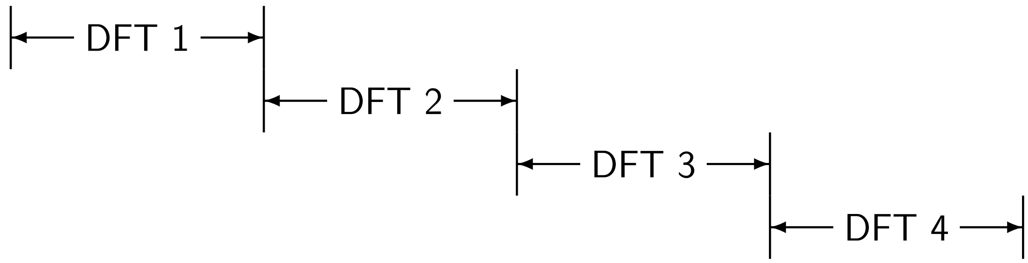
To improve the result,

1. blocks are overlapping,
2. each block is multiplied by a window that is tapered at its endpoints.

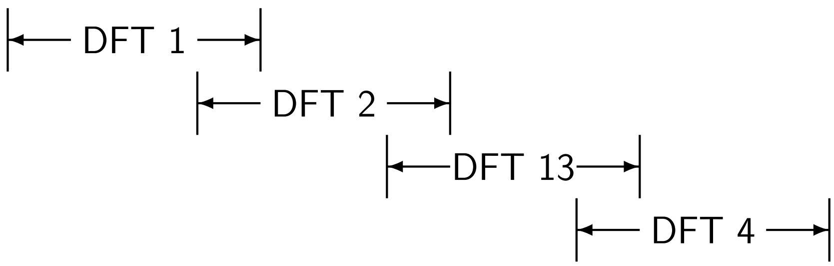
Several parameters must be chosen:

1. Block length, R.
2. The type of window.
3. Amount of overlap between blocks.
4. Amount of zero padding, if any.

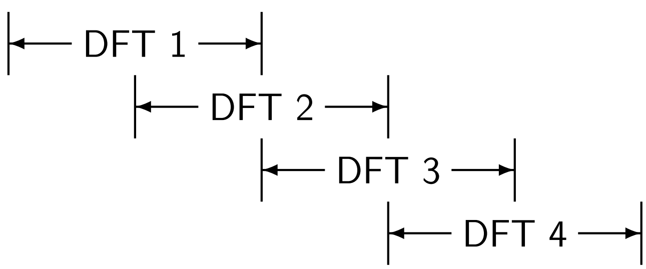
NO OVERLAP



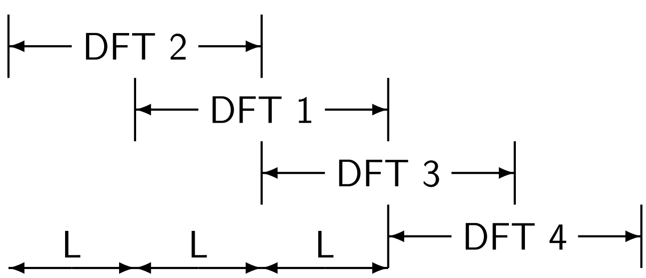
R/4 OVERLAP



R/2 OVERLAP



The parameter L



L is the number of samples between adjacent blocks.

SHORT TIME FOURIER TRANSFORM (STFT)

The short-time Fourier transform is defined as

$$\begin{aligned}
X(\omega, m) &= \text{STFT}\,\{x(n)\} \\
&:= \text{DTFT}\,\{x(n+m)\,w(n)\} \\
&= \sum_{n=-\infty}^{\infty} x(n+m)\,w(n)\,e^{-j\omega n} \\
&= \sum_{n=0}^{R-1} x(n+m)\,w(n)\,e^{-j\omega n}
\end{aligned}$$

where $w(n)$ is the window function of length R.

1. The STFT of a signal $x(n)$ is a function of two variables: time and frequency.
2. The block length is determined by the support of the window function $w(n)$.
3. A graphical display of the magnitude of the STFT, $|X(\omega, m)|$, is called the *spectrogram* of the signal. It is often used in speech processing.
4. The STFT of a signal is invertible.
5. One can choose the block length. A long block length will provide higher frequency resolution (because the main-lobe of the window function will be narrow). A short block length will provide higher time resolution because less averaging across samples is performed for each STFT value.
6. A *narrow-band* spectrogram is one computed using a relatively long block length R, (long window function).
7. A *wide-band* spectrogram is one computed using a relatively short block length R, (short window function).

To numerically evaluate the STFT, we sample the frequency axis ω in N equally spaced samples from $\omega = 0$ to $\omega = 2\pi$.

$$\omega_k = \frac{2\pi}{N} k, \qquad 0 \le k \le N-1$$

We then have the discrete STFT,

$$\begin{aligned} X^d(k,m) &:= X(\frac{2\pi}{N}k, m) \\ &= \sum_{n=0}^{R-1} x(n+m)\, w(n)\, e^{-j\frac{2\pi}{N}kn} \\ &= \sum_{n=0}^{R-1} x(n+m)\, w(n)\, W_N^{-kn} \\ &= \mathsf{DFT}_N\{\{x(n+m)\, w(n)\}_{n=0}^{R-1}, \underbrace{0, \dots, 0}_{N-R}\} \end{aligned}$$

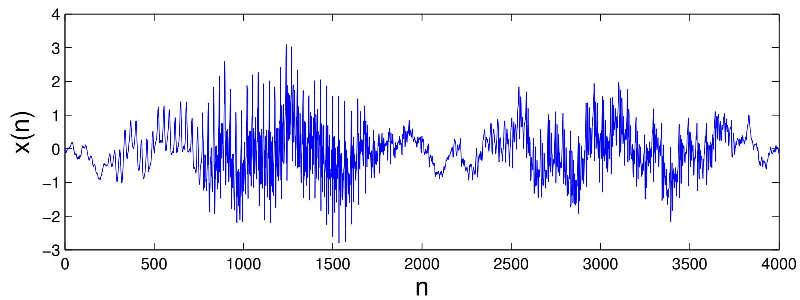
In this definition, the overlap between adjacent blocks is $R-1$. The signal is shifted along the window one sample at a time. That generates more points than is usually needed, so we also sample the STFT along the time direction. That means we usually evaluate

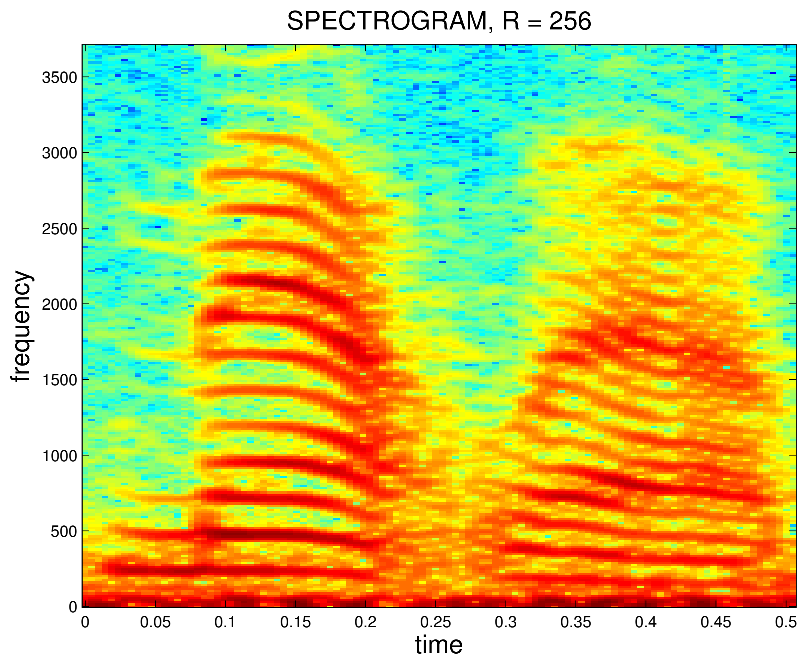
$$X^d(k, Lm)$$

where L is the time-skip. The relation between the time-skip, the number of overlapping samples, and the block length is

$$\text{overlap} = R - L.$$

SPECTROGRAM EXAMPLE





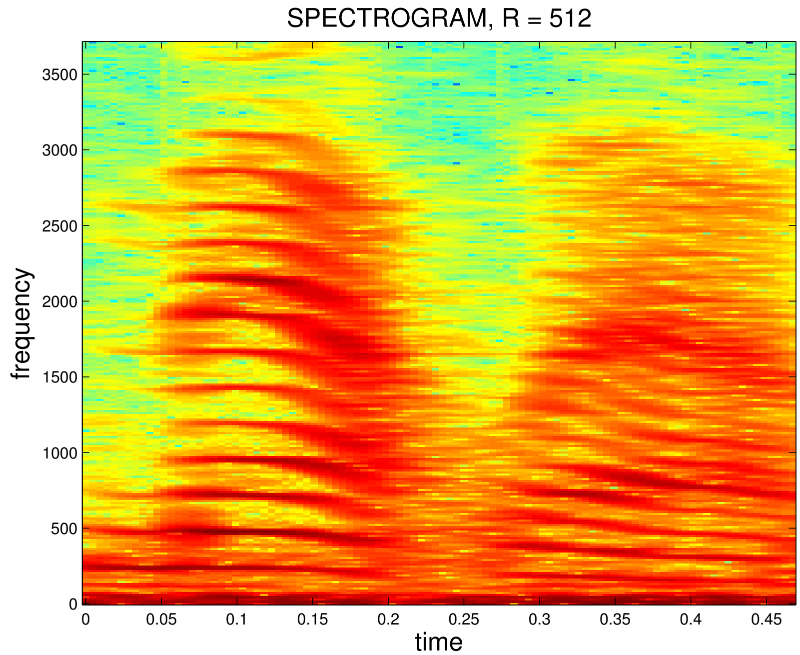
SPECTROGRAM EXAMPLE

The Matlab program for producing the figures on the previous page.

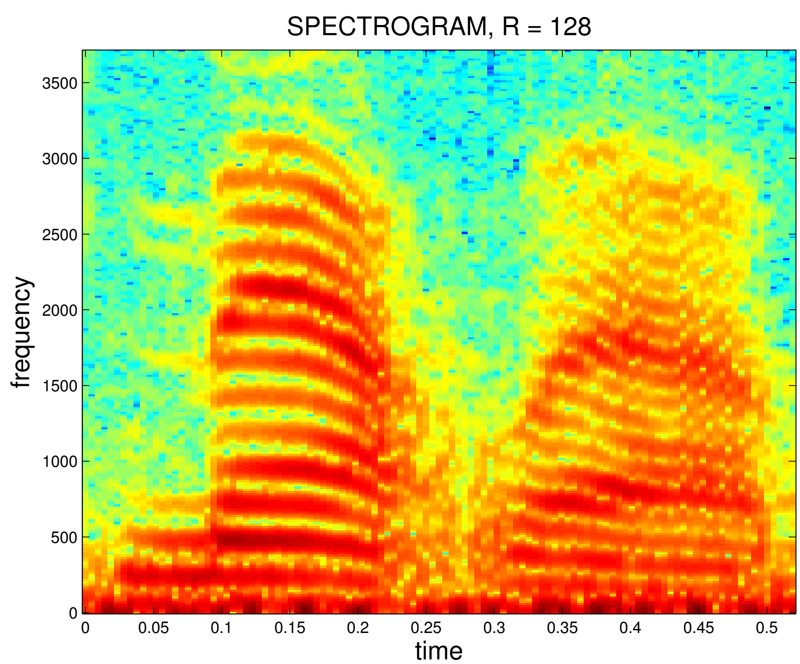
```
% ------ SPECTROGRAM EXAMPLE ------

% LOAD DATA
load mtlb;
x = mtlb;

figure(1), clf
plot(0:4000,x)
xlabel('n')
ylabel('x(n)')

% SET PARAMETERS
R = 256;                % R: block length
window = hamming(R);    % window function of length R
N = 512;                % N: frequency discretization
L = 35;                 % L: time lapse between blocks
fs = 7418;              % fs: sampling frequency
overlap = R - L;

% COMPUTE SPECTROGRAM
[B,f,t] = specgram(x,N,fs,window,overlap);

% MAKE PLOT
figure(2), clf
imagesc(t,f,log10(abs(B)));
colormap('jet')
axis xy
xlabel('time')
ylabel('frequency')
title('SPECTROGRAM, R = 256')
```

EFFECT OF WINDOW LENGTH R

Narrow-band spectrogram: better frequency resolution

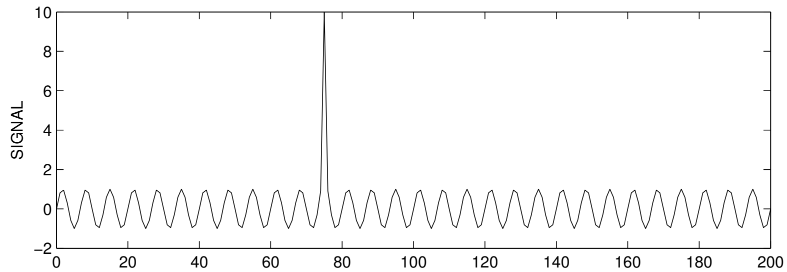


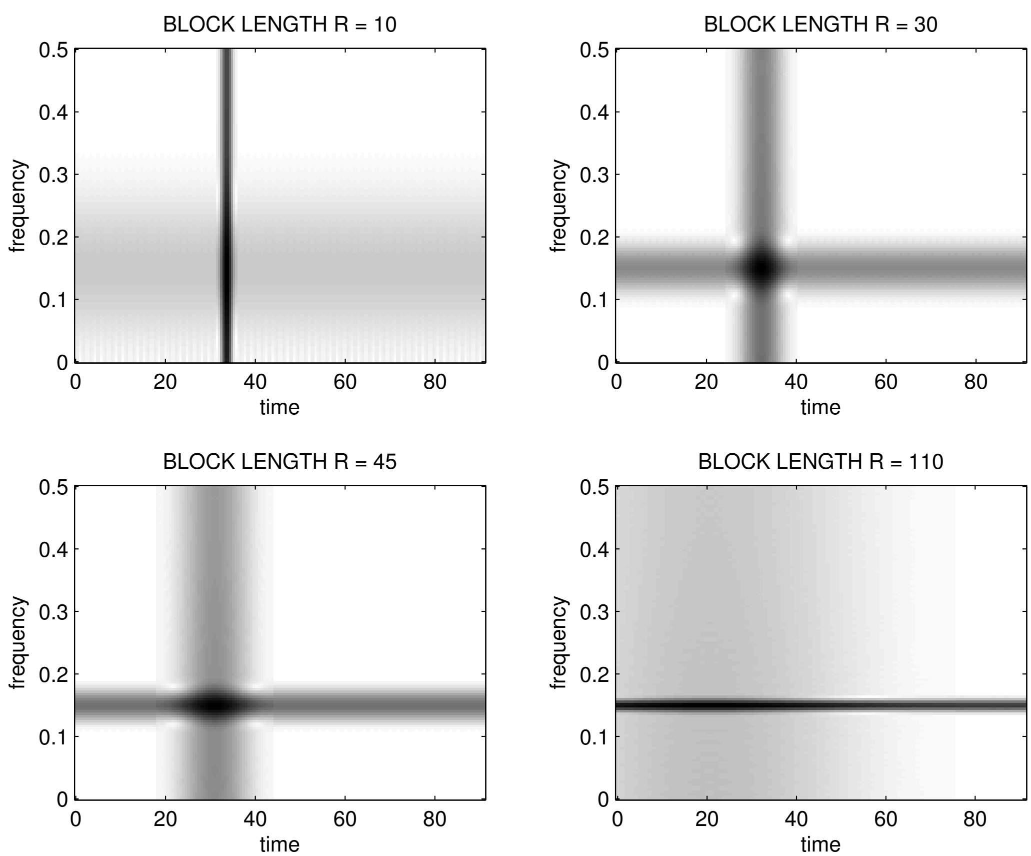
Wide-band spectrogram:: better time resolution



EFFECT OF WINDOW LENGTH R

Here is another example to illustrate the frequency/time resolution trade-off.



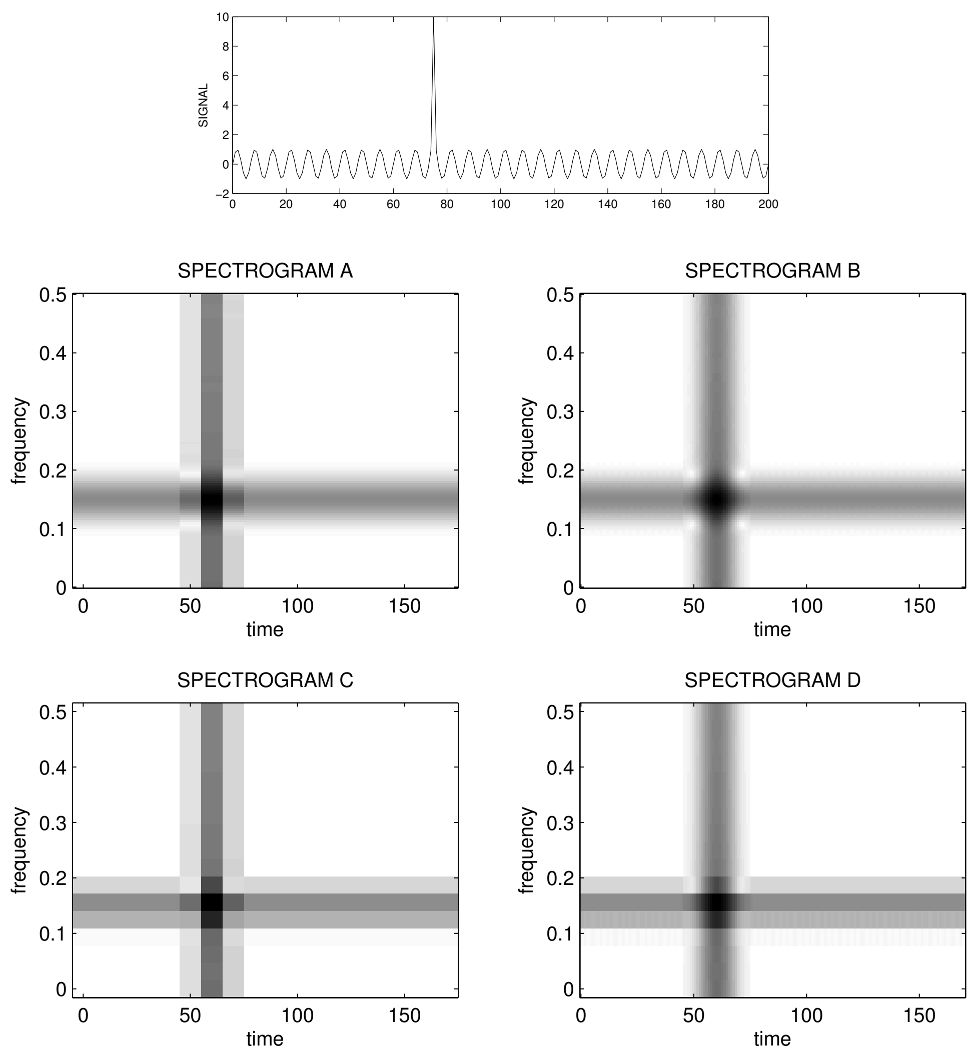


A spectrogram is computed with different parameters:

$$L \in \{1, 10\}, \qquad N \in \{32, 256\}$$

$L =$ time lapse between blocks.
$N =$ FFT length (Each block is zero-padded to length N.)
In each case, the block length is 30 samples.

For each of the four spectrograms, can you tell what L and N are?



L and N do not effect the time resolution or the frequency resolution. They only influence the 'pixelation'.