Digital Signal Processing (Lecture Note 6)

Jin Ho Chang

Professor

Dept. of Electrical Engineering and Computer Science

대구경북과학기술원
Daegu Gyeongbuk Institute of Science & Technology

Data Windowing

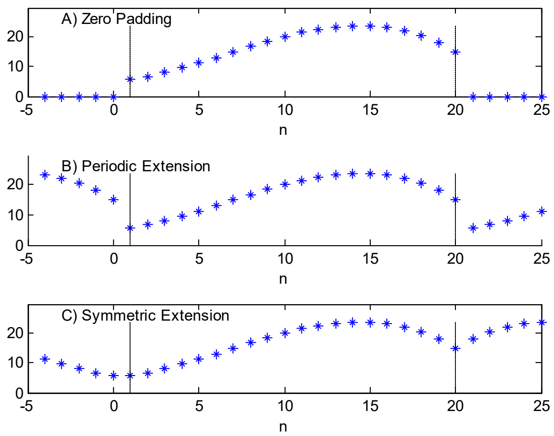
대구경북과학기술원
Daegu Gyeongbuk Institute of Science & Technology

Edge Effects

- An advantage of dealing with infinite data is that one need not to be concerned with the end points since there are no end points.
- However, finite data consist of numerical sequences having a fixed length with fixed end points at the beginning and end of the sequence.
  - Some operations, such as convolution, may produce additional data points while some operations will require additional data points to complete their operation on the data set.
  - The question becomes how to add or eliminate data points, and there are a number of popular strategies for dealing with these edge effects.

Three Common Strategies

- Using for extending a data set when additional points are needed
  - Extending with zeros (or Constant): Zero Padding
  - Extending using periodicity or wraparound
  - Extending by reflection known as symmetric extension



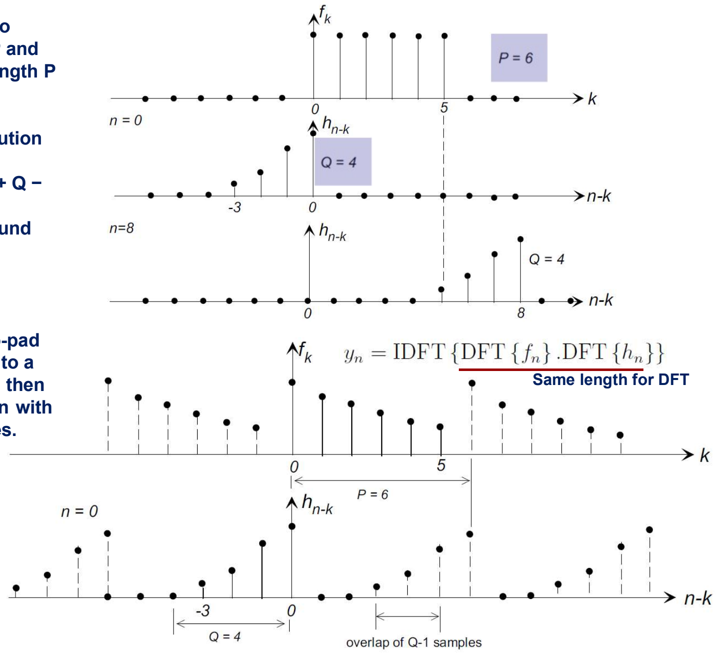
- Zero-Padding Approach
  - It is Frequently used in spectral analysis.
  - This approach is justified by the implicit assumption that the waveform is zero outside of the sample period anyway.
  - A variant of zero padding is constant padding, where the data sequence is extended using a constant value, often the last (or first) value in the sequence.
- Wraparound Approach
  - If the waveform can be reasonably thought of as one cycle of a periodic function, then this approach is justified.
  - The data are extended by tacking on the initial data sequence to the end of the data set and vice versa.

  ❖**These two approaches will, in general, produce a discontinuity at the beginning or end of the data set, which can lead to artifact.**

- Symmetric Reflection Approach
  - It eliminates the discontinuity by tacking on the end points in reverse order (or beginning points if extending the beginning of the data sequence).
- An Example: Convolution with small size of data sets
  - When the data are considered periodic and it is desired to retain the same period or when other similar concerns are involved.
  - The original data set is extended using the wraparound strategy, convolution is performed on the extended set.
  - The additional points produced by the convolution operation are removed symmetrically.
    - ✓ Symmetrically eliminate data from both ends of the data set.
  - The goal is to preserve the relative phase between waveforms pre- and post-convolution.

DFT convolution of two sequences of length P and Q (P ≥ Q) in DFTs of length P
1.Produces an output sequence of length P, whereas linear convolution produces an output sequence of length P + Q − 1.
2.Introduces wrap-around error in the first Q−1 samples of the output sequence.

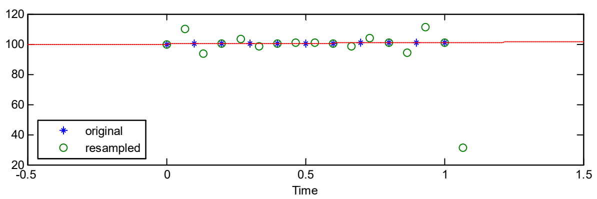
The solution is to zero-pad both input sequences to a length N ≥ P +Q−1 and then to use DFT convolution with the length N sequences.



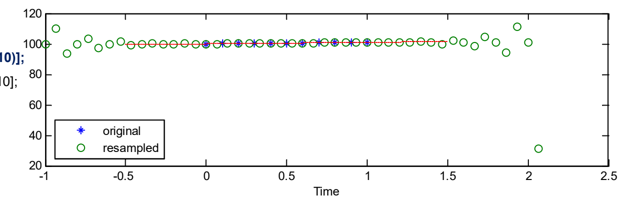
The RESAMPLE function assumes that the signal values before and after the specified signal are zeros. However, this may not be true for your particular biased signal as in the example below. This assumption is what causes the edge effects (or oscillations) to occur.

To work around this issue, you should pad your input signal at the beginning and end with the appropriate values, and then use RESAMPLE. The relevant signal can then be extracted from the results of resampling.

```
fs1 = 10; % Original sampling frequency in Hz
t1 = 0:1/fs1:1; % Time vector
x = t1 + 100; % Define a linear sequence
y = resample(x,3,2); % Now resample it
t2 = (0:(length(y)-1))*2/(3*fs1); % New time vector
plot(t1,x,'*',t2,y,'o',(-0.5:0.01:1.5),(-
0.5:0.01:1.5)+100,':')
```




```
fs1 = 10; % Original sampling frequency in Hz
t1 = 0:1/fs1:1; % Time vector
x = t1 + 100; % Define a linear sequence
xpad = [repmat(x(1), 1, 10), x, repmat(x(end), 1, 10)];
tpad = [-1/fs*10 : 1/fs: 0-1/fs, t1, 1+1/fs:1/fs:1+1/fs*10];
ypad = resample(xpad,3,2); % Now resample it
t2 = (0:(length(ypad)-1))*2/(3*fs1) - 1; % New time
vector
plot(t1,x,'*',t2,ypad,'o',(-0.5:0.01:1.5),(-
0.5:0.01:1.5)+100,':')
```



Data Windowing

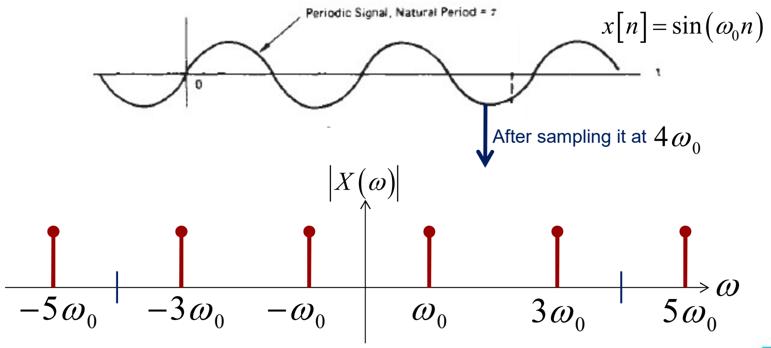
- More often, a waveform is neither periodic or aperiodic, but a segment of a much longer – possibly infinite – time series.
  - EEG and ECG analysis where the waveforms being analyzed continue over the lifetime of the subject are examples
  - Obviously, only a portion of such waveforms can be represented in the finite memory of the computer.
  - Some attention must be paid to how the waveform is truncated.
  - A segment is simply cut out from the overall waveform; that is, a portion of the waveform is truncated and stored, without modification, in a computer. ➔ The application of a **rectangular window** to the overall waveform.

- When a data set is windowed, then the frequency characteristics of the window become part of the spectral result.
  - All windows produce manly two types of artifacts.
  - Type 1: The actual spectrum is **widened by an artifact** termed the **mainlobe**.
  - Type 2: **Additional peaks** are generated termed the **sidelobes**.
- Most alternatives to the rectangular window **reduce the sidelobes** at the cost of **wider mainlobes**.

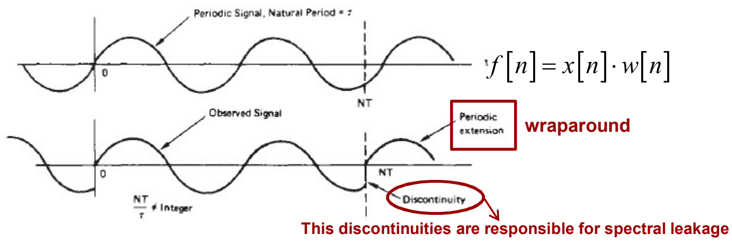
➔ Reading Assignment: F. J. Harris, "One the use of windows for harmonic analysis with the discrete Fourier transform," Proceedings of the IEEE, vol. 66, no. 1, 1978.

Spectral Leakage

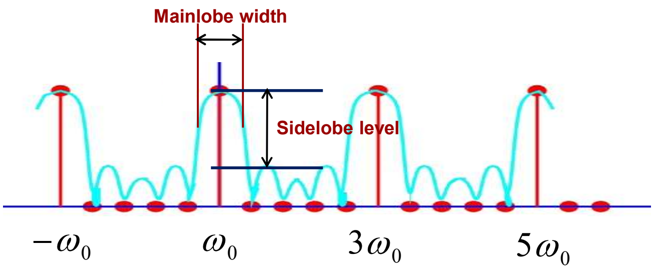
- An effect in the frequency analysis of finite-length signals or finite-length segments of infinite signals where it appears as if some energy has leaked out of the original signal spectrum into other frequencies.➔ Sidelobes



11



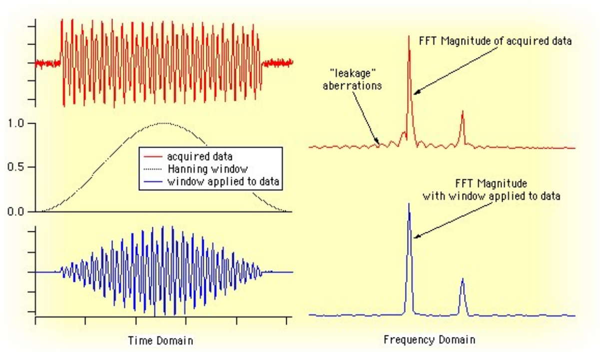
$$F[k]=\sum_{n=0}^{N-1}\sin(\omega_0 n)\cdot w[n]e^{-\frac{2\pi kn}{N}}=X[k]*W[k]$$



12

Example

- Windows are weighting functions applied to data to reduce the spectral leakage associated with finite observation intervals.
  - From one viewpoint, the window is applied to data to reduce the order of the discontinuity at the boundary of the periodic extension, i.e., sudden changes in the data at the start and end of data.



13

Summary of Spectral Leakage

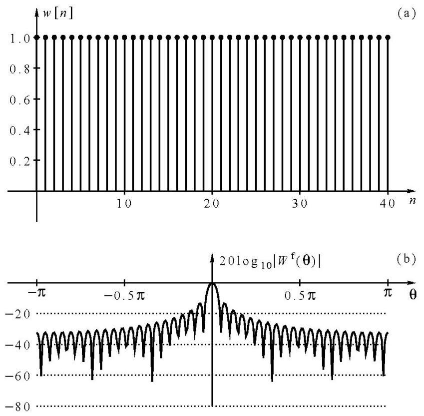
Fourier transform is theoretically defined for signals of infinite duration, and thus a finite segment of data is treated as a periodic signal with period equal to the duration of the data segment. Differences between the starting and ending values of the segment produces a discontinuity which generates high-frequency spurious components. These can be reduced tapering down start and end of the segment by multiplying the signal by a data window w(t). Popular data windows in the analysis of blood pressure and heart rate variabilities are the 10% cosine-taper and the Hann windows. Data windowing, however, **worsens frequency resolution** and **estimation variance of the spectrum**.

❖ **Desired Window Properties: Narrow mainlobe and Low Sidelobes**

14

Window Functions

- Rectangular Window



$$w_r[n] = \begin{cases} 1, & 0 \le n \le N-1 \\ 0, & otherwise \end{cases}$$

$$W[\theta] = \frac{\sin\left(\frac{\theta N}{2}\right)}{\sin\left(\frac{\theta}{2}\right)} e^{-j\frac{\theta(N-1)}{2}}$$

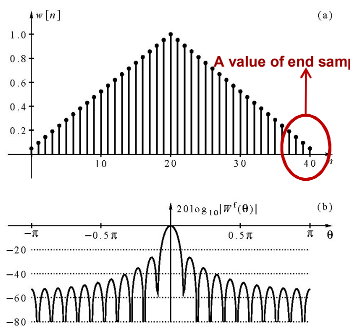
Mainlobe width: $\theta = \frac{2\times 2\pi}{N}$

Max. Sidelobe Level: -13.5 dB

at $\theta = \frac{3\pi}{N}$

DGIST 대구경북과학기술원 Daegu Gyeongbuk Institute of Science & Technology

- Bartlett (or Triangular) Window



A value of end sample is not zero

For odd N

$$\begin{aligned} w_t(n) &= \frac{2}{N+1} w_r(n) * w_r(n) \\ &= 1 - \frac{|2n - N + 1|}{N+1}, \quad 0 \le n \le N-1 \end{aligned}$$

where length of $w_r(n)$ = (N+1)/2

Mainlobe width: $4\frac{2\pi}{N+1} \approx 2\frac{4\pi}{N}$

Peak sidelobe level: Nearly twice that of a RW

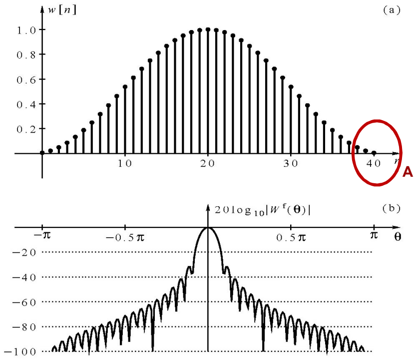
2x(-13.5) = -27 dB

$$\boxed{\begin{aligned} W_t(\Theta) &= \frac{2}{N+1} D^2(\Theta, (N+1)/2)\, e^{-j\Theta(N-1)/2} \\ &= \frac{2\sin^2[\Theta(N+1)/4]}{(N+1)\sin^2(\Theta/2)} e^{-j\Theta(N-1)/2} \end{aligned}}$$

DGIST 대구경북과학기술원 Daegu Gyeongbuk Institute of Science & Technology

- Hann Window

Increasing window length by 2 with respect to the desired N and deleting the two end points



$$w_{hn}(n) = 0.5\left[1-\cos\left(\frac{2\pi(n+1)}{N+1}\right)\right],\quad 0 \le n \le N-1$$

A value of end sample is zero

Same as that of Bartlett window

Mainlobe width: $8\pi/N$

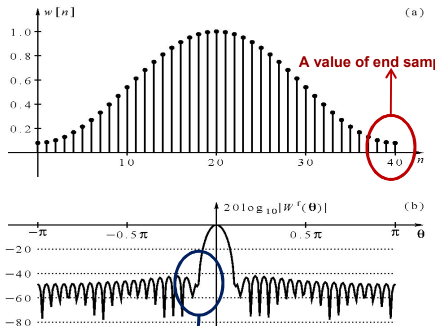
Peak sidelobe level: -32 dB

Lower than that of Bartlett window

17

- Hamming Window

$$w_{hm}(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N-1}\right),\quad 0 \le n \le N-1,$$



A value of end sample is not zero

Same as that of Bartlett window

Mainlobe width: $8\pi/N$

Peak sidelobe level: -43 dB

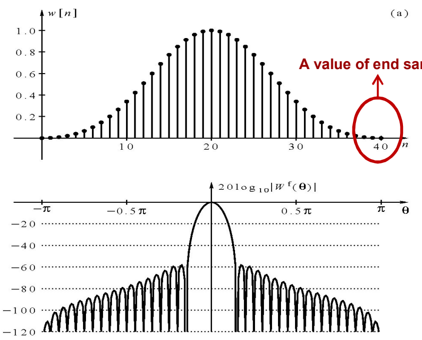
Lower than that of Bartlett and Hann windows

A peculiar property of this window is that the highest sidelobe is not the nearest to the main lobe.

18

Blackman Window

$$w_b(n) = 0.42 - 0.5\cos\left(\frac{2\pi n}{N-1}\right) + 0.08\cos\left(\frac{4\pi n}{N-1}\right),\ \ 0 \le n \le N-1,$$



A value of end sample is zero

Increasing window length by 2 with respect to the desired N and deleting the two end points

broadest mainlobe width

Mainlobe width: $12\pi/N$

Peak sidelobe level: -57 dB

Lower than that of Bartlett and Hann windows

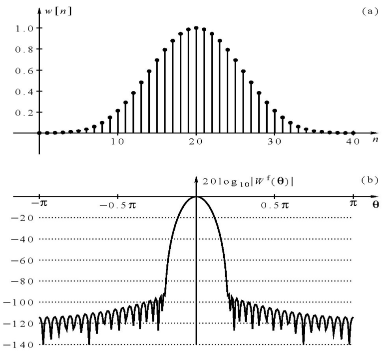
- Optimum Window
  - Modern windows are based on optimally criteria; they aim to be best in a certain respect, while meeting certain constraints.
- Optimality Criteria
  - Dolph's Criterion
    - ✓ Minimize the width of the mainlobe of the kernel, under the constraint that the window length be fixed and the sidelobe level not exceed a given maximum value.
    - ✓ The window values depend on the length and the permitted sidelobe level.
  - Kaiser's Criterion
    - ✓ Minimize the width of the mainlobe of the kernel, under the constraint that the window length be fixed and the energy in the sidelobes not exceed a given percentage of the total energy.
    - ✓ The window values depend on the length and the permitted energy in the sidelobes.

- Kaiser Window

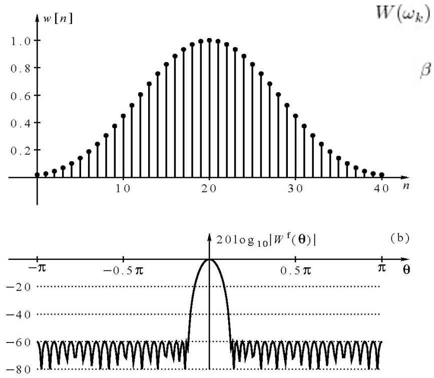
$$w_k(n)=\frac{I_0\left[\alpha\sqrt{1-\left(\frac{|2n-N+1|}{N-1}\right)^2}\right]}{I_0(\alpha)},\quad 0\le n\le N-1$$

$$\alpha\uparrow \rightarrow (mainlobe\ \ width\uparrow)\ (sidelobe\downarrow)$$

α is a control parameter



- Dolph Window
  - Equi-ripple window
  - Sensitive to coefficient accuracy



$$W(\omega_k) = \frac{\cos\left\{M\cos^{-1}\left[\beta\cos\left(\frac{\pi k}{M}\right)\right]\right\}}{\cosh\left[M\cosh^{-1}(\beta)\right]},\qquad k=0,1,2,\ldots,M-1$$

$$\beta = \cosh\left[\frac{1}{M}\cosh^{-1}(10^{\alpha})\right],\qquad (\alpha\approx 2,3,4).$$

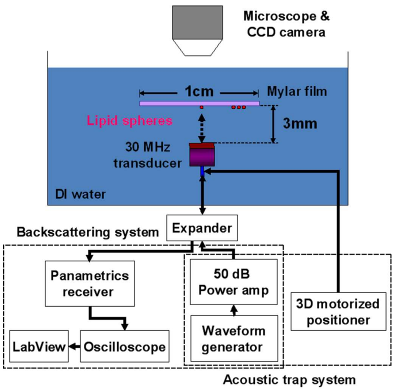
α is the sidelobe level in dB
(i.e., 20α dB)

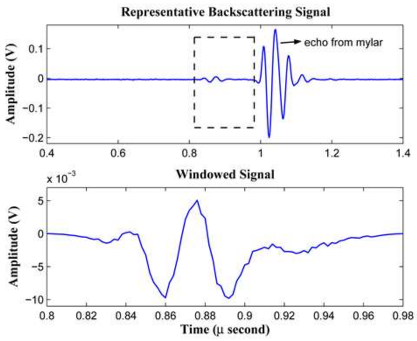
Selection of a Window Function

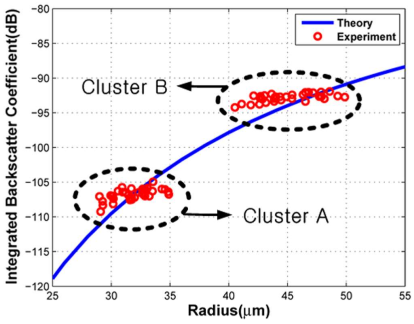
- Resolution vs. Dynamic Range
  - Resolution related to mainlobe width
  - Dynamic Range related to sidelob levels
    - ✓ If there are two sinusoids with different frequencies, leakage can interfere with the ability to distinguish them spectrally.
    - ✓ If their frequencies are not similar, the leakage interferes when on sinusoid is much smaller in amplitude than the other.➔ **Dynamic Range issue**
    - ✓ When the frequencies are near each other, the leakage can be sufficient to interfere even when the sinusoids are equal strength. ➔ **Resolution issue**
  - The rectangular window has excellent resolution characteristics for signals of comparable strength, but it a poor choice for signals of disparate amplitude.

- If the task is to resolve two narrowband signals closely spaced in frequency, then a window with **the narrowest mainlobe is preferred**.
- If there is a strong and a weak signal spaced a moderate distance apart, then a window with **rapidly decaying sidelobes** is preferred to prevent the sidelobes of the strong signal from overpowering the weak signal.
- If there are two moderate strength signals, one close and the other more distant from a weak signal, then a compromise window with a moderately narrow mainlobe and a moderate decay in sidelobes could be the best choice. ➔ **The most appropriate window is selected by trial and error**.

Example



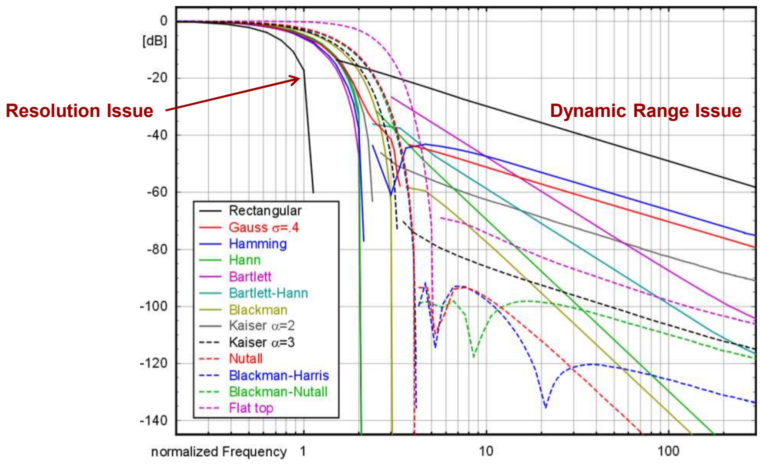




J. Lee, J. H. Chang, et al., "Backscattering measurement from a single microdroplet", IEEE Trans. UFFC, vol. 58, pp. 874 – 879, 2011.

Review of Data Windowing

$$F[k]=\sum_{n=0}^{N-1}\sin(\omega_0 n)\cdot w[n]e^{-\frac{2\pi kn}{N}}=X[k]*W[k]$$



Window Functions in MATLAB

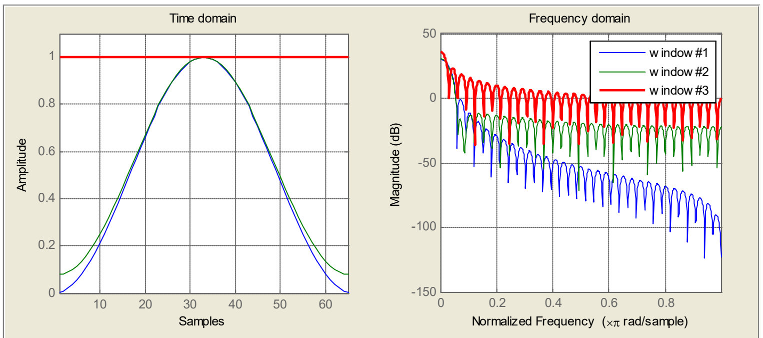
- MATLAB has a number of data windows available for us to easily use.
  - The relevant MATLAB routine generates an n-point vector array containing the appropriate window shape.
  - General Form
    - ✓ W =window_name(N)
      - Generate vector W of length N containing the window function of the associated name.
    - ✓ Example
      - W=hamming(8)
      - W=[0.08 0.2532 0.6424 0.9544 0.9544 0.6424 0.2532 0.08]^T
    - ✓ window_name: rectwin, bartlett, hanning, hamming, blackman, gausswin, kaiser, triang, and chebwin with option arguments

- All of the window functions can be constructed with one call
  - W = window(@name, N, opt)
    - ✓ @barthannwin, @bartlett, @blackman, @blackmanharris, @bohmanwin, @chebwin, @flattopwin, @gausswin,@hamming, @hann, @kaiser, @nuttallwin, @parzenwin, @rectwin, @taylorwin, @triang, @tukeywin

Window	winopt Description	winopt Value
blackman	sampling flag string	'periodic'or 'symmetric'
chebwin	sidelobe attenuation relative to mainlobe	numeric
flattopwin	sampling flag string	'periodic'or 'symmetric'
gausswin	alpha value (reciprocal of standard deviation)	numeric
hamming	sampling flag string	'periodic'or 'symmetric'
hann	sampling flag string	'periodic'or 'symmetric'
kaiser	beta value	numeric
taylorwin	1. number of sidelobes 2. maximum sidelobe level in dB relative to mainlobe peak	1. integer greater than or equal to 1 2. negative value
tukeywin	ratio of taper to constant sections	numeric

Example

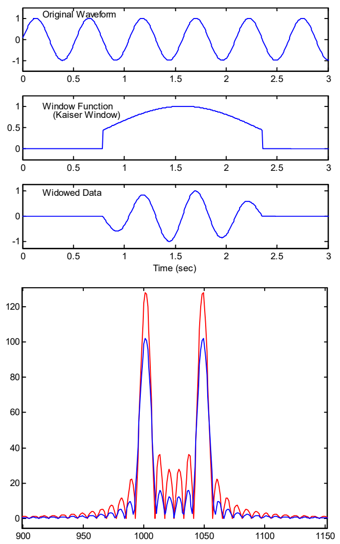
N = 65;
w1 = window(@hanning,N);
w2 = window(@hamming,N);
w3 = window(@rectwin,N);
wvtool(w1,w2,w3)
➔ a GUI that allows you to analyze windows



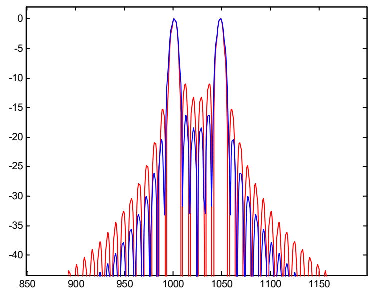
Leakage Factor: ratio of power in the sidelobes to the total window power

W1➔ Leakage Factor: 0.05%, Relative Sidelobe Attn: -31.5dB, Mainlobe Width (-3dB): 0.042969
W2➔ Leakage Factor: 0.04%, Relative Sidelobe Attn: -42.5dB, Mainlobe Width (-3dB): 0.039063
W3➔ Leakage Factor: 9.41%, Relative Sidelobe Attn: -13.3dB, Mainlobe Width (-3dB): 0.025391

- Application of a window function



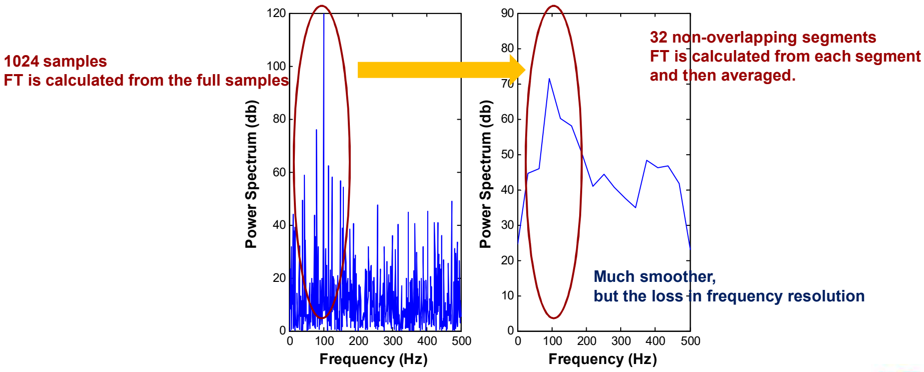

```
N = 512;
t = (1:N)* pi/N;
x = sin (12*t);          % Generate 12 cycles of a sin wave
wind = window(@kaiser,N/2,2);
wind_re=[zeros(128,1); ones(length(wind),1); zeros(128,1)];
wind = [zeros(128,1); wind; zeros(128,1)];
subplot(3,1,1);
plot(t,x,'b');
axis([0 3 -1.5 1.5]);
text(.2,1.2,'Original Waveform');
subplot(3,1,2);
plot(t,wind,'b');
axis([0 3 -.25 1.25]);
text(.2,1,'Window Function');
text(.3,.8,'(Kaiser Window)');
out = x .* wind';
subplot(3,1,3);
plot(t,out,'b');
axis([0 3 -1.25 1.25]);
xlabel('Time (sec)');
text(.2,.9,'Widowed Data');

x_fft=fftshift(fft(x.*wind_re‘, 2048));
xWin_fft=fftshift(fft(out, 2048));
figure,plot(1:2048, abs(x_fft), 'r', 1:2048, abs(xWin_fft), 'b')
```



Example: Power Spectrum Density

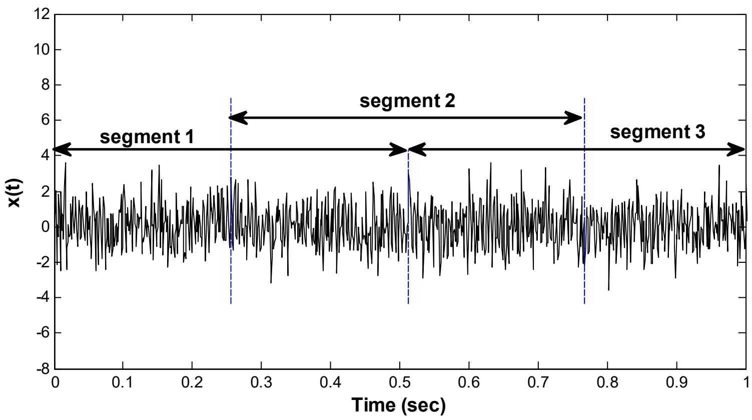
- Computation of PSD commonly involves truncation of data, particularly when the data contains some noise.
  - Average Periodogram
    - ✓ The periodogram is an estimate of the spectral density of a signal, which is computed using the fast Fourier transform.
    - ✓ The raw periodogram is not a good spectral estimate because of <u>spectral bias</u> and the fact that <u>the variance at a given frequency does not decrease as the number of samples used in the computation increases</u>. ➔ Need to use window function.
    - ✓ Averaging improves the statistical properties of the result.
    - ✓ The average periodogram is an estimate by performing Fourier transform and subsequently averaging.
    - ✓ The selection of the data window and the averaging strategy determines statistically reliable spectral estimates and resolution.

- Averaging is usually achieved by dividing the waveform into a number of segments and evaluating the Fourier transform on each of these segments. ➔ Reducing the number of data samples ➔ Lowering frequency resolution (1/NT)
- Choosing **<u>a short segment length</u>** (a small N) will provide more segments for averaging and **improve reliability** of the spectral estimate, but it will also **decrease frequency resolution**.



- Welch Method of spectral analysis
  - A segmentation scheme
    - ✓Overlapping segments
    - ✓A window is applied to each segment
  - By overlapping segment, more segments can be averaged for a given segment and data length.
  - Averaged periodograms obtained from noisy data traditionally average spectra from half-overlapping segments, i.e., segments that overlap by 50%.
    - ✓When computing time is not a factor, maximum overlap has been recommended.
      ➔ Shifting over by just a single sample to get the new segment.
  - The use of data windowing for sidelobe control is not as important when the spectra are expected to be relatively flat.
    - ✓Windows should be applied whenever the spectra are expected to have large amplitude differences.

- Overlapping Segmentation

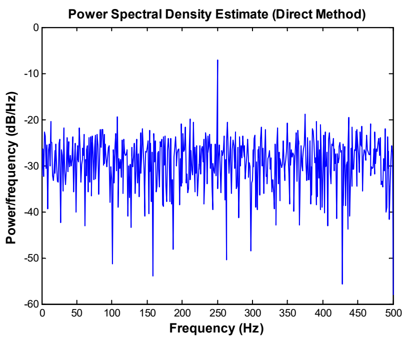


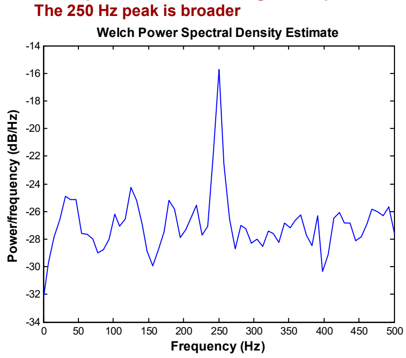
- Welch Method in MATLAB
  - [ps, f] = pwelch (x, window, noverlap, nfft, fs);
    - ✓x: a discrete-time signal vector using Welch's averaged
      - **By default**, x is divided into eight sections with 50% overlap, each section is windowed with a Hamming window, and eight modified periodograms are computed and averaged.
    - ✓window: a vector containing a window function
      - Dividing x into overlapping sections of length equal to the length of "window", and then windows each section with the vector specified in "window".
      - If "window" is an integer, x is divided into sections of length equal to that integer value, and a Hamming window of equal length is used.
    - ✓noverlap: specifying the overlap in smaples.
    - ✓nfft: specifying the number of FFT points used to calculate the PSD estimate.
    - ✓fs: specifying the sampling frequency and is used to fill the frequency vector, f, in the output with appropriate values.


```
N = 1024;
[data,t,snr] = sig_noise(250,-7,N); % Get data (250 Hz sin plus white noise)

% Estimate the Welch spectrum using 128 point segments, the triangular filter,
%   a 50% overlap, and get 95% confidence interval.  Fs = 1kHz Plot in db.

[PS,f] = pwelch(data, 128,64,128,1000);
plot(f,PS,'b');                                          % Plot PS in log scale
title('Power Spectrum (Welch Method)');
xlabel('Frequency (Hz)');
ylabel('Power Spectrum');
```

Using Hamming Window

The improvement in the background spectra
The 250 Hz peak is broader





– Two spectra computed for a waveform consisting of two closely spaced sine waves (235 and 250 Hz) in noise (SNR=-10 dB)
– Welch's method was used for both with the same parameters (nfft=128, overlap=64) except for the window functions.
– The triangular window with a smaller mainlobe gives rise to a spectrum that shows two peaks, in contrast to Chebyshev windowed spectrum.