Filtering and Spectral Analysis

Borbala Hunyadi

Delft University of Technology, The Netherlands

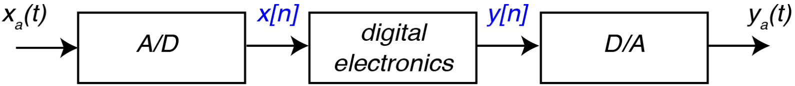
May 15, 2023

Part 1: Filtering using DFT

Recap: DSP

Digital Signal Processing

Processing of analog signals by means of discrete-time operations implemented on digital hardware



sampling
quantization

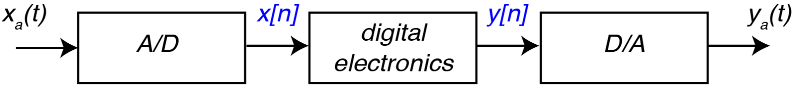
filtering
spectral analysis

reconstruction

Recap: DSP

Digital Signal Processing

Processing of analog signals by means of discrete-time operations implemented on digital hardware



sampling	*filtering*	reconstruction
quantization	*spectral analysis*	

Prerequisite: digital representation of the spectrum of X(f)!

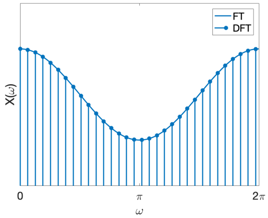
Recap: Discrete Fourier Transform

We have shown that, given a discrete time signal $x[n]$, it is possible to reconstruct its spectrum $X(\omega)$ from its N equidistant samples in case $x[n]$ is of finite length with $n = 0, 1, ..., L-1$ and $N \geq L$.

Definition

The Discrete Fourier Transform (DFT) of a sequence $x[n]$ is

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j2\pi \frac{kn}{N}}, \text{ for } 0 \leq k \leq N-1$$



Recap: Inverse Discrete Fourier Transform

We have also shown that the discrete time signal $x[n]$ can be reconstructed from N equidistant samples of its spectrum $X(\omega)$ in case $x[n]$ has finite length with $n = 0, 1, \ldots, L-1$ and $N \geq L$

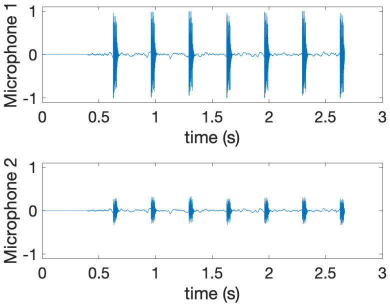
Definition

The DFT values (i.e. $X[k]$, $0 \leq k \leq N-1$) uniquely define the discrete time signal $x[n]$, $n = 0, 1, \ldots L-1$, $L \leq N$ through the Inverse Discrete Fourier Transform (IDFT):

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] e^{j2\pi \frac{kn}{N}}, \text{ for } 0 \leq n \leq N-1$$

Recap: DFT - Example 4

Channel estimation (EE2T11):

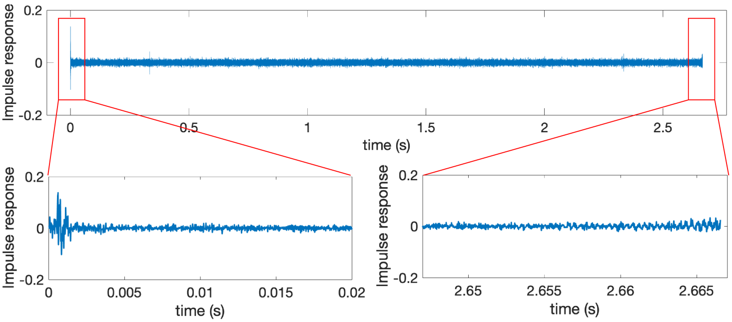


Deconvolution in the frequency domain:

1. Compute $X(k)$ and $Y(k)$, the DFTs of x and y (microphone close and far from the loudspeaker, resp.)
2. Compute $H(k) = \frac{Y(k)}{X(k)}$
3. Compute $h(k)$ from $H(k)$ using the IDFT

Recap: DFT - Example 4

Channel estimation result:



As expected, the impulse response tends to zero after the initial peak. However, an increase in values is observed in the end of the response. These values belong to the negative time of the impulse response. They appear in the end due to the circular convolution proptery of DFT.

DFT and circular convolution

Given two finite-length sequences $x_1[n]$ and $x_2[n]$ and their DFTs $X_1[k]$ and $X_2[k]$. If a certain DFT $X_3[k] = X_1[k] \cdot X_2[k]$, then the corresponding finite length sequence $x_3[n]$ can be written as the circular convolution of the two sequences:

$$x_3[m] = \sum_{n=0}^{N-1} x_1[n]x_2[m-n]_N \text{ , where}$$

$[]_N$ denotes the modulo operation.

Goal of the lecture

During this lecture we will see how to implement linear convolution using circular convolution, such that filtering (i.e. linear convolution in time domain) can be implemented using DFT in the frequency domain on a digital computer.

Outline

- Circular convolution
- Filtering using DFT: linear convolution via circular convolution
- Filtering long sequences

Circular convolution and the modulo operation

Meaning of the modulo operation $[\,]_N$: Given $[k-l]_N$, if $m = k - l$ is outside the range [0...N-1], then the modulo operation takes an integer p such that $k - l + pN$ is within the range [0...N-1]. Then $[k-l]_N = k - l + pN$.

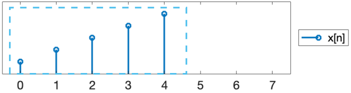
Example: $[2-3]_5 = 4$

Circular convolution and the modulo operation

Meaning of the modulo operation $[]_N$: Given $[k-l]_N$, if $m = k - l$ is outside the range [0...N-1], then the modulo operation takes an integer p such that $k - l + pN$ is within the range [0...N-1]. Then $[k-l]_N = k - l + pN$.

Example: $[2-3]_5 = 4$

Graphical example:

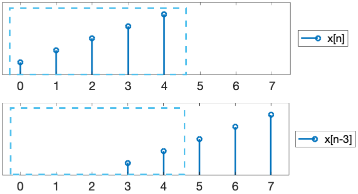


Circular convolution and the modulo operation

Meaning of the modulo operation $[]_N$: Given $[k-l]_N$, if $m = k - l$ is outside the range [0...N-1], then the modulo operation takes an integer p such that $k - l + pN$ is within the range [0...N-1]. Then $[k-l]_N = k - l + pN$.

Example: $[2-3]_5 = 4$

Graphical example:

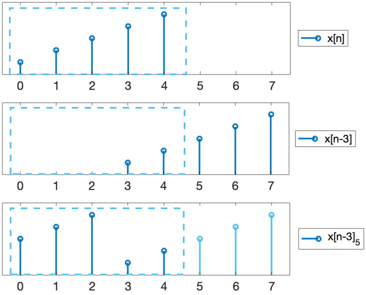


Circular convolution and the modulo operation

Meaning of the modulo operation $[]_N$: Given $[k-l]_N$, if $m = k - l$ is outside the range [0...N-1], then the modulo operation takes an integer p such that $k - l + pN$ is within the range [0...N-1]. Then $[k-l]_N = k - l + pN$.

Example: $[2-3]_5 = 4$

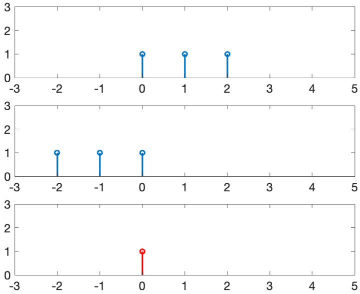
Graphical example:



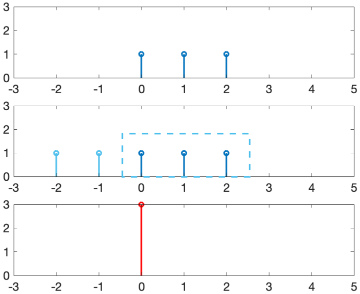
Linear vs circular convolution

Example: $x_1 = x_2 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$. What is x_3?

Linear convolution:



Circular convolution:



$$x_3[m] = \sum_{n=0}^{N-1} x_1[n] x_2[m-n],$$

$$m = 0, 1, \ldots L + M - 1$$

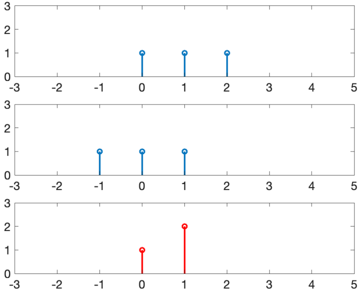
$$x_3[m] = \sum_{n=0}^{N-1} x_1[n] x_2[m-n]_N,$$

$$m = 0, 1, \ldots N - 1$$

Linear vs circular convolution

Example: $x_1 = x_2 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$. What is x_3?

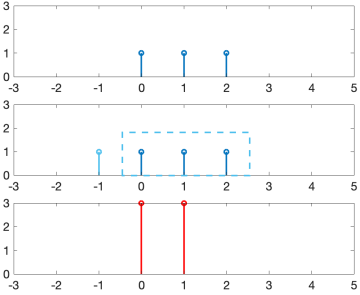
Linear convolution:



$$x_3[m] = \sum_{n=0}^{N-1} x_1[n] x_2[m-n],$$

$$m = 0, 1, \ldots L + M - 1$$

Circular convolution:



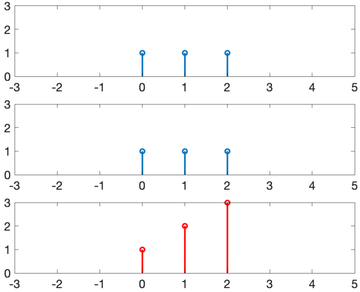
$$x_3[m] = \sum_{n=0}^{N-1} x_1[n] x_2[m-n]_N,$$

$$m = 0, 1, \ldots N - 1$$

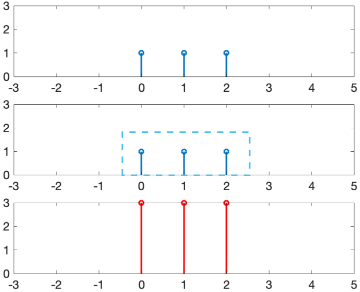
Linear vs circular convolution

Example: $x_1 = x_2 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$. What is x_3?

Linear convolution:



Circular convolution:



$$x_3[m] = \sum_{n=0}^{N-1} x_1[n] x_2[m-n],$$

$$m = 0, 1, \ldots L + M - 1$$

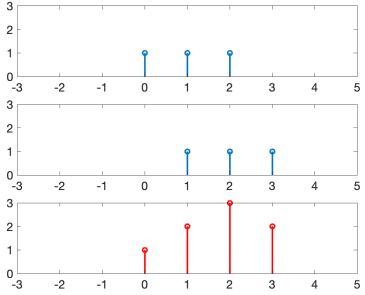
$$x_3[m] = \sum_{n=0}^{N-1} x_1[n] x_2[m-n]_N,$$

$$m = 0, 1, \ldots N - 1$$

Linear vs circular convolution

Example: $x_1 = x_2 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$. What is x_3?

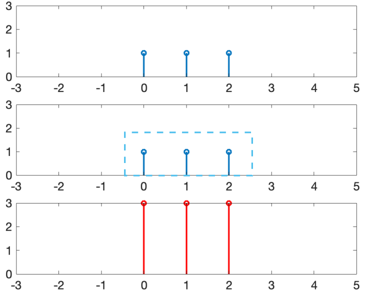
Linear convolution:



$$x_3[m] = \sum_{n=0}^{N-1} x_1[n] x_2[m-n],$$

$$m = 0, 1, \ldots L + M - 1$$

Circular convolution:



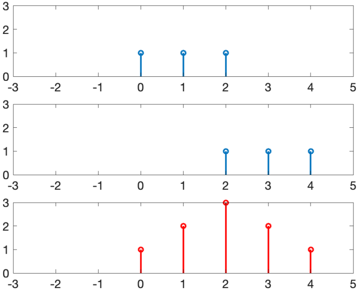
$$x_3[m] = \sum_{n=0}^{N-1} x_1[n] x_2[m-n]_N,$$

$$m = 0, 1, \ldots N - 1$$

Linear vs circular convolution

Example: $x_1 = x_2 = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}$. What is x_3?

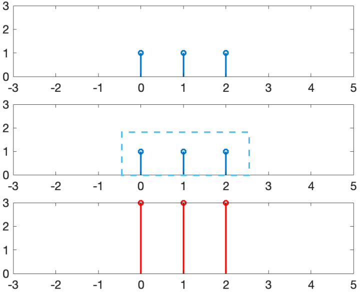
Linear convolution:



$$x_3[m] = \sum_{n=0}^{N-1} x_1[n]x_2[m-n],$$

$$m = 0, 1, \dots L + M - 1$$

Circular convolution:



$$x_3[m] = \sum_{n=0}^{N-1} x_1[n]x_2[m-n]_N,$$

$$m = 0, 1, \dots N - 1$$

Filtering using DFT

Let $x[n]$, $n = 0, 1, ..., L - 1$ be the finite-length input of a system with finite impulse response $h[n]$, $n = 0, 1, ...M - 1$. Then, the output of the system is a length $L + M - 1$ sequence that can be expressed as the linear convolution of the sequences:

$$y[n] = \sum_{k=0}^{M-1} h[k]x[n-k]$$

Or as a multiplication of their (continuous!) spectra:

$$Y(\omega) = X(\omega)H(\omega)$$

As $y[n]$ is of length $L + M - 1$, we need to represent $Y(\omega)$ with at least $L + M - 1$ samples in the frequency domain, i.e. with an N-point DFT where $N \geq L + M - 1$.

Filtering using DFT

Let $x[n]$, $n = 0, 1, ..., L-1$ be the finite-length input of a system with finite impulse response $h[n]$, $n = 0, 1, ...M-1$. Then, the output of the system is a length $L+M-1$ sequence that can be expressed as the linear convolution of the sequences:

$$y[n] = \sum_{k=0}^{M-1} h[k]x[n-k]$$

Or as a multiplication of their (continuous!) spectra:

$$Y(\omega) = X(\omega)H(\omega)$$

As $y[n]$ is of length $L+M-1$, we need to represent $Y(\omega)$ with at least $L+M-1$ samples in the frequency domain, i.e. with an N-point DFT where $N \geq L+M-1$. We cannot achieve this directly using the M-point DFT of $x[n]$ and L-point DFT of $h[n]$!

Filtering using DFT

Let's sample the continuous spectra at N equidistant samples:

$$Y(\frac{2\pi k}{N}) = H(\frac{2\pi k}{N})X(\frac{2\pi k}{N})$$

We have seen that the N-point DFT of a length $L < N$ sequence can be obtained by zero-padding the sequence. Thus, let

$$x_N[n] = \begin{cases} x[n], & n \leq L-1 \\ 0, & L < n \leq N-1 \end{cases}$$

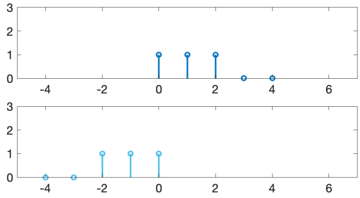
$$h_N[n] = \begin{cases} h[n], & n \leq M-1 \\ 0, & M < n \leq N-1 \end{cases}$$

Then, taking their N-point DFTs $X_N[k]$ and $H_N[k]$, we can obtain $Y[k] = H_0[k]X_0[k]$.

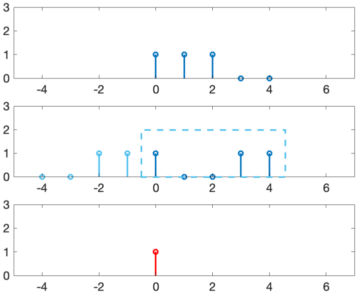
Linear convolution through circular convolution

The above result also implies that linear convolution of finite-length sequences is equivalent to the N-point circular convolution of their zero-padded (till length N) versions. Let $x_1 = x_2 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \end{bmatrix}$, and

$$x_3[m] = \sum_{n=0}^{N-1} x_1[n] x_2[m-n]_N .$$



$x_1[n]$ $x_2[0-n]$

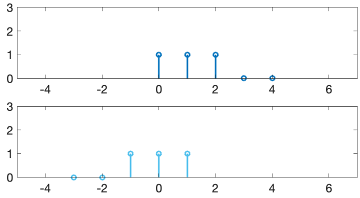


$x_1[n]$ $x_2[0-n]_5$ $x_3[0]$

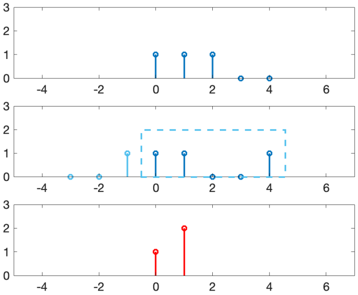
Linear convolution through circular convolution

The above result also implies that linear convolution of finite-length sequences is equivalent to the N-point circular convolution of their zero-padded (till length N) versions. Let $x_1 = x_2 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \end{bmatrix}$, and

$$x_3[m] = \sum_{n=0}^{N-1} x_1[n] x_2[m-n]_N.$$



$x_1[n]$ $x_2[1-n]$

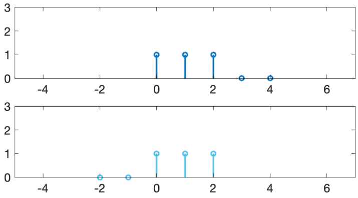


$x_1[n]$ $x_2[1-n]_5$ $x_3[1]$

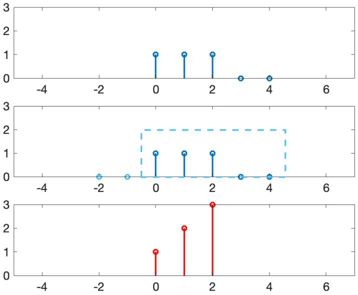
Linear convolution through circular convolution

The above result also implies that linear convolution of finite-length sequences is equivalent to the N-point circular convolution of their zero-padded (till length N) versions. Let $x_1 = x_2 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \end{bmatrix}$, and

$$x_3[m] = \sum_{n=0}^{N-1} x_1[n] x_2[m-n]_N.$$



$x_1[n]$ $x_2[2-n]$

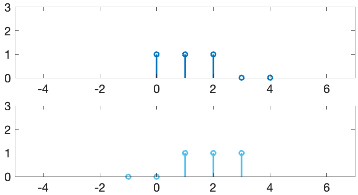


$x_1[n]$ $x_2[2-n]_5$ $x_3[2]$

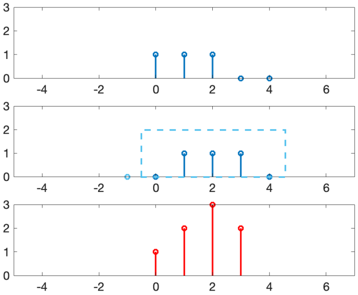
Linear convolution through circular convolution

The above result also implies that linear convolution of finite-length sequences is equivalent to the N-point circular convolution of their zero-padded (till length N) versions. Let $x_1 = x_2 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \end{bmatrix}$, and

$$x_3[m] = \sum_{n=0}^{N-1} x_1[n] x_2[m-n]_N.$$



$x_1[n]$ $x_2[3-n]$

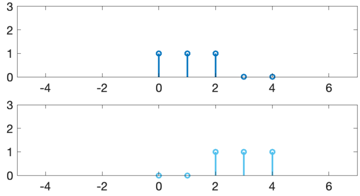


$x_1[n]$ $x_2[3-n]_5$ $x_3[3]$

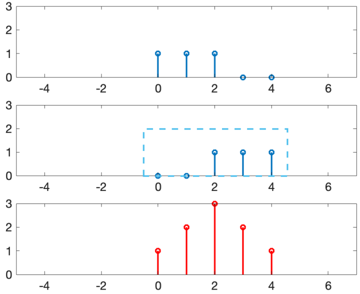
Linear convolution through circular convolution

The above result also implies that linear convolution of finite-length sequences is equivalent to the N-point circular convolution of their zero-padded (till length N) versions. Let $x_1 = x_2 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \end{bmatrix}$, and

$$x_3[m] = \sum_{n=0}^{N-1} x_1[n] x_2[m-n]_N.$$



$x_1[n]$ $x_2[4-n]$



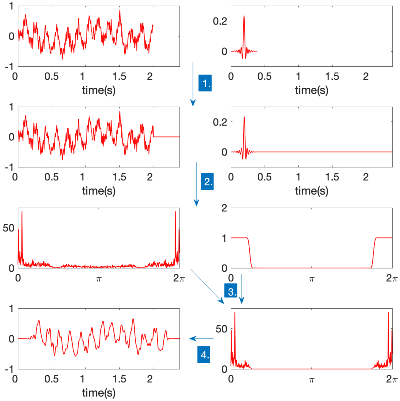
$x_1[n]$ $x_2[4-n]_5$ $x_3[4]$

Filtering using DFT: Example

How to filter a length N_x EEG segment (left) with a low-pass filter with an impulse reponse (right) of length N_y?

Procedure:

1. Zero-pad both sequences to $N = N_x + N_y - 1$
2. Compute their N-point DFTs
3. Multiply their DFTs
4. Compute the inverse DFT to obtain the filtered EEG segment



Filtering long sequences

Practical problems:

- filtering long blocks is computationally intensive
- requires large memory
- results are required in real-time

Various applications, e.g.:

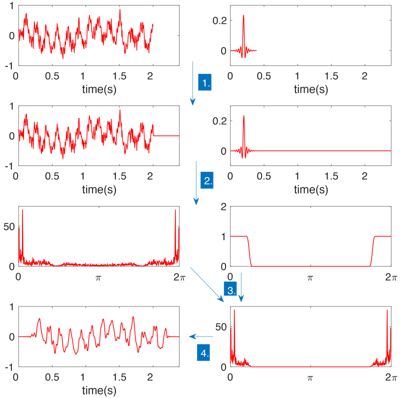
- EEG monitoring lasting several days
- audio processing for hearing aids

Solution:

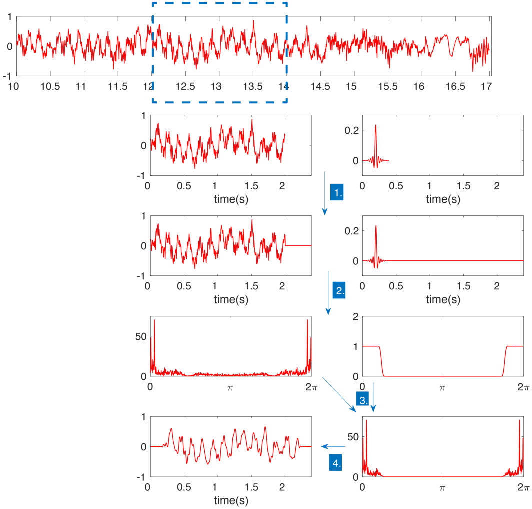
1. segment the sequence to short fixed-size blocks
2. process the blocks separately
3. fit the processed blocks together

Filtering long sequences: overlap-save

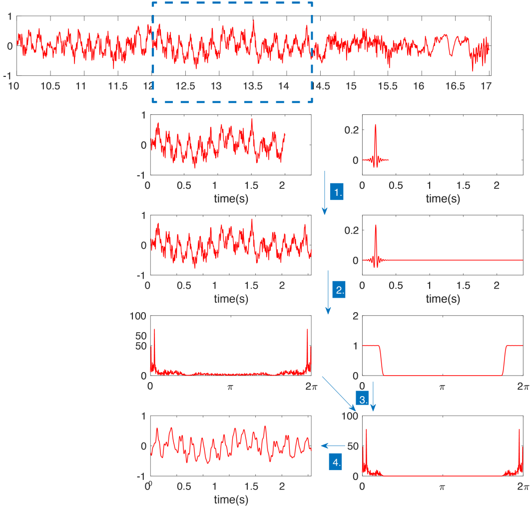
In the previous example, the EEG segment came from a long recording. Instead of zero-padding, we can use the upcoming data samples.



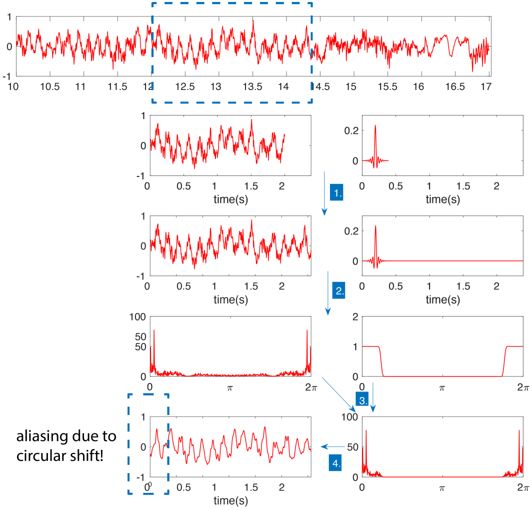
Filtering long sequences: overlap-save



Filtering long sequences: overlap-save



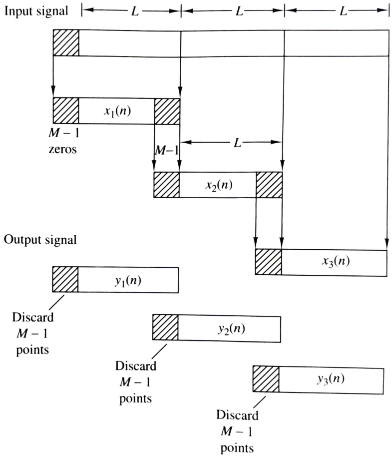
Filtering long sequences: overlap-save



Filtering long sequences: overlap-save

In order to avoid aliasing,

- we discard the first M-1 samples of the output block, where M is the length of the impulse reponse of the filter.
- these points become the first M-1 samples of the next output block
- in the very first output block the first M-1 samples are replaced by zeros.

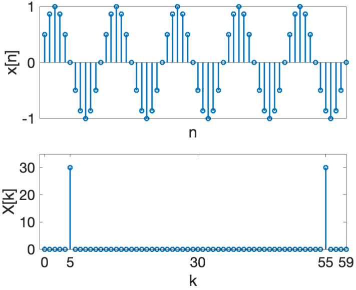


Part 2: Spectral Analysis using DFT

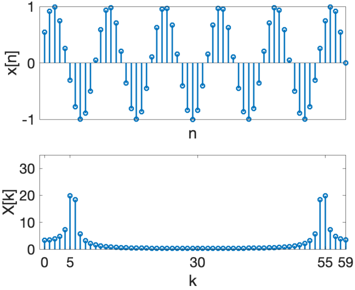
Recap: DFT - Example 1

$$x[n] = sin(\theta n)\ ,\ 0 \leq n \leq N-1\ ,\ N = 60$$

Case 1:
θ is an integer multiple of $2\pi/N$
e.g. $\theta = 2\pi \cdot 5/N$



Case 2:
θ is not an integer multiple of $2\pi/N$
e.g. $\theta = 2\pi \cdot 5.5/N$



Recap: Example 1 - Explanation

Why in case 1 we have 2 non-zero DFT coefficients, while in case 2 all coefficients are non-zero?

- explanation 1: considering DFT as a projection on a linear basis.
- explanation 2: considering that the DFT is related to the Fourier coefficients of the periodic extension of the sequence.

Recap: Example 1 - Explanation

Why in case 1 we have 2 non-zero DFT coefficients, while in case 2 all coefficients are non-zero?

- explanation 1: considering DFT as a projection on a linear basis.
- explanation 2: considering that the DFT is related to the Fourier coefficients of the periodic extension of the sequence.

During this lecture we will see one more alternative explanation for this phenomenon.

Spectral analysis: general considerations

To compute the spectrum of a signal $x[n]$, we need all the samples over an infinite interval.

$$X(\omega) \equiv \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}$$

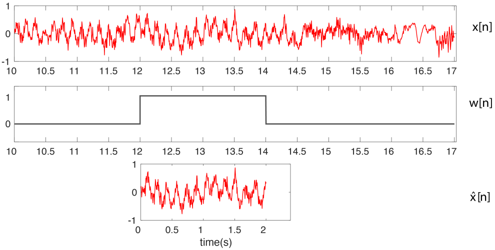
However, in practice, we only have a finite number of samples available, i.e $\hat{x}[n]$, with $n = 0, 1, ..., L-1$.

Spectral analysis: general considerations

To compute the spectrum of a signal $x[n]$, we need all the samples over an infinite interval.

$$X(\omega) \equiv \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}$$

However, in practice, we only have a finite number of samples available, i.e $\hat{x}[n]$, with $n = 0, 1, \ldots, L-1$.
$\hat{x}[n]$ can be viewed as a windowed version of $x[n]$:



$$\hat{x}[n] = x[n]w[n]$$

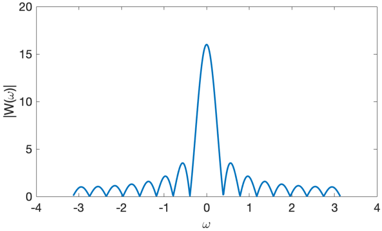
Spectral analysis: general considerations

Due to the windowing, the spectrum of the finite sequence is expressed as a convolution of the spectrum of the original sequence and the Fourier transform of the window sequence:

$$\hat{X}(\omega) = \int_{-\pi}^{\pi} X(\theta) W(\omega - \theta) d\theta$$

$$W(\omega) = \sum_{n=0}^{L-1} e^{-j\omega n} = \frac{1 - e^{-j\omega L}}{1 - e^{-j\omega}}$$

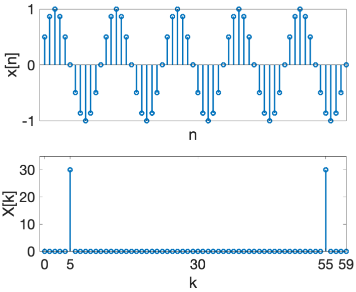
$$= \frac{sin(\omega L/2)}{sin(\omega/2)} e^{-j\omega(L-1)/2}$$



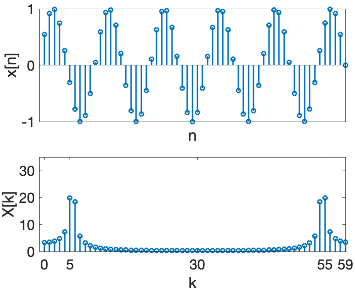
Example 1 - Alternative explanation

$$x[n] = sin(\theta n)\ ,\ 0 \leq n \leq N-1\ ,\ N = 60$$

Case 1:
θ is an integer multiple of $2\pi/N$
e.g. $\theta = 2\pi \cdot 5/N$



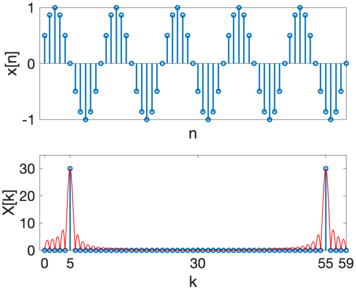
Case 2:
θ is not an integer multiple of $2\pi/N$
e.g. $\theta = 2\pi \cdot 5.5/N$



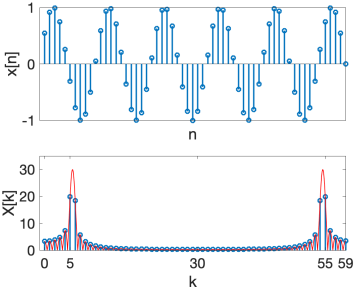
Example 1 - Alternative explanation

$$x[n] = sin(\theta n)\ ,\ 0 \leq n \leq N-1\ ,\ N = 60$$

Case 1:
θ is an integer multiple of $2\pi/N$
e.g. $\theta = 2\pi \cdot 5/N$

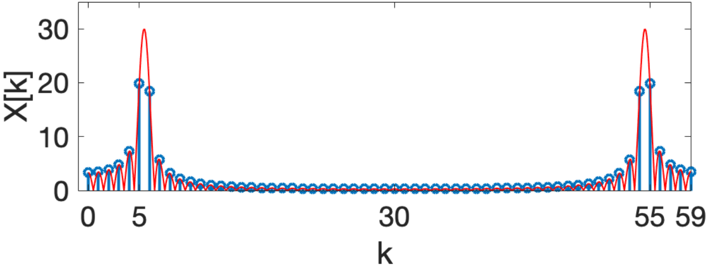


Case 2:
θ is not an integer multiple of $2\pi/N$
e.g. $\theta = 2\pi \cdot 5.5/N$



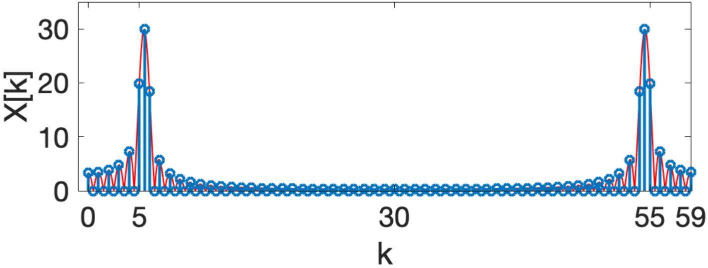
Zero-padding

Case 2:



Zero-padding

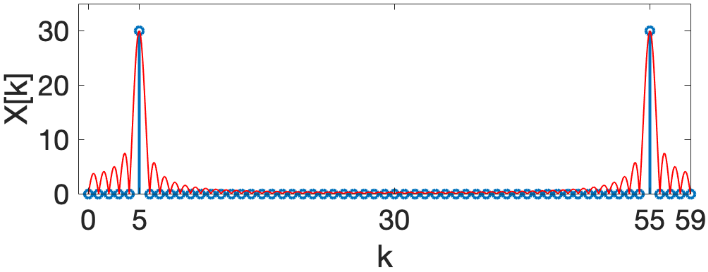
Case 2:



Zero-padding gives us more samples of the underlying continuous spectrum, i.e. improves *apparent* resolution.
However, we will see that it does not improve the actual spectral resolution, i.e. our ability to distingusih closely spaced frequencies.

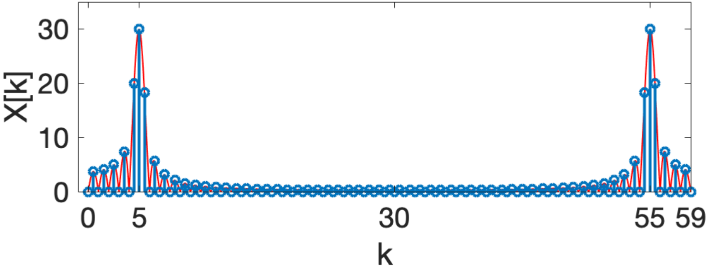
Zero-padding

Case 1:



Zero-padding

Case 1:

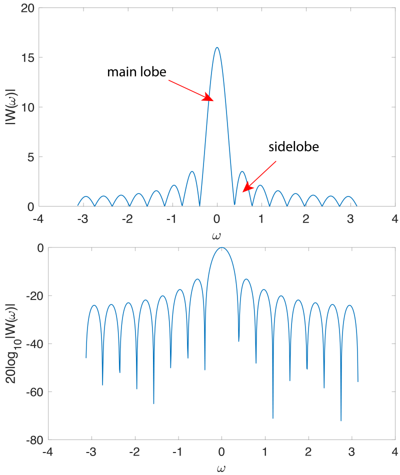


As the window function is not localized in frequency, the windowed spectrum is spread out (leaked out) over the whole frequency range - "spectral leakage"

Rectangular window

$$W(\omega) = \frac{sin(\omega L/2)}{sin(\omega/2)} e^{-j\omega(L-1)/2}$$

- Main lobe has a width of $\Delta\omega = \frac{4\pi}{L}$
- Sidelobes have an amplitude of -13dB.



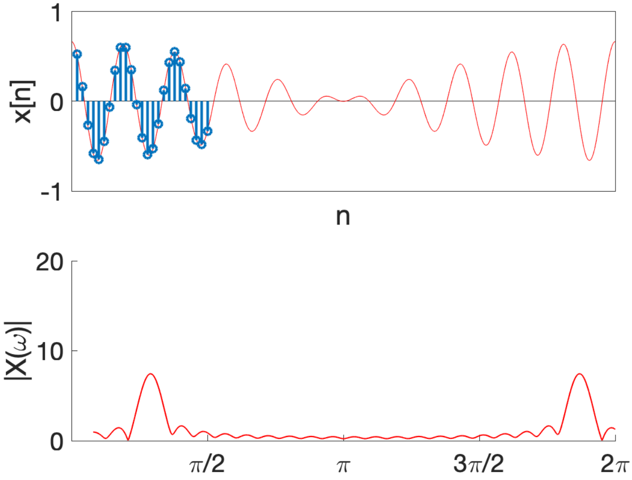
N=16

Effect of windowing

- Spectral smoothing: Due to the non-zero width of the main lobe, two closely spaced peaks in the Fourier spectum may appear as a single peak in the DFT of the finite sequence.
- Spectral leakage: The spectrum is spread out to the whole frequency range. Besides, a weak peak in the original spectrum may be masked by the "leakage" from a large peak.

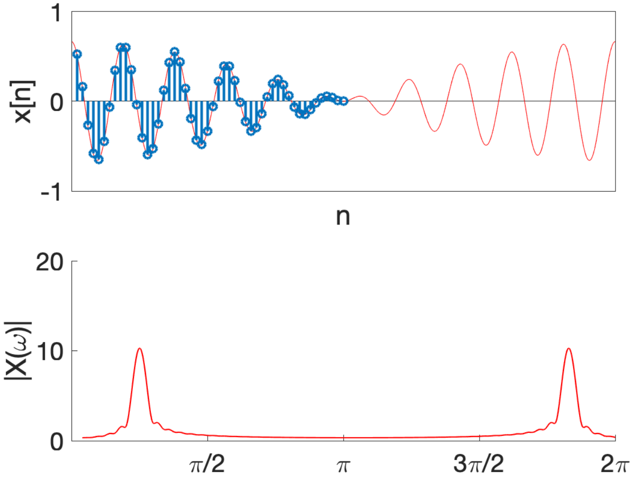
Effect of windowing - Example 1

$x[n] = 1/3(cos(\omega_1 n) + cos(\omega_2 n))$, $\omega_1 = 0.2\pi$, $\omega_2 = 0.22\pi$ with $n = 0, 1, \ldots L-1$, and $L = 25$



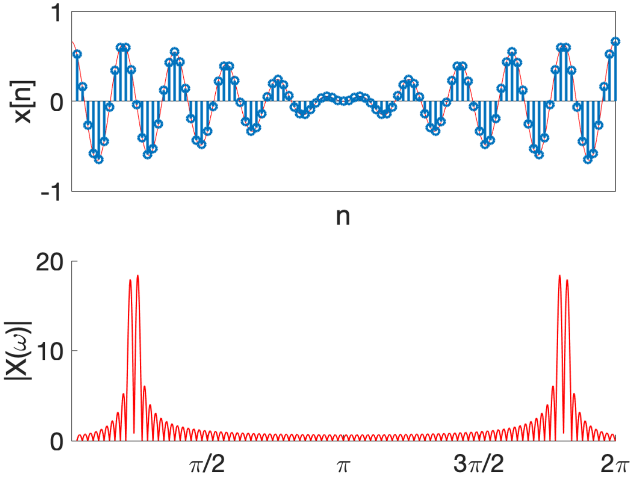
Effect of windowing - Example 1

$x[n] = 1/3(cos(\omega_1 n) + cos(\omega_2 n))$, $\omega_1 = 0.2\pi$, $\omega_2 = 0.22\pi$ with $n = 0, 1, \ldots L-1$, and $L = 50$



Effect of windowing - Example 1

$x[n] = 1/3(cos(\omega_1 n) + cos(\omega_2 n))$, $\omega_1 = 0.2\pi$, $\omega_2 = 0.22\pi$ with $n = 0, 1, \ldots L-1$, and $L = 100$



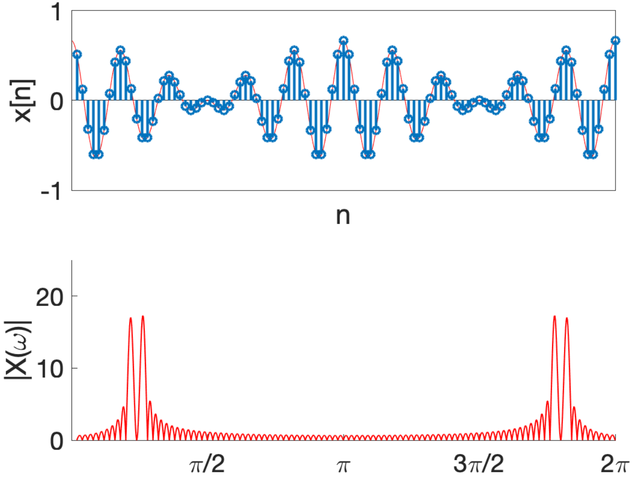
Observations from example 1

The more samples I take[1], the better the spectral resolution.

[1] Here, consider a fixed sampling rate that matches the bandwidth of the signal. That is, more samples mean a longer observation window.

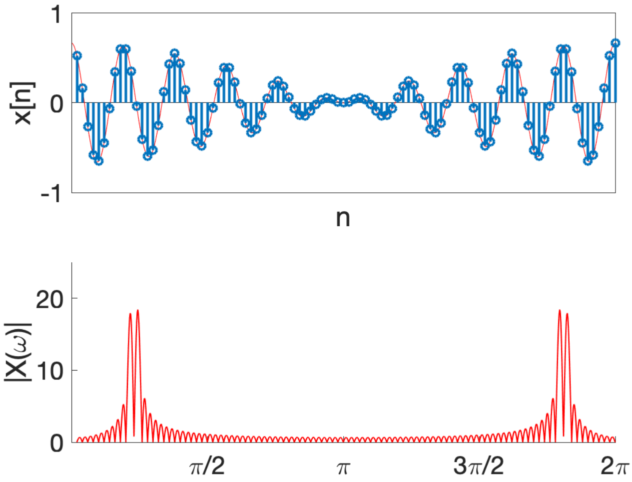
Effect of windowing - Example 2

$x[n] = 1/3(cos(\omega_1 n) + cos(\omega_2 n))$, $\omega_1 = 0.2\pi$, with $n = 0, 1, \ldots L-1$, $L = 100$ and $\omega_2 = 0.24\pi$



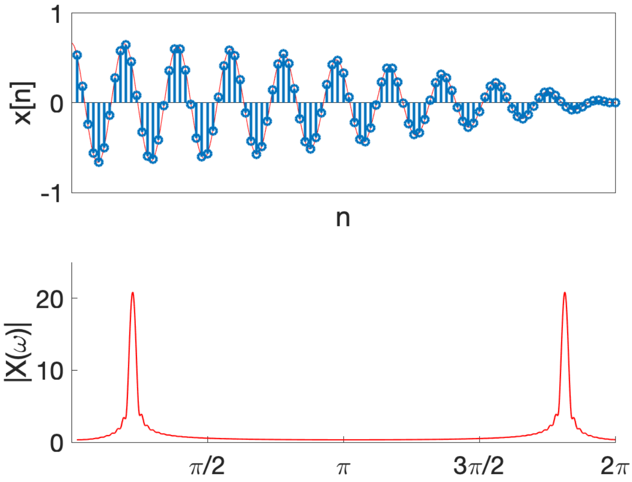
Effect of windowing - Example 2

$x[n] = 1/3(cos(\omega_1 n) + cos(\omega_2 n))$, $\omega_1 = 0.2\pi$, with $n = 0, 1, ... L-1$, $L = 100$ and $\omega_2 = 0.22\pi$



Effect of windowing - Example 2

$x[n] = 1/3(cos(\omega_1 n) + cos(\omega_2 n))$, $\omega_1 = 0.2\pi$, with $n = 0, 1, ...L-1$, $L = 100$ and $\omega_2 = 0.21\pi$



Observations from example 2

The width of the main lobe limits the spectral resolution, i.e. our ability to distinguish closely spaced spectral lines.

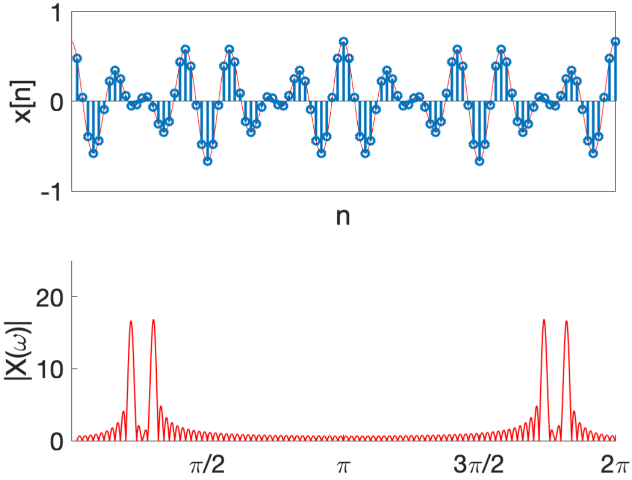
Spectral resolution and the rectangular window

The spectral resolution depends on the width of the main lobe of the window function:

- The spectrum $W(\omega)$ has its first zero-crossing at $\omega = 2\pi/L$
- Therefore, two spectral lines ω_1 and ω_2 are not distinguishable if $|\omega_1 - \omega_2| < 2\pi/L$.
- If $|\omega_1 - \omega_2| \geq 2\pi/L$ we will see two separate lobes in the frequency spectrum.

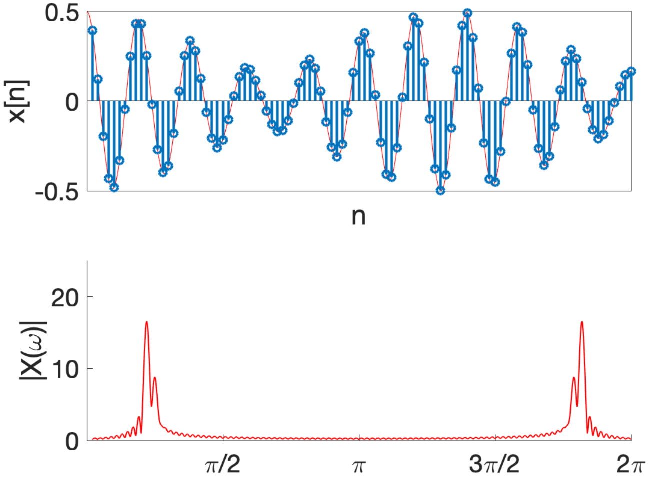
Effect of windowing - Example 3

$x[n] = 1/3(cos(\omega_1 n) + \gamma cos(\omega_2 n))$, $\omega_1 = 0.2\pi$, $\omega_2 = 0.28\pi$ with $n = 0, 1, \ldots L-1$, $L = 100$ and $\gamma = 1$



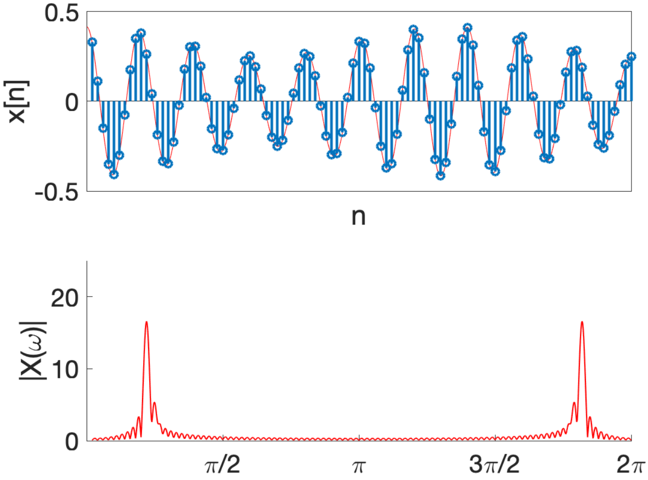
Effect of windowing - Example 3

$x[n] = 1/3(cos(\omega_1 n) + \gamma cos(\omega_2 n))$, $\omega_1 = 0.2\pi$, $\omega_2 = 0.28\pi$ with $n = 0, 1, ...L-1$, $L = 100$ and $\gamma = 0.5$



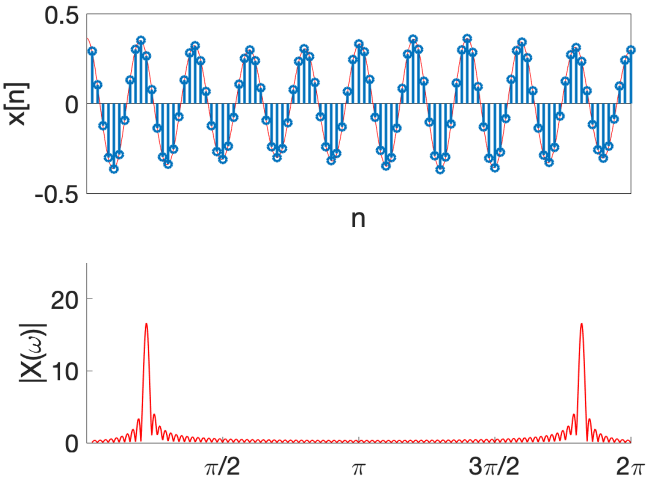
Effect of windowing - Example 3

$x[n] = 1/3(cos(\omega_1 n) + \gamma cos(\omega_2 n))$, $\omega_1 = 0.2\pi$, $\omega_2 = 0.28\pi$ with $n = 0, 1, \ldots L-1$, $L = 100$ and $\gamma = 0.25$



Effect of windowing - Example 3

$x[n] = 1/3(cos(\omega_1 n) + \gamma cos(\omega_2 n))$, $\omega_1 = 0.2\pi$, $\omega_2 = 0.28\pi$ with $n = 0, 1, \ldots L-1$, $L = 100$ and $\gamma = 0.1$



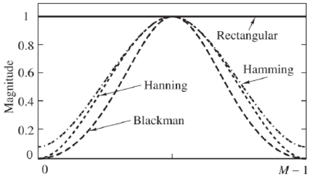
Observations from example 3

The sidelobe of the window function can mask a weak spectral line.

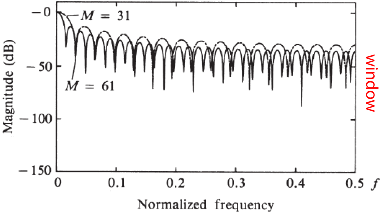
Choice of the window function

In general, there is a trade-off between the width of the main lobe and the amplitude of the sidelobes:

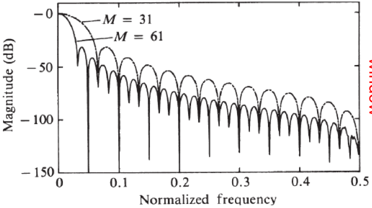
window functions in time domain:



window function in frequency domain:



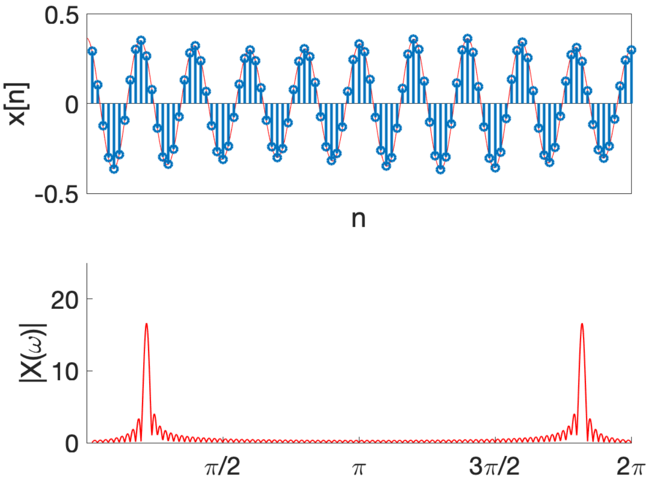
rectangular window



Hanning window

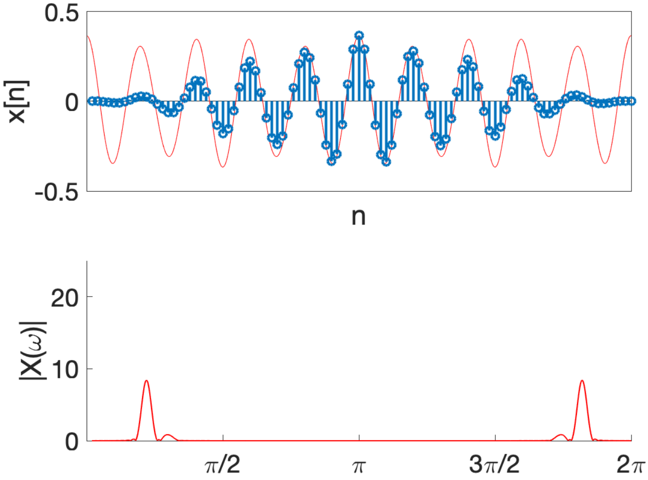
Choice of window - Example 1

$x[n] = 1/3(cos(\omega_1 n) + \gamma cos(\omega_2 n))$, $\omega_1 = 0.2\pi$, $\omega_2 = 0.28\pi$ with $n = 0, 1, \ldots L-1$, $L = 100$, $\gamma = 0.1$ and using a rectangular window



Choice of window - Example 1

$x[n] = 1/3(cos(\omega_1 n) + \gamma cos(\omega_2 n))$, $\omega_1 = 0.2\pi$, $\omega_2 = 0.28\pi$ with $n = 0, 1, \ldots L-1$, $L = 100$, $\gamma = 0.1$ and using a Hanning window



Summary

We considered the spectral analysis of an (infinite) sequence in practice, approximated from a finite data record.

- In time domain the finite sequence is equivalent to multiplying the infinite sequence with a window funcion
- In the frequency domain the spectrum (DTFT) of the finite sequence is equivalent to the convolution of the DTFT of the infinite sequence with the DTFT of the window sequence.
- The spectral resolution will depend on the width of the main lobe of the window, which, in turn, depends on the chosen window function and the number of samples L we take.
- There is a trade-off between resolution and spectral leakage, i.e. the width of the main lobe and the amplitude of the sidelobes.
- Taking an N-point DFT with $N > L$ (i.e. zero-padding) will not increase the spectral resolution (i.e. the information contained in the DTFT) but it will give us more samples of it (i.e. may increase apparent resolution)