DSZOB
Digitálne spracovanie zvuku, obrazu a biosignálov

Fourier transform
FFT

Some basic correspondences

Príklad: pravoúhly puls
Example : Rectangular Pulse ⇔ sinc

Uvažujeme pravoúhly puls definovaný okolo počiatku

(Consider the non-periodic rectangular pulse at zero)

$$x(t) = \begin{cases} 1 & |t| < T_1 \\ 0 & |t| \geq T_1 \end{cases}$$

$$X(j\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt = \int_{-T_1}^{T_1} e^{-j\omega t} dt$$

$$= \frac{1}{-j\omega} e^{-j\omega t} \Big|_{-T_1}^{T_1}$$

$$= \frac{2\sin(\omega T_1)}{\omega}$$

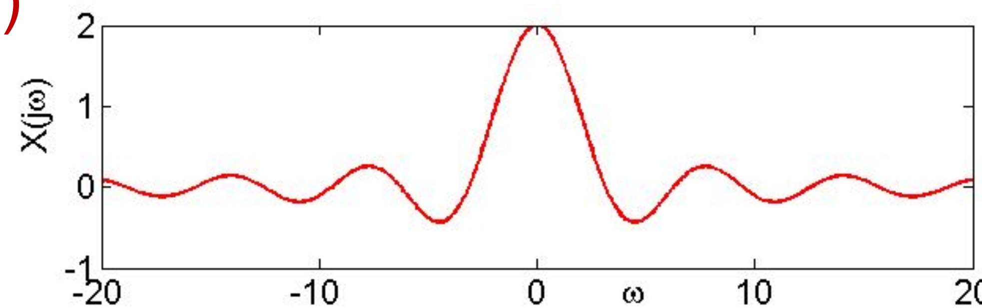
Fourierova transformácia pravoúhleho pulsu je funkcia:

The Fourier transform of single rectangular pulse is function :

$\text{sinc}(\omega T_1) = \sin(\omega T_1)/(\omega)$

Note, the values are real

$T_1 = 1$

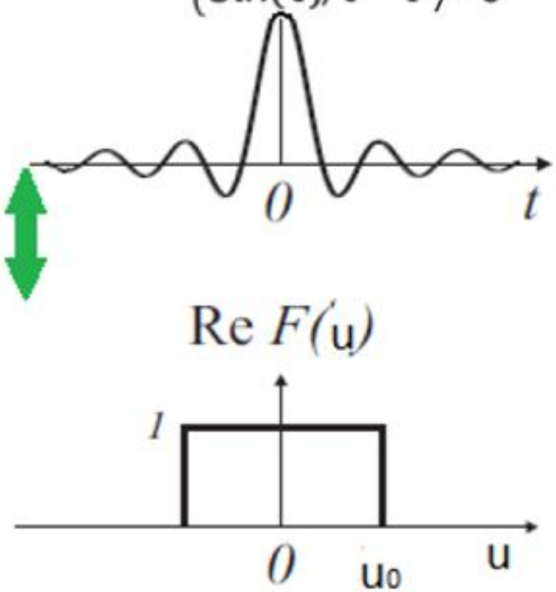


Fukcia a jej Fourierovské spectrum sinc(t)⇔ rectangle pulse

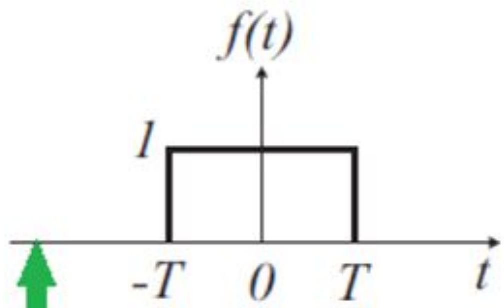
Sinc(x) function is defined for x ≠ 0 by t $\mathrm{sinc}(x) = \dfrac{\sin(x)}{x}$

the value at x = 0 is defined to be the limiting \ $\mathrm{sinc}(0) = \lim\limits_{x \to 0} \dfrac{\sin(x)}{x} = 1$

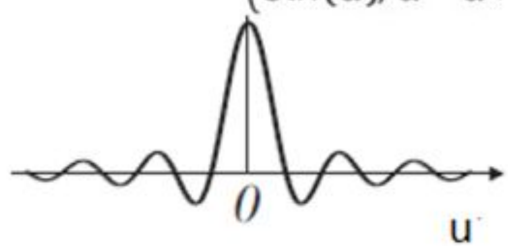
$$sinc(t) = \begin{cases} 1 & t = 0 \\ sin(t)/t & t \neq 0 \end{cases}$$



Obrázok 1.5: Funkcia sinc(t) a jej reálne spektrum.

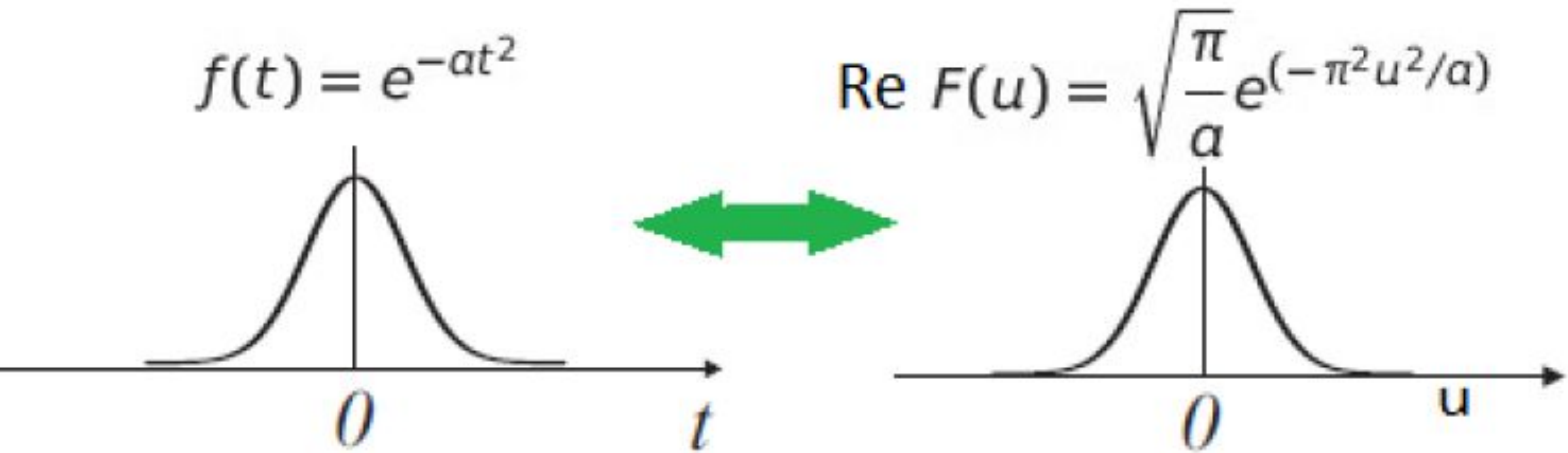


$$\mathrm{Re}\ F(u) = \begin{cases} 1 & u = 0 \\ sin(u)/u & u \neq 0 \end{cases}$$



Obrázok 1.6: Obdĺžniková funkcia a jej reálne spektrum.

Gaussova funkcia a jej Fourierovské spectrum
Gauss function ⇔ Gauss function



Obrázok 1.7: Gaussova funkcia a jej reálne spektrum.

Vlastnosti Fourierovej transformácie
Fourier Transform Properties

Linearita Fourierovej transformácie
(Linearity of Fourier Transform)

Majme pár:

funkciu f1(t) a k nej príslušné spektrum F1(u).

Majme druhý pár:

funkciu f2(t) a k nej príslušné spektrum F2(u).

Takéto páry budeme označovať ako:

$f1(t) \Longleftrightarrow F1(u)$,

$f2(t) \Longleftrightarrow F2(u)$.

Ďalej sú dané konštanty: K1 a K2.

Potom platí, že:

$$K1*f1(t)+K2*f2(t) \Longleftrightarrow K1*F1(u)+K2*F2(u)$$

The Fourier Transform is a linear transform. That is, let's say we have two functions f1*(t) and* f2*(t), with Fourier Transforms given by* F1(u) *and* F2(u)*, respectively. Then the Fourier Transform of a linear combination of f1 and f2 (using constats K1 and K2) can be easily found:*

$$K1*f1(t)+K2*f2(t) \Longleftrightarrow K1*F1(u)+K2*F2(u)$$

Vlastnosti Fourierovej transformácie Veta o posunutí v čase (time shifting)

Majme transformačný pár: funkciu $f(x)$ a k nej príslušné spektrum $F(u)$. $f(x) \Longleftrightarrow F(u)$.

Keď signál $f(x)$ posunieme v čase o t_0, tak pre jeho spektrum platí

$$f(x - t_0) \Longleftrightarrow e^{-j\,2\pi u t_0} F(u).$$

Na strane spektra po posunutí signálu v čase pribudol multiplikatívny člen, spektrum bude vynásobené komplexnou jednotkou : $e^{-j\,2\pi u t_0}$.

Magnitúdové spektrum posunutého signálu zostáva teda nezmenené, zmení sa len fázové spektrum.

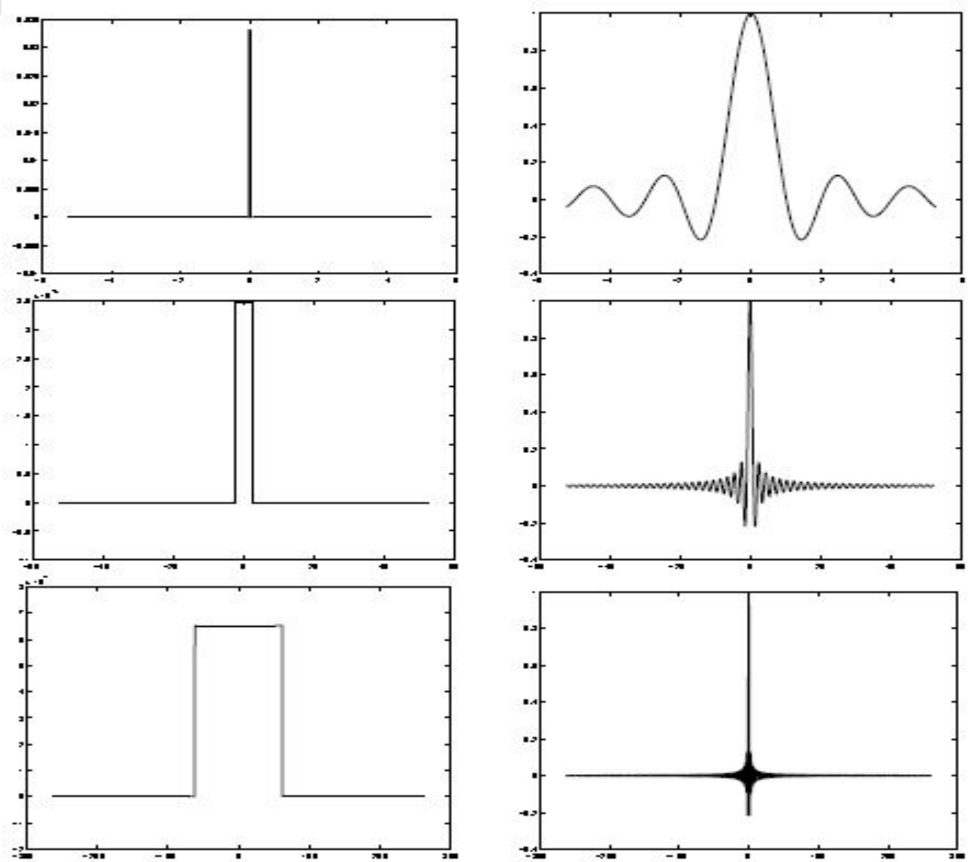
The time-shifting property means that a shift in time corresponds to a phase rotation in the frequency domain: ***F{f(x-t0)}=exp(− j 2πfu t0) F(u).***

Veta o podobnosti
Veta o zmene mierky času -Time scaling

Veta o podobnosti. Nazývaná je tiež ako veta o zmene mierky času

$$f(at) \Longleftrightarrow \frac{1}{|a|} F(u/a).$$

„Roztiahnutie“ signálu v časovej osi sa prejaví ako „zúženie“ signálu jeho spektra a naopak.



Ilustrácia vety o podobnosti. Na x-ovej osi je premenná *x* pre vstupný signál resp. premenná *u* pre spektrum. Na y-novej osi je amplitúda signálu resp. spektra.

DFT matrix

DFT matrix

Calculation of DFT using DFT matrix

$$W = \left(\frac{\omega^{jk}}{\sqrt{N}}\right)_{j,k=0,\ldots,N-1}$$

$$\omega = e^{-\frac{2\pi i}{N}}$$

$$\begin{pmatrix} F_0 \\ F_1 \\ F_2 \\ \vdots \\ F_{(N-1)} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & \ldots & 1 \\ 1 & W^1 & \ldots & W^{N-1} \\ 1 & W^2 & \ldots & W^{2N-2} \\ \vdots & \vdots & & \vdots \\ 1 & W^{N-1} & \ldots & W^{(N-1)^2} \end{pmatrix}}_{\mathbf{W}} . \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{(N-1)} \end{pmatrix}$$

Maticový zápis výpočtu DFT skrátene napíšeme ako

$$\mathbf{F} = \mathbf{W}.\mathbf{f},$$

DFT spôsoby výpočtu

1. Z definície:

$$F(u)=\frac{1}{M}\sum_{x=0}^{M-1} f(x)e^{-j2\pi ux/M}$$

$$u = 0,\ldots,\text{M-1}$$

2. Transformačnou maticou

$$\begin{pmatrix} F_0 \\ F_1 \\ F_2 \\ \vdots \\ F_{(N-1)} \end{pmatrix} = \underbrace{\begin{pmatrix} 1 & 1 & \ldots & 1 \\ 1 & W^1 & \ldots & W^{N-1} \\ 1 & W^2 & \ldots & W^{2N-2} \\ \vdots & \vdots & & \vdots \\ 1 & W^{N-1} & \ldots & W^{(N-1)^2} \end{pmatrix}}_{\mathbf{W}} \cdot \begin{pmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{(N-1)} \end{pmatrix}$$

Maticový zápis výpočtu DFT skrátene napíšeme ako

$$\mathbf{F} = \mathbf{W}.\mathbf{f},$$

3. FFT

FFT

Fast Fourier Transform FFT

Keď v roku 1965 Cooley a Tukey ohlásili objavenie rýchlej Fourierovej transformácie (FFT) , spôsobili revolúciu digitálneho spracovania signálov.

When in 1965 Cooley and Tukey announced discovery of Fast Fourier Transform (FFT) it revolutionised Digital Signal Processing.

- jedna z najviac rozvinutých oblastí DSP (Digital Signal Processing)
- existuje veľa rôznych typov a variácií algoritmu FFT.
- najzákladnejší algoritmus radix-2 vyžaduje, aby N bolo mocninou 2.

FFT je veľmi elegantný a efektívny algoritmus, ktorý je stále jedným z najpoužívanejších algoritmov v spracovaní reči, komunikácii, frekvenčnom odhade atď

Fast Fourier Transform FFT
Základný koncept FFT

Základný princíp FFT pre vektory s dĺžkou N s ľubovoľným základom (mixed-radix FFT), ktorý predstavili už autori Danielson a Lanczos v roku 1942, je faktorizácia.

Ak N môže byť rozložené na súčin n_f prirodzených čísiel $N = f_1 f_2 ... f_{Nf}$, tak výpočet DFT môže byť rozdelený na výpočet viacerých DFT pre každý činiteľ zvlášť.

(N/f_1) výpočtov DFT s dĺžkou f_1 ,

(N/f_2) výpočtov DFT s dĺžkou f_2,. . .

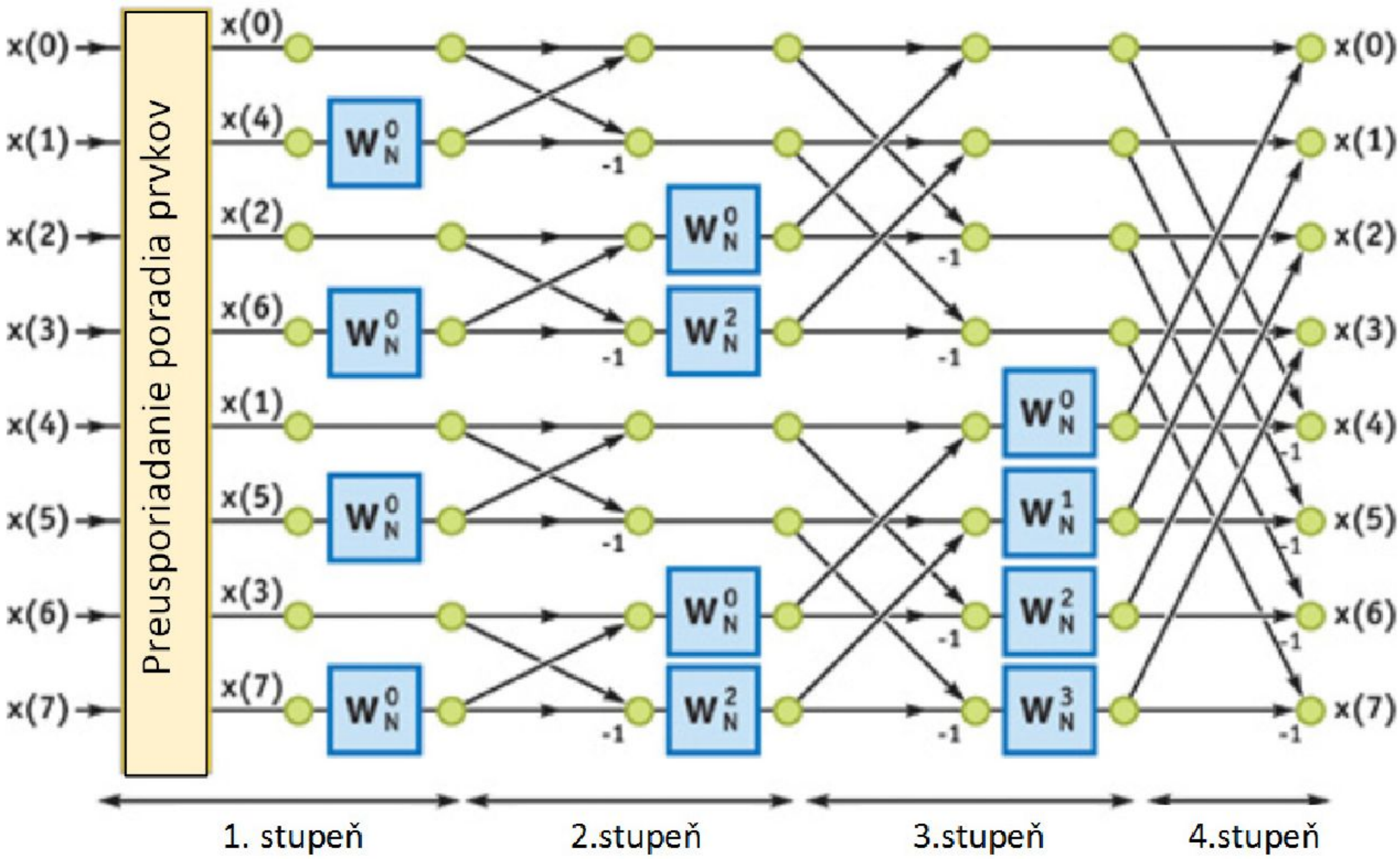
(N/f_{nf}) výpočtov DFTs s dĺžkou f_{nf} .

Celkový počet operácií pre tieto suboperácie potom bude:

$O(N(f_1 + f_2 + ... + f_{nf}))$.

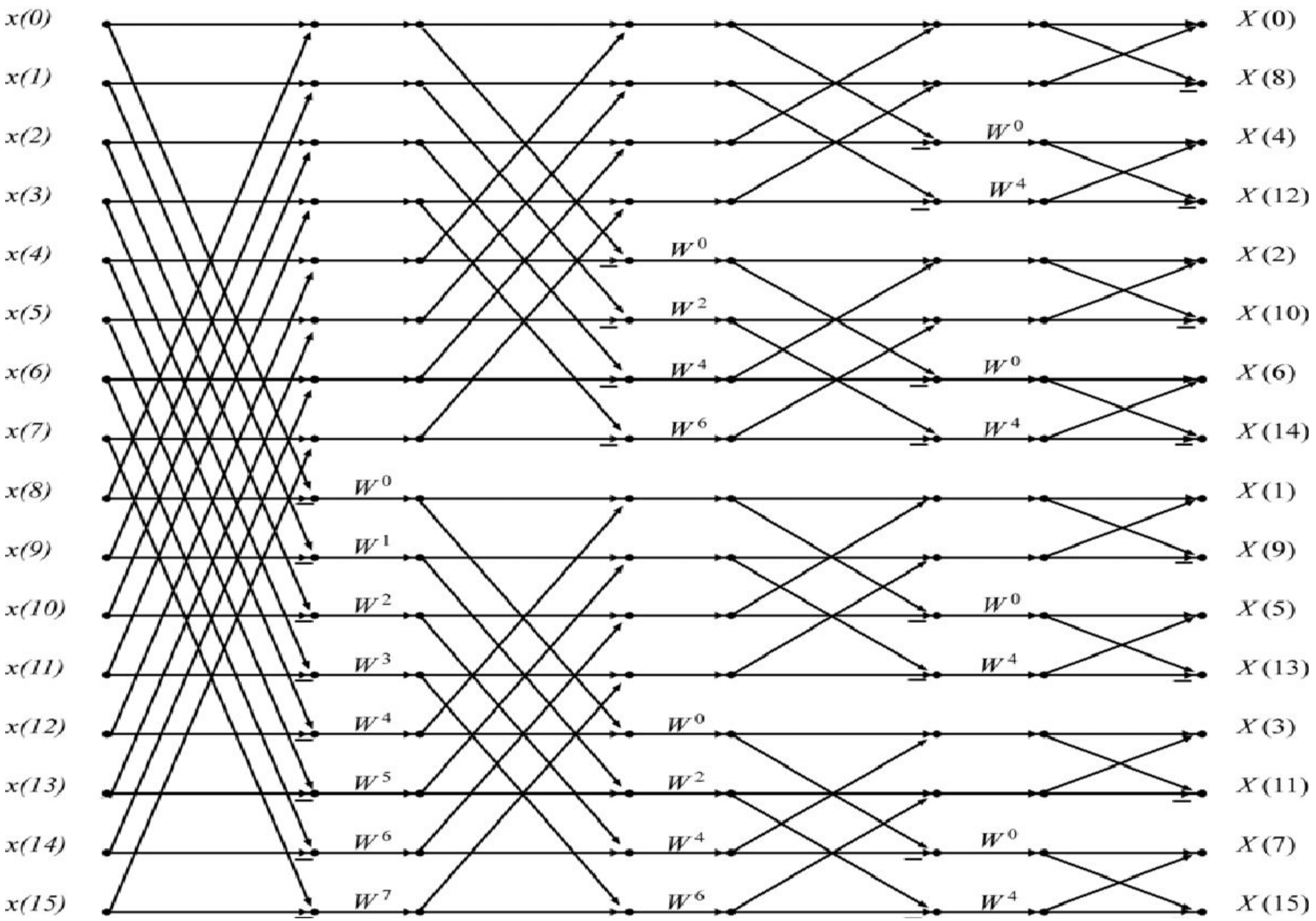
V prípade, že činitele N sú malé celé čísla, tak výpočtová náročnosť bude podstatne menšia ako $O(N^2)$.

Fast Fourier Transform FFT



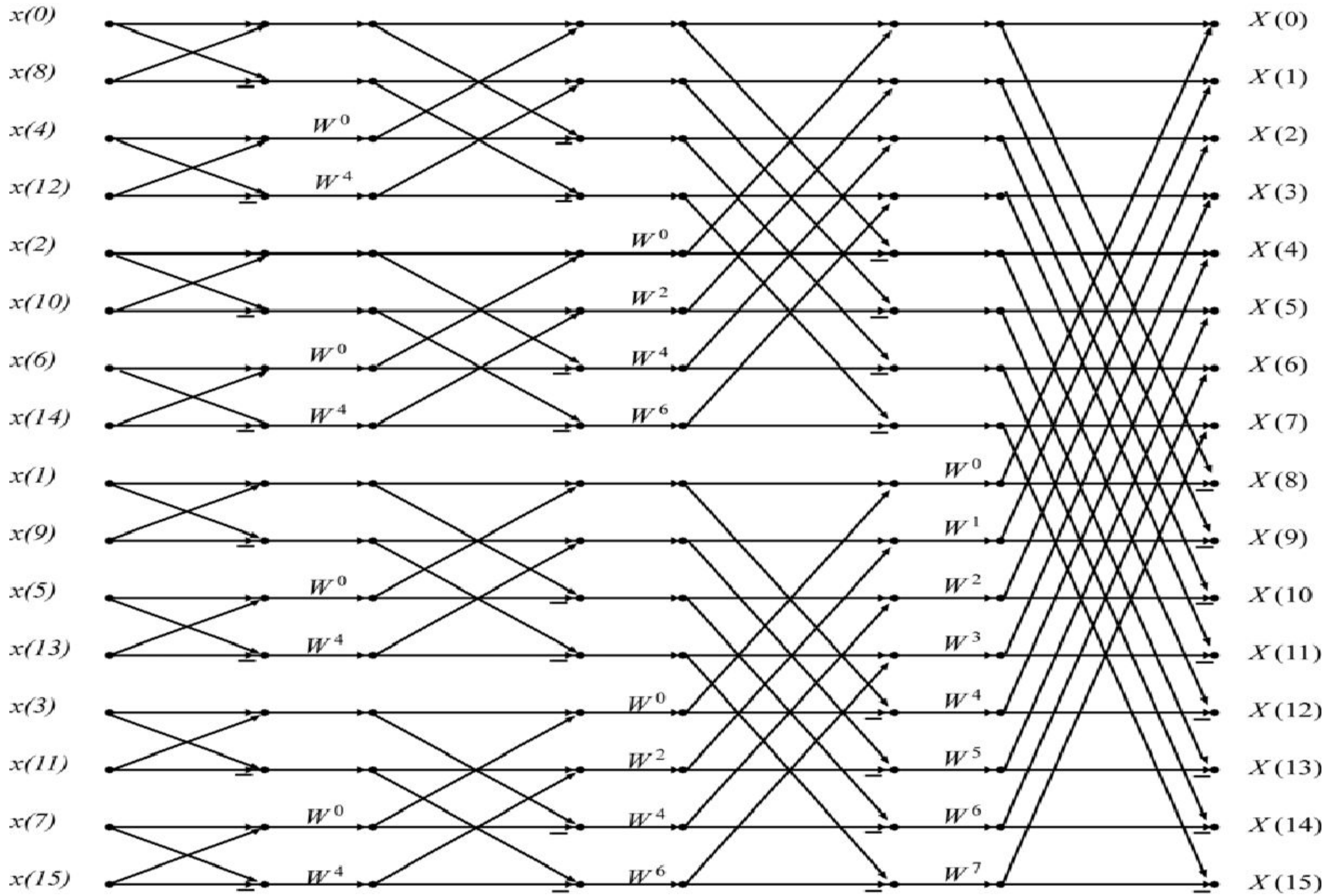
Fast Fourier Transform FFT

decimácia vo frekvencii.



Fast Fourier Transform FFT

decimácia v čase.



FFT - Bitovo invertované poradie prvkov

Poradie prvkov napíšeme ako **bitový reťazec**. Jeho bitovo invertované poradové číslo potom získame tak, že tento bitový reťazec interpretujeme opačnýmsmeromzápisu, **čiže odzadu smerom dopredu**.

V signálovom diagrame FFT je poradie prvkov bitovo invertované, a to na vstupe alebo na výstupe, podľa toho, či sa jedná o FFT typu decimácia v čase alebo decimácia vo frekvencii.

Optimalizácia výpočtu FFT.

Algoritmus výpočtu FFT sa nechá paralelizovať, je vhodný na rýchle hardvérové implementácie na špeciálnych čipoch i na výpočet pomocou grafického procesora GPU.

Na záver si pripomeňme, že algoritmus FFT (Cooley–Tukey) je pre vektory s dĺžkou mocniny dvoch: $N = 2^M$.

Doplnenie nulami (zero padding)

Metódu doplnenia nulami nájdeme v anglickej literatúre pod názvom „zero padding“.

Používajú sa dva základné prístupy:

- doplnenie nulami v spektrálnej oblasti,
- doplnenie nulami v priestorovej oblasti.

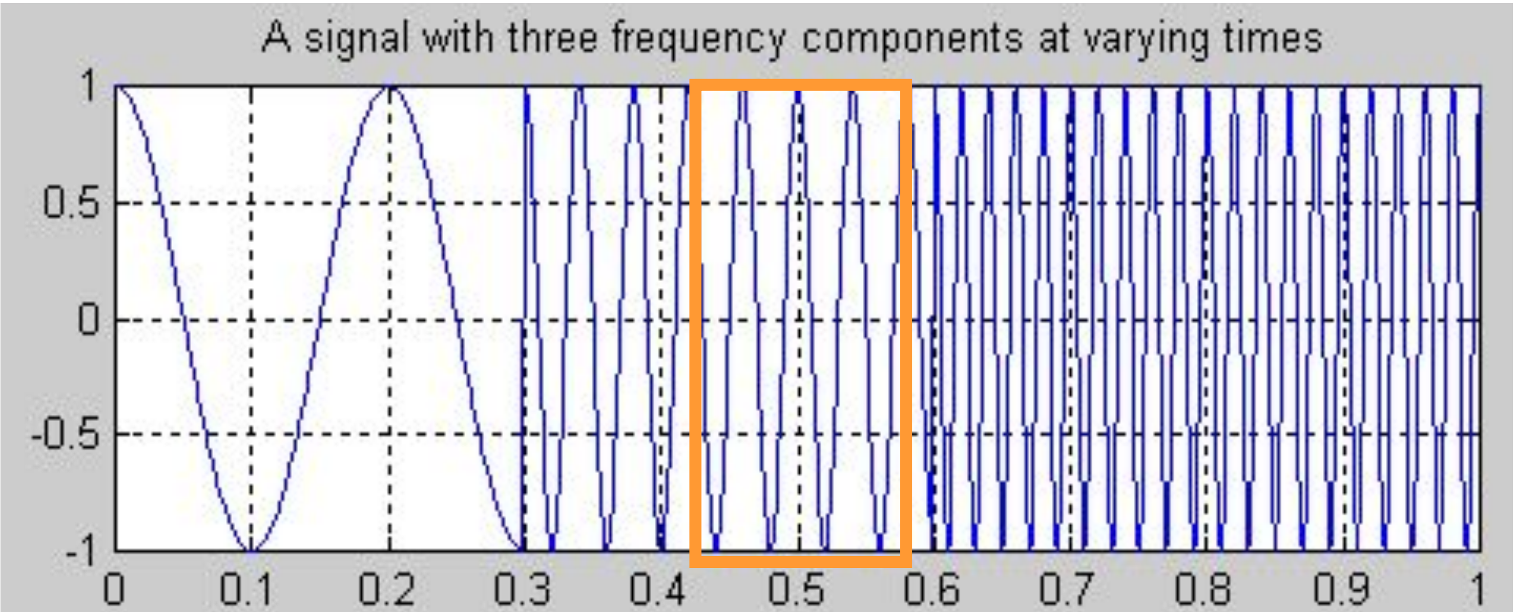
Doplnenie nulami v spektrálnej oblasti môžeme použiť pre zvýšenie rozlíšenia vstupného obrazu. Novozískané hodnoty sú vlastne interpolované existujúce hodnoty obrazu, neobsahujú novú informáciu.

Doplnenie nulami v priestorovej oblasti podobne zvýši rozlíšenie. V prípade, že náš vstupný signál nespĺňa dĺžku mocniny dvoch: $N = 2^M$, tak výhodnemôžeme použiťmetódu doplnenia nulami.

Short-Time Fourier Transform STFT

Short-Time Fourier Transform
Window
STFT

- Segmentovanie signálu do úzkych časových intervalov (to znamená, že je dostatočne úzky na to, aby sa považoval za stacionárny - "kvázi-stacionárny").
- Výpočet Fourierovej transformácie pre každý segment.

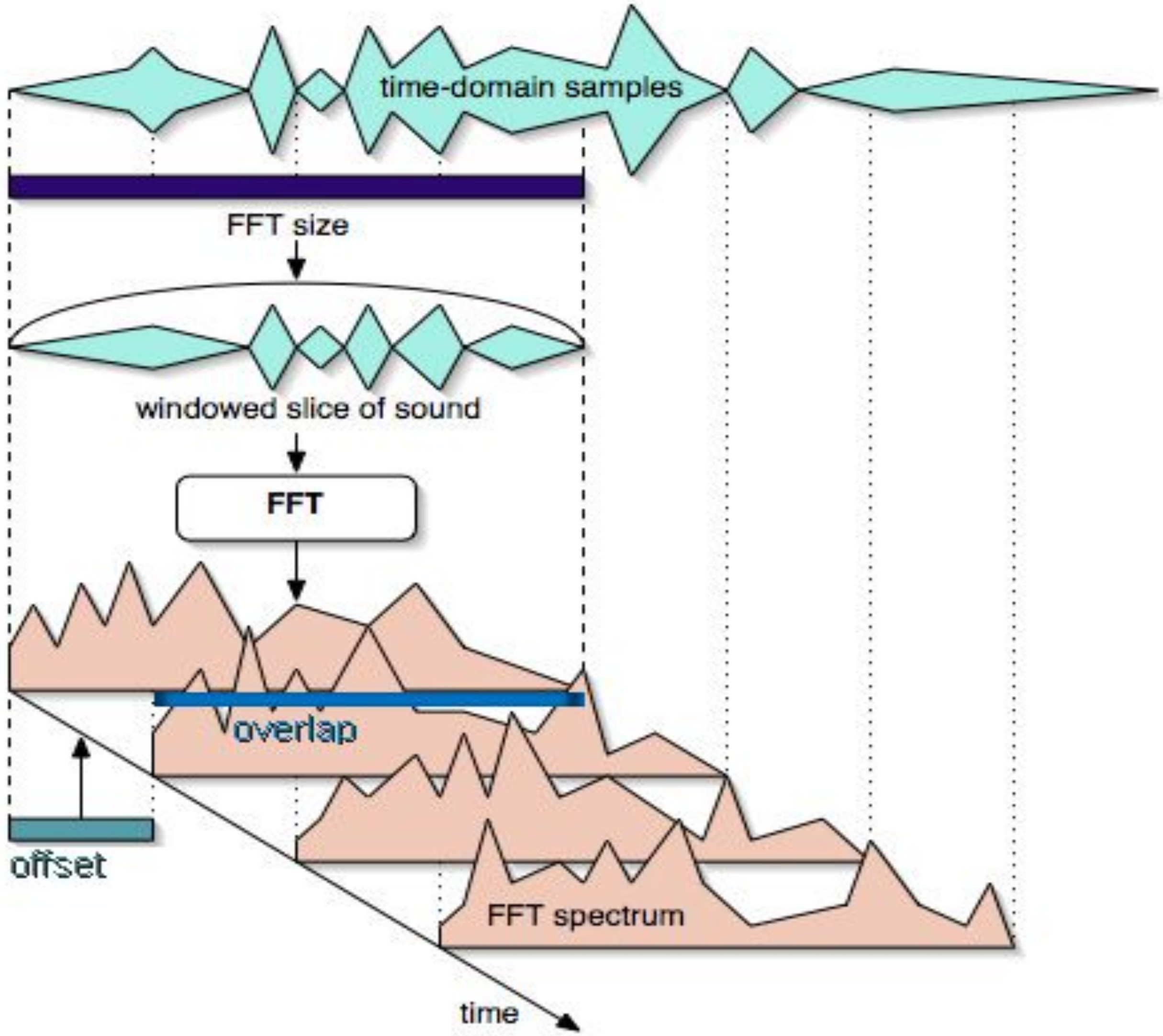


Short-Time Fourier Transform (STFT)

Offset

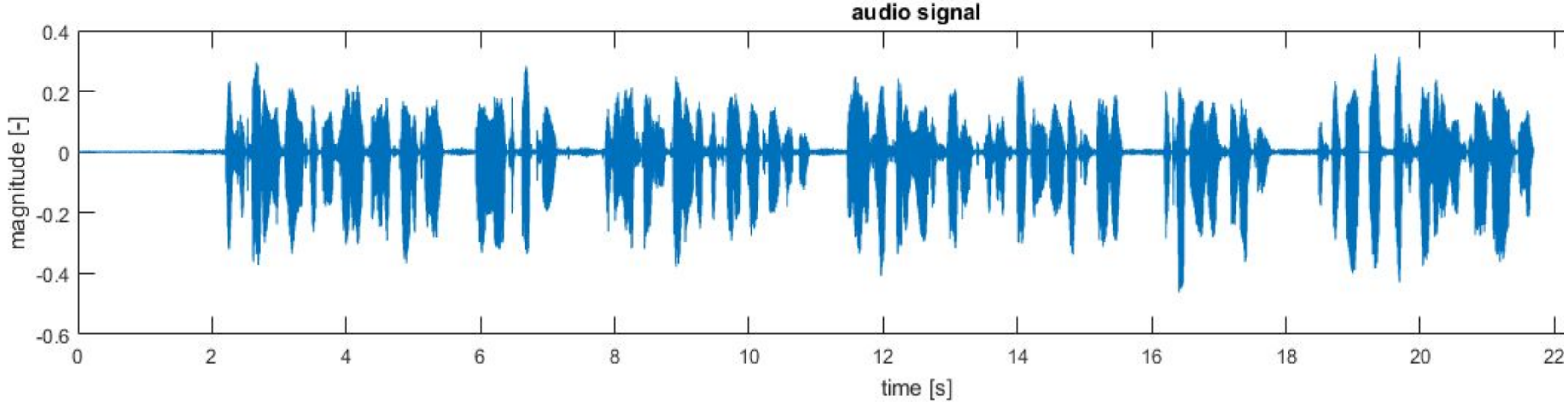
or

Overlap

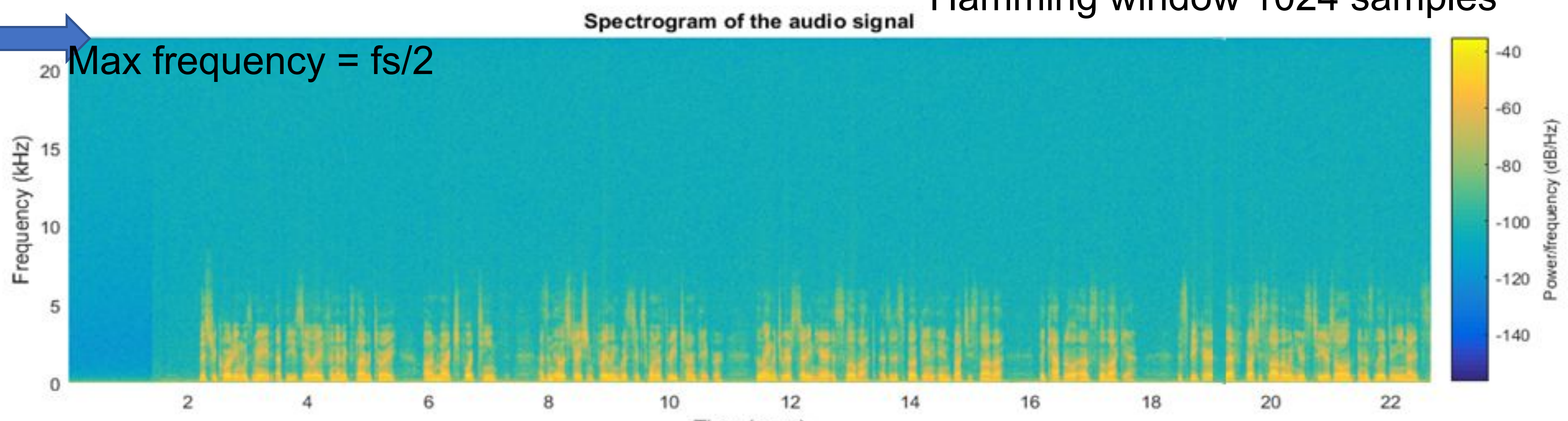


Short Time Fourier Transform Visualisation - Spectrogram (how to scale axes?)

Example of a audio signal
Sample frequency fs = 44100 Hz



Hamming window 1024 samples



Max frequency = fs/2

Demo

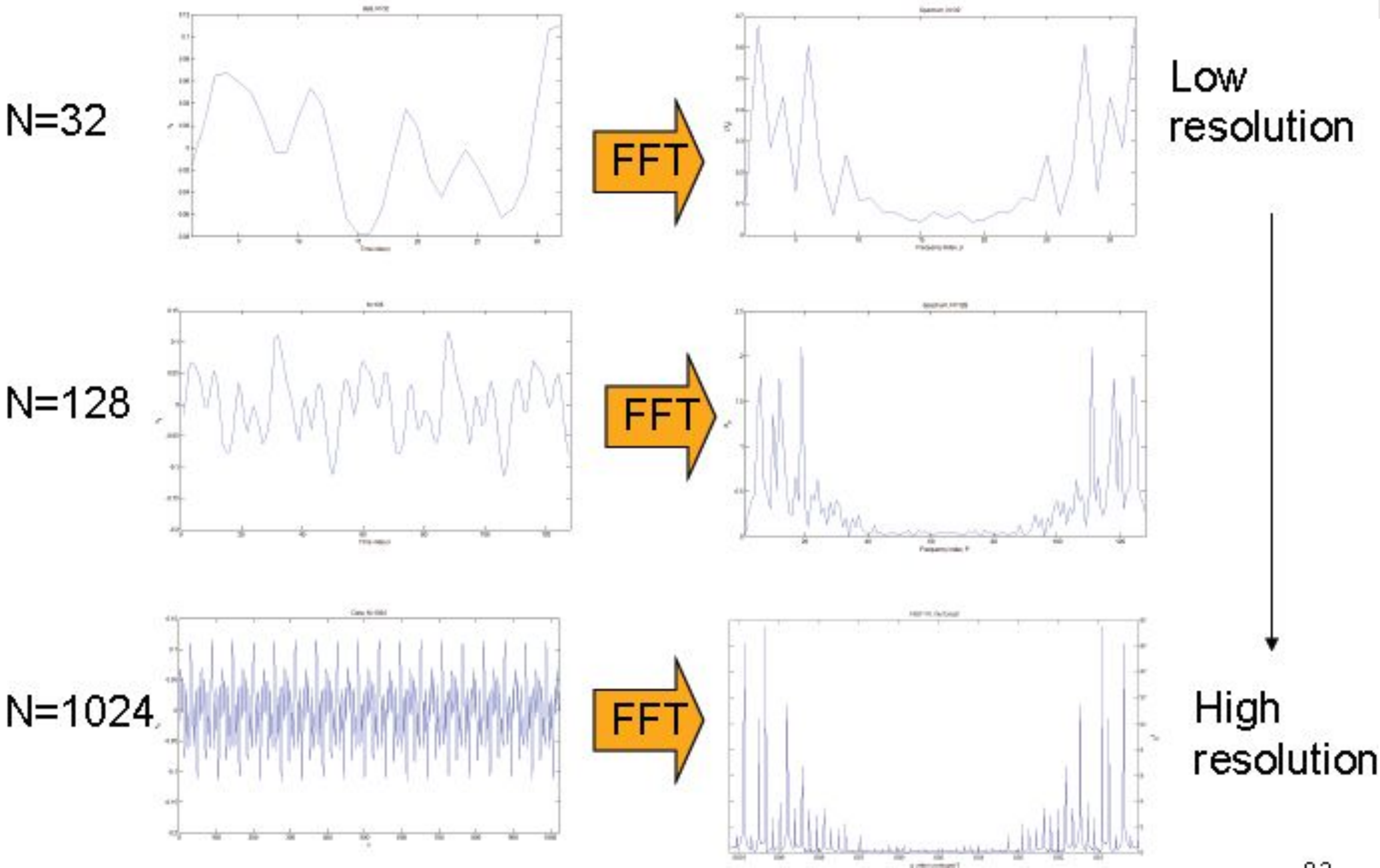
Demo spectrogram:

https://academo.org/demos/spectrum-analyzer/

Interaktivny nastroj:
http://www.falstad.com/fourier/

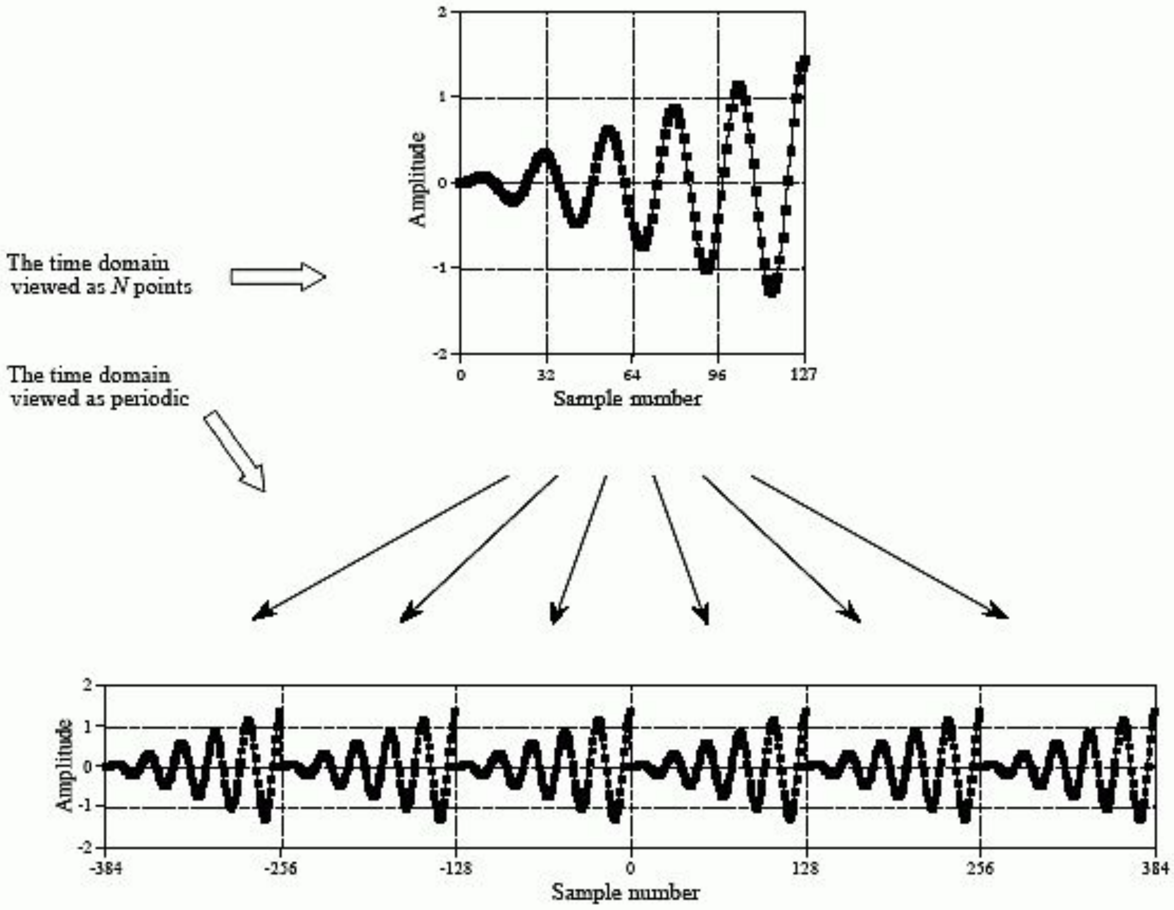
VIDEO:
https://www.youtube.com/watch?v=nmgFG7PUHfo

The Effect of data length N (window size)



Spectral Leakage
Gipps phenomenon

Periodicity of the DFT



Periodicity of the DFT's time domain signal. The time domain can be viewed as N samples in length, shown in the upper figure, or as an infinitely long periodic signal, shown in the lower figure.

Periodicity of the DFT

$$\begin{aligned} X[k+N] &= \sum_{n=0}^{N-1} x[n]\, e^{-j2\pi\frac{(k+N)n}{N}} \\ &= \left(\sum_{n=0}^{N-1} x[n]\, e^{-j2\pi\frac{kn}{N}}\right) e^{-j2\pi n} \\ &= X[k]\, e^{-j2\pi n} = X[k] \qquad \Rightarrow \end{aligned}$$

The DFT spectrum X[k] is periodic with period N

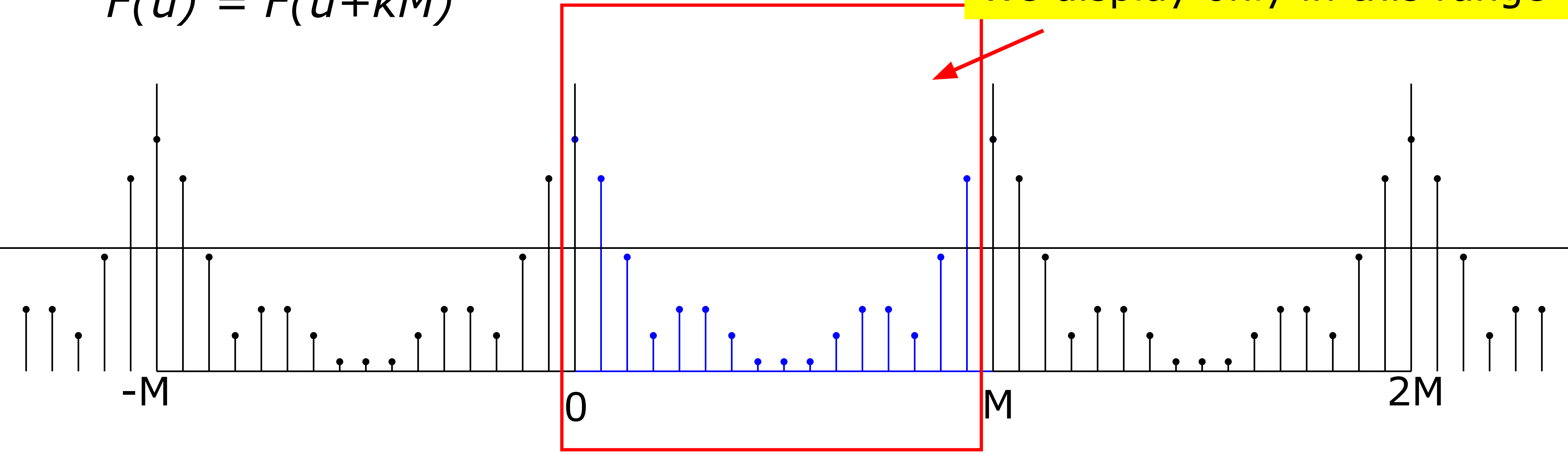
math. recap: $e^{-j2\pi} = 1 + 0j$

Periodicity of the DFT

From DFT: $$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x) e^{-j2\pi ux/M}$$

F(u) = F(u+kM)

We display only in this range

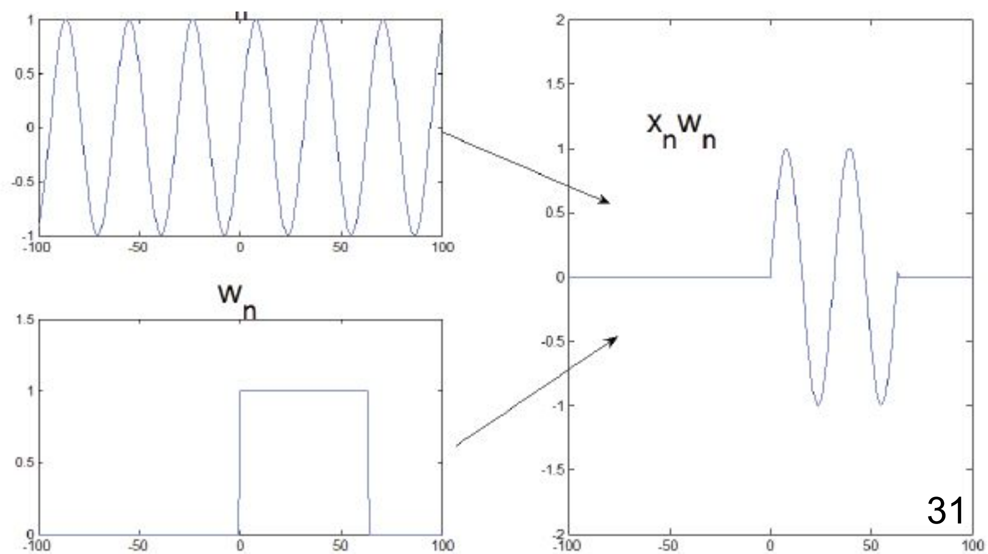


DFT repeats itself every M points (Period = M)

Applying rectangular window

A finite sequence x[n] that is M samples long can be obtained from a longer sequence y[n] by applying a rectangular window of length M:

$$x[n] = \begin{cases} 0\,, & n < 0, \\ y[n]\,, & 0 \le n \le (M-1)\,, \\ 0\,, & n \ge M. \end{cases}$$



Spectral Leakage in STFT

An assumption in the DFT algorithm:

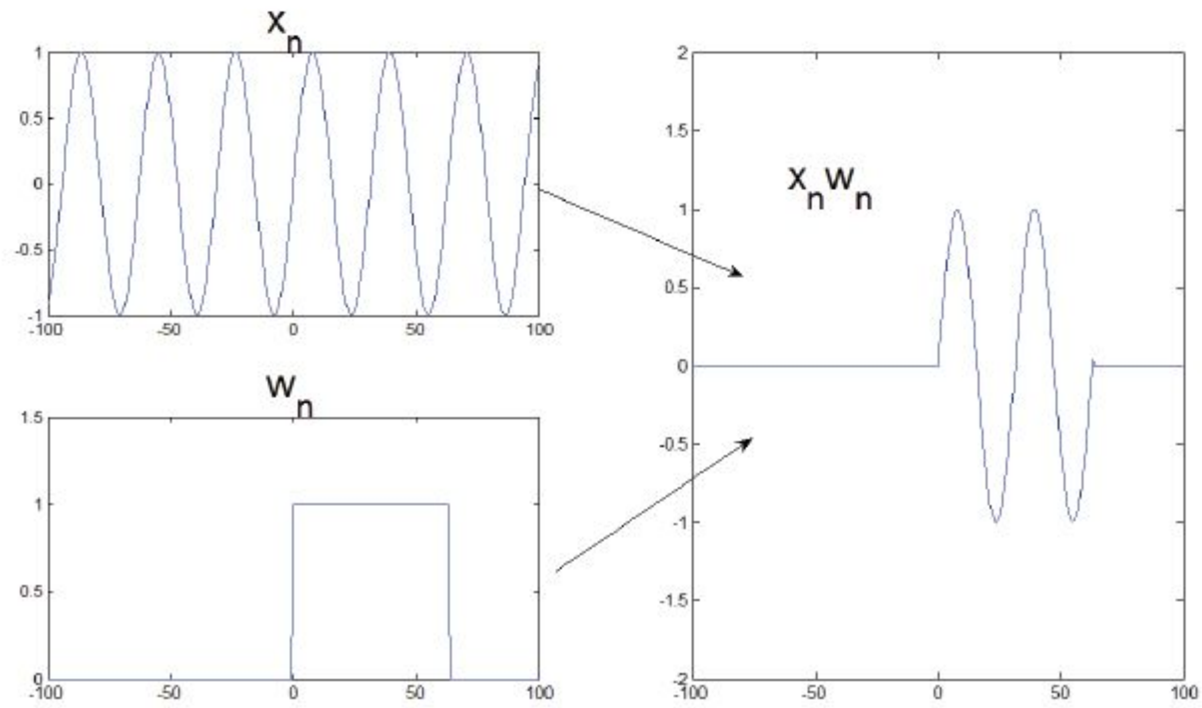
the time record (rectangular window size) is exactly synchronized with the signal

-> hence, the window should cover exactly the part of signal which has a integral number of periodical cycles.

If the time record (rectangular window size) corresponds to a non-integral number of cycles, this assumption is violated and spectral leakage occurs.

Rectangle window in STFT

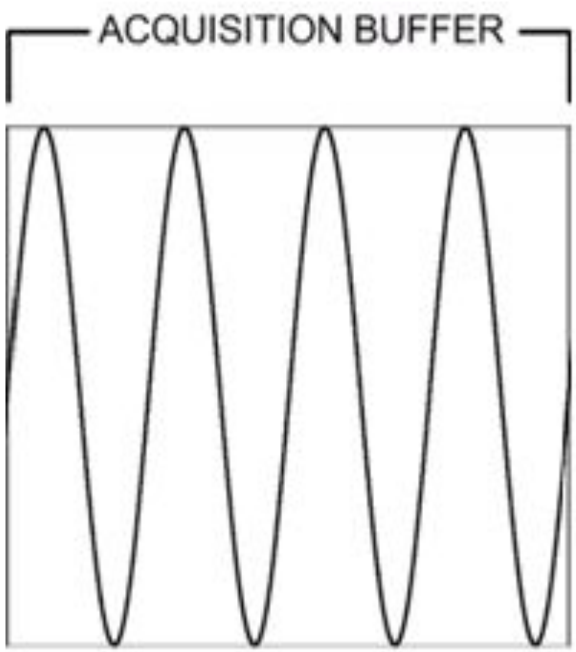
Example: window size is exactly synchronized with the signal

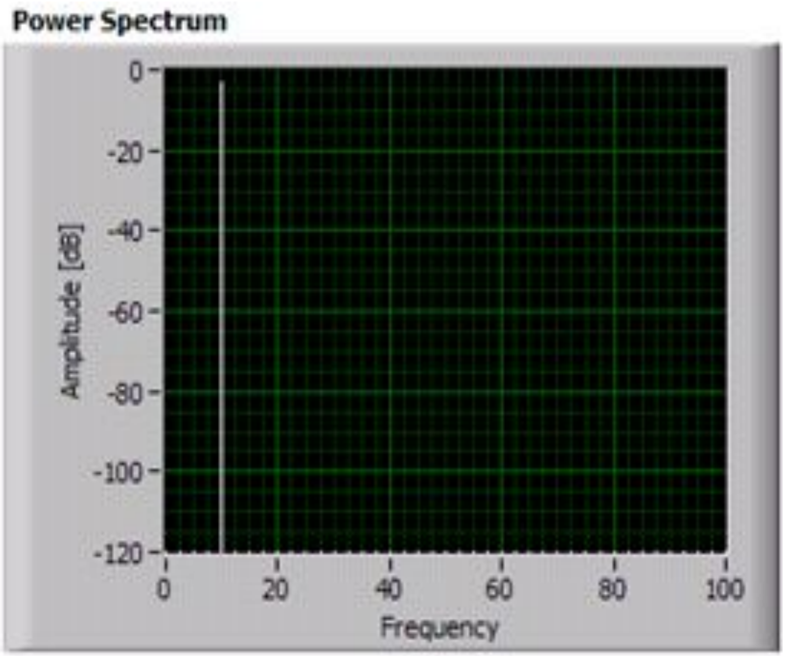


Integer Number of Periods in Acquisition Time Interval

Acquisition buffer – rectangular window

The size is an integer number of periods –> spectrum after STFT is exact

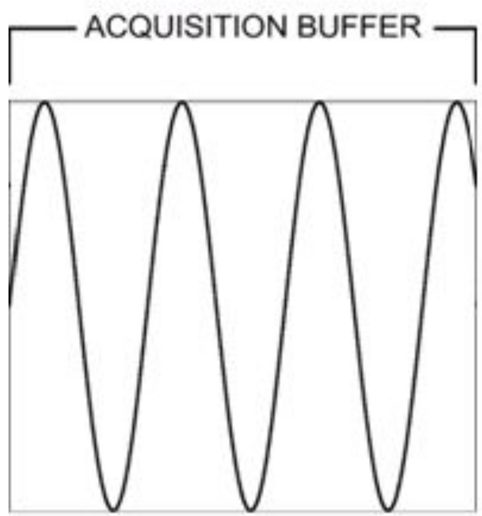


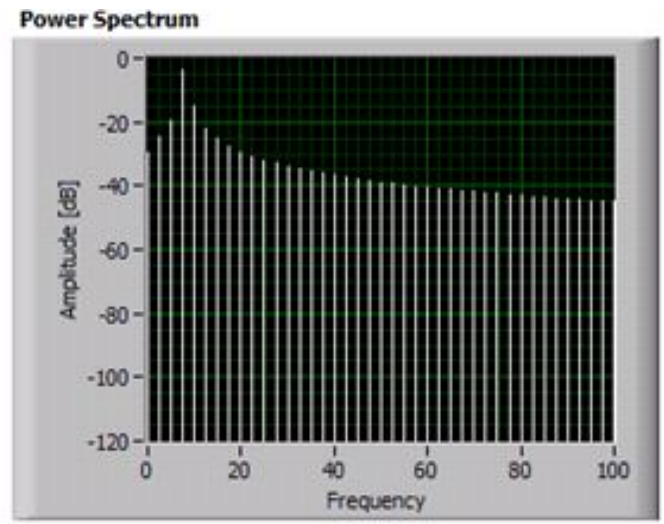


Non-Integer Number of Periods in Acquisition Time Interval- Spectral Leakage

Acquisition buffer – rectangular window

The size is an non-integer number of periods





!Log scale! (Db is logarithmic unit)

$$Amplitude_{dB} = 10 \cdot \log_{10}(P_1 / P_0)$$

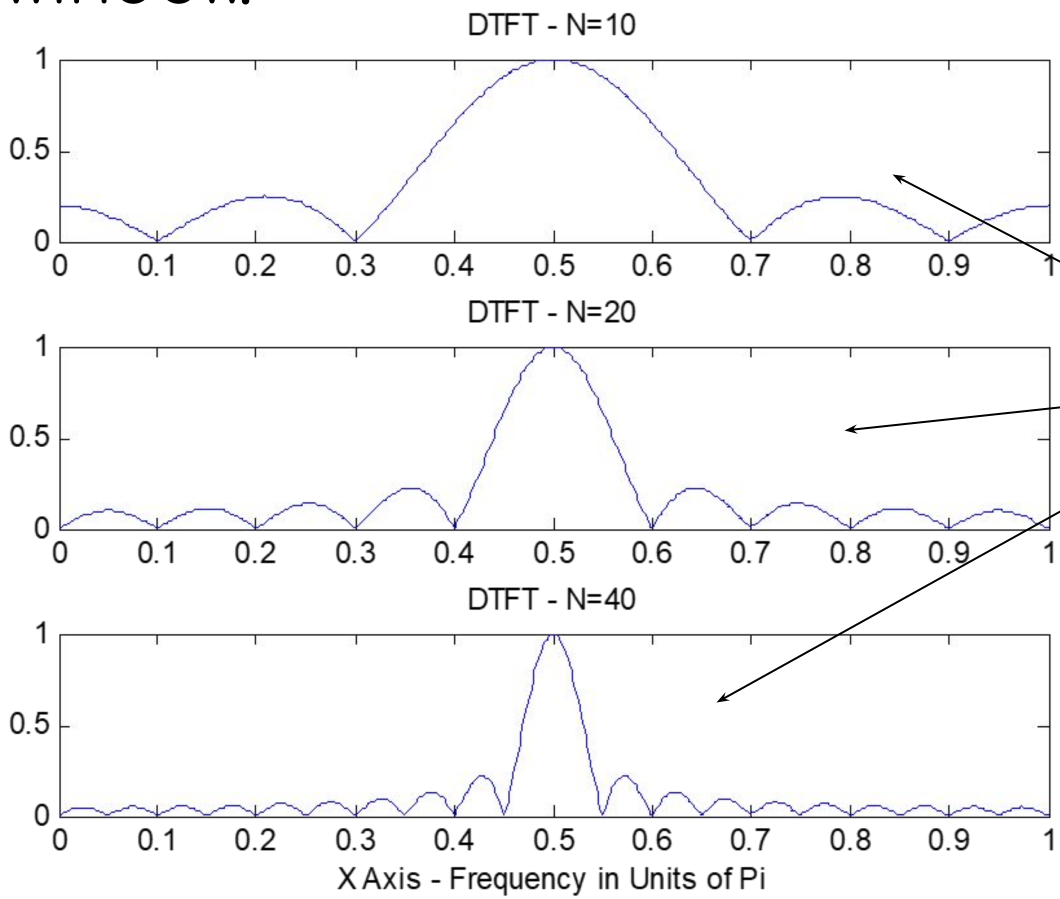
I

n the spectrum are high side lobes

This phenomena is called Spectral Leakage.

Spectral Leakage
Gibbs Phenomenon

Windowing using non synchronized rectangular window.



the rectangular window (abrupt truncation of the signal) results in side lobes (Gibbs phenomenon) in spectrum

Windowing

Windowing

Solution of the problem of spectral leakage :

Use other than rectangular type of window ->„smooth“ window

Frequently used windows:

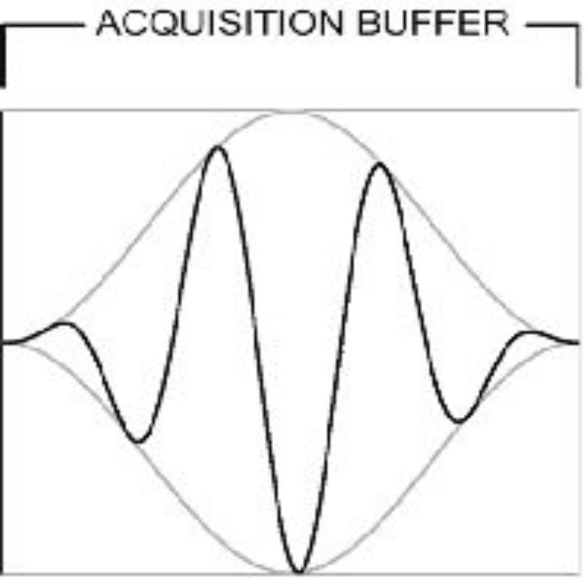
Hamming, Hanning, Blackman, Barlett, Gaussian window and more…

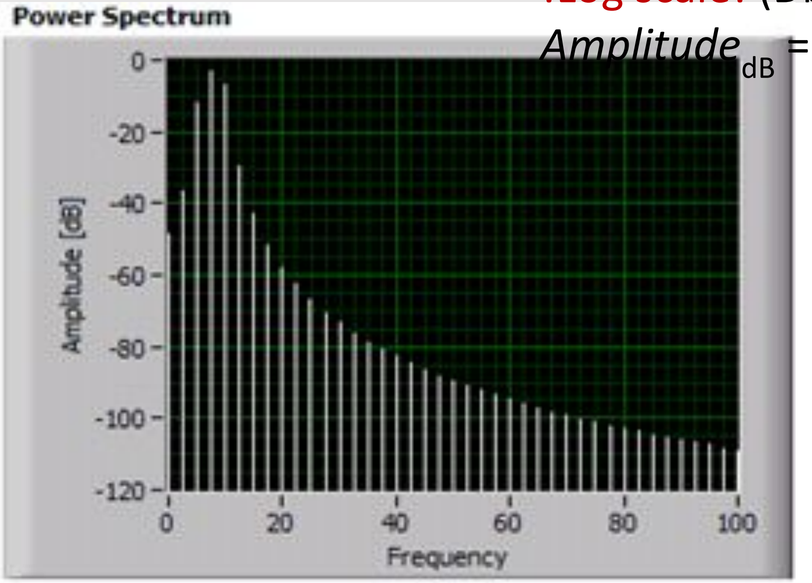
This type of windows reduce (! but not completely remove!) the side-lobes associated with the rectangular window

Windowing
Using „smooth“ window functions

!Log scale! (Db is logarithmic unit)

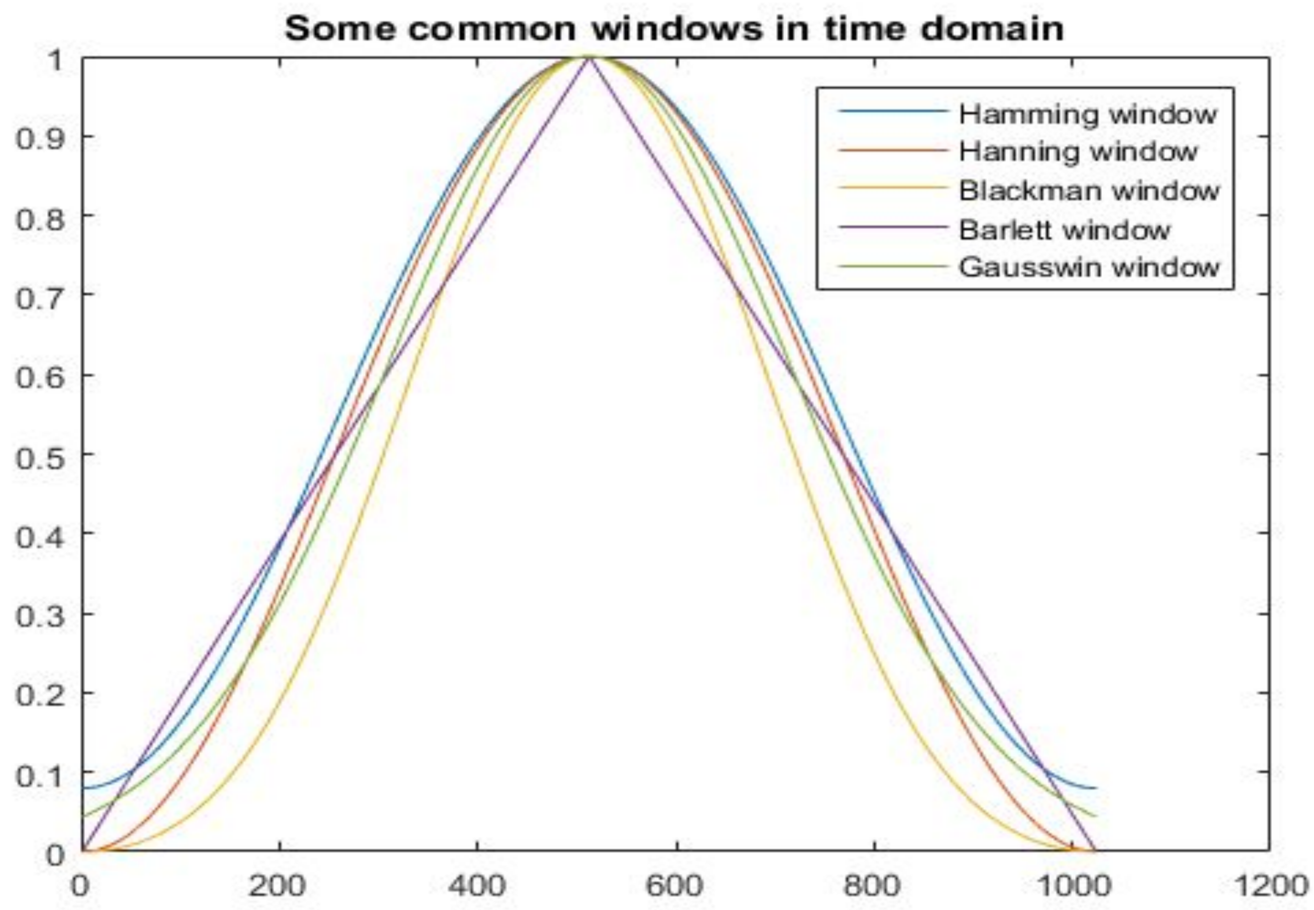
$Amplitude_{dB} = 10 \cdot \log_{10}(P_1 / P_0)$



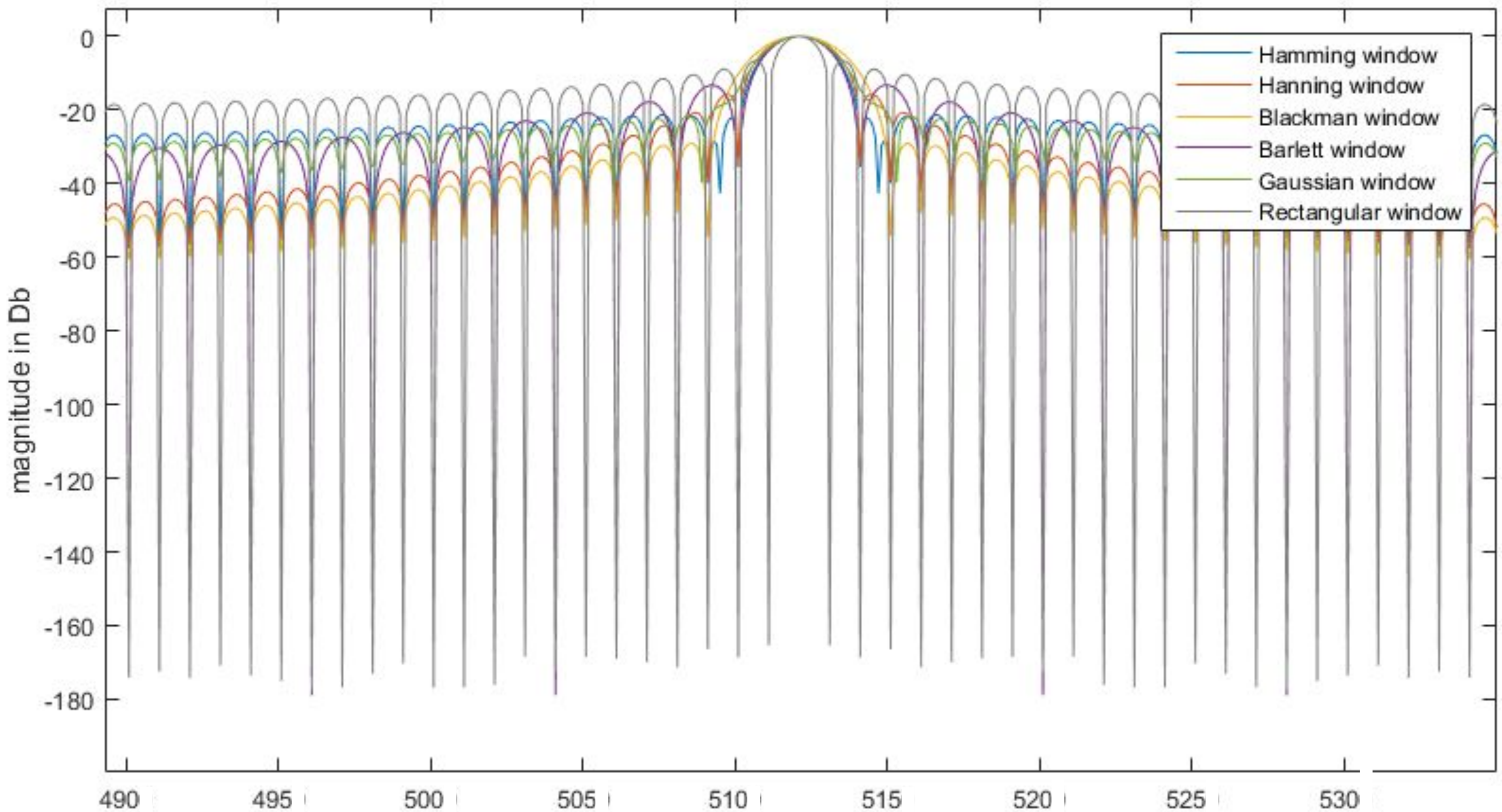


- Signal in acquisition buffer will be multiplied with a „smooth window function“ per elements before computing the FFT.
- using this „smooth“ window will suppress the spectral leakage
- This technique is also referred to as 'applying a window' or simply 'windowing'.

Windowing
selected windows in the time domain



Windowing
selected windows in the spectral domain



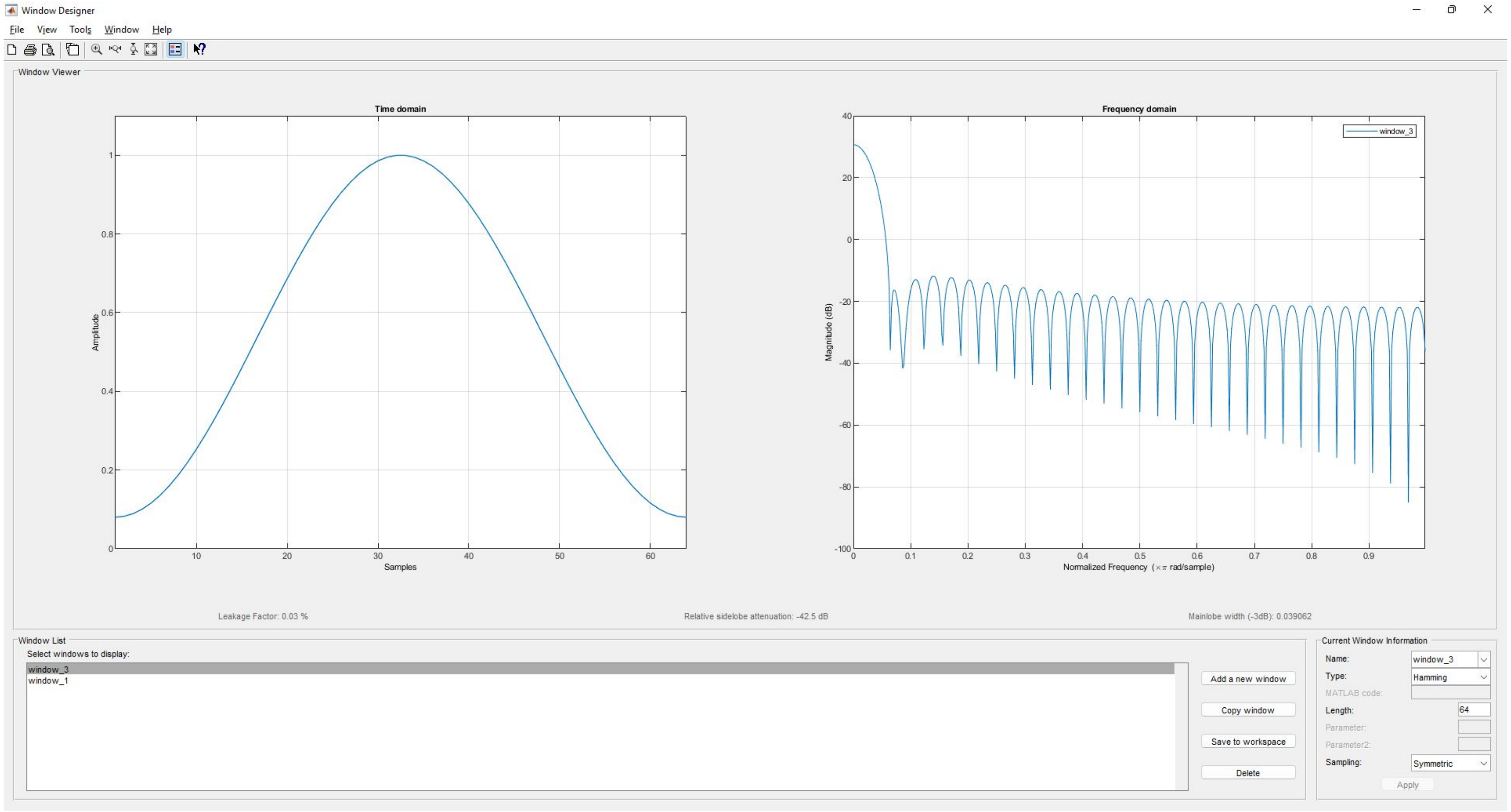
!Log scale! (Db is logarithmic unit)

$Amplitude_{dB} = 10 \cdot \log_{10}(P_1 / P_0)$

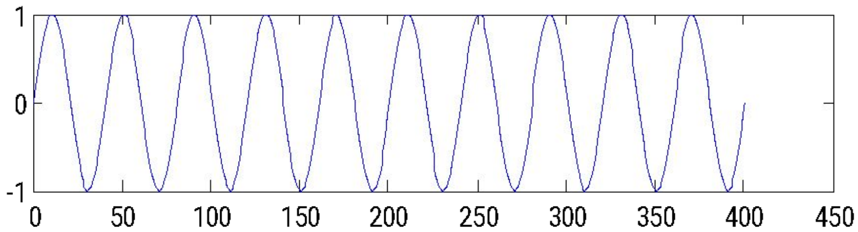
Window Designer in Matlab

windowDesigner

- Leakage factor - ratio of power in the sidelobes to the total window power
- Relative sidelobe attenuation - difference in height from the main lobe peak to the highest sidelobe peak
- Main lobe width (–3dB) - width of the main lobe at 3 dB below the main lobe peak

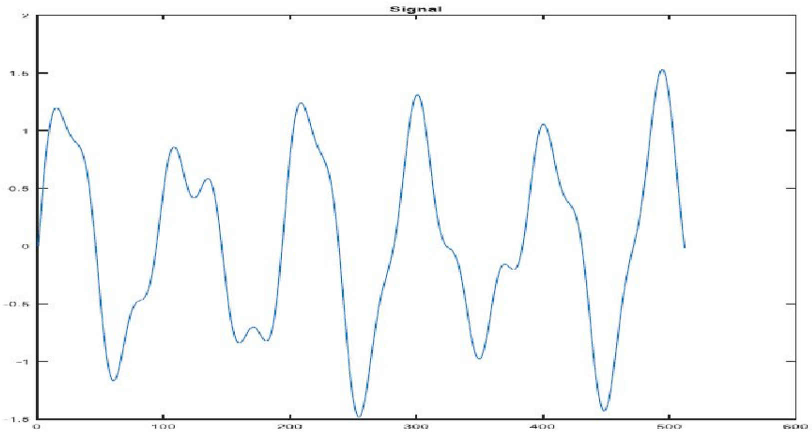


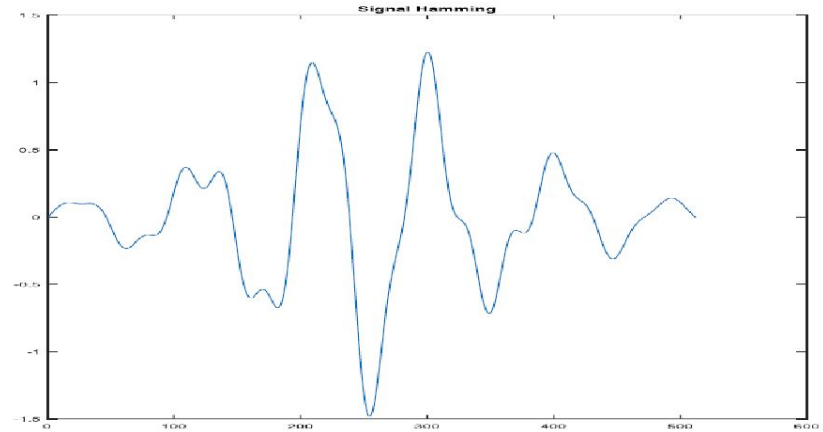
Windowing
Applying a window function



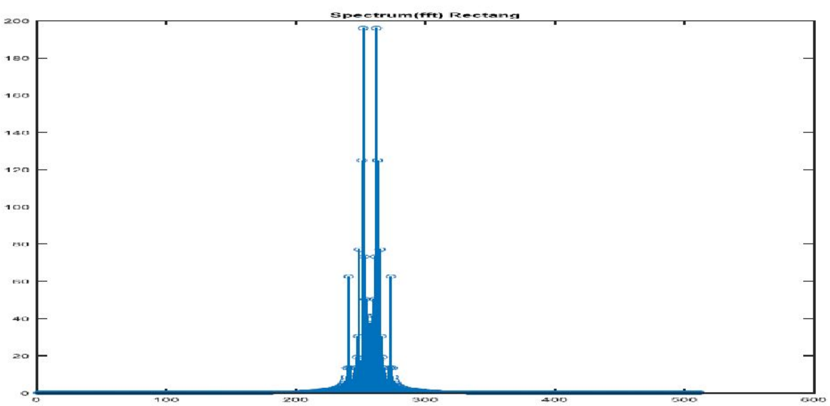
The window function (Hamming) tapers the abrupt truncation of the signal but preserves its frequency characteristics

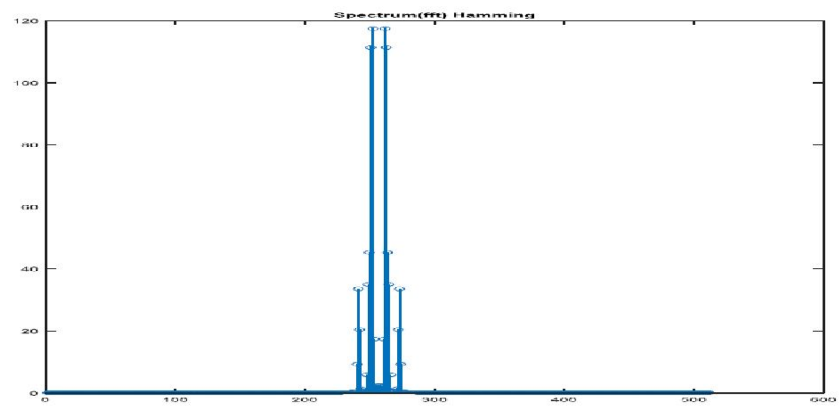
Windowing - Examples





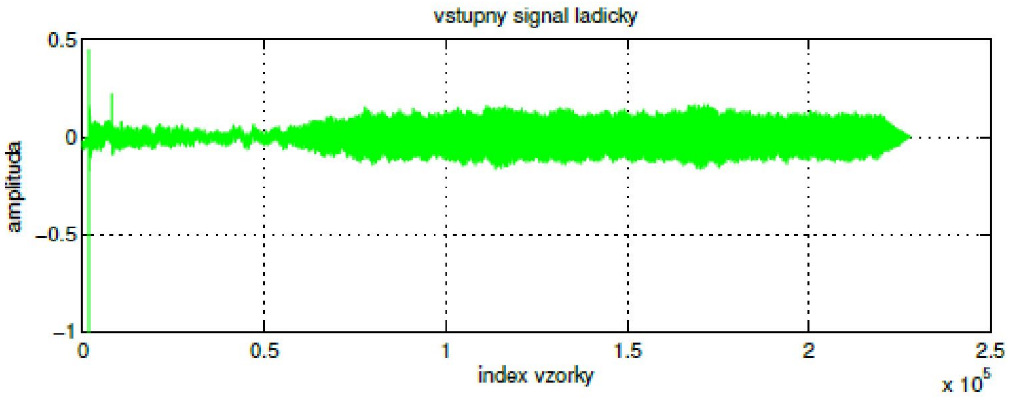
Pôvodný signál a signál vynásobený pravouhlým(vľavo) a Hammingovým okienkom(vpravo)



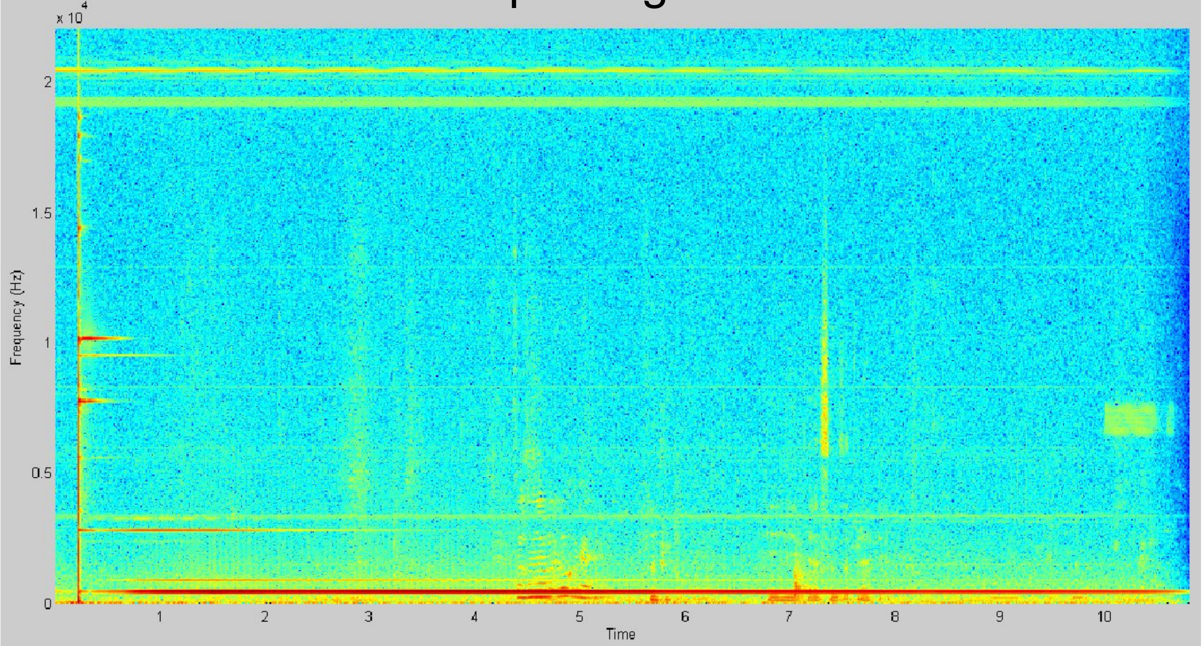


FFT vypočítané zo signálu násobeného pravouhlým okienkom (vľavo) a Hammingovým okienkom (vpravo)

Príklad - ladička



Spectrogram



Príklad – ladička
fft size= 1024

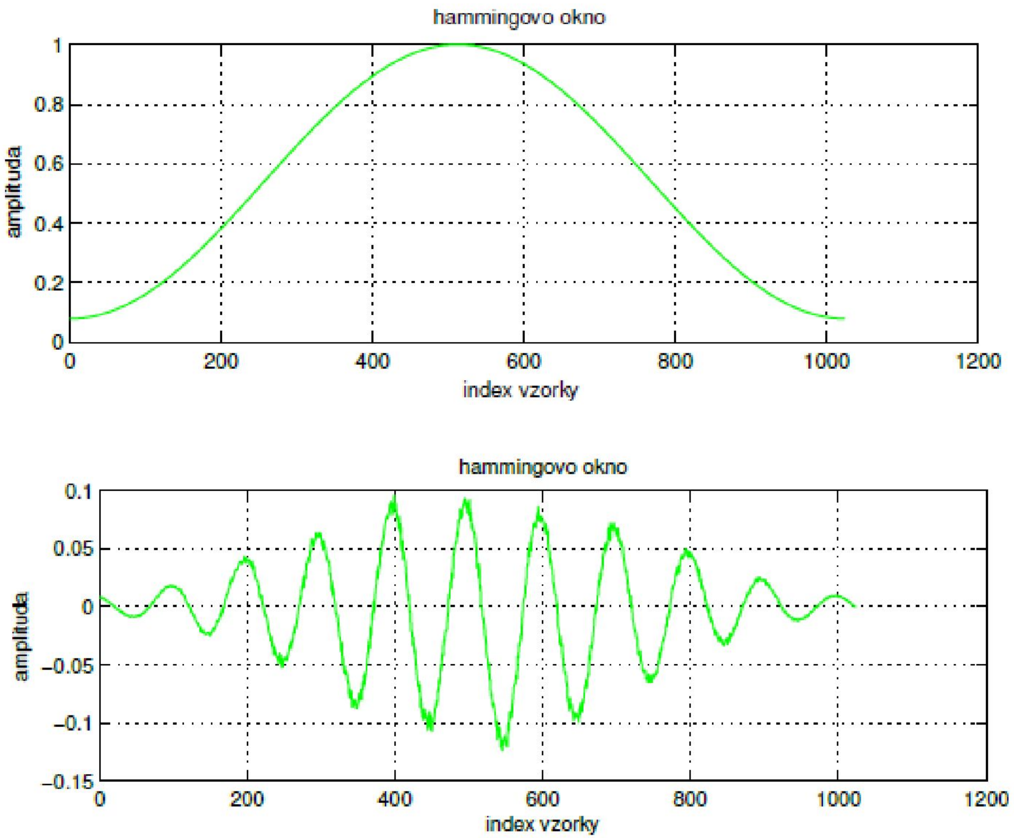


Figure 3: Periodický signál so šumom v spektre vysokých frekvencií, na ktorý bolo aplikované hammingovo okno.

Príklad - ladička

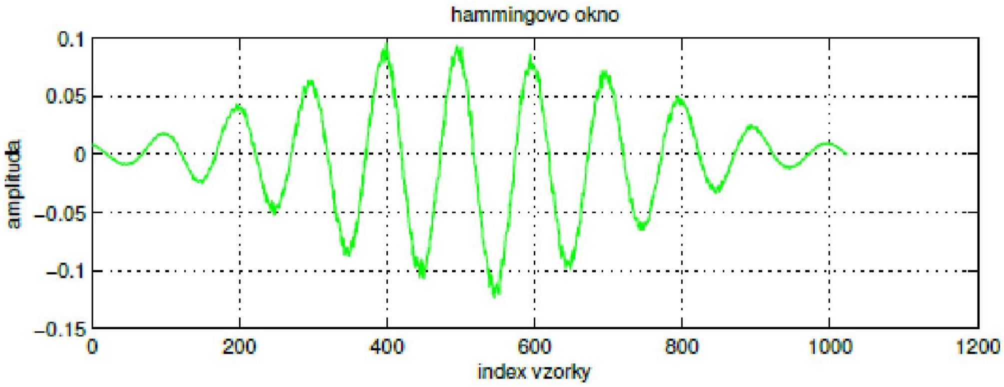
Fs = 44100 Hz

Veľkosť okna: 1024

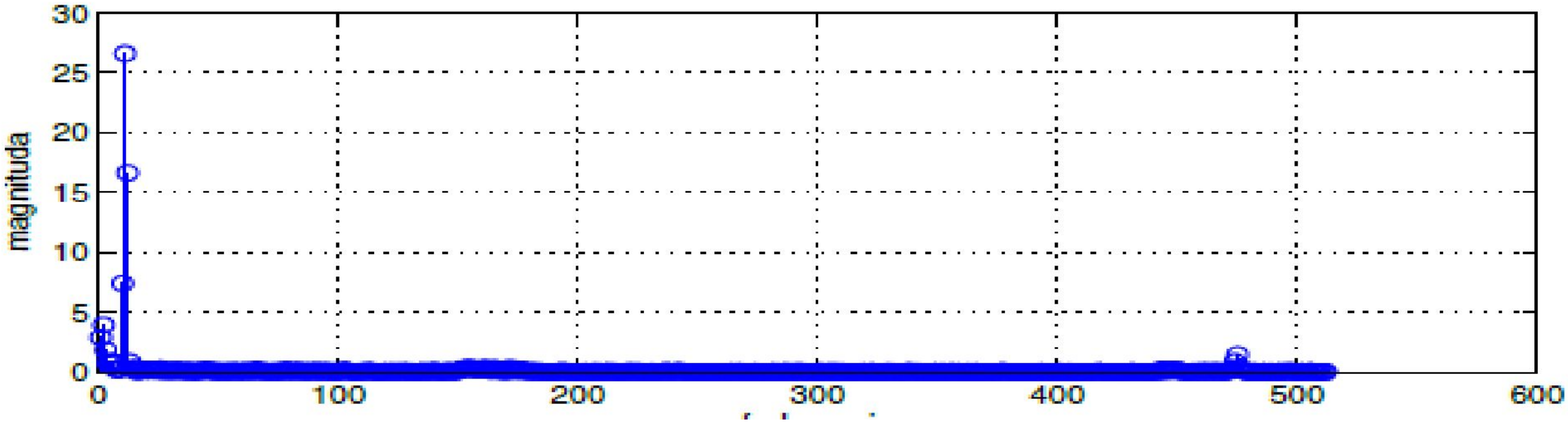
440 Hz je dominantne v

10. komponente (presne 10,2)

Signál po vynásobení Hammingovým oknom



Magnitude spectrum



Zero padding

Zero padding

- In the time domain
- In the frequency domain

Why Zero-padding?

Main reason for zero-padding:

to reach power-of-two input samples number by FFT

Other reason for zero padding:

- zero padding in frequency domain increases sampling rate in time domain
- zero padding the data in the time domain increases the frequency resolution after DFT and thus improve the estimate.

-Frequencies in the discrete Fourier transform (DFT) are spaced at intervals of Fs/N where Fs is the sampling frequency and N is the length of the input time series (with or without zero padding).

! It works only if sampling theorem satisfied!

Zero padding-> interpolated data

DFT increases the frequency resolution by INTERPOLATION.

We can not get more information, we can only get more data

-> interpolated data

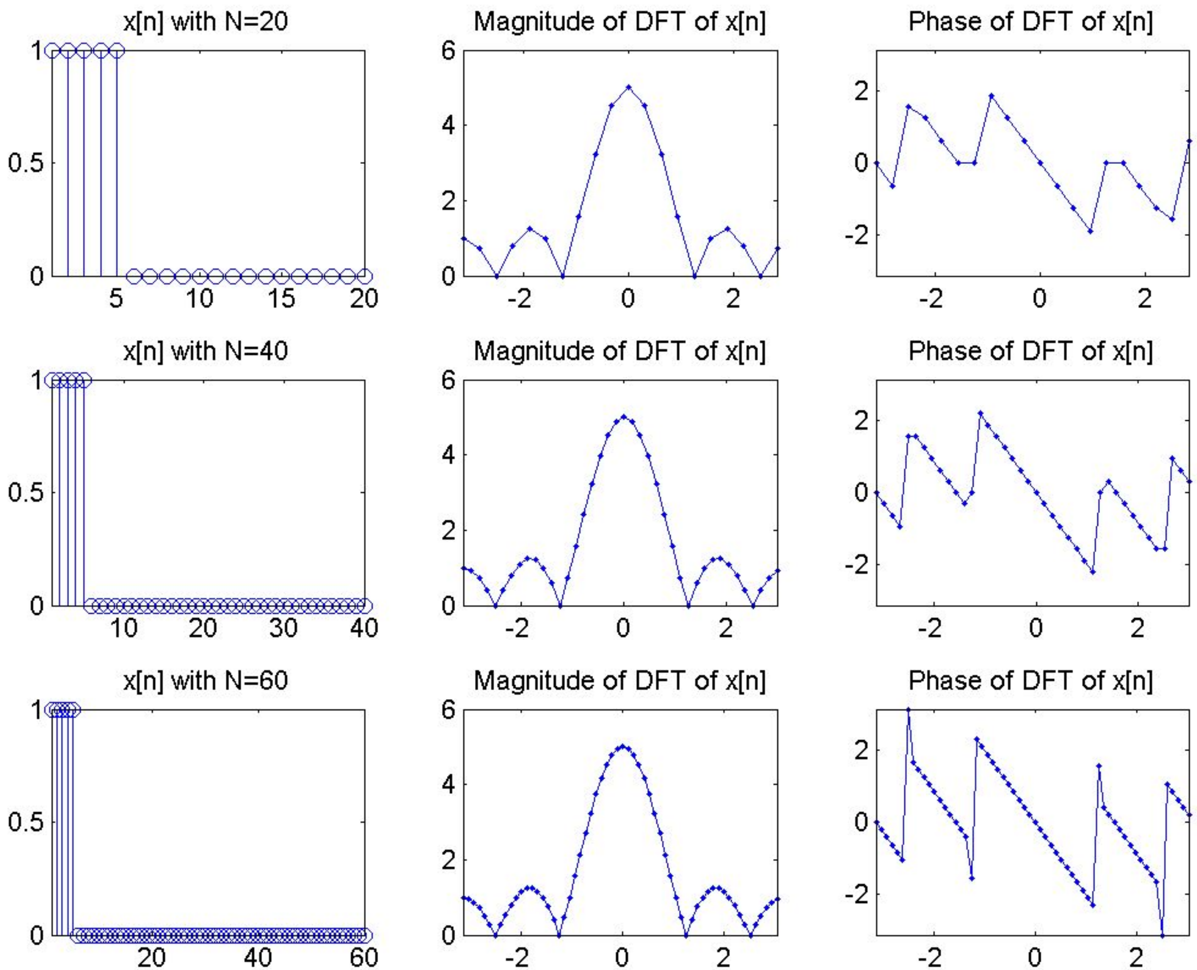
Capability to distinguish two closely-spaced frequencies :

NOT IMPROVED by zero-padding

Frequency inter-sampling spacing:

INCREASED by zero padding interpolation

Zero padding Example



Zero padding

Apply zero-padding AFTER windowing!