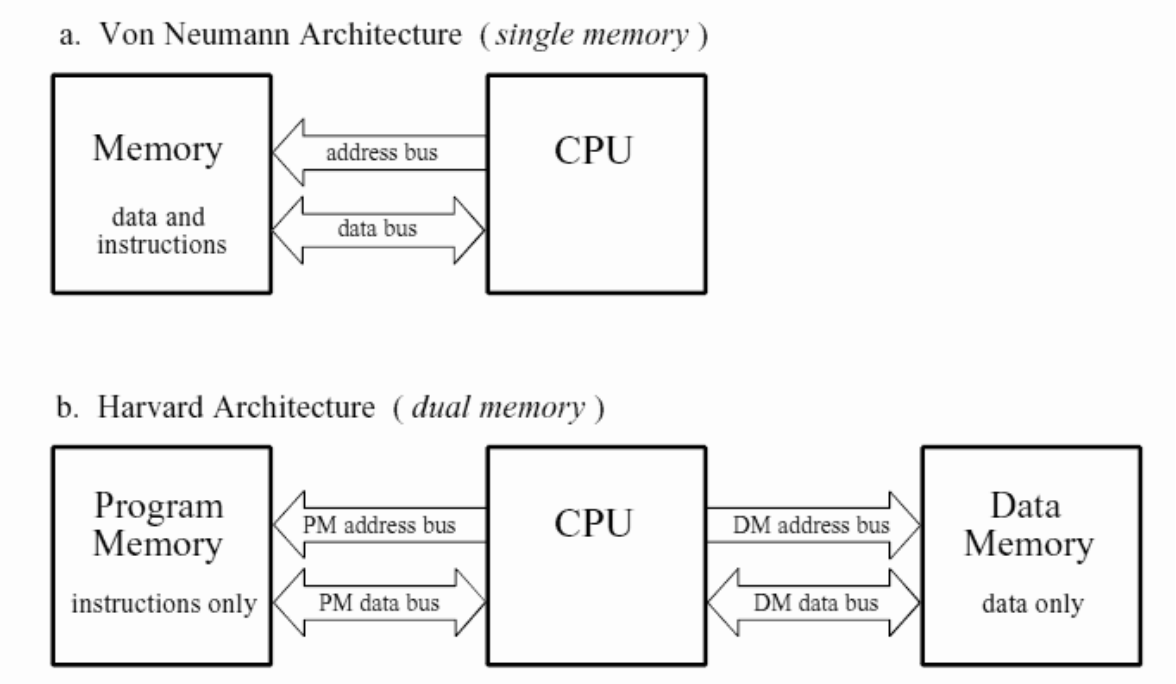
DSP processzorok:



c. Super Harvard Architecture (*dual memory, instruction cache, I/O controller*)

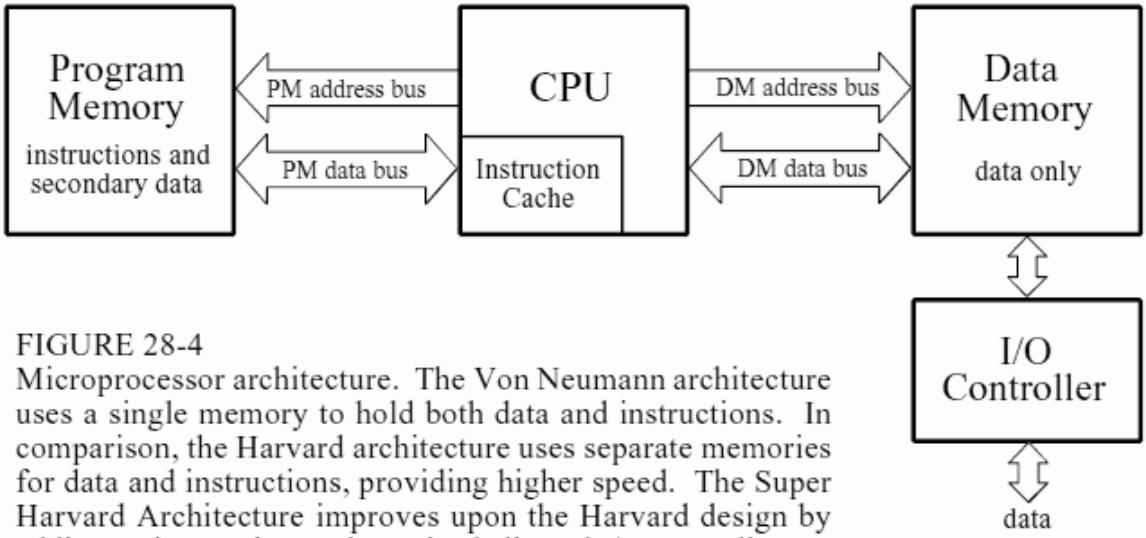
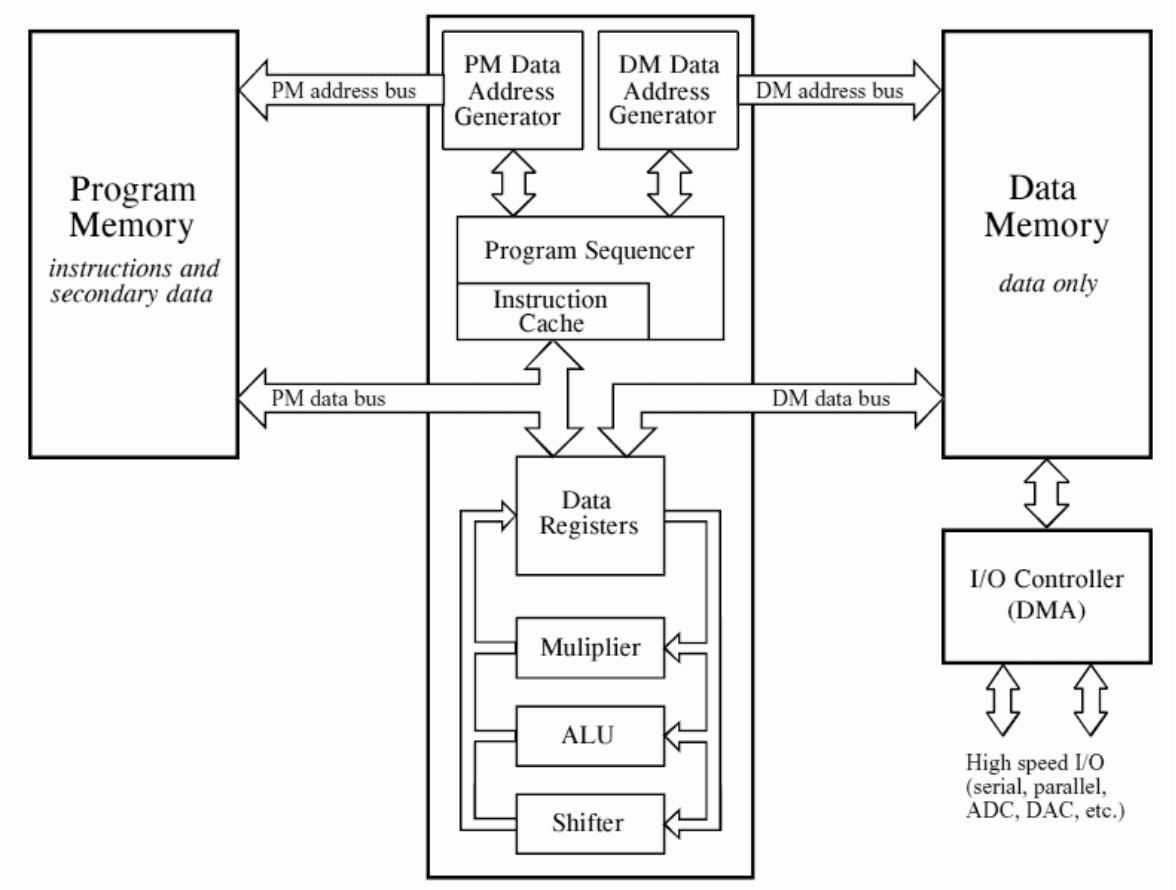
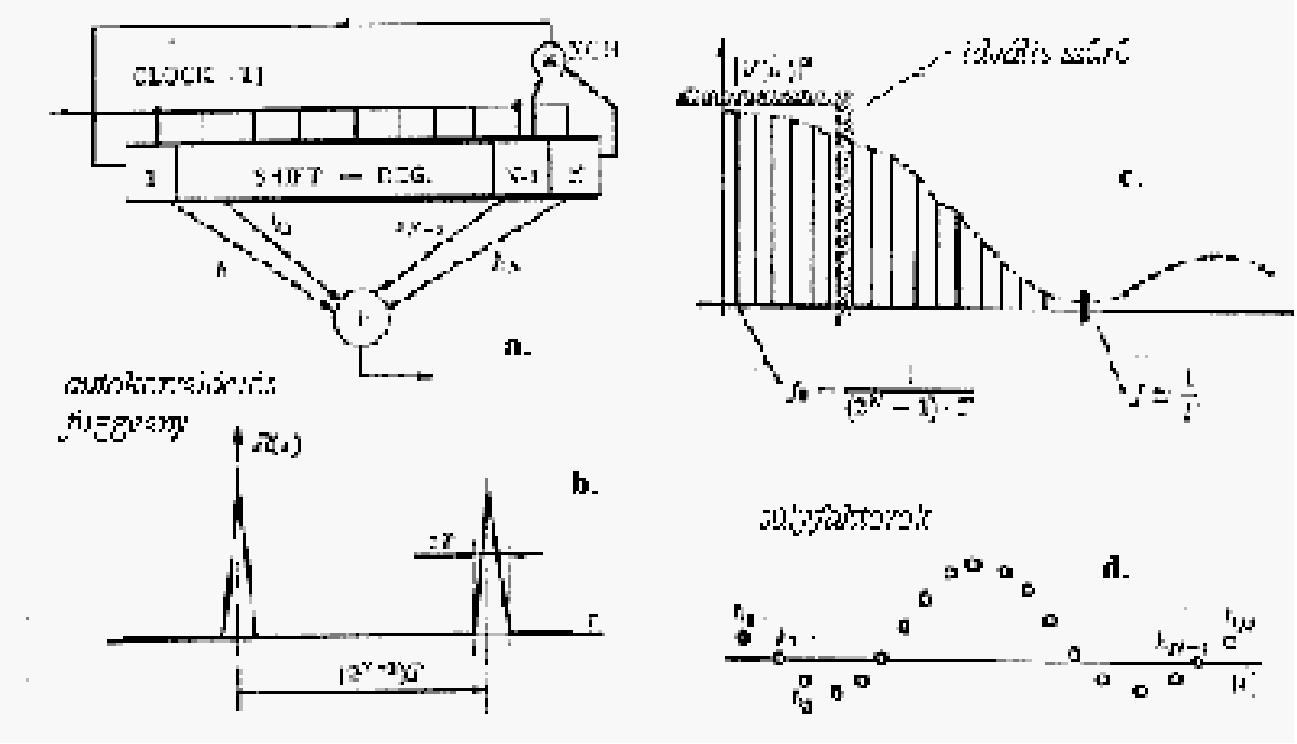
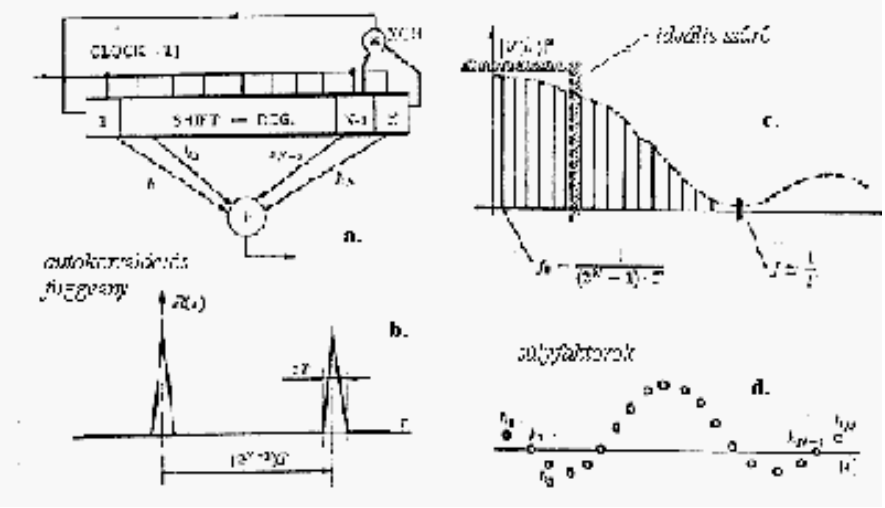


FIGURE 28-4
Microprocessor architecture. The Von Neumann architecture uses a single memory to hold both data and instructions. In comparison, the Harvard architecture uses separate memories for data and instructions, providing higher speed. The Super Harvard Architecture improves upon the Harvard design by adding an instruction cache and a dedicated I/O controller.



HP zajgenerátor:





Shift regiszter + XOR kapu: 2^n állapot
Autókorrelációs függvény: l. pénzdobálás: $(\sin x/x)^2$ burkoló!

Fehérzajhoz a konstans érték kell - megoldás a digitális szűrő
Összegezési súlyok $\sin x/x$ szerint (ez akár analóg is lehet!!!)
Lépcsőfüggvény az órajelnél rövidebb idő alatt: további simítás

Szűrő típusok

Alul/felül/sáváteresztő/sávzáró
Aluláteresztő:

$$H(\omega) = \frac{1}{1 + F(\omega/\omega_0)}$$

- ingadozás az átviteli sávban rossz
- ingadozás a vágási részen elfogadható
  pl. -3 dB adhatja meg a vágási pontot (mint az RC esetén)
- vágás mértéke: dB/dekád
- vágási szint (pl. -40 dB)
- Fázis linearitás csak az átviteli sávban fontos
- Fázismenetben a kicsi nemlinearitás még elfogadható lehet

Tervezés: bonyolult, n elemre tervezőprogramokkal optimalizálják a fázis és frekvenciamenetet.

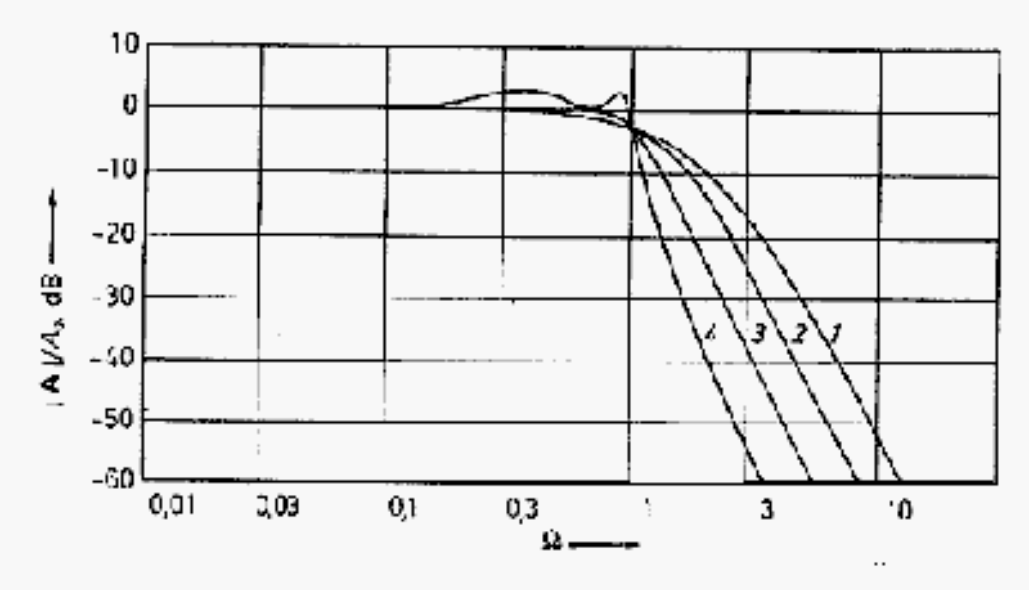
Példák:

- Bessel: $F(\omega/\omega_0)$ Bessel-fv.
  maximálisan sima fázismenet
  lassú felfutás

- Butterworth: $F(\omega/\omega_0) = ((\omega/\omega_0)^{2n}$
  teljesen sima az átviteli sávban
  „legsimább" átvitel

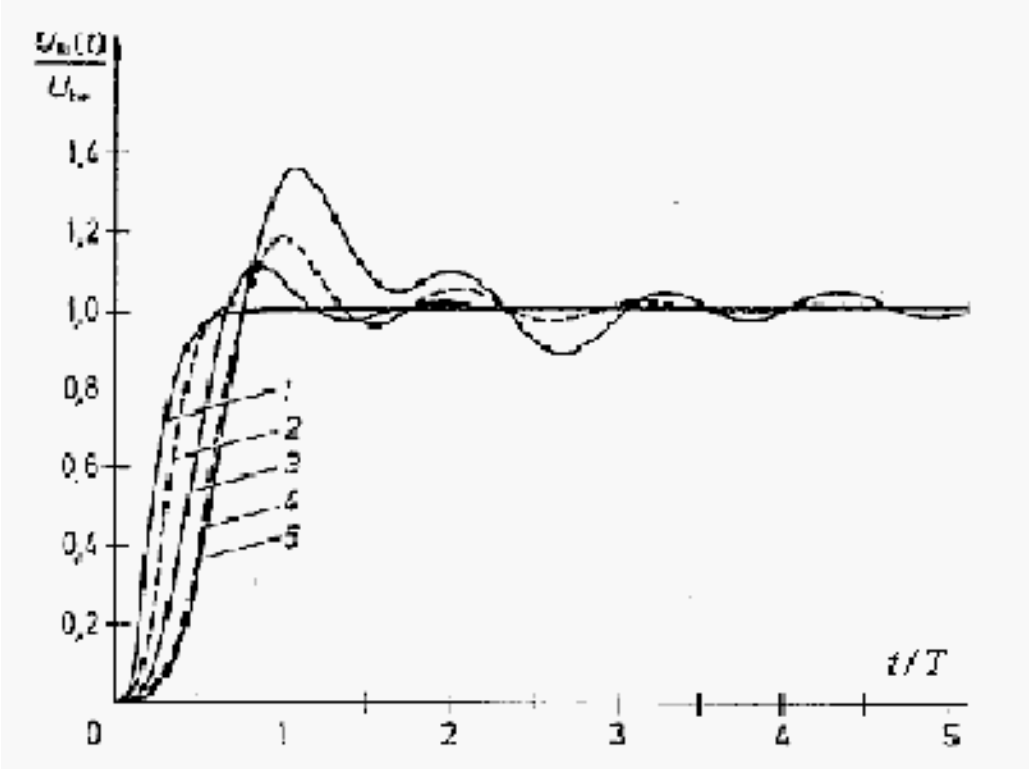
- Csebisev: $F(\omega/\omega_0)$ Csebisev-fv.
  gyors lefutás, de ingadozás is van
  rossz fázismenet

Frekvenciamenet:



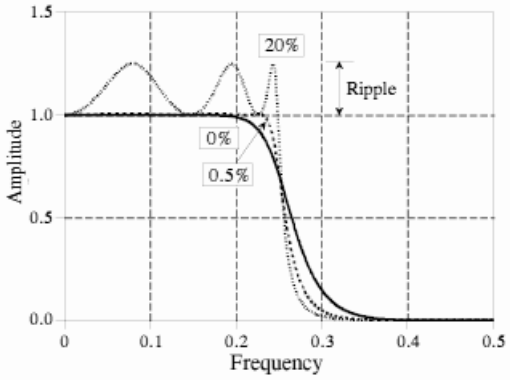
1 = kritikusan csillapított; 2 = Bessel; 3 = Butterworth; 4 = Csebisev szűrő

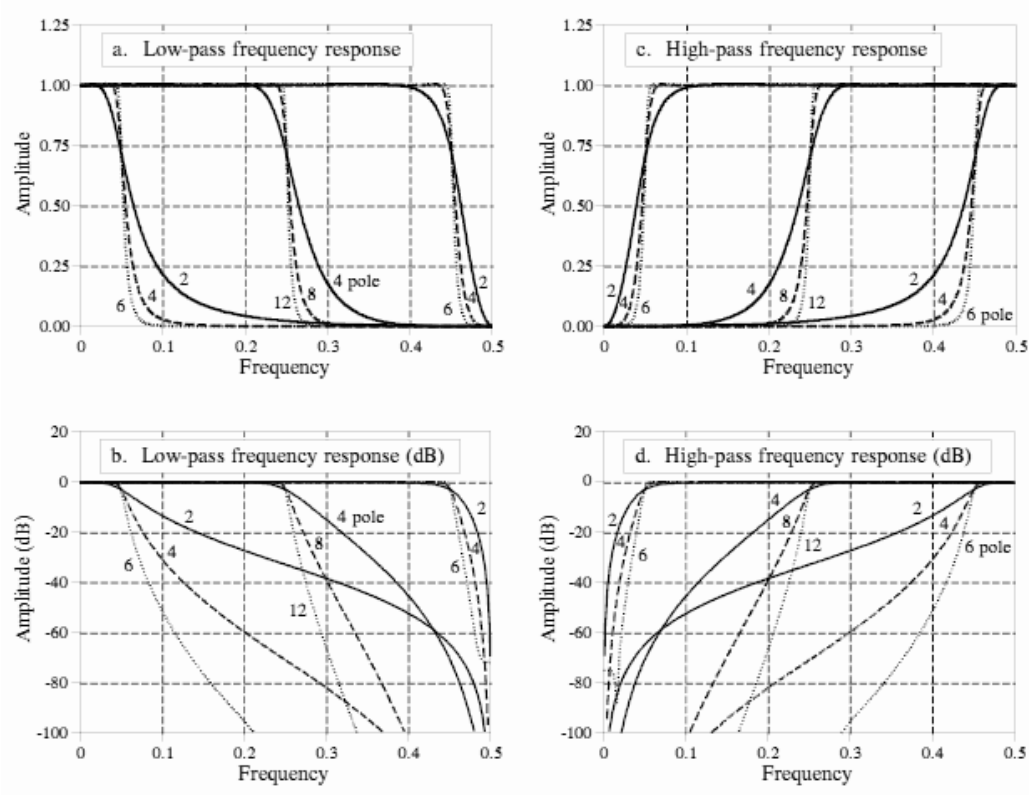
Ugrásfüggvény-átvitel:

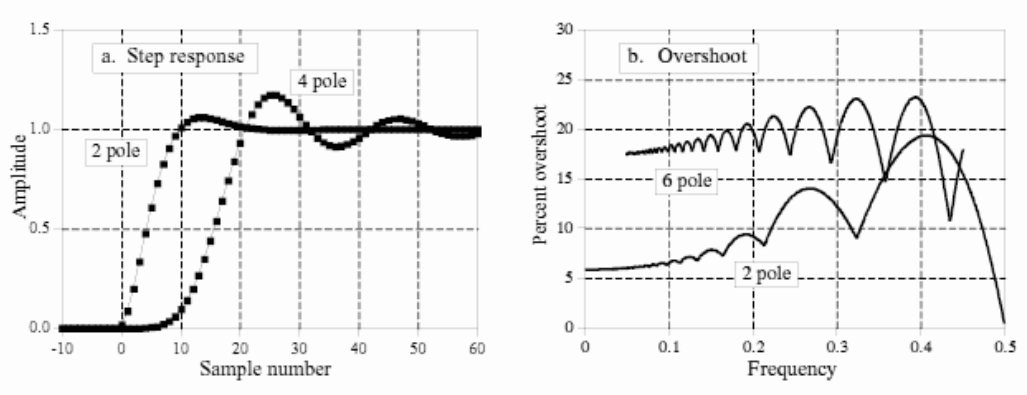


1 = kritikusan csillapított; 2 = Bessel; 3 = Butterworth; 4/5 = Csebisev szűrő

FIGURE 20-1
The Chebyshev response. Chebyshev filters achieve a faster roll-off by allowing ripple in the passband. When the ripple is set to 0%, it is called a *maximally flat* or *Butterworth* filter. Consider using a ripple of 0.5% in your designs; this passband unflatness is so small that it cannot be seen in this graph, but the roll-off is much faster than the Butterworth.

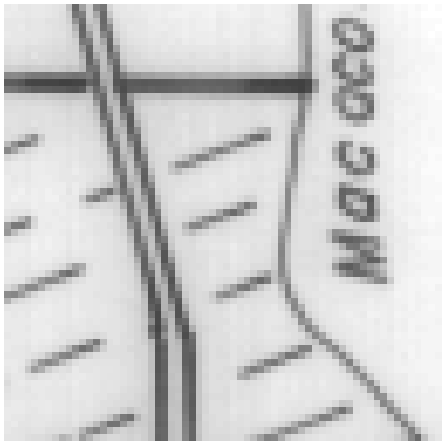


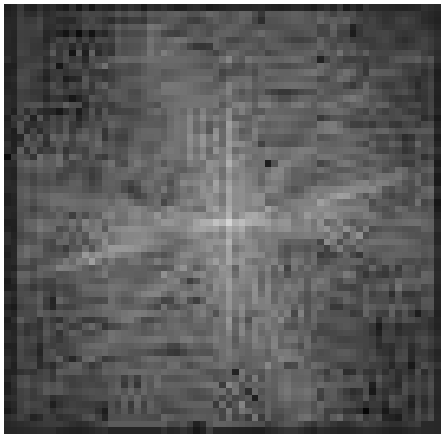



a. Step response
4 pole
2 pole
Amplitude
Sample number
1.5
1.0
0.5
0.0
-10
0
10
20
30
40
50
60
b. Overshoot
6 pole
2 pole
Percent overshoot
Frequency
30
25
20
15
10
5
0
0
0.1
0.2
0.3
0.4
0.5

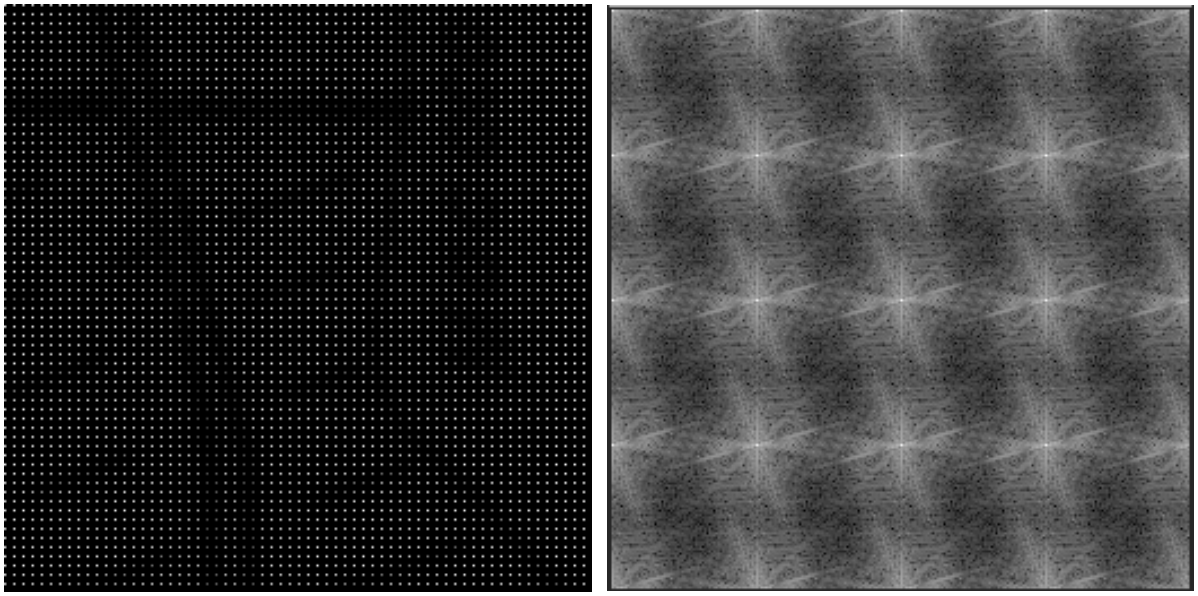
Képek nagyítása, interpoláció

Eredeti kép és DFT spektrum:

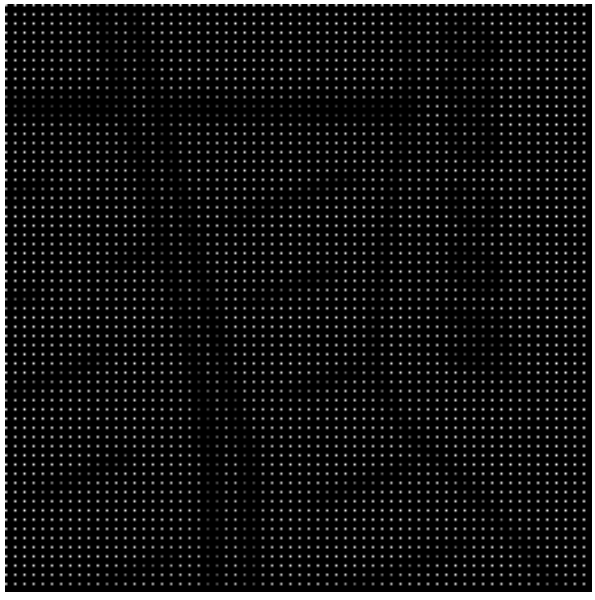




Nagyított kép és DFT spektruma:

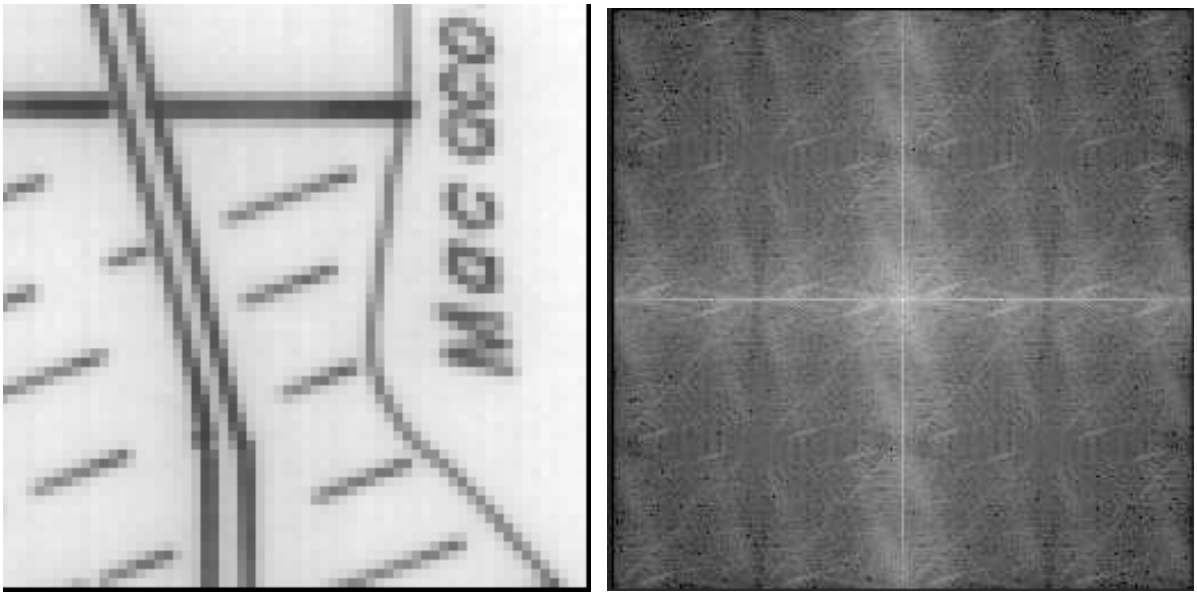


Legközelebbi szomszéd, átlagolás:
Nagyított kép és a simitó magfüggvény:

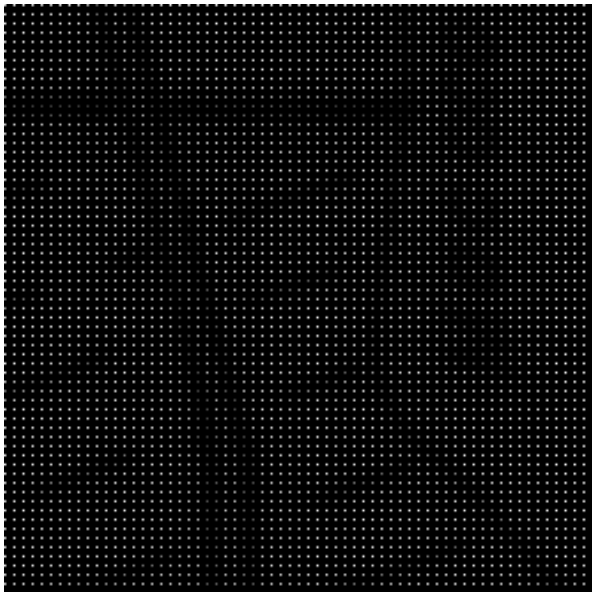


1 1 1 1
1 1 1 1
1 1 1 1
1 1 1 1

Interpolált kép és DFT spektruma:

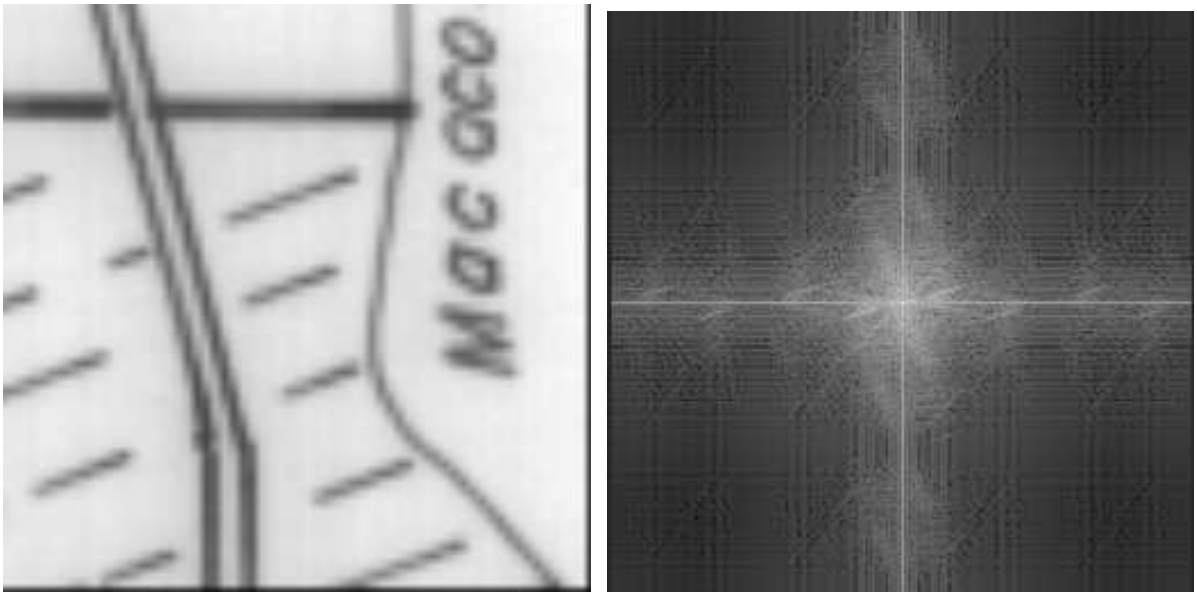


Lineáris interpoláció :
Nagyított kép és a simitó magfüggvény:

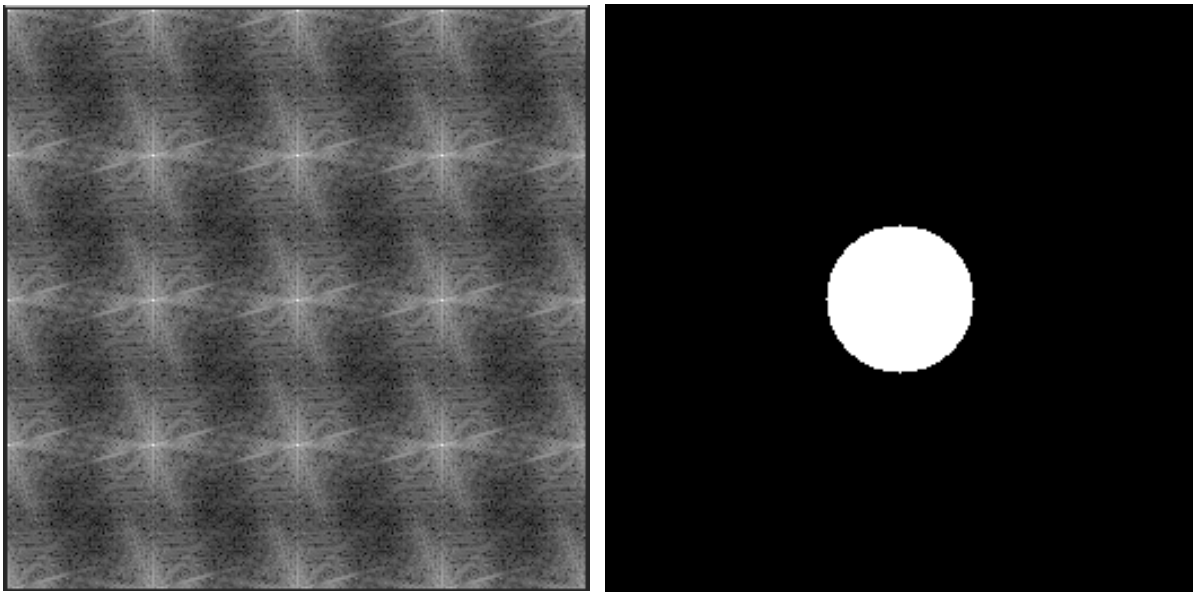


00	00	00	00	00	00	00	00	00
00	01	02	03	04	03	02	01	00
00	02	04	06	08	06	04	02	00
00	03	06	09	12	09	06	03	00
00	04	08	12	16	12	08	04	00
00	03	06	09	12	09	06	03	00
00	02	04	06	08	06	04	02	00
00	01	02	03	04	03	02	01	00
00	00	00	00	00	00	00	00	00

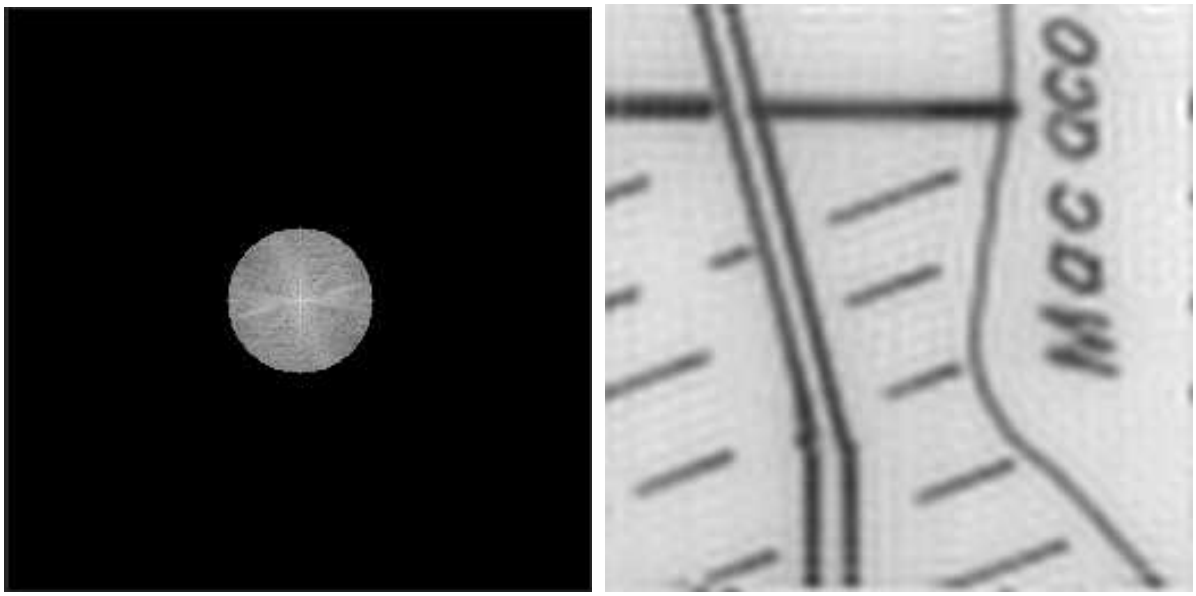
Interpolált kép és DFT spektruma:



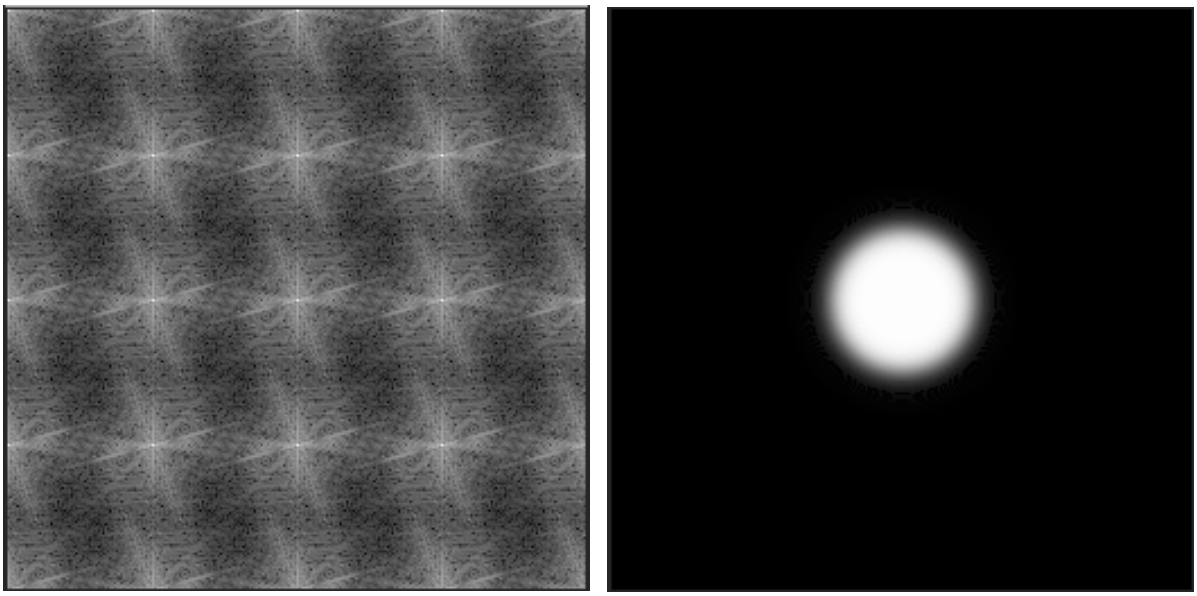
Ideális aluláteresztő:
Nagyított kép és az ideális aluláteresztő szűrő:



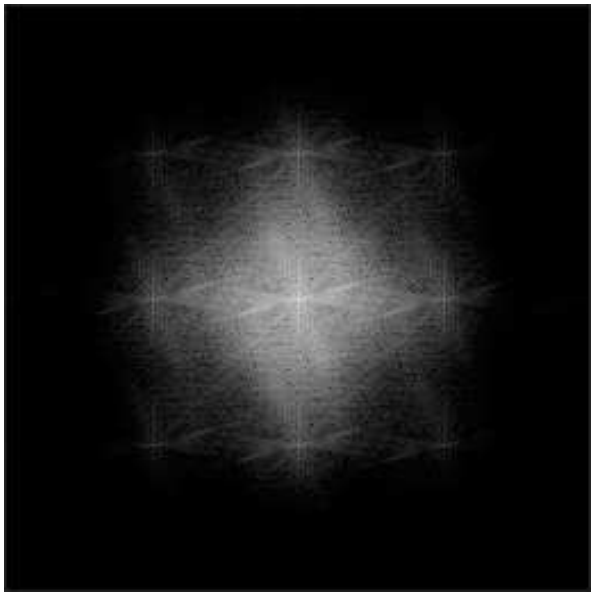
Interpolált kép és DFT spektruma:

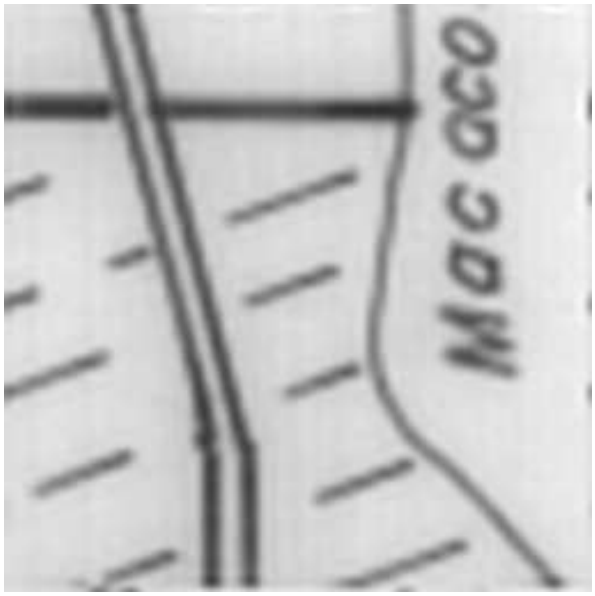


Butterworth szűrés:
Nagyított kép és a Butterworth szűrő:

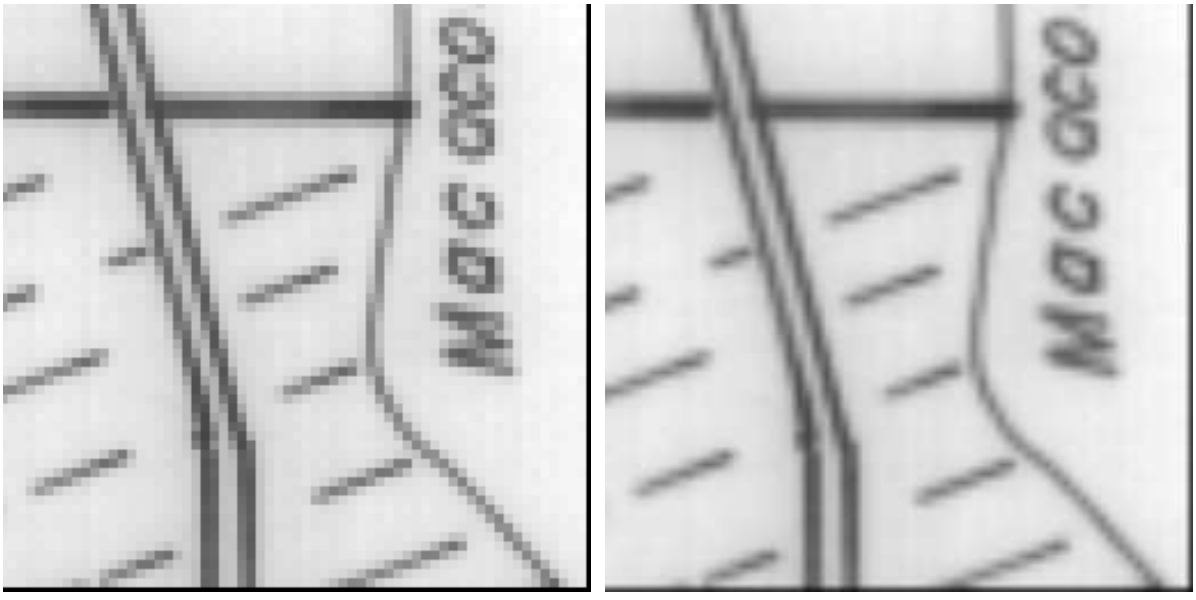


Interpolált kép és DFT spektruma:

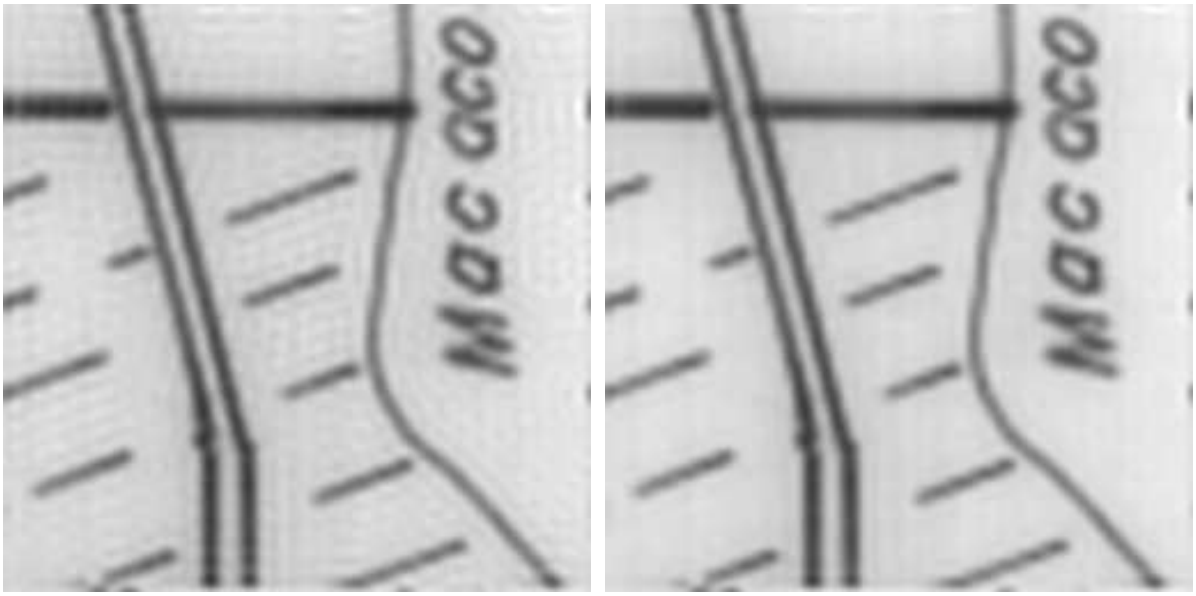


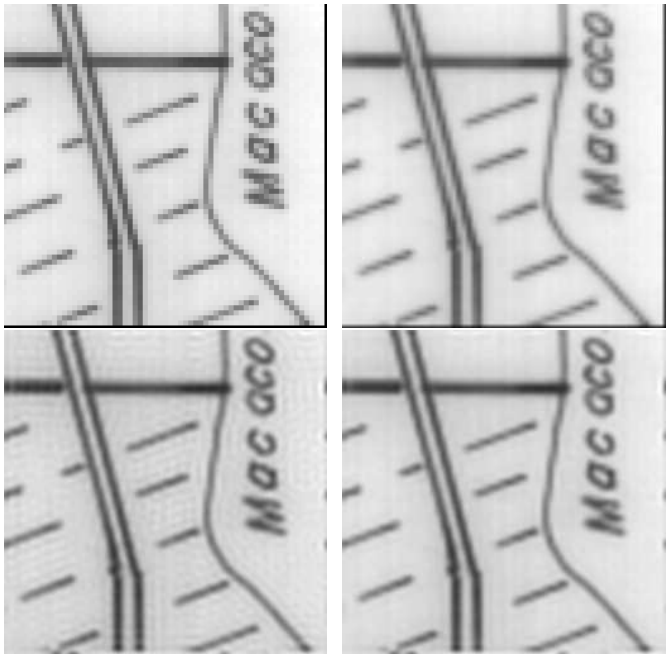


Összehasonlítás:
Legközelebbi szomszéd és lineáris interpoláció:



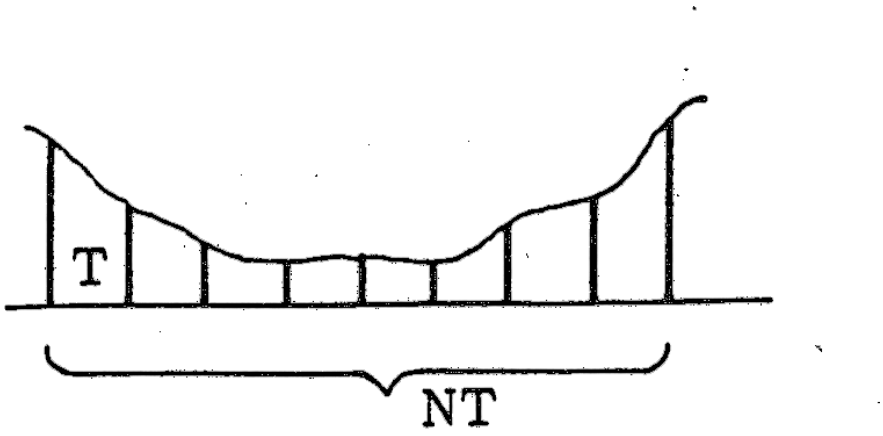
Ideális aluláteresztő és Butterworth




Mac Arco
Mac Arco
Mac Arco
Mac Arco

DFT - diszkrét Fourier transzformáció

Sávlimitált jel - N minta, periódikus!



Fourier transzformált:

$$V(k\Omega) = \sum_{n=0}^{N-1} v(nT) e^{-i\Omega knT}$$

itt $\Omega = \dfrac{2\pi}{NT}$

Mennyi információ, hány különböző jel? N mérési eredmény

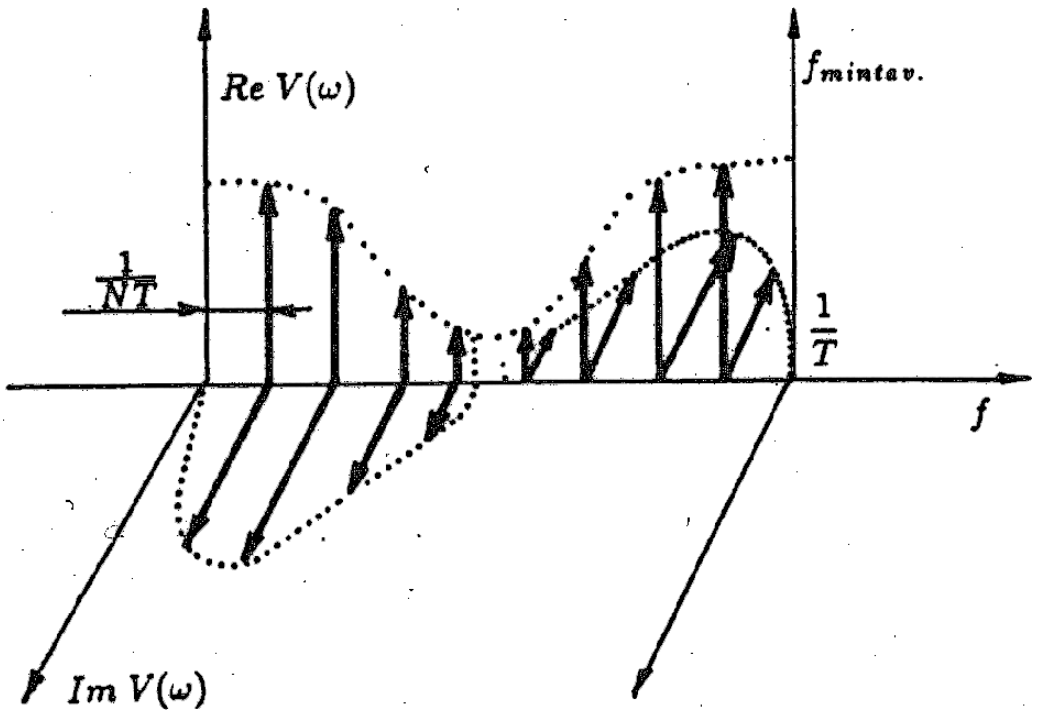
Ha $k = rN + k_0$ és r =egész

$$V(k\Omega) = \sum_{n=0}^{N-1} v(nT) e^{-i\Omega(rN+k_0)nT} = \sum_{n=0}^{N-1} v(nT) e^{-i\Omega k_0 nT}$$

$$\left(e^{-i\Omega TNr} \equiv 1\right)$$

A spektrum N periódussal ismétlődik!

N (valós) adat $\longrightarrow N/2$ független (komplex!) spektrumvonal!



A diszkrét Fourier transzformáció mátrixművelet: ($\omega = \exp(i2\pi/N)$):

$$\begin{pmatrix} V_0 \\ V_1 \\ V_2 \\ \vdots \\ V_{N-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \omega & \omega^{-2} & \dots & \omega^{-(N-1)} \\ 1 & \omega^{-2} & \omega^{-4} & \dots & \omega^{-2(N-1)} \\ & & \vdots & & \\ 1 & \omega^{-(N-1)} & \omega^{-2(N-1)} & \dots & \omega^{-(N-1)^2} \end{pmatrix} \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ \vdots \\ v_{N-1} \end{pmatrix}$$

$N = 4$-re nincs szorzás!

$$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & -i & -1 & i \\ 1 & -1 & 1 & -1 \\ 1 & i & -1 & -i \end{pmatrix}$$

Ha csak egy érték nem nulla (pl. Dirac delta) $\longrightarrow$ minden tag ugyanakkora!

Oda-vissza transzformáció: $\longrightarrow \frac{1}{N}$ faktor!

$$V_k = \sum_{n=0}^{N-1} \omega^{-nk} v_n$$

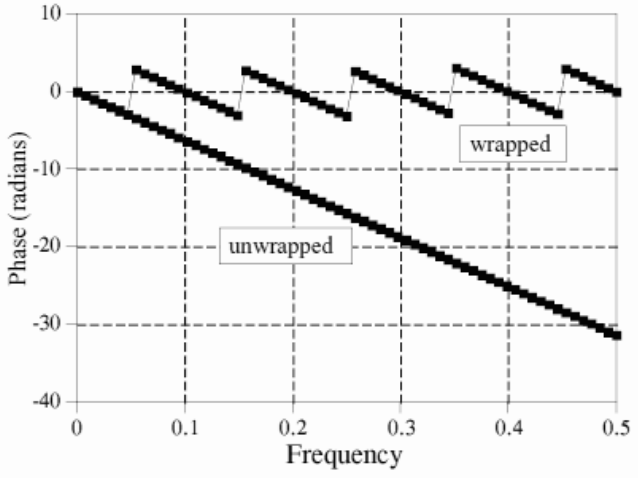
$$v_n = \frac{1}{N}\sum_{k=0}^{N-1}\omega^{nk}V_k = \frac{1}{N}\sum_{k=0}^{N-1}\omega^{nk}\sum_{l=0}^{N-1}\omega^{-lk}v_l$$

$$= \frac{1}{N}\sum_{l=0}^{N-1}v_l\sum_{k=0}^{N-1}\omega^{k(n-l)} = \begin{cases} v_n & (n=l) \\ 0 & (n\neq l) \end{cases}$$

Itt $\omega = e^{i2\pi/N}$, $\qquad (1-\omega^{(n-l)N})/(1-\omega^{n-l}) = 0$

Fázis problémák:

FIGURE 8-12
Example of phase unwrapping. The top curve shows a typical phase signal obtained from a rectangular-to-polar conversion routine. Each value in the signal must be between -π and π (i.e., -3.14159 and 3.14159). As shown in the lower curve, the phase can be *unwrapped* by adding or subtracting integer multiplies of 2π from each sample, where the integer is chosen to minimize the discontinuities between points.



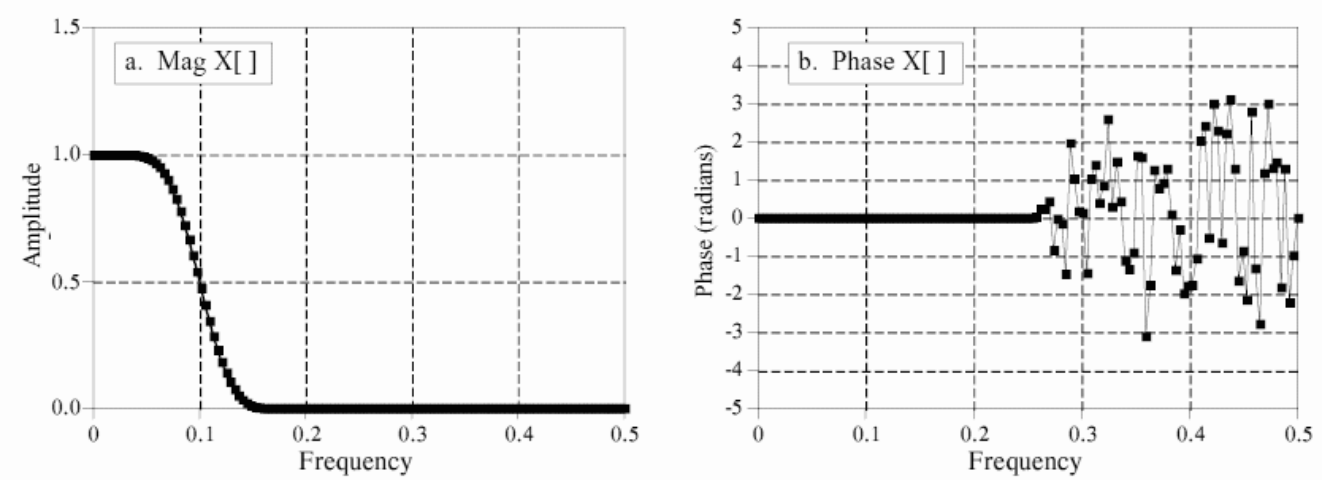
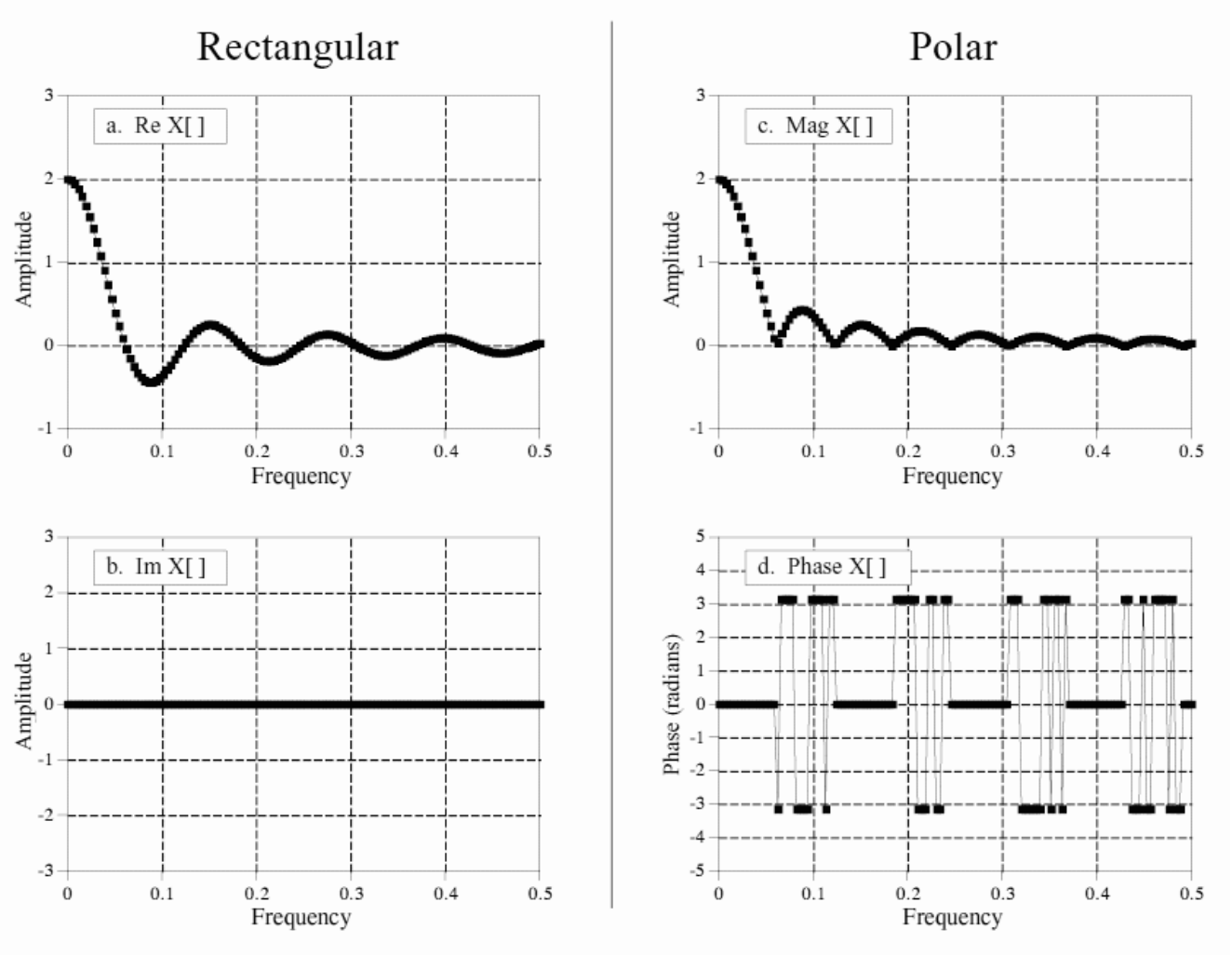
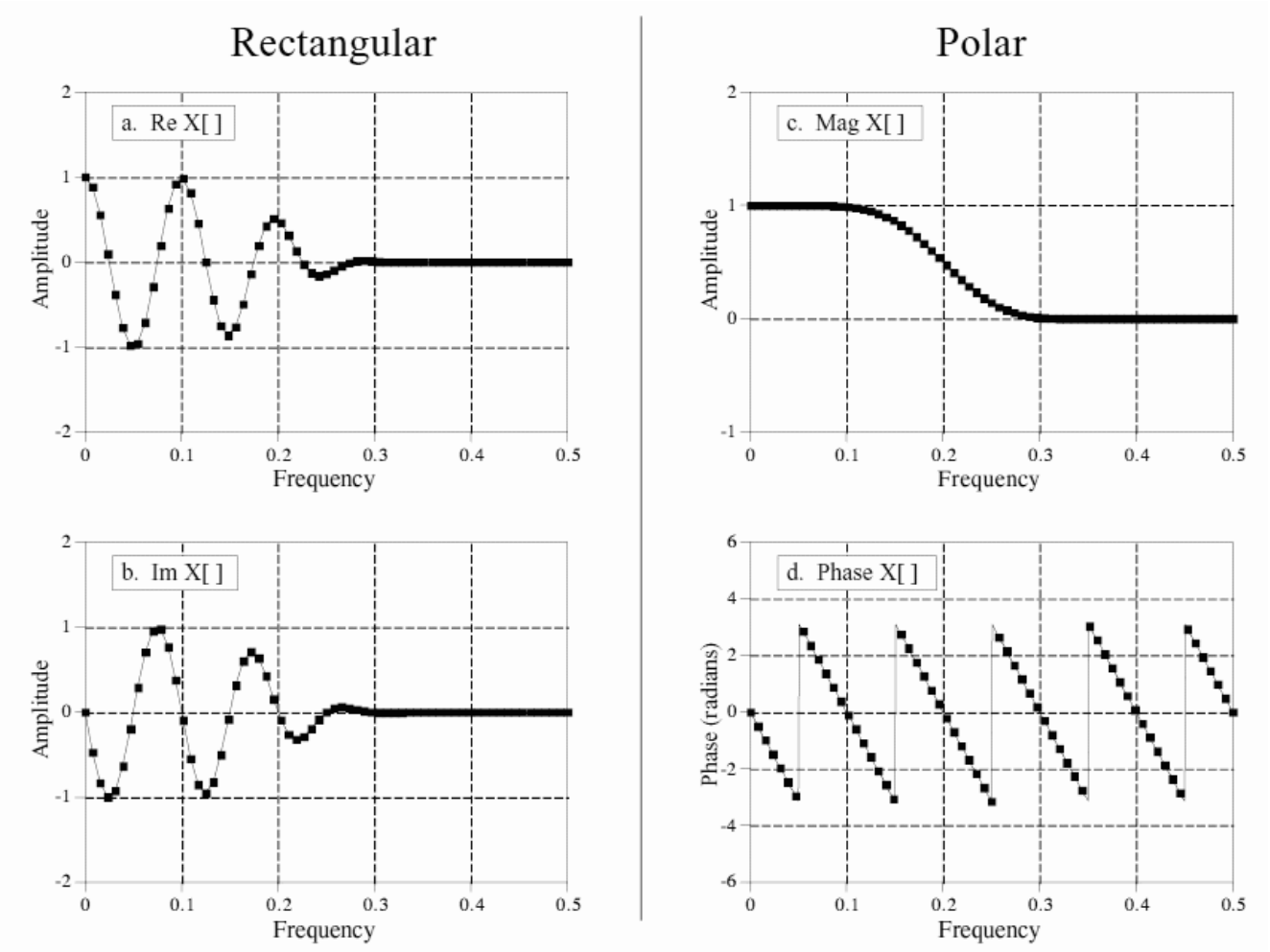


FIGURE 8-11
The phase of small magnitude signals. At frequencies where the magnitude drops to a very low value, round-off noise can cause wild excursions of the phase. Don't make the mistake of thinking this is a meaningful signal.


Rectangular
Polar
a. Re X[ ]
b. Im X[ ]
c. Mag X[ ]
d. Phase X[ ]
Amplitude
Phase (radians)
Frequency
0
0.1
0.2
0.3
0.4
0.5


Rectangular
Polar
a. Re X[ ]
b. Im X[ ]
c. Mag X[ ]
d. Phase X[ ]
Amplitude
Phase (radians)
Frequency
2
1
0
-1
-2
4
6
-4
-6
0.1
0.2
0.3
0.4
0.5

FFT - Fast Fourier transzformáció

Fourier transzformáció $\longrightarrow N^2$ szorzás (ugyanannyi, mint a konvolúció)

1965: Cooley és Tukey $\longrightarrow N \log_2 N$ algoritmus
1978: Winograd algoritmus

	szorzás/pont	összeadás/pont
DFT (1000 $\times$ 1000)	8000	4000
CT-FFT (1024 $\times$ 1024)	4O	6O
Winograd (1008 $\times$ 1008)	6	91

CT algoritmus: részekre bontás:
N pont DFT $\longrightarrow N^2$ szorzás
$N/2$ pont DFT $\longrightarrow 2(N/2)^2$ szorzás külön-külön:
eredeti v adatok: g páros és h páratlan adatokra bonthatók, ekkor G és H kombinációja ω^k-val szorzódik:

$$v(n) \quad (n = 0, 1, \ldots, N-1) \quad \Rightarrow \begin{cases} g(n) = v(2n) & n = 0, 1, \ldots N \\ h(n) = v(2n+1) & 0, 1, \ldots N/2 - \end{cases}$$

$$V_k = \sum_{n=0}^{N-1} v_n \omega^{-nk} = \sum_{n=0}^{N/2-1} g_n \omega^{-2nk} + \sum_{n=0}^{N/2-1} h_n \omega^{(2n+1)k} \qquad k = 0, 1, \ldots N$$

$$\omega_N^2 = \omega_{N/2}$$

$$V_k = \sum_{n=0}^{N/2-1} g_n \omega_{N/2}^{-nk} + \omega_N^{-k} \sum_{n=0}^{N/2-1} h_n \omega_{N/2}^{-nk} = G_k + \omega_N^k H_k \qquad k = 0, 1, \ldots$$

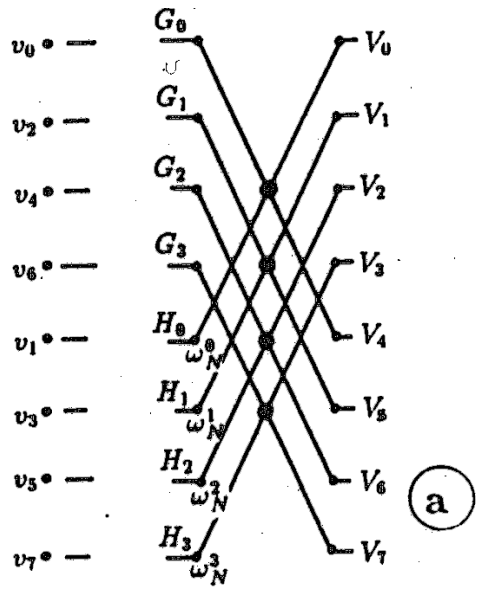
$k > N/2 - 1$ esetén ez $k - N/2$:

$$\begin{aligned} V_k &= G_{k-N/2} + \omega_N^{-k} H_{k-N/2} && N/2 \leq k \leq N-1 \\ V_{k+N/2} &= G_k + \omega_N^{-k+N/2} H_k && 0 \leq k \leq N/2 - 1 \end{aligned}$$

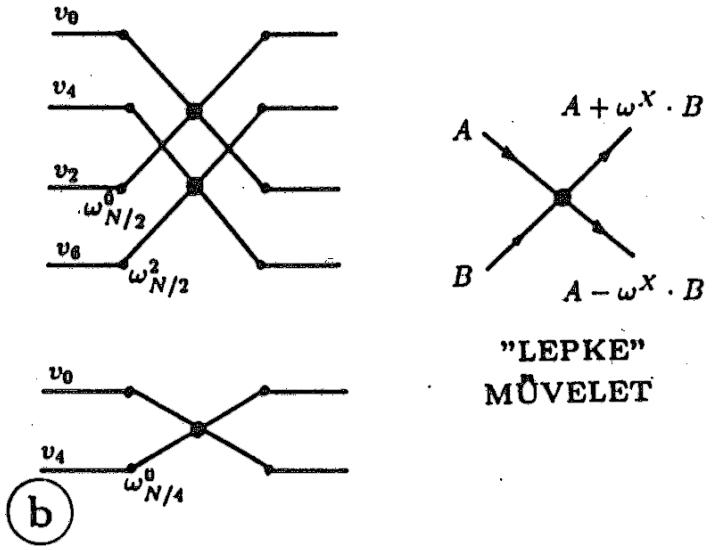
Ha $N = 2^k \rightarrow$ többszörösen ismételhető

Az FFT algoritmus részei $N = 8$ esetén

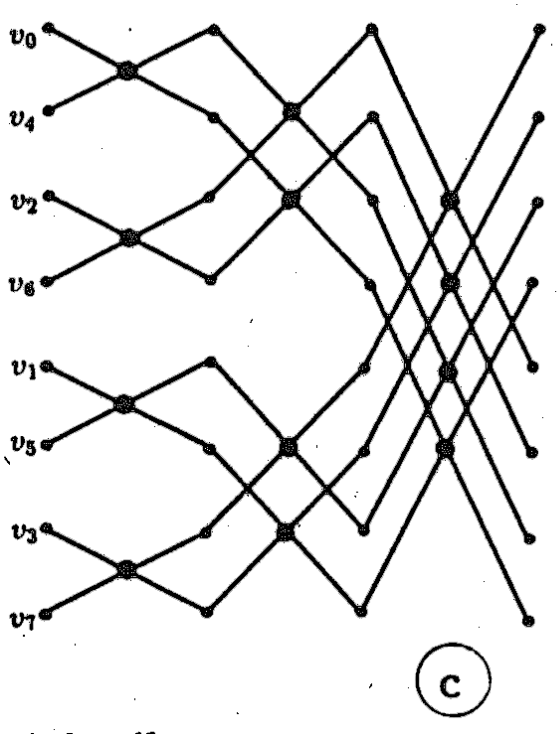
Kiinduló adatok és az első lépés



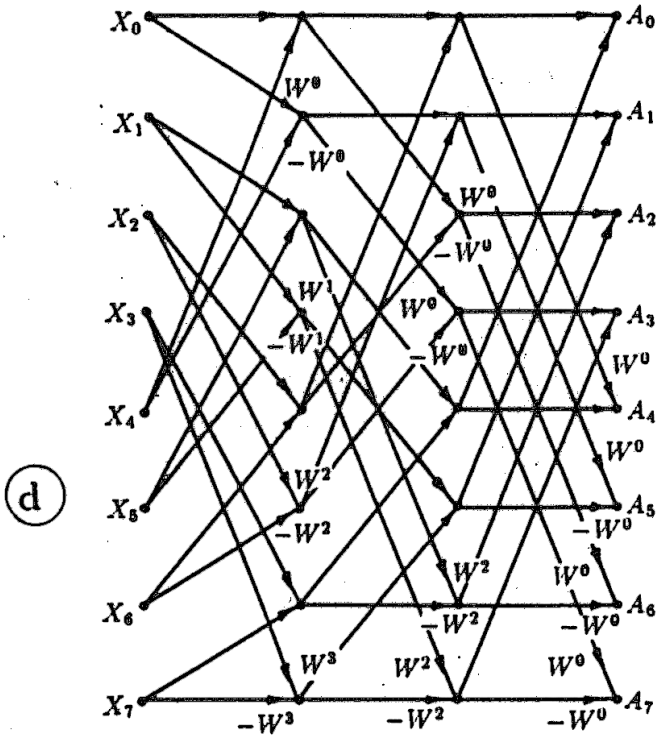
Lepkeműveletet



A 3 fázis



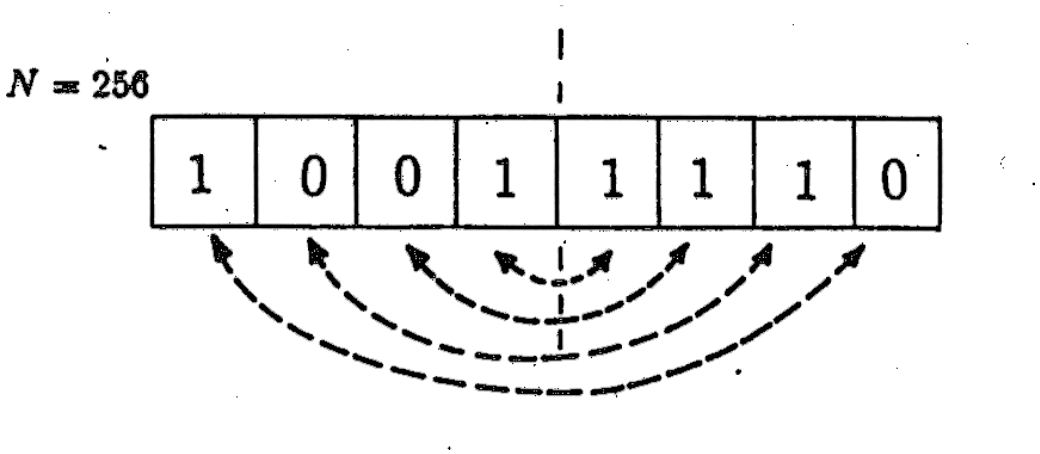
Teljes FFT


X0
X1
X2
X3
X4
X5
X6
X7
A0
A1
A2
A3
A4
A5
A6
A7
W0
-W0
W0
-W0
W1
-W1
W0
-W0
W0
W0
W2
-W2
W2
W0
-W2
-W0
W3
W2
W0
-W3
-W2
-W0

d

Teljes transzformáció: $(N/2)\log_2 N$ lepkeművelet
$N \log_2 N$: pl. $N = 1024$ esetén csak 1% szorzás marad!

Adatok keverednek az FFT-ben: bit-reverzálás



Sok algoritmus (Numerical Recipes, speciális progamok)

További sebességnövelés: pl. csak valós adatokkal:

$$
\begin{aligned}
v_1(n) \quad & \Longleftrightarrow V_1(k) \\
v_2(n) \quad & \Longleftrightarrow V_2(k) \\
V(k) \;=\; & V_1(k) + iV_2(k) \qquad\qquad V^*(k) = V_1(k) - jV_2(k) \\
V_1(k) \;=\; & \frac{1}{2}\left[V^*(N-k) + V(k)\right]
\end{aligned}
$$

$$V_2(k) \quad = \quad \frac{1}{2}\left[V^*(N-k) - V(k)\right]$$

FFT alkalmazásai

Univerzális eljárás: konvolúció-dekonvolúció:
Gyorsítás:

- gyorsabb számítógép
- aritmetikai processzor alkalmazása;
- lehetőleg mindent számítsunk ki előre (ún. look-up table)
- csökkentett fix számábrázolást;
- hatékony algoritmusok
- DSP (digital signal processor) chipek használata, szorzás és összeadás egy lépésben

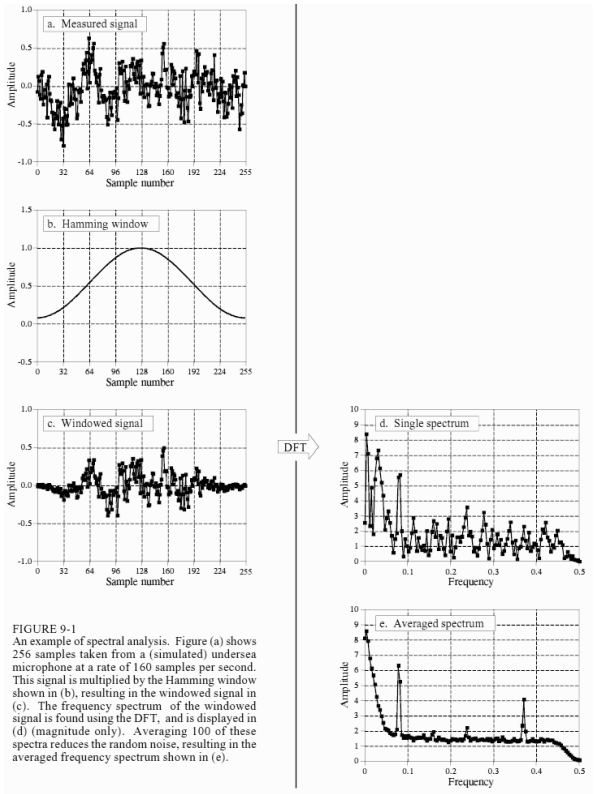
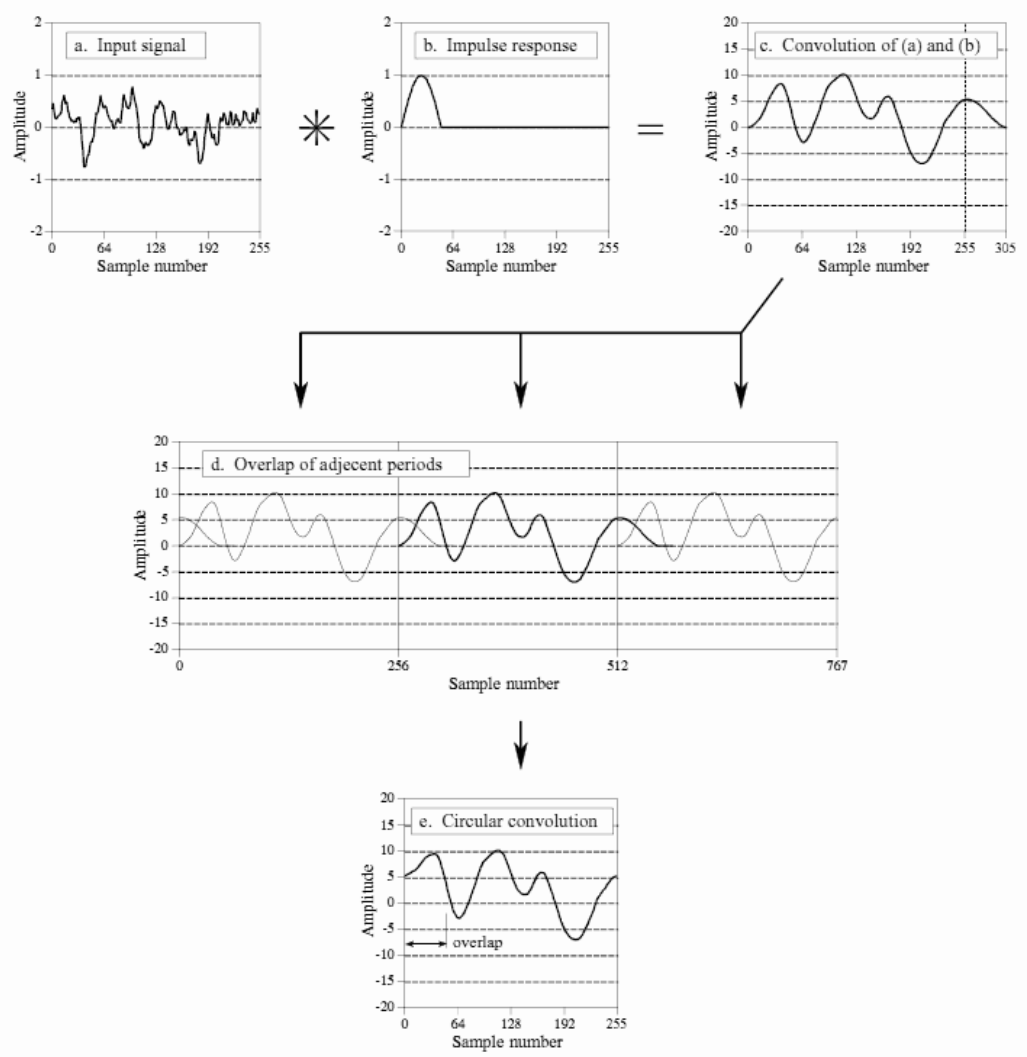
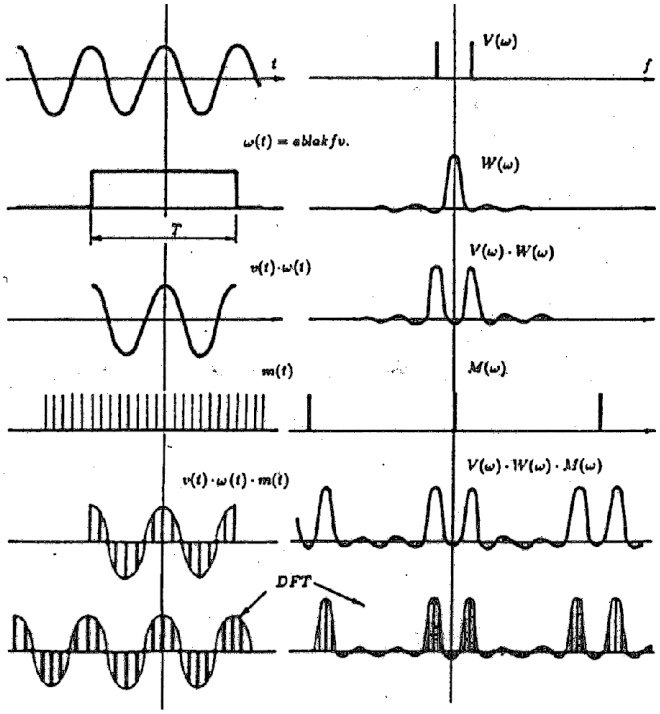


FIGURE 9-1
An example of spectral analysis. Figure (a) shows 256 samples taken from a (simulated) undersea microphone at a rate of 160 samples per second. This signal is multiplied by the Hamming window shown in (b), resulting in the windowed signal in (c). The frequency spectrum of the windowed signal is found using the DFT, and is displayed in (d) (magnitude only). Averaging 100 of these spectra reduces the random noise, resulting in the averaged frequency spectrum shown in (e).

Cirkuláris konvolúció!

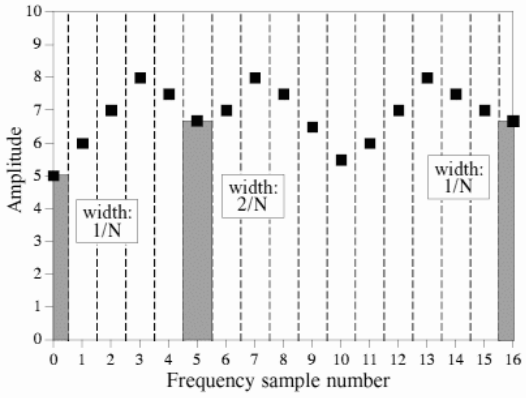


Problémák a kvázi-periodikusság miatt:



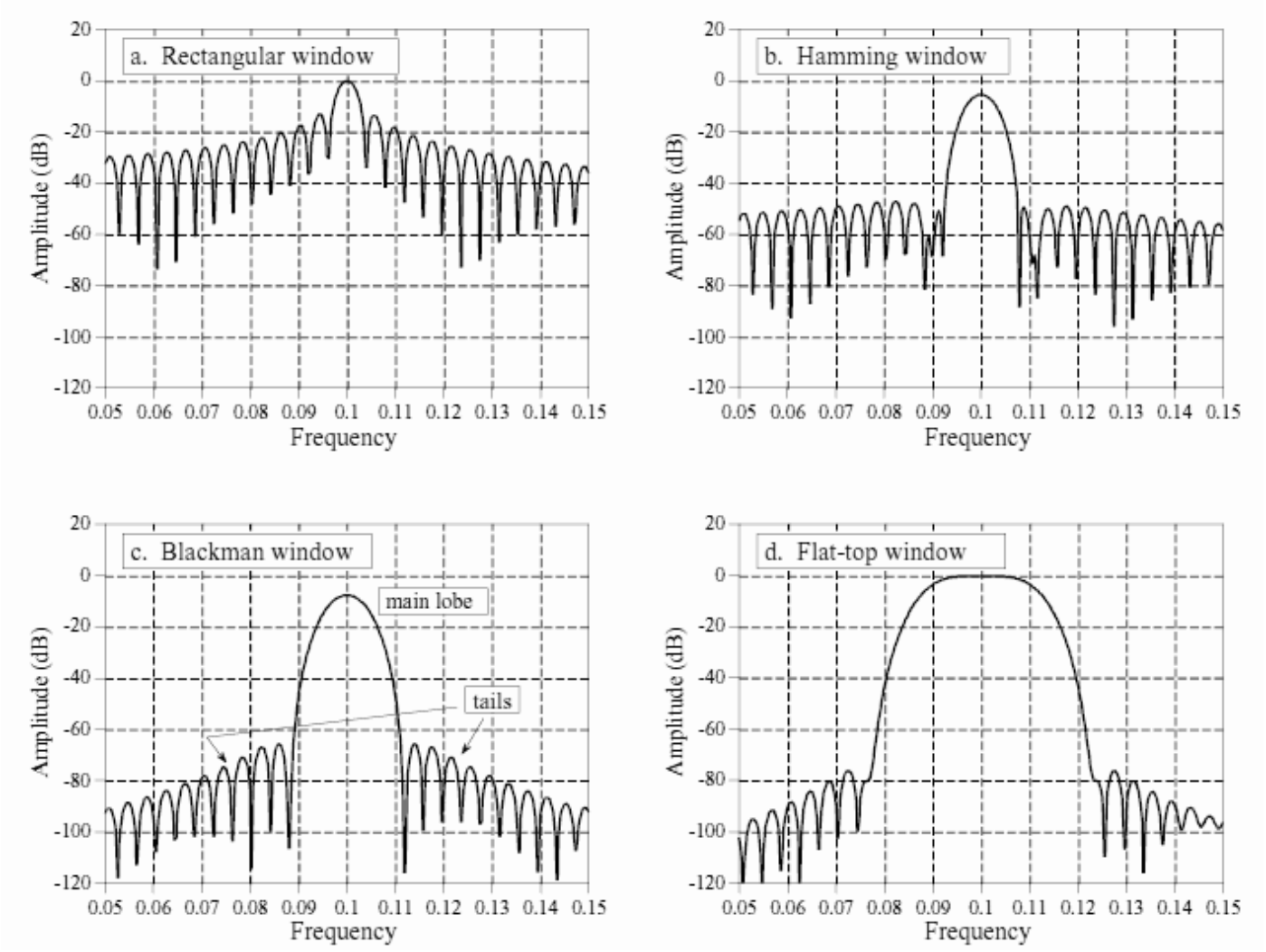
Diszkrét értékek, szivárgás, ha egy jel nem PONTOSAN egy adott spektrumvonalra esik!

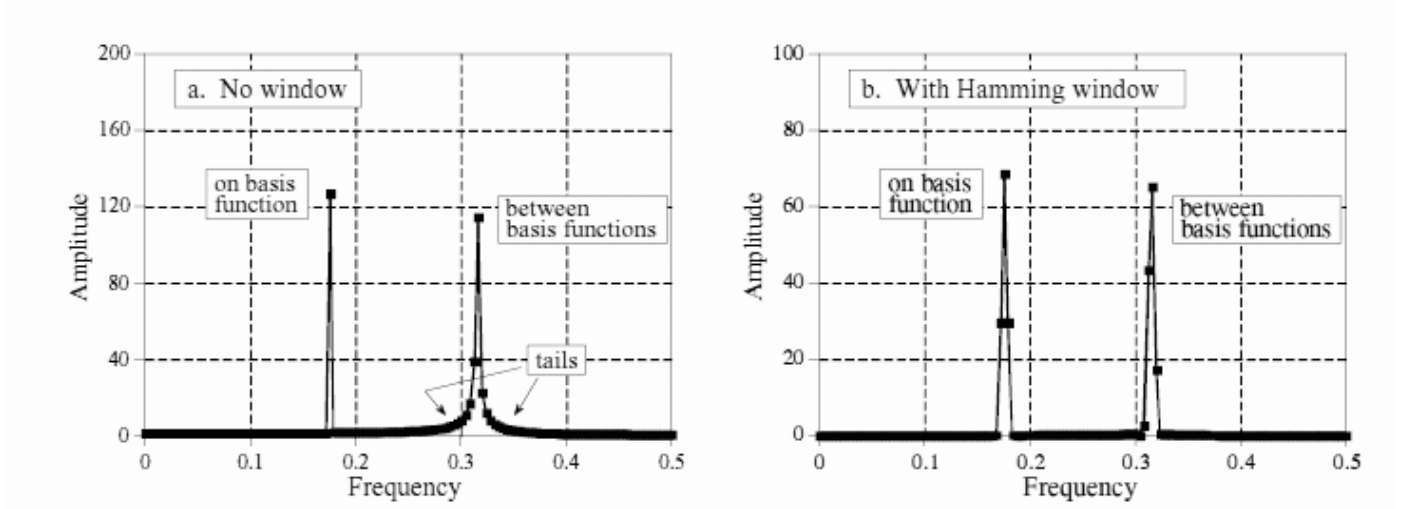
FIGURE 8-7
The bandwidth of frequency domain samples. Each sample in the frequency domain can be thought of as being contained in a frequency band of width $2/N$, expressed as a fraction of the total bandwidth. An exception to this is the first and last samples, which have a bandwidth only one-half this wide, $1/N$.



Megoldás: ablakfüggvény alkalmazása:

négyszög 1	4π/N	22 %
Bartlett (háromszög) $1 - 2\|n\|/(N-1)$	8π/N	4 %
Hamming $0.54 + 0.46\cos(\frac{2\pi n}{N-1})$	8π/N	1 %
Hanning $0.5 + 0.5\cos(\frac{2\pi n}{N-1})$	8π/N	2.6 %
Blackman $0.42 + 0.5\cos(\frac{2\pi n}{N-1}) + 0.08\cos(\frac{4\pi n}{N-1})$	12π/N	0.1 %


a. Rectangular window
b. Hamming window
c. Blackman window
main lobe
tails
d. Flat-top window
Amplitude (dB)
Frequency
20
0
-20
-40
-60
-80
-100
-120
0.05 0.06 0.07 0.08 0.09 0.1 0.11 0.12 0.13 0.14 0.15


a. No window
on basis function
between basis functions
tails
Amplitude
Frequency
200
160
120
80
40
0
0
0.1
0.2
0.3
0.4
0.5
b. With Hamming window
on basis function
between basis functions
Amplitude
Frequency
100
80
60
40
20
0
0
0.1
0.2
0.3
0.4
0.5

Egyéb eljárások, módszerek

Strassen algoritmus: műveletszám csökkentése:

$$e + jf = (a + jb)(c + jd) \qquad \Longrightarrow \qquad e = ac - bd, \quad f = ad + bc.$$

$$\text{Strassen}: \qquad e = (a-b)d + a(c-d), \quad f = (a-b)d + b(c+d).$$

Cirkuláris konvolució: pl. 8×8-as cikruláris konvolucióra 64 helyett 17 szorzás:

$$\begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} c & -d \\ d & c \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} \begin{pmatrix} c-d & 0 & 0 \\ 0 & c+d & 0 \\ 0 & 0 & d \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 0 & -1 \end{pmatrix}$$

Winograd algoritmus: primszámok hatványai, nem csak 2^n hatványok. Pl. ($\Theta = 2\pi/5$):

$$\begin{pmatrix} V_0 \\ V_1 \\ V_2 \\ V_3 \\ V_4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & -1 \\ 1 & 1 & -1 & 1 & 1 & 0 \\ 1 & 1 & -1 & -1 & -1 & 0 \\ 1 & 1 & 1 & -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} B_0 & 0 & 0 & \dots & & \\ 0 & B_1 & 0 & & & \\ 0 & 0 & B_2 & & & \\ \vdots & & & B_3 & & \\ & & & & B_4 & \\ & & & & & B_5 \end{pmatrix}$$

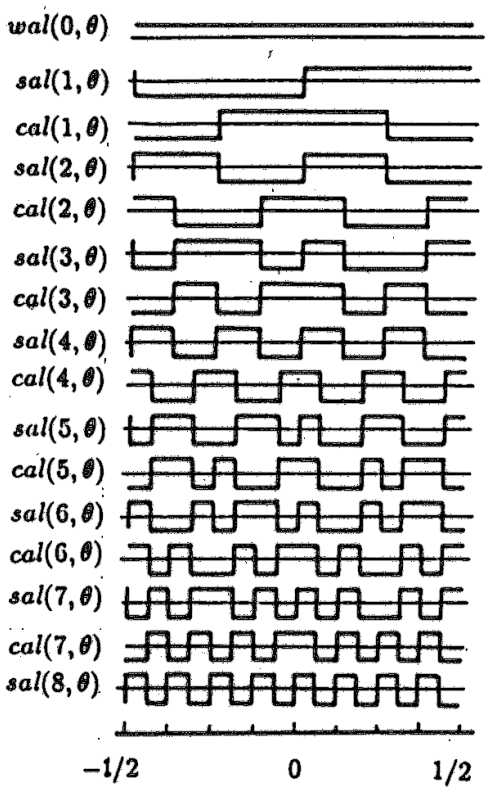
$$\times \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & -1 & -1 & 1 \\ 0 & 1 & -1 & 1 & -1 \\ 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & -1 & 1 & 0 \end{pmatrix} \begin{pmatrix} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \end{pmatrix}$$

$$B_0 = 1, \quad B_1 = 0.5(\cos\Theta + \cos 2\Theta) - 1, \quad B_2 = 0.5(\cos\Theta - \cos 2\Theta),$$

$$B_3 = j\sin\Theta, \quad B_4 = j(-\sin\Theta + \sin 2\Theta), \quad B_5 = j(\sin\Theta + \sin 2\Theta).$$

Bonyolult programozás!

Más felbontás: Walsh, Haar, wavelet alapok!


wal(0, θ)
sal(1, θ)
cal(1, θ)
sal(2, θ)
cal(2, θ)
sal(3, θ)
cal(3, θ)
sal(4, θ)
cal(4, θ)
sal(5, θ)
cal(5, θ)
sal(6, θ)
cal(6, θ)
sal(7, θ)
cal(7, θ)
sal(8, θ)
−1/2
0
1/2