Gaál József

Digitális jelfeldolgozás eszközei

II. kötet

A jelfeldolgozás szoftver eszközei

Matlab gyakorlatok

KHVM - BMGE HT (2001.)

3. kiadás
Budapest, 2006.

Bevezetés

1. Gyakorlat: ARMA jelek idő- és operátortartományi reprezentációinak grafikus vizsgálatára

2. Gyakorlat: Idő- és frekvenciatartományi reprezentációk grafikus vizsgálatára

3. Gyakorlat: Sávhatárolt KF jel alul-mintavételezési frekvenciájának meghatározása

4. Gyakorlat: Egyenletes és logaritmikus kvantálók vizsgálata

5. Gyakorlat: Differenciális kódolók vizsgálata

Bevezetés

A digitális jelfeldolgozás diszkrét idejű jeleinek és rendszereinek analízisét és szintézisét igen hatékonyan támogató lehetőségeket nyújt (a korábban már más tárgyak során megismert) MATLAB. Ezen tárgy keretében, néhány számítógéptermi gyakorlat során pár jellegzetes példa-feladat elkészítésére és működtetésére kerül sor.

A gyakorlatok fő célja, hogy kidogozott esetek kapcsán, mintegy példát mutatva segítsen kialakítani azt a készséget, mellyel a későbbiekben a mindenkori problémakörhöz illeszkedve mindenki elkészítheti majd a saját feladat-orientált eszközeit.

Néhány általános tanács:

- javasolt az on-line help intenzív használata
- a “szerszámainkat” .m file-ba írjuk
- folyamatos tesztelés MATLAB command window-ban
- használjuk az autohelp konvenciót
- összetettebb esetben hierarchikusan szervezett, egyszerű szerszámokból építsük fel a kísérleteinket
- jól átgondolt interfésszel (paraméter listával) szeparáljuk az algoritmikus és a megjelenítő, grafikus funkciókat
- kerüljük a for ciklusok használatát, amikor lehet tömb műveleteket és index-tömböket alkalmazzunk

1. Gyakorlat: ARMA jelek idő- és operátortartományi reprezentációinak grafikus vizsgálatára

A diszkrét idejű jelek (és rendszerek) idő- és operátortartománybeli leírásai közötti közlekedési eszköz a Z-transzformáció és az inverz Z-transzformáció.

A diszkrét idejű jeleket az időtartományban x(n) **számsorozatokkal** írjuk le (reprezentáljuk), az operátor tartományban pedig valamely körgyűrű felett analitikus X(z), $r1<|z|<r2$ komplex függvényekkel.

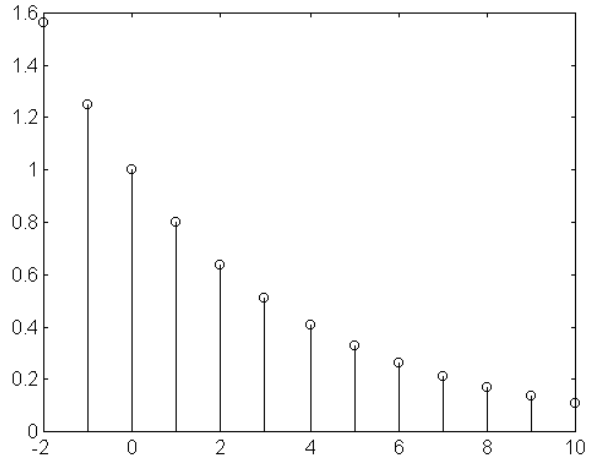
Az ARMA jelek - definíció szerint - azok a jelek, melyek Z-transzformáltja racionális törtfüggvény. Ebben az esetben az operátor(z)-tartományi teljes leírást megkapjuk a z változó racionális törtje nevező és számláló polinomjainak a gyökeivel. Tehát az operátor tartományban a grafikus leírás a **pólus-zérus ábra**.

Első lépésként készítsük el a sorozatokat ill. a racionálistörtek pólus-zérus ábrát megjelenítő MATLAB függvényeket.

Az xnplot.m file:

```
function xnplot(n,x);

% xnplot(n,x)
% n:  indexek vektora
% x:  mintak vektora

  for i = 1:3*length(n)
    nn(i) = n(1+floor((i-1)/3));
  end;
  xx = zeros(size(nn));
  for i = 1:length(n)
    xx(3*i-1) = x(i);
  end;
  plot(nn,xx,'-',n,x,'o');
```

Egy példa az idősor rajzoló függvény hívására:
» n=[-2:10]; x=0.8.^n; xnplot(n,x);



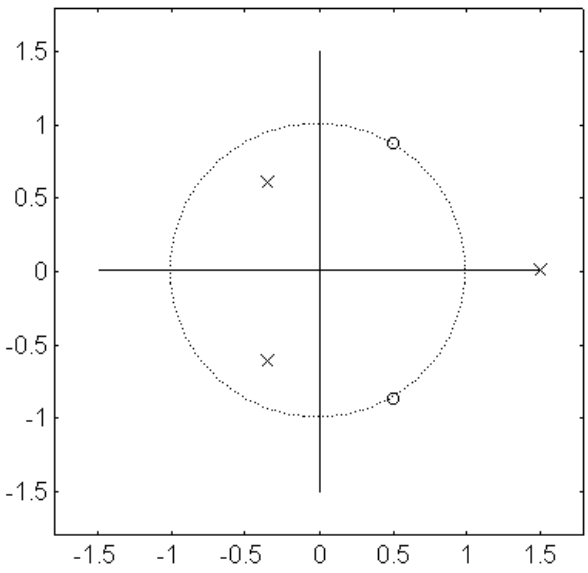
A pzabra.m file:

```
function pzabra(a,b);

%   pzabra(a,b);  pólus-zérus ábra rajzolása
%   a  a számláló (z)polinom együtthatói (csökkenő hatvány sorrendben)
%   b  a nevező (z)polinom együtthatói (csökkenő hatvány sorrendben)

 zk = roots(a);
 pk = roots(b);
 m = max([abs(zk);abs(pk);1]);
 kx = [-m,0,0,0,0,m];
 ky = [0,0,m,-m,0,0];
 korx = cos([0:2*pi/200:2*pi]);
 kory = sin([0:2*pi/200:2*pi]);
 plot(korx,kory,'-',kx,ky,'-', ...
       real(zk),imag(zk),'O',real(pk),imag(pk),'X');
 m = 1.2*m;
 axis([-m,m,-m,m]);
 axis('square');
```

A pzabra.m hívása:

```
» pzabra([1,-1,1],conv([1,0.7,0.49],[1,-1.5]));
```

és az eredménye:



A továbbiakan készítsük el racionális tört alakú Z-transzformáltak inverz Z-transzformáltját számító MATLAB függvényt.
Feltételezzük, hogy X(z)-nek adott az A(z) számláló, és B(z) nevező polinomja, továbbá adott a ρ konvergencia sugár, melyet tartalmazó körgyűrűhöz, mint konvergencia tartományhoz tartozó inverz z-transzformáltat akarjuk kiszámítani. A továbbiakban még feltételezzük, hogy X(z)-nek csak egyszeres pólusai vannak, így alkalmazhatjuk a parciális törtekre bontás egyszerű esetét.

$$X(z) = \frac{A(z)}{B(z)} = \frac{\sum_{i=0}^{M} a_i \cdot z^i}{\sum_{j=0}^{N} b_j \cdot z^j} = \frac{a_M \cdot \prod_{i=1}^{M} (z - z_i)}{b_N \cdot \prod_{j=1}^{N} (z - p_j)},$$

ahol a z_i-k X(z) zérusai és p_j-k X(z) pólusai.

Egyszeres pólusok esetén X(z) parciális törtek összegeként is felírható:

$$X(z) = \frac{A(z)}{B(z)} = \sum_{i=1}^{N} \frac{r_i}{z - p_i} + \sum_{i=0}^{M-N} k_i z^i ,$$

ahol $r_i = \operatorname{res}_{z=p_i}\{X(z)\} = \frac{A(p_i)}{B'(p_i)}$

továbbá k_i az (M-N)-ed fokú egészrész polinom együtthatói, melyek nullák, ha X(z) valódi tört.

Az r_i p_i k_i számok vektorai a MATLAB-ban a residue utasítással határozhatóak meg.

Mivel a z-transzformáció lineáris művelet, a fenti összeg tagjainak külön-külön vehetjük az inverz z-transzformáltját, és az így kapott részsorozatokat a végén összegezhetjük.

Ezeket a részsorozatokat a geometriai sorok összegére vonatkozó tétel segítségével határozzuk meg:

$$\mathcal{Z}^{-1}\left\{\frac{r_i}{z - p_i}, |z| = \rho\right\} = \begin{cases} r_i \mathcal{Z}^{-1}\left\{ z^{-1} \frac{1}{1 - \frac{p_i}{z}} \right\} = r_i \mathcal{Z}^{-1}\left\{ {p_i}^0 z^{-1} + {p_i}^1 z^{-2} + {p_i}^2 z^{-3} + \ldots \right\} & |\rho| > |p_i| \\ -r_i \frac{1}{p_i} \mathcal{Z}^{-1}\left\{ \frac{1}{1 - \frac{z}{p_i}} \right\} = -r_i \mathcal{Z}^{-1}\left\{ {p_i}^{-1} + {p_i}^{-2} z^1 + {p_i}^{-3} z^2 + \ldots \right\} & |\rho| < |p_i| \end{cases}$$

azaz,

$$\mathcal{Z}^{-1}\left\{\frac{r_i}{z - p_i}, |z| = \rho\right\} = \begin{cases} \begin{cases} r_i {p_i}^{-n-1}, & n > 0, \\ 0, & n < 1, \end{cases} & \text{ha} \quad |\rho| > |p_i| \\ \begin{cases} 0, & n > 0, \\ -r_i {p_i}^{-n-1}, & n < 1, \end{cases} & \text{ha} \quad |\rho| < |p_i| \end{cases}$$

A fenti meggondolások alapján megírhatjuk az alábbi invzp függvényt. (A függvény neve végén levő p betűvel azt hangsúlyozzuk, hogy az a és b input polinomok z pozitív hatványaihoz tartoznak.)

```
function x=invzp(a,b,ro,nl,nu);

%function x=invz(a,b,ro,nl,nu);
% inverz Z transzformacio (z pozitiv hatvanyai szerinti tortre)
% a: szamlalo a(1)*z^M + a(2)*z^(M-1)+...+a(M+1)
% b: nevezo   b(1)*z^N + b(2)*z^(N-1)+...+b(N+1)
% ro:konvergencia tartomanybeli sugar
% nl, nu : a kiszamitando idotengely feletti index-intervallum
% x : eredmeny: az nl...nu indexekhez tartozo idomintak sorozata

[res,pol,k]=residue(a,b);
if length(pol) == 0 k=a/b(1); end;
it=[nl:nu];                          % ido index tomb
mask1=sign(max(zeros(size(it)),it));
% ro-n beluli polus nemnulla mintainak indexei
mask2=-(ones(size(it))-mask1);
% ro-n kivuli polus nemnulla mintainak indexei

x=0*it;
for i=1:length(pol)
  xi=pol(i)*ones(size(it));
  xi=xi.^(it-1);
  if abs(pol(i)) <= ro xi=mask1.*xi;
  else                 xi=mask2.*xi;
  end;
  x=x+res(i)*xi;
end % for

nk=length(k);
if nk>0    % nem valodi tort eseten tovabbi kiegeszites
  [iw,xw]=addi(it,x,[-(nk-1):0],k);
  [iw,x]=multi(iw,xw,it,ones(size(it)));
end
```

A fenti megoldásban, az egészrészhez tartozó FIR szűrő impulzusválasz sorozatának meghatározásánál, felhasználtuk az alábbi két segédfüggvényt, melyet indekszelt vektorok összeadását és szorzását végzik:

```
function [i3,x3]=addi(i1,x1,i2,x2);

%function [i3,x3]=addi(i1,x1,i2,x2);
% indexelt vektorok index-helyes osszeadasa
% i1, i2, i3: indexek folyamatosan felsorolt vektorai
% x1, x2, x3: indexekhez tartozo mintak vektorai
%             indextartomanyon kivul nulla mintak feltetelezettek
% i3: i1 es i2 unioja

i3=[min(i1(1),i2(1)):max(i1(length(i1)),i2(length(i2)))];

x3=zeros(size(i3));
x3(i1-i3(1)+1)=x3(i1-i3(1)+1)+x1;
x3(i2-i3(1)+1)=x3(i2-i3(1)+1)+x2;
```

```
function [i3,x3]=multi(i1,x1,i2,x2);

%function [i3,x3]=multi(i1,x1,i2,x2);
% indexelt vektorok index-helyes osszeszorzasa
% i1, i2, i3: indexek folyamatosan felsorolt vektorai
% x1, x2, x3: indexekhez tartozo mintak vektorai
%             indextartomanyon kivul nulla mintak feltetelezettek
% i3: i1 es i2 metszete

il=max(i1(1),i2(1));
iu=min(i1(length(i1)),i2(length(i2)));

if iu >= il
  i3=[il:iu];
  i3s=[1:length(i3)];
  x3=x1(i3(1)-i1(1)+i3s).*x2(i3(1)-i2(1)+i3s);
end;
```

Az eddigi eszközeink kényelmes használatát támogatandó, írjuk meg a grafikát vezérlő ginvzp.m file-t.

```
function x=ginvzp(a,b,r,nl,nu,ny,nx,ni)

%function x=ginvzp(a,b,r,nl,nu,ny,nx,ni);
% inverz Z transzformacio grafikus outputtal
% a: szamlalo a(1)*z^M + a(2)*z^(M-1)+...+a(M+1)
% b: nevezo   b(1)*z^N + b(2)*z^(N-1)+...+b(N+1)
% r: konvergencia tartomanybeli sugar
% nl, nu : a kiszamitando idointervallum
% ny,nx,ni: subplot(ny,nx,ni) vezerlo adatok

subplot(ny,nx,ni);

pzabra(a,b);
%szoveg=['Konvergencia tartomanybeli kontursugar: ',num2str(r)];
szoveg=['ro=',num2str(r)];
xlabel(szoveg);
title('Polus-Zerus kep');
it=[nl:nu];
x=invzp(a,b,r,nl,nu);

subplot(ny,nx,ni+nx);

xnplot(it,x);
xlabel('Idosor');
subplot(111);
```

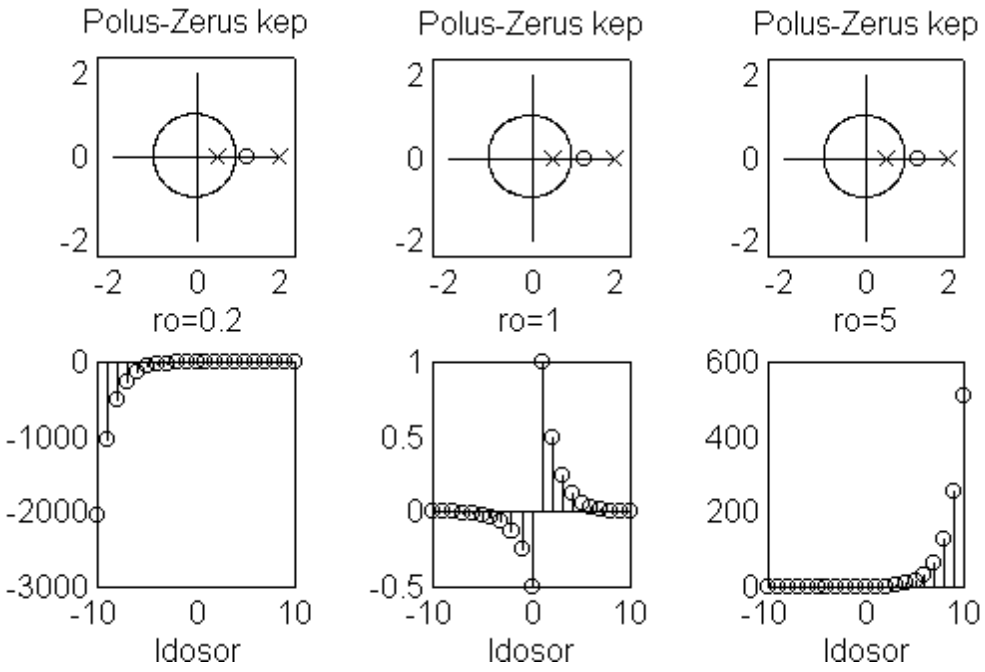
Ellenőrző és illusztratív példaként vizsgáljuk meg az

$$X(z) = \frac{2z - 2.5}{(z - 0.5)(z - 2.0)} = \frac{1}{z - 0.5} + \frac{1}{z - 2.0}$$

esetét ρ = 0.2, 1.0 és 5.0 konvergencia sugarak esetére.

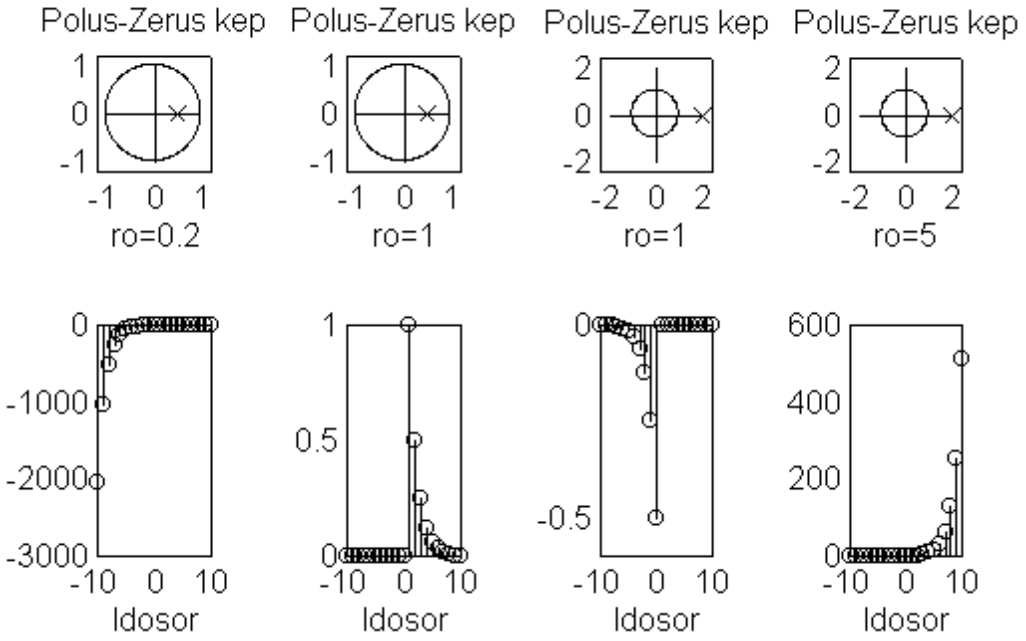
A három különböző konvergencia gyűrűvel kapott eredmények:

```
ginvzp([2,-2.5],conv([1,-0.5],[1,-2]),0.2,-10,10,2,3,1);
ginvzp([2,-2.5],conv([1,-0.5],[1,-2]),1.0,-10,10,2,3,2);
ginvzp([2,-2.5],conv([1,-0.5],[1,-2]),5.0,-10,10,2,3,3);
```



Tanulságos összehasonlításként itt vannak az eredmények a csak egy (az egységsugarú körön belüli vagy kívüli) pólus esetére is, különböző konvergencia sugarak mellett:

```
ginvzp([1],conv([1,-0.5],[1]),0.2,-10,10,2,4,1);
ginvzp([1],conv([1,-0.5],[1]),1.0,-10,10,2,4,2);
ginvzp([1],conv([1,-2.0],[1]),1.0,-10,10,2,4,3);
ginvzp([1],conv([1,-2.0],[1]),5.0,-10,10,2,4,4);
```



2. Gyakorlat: Idő- és frekvenciatartományi reprezentációk grafikus vizsgálatára

Készítsük el azt a t2f.m nevű szerszámot, mellyel egy fc órajelű, nt vektorban levő idő-indexek sorozatához tartozó xt mintavektorok komplex sx Fourier transzformáltját számítjuk ki a frekvenciatengelyen tetszőlegesen előírható fa-ff intervallum felett. Az fft eljáráson alapuló számítás frekvencia tengely feletti $\Delta f = fc/nfft$ felbontását az xt mintavektor hossza határozza meg!

t2f.m:

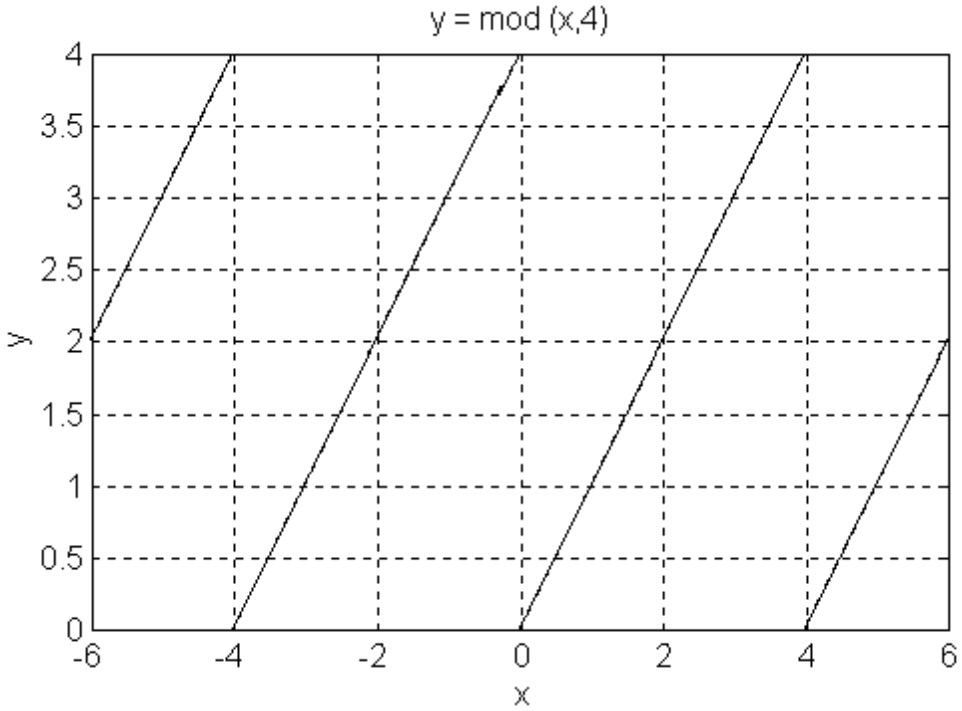
```
function [ft,sx]=t2f(fc,nt,xt,fa,ff);

% Time to Frequency
%[ft,sx]=t2f(fc,nt,xt,fa,ff);
% fc: orajel
% nt, xt: ido-index vektor es minta vektor
% fa, ff : a szamitando frekvencia sav
% sx-ben a komplex spektrum mintak az ft beli frekvenciak szerint

 % FFT pontszam meghatarozasa
 nfft = 2^ceil(log(length(xt))/log(2));

 sx = fft(xt,nfft);

% a szamitando frekvencia-index intervallum
 nf = [round(nfft*fa/fc):round(nfft*ff/fc)];
 ft = nf*fc/nfft;

% a periodikus spektrum nf feletti resze
sx = sx(mod(nf,nfft)+1);

% az xt mintak idobeli helyzetenek megfelelo fazis korrekcio
sx = sx .* exp(-j*2*pi*nt(1)*ft/fc);
```

A fenti eljáráshoz felhasználtuk az alábbi moduló függvényt:
mod.m:

```
function y = mod(x,a);
% y = mod(x,a)
y = x - a * floor(x/a);
```

Az egyébként sokszor használandó moduló függvényünk tesztelése, illusztrálása az alábbiak szerint történhet:

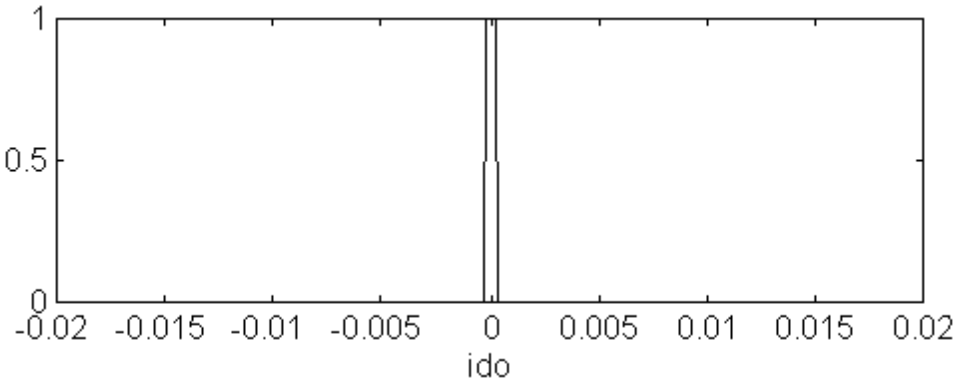
```
» plot([-6:0.01:6],mod([-6:0.01:6],4),'.'); grid;
» xlabel('x'); ylabel('y'); title('y = mod (x,4)');
```

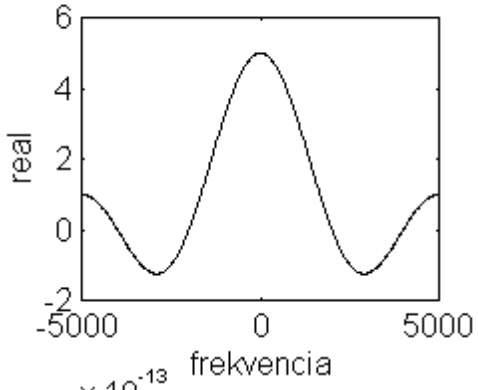


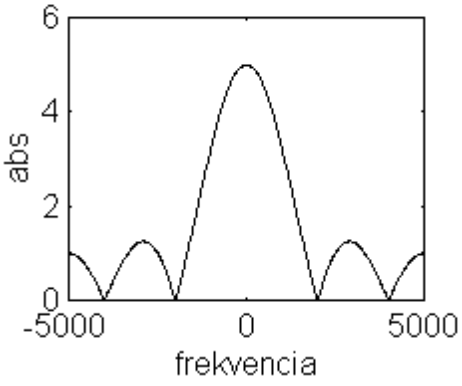
Elkészített eszközünket használjuk fel az időtartományi ablakolás frekvencia-tartományi következményeinek vizsgálatára.

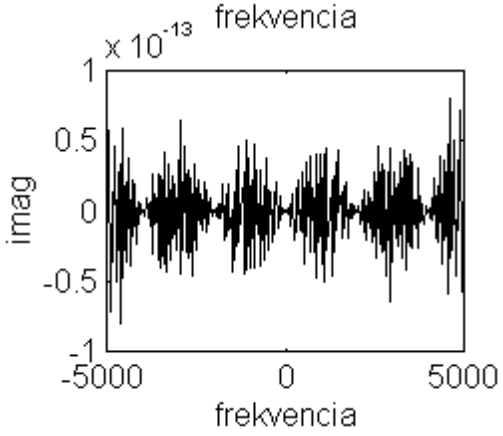
pelda21.m

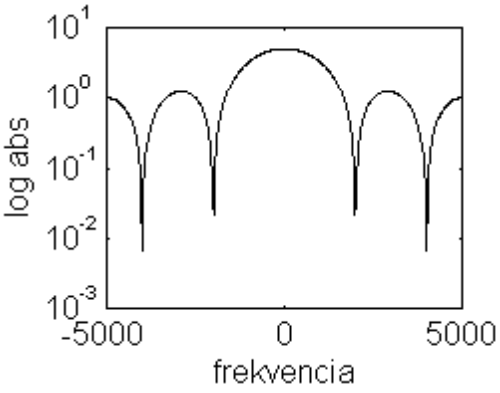
```
naa=-2;  naf=2;   %ablak index hatarok
      % a jel adatai
fc=10000;  T=1/fc;   % orajel
na=-200;  nf=200;  nt=[na:nf];
tt=T*nt;  %ido tengely

xat=sign(nt-(naa-0.1));
xft=sign((naf+0.1)-nt);

xt=(xat+xft)/2;

figure(1);
subplot(211);
plot(tt,xt,'-'); xlabel('ido');
[ft,sx]=t2f(fc,nt,xt,-fc/2,fc/2);
subplot(111);

figure(2);
subplot(221);
plot(ft,real(sx),'-'); xlabel('frekvencia'); ylabel('real');
subplot(223);
plot(ft,imag(sx),'-'); xlabel('frekvencia'); ylabel('imag');
subplot(222);
plot(ft,abs(sx),'-'); xlabel('frekvencia');  ylabel('abs');
subplot(224);
semilogy(ft,abs(sx),'-'); xlabel('frekvencia'); ylabel('log abs');
subplot(111);
```


1
0.5
0
-0.02 -0.015 -0.01 -0.005 0 0.005 0.01 0.015 0.02
ido


6
4
2
0
-2
real
-5000 0 5000
frekvencia

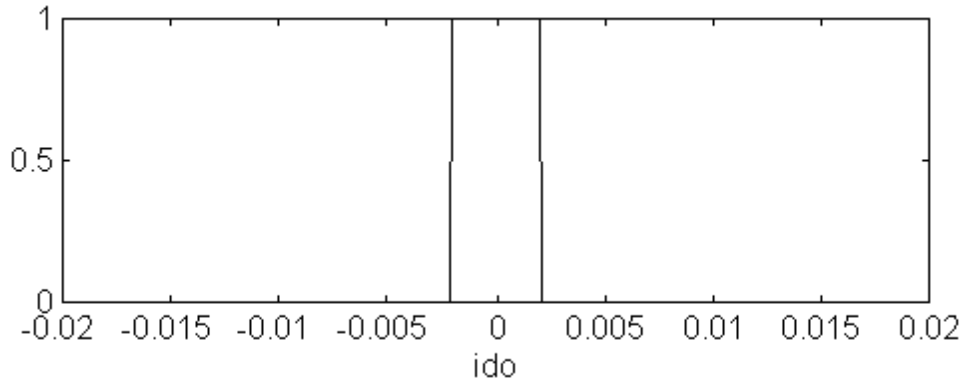

6
4
2
0
abs
-5000 0 5000
frekvencia

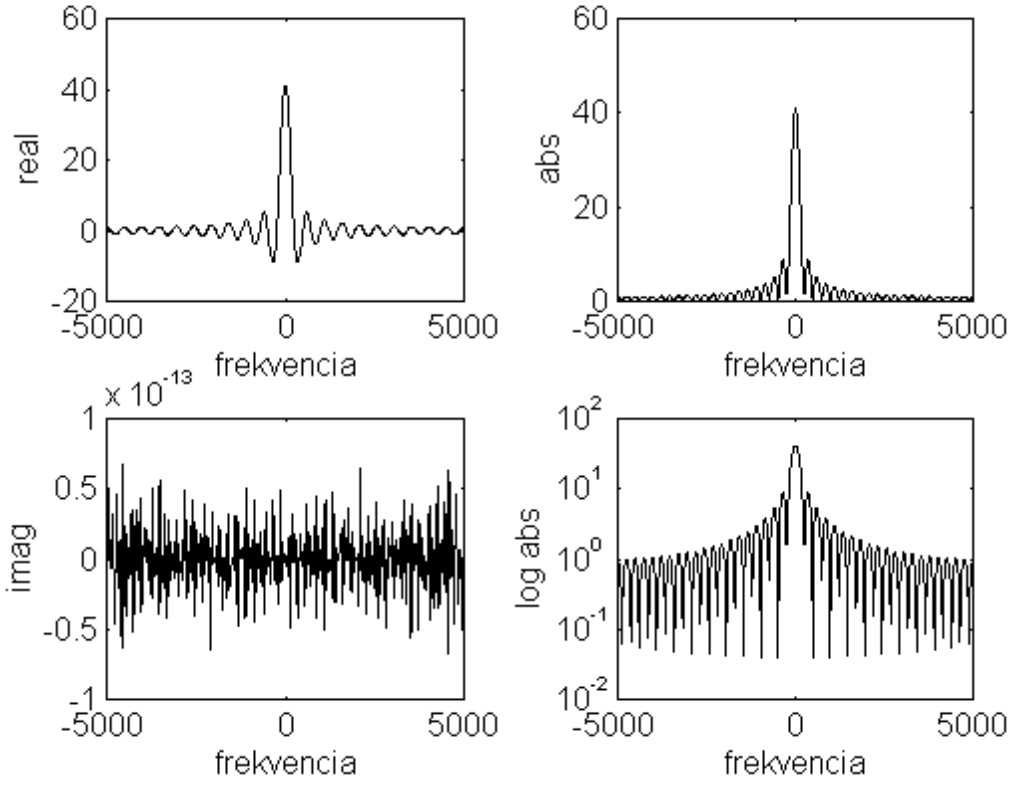

x 10^-13
1
0.5
0
-0.5
-1
imag
-5000 0 5000
frekvencia


10^1
10^0
10^-1
10^-2
10^-3
log abs
-5000 0 5000
frekvencia

Szélesebb ablak:

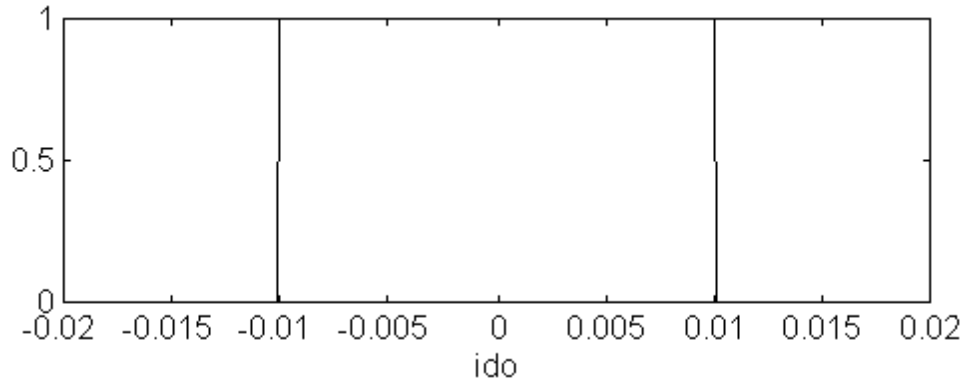
```
naa=-20;  naf=20;   %ablak index hatarok
```

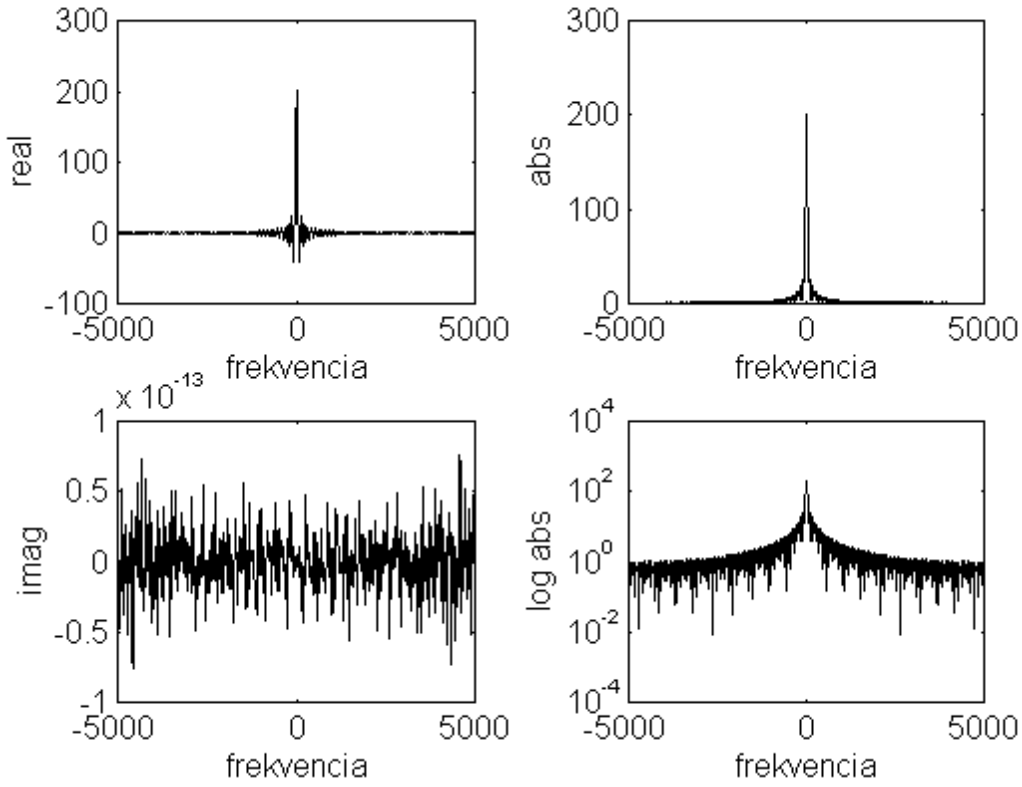




201 minta szélességű ablak esete:

```
naa=-100;  naf=100;   %ablak index hatarok
```



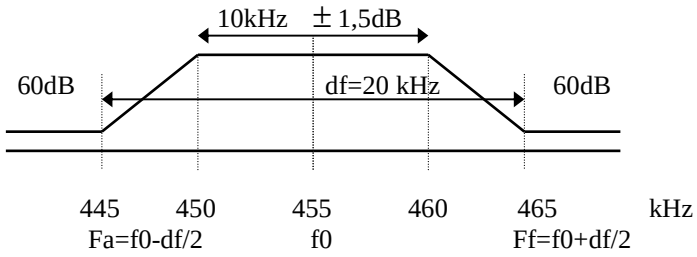


Megjegyzések:
- A képzetes részre kapott eredmény a numerikus pontosságra vonatkozó tájékoztatásként értelmezhető
- További kisérletek végezhetőek nem páros függvényű ablakkal
- További vizsgálatok: fazis, futási idő karakterisztikák
- További vizsgálat: ablakolt szinusz spektruma, rövid idejű burst hullam vizsgálata

3. Gyakorlat: Sávhatárolt KF jel alul-mintavételezési frekvenciájának meghatározása

A rádiótechnikában gyakran előforduló eset, hogy távközlő csatorna jelét valamely középfrekvenciás (KF) sávban dolgozzák fel. A feladatunk, hogy meghatározzuk, melyek az adott csatornaszűrő által meghatározott lehetséges mintavételi frekvenciák.

Példaként tekintsük az alábbi sávszűrő karakterisztikát:

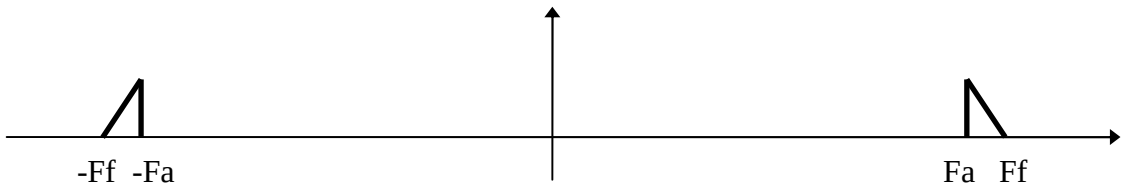


f0=455 kHz
df=20kHz

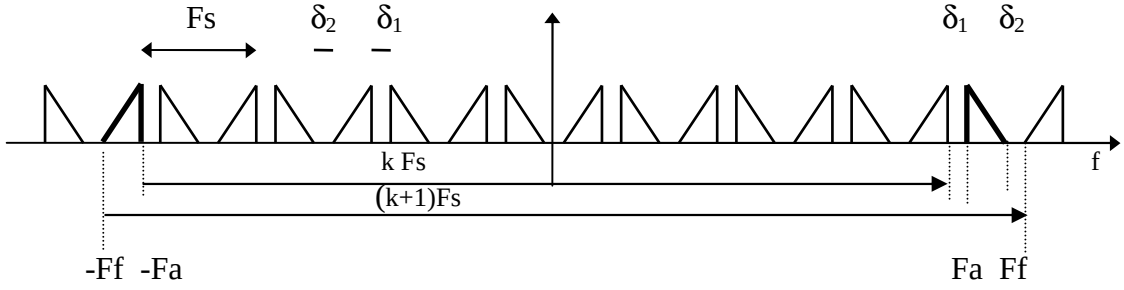
Készítsünk Matlab programot, mely az f0, df paraméterekből a frekvenciatengely fsa ... fsf intervalluma felett kirajzolja a lehetséges mintavételi frekvenciák indikátor függvényét (1 az érték lehetséges frekvencia felett, nulla egyébként)

A program megírása előtt tekintsük át a vonatkozó összefüggéseket:

A sávszűrő után maradó spektrum, tehát csak az Fa – Ff = df intervallumban tartalmaz komponenseket:



Az Fs frekvenciájú mintavételezés utáni spektrum:



A mintavett jel periodikus spektrumának átlapolódás mentességéhez szükséges az alábbi egyenlőtlenség rendszer teljesülése:

$\delta_1 , \delta_2 > 0$; k egész ;

$-Fa + k\,Fs + \delta_1 = Fa$
$-Ff + (k+1)\,Fs - \delta_2 = Ff \quad \rightarrow \quad Fs > 2\,(Ff - Fa) = 2\,df$

$k < 2\,Fa/Fs, \quad \rightarrow \quad kmax = floor(2\,Fa / Fs)$
$k > (2\,Ff - Fs)/Fs, \quad \rightarrow \quad kmin = ceil((2\,Ff - Fs)/Fs) = ceil\,(2Fa / Fs - (1-2\,df/Fs))$

Adott Fa, Ff értékek mellett a fenti egyenlőtlenségrendszernek csak olyan Fs mintavételi frekvencia értékeknél van egész megoldása k-ra, ahol

$$kmin(Fs) >= kmax(Fs).$$

Mivel $0 < (1-2\,df/Fs) < 1$, ezért kmin vagy egyenlő kmax-szal, vagy eggyel nagyobb, tehát a keresett indikátor függvény:

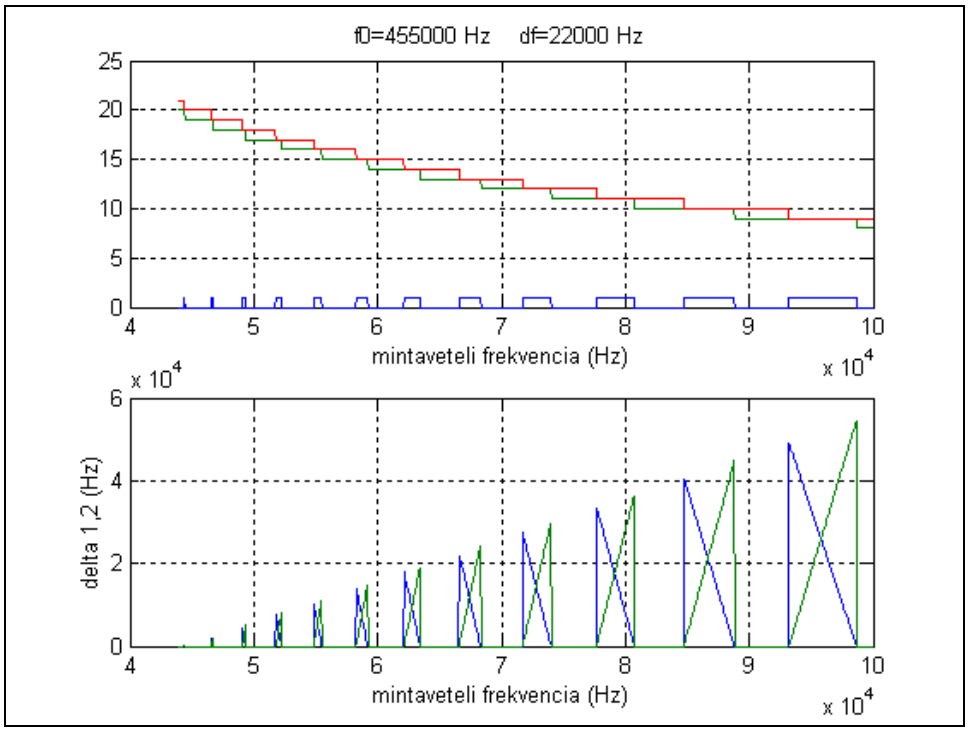
$$1 + kmax(Fs) - kmin(Fs)$$

Ezek után a Matlab program:

```
function subsamp(f0,df,fsa,fsf);

fa=f0-df/2; ff=f0+df/2;

if fsa < 2*df  fsa=2*df;
end;
fs=[fsa:(fsf-fsa)/1000:fsf];
kmax=floor(2*fa./fs);
kmin=ceil(2*ff./fs-1);
ke=1+ kmax-kmin;    % k engedelyezett indikatora

figure(1)
subplot(211)
    plot(fs,ke,fs,kmax,fs,kmin); grid;
    title(['f0=',num2str(f0),' Hz    df=',num2str(df),' Hz']);
xlabel('mintaveteli frekvencia (Hz)');
%ylabel('kmax(kek),  knim(zold)');

delta1=ke.*(2*fa-kmin.*fs);
delta2=-ke.*(2*ff-(kmin+1).*fs);

subplot(212)
plot(fs, delta1, fs, delta2); grid;
xlabel('mintaveteli frekvencia (Hz)');
ylabel('delta 1,2 (Hz)');
subplot(111);
```

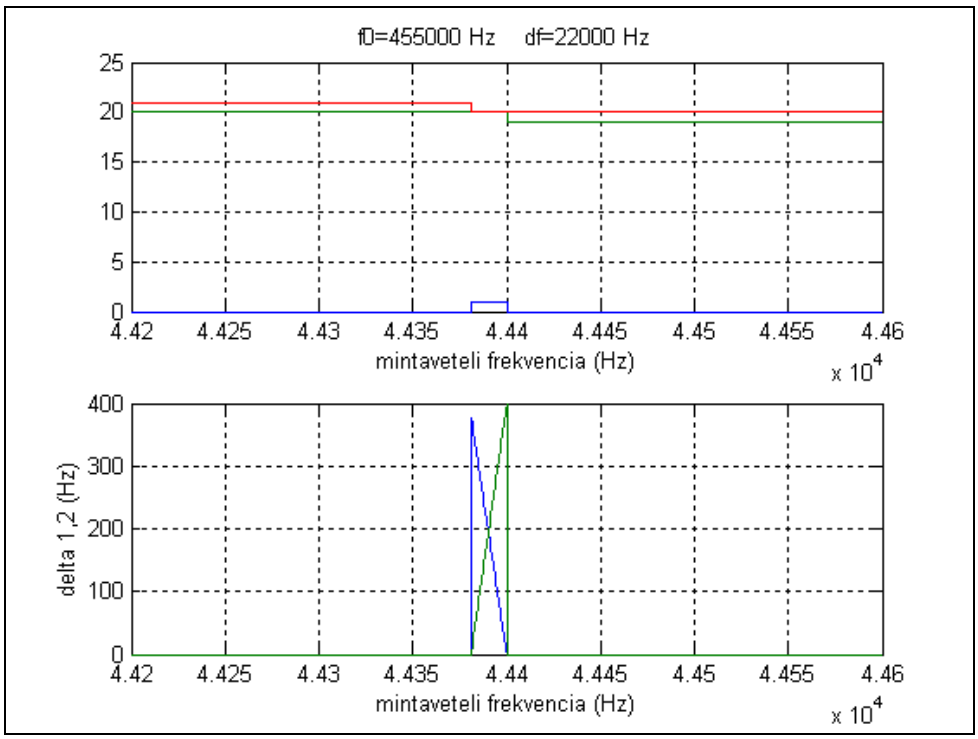
Futtatási eredmények:

clear; subsamp(455000,22000,44000);



Kinagyítva a lehetséges legkisebb frekvencia környezete:

subsamp(455000,22000,44200,44600);



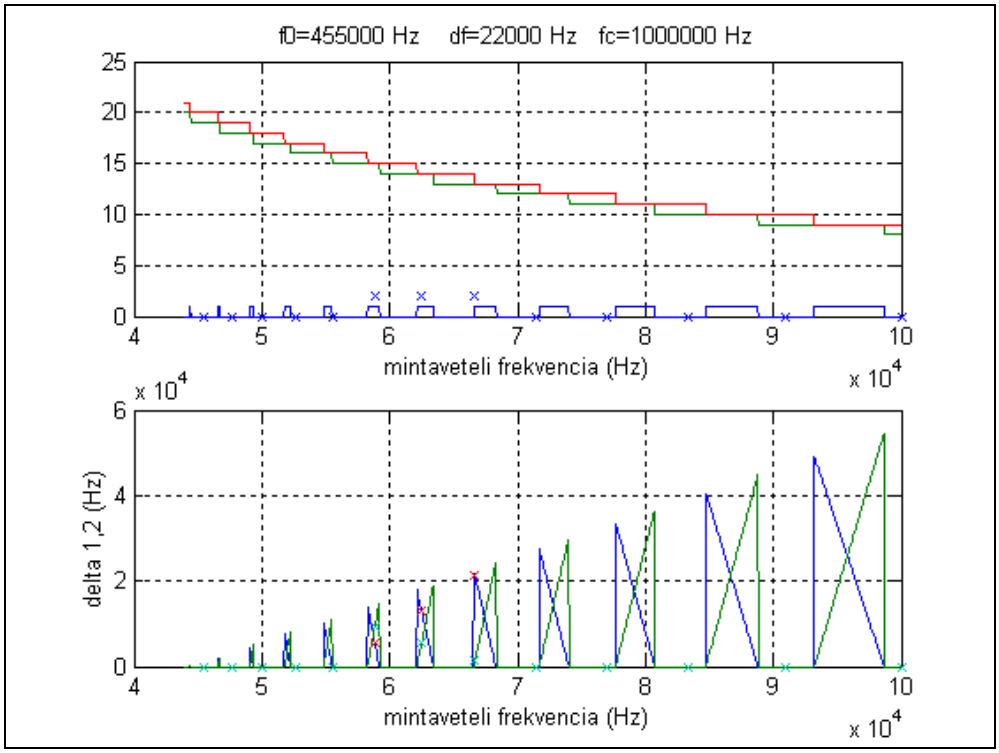
A továbbiakban vizsgáljuk meg, hogyan kell módosítani a számítást, ha figyelembe akarjuk venni, hogy nem folytonos idejű (analog) jel mintavételezési frekvenciáját kell meghatároznunk, hanem már egy fc előzetes mitavételi frekvenciáról (nagyfrekvencián digitalizált jel) akarunk áttérni egy kisebb alul-mintavételezési frekvenciára. Ebben az esetben a lehetséges Fs mintavételi frekvenciák, nem a folytonos értékek felett értelmezett intervallumokban lehetségesek, hanem csak a diszkrét Fs = fc/N szubharmónikus frekvenciák közül azok, melyek a korábban megismert egyenlőtlenség rendszert kielégítik.

A módosított MATLAB program:

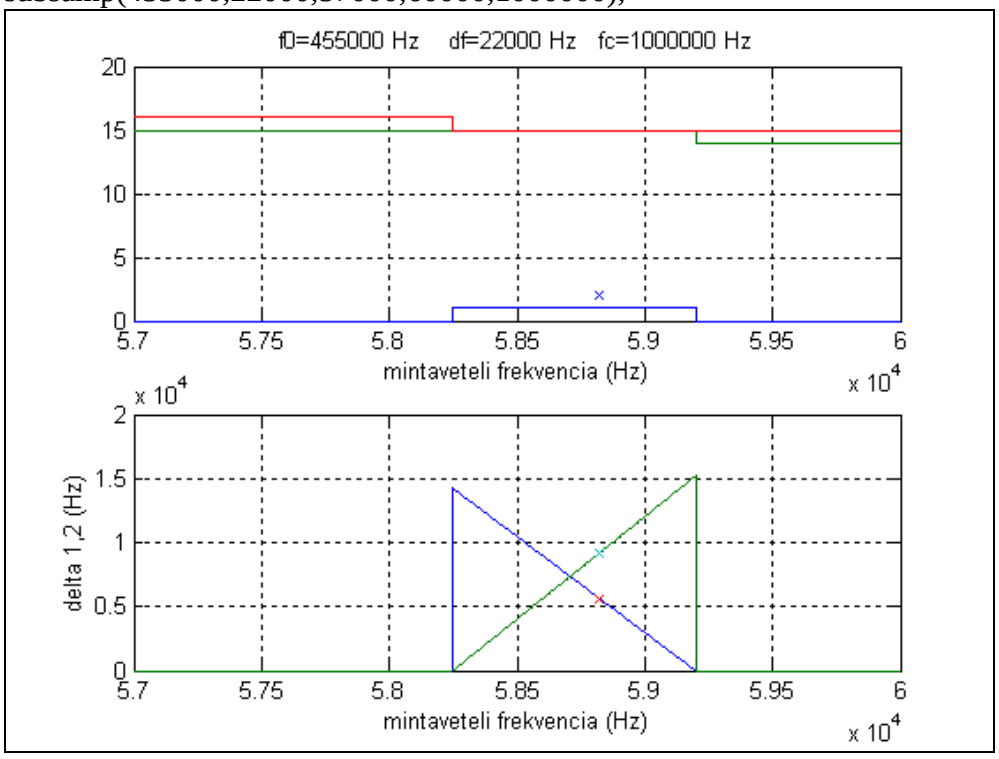
```
function subsamp(f0,df,fsa,fsf, fc);

fa=f0-df/2; ff=f0+df/2;
if fsa < 2*df  fsa=2*df; end;
fs=[fsa:(fsf-fsa)/1000:fsf];
kmax=floor(2*fa./fs);
kmin=ceil(2*ff./fs-1);
ke=1+ kmax-kmin;    % k engedelyezett indikatora
cim=['f0=',num2str(f0),' Hz    df=',num2str(df),' Hz'];
figure(1)
subplot(211)
if nargin > 4   % diszkret frekvenciak esete
    nmax=floor(fc/fsa);    nmin=ceil(fc/fsf);
    fsd=fc./[nmax:-1:nmin];
    kmaxd=floor(2*fa./fsd);
    kmind=ceil(2*ff./fsd-1);
    ked=(1+kmaxd-kmind); % k engedelyezett diszkret indikatora

    plot(fs,ke,fs,kmax,fs,kmin,fsd,2*ked,'x'); grid;
    title([cim,'  fc=',num2str(fc),' Hz']);
else
    plot(fs,ke,fs,kmax,fs,kmin); grid;
    title(cim);
end;
xlabel('mintaveteli frekvencia (Hz)');

delta1=ke.*(2*fa-kmin.*fs);
delta2=-ke.*(2*ff-(kmin+1).*fs);
subplot(212)
if nargin > 4
    delta1d=ked.*(2*fa-kmind.*fsd);
    delta2d=-ked.*(2*ff-(kmind+1).*fsd);
    plot(fs,delta1,fs,delta2,fsd,delta1d,'x',fsd,delta2d,'x');
grid;
else
    plot(fs, delta1, fs, delta2); grid;
end
xlabel('mintaveteli frekvencia (Hz)');
ylabel('delta 1,2 (Hz)');
subplot(111);
```

A kapott eredmények, a KF-jel előzetes 1MHz mintavételezési frekvencián történt digitalizálása esetén:

subsamp(455000,22000,44000,100000,1000000);



subsamp(455000,22000,57000,60000,1000000);



4. Gyakorlat: Egyenletes és logaritmikus kvantálók vizsgálata

Egyenletes, szimmetrikus, r bites kvantálási karakterisztikát az alábbi függvénnyel valósíthatunk meg:

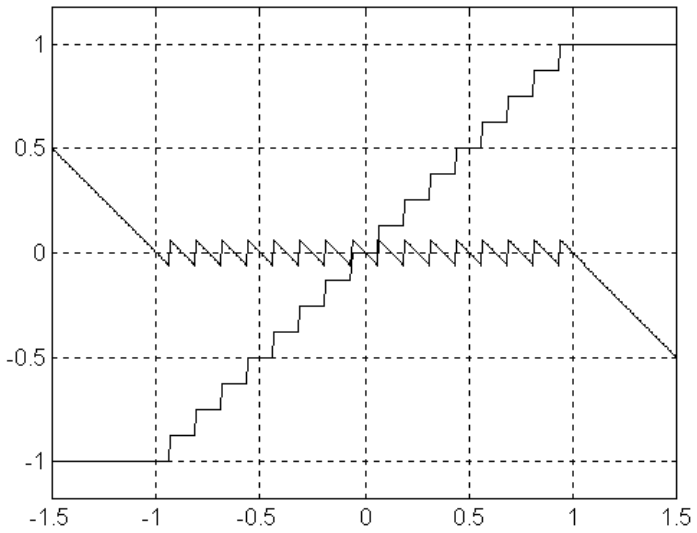
kvant_e.m

```
function y = kvant_e(x,r,xmin,xmax);
%  y = kvant_e(x,r,xmin,xmax);
%  Egyenletes kvantalo
%     x: bemeneti mintak
%     r: r bites kvantalas
%     xmin-xmax: felett egyenletes kvantalo, |hiba| < q/2
%     y: kvantalt kimeneti mintak
  l = 2^r;              % dontesi intervallumok, szintek szama
  q = (xmax-xmin)/l;               % a kvantalasi lepcso
% nagy jelu mukodes
  y = min(x,(xmax-q/2)*ones(size(x)));
% kis jelu mukodes
  y = max(y,(xmin+q/2)*ones(size(y)));
% kerekites
  y = xmin+q/2+q*round((y-(xmin+q/2))/q);
```

4 bites kvantáló esetére a lépcsős kvantálási karakterisztika és a fűrész alakú hiba karakterisztikák az alábbiak szerint jeleníthető meg:

pelda41.m

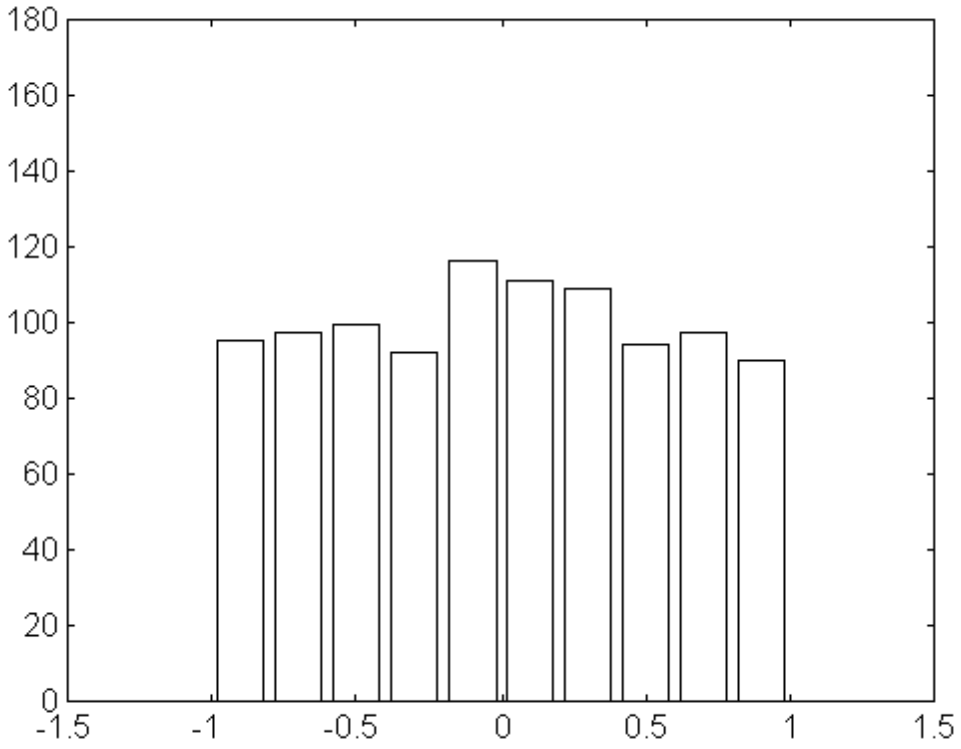
```
% Egyenletes kvantalo kvantalasi es hiba karakterisztikaja;
figure(1);
r=4; d=1;
x=1.5*[-d:d/300:d];
y=kvant_e(x,r,-d,d);
e=y-x;
plot(x,y,'-',x,e,'-'); grid;
axis('equal');
```



Kvantálók mérésére gyakran használunk szélessávú, véletlen jelet. 1000 darab, [-1...+1] intervallum felett egyenletes eloszlású független véletlen szám x vektorát és a hisztogramját az alábbiak szerint álithatjuk elő:

pelda41.m (folytatás)

```
% mérő jel
x=2*(rand(1,1000)-0.5);
figure(2);
hist(x,10); v=axis; axis(1.5*v);
```



A kvantálási torzítás mértékének számítására az alábbi függvényt használhatjuk:

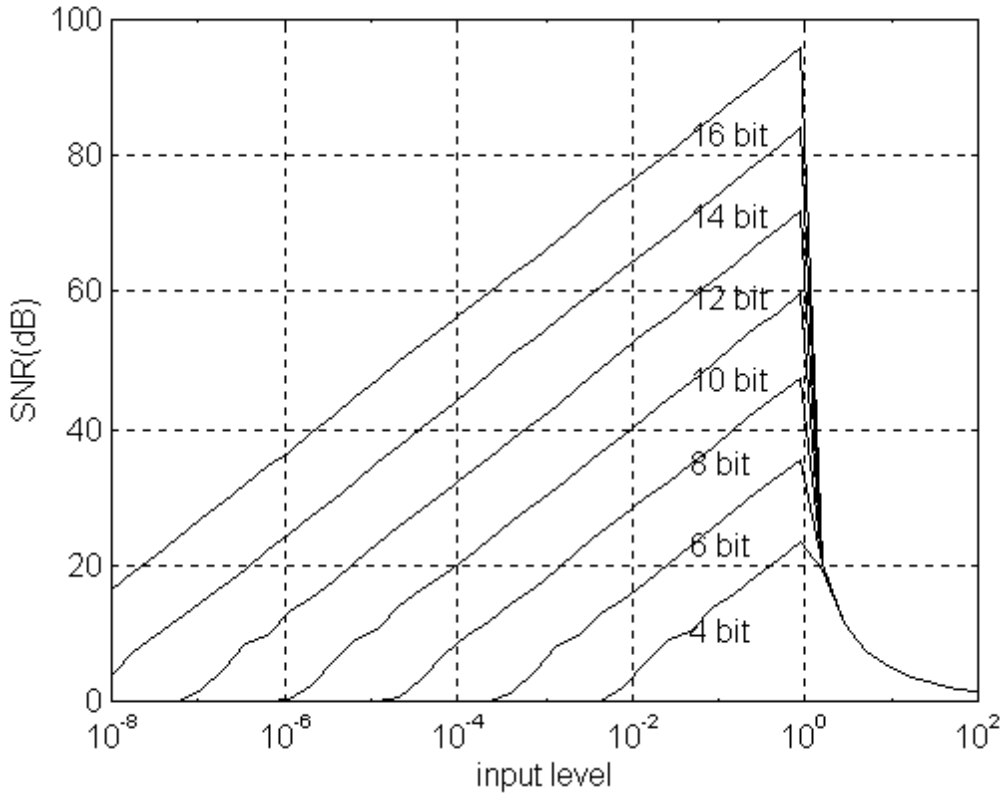
snrdb.m:

```
function s=snrdb(x,y,d);
%s=snrdb(x,y,d);
%Az x es az y mintak kozotti totorzitas jelzajviszonya dB-ben.
%Az osszetartozo mintak y-ban d-utemmel kesnek

lx=length(x);
ly=length(y);
l=min(lx,ly-d);
e=y(1+d:l+d)-x(1:l);
s=(sum(x.*x)/lx)/(sum(e.*e)/l);
s=10*log10(s);
```

A különböző felbontású (különböző bitszámú) egyenletes kvantálók kvantálási torzításának a bemeneti jelszinttől való függését az alábbiak szerint vizsgálhatjuk meg:

pelda41.m (folytatás)

```
ns=40;                                  % a bemeneti jelszintek szama
szigma=logspace(-4,1,ns);  % a bemeneti jelszintek ertekei

% a vizsgalando kvantalo-felbontasok:2,4,6,8,10,12,14,16 bit
bits=[4:2:16];

for j=1:length(bits),
  for i=1:ns,
    xs=szigma(i)*x;
    ys=kvant_e(xs,bits(j),-1,1);
    snr(j,i)=snrdb(xs,ys,0);      % jel/zaj meghatarozasa
  end;     %for i
end;  %j

figure(3);
semilogx(szigma.*szigma,snr,'-'); grid,
ylabel('SNR(dB)'); xlabel('input level');
for j=1:length(bits),
  text(szigma(27)*szigma(27),snr(j,27),[num2str(bits(j)),' bit']);
end;
```



Adott kvantálási dinamikatartomány felett szint független jel-zaj viszonyt logaritmikus kvantálóval érhetünk el.

kvant_l.m

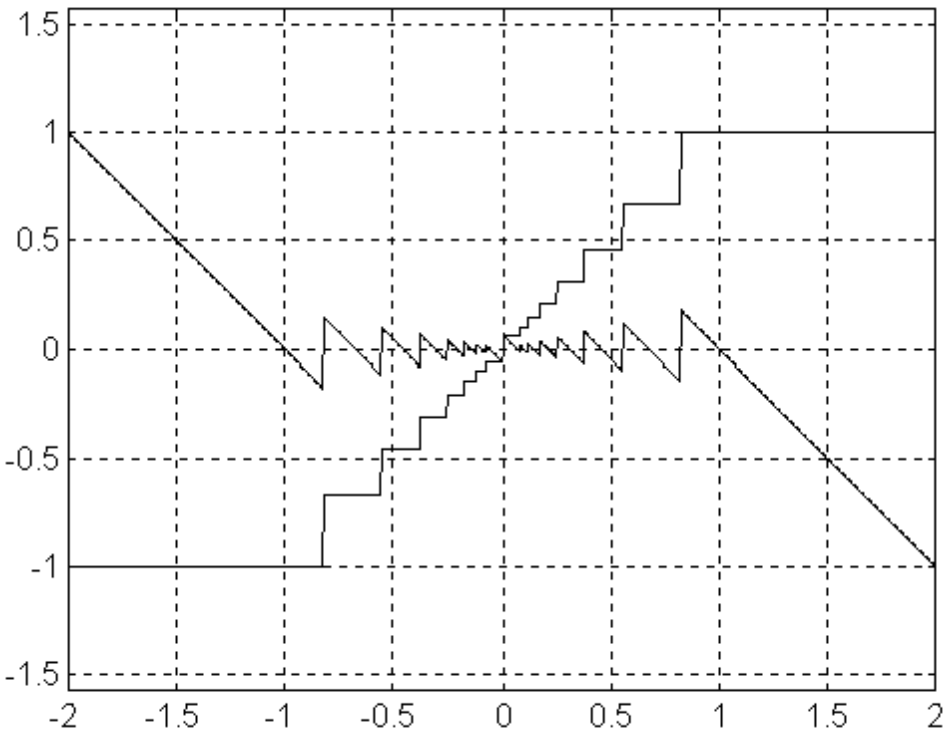
```
function y = kvant_l(x,r,d,ddb);
%  y = kvant_l(x,r,d,ddb);
%  logaritmikus kvantalo
%     x: bemeneti mintak
%     r: r bites kvantalas
%     d: +-d felett logaritmikus kvantalas
%     ddb: a logaritmikus kvantalo dinamikatartomanya dB-ben
%     y: kvantalt kimeneti mintak

l = 2^(r-1);      % dontesi intervallumok, szintek szama
xmin=10^(-ddb/20);
ql = (0-log(xmin))/(l-1); % a kvantalasi lepcso

y = min(abs(x/d),ones(size(x)));    % nagy jelu mukodes
y = max( y, xmin*ones(size(x)));    % kis jelu mukodes
yl=log(y);

y = (round((yl)/ql))*ql;            % kerekites

y = d* sign(x).*exp(y);
```

Természetesen a logaritmikus karakterisztika szerinti kvantálást is visszavezethetjük egyenletes kvantálásra a log() kompander- és exp() expander-karakterisztikák használatával:

```
function y = kvant_le(x,r,d,ddb);
%  y = kvant_le(x,r,d,ddb);
%  szimmetrikus, logaritmikus kvantalo
%     x: bemeneti mintak
%     r: r bites kvantalas
%     d: +-d felett logaritmikus kvantalas
%     ddb: a logaritmikus kvantalo dinamikatartomanya dB-ben
%     y: kvantalt kimeneti mintak

xmax=d;   xmin=d*10^(-ddb/20);
lxmax=log(xmax); lxmin=log(xmin);

ly=kvant_e(log(abs(x)),r-1,lxmin, lxmax);

y=sign(x).*exp(ly);
```

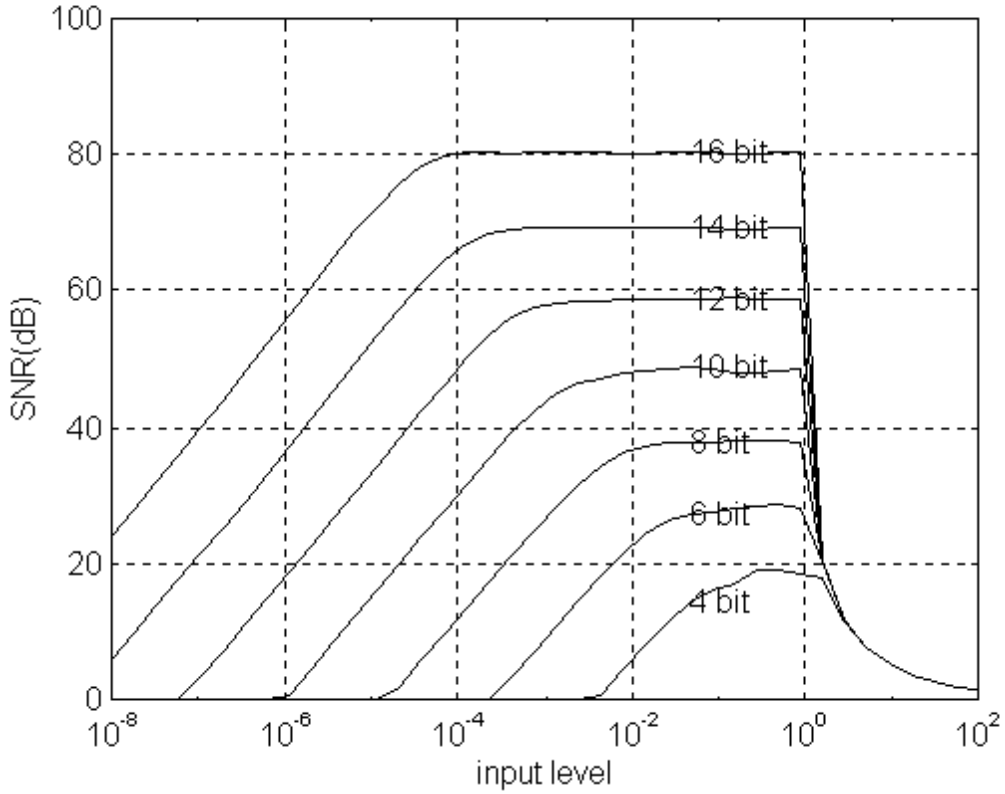
A logaritmikus kvantáló kvantálási- és torzítási karakterisztikájának megjenítése:

```
r=4; d=1; ddb=r*6;
x=2*[-d:d/500:d];
y=kvant_l(x,r,d,ddb); e=y-x;
plot(x,y,'w-',x,e,'w-'); grid;
axis('equal');
```



A jel-zaj viszony szint-függésének szimulációs vizsgálata:

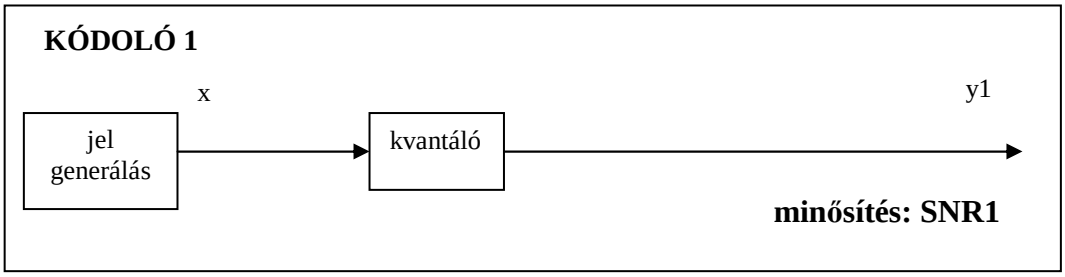
```
% mérő jel
x=2*(rand(1,1000)-0.5);
ns=40;                                  % a bemeneti jelszintek szama
xmax=logspace(-4,1,ns);  % a bemeneti jelszintek ertekei
bits=[4:2:16];
ddb=100; %input dinamikatartomany dB-ben

for j=1:length(bits),
  for i=1:ns,
    xs=xmax(i)*x;
    ys=kvant_le(xs,bits(j),d,ddb);
    snr(j,i)=snrdb(xs,ys,0);
  end;     %for i
end;  %j

figure(6);
semilogx(xmax,max(snr,zeros(size(snr))),'-'); grid;
ylabel('SNR(dB)'); xlabel('input level');
for j=1:length(bits),
  text(szigma(27)*szigma(27),snr(j,27),[num2str(bits(j)),' bit']);
end;
```


100
80
60
40
20
0
SNR(dB)
16 bit
14 bit
12 bit
10 bit
8 bit
6 bit
4 bit
10^-8
10^-6
10^-4
10^-2
10^0
10^2
input level

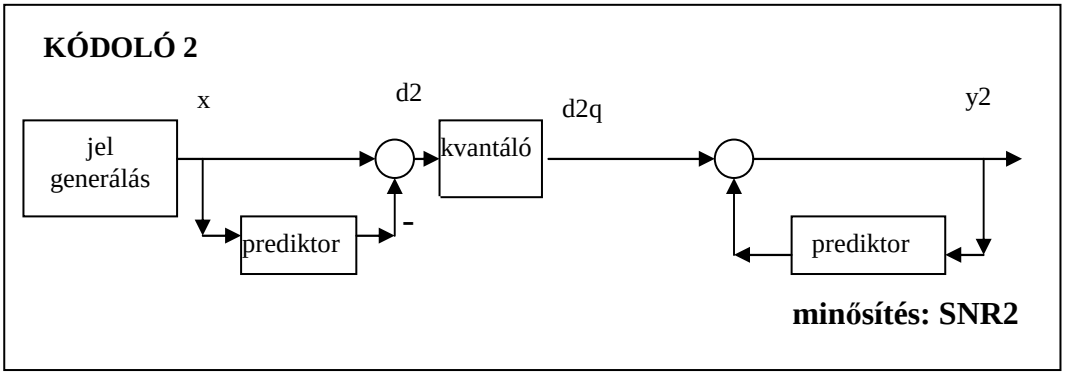
5. Gyakorlat: Differenciális kódolók vizsgálata

A MATLAB segítségével végezzük el az alábbi három hullámforma kódoló-dekódoló (jeldigitalizáló és rekonstruáló) rendszer összehasonlító elemzését.

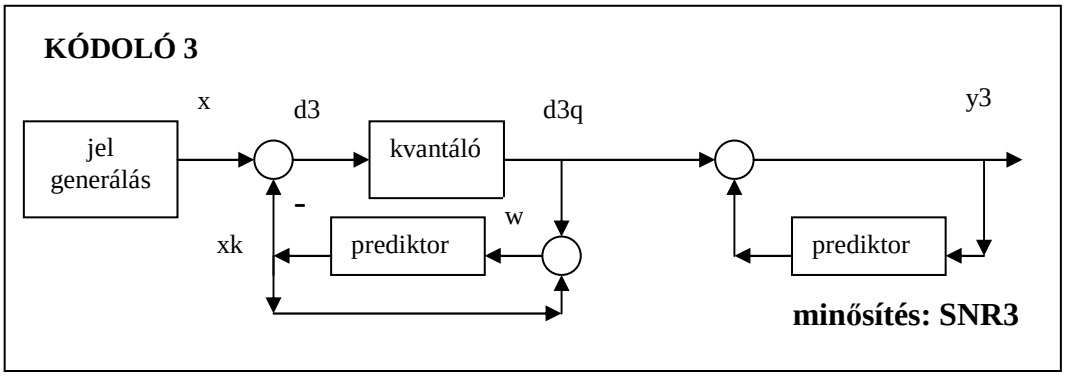
Az első rendszer pusztán egy illesztett kvantáló (a kvantáló dinamika tartománya a kvantálandó jel szórásának a háromszorosa):



A második esetben a jel és a predikáltja különbségeként előálló differenciális jelet kvantáljuk:



A harmadik esetben a differenciális-prediktív kódoló körbe hurkolja a kvantálót:

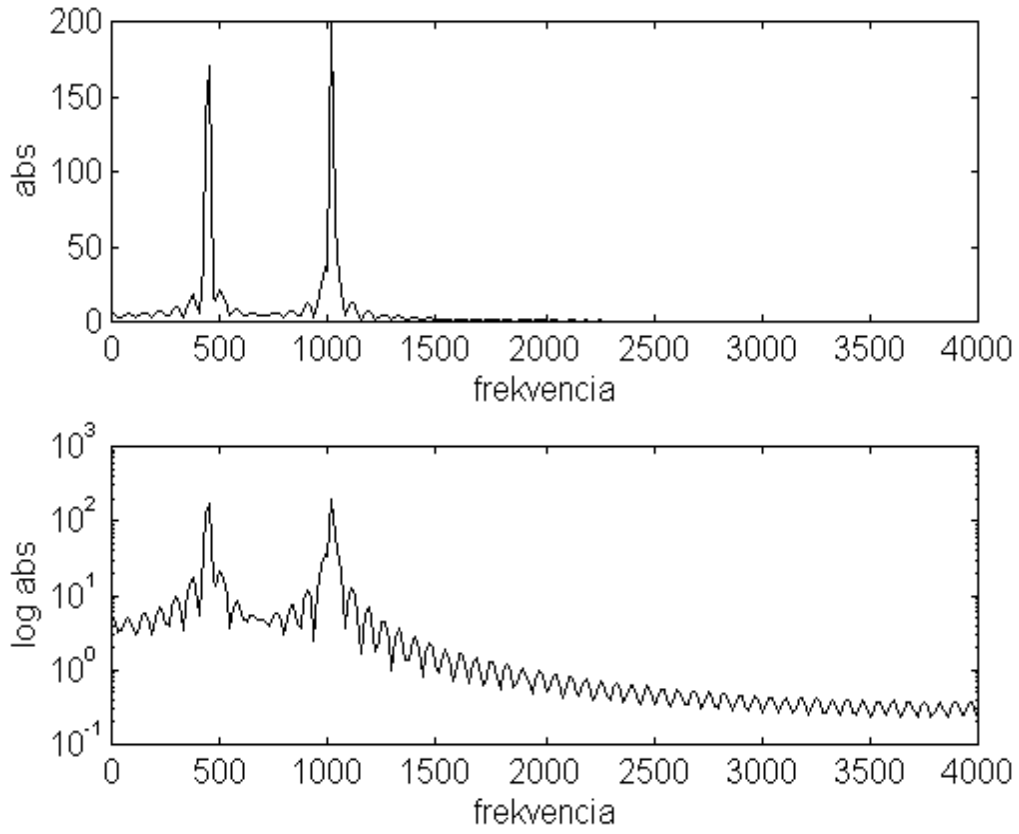


Vizsgáló jelként állítsuk elő két szinusz és fehé zaj összegét. A kezdeti vizsgálódásainknál zaj mentes esettel foglalkozunk:

pelda5.m

```
clear;

%jeleloallitas
fc=8000; dt=1/fc;
xn=400; nt=[0:xn]; tt=nt*dt;
f1=447; f2=1017;
a1=1; a2=1; az=0.0;
x=a1*sin(2*pi*f1*tt)+a2*sin(2*pi*f2*tt)+az*randn(size(tt));

figure(1);
[ft,sx]=t2f(fc,nt,x,0,fc/2);
subplot(211);
plot(ft,abs(sx),'-'); xlabel('frekvencia');  ylabel('abs');
subplot(212);
semilogy(ft,abs(sx),'-'); xlabel('frekvencia'); ylabel('log abs');
```

A vizsgáló jel spektruma:

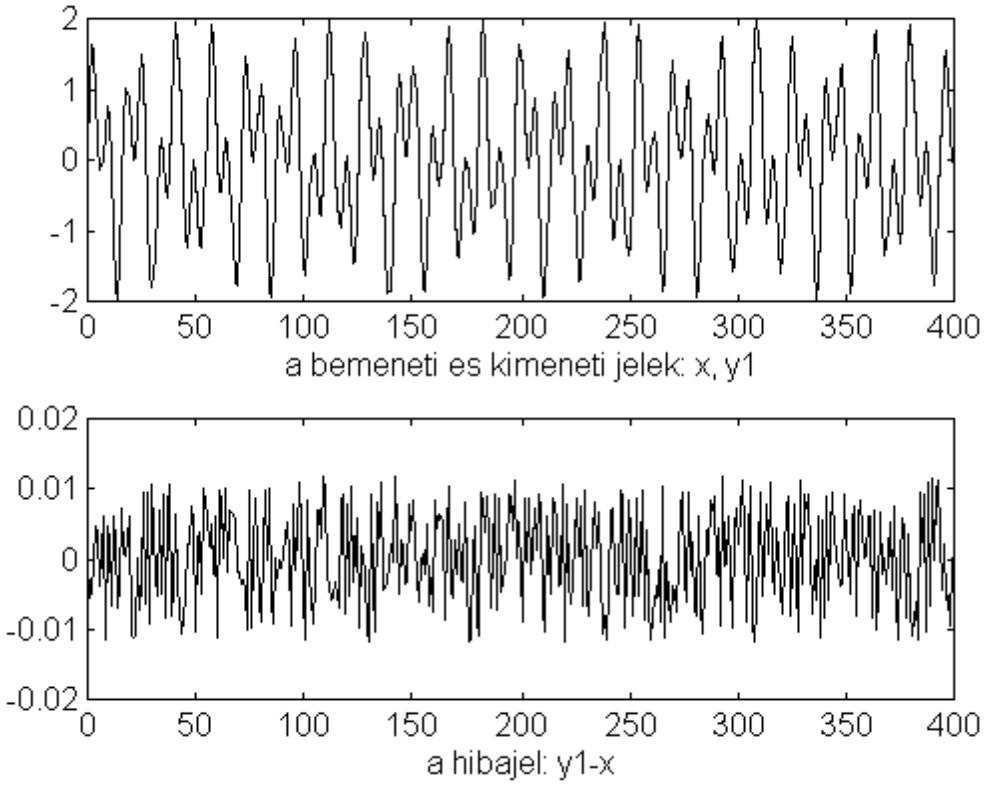


Az első kódoló vizsgálatához használjuk fel a korábban készített kvant_e.m és snrdb.m eszközeinket.

pelda5.m (folytatás)

```
% KÓDOLÓ 1
qbit=8;
qd=3*std(x);
y1=kvant_e(x,qbit,-qd, qd);

% minosites
snr1=snrdb(x,y1,0);

figure(2);
subplot(211);
plot(nt,x,'-',nt,y1,'-');
xlabel('a bemeneti es kimeneti jelek: x, y1');
subplot(212);
plot(nt,y1-x,'-');   xlabel('a hibajel: y1-x');
```

A kapott jelalakok:

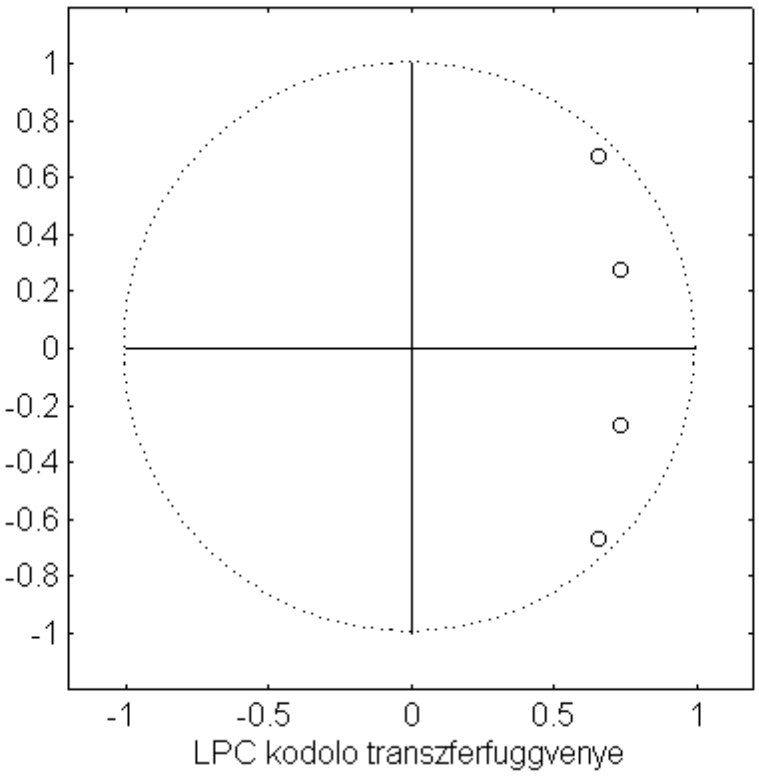


és a kapott jel-zaj viszony: SNR1 = 43.29 dB.

A második rendszerben a differenciális jel előállításához szükséges (negyed fokúnak választott) prediktort (a jelet a korábbi mintáiból optimálisan megbecslő szűrőt) a MATLAB lpc függvényével határozzuk meg a kódolandó x jelből. Egyúttal kirajzoltatjuk az így kapott kódoló P-Z ábráját is.

pelda5.m (folytatás)

```
% KODOLO 2
npred=4; pn=real(lpc(x,npred));   % 4-ed foku prediktor tervezes
figure(3); subplot(111);
pzabra(pn,1); xlabel('LPC kodolo transzferfuggvenye');
```



Figyeljuk meg, hogy a kódoló zérusai közel vannak a predikálandó jelben lévő szinuszok frekvenciájához.

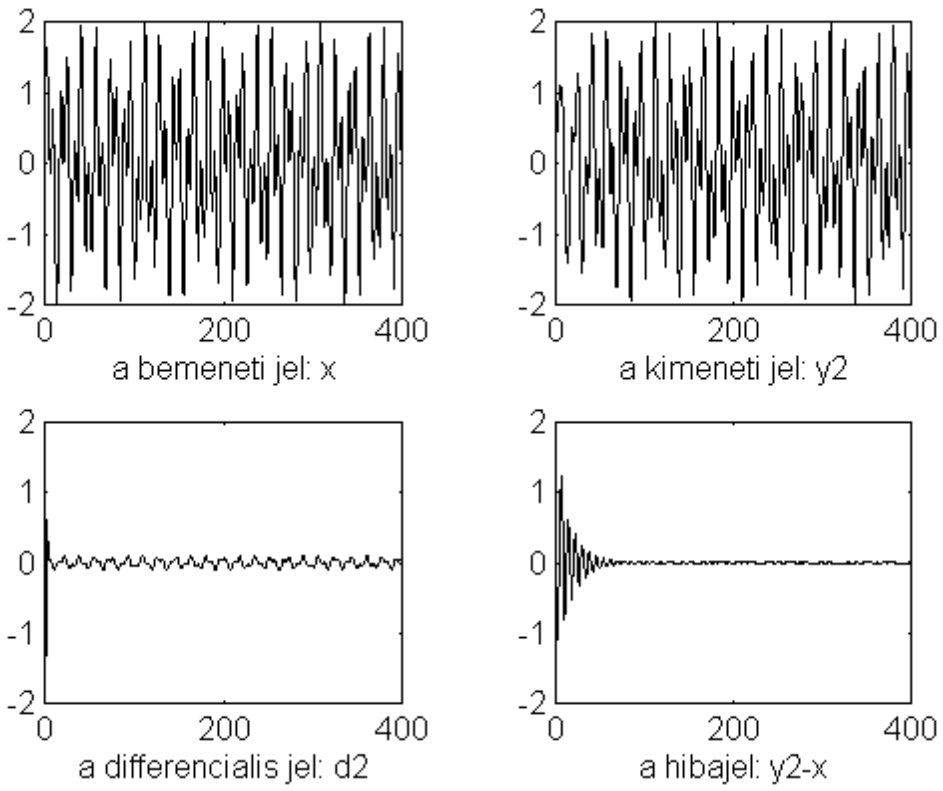
```
» (447/8000)*360 = 20.1150 fok
```
és
```
» (1017/8000)*360 =  45.7650 fok
```

A második kódoló analízise:

pelda5.m (folytatás)

```
d2=filter(pn,[1],x); %differencia kepzes
qd=3*std(d2); % az illesztett kvantalo dinamtartomanya
d2q=kvant_e(d2,qbit,-qd,qd); %kvantalas
y2=filter([1],pn,d2q); %differencialis jelbol visszallitas

%minosites
index=[100:length(x)];
snr2=snrdb(x(index),y2(index),0);
gp=10*log10((sum(x.*x)/length(x))/(sum(d2.*d2)/length(d2)));

figure(4);
subplot(221);
plot(nt,x,'-'); xlabel('a bemeneti jel: x');
subplot(222);
plot(nt,y2,'-'); xlabel('a kimeneti jel: y2');
subplot(223);
plot(nt,d2,'-');   xlabel('a differencialis jel: d2');
subplot(224);
plot(nt,y2-x,'-');   xlabel('a hibajel: y2-x');
```

A kapott jelalakok:



A rendszert minősítő SNR2 érték számításánál a tranziens mintákat figyelmen kívül hagytuk.
A kapott
SNR2 = 43.44dB

érték lényegében ugyan annyi, mint az előző egyszerű esetben, pedig a differenciális kódolótól várható predikciós nyereség számított értéke (az eredeti x jel és a kvantálandó d2 jel teljesítményeinek hányadosa):
Gp = 19.72 dB ,
azaz ezzel az értékkel megnővelt SNR lenne elérhető.
Természetesen ez az elvi lehetőség csak akkor realizálható, ha a prediktiv hurok közrefogja a kvantálót.
Ezen meggondolások szerint épül fel a harmadik kódoló.

pelda5.m (folytatás)

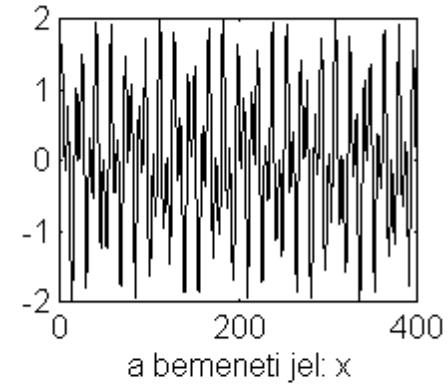
```
% KODOLO 3
w=zeros(1,npred);       % prediktor allapotvaltozoinak inicializalasa

% rekurziv rendszer
for i1=npred:length(x),
   xk=-pn(2:npred+1)*w(1:npred)';  % predikcio
   d3(i1)=x(i1)-xk;                % differencia kepzes
   d3q(i1)=kvant_e(d3(i1),qbit,-qd,qd); % kvantalas
   w=[d3q(i1)+xk,w(1:(npred-1))];   % prediktor allapot update
end;

y3=filter([1],pn,d3q);   %differencialis jelbol visszallitas

%minosites
index=[100:length(x)];
snr3=snrdb(x(index),y3(index),0);

figure(5);
subplot(221);
plot(nt,x,'-'); xlabel('a bemeneti jel: x');
subplot(222);
plot(nt,y3,'-'); xlabel('a kimeneti jel: y3');
subplot(223);
plot(nt,d3,'-');   xlabel('a differencialis jel: d3');
subplot(224);
plot(nt,y3-x,'-');   xlabel('a hibajel: y3-x');
```

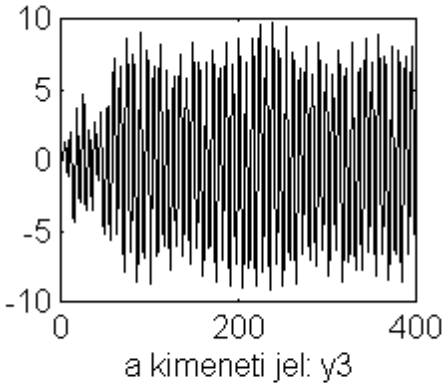
Összegező adatok, hisztogrammok kiírása:
pelda5.m (folytatás)

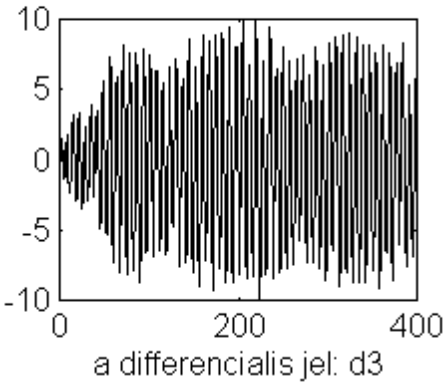
```
disp(['snr1=',num2str(snr1),'dB  ','snr2=',num2str(snr2),'dB  ',...
      'snr3=',num2str(snr3),'dB  ','Gp=',num2str(gp),'dB ' ]);

figure(6);
subplot(221); [nn,hh]=hist(x); bar(hh,nn,'-');xlabel('x hisztogram');
subplot(223); [nn,hh]=hist(d2); bar(hh,nn,'-'); xlabel('d2 hisztogram');
subplot(224); [nn,hh]=hist(d3); bar(hh,nn,'-'); xlabel('d3 hisztogram');
```

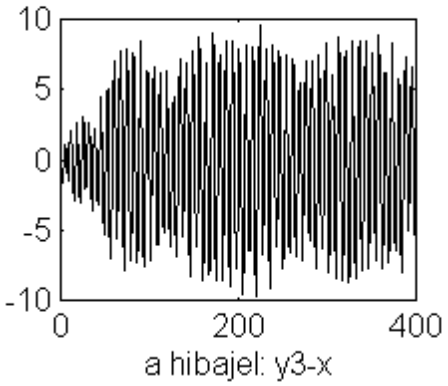
A kapott eredmények és ábrák:

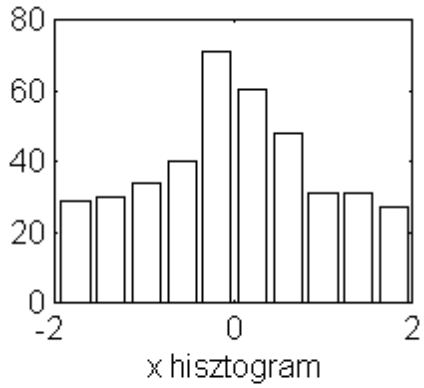
SNR1=43.29dB SNR2=43.44dB SNR3=-14.78dB Gp=19.72dB

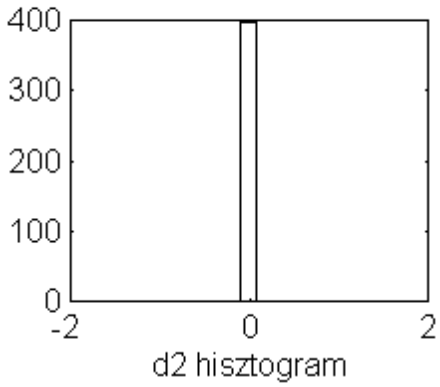

2
1
0
-1
-2
0
200
400
a bemeneti jel: x

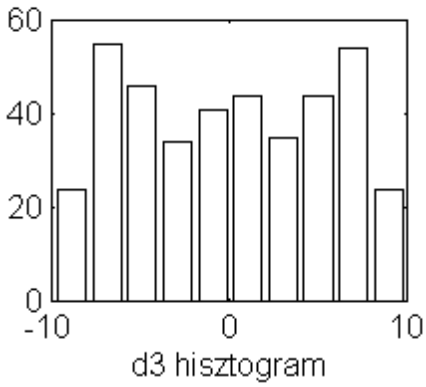

10
5
0
-5
-10
0
200
400
a kimeneti jel: y3


10
5
0
-5
-10
0
200
400
a differencialis jel: d3


10
5
0
-5
-10
0
200
400
a hibajel: y3-x


80
60
40
20
0
-2
0
2
x hisztogram


400
300
200
100
0
-2
0
2
d2 hisztogram


60
40
20
0
-10
0
10
d3 hisztogram

Mint több jelből látható, valami nagy baj van, a rendszer "gerjed":

- SNR3=-14.78dB ! értelmetlenség
- a differenciális- (d3) és a hibajel (y3-x) "felcseng", sokkal nagyobb mint maga az eredeti jel

A helytelen működés oka, a kvantáló nagyjelű nemlinearitásában (limiter karakterisztika) keresendő.

Ennek bizonyítására, ideiglenesen módosítsuk a **kvant_e.m** függvényünket úgy, hogy a nagyjelű limitálást "kikapcsoljuk", azza minden bemenetre csak kerekítést végez a kvantáló:

kvant_e.m (ideiglenesen módosítva)

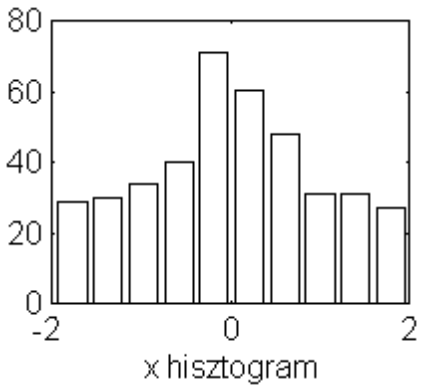
```
function y = kvant_e(x,r,xmin,xmax);
%  y = kvant_e(x,r,xmin,xmax);
%  Egyenletes kvantalo
%     x: bemeneti mintak
%     r: r bites kvantalas
%     xmin-xmax: felett egyenletes kvantalo, |hiba| < q/2
%     y: kvantalt kimeneti mintak
  l = 2^r;              % dontesi intervallumok, szintek szama
  q = (xmax-xmin)/l;                % a kvantalasi lepcso
% nagy jelu mukodes
%%%%  y = min(x,(xmax-q/2)*ones(size(x)));
% kis jelu mukodes
%%%%  y = max(y,(xmin+q/2)*ones(size(y)));
% kerekites
  y = xmin+q/2+q*round((y-(xmin+q/2))/q);
```

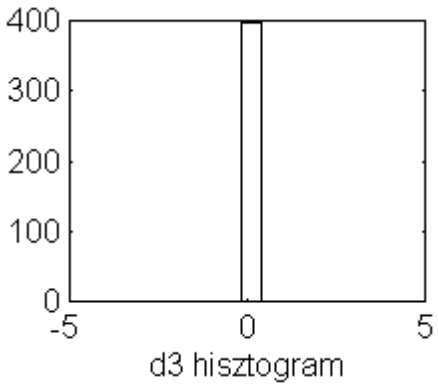
Ezek után újrafuttatva a pelda5.m programot, az alábbiak szerinti eredményeket kapjuk:

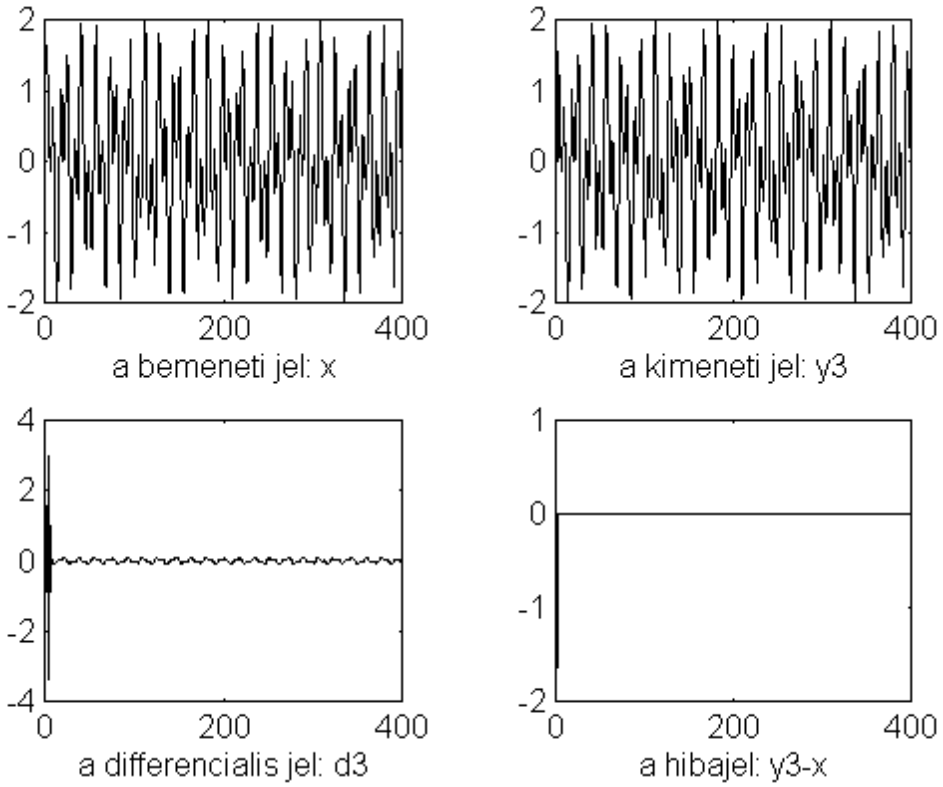
SNR1=43.29dB SNR2=43.51dB SNR3=62.87dB Gp=19.72dB

Most már SNR3 értelmes (a Gp predikciós nyereséggel megnövelt) értéket ért el.

A kapott ábrák:







Tehát, ami a jeleket illeti, most már minden az elvárások szerint alakult.
Ez a kisérlet, természetesen nem megoldása a predikciós kódoló müködésének, csupán azt bizonyítottuk, hogy a fixen a d2 differenciális jelhez (,mely a finom kvantálástól eltekintve ugyan az, mint a d3 differenciális jel és stacioner esetben igen kis jel) illesztett kis dinamika tartományú kvantáló a bekapcsoláskor (és a tranziens elején) olyan nagy túlvezérlést kap, hogy az ekkor keletkező jelekre a nemlináris, visszacsatolt rendszer nagyjelű, határciklusos gerjedést produkál.

A valódi megoldás skálázófaktor-adaptiv kvantáló alkalmazása a visszacsatolt hurkon belül. Ennek az elgondolásnak a MATLAB-beli implementációját a korábbi pelda5.m program megfelelő helyének az alábbi módosításával készíthetjük el:

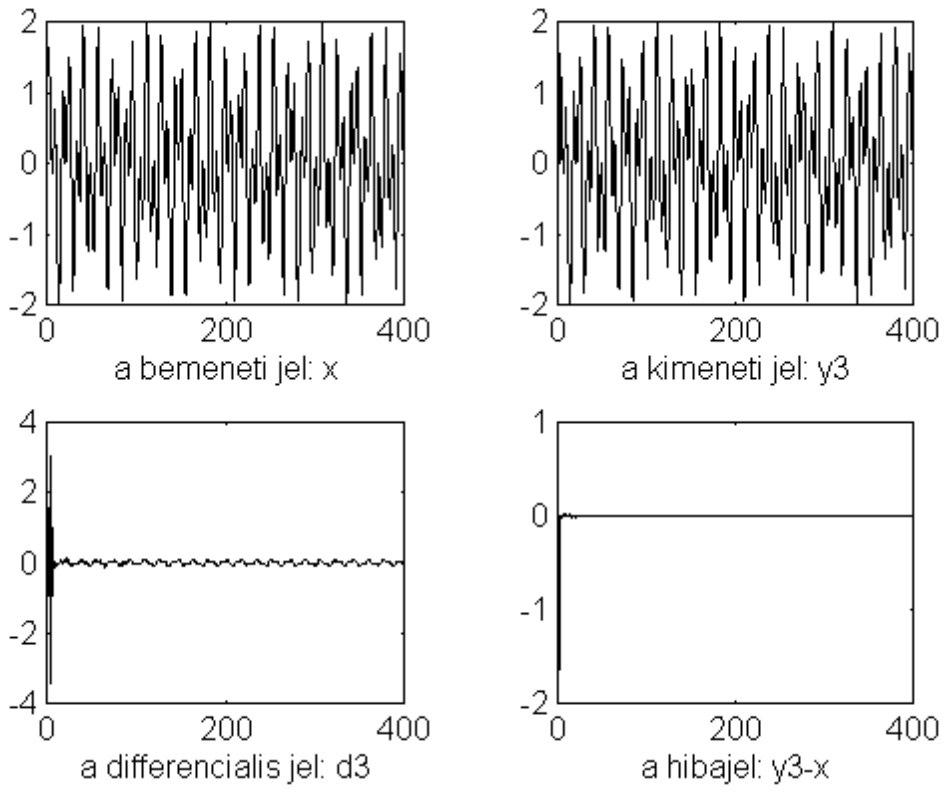
pelda5.m (kiegészítve adaptív kvantálóval)

```
% KODOLO 3.
w=zeros(1,npred);
powd3=1; alfa=2^(-5);            % adaptalas inicializalasa

% rekurziv rendszer
for i1=npred:length(x),
   xk=-pn(2:npred+1)*w(1:npred)';  % predikcio
   d3(i1)=x(i1)-xk;               % differencia kepzes

   % Adaptiv kvantalo
   powd3=(1-alfa)*powd3 + alfa*d3(i1)*d3(i1);
   qd=3*sqrt(powd3);

   d3q(i1)=kvant_e(d3(i1),qbit,-qd, qd); % kvantalas
   w=[d3q(i1)+xk,w(1:(npred-1))];   % prediktor allapot update
end;

y3=filter([1],pn,d3q);   %differencialis jelbol visszallitas
```

A kisérlet újra futtatásánál ne felejtsük el kvant_e.m eredeti működését visszaállítani!

A kapott eredmények:

SNR1=43.29dB SNR2=43.51dB SNR3=63.4dB Gp=19.72dB



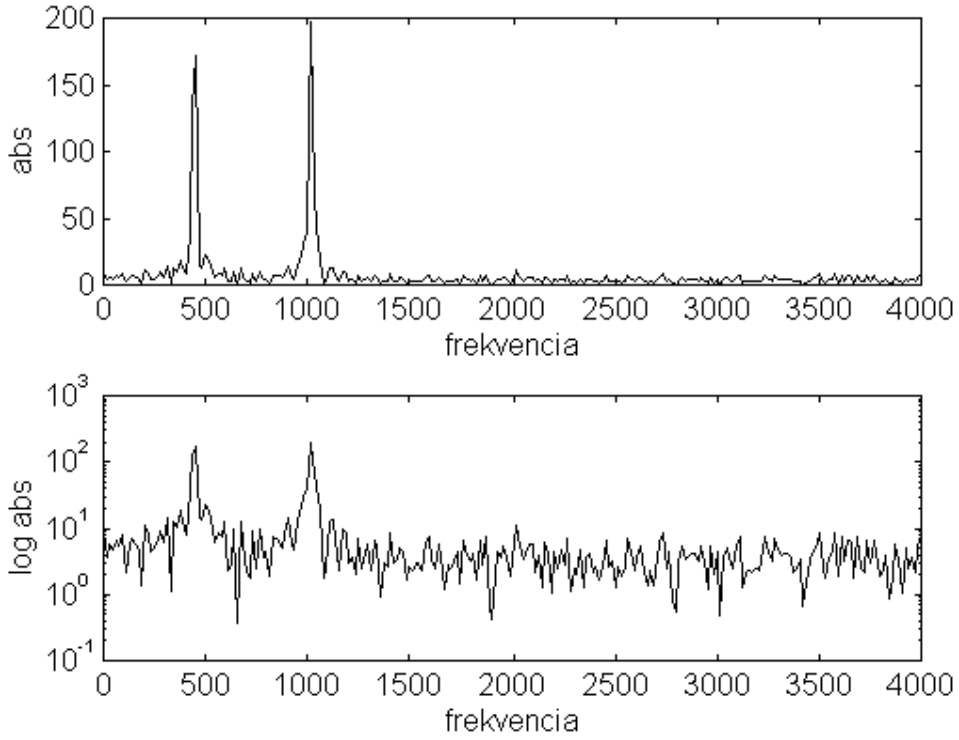
Az edigi vizsgálatoknál a kódólandó jelnek csak keskenysávú (szinuszos) összetevői voltak.

Az alábbiak szerint módosítva pelda5.m filet futtassuk újra a MATLAB programunkat.

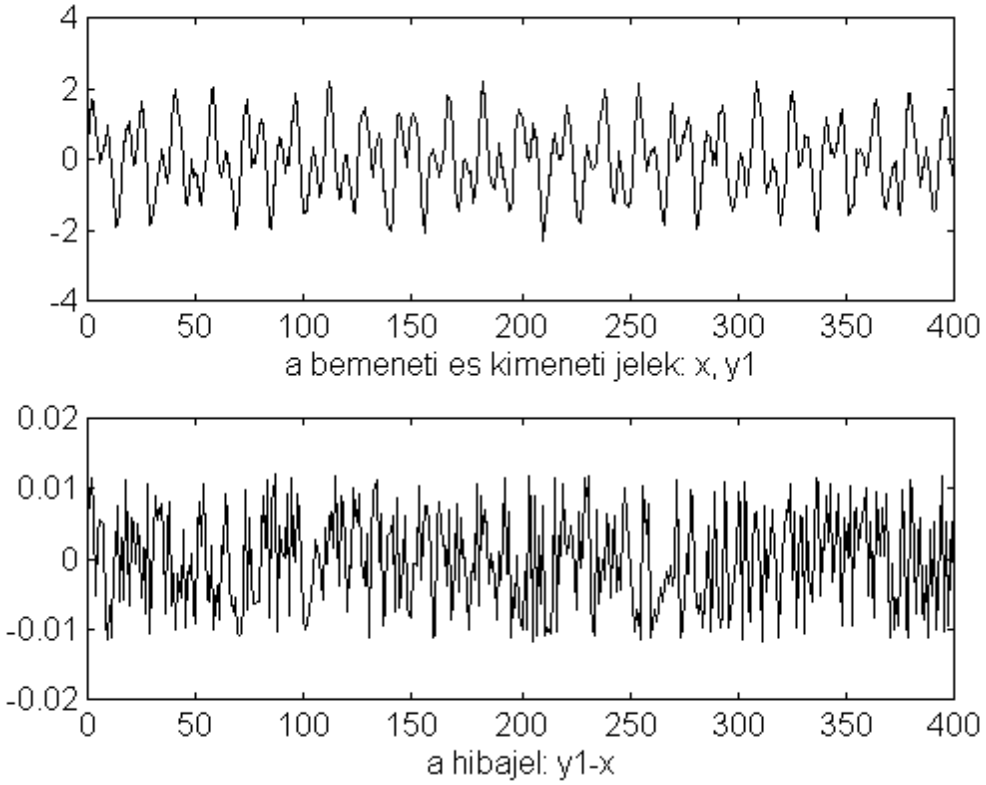
pelda5.m (adat módisításs)

```
%a1=1; a2=1; az=0.0;
a1=1; a2=1; az=0.2;
```

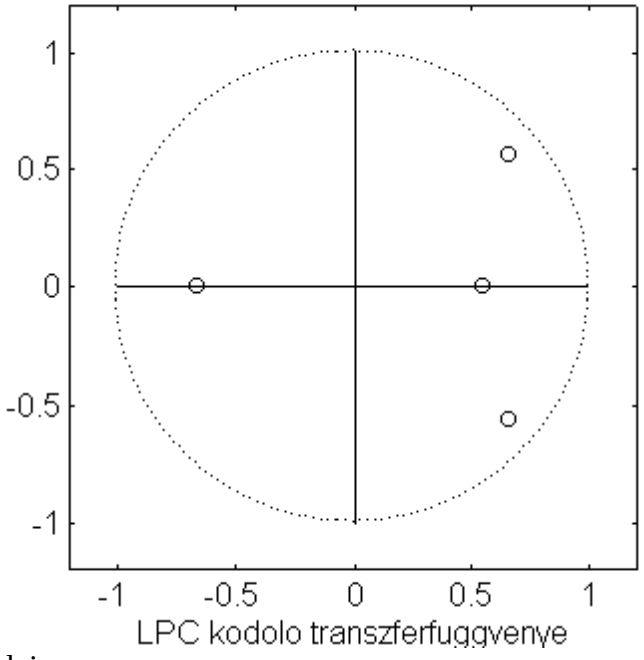
A keskeny és szélessávú (feher zaj) összetevőt is tartalmazó bemenő jel spektruma:



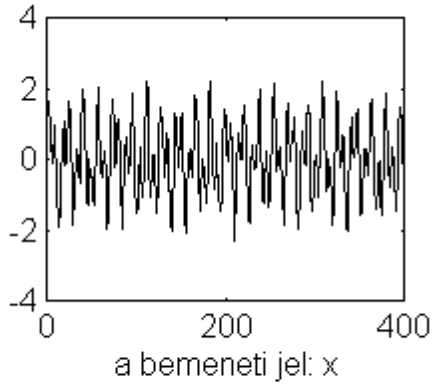
A következő oldalon a KODOLÓ 1. kimenő jelét és hiba jelét látjuk.

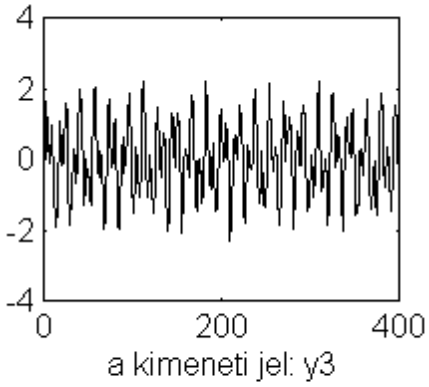


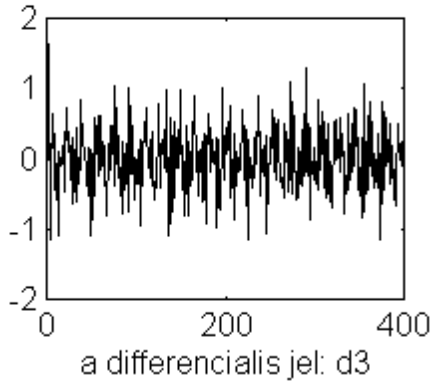
A differenciális kódolók prediktorainak zérusai a zaj öszetevő megjelenésének hatására az ábrán látható módon változtak.

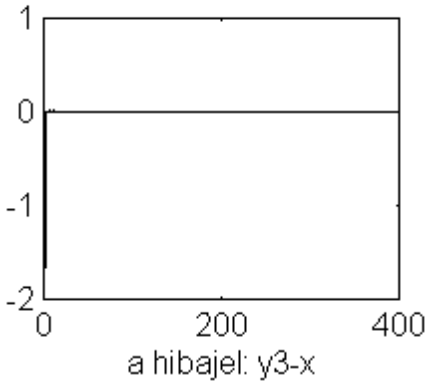


A KÓDOLÓ3 jelei:

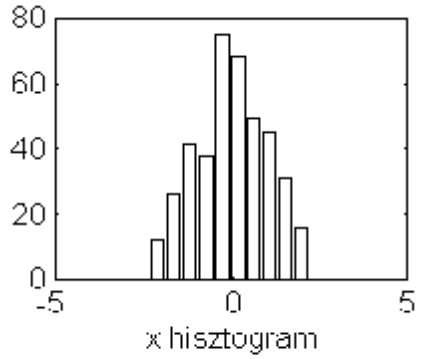


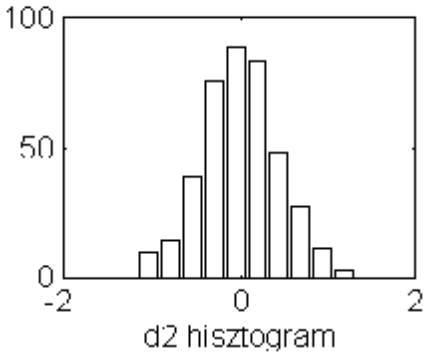


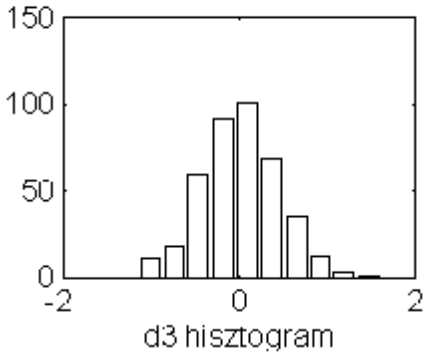




Hisztogrammok:







Végezetül a jel/zaj értékek:
SNR1=43.32dB SNR2=42.4dB SNR3=50.92dB Gp=7.374dB