



Budapesti Műszaki és Gazdaságtudományi Egyetem

Méréselmélet

(BMEVIMIM108)

2. Házi feladat

Készítette:

Sipos-Takáts Bence László
T35NOC

Budapest, 2011.04.20

```

clear all
close all

%%%% 1. Feladat

% Paraméterek

N=64;
p=0.76;

Bazis_M = zeros(N,N);
Reciprokbazis_M = zeros(N,N);

% Bázismátrix értékadás

for sor = 1:N
    for oszlop = 1:(N/2)
        Bazis_M(sor,2*oszlop-1) = sqrt(2/N) * real(exp(j*pi*(2*oszlop-1)*sor/N));
        Bazis_M(sor,2*oszlop) = sqrt(2/N) * imag(exp(j*pi*(2*oszlop-1)*sor/N));
    end
end

% Reciprokbázis mátrix értékadás

for oszlop = 1:N
    for sor = 1:(N/2)
        Reciprokbazis_M(2*sor-1,oszlop) = sqrt(2/N) * real(exp(j*pi*(2*sor-1)*oszlop/N));
        Reciprokbazis_M(2*sor,oszlop) = sqrt(2/N) * imag(exp(j*pi*(2*sor-1)*oszlop/N));
    end
end

%%%% 2. Feladat - Ellenőrzés

% vizuális ellenőrzés: a számolt reciprokbázis egybeesik-e a számolt bázis
% Matlab által előállított inverzével

szamoltRecBazis = Bazis_M';
figure(1)
x=[1:1:N]
for i=1:N;
    plot(x,szamoltRecBazis(i,:), 'or')
    hold on;
    plot(x,Reciprokbazis_M(i,:), 'xb')
end

% Ha a két számolt mátrixot összeszorozzuk, egységmátrixot kell kapnunk. Ha
% ebből kivonunk egy egységmátrixot, 0-t kell kapnunk. Azonban a Matlab
% hibája végesen kicsi, ezért 0 helyett 10^-10-en hibátűrést megengedünk.

Bazis_szorzat = Bazis_M * Reciprokbazis_M;
Kulonbseg = Bazis_szorzat-eye(N);

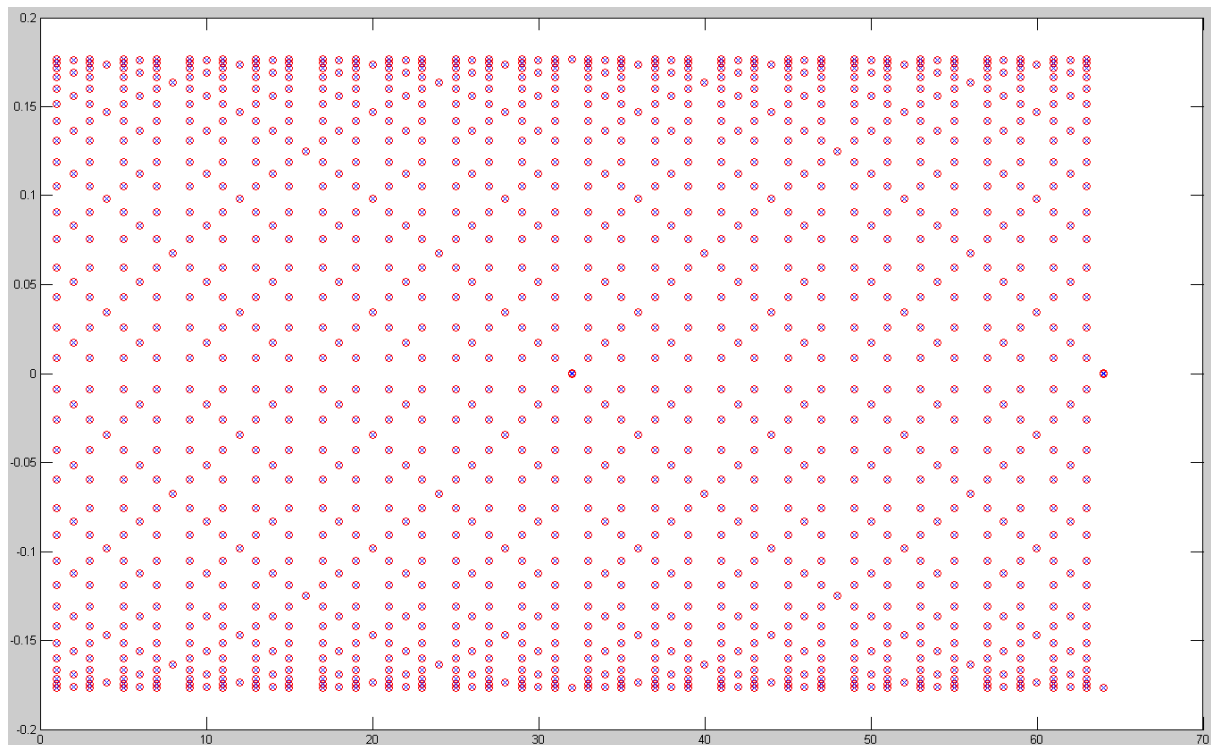
```

```

x=0;
for i=1:N
    for j=1:N
        if Kulonbseg(i,j)>10^-10
            'Nem megengedett pontatlanság'
            x=1;
        end
    end
end

if x==0;
    'Mátrixok megfelelnek'
end

```



1. Ábra: Vizuális ellenőrzés

A Mátrixok megfelelnek, vagyis a kettő különbsége (pontatlansága) 10^{-10} nagyságrendű.

%%% 3. Feladat - Jelszintetizátorok

```

% A súlytényezőket úgy állítom elő, hogy a bázis mátrix inverzét megszorodom
% a jellel. Ellenőrzésképp összeszorozom a súlytényezőt a bázismátrix-al és
% összehasonlítom az eredeti jellel

```

```

% 3.1 - Négyszögjel
% Ezt a jelet a square függvénnyel állítom elő, periódusa 2*8

```

```

t=0:N-1;
negyszog = square(t*pi/8);
negyszog_suly = inv(Bazis_M)*negyszog';

```

```

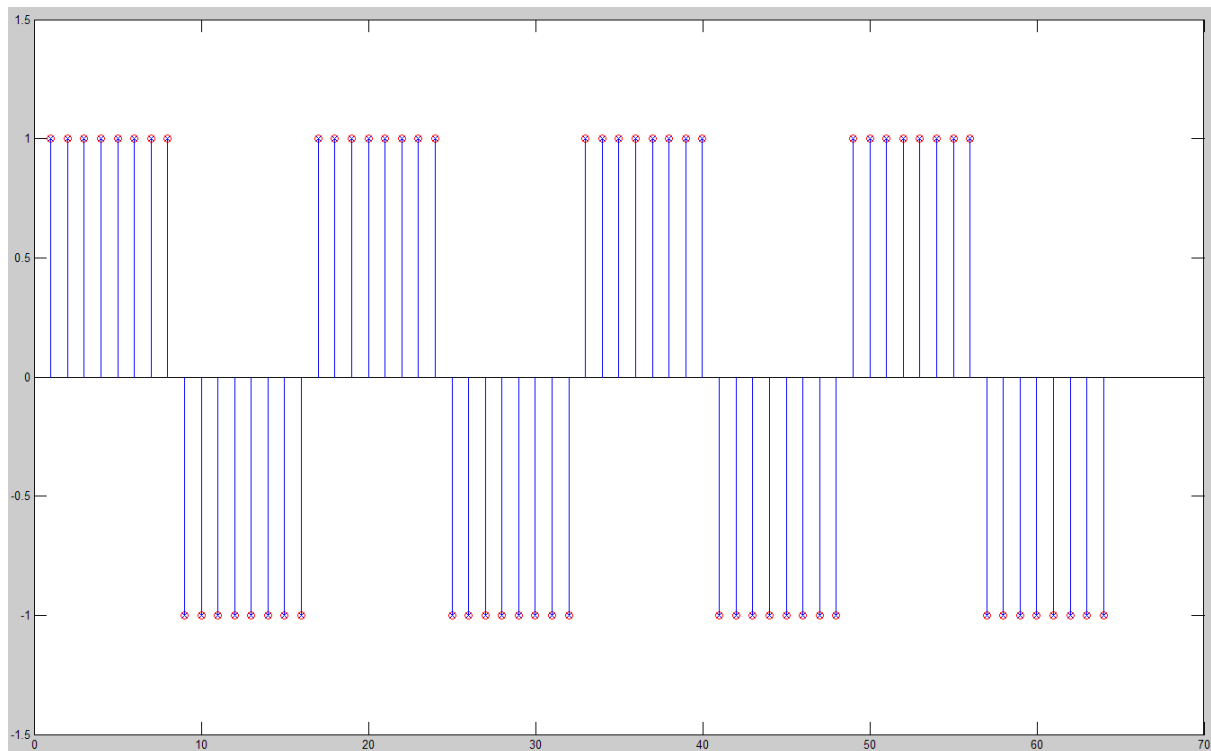
negyszog_suly_i=negyszog_suly';
proba_negyszog = Bazis_M*negyszog_suly;

```

```

figure(2);
stem(negyszog, 'or');
hold on;
stem(proba_negyszog, 'xb');

```



2. Ábra: Négyzögjel

```
% 3.2 - Háromszögjel
```

```

t=4:N+3;
haromszog = sawtooth(t*pi/8, 0.5);
haromszog_suly = inv(Bazis_M)*haromszog';

```

```

haromszog_suly_i=haromszog_suly';
proba_haromszog = Bazis_M*haromszog_suly;
figure(3);
stem(haromszog, 'or');
hold on;
stem(proba_haromszog, 'xb');

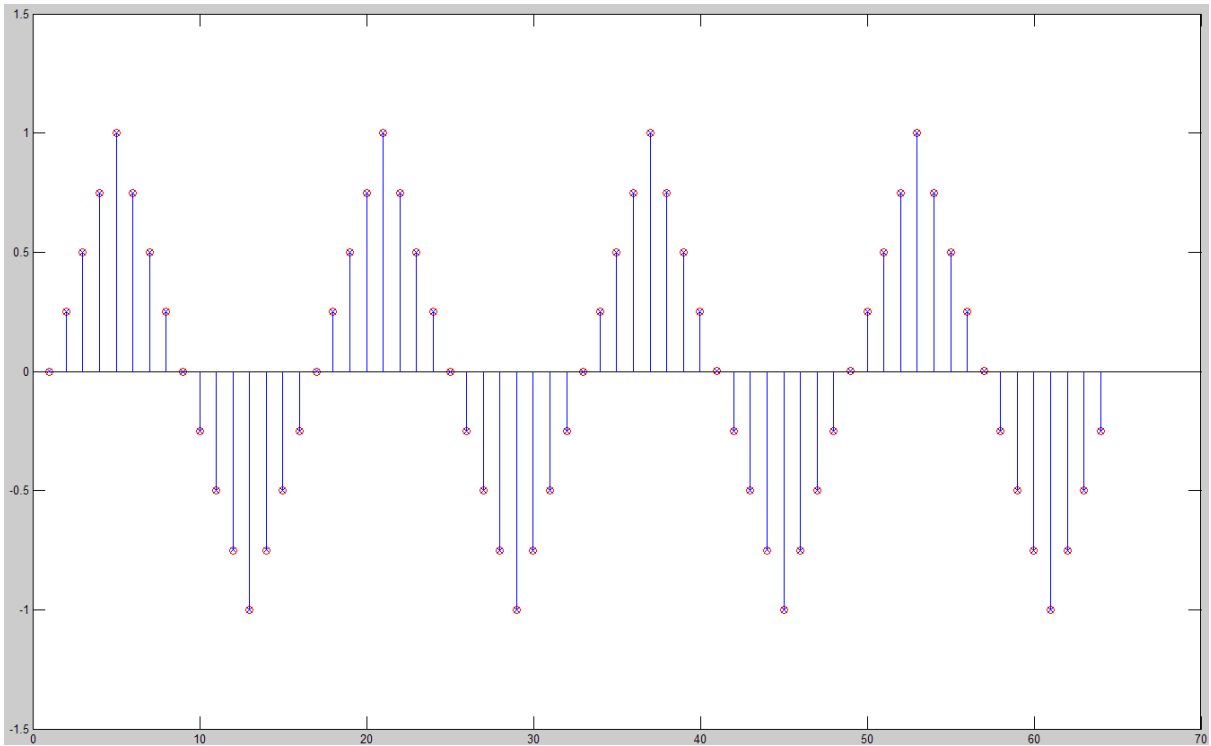
```

```
% 3.3 - Multiszinus jel
```

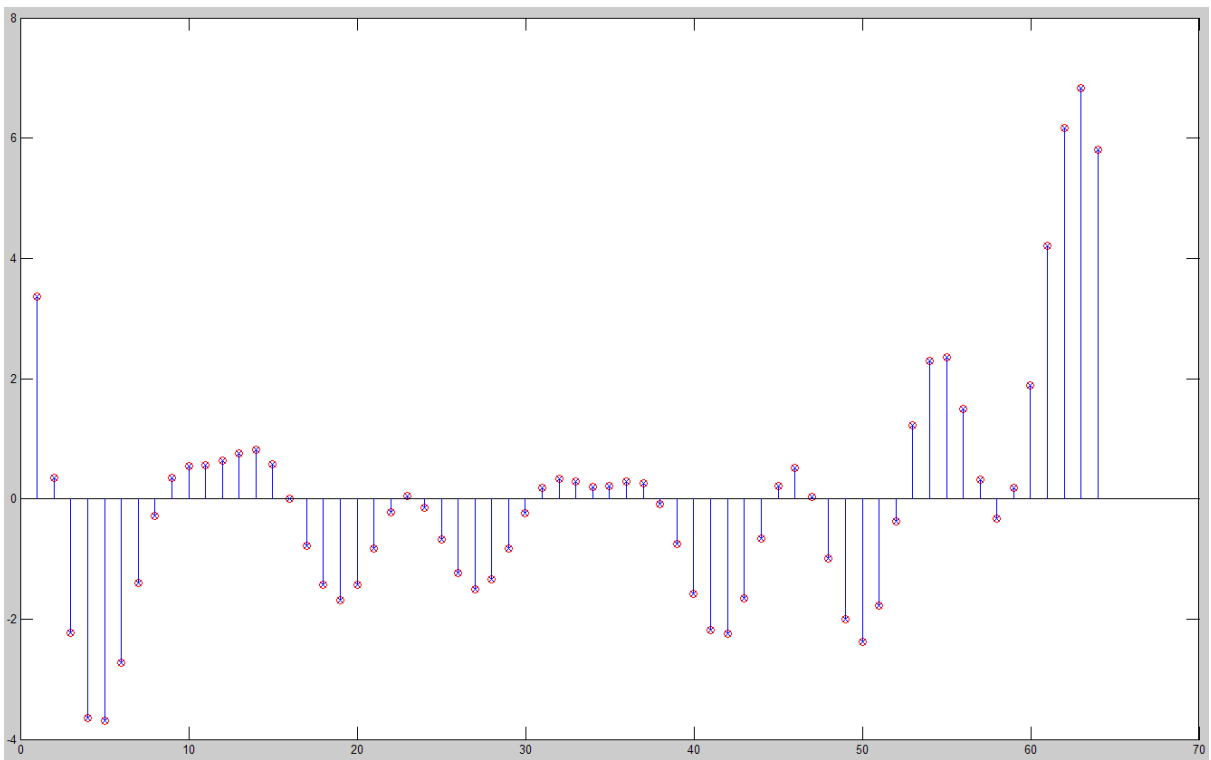
```

multiszinusz = zeros(N,1);
fazis=rand(8,1);
for i=1:N
    for j=1:8
        multiszinusz(i) = multiszinusz(i) + sin(i*j*2*pi/N+fazis(j)*pi);
    end
end
multiszinusz_suly = inv(Bazis_M)*multiszinusz;
proba_multiszinusz = Bazis_M*multiszinusz_suly;
figure(4);
stem(multiszinusz, 'or');
hold on;
stem(proba_multiszinusz, 'xb');

```



3. Ábra: Háromszögjel



4. Ábra: Multiszínusz jel

%%%% 4. Feladat - Súlytényező mérő

% A becslés ciklikusan N*N lépésben történik

```
negyszog_mero=zeros(N,1);
haromszog_mero=zeros(N,1);
multiszinusz_mero=zeros(N,1);
```

% Négyyszögjel

```
for i=1:N
    for j=1:N
```

```
negyszog_mero(j)=negyszog_mero(j)+Reciprokbazis_M(j,i).*negyszog(i);
    end
end
```

% Háromszögjel

```
for i=1:N
    for j=1:N
```

```
haromszog_mero(j)=haromszog_mero(j)+Reciprokbazis_M(j,i).*haromszog(i);
    end
end
```

% Multiszinusz jel

```
for i=1:N
    for j=1:N
```

```
multiszinusz_mero(j)=multiszinusz_mero(j)+Reciprokbazis_M(j,i).*multiszinusz(i);
    end
end
```

```
negyszog_mero_i=negyszog_mero';
haromszog_mero_i=haromszog_mero';
```

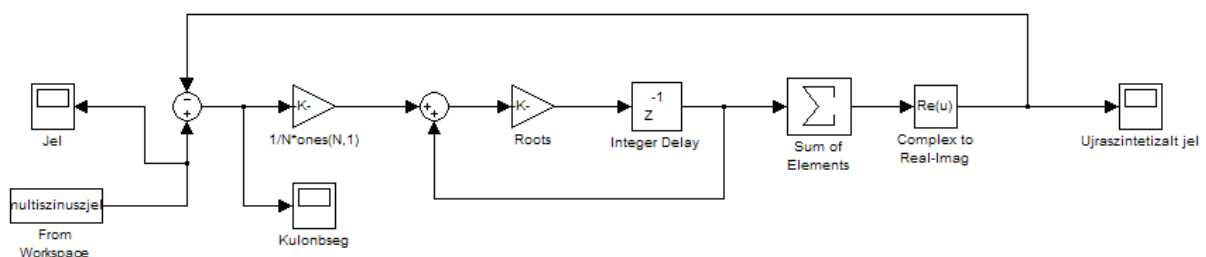
Súlytényezők:

N=1:8								
Négyyszög súlytényező	1,432385	0,035163	1,601292	0,118118	2,145037	0,264565	5,121799	0,888716
Négyyszög becslt	1,432385	0,035163	1,601292	0,118118	2,145037	0,264565	5,121799	0,888716
Háromszög súlyt.	0,717925	0,035269	0,81866	0,121437	1,144317	0,286636	2,936895	1,050838
Háromszög becslt	0,717925	0,035269	0,81866	0,121437	1,144317	0,286636	2,936895	1,050838
Multisin súlytényező	-5,99117	-2,99485	-6,09994	0,838789	-5,02958	2,893457	-1,31301	-1,34135
Multisin becslt	-5,99117	-2,99485	-6,09994	0,838789	-5,02958	2,893457	-1,31301	-1,34135
N=9:16								
Négyyszög súlytényező	-3,95763	-0,88872	-0,95602	-0,26457	-0,35755	-0,11812	-0,09116	-0,03516
Négyyszög becslt	-3,95763	-0,88872	-0,95602	-0,26457	-0,35755	-0,11812	-0,09116	-0,03516

Multisin súlytényező	-0,78188	0,714404	-0,79773	0,63528	-0,8109	0,561667	-0,82184	0,492565
Multisin becslt	-0,78188	0,714404	-0,79773	0,63528	-0,8109	0,561667	-0,82184	0,492565
N=49:56								
Négyszög súlytényező	0,013564	0,035163	0,039021	0,118118	0,073215	0,264565	0,199568	0,888716
Négyszög becslt	0,013564	0,035163	0,039021	0,118118	0,073215	0,264565	0,199568	0,888716
Háromszög súlyt.	0,037987	-0,03443	0,043361	-0,03216	0,057133	-0,03424	0,128537	-0,06079
Háromszög becslt	0,037987	-0,03443	0,043361	-0,03216	0,057133	-0,03424	0,128537	-0,06079
Multisin súlytényező	-0,83091	0,42716	-0,83839	0,364783	-0,84448	0,304866	-0,84934	0,246922
Multisin becslt	-0,83091	0,42716	-0,83839	0,364783	-0,84448	0,304866	-0,84934	0,246922
N=57:64								
Négyszög súlytényező	-0,15421	-0,88872	-0,03263	-0,26457	-0,00871	-0,11812	-0,00086	-0,03516
Négyszög becslt	-0,15421	-0,88872	-0,03263	-0,26457	-0,00871	-0,11812	-0,00086	-0,03516
Háromszög súlyt.	-0,08842	0,031639	-0,01741	0,00436	-0,00445	0,000661	-0,00043	2,13E-05
Háromszög becslt	-0,08842	0,031639	-0,01741	0,00436	-0,00445	0,000661	-0,00043	2,13E-05
Multisin súlytényező	-0,8531	0,190522	-0,85585	0,135284	-0,85765	0,080853	-0,85854	0,026898
Multisin becslt	-0,8531	0,190522	-0,85585	0,135284	-0,85765	0,080853	-0,85854	0,026898

Látható, hogy a súlytényezők és folyamatosan mért értékeik értékeik megegyeznek.

5. Feladat - Jelgenerátorra rákapcsolás



5. Ábra: Jelszintetizátor Simulink-es blokkvázlata

```
negyszogjel.time=[];
negyszogjel.signals.values=proba_negyszog;
haromszogjel.time=[];
haromszogjel.signals.values=proba_haromszog;
multiszinusjel.time=[];
multiszinusjel.signals.values=proba_multiszinusuz;
```

```
% N elemu egységkör létrehozása
```

```
clear j;
```

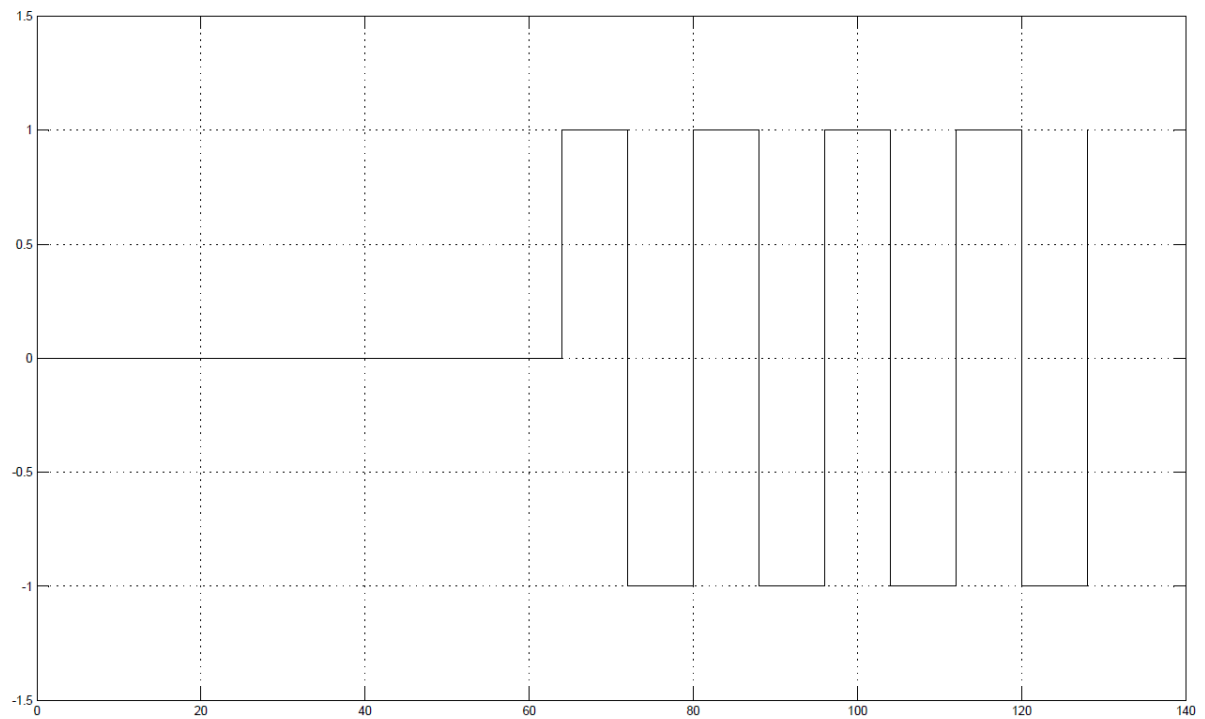
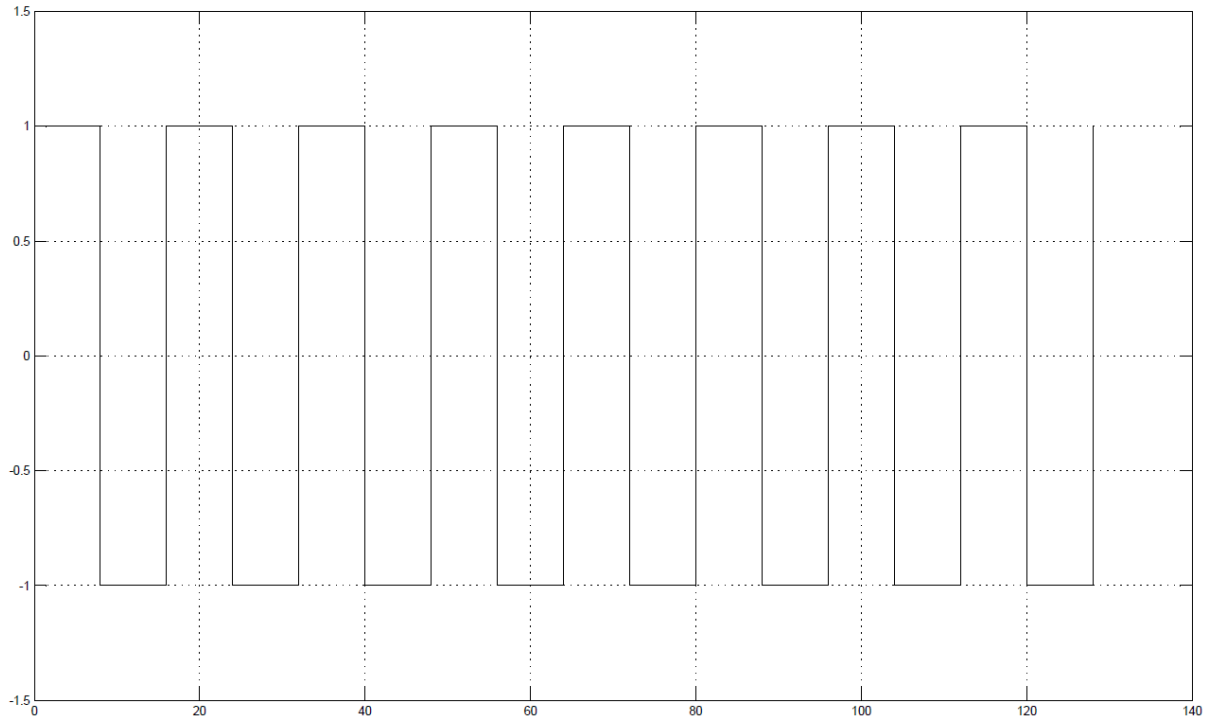


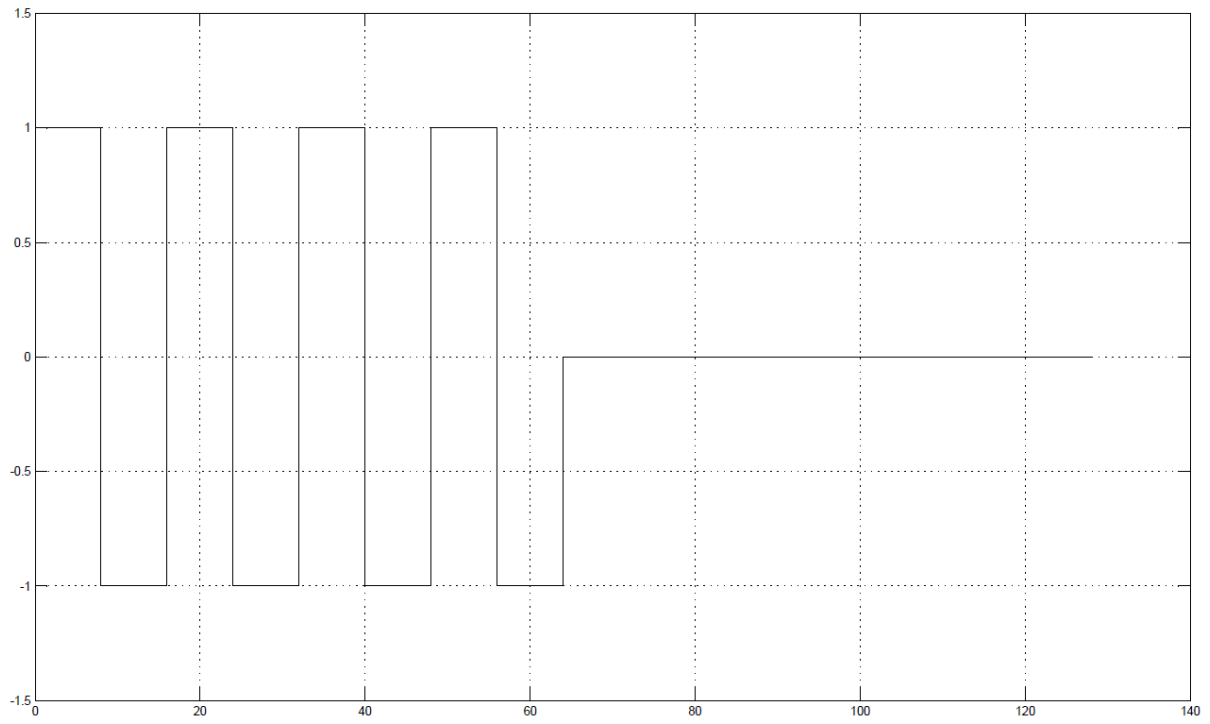
```
Roots=zeros(N,1);
```

```
for i=1:N  
    Roots(i,1)=exp(j*(2*pi/N)*i);  
end
```

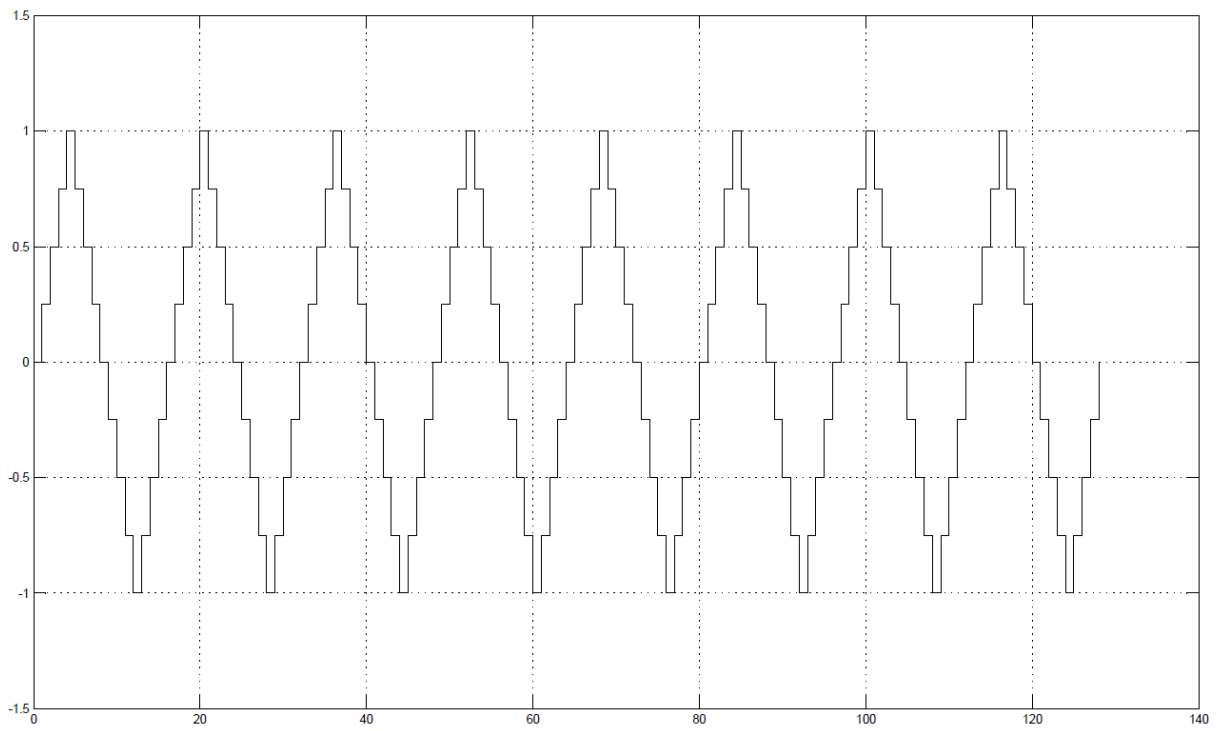
Az ábrák az egyes jelformáknál a következő sorrendben láthatók: jel, újrászintetizált jel és végül a különbség. A 2N elemig való futást Simulink-en állítottam be.

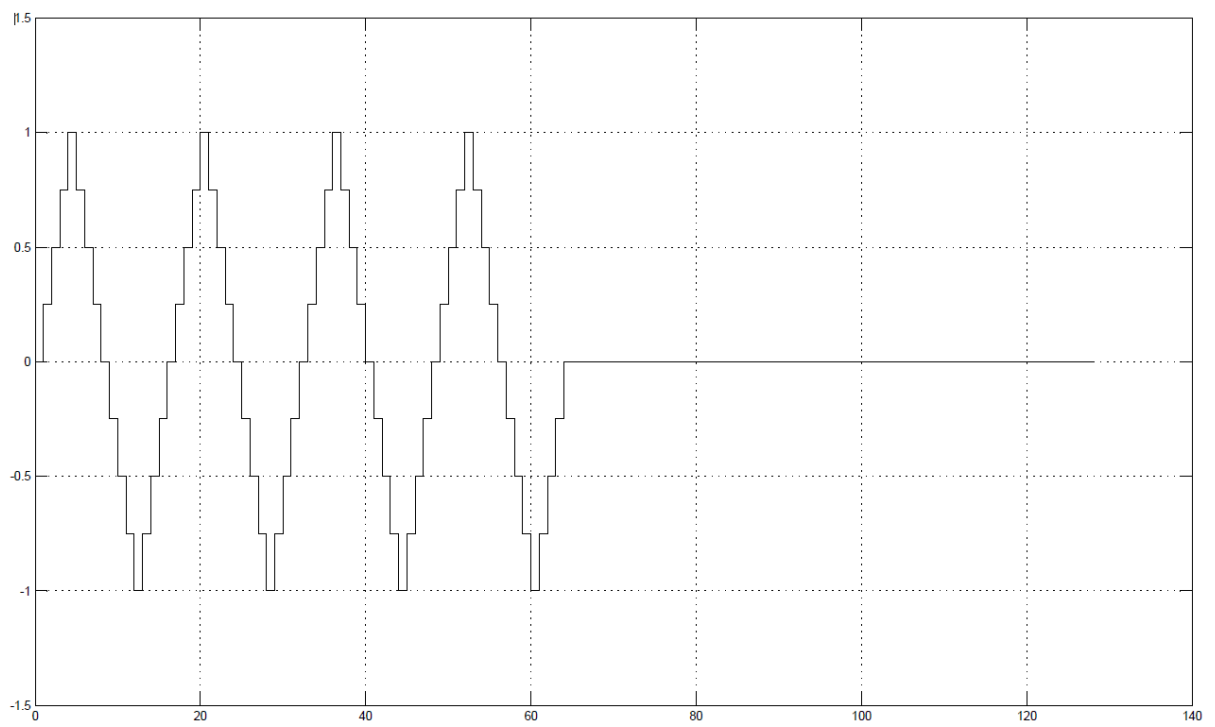
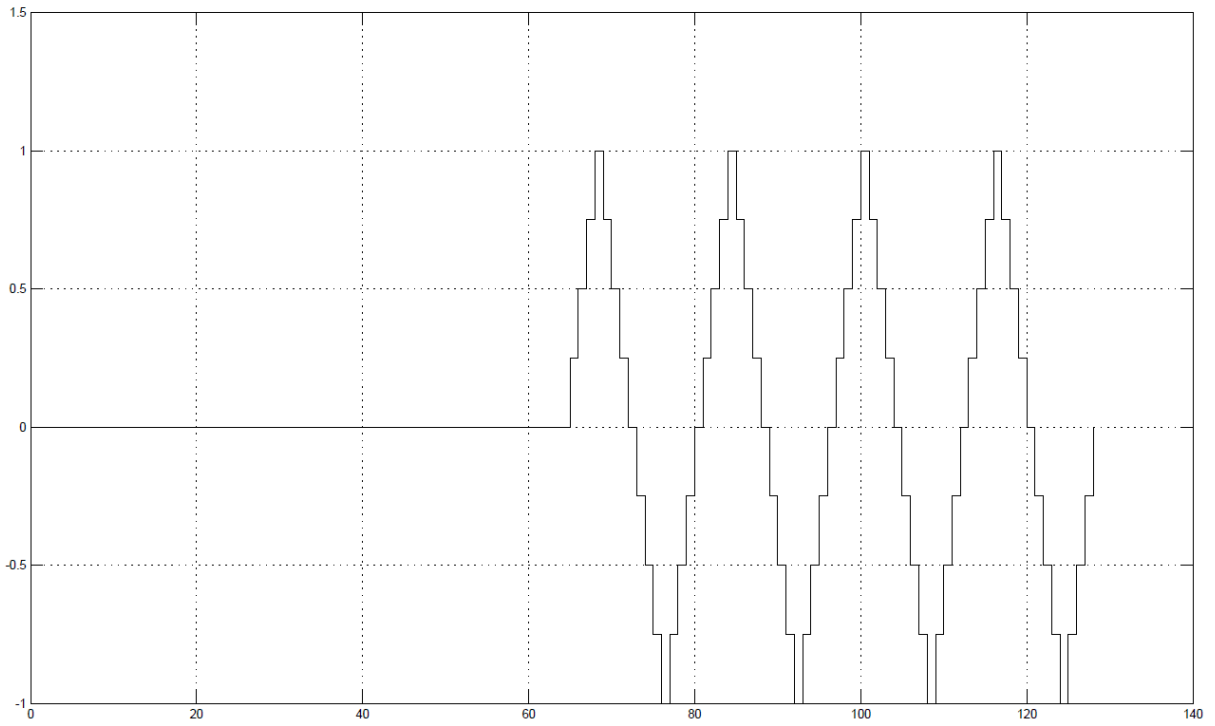
Négyszög:



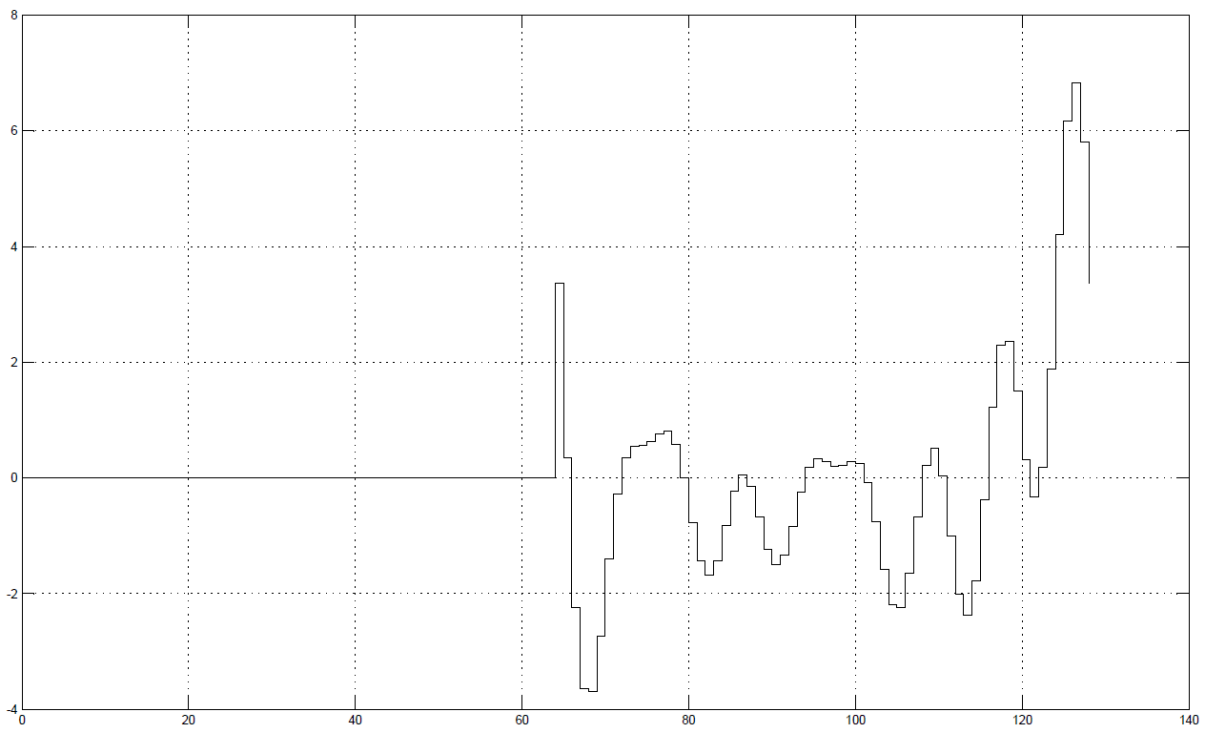
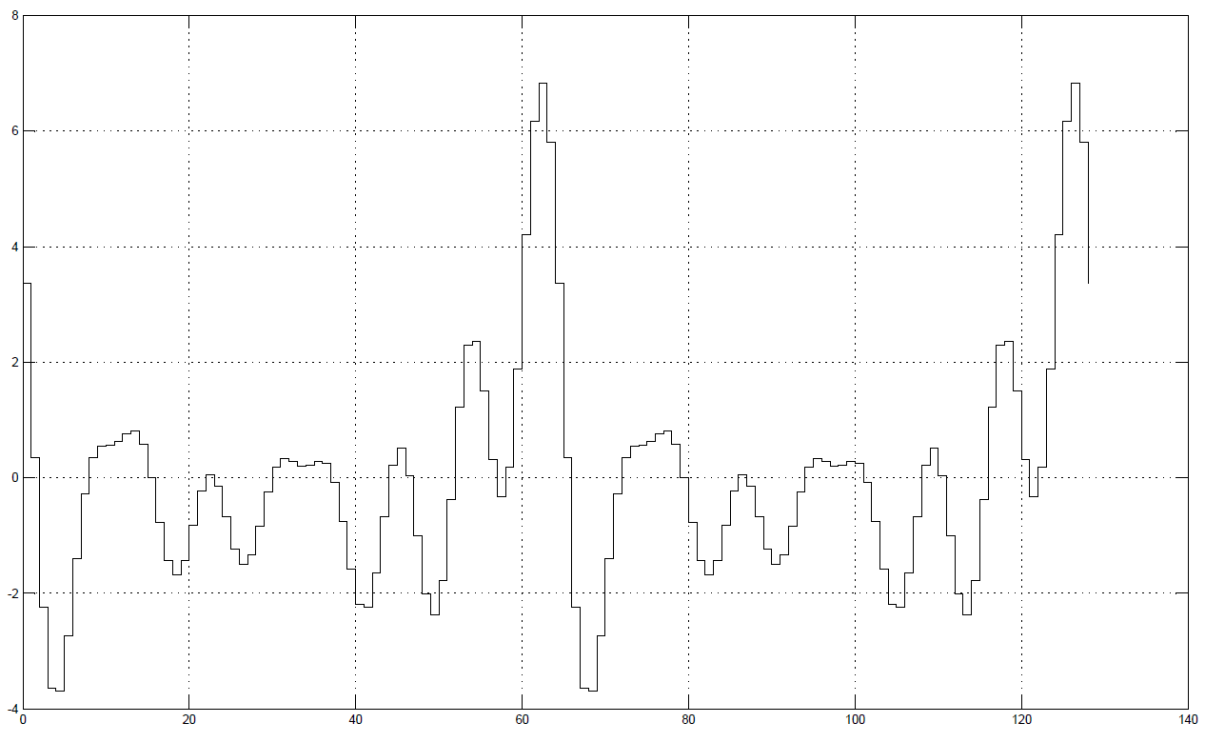


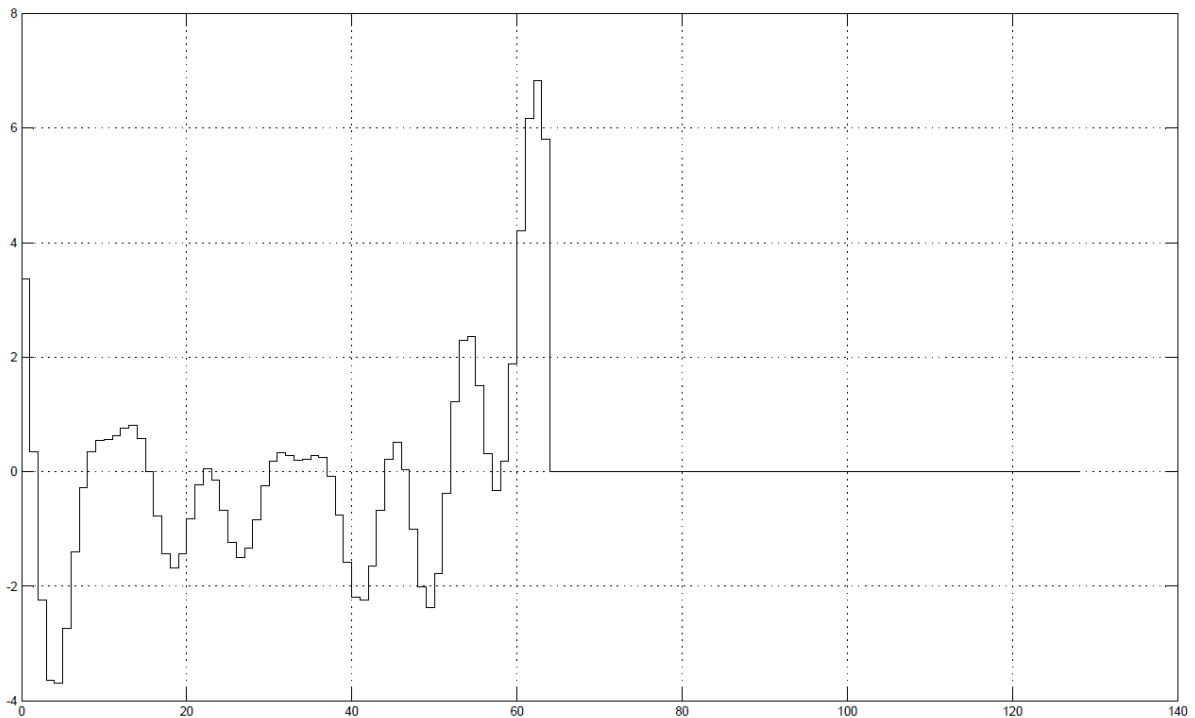
Háromszög:





Multiszínusz:





Látható, hogy N lépés után kioltja a hibát mindhárom jel esetén is.

6. Feladat - Amplitúdó- és fázis karakterisztika

Célszerű a multiszínusz jelet választani, mert Fourier transzformációja „sok vonal”, ez reprezentálja a különböző frekvenciákat, így könnyen, egy lépésben meg lehet határozni vele egy rendszer fázis- és amplitúdókarakterisztikát. Ilyen mérés volt Laboratórium c. tárgy kereteiben is.

```
szaml = [0 (1-p^2) 0];
nev = [1 0 p^2];
rdsz = tf(szaml, nev, 1);
figure(5);
bode(rdsz);
```

Az amplitúdó- (felül) és a fáziskarakterisztika (alul):

A mért és számított görbe jellegre helyesek.

