

Méréselmélet 2. házi feladat

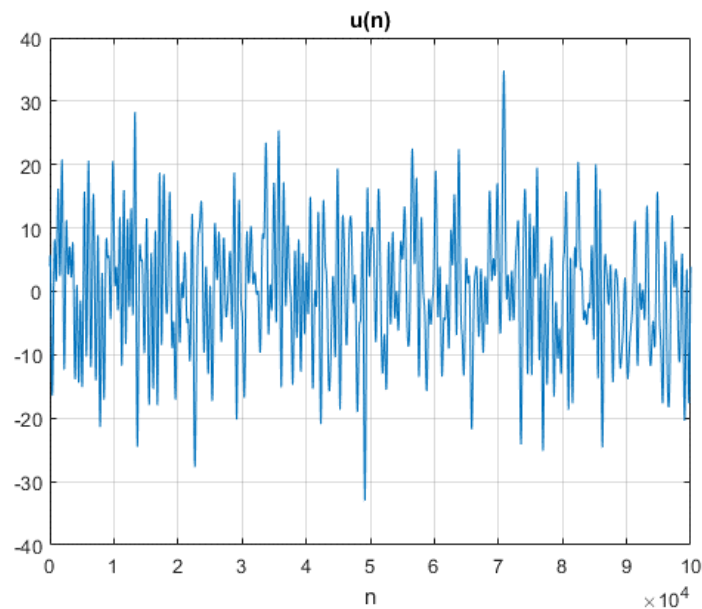
Ádám Tibor (adamtibi9@gmail.com)

ZK0MG3

1.1 Feladat

```
1 - M=1:180;
2 - r=0.83;
3 - q=0.1;
4 - P=25;
5
6 % 1.1
7 - N=100000; % minták száma
8 - u=zeros(N,1); % helyfoglalás a gerjeszt?jelnek
9 - phi=rand(1,max(M)); % véletlen fázis generálás
10 - for n=0:N-1
11 -     u(n+1)=sum(sin(2*pi*(M*n/N+phi)));
12 - end
13
14 - figure
15 - plot(0:N-1,u);
16 - title('u(n)');
17
18 % 1.2
```

A kapott jel:



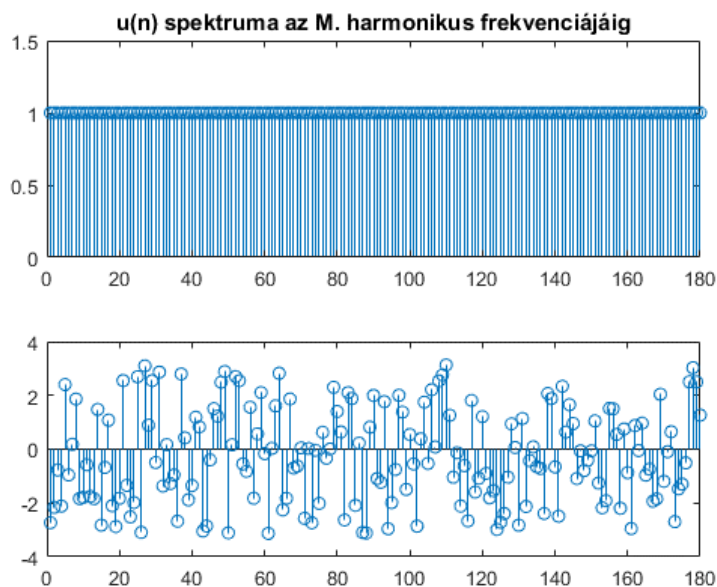
1.2 Feladat

```
18 % 1.2
19 - [A,B]=analyse(u,max(M));
20 - figure
21 - subplot(2,1,1);
22 - stem(A); % u(n) amplitúdóspektruma
23 - title('u(n) spektruma az M. harmonikus frekvenciájáig');
24 - subplot(2,1,2);
25 - stem(B); % u(n) fázisspektruma
```

Az *analyse* függvényben valósítottam meg a multiszinusz analizátort:

```
1 function [A,B]=analyse(a,M)
2 - N=size(a,1); % minták száma
3 - F=fft(a); % diszkrét fourier transzformáció
4 - A=abs(F)*2/N; % amplitúdó spektrum
5 - B=angle(F); % fázisspektrum
6 - A=A(2:M+1); % az első M frekvenciakomponens megtartása
7 - B=B(2:M+1);
8 - end
```

Az ellenőrzés eredménye:

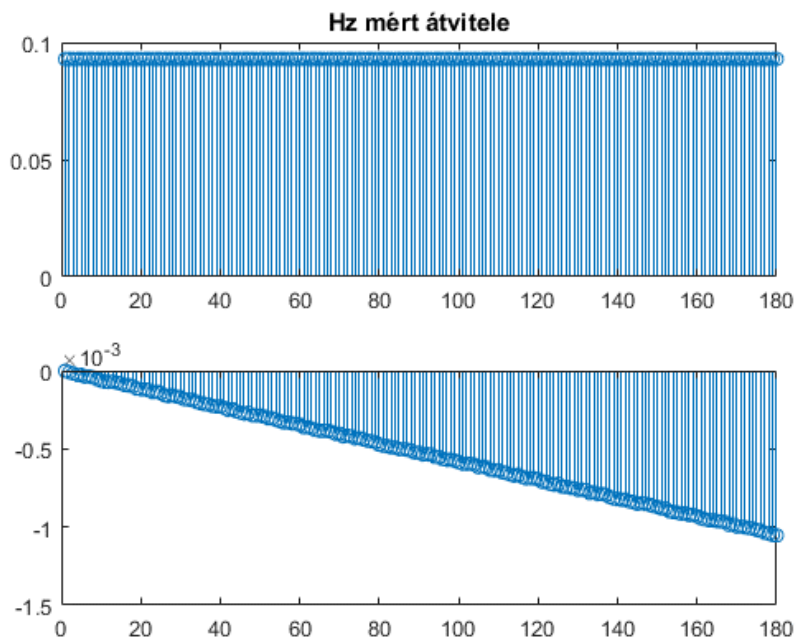


Látszik, hogy a jel $M=180$ komponenst tartalmaz, melyek amplitúdója 1, ez megegyezik a várt értékkel. A fázis $-\pi$ és $+\pi$ közötti véletlen érték.

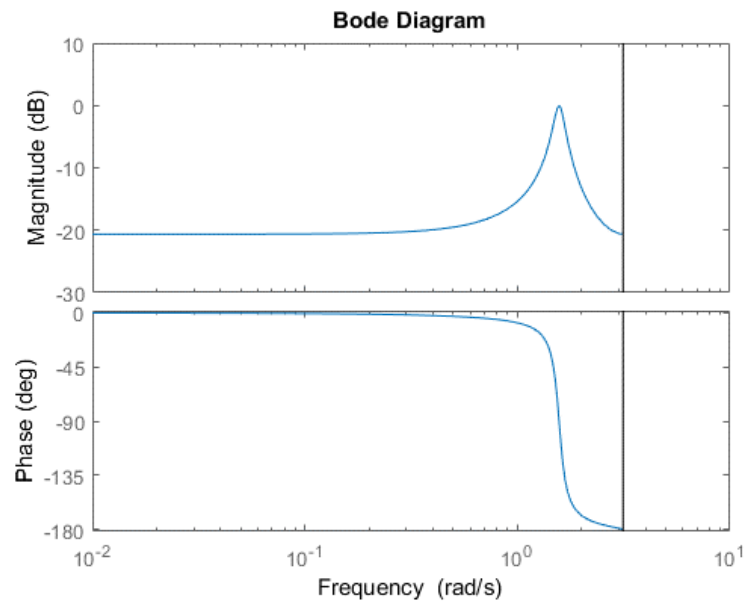
1.3 Feladat

```
28 % 1.3
29 - Hz=tf([1-r 0],[1 0 r],1); % átviteli függvény képzése
30 - y=lsim(Hz,u); % az átviteli függvény gerjesztése u(n)-el
31
32 - [Ay,By]=analyse(y,max(M)); % a kimenő jel amplitúdó- és fázisspektruma
33 - H=Ay./A; % erősítés
34 - dP=By-B; % fázistolás
35
36 - figure
37 - subplot(2,1,1);
38 - stem(H);
39 - title('Hz mért átvitele');
40 - subplot(2,1,2);
41 - stem(dP);
```

$H(z)$ átvitele a multiszínusz frekvenciákon:



A rendszer erősítése a multiszínusz frekvenciákon konstans 0.1 körül van. Ha megnézzük $H(z)$ Bode-diagramját a Matlab `bode()` függvényével is, akkor látszik, hogy a $[1/N \dots M/N]$ frekvencia-tartományon a rendszer erősítése -20 dB, vagyis 0.1. A közel nullának mért fázistolás is megegyezik a valós értékkel.



1.4 Feladat

```

44 % 1.4
45 % a 100. mintától már bőven állandósult állapotban van y(n)
46 K=100;
47
48 X=toeplitz(zeros(P,1),[0 u(1:end-1)']); % regressziós vektor előállítása
49 % a gerjesztő jel elejére bekerült egy 0, így egy mintával késleltetve lett
50 % tehát nem kell figyelni később az y és u indexelésére
51 X=X';
52 x=X(K:end,:); % (K-1)-ig tartó mintasorozat eldobása
53
54 w0=ones(P,1)/P; % a súlyok inicializálása
55 P0=eye(P); % a P mátrix inicializálása
56 w=recLS(w0,P0,x,y(K:end)); % rekurrzív LS függvény, w: súlytényezők
57 y2=x*w;
58 figure
59 plot(0:9999,y2(1:10000),'b',0:9999,y(K:K+9999),'r');
60 legend('lin.kombinátor','adaptálandó rendszer');
61 title('Az adaptálandó rendszer és a lin. kombinátor kiemenete');

```

A regressziós vektor értéke az n . ütemben megegyezik egy jól megkonstruált toeplitz-mátrix n . oszlopával. A mátrix első oszlopát nullákkal töltöttem fel, így toltam el egy mintányi idővel a gerjesztést a kimenethez képest. A rekurrzív LS becslő számításához az első házi feladatban írt függvényt módosítottam egy kicsit:

```

1  function [a ,e] = recLS(a0,P0,u,z)
2  -   N=900;
3
4      % kezdeti értékek beállítása
5  -   a=a0;
6  -   P=P0;
7      % rekurzív LS becslő egyenletei
8  -   for i=1:N
9  -       G=P*u(i,:) ' / (1+u(i,:)*P*u(i,:) ');
10 -       e=z(i)-u(i,:)*a;
11 -       a=a+G*e;
12 -       P=(eye(25)-G*u(i,:))*P;
13 -   end
14 - end

```

A becslő és a P mátrix kezdeti értékét most nem a sima LS becsléssel határoztam meg, hanem egyszerűen minden súlyt $1/P$ -vel (ahol P jelöli a súlyok számát is) inicializáltam, a P mátrix kezdeti értékének pedig az egységmátrixot választottam. A kimenet beállási idejét az idődiagramjáról olvastam le.

1.5 Feladat

Az átviteli függvényt a mértani sor együtthatói alapján írtam fel. A mértani sor összegképlete:

$$\sum_{i=0}^{\infty} a_1 q^i = \frac{a_1}{1-q}$$

Az átviteli függvényem:

$$H(z) = \frac{(1-r)z^{-1}}{1+rz^{-1}} = \frac{(1-r)z^{-1}}{1-(-rz^{-1})}$$

$$a_1 = (1-r)z^{-1}$$

$$q = (-rz^{-1})$$

A sorfejtett alak:

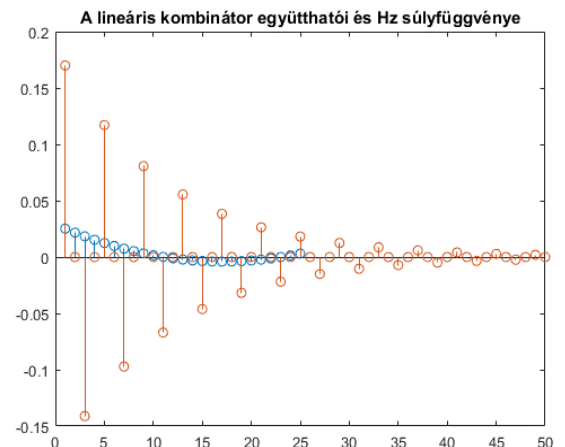
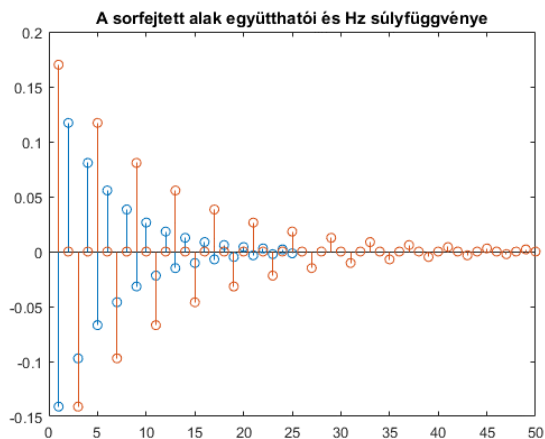
$$H(z) = \sum_{i=0}^{\infty} (1-r)(-r)^i z^{-(1+2i)}$$

```

64 - sor = (1-r)*(-r).^(1:P); % sorfejtett alak együtthatói
65 - suly = impulse(Hz,1:2*P); % Hz impulzusválaszának első 2P együtthatója
66
67 - figure
68 - stem(sor);
69 - hold on
70 - stem(suly);
71 - title('A sorfejtett alak együtthatói és Hz súlyfüggvénye');
72
73 - figure
74 - stem(w);
75 - hold on
76 - stem(suly);
77 - title('A lineáris kombinátor együtthatói és Hz súlyfüggvénye');
78
79 - y3=X*w; % lin. kombiátor kimenete a teljes u(n) gerjesztésre
80 - [Alin,Blin]=analyse(y3,max(M)); % a lin.kombinátor kimenetének spektruma
81 - figure
82 - subplot(2,1,1);
83 - % erősítések
84 - stem(Alin./A);
85 - title('Az adaptálandó rendszer és a lin. kombinátor erősítése');
86 - hold on
87 - stem(Ay./A);
88 - legend('lin.komb.','H(z)');
89 - subplot(2,1,2);
90 - stem(abs((Ay-Alin)./A)); % erősítések különbsége

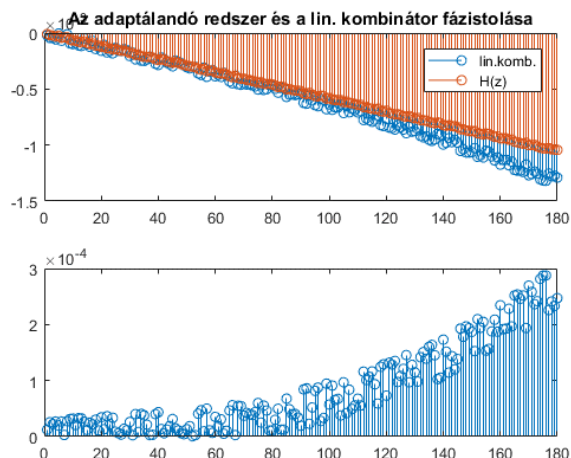
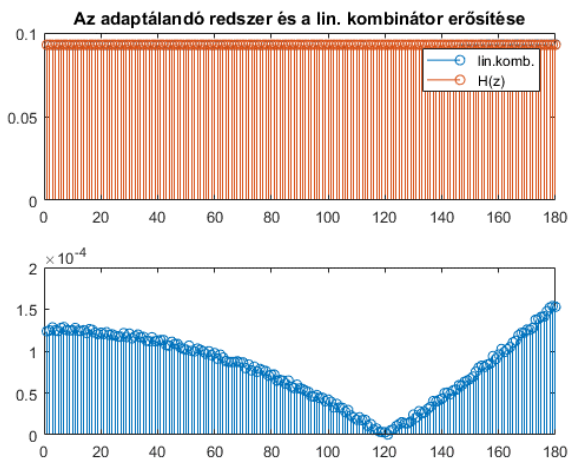
```

A sorfejtett alak első P tényezője és a lineáris kombinátor együtthatóinak összevetése $H(z)$ impulzusválaszának együtthatóival:



Az impulzusválasz páros sorszámú mintái rendre nullaértékűek, egyébként mind a három sorozat $\sin(x)/x$ alakú. A lineáris kombinátor súlyai sokkal kisebbek, és a lengése is kisebb.

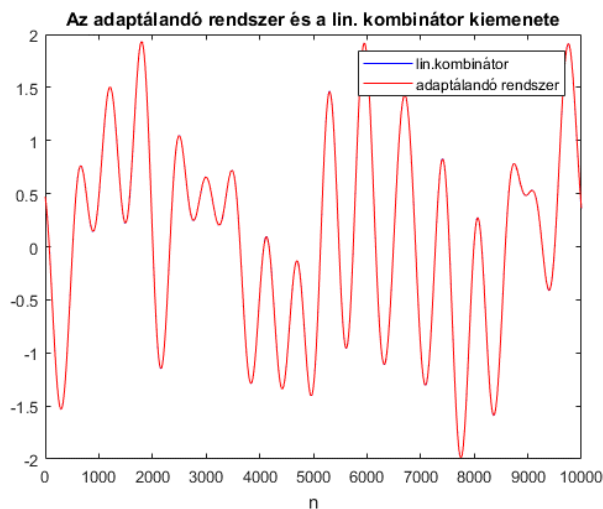
A modellezendő rendszer és a lineáris kombinátor átvitelének összevetése:



A fenti két képen látszik, hogy a lineáris kombinátor nagyon jól közelíti a kívánt modellt, hiszen az erősítés hibája minden frekvencián kisebb, mint 0.02%, a fázistolás hibája, pedig kisebb, mint 0.03%.

1.6 Feladat

Összegezve a korábban leírtakat, sikerült úgy kiszámolni a lineáris kombinátor súlyait, hogy a közelítés hibája nagyon kicsit legyen. Az alábbi képen a lineáris kombinátor és a modell kimenete látható ugyanarra a gerjesztésre:



A két jel szinte teljesen fedésben van.

A W mátrix elemeit az alábbi táblázat tartalmazza.

w1 0.021751	w2 0.019096	w3 0.01659	w4 0.014235	w5 0.01203	w6 0.009976	w7 0.008072
w8 0.006318	w9 0.004715	w10 0.003262	w11 0.00196	w12 0.000808	w13 -0.00019	w14 -0.00104
w15 -0.00175	w16 -0.0023	w17 -0.0027	w18 -0.00295	w19 -0.00305	w20 -0.003	w21 -0.0028
w22 -0.00246	w23 -0.00196	w24 -0.00132	w25 -0.00052			

A sorfejtés eredménye, a programkódok és diagramok a korábbi feladatoknál találhatóak.

2. Feladat

```

104      % 2 - LMS algoritmus
105
106      % a Wiener-Hopf egyenlet R mátrixának számítás - a bátorsági tényezőhöz
107
108      R = X'*X;
109      R=R/max(M);
110      lam = max(eig(R)); % lambda az R mátrix sajátértékeinek maximuma
111      mu = 1/(10*lam); % bátorsági tényező
112
113      W = zeros(P,N); % súlymátrix inicializálása nullákkal
114
115      for i=P:N/2
116          X=u(i-P+1:i);
117          e=y(i)-X'*W(:,i);
118          W(:,i+1)=W(:,i)+2*mu*X*e; % a súlyok rekurzív számítása LMS módszerrel
119      end
120
121      Hz_q = tf([(1-(r-q)), 0, 0], [1, 0, 0, (r-q)], 1); % H(z) csökkentett r-el
122      y_q = lsim(Hz_q,u); % az új modell kimenete u(n) gerjesztésre
123
124      % a súlyok számításának folytatása az előbbi módon
125      for i=N/2+1:N
126          X=u(i-P+1:i);
127          e=y_q(i)-X'*W(:,i);
128          W(:,i+1)=W(:,i)+2*mu*X*e;
129      end

```

Bátorsági tényező számítása az alábbi egyenlőtlenségből:

$\mu < \frac{1}{2\lambda}$, ahol μ a bátorsági tényező és λ az R mátrix legnagyobb sajátértéke.

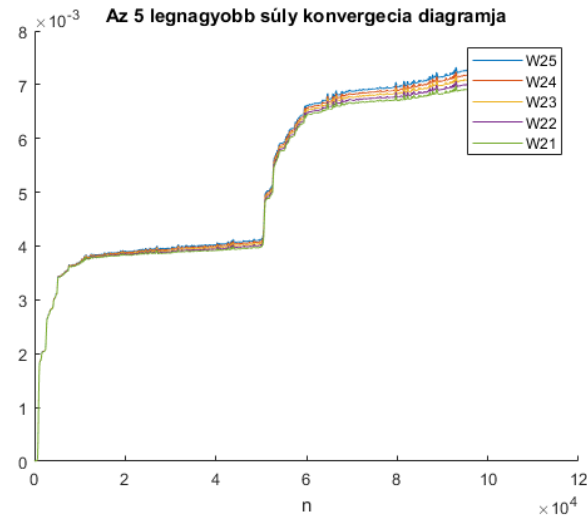
Túl kicsi bátorsági tényező esetén nagyon lassan konvergáltak a súlyok, viszont pontosan álltak be, túl nagy bátorsági tényező esetén pedig, gyorsan, de pontatlanul álltak be. A megfelelő értéket pár futtatás után sikerült megtalálnom.

A konvergencia diagramok és azok kirajzolásának kódja:


```

131 - figure
132 - % súlyok csökkenő sorrendben
133 - [maxW, maxNum] = sort(abs(W(:,end-1)),'descend');
134 - hold all
135 - for i=1:5
136 -     plot(abs(W(maxNum(i),:))); % az 5 legnagyobb súly kirajzolása
137 - end
138 - legend(strcat('W',num2str(maxNum(1:5),-1)));
139 - hold off
140 - title('Az 5 legnagyobb súly konvergencia diagramja');

```



3. Feladat

```

142 - % 3
143 -
144 - P2=6; % 6 együtthatót kell számolni
145 - X=toeplitz(zeros(P2,1),[0 u(1:end-1)']');
146 - R=X*X'/max(M); % ugyanúgy kiszámoljuk R-t a batorsági tényezőhöz
147 - lam=max(eig(R));
148 - mu=1/(10*lam); % batorsági tényező
149 -
150 - % W1 és W2 számolása közös ciklusban
151 - W1 = zeros(P2,N); % súlyok az normál modellhez
152 - W2 = zeros(P2,N); % súlyok a csökkentett r-ű modellhez
153 -
154 - % equation-error formulation és LMS egyenletek
155 - for i=P2:N-1
156 -     X1=[u(i-1:i)', y(i-3:i)']; % 2 bemeneti és 4 korábbi kimeneti minta
157 -     X2=[u(i-1:i)', y_q(i-3:i)'];
158 -     e1=y(i+1)-X1*W1(:,i);
159 -     e2=y_q(i+1)-X2*W2(:,i);
160 -     W1(:,i+1)=W1(:,i)+2*mu*X1'*e1;
161 -     W2(:,i+1)=W2(:,i)+2*mu*X2'*e2;
162 - end

```

```

164 - figure|
165 - for i=1:P2
166 -     plot(W1(i,:))
167 -     hold on
168 - end
169 - legend('a1','a2','b1','b2','b3','b4');
170 - title('Súlyok konvergencia diagramja a normál IIR rendszerhez');

174 - for i=1:P2
175 -     plot(W2(i,:))
176 -     hold on
177 - end
178 - legend('a1','a2','b1','b2','b3','b4');
179 - title('Súlyok konvergencia diagramja a csökkentett r-ű IIR rendszerhez');

```

Konvergencia diagramok:

