

Dr. Szabó Anita*

FIR és IIR szűrők tervezése digitális jelfeldolgozás területén

A Szabadkai Műszaki Szakfőiskola oktatójaként kutatásaimat a digitális jelfeldolgozás területén folytatom, ezen belül a fő érdeklődési területem a FIR és IIR szűrők. A magiszteri munkámat és doktori tézisémet is ezen területen végeztem. A fent nevezett szűrők tervezésének oktatása a tananyag része a Főiskolán. A tervezést a MATLAB programcsomagban és MATLAB Simulink-ban fejlesztem és oktatom a hallgatóknak. A szimulációk során a megtervezett szűrőket digitális hangfeldolgozásban alkalmazzuk.

Digitális szűrők

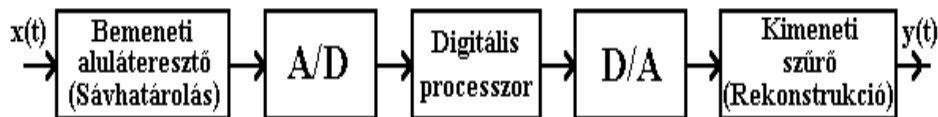
A digitális jelfeldolgozás egyik legfontosabb alkalmazási területe a szűrés. A szűrést két alapvető helyzetben használjuk a jelek helyreállítására vagy azok szétválasztására. A jelek szűrése alapvetően két fontos dolog miatt szükséges. Az egyik a jelekre rakódó nem kívánt zaj eltávolítása, a másik pedig két, vagy több egymásra modulálódott (modulált) jel szétválasztása. A digitális szűrők mindkét alkalmazási területen kiváló teljesítményt nyújtanak, és emiatt széleskörűen alkalmazzák is őket a legkülönbözőbb feladatokra.

Helyreállítást akkor alkalmazunk, ha a jelünk valamilyen okból sérültté vált. Ilyen eset lehet például a gyengébb minőségű készülékkel rögzített hangfelvétel, vagy a nem megfelelően fókuszált kamerával készített felvétel is.

Leegyszerűsítve: a szűrő egy hálózat, amely a bemenőjel hullámalakját, frekvencia- és/vagy fázismenetét változtatja az adott feladatnak megfelelően.

A digitális szűrő egy első- vagy másodrendű diszkrét hálózat átviteli függvényét, egy ennek megfelelő algoritmussal, egy adott hardver eszközre írt szoftverrel valósítja meg. Ha a kimenő jel a bemenő jelet állandó késleltetéssel követi, akkor real – time megvalósításról beszélünk.

* Dr. Szabó Anita, főiskolai tanár, Szabadkai Műszaki Szakfőiskola, Szabadka



1. ábra

A digitális szűrő egyszerűsített blokkvázlata látható

A digitális szűrő előnyei

- Megvalósíthatók vele olyan átviteli karakterisztikák, amelyek analóg eszközzel nem lehetségesek.
- Nem érzékenyek a környezeti hatásokra, ezért a rendszeres utánhangolásokra nincs szükség.
- Adaptív szűrési algoritmusok is megvalósíthatóak vele.
- A hardver sokszorozása nélkül is szűrhetünk több bemenő jelet ugyanazzal a szűrővel.
- Egyaránt tárolhatjuk a szűrt és a szűretlen jeleket a további feldolgozásra.
- Analóg szűrőknél nagyobb pontosság érhető el, ezt csak az alkalmazott szóhossz korlátozza.
- Nagyon alacsony frekvencián is használhatóak az analóg szűrőkkel ellentétben, ugyanis azok használata a nagyon alacsony frekvenciákon az induktivitások miatt már problémás.

A digitális szűrők hátrányai

- Sebességhatár.
- A véges szóhosszból adódó problémák.
- Tervezési idő.

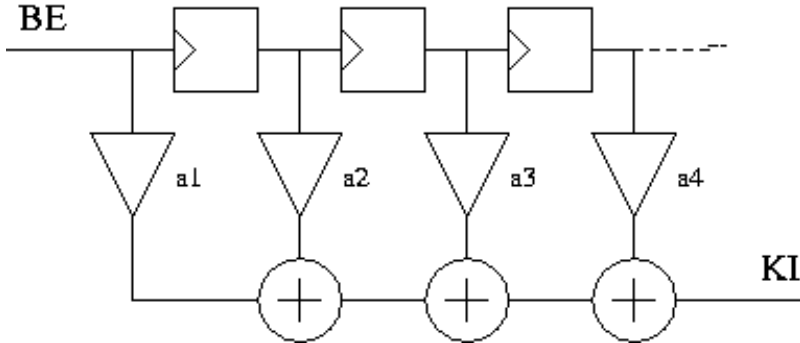
A digitális szűrők típusai

A digitális szűrők két típusra oszthatók:

- Véges impulzusválaszú **FIR** (Finite Impulse Response);
- Végtelen impulzusválaszú **IIR** (Infinite Impulse Response).

FIR szűrők

FIR (Finite Impulse Response, jelentése: véges impulzusválaszú), mivel a késleltetés után a bemenetre adott impulzus már nem hat ki az áramkörre, hiszen ez az impulzus az utolsó élvezérelt tárolót is elhagyta.



2. ábra
FIR szűrő blokkvázlata

A FIR szűrők fontosabb jellemzői:

- Fázis jelleggörbéje lineáris.
- Amplitúdó karakterisztikája lineáris.
- Mindig stabil (nem tartalmaz pólusokat).
- Nem tartalmaz visszacsatolási ágat.
- Impulzus válaszfüggvénye véges.

A FIR szűrők típusai az áteresztési tartományuk szerint:

- Aluláteresztő.
- Feluláteresztő.
- Sáváteresztő.
- Sávzáró.

FIR szűrők tervezése a Matlab nevű programban

Általános alak: $b = \text{fir1}(n, w_n, \text{'ftype'})$;

(1)

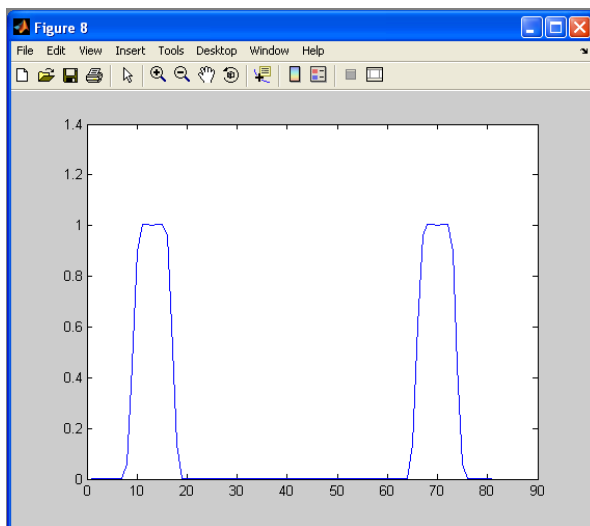
b – koefficiens, n - szűrő fokszáma, w_n – határfrekvencia, melynek értéke 0 és 1 között van, ahol az 1 a mintavételezési frekvencia felének felel meg (Nyquist frekvencia), ftype – típus (high – feluláteresztő szűrő, stop – sáváteresztő szűrő).

Ha a W_n vektornak két eleme van, $W_n = [w_1 \ w_2]$, akkor a fir1 sáváteresztő szűrőt ad vissza, ahol az áteresztési tartomány $w_1 < w < w_2$. Ha a W_n vektornak több eleme van, $W_n = [w_1 \ w_2 \ w_3 \ \dots \ w_n]$, akkor a fir1 n . rendű multiband szűrőt ad vissza, $0 < w < w_1$, $w_1 < w < w_2$, $w_n < w < 1$ tartományban.

Példa FIR szűrő tervezésére:

Tervezzünk egy sáváteresztő szűrőt, amely 80-ad rendű, ahol az áteresztési tartomány 0.2-0.4. Mutassuk be a spektrum abszolút értékét.

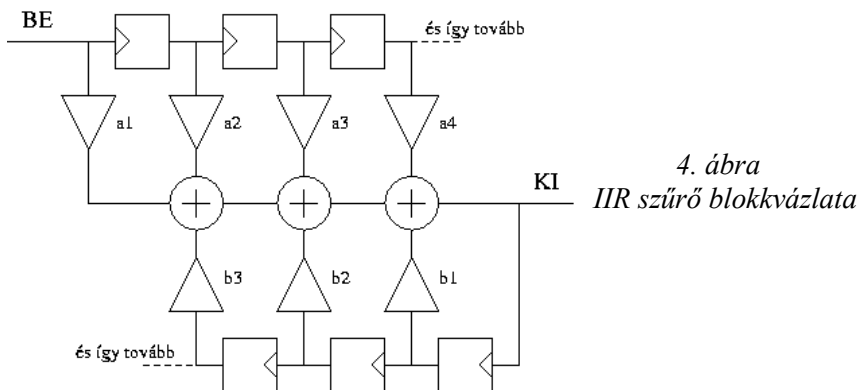
```
b=fir1(80, [0.2 0.4]);  
z=fft(b);  
zz=abs(z);  
plot(zz (1:length(zz)))
```



3. ábra
Eredmény a Matlab programba

IIR szűrők

Az IIR rendszerek tervezése alatt az analóg szűrők diszkrét tartományba való átképzését értjük.



4. ábra
IIR szűrő blokkvázlata

Az IIR szűrők fontosabb jellemzői:

- Impulzus válaszfüggvénye véges.
- Vannak pólusai és nulla helyei.
- Időben invariáns.
- Legtöbbször alkalmazott megvalósítások: közvetlen struktúra (Direkt Form), párhuzamos (Parallel Form) és sosos (Cascade Form).

Az IIR szűrők típusai az áteresztési tartományuk szerint

- Aluláteresztő.
- Felüláteresztő.
- Sáváteresztő.
- Sávzáró.

IIR implementációk

Butterworth szűrő

Általános alak: $[n, wn]=\text{buttord}(Wp, Ws, Rp, Rs);$
(2)

n - szűrő fokszáma,

wn – határfrekvencia,

Wp - áteresztési tartomány frekvenciája,

Ws – zárási tartomány frekvenciája,

Rp – csillapítás az áteresztési tartományban,

Rs – csillapítás a zárási tartományban.

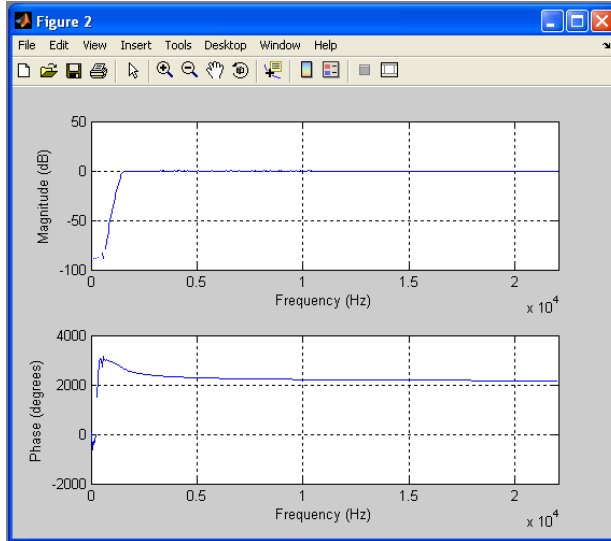
Példa Butterworth szűrő tervezésére:

Tervezzünk egy felüláteresztő Butterworth szűrőt a következő tulajdonságokkal: áteresztési tartomány frekvenciája 1000Hz; zárási tartomány frekvenciája 1500Hz; csillapítás az áteresztési tartományban 3dB; csillapítás a zárási tartományban 40dB; mintavételezési frekvencia 44100Hz.

$[n, wn]=\text{buttord}(1500/22050, 1000/22050, 3, 40);$

$[b, a]=\text{butter}(n, wn, \text{'high'});$

$\text{freqz}(b, a, 512, 44100)$



5. ábra
Eredmény

Chebyshev szűrő

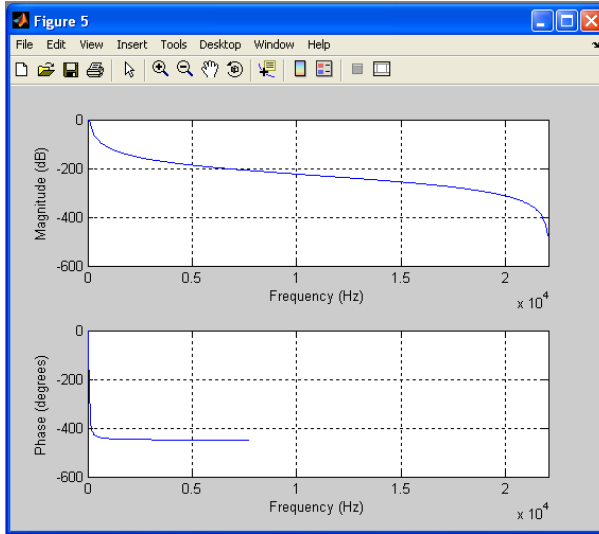
Általános alakja: $[n, wn]=\text{cheb1ord}(Wp, Ws, Rp, Rs);$
(3)

- n - szűrő fokszáma,
- wn – határfrekvencia,
- Wp – áteresztési tartomány frekvencia,
- Ws – zárási tartomány frekvencia,
- Rp – csillapítás az áteresztési tartományban,
- Rs – csillapítás a zárási tartományban.

Példa Chebyshev szűrő tervezésére:

Tervezzünk egy aluláteresztő Chebyshev szűrőt, amely a következő tulajdonságokkal rendelkezik: áteresztési tartomány frekvenciája 100Hz; zárási tartomány frekvenciája 300Hz; csillapítás az áteresztési tartományban 3dB; csillapítás a zárási tartományban 60dB; mintavételezési frekvencia 44100Hz.

```
[n, wn]=cheb1ord(100/22050, 300/22050, 3, 60);
[b, a]=cheby1(n, 0.5, wn);
freqz(b, a, 512, 44100)
```



6. ábra
Eredmény

Elliptikus szűrő

Általános alak: $[n, wn]=\text{ellipord}(Wp, Ws, Rp, Rs);$
(4)

n - szűrő fokszáma,

wn - határfrekvencia,

Wp - átérésztési tartomány frekvencia,

Ws - zárasi tartomány frekvencia,

Rp - csillapítás az átérésztési tartományban,

Rs - csillapítás a zárasi tartományban.

Példa Elliptikus szűrő tervezésére:

Tervezzünk egy sávzáró elliptikus szűrőt a következő tulajdonságokkal:

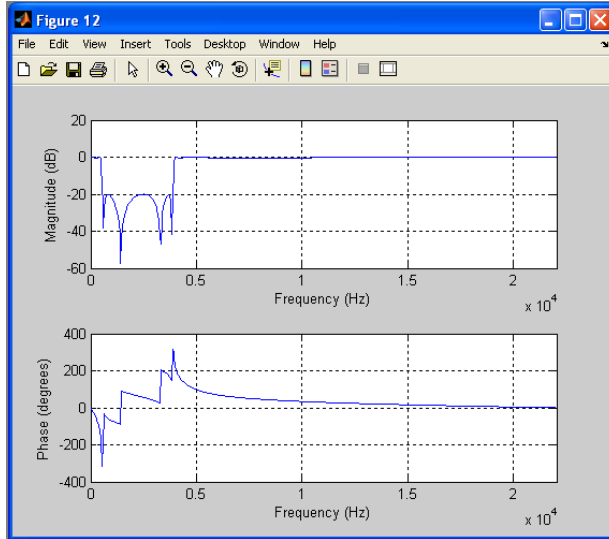
1000Hz–3000Hz;

500Hz–4000Hz; 3dB; 60dB; mintavételezési frekvencia 44100Hz.

$[n, wn]=\text{ellipord}([500/22050 \ 4000/22050], [1000/22050$
 $3000/22050], 3, 60);$

$[b, a]=\text{ellip}(n, 0.5, 20, wn, 'stop');$

$\text{freqz}(b, a, 512, 44100)$



7. ábra
Eredmény
Simulink

A rendszermodellezést és szimulációt a Simulink blokkorientált nyelv támogatja. A különböző szakterületek jelfeldolgozási és tervezési feladatait számos MATLAB-ra épülő toolbox segíti.

A Simulink grafikus felületének indítása a Matlab Command Windowban kiadott Simulink paranccsal lehetséges. A Simulink a blokkokból felépített dinamikus (differenciál és algebrai egyenletekkel leírható) rendszerek szimulációjára alkalmas program. A Simulink-et két lépesben használhatjuk fel, a modellalkotásban és a modellanalízisben. A modellt blokkokból építjük fel, amelyek különböző típus osztályokba vannak szervezve (pl. források, lineáris rendszerekhez tartozó blokkok, kijelzők, stb.). A blokkdiagram bevitele a standard grafikus programokéhoz hasonló. A modell szimulálásakor több paraméter beállítása szükséges, melyek közül legfontosabb a szimuláció időtartománya.

A struktúrák megvalósítása

A jelfeldolgozásban a digitális szűrő egy számítógépes algoritmus, ami egy bemenő számsorozatot alakít át egy kimenő számsorozattá. Az algoritmus célja, hogy a kimenő jel frekvencia- vagy időtartományban megfeleljen az adott korlátoknak, tehát a szükséges frekvencia

komponensek megjelenjenek a kimeneten, a nemkívánatosak viszont ne.

A szűrőtervezés végezhető folyamatos időben is, majd különböző transzformációk elvégzése után diszkrét idejű digitális szűrővé alakítható. Miután a digitális szűrőt megterveztük, bármelyik kanonikus alak alkalmazásával megvalósítható a fix-pontos rendszer, különböző beépített blokkok segítségével. A fix-pontos aritmetikánál túlsordulás fordulhat elő a késések bemeneténél. Az elkészített modell ráköthető egy Scope (oszilloszkóp) blokkra, amely precízen kirajzolja a jelet a meghatározott tartományban. Minden blokk teljes hozzáférést nyújt a paraméterek változtatásában, kétszer az adott blokkra való kattintás után.

Tipikus struktúrák: Közvetlen struktúra (Direct Form), Soros/kaszád struktúra (Series/Cascade Form), párhuzamos struktúra (Parallel Form).

A struktúráknál használt elemek

- az összeadó a bemeneteire érkező értékeket összegzi: $ki = be_1 + be_2$ (8. ábra/a),
- a késleltető egy ütemmel késleltetve helyezi a kimenetére a bemeneten levő jelet, amit egy konstanssal (K) való szorzás eredményez: $ki = K \cdot be$ (8. ábra/b),
- az erősítő vagy más néven szorzó, egy előre konfigurált konstanssal szorozza a jelet (8. ábra/c).

(a)  - összeadó

(b)  - késleltető

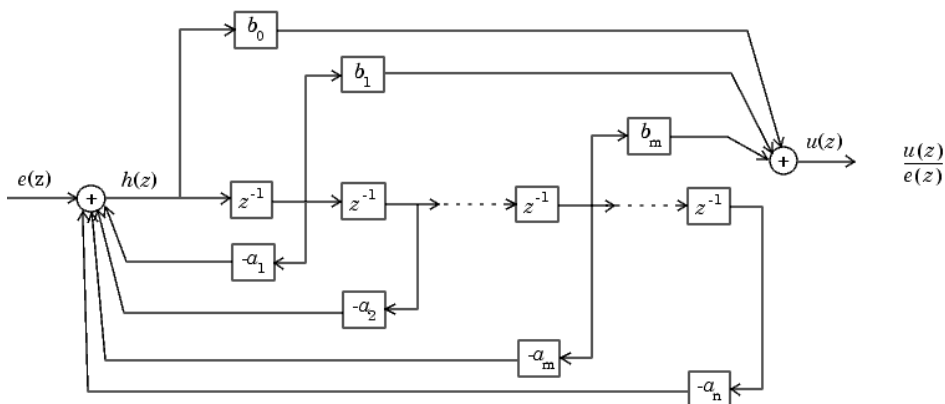
(c)  - erősítő (szorzó)

8. ábra

A struktúráknál használt elemek

Közvetlen struktúra (Direct Form II)

Általában a Direct Form II olyan struktúrát alkot, ahol az átviteli függvény együttthatói erősítő (Gain) blokkokban jelennek meg. Minimális n számú késés elemet tartalmaz, amely az átviteli függvény nevezőjének fokszámát jelenti. Ezt az alakot másképp kanonikus struktúrának is nevezik. A Direct Form II struktúra érzékeny a pólusok és zérusok együttthatóinak (a_i , b_i) kerekítési hibáira.



9. ábra
Direct Form II

Ügyeljünk arra, hogy az előrecsatolt elemek (direkt ág) a számláló együtthatóit, míg a visszacsatolt elemek (visszacsatolt ág) a nevező együtthatóit tartalmazzák.

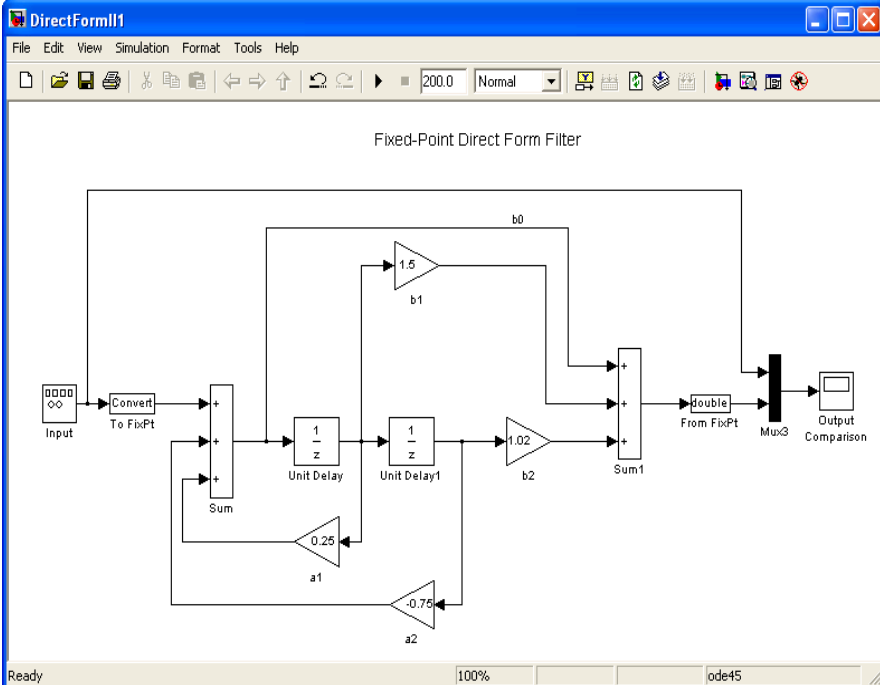
Az átviteli függvény a rendszer bemenete, illetve kimenete közötti kapcsolat meghatározására szolgál, amelyet a Direct Form II-nél a következő módon számolhatunk ki:

$$H(z) = \frac{b_0 + b_1 z^{-1} + \dots + b_m z^{-m}}{a_0 + a_1 z^{-1} + \dots + a_n z^{-n}} \quad (5)$$

Példa Direct Form II alkalmazására:

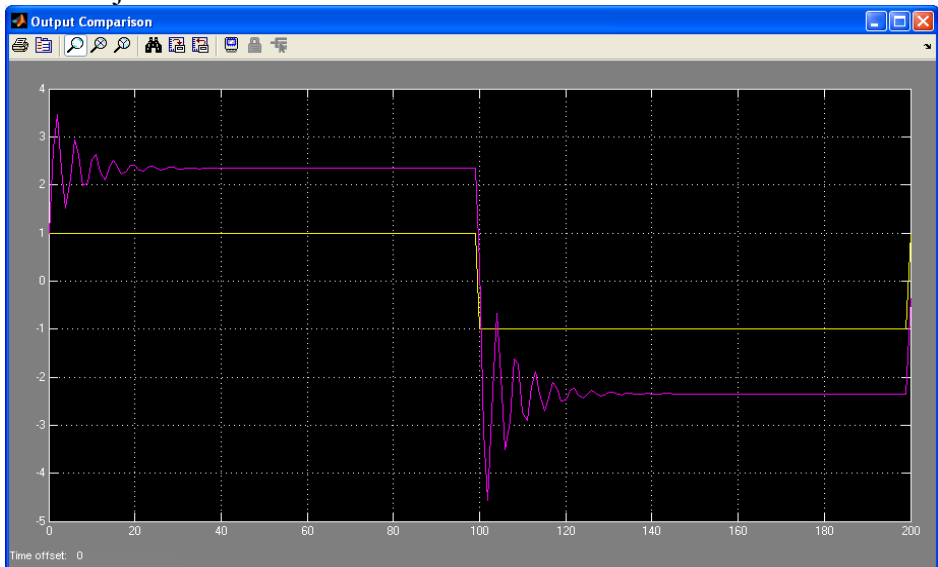
Az adott átviteli függvény alapján valósítsa meg a dinamikus rendszer modelljét és mutassa be a kimenő jelet.

$$H(z) = \frac{1 + 1.5z^{-1} + 1.02z^{-2}}{1 - 0.25z^{-1} + 0.75z^{-2}}$$



10. ábra
A dinamikus rendszer modellje

Kimenő jel:



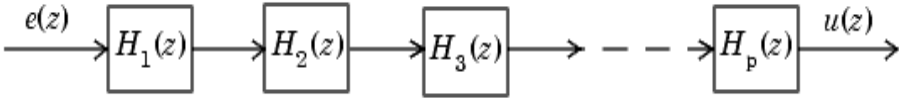
11. ábra
Kimenő jel

Soros/Kaszád struktúra (Series/Cascade Form)

Cascade Form-nál a $H(z)$ átviteli függvényt az első- és másodrendű átviteli függvény szorzásával kapjuk meg. A blokk kimenete a rákövetkező blokk bemenete. Matematikailag nem számít, hogy milyen sorrendben következnek a blokkok, mivel az eredmény ugyanaz lesz. A gyakorlatban a blokkok sorrendje viszont elengedhetetlen szempont a Cascade Form-nál.

$$H(z) = H_1(z) \cdot H_2(z) \cdot H_3(z) \cdot \dots H_p(z)$$

(6)

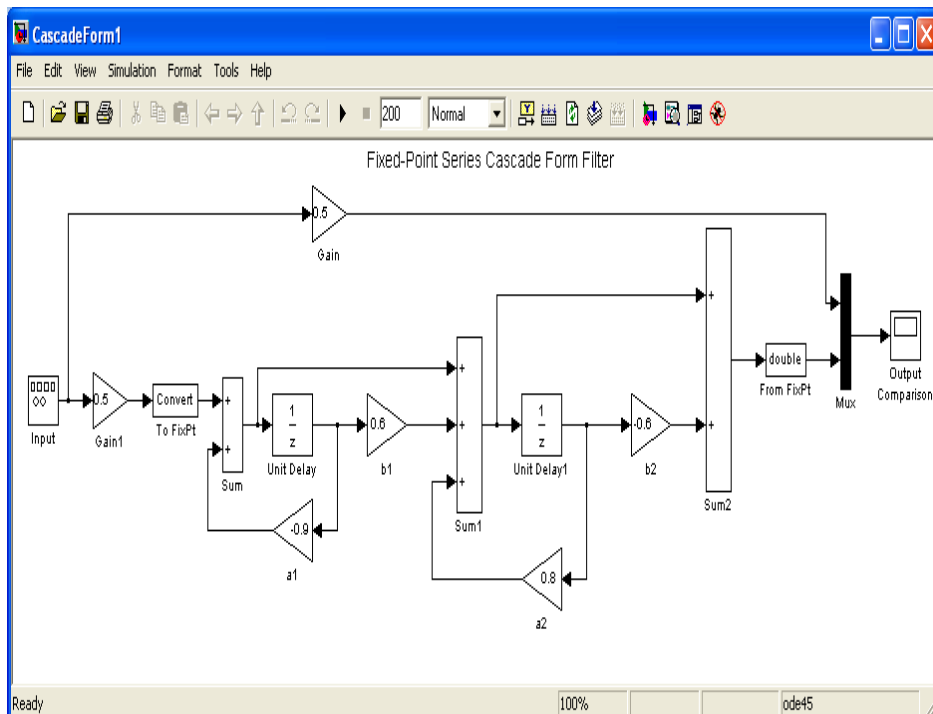


12. ábra
Cascade Form

Példa Cascade Form alkalmazására:

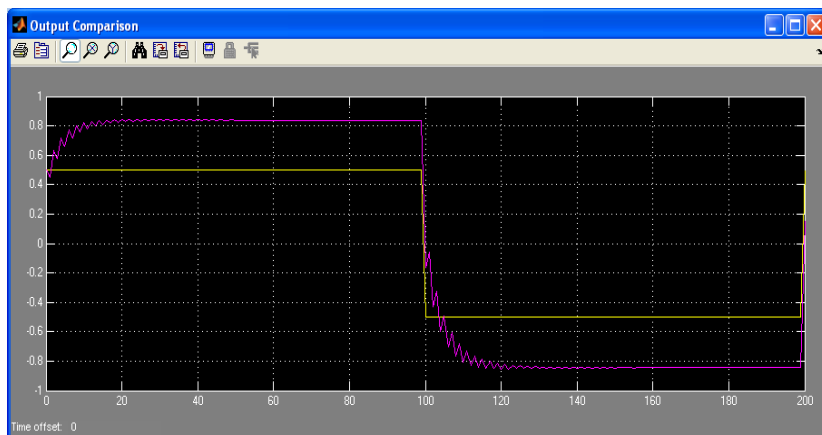
Az adott átviteli függvény alapján valósítsa meg a dinamikus rendszer modelljét és mutassa be a kimenő jelet.

$$H(z) = \frac{0.5(1 + 0.6z^{-1})(1 - 0.6z^{-1})}{(1 + 0.9z^{-1})(1 - 0.8z^{-1})}$$



13. ábra
A dinamikus rendszer modellje

Kimenő jel:



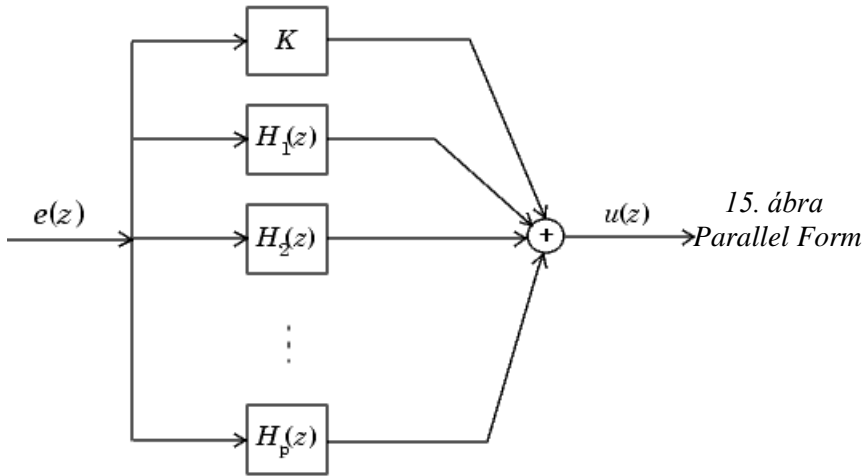
14. ábra
Kimenő jel

Párhuzamos struktúra (Parallel Form)

A Parallel Form esetében az átviteli függvényt $H(z)$ parciálisan bontjuk fel. Az átviteli függvényt a konstans együttható (K), az első- és másodrendű átviteli függvény összegeként kapjuk meg. A Sum blokk lehetővé teszi, hogy az eredmény egyszerűbb ormát vegyen fel a Cascade Form eredményétől.

$$H(z) = K + H_1(z) + H_2(z) + \dots + H_p(z)$$

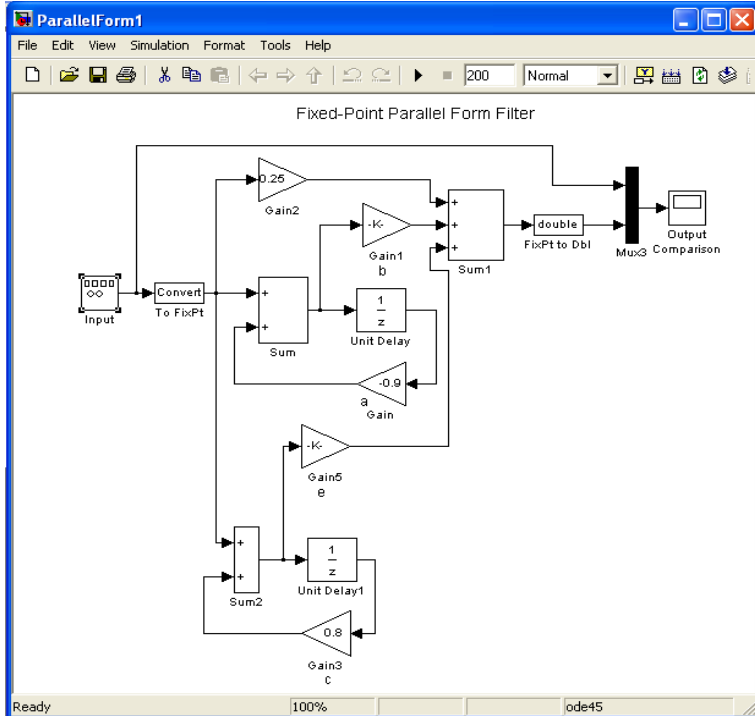
(7)



Példa Parallel Form alkalmazására:

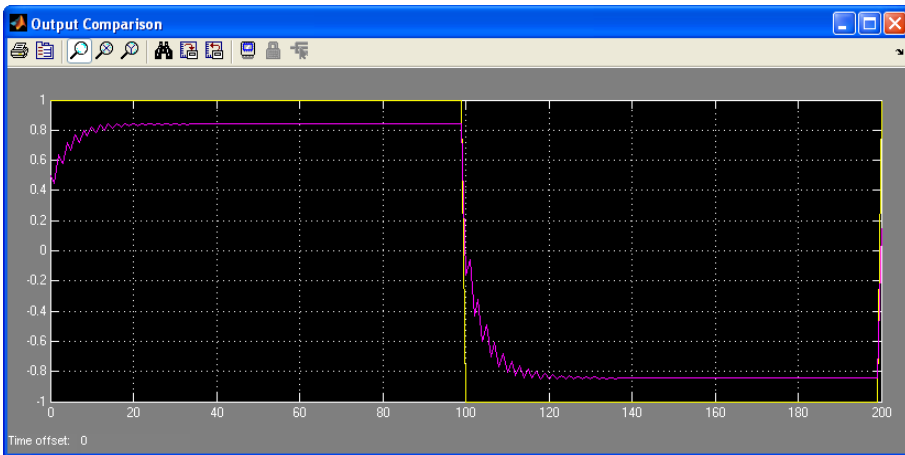
Az adott átviteli függvény alapján valósítsa meg a dinamikus rendszer modelljét és mutassa be a kimenő jelet.

$$H(z) = 0.25 + \frac{0.147}{1 + 0.9z^{-1}} + \frac{0.103}{1 - 0.8z^{-1}}$$



16. ábra
A dinamikus rendszer modellje

Kimenő jel:



17. ábra
Kimenő jel

Felhasznált irodalom:

1. Dr. Szabó Anita és Mr. Szakáll Tibor: Kép- és hangfeldolgozás gyakorlatok Matlab környezetben, Szabadka 2011.
2. Dr. Elek Kálmán: Jelfeldolgozó rendszerek jegyzet (BME, Híradástechnikai Tanszék).
3. Márki Ferenc: Digitális hangfeldolgozás, Budapesti Műszaki és Gazdaságtudományi Egyetem.
4. Digitális szűrők – Telekommunikáció szakos informatikusoknak (Miskolci Egyetem).