



Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Méréstechnika és Információs Rendszerek Tanszék

Logaritmikus felbontású szűrők kerekítési hibáinak vizsgálata

MSC ÖNÁLLÓ LABORATÓRIUM II. JEGYZŐKÖNYV

Készítette

Horváth Kristóf Szabolcs

Konzulens

dr. Bank Balázs

2014. december 19.

Tartalomjegyzék

Kivonat	3
Abstract	4
Bevezető	5
1. Warpolt szűrők	7
1.1. Közvetlen realizációk	8
1.1.1. WFIR	8
1.1.2. WIIR	10
1.2. Soros realizáció	11
1.3. Párhuzamos realizáció	12
2. Kvantálás	13
2.1. Bevezetés	13
2.2. Additív zaj modell	14
2.3. Zaj a direkt realizációjú struktúrákban	14
2.3.1. IIR direkt-I és II	14
2.3.2. WFIR közvetlen realizáció	15
2.3.3. WIIR közvetlen realizáció	16
2.4. Zaj soros struktúrákban	17
2.4.1. IIR	18
2.4.2. WFIR	19
2.4.3. WIIR	19
2.5. Zaj párhuzamos struktúrákban	19
2.5.1. IIR	20
2.5.2. WIIR	20
3. Szimulációk	21
3.1. Kvantálási zaj	22
3.1.1. WFIR	23
3.1.2. WIIR	24
3.2. Együttható-kerekítés	25
3.2.1. WFIR és WIIR közvetlen realizáció	25

3.2.2. Soros realizációk	25
3.2.3. Párhuzamos realizáció	27
4. Összefoglalás	29
Irodalomjegyzék	30

Kivonat

Audio területen gyakori jelfeldolgozási feladat egy adott rendszer átvitelének modellezése, illetve kompenzálása. A hagyományos FIR/IIR szűrők kiforrottak, rengeteg szakirodalom foglalkozik velük, de lineáris frekvenciafelbontásuk nem illeszkedik az emberi hallás logaritmus-szerű felbontásához. Ahhoz, hogy egy FIR vagy IIR szűrő alacsony frekvenciákon jól követni tudja a kívánt átvitelt, nagy foksám szükséges, ez viszont azt vonja maga után, hogy magasabb frekvenciákon az alkalmazás szempontjából szükségtelenül részletes lesz az átvitele.

A logaritmikus felbontású warped szűrők alacsonyabb frekvenciákon részletesebben, magasabb frekvenciákon pedig kevésbé részletesen követik a kívánt átvitelt. Legegyszerűbb módon úgy származtathatók a FIR és IIR szűrőkből, hogy a késleltető tagokat mindentáterestzőkre cseréljük, amelyek λ szabad paraméterét úgy állítjuk be, hogy a frekvenciategelyt közel logaritmikusra torzítsák. Az így megvalósított warped szűrő pólusai alacsonyabb frekvenciákon koncentrálódnak, ezáltal olyan átvitelt tesznek lehetővé, amely jobban illeszkedik az emberi halláshoz, mint a FIR vagy IIR struktúrák.

Digitális szűrőket valós idejű alkalmazásokban főleg jelfeldolgozó processzorokon (DSP) szokás implementálni. A DSP-k, noha speciális utasításaik segítségével gyorsabb műveletvégzést tesznek lehetővé általános célú társaiknál, legtöbbször fixpontos aritmetikával rendelkeznek, így bizonyos műveleteknél előforduló kerekítések (kvantálások) olyan zajt adnak a kimenethez, amely már nem elhanyagolható.

A kvantálási zaj különösen érdekes a warped szűrőknél, ugyanis többféle struktúra (direkt, soros, párhuzamos) szerint realizálhatóak, amelyekben különböző pontokon történik meg a kerekítés. Dolgozatomban ezen struktúrákon analízálom a kvantálási zajt a tényleges kerekítés szimulációjával, valamint az additív zajmodell szerinti elméleti alapon. Az egyes becsatolási pontokra felírt zajátvitel megfelelő összegzéséből megállapítható a kimeneti zaj spektruma. Az eredmények alapján javaslatokat teszek arra, hogy egy adott bitszélességű DSP-n melyik struktúra implementációja az optimális a kerekítési zaj szempontjából.

Abstract

Modelling or equalizing a transfer function is a common task of digital signal processing. Traditional FIR and IIR filters are commonly used, but their linear frequency resolution does not resemble the logarithmic-like characteristics of the human hearing. Thus, it requires high order filters to properly approximate the given transfer function at lower frequencies, but then the filter will be unnecessarily detailed at higher frequencies.

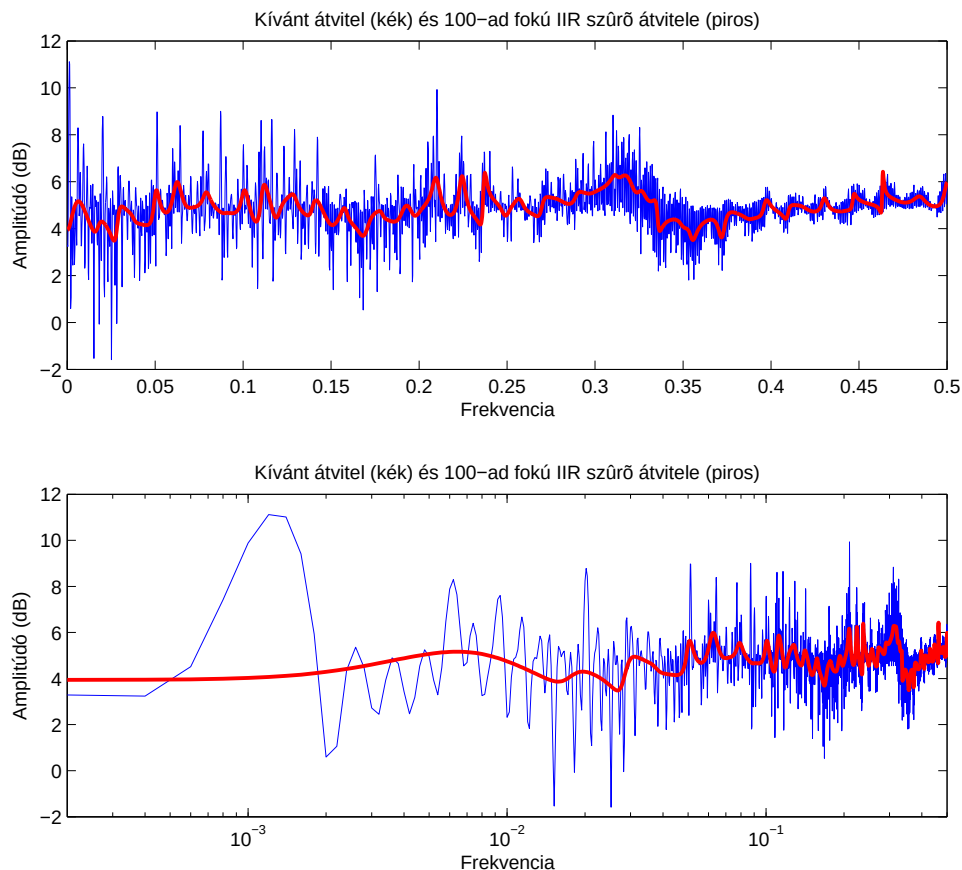
The logarithmic resolution warped filters follow the desired frequency response with higher resolution at low frequencies and vice versa. They are easily derived from FIR and IIR filters by substituting the unit delays with all-pass units. The λ free parameter of the all-pass filters determines the warping of the frequency scale, thus the poles of the filter are concentrated at lower frequencies. These types of filters fit better the characteristics of the human ear than the FIR and IIR structures.

In real-time applications, digital filters are implemented mainly on signal processors (DSPs). Although the DSPs are faster than their general-purpose counterparts in certain operations, they mostly use fixed-point arithmetic. Thus, round-off noise is added to the output of the filters, which should not be neglected.

The round-off noise is especially interesting in warped filters since they can be realized in different structures (direct, serial, parallel) in which rounding is performed in different places. In this paper I analyse the round-off noise using simulations, and additive noise model. The output noise can be calculated using the summation of elemental noise spectra. Using this results I make proposals for implementing optimal warped filters considering the round-off noise.

Bevezető

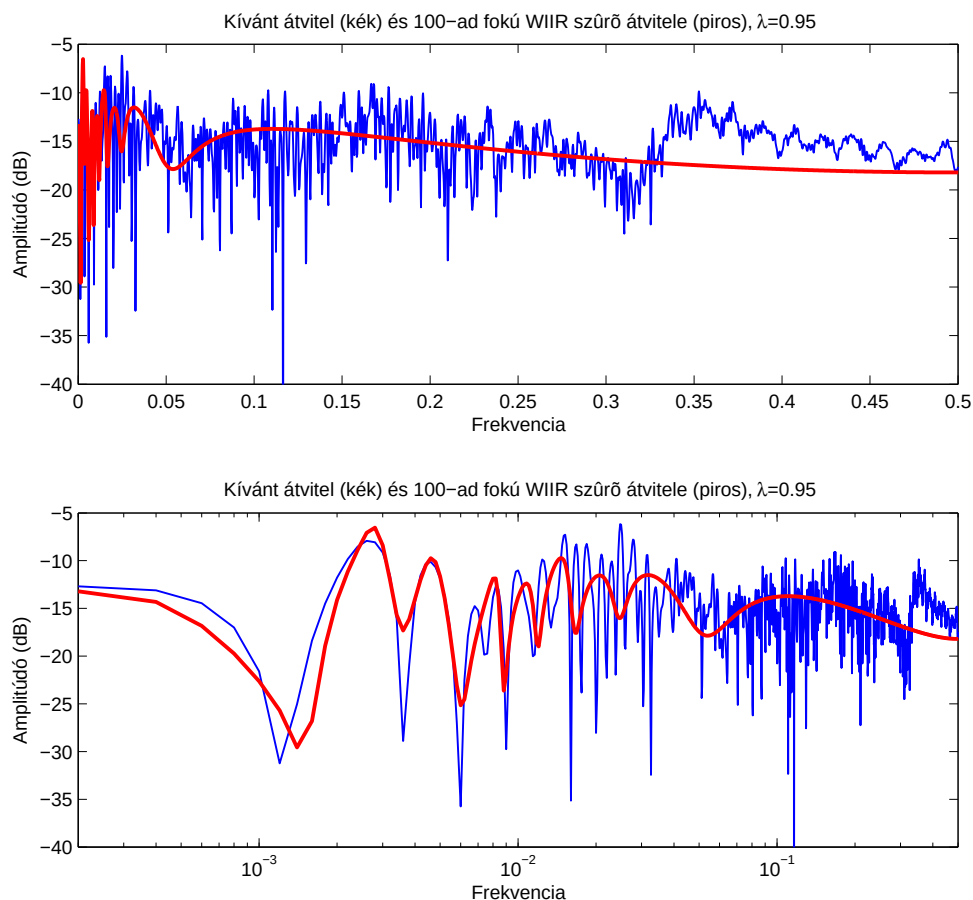
Audio területeken gyakori feladat egy adott átvitel modellezése vagy kompenzálása. A hagyományos FIR és IIR szűrők tervezési módszerei kiforrottak, kerekítési jelenségeik részletesen vizsgáltak különböző realizációk mellett is [3]. Audio jelfeldolgozásra viszont nem optimálisak: az emberi hallás logaritmikus frekvenciafelbontása miatt alacsony frekvencián pontosabban kellene követni a kívánt átvitelt, ez azonban nem teljesül egyik tervezési módszerrel sem.



1. ábra. IIR szűrő átvitele lineáris és logaritmikus frekvenciaskálán

A problémára egyik megoldás a frekvenciatartomány transzformációja: ez a warpolás. A módszer alapja, hogy a késleltetőelemeket frekvenciafüggő késleltetésekre (mindentátereztető tagokra) cseréljük, ezáltal az átvitel eltolódik a frekvenciatartományban. A transzformáció inverze is hasonlóképpen áll elő. Amennyiben a tervezés első lépéseként inverz transzformáljuk a kívánt átvitelt, majd erre illesztünk szűrőt, akkor megkapjuk a warpolt

szűrő együtthatóit. Az implementációba így bekerülnek a mindentáteresztő tagok, amelyek a transzformációt végzik. A végeredményként kapott átvitel magasabb frekvenciákon kevésbé részletes ugyan, alacsony frekvenciákon azonban jól követi a kívánt átvitelt.



2. ábra. WIIR szűrő átvitele lineáris és logaritmikus frekvenciaskálán

A warped szűrő fixpontos DSP-n történő implementációjakor kerekítési jelenségekkel kell számolni. Kvantálási zajról akkor beszélünk, amikor egy szélesebb bitszámú regiszter (akkumulátor) tartalmát alacsonyabb bitszámúvá kerekítjük. Tervezési időben kerül elő az együtthatók kerekítése, ami a szűrő átvitelét megváltoztatja, extrém esetben instabillá téve azt. Az akkumulátor regiszterek túlsordulása is gondot okoz, ezt a bemenet leskálázásával kerülhetjük el.

A warped szűrők kvantálási jelenségeivel már többen is foglalkoztak [2, 1], azonban ezt csak a WFIR közvetlen warped realizációján vizsgálták. Dolgozatomban a WFIR és a WIIR szűrők több realizációjának (direkt, soros, párhuzamos) is elemzem a kvantálási zaját.

1. fejezet

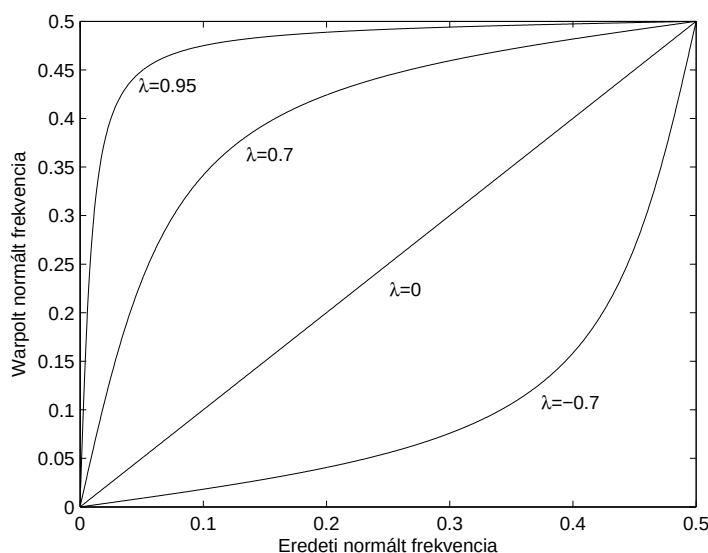
Warpolt szűrők

A frekvenciatengely transzformálására egyik, gyakran használt eljárás a warpolás. A módszer alapja, hogy a FIR/IIR szűrőkben használt egységnyi késleltetést olyan taggal helyettesítjük, amely fázistolása függ a frekvenciától. Az egyes frekvenciakomponensek amplitúdójának változatlansága érdekében kézenfekvő, hogy erre a célra mindentáteresztő tagot használjunk:

$$\Theta_\lambda(z^{-1}) = \frac{z^{-1} - \lambda}{1 - \lambda \cdot z^{-1}}, \quad (1.1)$$

ahol $-1 < \lambda < 1$. A λ paramétert warpolási tényezőnek nevezzük. Az így megvalósított transzformáció az egységkört az egységkörre képezi le és igaz rá, hogy

$$\Theta_\lambda^{-1}(z) = \Theta_{-\lambda}(z). \quad (1.2)$$



1.1. ábra. Warpolás hatása a frekvenciatartományban (normálás f_s -sel)

Belátható, hogy ha egy FIR/IIR szűrő átvitele $H(z)$, akkor a késleltetések mindent áteresztő tagokra cserélésével az új átvitel $H(\Theta(z))$ lesz. A transzformáció megőrzi a fok-

számot, a szűrő új pólusai és zérusai a

$$\tilde{p}_i = \Theta(z)|_{z=p_i}, \quad \tilde{s}_i = \Theta(z)|_{z=s_i} \quad (1.3)$$

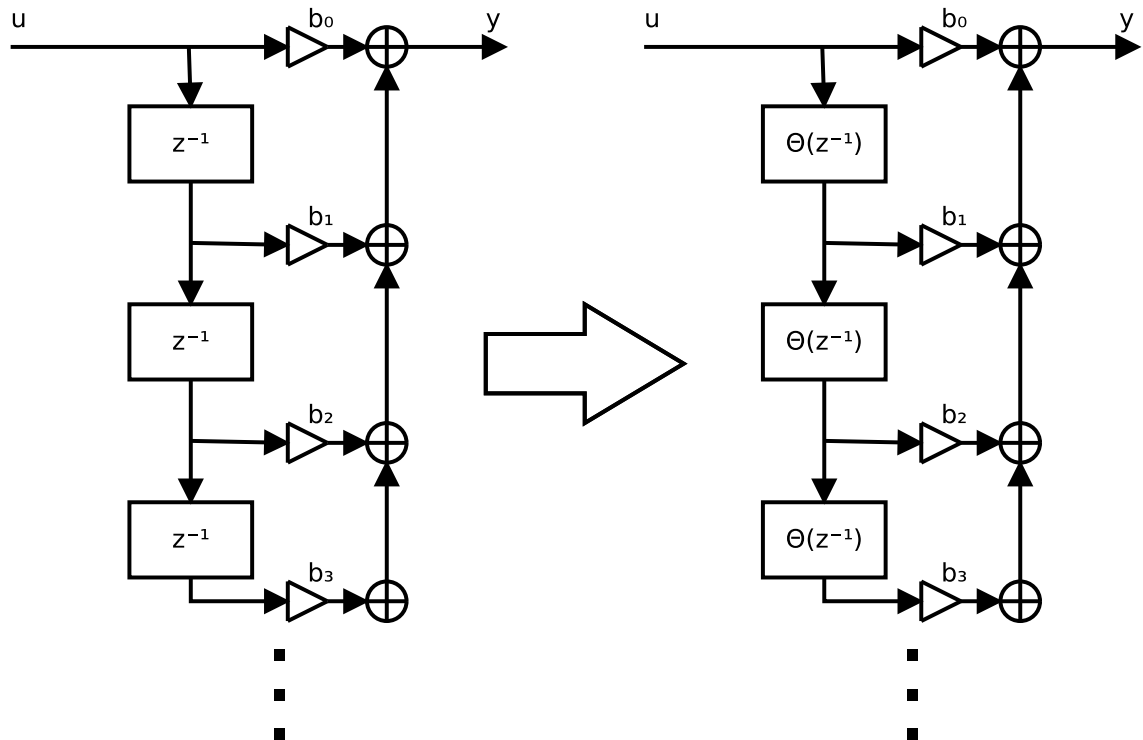
szerint állnak elő. Érdemes megjegyezni, hogy a WFIR struktúrának a pólusai nem a nullában lesznek, ezáltal az impulzusválasza nem lesz véges idejű [1].

Felmerülhet a kérdés, hogy ha ilyen egyszerűen, képletekkel kiszámíthatók az új pólusok és zérusok, akkor miért nem realizáljuk direkt struktúrájú IIR szűrővel. A módszer elméletben működik, de a gyakorlatban már viszonylag alacsony fókuszszámok (≈ 10) mellett is olyan szűrőegyütthatókat eredményez, amelyek kilógnak a dupla pontosságú lebegőpontos számábrázolás tartományából is.

A warped szűrők tervezése visszavezethető a hagyományos FIR és IIR tervezési algoritmusokra. Ha a szűrőben pozitív λ érték szerepel, akkor a tervezés első lépésében $-\lambda$ -val kell transzformálni a kívánt átvitelt. Ez egyébként megfelel az inverz transzformációnak. Az így létrejött warped átvitelre kell alkalmazni a hagyományos tervezési algoritmusokat (pl. legkisebb négyzetek módszere). A kiadódó értékek lesznek a warped szűrő együtthatói.

1.1. Közvetlen realizációk

1.1.1. WFIR

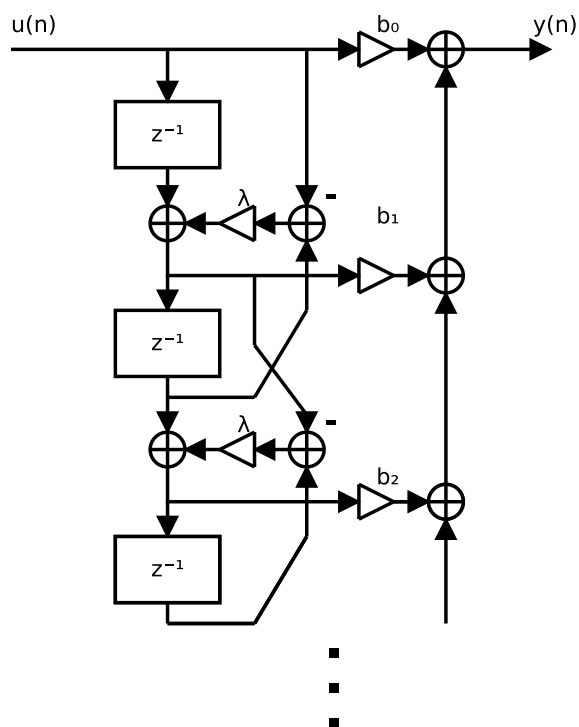


1.2. ábra. WFIR szűrő származtatása FIR-ből

Legegyszerűbb struktúrájú warped szűrő. Származtatása a FIR direkt realizációjából történik mindentátesztő tagok behelyettesítésével (1.2.. ábra). Implementációja

az 1.3. ábrán található. Látható, hogy ebben a struktúrában már vannak visszacsatolások is, ezért az impulzusválasz már nem lesz időben véges.

A struktúrából adódóan a DSP akkumulátorát és speciális utasításait egyedül a kimenet előállításánál lehet kihasználni, a λ -val beszorzásnál nem. A szükséges erőforrásigény emiatt az 1.1. táblázatban írtak szerint alakul.



1.3. ábra. *WFIR közvetlen realizáció*

Szükséges memória:	$N + 1$
Akkumulátorok száma:	1
MAC utasítások:	$N + 1$
Összeadások:	$2N$
Szorzás memória szélességű kimenettel:	N

1.1. táblázat. *N-ed fokú WFIR szűrő erőforrásigénye DSP implementációnál*

1.1.2. WIIR

A warped IIR szűrő esetén nem működik az egyszerű behelyettesítési módszer: késleltetésmentes hurok alakul ki a visszacsatoló ág miatt [5].

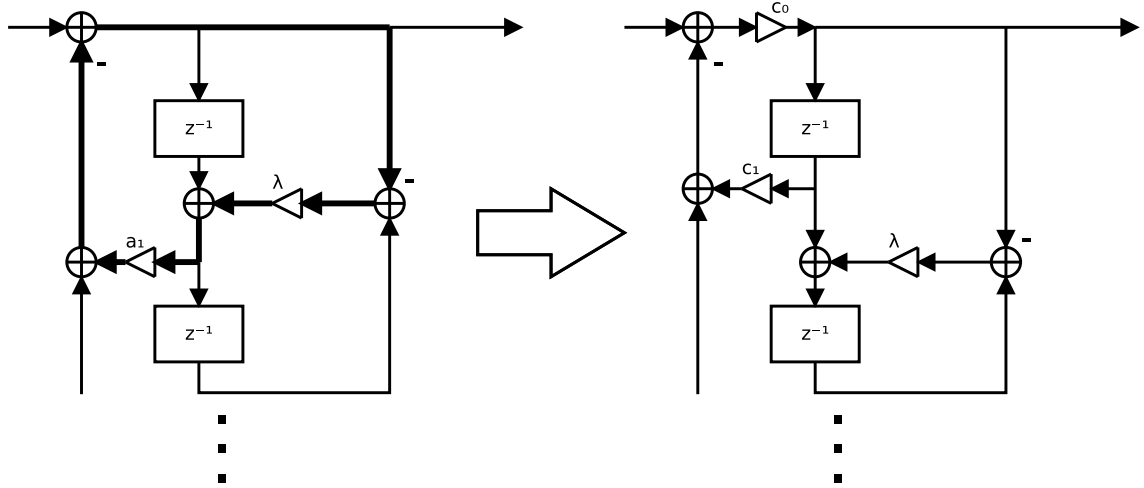
Amennyiben a az 1.4. ábra szerint módosítjuk a struktúrát, úgy az új szűrőegyütthatók [4]:

$$C_0 = 1 + \sum_{i=1}^N a_i (-\lambda)^i \quad (1.4)$$

$$C_1 = \sum_{i=1}^N a_i (-\lambda)^i \quad (1.5)$$

$$C_k = \sum_{i=k}^N a_i (-\lambda)^{i-k} - \sum_{i=k-1}^N a_i (-\lambda)^{i-k+2} \quad (1.6)$$

ahol N a foksám, a_i az i -edik együttható a visszacsatoló ágban, λ a warpolási tényező.

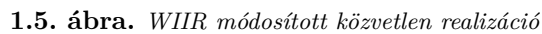


1.4. ábra. Balra: WIIR közvetlen realizációban fellépő késleltetésmentes hurok
Jobbra: hurokmentesített verzió

Az így létrehozott struktúra nagyon hasonlít a WFIR közvetlen realizációjához. Az implementáció ennek megfelelően problémás: noha a C_i tagokkal beszorzásnál és összegzésnél használható a DSP MAC utasítása, a λ -val beszorzásnál továbbra sem. Ennek eredményeként az erőforrásigény az 1.2. táblázat szerint alakul.

Szükséges memória:	$N + 1$
Akkumulátorok száma:	1
MAC utasítások:	$2N + 1$
Összeadások:	$2N + 1$
Szorzás memória szélességű kimenettel:	$N + 1$

1.2. táblázat. N -ed fokú WIIR szűrő erőforrásigénye DSP implementációnál



A szűrőket másodfokú IIR direkt tagok soros kapcsolásaként is lehet realizálni. A warped szűrők tervezésekor lényegében egy FIR/IIR szűrőt méretezünk az inverz transzformált átvitelre, majd a realizációban a másodfokú tagokkal valósul meg a kívánt eredmény. Ezt a FIR/IIR szűrőt felbonthatjuk másodfokú tagok soros kapcsolására:

Az egyes tagokba már behelyettesíthető a mindentáteresztő átvitele úgy, hogy még ne váljon numerikusan labilissá a tervezési folyamat:

Érdemes megjegyezni, hogy N -ed fokú WFIR szűrőnél az egyes másodfokú tagoknak λ -ban lesz pólusuk N -szeres multiplicitással.

1.3. Párhuzamos realizáció

A soros átvitelhez hasonlóan a tervezés során előálló IIR szűrő átvitelét felírhatjuk másodfokú tagok összegeként:

$$H(z) = \frac{\sum_{k=0}^N b_k \cdot z^{-k}}{1 + \sum_{k=1}^N a_k \cdot z^{-k}} = \sum_{i=1}^{\lceil N/2 \rceil} \frac{c_{0,i} + c_{1,i} \cdot z^{-1}}{1 + d_{1,i} \cdot z^{-1} + d_{2,i} \cdot z^{-2}} + K(z) \quad (1.9)$$

Az átvitelbe bekerülő $K(z)$ egy maradék FIR tag. Ha az átvitelben a számláló és a nevető fokszáma azonos, akkor ez a tag egy egyszerű konstans lesz. A másodfokú tagok átvitele az (1.8). képlet szerint alakul a WIIR szűrőben.

Párhuzamos realizációt ezzel a módszerrel csak WIIR szűrőnél lehet előállítani, WFIR-nél a többszörös multiplicitású pólusok miatt nem.

2. fejezet

Kvantálás

2.1. Bevezetés

Szűrők implementálásakor elkerülhetetlenek a véges számábrázolásból adódó kvantálási jelenségek, ezekből hármat különböztetünk meg: numerikus zaj (round-off noise), együtt-ható kerekítés, akkumulátor telítődés. A dolgozat keretén belül elsősorban a numerikus zajt tárgyalom, de ehhez szorosan hozzákapcsolódik az akkumulátor telítődése, amit a bemenet leskálázásával lehet elkerülni, rontva ezáltal a jel-zaj viszont. Az együtt-hatók kerekítése a szűrő átvitelét változtatják meg (akár instabillá is téve azt), de a kimeneti zajhoz nem járulnak hozzá közvetlenül. Ugyanakkor a szűrőt használhatatlannak minősíthetjük, ha az átvitele túlságosan eltér a kívánttól.

A numerikus zaj tárgyalásánál fontos leszögezni, hogy fixpontos, vagy lebegőpontos aritmetikát használunk. A ma használt jelfeldolgozó processzorok (DSP-k) többsége fixpontos számábrázolással dolgozik, így a dolgozatban is ezeket tárgyalom.

Fixpontos számábrázolásnál úgy tekintjük, hogy a kettedespont helye rögzített, ezáltal az egész és a tört helyiértékek száma is adott. A negatív számokat leggyakrabban kettes komplementum formátumban ábrázolják. Ha az ábrázolható számtartomány a $[-2^K, 2^K)$, a bitszám pedig B , akkor a legkisebb abszolútértékű szám:

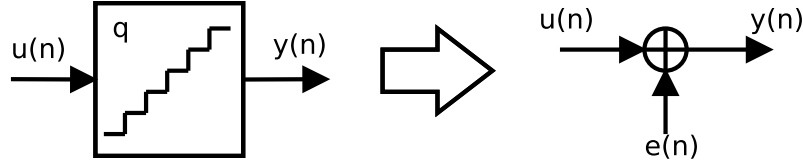
$$q = 2^{-B+K+1}. \quad (2.1)$$

Ez lényegében véve a kvantálási küszöb. DSP implementációnál K értéke leggyakrabban 0, így a számokat a $[-1, 1)$ tartományban ábrázoljuk.

Külön figyelmet érdemel a DSP-k felépítése: a gyártók a memória bitszélességét szokták feltüntetni bitszámként, azonban rendelkeznek ennél nagyobb pontosságú regiszterrel is: ez az akkumulátor. A méréseim során feltételeztem, hogy az akkumulátor szélessége legalább kétszer akkora, mint a memóriáé, így a regiszter szélességű szorzások eredményei biztosan adatvesztés nélkül beleférnek az akkumulátorba. Kerekítés csak akkor lép fel, amikor az akkumulátor tartalmát a memóriába írjuk. Fontos még megjegyezni, hogy az akkumulátor értékét közvetlenül nem lehet szorozni, azt előbb ki kell írni egy memória szélességű regiszterbe.

2.2. Additív zaj modell

A kvantálás alapvetően egy determinisztikus, nemlineáris művelet, a kezelhetőség kedvéért viszont sztochasztikus folyamatként modellezzük: a kvantálási pontban egy egyenletes eloszlású fehér zajt adunk hozzá a hálózathoz. Ezt megtehetjük, ha a kvantálási küszöb (q) sokkal kisebb, mint a jel.



2.1. ábra. Additív zaj modell

A hozzáadott fehérzaj egyenletes eloszlású a $(-\frac{q}{2}; +\frac{q}{2})$ tartományon és korrelálatlan a jellel. Várható értéke 0, szórásnégyzete (teljesítménye) pedig $\sigma_e^2 = \frac{q^2}{12}$.

A modell tulajdonsága, hogy a hozzáadott zaj szórásnégyzete megegyezik a teljesítmény spektrális sűrűségfüggvényével. Ezt egy adott frekvenciatartományra integrálva (összegezve) kiadódik a tartomány zajteljesítménye.

A különböző struktúrákra alkalmazott zajmodellnél egy elhanyagolással éltem: úgy modelleztem a szűrőket, hogy a kimenet az akkumulátorban keletkezik, nincs kimeneti kvantálás, az ugyanis implementációfüggő (pl. akkumulátor szélességű D/A konverter). A kimeneti kvantálás ráadásul minden struktúránál egy additív, konstans zajteljesítményt jelent, így az összehasonlítást nem befolyásolja.

2.3. Zaj a direkt realizációjú struktúrákban

2.3.1. IIR direkt-I és II

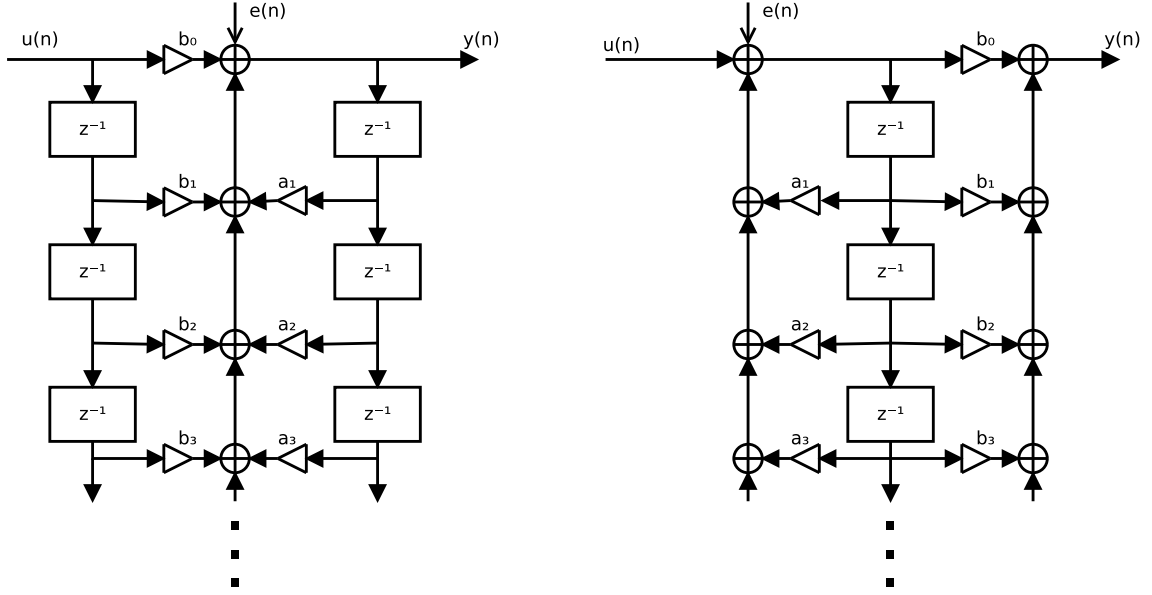
A soros másodfokú tagok miatt tárgyalni kell az IIR szűrő kvantálási zaját is. A direkt realizációjú IIR szűrőkben csak egy ponton van kvantálás, a becsatolási pontok $e(n)$ -nel jelölve a 2.2. ábrán láthatók. A transzponált struktúrákban több helyen csatolódik be a zaj, emiatt várható, hogy nagyobb zajjal bírnak [3]. A későbbiekben ezért nem tárgyalom a transzponált struktúrákat.

Vegyük észre, hogy míg a direkt-I realizációnál a zaj csak a visszacsatoló ágban jelenik meg, addig a direkt-II realizációban a teljes szűrőn áthalad. Ebből következik, hogy a zaj spektrális sűrűségfüggvénye [3]:

$$P_{\text{zaj,direkt-I}}(\omega) = \sigma_e^2 \cdot \left| \frac{1}{A(e^{j\omega})} \right|^2 \quad (2.2)$$

$$P_{\text{zaj,direkt-II}}(\omega) = \sigma_e^2 \cdot \left| \frac{B(e^{j\omega})}{A(e^{j\omega})} \right|^2 \quad (2.3)$$

A képletek alapján azt várjuk, hogy a direkt-I struktúra zaja lesz nagyobb, mert abba csak a kiemelés jelentő pólusok szólnak bele, míg a direkt-II realizációnál a zajt csillapítják



2.2. ábra. Jobbra: IIR direkt-I, Balra: IIR direkt-II realizáció

a szűrő zérusai. Összességében viszont ez nem mond még semmit a jel-zaj viszonyról: az akkumulátor túlsordulása ellen le kell skálázni a bemenetet, ezáltal a jel teljesítményét kell csökkenteni. A skálázás értéke viszont függ magától a szűrőtől és a gerjesztéstől is.

2.3.2. WFIR közvetlen realizáció

A warped szűrőknél már nem csak egy zaj becsatolási pont van, így az egyes zajátvitteleket valamilyen módon összegezni kell. Feltételezve, hogy az egyes zajok függetlenek egymástól, négyzetesen összegezhetővé válnak az átvitelek.

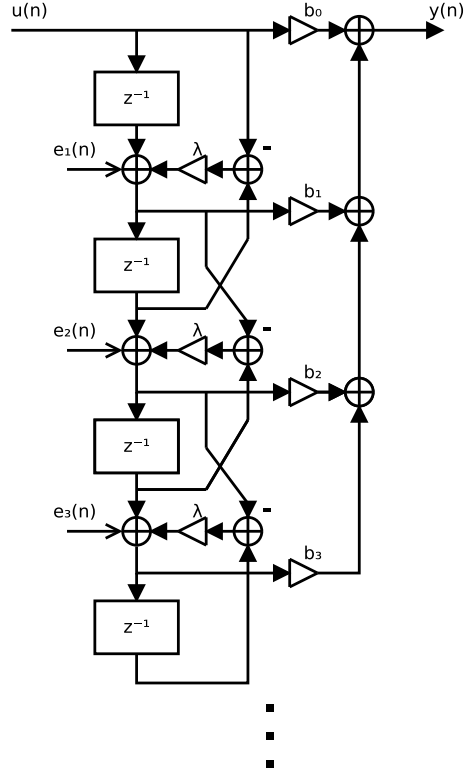
Érdemes még megjegyezni, hogy a λ értékét már eleve kvantálnak kell megválasztani a szűrőtervezés előtt.

A becsatolási pontok függenek az implementációtól, de belátható, hogy a 2.3. ábrán látható helyeken kerekítve lesz optimális a megvalósítás. A λ -val beszorzás előtt ugyanis mindenképpen ki kellene venni az akkumulátorból az értéket (szorozni csak memória szélességű adatokat lehet). A λ -val szorzás után pedig csak memória szélességű adatokat adunk hozzá a jelfolyamhoz, így a legközelebbi kerekítési pontnál csak a beszorzásnál bekerülő információ veszik el. Következésképpen kevesebb műveletet igényel, ha a λ -val beszorzást eleve memória szélességű végeredménnyel állítjuk elő.

Feltételezve, hogy az egyes zajforrások korrelálatlanak egymással, definiáljuk a részleges átviteli függvényeket a következőképpen:

$$H_i(z) = \sum_{k=i}^N b_k \cdot z^{-(k-i)} \quad (2.4)$$

Ekkor az i -edik tag zajbecsatolási pontja és a kimenet közötti átvitel $\frac{H_i(\Theta_\lambda(z))}{1-\lambda \cdot z^{-1}}$ lesz. A kimenetre vetített zajokat négyzetesen összegezve az alábbi zajteljesítmény-sűrűségfüggvényt



2.3. ábra. Zaj becsatolási pontok WFIR szűrőnél

kapjuk a kimenetre [2]:

$$P_{\text{zaj,WFIR}}(\omega) = \sigma_e^2 \cdot \sum_{i=1}^N \left| \frac{H_i(\Theta_\lambda(e^{j\omega}))}{1 - \lambda \cdot e^{-j\omega}} \right|^2 = \frac{\sigma_e^2}{|1 - \lambda \cdot e^{-j\omega}|^2} \left(\sum_{i=1}^N |H_i(\Theta_\lambda(e^{j\omega}))|^2 \right). \quad (2.5)$$

Érdemes megfigyelni a (2.5). képletet. Ugyan általános tanulságot nehéz levonni a szűrőegységgyűjtőktől való függés miatt, belátható, hogy az $1/|1 - \lambda \cdot e^{-j\omega}|^2$ alakú közös tag egy elsőfokú aluláteresztőt jelent, ezáltal a zajteljesítmény spektrális sűrűségfüggvénye alacsonyabb frekvenciákon nagyobb, magasabb frekvenciákon kisebb lesz.

2.3.3. WIIR közvetlen realizáció

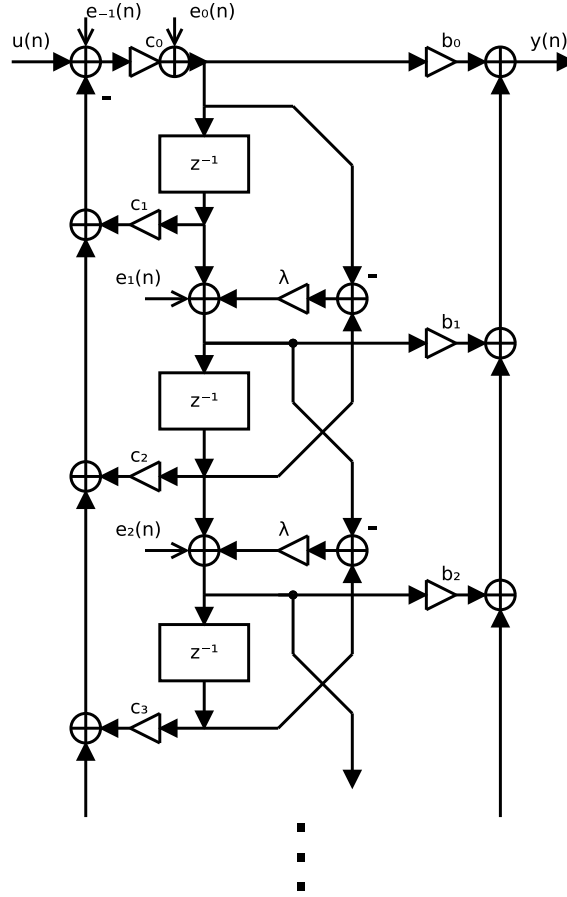
A szakirodalom nem érinti ennek a szűrőnek a vizsgálatát a numerikus zaj szempontjából.

Korábban láttuk, hogy a warped IIR szűrő direkt realizációja csak a struktúra átrendezésével lehetséges. A megvalósíthatóság érdekében módosított felépítés miatt kettővel több ponton történik a kerekítés, mint a WFIR-nél: a C_0 -val beszorzás előtt és után (akkumulátor tartalmát nem lehet közvetlenül szorozni). Ezt a két pontot a -1 és a 0 indexekkel jelöltem a 2.4. ábrán. Belátható, hogy ezekből a pontokból a kimenetre vetített zajátvitel:

$$H_{e,-1}(z) = H(\Theta_\lambda(z)), \quad (2.6)$$

$$H_{e,0}(z) = \frac{1}{C_0} \cdot H(\Theta_\lambda(z)). \quad (2.7)$$

A többi pontra már nem egyszerű felírni a zajátvitelt, így a méréseknél numerikus



2.4. ábra. Zaj becsatolási pontok WIIR szűrőnél

módszert alkalmazok: a becsatolási pontra vizsgálom az impulzusválaszt, így kapom meg az átvitelt ($H_i(z)$). Ezzel és a fenti átvitelekkel a kvantálási zaj spektrális teljesítménysűrűségfüggvénye:

$$P_{\text{zaj, WIIR}}(\omega) = \sigma_e^2 \cdot \sum_{i=-1}^N \left| H_i(e^{j\omega}) \right|^2 \quad (2.8)$$

2.4. Zaj soros struktúrákban

Egy tetszőleges átviteli függvény sorosításakor az egyes másodfokú tagok egy-egy konjugált komplex póluspárt és két zérust fognak tartalmazni, ezáltal azt várjuk, hogy ez a realizáció kevésbé lesz érzékeny az együtthatók kvantálására, mint a direkt realizáció. Látni fogjuk azonban, hogy numerikus zaj szempontjából viszont már nem optimális.

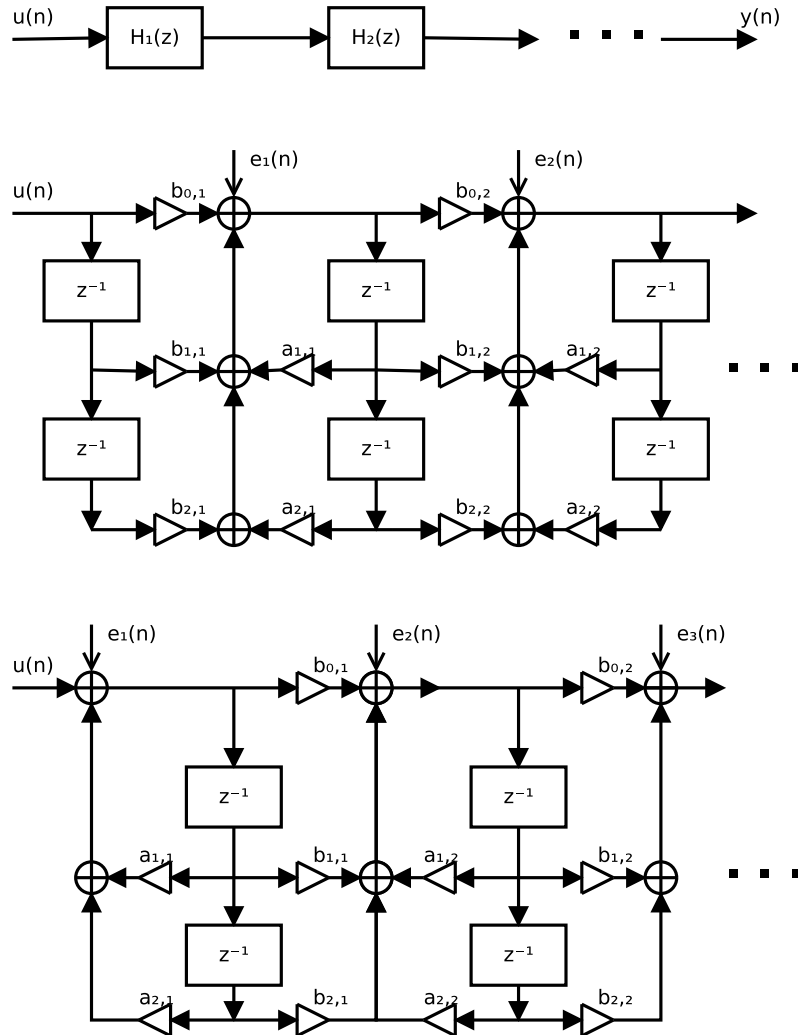
A másodfokú tagok implementálásakor fontos tényező, hogy az egyes tagokat milyen realizációban valósítjuk meg. Vizsgálataimban IIR direkt megvalósításokat vizsgáltam.

2.4.1. IIR

Mielőtt a warped szűrőkre térnénk, nézzük meg az IIR szűrő sorosítását! Az átviteli függvény a következők szerint alakul:

$$H(z) = \frac{\sum_{k=0}^N b_k \cdot z^{-k}}{1 + \sum_{k=1}^N a_k \cdot z^{-k}} = \prod_{i=1}^{\lceil N/2 \rceil} \frac{b_{0,i} + b_{1,i} \cdot z^{-1} + b_{2,i} \cdot z^{-2}}{1 + a_{1,i} \cdot z^{-1} + a_{2,i} \cdot z^{-2}} = \prod_{i=1}^{\lceil N/2 \rceil} H_i(z) \quad (2.9)$$

Vegyük észre, hogy a k -adik tag kimenete a $k+1$ -edik bemenete lesz! Ez azt jelenti, hogy ha az egyik tag erősít egy bizonyos frekvenciatartományban, úgy nem csak a hasznos jelet, hanem a kvantálási zajt is növelni fogja, emiatt nem elhanyagolható a tagok sorrendje. Ezen kívül minden egyes másodfokú tag visszacsatolásában megjelenik egy kerekítés, így ha N -ed fokú szűrőt sorosítunk, a kvantálási pontok száma $\lceil N/2 \rceil$ lesz.



2.5. ábra. Fent: soros struktúra
Középen: megvalósítás direkt-I tagokkal
Lent: megvalósítás direkt-II tagokkal

A fenti megfontolások szerint a kvantálási zaj spektrális teljesítmény-

sűrűségfüggvénye [3]:

$$P_{\text{zaj,soros}} = \sigma_e^2 \cdot \sum_{i=1}^{\lceil N/2 \rceil} \left| H_{e,i}(e^{j\omega}) \right|^2, \quad (2.10)$$

ahol $H_{e,i}(e^{j\omega})$ az i -edik ponton becsatolt zaj átvitele a kimenetre. Ennek alakja attól függ, hogy a másodfokú tagokat milyen realizáció szerint implementáltuk:

$$H_{e,\text{direkt-I},i}(e^{j\omega}) = \frac{1}{A_i(e^{j\omega})} \cdot \prod_{k=i+1}^{\lceil N/2 \rceil} \frac{B_k(e^{j\omega})}{A_k(e^{j\omega})} \quad (2.11)$$

$$H_{e,\text{direkt-II},i}(e^{j\omega}) = \prod_{k=i}^{\lceil N/2 \rceil} \frac{B_k(e^{j\omega})}{A_k(e^{j\omega})} \quad (2.12)$$

2.4.2. WFIR

Warpolt FIR szűrő soros realizációjánál ugyanazon képletek szerint fog adódni a kvantálási zaj, mint az IIR szűrőnél, de itt a pólusokra az (1.8). képlet alapján konkrét értékeket adhatunk. Az egyes másodfokú tagok átviteli függvénye:

$$\tilde{H}_i(z) = \frac{(b_{0,i} - b_{1,i}\lambda + b_{2,i}\lambda^2) + (b_{1,i} - 2b_{0,i}\lambda - 2b_{2,i}\lambda + b_{1,i}\lambda^2) \cdot z^{-1} + (b_{2,i} - b_{1,i}\lambda + b_{0,i}\lambda^2) \cdot z^{-2}}{1 - 2\lambda \cdot z^{-1} + \lambda^2 \cdot z^{-2}}, \quad (2.13)$$

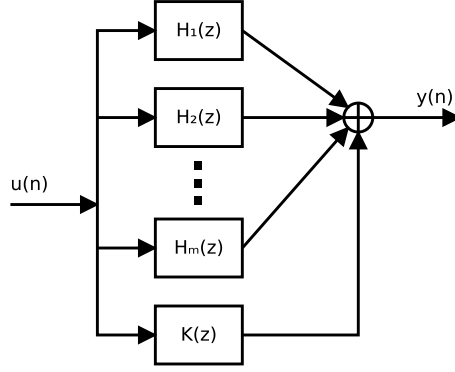
azaz mindegyik tagnak $z = \lambda$ -ban lesz egy kettes multiplicitású pólusa. Ettől azt várhatjuk, hogy a zaj elsősorban alacsonyabb frekvenciákra fog összpontosulni, mértéke pedig annál nagyobb, minél közelebb van λ az egyhez.

2.4.3. WIIR

Warpolt IIR soros realizációjánál már nem mondható el semmi általános, a zaj számításának képletei megegyeznek a (2.10)–(2.12) képletekkel.

2.5. Zaj párhuzamos struktúrákban

A párhuzamos struktúrában a direkt és a soros realizációkkal ellentétben az i -edik tagban becsatolt zaj közvetlenül a kimenetre jut. Ha a tervezett szűrő fokszáma N , akkor a párhuzamos realizáció $\lceil N/2 \rceil$ db másodfokú tagot tartalmaz, valamint egy $K(z)$ maradék FIR tagot. Ha a kívánt átvitel számlálója és nevezője azonos fokszerű, akkor $K(z)$ egy konstansszal szorzást jelent. Tekintve, hogy számításaimban a kimenet az akkumulátorban képződik, ez a FIR tag nem fog zajt hozzáadni a rendszerhez (nincs benne kerekítés).



2.6. ábra. Párhuzamos struktúra

2.5.1. IIR

A párhuzamos struktúrában a zaj teljesítmény-sűrűségfüggvénye függ attól, hogy az egyes másodfokú tagokat milyen direkt realizációban valósítottuk meg[3]:

$$P_{zaj,direkt-I}(\omega) = \sigma_e^2 \cdot \sum_{i=1}^{\lceil N/2 \rceil} \left| \frac{1}{A_i(e^{j\omega})} \right|^2, \quad (2.14)$$

$$P_{zaj,direkt-II}(\omega) = \sigma_e^2 \cdot \sum_{i=1}^{\lceil N/2 \rceil} \left| \frac{B_i(e^{j\omega})}{A_i(e^{j\omega})} \right|^2, \quad (2.15)$$

ahol B_i és A_i polinomok az i -edik tag átviteli függvényének számlálója és nevezője.

2.5.2. WIIR

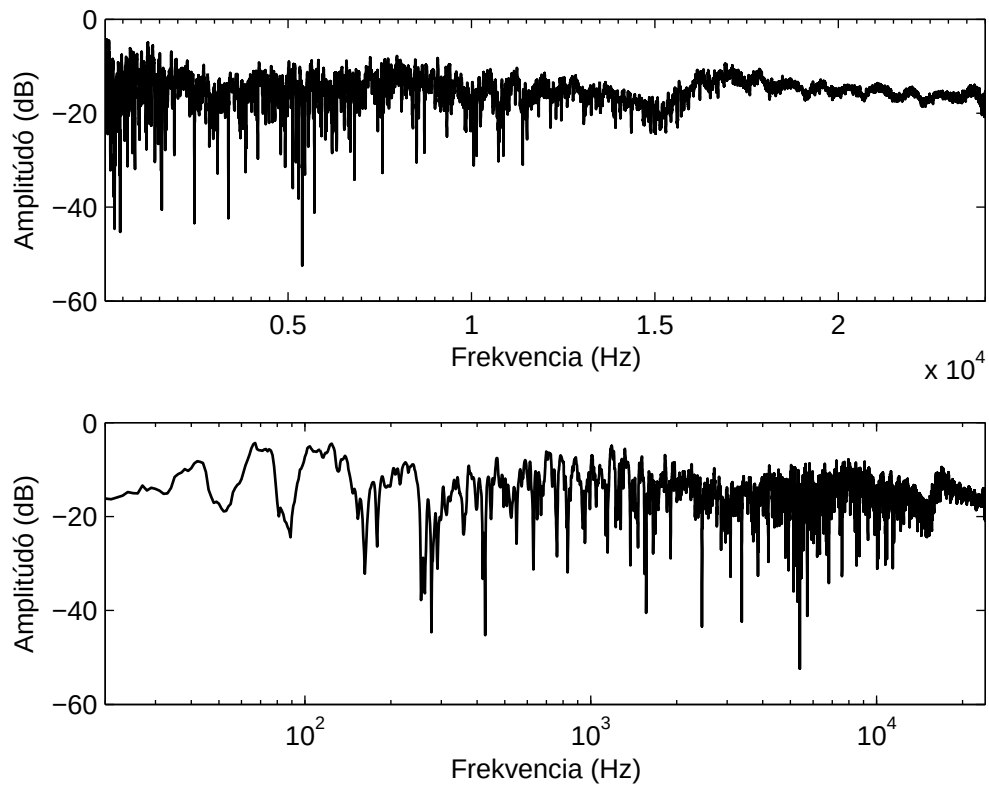
Warpolt IIR szűrő párhuzamos realizációjánál ugyanazon képletek szerint számolható a numerikus zaj, mint az IIR szűrőnél. Plusz információ, hogy az egyes pólusok és zérusok alacsonyabb frekvenciákra koncentrálódnak, így az várható, hogy a direkt-I tagokkal történő megvalósítás zaja az alacsonyabb frekvenciatartományokra koncentrálódik pozitív λ esetén.

3. fejezet

Szimulációk

A korábbi fejezetekben láthattuk, hogy miként számolható a kvantálási zaj különböző struktúrákban. Ebben a fejezetben konkrét számítások eredményei találhatók, melyek alapján következtetéseket lehet levonni a kvantálási jelenségek természetét illetően.

A szimulációk során egy szoba átvitelére terveztem szűrőket és ezek különböző realizációit vizsgáltam.



3.1. ábra. Szoba átvitel lineáris és logaritmikus frekvenciaskálán

3.1. Kvantálási zaj

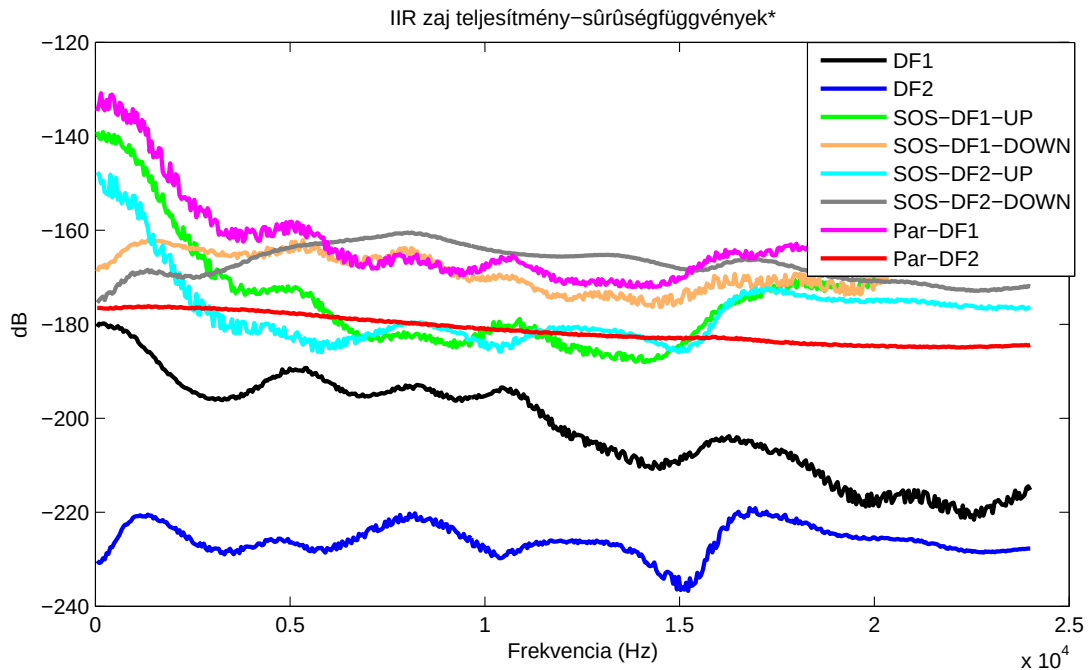
A szimulációk keretén belül a korábban tárgyalt teljesítmény-sűrűségfüggvényeket értékelem ki, ábrázolom őket és kiszámítom belőlük a zajteljesítményt. A szimulációkat elvégzem 16, 24 és 32 bites számábrázolásnál, valamint 0.5, 0.7 és 0.95-ös λ értékeknél is. A tervezett szűrő minden esetben $N = 20$ -as foksámú, a mintavételi frekvencia pedig 48 kHz volt.

A teljesítmény számításakor 0 dB-nek az egységnyi szórású zajt tekintem ($\sigma_e^2 = 1$). Amennyiben a zajteljesítmény -80 dB alatti, akkor az adott szűrőstruktúrát a zaj szempontjából jónak értékelem.

A szűrők kvantálási zajának vizsgálatánál van még egy jelenség, amit figyelembe kell venni: ez az akkumulátor túlsordulása. Ennek elkerülésére a bemenőjelet le kell skálázni, ezáltal nem fordulhat elő a túlsordulás. A skálázási tényező megállapítására fehérzaj gerjesztést használtam lebegőpontos akkumulátor mellett, majd a legnagyobb akkumulátorértékkel osztottam le a bemenetet. A vizsgálataimban az összehasonlíthatóság kedvéért a zajteljesítmény-sűrűségfüggvényekbe beépítettem ezt a skálázást: a bemenet leosztásának függvényében növeltem a kimeneti zajt.

Referenciaként megnéztem egy 20-ad fokú IIR szűrő zaját ugyanerre az átvitelre. A teljesítmény-sűrűségfüggvényeket a 3.2. ábra tartalmazza. Látszik, hogy zaj szempontjából a direkt struktúrák a legjobbak, ezt a párhuzamos DF2 követi, majd a soros struktúrák. Érdekes, hogy párhuzamos DF1 realizáció bizonyul a legrosszabbnak.

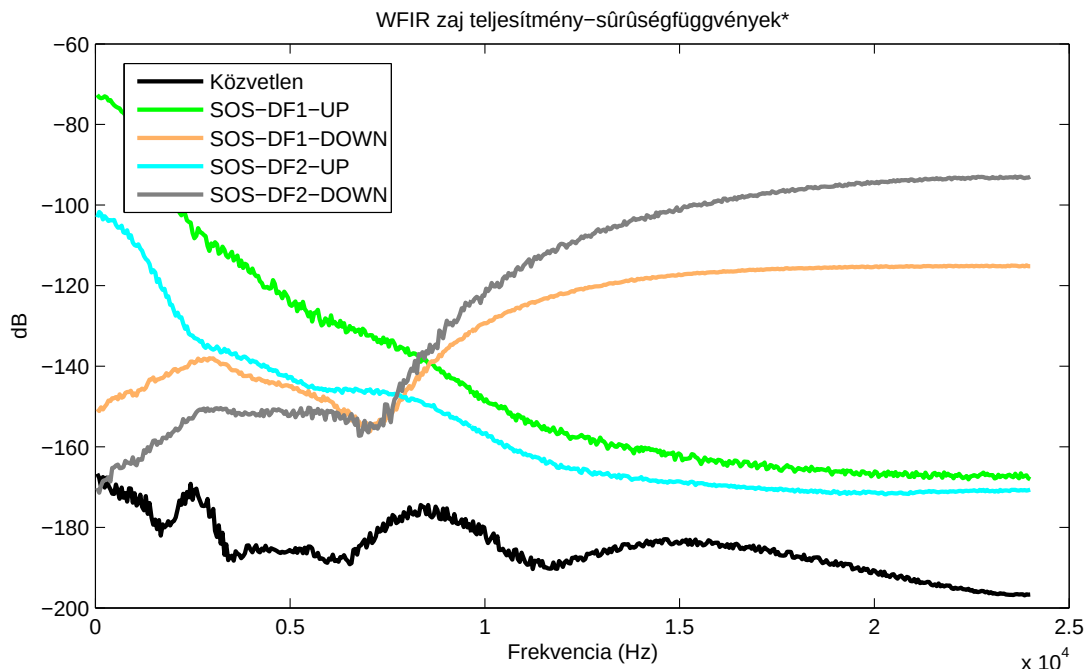
A soros másodfokú struktúránál az egyes tagok realizációján felül a tagok sorrendje is számít. Ezen belül kétfélet vizsgáltam: az első tagban van a legnagyobb pólus (UP), valamint azt, amikor az első tagban van a legkisebb pólus (DOWN).



3.2. ábra. 20-ad fokú IIR szűrő különböző realizációinak zaja 16 biten

3.1.1. WFIR

Warpolt FIR szűrőt a közvetlen realizáción felül még soros másodfokú tagokkal lehet megvalósítani. A 16 bites ábrázolás teljesítmény-sűrűségfüggvényeit mutatja a 3.3. ábra.



3.3. ábra. 20-ad fokú WFIR szűrő különböző realizációinak zaja, $\lambda = 0.5$, 16 bit

Az ábrán látszik, hogy a soros struktúra egyértelműen rosszabb zajjal rendelkezik, mint a közvetlen realizációjé. Ezen felül lényeges különbség mutatkozik a másodfokú tagok realizációjában is: a DF1 struktúrájú megvalósítás alacsony frekvenciákon nagy zajjal bír. Lévén, hogy ez a számunkra fontos frekvenciatartomány, ez nagy valószínűséggel nem lesz optimális. A DF2 realizációjú másodfokú tagoknál is látható, hogy nem hanyagolható el a tagok sorrendje. Itt csak két sorrendet vizsgáltam, de elképzelhető, hogy nem ezek között lesz az optimális.

A vizsgálatokat elvégeztem különböző λ értékekre és bitszámokra is, a zajteljesítményeket a 3.1. táblázat mutatja. A cellákban az adott struktúra esetén elérhető legkisebb zajteljesítmény található. A soros realizációnál az összes megvalósítás közül a legjobbat választottam ki és azt írtam a táblázatba.

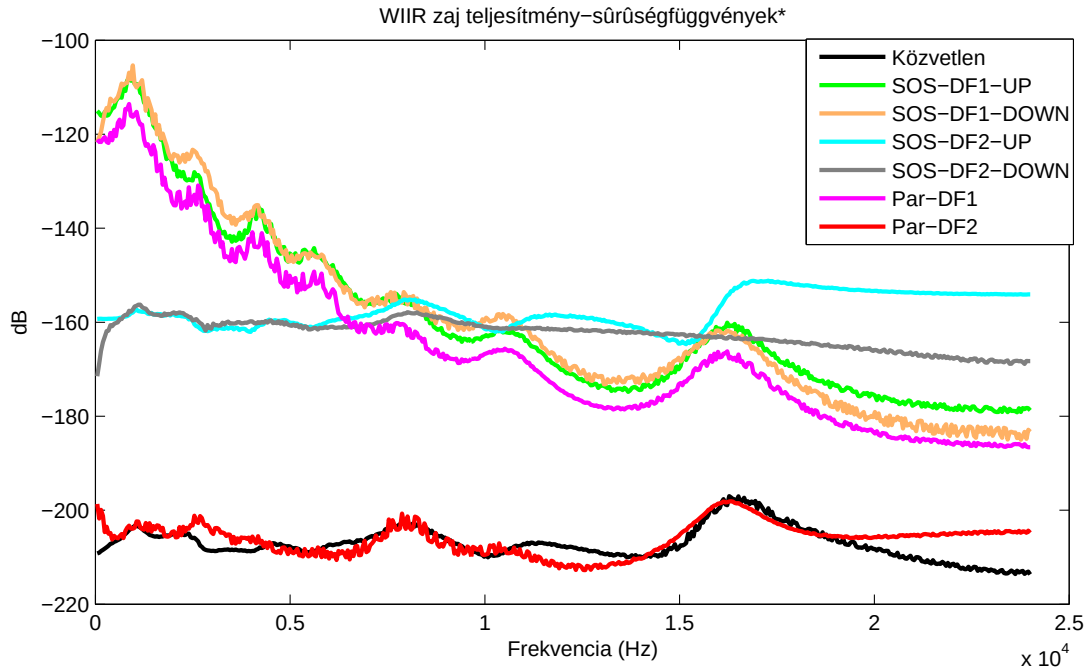
Jól látható, hogy a soros realizáció csak nagy bitszámú számábrázolásnál alkalmazható, de ott is csak kisebb λ értékekre. 16 biten WFIR szűrőt nem érdemes megvalósítani, még közvetlen alakban sem.

Közvetlen realizáció			
Lambda	16 bit	24 bit	32 bit
0,5	-53.1	-101.2	-149.4
0,7	-49.3	-97.6	-146.2
0,95	-33.4	-82.7	-129.7
Soros realizáció			
0,5	-35.4	-83.6	-131.8
0,7	-25.7	-77.7	-126.1
0,95	-25.4	-73.3	-79.3

3.1. táblázat. *WFIR szűrő zajteljesítményei (dB)*

3.1.2. WIIR

Warpolt IIR-nél a közvetlen realizáción felül létezik soros és párhuzamos struktúra is. A 16 bites ábrázolás teljesítmény-sűrűségfüggvényeit mutatja a 3.4. ábra.



3.4. ábra. *20-ad fokú WIIR szűrő különböző realizációinak zaja, $\lambda = 0.5$, 16 bit*

Az ábrán látható, hogy a soros struktúránál alig befolyásolja a tagok sorrendje a kvantálási zajt. Ez amiatt van, hogy az ábrán a $\lambda = 0.5$ esetet ábrázoltam. Magasabb warpolási tényezővel már szembetűnőbb a különbség.

Az összesített eredményeket a 3.2. táblázat tartalmazza. Itt is az adott struktúrával elérhető legkisebb zajteljesítményt vettem figyelembe. Konklúzióként hasonló vonható le, mint a WFIR szűrőnél: 16 biten nem érdemes implementálni, de fölötte is a közvetlen realizáció a legjobb. Kisebb λ értékekre azonban érdemes lehet elgondolkozni a párhuzamos megvalósításon, ugyanis 24 bittől felfelé már ez is kellően alacsony zajú, ugyanakkor viszont számításigény szempontjából jobb.

Közvetlen realizáció			
Lambda	16 bit	24 bit	32 bit
0,5	-72.6	-120.5	-168.4
0,7	-75.2	-123.2	-172.1
0,95	-59.5	-106.7	-156.1
Soros realizáció			
0,5	-47.2	-95.9	-144.8
0,7	-29.3	-77.6	-127.5
0,95	-7.0	-61.9	-110.4
Párhuzamos realizáció			
0,5	-63.3	-117.3	-165.5
0,7	-46.7	-95.8	-143.8
0,95	-26.7	-76.3	-123.4

3.2. táblázat. *WIIR szűrő zajteljesítményei (dB)*

3.2. Együttható-kerekítés

A szűrők együtthatóit véges pontossággal tudjuk csak ábrázolni. Ennek eredményeként például a dupla pontosságú lebegőpontos ábrázolás mellett megtervezett szűrőnk a fix-pontos ábrázolásra áttérve megváltoztatja a kerekítés. Más lesz az átvitel, szélsőséges esetben akár instabillá is válik a szűrő.

Az együtthatók kerekítésének átvitelre gyakorolt hatását érzékenységszámítással lehet megkapni. Szimulációimban azonban egy gyakorlatiasabb módszert alkalmaztam: a dupla pontosságú lebegőpontos szűrőegyütthatókhoz véletlenszerűen $\pm \frac{q}{2}$ -t adtam, ahol q a fix-pontos ábrázolás legkisebb abszolút értékű száma. Az így kapott átviteleket ábrázoltam, ezzel tulajdonképpen egy worst-case hibát számítottam a kerekítés hatására. Amennyiben a kívánt átviteltől való eltérés nem haladta meg a 0.5 dB-t, akkor a szűrőt jónak tekintetem együttható-kerekítés szempontjából.

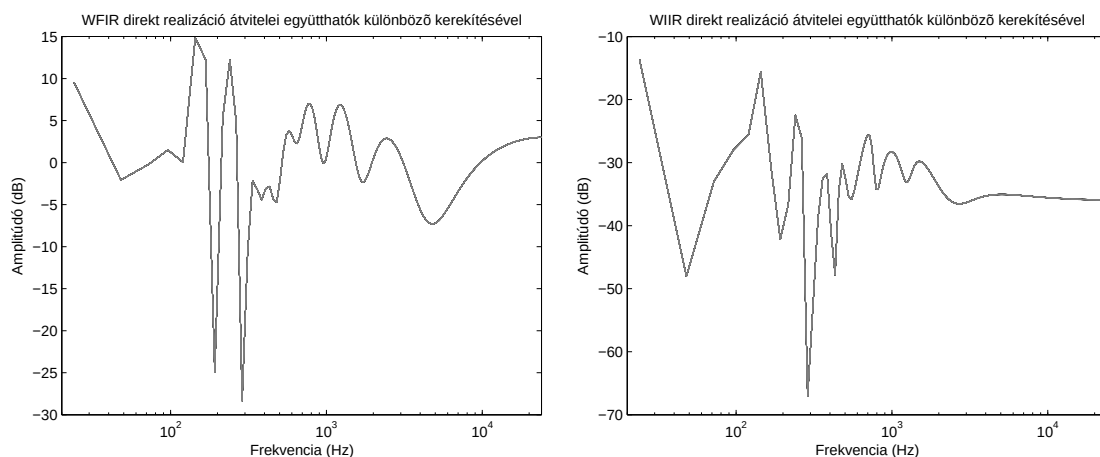
Előzetes szimulációk alapján az együtthatók kerekítése annál nagyobb hibát okoz, minél nagyobb λ . A következő λ értékekre vizsgáltam: 0.5, 0.7, 0.95.

3.2.1. WFIR és WIIR közvetlen realizáció

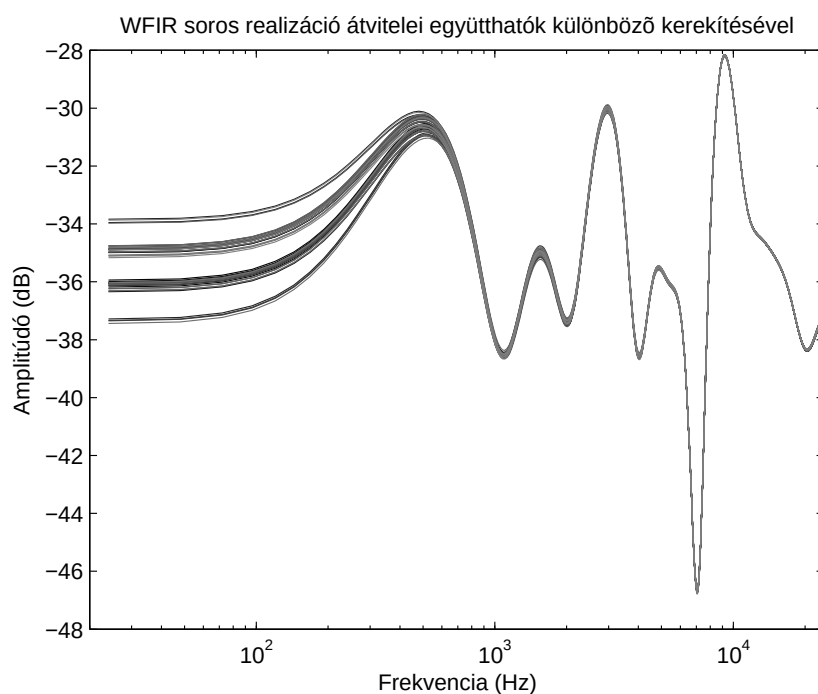
A szimulációk szerint a warped szűrők közvetlen realizációja kifejezetten robosztus az együtthatók kvantálásával szemben. A 3.5. ábrán látható, hogy még $\lambda = 0.95$ választásával sem jelentős az átvitel megváltozása 16 bites számábrázolásnál.

3.2.2. Soros realizációk

Mind a WFIR, mind a WIIR szűrőt lehet soros másodfokú tagokkal realizálni. Az egyes tagok sorrendje nem számít bele a szűrő átvitelébe, így az sem, hogy DF1, vagy DF2 formában implementáljuk a másodfokú tagot. Közös azonban bennük, hogy direkt realizációjuk, ennek pedig komoly hatása van az együttható-kerekítés végeredményére.



3.5. ábra. 20-ad fokú WFIR és WIIR szűrő érzékenysége az együtthatókra, $\lambda = 0.95$, 16 bit

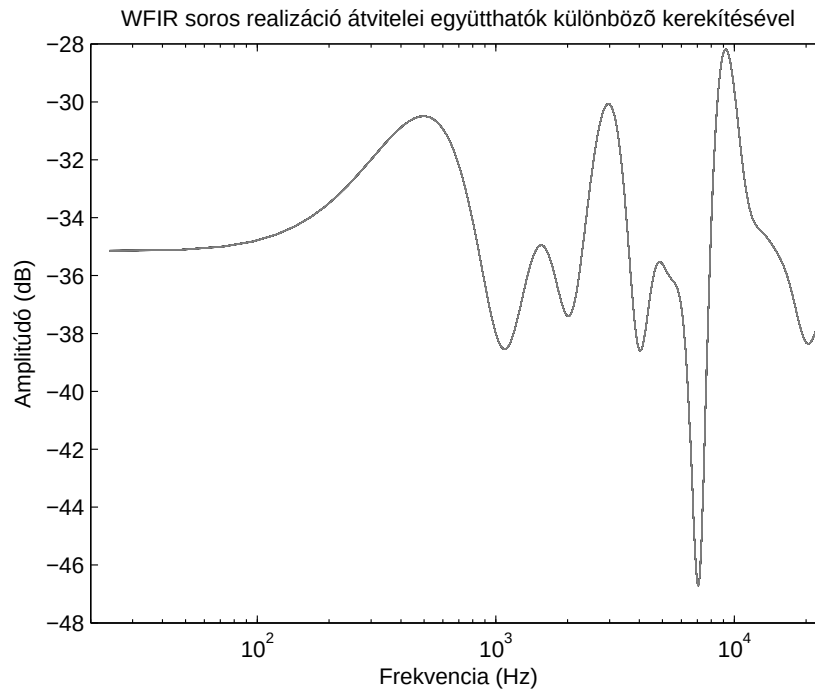


3.6. ábra. 20-ad fokú WFIR kaszkád szűrő érzékenysége az együtthatókra 16 biten, $\lambda = 0.5$

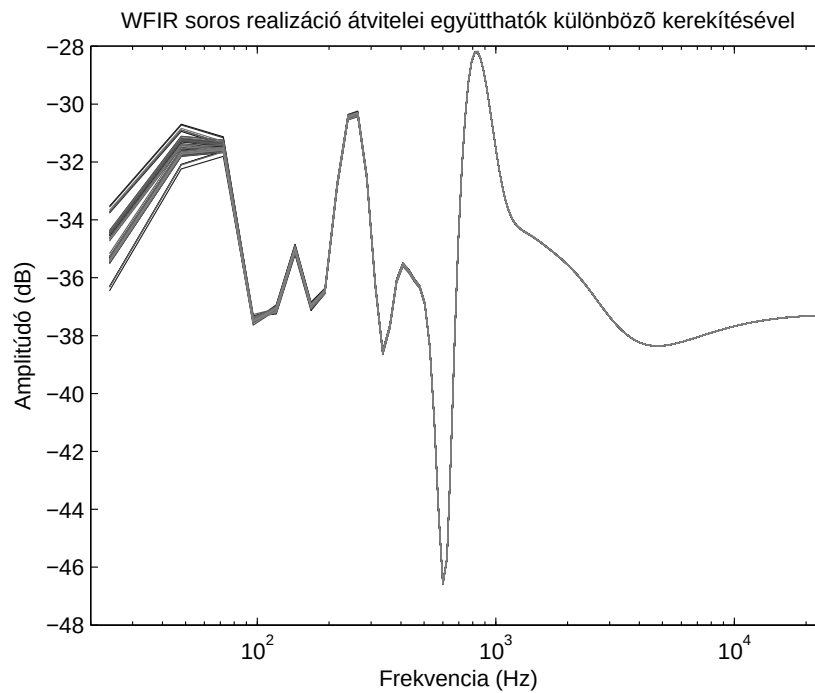
A 3.6. 3.7. és 3.8 ábrákon látható, hogy az együttható-kerekítés hatása nem csak a bitszámtól, hanem λ -tól is függ. Ezért itt is egy hasonló táblázatot fogok adni az eredményekről, mint a kvantálási zaj vizsgálatánál.

WFIR soros realizáció			
Lambda	16 bit	24 bit	32 bit
0,5	×	✓	✓
0,7	×	✓	✓
0,95	×	×	✓

3.3. táblázat. WFIR kaszkád szűrő együttható érzékenységeinek eredményei



3.7. ábra. 20-ad fokú WFIR kaszkád szűrő érzékenysége az együtthatókra 24 biten, $\lambda = 0.5$



3.8. ábra. 20-ad fokú WFIR kaszkád szűrő érzékenysége az együtthatókra 24 biten, $\lambda = 0.95$

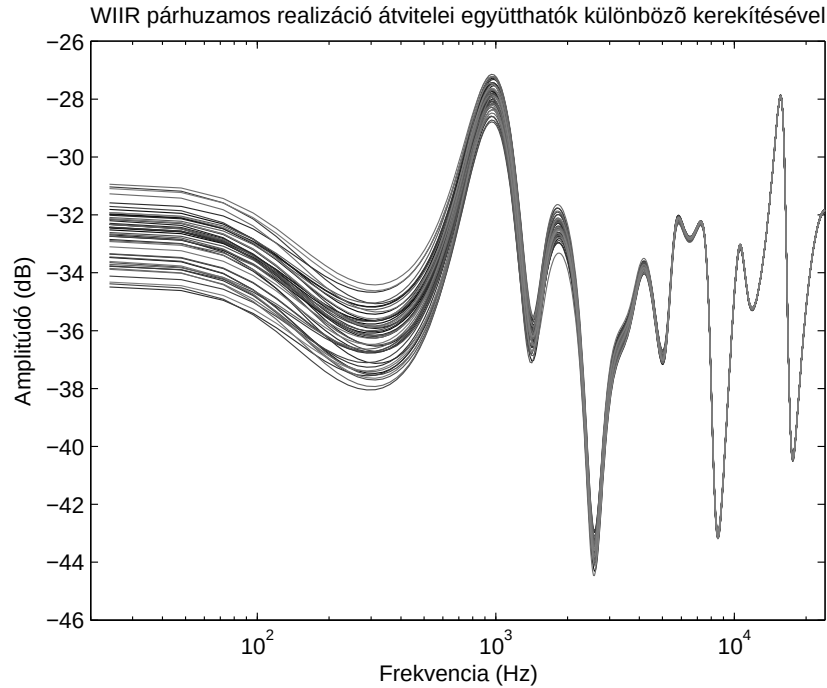
3.2.3. Párhuzamos realizáció

Párhuzamos másodfokú tagokkal a WFIR-t nem, csak a WIIR szűrőket lehet realizálni.

Az eredmények hasonlóak a soros realizációkhoz, ennek oka a másodfokú IIR direkt ta-

WIIR soros realizáció			
Lambda	16 bit	24 bit	32 bit
0,5	×	✓	✓
0,7	×	✓	✓
0,95	×	×	✓

3.4. táblázat. WIIR kaszkád szűrő együttható érzékenységének eredményei



3.9. ábra. 20-ad fokú WIIR párhuzamos szűrő érzékenysége az együtthatókra 16 biten, $\lambda = 0.5$

WIIR párhuzamos realizáció			
Lambda	16 bit	24 bit	32 bit
0,5	×	✓	✓
0,7	×	✓	✓
0,95	×	×	✓

3.5. táblázat. WIIR párhuzamos realizáció együttható érzékenységének eredményei

gokban keresendő. A warpolt szűrők pólusai és zérusai alacsonyabb frekvenciákon helyezkednek el, ráadásul az egységkörhöz közel. A direkt tagoknak pont ebben a tartományban legmagasabb az érzékenységük az együtthatók értékeire.

4. fejezet

Összefoglalás

Láthattuk, hogy egy szűrő kvantálási szempontból jónak minősítéséhez két feltételt kell teljesítenie: kis zaj és kellően pontos átvitel. Az eredményeket táblázatban szemléltetem, ahol zöld cella jelenti azt, hogy mind a kettő feltételnek megfelel a szűrő.

WFIR közvetlen realizáció			
Lambda	16 bit	24 bit	32 bit
0,5			
0,7			
0,95			
WFIR soros realizáció			
0,5			
0,7			
0,95			
WIIR közvetlen realizáció			
Lambda	16 bit	24 bit	32 bit
0,5			
0,7			
0,95			
WIIR soros realizáció			
0,5			
0,7			
0,95			
WIIR párhuzamos realizáció			
0,5			
0,7			
0,95			

4.1. táblázat. Eredmények WFIR és WIIR szűrőnél

Az eredmények alapján elmondható, hogy 16 biten warpolt szűrőt nem érdemes implementálni és WFIR-t nem érdemes sorosítani. A WIIR szűrőket azonban érdemes lehet 32 bites DSP-n soros, esetleg párhuzamos struktúrában megvalósítani, mert számításigény szempontjából ideálisak, kvantálási jelenségeik pedig közel elhanyagolhatóak. 24 bites számábrázolás esetén a közvetlen megvalósítás mindig kielégítő eredményeket ad.

A tárgyalt kvantálási jelenségek a másodfokú direkt IIR megvalósításból adódnak. A tagok implementálására léteznek más struktúrák, például a Kingsbury, valamint a Zölzer struktúra [6]. A későbbiekben ezeket fogom megvizsgálni.

Irodalomjegyzék

- [1] Chalee Asavathiratham. *Digital audio filter design using frequency transformations*. PhD thesis, Massachusetts Institute of Technology, 1996.
- [2] Chalee Asavathiratham, Paul E Beckmann, and Alan V Oppenheim. Frequency warping in the design and implementation of fixed-point audio equalizers. In *Applications of Signal Processing to Audio and Acoustics, 1999 IEEE Workshop on*, pages 55–58. IEEE, 1999.
- [3] Wei Chen. Performance of cascade and parallel iir filters. *Journal of the Audio Engineering Society*, 44(3):148–158, 1996.
- [4] Matti Karjalainen, Aki Harma, and Unto K Laine. Realizable warped iir filters and their properties. In *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, volume 3, pages 2205–2208. IEEE, 1997.
- [5] K Steiglitz. A note on variable recursive digital filters. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 28(1):111–112, 1980.
- [6] Udo Zölzer. *Digital audio signal processing*. John Wiley & Sons, 2008.