

Diplomaterv

Készítette: Tölyhi Tamás

2004.

Nyilatkozat

Alulírott Tölyhi Tamás, a Budapesti Műszaki és Gazdaságtudományi Egyetem hallgatója kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, és a diplomatervben csak a megadott forrásokat használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

.....
Tölyhi Tamás

Tartalomjegyzék

Összefoglalás	7
Summary	9
Előszó	11
1 Bevezetés	13
2 Rendszerterv	15
2.1 A hardverkörnyezet.....	15
2.2 A szoftver.....	18
3 Szűrők tervezése	21
3.1 Előírások a szűrőkkel kapcsolatban.....	22
3.2 Szűrőtípus, approximációválasztás.....	24
3.3 A FIR szűrők jellemzői.....	24
3.3.1 Frekvencia mintavételezés.....	25
3.3.2 Ablakozás.....	26
3.3.3 Optimális tervezés.....	31
3.3.4 Összefoglalás.....	33
3.4 Az IIR szűrők jellemzői.....	33
3.4.1 Szűrőkarakterisztikák közelítésének módszerei.....	34
3.4.2 Összefoglalás.....	40
3.5 A megvalósítási forma kiválasztása.....	40
3.5.1 Kaszkád megvalósítás.....	42
3.5.2 Összefoglalás.....	44
3.5.3 Rezonátoros megvalósítás.....	46
3.5.4 Összefoglalás.....	49
3.6 Következtetés.....	51
4 Megvalósítás	55
4.1 A jelfeldolgozó kártya felépítése.....	55
4.2 Szűrőegyütthetők előállítása.....	57
4.3 A program ismertetése.....	59
4.3.1 A szűrést megvalósító rutin.....	60
4.3.2 Szűrőkarakterisztikák előállítása.....	66
4.3.3 Szűrőegyütthetők elhelyezése, elővétele.....	69
4.4 A kiegészítő áramkör.....	72
5 Mérési eredmények feldolgozása	78
6 Továbbfejlesztési lehetőségek	84
Irodalomjegyzék	86
Függelék	88
A CD melléklet tartalma.....	88
Adatlapok.....	90
További mérési eredmények.....	96
A kiegészítő áramkör kapcsolási rajza.....	99

Összefoglalás

A diplomaterv hangfrekvenciás sávban működő, digitális rezonátoros szűrőkkel megvalósított, programozható szűrő tervezésével foglalkozik. A fejlesztéshez az Analog Devices EZ-KIT Lite Board néven ismert fejlesztőrendszere lett felhasználva, amely az ADSP-2181-es jelfeldolgozó processzorra épül.

A programozható szűrők szerepe a mérés technikában igen fontos. Gyakran szükséges ugyanis a mérendő jelet frekvenciatartományban korlátozni. Ezzel a jelet terhelő nem kívánt zajok mértéke csökkenthető vagy a jel egy bizonyos frekvenciasávba eső része vizsgálható.

A diplomaterv lépésről-lépésre mutatja be az elkészült digitális programozható szűrő tervezésének folyamatát, különös hangsúlyt fektetve a megfelelő szűrőtípus, majd a megvalósítási forma kiválasztására. Emellett tartalmaz több olyan elvárást a hardverrel és szoftverrel szemben, amelyek hasonló áramkörök tervezésénél nyújthatnak segítséget. Ehhez különböző gyártók által forgalmazott programozható szűrők paraméterei lettek összehasonlítva. Ez alapján elkészült egy hardver rendszerterv, amely a tervezés kiindulópontjául szolgált.

Mivel a programozható szűrő egyik fontos paramétere magának a szűrésnek a pontos megvalósítása, ezért a megfelelő szűrőtípus kiválasztásának tárgyalása a diplomaterv lényeges részét képezi. Összehasonlításra kerültek az egyes szűrőtípusok (FIR, IIR) és az egyes közelítési eljárások tulajdonságai, figyelembe véve a programozható szűrő támasztotta követelményeket. Ezek alapján kiválasztásra került egy, a megvalósítás szempontjából optimális approximáció.

A szakdolgozat részletesen tárgyalja két megvalósítási forma, a kaszkád és a rezonátoros megvalósítás tulajdonságait. Ehhez elsősorban a digitális jelfeldolgozó processzorra való alkalmazhatóságot veszi figyelembe. Ahhoz, hogy az eredmények ne csak elméleti síkon, hanem gyakorlati megfigyelésekkel is össze legyenek hasonlítva, elkészültek a megfelelő DSP programok. Pontos mérésekkel alátámasztva bizonyítható volt, hogy a rezonátoros struktúra jóval kedvezőbb megvalósítást eredményez, mint a kaszkád.

Az elkészült DSP program biztosítja a szűrők közel oktávonkénti beállíthatóságát és a szűrőkarakterisztikák közti váltást (alul- felüláteresztő, sávzáró vagy sáváteresztő szűrő). Továbbá egy saját tervezésű, egyszerű kezelőfelülettel való kommunikációt is megvalósít. A kezelőfelület a fejlesztőkártya kiegészítőjeként, a szűrő egyszerűbb programozhatóságát segíti.

Végül a mérési eredmények feldolgozásával a megvalósított programozható szűrő értékelése történik. Ehhez néhány olyan tipikus beállítás lett felhasználva, amely jól tükrözi a szűrő használhatóságát, illetve annak korlátait. Emellett javaslatok találhatók a hibák és hiányosságok kiküszöbölésére, amelyek egy későbbi továbbfejlesztésekhez nyújthatnak segítséget.

Summary

The thesis deals with planning of a programable filter, which is working in the band of sound frequency and which is implemented by digital resonator-based filters. For the development I used the developing system of Analog Devices known as EZ-KIT Lite Board, based on the digital signal processor, ADSP-2181.

Programable filters have a significant role in the measurement technology, because we often need to keep the signal we want to measure within the limits of frequency domain. Therefore we can reduce the undesired noises are troubling the signal or we can examine that part of the sign, which is within this frequency domain.

The thesis demonstrate step by step the planning process of the digital programable filter. It emphasizes the importance of the selection of the suitable filter type and the form of realization. In addition it contents several expectations regarding to the hardware and software, which can be useful at planning of similar electric circuits. For this I compared the parameters of different tunable filters made by different manufacturers. Based on this I created a hardware planning system, which was the starting-point of the planning.

The selection of the suitable type of filter take important part in the dissertation, because one of the most important parameters of the tunable filter is the accurate realization of the filtering. I compared different types of filters (FIR, IIR) and the features of the approaching methods, takeing into consideration the requirements regarding to the tunable filter. Based on this I selected one approximation, which is the optimum in the respect of realization.

The diplom work deals with the features of the two realization methods, cascade and resonator-based in detail. For this it mainly takes into consideration how suitable they are for the digital signal processor. I prepared the suitable DSP programs with which we can compare the results in practical observation as well, not just in theoretical way. Supported by precise measurements it was demonstrable that the resonator-based structure has a most favourable realization than the cascade one.

The ready DSP program assures the adjustability of filters nearly by octaves and the change among filter characteristics (Low-Pass, High-Pass, Band-Pass or Band-Reject). Furthermore it accomplishes an own planning communication with a simple operating-board. The operating-board, as a supplementary unit of the developing-card, helps to program the filter in a more simple way.

Finally I assessed the created programable filter by using the results of measurements. For this I used some typical settings, which perfectly reflect the usability and the bands of the filter. In addition I put some suggestions how to eliminate the errors and inadequacies, which can be useful in the further developments afterwards.

Előszó

A digitális jelfeldolgozás matematikai hátterének megteremtése után még évtizedeket kellett várni, amíg megjelent egy olyan hardver architektúra, amely már rendelkezett a szükséges erőforrásokkal és ezáltal lehetővé tette a valós idejű jelfeldolgozást. A nagy számítási igény és sebesség miatt ezek a digitális jelfeldolgozó processzorok (DSP) egy speciális architektúrával rendelkeznek, amelyek a jelfeldolgozásnál használt alapműveleteket (szorzás, összegzés és együtthatók mozgatása) párhuzamosan valósítják meg.

A jelfeldolgozó processzorok fejlődése mára már elérte azt a szintet, hogy segítségükkel komoly és összetett számításokat igénylő feladatok valósíthatók meg. Ezért felhasználásuk egyre szélesebb körben terjedt.

A diplomaterv célja egy DSP környezetre épülő, a mérés technikában gyakran használt programozható digitális szűrő fejlesztésének ismertetése. A jelfeldolgozó processzorok fejlődése ugyanis már lehetővé teszi, hogy az alacsonyabb frekvenciasávban (néhány 100 kHz) az analóg szűrőket egy sokkal megbízhatóbb és pontosabb digitális szűrővel válthassuk ki.

A mai mérés technikában a szűrők alkalmazásának és használhatóságának ténye vitathatatlan. Rengeteg olyan terület van, ahol ezek használata nélkülözhetetlen. Ilyenek például az átvitel technikai mérések, a mérendő jelek járulékos zajának csökkentése vagy például az orvos biológiai mérések, és még számos olyan terület, ahol a programozható szűrőket előnyösen lehet alkalmazni.

Azzal hogy egy szűrő programozható, és nem egy előre meghatározott beállítással működik, a felhasználhatósági köre jóval szélesebbé válik, többféle mérési célra lesz alkalmazható. Tehát amellet, hogy egy ilyen szűrő átvitele állítható, rugalmasan illeszkedik a különféle mérőrendszerekhez.

A szakdolgozat célja tehát egy olyan programozható szűrő tervezése, amely a hangfrekvenciás tartományt átfogva, valósítja meg a programozhatóságot és lehetővé teszi a különböző mérőrendszerekhez való illesztést.

Mivel sok gyártó foglalkozik ilyen szűrők forgalmazásával, először el kellett végezni egy piackutatást, hogy a hasonló célra tervezett műszerek milyen specifikációval rendelkeznek. A cél itt nem egy nagy pontosságú, minden igényt kielégítő szűrő elkészítése, hanem egy általánosan használható, olcsón megvalósítható, egyszerűbb kivitelű eszköz tervezése volt. A nagy pontosságú műszerek ugyanis meglehetősen drágák. Azonban az ezeknél alkalmazott funkciók és beállítási lehetőségek jó kiindulást jelentettek a tervezés során.

Azzal, hogy a hangfrekvenciás sáv lett megcélözva a tervezésénél, lehetőség nyílt a jelfeldolgozó processzorral történő megvalósításra. Ebben a frekvenciasávban már számos gyártó által forgalmazott processzorral meg lehet valósítani a programozható szűrőt. Ki kellett tehát választani egy megfelelő jelfeldolgozó processzort, amellyel a szűrővel szemben támasztott követelmények még teljesíthetők.

Ezután következett az irodalomkutatás, melynek a célja, hogy a különböző approximációkat majd megvalósítási formákat összehasonlítva ki lehessen választani a megvalósítás szempontjából legkedvezőbb megoldást. A szűrőtervezésnek ez egy meglehetősen fontos része. Ugyanis a kiválasztott struktúra és approximáció határozza meg a megvalósítás korlátait.

Ezt követően lehetett elkezdni a tényleges fejlesztést. Ennek eredménye egy jelfeldolgozó processzorra íródott program, amely a kitűzött specifikációnak megfelelően valósítja meg a programozható szűrő működtetését.

Végül a szűrők bemérése és a mérési eredmények kiértékelése következett, majd a szakdolgozat végén az elkészült szűrő hibáit figyelembe véve és a tervezés során felhasznált tapasztalatok alapján a későbbi továbbfejlesztést elősegítő javaslatok találhatóak.

1 Bevezetés

A feladat programozható szűrő tervezése. Az, hogy egy szűrő programozható (hangolható) azt jelenti, hogy nem kell az áramkör belsejébe nyúlni és ott bármiféle fizikai változtatást végezni, hogy a szűrő paramétereit módosítani tudjuk. A programozhatóság tehát azt jelenti, hogy például külső kezelőfelületen történő beállítások segítségével a szűrőt átállítjuk, átprogramozzuk, paramétereit megváltoztatjuk. Ilyen paraméterek a törésponti frekvencia vagy a szűrőtípus kiválasztása.

A programozható szűrő tehát egy olyan eszköz, amely könnyen átkonfigurálható, ezáltal nyújt rugalmas felhasználást.

A mérnöki gyakorlatban nagyon sokszor előforduló feladat a különböző, főleg elektromos jelek paramétereinek meghatározása. Ehhez a jelet illeszteni kell egy adott mérőrendszerhez. Ilyenkor egy lényeges elvárható követelmény, hogy a mérendő jelet a lehető legkevesebb járulékos hibával terheljük. A mérendő jelek azonban sokszor olyan zajos környezetből származnak, hogy a valódi jel leválasztása a jelforrásból, csak valamilyen szelektív mérési eljárással lehetséges. Ilyen áramkörök például a szűrők is. Azzal ugyanis, hogy a mért jelet a frekvenciatartományban korlátozzuk, nagyon sok egyéb forrásból adódó zaj hatása kiküszöbölhető.

A mérési célokra kifejlesztett szűrőáramkörök feladata, hogy minél egyszerűbben, minél kevesebb járulékos hibát okozva csatlakoztatni tudjuk egy adott mérőrendszerhez. Ehhez azonban meg kell felelnie néhány követelménynek. Például a hasznos, azaz a mérendő jelet a lehető legkevésbé torzítsa, a nemkívánatos zajokat viszont a lehető legnagyobb mértékben nyomja el, vagyis azoknak a mérést befolyásoló hatását csökkentse minimálisra.

Egy ilyen szűrőáramkörrel kapcsolatban alapvető követelmény, hogy lehetőleg minél szélesebb körben alkalmazható legyen, azaz minél több beállítási lehetőséggel rendelkezzen. Bár a konkrét feladatra kifejlesztett áramkörök jóval kedvezőbb tulajdonságúak, de a gyakorlatban mégis egyre jobban elterjedtek a programozható (átkapcsolható) szűrők. Ezek előnye, hogy a használható frekvenciatartományban a kezelőszervek segítségével az áteresztősávi törésponti frekvenciák határai bizonyos tartományok között, valamilyen felbontással állíthatók. További beállítási lehetőség lehet még a szűrő típusának és fokszámának kezelőszervek segítségével történő kiválasztása is.

A műszerpiacon az ilyen átkapcsolható szűrők már régóta jelen vannak. Több gyártó is forgalmaz, különböző sávszélességben és felbontással dolgozó szűrőket. A függelékben példaként megtalálható két gyártó programozható szűrőjének az adatlapja. Mint az adatlapok alapján is látszik, vannak egyszerűbb kivitelű és ezáltal olcsóbb változatok, ahol a beállítható törésponti frekvenciák viszonylag ritkán követik egymást és megjelentek már olyan nagy pontosságú széles frekvenciasávban működő műszerek is, amelyek már majdnem folyamatos törésponti frekvencia beállítást tesznek lehetővé a teljes sávban, többféle kiválasztható szűrő karakterisztika mellett. Emellett általában még lehetséges a bemeneti fokozat erősítésének állítása, kiválasztható, hogy a jel AC vagy DC csatolt legyen, valamint a kimenő jel amplitúdójának nagysága is legtöbbször külön beállítható.

Ezek után felmerülhet a kérdés, hogy miért is van szükség programozható szűrő tervezésére, hiszen ezek már évek óta megvásárolhatók a piacon. Itt az elsődleges szempont az ár, hiszen a nagypontosságú szűrőáramkörök ma még meglehetősen drágák, és nem biztos, hogy az adott felhasználói környezetben az igényeket, esetleg némi kompromisszum árán nem elégíti-e ki egy-egyszerűbb kivitelű, kevesebb beállítási lehetőséget tartalmazó szűrőáramkör. Sajnos azonban még ezek is meglehetősen drágák. A szerényebb kivitelhez itt jóval szerényebb beállítási lehetőségek is tartoznak. Sokszor csak egyetlen szűrő karakterisztika (általában alul vagy felüláteresztő karakterisztika) használata lehetséges és nem támogatják külön a sávszűrés lehetőségét. Ehhez két ilyen szűrőt kellene párhuzamosan, illetve sorba kötni, ami ismét drágítja vagy bonyolultabbá teszi a felhasználást.

A digitális jelfeldolgozó processzorok elterjedésével lehetőség nyílt arra, hogy alacsonyabb frekvenciatartományokban ki lehessen váltani a drága nagypontosságú alkatrészekből felépülő szűrőket digitálisan megvalósított szűrőkkel. Ezek előnye, hogy a szűréshez nem igényelnek precíz alkatrészeket, ezáltal az alkatrészek öregedése nem vezethet elhangolódásához.

A diplomaterv célja tehát egy olyan digitális jelfeldolgozó processzorra épülő programozható szűrő tervezése, amely a hangfrekvenciás tartományt fedi le. A frekvenciasáv azért lett erre a tartományra korlátozva, mert a ma már könnyen beszerezhető, kisebb számítási sebességű jelfeldolgozó processzorokkal ez olcsón megvalósítható. Emellett ez a frekvenciasáv már meglehetősen tágas alkalmazási területet biztosít: például beszéd és hangátviteli mérések, adatátviteli mérések, orvosi alkalmazások, zavarsszűrés.

Összefoglalva tehát, a cél egy olyan programozható szűrőáramkör fejlesztése, amely a hangfrekvenciás tartományban használható, viszonylag sűrűn felosztva ezt a tartományt a törésponti frekvenciák beállíthatósága szempontjából. Továbbá, mint önálló hordozható műszer könnyen illeszthető legyen mérőrendszerekhez. Tehát viszonylag kevés járulékos hibát okozzon. Emellett lehetőleg olcsó, könnyen beszerezhető alkatrészekből épüljön fel az áramkör.

2 Rendszerterv

Ebben a fejezetben a digitális szűrő egy lehetséges megvalósításának hardver és szoftver lehetőségei kerülnek bemutatásra. A tervezés során ezek az elvárások és szempontok jelentették a kiindulási alapot. A diplomaterv keretébe az idő rövidege miatt nem fért bele egy kész hardver eszköz megvalósítása, ezért itt csak egy rendszerszintű felépítés lesz bemutatva, és néhány olyan fontos követelmény, amelyek egy későbbi megépítés során segítséget jelenthetnek.

A tervezés folyamata három részre osztható. Az első részben ki kell választani egy olyan megfelelő hardverkörnyezetet, amelynek segítségével a programozható szűrő, mint önálló eszköz megvalósítható. Ezután meg kell vizsgálni több szűrési eljárást, el kell dönteni, hogy milyen struktúrával legyenek a szűrők megvalósítva, melyik a legmegfelelőbb approximáció és melyik megvalósítási forma biztosítja a lehető lekedvezőbb működést az adott frekvenciasávban. Végül fel kell építeni egy olyan programstruktúrát, amely illeszkedik az elviekben megtervezett hardver architektúrához.

A tervezés elején, első lépésként el kellett dönteni, milyen módon legyen a szűrés megvalósítva. Ennek eredményeként született meg a digitális szűrők alkalmazása. A digitális szűrők nagy előnye, hogy az átviteli karakterisztika mindig állandó. Nincsenek az alkatrészek öregedéséből adódó elhangolódási problémák, mivel a rezonanciafrekvenciákat nem a diszkrét alkatrészek és azok viszonylagos pontossága határozza meg. Továbbá a jelfeldolgozó processzorok az utóbbi években rohamos fejlődésnek indultak. Az architektúrák egyre összetettebbek egyre gyorsabb működést biztosítanak, és emellett az áruk is jelentősen csökkent. Ami szintén a digitális szűrők megvalósítása mellett szólt. Mivel a digitális szűrők együtthatói memóriából kerülnek kiolvasásra, anélkül, hogy további alkatrészeket építenénk az áramkörbe, többféle szűrő karakterisztika is megvalósítható. Ezek között egyszerűen megoldható az átváltás, ami lényegesen kezelhetőbbé és rugalmasabbá teheti a rendszert. Végül, a digitális szűrőkkel sokkal nagyobb levágási meredekség és szelektivitás oldható meg, mint analóg szűrők segítségével, ezért ezek e tekintetben is előnyösebbek.

2.1 A hardverkörnyezet

Mivel az előnyös tulajdonságai miatt a szűrés digitálisan lesz megvalósítva, ki kell választani egy megfelelő processzorcsaládot, amellyel a feladat megoldható. Érdemes a nagyobb gyártók választékát figyelni (Analog Devices, Texas Instruments, Motorola), mivel ezek termékei jóval könnyebben beszerezhetők. A kedvezőbb ár miatt itt elsősorban a 16 bites processzorok kerültek előtérbe. A Texas TMS320C2x, -C5x processzorai, bár kategóriájukban általában erősebbek, programozásuk mégis nehézkes. Az egyes generációk tagjai nem minden esetben kód kompatibilisek egymással. Ez gyakran hátrányt jelent, mivel előfordulhat, hogy a fejlesztés során át kell térni egy másik processzorcsaládra. Az Analog Devices már jóval felhasználóbarátabb processzorokkal jelent meg a piacon. Ezek belső felépítése már kiforrottabb, ezen kívül a család tagjai egymással kód kompatibilisek. A másik előnyük, hogy támogatják az algebrai assemblert. Ez a program fejlesztésénél nagyon előnyös, hiszen a kódot sokkal

áttekinthetőbbé teszi. A beszerezhetőség és programozhatóság mellett a döntést a rendelkezésre álló fejlesztőrendszerek választéka is befolyásolta. Ezért a választás végül az Analog Devices ADSP-21xx családján belül a 2181-es processzortípusra esett. Ez egy 16 bites processzor jól felépített belső architektúrával és perifériatámogatással. Emellett viszonylag nagy (16K) belső program és adatmemóriával rendelkezik

A választás azért esett 16 bites processzorra, mivel ezek jelenleg jóval olcsóbbak, mint a 32 bites lebegőpontos utódaik és számítási sebességük is megfelelőnek látszott a hangfrekvenciás sávban működő szűrők megvalósításához. Bár felmerült a kérdés, hogy a 16 bites számítási pontosság esetleg nem lesz elegendő ebben, a viszonylag széles frekvenciasávban állítható szűrők megvalósítására, hiszen az alacsony frekvenciákon már viszonylag nagy szelektivitású szűrőket kell alkalmazni a mintavételi frekvenciához képest. Ehhez pedig az együtthatók tárolását és a számításokat is viszonylag pontosan kell végrehajtani. Ezért a kritikus helyeken előfordulhat, hogy a számításokat duplapontosan, 32 bitesen kell elvégezni. Viszont a közelítő számítások azt igazolták, hogy a szűrőrutinok még duplapontos számítás esetén sem haladják meg a processzor szabta időkorlátokat.

A jelfeldolgozó processzor tehát adott, már csak meg kell teremteni a szükséges jelfeldolgozó környezetet, amelynek segítségével a programozható szűrő fejlesztése elkezdhető. Ebben nagy segítséget jelentett, hogy a tanszék rendelkezik néhány, ezzel a processzorral szerelt, egyszerűbb felépítésű fejlesztőkártyával és az ehhez szükséges szoftverekkel. Ezt az Analog Devices ADSP-2181 EZ-KIT Lite Board néven forgalmazza, kifejezetten fejlesztési célokra. A kártya felépítése olyan, hogy illeszkedjen az általános követelményekhez, azaz tartalmaz néhány olyan kiegészítő áramkört, amelyek alapvetően szükségesek egy jelfeldolgozó környezetben. Mivel a digitális szűrés nem támaszt komoly hardverigényeket, eltekintve a szükséges kezelőszervektől és esetleges kijelzőktől, a kártya teljes mértékben alkalmas a szűrés megvalósító és az ezt kezelő program fejlesztéséhez és teszteléséhez.

A fejlesztőkártyán az analóg interfészt egy AD1847-es típusú sztereó delta-sigma Codec képviseli. Ezzel a jelfeldolgozó kártya alkalmassá válik arra, hogy a tervezett hangfrekvenciás tartományban megvalósíthassuk a programozható szűrőt. Az AD1847 legmagasabb mintavételi frekvenciája 48 kHz lehet, alacsony frekvenciás tartományban pedig egy RC-tag gondoskodik a 20 Hz-nél alacsonyabb frekvenciájú komponensek levágásáról.

A fejlesztőkártya egyik hátránya, hogy csak egyetlen nyomógomb használható fel a program futásának módosítása céljára. Ezért el kellett készíteni egy kiegészítő áramkört (lásd később a 4.4 fejezetben), amely a felhasználó felé egy egyszerű kezelőfelületet nyújt a szűrők beállításához.

A fejlesztőkártya tehát nem nyújt egy ideális hardver környezetet a programozható szűrő megvalósításához. Ez csak fejlesztési, tesztelési és bemérési célokra használható. Ahhoz, hogy egy ilyen áramkört ténylegesen meg lehessen építeni, meg kell határozni néhány fontos követelményt, amelyet a hardvertervezésnél figyelembe kell venni. Ilyen szempontok a kis méret, hordozhatóság és az egyszerű, jól áttekinthető kezelőfelület, amelyek felhasználó oldalról nézve fontosak. Gyakran szükség van ugyanis a nagypontosságú laborműszerek mellett olcsóbb, hordozható kézi műszerekre.

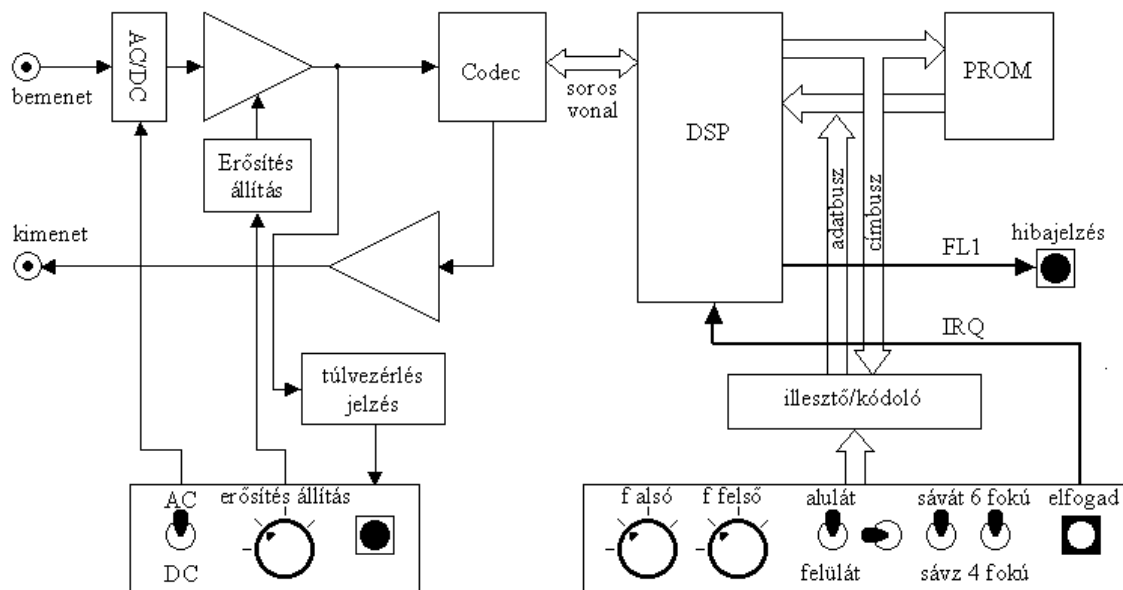
A hordozhatóság feltétele, hogy viszonylag hosszú ideig biztosítani lehessen a szűrő működéséhez szükséges energiát. Ehhez azonban kis fogyasztású alkatrészek használata szükséges. A szűrő bekapcsolt állapotban folyamatos működésű kell legyen, vagyis működés közben nem lehet az alkatrészeket alacsony fogyasztású üzemmódba kapcsolni. Fogyasztás szempontjából a DSP és a Codec a kritikus alkatrészek. Ezek együttes áramfelvétele elérheti a 200 mA-t 5 V-os tápfeszültség mellett. Tehát az akkumulátoros működés mellett mindenképp biztosítani kell az adapteres töltés és tápellátás lehetőségét. A kezelőfelületnek is olyan kialakításúnak kell lenni, hogy a beállítások egyértelműen leolvashatók legyenek a kezelőszervek állásából. Vagyis csak a legszükségesebb kijelzők kerülhetnek az előlapra. Ezt szintén a fogyasztás alacsony szinten tartása indokolja. Érdemes a tápellátást úgy kialakítani, hogy az analóg és digitális oldal tápfeszültsége egységes, jelen esetben 5 V legyen. Az analóg tápfeszültségből regulátor segítségével előállított 5 V mindenképp jelentős energia-vesztéséget eredményez.

Az analóg áramköröknél törekedni kell arra, hogy minél kisebb járulékos zaj adódjon a mérendő jelhez, tehát kiszajú bemeneti erősítőt kell választani. További követelmény, hogy az erősítő egy tápfeszültségről is működtethető és ebben a tartományban teljesen kivezérelhető legyen. Így a bemenetre maximálisan 5 V-os jel kapcsolható. Ezt a jelet illeszteni kell az ADC-hez. Az AD1847 típusjelű 16 bites soros Codec erre a célra tökéletesen megfelel. Használatával biztosítható a programozható szűrő hangfrekvenciás sávban való működtetése. A 16 bites Codec-el az elméletileg elérhető jel/zaj viszony 96 dB lehet. Ez az érték jelentősen romlik a bemeneti jel csökkenése esetén. Ezért a bemeneti fokozatnak mindenképpen állítható erősítésűnek kell lenni. Érdemesebb nagy pontosságú ellenállásokkal diszkrét értékeket beállítani. Az átkapcsoláshoz pedig forgókapcsolót alkalmazni. Ahhoz, hogy a jel/zaj viszony értéke túlságosan ne romoljon el, az erősítés beállítás legalább 10 dB-es ugrásokkal történjen. Így az elméletileg elérhető jel/zaj viszony nem csökken 80 dB alá. A bemeneten az állítható erősítés miatt szükség van egy, az ADC túlvezérlését jelző LED-re. A kimeneten nem szükséges az erősítés állítás, hiszen a DAC kimenetén teljes kivezérlés esetén 2 V csúcsertékű jel jelenik meg. A kimeneti erősítővel szembeni elvárás, hogy legalább egy fejhallgató meghajtására képes legyen. Érdemes lehet még a hangfrekvenciás sávot kitolni egészen a DC tartományig, azonban mindenképp szükséges a DC leválasztás lehetőségét megtartani. Ez egy külön kapcsolóval megoldható.

A digitális oldalon meg kell oldani a DSP bekapcsolás utáni automatikus felprogramozását. Tehát PROM-ot kell illeszteni a jelfeldolgozó processzorhoz. Erre a célra megfelel a 27C128 vagy a 27C256 típusú párhuzamos 8 bites EPROM. Ezek mérete 16 illetve 32 kilószó, amely bőven elegendő az elkészült program beégetéséhez. Meg kell oldani még a Codec illesztését is a processzorhoz, valamint az analóg interfészhez. Az AD1847 típusú Codec előnye, hogy eleve sztereo felépítésű, tehát egyetlen IC segítségével megvalósítható a kétcsatornás működés. A Codec soros vonalon keresztül kapcsolható a DSP-hez. Ezt a lehetőséget a DSP is támogatja, mivel rendelkezik két soros interfésszel. A kezelőfelület egy része is a digitális oldalhoz kapcsolódik. Ezek a szűrők paramétereit beállító kapcsolók. A törésponti frekvenciaváltást érdemes forgókapcsolóval megoldani. Erről egyértelműen leolvasható a beállított frekvencia érték. A szűrőtípusok közti átkapcsolásra használható billenő kapcsolók. Ezeknél a kapcsoló állapota (0 vagy 1) könnyen kiolvashatók. A forgókapcsolónál azonban meg kell oldani az egyes beállításokhoz tartozó

kapcsolóállások dekódolását. Ezen kívül meg kell oldani az adatbuszhoz történő illesztést és a kapcsolók, mint periféria címzését és olvasását. Az olvasás engedélyezésére, vagyis a beállítások elfogadására itt is célszerű nyomógombot használni, és hasonlóan a fejlesztőkártyán lévő megoldáshoz, a processzor IRQ lábára kötni. A hibás szűrőbeállítás jelzésére pedig egy LED használható.

A kezelőfelületet tehát két részre osztható. A kapcsolók egyik része az analóg oldal működését befolyásolja (erősítés állítás, AC/DC csatlós kiválasztása), másik része viszont a digitális oldal, vagyis a szűrők beállítását végzi (törésponti frekvencia váltás, szűrőtípus, karakterisztika beállítás). A 2.1 ábra blokkvázlat szinten mutatja be a programozható szűrő egy lehetséges megvalósítását.



2.1. ábra A programozható szűrő blokkvázlata

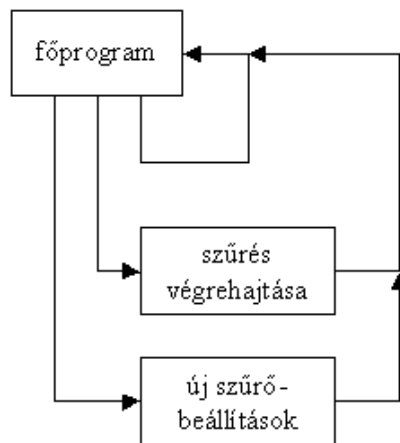
2.2 A szoftver

A szoftver tervezésénél, mivel ez nem az elkészült célhardverrel lesz fejlesztve, figyelembe kellett venni azokat a szempontokat, amelyeket egy programozható szűrő hardverkörnyezete támaszt a szoftverrel szemben. Ideális esetben a hardver és szoftver fejlesztése együtt történik. Itt a hardverkörnyezetet most a fejlesztőkártya biztosítja, tehát a szoftver ennek áramköri megoldásához kell, hogy igazodjon. Emellett, mivel elkészült egy rendszerterv, és ott bizonyos fontosabb elvárások definiálva lettek, ezeket is figyelembe kell venni a fejlesztésnél. Itt elsősorban a kezelőfelület lekérdezése és annak megfelelő szűrőtípusok beállítása az, amelyet az eredeti hardverkörnyezet nem támogat. Ez külön kiegészítő hardverként kapcsolódik a fejlesztőkártyához.

A szoftvernek így két feladatot kell ellátnia, ezek közül az egyik és a futásidő nagyobb részét kitevő programrutin a szűrés végrehajtása. Itt meg kell valósítania az adatok beolvasását és a szűrés eredményének kiírását a Codec-re minden mintavételi időpillanatban. Két mintavételi idő között pedig a kiválasztott struktúrával meg kell valósítani a tényleges szűrést.

Másik feladat a kezelőfelületen található kapcsolók állapotának beolvasása és ennek megfelelően a szükséges szűrőegység elővétele a programmemóriából. Mint ahogy az a 2.1 ábrán is látható külön forgókapcsoló van az alsó és külön a felső törésponti frekvencia beállítására. Természetesen ez sávszűrő beállítása esetén érvényes. Alul vagy felüláteresztő szűrők esetén csak az egyik forgókapcsoló állítható. Az, hogy alul vagy felüláteresztő illetve ezek párhuzamos vagy sorba kapcsolásából előállított sávzáró vagy sáváteresztő szűrő működjön, egy külön választókapcsolóval állítható be. Végül van még egy kapcsoló, amellyel a szűrő karakterisztika váltása lehetséges. Az elkészült programban itt negyedfokú vagy hatodfokú inverz Csebisev karakterisztika között lehet váltani. A beállított paraméterek egy nyomógomb segítségével lesznek elfogadva. A nyomógomb lenyomásának hatására a program végrehajtja a szűrőegység cseréjét.

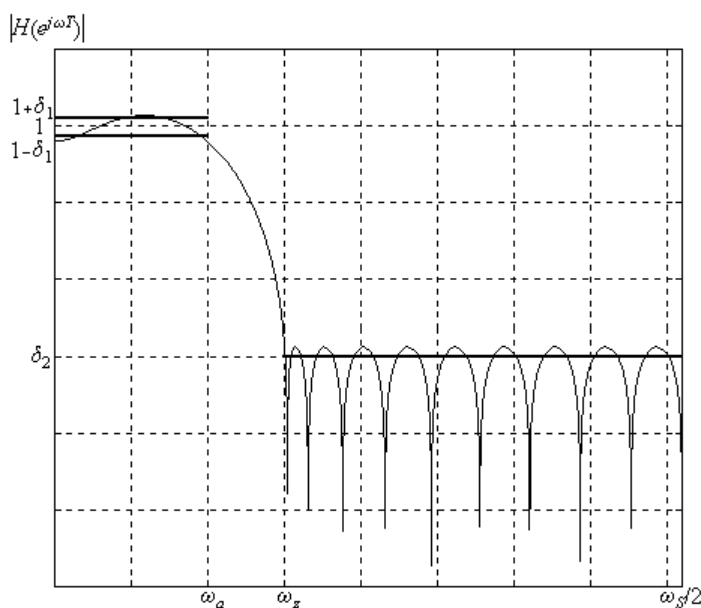
A programnak tehát ezt a két feladatot kell ellátnia a megfelelő megszakítás kérés hatására. Tehát ezeket a programrutinokat célszerű a főprogramtól különválasztani. A főprogram feladata a megszakítások figyelése, és ennek megfelelően a két programrutin egyikének meghívása. Ezt a struktúrát szemlélteti a 2.2 ábra.



2.1. ábra A program felépítése

3 Szűrők tervezése

Digitális szűrők tervezése a mai mérnöki gyakorlatban még mindig komoly odafigyelést és összetett számításokat igénylő feladat. Különböző eljárások léteznek, amelyek segítségével a kívánt karakterisztikához tartozó szűrőegyütthatók meghatározhatók. Végtelen impulzusválaszú, azaz IIR szűrőknél ez leggyakrabban analóg szűrőkből kiindulva, frekvenciatranszformálással történik. Ez meglehetősen hosszadalmas számításokat igényelne. Azonban ma már a tervezés folyamatát rengeteg segédprogram és függvény segíti és teszi kényelmesebbé. A legtöbb matematikai program támogatja a digitális szűrők tervezését. Ilyen például a MATLAB is, amely szintén rendelkezik a digitális szűrők tervezését támogató csomagokkal. Ezek a MATLAB Signal Processing Toolbox-ában található meg. A beépített függvények segítségével nemcsak a különböző approximációkhoz tartozó együtthatók generálása, hanem a megtervezett szűrő szimulációja is leegyszerűsödött. Így a tervezés folyamán a szűrőegyütthatók meghatározása egy kiválasztott approximációhoz már nem okoz nagy problémát. Ahhoz, hogy az együtthatókat megkapjuk, elegendő megadni a szűrő fontosabb paramétereit, mint az áteresztő sáv határa, megengedett ingadozás, fokszám, elnyomás mértéke (lásd 3.1 ábra).



3.1. ábra Szűrőspecifikációk meghatározása

Mivel az együtthatók generálása viszonylag egyszerűen megvalósítható, a tervezés nagyobb része inkább abból állt, hogy meg kellett határozni a szűrővel szemben támasztott követelményeket, és ehhez kellett egy optimális szűrőapproximációt, majd egy megvalósítási formát találni. Itt olyan követelmények merülnek fel, mint például az átviteli karakterisztika azonossága minden szűrőtípusnál bármely beállított törésponti frekvencia mellett. Vagyis a levágási meredekség értéke (dB/oktáv) a teljes frekvenciasávban azonos legyen. Az elnyomás mértéke is minden szűrőnél a specifikációban meghatározott szint fölött kell, hogy legyen. Ez elsősorban mérés technikai szempontból fontos. A nehézséget az okozta, hogy a szűrőknek viszonylag széles frekvenciasávban kell dolgozniuk. Ugyanolyan levágási meredekséget biztosítása a mintavételi frekvenciától távolabb eső szűrőknél jóval pontosabb együttható ábrázolást igényel,

mint a mintavételi frekvenciához közelebb eső szűrőknél. A mintavételi frekvenciától egyre távolabb kerülve a számított pólusok egyre közelebb kerülnek a komplex frekvenciatartomány egységkörének határához, tehát a megvalósított szűrők a stabilitás határán működnek. Ezért lehetőségekhez képest olyan megvalósítási formát kell keresni, amelynek kedvezőek az érzékenységi tulajdonságai még egy ilyen szélesebb frekvenciatartományban is, így elkerülhető a sebességkonverziós jelfeldolgozás (mintaritkítás, szűrés, interpoláció), ami a programot jóval bonyolultabbá teszi. Másik szempont, hogy a specifikáció megvalósításához szükséges fokszám minél alacsonyabb legyen. Ennek elsősorban programozás szempontjából van jelentősége, hiszen a két mintavétel közötti idő meglehetősen rövid egy túlságosan sok együtthatót igénylő szűrőrutin kiszámításához.

A feladat azért volt meglehetősen összetett, mert bár rengeteg segédprogram létezik, azonban ezek nem kifejezetten támogatják a jelfeldolgozó processzoros környezetet. Így egy megtervezett szűrő MATLAB függvényvel kirajzoltatott átviteli karakterisztikája nem biztos, hogy meg fog egyezni ugyanezen szűrő jelfeldolgozó processzoron történő megvalósításakor kapott átviteli karakterisztikával. Ennek oka a számítási pontosságból adódó különbség. Bár az együtthatókat megadhatjuk a MATLAB-nak csonkolt formában, ahogy azok a DSP adatmemóriájában is szerepelni fognak, de ezzel is csak közelítjük a DSP által számított eredményt, ugyanis a MATLAB számítási pontossága jóval nagyobb, mint amire egy 16 bites jelfeldolgozó processzor, akár dupla pontosan is, képes. Ezért a DSP által termelt aritmetikai zajt itt nem tudjuk figyelembe venni. Mégis az így elvégzett számítások és szimuláció már jó közelítést szolgáltat, hogy ellenőrizhessük, a megtervezett szűrőnk teljesíti-e a megkívánt előírásokat. Mivel a mintavételi frekvenciától távolabb, az alacsonyabb frekvencián működő szűrők megvalósítása jelenti a nagyobb gondot az együtthatók érzékenysége miatt, azért a megvalósítás szempontjából az itt kapott eredményeket a megvalósíthatóság korlátjának kell tekinteni. Ezek lehetnek a levágási meredekségből adódó korlátok vagy az elérhető elnyomás mértékének a felső határát jelentő értékek.

3.1 Előírások a szűrőkkel kapcsolatban

A szűrő tervezésénél először meg kellett határozni, hogy az adott frekvenciatartomány milyen mértékben legyen lefedve, vagyis a beállítható törésponti frekvenciák milyen sűrűn kövessék egymást. Nagypontosságú műszereknél, mint ahogy ez a bevezetőben is olvasható, ez a beállítás akár Herz pontossággal is történhet, míg egyszerűbb eszközöknél a beállítható törésponti frekvenciák ritkábban, például oktávonként követik egymást. Ebben az alkalmazásban azonban nem a törésponti frekvenciák minél pontosabb beállítása, hanem a különböző beállításokhoz tartozó szűrőkarakterisztikák azonos menetének biztosítása volt a cél. Approximációtól és fokszámtól függően ugyanis egy szűrő átviteli karakterisztikája a beállított törésponti frekvenciától néhány százalékkal eltérve még nem okoz az amplitúdóban jelentős eltérést. Így nem látszott célszerűnek a frekvenciasáv túlságosan sűrű felosztása. Ezért, mivel ez más programozható szűrőknél is gyakran alkalmazott megoldás, az oktávonkénti felosztás mellett döntöttem, mivel itt még elég sűrűn követik egymást a beállítható törésponti frekvenciák.

A megcélzott hangfrekvenciás tartományban (20 Hz és 20 kHz közötti tartomány) a következő törésponti frekvencia beállítások tűntek célszerűnek: 50 Hz, 100 Hz, 200 Hz,

500 Hz, 1 kHz, 2 kHz, 5 kHz, 10 kHz. Ezek a beállítások az áteresztő sáv határát adják meg, ahol az amplitúdó hiba még nem haladja meg a specifikációban megadott értéket. A 20 Hz-es szűrő megvalósítása azért nem látszott célszerűnek, mivel az analóg bemeneten egy RC-tag gondoskodik a DC jelek leválasztásáról, ezért a 20 Hz alatti komponensek eleve kiszűrésre kerülnek. A 20 kHz-es szűrőnél pedig már a mintavételi frekvencia fele korlátozza az átvitelt.

A tervezés következő lépéseként meg kellett határozni egy olyan specifikációt a szűrőkkel szemben, amely elég szigorú ahhoz, hogy nagy pontosságú mérésekhez illeszkedhessen, de nem túlságosan szigorú ahhoz, hogy mindezt a szűrők a teljes frekvenciatartományban teljesíteni tudják.

Egy ilyen előírás az elnyomási tartományra vonatkozólag, hogy az áteresztő tartományhoz képest a maradék jel legalább 60 dB-lel alacsonyabb szinten legyen. Ennél nagyobb elnyomású szűrő megvalósítása nem csak az együttható érzékenységek szempontjából jelent problémát, hanem azért is, mert az alkalmazott áramköri elemek, a bemeneti erősítő és a kvantálás okozta hiba is járulékos zajjal terheli a jelet.

Elméletileg a 16 bites AD átalakítóval elérhető jel/zaj viszony 96 dB lehet. A bejövő jelet azonban különböző okok miatt, például skálázás, le kell osztani. Továbbá az AD és DA átalakítónál is jelszint illesztést kell végezni. A szűrendő jel sem mindig vezérli ki teljesen az AD átalakítót, és valamekkora amplitúdó tartalékot is kell hagyni a túlvezérlés elkerülése végett. Ezek természetesen rontják az elméletileg elérhető jel/zaj viszonyt. Emellett a számítások következtében is adódik valamekkora, a kerekítésekből származó aritmetikai zaj a jelhez. Ezért nem látszott célszerűnek a 60 dB-nél nagyobb elnyomás biztosítása a zárósávban.

A másik követelmény a szűrővel kapcsolatban, hogy az áteresztő tartományban a megszürt jelet minél kisebb mértékben módosítsa. Ehhez az kellene, hogy az áteresztő sáv ingadozása a lehető legkisebb mértékű, esetleg nulla legyen. Ezzel az amplitúdó torzulás hibáját tudjuk egy meghatározott szint alatt tartani. Természetesen az áteresztő sáv határán, az approximációtól függően fellép valamekkora amplitúdó csökkenés. Hogy ne legyen túl szigorú az előírás, és hogy az elnyomás hatása is hamarabb okozzon érzékelhető változást a jelben, ennek értékét 0,5 dB-re választottam, amely 5%-nál nagyobb amplitúdó csökkenést nem enged meg az áteresztő tartomány határán.

Fázis szempontjából a lineáris fázismenet lenne a legkedvezőbb, hiszen ez sok esetben követelmény lehet, de ez csak FIR szűrők alkalmazásával lehetséges. Ennél viszont nagyobb levágási meredekséggel rendelkező szűrők csak viszonylag magasabb fókuszám mellett valósíthatók meg.

Meg kell határozni még az átmeneti sáv szélességét. A szűrőnek lehetőleg minél nagyobb szelektivitást kellene biztosítani, hiszen a digitális megvalósításnak éppen ez az egyik előnye, hogy jóval meredekebb levágású szűrők készíthetők, mint analóg áramkörök segítségével. Azonban a meredek levágás is több olyan problémát von maga után, amelynek következményei jelentősen kihatnak a szűrők működésére. Az egyik ilyen probléma a stabilitás. A nagyobb levágási meredekséghez nagyobb fókuszámú szűrő tartozik. A nagyobb fókuszám következtében a pólusok sűrűbben fognak elhelyezkedni, egymáshoz viszonyított távolságuk kisebb lesz, így a szűrő hajlamosabb lesz a gerjedésre, mint kisebb fókuszám esetén. A másik problémát a számítási idő

okozza, ugyanis a szűrést megvalósító programrutinnak le kell futnia két mintavételi idő között, ami annyit jelent, hogy a szűrés folyamata nem haladhatja meg a 750 utasítást.

A választás szempontjából célszerűbbnek látszott minél alacsonyabb fokszerű számú szűrőket alkalmazni. Igazodva a szűrők frekvenciabeli elhelyezkedéséhez, amely szerint a beállítható törésponti frekvenciák körülbelül 1 oktávonként követik egymást, a kitűzött cél olyan fokszerű számú szűrők alkalmazása, amelyek a következő beállítható törésponti frekvenciáig megvalósítják a 60 dB elnyomást.

3.2 Szűrőtípus, approximációválasztás

A 3.1 fejezetben megtalálható néhány olyan követelmény, amelyeket a szűrőknek teljesítenie kell. Ahhoz, hogy a tervezésben tovább lehessen lépni, el kellett dönteni, hogy FIR vagy IIR szűrőket célszerűbb-e alkalmazni. Itt a legfőbb szempont a jelfeldolgozó processzoron történő megvalósíthatóság. A megtervezett szűrőknek olyanoknak kell lenniük, hogy a rendelkezésre álló számítási és együtttható pontosság mellett is teljesítsék a megszabott követelményeket a teljes frekvenciasávban. A másik követelmény a végrehajtási idő. A szűrés folyamatának mindenképp le kell zajlania két mintavételi idő között.

A szűrőkkel szemben támasztott előírások általában hibatarományokat határoznak meg, amely tartományon belül kell elhelyezkednie a szűrő átviteli függvénynek. Ilyen előírást mutatott be a 2.1 ábra. Az áteresztő-tartományi ingadozás mértéke az amplitúdóhiba legnagyobb értékét határozza meg. Az elnyomás mértéke pedig a maradékjel legnagyobb értékét írja elő a zárótartomány határa után. A záró és áteresztőtartomány között lévő átmeneti sávra általában nem adhatók meg előírások. Ennek alakulása az alkalmazott approximációtól függ. Az előírások meghatározását követően ki kell választani tehát egy megfelelő szűrőapproximációt, amelynek ismeretében a szükséges fokszerű szám már számítható.

A következő részben ismertetésre kerülnek mind a FIR, mind az IIR szűrők fontosabb jellemzői, amelyek segítségével eldönthető, melyik eljárással valósítható meg hatékonyabban, jelfeldolgozó processzor segítségével, a programozható szűrő.

3.3 A FIR szűrők jellemzői

A FIR szűrők a szűrőstruktúráknak egy speciális osztályát képviselik. Impulzusválaszuk az IIR szűrőkkel szemben véges. Átviteli függvénye a következő alakban írható fel:

$$H(z) = \sum_{n=0}^{N-1} b_n z^{-n}, \quad (3.3.1)$$

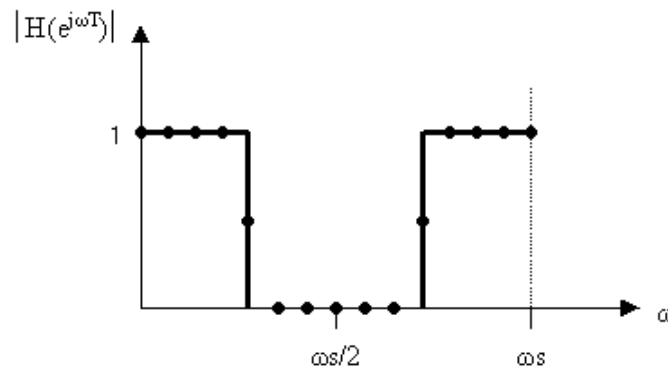
ahol N a megtervezett szűrő fokszerű számát jelöli. Ennek a szűrőosztálynak van néhány nagyon kedvező tulajdonsága, ami miatt érdemes vele foglalkozni. A 3.3.1 összefüggésből látható, hogy a FIR szűrők átviteli függvénye nem tartalmaz nevező polinomot, vagyis a nevező polinom összes együttthatója nulla. Annak következtében, hogy a struktúra mentes a visszacsatolástól, biztos, hogy stabil működésű lesz. A másik fontos tulajdonsága, hogy lineáris fázismentű szűrők tervezhetők, ha sikerül biztosítani

azt, hogy a rendszer $h(n)$ sűlysorozata konjugált komplex szimetriával rendelkezzen [1]. Mindezen előnyök után nézzük meg, milyen lehetőségeket kínál a FIR szűrők osztálya.

A FIR szűrők tervezése teljes mértékben eltér a hagyományos, analóg szűrőtervezés módszerétől. Három általános eljárás ismeretes, ezek a frekvencia mintavételezés, az ablakozás, és az optimális tervezés. Mindegyik eljárás más-más módon igyekszik optimalizálni a tervezett szűrőt.

3.3.1 Frekvencia mintavételezés

Az eljárás lényege, hogy a megkívánt amplitúdókarakterisztikát nem folytonosnak választjuk, hanem diszkrét frekvencia pontokban írjuk csak elő, milyen legyen az átvitele. Az amplitúdóátvitelt választhatjuk ideálisnak is, azaz az áteresztőtartományban az amplitúdó abszolút értéke egy, míg a zárótartományban nulla. Az átmeneti tartomány pedig nulla szélességű. Ez látható a 3.2 ábrán.



3.1. ábra Az ideális aluláteresztő szűrő amplitúdóválasza

A módszertől elvárjuk, hogy a meghatározott pontokban a számított karakterisztika teljes mértékben egyezzen meg az előírt pontok értékével. A köztes frekvenciákon a módszer nem garantál egyezést, de elég sűrűn felvéve a pontokat, az eltérés minimalizálható. Ez a módszer lényegében egy Lagrange-féle interpolációt hajt végre a meghatározott pontok között. Az interpoláló függvény felírható az egységgyökök segítségével [2]:

$$z^N = 1 = e^{jk2\pi} \quad (3.3.2)$$

Ennek az N -ed fokú egyenletnek a gyökein az úgynevezett egységgyökök, amelyekkel az interpoláló polinom

$$F_k(z) = \frac{1}{N} \frac{1 - (z_k z^{-1})^N}{1 - (z_k z^{-1})}, \quad k = 0, 1, \dots, N - 1, \quad (3.3.3)$$

alakban írható fel. Ez gyakorlatilag egy mértani sor összegképlete. Áttérve a frekvencia-tartományra, a következő összefüggést kapjuk:

$$F_k(\omega) = F_k(e^{j\omega T}) = \frac{1}{N} \frac{1 - (e^{j2\pi k/N} e^{-j\omega T})^N}{1 - (e^{j2\pi k/N} e^{-j\omega T})}, \quad (3.3.4)$$

amelyet továbbalakítva a 3.3.5-ös összefüggéshez jutunk

$$F_k(\omega) = e^{-j(N-1)\omega T/2} e^{jk\pi(N-1)/N} \frac{1}{N} \frac{\sin(N(\omega - \omega_k)T/2)}{\sin((\omega - \omega_k)T/2)}, \quad (3.3.5)$$

ahol:

$$\omega_k = k \frac{\omega_S}{N}, \quad k = 0, 1, \dots, N-1, \quad (3.3.6)$$

amely már teljes mértékben megfelel az interpoláció céljára, hiszen az adott $F_k(\omega_k)$ helyeken az értéke 1, míg a többi helyen 0. Ha ezután az F_k értéket megszorozzuk a meghatározandó átviteli függvény k -adik helyen lévő H_k értékével, akkor az adott pontban biztos, hogy meg fog egyezni az interpolációval előállított görbe a megkívánt függvényértékkel.

Felmerül azonban a foksám problémája. Mivel a frekvenciaskála egyenletes felosztású, így keskeny átmeneti tartományhoz nagyon magas foksámú szűrő tartozik. A foksám azonban csökkenthető, ha az átmeneti tartományban is meghatározunk pontokat. A 3.1 táblázat bemutatja, milyen összefüggés van az elérhető elnyomás és az átmeneti tartományban felvett pontok száma között [1].

Átmeneti sávban felvett pontok száma [db]	Átmeneti sáv szélessége [$\Delta\omega$]	Elérhető maximális elnyomás [dB]
0	1	<20
1	2	45-55
2	3	65-75
3	4	85-95

3.1. táblázat A zárósávi elnyomás az átmeneti sáv szélességének függvényében

3.3.2 Ablakozás

Az ablakozás lényege, hogy a megkívánt amplitúdókarakterisztikát, amely lehet akár egy ideális amplitúdókarakterisztika is (3.2 ábra), Fourier transzformáljuk, ezzel az impulzusválasz sorozathoz jutunk, majd ezt megszorozzuk egy $w(n)$ ablakfüggvénnyel. Erre azért van szükség, mert az előírt karakterisztika Fourier-sora egy végtelen sorozat. Az együtthatókkal szemben támasztott követelmény a véges számú tag és, hogy a sorozat kauzális legyen. Ezért a Fourier transzformáció következtében keletkező végtelen sorozatot végessé kell tenni, majd el kell tolni annyival, hogy ne legyenek negatív együttható érték. Az impulzussorozat csonkolása természetesen a szűrő karakterisztikáját is módosítja, így az kisebb-nagyobb mértékben el fog térni a megkívánt értékektől. Az ablakozás viszont csökkenteni igyekszik a csonkolás által okozott éles törést az impulzusválasz sorozatban. Az ablakozás hatására keletkező új súlysorozatot

$$g(n) = h(n)w(n), \quad n = 0, 1, \dots, N-1, \quad (3.3.7)$$

alakban írhatjuk fel. Most nézzünk meg néhány ismertebb ablakozó sorozatot, majd ezek tulajdonságait táblázatosan összefoglalva.

Négyszögablak:

$$w_R(n) = \begin{cases} 1, & n = 0, 1, \dots, N-1 \\ 0, & \text{egyébként} \end{cases}, \quad (3.3.8)$$

alakban írható fel, amely gyakorlatilag megegyezik a csonkolással. Ez meglehetősen gyenge minőségű szűrő tervezését teszi lehetővé, mivel fellép a Gibbs oszcilláció jelensége [2], amelynek következtében az első oldalhullám legnagyobb elnyomása nem haladja meg a -21 dB-t. Ez az érték a fokszámtól függetlenül állandó.

Hamming és Hanning ablakok:

$$w_H(n) = \begin{cases} a - 2b \cos \frac{2\pi n}{N}, & n = 0, 1, \dots, N-1 \\ 0, & \text{egyébként} \end{cases}, \quad (3.3.9)$$

amelyek csak az együtthatók értékeiben különböznek egymástól. Ezeket a két ablaknak megfelelően a következő értékekre kell beállítani [2]:

Hamming ablaknál: $a = 0,54$; $2b = 0,46$;

Hanning ablaknál: $a = 0,5$; $2b = 0,5$;

Ezek az ablakok már megpróbálják minimalizálni az áteresztő tartomány határán fellépő ingadozást, méghozzá úgy, hogy az első oldalhullámot, amely a legnagyobb hibát okozza, kiejtik. A hátránya viszont, hogy a négyszögablakhoz képest közel kétszeres fokszám szükséges ugyanakkora levágási meredekség eléréséhez.

Blackman ablak:

$$w_{Bl}(n) = \begin{cases} a - 2b \cos \frac{2\pi n}{N} + 2c \cos \frac{4\pi n}{N}, & n = 0, 1, \dots, N-1 \\ 0, & \text{egyébként} \end{cases}, \quad (3.3.10)$$

amely az eddigiekhez képest a legnagyobb elnyomást tudja biztosítani, amely annak a következménye, hogy a helyes együttható beállítás következtében nem csak az első, hanem a második oldalhullámot is minimálisra csökkentik. Ehhez az együtthatók értékeit a következő képen kell beállítani [2]:

$a = 0,42$; $2b = 0,5$; $2c = 0,08$;

Természetesen a nagy elnyomásnak is megvan a maga ára, amely a főhullám kiszélesedésében jelentkezik, ezen kívül a négyszögablakhoz képest itt már háromszoros fokszám szükséges ugyanakkora levágási meredekség eléréséhez.

A 3.2 táblázat összefoglalja az eddig ismertett ablakozó függvények segítségével tervezhető szűrők tulajdonságait [1]. A táblázatban ω_S a mintavételi körfrekvenciát jelöli.

	Főhullám szélessége [ω_s/N]	Átmeneti tartomány szélessége [ω_s/N]	Elérhető elnyomás [dB]
Négyszög	2	0,89	21
Hanning	4	3,1	44
Hamming	4	3,3	53
Blackman	6	5,5	74

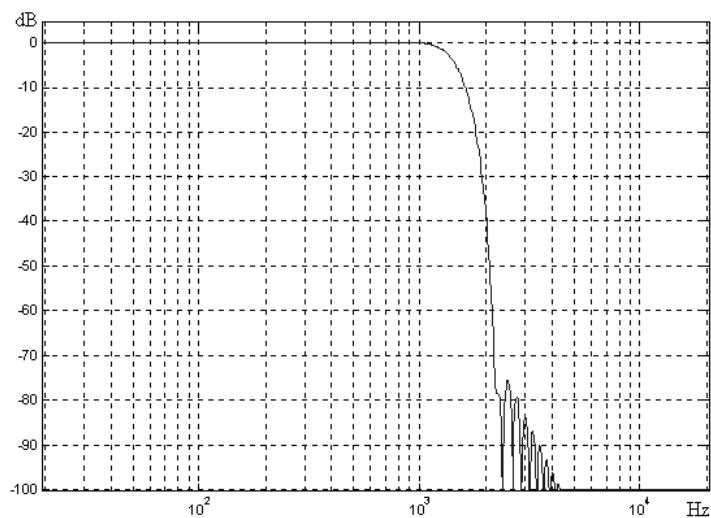
3.1. táblázat Ablakozással tervezett szűrők jellemzői

A FIR szűrőkre jellemző, hogy a főhullám szélessége határozza meg az átmeneti tartomány szélességét. Ezt csak a fokszám növelésével lehet keskenyebbre venni, tehát a kívánt levágási meredekségből meghatározható a szűrő fokszáma.

Figyelembe véve a 3.1 fejezetben támasztott követelményeket, megállapítható, hogy a szükséges 60 dB-es zárósávi elnyomást négyszögablakkal csak úgy lehet megvalósítani, ha az átmeneti sávot nagyon keskenyre választjuk. Ezzel a 60 dB-nél kisebb oldalhullámokat az előírt egy oktáv széles átmeneti sávba zsúfoljuk. Ehhez viszont extrém nagy fokszám szükséges. Hamming ablaknál is az első oldalhullámok értéke még nem haladja meg a 60 dB-t, ezért itt is csak nagy fokszámmal valósítható meg az előírt specifikáció. A legkedvezőbb a Blackman ablakkal tervezett szűrő. Bár, ha figyelembe vesszük, hogy a legkeskenyebb sávszélességű aluláteresztő szűrő 50 Hz-ig enged át, és közelítőleg egy oktáv után el kell nyomnia 60 dB-t, az átmeneti sáv szélessége 75 Hz-re adódik. Így a szükséges fokszám Blackman ablak esetén is:

$$N = 6 \frac{44,1kHz}{75Hz} = 3528,$$

ami szintén egy meglehetősen extrém érték. Ugyanilyen specifikáció mellett, 1 kHz-es szűrő esetén már csak 175 együtthatóra van szükség a megvalósíthatósághoz (3.3 ábra), ami már egy kedvezőbb érték.



3.1. ábra Blackman aluláteresztő szűrő amplitúdókarakterisztikája: $N = 175$

A paraméterek ismeretében a

`FIR1(N, Wn, Blackman(N+1))`

MATLAB függvénnyel meghatározhatók a szűrőegyütthetők, ahol N a foksám, Wn a relatív törésponti frekvencia, a mintavételi frekvencia felére vonatkoztatva. A Wn frekvenciához viszonyítva $+B/2$ -re van a zárósáv kezdete, ahol B az átmeneti tartomány szélességét határozza meg. Az együtthetők ismeretében a karakterisztika a például a `freqz()` függvénnyel kirajzolható. Ez látható a 3.3 ábrán.

Sajnos a FIR szűrők egyik legnagyobb hátránya, hogy a frekvenciatengelyt lineárisan kezelik, így keskenysávú, alacsony frekvenciatartományban dolgozó szűrőket csak igen magas foksám mellett lehet megvalósítani.

A Kaiser ablak:

$$w_K(n) = \frac{I_0 \left(\beta \sqrt{1 - \left(\frac{2n}{N-1} \right)^2} \right)}{I_0 \beta}, \quad n = 0, 1, \dots, N-1, \quad (3.3.11)$$

ahol I_0 az elsőfajú, zérusrendű, módosított Bessel függvény. A β paraméter pedig a következő összefüggés segítségével számítható ki:

$$\beta = 0,1102(A - 8,7), \quad A > 50 \text{ esetén.} \quad (3.3.12)$$

ahol A , a kívánt elnyomás mértéke az elnyomási tartományban. A β paraméter szerepe, hogy kompromisszumos megoldást teremtsen a főnyaláb szélessége, és az oldal-hullámok maximális szintje között. A 3.3 táblázat szemlélteti az átmeneti tartomány szélessége és az elérhető legnagyobb elnyomás közti összefüggést a β paraméter függvényében [1].

B	Átmeneti tartomány [ω_s/N]	Elérhető elnyomás (A) [dB]
2,0	1,5	29
3,0	2,0	37
4,0	2,6	45
5,0	3,2	54
6,0	3,8	63
7,0	4,5	72
8,0	5,1	81
9,0	5,7	90
10,0	6,4	99

3.2. táblázat A Kaiser ablak együtthetői és az elnyomás kapcsolata

A megvalósításhoz szükséges foksám a 3.3.13 összefüggés segítségével számítható:

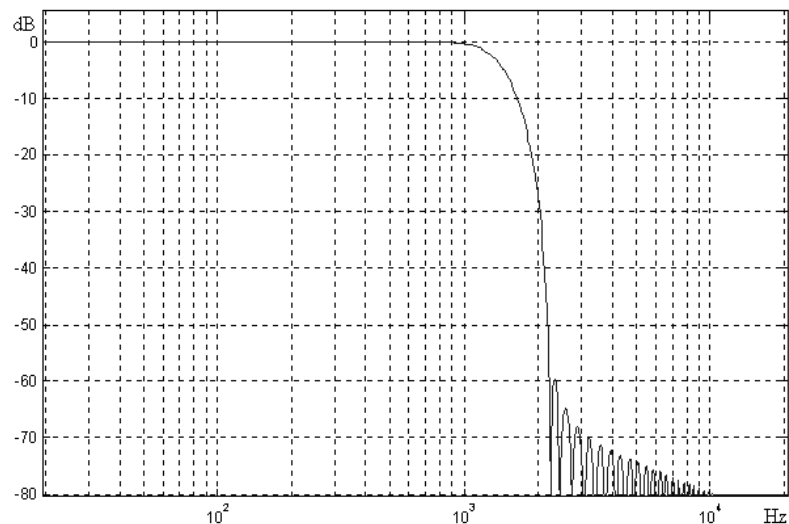
$$N \geq \frac{A - 7,95}{14,36\Delta\omega} + 1, \quad (3.3.13)$$

ahol $\Delta\omega$ az átmeneti sáv szélességét jelöli, a mintavételi körfrekvenciára vonatkoztatva.

Számításokkal itt is meghatározható, mekkora foksám szükséges a 3.1 fejezetben megadott specifikációjú szűrő megvalósításához. Például, ha 50 Hz-ig áteresztő szűrőt szeretnénk, amely egy oktáv után megvalósítja a 60 dB elnyomást, közelítőleg 3000 együtthetős szűrőre lenne szükség. Ugyanezt kiszámolva 1 kHz-es aluláteresztő szűrőre, már csak közelítőleg 110 együtthetőre van szükség. A paraméterek ismeretében a

```
FIR1(N, Wn, Kaiser(N+1, b))
```

MATLAB függvénnyel meghatározhatók a szűrőegyütthetők, ahol N a foksám, b a β paraméter értékét veszi fel, Wn pedig a relatív törésponti frekvencia, a mintavételi frekvencia felére vonatkoztatva. A Wn frekvenciához viszonyítva $+B/2$ -re van a zárósáv kezdete, ahol B az átmeneti tartomány szélességét határozza meg. Az együtthetők ismeretében a karakterisztika a például a `freqz()` függvénnyel kirajzoltatható. Az így megtervezett szűrő karakterisztikája látható a 3.4 ábrán.



3.2. ábra Kaiser aluláteresztő szűrő amplitúdókarakterisztikája: $N = 110$, $\beta = 5,6$

3.3.3 Optimális tervezés

Az eddig ismertett eljárásokkal tervezett szűrők egyik nagy hibája, hogy az átmeneti sáv szélessége, és az elérhető legnagyobb elnyomás között szoros összefüggés van. A szűrők elnyomása viszont nem egyenletes, aminek következtében néhány oktávval odébb már esetleg jelentősen túlteljesítjük a specifikációt. A nagy foksámra azonban szükség van, mivel az első oldalhullám csökkentése csak az átmeneti tartomány növelése árán érhető el. A probléma megoldására létezik egy optimális módszer, amely a zárósáv egyenletes ingadozás mellett a lehető legkisebb foksámmal valósítja meg az adott specifikációnak megfelelő szűrőt. Ez nem más, mint a Parks-McClellan algoritmus, amely Remez-féle kicserélési algoritmus néven is ismeretes. [2] Előnye, hogy az alacsonyabb foksám mellett megtartja a FIR szűrők kedvező tulajdonságait, így a lineáris fázismenetet is. Ez az eljárás az előzőekkel szemben azonban nem adható meg zárt formulákkal.

Optimális szűrő tervezhető a MATLAB Remez() függvényével is, amely a következő paraméterekkel hívható meg:

$$B = \text{REMEZ}(N, F, A, W);$$

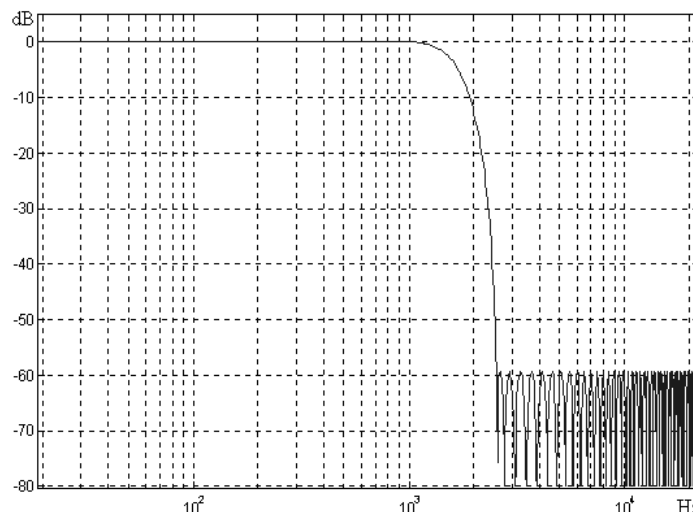
Ahol N a szükséges foksám, B a számított együtthatók $N+1$ elemű vektora, F a relatív frekvenciamenet leíró vektor a mintavételi frekvencia felére vonatkoztatva, A paraméter a megadott frekvenciapontokhoz tartozó elnyomás értékek definiálására szolgál, W pedig a súlyozó vektor (megadása nem szükséges).

Arra azonban ügyelni kell, hogy az F , A és W vektorok egyforma hosszúak legyenek. A szükséges foksám a 3.3.14 egyenlet alapján, a paraméterek behelyettesítése után adódik.

$$N \geq \frac{-10 \lg(\delta_1 \delta_2) - 11,7}{14,6 \Delta \omega} + 1, \quad (3.3.14)$$

ahol δ_1 az áteresztő sávi ingadozás, δ_2 pedig a zárósávi elnyomás relatív egységben és $\Delta \omega$ az átmeneti sáv szélességét jelöli, a mintavételi körfrekvenciára vonatkoztatva.

Az eddigiekhez hasonlóan itt is elvégeztem a számításokat, hogy nagyságrendileg megkapjam a szükséges együtthatószámot. A szigorúbb specifikáció itt is 50 Hz-es aluláteresztő szűrő, amelynek egy oktáv után el kell nyomnia 60 dB-t. Az áteresztő sávi ingadozást 0,01-ra választottam. A 3.3.14 összefüggés alapján a szükséges foksám közelítőleg 1950. Ez valóban jobb az eddigi eredményeknél, de még mindig meglehetősen nagy érték ahhoz, hogy jelfeldolgozó processzorral, két mintavételi idő között meg lehessen valósítani. Ugyanezt a számítást elvégezve 1 kHz-es szűrőre, a foksám 65-re adódik. Ez a karakterisztika látható a 3.5 ábrán. Az érdekesség, hogy a MATLAB a számított értéktől eltérően, csak 90-edfokú szűrővel tudta megvalósítani az előírt specifikációt.



3.1. ábra Remez algoritmussal tervezett aluláteresztő szűrő: $N = 90$

3.3.4 Összefoglalás

Sajnos az eddigiekből kiderül, hogy a FIR szűrők számos előnyös tulajdonsága mellett olyan problémák is felmerülnek, amelyek miatt a valós megvalósításuk meglehetősen nehézkes. A túlzottan keskeny sávú szűrők csak nagyon magas fokszámmal valósíthatók meg. Ezen csak a mintavételi frekvencia csökkentésével lehetne enyhíteni, hiszen így az átmeneti sáv relatív szélessége megnövekedne, amely egyértelműen kisebb fokszámot eredményez. Egy ilyen lehetőség a decimálás. De ez igazából csak az aluláteresztő szűrőknél jelentene megoldást. Átlapolódásgátló szűrőkkel, és mintaritkítással a mintavételi frekvenciát olyan tartományba lehet letranszformálni, ahol az előírt specifikáció már néhány tíz együtthatós szűrővel is megvalósítható. Természetesen ez egy többlépéses folyamat, de nagy előnye, hogy pipe-line feldolgozható. Igazából a módszer alkalmazását, és így a FIR szűrők használatát azért nem tartom célszerűnek a feladat megoldására, mert feluláteresztő és sávszűrő megvalósításához már nem elég hatékony. Az előírt követelmény ugyanis a 20 kHz-es sáv szélesség biztosítása. Tehát a lecsökkentett mintavételi frekvenciát mindenképp vissza kell transzformálni az eredeti 44,1 kHz-es frekvenciára. Ehhez elengedhetetlen az interpoláció alkalmazása. Így amit nyerünk a decimálással (utasításszám csökkenés), azt elveszítjük az interpolálás végrehajtásával. Az interpoláció másik hátránya, hogy a jelalaknak a közelítését csak bizonyos hibával tudja elvégezni. Ez természetesen függ az alkalmazott interpolációs eljárástól. Így tehát a feladat megvalósításához a FIR szűrők alkalmazását, minden előnyös tulajdonságával együtt elvettem.

A CD mellékleten megtalálható a `szuro_FIR.m` fájl, amellyel az eddig bemutatott szűrők amplitúdó átviteli függvényét lehetett kiszámítani, majd kirajzoltatni. Ennek a MATLAB fájlnak a segítségével készültek a 3.3, 3.4 és 3.5 ábrák.

3.4 Az IIR szűrők jellemzői

Az IIR szűrők olyan speciális szűrőstruktúrák, amelyek végtelen impulzusválasszal rendelkeznek. Átviteli függvénye a következő:

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{m=0}^M b_m z^{-m}}{1 - \sum_{k=1}^N a_k z^{-k}} = b_0 \frac{\prod_{m=0}^M (1 - z_m z^{-1})}{\prod_{k=1}^N (1 - p_k z^{-1})}. \quad (3.4.1)$$

A $B(z)$ és $A(z)$ polinomok együtthatói valósak, míg a polinomok gyökeit jelölő z_m zérusok és p_k pólusok lehetnek valósak és komplexek is. Ahhoz, hogy a rendszer stabil maradjon, a pólusok csak az egységkörön belül helyezkedhetnek el. A zérusokra elvileg nincs semmilyen megkötés, de célszerű azokat az egységkörön elhelyezni, mivel így a zárttartomány gazdaságosabban közelíthető [1].

Mivel az IIR szűrők impulzusválasza végtelen, hasonlóan az analóg szűrőkéhez, ezért az IIR szűrők tervezését leggyakrabban az analóg szűrőtervezésre vezetik vissza. Ennél a módszernél egy gyakran használt eljárás a bilineáris transzformáció, amely a diszkrét és folytonos időtartomány közötti átmenetet valósítja meg, vagyis az s sík és z sík között teremt kapcsolatot. Az eljárás lényege, hogy először a követelményeket át kell transzformálni a folytonos időtartományba, majd itt megtervezni az előírásoknak

megfelelő analóg szűrőt, ezután a kapott zérusokat és pólusokat visszatranszformáljuk a diszkrét időtartományba.

Ez a többlépcsős eljárás azért előnyös, mert az analóg tervezés módszerei gyakorlatilag már teljesen kiforrottak, és nagyon sok függvény és eljárás ismeretes, amely a tervezést segíti. Az analóg tervezés alapja a referencia aluláteresztő szűrő. Ennek határfrekvenciája az egység. A specifikációt mindig erre a normalizált tartományra transzformáljuk, majd katalógus segítségével kikeressük, mely referenciaszűrő teljesíti az előírást. Ezután el kell végezni egy típus-transzformációt, hogy megkapjuk az eredeti követelményeknek eleget tevő szűrőt.

A megtervezett szűrő frekvenciamenetét a felhasznált approximáció határozza meg. Ez megegyezik az analóg szűrőknél megismert karakterisztikus függvényekkel. Ezeket átranzformálva a diszkrét időtartományba, felhasználhatók a szűrők megvalósításához.

3.4.1 Szűrőkarakterisztikák közelítésének módszerei

Vezessünk be néhány jelölést: ω_{aref} és ω_{zref} a referencia aluláteresztő szűrő áteresztő sávjának határát és a zárósáv kezdetét jelölik. A δ_1 és δ_2 az áteresztőtartománybeli ingadozás, és a zárósávi elnyomás értéke (3.1 ábra). Ezek segítségével definiálható még két változó:

$$\varepsilon^2 = \frac{1}{(1 - \delta_1)^2} - 1, \quad d = \sqrt{(1/\delta_2^2 - 1)/\varepsilon^2}, \quad (3.4.2)$$

ahol ε a szűrő csillapítását határozza meg az ω_{aref} frekvencián.

A maximális-lapos (Butterworth) közelítés:

Ez a közelítés olyan, hogy mind az áteresztő, mind a zárósávban maximálisan lapos amplitúdó átviteli karakterisztikát eredményez. Átviteli függvénye a következő:

$$|H(e^{j\omega T})|^2 = \frac{1}{1 + \varepsilon^2 (\omega / \omega_{aref})^{2N}} \quad (3.4.3)$$

Az áteresztősáv határán, $\omega = 1$, a maximális hiba $\varepsilon = 1$ esetén -3 dB, mivel itt a függvény értéke:

$$20 \lg(H^{ej1T}) = 20 \lg \sqrt{\frac{1}{1 + \varepsilon^2}}, \quad [dB] \quad (3.4.4)$$

csak az ε értékétől függ. Ha meghatározzuk az áteresztő sáv határán a megengedhető maximális eltérés nagyságát, ami gyakran követelmény, az ε értéke a 3.4.2 összefüggés segítségével számítható. Az áteresztő sáv határától elegendően nagy frekvencián az elnyomás értéke már $1/\omega^N$ -nek megfelelő, azaz $6N$ dB oktávonként, ahol N a fokszámot jelöli. A teljes tartományban a csillapítás a 3.4.5 szerint számítható.

$$A_{\omega \rightarrow \infty} = 20 \lg(\varepsilon) + 20N \lg(\omega / \omega_{aref}) \quad [dB]. \quad (3.4.5)$$

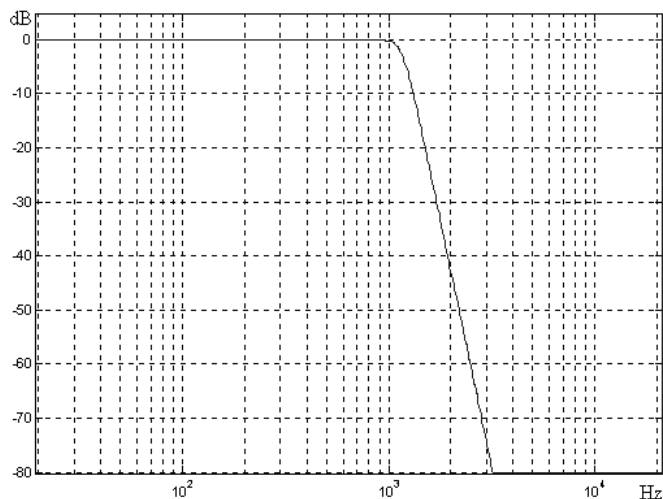
A szükséges foksám a zárótartomány határán megszabott elnyomásérték alapján számítható:

$$N \geq \frac{\lg(d)}{\lg(\omega_{zref})}. \quad (3.4.6)$$

A polinomra jellemző, hogy $2N$ darab pólusa van, amelyek az egységkörön helyezkednek el, egymástól egyenletes távolságban. Páratlan foksám esetén a valós tengelyre is kerülhet pólus, azonban a képzetes tengelyre még páratlan foksám esetén sem jut a pólusokból. A zérusok, amelyből N darab van, mind az $\omega = \infty$ frekvencián helyezkednek el [1]. Az, hogy a pólusok ilyen egyenletesen, és egymástól a lehető legtávolabb helyezkednek el, ezt a struktúrát sokkal stabilabbá teszi a többinél.

Ennek a karakterisztikának a hátrány, hogy nem biztosít elég meredek levágást. Az elnyomás is csak jóval az áteresztő tartomány határa után kezd meredekké válni, így nem biztosít kellő szelektivitást sem. Az áteresztő sáv határától távolabb azonban már feleslegesen nagy az elnyomás mértéke.

Ahogy a FIR szűrőknél, itt is elvégeztem a számításokat a foksámra vonatkozólag. A specifikáció ugyanaz: 50 Hz-es aluláteresztő szűrő, amelynek egy oktáv után el kell nyomnia 60 dB-t és az áteresztő sáv határán az eltérés nem lehet nagyobb 0,5 dB-nél. A 3.4.6 alapján elvégzett számítások után a foksám 9-re adódik. Mivel a foksám számításánál az áteresztősáv határához, és nem pedig a mintavételi frekvenciához viszonyítjuk ω_{zref} -et, ezért az így megtervezett szűrő az egész frekvenciatartományban biztosítja ugyanezt a levágási meredekséget. A 3.6 ábra egy 1 kHz-ig áteresztő, Butterworth szűrő amplitúdó karakterisztikáját mutatja be.



3.1. ábra Butterworth aluláteresztő szűrő amplitúdó karakterisztikája: $N = 9$

A tervezést természetesen itt is számos MATLAB függvény segíti. A

$$[B, A] = \text{BUTTER}(N, Wn);$$

függvény segítségével a szűrőegyütthatókat határozhatjuk meg. N a foksámot, Wn pedig az áteresztő sáv határát jelöli relatív értékben a mintavételi frekvencia felére vonatkoztatva. Ez a függvény alapértelmezésben az áteresztő sáv határán -3 dB hibát enged meg. Célszerű az együtthatók számítása előtt előbb meghívni a

$$[N, W_n] = \text{BUTTORD}(W_p, W_s, R_p, R_s);$$

függvényt. Itt W_p jelöli az áteresztősáv határát, W_s a zárósáv kezdetét, R_s a zárósávi elnyomás értékét W_s frekvencián, R_p pedig az áteresztő sáv határán megengedhető hiba legnagyobb értékét. W_s és W_p értékét relatív frekvenciaként kell megadni. Ha az áteresztősáv határán nem 3 dB hibát írunk elő, akkor a W_n vektor tartalmazni fogja az ehhez tartozó relatív frekvencia értékeket. N természetesen a szükséges fokszámot tartalmazza.

Az egyenletes ingadozású (Csebisev) közelítés:

A Csebisev közelítés már jóval nagyobb szelektivitást és levágási meredekséget biztosít, mint a Butterworth közelítés. Ennek azonban megvan az ára. Elveszítjük azt a kedvező tulajdonságot, hogy az áteresztőtartományt monoton közelítsük. A Csebisev polinom ugyanis az áteresztő tartományban egyenletes ingadozású, míg a záró tartományban maximálisan lapos az amplitúdó karakterisztikájának átvitele. Mivel az egyenletes amplitúdómenet gyakran követelmény az áteresztő tartományban, ezért a Csebisev approximációból, transzformációk útján előállították az úgynevezett inverz Csebisev polinomot, amely minden tulajdonságában megegyezik az eredetivel, kivéve, hogy az egyenletes ingadozás most a zárósávba került. Sajnos a szelektivitás így valamelyest csökkent, az áteresztősáv maximálisan lapos közelítés miatt. Az N -edfokú Csebisev polinom definíciója [1]:

$$T_N(x) = \begin{cases} \cos[N \cos^{-1}(x)], & x \leq 1 \\ \cosh[N \cosh^{-1}(x)], & x \geq 1 \end{cases}, \quad (3.4.7)$$

amelyet felhasználva a szűrő amplitúdó válaszának a négyzete:

$$|H(e^{j\omega T})|^2 = \frac{1}{1 + \varepsilon^2 T_N^2(\omega / \omega_{aref})}, \quad (3.4.8)$$

alakban írható fel. Az elnyomás értéke a teljes frekvenciatartományon az:

$$A_{\omega \rightarrow \infty} = 20 \lg(\varepsilon) + (N - 1)6,0206 + 20N \lg(\omega), \quad [dB], \quad (3.4.9)$$

összefüggésből számítható. Látható, hogy ez az érték $(N - 1) 6,0206$ dB-lel magasabb, mint az azonos áteresztő csillapítású maximálisan lapos szűrőé. Ez a meredekebb levágás már nem sokkal az áteresztő tartomány határa után is megfigyelhető. A szükséges fokszám, amely itt jóval kisebb, mint a Butterworth szűrőnél, a következő képlet segítségével számítható:

$$N \geq \frac{\cosh^{-1}(d)}{\cosh^{-1}(\omega_{zref})}. \quad (3.4.10)$$

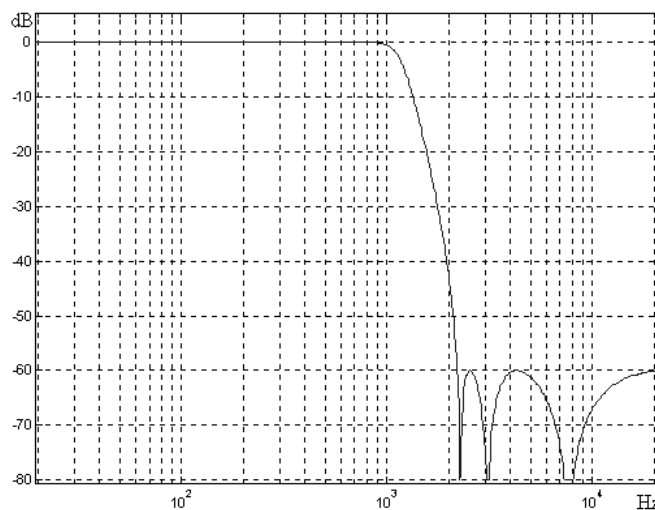
Az inverz Csebisev approximáció az előző approximációból származtatható kétszeres transzformáció segítségével:

$$|H(e^{j\omega T})|^2 = \frac{1}{1 + \varepsilon^2 \left[\frac{T_N^2(\omega_{zref})}{T_N^2(\omega_{zref} / \omega)} \right]} \quad (3.4.11)$$

Ezekre a polinomokra jellemző, hogy pólusaik az egységkör belsejében egy ellipszisen helyezkednek el. A zérusok, Csebisev polinom esetén a végtelen frekvencián, míg az inverz Csebisev szűrőknél a $j\omega$ tengelyen helyezkednek el [2].

Az előző két approximációból a feladat megoldásához az áteresztő sávot egyenletesen közelítő eljárás tűnik a kedvezőbbnek, hiszen a cél az volt, hogy az amplitúdó nagysága az áteresztő sávban állandó legyen. Bár a normál Csebisev approximációnál is megadható olyan kis ingadozás, amely csak nagyon kismértékben módosítaná az amplitúdó értékét az áteresztő tartományban, de ennek következménye az együtthatók jóval nagyobb érzékenysége lenne. Ez viszont mindenképp rontaná a szűrő tulajdonságait.

A maximálisan lapos közelítésnél megadott szűrőspecifikáció alapján itt is elvégeztem a számításokat. Ugyanaz a szűrő inverz Csebisev közelítéssel már kevesebb, mint hatodfokú szűrőt igényel. A MATLAB `freqz()` függvénnyel kirajzoltatott amplitúdó-karakterisztikája a 3.7 ábrán látható.



3.2. ábra Inverz Csebisev aluláteresztő szűrő amplitúdó-karakterisztikája: $N = 6$

A tervezést segítő MATLAB függvények a következők:

$$[B, A] = \text{CHEBY2}(N, R, Wn);$$

ahol B és A az $N+1$ -edfokú számláló és nevező polinom együtthatói, R a zárósávi elnyomás értéke decibelben, Wn pedig a zárósáv kezdetét megadó relatív számérték, a mintavételi frekvencia felére vonatkoztatva. Itt is érdemes még lefuttatni a

$$[N, Wn] = \text{CHEB2ORD}(Wp, Ws, Rp, Rs);$$

függvényt, ahol Wp jelöli az áteresztősáv határát, Ws a zárósáv kezdetét, Rs a zárósávi elnyomás értékét Ws frekvencián, Rp pedig az áteresztő sáv határán megengedhető hiba legnagyobb értékét. Ws és Wp értékét relatív frekvenciaként kell megadni. A függvény

az előírásnak megfelelően beállítja az áteresztőtartomány határát relatív frekvenciában, amely a Wn vektorba kerül. N természetesen a szükséges fokszámot tartalmazza.

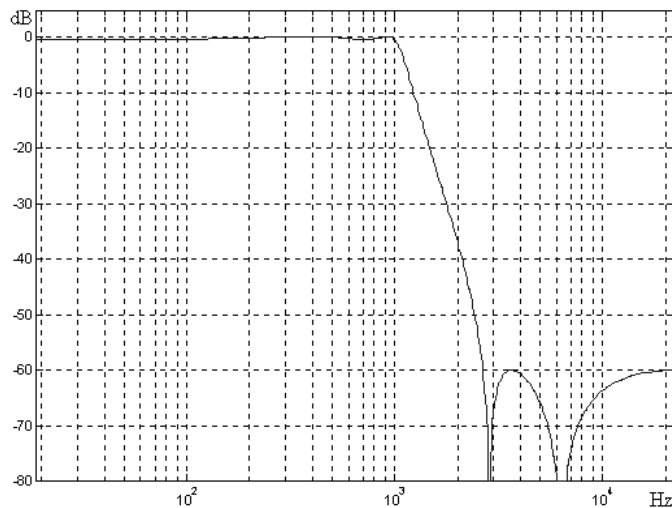
Mint az eddigiekből kiderül, a zárótartomány vagy az áteresztőtartomány egyenletes közelítése a fokszám szempontjából optimálisabb szűrőt eredményez, mint maximálisan lapos közelítést alkalmazva. Ezt a gondolatmenetet folytatva, a legnagyobb szelektivitást az az approximáció biztosítja, amely mind az áteresztő, mind a zárósávot egyenletes ingadozással közelíti.

Az egyenletes ingadozású (Cauer) közelítés:

A Cauer approximáció tehát oly módon valósítja meg az átviteli karakterisztikát, hogy mind az áteresztő, mind a zárótartományt egyenletes ingadozással közelíti. A polinom alakja a következő:

$$\left|H(e^{j\omega T})\right|^2 = \frac{1}{1 + \varepsilon^2 U_N^2(\omega / \omega_{ref})}, \quad (3.4.12)$$

ahol $U_n(x)$ a Jacobi elliptikus függvény [1]. A Csebisev approximációhoz hasonlóan, a pólusok itt is az egységkör belsejében, míg a zérusok a $j\omega$ tengelyen helyezkednek el. Annak ellenére, hogy az így megvalósított szűrő a fókuszszám szempontjából optimálisnak tekinthető, nem szabad megfeledkezni arról, hogy az együtthatók érzékenységének tekintetében viszont ez a legkritikusabb szűrő, valamint, hogy az összes approximáció közül ennek a fázisa tér el a legjobban a lineáristól. A Cauer szűrő amplitúdóválasza a 3.8 ábrán látható.



3.3. ábra Cauer aluláteresztő szűrő amplitúdóválasza: $N = 4$, $R_p = 0,5$ dB

A tervezésnél felhasznált MATLAB függvények:

$$[B, A] = \text{ELLIP}(N, R_p, R_s, W_n);$$

Itt a zárósávi elnyomás R_s mellett, meg kell adni az áteresztősávi ingadozás R_p értékét is. Mivel itt ismeretes az áteresztő sávban az ingadozás legnagyobb értéke, ezért W_n az áteresztő tartomány határát jelöli ki, ahol az eltérés nem haladja meg az előírt értéket. A fókuszszám meghatározásához érdemes felhasználni az

$$[N, W_n] = \text{ELLIPORD}(W_p, W_s, R_p, R_s);$$

függvényt, mivel kézi módszerrel a fókuszszám meghatározása jóval bonyolultabb, mint az előző két esetben. Az eddigiekhez képest azonban ugyanazon specifikációt már 4-ed-fokú szűrővel is teljesíteni lehet.

3.4.2 Összefoglalás

Amint az eddigiekből kiderült, az IIR szűrők egyik nagy előnye, hogy az adott specifikáció teljesítése jóval kisebb fokszámmal lehetséges, mint FIR szűrők esetén. Mivel az IIR szűrőknél a frekvenciatartomány felosztása logaritmikus léptékű, ezért egy adott fokszámhoz tartozó levágási meredekség (decibel/oktávban mérve) az egész frekvenciatartományban azonos lesz. Azonban az IIR szűrők alkalmazásának is megvannak a hátrányai. A fázismenet nemlineáris, amely viszont sok alkalmazás szempontjából követelmény lehet. Másik kevésbé előnyös tulajdonság, hogy a rendszer tartalmaz visszacsatolást, így, míg a FIR struktúra teljesen stabil, az IIR szűrők hajlamosak a gerjedésre.

IIR szűrőknél a stabil működés feltétele, hogy a 3.4.1 összefüggésben a nevező polinom pólusai az egységkör belsejében helyezkedjenek el. Nem túlságosan szigorú specifikáció esetén ez nem is igen okoz problémát. Ilyenkor az együtthatók érzékenysége is jóval kisebb. Viszont keskenysávú szűrőknél a pólusok abszolút értéke már nagyon megközelítik az egyet, és az egymáshoz viszonyított távolságuk is jelentősen lecsökken, amely nagyon érzékenyvé teszi a rendszert. Ez a probléma DSP környezetben, a véges szóhosszúság miatt fokozottan jelentkezik.

A választás mégis az IIR szűrőkre esett. Ennek egyik oka, hogy jelentősen kisebb fokszámmal valósítható meg a tervezett specifikáció, továbbá sok olyan struktúra került már kidolgozásra, amelyek gyakorlati szempontból már megfelelően stabil működést garantálnak.

Tulajdonságai alapján az inverz Csebisev approximáció a legkedvezőbb. Az áteresztő tartományt monoton közelíti, amely előírt követelmény, a levágási meredeksége pedig jóval nagyobb, mint az azonos fokszámú Butterworth szűrőé. Bár a maximális lapos közelítés hátránya, hogy csak jóval az áteresztő sáv határa után kezd érezhető elnyomást produkálni. Viszont nem okoz ingadozást a jel amplitúdójának nagyságában. A legnagyobb hiba itt az áteresztő sáv határán lép fel, amely a specifikációnak megfelelően nem lehet nagyobb -0.5 dB-nél.

A 3.1 fejezetben leírt követelmények inverz Csebisev approximációval már $N = 6$ fokszám esetén teljesíthetők, így a szűrőegyütthatók előállításához már minden specifikációs adat rendelkezésre áll. Természetesen ezek csak egy irányelvet adnak, amit a lehetőségekhez képest be kell tartani. Ugyanis a valós működés dönti el, hogy a megkövetelt specifikáció nem túlzottan szigorú-e, azaz hogy az alkalmazott struktúrával és az adott jelfeldolgozó processzorral meg lehet-e valósítani a szűrőket.

Az IIR szűrők együtthatóinak számításához, és azok amplitúdó átvitelének megjelenítéséhez elkészítettem a `szuro_IIR.m` MATLAB programot, amely a CD mellékleten is megtalálható. A 3.6, 3.7 és a 3.8 ábrák is ennek segítségével készültek el.

3.5 A megvalósítási forma kiválasztása

A 3.3 és a 3.4 fejezetekben megvizsgáltam a lehetőségeket, mely approximáció alkalmazása lenne a legkedvezőbb a szűrők megvalósításához. A specifikáció meghatározása és a kiválasztott approximáció ismeretében a szükséges szűrőegyütthatók már kiszámíthatók. A probléma már csak az, hogy milyen struktúrával

legyenek megvalósítva a szűrők, mely struktúra biztosítja a lehető legkedvezőbb tulajdonságokat működés közben. Mivel a megvalósítás jelfeldolgozó processzorral történik, ezért a választásnál figyelembe kellett venni néhány fontos szempontot.

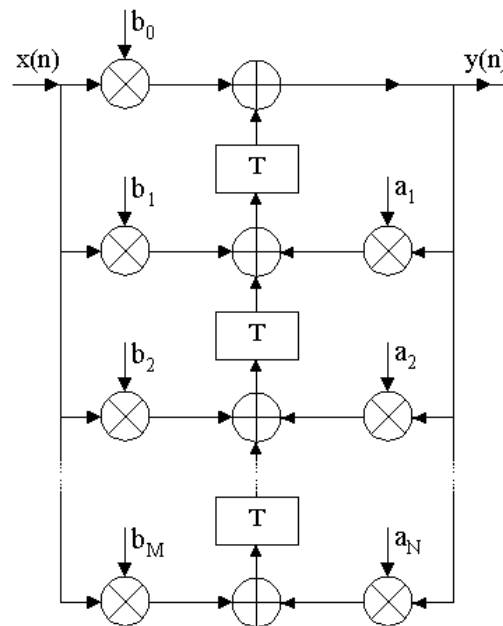
Mivel a szóhosszúság és a számítási pontosság véges, olyan struktúrára van szükség, ahol az együtthatók pontatlanságára, és a számítások elvégzése után szükséges kvantálásra nem túlságosan érzékeny a kiválasztott megvalósítási forma. Továbbá a számításoknak bele kell férni egy mintavételi ciklusba, amely szerint a szűrés, kétszatornás működést feltételezve, nem haladhatja meg a 375 utasítást.

IIR szűrőknél a legegyszerűbb, az úgynevezett direkt struktúra (3.9 ábra) azért nem alkalmazható, mert az együtthatók szempontjából igen érzékeny.

A direkt struktúra a 3.4.1 átviteli függvényt oly módon állítja elő, hogy az abból felírható differenciaegyenletet

$$y(n) = \sum_{m=0}^M b_m x(n-m) + \sum_{k=1}^N a_k y(n-k), \quad (3.5.1)$$

közvetlenül valósítja meg. A fenti differenciaegyenletben b_m és a_k konstans szorzótényezők. Ezen kívül azonban még számos egyéb megvalósítási forma is létezik, amelyek tulajdonságai jóval kedvezőbbek, mint a direkt struktúráé.



3.1. ábra IIR szűrő direkt megvalósítása

A döntésben, hogy mely realizációt érdemes alkalmazni, egy korábbi diplomatervezés segített [3]. Ez alapján két forma, a kaszkád és a rezonátoros struktúra kipróbálása tűnt célszerűnek. A kaszkád struktúra elsősorban az egyszerűbb programozás miatt látszott alkalmasnak a feladat megvalósítására. A belsejében rezonátorokat tartalmazó forma viszont nagyon kedvező érzékenységi tulajdonságokkal rendelkezik. Ennek programozása azonban már komolyabb feladatot jelent.

3.5.1 Kaszkád megvalósítás

A probléma abból adódik, hogy keskenysávú szűrők direkt megvalósítása gyakorlatilag lehetetlen feladatnak bizonyul. A mintavételi frekvenciától távol, a mintavételi frekvenciához képest relatív keskeny sávban működő szűrőknél az együtthatók érzékenysége olyan nagy lehet, hogy már nagyon kismértékű változás is teljesen elronthatja a megvalósított szűrő átvitelét. Erre adhat megoldást a kaszkád tagokra bontás.

Az IIR szűrő átviteli függvénye a következő alakban írható fel:

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{m=0}^M b_m z^{-m}}{1 - \sum_{k=1}^N a_k z^{-k}} = b_0 \frac{\prod_{m=0}^M (1 - z_m z^{-1})}{\prod_{k=1}^N (1 - p_k z^{-1})}, \quad (3.5.2)$$

ahol z_m zérusok a $B(z)$ polinom gyökeit, míg p_k pólusok az $A(z)$ polinom gyökeit jelölik. A pólusokra elvégezve a számítást, az i -edik pólus a 3.5.2 összefüggés nevező polinomjának gyökeként áll elő, gyökmegoldó képlet segítségével:

$$p_i = f(a_1, a_2, \dots, a_N). \quad (3.5.3)$$

Az elvi pontos együttható a_k felírható a számított együttható és a köztük lévő eltérés segítségével:

$$a_k = a_k^0 + \Delta a_k, \quad (3.5.4)$$

ahol a_k^0 jelöli a számított értéket. A 3.5.3-ban megadott N változós függvény együttható pontatlanságának hatására történő gyök megváltozása felírható a teljes differenciál segítségével:

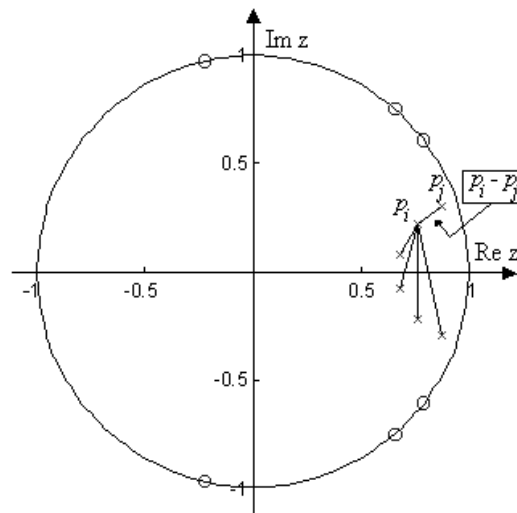
$$p_i = p_i^0 + \sum_{k=1}^N \frac{\partial p_i}{\partial a_k} \Delta a_k. \quad (3.5.5)$$

Az i -edik gyök k -adik együtthatóra vonatkoztatott érzékenysége kiszámítható a 3.5.6 összefüggés segítségével:

$$\frac{\partial p_i}{\partial a_k} = \frac{\frac{\partial A(z)}{\partial a_k} \Big|_{z=p_i}}{\frac{\partial A(z)}{\partial p_i} \Big|_{z=p_i}} = - \frac{p_i^{N-k}}{\prod_{j=1; j \neq i}^N (p_i - p_j)}. \quad (3.5.6)$$

Az érzékenység a számláló zérusaira vonatkoztatva is ugyanezzel a módszerrel határozható meg. A 3.5.6 formula alapján látható, hogy az i -edik gyök érzékenysége a többi gyök egymáshoz viszonyított távolságától függ. Ezt szemlélteti a 3.10-es ábra is. Keskeny sáv szélességű szűrők esetén a gyökök egymáshoz képest nagyon közel helyezkednek el. Ennek következménye, hogy szorzatuk nagyon kis szám lesz, vagyis az érzékenység nagyon megnő. Már kis együttható pontatlanság esetén is a pólusok

jelentősen elmozdulhatnak, aminek következtében az átviteli karakterisztika jelentősen el fog térni a megkívánt karakterisztikától, még rosszabb esetben a pólus az egységkörön kívülre kerül, ami instabil működést eredményez.



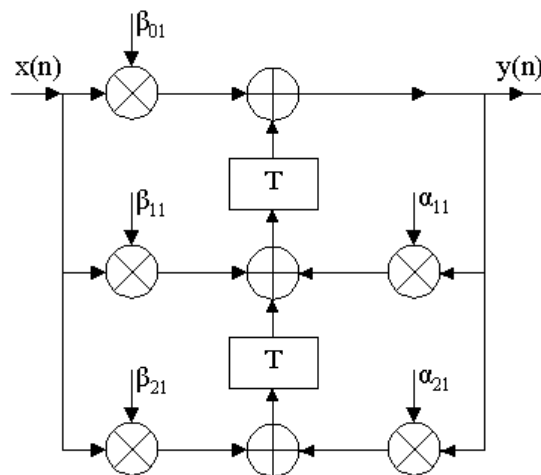
3.1. ábra IIR szűrő pólus-zérus elhelyezkedése

Az eddig leírtakra ad megoldást a másodfokú tagokra bontás módszere. A másodfokú tagokban (3.11 ábra) a gyökök komplex konjugált párjukkal szerepelnek, így a 3.5.6 összefüggésben a nevező értékét e két gyök távolsága adja, amely jóval nagyobb érték, mint az összes gyök együttes távolságának szorzata, tehát az érzékenység nagyságrendekkel csökkent. Probléma csak a mintavételi frekvenciához képest nagyon kis frekvenciákon léphet fel, hiszen ekkor már a két gyök távolsága ismét jelentősen lecsökken, és az érzékenység megnő.

A kaskád struktúra tehát alapvetően másodfokú tagokból épül fel:

$$H_i(z) = \frac{\beta_{0i} + \beta_{1i}z^{-1} + \beta_{2i}z^{-2}}{1 - \alpha_{1i}z^{-1} - \alpha_{2i}z^{-2}} \quad (3.5.7)$$

A 3.5.7 összefüggéssel megadott átviteli függvényt megvalósító struktúra a 3.11 ábrán látható.



3.2. ábra másodfokú alaptag

A teljes átvitel ezeknek a tagoknak a szorzatából áll:

$$H(z) = \prod_{i=1}^L H_i(z), \text{ ahol } L = [(N+1)/2]_{\text{egészrész}}. \quad (3.5.8)$$

Páratlan foksám esetén tehát marad még egy elsőfokú alaptag is. Ez minden esetben valós. Az átviteli függvény gyökeinek ismeretében a kaszkád együtthatók a következő összefüggések segítségével határozhatók meg:

$$\begin{aligned} \alpha_{1i} &= P[i_1] + P[i_2], & \alpha_{2i} &= -P[i_1]P[i_2], \\ \beta_{0i} &= \frac{1 - \alpha_{1i} - \alpha_{2i}}{1 + Z[i_1]Z[i_2] - (Z[i_1] + Z[i_2])}, & & (3.5.9) \\ \beta_{1i} &= -\beta_{0i}(Z[i_1] + Z[i_2]), & \beta_{2i} &= \beta_{0i}Z[i_1]Z[i_2], \end{aligned}$$

ahol $Z[i_1]$, $Z[i_2]$ az i -edik kaszkád taghoz tartozó zérusokat, $P[i_1]$, $P[i_2]$ pedig ugyanehhez a taghoz tartozó pólusokat jelöli. A zérusok és pólusok helyes összerendelése a stabil működés szempontjából nagyon lényeges. A komplex konjugált párt alkotó pólusokhoz csak komplex konjugált zérusok tartozhatnak. A kaszkád tagok sorrendjénél figyelembe kell venni a jósági tényezőt. A kivezérelhetőség és stabilitás szempontjából a legkedvezőbb megoldás, ha a másodfokú tagokat növekvő jósági tényező szerint kapcsoljuk sorba [1]. Egy másodfokú tag jósági tényezője a z síkon a pólusainak egysugarú körtől mért távolságával arányos. A nagyobb jósági tényezővel rendelkező tagok kiemelése is nagyobb az áteresztő tartományban, ami túlvezérlést okozna. Ezért kell legelől lennie az egysugarú körtől legtávolabbi elhelyezkedő pólusokat tartalmazó tagnak, mert így a túlvezérlés elkerülhető. A zéruspárokat úgy kell hozzárendelni a póluspárokhoz, hogy azok távolsága a legkisebb legyen. Az összerendelést itt is csökkenő jóság szerint kell elvégezni.

A fenti összerendelést és a kaszkád együtthatók meghatározását elvégezhetjük a MATLAB

```
[SOS,G] = TF2SOS(B,A);
```

függvényével. A két bemeneti paraméter a megtervezett szűrő számláló A és nevező B polinomja. Ezek alapján a függvény előállít egy $6 \times N$ típusú mátrixot SOS , ahol a mátrix sorai tartalmazzák az együtthatókat $\beta_{0i} \beta_{1i} \beta_{2i} 1 \alpha_{1i} \alpha_{2i}$ formában. A G a kaszkád tagok eredő erősítését tartalmazza, amelyet a β_{0i} együtthatók között kell elosztani.

3.5.2 Összefoglalás

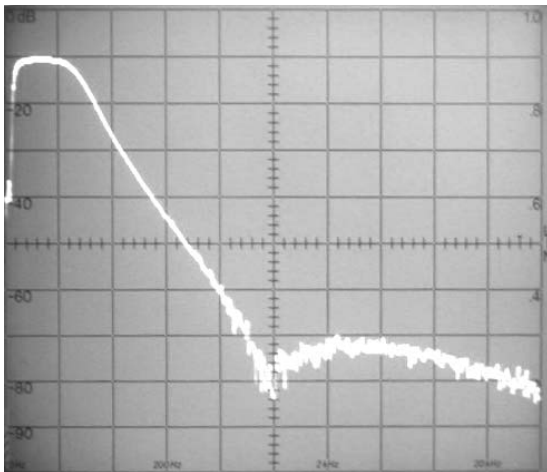
Az együttható érzékenység ugyan jelentősen csökkent, nagyságrendekkel jobb eredmény érhető el, mint direkt megvalósítás esetén, de ennek is megvan a maga korlátja. Igen alacsony frekvenciákon, keskenysávú szűrők esetén az együttható érzékenység problémája ismét jelentkezik. Ennek eredményeként az átviteli karakterisztika torzul, rosszabb esetben gerjedés lép fel. A kaszkád struktúra egyik nagy hátránya, hogy a bemeneti jelet le kell osztani ahhoz, hogy elkerüljük a túlvezérlést. Ez mindenképpen zajosabbá teszi a megszárt jelet. Ez főleg az alacsony frekvenciás tartományban jelentkezik. A másik hátrány, hogy az együtthatók értéke a $[-2...2]$ tartományba esik. Jelfeldolgozó processzor esetén a leggyakoribb számábrázolási

tartomány a $[-1...1]$ intervallum. Ezért az együtthatókat felére kell osztani, majd a sorbakapcsolt tagok végén ezt a leosztást egy kétszeres visszaszorzással korigálni kell. Ez az eljárás tovább rontja az elérhető jel/zaj viszonyt.

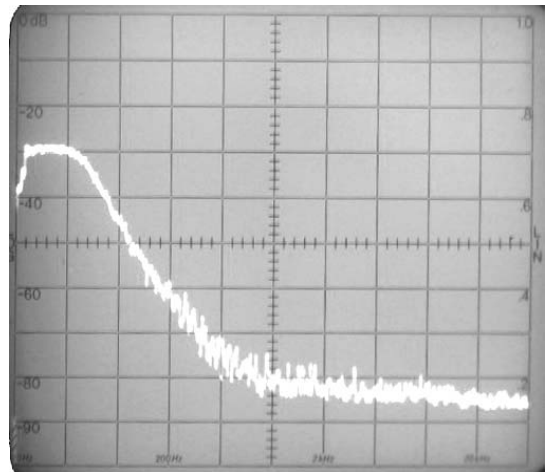
Mindezek az eredmények természetesen mérésekkel is alá lettek támasztva. Először elkészült a kaszkád struktúrát 16 bites számítási pontossággal megvalósító DSP program. Ezzel azonban csak a 200 Hz törésponti frekvencia fölötti szűrőket lehetett megvalósítani. Ennél kisebb törésponti frekvencián működő szűrőknél a bemeneti jelet jelentősen le kellett osztani, hogy a kimeneten ne lépjen fel gerjedés. Ezután elkészült a 32 bites számítási pontossággal dolgozó kaszkád megvalósítású szűrő program. Ez megtalálható a CD mellékleten `kaszkad2.dsp` néven.

Ezzel a számítási pontossággal már jelentősen jobb eredményeket lehetett elérni. Azonban az 50 Hz-es szűrőknél még mindig jelentős leosztást kellett alkalmazni, hogy a kimenet ne gerjedjen be.

A mérések negyedfokú, inverz Csebisev közelítéssel megvalósított 100 Hz-es aluláteresztő szűrő beállításánál készültek, hiszen ennél már érezhetően romlott a kisszintű jelek megszűrése után kapott kimeneti jel minősége. A spektrumanalizátorral felvett frekvenciasáv 0 Hz és 1 kHz közé esett. A 3.12 ábra 2 V csúcsértékű bemenőjelnél, míg a 3.13 ábra ugyanolyan beállítások mellett, 0,4 V csúcsértékű bemenőjel esetén mutatja a szűrő amplitúdóválaszát.



3.1. ábra Amplitúdóválasz 2 V-os bemenőjel esetén (függőlegesen 100 Hz/osztás, vízszintesen 10 dB/osztás)

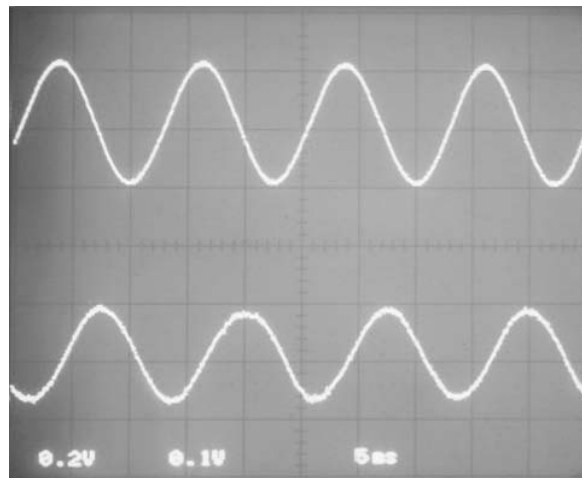


3.2. ábra Amplitúdóválasz 0,4 V-os bemenőjel esetén. (függőlegesen 100 Hz/osztás, vízszintesen 10 dB/osztás)

Az ábrákról jól látszik, hogy míg 2 V csúcsértékű jelnél a szűrő teljesíti az előírt 60 dB elnyomást, addig a 0,4 V csúcsértékű jel esetén már csak 55 dB elnyomás érhető el. Emellett az áteresztő tartományban is megfigyelhető, hogy a második esetben a kimeneti jel jóval zajosabb, azaz az amplitúdó átvitel nem teljesen egyenletes. Ez a fajta lengés oszcilloszkóppal is kimutatható volt (3.14 ábra). Míg rezonátoros megvalósítás esetén (lásd később) a kimeneti jel még ilyen kisszintű bemenet esetén is torzításmentes marad.

A 3.14 ábra 0,4 V csúcsértékű 80 Hz frekvenciájú bejövő jel esetén (felső csatorna) mutatja a szűrő kimenetén mérhető jelalakot (alsó csatorna). A kimeneten közelítőleg

0,18 V csúcsertékű jelet kapunk. Az oszcilloszkóp felső csatornája 0,2 V/osztásra, míg alsó csatornája 0,1 V/osztásra volt beállítva.



3.3. ábra A megszürt jel 0,4 V-os bemenőjel esetén

3.5.3 Rezonátoros megvalósítás

A struktúra alapvetően digitális rezonátorokra épül [4], amelyek tulajdonságaikban gyakorlatilag megegyeznek az analóg technikában megismert rezonátorok tulajdonságaival. A digitális rezonátor egy pozitív visszacsatolt rendszert alkot, ezért erre is jellemző, hogy a rezonátor frekvencián a hurokerősítése végtelenné válik. Ilyen módon a kimenete független lesz a bemenettől, vagyis az adott frekvencián rezonancia lép fel. A rezonátor átviteli függvénye a következő:

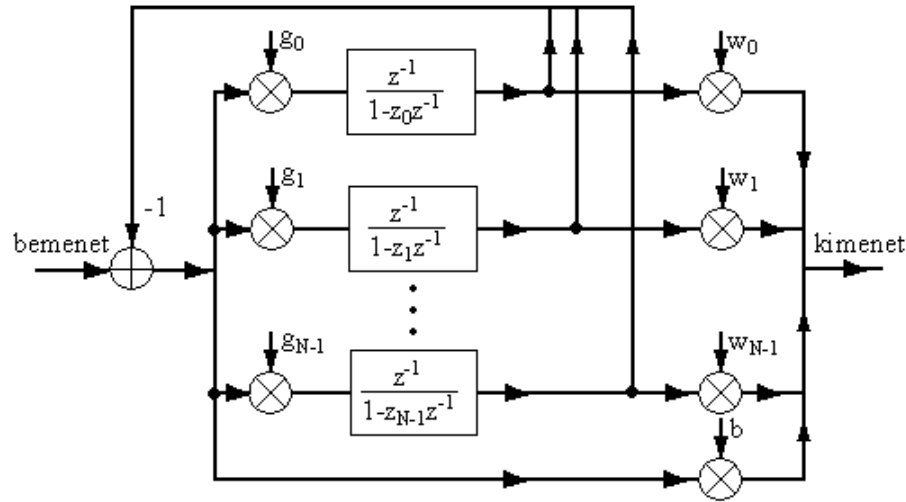
$$H_0(z) = \frac{z^{-1}}{1 - z_0 z^{-1}}, \quad (3.5.10)$$

ahol z_0 a rezonátor komplex együtthatója, és a rezonanciafrekvenciát határozza meg. Mivel a rezonátor együttható abszolút értéke egy, vagyis az egységkörön helyezkedik el, ebből adódik, hogy amikor a bemeneti jel eléri a rezonancia frekvenciát, azaz $z = z_0$ lesz, a $z_0 z^{-1}$ szorzat eggyé válik, és ilyenkor a nevező értéke nulla, tehát a rezonátortag erősítése végtelen lesz.

Több rezonátort párhuzamosan kapcsolva, és ezeket egy közös visszacsatolással ellátva egy olyan rendszert kapunk, amelynek az adott rezonátorfrekvenciákon az átvitele egységnyi, míg a többi helyen nulla. Így felépíthetünk egy olyan szűrőt, amely interpolációs jelleggel közelíti a megkívánt átviteli függvényt, hasonlóan a Lagrange-interpolációhoz. Ez nem más, mint a frekvencia mintavételezési eljárás általánosítása. Mivel ez a módszer alapvetően FIR szűrők megvalósítására alkalmas, módosítani kellett a struktúrát, hogy ne csak véges impulzusválaszú, hanem IIR szűrők realizálására is alkalmas legyen. A 3.15 ábra egy ilyen, rezonátorokra épülő struktúrát mutat be.

A g_n paraméterek helyes megválasztásával érhető el, hogy a beállítás ne csak véges lépésben történjen, így tehát az IIR szűrés lehetősége adottá válik. A g_n szorzótagok szerepe, hogy adott z_n értékek mellett beállítsák az átvitel pólusait.

A w_n szorzótagok tulajdonképpen már nem befolyásolják a hurok működését. Ezek feladata a kívánt karakterisztika előállítása. Vagyis adott g_n és z_n értékek mellett w_n állítja be a zérusokat.



3.1. ábra: Rezonátor alapú szűrőstruktúra

Ahhoz, hogy meg tudjuk határozni az egyes együtthatók értékét, először nézzük meg, hogyan is alakul a 3.15 ábrán megadott elrendezés átviteli függvénye. Az egyes rezonátortagok átvitele a következő:

$$H_n(z) = \frac{g_n z^{-1}}{1 - z_n z^{-1}}, \quad n = 0, 1, \dots, N-1. \quad (3.5.11)$$

Mivel mindegyik rezonátortag egy közös visszacsatolt hurokon belül helyezkedik el, ezért ezek közösen alakítják ki az eredő átviteli függvényt. Felírható tehát a teljes hurok átvitele, egyetlen rezonátor csatornára vonatkoztatva:

$$G_n(z) = \frac{H_n(z)}{1 + \sum_{n=0}^{N-1} H_n(z)}, \quad n = 0, 1, \dots, N-1, \quad (3.5.12)$$

valamint a különbségi hibajelét kicsatoló csatornára vonatkoztatva:

$$G_n(z) = \frac{1}{1 + \sum_{n=0}^{N-1} H_n(z)}, \quad n = 0, 1, \dots, N-1. \quad (3.5.13)$$

A kimenet ezeknek csatornáknak az összegeként áll elő, csatornánként megszorozva a kicsatoló együtthatókkal. A rendszer eredő átvitele tehát, amelynek meg kell egyeznie a megvalósítandó szűrő átvitelével, a következő alakban írható fel:

$$H(z) = \frac{b + \sum_{n=0}^{N-1} w_n H_n(z)}{1 + \sum_{n=0}^{N-1} H_n(z)} = \frac{B(z)}{A(z)}, \quad n = 0, 1, \dots, N-1, \quad (3.5.14)$$

ahol $B(z)$ a megvalósítandó szűrő számláló, míg $A(z)$ a nevező polinomja. Ezek legfeljebb N -edfokú polinomjai lehetnek a z komplex változónak.

A következőkben nézzük végig, hogyan is határozhatók meg a rezonátoros szűrő együtthatói.

Legegyszerűbb a kicsatoló együtthatók meghatározása. A 3.5.14 összefüggés alapján, ha a komplex változó, z helyére nullát helyettesítünk, a különbségi jelet megszorozó b együttható értéke határozható meg. Ez minden esetben egy valós konstans.

$$b = \frac{B(z)}{A(z)}, \quad z = 0. \quad (3.5.15)$$

Továbbá az átviteli függvénynek a rezonátor frekvenciákon meg kell egyeznie az eredeti szűrő átvitelével, amelyből a w_n kicsatoló együtthatók határozhatók meg. Mivel z_n rezonancia frekvencián a $H_n(z)$ értéke tart a végtelenhez, kiemelve a 3.5.14 összefüggés számlálójából és nevezőjéből is $H_n(z)$ értékét, a teljes határérték tartani fog w_n -hez. Tehát az n -edik kicsatoló együttható meghatározható, ha a megtervezett szűrő polinom z komplex változójának helyére z_n -et helyettesítünk:

$$w_n = \frac{B(z)}{A(z)}, \quad z = z_n, \quad n = 0, 1, \dots, N-1. \quad (3.5.16)$$

A rezonátor pozíciók számításánál célszerű figyelembe venni a megtervezett függvény eredeti pólusainak elhelyezkedését. Ezt a választást azért tehetjük meg, mert a rezonátoros struktúra paramétereiben redundáns, így a rezonátorpólus pozíciók elméletileg tetszés szerint felvehetők. Ennek bizonyítása megtalálható a felhasznált irodalomban [4]. Továbbá ahhoz, hogy a rendszer strukturálisan stabil maradjon, vagyis a közösen visszacsatolt hurok átvitele ne legyen nagyobb egynél, a következő feltételeknek kell teljesülnie:

$$|z_n| = 1, \quad \text{Im} \left[\frac{g_n}{z_n} \right] = 0, \quad \sum_{n=0}^{N-1} \text{Re} \left[\frac{g_n}{z_n} \right] = \sum_{n=1}^{N-1} r_n \leq 1, \quad (3.5.17)$$

ahol $n = 0, 1, \dots, N-1$. Tehát csak olyan z_n paraméterek felelnek meg, amelyekkel a fenti feltételeket teljesíteni tudjuk. Egyik ilyen megoldás, ha a rezonátorpólus pozíciók meghatározásához a visszacsatolt hurkot mindentáteresztő hálózatként írjuk fel. Tehát meg kell határozni a

$$\prod_{n=0}^{N-1} (1 - p_n z^{-1}) \pm \prod_{n=0}^{N-1} (z^{-1} - p_n) = 0 \quad (3.5.18)$$

polinom gyökeit, ahol p_n a megtervezett szűrő pólusait jelöli. Ha az eredeti szűrő nevező polinomját a következő alakúnak vesszük,

$$A(z) = (a_0 + a_1z^{-1} + a_2z^{-2} + a_{N-1}z^{N-1}), \quad (3.5.19)$$

akkor a 3.5.18 összefüggés alapján a rezonátor pólusok számítása a nevező polinom, és azok pólusainak az egységkörre tükrözéséből keletkező polinomok összegéből és különbségéből adódó polinom:

$$(a_0 + a_1z^{-1} + a_2z^{-2} + a_{N-1}z^{N-1}) \pm (a_{N-1} + a_{N-2}z^{-1} + a_{N-3}z^{-2} + a_0z^{N-1}) = 0 \quad (3.5.20)$$

gyökeinek a meghatározását jelenti. A 3.5.20 képletben N a polinom fokszámával egyezik meg. Az eredmény két rezonátorpólus készlet lesz, amelyből majd ki kell választani a kedvezőbbet. A rezonátor pólusok ismeretében már meghatározható az r_n együtthatók értéke:

$$r_n = \frac{\prod_{m=0}^{N-1} (1 - p_m z_n^{-1})}{\prod_{m=0, m \neq n}^{N-1} (1 - z_m z_n^{-1})}, \quad n = 0, 1, \dots, N-1, \quad (3.5.21)$$

ahol p_m , $m = 0, 1, \dots, N-1$, a megvalósítandó szűrő pólusai, míg z_m , $m = 0, 1, \dots, N-1$, az m -edik szűrőpólushoz tartozó rezonátorpólus értéke. Természetesen itt is két együtthatókészlet adódik, amelyből a 3.5.17 feltételhez igazodva ki kell választani a megfelelőt, és az ahhoz tartozó z_n pólusokkal együtt ezek alkotják majd a szűrő paramétereit. Az, hogy az r együttható készlet csak valós lehet, a fenti számításokat alkalmazva mindig teljesül.

3.5.4 Összefoglalás

Rezonátorok alkalmazásával, és a kívánt átviteli függvény interpoláció jellegű közelítésével egy nagyon kedvező struktúrához jutottunk. Ez az együtthatók pontatlanságára meglehetősen érzéketlen, amely főleg kevés bitszámon történő számábrázolásnál előnyös, ami annyit jelent, hogy az együtthatók ilyen jellegű megválasztása mellett is biztosítható a strukturális passzivitás. Vagyis a rendszer korlátos bemenetre mindig korlátos kimenetet szolgáltat. Skálázásra ugyan szükség van, de ennek mértéke az adott frekvenciatartományban (50 Hz – 10 kHz) kétszeresnél nem nagyobb. Számítás során a kerekítés helyett célszerű az eredmények csonkolást elvégezni és így eltávolítani azokat, ezzel elkerülhető a rendszer bármiféle határoszcillációja. A megvalósítási forma előnye, hogy az átmeneti tartományban nagyon kedvező tulajdonságokat mutat a rendszer. A kismértékű skálázás következtében az aritmetikai zaj is jóval kevesebb hibát okoz, mint a kaszkád struktúránál. Még igen alacsony frekvenciákon is megmaradnak az átmeneti tartomány kedvező tulajdonságai, vagyis a kis zaj, és a stabilitás. Egyedül a zárótartományban az elnyomásnál lépnek fel hibák, ami abban nyilvánul meg, hogy a maradékjel nem minden esetben csökken a megkívánt érték alá, amely a számítási pontatlanságból adódó aritmetikai zajnak a következménye.

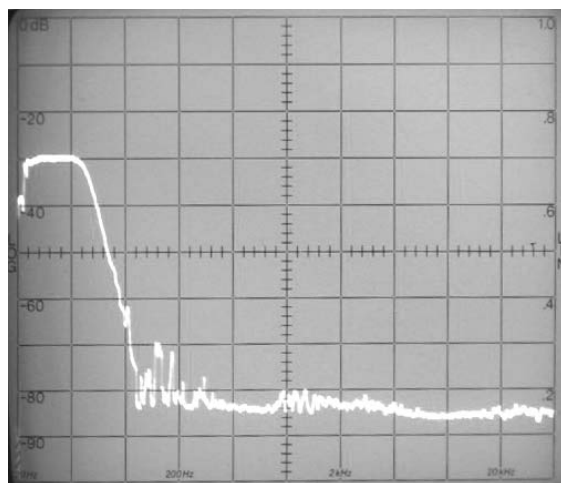
A rezonátoros szűrő együtthatóinak meghatározása meglehetősen bonyolult és számításigényes. Ezért felhasználtam olyan MATLAB programokat, melyek a szükséges együtthatók egyszerű kiszámítását tették lehetővé. Emellett MATLAB kísérletek történtek az így előállított együtthatókkal, hogy azok milyen pontossággal valósítják meg az előírt amplitúdókarakterisztikát és meg lett vizsgálva, hogy az

együtthatók csonkolása milyen hatással van az átviteli függvényre. Az ehhez felhasznált programok egy korábbi diplomaterv [3] keretében lettek elkészítve. A kísérletek eredményeként kiadódott, hogy 16 bites együttható pontossággal a karakterisztikák még viszonylag kis hibával megvalósíthatók. Persze hangsúlyozni kell, hogy a MATLAB belső számítási pontossága jóval nagyobb, mint a jelfeldolgozó processzoré.

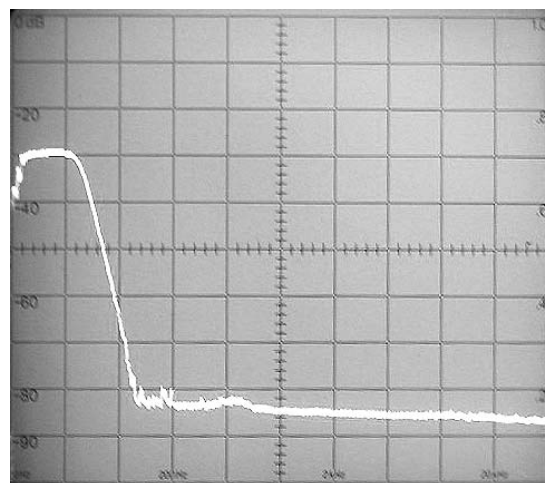
Megvalósítás szempontjából a komplex műveletek elvégzése okozhat némi nehézséget. Ez a programozás szempontjából nagyobb odafigyelést igényel. A komplex műveletek és a nagy számú együttható következtében a program hosszabbá, míg a számítás lassabbá válik. Ez annyiból jelent korlátot, hogy a rendelkezésre álló ciklusidőben csak alacsonyabb fokszámú szűrők képesek futni, mint kaskád megvalósítás esetén.

Természetesen az eredmények itt is mérésekkel lettek alátámasztva. Ehhez először elkészült a rezonátoros szűrőt megvalósító programrutin 16 bites számítási pontossággal működő változata. A mérési eredmények alapján ez a változat már alkalmas volt a teljes hangfrekvenciás sávban, az összes beállítható törésponti frekvencián a szűrők megvalósítására.

Az áteresztő sávban a szűrés utáni jelalak teljesen torzítás és zajmentes volt. Azonban a zárósávban, kisszintű bemenő jelek esetén már fellépett valamekkora hiba. Ez alacsonyfrekvencián működő szűrők esetén volt tapasztalható. A zárósávban ekkor az elnyomás értéke alig haladta meg az 50 dB-t (3.16 ábra). Ezért elkészült a rezonátoros szűrő 32 bites számítási pontossággal dolgozó változata. Ennél az utóbbi hiba, vagyis az elnyomás mértéke valamelyest javult (3.17 ábra) és az átmeneti tartomány után közvetlenül tapasztalható hiba is megszűnt. Tehát megállapítható, hogy rezonátoros megvalósítás esetében is szükséges a 32 bites számítási pontosság alkalmazása. Történt még egy kísérlet, amelyben az együtthatók is duplapontosan lettek letárolva és a számításokhoz is 32 bites formában lettek felhasználva, de ez a megoldás érdemi változást nem hozott, csak a programkódot tette feleslegesen bonyolulttá és a számítási időt hosszabbá.



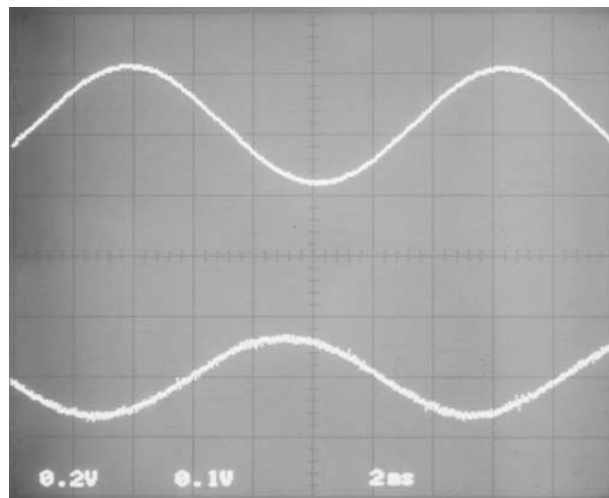
3.1. ábra Amplitúdóválasz 16 bites számítási pontosság esetén. (függőlegesen 100 Hz/osztás, vízszintesen 10 dB/osztás)



3.2. ábra Amplitúdóválasz 32 bites számítási pontosság esetén. (függőlegesen 100 Hz/osztás, vízszintesen 10 dB/osztás)

A spektrumanalizátorral felvett frekvenciasáv 0 Hz és 1 kHz közé esett. A szűrő egy hatodfokú, inverz Csebisev közelítéssel megvalósított, 100 Hz törésponti frekvenciájú aluláteresztő szűrő. A bemenetre 0,4 V csúcsertékű jel volt kapcsolva.

A 3.18 ábra a megszürt jelet mutatja. Ezen jól látható, hogy a kimenő jel (alsó csatorna) még ilyen kisszintű bemenet esetén is torzítás és zajmenetes. Ez a kedvező tulajdonság a teljes frekvenciasávban megmarad.



3.3. ábra A megszürt jel 0,4 V-os bemenőjel esetén

A bemeneten (felső csatorna) 0,4 V csúcsértékű 80 Hz-es jel van. Az oszcilloszkóp beállítása ezen a csatornán 0,2 V/osztás. A kimenő jel nagysága (alsó csatorna) ekkor megközelítőleg 0,18 V. Ez a csatorna 0,1 V/osztás állásban volt.

3.6 Következtetés

A tervezés tehát eljutott arra a szintre, hogy el lehessen kezdeni a szűrőprogram fejlesztését a jelfeldolgozó kártyára. Azt azonban még el kellett dönteni, hogy a szűrő magját alkotó, a szűrést elvégző rutin, kaszkád vagy rezonátoros struktúrával legyen-e megvalósítva.

Kaszkád esetben az eredmények azt mutatták, hogy 16 bites számítási pontosság mellett, az alacsony frekvenciás szűrők már nem valósíthatók meg. Ez egyértelműen az együttható érzékenység következménye. 32 bites együttható pontosság és duplapontos számítás mellett az eredmények már jóval kedvezőbbek lettek. Ezek alapján látható, hogy elég nagy hibával ugyan, de az alacsony frekvenciatartományban működő szűrők is megvalósíthatók.

A mérések tehát azt bizonyították, hogy a 32 bites együttható pontosság, és a 32 bites számítási pontosság az 50 Hz törésponti frekvenciájú alul és felüláteresztő szűrők megvalósításához már elegendők. Azonban itt a bemenő jel igen nagy leosztása miatt a jel/zaj viszony erősen leromlott, ezért a szűrő már nem teljesítette a 60 dB elnyomást, másrészt az áteresztőtartományban a jel már a 200 Hz törésponti frekvencia alatt működő szűrők alkalmazása után meglehetősen zajossá vált. Ez főleg kisszintű jelek esetén volt tapasztalható. Ez egyértelműen az aritmetikai zajnak tudható be.

Az aritmetikai zaj meghatározásánál abból indulhatunk ki, másodfokú alaptagot feltételezve (3.11 ábra), hogy a bemeneti minta pontos, a számított $2N$ bites szorzat és összegzés pontos, csak az eredmények N bitre történő kvantálásakor keletkezik hiba. A hiba ($e(n)$) nagysága ekkor [2]:

$$-\frac{q}{2} \leq e(n) < \frac{q}{2}, \quad (3.6.1)$$

ahol q az N bit legalsó helyiértéke. $N = 16$ bit esetén $q = 2^{-15}$. Véletlen változó jelet feltételezve, amely korrelálatlan, független a bemenő jeltől és egyenletes eloszlású, azt mondhatjuk, hogy a hiba a keletkezés helyén fehér zajnak tekinthető, amelynek a spektrális sűrűségfüggvénye:

$$S_{ee}(\omega) = \frac{q^2}{12}, \quad (3.6.2)$$

konstans a teljes frekvencia tartományon. A visszacsatolás hatására azonban ez a hiba felsokszorozódik. A többszöri kerekítés és visszacsatolás következtében a 3.6.2 alapján számítható hiba többszöröse kerül a kimenetre. A kimeneti zaj teljesítménye, másodfokú alaptagot feltételezve:

$$\sigma_f^2 = \frac{q^2}{12} \frac{1+a_2}{1-a_2} \frac{1}{(1+a_2)^2 - a_1^2}. \quad (3.6.3)$$

A fenti kifejezésben a_1 és a_2 a kaszkád tag együtthatóit jelöli. Az együtthatók és a nevező zérusai között a következő összefüggés van:

$$a_1 = -(p_1 + p_2), \quad a_2 = p_1 p_2. \quad (3.6.4)$$

A 3.6.3 és a 3.6.4 összefüggések alapján nyilvánvalóvá válik, hogy minél közelebb esnek a pólusok az egységkör széléhez, a kimeneti zaj teljesítménye annál jobban megnő. Például egy hatodfokú, 50 Hz-ig aluláteresztő szűrő első kaszkád fokozatának a pólusai a z síkon a

$$0.99079564544472 + 0.00300851037399i$$

$$0.99079564544472 - 0.00300851037399i$$

pozíciókban helyezkednek el. Alkalmazva a 3.6.3 összefüggést, a kimeneti zaj teljesítménye:

$$\sigma_f^2 = \frac{q^2}{12} 291146,03 \text{ -re adódik,}$$

amely 16 bites számábrázolás esetén $-46,5$ dB, míg 32 bites számábrázolást feltételezve már csak $-142,8$ dB-re adódik az aritmetikai zaj teljesítménye. Itt azonban figyelembe kell még venni, hogy a további fokozatok nem csak a hasznos jelet, hanem ezt a zajt is felerősítik, és az így megtöbbszörözve adódik hozzá a kimeneti jelhez.

A mérések eredményére hagyatkozva azt lehetett megállapítani, hogy ilyen széles frekvenciatartományban a kaszkád realizáció nem biztosít megfelelően pontos átvitelt. A specifikáció enyhítésével az elnyomás értékét még be lehetett volna állítani egy megfelelő érték alá, de az áteresztő sáv zajossága nagyon kedvezőtlen az átvitel szempontjából.

4 Megvalósítás

Ebben a fejezetben az elkészített szűrőprogram felépítése és működése lesz ismertetve. A program megírása gépi kódban történt. Ennek oka, hogy a fejlesztőrendszer nem tartalmaz olyan fordítót, amely magasabb szintű nyelven megírt programot a jelfeldolgozó processzor „nyelvére” le tudna fordítani. Ennek természetesen van előnye és hátránya is. A hátrány az, hogy elveszítjük a magas szintű programnyelvek kínálta kényelmi megoldásokat. Az előny viszont a gyorsabb működés, ami általában követelmény a jelfeldolgozó rendszerekkel szemben. Gépi kódban ugyanis jobban kihasználhatók a párhuzamos utasítás-végrehajtás előnyei. A programkód is sokkal tömörebbé válik, ami egyértelműen gyorsabb működést eredményez. Bár, mint az előző fejezet végén kiderült, a rezonátoros szűrők alkalmazásával már nem valósítható meg a kétcsatornás feldolgozás. Az egycsatornás szűréshez viszont a rendelkezésre álló két mintavétel közti ciklusidő bőven elegendő. Ennek ellenére igyekeztem a program szempontjából kritikus részt, ami maga a szűrést végrehajtó rutin, minél tömörebbé és gyorsabbá tenni.

A DSP program `szuro32.dsp` mellett elkészült egy MATLAB program is `rezonator_ehk2.m`, amely a szűrőegyütthetők kiszámítását végzi el, és azokat olyan formában tárolja, amit a DSP programba fordítási időben könnyen be lehet illeszteni. Ezek a programok megtalálhatók a CD mellékleten.

A megvalósításnál figyelembe kellett még venni a rendelkezésre álló hardver környezetet. Ez természetesen némi kötöttséget jelentett a tervezésnél. Ezért a következő fejezetben bemutatásra kerül a fejlesztőkártya és annak felépítése.

4.1 A jelfeldolgozó kártya felépítése

A jelfeldolgozó kártya egy ADSP-2181-es processzorra épül [5]. A processzor 16 bites számítási pontossággal rendelkezik, mivel az adatmemóriája és a belső adatbusz is 16 bites kialakítású. A számítások végrehajtása valamint az adatok tárolása fixpontos formátumban történik. A programmemória 24 bit széles, amely szintén alkalmas adatok tárolására. Ezek tipikusan valamilyen szűrőegyütthetők, amelyek statikusan, fordítási időben kerülnek eltárolásra. Adat beolvasáskor a 24 bitnek csak a felső 16 bitje kerül felhasználásra. A processzor belső felépítése olyan, hogy az adatmozgatás és a műveletvégzés párhuzamosan is történhet, így az összetettebb és legtöbbet használt utasítások, ilyen például a szorzás és összegzés, egy lépésben végrehajthatók. Tipikusan egy ilyen párhuzamosan végrehajtható műveletet a következő utasítás sorozat:

```
mr = mr + mx0 * my0, mx0 = DM(i2, m2), my0 = PM(i4, m4);
```

amely az adatmemóriából és a programmemóriából már beolvasott előző értékeket összeszorozza, és a szorzat eredményét hozzáadja a korábbi szorzat eredményéhez, majd beolvassa az újabb két adatot. Mindez egy utasításciklus alatt hajtódik végre.

A processzor tehát tartalmaz egy szorzó egységet, amelynek négy darab 16 bites bemeneti regisztere van: MX0, MX1, MY0, MY1 és egy darab eredmény regisztere: MR. Az MR regiszter három részből áll, ahol MR0 a szorzat alsó, MR1 pedig a felső 16

bitje, továbbá van még egy 8 bites része: MR2, amelyben az összegzések következtében fellépő túlsordulások kerülnek. Így egy cikluson belül legalább 256 darab MAC (szorzás és összegzés) utasítás hajtható végre, anélkül, hogy az eredmény túlsordulna.

Az összegzést és a logikai műveleteket az aritmetikai egység végzi. Ennek szintén négy darab 16 bites bemeneti regisztere van: AX0, AX1, AY0, AY1 és egy darab 16 bites kimeneti regisztere az eredmény tárolására: AR. Az aritmetikai és logikai műveletek szintén végezhetők az adatmozgatással párhuzamosan.

A regiszterek tartalmának léptetése a shift regiszter segítségével oldható meg. Ennek a bemeneti regisztere: SI 16 bites, míg az eredmény 32 biten kerül tárolásra az SR regiszterben. Az SR regiszter alsó 16 bitjét SR0, míg felső 16 bitjét SR1 regiszter tartalmazza. Az eltolás mértéke jobbra illetve balra egyaránt 32 lépés lehet, amelyet a shifter az SE regiszterből olvas ki. Pozitív érték esetén a léptetés balra, míg negatív értéknél jobbra történik. Bár az SE regiszter 8 bites, ami ± 127 lépést tesz lehetővé, de 32-nél nagyobb eltolás érték esetén az adat teljesen kikerül az SR regiszterből. Az utasításnak meg kell jelölnie az eredmény regisztert is, erre szolgál a (HI) illetve (LO) referencia, amely megadja, hogy az eredmény az SR regiszter alsó vagy felső 16 bitjébe kerüljön.

A processzor ezen kívül két soros vonalat képes kezelni, amellyel a külső kommunikáció oldható meg, valamint rendelkezik egy belső időzítővel is. A soros portok közül az egyik (SPORT1), egy számítógép soros portjával kapcsolódva teszi lehetővé a programok letöltését, míg a másik (SPORT0) az analóg-digitál átalakítóval kommunikál.

A processzor bekapcsolás utáni automatikus felkonfigurálása a kártyán található EPROM-ról történik. Ebbe jelenleg az eredeti gyári bemutatóprogram van beégetve. Azonban az EPROM cseréjével lehetőség nyílik saját program betöltésére is. A kártyán ennek helye úgy van kiépítve, hogy 32 kbyte-tól 1 Mbyte-ig lehessen rajta EPROM-ot elhelyezni.

Mindezek mellett a processzor 33 MHz-es utasítás-végrehajtási gyakoriságra képes, ami első közelítésben meglehetősen gyorsnak tűnik. A szűrésnek azonban két beérkező minta között le kell zajlania. Hangfrekvenciás tartományt nézve a minták 1/44100 Hz gyakorisággal következnek, ami azt jelenti, hogy megközelítőleg 750 utasítást lehet végrehajtani két beérkező minta között.

Az analóg interfészt egy AD1847-es sztereó delta-sigma Codec képviseli [6]. Ezzel a jelfeldolgozó kártya a hangfrekvenciás tartományban képes az adatok feldolgozására. Az AD1847 típusú Codec legmagasabb mintavételi frekvenciája 48 kHz lehet, az alacsony frekvenciás tartományban pedig egy RC-tag gondoskodik a 20 Hz-nél alacsonyabb frekvenciájú komponensek levágásáról. A legnagyobb megengedhető jel a line-in bemeneten csúcsértékben 4,5 V. Efölött az AD átalakító telítésbe kerül, és csenkolja a bejövő jelet. A kimenő jel nagysága a DA kimenetén csúcsértékben 2 V lehet. Az AD1847-es Codec felbontása 16 bit, amellyel elméletileg 96 decibeles jel/zaj viszonyt lehetne elérni.

A jelfeldolgozó kártyán a könnyebb bemérés és a működés ellenőrzésének céljából minden processzorláb ki van vezetve. Továbbá ki van építve egy nyomógomb, amely a processzor külső megszakítást kérő lábára van rákötve, így a processzor működése ezzel

a gombbal befolyásolhatóvá válik. A belső állapotok kijelzésére egyetlen LED-et használhatunk, amelyet programból lehet vezérelni. A vezérlési és kijelzési lehetőségek tehát meglehetősen szerények. Ahhoz, hogy a szűrőket a későbbiek folyamán könnyebben lehessen átállítani, kiegészítésként egy egyszerű kezelőfelületet kellett a fejlesztőkártyához illeszteni.

A hardver környezet tehát adott volt, így a programot úgy kellett fejleszteni, hogy az igazodjon a jelfeldolgozó kártya lehetőségeihez. Mivel ezzel a processzorral már többen dolgoztak, ezért rendelkezésre állt néhány olyan programrészlet, amelyet felhasználtam a saját programom fejlesztéséhez. Így a processzor konfigurálását végző rutin, és az analóg-digitál átalakítóval történő kommunikáció már adott volt. Ezeket változtatás nélkül használtam fel a programomhoz. Az AD1847-es analóg interfész a processzor szabad soros vonalán SPORT0 keresztül kommunikál, küld, illetve fogad adatokat. Ehhez a

```
right_in( ); left_in( );
```

illetve a

```
right_out( ); left_out( );
```

parancsokat kellett megadni, amelyek meghívták az adat kiírást, illetve olvasást végrehajtó rutint. A programrészlet a kommunikáció lebonyolítása mellett jelszint illesztést is végez, mivel az AD és DA átalakítók érzékenysége nem egyforma. A mérést 1 kHz-en elvégezve 1,4 V-os bemeneti jel mellett a kimeneten 0,4789 V nagyságú jelet mérhetünk. Megállapítható tehát, hogy a bejövő jel 2,92-ed részére van leosztva. Ez a teljes tartományban állandónak tekinthető.

4.2 Szűrőegyütthetők előállítása

A DSP program futásához szükség van a megfelelő szűrőegyütthetők előállítására. Ezért el kellett készíteni egy olyan MATLAB programot, amely az inverz Csebisev közelítéssel meghatározott szűrőparamétereket átszámítja a rezonátoros szűrők megvalósításához szükséges együtthetőká. Ezután az együtthetőket egy adatfájlba menti, amely ezután biztosítja a DSP program számára a fordítási időben a gyors elérést. A MATLAB program írásánál nagy könnyedséget jelentett, hogy egy korábbi diplomaterv [3] is foglalkozott már a rezonátoros megvalósítás lehetőségeivel, és ennek végeredményeként elkészült egy hasonló program, amely a rezonátoros struktúra együtthetőit számítja ki. Ezt a programot használtam fel kiindulásként a paraméterek számításához. Természetese az én esetemben nem egyetlen szűrő együtthetőit kellett meghatározni, hanem együtthető csoportokat. A megvalósításnál a következőkből indultam ki. Célszerűnek látszott az együtthetőket külön kezelni, azaz minden egyes együtthetőnek egy külön adatfájl létrehozni. Ez a DSP programban is így lett megvalósítva, amelynek eredményeként a beolvasott adatfájl tartalmára az adott együtthető nevével lehet hivatkozni. A másik elgondolás, hogy az együtthetők, mint adatvektor kerüljenek eltárolásra, vagyis a különböző frekvencia beállításokhoz és szűrőtípusokhoz tartozó paraméterek egymás után, sorban legyenek elhelyezve. A DSP program pedig majd gondoskodik róla, hogy mindig az aktuális együtthető csoport legyen kiválasztva a szűrők számításakor.

A `rezonator_ehk2.m` program első fele a `CHEBY2()` függvény által kiszámított együtthatókat alakítja át rezonátoros szűrőegyütthatókká. Ehhez felhasználja a törésponti frekvenciákat tartalmazó vektort. A vektorban sorban a negyedfokú majd a hatodfokú szűrőkhöz tartozó relatív törésponti frekvencia értékek vannak. A törésponti frekvenciákat tartalmazó vektor értékeit a `szuro_IIR.m` programmal ellenőriztem. Ez a program direkt struktúrát feltételezve kirajzolja a megtervezett szűrő amplitúdó átviteli függvényét, amelynek segítségével grafikusan ellenőrizhető, teljesíti-e a megszabott specifikációt.

A `rezonator_ehk2.m` program ezután meghatározza a rezonátor alapú szűrő paramétereit. Ehhez a 3.5.3 fejezet összefüggéseit használja fel (3.5.15, 3.5.15, 3.5.17, 3.5.20 és a 3.5.21). Mivel a szűrők páros fokszámúak, ezért a `CHEBY2()` függvény által generált polinom számlálója és nevezője azonos fokú:

$$\frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_{N-1} z^{-(N-1)}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{N-1} z^{-(N-1)}}, \quad (4.2.1)$$

ahol N megegyezik a szűrő fokszámával. A különbségi jelet kicsatoló együttható b ebben az esetben megegyezik a számláló polinom b_0 értékével. Mivel a 3.5.20 összefüggés értelmében két z és ennek következtében két r és w paraméterkészlet adódik, a 3.5.17 feltételt alkalmazva, eldönthető melyik paraméterkészlet biztosítja a helyes működést. Az együtthatók száma azonban még így is redundáns, mivel a DSP program a komplex konjugált párok egyik tagját használja fel a szűrők megvalósításához. Ez szemléletesen annyit jelent, hogy azokat a rezonátorpólusokat kell kiválasztani, amelyek a komplex sík felső felén helyezkednek el. Azaz amelyek képzetes része pozitív. Az így kiválasztott együtthatók a program végén az együtthatóval azonos nevű fájlban lesznek eltárolva. Mivel a program annyiszor fut le, ahány szűrőtípus és azon belül ahány frekvenciaállítási lehetőség van, az együtthatók egymás utáni csoportokban íródnak ki a megfelelő adatfájlba. Végül a konstansok előállítására történik, amelyre a DSP programnak van szüksége, az együttható vektorokban történő lépéskészítéshez, frekvencia illetve szűrőtípus váltás esetén.

Érdeemes még néhány szót szólni arról, milyen formátumban kell az adatokat eltárolni ahhoz, hogy a jelfeldolgozó processzor helyesen tudja értelmezni azt a számítások elvégzésekor. Ugyanis a programban a felhasználás helyén dől el, hogy milyen formátumban kezeli az adott számot. A `szuro32.dsp` program végig előjeles törtszámábrázolást használ. A számábrázolási tartomány a $[-1; 1-2^{-15}]$ intervallum, tizenhat bites felbontás esetén. Szerencsére a rezonátoros szűrő paramétereit belül vannak ezen a tartományon, így nem kellett az együtthatókat leosztani. Ahhoz azonban, hogy a processzor helyesen értelmezze, mindenképpen módosítani kell azokat. A program illetve adatmemóriába csak előjeles egész értékek írhatók. A fordító ugyanis csak ezt a formátumot fogadja el. A decimálisan megadott értéket át kell számítani hexadecimálissá, de úgy, hogy a negatív számot kettes komplementes kódban tárolja. Ha adatmemóriába kívánunk elhelyezni együtthatót, az adott számot csak meg kell szorozni 2^{15} -nel, majd az egészrészt, mint felhasználható adatot, kell elmenteni. Ekkor az előjeles számok is helyesen lesznek eltárolva. A programmemóriába szánt adatoknál figyelembe kell venni, hogy műveletvégzésnél csak a felső tizenhat bit kerül kiolvasásra. Negatív számnál ügyelni kell arra, hogy a kettes komplementes képzéskor a kiolvasott tizenhat bites érték helyes legyen. Ehhez is először 2^{15} -nel kell megszorozni az együtthatót, majd a törtrész csonkolása után még meg kell szorozni 2^8 -nal. Így az utolsó nyolc bit nulla

marad. Kettes komplement képzéskor, amely úgy állítható elő, hogy az egyes komplementhez hozzáadunk egyet, a hozzáadott bit a felső tizenhat bitet fogja módosítani, így valóban helyesen lesz tárolva az adat. A következő programrészlet az együtthatók ilyen módon való számítására és fájlba mentésére mutat példát.

```

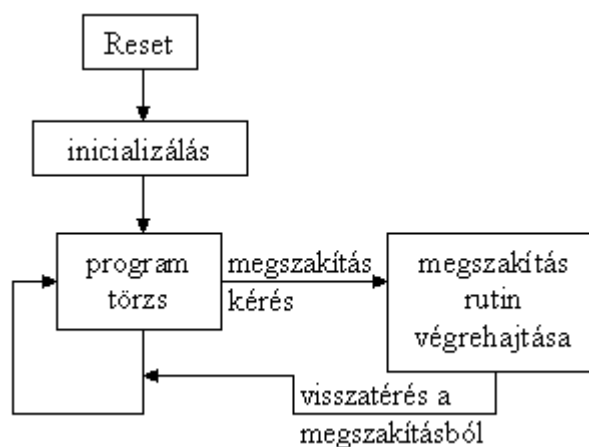
*****változók, a DSP által használható formátumú, beállítása*****
%DM 16 bites PM 24 bites
    rn= (fix(rr*2^15))*2^8;           %rn a PM-ban (mindig pozitív)
    ReW= (fix(real(ww)*2^15))*2^8;   %ReW a PM-ban lesz tárolva
    ImW= (fix(imag(ww)*2^15))*2^8;   %ImW a PM-ban lesz tárolva
    ReZ= (fix(real(zz)*2^15))*2^8;   %ReZ a PM-ban lesz tárolva
    ImZ= (fix(imag(zz)*2^15))*2^8;   %ImZ a PM-ban lesz tárolva
    b= (fix(bb*2^15))*2^8;           %b a PM-ban lesz tárolva

*****mentés fájlba*****
%formátum: a szám 8 helyiértéket foglal, plusz az el jel
%minden együttható után vessz , az utolsó után pontosvessz
for i= 1: N/2-1
    fprintf(fn1, '%8d, \r\n', rn(i)); %nincs el jel
    fprintf(fn2, '%9d, \r\n', ReW(i)); %van el jel
    fprintf(fn3, '%9d, \r\n', ImW(i));
    fprintf(fn4, '%9d, \r\n', ReZ(i));
    fprintf(fn5, '%8d, \r\n', ImZ(i));
end

```

4.3 A program ismertetése

Most nézzük tehát a szűrő programjának felépítését. A jelfeldolgozó processzoroknál egy általános programszervezési struktúra látható a 4.1 ábrán. Ez a logikus felépítés annak köszönhető, hogy a programrutinok mindig csak egy újabb adat beérkezésekor szolgáltatnak az előzőtől eltérő eredményt. Egy újabb adat pedig mindig valamilyen megszakításkérés következtében áll elő. Ezért a program magját alkotó függvény önmagára mutat vissza, vagyis egy végtelen ciklusban fut. Ebből a ciklusból csak megszakítás hatására lép ki. Ekkor a megszakítás típusától függően átadódik a vezérlés a szükséges programrutinra, majd ennek végrehajtása után, ismét visszatér a program törzsbe.



4.1. ábra A program egyszerű felépítése

Bekapcsolás után a processzor egy úgynevezett reset állapotba kerül. Ennek hatására a státusz regiszterek és a vezérlő regiszterek egy előre definiált értéket vesznek fel. Az adatregiszterek és címregiszterek értéke nem meghatározott.

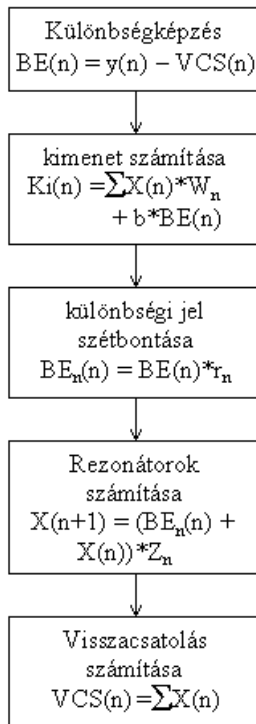
Reset után a programszámláló az első utasítás címére mutat. A program az inicializálással, azaz a kezdeti értékek beállításával kezdődik. Ekkor az összes megszakítást tiltani kell. Itt történik a helyfoglalás a felhasznált változóknak és konstansoknak. A konstansok kezdeti értékét is itt kell meghatározni. A programban ez egy `.include` utasítással történik, amely egy külső fájlból, annak tartalmát illeszti a program megadott helyére. Ebben vannak a szükséges konstansok, amelyeket a már említett MATLAB program generál. Ezután történik a helyfoglalás mind az adat, mind a programmemóriában a felhasznált együttható vektorok és változók számára. Az együttható vektorok cirkulárisan vannak címezve. A következő lépés a szűrő-együtthatók programmemóriába illesztése. Ez szintén az `.include` utasítással történik. Ezeket az adatfájlokat is a MATLAB program állítja elő. A változók egyik része a részeredmények tárolására szolgál, míg másik részük a kiválasztott, és éppen futó szűrő együtthatóinak kezdőcímére mutat az együttható vektorban. Be kell még állítani a státusz és vezérlőregisztereket is, valamint engedélyezni kell a soros vonal működését, amelyen keresztül a processzor az analóg interfésszel kommunikál. Ezek már előre megírt programrészletek formájában szintén az `.include` utasítással vannak a programhoz kapcsolva. A programrészletek a tanszéken dolgozó hallgatók korábbi munkáinak eredménye. Szükséges még a mintavételi frekvencia kiválasztása. Ez táblázatból történik, mivel az AD1847 csak ezeket a beállításokat támogatja. Igazodva a tervezés során megszabott 20 Hz - 20 kHz-es tartományhoz, a mintavételi frekvencia 44,1 kHz-re lett választva. A cirkuláris bufferek címregisztereinek inicializálása után a változók kezdeti értékeinek beállítása, és a részeredmények nullázása történik. Ez a folyamat a `szuro32.dsp` programon jól követhető. Miután az inicializálás lezajlott, a megszakításokat újra engedélyezni kell. Ezt követően az utasítás számláló a program törzsébe ugrik, amely végtelen ciklusban várakozik a megszakításra.

A programban normál működése közben kétféle megszakítás lehetséges (2.2 ábra). Az egyik az AD átalakító felől jön, amely jelzi, hogy lezajlott a konverzió és az új adat kiolvasható a soros vonalon keresztül. Ekkor a szűrést végző rutin hajtódik végre. A másik megszakítás kérés a processzor `IRQE` lábára adott lefutó él hatására történik. Ezt a kártyára szerelt nyomógombbal lehet kiváltani. Ekkor a megszakítást kiszolgáló rutin a kezelőfelület állapotának megfelelően beállítja a szűrőtípust és a törésponti frekvenciát. Most nézzük meg, hogyan is működik ez a két programrutin.

4.3.1 A szűrést megvalósító rutin

A harmadik fejezet eredményei alapján megállapítható volt, hogy rezonátoros struktúrával lehetséges a legjobb tulajdonságú szűrőket megvalósítani. A 3.15 ábrán látható struktúra a következő részekre bontható: különbségi jel képzése a bemeneti jel és a visszacsatolt jelből, kimeneti jel számítása a kicsatoló együtthatók segítségével, különbségi jel szétbontása a rezonátorok számára és a megfelelő előosztás elvégzése, a rezonátorok számítása, és végül a visszacsatolandó jel számítása. Ez a fajta felbontás (lásd 4.2 ábra) a programot jól áttekinthetővé és könnyen értelmezhetővé teszi.

Az ábrán látható együttható és változó jelölések megegyeznek a programban használt elnevezésekkel. Természetesen a komplex tagok külön vannak bontva valós és képzetes részre.



4.1. ábra A rezonátoros struktúra megvalósítása

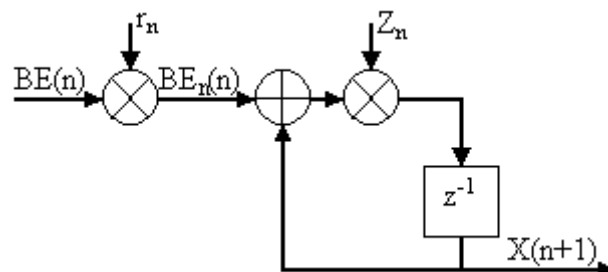
A program magját a rezonátorok képezik. Ezek átviteli függvénye a 3.5.10 összefüggés alapján:

$$H(z) = \frac{z^{-1}}{1 - z_n z^{-1}}, \quad N = 0, 1, \dots, N-1$$

alakú, ahol z_n a rezonancia frekvenciát határozza meg. Minden egyes rezonátor tag előtt található még egy g_n szorzótag (3.15 ábra) is. Ha a g_n szorzótag helyett az

$$r_n = \frac{g_n}{z_n}$$

együtthatót használjuk, amelynek kiszámítása a 3.5.21 összefüggés alapján történik, akkor a rezonátortagok a 4.3 ábra szerinti struktúrával valósíthatók meg.



4.2. ábra Rezonátortag megvalósítása

Egy rezonátortag bemenete és kimenete között, a program jelöléseit használva, az n -edik időpillanatban a következő kapcsolat áll fenn:

$$X(n+1) = (BE(n) * r_n + X(n)) * Z_n, \quad (4.3.1)$$

ahol $BE(n)$ a különbségi jel az n -edik időpillanatban. Ezt megszorozva az n -edik rezonátortaghoz tartozó r_n együtthatóval, előáll $BE_n(n)$ értéke. Z_n a rezonancia frekvenciát meghatározó együttható, kiszámítása a 3.5.18 formula alapján lehetséges. A rezonátorok számítása a programban a következő összefüggés alapján történik:

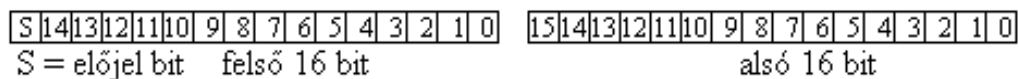
$$X(n+1) = (BE_n(n) + X(n)) * Z_n. \quad (4.3.2)$$

A nehézséget az okozza, hogy bár a $BE_n(n)$ értékei valósak, a rezonátor együtthatók viszont komplexek. Így minden egyes rezonátortag kimenetén komplex eredmény keletkezik. A rezonátor hurkon belül tehát a számításokat komplex formában kell végezni. Ehhez fel kell bontani Z_n és $X(n)$ értékét valós és képzetes részre:

$$ReX(n+1) = (BE_n(n) + ReX(n)) * ReZ_n - ImX(n) * ImZ_n,$$

$$ImX(n+1) = (BE_n(n) + ReX(n)) * ImZ_n - ImX(n) * ReZ_n. \quad (4.3.3)$$

A rezonátorokat számító rész tehát a 4.3.3 összefüggései alapján generálja a következő kimeneti mintát komplex formában. Ez jól követhető a programkódban is. Mivel a harmadik fejezet végén megállapítást nyert, hogy a 16 bites számítási pontosság nem szolgáltatót kellően pontos eredményt, ezért a 4.3.3 összefüggésekben megadott műveleteket duplapontosan kellett elvégezni. Az együtthatók értékei, mivel a struktúra nagy előnye, hogy ezek érzékenysége kicsi, továbbra is 16 bites formában lesznek tárolva és felhasználva. A részeredmények tárolása és felhasználása a műveletek végzésekor 32 bites formában történik. Ehhez az eredményeket a kettes komplement kódoknak megfelelően a 4.4 ábra alapján kellett eltárolni.



4.3. ábra 32 bites eredmény tárolása

A 32 bites eredmények tehát két részből állnak: a felső 16 bitből, amelynek a legfelső bitje az előjelbit, valamint a törtrész maradék részéből, amely az alsó 16 bitet foglalja el. A szám ábrázolása tehát valójában 31 biten történik. A szorzásnál a műveleteket a legkisebb helyiérték felől kell kezdeni. Ez a két alsó 16 bit összeszorozását jelenti. Mivel itt nincs előjel meghatározva, ezért a szorzást is előjel nélkül kell végezni. A szorzat eredményének felső 16 bitjét hozzá kell adni a következő helyiértékek szorzatának eredményéhez. Ebben az esetben az egyik tag felső és a másik tag alsó 16 bitjének szorzását jelenti. Ezt el kell végezni fordítva is, az első tag alsó 16 bitjét megszorozni a második tag felső 16 bitjével. Az így kapott eredményt is hozzá kell adni az előző részeredményhez. Itt arra kell ügyelni, hogy a felső 16 bit már előjeles, és ezt a szorzásnál jelezni is kell. Erre szolgál az (u) -előjel nélküli, illetve az (s) -előjeles szorzás, kapcsoló. Végül a kapott eredményt hozzá kell adni a két felső 16 bit szorzatának eredményéhez. Ezt az eljárást követve az eredmény a szorzat regiszterben (MR) megfelel a 4.4 ábrán bemutatott formátumnak. Az elmondottakra nézzünk egy általános példát:

```

mx0= DM(i2,m2), my0= PM(i4,m4);      {alsó rész betöltése}
mx1= DM(i2,m2), my1= PM(i4,m4);      {fels  rész betöltése}
mr= mx0*my0(uu);                      {alsó 16 bit * alsó 16 bit}
mr0= mr1;                              {szorzat fels  16 bitje}
mr1= mr2;                              {bekerül mr alsó 16 bitjébe}
mr= mr+mx1*my0(su);                   {fels  16 bit * alsó 16 bit}
mr= mr+mx0*my1(us);                   {alsó 16 bit * fels  16 bit}
mr0= mr1;
mr1= mr2;
mr= mr+mx1*my1(ss);                   {fels  16 bit * fels  16 bit}
DM(EA)= mr0;                           {eredmény alsó rész mentése}
DM(EF)= mr1;                           {eredmény fels  rész mentése}

```

Mint látható az alsó helyiértékek szorzatának nem csak a felső 16 bitje kerül át az MR regiszterbe, hanem a túlcsoordulás (MR2) értéke is átíródik az MR regiszter felső 16 bitjébe. Ez azért szükséges, mivel a szorzandók maradék része kettes komplementes számként lehet olyan nagy érték is, hogy a szorzás és összegzések hatására az eredmény meghaladhatja a 32 bitet.

A szuro32.dsp programban, mivel az együtthatók csak 16 biten vannak ábrázolva, a fenti szorzás egyszerűsödik. Az együtthatók alsó 16 bitje nulla, így azzal felesleges elvégezni a műveletet. A nyolc utasítás helyett így elegendő négy sor a 32 bites eredmény kiszámításához.

```

mr= mx0*my1(us);                      {alsó 16 bit * fels  16 bit}
mr0= mr1;
mr1= mr2;
mr= mr+mx1*my1(ss);                   {fels  16 bit * fels  16 bit}

```

A duplapontos számítás nem csak a szorzásnál lett megvalósítva. Mivel minden részeredmény 32 bites formában van tárolva, az összegzésnél és kivonásnál is biztosítani kellett ezt a pontosságot. Természetesen így az aritmetikai műveletek elvégzése is több utasításból áll. Összeadásnál először a két szám alsó helyiértékeit kell összeadni, és az eredményt tárolni, majd a két felső helyiérték következik, amelyhez hozzá kell adni a túlcsoordulás bitet.

```

ar= ax0+ay0;                          {alsó 16 bit + alsó 16 bit}
DM(tarlA)= ar;                         {eredmény alsó rész mentése}
ar= ax1+ay1+C;                         {fels  16 bit + fels  16 bit}
DM(tarlF)= ar;                         {eredmény fels  rész mentése}

```

A különbségképzés is ugyanilyen módon négy utasítással megvalósítható. Ilyen műveletet kell elvégezni, például a bemeneti jel és a visszacsatolt jel különbségeként előálló hibajel számításakor. Itt is figyelembe kell venni az áthozatal értékét. A duplapontos kivonás tehát a következő négy utasításból áll:

```

ar= ax0-ay0;                          {alsó 16 bit - alsó 16 bit}
DM(BEA)= ar;                           {különbségi jel alsó rész tárolása}
ar= ax1-ay1+C-1;                       {fels  16 bit - fels  16 bit}
DM(BEF)= ar;                           {különbség fels  rész tárolása}

```

Ezek a műveletek biztosítják a kritikus részek, vagyis a rezonátorok számítása, és a teljes hurkon belül lévő összes részeredmény számításánál a megfelelő pontosságot. Egyedül a kimeneti jel számítása történik 16 biten, mivel itt a pontosság nem annyira

kritikus. A kicsatoló együtthatók ugyanis már kívül helyezkednek el a visszacsatolt hurkon, ezért ezek hibája nem halmozódhat.

Mint a programrészletekből is kiderült, a legtöbb időt a szorzások elvégzése igényli. Ezért célszerű olyan helyeken, ahol kettő egész hatványa szerinti szorzást vagy osztást kell végrehajtani, shiftelést alkalmazni. A bitminta N értékkel jobbra vagy balra léptetése megfelel a szám 2^N -el való szorzásával vagy osztásával. Egy 32 bites szám shiftelése kevesebb utasítást igényel, mint a szorzás végrehajtása. Ezt a programban több helyen is alkalmaztam. Így például a túlvezérlés elkerülése érdekében a bemeneti jelet bizonyos frekvenciájú szűrők esetén felére le kellett osztani, majd a végeredményt visszaszorozni. Ez a művelet a skálázás. Ha a bemeneti jelet 32 bites formában shifteljük, elkerülhető a legalsó bit elvesztése. Természetesen az aritmetikai zaj szempontjából ez a művelet nem előnyös, ezért csak azoknál a szűrőknél alkalmaztam skálázást, amelyeknél a mérések egyértelműen bebizonyították, hogy szükséges. A 32 bites shiftelés a szorzáshoz képest mindössze két utasítást vesz igénybe:

```
right_in(ar);           {következ adat beolvasás AD-ről}
sr= ashift ar by -1 (hi); {beolvasott jel felére osztása}
sr= sr or lshift ar by -1 (lo);
```

Visszatérve a rezonátorok számítására, itt is adódik egyszerűsítési lehetőség, amellyel a program futása gyorsítható. Eredetileg a megtervezett szűrő fokszámával megegyező számú rezonátortagra lenne szükség, hiszen az eredeti IIR szűrő minden pólusa meghatároz egy rezonátor pólust (lásd 3.5.18 összefüggés). Természetesen itt már csak azokról a rezonátor pólusokról van szó, amelyekkel teljesíthetők a 3.5.17 feltételek. Az IIR szűrő pólusai azonban komplex konjugált párokat alkotnak, amelynek következménye, hogy a rezonátorok Z_n együtthatói is komplex konjugált párok lesznek. A rezonátortagok bemeneti jele viszont tisztán valós, mivel a különbségi jelet szorzó r_n együtthatók értékei is valósak. Bebizonyítható [4], hogy a 4.3.2 összefüggés alapján számított, a rezonátortagok kimenetén megjelenő eredmények is komplex konjugált párokat alkotnak. Ez azért nagy jelentőségű megállapítás, mert visszacsatoláskor a rezonátortagok kimenetének az összege van visszavezetve a bemenetre. Komplex konjugált pároknál az összegzés hatására a képzetes rész kiesik. Ugyanez a helyzet a kicsatoló W_n együtthatóknál is. Egy rezonátortaghoz, és a hozzá tartozó konjugált párjához rendelt kicsatoló együtthatók is komplex konjugált párt alkotnak. Mivel a végeredmény a rezonátorok kimeneti értékeinek és a hozzájuk tartozó kicsatoló együtthatók szorzatának az összegzéséből áll elő, itt is megállapítható az a tény, hogy a konjugált párok miatt a képzetes rész kiesik. A végeredmény tehát tisztán valós lesz. A számítások szempontjából ez nagyon előnyös, mivel a visszacsatolásnál és a kimenet számításánál a képzetes résszel nem kell törődni, hiszen annak értéke amúgy is nulla lesz. Sokkal nagyobb jelentőséggel bír azonban az a tény, hogy ugyanazt az eredményt tudjuk produkálni akkor is, ha a konjugált páronként működő rezonátor tagoknak csak az egyik tagjából nyert eredményt használjuk fel az összegzésnél, majd ezt az összeget megszorozzuk kettővel. Ezek szerint elegendő csak a megtervezett szűrő fokszámának a fele számú rezonátort megvalósítani. Ez mindekképp előnyös két szempontból is. Az egyik, hogy az idő legnagyobb részét kitevő, a rezonátorok számítását végző programrutin futásidejét a felére lehet csökkenteni. A másik előny a pontosság. Ugyanis a többszörös visszacsatolás és az együtthatók pontatlansága miatt kialakulhatna olyan eredmény, ahol a konjugált szimmetria kissé felbomlik, és a képzetes rész nem válik nullává. Azzal, hogy a visszacsatolásnál csak a valós részt számítjuk, és ennek a

kétszeresét vesszük, az előbb említett hiba nem fordulhat elő. Természetesen a kimeneti jel előállításához is felhasználhatjuk ugyanezt a kedvező tulajdonságot.

A kimenet a rezonátortagok összegének kicsatolásából és a különbségi hibajel értékének kicsatolásából adódik össze:

$$Ki(n) = \sum_{n=0}^N (X(n) * W_n) + b_n * BE(n), \quad (4.3.4)$$

ahol N a rezonátortagok száma. Mivel ezek a kicsatolások már nem vesznek részt a számítás szempontjából kritikus visszacsatolt hurokban, ezért az itt lévő úgynevezett kicsatoló együtthatók pontossága sem kritikus a végeredmény szempontjából. A szűrő vizsgálatánál elvégzett mérések is azt igazolták, hogy nincs szükség a kimenet számításakor a duplapontos számábrázolásra. Itt tehát a kicsatoló együtthatók mellett a számítások is 16 bites pontossággal történnek.

Mivel a W_n kicsatoló együtthatók (3.5.16 összefüggés) is komplex konjugált párokat alkotnak és a rezonátortagok kimenete is ugyanilyen szimmetriával rendelkezik, ezért itt is élhetünk azzal az egyszerűsítési lehetőséggel, hogy a műveleteket csak a konjugált párok egyik felével végezzük el, és az eredményt megszorozzuk kettővel. Itt is bebizonyítható ugyanis [4], hogy a W_n kicsatoló együtthatókkal elvégzett szorzás után is megmarad a konjugált komplex szimmetria, amelynek eredménye, hogy az összegzés következtében a képzetes rész kiesik, azaz nulla lesz. Az elmondottak alapján a kimeneti jel a következő kifejezés segítségével számítható:

$$Ki(n) = 2 \sum_{n=0}^{N/2} (ReX(n) * ReW_n - ImX(n) * ImW_n) + b_n * BE(n), \quad (4.3.5)$$

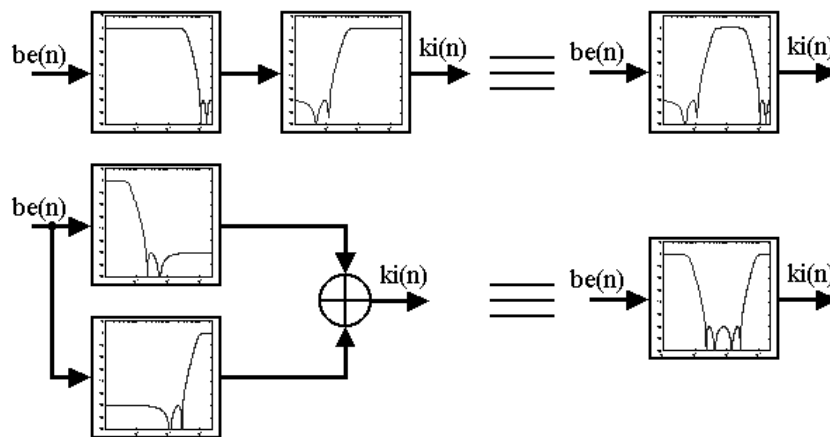
ahol N a rezonátortagok számát jelöli. Ennek páros számnak kell lennie, ami azt jelenti, hogy csak páros fokszámú szűrők valósíthatók meg a program segítségével. Páratlan fokszám esetén ugyanis keletkezne egy valós rezonátor pólus is, amelyet a program nem tud kezelni. Ez azért nincs megvalósítva, mert gyakorlatilag feleslegesen bonyolítaná a szűrőrutin felépítését. A 4.3.5 összefüggésben b_n a különbségi jel kicsatoló együtthatója. Ennek számítása a 3.5.15 képlet segítségével lehetséges. Ez az együttható egy tisztán valós érték.

4.3.2 Szűrőkarakterisztikák előállítása

A szűrők megvalósításánál felvetődött még egy kérdés. Ez pedig az, hogy mennyire biztosítható a rendelkezésre álló frekvenciatartományban a beállításoktól függetlenül, hogy azonos átviteli tulajdonságokat produkáló szűrőket tudjunk alkalmazni. Ugyanis az elvárás az ilyen programozható szűrőkkel szemben az, hogy ugyanazon törésponti frekvenciákat lehessen beállítani az áteresztő sáv határának sávszűrőknél, mint alul és felüláteresztő szűrők esetén. Emellett a levágási meredekségnek is minden törésponti frekvencián azonosnak kell lennie, így biztosítható, hogy a szűrő a teljes tartományban egyformán teljesíti a megszabott előírásokat.

Alul és felüláteresztő szűrőknél a direkt realizálásból indultam ki, vagyis a megtervezett szűrő $H(z)$ átviteli függvénye alapján lettek meghatározva a rezonátor együtthatók. Ez a megoldás a sávszűrők esetén is alkalmazható. Ki kell tehát számítani a direkt struktúra

együtthatóit, majd ezekből meghatározhatók a rezonátoros paraméterek. A probléma itt az, hogy keskeny áteresztő illetve elnyomási sávot előírva a levágási meredekség jóval nagyobb, mint szélesebb tartomány esetén. Tehát az átviteli karakterisztikák nem mutatnak egyezést a különböző áteresztősáv szélességűre tervezett sávszűrőknél. Természetesen a meredekebb levágás előnyös tulajdonság, de ebben az esetben csak akkor, ha ez minden frekvencián azonosnak mondható. Továbbá a stabilitás szempontjából is igen kritikus a keskenysávú sávszűrők megvalósítása. A másik megoldás, amely jobban illeszkedik az elvárásokhoz, miszerint a szűrők amplitúdó átviteli karakterisztikája minden frekvencián jellegre azonos legyen az, hogy a sávszűrőket alul és felüláteresztő szűrők segítségével állítjuk elő. Ezt a megoldást szemlélteti a 4.5 ábra.



4.1. ábra Sáváteresztő és sávzáró szűrők egy lehetséges megvalósítása

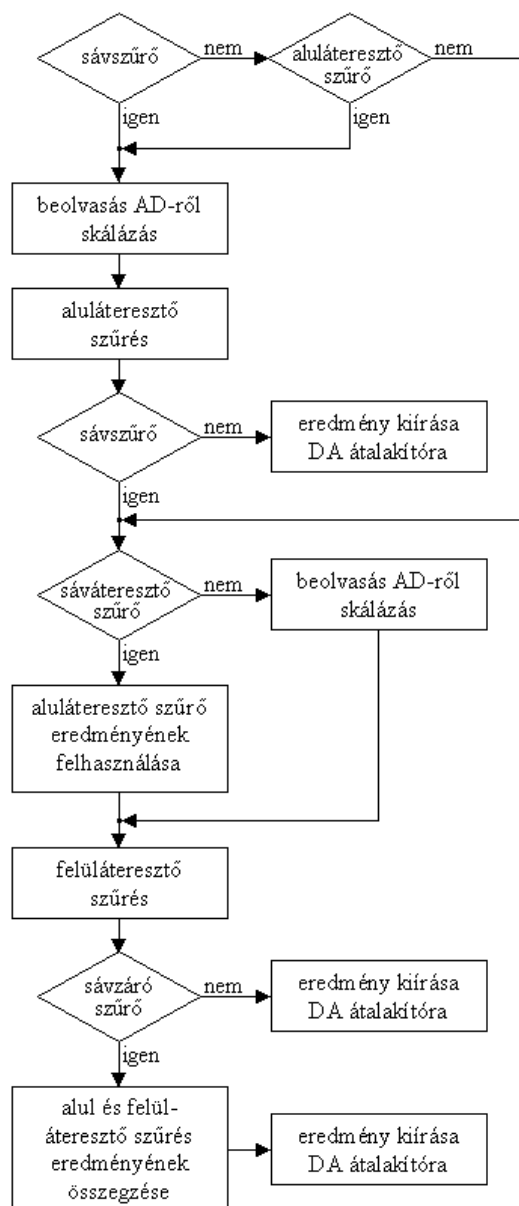
Régebbi, analóg típusú szűrőknél általános megoldás volt a sávszűrt jel ilyen módon történő előállítása. Sáváteresztő szűrőt a megfelelően beállított alul és felüláteresztő szűrőtagok sorba kapcsolásával nyerhetünk, míg sávzáró szűrő esetén ezeket a tagokat párhuzamosan kell kapcsolni. Az átviteli karakterisztika ekkor jellegre biztos, hogy azonos lesz minden frekvencián, hiszen a megtervezett alul és felüláteresztő szűrők meredeksége az átmeneti sávban minden törésponti frekvencián azonos.

Sajnos némi pontatlanság így is adódik. Soros kapcsolás esetén, sávközépen összeadódik az áteresztő tartomány hibája. Inverz Csebisev típusú szűrőnél, mivel az áteresztő tartomány maximálisan lapos közelítésű, ez a hiba egy amplitúdó csökkenést eredményez. A legnagyobb hiba akkor lép fel, amikor az alul és felüláteresztő szűrők törésponti frekvenciái egybeesnek. A törésponti frekvencián a legnagyobb megengedhető hiba $-0,5$ dB, amely ebben az esetben összegződik. Így sávközépen, amely ebben az esetben a beállított frekvenciára esik, a hiba nagysága $-0,9$ dB. Ha az alsó és felső törésponti frekvencia között egy oktáv a különbség, azaz két szomszédos frekvencia jelöli ki az áteresztő tartomány határait, a hiba nagysága már csak $-0,1$ dB, negyedfokú szűrők esetén. Hasonló hiba lép fel sávzáró szűrők esetén is. Itt az összegzés miatt a zárótartományi elnyomás értékében léphet fel ingadozás. Kedvezőtlen esetben a maradék jel összegződik, amelynek következménye, hogy az elnyomás mértéke lecsökken -54 dB-re.

Analóg műszereknél a szűrők sorba illetve párhuzamos kapcsolását a kimeneten lehetett megvalósítani. Ennek hátránya, hogy a kétcsatornás műszer sávszűrés esetén már csak egycsatornás szűrőként működik. Jelfeldolgozó processzor alkalmazásával ez a

probléma kiküszöbölhető. A soros illetve párhuzamos kapcsolás a programon belül megoldható, a kiválasztott működési módnak megfelelően.

Az egyszerűbb programszervezés szempontjából két teljesen azonos szűrőrutin fut egymástól függetlenül. Az egyik az aluláteresztő, míg a másikon a feluláteresztő beállításoknak megfelelő szűrők futnak. A szűrést végrehajtó programrész folyamatábrája a 4.6 ábrán látható. A program futása közben, a beállításoktól függően dől el, hogy éppen melyik rutinra adódik át a vezérlés. A beállításokat tartalmazó paramétervektort egy kiegészítő áramkörnek, egy kapcsolósornak a segítségével lehet állítani. Jelenleg ez valósítja meg a szűrő kezelőfelületét. A kapcsolósor segítségével a szűrő-karakterisztikát, a fokszámot és az áteresztő tartomány szélét (széleit) lehet módosítani. A szűrést megvalósító programrész feladata tehát az előkészített együttthatók felhasználásával a megfelelő szűrőrutinok futtatása. Az együttthatókat egy másik programrutin készíti elő. Ennek ismertetése a következő fejezetben lesz.

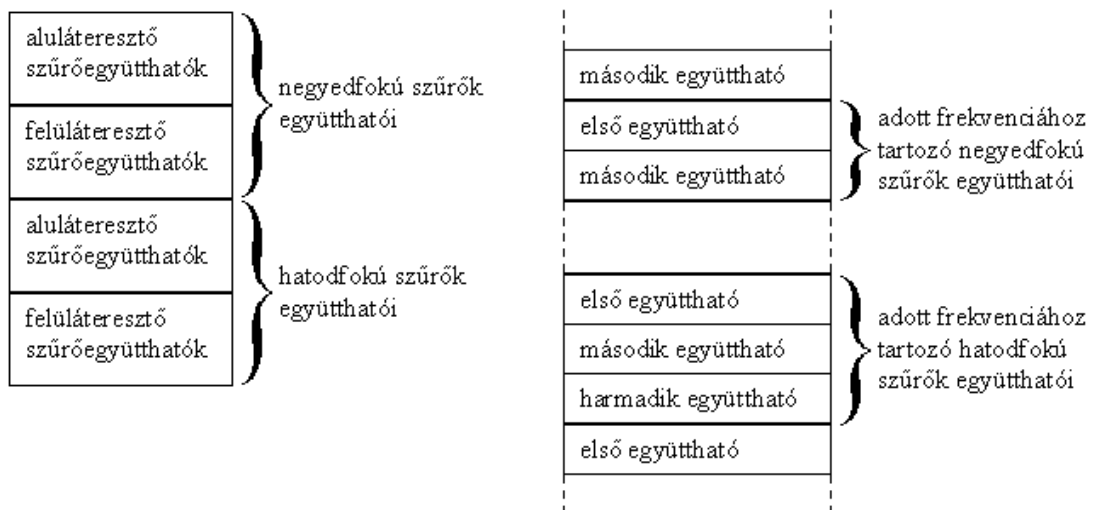


4.2. ábra A szűrést megvalósító programrész folyamatábrája

Alul vagy felüláteresztő szűrőbeállításnál az AD átalakító megszakítás kérelme, azaz a következő minta beolvasása után a vezérlés átadódik a megfelelő szűrőrutinra, amely közvetlenül szolgáltatja az eredményt (4.6 ábra). Sávszűrők esetén azonban mindkettő rutin futása szükséges, és az eredményképzésnél figyelembe kell venni, hogy a két rutin sorban egymás után, vagy párhuzamosan hajtódjon végre. Ezért az aluláteresztő szűrőket megvalósító programrutin végén feltételvizsgálattal el kell dönteni, hogy szükséges-e még a másik rutin végrehajtása (sorba kapcsolás). A felüláteresztő szűrőket megvalósító programrutinnál viszont az elején, az új adat beolvasásakor és a kimenet képzésekor is szükséges a feltétel vizsgálat. Az új adat beolvasása történhet közvetlenül az AD átalakítóról, vagy sáváteresztő szűrők esetén az előző aluláteresztő szűrőrutin eredményét kell tovább feldolgozni. Sávzáró szűrők esetén az aluláteresztő szűrőt megvalósító programrutin eredményéhez kell hozzáadni a felüláteresztő szűrő által szolgáltatott eredményt (párhuzamos kapcsolás).

4.3.3 Szűrőegyütthetők elhelyezése, elővétele

A szűrőegyütthetők a program elején adatfájlokból, a számukra lefoglalt programmemória helyekre lesznek beillesztve. Az adatfájlokat a már ismertetett MATLAB program generálja. Elhelyezkedésüket a memóriában a 4.7 ábra szemlélteti.



4.1. ábra Együtthetők elhelyezkedése a programmemóriában

Ez az elhelyezés mindegyik együtthetőcsoportra igaz (Z , W , b , rn). Természetesen a komplex együtthetőcsoportoknál (Z , W) külön van még választva a valós és képzetes rész is. Az 4.7 ábra alapján látható, hogy egy adott együtthetőcsoporton belül előbb a negyedfokú, majd a hatodfokú szűrőkhöz tartozó együtthetők vannak elhelyezve. Ezen belül az adattömb első felében a beállítható frekvenciákhoz tartozó aluláteresztő együtthetők, majd a felüláteresztő szűrőkhöz tartozó együtthetők helyezkednek el. A negyedfokú szűrőnél egy adott beállításhoz két együtthető, a hatodfokú szűrőnél pedig három együtthető tartozik, mivel ezek mindegyike komplex konjugált párt alkot. Így a lefoglalt memóriaterület nagysága $2*2*(N/2)$ illetve $2*3*(N/2)$ az egyes együtthető csoportoknál, ahol N a szűrő fokszáma. Komplex együtthetőknél a lefoglalt terület ennek még a duplája. Természetesen a különbségi jelet kicsatoló szorzótagnál (b), mivel minden beállításhoz csak egy ilyen együtthető tartozik, a memóriában lefoglalt terület nagysága, mind negyedfokú mind hatodfokú szűrő esetén $2*N$.

A programban ezek után meg kellett oldania, hogy a beállításoknak megfelelően a szükséges együttthatók mindig a megfelelő helyről legyenek elővéve. Ehhez mindegyik együttthatóhoz rendelni kellett egy a kezdőcímére mutató változót, amely a program futása közben mindig átadja az együttthatók kezdőcímének értékét a cirkuláris címzést végző mutatónak.

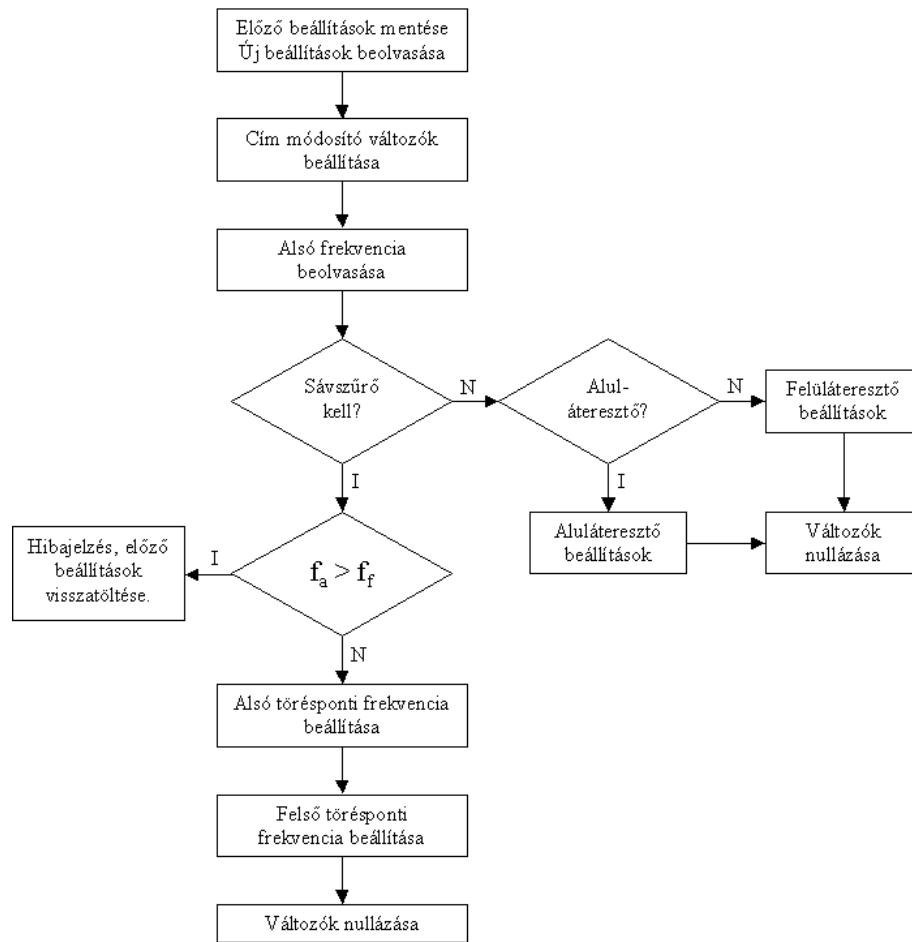
A programnak tehát minden új beállítás beolvasása után be kell állítani ezeket a kezdőcímekre mutató változókat (4.9 ábra). Ehhez a szükséges információt a kiegészítő áramkör kapcsolósorának 1-6 kapcsolói szolgáltatják. Az első három kapcsoló az alsó frekvenciát, a másik három a felső frekvenciát állítja be. Ez 2^3 azaz nyolc beállítható frekvenciaérték, amelyekhez tartozó együttthatók egymásután sorban helyezkednek el a memóriában, tehát a beállított frekvencia kódja egyértelműen meghatározza az együttthatónak az adatvektorban való elhelyezkedését. A beállításhoz valójában két címmutató lett felhasználva. Az egyik az adott együttthatótömbön belül a szűrő fokszámának megfelelő aluláteresztő vagy feluláteresztő együttthatócsoport kezdőcímére mutat. A másik mutató viszont ehhez a címhez képest a beállított szűrő sorszámának megfelelő együtttható párra vagy hármasra ugrik, amely a megfelelő törésponti frekvenciához tartozó együttthatók helyét adja meg. Ehhez ismerni kell az ugrás mértékét, amellyel a következő törésponti frekvenciákhoz tartozó együttthatókra lehet ugrani. Negyedfokú szűrőknél ehhez kettesével, hatodfokú szűrőknél pedig hármasával kell lépkedni a programmemóriában (4.7 ábra).

Az új beállításvektor beolvasása a fejlesztőkártyán található INTERRUPT nyomógomb lenyomása után történik. A főprogram (amely egy önmagára mutató végtelen ciklus) futása ekkor megszakad, és átadódik a vezérlés a beállítást elvégző rutinra. Ennek folyamatábrája látható a 4.8 ábrán.

A programrutin első lépésként elmenti a korábbi beállításhoz tartozó tízbites állapotvektort. Ennek szerepe, hogy hibás beállítás esetén az előző szűrő karakterisztika automatikusan visszatölthető legyen. Hibás beállítás sávszűrők esetén akkor fordul elő, ha sáváteresztő szűrőnél a felső törésponti frekvencia értéke kisebb, mint az alsó törésponti frekvencia érték. Sávszűrő szűrő esetén pedig, ha kisebb vagy egyenlő a felső törésponti frekvencia, mint az alsó. Ezt a kártyán a piros LED kigyulladás is jelzi, és mindaddig égve marad, míg érvényes beállítást nem kap a DSP a kapcsolósorról.

A 4.8 ábrán bemutatott programrutin, miután a korábbi beállítást elmentette, beállítja az elsődleges címmutatók értékét. Ehhez meg kell vizsgálnia, hogy negyedfokú vagy hatodfokú szűrő lett-e beállítva, majd ezután a 4.7 ábrának megfelelően az adott együttthatócsoport elejére állítja azt. Emellett beállítja a szűrő fokszámát tartalmazó változót és az együttthatóvektorban való ugrás mértékét.

A program ezután kétfelé ágazik. Alul vagy feluláteresztő szűrést alkalmazva csak egyetlen törésponti frekvencia állítható. Ezt az értéket a kapcsolósor első három kapcsolója tartalmazza. A beállítástól függően vagy az aluláteresztő együttthatóblokkban vagy a feluláteresztő együttthatókat tartalmazó blokkban lépkedve kell kiválasztani a beállított frekvenciához tartozó együttthatókat. Természetesen ezt mind a Z , W , b és rn együttthatókra el kell végezni és az együttthatóhoz rendelt címmutatót ennek megfelelően kell az adott együtttható memóriacíméhez rendelni. Miután a címmutatók beállítása, és a szükséges együttthatók adatmemóriába másolása megtörtént, el kell még végezni a rezonátoros szűrő belső jeleinek törlését. Ezután a program visszatér a megszakításból a főprogramba.

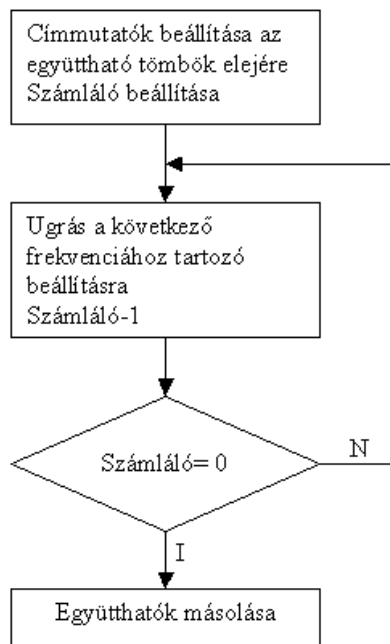


4.2. ábra A szűrőbeállítást végző programrész folyamatábrája

Sávszűrő esetén már mind az alsó mind a felső frekvencia beállításra szükség van. A hibavizsgálat elvégzése után itt is a megfelelő együtthatókra mutató címek beállítása következik. Sáváteresztő szűrőnél az alsó törésponti frekvenciát a felüláteresztő szűrőtag együtthatói határozzák meg, míg a felső törésponti frekvenciát az aluláteresztő szűrőtag együtthatói. Sávszűrő szűrőnél ez pont fordítva van. Ezért az együttható címének beállításánál vagy az aluláteresztő vagy a felüláteresztő együttható beállítást kell meghívni először, majd utána természetesen a másikat. A címmutatók beállítása után itt is meg kell hívni a rezonátoros szűrő belső változóinak nullázását végző programrészt.

Az, hogy az együttható vektorban az elsődleges címhez képest az adott frekvenciához tartozó együtthatók címe hogyan kerül kiválasztásra a 4.9 ábra mutatja be. A 4.8 ábrán ezt a folyamatot az aluláteresztő és felüláteresztő beállítások jelölik, sávszűrők esetén is természetesen ugyanez hajtódik végre a megfelelő sorrendben.

Az elsődleges címmutató a szűrő fokszámától függően a hozzá tartozó együtthatók (aluláteresztő, felüláteresztő) csoportjának elejére mutat (4.7 ábra). Azt, hogy a blokkban mennyit kell lépkedni, a beállított törésponti frekvencia száma valamint az együtthatóvektorban szükséges ugrás mértéke határozza meg (4.7 ábra). Negyedfokú szűrőknél kettő, míg hatodfokú szűrőnél három együttható tartozik egy beállításhoz. A b különbségi jelet kicsatoló együtthatónál minden beállításhoz csak egyetlen érték tartozik. Ugyanígy a leosztást tartalmazó s vektor esetében is.



4.3. ábra Az együtthatók címének kiválasztása

A címmutatók az elsődleges címnek megfelelően az alul vagy felüláteresztő együtthatókat tartalmazó blokk elejére állítódnak (4.9 ábra). Ezután egy ciklus elkezdni növelni a címmutató értékét a beállított ugrás mértékének megfelelően, egészen addig, míg a törésponti frekvencia számát tartalmazó számláló nulla nem lesz. Ez azt jelenti, hogy minden mutató a programmemóriában a neki megfelelő címre mutat. Ezután a b , a skálázás mértékét beállító s és az rn együtthatók átmásolása következik az adatmemóriába. Az, hogy némely együttható az adatmemóriába kerül, a program futása szempontjából előnyös. Párhuzamos utasítás végrehajtásnál ezen együtthatók egyszerre érhetők el a programmemóriában tárolt együtthatókkal. És így a számítási idő lerövidíthető volt.

A program elején el van helyezve egy konstans bináris számsorozat, amely egy bekapcsolás utáni alapértelmezett szűrőbeállítást tartalmaz. Tehát a szűrőprogram elindulása után ez a beállítás fog működni. Az inicializálás és változók beállítása után a program, még mielőtt a vezérlés a főprogramra adódna, meghívja ezzel a beállítással az eddig ismertett szűrőegyüttható beállítást végző rutint. Programindítás után tehát a jelfeldolgozó kártya egy hatodfokú inverz Csebisev approximációval megvalósított, 500Hz és 2kHz között sáváteresztő szűrőként fog viselkedni.

4.4 A kiegészítő áramkör

Miután a szűrést megvalósító programrészlet már megbízhatóan működött, meg kellett valósítani a különböző szűrőkarakteristikákhoz tartozó együtthatók cseréjét. Mint ez már a korábbi fejezetekből is kiderült, a kiszámított 16 bites együtthatók a program-, illetve az adatmemóriában kerülnek tárolásra, ahonnan aztán szűrőkarakterisztika váltás esetén a régi együtthatók helyére töltődnek. A fejlesztőkártyán található nyomógomb azonban nem sok segítséget tud nyújtani a szűrők beállításához. Ezért vált szükségessé egy olyan kezelőfelület megépítése, amellyel ezek a beállítások könnyebben elvégezhetővé válnak.

Az ADSP-2181-es processzor az ADSP-21xx családhoz képest már egy módosított memória-felépítést és memóriakezelést tartalmaz [5]. A belső 16K x 24 bites programmemória és a belső 16K x 16 bites adatmemória mellett négy elkülönülő külső memóriamezőt tud kezelni. Ezek a program, adat, és bajtmemória szegmensek, valamint a külső I/O eszközök illesztésére lefoglalt I/O memória terület. Számunkra most ez utóbbi a fontos, hiszen a processzorhoz illesztett külső perifériák ezen keresztül mint memóriába ágyazott címtartományok könnyedén írhatók, illetve olvashatók. A címezhető tartomány 11 bit, vagyis a rendelkezésre álló memóriatartomány 2048 szegmens, amelyek 16 bites adatszélességgel rendelkeznek. A periféria hozzáférésnél az ADSP-2181 a címbusz alsó 11 bitjét használja, a többi állapot ilyenkor közömbös. Az I/O területet az ADSP-2181 mint egy 16 bites memóriát kezeli. Ehhez az adatbusz felső 16 bitjét használja adatátvitelnél. Az I/O memóriaterületnek a kiválasztásához egy külön engedélyező vonal van kiépítve a processzorban. Ez az $\overline{\text{IOMS}}$ jel. Ennek alacsony állapota jelzi, hogy a processzor ehhez a memóriaterülethez kíván hozzáférni.

A 2K (16 bit széles) I/O memória terület további négy részre van osztva. Ezen négy területen belül a memória hozzáféréshez rendelt várakozási idő külön-külön meghatározható. Ezzel a processzor a lassabb perifériákkal való kompatibilitást valósítja meg. A várakozási idő maximális értéke 8 órajel ciklus lehet. A beállítására a Wait State Control Regiszter használható, melynek az alsó 4x3 bitje szolgál a négy forma I/O memóriaterület írás vagy olvasás késleltetésének beállítására.

Ezek után tehát egy külső periféria illesztése a következő módon történhet. A periféria számára a memóriatartományban kijelölünk egy szabad területet. A periféria kiválasztását egy címdekódolóra bizzuk. A címdekódolónak a címbusz alsó 11 bitjét kell figyelnie. A perifériacím kiválasztása előtt a processzor az $\overline{\text{IOMS}}$ jel alacsony szintjével jelzi, hogy az I/O memóriaterülethez kíván hozzáférni. Íráskor vagy olvasáskor a processzor az $\overline{\text{RD}}$ illetve $\overline{\text{WR}}$ jelet alacsony szintre viszi és az adatbusz felső 16 bitjének tartalmát, a beállított várakozási idő után beolvassa, illetve a kiírti kívánt adatot kihelyezi az adatbuszra.

Programban erre az IO utasítás használható. Ennek szintaktikája a következő:

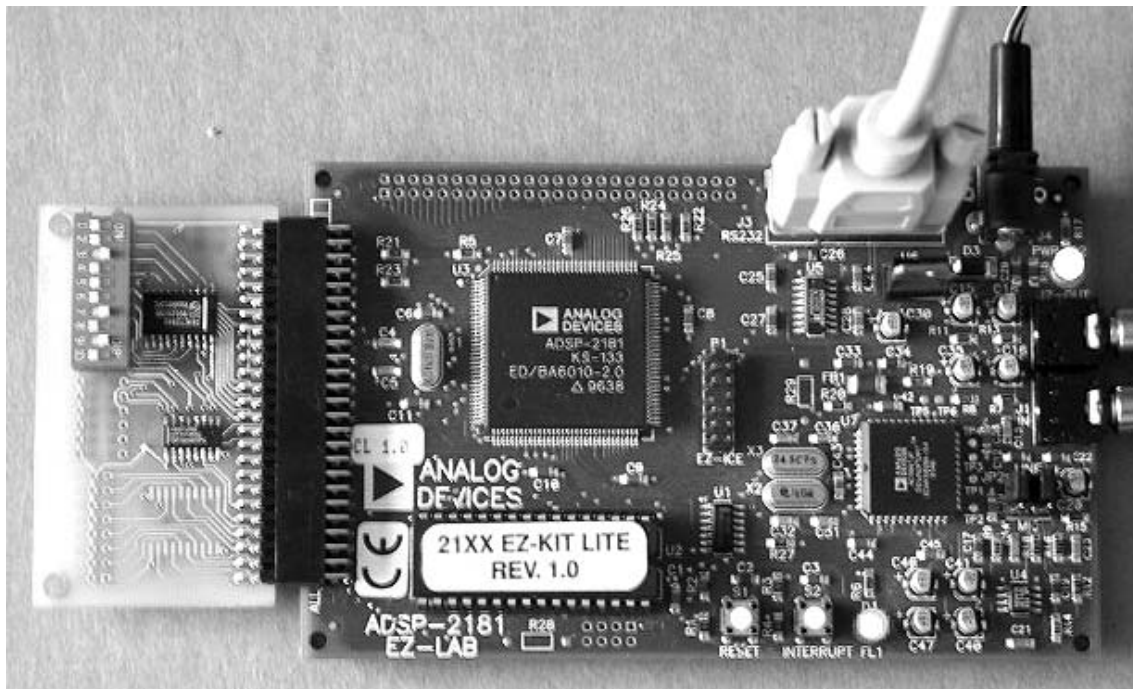
```
Perifériaolvasás:   adatregiszter= IO(<memóriacím>);
```

```
Perifériaírás:     IO(<memóriacím>)= adatregiszter;
```

Az I/O memóriaterület hozzáférése előtt azonban érdemes a periféria sebességének megfelelő késleltetési időt beállítani.

Az ADSP-2181 segítségével tehát könnyedén megvalósítható a külső eszközök illesztése. A kiegészítő áramkör feladata, hogy egyfajta egyszerű kezelőfelület biztosítson a felhasználók számára, a programozható szűrő beállításához. Ezzel a tesztelés és a bemérés is egyszerűbbé tehető. A fejlesztőkártyán ugyanis meglehetősen kevés nyomógombot helyeztek el. A programozható szűrőknél pedig éppen az egyszerű kezelhetőség az egyik alapfeltétel. Mivel azonban a programozható szűrő mint önálló áramkör nem kerül megépítésre, úgy gondoltam, nem érdemes egy komoly, forgókapcsolókkal és kapcsolókkal ellátott kezelőfelületet illeszteni az ADSP-2181-hez. Ehelyett inkább egyszerűbb, a kártyához könnyen illeszthető kezelőfelület megtervezését tűztem ki célul. Ezért döntöttem végül a kapcsolósor alkalmazása mellett. Ez egyetlen memóriacímen érhető el, és a tartalmát, mint bináris vektort olvassa

be a processzor. A fejlesztőkártya és a hozzá csatlakoztatott kiegészítő áramkör a 4.10 ábrán látható.



4.1. ábra A fejlesztőkártya és a kiegészítő kezelőfelület

A programozható szűrő az alábbi beállítási lehetőségeket tartalmazza:

- Választási lehetőség két szűrőtípus, jelenleg negyedfokú és hatodfokú szűrő alkalmazása között.
- Szűrő karakterisztika választás: alul-, felüláteresztő vagy sávzáró, sáváteresztő szűrő használata.
- Törésponti frekvencia választás (50 Hz, 100 Hz, 200 Hz, 500 Hz, 1 kHz, 2 kHz, 5 kHz, 10 kHz).

A kapcsolósor tehát mint bináris vektor tartalmazza a fenti beállításokat. A programban ez egy megadott címről történő 16 bites vektor beolvasásával lett megvalósítva. A fent leírt funkciók megvalósításához elegendő 10 bit. Mivel egy memóriamező 16 bit széles, ezért a kapcsolósor csak egyetlen mezőt foglal le az I/O memóriaterületből.

Hardver szempontból meg kellett valósítani, hogy a kapcsolósor a kártya normál működését ne zavarja. Amikor nincs hozzáférés, nagyimpedanciás állapotba kell kapcsolni.

A kapcsolósor állapotának beolvasása az \overline{IOMS} és az \overline{RD} jel aktívva válása után történik. Mivel a kapcsolósor eleve csak olvasható eszköz, ezért a hardver nem támogatja a perifériaírást. Az adatbuszról történő leválasztást két darab 74AHCT244 típusú buszmeghajtóval valósítottam meg. Ezek, ha nincs engedélyezve a kimenetük, azt nagyimpedanciás állapotba viszik. Ezzel a leválasztás megoldott. Az engedélyezés feltétele, vagyis, hogy a kapcsolósor állapotát a buszmeghajtó az adatbuszra tegye, a helyes perifériacím és az IOMS továbbá az RD együttes megjelenése. A cím

dekódolását egy 74ALS520 típusú komparátor végzi. Mivel ennek az áramkörnek TTL bemenetei vannak, a szabadon hagyott lábak logikai egynek számítanak. A perifériacím alsó nyolc bitje egy jumper segítségével (JP1) konfigurálható. A cím felső három bitje már egy 74AHCT138 típusú demultiplexerre van kötve. Ezért ezek már fix értékek, nem konfigurálhatók át. A demultiplexer maradék két kapuzó lábára az \overline{IOMS} és az \overline{RD} jelek vannak kötve. Így a demultiplexer kimenete vezérli a buszmeghajtó engedélyező bemenetét, és aktív állapotba kapcsolja, ha periféria olvasás következik be és a perifériacím megegyezik a jumperek által megadott címmel. A kapcsolósor címe ezek után: 100xxxxxxxx binárisan, ahol az alsó nyolc bit a jumperek segítségével konfigurálható. A kiegészítő áramkör kapcsolási rajza megtalálható a függelékben.

Mivel a programozható szűrő beállításaihoz elegendő a 10 kapcsoló, ezért a beolvasott 16 bites adat felső 6 bitjét a program a beolvasás után figyelmen kívül hagyja, hiszen ilyenkor ezek állapota, a buszmeghajtó bemenetén található felhúzó ellenállások miatt, azonosan logikai magas szinten áll.

A beolvasó rutin tartalmazza még a Wait State Control Register beállítását is. A mérések ugyanis azt mutatták, hogy a buszmeghajtó kimenetei a leggyorsabb hozzáférésnél (a processzornál ez, mint alapértelmezett érték áll be reset után) még nem állnak be stabilan a kapcsolósor állapotának megfelelően. A kiegészítő áramkör teljes késleltetése, az \overline{RD} jel megjelenését követően a kimenetek stabil beállításáig, katalóguslapok alapján, a legrosszabb esetben 15ns. Ezért a memória-hozzáférés késleltetését a beolvasás előtt a legnagyobb értékre kell állítani, hogy a kapcsolósor állapota az adatbuszon stabilan megjelenjen a tényleges beolvasás pillanatáig.

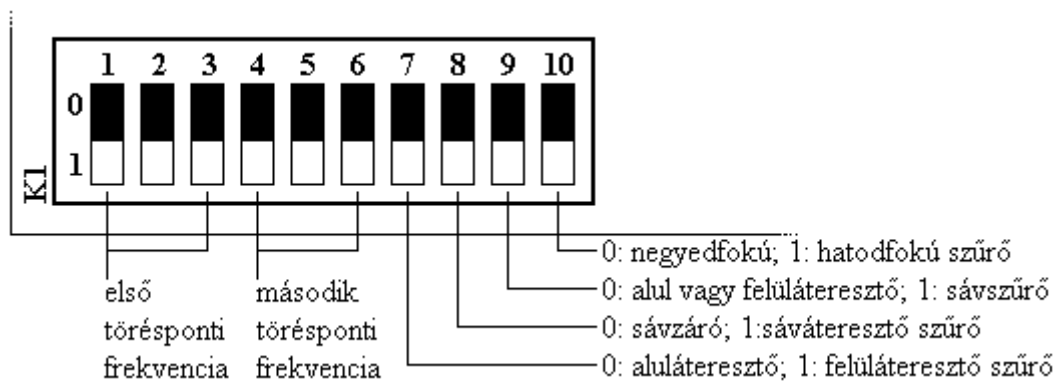
```

ar= DM(tipus);           {korábbi beállítások elmentése}
DM(regi)= ar;
ar= DM(Dm_Wait_Reg);    {késleltetés IO olvasáshoz}
DM(Dm_Wait_Reg_Regi)=ar; {eredeti állapot mentese}
ar= 0xFFFF;
DM(Dm_Wait_Reg)= ar;    {maximális várakozás}
ar= IO(b#10011111111);  {beolvasása az 10011111111 címr 1}
DM(tipus)= ar;
ar= DM(Dm_Wait_Reg_Regi); {késleltetés visszaállítása}
DM(Dm_Wait_Reg)= ar;
call beall;              {új beállítások végrehajtása}
rti;

```

A kiegészítő kártyáról történő adatbeolvasást a DSP kártya INTERRUPT nyomógombjával lehet elindítani. Ennek hatására hajtja végre a program a fenti rutint, majd átadódik a vezérlés a 4.3.3 fejezetben ismertetett, a szűrőegyütthetők cseréjét megvalósító programrészre.

A kapcsolósor egyes kapcsolóinak szerepét a 4.11 ábra mutatja be. A kapcsolók állapota logikai 0 vagy 1 lehet. A 10. kapcsolóval lehet a kétfajta szűrőtípus között váltani. Jelen esetben ez negyedfokú vagy hatodfokú inverz Csebisev approximáció lehet. A 9. kapcsoló szerepe, hogy a program számára jelezze, csak az alsó törésponti frekvenciát (alul vagy felüláteresztő szűrő esetén) vagy az alsó és felső törésponti frekvenciát is figyelembe kell venni (sáváteresztő vagy sávzáró szűrő esetén). A 8. kapcsoló sávzáró és sáváteresztő szűrő közti átkapcsolást, míg a 7. kapcsoló az alul és felüláteresztő szűrő közti váltást valósítja meg.



4.2. ábra A lehetséges beállítások a kapcsoló segítségével

A törésponti frekvenciák beállítására tehát három bit szolgál. Ezzel a nyolc frekvencia-érték már beállítható. A törésponti frekvenciák és a kapcsolók állása közti össze-
rendelést a 4.1 táblázat mutatja.

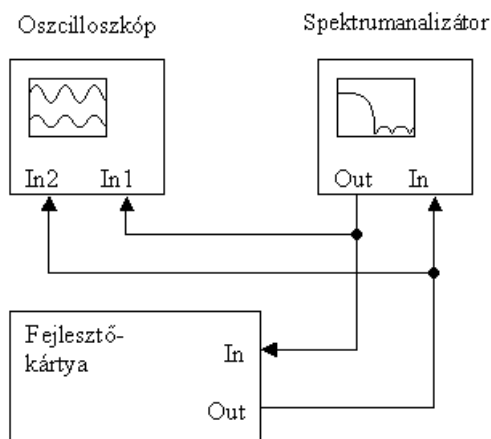
Kapcsolók állása	Beállított törésponti frekvencia
000	50Hz
100	100Hz
010	200Hz
110	500Hz
001	1kHz
101	2kHz
011	5kHz
111	10kHz

4.1. táblázat A beállítható törésponti frekvenciák

Aluláteresztő és felüláteresztő szűrőnél az első törésponti frekvencia állítja be az áteresztő sáv határát. Sáváteresztő és sávzáró szűrőnél az első törésponti frekvencia az alsó, míg a második a felső törésponti frekvencia beállítására szolgál.

5 Mérési eredmények feldolgozása

Ebben a fejezetben az elkészült programozható szűrő bemérése során készült mérési eredmények kiértékelése következik. A kiértékelés során a mérési eredményeknek csak egy része, a működés szempontjából kritikusabbnak mondható szűrőbeállításokról készült ábrák lesznek bemutatva. További, a mérés során felvett eredmények megtalálhatók a függelékben. A mérés a 5.1 ábra szerinti elrendezésben zajlott.

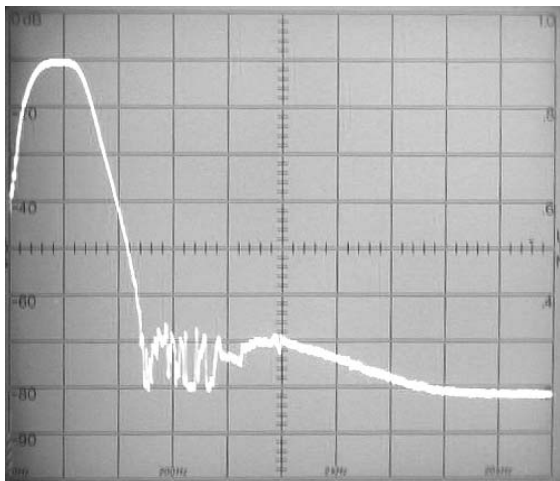


5.1. ábra A mérési elrendezés

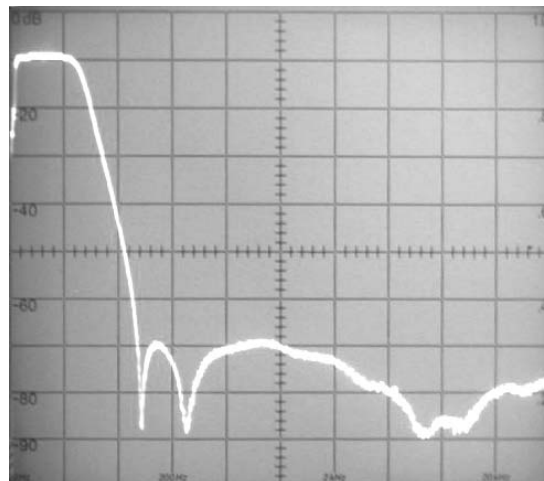
A spektrumanalizátor sávszélessége 50 kHz, amellyel a hangfrekvenciás tartományban működő szűrő átviteli karakterisztikája a teljes sávban felvehető. Emellett rendelkezett azzal a dinamikatartománnyal, amellyel a szűrők zárósávi viselkedése is vizsgálható. A kimeneti jel amplitúdója viszonylag széles határok között (0,1 V – 5 V) állítható, tehát alkalmas a szűrő teljes kivezérlés melletti és kisszintű jelekkel történő meghajtás esetén is, a megszürt jel vizsgálatára. A kimeneti jelalak méréséhez ezért szükség volt egy oszcilloszkópra is. Ezzel ellenőrizni lehetett az esetleges túlvezérlést, illetve a szűrés következtében esetleg fellépő amplitúdó torzulásokat vagy a kimeneti jel zajosságát.

A mérés során tehát elsősorban a működés szempontjából kritikus szűrőkarakterisztikák lettek felvéve. Ezek az alacsony törésponti frekvenciájú szűrők. Mivel a 3.5.4 fejezetben, a szűrőtípusok összehasonlítása után kiadódott, hogy a 3.1 fejezetben meghatározott specifikációk teljesítéséhez hatodfokú inverz Csebisev szűrő alkalmazása lenne célszerű, ezért a kiértékelés során is ezekkel a beállításokkal felvett eredmények lesznek bemutatva. Összehasonlításként a negyedfokú szűrőbeállítás mellett kapott eredmények megtalálhatók a függelékben.

Korábban, a szűrőtervezést ismertető fejezetben többször említésre került, hogy a mintavételi frekvenciától távol, alacsony törésponti frekvencián működő szűrők esetén a pólusok egymáshoz és a komplex frekvenciatartomány egységsugarú köréhez viszonyított távolságuk csökkenése miatt a szűrőkarakterisztikákban torzulás léphet fel. Az 5.2 ábrán az 50 Hz-es aluláteresztő szűrő amplitúdóválasza látható. A mérések 4 V csúcsertékű bemenő jel mellett lettek regisztrálva. Ez közel esik az AD-átalakító kivezérlésének határához, tehát a jel/zaj viszony itt a legkedvezőbb.



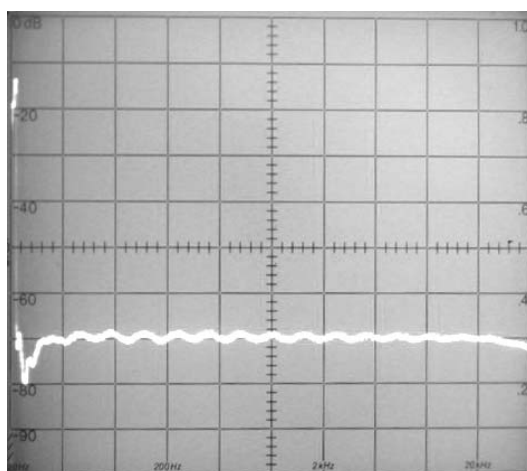
5.2. ábra Hatodfokú, 50 Hz-es aluláteresztő szűrő (függőlegesen 50 Hz/osztás, vízszintesen 10 dB/osztás)



5.3. ábra Hatodfokú, 1 kHz-es aluláteresztő szűrő (függőlegesen 1 kHz/osztás, vízszintesen 10 dB/osztás)

Az 5.2 ábrán látható amplitúdó karakterisztika a 0 Hz és 500 Hz közötti frekvenciasávot ábrázolja. A görbe elején a felfutó szakaszt a DC leválasztást végző szűrőtag okozza. Az ábrán látható, hogy az zárótartomány kezdetén a szűrő még nem tudja teljesíteni az előírt 60 dB elnyomást, míg magasabb fokszámú szűrőknél (5.3 ábra) ez a hiba már megszűnik. A jelenség a rezonátorok és a teljes visszacsatoló hurokban végrehajtott kerekítések és ezen hibák halmozódásából keletkező számítási pontatlanság eredménye. Ugyanis itt a rezonátor pólusok értéke olyan kicsi, hogy a velük végzett műveletek eredménye a számábrázolási tartomány alsó határára kerül. Ezt a fajta hibát a rezonátor pólusok pontosabb meghatározása nem befolyásolja. Ugyanis itt nem a pólusok pozíciójának pontatlansága, hanem azok nagyon kis értéke okozza a hibát. Ennek bizonyítása mérésekkel is megtörtént. A rezonátoros szűrőnek elkészült egy olyan változata, amely az együtthatókat is 32 bites formában használta fel. Ez azonban lényegi változást nem eredményezett.

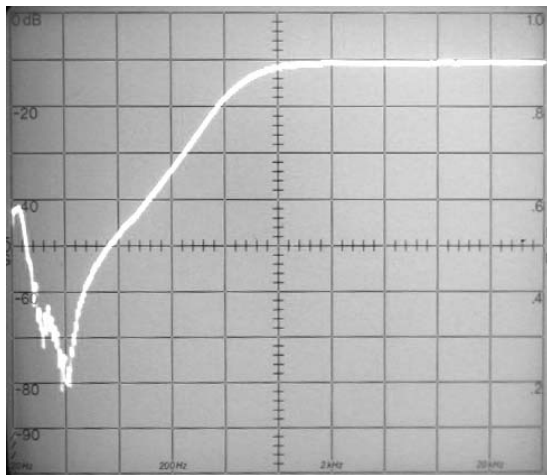
Az 5.4 ábrán az 50 Hz-es aluláteresztő szűrő átviteli karakterisztikája látható a 0 Hz és 20 kHz-es frekvenciasávban.



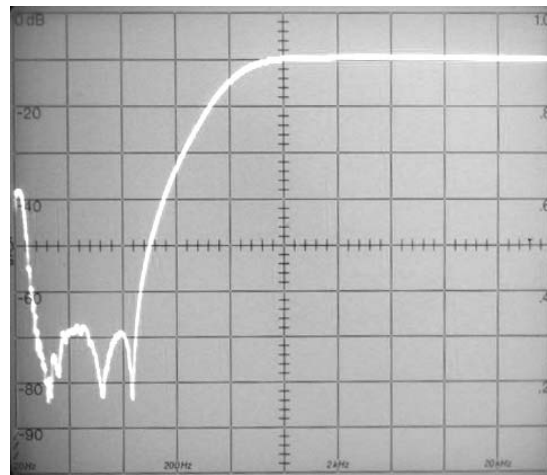
5.4. ábra Az 50 Hz-es aluláteresztő szűrő átvitele a teljes frekvenciasávban (függőlegesen 2 kHz/osztás, vízszintesen 10 dB/osztás)

Az 5.4 ábrán látható mérési eredmény már olyan széles frekvenciasávban lett felvéve, hogy az aluláteresztő szűrő áteresztő tartomány már alig mérhető. A mérés célja azonban itt az volt, hogy szemléltesse, az első oldalhullámban jelentkező hiba a teljes működési tartományban máshol már nem tapasztalható. Ez ugyanígy a magasabb törésponti frekvenciájú szűrőkre is igaz, vagyis sehol nem tapasztaltam kiugrást a zárótartományban.

Mérések alapján az 50 Hz-es felüláteresztő szűrők viselkedése is hasonló hibákat produkál 5.5 ábra. A magyarázat itt is ugyanaz, a rezonátorpólus frekvenciák túlságosan kis értékűek, a számítások eredményei közel kerülnek a számábrázolás alsó határához.



5.5. ábra Hatodfokú 50 Hz-es felüláteresztő szűrő (függőlegesen 10 Hz/osztás, vízszintesen 10 dB/osztás)



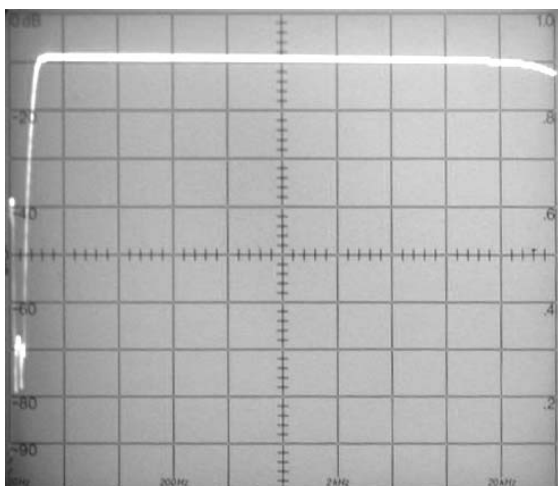
5.6. ábra Hatodfokú 1 kHz-es felüláteresztő szűrő (függőlegesen 200 Hz/osztás, vízszintesen 10 dB/osztás)

Az 5.5 ábra az amplitúdókarakterisztika 0 Hz és 100 Hz közé eső részét mutatja. Látható, hogy itt még az átmeneti tartományban is fellép némi karakterisztika torzulás és az első oldalhullám okozta hiba is valamivel nagyobb, mint az 50 Hz-es aluláteresztő szűrő esetén. Az elnyomás értéke itt 55 dB. Ez a hiba magasabb fokszámú szűrőknél már megszűnik (5.6 ábra).

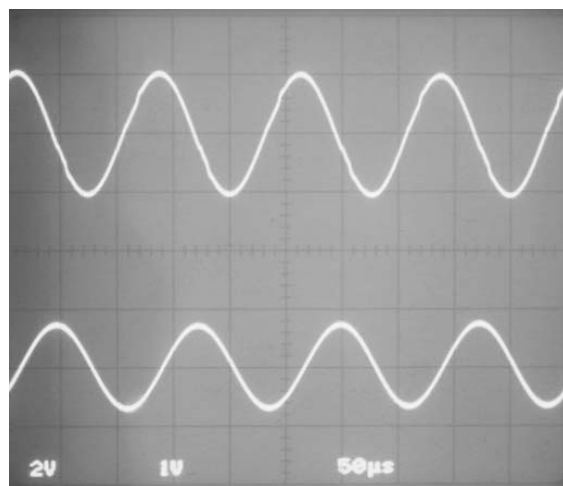
A karakterisztikák bevezető szakaszán látható kiugrás a spektrumanalizátor hibájából adódik. A mérés 0 Hz-ről indul, ahol még belezavar a mérésbe a kondenzátorok feltöltöttségéből adódó hiba. A letapogatási időt nagyon nagyra kellett volna választani ahhoz, hogy a szűrőkondenzátorok kisülése ne legyen látható az ábrákon.

Itt is elkészült egy, a telje frekvenciasáv átvitelét szemléltető ábra (5.7 ábra). Ehhez a méréshez az 1 kHz-es felüláteresztő szűrő lett beállítva. A mérés azt bizonyítja, hogy a teljes frekvenciasávban, az AD-átalakító kivezérléséhez közeli, 4 V csúcserértékű bemeneti jel esetén sehol sem lép fel gerjedés az áteresztő tartományban. Ez mindegyik szűrőbeállítás mellett igaz.

Az 5.8 ábra a megszürt jel jelalakját (alsó csatorna) mutatja be 4 V csúcserértékű, 8 kHz-es bemeneti jel esetén (felső csatorna). Az ábrán jól látható, hogy a jelben semmiféle torzulás nem lép fel, tehát a szűrők valóban a teljes bemeneti feszültségtartományban használhatók. A kimeneten gyakorlatilag nem mérhető a bemenő jelen lévő zajnál nagyobb érték.



5.7. ábra Az 1 kHz-es felüláteresztő szűrő átvitele a teljes frekvenciasávban (függőlegesen 2 kHz/osztás, vízszintesen 10 dB/osztás)

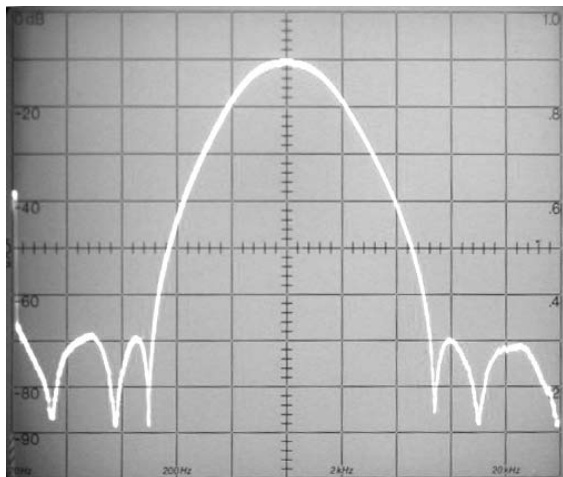


5.8. ábra A szűrés utáni jelalak (felső csatorna 2 V/osztás, alsó csatorna 1 V/osztás)

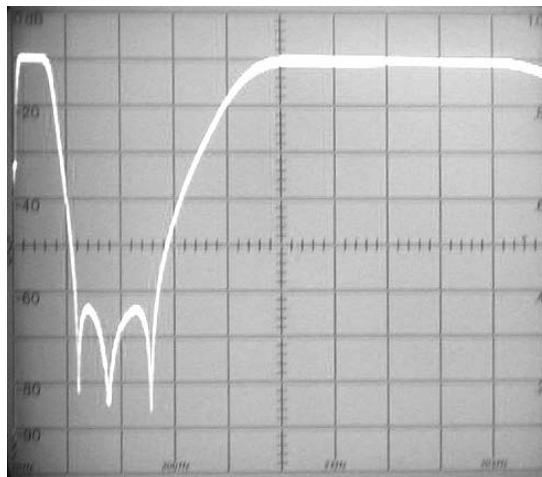
Megállapítható tehát, hogy mind aluláteresztő, mind felüláteresztő szűrőnél a legalsó, 50 Hz-es törésponti frekvenciájú karakterisztikánál már föllép valamekkora hiba a zárótartományban. Azzal, hogy hatodfokú szűrő helyett negyedfokút alkalmazunk, ez a hiba csökken. Ugyanis itt a pólusok pozíciója kedvezőbb elhelyezkedést mutat, tehát a számításokból adódó hiba itt kevésbé jelentkezik. A negyedfokú szűrőkkel felvett átviteli karakterisztikák megtalálhatók a függelékben.

Eddig csak a kivezérlés határához közel eső, nagyszintű bemenő jelekkel történt mérésekről volt szó. Az ilyen szűrőknél azonban elvárt követelmény, hogy a kisszintű jeleket is a specifikációnak megfelelően szűrjék meg. A mérések természetesen ilyen beállítások mellett is elkészültek. A méréshez az eddiginél 20 dB-lel alacsonyabb szintű, azaz 0,4 V csúcsértékű jel lett felhasználva. Az így felvett átvitel, 100 Hz-es aluláteresztő szűrő esetén a 3.17 ábrán látható. A kimeneten mért jelalakot pedig a 3.18 ábra mutatja. Ezek alapján látható, hogy az áteresztő tartomány továbbra is torzításmentes. A zárótartományban azonban csökken az elnyomás mértéke, ez kevéssel haladja meg az 50 dB-t. Tehát valóban elengedhetetlen a bemeneti erősítésszabályozás alkalmazása. Ezzel a bemeneti jel mindig beállítható olyan feszültségértékre, ami még megfelelően kivezéri az AD-átalakítót.

A következőkben a sávszűrők okozta hibára láthatunk példát. Mint ahogy az a 4.3.2 fejezetben ismertetésre került, a sáváteresztő szűrő az alul és felüláteresztő szűrők sorba kapcsolásával, míg a sávzáró szűrő ezek párhuzamos kapcsolásával lett megvalósítva. Ennek következménye, hogy sáváteresztő szűrőknél sávközépen egy amplitúdó csökkenést tapasztalhatunk (5.9 ábra), míg sávzáró szűrők esetén az elnyomási tartományban lép fel valamekkora hiba (5.10 ábra). Ennek a hibának az értéke sáváteresztő szűrő esetén $-0,9$ dB akkor, ha az alsó és felső törésponti frekvenciák egybeesnek. Sávzáró szűrőnél a zárósávi maradékjel a párhuzamos kapcsolat miatt összegződik. Az eredetileg 60 dB-es elnyomás ekkor lecsökken 54 dB-re. Ez a zárósáv egyenletes ingadozású közelítésének következménye.



**5.9. ábra Sáváteresztő szűrő hibája
(függőlegesen 2 kHz/osztás, vízszintesen
10 dB/osztás)**



**5.10. ábra Sávzáró szűrő hibája (függőlegesen
2 kHz/osztás, vízszintesen 10 dB/osztás)**

Mindkét ábra a teljes 0 Hz – 20 kHz-es frekvenciasávot mutatja. Az 5.9 ábrán olyan sáváteresztő szűrő van beállítva, amelynek alsó és felső törésponti frekvenciája egybeesik és ennek értéke 10 kHz. Az amplitúdóhiba tehát itt $-0,9$ dB. Az 5.10 ábrán látható szűrő 1 kHz és 10 kHz között valósítja meg az 54 dB elnyomást.

6 Továbbfejlesztési lehetőségek

Mint ahogy az az előző fejezetből is kiderült, a megvalósított programozható szűrőnek vannak hibái. Ezek korrigálásához további fejlesztésekre lenne szükség mind hardver, mind szoftver szinten.

Elsősorban azonban a hardver továbbfejlesztése lenne a fontosabb, mivel a jelenlegi állapot még biztosítja a programozható szűrő teljes körű használatát. Figyelembe véve a 2.1 fejezetben leírt követelményeket és a mérések során kapott eredményeket, a következő szempontokat kell a későbbi fejlesztéseknél figyelembe venni:

- Meg kell valósítani a bemeneten az erősítésállítás lehetőségét. Mint a mérések is mutatták, 20 dB-lel alacsonyabb szintű jelnél a szűrés a zárósávban már csak alig nagyobb, mint 50 dB elnyomást biztosít. Ahhoz, hogy a specifikációt tartani lehessen, 20 dB-nél kisebb felbontású erősítésállítást kell alkalmazni. A felbontás mértéke legalább 10 dB, pontosabb esetben 5 dB legyen. Ezzel a specifikáció, helyes erősítés beállítás mellett már teljesíthető.
- Célszerű a hangfrekvenciás tartományt kitolni egészen a DC szintig. Ehhez további törésponti frekvenciák felvétele szükséges. Ezzel a programozható szűrő további felhasználási területeken is alkalmazhatóvá válik. Ilyenek például az orvos biológiai mérések, vagy az igen alacsony frekvenciás geológiai mérések. Ehhez érdemes még megvalósítani a 20 Hz, 10 Hz, esetleg még 5 Hz-es szűrőket.
- A frekvenciasáv ilyen módon való kitolása a kezelőfelület és a program némi megváltoztatását vonja maga után. A beállítható frekvenciákat már nem lehet 3 biten beolvasni, ezért a programnak a szűrőegytárhók cseréjét végrehajtó részét ennek megfelelően módosítani kell. Emellett a MATLAB programba is meg kell adni az új relatív frekvenciaértékeket, ezután a program a szükséges egytárhókat automatikusan generálja.

Azonban, mint a mérési eredmények is bizonyítják, bár a struktúra nagyon kedvező tulajdonságokkal bír, mégis megvannak a megvalósítás korlátai. A 100 Hz alatti szűrők már közel vannak a megvalósítás határához. A számítási pontosság itt már nem minden esetben garantálja a szükséges elnyomást a zárósávban. Ezt a problémát csak egy nagyobb számítási pontossággal rendelkező jelfeldolgozó processzorral lehet feloldani. Érdemes lehet tehát áttérni egy 32 bites lebegőpontos processzorra. Ezeknek, amellett, hogy a számbábrázolásuk jóval pontosabb, a műveletvégzési sebességük is magasabb, mint a 16 bites elődeiké. Ezzel tehát megoldható lenne a kétcsatornás működés, amit az ADSP-2181-nél a szűrők nagy számítási ideje mellett fel kellett adni.

A tanszéken az ilyen irányú fejlesztések már megvan a lehetőség, hiszen rendelkezésre áll már néhány ADSP-21061 SHARC típusú processzorral szerelt fejlesztőkártya és az ezzel végzett hallgatói kísérletek igazolják a nagy pontosságú számítások előnyeit.

Irodalomjegyzék

1. Dr. Simán István: Digitális jelfeldolgozás (főiskolai jegyzet), Budapest, KKMF 1997.
2. Elek Kálmán: Jelfeldolgozó rendszerek tervezése (hallgatói segédlet), BME Híradástechnikai Tanszék
3. Dr. Sujbert László: Programozható digitális szűrők realizálási lehetőségeinek vizsgálata (diplomaterv), BME Villamosmérnöki Kar 1992.
4. Péceli Gábor: Digitális rezonátoros szűrők és transzformációk kapcsolatrendszere Híradástechnika XLII. Évfolyam 1991. június
5. Analog Devices: ADSP-2100 Family User's Manual September 1995.
6. Analog Devices: ADSP-2100 Family EZ-Kit Lite Reference Manual 1995.

Függelék

A CD melléklet tartalma

A CD mellékleten megtalálható a diplomaterv eredményeként elkészült, a programozható szűrőt megvalósító `szuro32.dsp` program és ennek együttthatóit generáló `rezonator_ehk2.m` MATLAB program. Továbbá megtalálhatók ezeknek a programoknak a korábbi változatai és az egyéb kísérleti programok, amelyek a szűrők megvalósítására irányultak.

A `szuro32.dsp` és a `rezonator_ehk2.m` programok

A diplomaterv részét képező DSP program. A program a rezonátorhurkon belül 32 bites számítási pontossággal dolgozik. Az együttthatókat 16 bites formátumban adattömbökbe rendezve várja, amelyet fordítási időben illeszt a programmemóriába. A szűrőegyütthatók tömbjét előállító MATLAB programot tehát a DSP program fordítása előtt le kell futtatni. Ehhez a programban megadott könyvtárnak léteznie kell, különben hibüzenetet kapunk.

A `szuro16.dsp` és a `rezonator_ehk.m` programok

A szűrőprogramnak ez a változata még 16 bites számítási pontossággal dolgozik. Az együttthatók azonban már tömbként vannak a programmemóriában letárolva. A program azonban még nem támogatja a kiegészítő áramkör kezelését. Együttható (törésponti frekvencia) váltás az INTERRUPT nyomógomb segítségével történik. Ennek dupla megnyomása az aluláteresztő és feluláteresztő együttthatók közti váltást valósítja meg. A `rezonator_ehk.m` a DSP program számára állítja elő az együttthatók tömbjét. A program negyedfokú szűrőegyütthatókat generál.

A `rez.dsp`, `rez32.dsp` és a `rezonator.m` programok

A két DSP program a rezonátoros szűrő magját valósítja meg. A szükséges együttthatókat mindkét esetben a MATLAB program generálja, amelyek fordítási időben kerülnek a programmemóriába. A MATLAB programnál meg kell adni a kívánt fokszámot, a törésponti frekvenciát és az approximációt (ehhez a programkódba kell belenyúlni). A `rez.dsp` 16 bites számítási pontossággal, míg a `rez32.dsp` duplapontos számítást alkalmazva valósítja meg a rezonátoros szűrőt.

A `kaszkad2.dsp` és a `kaszkad_ehatok32.m` programok

A DSP program kaszkád struktúrával és belső 32 bites számítási pontossággal valósítja meg a szűrőt. Ehhez a szükséges együttthatókat a MATLAB program generálja.

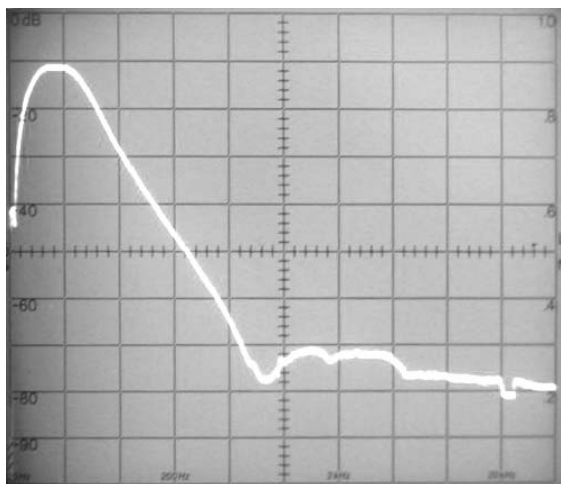
A `szuro_FIR.m`, `szuro_IIR.m` és a `pzabra.m` programok

Ezek a programok a MATLAB-kísérleteknél az átviteli karakterisztika ellenőrzésében és a szűrők tervezésében segítettek.

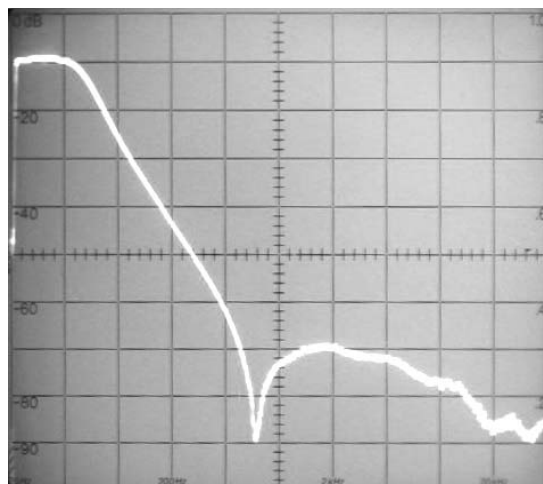
Adatlapok

További mérési eredmények

A mérési eredmények értékelésekor csak a tipikus és a hibák szempontjából lényeges eredmények lettek bemutatva. Itt összehasonlítás képen bemutatásra kerül a negyedfokú szűrőbeállítások mellett felvett néhány amplitúdó karakterisztika.



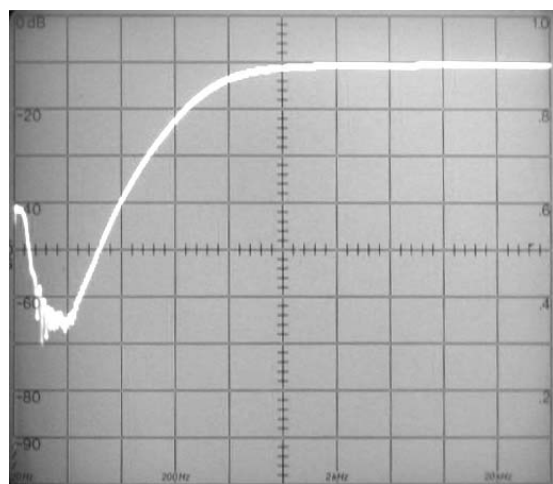
0.1. ábra Negyedfokú, 50 Hz-es aluláteresztő szűrő (függőlegesen 50 Hz/osztás, vízszintesen 10 dB/osztás)



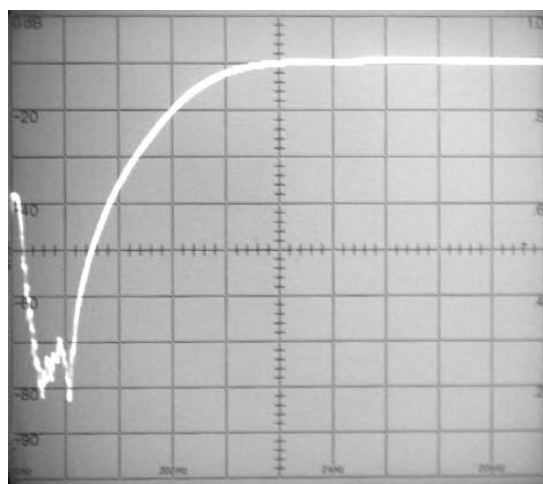
0.2. ábra Negyedfokú, 1 kHz-es aluláteresztő szűrő (függőlegesen 1 kHz/osztás, vízszintesen 10 dB/osztás)

A bemenetre mindkét esetben 4 V csúcsértékű jel volt adva. Látható, hogy az 50 Hz-es negyedfokú szűrőnél már nem jelentkezik az első oldalhullámnál tapasztalható hiba.

A következő két ábra a feluláteresztő szűrők amplitúdó karakterisztikája látható.



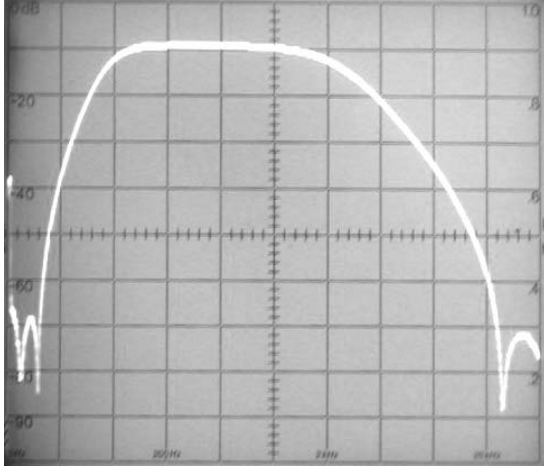
0.3. ábra Negyedfokú, 50 Hz-es feluláteresztő szűrő (függőlegesen 10 Hz/osztás, vízszintesen 10 dB/osztás)



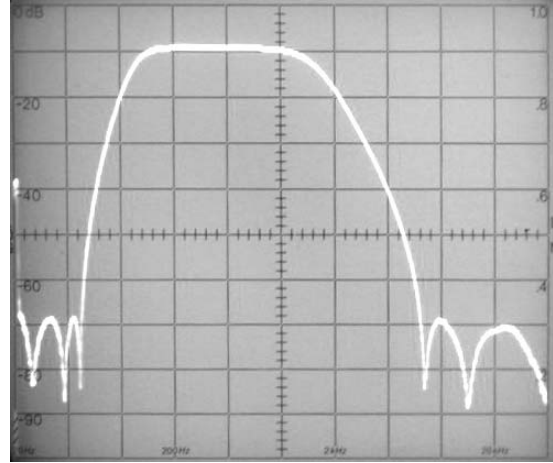
0.4. ábra Negyedfokú 1 kHz-es feluláteresztő szűrő (függőlegesen 200 Hz/osztás, vízszintesen 10 dB/osztás)

Itt a negyedfokú 50 Hz-es aluláteresztő szűrőnél továbbra is tapasztalható a zárósávi hiba, azonban a karakterisztika torzulás már nem jelentkezik. A két karakterisztika, 0.3 és 0.4 ábra jellegre teljesen megegyezik.

A következőkben a szemléletesség kedvéért összehasonlítható a negyedfokú és hatodfokú sávszűrők szelektivitása. Az átviteli karakterisztika a teljes hangfrekvenciás sávot mutatja. A szűrők beállításánál mindkét esetben az alsó törésponti frekvencia értéke 5 kHz, míg a felsőé 10 kHz. A negyedfokú sáváteresztő szűrő a 0.5 ábrán, míg a hatodfokú a 0.6 ábrán látható.



0.5. ábra Negyedfokú sáváteresztő szűrő
(függőlegesen 2 kHz/osztás, vízszintesen
10 dB/osztás)



0.6. ábra Hatodfokú sáváteresztő szűrő
(függőlegesen 2 kHz/osztás, vízszintesen
10 dB/osztás)

A kiegészítő áramkör kapcsolási rajza