



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Méréstechnika és Információs Rendszerek Tanszék

Fizikai alapú dobszintetizátor megvalósítása beágyazott környezetben

DIPLOMATERV

Készítette

Varjasi Tamás

Konzulens

dr. Bank Balázs Lajos

2017. december 13.



DIPLOMATERV-FELADAT

Varjasi Tamás (FRWQWF)
szigorló villamosmérnök hallgató részére

Fizikai alapú dobszintetizátor megvalósítása beágyazott környezetben

Az elektromos gitár esetén alapvetés, hogy a hangzást leginkább a jel útjába kötött torzítók és effektprocesszorok határozzák meg, melyekből számtalan típus elérhető. Természetesen más hangszerek esetén is van lehetőség új, elektronikus hangzások létrehozására, bár az elérhető lehetőségek sokkal korlátozottabbak. Különösen igaz ez az ütőhangszerek esetében.

A kereskedelemben kapható ütőhangszer- és dobszintetizátorok előre felvett hangminták lejátszásán alapulnak, amit egy (tipikusan gumi) ütőfelületre szerelt piezokristály jele alapján indítanak el. Hátrányuk, hogy az ütősjáték finomságai elvesznek, hiszen a piezokristály jelét csak triggerjelként alkalmazzák, ill. a gumi ütőfelületen a játékélmény is korlátozott.

A fenti hátrányok kiküszöbölhetők lennének, ha akusztikus ütőhangszerre szerelt érzékelők jelei alapján vezérelnénk a szintetizátort, ill. ha az érzékelők jelét egy fizikai alapú dobmodell erőbemeneteként használnánk, hiszen ilyenkor nem csupán az ütés erőssége, hanem annak jellege (pl. dobverő, seprű, kézi ütés) is befolyásolná a hangzást. A hallgató feladata egy ennek megfelelő dobszintetizátor megvalósítása, amely képes az akusztikus hangszer hangját gazdagítani, új, pl. elektronikus zenében is használható hangzásokat létrehozni.

A hallgató munkájának a következőkre kell kiterjednie:

- Vizsgálja meg a különböző erő- és rezgésszenzorok működését (piezo kristály, erőérzékeny ellenállás, MEMS gyorsulásérzékelő), a szükséges jelfeldolgozási lépéseket, ill. végezzen ezekkel összehasonlító méréseket!
- A kiválasztott érzékelő felhasználásával készítsen ütőhangszerekre helyezhető érzékelőt, ill. alakítsa ki a szükséges analóg és digitális jelfeldolgozási egységeket!
- Tekintse át a fizikai alapú hangszintézis irodalmát, különös tekintettel a dobok és ütőhangszerek szintézisére!
- Valósítson meg modális alapú ütőhangszermodellt Matlab környezetben, ill. vizsgálja meg, hogyan lehetne a modellbe integrálni az analóg dob-szintetizátorok jelfeldolgozási egységeit az elektronikus hangzás elérése érdekében!
- A Matlabban megvalósított algoritmus számításigénye alapján válasszon megfelelő beágyazott platformot a dobszintetizátor megvalósításához!
- Készítse el a dobszintetizátor prototípusát a kiválasztott beágyazott platformon, és végezzen méréseket az elkészült rendszer késleltetésére vonatkozóan!

Tanszéki konzulens: Dr. Bank Balázs, docens

Budapest, 2017. március 10.

.....
Dr. Dabóczy Tamás
tanszékvezető

Tartalomjegyzék

Kivonat	6
Abstract	7
1. Bevezető	8
1.1. Elektronikus hangszerek fejlődése	8
1.2. A feladat bemutatása	9
1.2.1. Rendszer áttekintése	10
1.3. Piaci körkép	11
1.3.1. Elektronikus dobok	11
1.3.2. Elektronikus cajon	11
1.3.3. Szoftver hangszerek	12
1.4. A dolgozat felépítése	15
2. Hangszintézis eljárások	16
2.1. Előre felvett hangminta visszajátszáson alapuló eljárások	16
2.1.1. Hangminta alapú hangszintézis	16
2.1.2. Hullámtábla alapú szintézis	17
2.2. Frekvenciatartománybeli modell alapú eljárások	18
2.2.1. Additív szintézis	18
2.2.2. Szubtraktív szintézis	18
2.3. Fizikai alapú szintézis	21
2.3.1. Digitális hullámvezető	22
2.3.2. Véges differenciák módszere	23
2.4. Absztrakt algoritmusok	23
2.4.1. FM szintézis	23
3. Modális hangszintézis	25
3.1. Mechanikai rendszer leírása	25
3.2. Modális megoldás	26
3.2.1. Módusok	26
3.3. Modális megoldás alapú jelszintézis	27
3.4. Kör alakú membrán módusai	28
3.4.1. Az ideális húr mozgásegyenlete	28

3.4.2.	Az ideális membrán mozgásegyenlete	29
3.4.3.	A membránegyenlet megoldása, módusok meghatározása	30
3.4.4.	Membrán-feszültség moduláció	31
3.5.	Szintézis struktúra	32
4.	Szenzorválasztás	34
4.1.	Piezokristály	34
4.2.	FSR	35
4.3.	MEMS technológiás szenzorok	37
4.4.	Szenzorokon végzett mérések	38
4.5.	Összefoglalás	39
5.	Hangszintézis megvalósítása modellező környezetben	40
5.1.	Előfeldolgozás	40
5.1.1.	Alul- és felüláteresztő szűrő	41
5.1.2.	Burkológörbe generálás	41
5.1.3.	Küszöbölés	41
5.1.4.	Simítás	41
5.2.	Rezonátorbank megvalósítása	43
5.3.	Pergőmodell	45
5.4.	Lábdobmodell	45
6.	Dobszintetizátor megvalósítása beágyazott környezetben	47
6.1.	Felhasznált hardverek	47
6.1.1.	DSP kártya	47
6.1.2.	Kezelőfelület	51
6.1.3.	Fejlesztéshez használt mérési elrendezés	51
6.2.	Beágyazott szoftver magasszintű működésének áttekintése	52
6.3.	Szoftvermodulok működése	53
6.3.1.	Hardver inicializálás	53
6.3.2.	Szintézis alkalmazás	54
7.	Beágyazott rendszer kezelőfelülete	59
7.1.	Szoftverterv magasszintű áttekintése	60
7.2.	Szoftvermodulok működése	61
8.	Beágyazott rendszer vizsgálata	63
8.1.	Rezonátorbank vizsgálata	63
8.1.1.	Pergőmodell vizsgálata	64
8.2.	A lábdobmodell vizsgálata	65
8.3.	A dobszintetizátor késleltetésének vizsgálata	68
9.	Összefoglalás	70
9.1.	Értékelés	70

9.2. Továbbfejlesztési lehetőségek	71
Ábrák jegyzéke	73
Irodalomjegyzék	76

HALLGATÓI NYILATKOZAT

Alulírott *Varjasi Tamás*, szigorló hallgató kijelentem, hogy ezt a diplomatervet meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Budapest, 2017. december 13.

Varjasi Tamás
hallgató

Kivonat

Az elektronikus hangszerek hangja egyre meghatározóbb a modern zenében, akár új zenei stílusok alapját is képezheti. A dolgozatomban egy elektronikus dobhang keltésére alkalmas eszközt terveztem és valósítottam meg beágyazott környezetben.

A ma kapható dobszintetizátorok működése többnyire előre felvett hangminta visszajátzáson alapul. A minta lejátszását az ütőfelületre szerelt szenzorok jele alapján generált triggeresemények indítják, amelynek nagy hátránya, hogy a hangszeres játék finomságai eltűnnek a generált jelből.

Erre megoldást jelenthet egy olyan fizikai alapú dobmodell megalkotása, ahol a hangszintézist nem triggeresemény indítja, hanem a szenzor jelét a dobmodell erőbemeneteként használjuk. Ennek köszönhetően nem csak az ütés erőssége, hanem a jellege is megjelenik a szintetizált jelben.

A dolgozatomban egy olyan dobszintetizátort valósítottam meg, mely képes egy akusztikus ütőhangszerre szerelt piezokristály kimeneti jele alapján olyan dobhangzást szintetizálni, amiben benne maradnak a hangszeres játék részletei. Akusztikus ütőhangszernek az afroperui eredetű, egyre népszerűbb cajont választottam.

A munkám részeként megvizsgáltam több eltérő technológián alapuló szenzort, hogy eldöntsem melyik a legalkalmasabb a fizikai alapú dobmodell gerjesztésére. Megvizsgáltam különböző tulajdonságokkal rendelkező piezo kristályokat, egy erőérzékeny ellenállást és tanulmányoztam a MEMS mikrofont és gyorsulásmérőt.

A szintézis alapegységeit először Matlabban valósítottam meg. A cajon jellegéből fakadóan egy mélyebb és egy magasabb ütés megszólaltatására alkalmas, ennek megfelelően egy lábdob- és egy pergőmodellt hoztam létre. A fizikai dobmodell a kör alakú membránra felírt mozgásegyenlet modális megoldásán alapul.

A modelleket saját építésű cajon-ba szerelt szenzorok jeleivel teszteltem. Ez alapján kidolgoztam egy előfeldolgozási lépést, mely a beérkező jelek frekvenciatartománybeli elemzése alapján eldönti, hogy a beérkező jel melyik modelleket gerjessze.

A modellező környezetben szerzett tapasztalatok alapján az Analog Devices SHARC ADSP-21489 lebegőpontos architektúrájú jelprocesszorát választottam a beágyazott környezetben való megvalósításra, valamint egy érintőképernyő-alapú vezérlő teszi lehetővé a szintézis paraméterek módosítását valós időben.

Abstract

The sound of electronic music instruments plays more and more dominant role in the modern music. Many musical genres are based on special electronic sound. In my diploma thesis, I have designed and realized a drum synthesizer in embedded environment, which is able to generate electrical drum sounds.

The drum synthesizers available today are mostly based on pre-recorded sound sample playback. This playback is triggered by events generated by the sensor mounted on the electrical drum surface. It has a large disadvantage that the details of the playing disappear from the generated signal.

In my diploma thesis I have implemented a drum synthesizer which keeps the details of the playing in the generated output. My selected musical instrument was a cajon which has an Afro-Peruvian origin. Thus my solution is not only an electrical drum surface, it is an acoustic instrument. The synthesizer extends the acoustic instrument sound with electronic drum sounds.

I have analyzed sensors based on different technologies to decide which is the best for exciting the physics-based models. I have analyzed different piezo sensors, a force sensitive resistor, electromechanical microphone and an acceleration meter.

First of all, I have implemented the base units of the synthesis in Matlab. I have designed two models: one for lower sounds (like a big drum) and another one for higher sounds (like a snare drum) based on the typical cajon sounds. The basis of the physics-based drum models is the modal solution of the partial differential equation that describes the vibration of a circular membrane.

I have tested the models with signals of sensors which are mounted on my self-made cajon. Based on these tests, I have designed a preprocessing step to make a signal routing between the models according to the behavior of input signal in the frequency domain.

Based on the results of the testing in the modeling environment, I have implemented the drum sound synthesis to the floating-point Analog Devices SHARC ADSP-21489 digital signal processor and I made a touch screen based solution for the real-time parameter settings.

1. fejezet

Bevezető

1.1. Elektronikus hangszerek fejlődése

A manapság tapasztalható technológiai innováció folyamatosan formálja életünket, kultúránkat. Így van ez a zenehallgatással és a zenéléssel is. A hallgatóság és a zeneipar is érdekelt abban, hogy új stílusok, új hangzások jöjjenek létre, az újdonság kíváncsiságot válthat ki, új típusú produkciók születhetnek.

Ha visszatekintünk az elektronikus hangszerek megjelenéséig, akkor megállapíthatjuk, hogy a legelső elektronikus hangszerek technológiájukat tekintve az elektroncsőre és elektromechanikus rendszerekre alapulnak [4]. Az egyik legelső például az 1929-ben megjelent Németországban kifejlesztett Trautonium [23], vagy az 1935-ben megjelent Amerikában tervezett Hammond-orgona [13]. Az utóbbi jó példa az elektromechanikus hangkeltésre, ugyanis a Hammond-orgonában a hangkeltésről elektromos motor által forgatott szinuszevolvens fogprofilal ellátott lágyvas kerék gondoskodik, amelyek mellé helyezett elektromos hangszedőn jelenik meg az elektromos hangjel. Fontos szerepet játszott még a Hammond-orgona hangjában az un. Leslie hangszóró, ahol a jellegzetes hatást a doppler jelenség okozza a mechanikusan forgatott hangtölcsér segítségével.

Nagy ugrást jelentett az elektronikus hangszerek megjelenésében a félvezető-technológia megjelenése, ekkor jelentek meg az első analóg szintetizátorok. Ez a technológia lehetővé tette olyan modulok kifejlesztését, mint a feszültség vezérelt oszcillátor (VCO), feszültség vezérelt szűrő (VCF) és feszültség vezérelt erősítő (VCA) valamint a vezérlő feszültségeket létrehozó burkológörbe generátorok (EG). Eleinte ezek egymástól teljesen független részek voltak, amiket - az akkori telefonközpontokban elterjedt - jack kábelek segítségével lehetett tetszőleges kombinációban összekötni. A hangszeres játék itt már a zongoránál megszokott billentyűzeten történt, ez állította elő a hangmagasságot vezérlő feszültséget (Control Voltage - CV). A modularitás miatt ezeket a szintetizátorokat moduláris szintetizátoroknak hívták [15]. Ilyen termékeket gyártott pl. a Moog Music, ARP Instruments, Inc. vagy a Buchla & Associates.

A dolgozat fő témáját adó dobszintetizátorok története az 1920-as évekig nyúlik vissza. A legelső dobgépek szintén elektromechanikus és elektroncsöves működésűek voltak. Az első kereskedelmi forgalomban is kapható dobgép 1959-ben jelent meg (Wurlitzer Side Man)

[1], ahol a hangkeltés már analóg elektronikus módon történt és képes volt előre beállított dobritmusokat lejátszani. Ezek egy - a ritmusnak megfelelő pozícióban elhelyezett - érintkezőkkel ellátott lemezen voltak, melyet egy kefése érintkezőkkel ellátott elektromos motorral meghajtott forgó kar olvasott le. A kar forgási sebessége határozza meg a lejátszott ritmus tempóját.

1972-ben jelent meg az első olyan dobgép (Eko ComputeRhythm), amely nem csak előre bekonfigurált ritmusokat tudott lejátszani, hanem gombsorok segítségével programozható volt.

Az igazán nagy áttörést az 1980-as év hozott, ugyanis ekkor jelent meg az első digitális hangminta-visszajátszáson alapuló dobgép (Linn LM-1). Annak ellenére, hogy összesen 12 db 28 kHz-en mintavételezett 8-biten kvantált hangmintát tárolt, meghatározó volt a zeneiparban.

Ugyancsak ebben az évben jelent meg a Roland márkanévvel viselő TR-808, ami az első igazán közkezdeltté vált dobszintetizátor. Ez az eszköz a moduláris szintetizátorokban elterjedt komponensek segítségével (szubtraktív hangszintézis) állít elő perkusszív jellegű hangokat (lábdob, pergő, tamok), melyek hangzása alapja annak, amit ma modern elektronikus zenének hívunk. Nem sokkal később megjelent a TR-909 is, ami az előző modell továbbfejlesztett változata. Mindkét változat rendelkezett programozható szekvenszerrel is, aminek a segítségével tetszőleges ritmust tudunk megszólaltatni vele. A jelentőségét jól jelzi az a tény, hogy egy újragondolt, teljesen digitális alapon, de ugyanazzal a hangzással és funkcionalitással bíró változata ma is kapható (Roland TR-09 Boutique).



1.1. ábra. *Roland TR-909*

1.2. A feladat bemutatása

A munkám célja egy olyan - akár színpadon is használható - dobszintetizátor megtervezése, amely alkalmas valós időben egy piezo érzékelőkkel felszerelt cajon hangját felhasználva dobhangok előállítására, mely kiegészíti a hangszer akusztikus hangját. Az alapötlet az, hogy az ütőhangszer által szolgáltatott jel ne csupán triggereseményt és ehhez tartozó dinamikai információt adjon, hanem hozzunk létre egy fizikai alapú dobmodellt, aminek ez a jel képezze a bemenetét, gerjesztését, így lehetőség adódik arra, hogy a hangszerjáték apró finomságai mind érvényesüljenek a szintetizált jelben.

A munkám során ütőhangszer gyanánt egy cajont választottam. A cajon (kiejtése: "ká-hon") egy afroperui eredetű, fából készült ütőhangszer, leginkább egy fadobozhoz hasonlít.

Ezt a hangszer a zenész a tetején ülve szólaltatja meg [8]. A leglényegesebb része egy rétegelt lemezből készült rezonáns lap, amire - a pergődobnál is használatos - sodrony van feszítve. Fontos megemlíteni még, hogy alapvetően kétféle tónusú hang kelthető vele, az előlapot középen ütve mélytónusú, fent ütve pedig magas tónusú. A zenében a mély ütésnek van lábdob szerepe, míg a felső ütés inkább pergőszerű, ami részben annak is köszönhető, hogy a felső ütésnél dominánsabb a sodrony. A 1.2. ábrán látható példaképpen egy Schlagwerk cég által gyártott cajon.

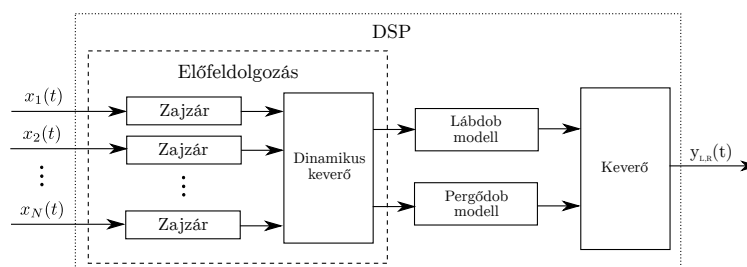


1.2. ábra. Cajon [9]

1.2.1. Rendszer áttekintése

Az általam megtervezett és megvalósított hangszer alapegységei a következő módon kapcsolódnak össze:

A bemeneti jeleket egy piezo szenzorokkal felszerelt cajon szolgáltatja. Ahhoz, hogy pontosabban tudjuk működtetni a hangmodult, több ponton is elhelyezhetünk érzékelőket, ami segít abban, hogy meghatározzuk az ütés pozícióját. Ezek több csatornán érkezik egy DSP kártyára, amin a szintézishez szükséges modell fut. A kimenet pedig egy sztereó hangjel, illetve a cél az, hogy a szintetizált jel kiegészítse a hangszer eredeti hangját.



1.3. ábra. Rendszerterv áttekintés

A 1.3. ábra szerint a cajonra szerelt érzékelők jelei első lépésben át vannak vezetve egy zajzáron. Abban különbözik ez az általános zajzártól, hogy nem a közvetlen jel hangereje alapján működik, hanem egy megfelelő frekvenciatartományra korlátozott jel erőssége

alapján vezérli a szüretlen jelet. Ez segíti a következő fokozatot, ahol eldől, melyik modellbe vezessük be a jelet. Optimális esetben a felső pozícióban történt ütések a pergődob modellt, a mélyütések pedig a lábdob modellt gerjesztik. Ha például egy mély ütéset szeretnénk detektálni, akkor alapvetően a jel mélyfrekvenciás tartománya alapján hozzuk meg a döntést. Természetesen az a lehetőség is fenn áll, hogy egy időben mindkét helyen megüssük a hangszer, ezt is megfelelően kell detektálni.

Hangsúlyozandó még, hogy a 1.3. ábra logikai kapcsolatokat jelöl a fokozatok között. Az implementációban lehetséges, hogy egy későbbi fokozat egy korábbiak nem csak a kimenőjelét, hanem belső állapotát is felhasználja a működéséhez.

1.3. Piaci körkép

Az alábbi pontokban megvizsgálom azokat a termékeket, termékkategóriákat, melyek működése vagy funkcionalitása hasonlít a dolgozat során elkészített dobszintetizátorhoz.

1.3.1. Elektronikus dobok

Az elektronikus dobok piacán az ezredforduló környékén kezdett elterjedni egy új termék-vonal, aminek a célja az volt, hogy a könnyűzenében elterjedt akusztikus dobfelszerelést modellezze, mind hangzás, mind játékélmény tekintetében. Jelenleg a piacon kapható, ebbe a kategóriába sorolható elektronikus dobszettek alapvetően triggeresemény hatására, hangminta visszajátszás alapján működnek. A mai beágyazott processzoroknak már nem kihívás, hogy elegendő mennyiségű memóriával és feldolgozó kapacitással rendelkezzenek ahhoz, hogy emberi érzékeléshez viszonyítva késleltetés nélkül, magas minőségű hangzást állítsanak elő. Ennek a technológiának a legnagyobb hátránya, hogy a bemeneti jelet kizárólag csúsdetektálásra és ütéserősség meghatározására használja, így a hangszeres játék finomságát nagyban eltünteti. Egyes drágább termékek alkalmasak az ütési pozíció becsülésére is, ami javítja a játékélményt és a hangzást is közelebb hozza az eredeti dobfelszereléshez. Ilyen pl. a Roland V-Drums TD-50.

Korg Wavedrum

A Korg által kifejlesztett elektromos dob (Wavedrum) rendelkezik egy szenzorokkal ellátott ütőfelülettel és hibrid szintézis technológiát használ. A feldolgozott szenzorjelek 36 féle fizikai modell algoritmusnak szolgálnak erőbemenetéül, de 200 előre letárolt hangminta megszólaltatására is alkalmas [21]. A gyártó nem fedi fel, hogy a fizikai modellben milyen tényleges megvalósítási módszert választott. A hangszer csak elektronikus hangkeltésre alkalmas, saját akusztikus hangja nincs, illetve nem játszik szerepet a hangzásban. A 1.4. ábrán megfigyelhető a kereskedelemben is kapható hangszer.

1.3.2. Elektronikus cajon

Ugyancsak a Roland hangszergyártó cég kifejlesztett egy modult, ami vagy egy cajon-ba integrálva vagy külön egységként a cajon mellett működik. A termékleírásból kiderül, hogy



1.4. ábra. Korg Wavedrum [21]

a modul egy hangbemenettel rendelkezik, ebből a jelből állapítja meg, hogy történt-e ütés, és ha igen akkor mély vagy magas pozícióban. Az ütés esemény hatására előre letárolt hangminta visszajátszást végez. További funkció, hogy a hangszerbe épített hangszórón szólal meg a hang. A termék neve (Electronic Layered Cajon) arra utal, hogy a cajon akusztikus hangja és a generált hang együtt képzik a hangszer hangját, tehát ebben megegyezik azzal a megközelítéssel amit a dobszintetizátor fejlesztése közben követtem.



1.5. ábra. Roland El Cajon EC-10 [20]

1.3.3. Szoftver hangszerek

Fontos megemlíteni, hogy nem csak a fizikai termékek jelenthetnek konkurenciát, ugyanis nagyon nagy számban léteznek csupán szoftveres megoldások is hangszintézisre.

Native Instruments Reaktor Prism Fx

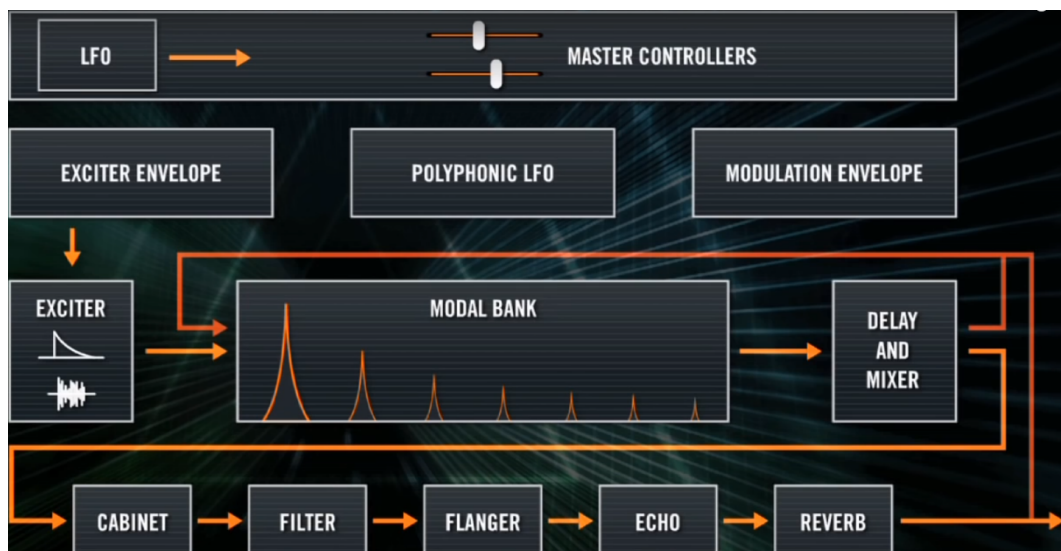
A stúdiókban, illetve otthoni produceri zeneszerkesztéshez használt hangszerkesztő szoftverek, avagy digitális audió munkaállomások (Digital Audio Workstation - DAW) segítségével készülnek azok felvételek amelyekből zeneipari termék lesz. Ilyen szoftverek többek között: Avid Pro Tools, Steinberg Cubase, Logic, stb. Mindegyik ilyen DAW rendelkezik olyan funkcióval, hogy képes külső fejlesztők által készített, megfelelő formátumú szoftvermodulok betöltésére (VST, AudioUnit, AAX, stb), melyek szoftverhangszereket vagy effektet valósítanak meg. Az effekt audio ki és bemenettel rendelkezik, a hangszerek pedig MIDI bemenettel (ami a hangszeres játékot reprezentálja) és audió kimenettel.

A Native Instruments audió plug-in-eket készítő cég egyik modulja a Reaktor Prism FX.

A bemutatott termékek közül az ebben implementált szoftver hasonlít leginkább az általam fejlesztett eszközre, ugyanis a működésének ugyanúgy a modális hangszintézis az alapja. A nagy különbség viszont az, hogy a Prism FX inkább speciális hangeffektek létrehozására használható, nem célja elektronikus dobhangok keltése.



1.6. ábra. Native Instruments Reaktor Prism FX [18]



1.7. ábra. Native Instruments Reaktor Prism FX blokk diagram [18]

A 1.7. ábrán láthatjuk a szoftvermodul belső felépítését. A modulba érkező audió jel először az "exciter envelope" feldolgozási egységbe kerül ami két feladatot lát el. Alapvetően egy zajzár, ami csak egy meghatározható bemeneti szint felett engedi át a jelet, másrészt generálja a bejövő jel burkolóját is. Az így létrehozott vezérlőjelet felhasználhatjuk a teljes beépülő modul bármilyen módosítható paraméterének vezérléséhez.

A következő lépésben az "Exciter" nevű blokkban a bejövő jel a dinamikai paraméterein változtathatunk, hozzákeverhetünk zajt vagy többféle szélessávú audió jelet. Erre azért lehet szükség, hogy a szintetizált hangjel is gazdagabb legyen, az eredetnél több frekvenciakomponens jelenjen meg benne.

A "modal bank" lépés az, ami a tényleges hangszintetizálást végzi. Egy ehhez hasonló szoftvermodult a munkám során implementáltam, ezt a dolgozatban részletesen bemutatom.

A Native Instruments szoftvere lehetőséget nyújt még, hogy a generált jelet különböző effektekkel tovább módosítsuk. Ezeknek a tetszőleges beállításával vagy automatikus vezérlésével teljesen meg is változhat a hangzás jellege. Érdekesség még, hogy ezeknek az effektek kicsit máshogy hatnak a bal illetve a jobb kimeneti csatornára és az így létrejött dekorreláció által kiszélesedik a hangzás sztereó képe.

Mogees

A virtuális hangszerek, szoftver szintetizátorok futtatására manapság már az okostelefonok vagy tabletek is alkalmasak. Ezt használja ki a Mogees megoldása is, akik fejlesztettek egy piezo technológiájú, bármire rátehető szenzort, mely egy okos mobil eszközhez van csatlakoztatva.

A működés alapja az, hogy egy tetszőleges tárgyra rárögzítjük a szenzort (pl.: gördeszkára, serpenyőre, drótkerítésre, hajópadlóra, stb. - a cég reklámanyagában szereplő példák szerint), majd a tárgyra mért ütések által keltett vibrációt a szenzor érzékeli és a mobil eszközön futó szoftver fizikai modell alapú szintézis segítségével ez alapján végzi a hangkeltést. Van a használat előtt egy tanulófázis, ahol megtaníthatjuk a szoftvernek, hogy a tetszőleges tárgy különböző pontjait megütve milyen típusú hangzást szeretnénk (pl.: a serpenyő főzőlapja más hangot adhat, mint ha a nyelét ütjük meg) [16].

Kitekintésképpen megjegyzem, hogy a Londonban működő start-up cég audió alapú gesztus felismerő technológiáját (gesture recognition) egyre többször használják nem zenei céllal is, lényegében bármilyen elektronikus eszköz ember-gép interfészeként szolgálhat. (pl.: asztalra koppintással felkapcsolhatunk egy lámpát.)

Impaktor

Ugyancsak okostelefonos alkalmazásként elérhető a BeepStreet által fejlesztett Impaktor, melyben megvalósított hangszintézis teljesen fizikai modellezésen alapul. Nagyon sok előreprogramozott hangzás érhető el (pl.: Conga, Hang drum, stb.), melyek valódi perkusszív hangszer modellezésén alapulnak. A legtöbb paraméter valós időben változtatható [6].

Ez abban hasonlít a megvalósított dobszintetizátorra, hogy a fizikai modellt a mikrofon bemenete alapján gerjeszti. A tapasztalatom szerint ennek ellenére kevés bemeneti információ határozza meg a szintézist, valószínűleg a mikrofon jelét triggerként használja.



1.8. ábra. *Bögrére rögzített Mogeas szenzor [17]*

1.4. A dolgozat felépítése

A bevezetőben szó volt az elektronikus hangszerek megjelenéséről és szerepéről a modern zenében, valamint bemutatásra került a dobszintetizátor alapötlete és működése, valamint a piacon jelenleg is kapható termékek, melyek közel állnak a feladathoz.

A további fejezetekben szó lesz az alapvető hangszintézis eljárásokról, kiemelve a modális hangszintézist, mivel a kifejlesztett hangszer működése ezen alapul. Bemutatom a tervezőmunka során kipróbált és tesztelt szenzorokat, ezután pedig a beágyazott rendszer építőelemeit. Ezek után részletesen fogom tárgyalni az implementált megoldásomat modellező és beágyazott környezetben, a dolgozat végén pedig sor kerül a megvalósítás értékelésére is.

2. fejezet

Hangszintézis eljárások

Ebben a fejezetben szeretnék áttekintést adni azokról a hangszintézis-módszerekről, amelyeknek a leginkább szerepe lesz a fejlesztésben.

Az analóg szintetizátorok megjelenése óta rengeteg hangszintézis-módszert találtak ki, melyek a legkülönbözőbb célokra használatosak (pl. emberi beszéd szintézis, akusztikus hangszer modellezése, stb.). Itt a hangszintézis-eljárásokat abból a szempontból is vizsgálni fogom, hogy mennyire alkalmasak perkusszív hangzások előállítására.

A mai modern hangszereknél legtöbbször nem kifejezetten egy módszert alkalmaznak a meglévők közül, hanem rendszerint ötvözik egymással őket, ezért érdemes minél többet megérteni.

Hangszintézis eljárások főbb típusai

Sokféle megközelítés alapján lehet csoportosítani a zenei hangszintézis típusokat, a legismertebbek egy lehetséges felosztása a következő [34]:

- **Előre felvett hangminta visszajátszáson alapuló:** Hullámtábla alapú, Hangminta alapú
- **Frekvenciatartománybeli modell alapú:** Additív, szubtraktív
- **Fizikai alapú:** Digitális hullámvezető, véges differenciák módszere, modális hangszintézis
- **Absztrakt algoritmusok:** Modulációs szintézisek, FM szintézis

2.1. Előre felvett hangminta visszajátszáson alapuló eljárások

2.1.1. Hangminta alapú hangszintézis

A hangminta visszajátszás alapú megoldás a legegyszerűbb módszerek közé tartozik. Működésének alapja, hogy a modellezendő hangszer akár minden lehetséges hangjáról felvételt készítünk, majd egy triggeresemény hatására (pl.: zongorabillentyű leütése) visszajátsszuk. A mai hangszerekben ezt úgy alkalmazzák, hogy a triggeresemény hatására elindul a hangminta lejátszás, majd a kezdeti tranziens jelenségek után exponenciálisan csökkenő bur-

kolóval egy rövid részletet folyamatosan ismételnék. A billentyű elengedését jelző esemény bekövetkezése után az ismételt részletet megszakítva egy lecsengő részlet következik, ami lekeveri a hangmintát.

A módszer előnye, hogy ha elég részletesen felvesszük a hangszer hangját, akkor nagyon élethű modell érhető el. Felvehetjük a hangszer összes lehetséges hangmagasságát, többféle erősségű játékmódban, alternatív hangmintákkal, amik közül a lejátszás során véletlen módon választunk, akár sztereóban, vagy több csatornán. A módszer nagy hátránya, hogy egy-egy jó virtuális hangszer tárhelyigénye akár több gigabájt is lehet, amit teljes egészében a lejátszás közben a gyors elérésű memóriában kell taranni.

A hangminta alapú szintézis perkusszív hangszereknél hatékonyan alkalmazható

2.1.2. Hullámtábla alapú szintézis

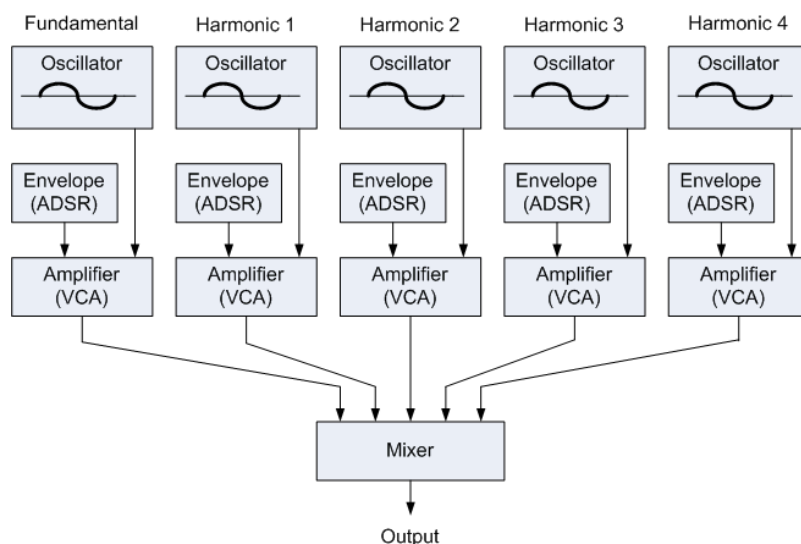
A módszer alapja az, hogy a megszólaltatni kívánt periodikus jelből letárolunk a memóriába legalább egy teljes periódust egy adott frekvencián, majd ezt a kívánt frekvenciához tartozó sebességgel kiolvassuk. Az esetek döntő többségében olyan fázishelyzetben lévő adatpontra lenne szükségünk, ami nincs letárolva, így szükségessé válik valamilyen interpoláció használata. Előnye, hogy egyszerűen megvalósítható, hátránya, hogy a szintetizálandó jel nem paraméterezhető. Mivel minden hang megszólaltatása esetén jelfeldolgozási szempontból decimálást vagy interpolálást végzünk, ezért felléphetnek átlapolódási jelenségek, aminek az elkerüléséhez átlapolódásgátló szűrők megfelelő használata szükséges.

Létezik olyan változata is, amikor a hangminta visszajátszás alapú hangkeltéssel ötvözik. Ha nem tárolunk el minden megszólaltatni kívánt frekvencián egy-egy hangmintát, hanem pl. csak 5-10 hangonként, akkor a hullámtábla alapú szintézis segítségével lecsökkenthetjük a szükséges hangminta adatbázis tárhely szükségletét. Hátránya, hogy az egymásmelletti hangok így jellegben azonosak lesznek, a módszer csökkenti a hangkeltés változatosságát.

2.2. Frekvenciatartománybeli modell alapú eljárások

2.2.1. Additív szintézis

Ez az egyik olyan módszer, amely a szintetizálendő jel spektrális leírásán alapul. Az eljárás lényege az, hogy a Fourier együtthatók segítségével jellemzett jelet párhuzamosan kötött és a megfelelő frekvenciákra hangolt szinuszos oszcillátorok segítségével állítja elő. Bemeneti paramétere lényegében a generálendő jel spektruma, az egyes frekvencia komponensek időfüggése, valamint a harmonikusok időbeli viselkedése, lecsengése. Ez a módszer periodikus jelek előállítására a legkézenfekvőbb (pl. orgona), nagy belépő tranzienssel és széles spektrumú komponensek esetén nem jól használható. Előnye, hogy egyes szintézis eljárásokhoz képest könnyen implementálható, hátránya a hatalmas paraméterkészlet, aminek a helyes beállítása nem egyszerű feladat. Az 2.1. ábrán látható a szintézis egy lehetséges struktúrája [35], [40].



2.1. ábra. Additív szintézis [2]

2.2.2. Szubtraktív szintézis

Az analóg szintetizátorok által leginkább használt módszer. A szubtraktív szintézis jó példa arra, mikor egy hangszer által gerjesztett jelet intuitív módon próbáljuk reprodukálni. Az eljárás alapja, hogy kiindulunk valamilyen széles spektrumú jelből (pl. négyszögjel, háromszögjel, fűrészjel, akár fehér zaj), majd rezonáns szűrőtagokkal ezt tovább formáljuk. Gyakori, hogy több oszcillátor jelét párhuzamosan használják fel, akár panorámázva, azaz előállítják a sztereó hangképet is. Megfelelően paraméterezve ez a módszer már alkalmas lehet perkusszív jellegű hangok előállítására, de fontos szempont, hogy az eredeti oszcillátorok frekvenciája is állítható legyen tetszőleges burkológörbe alapján [40].

A szubtraktív szintézist használó hangszerek nagyon elterjedtek a modern zenei hangzásokban is. Ez annak köszönhető, hogy sok modern digitális szintetizátorban vagy virtuális hangszerben visszaköszön ugyanaz a szubtraktív szintézis struktúra, mint az eredeti analóg

rendszerű hangszerekben. Azokat a szintetizátorokat, amelyek az analóg eljárást próbálják meg modellezni, virtuál-analóg szintetizátoroknak hívjuk. Ezeknek a hangszereknek nagy előnye, hogy amellet, hogy rendelkeznek az analóg hangzás keltésének képességével, a digitális funkciók még sokszínűbb használatot tesznek lehetővé (pl.: beállítások elmentése, visszahívása, MIDI vezérlés, digitális hangrögzítés, stb.).

A következő pontokban bemutatok a szubtraktív szintézisre néhány példát a Roland TR-909 dobszintetizátor alapján. A bemutatott megvalósítás teljes mértékben analóg összetevőkből épül fel.

Az ábrákon látható struktúra elemei:

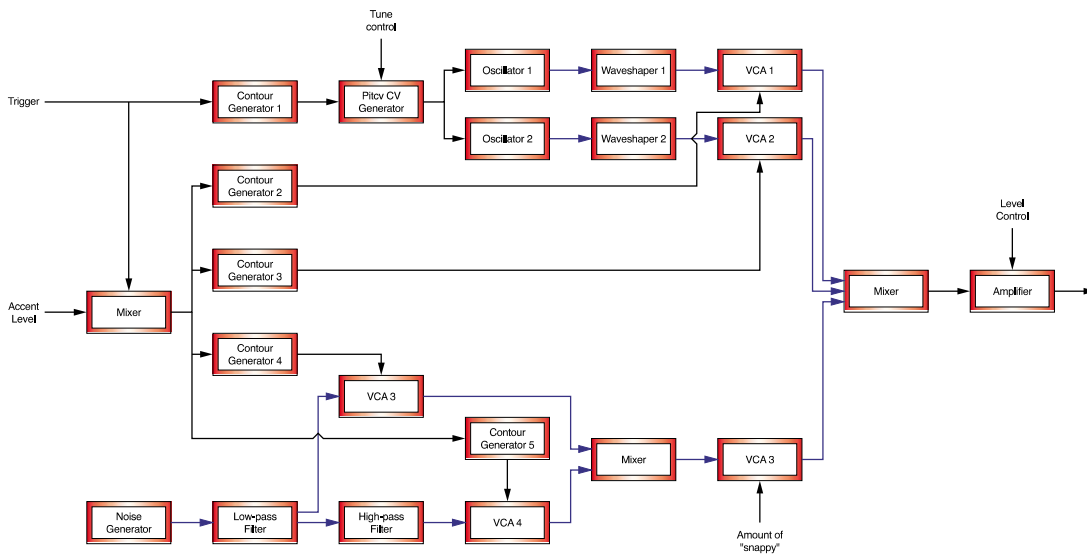
- **Trigger:** Külső forrásból származó analóg vezérlő jel, mely jelzi a dobütés kezdetét.
- **Accent Level:** A szintetizált hangjel erősségét határozza meg.
- **Noise Generator:** Fehérzaj forrás.
- **Contour Generator:** Vezérlőfeszültség generátor, legtöbbször valamilyen T időállandóval lecsengő exponenciális. Feszültségvetérelt erősítőre kötve a jel burkolóját határozza meg. (**EG - Envelope Generator**)
- **Low-pass Filter:** Aluláteresztő szűrő.
- **High-pass Filter:** Feluláteresztő szűrő.
- **Waveshaper:** Aluláteresztő szűrő.
- **VCA:** Voltage Controlled Amplifier - Feszültségvezérelt erősítő.
- **Oscillator, VCO:** Audió jelforrás, a jelalak lehet szinusz, háromszög, négyszög, stb. A bemeneti vezérlő feszültség határozza meg a frekvenciát.
- **Pitch CV generator:** Hangmagasság vezérlő feszültség generátor.
- **Mixer:** Keverő.
- **Pulse generátor:** Impulzus generátor.

Roland TR-909 pergő hang

A 2.2. ábrán látható a TR-909 dobszintetizátor pergődob moduljának struktúrája.

Az ábrán két bemenetet láthatunk. Az egyik a trigger impulzus, aminek határára a burkológenerátorok működésbe lépnek, a másik pedig az ütéseerősséget adja meg. A modul két fő jelútból áll. A felső jelút két háromszög generátort tartalmaz, az alapfrekvenciájuk 180 Hz és 330 Hz. A jelgenerátorok kimenetére két aluláteresztő szűrő van kötve, amik lecsökkentik a magasfrekvenciás összetevők arányát, majd két feszültségvezérelt erősítő segítségével válik a jel perkusszív jellegűvé. A két jelgenerátor modellezi a pergődob alsó és felső membránját, frekvenciájuk időbeli lefutását a 1-es vezérlőfeszültség-generátor kimenőjele határozza meg. A két membrán modellezéséből adódó tonális jelek időbeli lefutását a 2-es és 3-as vezérlőfeszültség generátor határozza meg egymástól függetlenül.

Az alsó jelút lehetőséget ad arra, hogy a generált jelhez zajkomponenst is keverjünk. A zajgenerátor kimenete először egy aluláteresztő, majd egy feluláteresztő szűrőn kerül átvezetésre. Az aluláteresztő szűrő és a feluláteresztő szűrő kimenete külön időbeli lefutással kerülnek a generált jelbe. Ezeket a lefutásokat a 4-es és az 5-ös vezérlőfeszültség határozza meg.

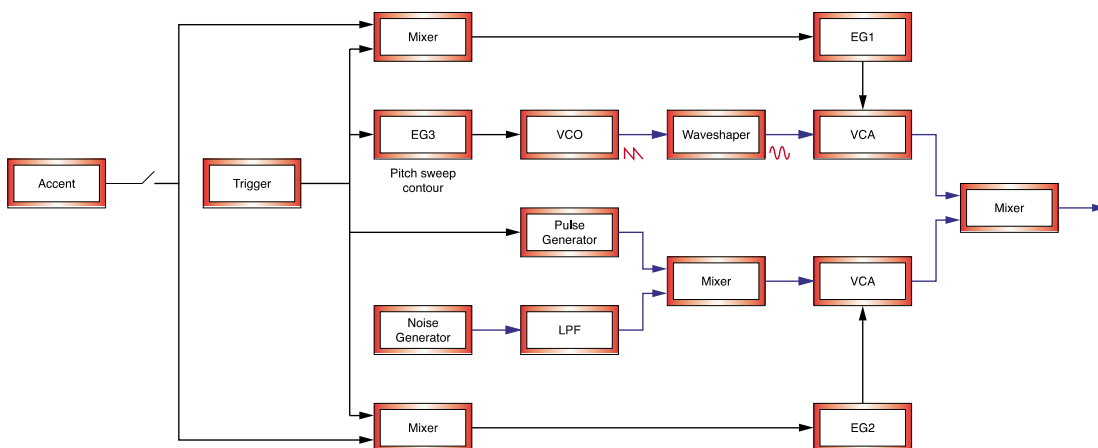


2.2. ábra. Roland TR-909 Pergődob [22]

A VCA 3 kézi paraméterezésével a felhasználó meghatározhatja a bekevert zajkomponens arányát, illetve a kimenőjelen lévő erősítő segítségével beállíthatja a pergőhang helyes arányát a teljes hangképben.

Roland TR-909 lábdob hang

A TR-909 által szintetizált lábdob hangja három fő komponensből épül fel.



2.3. ábra. Roland TR-909 Lábdob [22]

A 2.3. ábrán ennek a három komponensnek a struktúráját figyelhetjük meg.

A lábdob hangzás mélytónusú hangját az EG3 által vezérelt háromszögjel oszcillátor (VCO) adja. A kimenőjele át van vezetve egy aluláteresztő szűrőn, az így kapott jelalak sokkal inkább egy szinuszjel lesz, eltűnik belőle a felharmonikusok nagy része. A mélykomponens időbeli lefutását az EG1 vezérlőfeszültsége adja meg.

A hangzás karakterisztikáját nagyban befolyásolja a triggerimpulzus hatására a jelhez hozzákevert kattánás a "Pulse Generator" segítségével. Ez főleg nagyfrekvenciás komponensekkel gazdagítja a végeredményt.

A szintézist megszürt zaj hozzáadása teszi teljessé, ennek az időbeli lefutását az EG2 burkoló generátor határozza meg.

2.3. Fizikai alapú szintézis

A többi szintéziseljárással szemben, ahol a modellezendő hangszer által gerjesztett jelet próbáljuk valamilyen intuitív módszerrel közelíteni, a fizikai alapú szintézisnek a lényege, hogy megpróbáljuk megérteni és modellezni a hangszerekben történő fizikai folyamatokat, melyek a hangkeltésért felelősek. Ez a fizikai folyamatok alapján felépített matematikai modell legtöbbször két fő részből áll. Az egyik magának a hangszernek a viselkedését írja le tetszőleges gerjesztés hatására, a másik pedig a gerjesztés modellezése: dobmodell esetén pl. a dobverő milyen időbeli lefutással fejt ki a membránra a feszítő erőt. Fontos még megemlíteni, hogy a realiztikus hangzás érdekében azt a folyamatot is modellezhetjük, ahogyan a rezgésbe hozott hangszertestből az emberi fülbe érkezik a hang [40].

A matematikai modell a fizikai jelenségek további megértésével pontosítható, akár olyan részleteket beleépítve, minthogy a hangszer teste milyen fából készül, vagy a zongorában a hűrt megütő kalapács bevonatának keménysége hogyan változik a levegő hőmérsékletétől és páratartalmától függően.

Mikor kiválasztjuk azokat a fizikai jelenségeket, amiket szeretnénk a modellbe beleépíteni, akkor mérlegelendő az adott folyamat bonyolultságából származó számítási kapacitás, illetve, hogy az adott jelenség beépítése a modellbe mekkora hangzásbeli javulást okoz.

A modell lehetővé teszi azt, hogy a paraméterek beállításával közvetlenül beavatkozzunk a szintézisbe, megváltoztatva ezzel az eredeti hangzást. Ezek a paraméterek legtöbbször valamilyen valós fizikai tulajdonságot jellemeznek, pl. egy dobmodell esetén lehetőségünk van a dob méretét vagy a membránt feszítő erőt szabadon megválasztani. Ebből az is következik, hogy a modell paramétereit úgy is beállíthatjuk, hogy azzal a valós fizikai világban nem megvalósítható hangokat keltsünk.

Ennek a módszernek nagy előnye, hogy a generált hang természetes hatást kelt, sokkal változatosabb, mint a hangminta-visszajátszás alapú módszerek, valamint nem kell hozzá nagy tárhelyet elfoglaló hangminta adatbázis. Hátránya viszont a nagy számításgéni, emiatt a valós időben történő alkalmazása sokáig nehézségekbe ütközött.

Minden ilyen módszer alapja egy differenciálegyenlet-rendszer, ami a hangszer fizikai leírása. A különböző módszerek abban térnek el egymástól, hogy ezt az egyenletet milyen matematikai megközelítéssel próbálja megoldani, közelíteni. A különböző megoldások más-

más helyzetekben hozhatnak optimális eredményt.

A következő pontokban részletesebben is tárgyalt módszerek például a véges differenciák módszere, a digitális hullámvezető és a modális hangszintézis, aminek a részletes leírására egy önálló fejezetet kapott.

2.3.1. Digitális hullámvezető

A digitális hullámvezető (Digital Waveguide Method) eljárás alapja, hogy a hangszerben keletkezett hanghullámok viselkedését, terjedését próbálja meg leírni [31]. A J. O. Smith által kidolgozott módszer az ideális húr viselkedést leíró hullámegyenlet megoldására épül, mely szerint a húr pontjainak kitérése az idő függvényében felírható két ellentétes irányban haladó hullám összegeként. Az ideális, veszteségmenetes húr esetén a hullámegyenlet a 2.1 képlet írja le [33], [39].

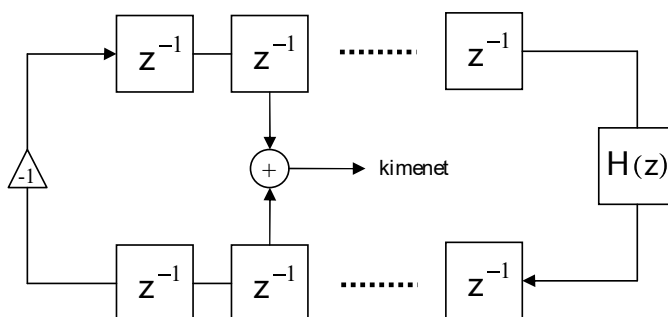
$$\frac{\partial^2 y}{\partial t^2} = c^2 \frac{\partial^2 y}{\partial x^2}. \quad (2.1)$$

A szimbólumok jelentése: y a rezgő test kitérése a rezgés síkjában, x a hullám terjedésének irányába mutató koordináta, t az idő, c a hullámterjedés sebessége [39]. A 2.1 egyenlet d'Alembert (1717-1783) által kidolgozott megoldása:

$$y = f_1(ct - x) + f_2(ct + x). \quad (2.2)$$

A 2.2 egyenletben $f_1(ct - x)$ jellemzi a c sebességű, pozitív irányú hullámot, $f_2(ct + x)$ pedig a negatív irányút. Eszerint a kérdéses pontban a kitérést megkaphatjuk két ellentétes irányban haladó hullám összegeként.

Az ideális veszteségmentes húrban a hullám torzítatlanul, csillapítás nélkül terjed, a végeken pedig ellentétes előjellel visszaverődik. A digitális modell két késleltetővonal segítségével implementálható. A húr nemideális tulajdonságait modellezhetjük a késleltetővonal egyik végére elhelyezett digitális szűrővel. A 2.4. ábra egy lehetséges megvalósítást mutat be.



2.4. ábra. Digitális hullámvezető egy lehetséges implementációja [39].

Ez a módszer olyan hangszerek esetében használható optimálisan, ahol az egydimenziós hullámterjedés határozza meg leginkább a hangszer hangját. (pl. vonósok, húros vagy fúvós hangszerek)

Előnye, hogy egydimenziós esetben egyszerű az implementálása (a modern DSP-kben külön periféria segíti a késleltetővonalak megvalósítását), hátránya a nagy számításigény ha több dimenzióban szeretnénk megvalósítani, valamint a rezgő rendszer nemlinearitásainak figyelembevétele korlátozott.

2.3.2. Véges differenciák módszere

Mint minden fizikai alapú hangszintézis módszer, ez is a modellezendő rendszert leíró differenciálegyenletekből indul ki. Ezek folytonos tartománybeli egyenletek, a rendszer viselkedését végtelensok pozícióban írja le végtelen sok időpontban. A módszer az alapegyenletek diszkrétizálásán alapul, a differenciálegyenletekben szereplő deriváltakat differenciahányadosokkal közelíti. A következő egyenlet az elsőrendű derivált értékének határértékes definíciója az x_0 pontban.

$$\left. \frac{dy}{dx} \right|_{x_0} = \lim_{\Delta x \rightarrow 0} \frac{y(x_0 + \Delta x) - y(x_0)}{\Delta x}. \quad (2.3)$$

A véges differenciák módszerének alkalmazásakor a parciális differenciálegyenletben szereplő deriváltakat véges differencia egyenletekkel helyettesítjük, vagyis az adott pontbeli deriváltakat a környező értékek lineáris kombinációjával közelítjük [39].

Ezzel elérjük, hogy elég véges pontban kiszámolni a vizsgált mennyiségeket. Megválaszthatjuk, hogy a modellezni kívánt rendszer viselkedését mekkora pontossággal szeretnénk kiszámolni, azaz nagyobb számítási kapacitás felhasználásával pontosabb eredményre jutunk. Előnye, hogy egyszerűen megvalósítható, hátránya, hogy bonyolult rendszerek esetén stabilitási problémák merülhetnek fel [40].

2.4. Absztrakt algoritmusok

Az absztrakt hangkeltő algoritmusok kategóriájába azok a metódusok tartoznak, ahol a cél nem valamilyen tradicionális hangszer modellezése, hanem önálló elektronikus hangzások keltése. Sokáig a szintetizátorok működési elvükből adódóan nem voltak képesek realisztikus hangok generálására, többek között azért, mert nem állt még rendelkezésre az ezekhez szükséges technológia és tudás.

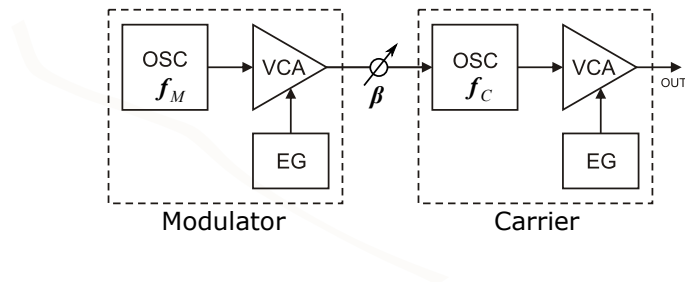
Mivel sok zeneszerző és együttes mégis elkezdte használni ezeket a szintetizátorokat, ezek elterjedésével új zenei stílusok jöttek létre. Manapság már kifejezetten cél a szintetizátor gyártók részéről, az új típusú, minél komplexebb effektek készítése.

Ebbe a kategóriába tartozik a szubtraktív és az additív szintézis bizonyos változata, de ide sorolhatók a híradástechnikából átvett modulációs eljárások is, pl az FM szintézis, Ring modulációs eljárások [39].

2.4.1. FM szintézis

Az FM (Frequency modulation) hangszintézis a modulációs szintézisek kategóriájába sorolható. Az alapja az, hogy a generált jelet egy alapjel és egy moduláló jel szorzatából

kapjuk. Minkét jel amiből kiindulunk lehet szinuszos vagy szélessávú. Amennyiben mindkét jel a hangfrekvenciás tartományban van, FM hangszintézisről beszélünk. A generált jel legtöbbször széles spektrumú, felharmónikusokban gazdag jel.



2.5. ábra. FM szintézis egy lehetséges megvalósítása [12]

A 2.5. ábra az FM moduláció általános struktúráját szemlélteti. Az FM szintézis előnye, hogy egyszerűen megvalósítható. Az első FM szintézist használó hangszer a YAMAHA DX7, mely 1983-ban jelent meg a piacon. A hangzás olyannyira elterjedt, hogy a mai napig is még készülnek erre a módszerre épülő megoldások (pl. Native Instruments FM8). Hátránya, hogy a paramétereit semmilyen fizikai tartalommal nem bírják, így a jó hangzások kikeverése empirikus módon történik [40].

3. fejezet

Modális hangszintézis

Kiemelten szeretnék foglalkozni a modális szintézissel, ugyanis a megvalósított dobszintetizátor alapvető működése erre épül. A fejezetben bemutatom, hogy a modális szintézist hogyan használom fel a dob hangjának szintéziséhez.

3.1. Mechanikai rendszer leírása

A módszer alapja az, hogy egy hangszert, mint mechanikai rendszert szeretnék modellezni, hogy ki tudjuk számolni tetszőleges gerjesztés esetén a választ [30].

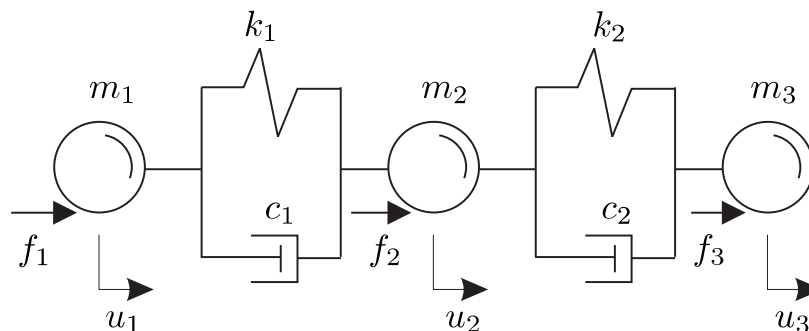
Ahhoz, hogy egy mechanikai rendszert modellezni tudjunk, három alapelem szükséges:

- **Tömeg** - m
- **Rugó**: rugómerevség - k
- **Csillapítás** - c

A rendszer bemenete és kimenete:

- **Erőgerjesztés** - a tömegekre ható $f(t)$ erők
- **Elmozdulásválasz** - a tömegek $u(t)$ elmozdulásai

A 3.1. ábrán egy egyszerű, három szabadsági fokú rendszer vázlatja látható.



3.1. ábra. Három szabadsági fokú koncentrált paraméteres rezgő rendszer [30]

Ha a különböző mennyiségeket mátrixban kezeljük, akkor felírhatjuk a rendszer mozgásegyenletét:

$$\mathbf{K}\mathbf{u}(t) + \mathbf{C}\dot{\mathbf{u}}(t) + \mathbf{M}\ddot{\mathbf{u}}(t) = \mathbf{f}(t). \quad (3.1)$$

A feladat tehát inentől kezdve adott, kíváncsiak vagyunk a mozgásegyenlet megoldására. Létezik direkt megoldása a problémának. Ha felírjuk frekvenciatarományban, akkor algebrai úton előállítható a megoldás. A Fourier-transzformáció szerint az idő szerinti deriválás $j\omega$ -val való szorzássá transzformálódik.

A 3.1 egyenlet felírva frekvencia tartományban:

$$(\mathbf{K} + j\omega\mathbf{C} - \omega^2\mathbf{M})\hat{\mathbf{u}}(\omega) = \hat{\mathbf{f}}(\omega). \quad (3.2)$$

Ha minden frekvencián előállítjuk az $\mathbf{S}(\omega) = \mathbf{K} + j\omega\mathbf{C} - \omega^2\mathbf{M}$ rendszer mátrixot, akkor $\mathbf{S}(\omega)$ inverzének segítségével kiszámolhatjuk az elmozdulásválaszt frekvenciatartományban:

$$\hat{\mathbf{u}}(\omega) = \mathbf{S}^{-1}(\omega)\hat{\mathbf{f}}(\omega). \quad (3.3)$$

Innentől kezdve $\hat{\mathbf{u}}(\omega)$ inverz Fourier-transzformálásával megkaphatjuk $\mathbf{u}(t)$ -t. Hátránya ennek a módszernek az, hogy minden frekvenciára ki kell számolnunk a választ, ami nagyon számításigényes, valamint az, hogy nem megkerülhető a rendszer leírására szolgáló mátrixok ismerete. Ez viszont komoly gond, ugyanis egy hangszer esetében akár több száz szabadsági fokú rendszerünk is előállhat.

3.2. Modális megoldás

Létezik a problémának egy másik típusú megoldása is. Ehhez be kell vezetni a módus fogalmát.

3.2.1. Módusok

A módusok olyan rezgésformák, amelyek a gerjesztés megszűnése után is fennmaradnak. Minden módushoz tartozik egy frekvencia és egy rezgésalak.

A módszer arra épít, hogy a rendszer $u(t)$ elmozdulásválasza tetszőleges $f(t)$ gerjesztés esetén felírható a rendszer ω_n módusalakjainak szuperpozíciójaként. Ha előállítottuk a módusainkat és részesedési tényezőiket, ki tudjuk számolni a választ tetszőleges esetben.

A jó hír az, hogy ezek a módusok mérésel is jól becsülhetőek, nem szükséges a rendszer mátrixok pontos ismerete. Folytonos rendszernek végtelen számú módusa van, viszont egy felső határfrekvenciáig véges. A Rubin-kritérium szerint ha a $f(\omega)$ gerjesztés rendelkezik Ω határfrekvenciával, akkor elegendő $\omega_n < 1.5\Omega$ módusokra kiszámolni a válaszokat, tehát véges idő alatt eredményre jutunk. Diszkrét rendszerben további korlátot jelent a f_s mintavételi frekvencia, tehát további korlátozó feltétel, hogy a választ nem számoljuk $f_s/2$ felett [30].

A módusok matematikai származtatására úgy jutunk, hogy keressük a 3.2 frekvenciatar-

tománybeli mozgásegyenlet megoldását gerjesztés nélküli ($\mathbf{f} = 0$) és csillapítatlan ($\mathbf{C} = 0$) esetben:

$$\mathbf{K}\varphi_{\mathbf{n}} = \omega_n^2 \mathbf{M}\varphi_{\mathbf{n}}, \quad n = 1, \dots, D. \quad (3.4)$$

Ha megoldjuk ezt a sajátérték problémát, akkor az így kiszámolt ω_n -ek (sajátértékek) megadják a módusfrekvenciákat, a φ_n -ek pedig a módusalakokat.

A modális megoldás azon alapszik, hogy a rendszer $\hat{\mathbf{u}}(\omega)$ elmozdulásvektora tetszőleges $\hat{\mathbf{f}}(\omega)$ gerjesztés hatására előállítható a rendszer φ_n módusalakjainak szuperpozíciójaként:

$$\hat{\mathbf{u}}(\omega) = \sum_{n=1}^D \varphi_n \hat{\alpha}_n(\omega), \quad (3.5)$$

ahol D a módusok száma, $\hat{\alpha}_n$ pedig a módusok részesedési tényezője. A részesedési tényezőt modális koordinátaként is szokás nevezni, mivel a módusalakok a rendszermátrixokra nézve ortogonális rendszert alkotnak.

3.3. Modális megoldás alapú jelszintézis

A modális hangszintézis alap gondolata, hogy ha a módusfrekvenciákon szinuszjeleket generálunk a gerjesztésnek megfelelő amplitudóval, akkor megkapjuk a modellezendő rendszer válaszát az adott gerjesztésre. Megjegyzendő, hogy a rendszer pontos viselkedéséhez a módusalakokkal is számolnunk kellene, de a hangszintézis eljárás során nem vagyunk kíváncsiak a dob membránjának tetszőleges pontban és tetszőleges időben a kitérésére, hanem a membrán által lesugárzott hang frekvenciája számunkra a rendszer kimenete. A szintézis szempontjából egy módust négy mennyiséggel jellemezünk: f_k módusfrekvencia, τ_k a lecsengés időállandója, ϕ_k kezdőfázis, A_k amplitudó. Ezt a négy mennyiséget optimális esetben mérésrel meg tudjuk határozni.

Egy lehetséges mód a mérések elvégzésére, hogy ha a mérendő rendszert (hangszert) egy impulzuskalapács segítségével rezgésbe hozunk és regisztráljuk a kalapács által leadott erő-idő lefutást és a hangszer egy meghatározott pontjának elmozdulás-idő viszonyát, ezekből pedig meghatározzuk az impulzusválaszt. A feladat, hogy az impulzusválasz alapján határozzuk meg a keresett módusokat.

Optimális esetben - ha a módusfrekvenciák nem változnak időben - egyszerű spektrumanalízissel is könnyen eredményre jutunk a kezdeti amplitudókat és módusfrekvenciákat illetően, de a lecsengések meghatározása nem ennyire egyszerű. A problémára többféle módszer is létezik, az egyik legegyszerűbb pl. a heterodin lekeveréses módszer. A lényege ennek az, hogy kiindulunk az analizálni kívánt hangjelből, majd egy kiválasztott frekvenciájú módust lekeverjük alapsávra. A kapott jelet aluláteresztő segítségével megszűrjük úgy, hogy a többi módus már ne tudjon belapolódni a vizsgált jelbe. Ha az így kapott időfüggvényben jelen lévő exponenciálisan lecsengő görbének meghatározzuk az időállandóját, akkor megtaláltuk a keresett értéket.

A módszer magában hordozza azt a tulajdonságot, hogy a kiválasztott hangmintában

egy frekvenciára fókuszál, viszont a perkusszív hangszerek módusainak alapfrekvenciája a sok nemlineáris jelenség miatt időben változhat. A legnagyobb eltérések rögtön az impulzusszerű gerjesztés után jelentkeznek, így ez a módszer perkusszív jellegű hangok analízisére nem optimális. Az ilyen hangszereknél eredményre vezethet a fizikai folyamatok mélyebb feltérképezése.

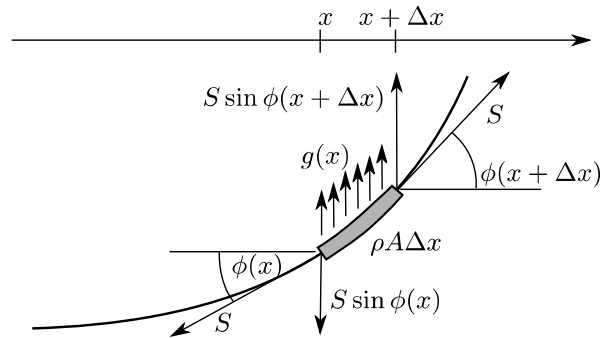
3.4. Kör alakú membrán módusai

Bár a modális hangszintézis egyik nagy előnye, hogy egy rezgő rendszer módusai mérésrel is hatékonyan meghatározhatók, de a 3.3 pontban már megállapításra került, hogy ez perkusszív hangszerek esetében mégis nehéz feladatot jelent.

Egy másik megközelítés a módusok meghatározására, ha felépítünk egy fizikai modellt a kifeszített kör alakú membránra és ennek segítségével elméleti úton határozzuk meg, számoljuk ki a módusokat.

3.4.1. Az ideális húr mozgásegyenlete

A húr rezgéstani szempontból a legegyszerűbb rendszer, melyben külső gerjesztés hatására hullámterjedés léphet fel. Ennek feltétele a húr kifeszítése, ugyanis ebből származik az a feszítő erő, aminek hatására létrejöhet rezgés, valamint hullámterjedés. A húr mozgásának leírásához tekinthetjük a rendszert egydimenziós kiterjedésűnek, ugyanis a húr keresztmetszete lényegesen kisebb a hosszirányú kiterjedésénél és a rajta kialakuló rezgések hullámhosszánál. Ez azt is jelenti, hogy transzverzális hullámterjedést feltételezünk.



3.2. ábra. A húr Δx hosszú szelete [32]

A 3.2. ábrán láthatunk egy húrszeletet és a rá ható erőket. Az S a húrt kifeszítő erő, a $g(x)$ a hosszegységre jutó külső gerjesztő erő, az ebből származó erő: $g(x)\Delta x$. $\phi(x)$ jelöli a húrt kifeszítő erő vektorának x tengellyel bezárt szögét, valamint $\mu = \rho A$ a húr hosszegységre eső tömege.¹ Ezek segítségével felírhatjuk a rezgő rendszerre Newton második törvényét [32]:

$$g(x)\Delta x + S[\sin \phi(x + \Delta x) - \sin \phi(x)] = \mu \Delta x a(x), \quad (3.6)$$

¹ ρ a fajlagos tömeg, A a húr keresztmetszete.

ahol $a(x)$ a szelet gyorsulását jelenti. Ha figyelembe vesszük, hogy kis szögek esetén élhetünk a $\sin \phi \approx \phi$ közelítéssel és bevezetjük az $u(x)$ kitérés függvényét, akkor az alábbi alakra juthatunk:

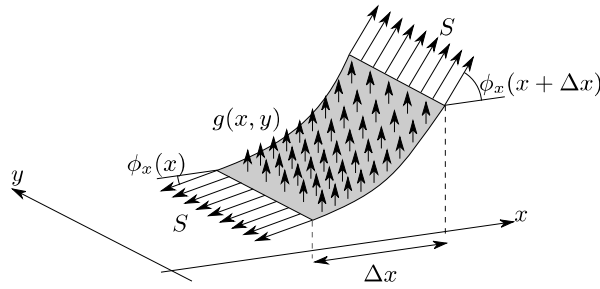
$$\phi(x) \approx \tan \phi(x) = \frac{du(x)}{dx}, \quad (3.7)$$

$$g(x, t) + Su''(x, t) = \mu\ddot{u}(x, t), \quad (3.8)$$

ahol a függvény feletti pont az idő szerinti, a vessző pedig a hely szerinti deriváltat jelenti.

3.4.2. Az ideális membrán mozgásegyenlete

Az ideális membrán rezgéstani szempontból tekinthető kétdimenziós húrnak, tehát a transzverzális hullámok terjedési iránya tetszőleges lehet az x - y síkon. Az ideális membrán a húrhoz hasonlóan nem rendelkezik önálló merevséggel, rezgést csak külső feszítő erő jelenlétében tud végezni. A 3.3. ábrán egy membrán $\Delta x, \Delta y$ méretű darabja látható x, y



3.3. ábra. A membrán $\Delta x, \Delta y$ méretű darabja [32]

pontban. A húrhoz hasonló módon felírhatjuk a membránra Newton második törvényét a membránra ható erők és a tömeg ismeretében ($\Delta x \Delta y$ felületelemmel végigosztva):

$$g(x, y, t) + S\left(\frac{\partial \phi_x(x, y, t)}{\partial x} + \frac{\partial \phi_y(x, y, t)}{\partial y}\right) = \sigma a(x, y, t), \quad (3.9)$$

ahol $a(x, y, t)$ membrán gyorsulását jelenti, σ pedig az egységnyi felületre vonatkoztatott tömegét. Ha itt is bevezetjük az $u(x, y, t)$ kitérésfüggvényét az előző pontban említett közelítésre alapozva, felírhatjuk a membrán mozgásegyenletét:

$$g(x, y, t) + S\nabla^2 u(x, y, t) = \sigma\ddot{u}(x, y, t). \quad (3.10)$$

A ∇^2 a Laplace operátort jelenti, derékszögű koordináta rendszerben:

$$\nabla^2 u = \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}. \quad (3.11)$$

3.4.3. A membránegyenlet megoldása, módusok meghatározása

A membránegyenlen megoldása visz minket közelebb ahhoz, hogy meg tudjuk határozni a kör alakú membrán módusait, ami a dob hangzás szintéziséhez szükséges. A 3.2.1 pont szerint a módusok olyan rezgésformák, melyek gerjesztés megszűnése után is fennmaradnak. Ezt az jelenti, hogy innentől kezdve a gerjesztés nélküli mozgásegyenletet használjuk: $g(x, y, t) = 0$. Amennyiben bevezetjük a rezgésterjedési sebességet:

$$c = \sqrt{\frac{S}{\sigma}}, \quad (3.12)$$

a következő egyenletre jutunk:

$$c^2 \nabla^2 u(x, y, t) = \ddot{u}(x, y, t). \quad (3.13)$$

Észrevehetjük, hogy a c rezgésterjedési sebesség függ az S feszítettségtől. Ez teszi lehetővé azt, hogy a feszítettség változtatásával hangolhatjuk a rezgő rendszer módusait. Ez éppúgy igaz a gitárhúrra, mint a dob membránjára. Mivel kör alakú membránt vizsgálunk, geometriai szempontok miatt érdemes polárkoordinátákra áttérni a következő módon.

$$x = r \cos \theta, \quad y = r \sin \theta. \quad (3.14)$$

Ahhoz, hogy az egyenlet megoldására eljuthassunk, szükséges alkalmazni a változók szeparálásának módszerét. Így a kitérésfüggvényt a következő módon bonthatjuk fel egy helyfüggő és egy időfüggő tagra:

$$u(r, \theta, t) = \psi(r, \theta) \alpha(t). \quad (3.15)$$

A felbontott kitérésfüggvényt a homogén membránegyenletbe helyettesítve:

$$c^2 \frac{\nabla^2 \psi(r, \theta)}{\psi(r, \theta)} = \frac{\ddot{\alpha}(t)}{\alpha(t)} = -\omega^2, \quad (3.16)$$

ahol az ω a keresett módus sajátfrekvenciája.

Ennek az egyenletnek a megoldása során jutunk el odáig, hogy meghatározhassuk a számunkra szükséges paramétereket. A teljes levezetés nélkül az eredmény [32]:

$$\psi_{mn}(r, \theta) = J_n(k_{mn}r) \cos(n\theta + \zeta), \quad k_{mn} = \frac{\gamma_{mn}}{R}, \quad (3.17)$$

ahol $k = \omega/c$ a hullámszám, R a membrán sugarát, J_n az n -edik Bessel függvényt jelenti, γ_{mn} pedig a $J_n(\gamma)$ függvény zérushelyeit. A lehetséges megoldásokat a $kR = \gamma_{mn}$ választja ki, a ψ_{mn} pedig a módusokat jelöli. A módusokhoz tartozó sajátfrekvenciák:

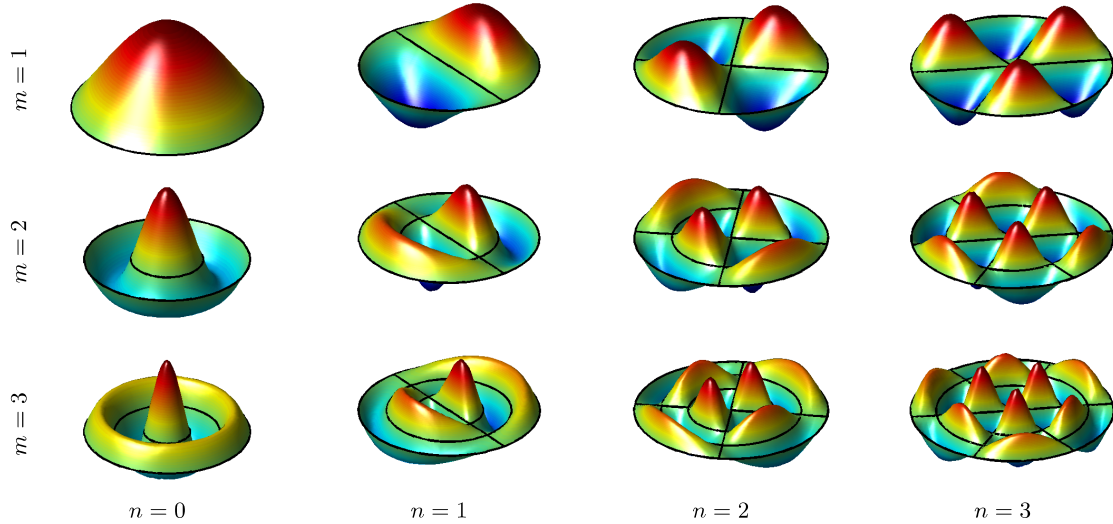
$$\omega_{mn} = k_{mn}c = \frac{\gamma_{mn}c}{R}. \quad (3.18)$$

A Bessel függvényeket n -nel indexeljük, jellemzőjük, hogy a harmonikus függvényekhez

hasonlóan oszcillálnak, de monoton csökkenő burkolóval. Nagy argumentumokra ($\gamma \gg n^2 - 1/4$) jól közelíthetők a következő képlettel:

$$J_n(\gamma) \approx \sqrt{\frac{2}{\pi\gamma}} \cos\left(\gamma - \frac{(2n+1)\pi}{4}\right). \quad (3.19)$$

A 3.4. ábrán láthatjuk a kör alakú membrán néhány módusának szemléltetését.



3.4. ábra. A kör alakú membrán módusai [32]

A módusok meghatározásához az ideális membrán mozgásegyenletéből indultunk ki, így a módusok is az ideális esetre vonatkoznak. Ha további fizikai folyamatokkal bővítjük a mozgásegyenletet, akkor az a következőképpen alakul [38]:

$$D\nabla^4 u(r, \theta, t) + \sigma \frac{\partial^2 u(r, \theta, t)}{\partial t^2} - S\nabla^2 u(r, \theta, t) + d_1 \frac{\partial u(r, \theta, t)}{\partial t} + d_3 \frac{\partial \nabla^2 u(r, \theta, t)}{\partial t} = g(r, \theta, t), \quad (3.20)$$

ahol $u(r, \theta, t)$ a kitérés függvény, $g(r, \theta, t)$ a külső gerjesztő erő.

$$\nabla^4 = \nabla^2 \cdot \nabla^2. \quad (3.21)$$

A Laplace operátor (∇^2) polár koordinátákkal:

$$\nabla^2 u = \frac{\partial^2 u}{\partial r^2} + \frac{1}{r} \frac{\partial u}{\partial r} + \frac{1}{r^2} \frac{\partial^2 u}{\partial \theta^2}. \quad (3.22)$$

A d_1 a disszipációs veszteség frekvenciafüggetlen része, a d_3 pedig a frekvenciafüggő. D pedig a negyedfokú taggal a hajlító merevségből adódó hatásokat modellezi.

3.4.4. Membrán-feszültség moduláció

A fizikai modell alapú szintézisek alapvető tulajdonsága, hogy a modellezendő rendszer fizikai folyamatainak mekkora részhalmazát valósítja meg, mennyire közelíti meg.

A doboknál a kifeszített membrán viselkedésén kívül sok más tényező is szerepet játszik a hangzásban, többek között, hogy legtöbbször két membrán van, amik a levegőn keresztül valamilyen csatolásban vannak egymással. Pergődoboknál közre játszhat az alsó membránra feszített sodrony és ennek a kölcsönhatása a felette lévő duplamembrános rezgő rendszerrel.

Ilyen hatás még a membrán megnyúlása illetve a rugalmassága. Kimérhető ugyanis, hogy ha egy bizonyos erőnél nagyobb a $g(x, y)$ külső gerjesztő erő, akkor megnő az S feszítettség. A 3.12 összefüggésből is látszik, hogy ha a feszítettség nő, az befolyásolja a hangszer hangolását is. A megfigyelt jelenség az az, hogy ebben az esetben a módusok sajátfrekvenciája megnő a membrán megnyúlásából adódó felületnövekedésével arányosan [24]. A 3.20 mozgásegyenletben lévő S feszítettséget a következő módon bonthatjuk fel:

$$S(z) = S_0 + S_{NL}(z), \quad (3.23)$$

ahol S_0 a nyugalmi feszítettség, S_{NL} pedig a nemlinearitásokból következő addicionális feszítettség. Ez általánosan, a membrán alakjától függetlenül így alakul:

$$S_{NL}(z) = \frac{Qh}{1-v^2} \frac{A(z) - A_0}{A_0}, \quad (3.24)$$

ahol Q a membrán rugalmassági tényezője, h a membrán magassága, v a Poisson-tényező, A_0 a nyugalmi felület, $A(z)$ pedig a nagy erőhatások hatására kialakuló felületnövekedés. Ez a jelenség nagyon erősen módosítja a dob (kifejezetten a lábdob) hangját, emiatt a megvalósított dobszintézis eljárásnak is része.

3.5. Szintézis struktúra

A fejezet korábbi részei leírják, hogy miként tudjuk meghatározni a hangszintézishez szükséges paramétereket (módusokat) méréssel vagy elméleti úton. Amennyiben a megfelelő paraméterek előálltak, a következő módon tudom felépíteni a modellt.

Egy módus modellezéséhez szükséges szinuszos hullámformát rezonátorral állítom elő. Ez alapvetően egy rezonáns, éles átvitelrel rendelkező IIR szűrő, aminek a középfrekvenciája a módusfrekvenciára van hangolva, valamint a szűrő impulzusválaszának lecsengése a módus lecsengésével egyezik meg.

A szűrőparamétereket a következő módon számoltam ki [37]:

$$H_{res,k}(z) = \frac{b_k z^{-1}}{1 + a_{1,k} z^{-1} + a_{2,k} z^{-2}}, \quad (3.25)$$

$$b_k = \frac{A_k}{f_s} \text{Im}\{p_k\}, \quad (3.26)$$

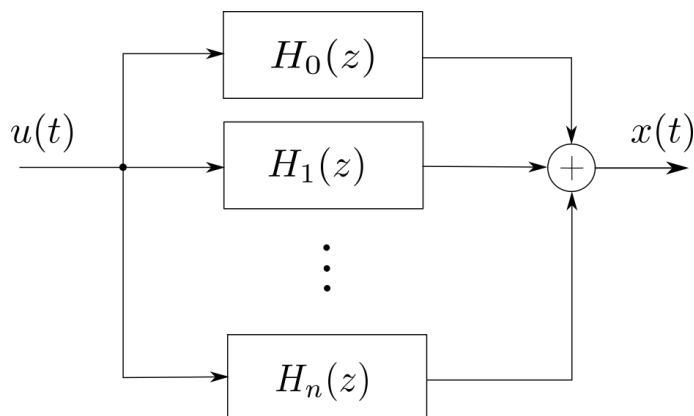
$$a_{1,k} = -2\text{Re}\{p_k\}, \quad (3.27)$$

$$a_{2,k} = |p_k|^2, \quad (3.28)$$

$$p_k = e^{j2\pi \frac{f_k}{f_s}} e^{-\frac{1}{\tau_k f_s}}. \quad (3.29)$$

Itt f_k a módusfrekvencia, τ_k a lecsengés időállandója, p_k a szűrő pólusa, b_k , $a_{1,k}$ és $a_{2,k}$ a szűrőegyütthatók.

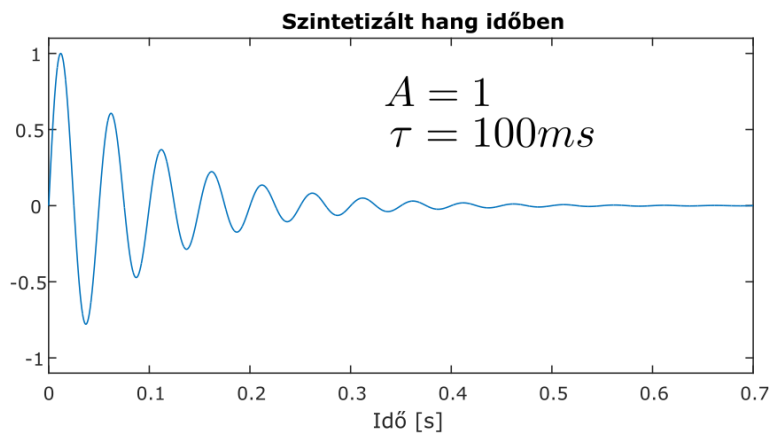
A következő ábrán szeretném illusztrálni a modális hangsintézisre alkalmas elrendezést.



3.5. ábra. Párhuzamosan kapcsolt rezonátorok

A 3.5. ábrán látható, hogy minden rezonátorra ugyanaz a gerjesztőjel kerül, majd a szintézis eredménye a rezonátorok kimeneteinek összegzésével áll elő.

A 3.6. ábrán látható egy rezonátor kimenete, ami egy exponenciálisan lecsengő szinuszjel.



3.6. ábra. Rezonátor kimenete

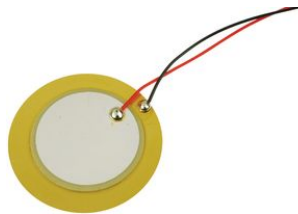
Az dobszintetizátor fejlesztése során az első lépés az volt, hogy megkeressük azt a struktúrát vagy struktúrákat, ami alkalmas triggeresemény nélkül dobhangok előállítására. Minden alkalommal a párhuzamosan kapcsolt rezonátorokból indultunk ki. A különböző típusú modellek pontos megvalósítását külön fejezetben tárgyalom.

4. fejezet

Szenzorválasztás

Az elektronikus hangszerünk jelfeldolgozási folyamatának belépési pontja a szenzor. Fontos, hogy már az érzékelő által szolgáltatott jelben benne legyen mindaz az információ, ami a hangszeres játékot a lehető legrészletesebben jellemzi. Emiatt a munkám során nagy hangsúlyt fektettem arra, hogy megvizsgáljam, a különböző technológiával rendelkező jeledők alkalmasak-e a feladatra, illetve ha alkalmasak, akkor hogyan kell előfeldolgozni a jelüket, hogy a fizikai modell erőbemenete lehessen.

Továbbá egy másik szempont volt a szenzorok vizsgálatánál az, hogy a hangszer különböző pontjaira felszerelve mekkora áthallás van köztük, alkalmas-e egy jó szenzorelrendezés arra, hogy következtetni tudjunk nagy biztonsággal az ütés pozíciójára. A feladat bemutatásánál már említettem, hogy a vizsgálatom tárgyát képező cajon alapvetően kétféle jellegű hangot képes kelteni. Ez abból a szempontból is érdekes, hogy a szintetizátor működése is erre alapul, két modell kerül megvalósításra. Egy megütés során el kell dönteni, hogy a jelet melyik modellbe vezessük be.



4.1. ábra. Piezo jeledő [10]

4.1. Piezokristály

A piezokristály a legelterjedtebben használt bemeneti érzékelő az elektronikus dobok esetében, így először én is ezt próbáltam ki.

A szenzor a működéséhez a piezoelektromos effektust használja ki, aminek köszönhetően alakváltozást alakít át töltéskülönbséggé. Előnye, hogy működéséhez nem igényel külső energiát. Ennek értelmében a szenzor alkalmazható minden olyan fizikai mennyiség mérésére, mely alakváltozást idéz elő. Ilyenek pl. az erő, nyomás, gyorsulás. Az alkalmazott technológia nem alkalmas statikus, csak dinamikus mérésekre, ugyanis az alakváltozás (kris-

tálydeformáció) hatására létrejött töltéskülönbség gyorsan elszivárog. Az átalakító anyaga régebben a természetes kristályok (kvarc, turmalin, Rochell só) köréből került ki, napjainkban azonban az báriumtitanát, ólomtitanát és ólomcirkonát kerámiákat a nagyobb érzékenységük miatt alkalmazzák. Ezeket porkohászati úton állítják elő, majd megfelelő elektromos térerő segítségével állítják be a piezoelektromos dipólusokat a kívánt irányba [26].

Az általunk alkalmazott szenzor egy vékony rézkorongból és a ráragasztott szintén korong alakú piezokristályból áll, melyre vékony vezetőréteget képeztek. Egy gyakran használt felépítésű érzékelő a 4.1. ábrán látható. Esetünkben a hangszer testéhez ragasztva a lemez gyorulásával arányos jelet mérünk. A pontos gyorsulásméréshez olyan szenzorra lenne szükség, amelyikben ismert szeizmikus tömeg elmozdulását mérjük.

Megjegyzendő még, hogy a szenzor egyik jellegzetessége, hogy van egy a felépítéséből fakadó rezonanciafrekvenciája, ez az átviteli függvényében kiemelkedő csúcsként jelentkezik.

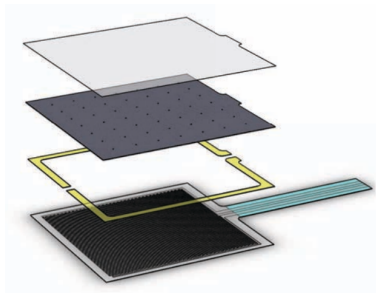
4.2. FSR

Egy kevésbé ismert szenzort, az erőérzékeny ellenállást (angolul: force sensitive resistor) is vizsgálatam a munkám során. A tesztekhez az Interlink Electronics FSR Model 406 termékét használtam [28].

A szenzor alapvetően két membránból áll, a kettő között légréteggel. A légréteget egy távtartó réteg biztosítja a membránok között. A felső membrán egy szén tartalmú tintával van bevonva mely megvalósítja a nyomásérzékelést. A legalsó réteg pedig egy speciális rajzolattal ellátott nyomtatott áramkör. Mindkét membrán flexibilis, poliimid rétegre készül. Nyomás hatására a felső membrán nekinyomódik az alsónak, így rövidre zárja az áramkört. Minél nagyobb a nyomóerő, annál nagyobb az alakváltozás, így az alsó áramkör eredő ellenállása lecsökken.



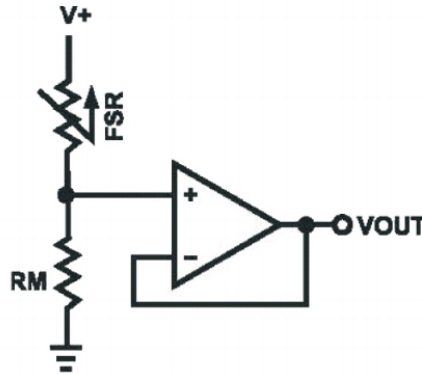
4.2. ábra. FSR rétegei [28]



4.3. ábra. FSR robbantott ábra [28]

A munkám során többféle módon vizsgáltam a szenzort, hogy kiderítsem alkalmas-e

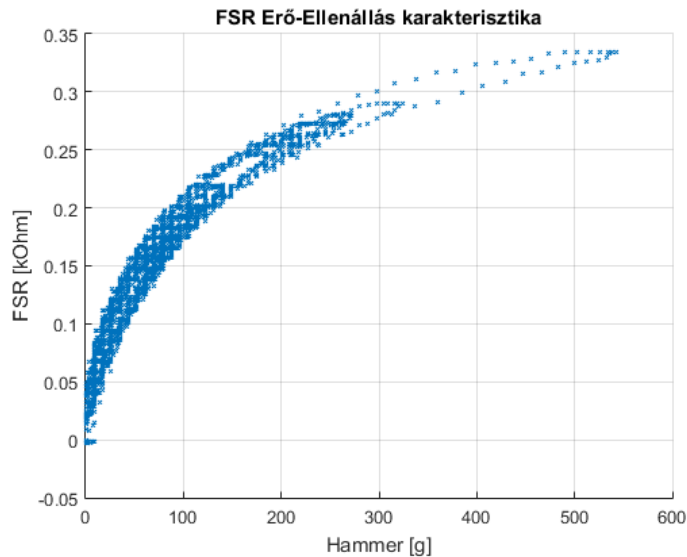
az általam használt fizikai modell megfelelő gerjesztésére. A gyártói ajánlásban többféle lehetőséget mutatnak be, hogy az ellenállásváltozást feszültséggé alakítsuk. Én a 4.4. ábrán látható feszültségosztó kapcsolást próbáltam ki.



4.4. ábra. Az FSR kimenetét feldolgozó feszültségosztó kapcsolás [28]

Kíváncsi voltam az erő-ellenállás karakterisztikájára. Ezt úgy mértem meg, hogy a szenzor felületét impulzuskalapáccsal gerjesztettem, és mértem a kalapácsból származó erőjelet és szenzor ellenállásváltozása hatására történő feszültségváltozást.

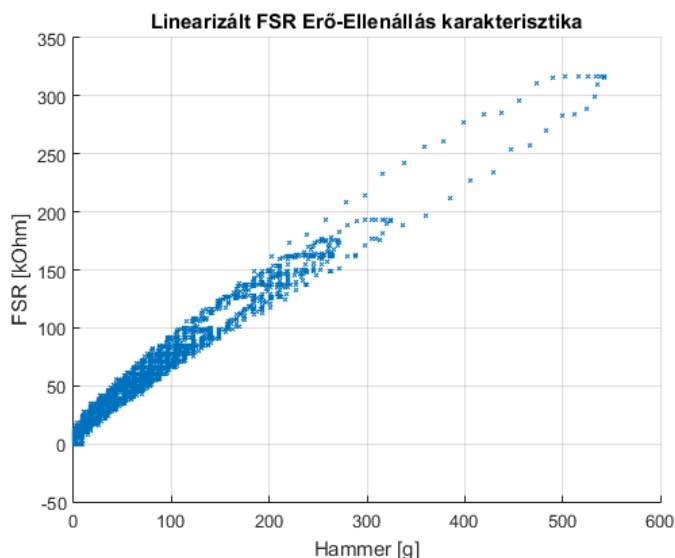
A mérési adatok feldolgozása úgy történt, hogy a leadott és mért impulzus több pontjáról mintát vettem és ezt jelenítettem meg.



4.5. ábra. FSR erő-ellenállás karakterisztikája

Megfigyelhető az ábrán egy hiszterézis jelenség is. Az adatlap is ír erről a jelenségről, ez az alkalmazott technológiából fakad és egyben egy aluláteresztő viselkedésre is utalhat. Ha ránézünk az ábrára, azt látjuk, hogy a görbe erősen kompresszált, azaz a kisebb erő esetén egységnyi erőváltozás nagyobb ellenállásváltozást idéz elő, mint ugyanekkora erőváltozás nagyobb erő mérése esetén. Ezt a torzulást a feszültségosztó kapcsolás okozza ugyanis, az ellenállásváltozásra nézve ez a kapcsolás nem lineáris. Ha a feszültségosztó-kapcsolás

hatását kompenzálom, akkor a karakterisztika nagymértékben lineárisra tehető.



4.6. ábra. FSR linearizált erő-ellenállás karakterisztikája

Fontos még kihangsúlyozni, hogy itt nem statikus karakterisztikát ábrázoltam, mert az ellenállásváltozás választ dinamikus gerjesztés (kalapácsütés) segítségével határoztam meg.

A mérés során rögzített adatok alapján vizsgáltam a jelet frekvencia tartományban is, ugyanis a magasabb frekvenciára elhelyezett rezonátorok csak akkor kapnak gerjesztést, ha van a bemenő jelben magasabb frekvenciás komponens is. A vizsgálat eredménye összefoglalva az, hogy a szenzor impulzusgerjesztés hatására a hangfrekvenciás tartomány legalján szolgáltat gerjesztőjelet, a jel energiájának jelentős része 200 Hz alatt van, így a lassúsága miatt közvetlenül erőbemenetként nem optimális.

További gyakorlati tapasztalat, hogy a frekvenciatartománybeli rossz viselkedése mellett nagyon jó tulajdonsága a gyorsulás gerjesztésre való érzéketlensége. Ez praktikus azt jelenti, hogy a cajonra felszerelve csak akkor észlelhető kimenőjel ha közvetlen a szenzort éri az erőhatás. Ha a cajon rezonáns lapját ütjük, akár a szenzorhoz nagyon közel, nem látható a kimeneten változás. Ez a tulajdonsága nagyon előnyös abban a helyzetben, ha olyan hangzást szeretnénk megszólaltani vele, aminek a hangja jellegében különbözik a cajon hangjától (pl.:cintányér). Ekkor biztosak lehetünk benne, hogy az áthallási jelenségeknek köszönhetően nem szólal meg, csak szándékolt esetben. További előnyös tulajdonsága, hogy DC jelet ad, tehát a folyamatosan jelen lévő erő esetén tartja a felvett ellenállás értéket. A kimenet így akár valós időben vezérelheti a szintetizátor valamely paraméterét.

Elektronikai szempontból még egy megjegyzés, hogy a feszültségosztó kapcsolás a szenzor tápfeszültségén található zajt is hozzáadja a hasznos jelhez, tehát külön gondoskodni kell egy alacsony zajú tápegységről a szenzor működtetéséhez.

4.3. MEMS technológiás szenzorok

A fejlesztés kezdeti fázisában felmerült, hogy vizsgáljuk meg a MEMS (Microelectromechanical Systems) technológiájú gyorsulásmérőket és mikrofonokat. Általánosan a gyorsu-

lasmérő azért merült fel, mert mint mérőműszer gyakran használatos hangfrekvenciás jelek mérésére. A piacon kapható termékeket átnézve arra a megállapításra jutottam, hogy az elérhető árú MEMS gyorsulásmérők felső határfrekvenciája alacsony a mi céljainkhoz. Magasabb árkategóriában vannak olyanok, amik adatlap szerint megfelelnek (pl.: Analog Devices - ADXL356), de olyat, ami kifejezetten audió célra lenne fejlesztve, nem találtam.

A munkám során teszteltem két MEMS mikrofont (Analog Devices - ADMP401). Tapasztalataim szerint a kipróbált modellek sokkal zajosabbak voltak, mint a piezo alapú megoldások, és a cajonba szerelve nagyobb ütésnél túlvezérlődtek. Az ADMP401 chipbe integrált előerősítővel van szerelve, így annak erősítése egy fix beállított érték, nem tudtam javítani a túlvezérlődési problémán. Megjegyzem, nem tudom pontosan meghatározni, hogy a beépített erősítő vezérlődik túl, vagy már a mikrofon sem képes az adott hangnyomást torzításmentesen elektromos jellé alakítani.

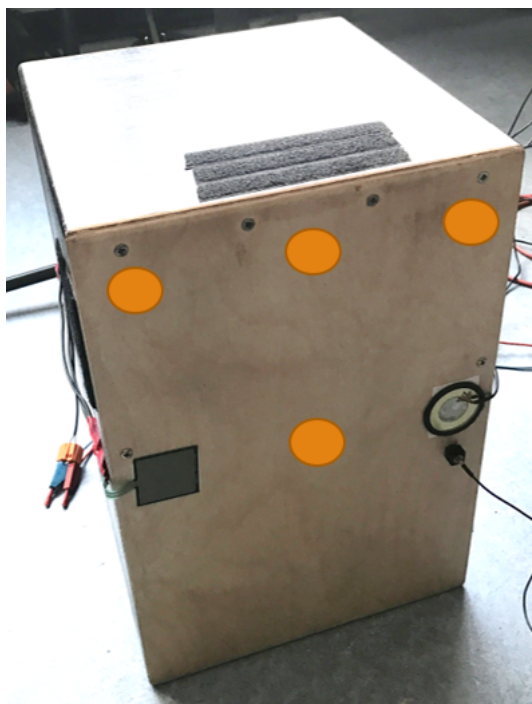
4.4. Szenzorokon végzett mérések

A jelfeldolgozás tervezése során több kérdés is felmerült a szenzorok jeleivel kapcsolatban. Ilyen kérdés volt, hogy milyen a jelek frekvenciatartománybeli viselkedése, mekkorák a kimeneti jelszintek, mekkora az érzékelők közötti áthallás a felszerelési pozíció függvényében. Hogy ezeket megválaszoljuk méréseket végeztem cajonra felszerelt szenzorok segítségével. A mellékelt ábra szerint helyeztem el a jeladókat. A 4.7. ábrán látható módon négy piezót a rezonáns lapra ragasztottam fel hátulról, valamint egy nagy átmérőjű piezót és egy FSR-t az előlapra. Referenciaként egy Shure Beta57 típusú mikrofont használtam és egy mérés-technikai célra szánt (Brüel & Kjaer) gyorsulásmérőt az előlapon. Összesen 8 csatornán rögzítettem a jeleket.

A mérések kiértékelése során több megállapítást is tettem.

1. Ha az ütés pozícióját oly módon szeretnénk meghatározni, hogy az ütés pillanatában összehasonlítjuk a szenzorok kimenetén mérhető pillanatnyi amplitudókat - azzal a feltételezéssel élve, hogy amelyik szenzorhoz történt legközelebb az ütés ott a legnagyobb az amplitudó - akkor az esetek legnagyobb részében ez működőképes megoldás lehet a függőleges dimenzióban.¹ A hátránya viszont az, hogy mindegyik bemeneti fokozat erősítését nagy pontossággal egyformára kell tervezni, ugyanis a szenzorok jelei között nincs nagy különbség.
2. A fent elhelyezett szenzorok között nagy áthallás van, a felső három pozícióban elhelyezett szenzorok kiemenete szinte egyforma. Itt nincs szükség ennyi szenzorra, csatornára.
3. A piezo szenzor által mért jelek széles spektrumúak, alkalmasak a rezonátoros architektúra gerjesztésére.
4. Az FSR kimenő jelében csak akkor tapasztaltunk változást, ha közvetlen a szenzort érte ütés.

¹A pozíció pontos definíciója esetünkben az, hogy az ütés fent történt-e, pergő szerű volt-e, vagy középen, tehát lábdob szerű volt-e.



4.7. ábra. *Felszenzorozott cajon - a narancssárga jelzés jelenti a belülre felszerelt szenzort*

4.5. Összefoglalás

A tesztekéből kiderült, hogy az adott feladatra (a fizikai modellek erőbemeneteként) a piezokristály a legalkalmasabb. Nagy előnye, hogy passzív, nem igényel külső táplálást. Ha a piezo utáni bemeneti fokozat erősítését úgy állítom be, hogy az ütések hangereje az optimális tartományban legyen, akkor a nyugalomban hagyott szenzorból nem mérhető a zaj. Az egyedüli hátránya a rezonanciája, de ha ez a frekvencia elég magas, akkor a tapasztalataim szerint jól kezelhető a jelben. Egyszerűen kiszűrhető, és nem is okoz gondot a szintézis során. Ezek az állítások az általam tesztelt szenzorra igazak, aminek a rezonancia-frekvenciája 6,5 KHz-en található.

5. fejezet

Hangszintézis megvalósítása modellező környezetben

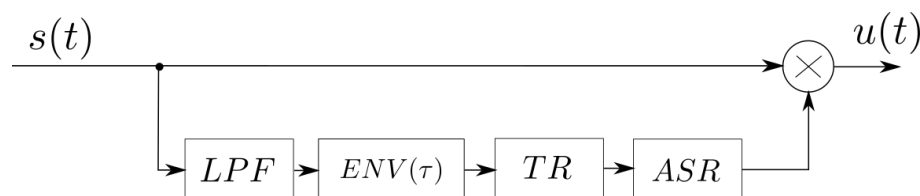
A jelfeldolgozási algoritmusok tervezésénél bevált gyakorlat, hogy a konkrét implementáció előtt megvalósítjuk, teszteljük az ötleteinket. A Matlab nagyon hatékonyan alkalmazható az audió rendszerek fejlesztésénél, mert gazdag jelfeldolgozási eszközkészletet, alapvető építőelemeket biztosít ehhez. Megkönnyíti a hibakeresést is és a mérnöki gyakorlatban rendkívül fontos, hogy változatos eszközöket használhatunk az adatok megjelenítésre.

A dobszintetizátor fejlesztésénél a munka első felében a modellező környezetet használtam arra, hogy kipróbáljam a modális szintézist és megtaláljam az optimális struktúrát a különböző modellek megvalósításához.

A Matlabban megvalósított ütőhangszermodell három fő részre tagolható a következő módon.

5.1. Előfeldolgozás

A mérési adatok feldolgozása során rögtön kiderült, hogy nagy áthallás van a jeladók között, így mindenképpen szükséges egy lépés, ahol eldöntjük, hogy az adott szenzorból érkező jelet beküldjem-e a dobmodellbe, vagy sem. Az első megoldásom erre egy zajzár, ami annyiban különbözik a hangtechnikában általánosan használt módszertől, hogy a vezérléshez egy szűrt jelet használok fel, nem pedig az eredetit. Az így kialakított feldolgozást a 5.1. ábra szerint valósítottam meg.



5.1. ábra. Alkalmazott zajzár struktúrája

Az ábrán lévő feldolgozási lépéseket a következő módon valósítottam meg.

5.1.1. Alul- és felüláteresztő szűrő

A 5.1. ábrán az *LPF* blokkal jelölt lépés, itt történik a megfelelő frekvenciatartomány kiválasztása. Ha a lábdob hangjára vagyunk kíváncsiak, akkor itt aluláteresztő szükséges, ha viszont a pergőre, akkor ott a célunk a kismagyas frekvenciás komponensek kiszűrése. Tapasztalataim alapján 70 Hz törésponti frekvenciát érdemes választani.

A szűrőparaméterek meghatározásához a Butterworth approximációt használtam. Ez a módszer klasszikus analóg szűrőtervezési technikát használ, majd az analóg szűrőt transzformálja diszkrét tartományba. Az átviteli karakterisztikája, ha a törésponti frekvencia $\omega = 1$:

$$|H(j\omega)|^2 = \frac{1}{1 + \omega^{2N}}, \quad (5.1)$$

ahol N a szűrő fokszáma [27]. A Matlabban a *butter()* függvény használatával érhető el ez a típusú szűrőtervezési módszer.

5.1.2. Burkológörbe generálás

A 5.1. ábrán az *ENV*(τ) blokkal jelölt lépés kimenete a jel abszolútértéke alapján képződik exponenciális átlagolással. Az egyszerű csúszóablakos átlagolás és az exponenciális átlagolás között az a különbség, hogy a csúszóablakban lévő minták az átlagban mind ugyanakkora súllyal rendelkeznek, az exponenciális esetben pedig az éppen érkező minta nagyobb súllyal rendelkezik, mint a régebbi minták egy lecsengő exponenciális függvény szerint. Jelen esetben az exponenciális függvény úgy van megalkotva, hogy a görbe alatti területe megegyezik az ugyanakkora időállandóval rendelkező csúszóablak területével. Mivel ezt a blokkot több helyen is használtam, Matlab függvényt valósítottam meg:

a függvény bemeneti paramétere az *fs* mintavételi frekvencia, a *tau* időállandó és a *dataIn* bemeneti adatvektor, amin az átlagolást szükséges elvégezni.

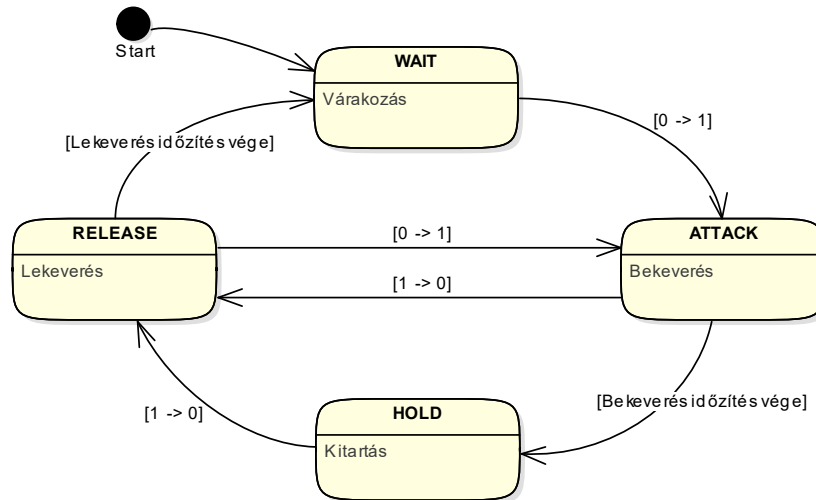
5.1.3. Küszöbölés

A 5.1. ábra szerinti *TR* blokkal jelzett küszöbölés annyit tesz, hogy ha a jel elér egy bizonyos szintet, akkor a kimenete egységnyi, egyébként pedig nulla.

5.1.4. Simítás

Ha a küszöbölés kimenetével vezérelnénk közvetlenül a jelet, könnyen előfordulhat nem nullaértékű jel esetén, hogy a kimeneten ugrás jelenik meg, amit a jelben kattánásnak hallunk. Ezt elkerülendő, minden esetben mikor a jel eléri a küszöbszintet, egy lineáris vezérlőjellel keverjük a jelet a kimenetre, így a kimeneti jelben csak folyamatos átmenet lehetséges.

Megvalósítása állapotgéppel lehetséges, amiben nyilvántartjuk, hogy éppen felkeverés, kitartás, vagy lekeverés állapotban vagyunk. Az állapotnak megfelelő kifejezés adja a kimenetet. A működését a következő ábra szemlélteti.



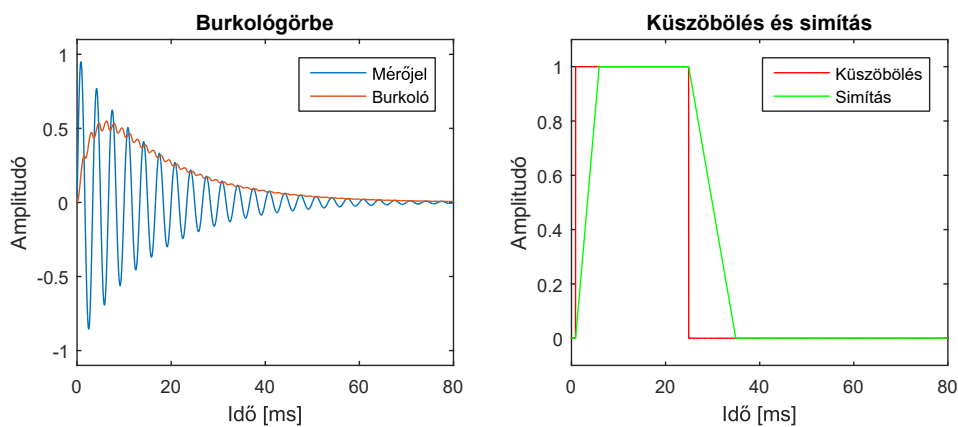
5.2. ábra. *Símtást végző modul belső állapotgépe*

A 5.2. ábra szerint program indulása után az állapotgép a WAIT állapotba kerül, itt várakozik a bemenet megváltozására. Ha a bemeneten logikai 1 érkezik, akkor az ATTACK állapotba lép, ahol a kimenet a kívánt meredekséggel felfelé indul. Abban a pillanatban, mikor a kimenet eléri a maximális értékét (1-et), akkor HOLD állapotba kerül. Itt egészen addig kitartja a kimenetet, amíg a bemeneti jel újra vissza nem lép 0-ba. Ekkor a lecsengési meredekség szerint a kimenet visszatér nullába, és a rendszer újra WAIT állapotba kerül.

Előfordulhat olyan eset is, amikor nem tudjuk bejárni a teljes állapotsorozatot, még a felkeverési fázisban visszavált a bemenet nullába, vagy lekeverés közben újra aktív lesz a bemenet. Az ilyen esetekben is biztosítani kell, hogy a kimeneten ne jelenhessen meg ugrás a jelben.

Az előfeldolgozás folyamata

A következő példa egy exponenciálisan lecsengő szinusz mérőjel segítségével mutatja be az előfeldolgozás lépéseit:

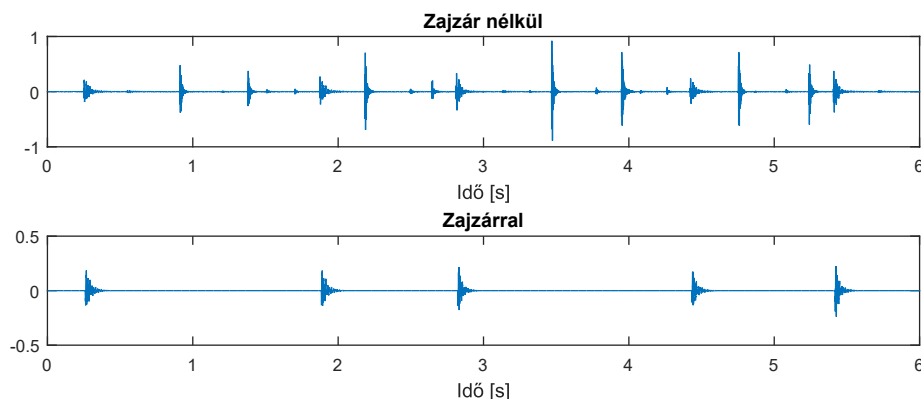


5.3. ábra. *A zajzár hatása a jelre mély ütés esetén*

A bal oldalon láthatjuk, hogy az átlagoló szűrővel megvalósított burkológörbe a jel meg-

jelenése után felfut, majd követi a mérőjel lecsengését. A jobb oldali diagramon pedig láthatjuk a küszöbölés logikai kimenetét ami a burkológörbe meghatározott szintje felett vált 1-be, valamint az ezt simító vezérlőjelet. Szemléltetési okokból az ábrán a simítási algoritmus időállandója eltúlzott mértékű. Valós körülmények között az időállandó nagyságrendileg 1 ms, amit úgy állítok be, hogy az impulzusjel kezdeti tranziense is már megjelenjen a kimeneten.

A következő ábrán megfigyelhető, hogy mit eredményez az előfeldolgozó szűrés egy valós mért jelen:



5.4. ábra. A zajzár hatása a jelre mély ütés esetén

A 5.4. ábrán egy cajon-on történő hangszeres játék felvétele látható, amiben magasabb és mélyebb tónusú ütések is vannak. A példában a mély ütéseket kiválasztó előfeldolgozási folyamat figyelhető meg, miszerint a szűrt jelben már nincs ott minden impulzus, valamint látszik, hogy a komparálás nem az eredeti jel szintje alapján történt.

5.2. Rezonátorbank megvalósítása

A hangszintézis legfontosabb építőeleme a rezonátorbank, ami párhuzamosan kapcsolt rezonátorszűrőket tartalmaz. Mindegyik szűrő ugyanazt a bemenetet kapja meg és a kimenetük pedig összegződik. A rezonátorbank egy alapprofrekvenciával rendelkezik (amire a legelső szűrő van hangolva), ez modellezi a dob hangolását. A többi rezonátor a kör alakú membrán módusai szerinti felharmónikusokra van hangolva. A Matlabban megvalósított rezonátorbank egyik bemeneti paramétere a frekvenciavektor, erre hangolja a szűrőket. A paraméterekben megadható a lecsengésvektor, ami tartalmazza a különböző frekvenciakomponensekhez tartozó lecsengési időket. Rendszerint ahogyan haladunk a magasabb frekvenciakomponensek felé, annál gyorsabb a rezonátorok lecsengése. A bemeneti amplitúdóvektor azt határozza meg, hogy a rezonátorok kimenete milyen aránnyal kerüljön a kimeneti jelbe.

A 3.4.4 pontban bemutatott membránfeszültség modulációt úgy építettem be a szintézisbe, hogy rezonátorbank kimenete visszacsatolásban képes elhangolni a szűrők frekvenciáját. Az elméleti levezetésből az következik, hogy a visszacsatolásban a kimeneti jel energiájával kell a szűrőket elhangolni. A gyakorlatban ez úgy jelentkezett, hogy a kisebb ütéseknél a jelenségnek nem volt hallható hatása, nagy ütéseknél viszont túlságosan do-

minált, így a végleges megvalósításban a jel abszolútértékével történik a visszacsatolás, és egy módosítható szorzóval lehet finomhangolni az elhangolódás mértékét.

Megjegyzendő, hogy az akusztikus doboknál játék közben nem módosítható a membrán rugalmassága és hajlítómerevsége így azzal, hogy a visszacsatolásban lévő erősítést változtatjuk, olyan hangzásokat is kaphatunk eredményül, ami a fizikai világban nem lenne megvalósítható.

Kiemelendő, hogy a membránfeszültség moduláció során minden mintára újra kell számolni a szűrőparamétereket a frekvencia változása miatt, ez jelentősen megnöveli az egy mintára jutó műveletek számát.

A fejlesztés során az elméleti modell ki lett bővítve úgy, hogy a visszacsatolásban a frekvencia-elhangolás a kimenet burkolója szerint történik, ezzel kiszűrve a hirtelen ugrásokat. Ezt az előfeldolgozásnál bemutatott átlagolással oldottam meg.

5.3. Pergőmodell

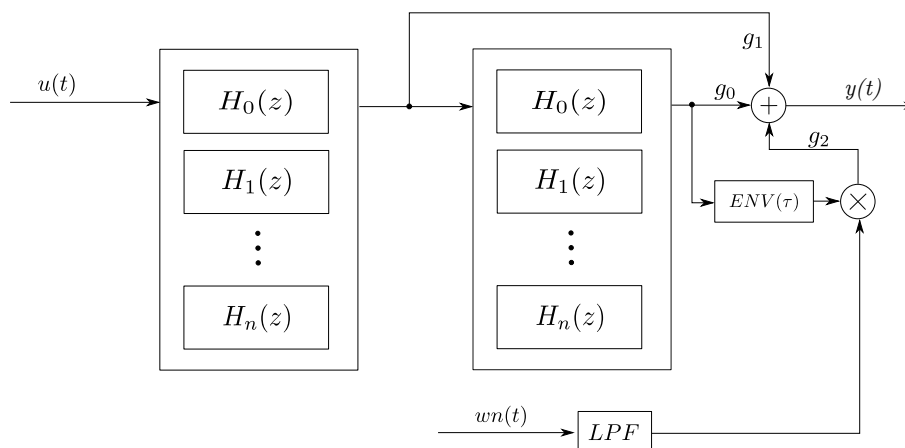
Itt a pergődob akusztikus megvalósításából indultam ki. Egy pergődob két dobtestre kifeszített membránból és egy sordonyból áll, ami az alsó membrán alá van kifeszítve. Ezt rendszerint 10 darab spirálrugó alkotja. A felső membrán megütése a két membrán közötti csatolás miatt az alsón is érvényesül, rezgésbe hozva a sordonyt is.

A pergődob szintetizálásához a 3.5 pont alapján rezonátorbankot hoztam létre. A két membrán csatolását úgy oldottam meg, hogy két rezonátorbankot futtatok, ami a két membránt modellezi. A felső kimenetét az alsó bemenetére kötöttem, így szimulálva a csatolást.

A rezonátorbankokat a kör alakú membrán módusaira hangoltam, a felsőt 200 Hz-es, az alsót meg kicsit elhangolva 205 Hz-es alapfrekvenciával.

A sordonyt pedig szűrt fehér zaj segítségével modelleztem. Azért, hogy ez a megfelelő pillanatban szóljon, az alsó membrán kimenőjelének burkolójával súlyoztam. Az így összeállt elrendezés látható a 5.5. ábrán.

A pergődob esetében a membránfeszültség moduláció kevésbé befolyásolja a hangzást, ezért ennek a modellezése nem került bele a szintézisbe.



5.5. ábra. Pergődob hangjának szintetizálása

Gauss eloszlású fehér zajt Matlabban legegyszerűbben a `wgn()` függvény segítségével lehet képezni. A zaj egy aluláteresztő szűrőn keresztül kerül a kimenetre, melynek törésponti frekvenciája kb. 5 kHz.

Megjegyzendő, hogy léteznek komplexebb modellek a pergődob, valamint a sordony hangjának modellezésére [25], [29], [36].

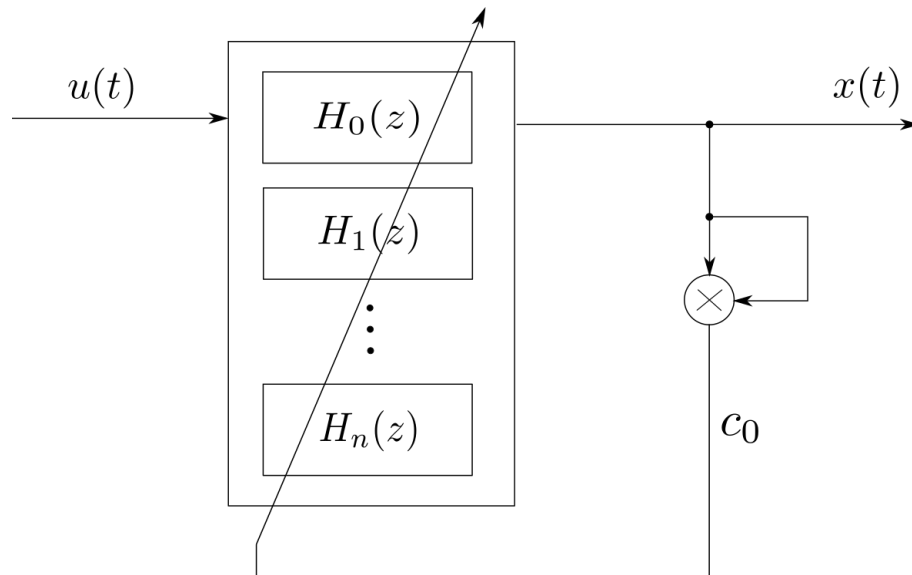
5.4. Lábdobmodell

A pergődobbal ellentétben a lábdob hangzásban jeletős hatása van a feszítettség növekedés hatására megugró módusfrekvenciáknak, így ez a lábdob modell szerves része. A módusfrekvenciák elhangolását egy visszacsatolással oldottam meg, miszerint a frekvencianövekedés nagysága a modell kimeneti jel abszolútértékével illetve energiájával arányos. Ha a jel energiáját használom fel a visszacsatolásnál, akkor a nagyobb ütéseknel sokkal nagyobb

a frekvencianövekedés, az abszolútérték esetében pedig már a kisebb erősségű ütéseknel is jobban jelentkezik a frekvencianövekedés hatása. Az ábrán látható c_0 változtatásával befolyásolni tudom, hogy a visszacsatolás mekkora frekvenciaváltozást hozzon létre. Ennek a paraméternek a módosításával olyan hangzás is kelthető, mely akusztikus hangszerben nem fordulhat elő.

A rezonátorbank alapfrekvenciáját 30-50 Hz tartományba állítottam be.

Az így konstruált széntézisstruktúra a 5.6. ábrán látható:



5.6. ábra. Lábdob hang szintetizálása

Ennek megfelelően a lábdob modellben a `res_filt()`; Matlab függvényt használtam.

6. fejezet

Dobszintetizátor megvalósítása beágyazott környezetben

6.1. Felhasznált hardverek

A beágyazott környezetben történő fejlesztések egyik legfőbb kérdése a központi feldolgozóegység kiválasztása. Egy elhibázva kiválasztott platform komoly korlátokat szabhat a kifejlesztett termék képessége és minősége tekintetében. A választott technológiának meg kell felelnie a feladatra mérnöki és gazdasági szempontból is. Ideális, de egyben szerencsés helyzet az, amikor a projekt elején teljesen specifikálni tudjuk a hardveres követelményeket. A választásban további szempont lehet, hogy melyik platformhoz juthatunk hozzá megfelelő időn belül, milyen fejlesztést támogató szolgáltatások vannak (pl.: az adott platformra való fejlesztéshez használt szoftverekért kell-e külön fizetni, ha igen mennyit), a platformra elérhetőek-e alapvető szoftvermodulok, illetve a dokumentáltság minősége és a fejlesztők korábbi tapasztalatai.

6.1.1. DSP kártya

Mivel jelen diplomaterv céljai között nem szerepelt a saját hangfeldolgozó modul megtervezése és legyártása, így alapelveként megfogalmazható, hogy a gyártók által kínált fejlesztőpanelekre szorítkozzam.

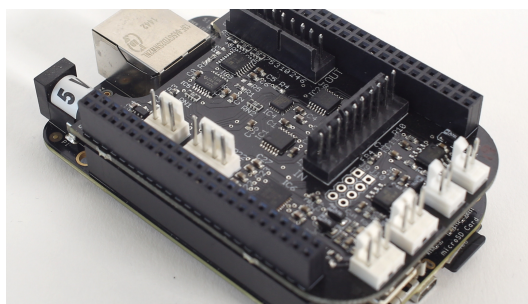
A valós idejű hangszintézishez használandó hardver esetében a legfőbb követelmények:

- **Audio bemenetek:** Rendelkezzen legalább 2 teljes értékű hangfrekvenciás analóg bemenettel, amely képes a cajonra szerelt szenzorok jelét fogadni.
- **Audio kimenetek:** Rendelkezzen legalább 2 kimeneti csatornával.
- **Számítási kapacitás:** Tudjon valós időben futtatni egyszerre két fizikai modellt. Becsléseim alapján ha kb. 100 szorzásműveletet el tud végezni minden hangmintára, az elegendő lesz.
- **Memória:** Szükséges az összes szűrőegytárható és paraméterkészlet, valamint a belső állapotok letárolása.

- **Komminuikáció:** Rendelkezzen legalább egy olyan kommunikációs csatornával, aminek a segítségével valós időben tudjuk a szintézis paramétereit hangolni.
- **Számábrázolás:** A platform képes legyen optimálisan 32-bites lebegőpontos műveletek végzésére.

A lebegőpontos architektúrára vonatkozó követelményt azért írtam elő, mert korábbi fejlesztések során már futottam bele olyan problémába, amit a fixpontos adatábrázolás okozott. Egy egyszerű példa erre, ha a program valamely pontján kíváncsiak vagyunk a jel energiára, ehhez négyzetre emelés művelet szükséges. Ebben az esetben fixpontos ábrázolásnál az ábrázolható tartomány a felére csökken, ami ennek a tartománynak a határán nem várt nemlineáris működéshez vezethet.

A keresést két irányból kezdtem meg. Egyik lehetséges út a nagyobb teljesítményű, akár többmagos, általános felhasználásra szánt beágyazott linux alapú fejlesztőkártya. Ezeknek a kártyáknak az átfogó elnevezése a *Single Board Computer* (SBC). Ilyen megoldás lehet pl.: egy *Raspberry Pi* [19] egy megfelelő audióinterfészsel. Ebben a termék kategóriában a legkomolyabban felmerülő megoldás az a *Bela IO* platform [7] volt. Ez a platform két modulból áll, egyrészt egy Texas Instruments Sitara AM335x (ARM Cortex-A8) alapú processorkártyából, mely rendelkezik a megfelelő memória modulokkal, tápegységgel, ki- és bemenetekkel (Beagle Bone Black [5]), valamint egy audió illesztőmodulból, amin található két audióbemenet, két audiókimenet, valamint nyolc szenzor be és kimenet.



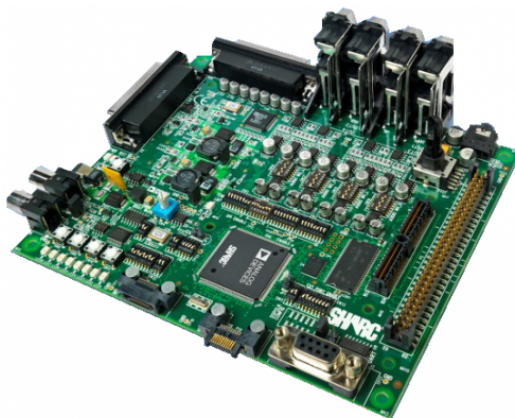
6.1. ábra. *Bela IO* fejlesztőkártya

Előnye ennek a kártyának, hogy a teljes szoftverkörnyezet elő van készítve rá, példa-programok segítik a fejlesztést, ami nagyban felgyorsíthatja a programozást. Hátránya, hogy ha belefutunk valamilyen alacsonyszintű hibába, vagy ha már túl szeretnénk lépni a prototípus fejlesztésén és saját hardvert tervezni, akkor mégis meg kell érteni a teljes hardver és szoftver működést, például saját audió codec kártya esetén meg kell írni teljes eszközzillesztő szoftverréteget, ami komoly linux fejlesztői tudást követel meg.

A másik irány a kimondottan audió jelfeldolgozásra szánt digitális jelfeldolgozó processzor (Digital Signal Processor - DSP) használata. Ezek általában kisebb belső órajelfrekvenciával működnek mint a linuxos kártyák¹, de a célspecifikus perifériák segítségével képesek magas hatásfokú műveletvégzésre.

¹A linuxos kártyák órajele nagyságrendileg 1 GHz, a DSP processzoroké, melyek erre a célra szóba jöttek modelltől függően 70-500 MHz.

A feladat megoldására az Analog Devices SHARC ADSP-21489 alapú fejlesztőkártyát választottam. Ez egy audió jelfeldolgozásra szánt célprocesszor, mely lebegőpontos műveletvégzési architektúrával rendelkezik [3].



6.2. ábra. *Analog Devices ADSP-21489 fejlesztőkártya*

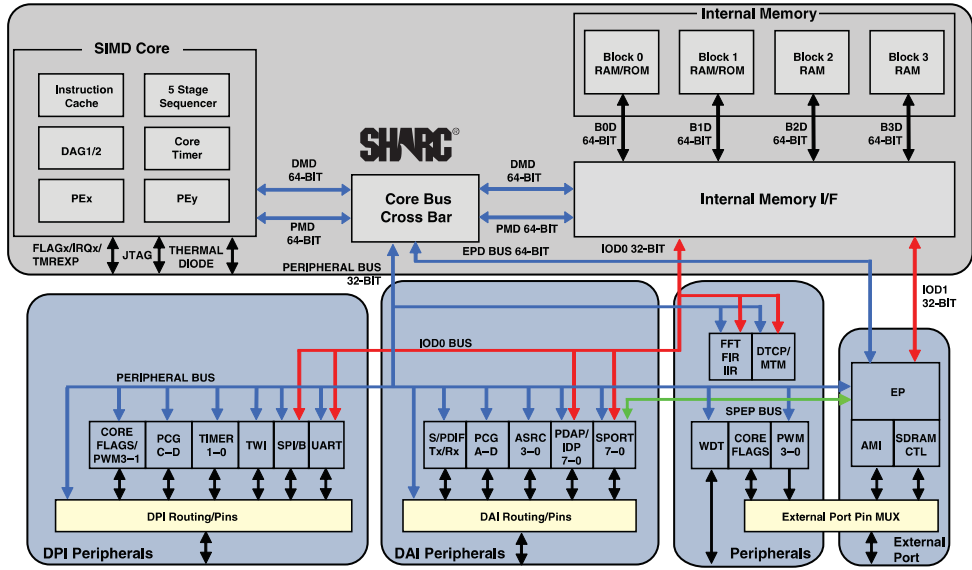
A fejlesztőkártyán elérhető egy digitális audió interfész (S/PDIF), 4 analóg bemenet és 8 analóg kimenet, ami a kártyán lévő AD1939 típusú audió A/D, D/A konverter IC-nek köszönhető. Rendelkezik egy RS232 kommunikációs csatlakozóval, valamint két bővítőporttal, amiken keresztül külső eszközöket is lehetséges a kártyához illeszteni. Nyolc szabadon felhasználható LED és 4 szabadon programozható nyomógomb segíti a kártyára történő szoftverfejlesztést.

A processzor rendelkezik 5Mbit chipbe épített RAM-mal, 450 MHz-es órajelfrekvenciával, SIMD műveletvégző architektúrával (Single instruction multiple data - egy művelet elvégzése egyszerre két adaton) valamint, nyolc nagysebességű soros porttal, ahol az adatot DMA tölti be a memóriába vagy olvassa ki onnan. A soros portok bekonfigurálhatóak minden elterjedt digitális audió formátumra (pl.: I2S - Inter IC Sound, TDM - Time Division Multiplex, stb.). Egy port egyszerre nyolc digitális audió be- és kimenetet tud kezelni. Nagy pontosságú órajel-generátor periféria gondoskodik a digitális audió kommunikációhoz szükséges órajelekről. Rendelkezik egy univerzális aszinkron soros porttal is, ami lehetővé teszi az egyszerű kommunikációt PC (RS-232) vagy másik mikrokontroller felé. A DSP teljes belső felépítése a 6.3. ábrán látható, a műveletvégző egység felépítése pedig a 6.4. ábrán.

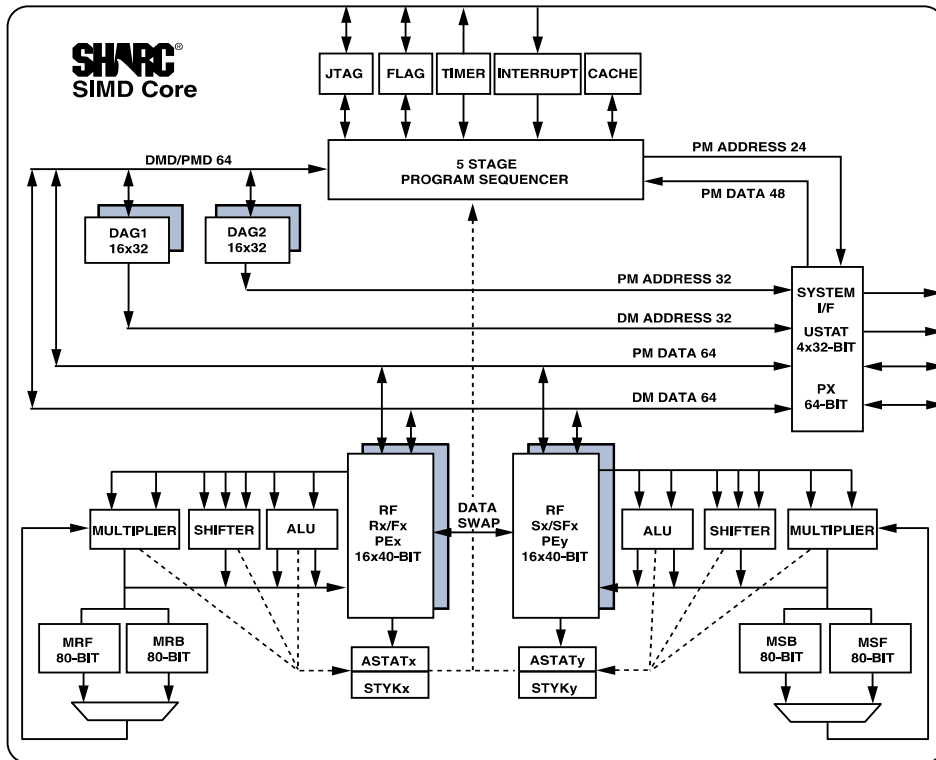
Előnye ennek a platformnak, hogy - a linuxos rendszerrel ellentétben - a teljes beágyazott kód rendelkezésre áll, a működéshez nem szükséges külső fejlesztő által írt szoftverkörnyezet. További előnye, hogy az alacsony szintű működés is konfigurálható, egyszerűen hozzáilleszthető új audió interfész és a hardveresen gyorsított FIR, IIR, és FFT műveletvégzés, melyekhez optimalizált gyártói függvénykönyvtár tartozik. Fontosnak tartom még kiemelni, hogy a gyártó jól dokumentált támogatást ad ahhoz, hogy saját hardvert tervezzünk hozzá, így termékfejlesztés esetén a hardvertervezési költségek alacsonyabbak lehetnek.

Hátránya, hogy a konfigurálása közvetlenül a perifériaregisztereken keresztül történik, így a konfigurálás során szükséges az összes periféria részletes ismerete.

A beágyazott kód fejlesztését az Analog Devices CrossCore Embedded Studio v2.6 in-



6.3. ábra. Analog Devices ADSP-21489 funkcionális blokkvázlata [3]



6.4. ábra. Analog Devices ADSP-21489 műveletvégző egység blokkvázlata [3]

tegrált fejlesztői környezetben végeztem.

6.1.2. Kezelőfelület

A modellező környezetben történő fejlesztés során, a futtatás nem valós időben történt, így a hangzás paramétereit csak a lefuttatások között tudtam hangolni. A DSP-ben a szintézis algoritmus valós idejű futtatása történik, így lehetőség van a hangzás paramétereinek azonnali manipulálására.

Több ötletem is volt arra, hogy hogyan tegyem lehetővé a hatékony paraméter beállítást. Felmerült, hogy soros porton számítógéphez csatlakoztatom és egy egyszerű paraméterállító szoftvert készítek. Lehetőség lett volna arra, hogy egy BLE (Bluetooth Low Energy) modul segítségével okostelefonos alkalmazáson keresztül tegyem lehetővé a vezérlést. Ezeknek az a hátránya, hogy a színpadi használatra korlátozottan lettek volna alkalmasak. Pl.: egy telefonhívás megszakíthatja a vezérlést.

A felmerült lehetőségeket mérlegelve végül egy másik fejlesztőpanel DSP-hez csatlakoztatásával oldottam meg a problémát. Itt a választást leginkább az befolyásolta, hogy melyik megoldás érhető el a legkisebb határidővel, így a már korábbi projektben használt STM32F7-es fejlesztőkártya mellett döntöttem.



6.5. ábra. Az STM32F746G Discovery fejlesztői kártya [11]

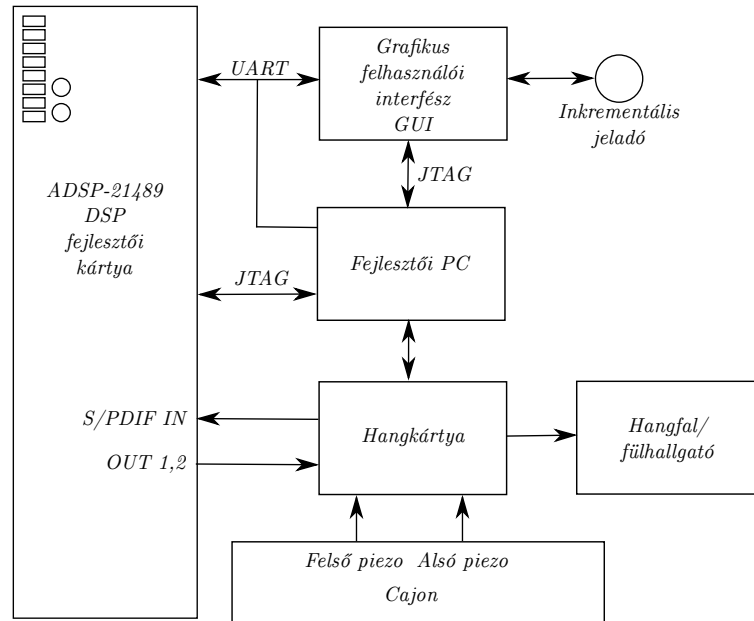
A fejlesztőkártya alapja egy STM32F746NGH6 típusú mikrokontroller, mely egy ARM Cortex-M7 maggal rendelkezik. Továbbá egy 4.3' inch átmérőjű színes LCD-TFT kijelző is megtalálható a kártyán, ami lehetővé teszi egy egyszerű felhasználói interfész megvalósítását.

6.1.3. Fejlesztéshez használt mérési elrendezés

A fejlesztés közben összeállított elrendezést a következő ábra segítségével szeretném bemutatni.

A 6.6. ábrán megfigyelhetjük, hogy fejlesztésben fontos szerepe volt a fejlesztői számítógéphez között hangkártyának is. Az elrendezés lehetőséget adott rá, hogy a fejlesztői gépről hangmintát tudjak bejátszani a DSP-n futó szintézis bemenetére, valamint a szintézis kimenetét rögzíteni tudjam, vagy a cajon eredeti hangja mellett meghallgatni.

A fejlesztést segítette, hogy a DSP fejlesztőpanelen lévő 8 ledet a kimeneti jelet mérő



6.6. ábra. Az STM32F746G Discovery fejlesztői kártya [11]

kivezélésjelzőként használtam, valamint a kártyán lévő két szabadon használható nyomógombot felhasználtam arra, hogy a bemeneti jelbe egységimpulzust juttassak, ezzel gerjesztve a fizikai modelleket.

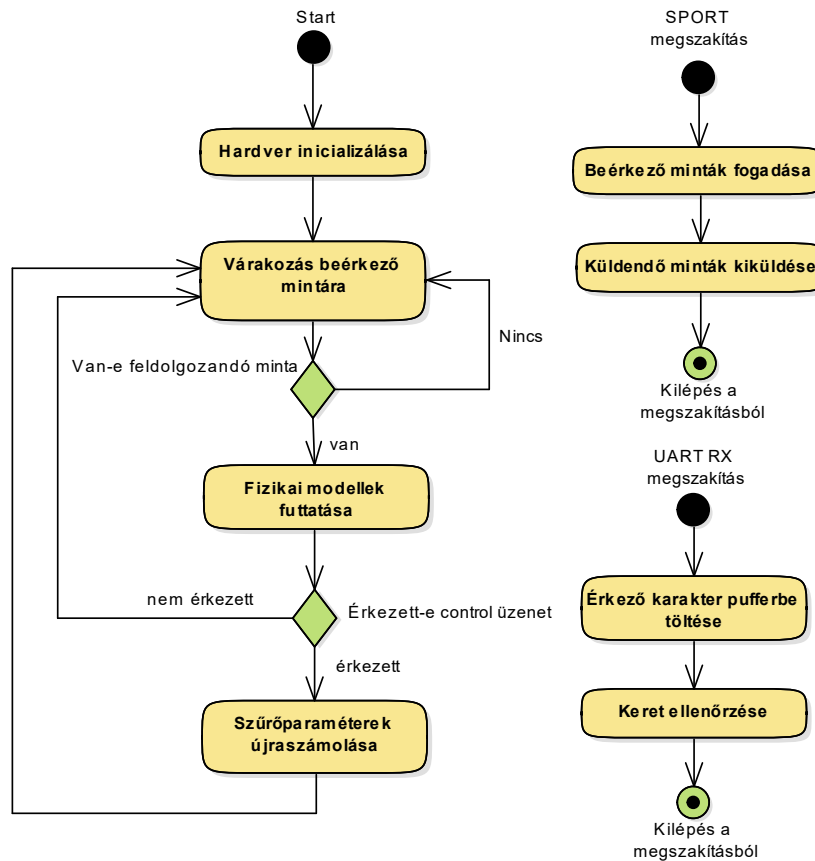
A DSP és a grafikus felhasználói interfészt megvalósító fejlesztőkártya között aszinkron soros kapcsolat van. Ezt a portot a PC-re is rákötve lehetőség adódik onnan is bájtsorozatot kiküldeni, így külön lehetett tesztelni az adó és a vevő oldalt.

6.2. Beágyazott szoftver magasszintű működésének áttekintése

A beágyazott rendszerben megvalósított valós idejű alkalmazások alapvető követelménye, hogy egy beérkező eseményre egy kritikus határidőn belül reagáljon. Jelen esetben a beérkező esemény a bemeneti csatornákon érkező új hangminta, a reakció pedig a szintetizált jel audió kiemeltre való kiküldése. A határidő felé pedig két követelmény van, egyrészt, hogy a bemenetre juttatott impulzusra érkező válasz az ember számára késleltetés mentes legyen (ez jellemzően 10 ms körül van), a másik, hogy a kimenetre a választott mintavételi frekvenciának megfelelő sebességgel új mintákat kell szolgáltatni.

Mivel a feldolgozás jellege olyan, hogy statikus esetben egy mintára jutó műveletvégzések száma állandó, nem szükséges összetett ütemező algoritmusokat használni a stabil működéshez. Akkor változhat meg a műveletvégzések száma, amikor beavatkozás történik a paraméterekbe, ilyenkor a szűrőegység újraszámítása plusz terhelést jelent. A tapasztalataim alapján a jelenleg implementált struktúra statikus esetben is csak néhány százalékát használja a DSP kapacitásának, így nem okoz gondot a paraméterállításakor felmerülő nagyobb kapacitásigény.

A 6.8. ábrán látható a program alapvető működésének folyamatábrája. Miután a DSP alapállapotba kerül a tápfeszültség felépülése után, első lépésben megtörténik a perifériák felkonfigurálása a kívánt működésre.



6.7. ábra. A DSP-n futó beagyazott kód magasszintű folyamata

A főprogram egy végtelen ciklust tartalmaz, mely ellenőrzi, hogy van-e feldolgozandó hangminta, ha igen, akkor beküldi a szintézist végző szoftvermodulba. További feladat a főprogram végén, hogyha érkezett olyan üzenet, amely valamely belső paraméter megváltoztatását kezdeményezi, akkor meghívja a funkciónak megfelelő kezelőfüggvényt.

Alapvetően két megszakítás játszik szerepet a szoftver működésében, egyik a SPORT (nagy sebességű soros port) megszakítás, mely a digitális audió kommunikáció folytán kerül meghívásra, illetve ha az aszinkron soros vonalon érkezik egy karakter akkor az UART RX megszakítás üt be.

6.3. Szoftvermodulok működése

6.3.1. Hardver inicializálás

A következő lépések szükségesek a DSP konfigurálásához:

1. **initPLL_SDRAM()**. A DSP alapvető működéséhez szükséges első lépésben a rendszerórajel és a perifériaórajelek beállítása. Ez függ a külső órajelforrás frekvenciájától és a kívánt működési frekvenciától. A beállítás a DSP-ben lévő PLL (Phase Locked Loop - fáziszárt hurok) konfigurálásán keresztül oldható meg. Ebben a lépésben történik meg a fejlesztőkártyán található SDRAM inicializációja is.
2. **InitSRU()**. A 6.3. ábrán megfigyelhető, hogy a perifériák nem közvetlenül valamely

processzorlábba vannak kötve, hanem egy útválasztó egység előzi ezt meg. Az SRU (Signal Routing Unit) segítségével határozhatjuk meg, hogy a fizikai processzorlábak melyik periféria melyik portjához legyenek hozzárendelve.

3. **Init1939viaSPI()**. Itt történik az AD1939 analóg audió interfész konfigurálása. Legfontosabb paraméterek a mintavételi frekvencia és a digitális audió adatfolyam formátumának megadása.
4. **InitSPORT()**. Az SPORT periféria végzi az audió adatfolyamok küldését és fogadását. Meg kell határozni itt is az adatfolyam formátumát, illetve azt a memóriatarományt ahová a beérkező és kimenő minták lesznek tárolva, ezzel együtt a DMA-t is bekonfigurálva. Ennek köszönhetően az audió adatfolyamok úgy kerülnek egy belső memóriaterületre, hogy a központi műveletvégző egységnek egy adatmozgató műveletet sem kell végeznie.
5. **InitSPDIF()**. Az S/PDIF digitális audióbemenetet engedélyezi.
6. **InitSRC()**. Az SRC periféria (Sample Rate Converter) lehetővé teszi, hogy a S/PDIF digitális audióbemenet mintavételi frekvenciájának nem kell megegyeznie a processzor belső mintavételi frekvenciájával, de mégis lehessen az onnan érkező adatot fogadni.
7. **InitUART()**. Az aszinkron soros interfészt állítja be a megfelelő formátumra és adatsebességre.
8. **InitSynthApp()**. Itt történik a hangszintézis inicializálása. Minden indulás után egy alapértelmezett paraméterkészletet használ amivel azonnal el is indul a szintézis.

6.3.2. Szintézis alkalmazás

A hangszintézishez szükséges szoftvermodulokat a következő módon osztottam fel:

`calc_biquad_coefs.c`

Ebben a modulban implementáltam a `calc_resonator_coefs()` függvényt, ahol történik egy rezonátor szűrőparamétereinek meghatározása a 3.5 pontban meghatározott módon.

A matematikai műveletek elvégzéséhez a gyártó által kiadott `math.h` könyvtárat használtam. A belső műveletek dupla pontosságú (64-bites `double`) lebegőpontos aritmetikával számolódnak, ezzel lényegében ugyanazzal a pontossággal számolok, mint ahogyan a modellező környezetben is.

Az eredeti matematikai összefüggések komplex számokat használtak, ezeket szükséges volt átírni olyan alakba, amit a beágyazott környezetben is kezelni tudok.

Ezt az Euler-képlet segítségével oldottam meg:

$$e^{ix} = \cos(x) + i \sin(x) \tag{6.1}$$

envelopes.c

Az `envelopes.c` tartalmazza azokat a programrészleteket, amelyek a burkológörbe számításához tartoznak, valamint ez tartalmazza a fejlesztőkártyán található LED-ek vezérléséhez használható jelszintmérést.

A számításokhoz itt is felhasználtam a `math.h` gyári könyvtárat. A burkolósűrítő együtthatói egyszeres pontosságú lebegőpontos aritmetikával számolódnak, ebben az esetben nincs olyan nagy pontosságra szükség mint a rezonátorok esetében.

led_if.c

A LED interfész modul felel a fejlesztőkártyán lévő LED-ek és nyomógombok működéséért. Itt történik az SRU konfigurálása, valamint a megfelelő processzorlábak be- és kimenetbe állítása, megszakítások konfigurálása.

Egy nyomógomb megnyomása megszakítást idéz elő, ennek segítségével lehet a bemeneti jelfolyamba impulzus generálni.

main_process.c

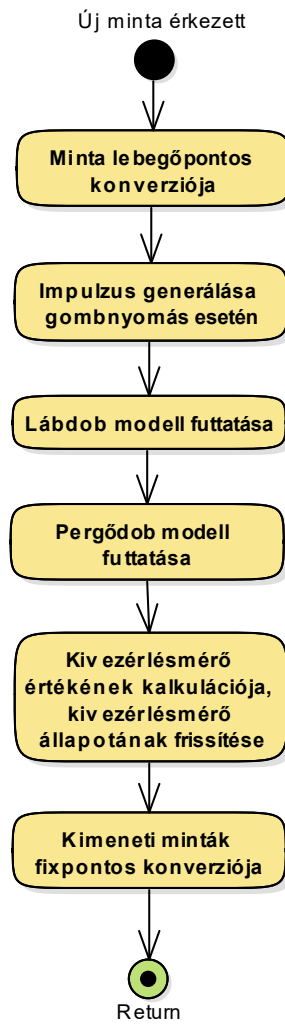
A szintézis algoritmus legfelső modulja a `main_process.c`. Itt történik a többi alegység inicializálása és egymáshoz kapcsolása.

Ha rendelkezésre áll bemeneti hangminta, akkor belép a szintézist végző ágba. A bemenetekről (A/D konverter és S/PDIF digitális audió bemenet) 24-bites bitmélységgel rendelkező előjeles fixpontos adatok érkeznek. Ez egy előjelbitet és 23 törtrész bitet jelent. Matematikai reprezentáció szempontjából a beérkező adat értékkészlete -1 és 1 közé esik. Fontos, hogy nincs 24 bites (3 bájtos) adattípus, csak 16 és 32 bites. Emiatt a 24 bites adatkeretet 32-bites változó tartalmazza, úgy, hogy a legfelső bájt marad használatlanul. Ez azt is jelenti, hogy az előjelbit nem a legfelső pozícióban, hanem egy bájjal alacsonyabban helyezkedik el, így ahhoz, hogy a feldolgozáshoz szükséges lebegőpontos formátumra alakíthassam szükséges az adat 8 bittel való eltolása.

A kimenetnél ennek a műveletnek a fordítottja is szükséges, hogy a feldolgozott lebegőpontos adatot a D/A konverterre lehessen küldeni.

Megjegyzendő még, hogy a lebegőpontos formátumra áttéréssel a feldolgozás során nagyobb az ábrázolható dinamikai tartomány. Ez a tulajdonság növeli a rendszer stabilitását, nagyobb pontosságot eredményez, de körültekintően kell eljárni, mikor a kimenő adatot visszaállítjuk az eredeti fixpontos ábrázolásra, ugyanis nem tudjuk 24-biten a 32-bit-en ábrázolható összes információt letárolni.

A mintavételi frekvencia szoftveres leosztásával oldottam meg a lassabb időzítéseket. Ennek segítségével meghívódik egy időzített függvény 10 ms-enként (`u10s_hook()`) és egy másodpercenként (`s_hook()`). Abban az esetben hasznos ez, ha fix időzítéssel szeretnénk például a soros portra valamilyen belső változó értékét kiírni. Itt valósul meg a kivezérlésjelzőnek használt LED-ek állapotfrissítése is, ez 10 ms-enként történik.



6.8. ábra. `main_process.c` folyamatábra

modal_bank_kick.c

Ebben a modulban valósul meg a lábdob modell. A modell alapját képző rezonátorbank paraméterei induláskor egy alapértelmezett beállítás szerint vannak meghatározva. A szűrőparaméterek a `calc_resonator_coeffs()` segítségével számolódnak a beállított tulajdonságoktól függően.

Mivel a rezonátor egy IIR szűrő, a gyártó `filter.h` könyvtárában megtalálható `biquad()` függvényt használni lehetett a megvalósítás során. Előnye, hogy optimálisan használja a DSP-t. A gyári könyvtár további optimalizált függvényt biztosít arra az esetre, ha több SOS (Second Order Section) blokkot szeretnék kaszkádosítani egymással sorba kapcsolva. Viszont ezt a további optimalizálást nem tudtam használni, mert a rezonátorok párhuzamosan kapcsolódtak a szintézis struktúrája szerint.

Az implementált `float modal_bank_kick(float in)` függvény minden hangmintára lefut egyszer és egy hangmintát ad kimenetül. Az első felében minden mintánál újraszámolódnak a szűrőparaméterek, mert a membránfeszítettség moduláció miatt az alapfrekvenciák folyamatosan változnak.

A következő lépésben történik a rezonátor szűrőbank futtatása. Ennek a kimenetét kapja meg a feszítettség kiszámolásához egy simítószűrő.

A függvényben implementálásra került még egy limitáció a kimenetre, mert a következő lépésben a fixpontos konverzió csökken a dinamik tartomány, valamint a membránfeszítettség modulációja sem növekedhet minden határon túl.

modal_bank_snare.c

A pergődob szintézis modul szerkezete nagyon hasonlít a lábdob modellhez. A fő különbség, hogy itt nincs membránfeszítettség moduláció, de két rezonátorbank van, ami a felső és az alsó membránt modellezi. A sodrony hanghatását az alsó membrán kimenőjelére ültetett burkolóval vezérelt fehérzaj modellezi.

Az első rezonátorbank kimenetét használja fel a második szűrőbank.

A fehérzajt a `rand()` függvény segítségével állítottam elő. Minden minta esetén lefut a függvény, így kapok eredményül időfüggvényt. A függvény kimenete a teljes 32-bites fixpontos tartományt lefedi, emiatt úgy skálázom a jelet, hogy a kívánt tartományba essen.

process_serial.c

Ez a modul felel azért, hogy a soros porton beérkezett üzenetet feldolgozza és továbbítsa az információt.

A soros porton ASCII karakterek hordozzák az adatokat. A binárisan közölt adat előnye, hogy egységnyi nyers információt a legtömörebben közli, a karakter alapú átvitel viszont lehetővé teszi, hogy egy egyszerű PC-n futó soros port terminálból (pl.: Putty) közvetlenül adatot lehessen küldeni, ezért gyorsabban tesztelhető.

Az adatformátum kétszer négy karakterből épül fel. Az első négy karakter (0000 - 9999) azonosítja, hogy melyik paraméterre vonatkozik az információ, a második négy karakter pedig a paraméter értékét tartalmazza.

A C könyvtári függvénykönyvtárban található `atoi()` függvény segítségével lehetséges karaktersorozatot integer típusú változóvá alakítani.

A későbbi továbbfejleszetheőség miatt felosztottam a szoftvermoduloknak egy-egy azonosító tartományt, amelyben az ahhoz tartozó paraméterek vannak. Pl.: a lábdob modellhez az 1000 és 2000 közötti azonosítók tartoznak. Ha ebbe a tartományba esik a beállítandó paraméter, akkor a `set_kick_param(param_id, param_value_str)` függvény kerül meghívásra.

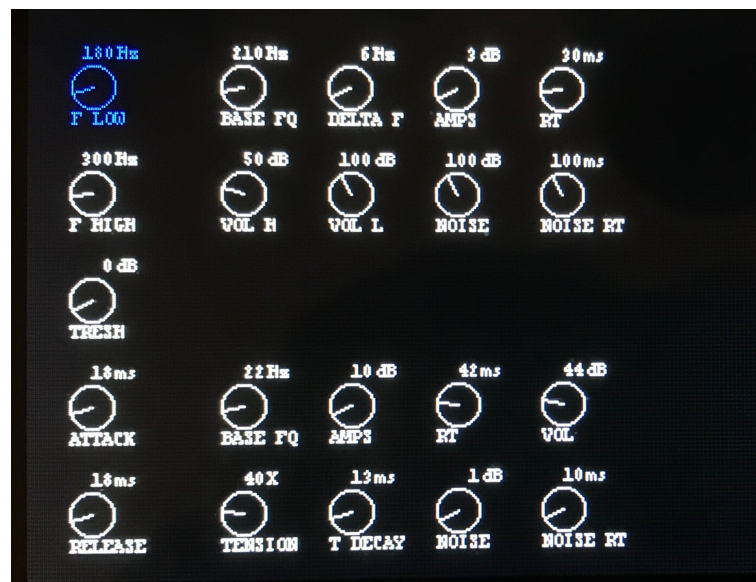
Kiemelném még, hogy a fejlesztés során a DSP-hez csak blokkoló elven működő soros port kezelőt találtam a példakódok között, így implementálásra került egy teljesen megszakításos rendszerű soros port kezelés. Ez azt jelenti, hogy se a küldés se a fogadás közben nem várakozik a processzor. Minden kiküldött és fogadott karakterhez egy megszakítás tartozik. Továbbfejlesztési lehetőség, hogy egy teljes kiküldött vagy fogadott keret összesen csak egy megszakítást eredményezzen. Ezt a DMA periféria segítségével lehetne megoldani.

7. fejezet

Beágyazott rendszer kezelőfelülete

Ahogy a 6.1.2 pontban említettem, felhasználói interfész gyanánt egy STM32F746NGH6 típusú fejlesztőkészletet használtam.

A választásban az is közrejátszott, hogy a kártyát az ARM által fejlesztett Mbed [14] platform segítségével programozni tudtam. Ez a platform magában foglal egy beágyazott operációs rendszert és egy olyan absztrakciós réteget, mely segítségével az alacsonyszintű hardverek konfigurációja is egyszerűvé válik. A fejlesztésben azt használtam ki, hogy a fejlesztőkártyához már készen használható programmodulok voltak elérhetőek, mint pl. a kijelző illesztése a processzorhoz, grafikus könyvtár, aminek segítségével egyszerű objektumok és karakterek jeleníthetőek meg, valamint a kijelzőn található kapacitív érintésérzékelő is elérhető. Teljesen megoldott a soros port kezelése is, csak a kívánt beállítások megadására volt szükség (melyik processzorláb, formátum, adatsebesség).



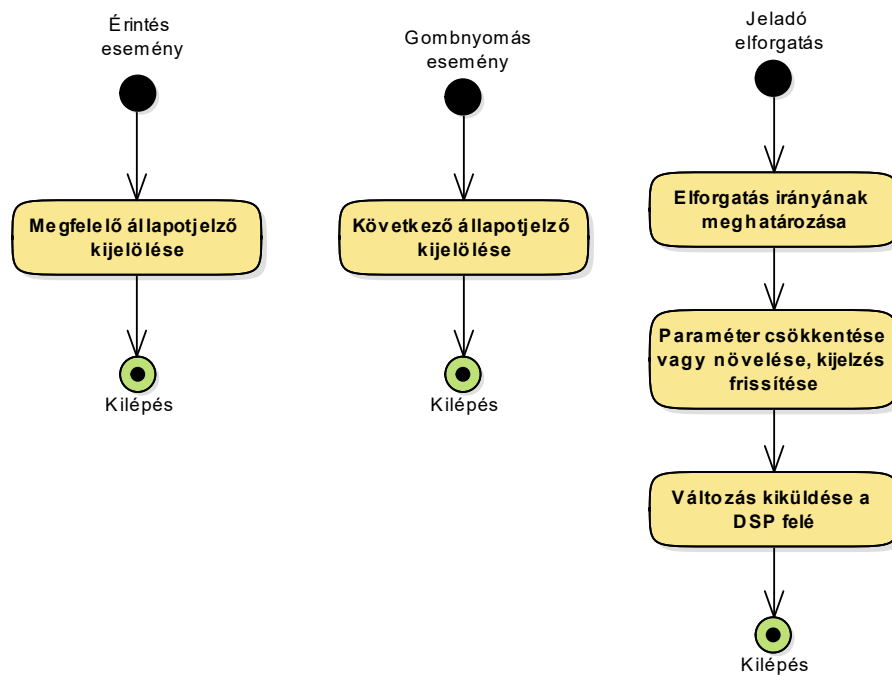
7.1. ábra. Dobszintetizátor kezelőfelülete

Felhasználói szempontból a működése a következő. Minden újrainduláskor leküldi a DSP-re az alapértelmezett paraméterkészletet, hogy ebben az esetben is megmaradjon a szinkronizáció a DSP-vel. A grafikus interfészen három csoportban vannak a virtuálisan elfor-

gatható állapotjelzők. Külön csoportban találhatóak az előfeldolgozásra vonatkozó paraméterek, külön a pergődobra és a lábdobra vonatkozóak. Mindig pontosan egy állapotjelző van kijelölt állapotban, újraindulás után a bal felső sarokban lévő. A kijelölést az jelzi, hogy az állapotjelző fehérről kék színre változik. A kijelölés pozíciója kétféleképpen változtatható meg. A fejlesztőkártyához csatlakoztatott inkrementális jeladó megnyomására a kijelölés a szomszédos állapotjelzőre ugrik, illetve a megfelelő állapotjelző megérintésével is kiválaszthatjuk a módosítani kívánt paramétert. Az így kiválasztott paraméter a jeladó elforgatásával módosítható. Minden állapotjelzőhöz tartozik egy felirat, ami a paramétert azonosítja, és egy érték a megfelelő mértékegységgel. Minden esetben, amikor paraméterváltozás történik, a soros porton kiküldésre kerül a módosított érték, de leggyakrabban 100 ms-onként.

7.1. Szoftverterv magasszintű áttekintése

A felhasználói interfészt megvalósító szoftver alapvetően eseményvezérelt módon működik. Ha nem történik külső beavatkozás, akkor a processzor alapállapotban várakozik.



7.2. ábra. Kezelőfelület eseményvezérelt működése

Háromféle eseményt fogad a szoftver. Érintés esemény akkor keletkezik, ha megérintjük a kijelzőt. Ha az érintés környezetében van állapotjelző, akkor kijelölésre kerül. Gombnyomás eseményt a csatlakoztatott jeladó megnyomása generál, ennek hatására a következő paraméter kerül kijelölésre. A jeladó elforgatása is eseményt generál. Ekkor eldönti a szoftver, hogy az adott paramétert növelni, vagy csökkenteni kell. Abban az esetben nem történik paraméterállítás, ha az adott állapotjelző már végállapotban van.

Az események főprogramban való lekezelése jelzőbiteken keresztül történik. Hogy a DSP-t ne terheljem túl egyszerre sok üzenettel, a főprogramba egy várakozás van beiktatva,

lassítva a frissítési sebességet. Felhasználói szempontból nem zavaró, hogy a kijelzett állapotjelzők csak 100 ms-onként frissülnek.

7.2. Szoftvermodulok működése

Az Mbed platform C++ nyelven íródott, így nekem is lehetőségem volt objektum orientált módon programozni.

A belső működést öt osztály határozza meg:

LCD_DISCO_F746NG

Ez az osztály valósítja meg a kijelzővel való kapcsolatot és a grafikus funkciókat. Szöveget kirajzolni egy meghatározott pixelkoordinátára a `DisplayStringAt` függvénnyel lehet, az előtte beállított tulajdonságok szerint (betűtípus mérete, színe, háttérszín). Téglalapot a `FillRect()` metódus segítségével lehet kirajzolni, kört a `DrawCircle()` metódussal, vonalat pedig a `DrawLine()` metódussal. Minden esetben négy függvényparamétert kell megadni, ezek segítségével határozhatjuk meg a kirajzolás pontos helyét és az objektum méretét.

TS_DISCO_F746NG

Az érintésérzékelő eseményeit tudjuk ennek az osztálynak a segítségével elérni. A főprogramban a `GetState()` tagfüggvény meghívására kapunk egy állapotleíró struktúrát, amelyből kiolvashatjuk a belső állapotokat. Megkapjuk, hogy éppen hány érintést detektált és ezeknek a pixelkoordinátáit.

Serial

A soros portok kezelésére a `Serial` osztály használható. Inicializáláskor megadható, hogy melyik processzorlábakat szeretnénk használni és milyen adatsebességgel. Az alapértelmezett formátum a 8N1, tehát 8 adatbit, 1 stopbit és nincs paritásbit. Ha a processzorláb helyére az `USBTX`, `USBRX` értékeket írjuk, akkor az USB-n keresztül a fejlesztői számítógépre csatlakoztatott fejlesztőkártyával virtuális soros port segítségével tudunk adatot cserélni. Az `attach()` tagfüggvény segítségével megadhatunk egy általunk megírt függvényt, ami meghívódik az adott porton érkező karakter hatására, illetve a `printf()` tagfüggvénnyel tudunk karaktersorozatot küldeni.

mRotaryEncoder

Az inkrementális jeladó kezelését oldja meg. Inicializáláskor meg kell adni, hogy a jeladó melyik portokra van rákötve. Inicializálás után bármikor lekérdezhető egy belső számláló, ami a forgatás hatására inkrementálódik vagy dekrementálódik iránytól függően. Az `attachSW()`, `attachROTCW()` és a `attachROTCCW()` metódusok segítségével megadhatunk olyan általunk definiált megszakításkezelő függvényeket melyeket szeretnénk ha

meghívódnának gombnyomáskor, óramutató járása szerinti és azzal ellentétes irányba való elforgatáskor.

Knob

A kijelzőn megjelenő állapotjelző megvalósítását ez az általam megvalósított osztály végzi. Inicializáláskor meg kell adni az állítani kívánt paraméter négy karakterből álló azonosítóját, a néhány karakterből álló feliratot, ami a kijelzőn megjelenjen alatta, a pozícióját a kijelzőn, a mennyiség mértékegységét, egy skálafaktort, az alapértelmezett értékét, amit bekapcsoláskor felvesz, valamint referenciát a kijelzőkezelő és az érintőpanelkezelő osztályokra.

A működésének magját a `drawKnob()` függvény adja. Meghívására a tagváltozóknban tárolt értékek alapján kirajzolja a kijelzőre az állapotjelzőt, és a módosított paraméterértéket kiküldi a soros porton a DSP-nek. A tagváltozók az inicializáláskor megadott paramétereken kívül tárolják, hogy éppen kijelölt állapotban van-e, illetve, hogy mi a beállítás aktuálisan felvett értéke. Az állapotjelzőt egy kör és egy vonal kirajzolásával valósítom meg. A vonal egyik végpontja mindig a kör középpontja, a másik végpontja pedig a megjelenítendő értéktől függ.

A `setValue()` és a `getValue()` tagfüggvények segítségével módosíthatjuk vagy kérdezhetjük le az aktuális beállított értéket.

Az `isTouched()` függvény pedig logikai 1-el tér vissza, ha a paraméterben megadott pixelkoordináta fedésben van az állapotjelzővel. Ez használható arra, hogy ha érkezik egy érintés esemény, akkor arról el lehessen dönteni, hogy melyik állapotjelzőhöz tartozik.

A főprogramban létrejön egy `KnobList[]` nevű tömb, amiben minden állapotjelző referenciája le van tárolva, ezáltal könnyebben kezelhetőek.

8. fejezet

Beágyazott rendszer vizsgálata

Minden szoftverfejlesztés kulcseleme a megvalósított rendszer átfogó tesztelése. Termékfejlesztés esetében általában elmondható, hogy a tesztelés minősége nagy szerepet játszik a végleges termék minőségében is. A tesztelés pontos menete közvetlenül a termék funkcionalitásától és a specifikációjától függ, de általánosan elmondható, hogy először a részegységekről önállóan meg kell bizonyosodni, hogy megvalósítják a kívánt specifikációt, aztán ha külön-külön mindegyik részegység helyesen működik, akkor rendszerszinten is bizonyosságot kell szerezni az elvárt működésről.

A fejlesztés során az első lépés volt egy olyan DSP szoftver létrehozása, mely feldolgozás nélkül átengedi az adatot a processzoron. Ezen keresztül tesztelni tudtam a különböző számábrázolások közötti konverzió helyességét, valamint a rendszer alapvető stabilitását.

Szerencsére nem tapasztaltam működésbeli hibákat ezen a ponton, így megkezdhettem a szintézisalgorithmus implementálását.

8.1. Rezonátorbank vizsgálata

Egy megvalósított rezonátor tesztelés szempontjából két részből áll. Egyik programrészlet azért felelős, hogy a megadott feltételek alapján (középfrekvencia, lecsengés, amplitúdó) kiszámolja a szűrőegyütthatókat, a másik pedig a szűrő, illetve szűrőbank futtatásáért felel.

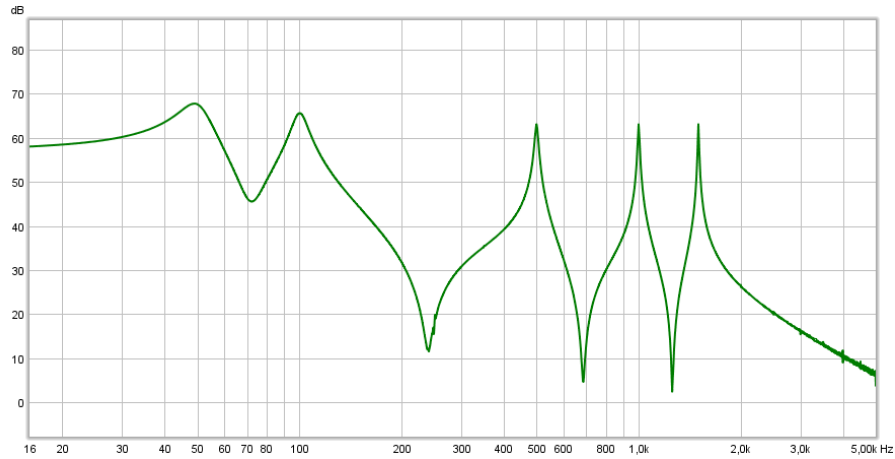
A szűrőegyütthatók kiszámolásának helyessége egyszerűen tesztelhető. A tervezett működési tartományt minél részletesebben lefedve meg kell határozni a bemeneti paramétereket, lefuttatni az algoritmust ezekre a bemenetekre a DSP-n, majd ellenőrizni az értékeket.

A vizsgálatom eredménye az lett, hogy az kiszámolt együtthatók megegyeznek a Matlabban kiszámított értékekkel a dupla pontosságú lebegőpontos számábrázolás pontosságával.

A szűrők futtatásának helyességéről úgy bizonyosodtam meg, hogy egy segédprogrammal, sepiert szinuszjel bemenettel megvizsgáltam a szűrők átviteli karakterisztikáját. Egyik esetben beállítottam 5 rezonátort 50 Hz, 100 Hz, 500 Hz, 1 kHz és 1,5 kHz frekvenciákra 30 ms-os lecsengéssel.

A következő ábrán látható a mérés eredménye:

A 8.1. ábrán 5 párhuzamosan kapcsolt rezonátor átviteli karakterisztikája látható, mely megfelel a várt karakterisztikának.

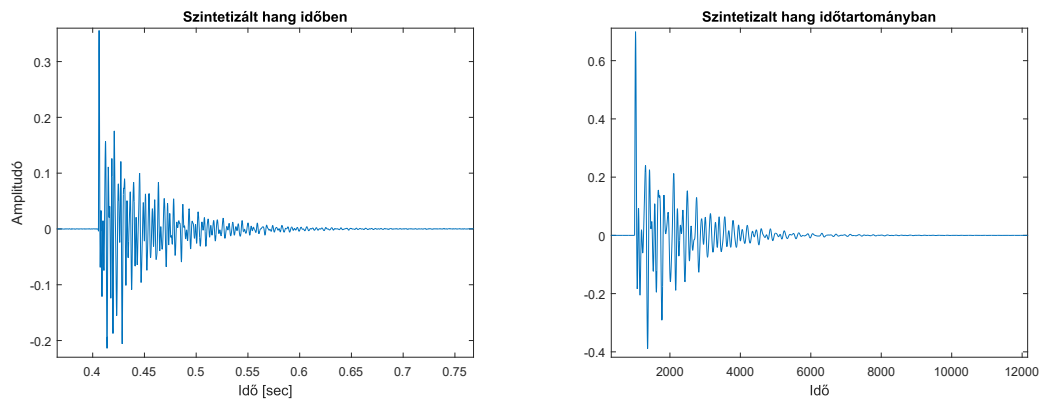


8.1. ábra. Rezonátorok tesztelése

8.1.1. Pergőmodell vizsgálata

A megvalósított modellt impulzusokkal gerjesztettem és vizsgáltam az így keletkező válaszjeleket.

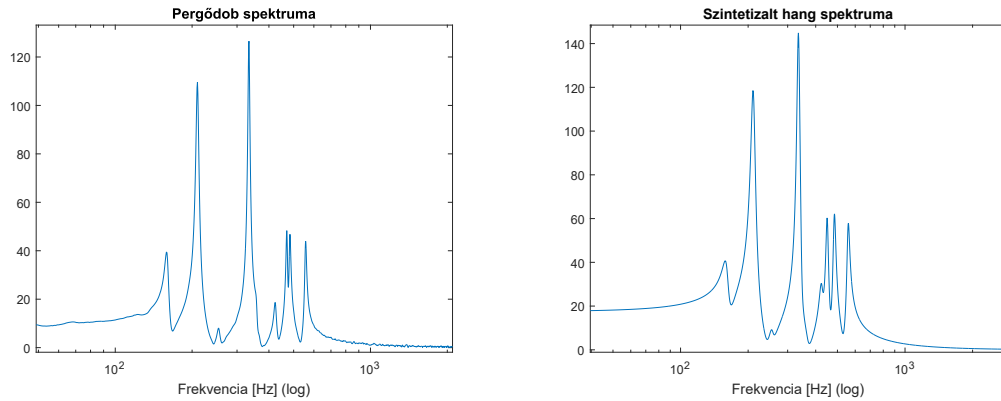
A pergőmodell két rezonátorbankból és egy vezérelt zajgenerátorból áll. Mivel a rezonátorbank helyes működéséről már megbizonyosodtam, azt vizsgáltam meg, hogy két rezonátorbank egymás után kapcsolva is megfelelően működik-e. További lehetőség az ellenőrzésre, hogy a DSP-n valósidőben futó modell kimeneteit összehasonlítom a modellező környezetben implementált modell kimeneteivel. Ahhoz, hogy a hasonlóságot elemezni tudjam, mindkét esetben ugyanazt a bemeneti paraméterkészletet használtam.



8.2. ábra. Pergődob jelalak: balra a DSP, jobbra a Matlab kimenete

A 8.2. ábra a pergő modell kimenetét ábrázolja az idő függvényében. A felső membrán alapharmónikusa 210 Hz, az alsó membrán elhangoltsága a felsőtől 50 Hz, az alapharmónikus lecsengési időállandója pedig 52 ms. Látható, hogy 0.6 - 0.65 s-nél, azaz a lecsengési időállandó 4-5x-ösénél teljesen lecseng a jel, ez megfelel az elvárt működésnek.

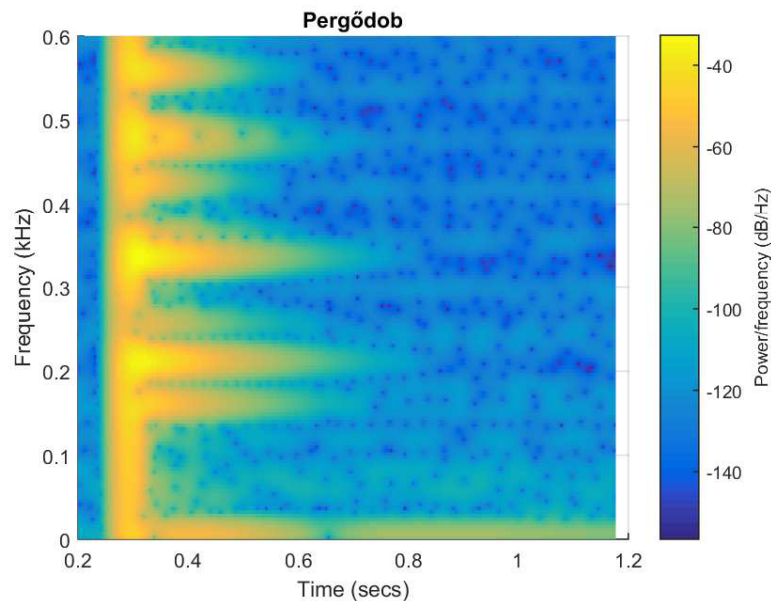
A 8.3. ábrán látható a DSP-n és a Matlabban futtatott pergődob válaszáinak spektruma. Jól látható a két membrán alapharmónikusa (210 Hz és 160 Hz), valamint a felharmónikusok. Természetesen lehetséges úgy hangolni az alapharmónikusokat, hogy a felharmónikusok egybeessenek, ezen a frekvencián nagyobb erősítést okozva. Továbbá megfigyelhető,



8.3. ábra. Pergődob spektruma: balra a DSP, jobbra a Matlab

hogy a Matlab szkript kimenete jellegre teljesen azonos. A kisebb eltérések származhatnak abból, hogy a két modell teljesen eltérő kódbázist használ a megvalósításhoz, valamint a DSP kimenetét egy analóg csatornán keresztül rögzítettem.

A frekvenciakomponensek időbeli viselkedését spektrogramm segítségével lehet ábrázolni.



8.4. ábra. Pergődob válaszáinak spektrogrammja

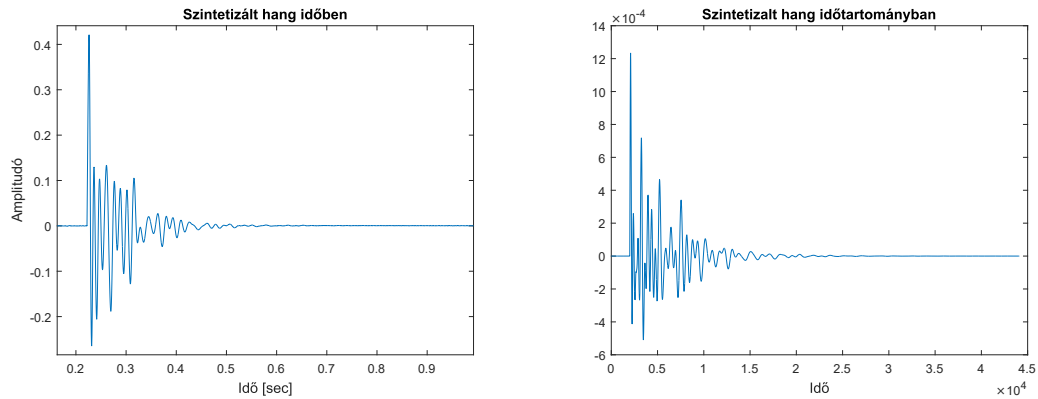
A 8.4. ábrán megfigyelhető, hogy a nagy frekvenciás komponensek hamarabb lecsengenek, a rezonátorbank konfigurációjának megfelelően.

8.2. A lábdobmodell vizsgálata

A lábdob esetében csak egy rezonátorbank fut, de a membránfeszítettség-moduláció modellezése révén visszacsatolást is tartalmaz. A kimeneti jel energiájával arányosan megnövekszik a membrán alapharmonikusának és a felharmonikusainak a frekvenciája. Hasonlóan a pergődobhoz, itt is összehasonlítottam a modellező környezetben és a DSP-n szintetizált

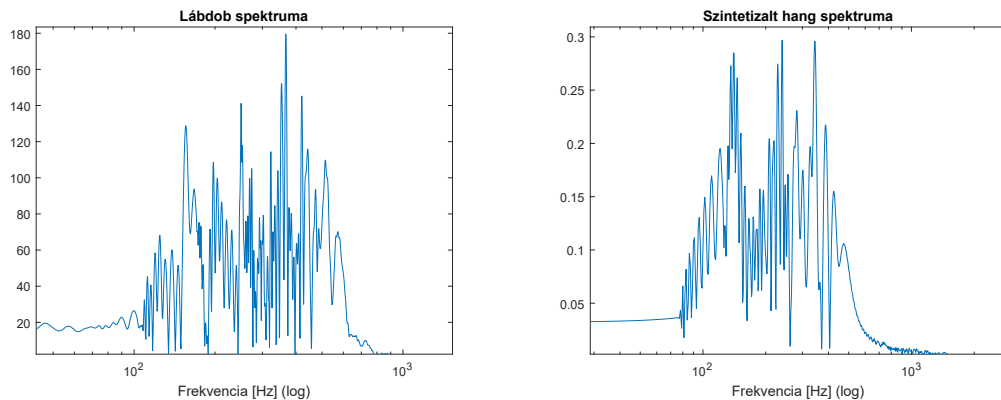
jeleket.

A pergődobhoz hasonlóan itt is megvizsgáltam a modell impulzusválasztát.



8.5. ábra. Lábdoz jelalak: balra a DSP, jobbra a Matlab kimenete

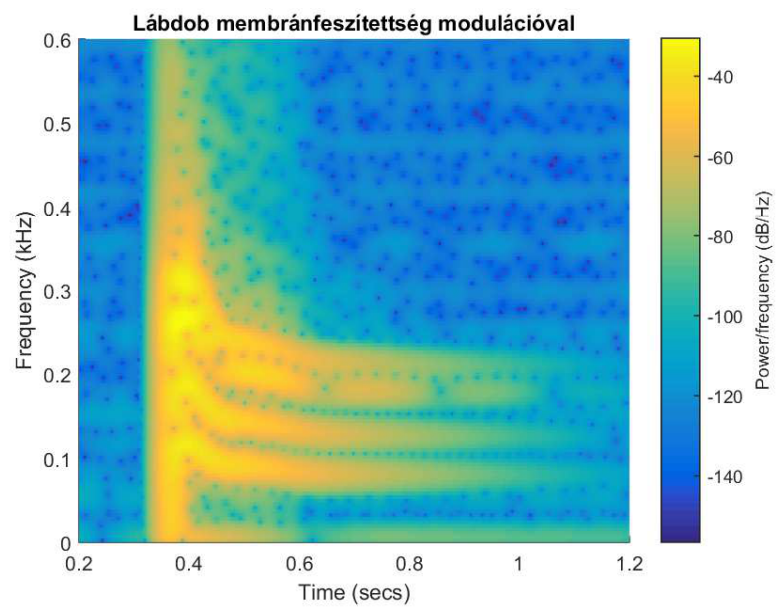
A 8.5. ábrán a válaszjel időfüggése látható. Az alapharmónikus frekvenciája 78 Hz, a lecsengési időállandó pedig 97 ms. Ha összehasonlítjuk a pergőmodell és a lábdozmodell spektrumát, alapvető különbségeket láthatunk.



8.6. ábra. Lábdoz spektruma: balra a DSP, jobbra a Matlab

A 8.6. ábrán észrevehetjük, hogy a módusfrekvenciákhoz tartozó csúcsok kevésbé tisztán láthatóak, mivel ezek a frekvenciakomponensek időben változnak. Továbbá látható itt is, hogy a két megvalósítás kimenete ugyanazt a jelleget hordozza. A környezeti sajátosságok miatt van strukturális különbség a visszacsatolás megvalósításában, ez okozhat egy kis eltérést. A membránfeszítettség-moduláció hatását legjobban a spektrogram segítségével érhetjük tetten.

A 8.7. ábrán jól megfigyelhető, hogy a kezdeti impulzus hatására minden frekvenciakomponens megugrik, majd visszatérnek a kiindulási frekvenciára.



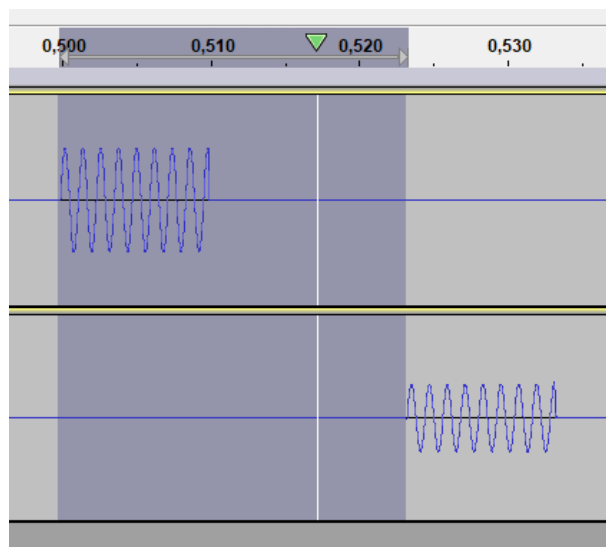
8.7. ábra. Lábdob válaszának spektrogramja

8.3. A dobszintetizátor késleltetésének vizsgálata

Az analóg audió rendszerek természetükből fakadóan nem okoznak késleltetést a feldolgozási láncban, mivel nincs olyan analóg elektronikai alkatrész, mely tárolóként viselkedne. Ha digitális audió rendszerrel szeretnénk megvalósítani jelfeldolgozást, akkor nagyon fontos paraméter a rendszer válaszideje. Egy nagy késleltetéssel rendelkező dobszintetizátor nem használható élőzenei körülmények között.

A DSP-ben történő feldolgozás mintáról mintára történik, nem használok be és kimeneti puffert, tehát az elvárásunk a feldolgozás késleltetésével szemben az a rendszer ismerete alapján, hogy bőven az emberi érzékelési küszöb alatt lesz.

A késleltetésre a fejlesztői számítógép segítségével végeztem egy mérést az Audacity nyílt forráskódú szoftver segítségével. Első lépésben arra voltam kíváncsi, hogy mekkora a számítógépre kötött audióinterfész és az operációs rendszer okozta időkésés. A hangkártyán egy analóg kimenetet egy analóg bemenetre kötöttem. Az Audacity-vel a kimenetre kijátszottam egy szinuszelet, majd az analóg bemeneten keresztül rögzítettem is ugyanezt.

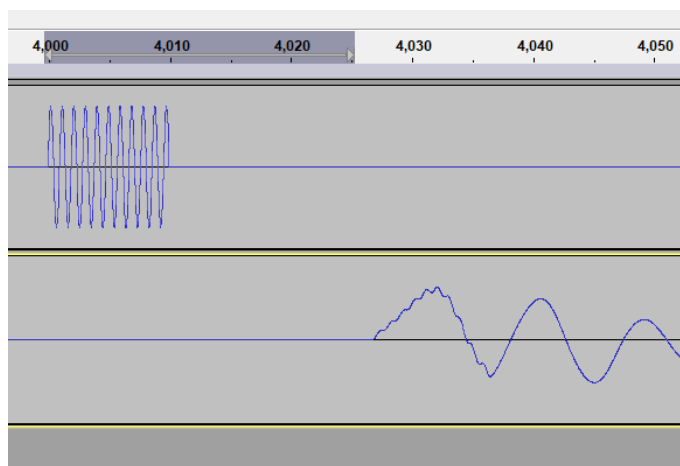


8.8. ábra. Mérési eljárás kalibrációja

A 8.8. ábrán látható mérés eredménye az, hogy a felvett és a kiadott jel között 24 ms az eltérés. A mérés többszöri megismétlésével szerzett tapasztalatom az, hogy a valós késleltetés 24 ms +/- 1 ms.

A mérést megismételtem úgy, hogy a kimenet és a bemenet közé bekötöttem a DSP-t is.

A 8.9. ábra szerint a DSP hurokba kötésével a válaszidő 26 ms-ra nőtt. A mérés bizonytalanságát beleszámítva kijelenthető, hogy a DSP által okozott késleltetés 1 ms és 3 ms között van. Eddigi tapasztalataim alapján az emberi érzékelési küszöb kb. 10 ms hasonló esetekben, tehát a mérés által adott becslés beigazolja az elvárásaimat, miszerint a késleltetés nem érzékelhető. Pontosabb mérést pl.: kétcsatornás oszcilloszkóppal végezhetnénk.



8.9. ábra. A DSP által szintetizált jel pozíciója a gerjesztőjelhez képest.

9. fejezet

Összefoglalás

A dolgozat készítése során egy fizikai alapú dobszintetizátort valósítottam meg modellező és beágyazott környezetben. A munkám során megvizsgáltam, a jelenleg piacon lévő dobszintetizátorok alapvető képességeit és működését, foglalkoztam olyan szenzorok működésével amik segítségével képesek lehetünk egy cajon-on történő hangszeres játék paramétereinek szoftveres modellbe történő beküldésére. Ilyen szenzor volt a piezo illetve az erőérzékeny ellenállás. Laboratóriumi mérés során vizsgáltam az erőérzékeny ellenállás statikus karakterisztikáját, valamint tanulmányoztam, milyen jelek nyerhetők a piezo jeledők segítségével a cajon különböző pontjaira rögzítve. Kipróbáltam a modális hangszintézis eljárást, az ehhez szükséges fizikai modelleket Matlab környezetben hoztam létre. Lábdob esetén implementáltam a feszítettség-moduláció technikát. Ez a módszer lehetővé teszi, hogy modellezem a nagyobb gerjesztőerő hatására bekövetkező nemlineáris jelenségeket. A pergő esetén modelleztem, hogy a két membrán csatolásban van, illetve zaj hozzákeverésével a sodrony által létrehozott hangzást is beépítettem az algoritmusba.

A modellező környezetben gyűjtött tapasztalatok alapján egy olyan beágyazott platformot választottam, amelyen meg lehet valósítani valós időben a hangszintézist. A választott jelprocesszor az Analog Devices SHARC ADSP-21489 lett, ami megfelelő számítási kapacitással rendelkezik a modellek futtatásához. Az implementáció alkalmas arra, hogy két audió bemeneti csatorna (a cajon-ra szerelt piezo szenzorok jele) alapján valós időben futtassa a lábdob és a pergődob fizikai modelljét. Egy kiegészítő mikrokontrolleres fejlesztőkészlet segítségével grafikus módon valós időben módosíthatók a hangszintézis paramétere.

9.1. Értékelés

A megvalósított rendszer alkalmas a modális hangszintézis alapú fizikai modellek valós időben történő futtatására, valamint a beágyazott szoftver frissítése nélkül módosíthatók a szintézis belső paramétere. A kimeneten mérhető jel megfelel azoknak az elvárásoknak, amelyeket a modellező környezetben kipróbált szintézis alapján támasztottam. A beágyazott szoftver stabilan fut, a tesztek során nem volt tapasztalható instabilitás.

Véleményem szerint a munka során bebizonyosodott, hogy a modális szintézisre épített ütőhangszer szintetizátor hangjában valóban nagyon sok részlet megmarad az eredeti

hangszeres játékból, de megváltoztatja a hangzás karakterisztikáját. Ezt szemléltetik a mellékletben található felvételek is. A tapasztalataim alapján azt gondolom, hogy további fejlesztőmunkát befektetve akár piacképes termék alapja is lehet.

9.2. Továbbfejlesztési lehetőségek

A szintézis modell struktúrákat lehetne még továbbfejleszteni több irányból. Egyik lehetőség, hogy növeljük a dobmodell részletességét és a valódi dobban lejátszódó fizikai folyamatok nagyobb részhalmozát valósítjuk meg, vagy a modális szintézis alapelvétől nem eltérve, de kibővítve, további új fajta hangzások konstruálhatóak. A belső paramétereket modulálhatjuk új vezérlőjelekkel, pl.: LFO-kkal (Low Frequency Oscillator - alacsonyfrekvenciájú oszcillátor).

Javítható lenne a feszítettség-moduláció megvalósítása, mert a jelenlegi visszacsatolt struktúrában lehet olyan állapotot előidézni, amikor a modell instabillá válik. Ha sikerülne úgy közelíteni ugyanezt a hatást, hogy nem használok hozzá visszacsatolást, akkor kiküszöbölhető lenne az instabilitás oka.

Új akusztikus ütőhangszerekre alapozva is lehetne létrehozni szintézis modelleket, tovább bővítve ezzel a szintetizálható hangzásokat.

Gazdagabbá lehetne tenni a hangzást, ha a modellek kimenetét tovább lehetne effektezni, törekedve arra, hogy kihasználjuk a sztereó tér lehetőségeit (pl.: EQ, visszhang, enyhe szaturáció, stb.).

Ábrák jegyzéke

1.1. Roland TR-909	9
1.2. Cajon [9]	10
1.3. Rendszerterv áttekintés	10
1.4. Korg Wavedrum [21]	12
1.5. Roland El Cajon EC-10 [20]	12
1.6. Native Instruments Reaktor Prism FX [18]	13
1.7. Native Instruments Reaktor Prism FX blokk diagram [18]	13
1.8. Bögrére rögzített Mogeeg szenzor [17]	15
2.1. Additív szintézis [2]	18
2.2. Roland TR-909 Pergődob [22]	20
2.3. Roland TR-909 Lábdob [22]	20
2.4. Digitális hullámvezető egy lehetséges implementációja [39].	22
2.5. FM szintézis egy lehetséges megvalósítása [12]	24
3.1. Három szabadsági fokú koncentrált paraméteres rezgő rendszer [30]	25
3.2. A húr Δx hosszú szelete [32]	28
3.3. A membrán $\Delta x, \Delta y$ méretű darabja [32]	29
3.4. A kör alakú membrán módusai [32]	31
3.5. Párhuzamosan kapcsolt rezonátorok	33
3.6. Rezonátor kimenete	33
4.1. Piezo jeladó [10]	34
4.2. FSR rétegei [28]	35
4.3. FSR robbantott ábra [28]	35
4.4. Az FSR kimenetét feldolgozó feszültségosztó kapcsolás [28]	36
4.5. FSR erő-ellenállás karakterisztikája	36
4.6. FSR linearizált erő-ellenállás karakterisztikája	37
4.7. Felszenzorozott cajon - a narancssárga jelzés jelenti a belülről felszerelt szenzort	39
5.1. Alkalmazott zajzár struktúrája	40
5.2. Símitást végző modul belső állapotgépe	42
5.3. A zajzár hatása a jelre mély ütés esetén	42
5.4. A zajzár hatása a jelre mély ütés esetén	43
5.5. Pergődob hangjának szintetizálása	45

5.6.	Lábdob hang szintetizálása	46
6.1.	Bela IO fejlesztőkártya	48
6.2.	Analog Devices ADSP-21489 fejlesztőkártya	49
6.3.	Analog Devices ADSP-21489 funkcionális blokkvázlata [3]	50
6.4.	Analog Devices ADSP-21489 műveletvégző egység blokkvázlata [3]	50
6.5.	Az STM32F746G Discovery fejlesztői kártya [11]	51
6.6.	Az STM32F746G Discovery fejlesztői kártya [11]	52
6.7.	A DSP-n futó beágyazott kód magasszintű folyamata	53
6.8.	main_process.c folyamatábra	56
7.1.	Dobszintetizátor kezelőfelülete	59
7.2.	Kezelőfelület eseményvezérelt működése	60
8.1.	Rezonátorok tesztelése	64
8.2.	Pergődob jelalak: balra a DSP, jobbra a Matlab kimenete	64
8.3.	Pergődob spektruma: balra a DSP, jobbra a Matlab	65
8.4.	Pergődob válaszának spektrogrammja	65
8.5.	Lábdob jelalak: balra a DSP, jobbra a Matlab kimenete	66
8.6.	Lábdob spektruma: balra a DSP, jobbra a Matlab	66
8.7.	Lábdob válaszának spektrogrammja	67
8.8.	Mérési eljárás kalibrációja	68
8.9.	A DSP által szintetizált jel pozíciója a gerjesztőjelhez képest.	69

Irodalomjegyzék

- [1] 1959 Wurlitzer Side Man. <https://emhistory.wikispaces.com/1959+Wurlitzer+Side+Man>. Elérés dátuma: 2017. 11. 17.
- [2] Additive synthesis. <http://www.cchrispeters.com/SynthesisTerms/AdditiveSynthesis.htm>. Elérés dátuma: 2017. 12. 02.
- [3] Analog Devices SHARC ADSP-21489 datasheet. http://www.analog.com/media/en/technical-documentation/data-sheets/ADSP-21483_21486_21487_21488_21489.pdf. Elérés dátuma: 2017. 12. 05.
- [4] Analog synthesizer. https://en.wikipedia.org/wiki/Analog_synthesizer. Elérés dátuma: 2017. 11. 17.
- [5] Beagle bone official website. <http://beagleboard.org/bone>. Elérés dátuma: 2017. 12. 05.
- [6] BeepStreet - Impaktor. <http://www.beepstreet.com/ios/impaktor>. Elérés dátuma: 2017. 12. 02.
- [7] Bela io embedded platform. <http://bela.io/>. Elérés dátuma: 2017. 12. 05.
- [8] Cajón. <https://en.wikipedia.org/wiki/Caj%C3%B3n>. Elérés dátuma: 2017. 05. 17.
- [9] Cajon agile. <http://www.schlagwerk.com/en/products/cajon/cajon-agile/super-agile-cappuccino/>. Elérés dátuma: 2017. 05. 17.
- [10] Detect vibrations using a piezo element. <http://runtimeprojects.com/2017/02/detect-vibrations-using-a-piezo-element/>. Elérés dátuma: 2017. 05. 17.
- [11] Discovery kit with STM32F746NG MCU. <http://www.st.com/en/evaluation-tools/32f746gdiscovery.html>. Elérés dátuma: 2017. 12. 05.
- [12] Frequency modulation synthesis. https://en.wikipedia.org/wiki/Frequency_modulation_synthesis.
- [13] Hammond organ. https://en.wikipedia.org/wiki/Hammond_organ. Elérés dátuma: 2017. 11. 17.

- [14] Mbed official website. <https://www.mbed.com/en/>. Elérés dátuma: 2017. 12. 05.
- [15] Modular synthesizer. https://en.wikipedia.org/wiki/Modular_synthesizer. Elérés dátuma: 2017. 11. 17.
- [16] Mogeegs official website. <https://www.mogeegs.co.uk/>. Elérés dátuma: 2017. 12. 02.
- [17] Mogeegs picks up seed funding to put audio-based gesture recognition tech into new devices. <https://techcrunch.com/2017/05/25/mogeegs-seed/>. Elérés dátuma: 2017. 12. 02.
- [18] Native instruments. <https://www.native-instruments.com/en/products/komplete/synths/reaktor-prism/>. Elérés dátuma: 2017. 11. 17.
- [19] Raspberry pi official website. <https://www.raspberrypi.org/>. Elérés dátuma: 2017. 12. 05.
- [20] Roland El Cajon EC-10. <https://www.soundonsound.com/reviews/roland-el-cajon-ec-10>. Elérés dátuma: 2017. 11. 17.
- [21] Sound on sound - korg wavedrum. <https://www.soundonsound.com/reviews/korg-wavedrum>. Elérés dátuma: 2017. 12. 02.
- [22] Sound on sound - synth secrets. <https://www.soundonsound.com/techniques/synth-secrets>. Elérés dátuma: 2017. 11. 20.
- [23] Trautonium. <https://en.wikipedia.org/wiki/Trautonium>. Elérés dátuma: 2017. 11. 17.
- [24] Riccardo Marogna, Federico Avanzini, Balázs Bank. Energy based synthesis of tension modulation in membranes. September 6-10 2010.
- [25] Stefan Bilbao. *Numerical Experiments with Coupled Membranes and the Snare Mechanism*. Lyon, 2010. Acoustics and Fluid Dynamics Group/Music, University of Edinburgh.
- [26] Dr. Huba Antal, Dr. Aradi Petra, Czmerk András, Dr. Lakatos Béla, Dr. Chován Tibor, Dr. Varga Tamás. Szenzorok és aktuátorok dinamikai modelljei és tervezésük. http://www.mogi.bme.hu/TAMOP/mechatronikai_berendezesek_tervezese/math-ch06.html, 2014. Elérés dátuma: 2017. 05. 17.
- [27] Balogh László, Dr. Kollár István, Dr. Péceli Gábor, dr. Németh József, Dr. Sujbert László, Vargha Balázs, Bank Balázs. Digitális jelfeldolgozás - tantárgyi jegyzet. <https://www.mit.bme.hu/system/files/oktatas/targyak/vedett/10094/vimm4084-digit-jelfeldolg.pdf>. Elérés dátuma: 2017. 12. 03.

- [28] Interlink Electronics. Fsr 400 series data sheet. https://www.interlinkelectronics.com/datasheets/Datasheet_FSR.pdf. Elérés dátuma: 2017. 05. 17.
- [29] Davide Rocchesso Federico Avanzini. *Modeling Collision Sounds: Non-Linear Contact Force*. Limerick, Ireland, 2001. Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-01).
- [30] Fiala Péter. Móduselemzés. <http://last.hit.bme.hu/sites/default/files/documents/modus.pdf>. Egyetemi jegyzet, Stúdiótechnika Laboratórium.
- [31] Thomas D. Rossing Neville H. Fletcher. *The Physics of Musical Instruments*. 1991. Springer-Verlag.
- [32] Fiala Péter. *Hangszerek fizikája tantárgyi jegyzet*. Budapesti Műszaki és Gazdaságtudományi Egyetem, 2015. Hálózati rendszerek és szolgáltatások tanszék.
- [33] J. O. Smith. Physical modeling using digital waveguides. *Computer Music Journal*, 16(4):74–91, 1992. Department of Music, Stanford University, Stanford, CA 94305.
- [34] Julius Orion Smith. Taxonomy of digital synthesis techniques. https://ccrma.stanford.edu/~jos/kna/Taxonomy_Digital_Synthesis_Techniques.html. Elérés dátuma: 2017. 05. 17.
- [35] Tero Tolonen, Vesa Välimäki, and Matti Karjalainen. *Evaluation of Modern Sound Synthesis Methods*. Espoo, Helsinki University of Technology, 1998. Department of Electrical and Communications Engineering.
- [36] Wilhelm S. Wermelin. Modelling of drums. January 21, 2015. Karlstad University, Analytical Mechanics.
- [37] Balázs Bank, Stefano Zambon, and Federico Fontana. A modal-based real-time piano synthesizer. *IEEE Trans. On Audio, Speech, and Language Processing*, 18(4):809–821, May 2010.
- [38] Stefano Zambon, and Federico Fontana. A modular physically based approach to the sound synthesis of membrane percussion instruments. *IEEE Trans. On Audio, Speech, and Language Processing*, 18(4), May 2010.
- [39] Szalontai Péter Zoltán. Fizikai alapú elektronikus ütőhangszer megvalósítása - diplomaterv, 2012.
- [40] Garamvölgyi Zsolt. Dobhang fizikai alapú szintézise - TDK dolgozat. http://home.mit.bme.hu/~garam/download/gzs_tdk.pdf, 2006.