



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Hálózati Rendszerek és Szolgáltatások Tanszék

Horváth Bence

MARIMBA HANGSZINTÉZISE VST KÖRNYEZETBEN

KONZULENS

Dr. Rucz Péter

BUDAPEST, 2022

Tartalomjegyzék

Összefoglaló	5
Abstract.....	6
1 Bevezetés	7
1.1 Motiváció és célkitűzés.....	7
1.2 A Dolgozat felépítése.....	7
2 Marimba	8
2.1 A marimba eredete és története	8
2.2 A marimba hangképző részei.....	9
2.2.1 Hanglap.....	9
2.2.2 Rezonátor	10
2.2.3 Ütők	11
3 A modell felépítése	13
3.1 A hanglap modellezése	13
3.1.1 Analitikus módszer	13
3.1.2 Végeselem módszer	16
3.2 Az ütő-hanglap kölcsönhatás	18
3.3 Rezonátor hatása a rendszerre.....	21
3.4 A teljes modell elkészítése.....	24
3.5 Példák a modell tesztelésére	27
4 A szintetizátor VST implementációja	30
4.1 Virtual Studio Technology.....	30
4.2 Digital Audio Workstation.....	31
4.3 Steinberg VST SDK.....	32
4.3.1 Note Expression Synth.....	33
4.4 További felhasznált szoftverek	34
4.4.1 Eigen	34
4.4.2 FFTW	35
4.5 A szintetizátor megvalósítása	35
4.5.1 Controller	36
4.5.2 UI	36
4.5.3 Processor.....	37

4.5.4 Voice	38
4.5.5 Grafikus felület elkészítése	42
5 Konfigurálható paraméterek	44
5.1 A felhasználói felületen beállítható értékek.....	44
5.2 Konfigurációs fájl tartalma	45
6 Összegzés.....	47
6.1 Értékelés.....	47
6.2 Továbbfejlesztési lehetőségek	47
6.3 Köszönetnyilvánítás	48
Irodalomjegyzék.....	49
Függelék.....	51

HALLGATÓI NYILATKOZAT

Alulírott **Horváth Bence**, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot 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 hitelesített 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.

Kelt: Budapest, 2022. 12. 08.

.....
Horváth Bence

Összefoglaló

Az utóbbi pár évtizedben a számítógépek fejlődésével és a feldolgozási teljesítmény növekedésével megnyílt a lehetőség különböző hangszerek működésének valós idejű szimulációjára. Ennek előnye, hogy nagyobb rétegek számára elérhetővé és kényelmesebbé teszi a hangszerek használatát, illetve szemlélteti a felhasznált fizikai modellek hatékonyságát, pontosságát.

A dolgozatomban egy ütős hangszer, a marimba hangkeltésének fizikai alapú modellezését, majd a kész modell VST környezetben történő szoftveres implementációját mutatom be. A marimba hangját elsősorban a hangszer három fő komponense határozza meg. Az ütő adja a gerjesztést a rendszerben, a gerjesztő erőt befolyásolja az ütő keménysége, tömege, és a leütés sebessége. A leütés hatására a hanglapban rezgések keletkeznek, amelyet a hanglap alakja és az anyagára vonatkozó fizikai tényezők határoznak meg. Végül a hanglap által lesugárzott hangot a rezonátor formálja, itt figyelembe kell venni a rezonátor geometriáját, elhelyezkedését és egyéb akusztikai jellemzőit.

A modell létrehozásánál több, fizikai közelítési módszert használtam fel. A hanglap esetén a hajlító rezgést, és a módusalakokat vettem figyelembe, ezeknek kiszámítására a végeelem módszert használtam. A kalapács-hanglap, és a hanglap-rezonátor kölcsönhatások modellezését pedig a Newmark-módszer segítségével oldottam meg. Néhány egyszerűsítést végeztem a komplexitás és számolásigény csökkentése érdekében, ugyanis a hanglap és a rezonátor is egydimenziós rendszerként modelleztem, a kölcsönhatásuk pedig koncentrált paraméterek szintjén történik. Ezen módszerek megfelelő alkalmazásával és összekapcsolásával elkészíthető a végleges egyszerűsített modell.

Ismertetem a dolgozatban a VST Plugin megvalósítását, a felhasznált VST fejlesztőkörnyezetet, valamint az egyéb könyvtárakat is bemutatom. Az implementáció során szem előtt tartottam a hatékonyságot, hogy valós időben lehessen előállítani a hangot, valamint a felhasználói felület tervezésére is ügyeltem, több állítható paraméterrel lehet változtatni a fizikai jellemzőket a különböző hangzások eléréséhez.

Abstract

With the evolution of computers and the growth of processing power in the last few decades, real time simulation of certain musical instruments became possible. This has the benefit of making instruments more accessible, comfortable to play as well as illustrating the efficiency and accuracy of physical models.

In my thesis, I present the physical model for the sound of a percussion instrument, the marimba. Afterwards I will show the software implementation of this model in VST environment. The sound of the marimba is shaped by three main components. The mallet generates the excitation in the system. This excitation force is influenced by the stiffness, mass of the mallet and the velocity of the strike. After the impact, vibrations start to occur in the marimba bar, which are affected by the structure of the bar and the physical characteristics of its material. Finally, the sound radiated by the bar is shaped by the resonator, considering its geometry, position, and other acoustical properties.

In the model creation process I used multiple approximation methods. Regarding the marimba bar I have only considered the transversal vibration and its modal shapes, which I calculated using the finite element method. For the modelling of the mallet-bar and bar-resonator interaction I used the Newmark method. I made a couple of simplifications for the purpose of reducing complexity and computing intensity, namely the bar and the resonator is viewed as a one-dimensional system, and their interaction is characterized by concentrated parameters. With the help of these methods the final simplified model can be completed.

The implementation of the VST plugin will also be presented in the thesis, along with the VST Software Development Kit and additional libraries I used for making the software. During the implementation effectivity, real time sound synthesis and the graphical interface were my top priority. I also made sure that physical parameters can be changed to allow for versatility in sound creation.

1 Bevezetés

1.1 Motiváció és célkitűzés

Gyermekkorom óta érdekelt a zene, hobbiszinten foglalkoztam hangszeres zenéléssel (gitár, basszusgitár), majd az egyes hangszereket elektronikusan megvalósító szintetizátorok felé fordultam. Ezért nagyon örültem, hogy lehetőséget kaptam egy ehhez szorosan kapcsolódó témában mélyebben elmerülni. Nagyon inspiráló volt, hogy számomra két fontos terület (zene és informatika) összekapcsolásával dolgozhattam ezen a projekten.

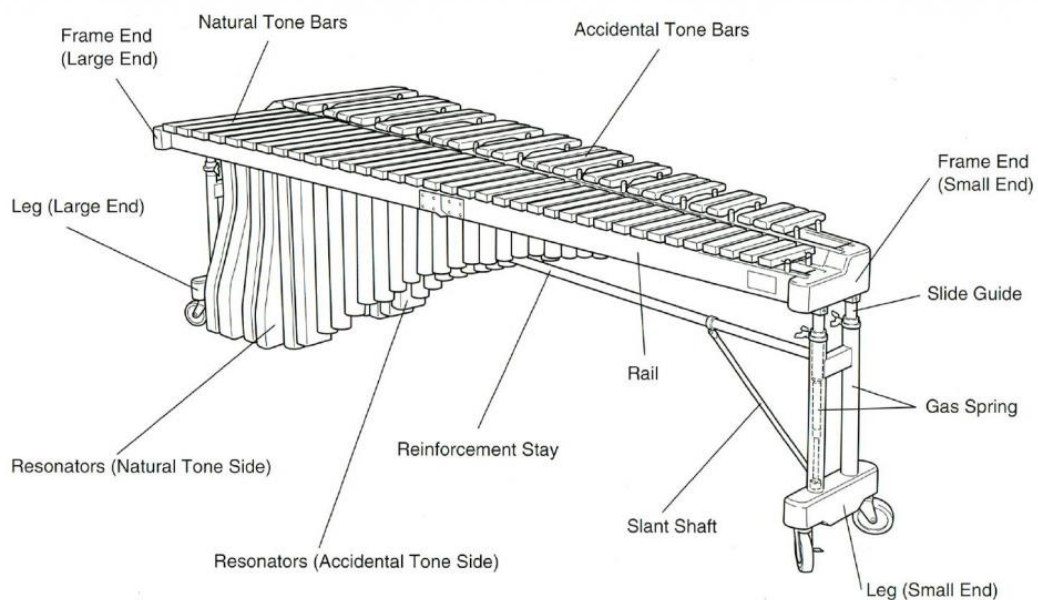
Azért a marimbára esett választásom a feladat megvalósításához, mert viszonylag kevésbé ismert, érdekes hangzása van és ehhez hasonló szintetizátormegvalósítást nem találtam erre a hangszerre. A tudomány a 20. század óta foglalkozott a hasonló hangszerek (hangolt idiofonok) hangkeltésének modellezésével. Ebben a témában számos tudományos cikk született. Ezen megoldások felhasználásával és összekapcsolásával készítettem el a teljes modellt. Úgy gondolom, hogy ezzel egy egyedi megoldást sikerült létrehozni ebben a projektben.

1.2 A Dolgozat felépítése

Először bemutatom a marimba hangszert és fontosabb jellemzőit a hangkeltés szempontjából. Később részletesen bemutatom a modellalkotás folyamatát, az egyes megfontolásokkal, egyszerűsítésekkel és ezek megfelelő indoklásával együtt. Ismertetem a kész modell szoftveres implementációját, amely egy VST szintetizátor plugin részeként valósul meg. A plugin elkészítéséhez használt eszközöket, könyvtárakat és ezek szükségességét is elmagyarázom. Valamint bemutatom a kezelői felületet és a változtatható paramétereket, amelyekkel a modell fizikai jellemzőit (pl. hangszer anyaga és alakja, az ütés pozíciója) lehet befolyásolni. Végül értékelem a megoldott feladatot, illetve felsorolok néhány továbbfejlesztési lehetőséget.

2 Marimba

A marimba egy ütős hangszer, a hangolt idiofonok családjába tartozik, a sokkal ismertebb és elterjedtebb xilofonhoz nagy mértékben hasonlít. Viszont ettől eltérően a marimba fából készült rudakból, vagy hanglapokból áll, amelyek megszólaltatását ütők segítségével végzik. A hanglapok alatt rezonátorcsövek helyezkednek el, ezek a hanglapból érkező hang felerősítésére szolgálnak. A hanglapok és a rezonátorcsövek egy kereten helyezkednek el, amit egy állvány nagyjából 1 méter magasan tart a földfelszíntől, ahogy a 2.1. ábra mutatja.



2.1. ábra: A marimba kinézete és részei [1]

2.1 A marimba eredete és története

A marimba a xilofonból fejlődött ki, amely Afrikából származik. A xilofonnak több fajtája alakult ki, főleg Közép- és Dél-Afrikában terjedt el, feltehetőleg a mostani Mali országból származik a 13. századból.

A xilofon egyik fajtájáról, ahol lopótkökből készítették a rezonátort, mintázták a később Latin-Amerikában elterjedt marimbát. Ez a típus elsősorban Tanzániában és Kongóban volt elterjedt.

A 16-17. században Közép- és Dél-Amerikában is megjelentek a marimbaként ismert xilofonok. Itt indult el a marimba fejlődése és elterjedése, főleg Brazíliában, Guatemalában és Mexikóban. A lopótökből készült rezonátort lecserélték pontosabban behangolt, fából készült, üreges kamrákra.

1910-ben jött létre a jelenlegi marimba hangszer, amit koncerteken és zenekarokban is használnak. Elkezdtek gyártani a latin-amerikai marimba mintájára készült hangszereket európai és amerikai szimfóniákhoz a zenekarok számára. A korábbi fából készült rezonátorokat lecserélték fémcsőkre, amelyek már hangolhatók voltak. Így jött létre a marimba, ami a 20. század második felében terjedt el igazán. [2]

2.2 A marimba hangképző részei

A következőkben bemutatom a marimba azon részeit, amelyek a hang kialakulásában is fontos szerepet játszanak, amit a modell alkotásnál is figyelembe vettem.

2.2.1 Hanglap



2.2. ábra: Hanglapok [3]

A hangképzés szempontjából a legfontosabb elemek a hanglapok, hiszen a hanglap rezgésének komponensei határozzák meg a lesugárzott hang frekvenciaösszetételét is. A marimba hanglapjai rúd alakúak, többféle méretben is készülnek attól függően, hogy milyen hangolást szeretnének elérni. Általában valamilyen faanyagból készülnek, például rózsafából.

A 2.2. ábrán is jól látszik, hogy a rudak közepén különböző méretű bevágás található, amitől vékonyabb lesz a rúd bizonyos szakaszain. Ez a bevágás is a megfelelő hangolást szolgálja, ugyanis ez nem csak a rúd méretétől függ. A rúd méretének változtatásával lehet befolyásolni az alaphang frekvenciáját, a többi módus egymáshoz vonatkoztatott aránya nem változik, ami egy szabályos rúd esetén nem eredményez megfelelő zenei hangzást. Ezt a problémát orvosolják a bevágással, ennek mértéke változhat különböző hanglapok között. A marimba esetében ezzel a hangolójáttal az 1:4:10 frekvenciaarányok beállítására törekszenek, amelyekkel tiszta dúrakkordokat lehet megszólaltatni. A gyakorlatban használt egy másik fajta hangolás is hasonló hangszereknél, ami az 1:3:9 arányok kialakításával történik.

A rudak két szélén vízszintes irányban lyukak is találhatóak, ez is látszik a 2.2. ábrán. Ez a rudak rögzítésére szolgál a hangszer keretére, hogy ne mozduljanak el oldalirányba a használat során. A rögzítést általában valamilyen rugalmas anyaggal (például a rúdon átfűzött kötelekkel, egyes hangszereknél filc alátámasztással) végzik, mert ha nagyon szorosan rögzítenék, az akadályozná a rúd keresztirányú rezgéseit is, tehát tompább, halkabb lenne a hangja.

2.2.2 Rezonátor



2.3. ábra: Rezonátorcsövek [4]

A marimba hangját a hanglap szolgáltatja elsősorban, ugyanakkor a hanglapokból lesugárzott hang nem túl erős, pár méternél messzebről már nem igazán hallható. A jobb hangzás segítése érdekében hozták létre a rezonátort, amely gyakorlatilag erősítőként szolgál, így a hangszer hangja messzebbre is eljut, könnyebben használható nagyobb közönség előtt. Fizikai szempontból ennél kicsit többről van szó, ennek részleteit a modell ismertetésénél részletezem tovább.

A rezonátor manapság valamilyen fémből készült cső, aminek a teteje nyitott, az alja zárt. A rezonátorcsövek a hanglapok alatt helyezkednek el, ahogyan a 2.3. ábrán is látható. Általában minden hanglap alatt található egy hozzátartozó rezonátorcső, amely a hanglaphoz van hangolva, így az egyes csövek főleg csak a hozzájuk tartozó hanglap hangját erősítik.

A csövek hossza változó, főleg a hanglap frekvenciájától függ. A mélyebb hangokhoz hosszabb csövek tartoznak. Ez a hanglapok esetén is igaz, ott főleg a rudak hossza változik, a mélyebb frekvenciákhoz hosszabb rudak tartoznak. A rezonátor alján lévő zárt rész bizonyos keretek között állítható, ezzel lehet a rezonátort hangolni. Legtöbbször a hanglapok frekvenciájához nagyon közel hangolják, de nem teljesen megegyező frekvenciára. Ettől a hangzás kicsit gazdagabb, karakteresebb lesz.

2.2.3 Ütők



2.4. ábra: Ütők [5]

A marimba egy ütős hangszer, amiből az következik, hogy ütővel történik a hangszer megszólaltatása. Az ütő nem közvetlenül vesz részt a hang kialakulásában, de ugyanúgy fontos szerepet játszik, mert a segítségével lehet erőt kifejteni a hanglapokon, amely hatására azok rezegni kezdenek.

Az ütők nem olyan bonyolult felépítésűek, fizikai szempontból is csak pár paraméter befolyásolja a keltett rezgést a hanglapban, azonban van néhány tényező, amit érdemes figyelembe venni. A 2.4-es ábra mutat pár példát különböző típusú ütőkre. Az ütők általában egy 28-40 cm hosszú nyélből, és egy gömb alakú, 2.5-3.5 cm átmérőjű fejből állnak. Legfontosabb tulajdonságuk az ütő fejének anyaga, valamint keménysége.

Az anyagát legtöbbször valamilyen puhább fa, vagy gumi képezi, külső felületét fonállal szokták betekerni.

A fej rugalmassága nagyban befolyásolja, hogy az ütő milyen mértékben tudja megszólaltatni a hangszert, ez a modell szempontjából is fontos lesz. A megszólaltatás mértéke egyrészt arra vonatkozik, hogy mennyi energiát tud átadni az ütő feje a hanglaphoz, másrészt ennek milyen a frekvenciaeloszlása, spektruma. Egy másik fontos szempont a leütés sebessége, amely nem az ütő konkrét tulajdonsága, hanem a hangszer használója tudja befolyásolni, hogy milyen erősséggel szeretné leütni, ezáltal milyen hanghatást szeretne elérni a hangszerrel. Az ütő sebessége a hanglappal való érintkezéskor általában 0.05-1.5 m/s közé szokott esni [6].

3 A modell felépítése

A marimba hangszintézise során fizikai alapú módszereket használtam a modell megalkotására, amely értelmében a hangszer fizikai paramétereit, valamint a hangját befolyásoló fizikai jelenségeket megpróbálom a lehető legpontosabban figyelembe venni, és ez alapján a hang kialakulását és terjedését szimulálni. Bizonyos mértékben egyszerűsítésekre volt szükség a valósághoz képest, hiszen a hangszer megszólaltatása során hallható hangot sok tényező befolyásolja, például a terem, vagy szoba akusztikája, ahol a hangszer elhelyezkedik. Az általam alkotott modell a marimba fő hangképző, illetve a képzett hangot formáló elemeit, és ezen elemek kölcsönhatását veszi figyelembe, mivel ez jó közelítést ad a hangszer valódi hangjához.

3.1 A hanglap modellezése

Fizikai szempontból a hanglapot egy rendszerként is lehet értelmezni, ahol egy bevitt erő hatására valamilyen válasz keletkezik. A rendszert gerjesztő erő az ütővel való megütés során jön létre, ez az erő azonban csak nagyon rövid ideig hat a hanglapra. Ennek hatására a rúd szabadrezgést végez, minden módus a sajátfrekvenciáján oszcillál.

Azt is fontos megjegyezni, hogy a rúdban többféle rezgés is keletkezhet, például longitudinális, hajlító irányú, vagy torziós rezgések is kialakulhatnak. Ezek mértéke függ a leütés helyétől is, a rúd tetején történő leütés hatására nem keletkeznek jelentős hosszanti irányú rezgések. A modellnél a hajlító rezgéseket vettem figyelembe, mivel ezek a legmeghatározóbbak a hangképzés során.

3.1.1 Analitikus módszer

Kezdetben megpróbáltam analitikus módszerrel létrehozni a hanglap modelljét. Egy alkalmas megoldás volt erre az Euler-Bernoulli rúdmodell, azonban ez csak néhány egyszerűsítéssel volt használható. Olyan feltételezések mellett érvényes ez a modell, ahol a vizsgált rúd szabályos alakú és vékony, tehát lényegesen kisebb a vastagsága a hosszánál. Ez abban az esetben működőképes, ha a marimba hanglapját is egy szabályos vékony rúdként értelmezem, ezért kezdetben így is tettem, a későbbiekben részletezni fogom, hogy miért nem lesz megfelelő ez a modell a kívánt célokhoz. A jelenlegi feltételekkel a rúd hajlító rezgéseit meghatározó módusalakokat és sajátfrekvenciákat ki lehet számolni analitikusan, ezek a következő képletekkel megkaphatók [7]:

Módusalakok kiszámítása egy adott pontban:

$$\psi_n(x) = A [\sin(k_n x) + \text{sh}(k_n x)] + B [\cos(k_n x) + \text{ch}(k_n x)]$$

$$B = A \frac{-\sin(k_n L) + \text{sh}(k_n L)}{\cos(k_n L) - \text{ch}(k_n L)}$$

ahol

$$k_n = \frac{\pi}{2L} [3, 0112, 5, 7, 9, \dots]$$

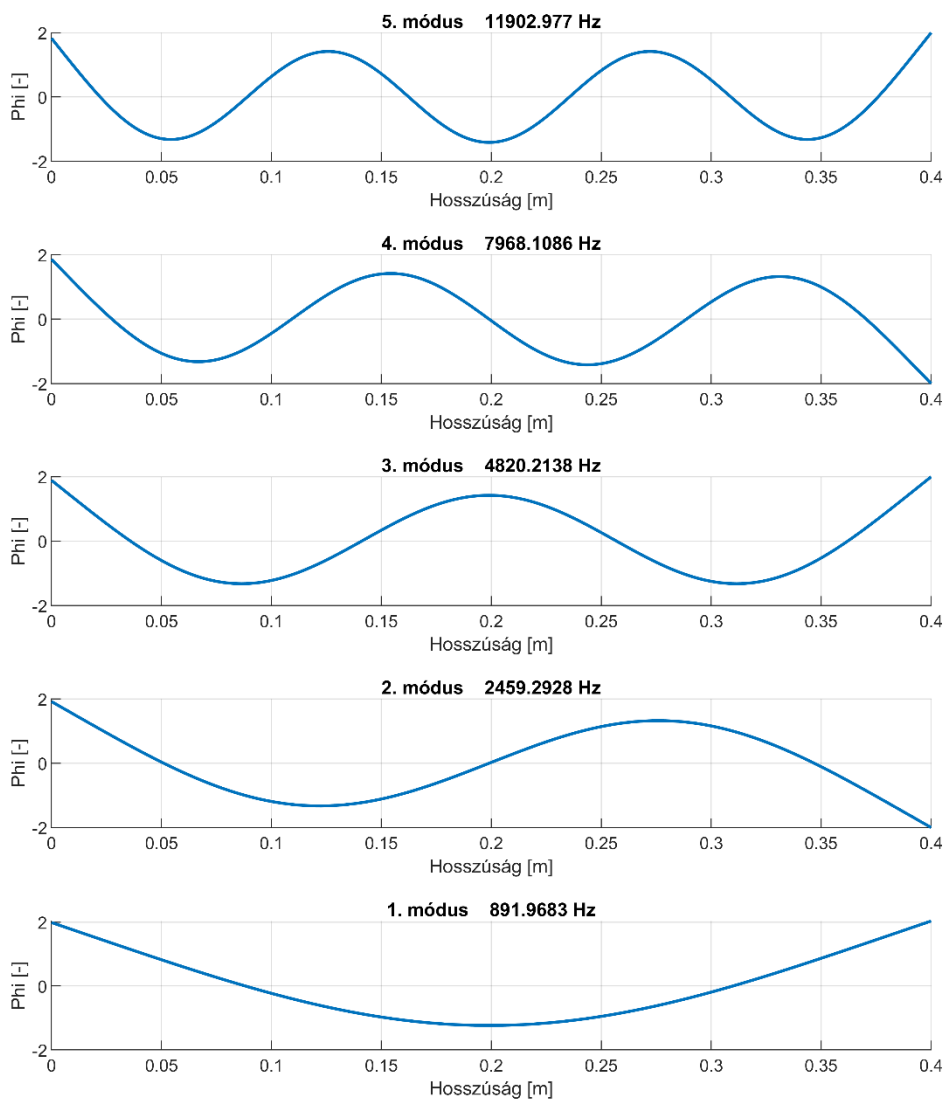
Sajátfrekvenciák kiszámítása:

$$\omega_n = c_L K \frac{\pi^2}{4L^2} [3, 0112^2, 5^2, 7^2, 9^2, \dots]$$

$$c_L = \sqrt{\frac{E}{\rho}}$$

$$K = \frac{h}{\sqrt{12}}$$

A ψ_n jelöli az n -edik módusalak értékét egy x pontban, ami 0-tól a rúd hosszáig (L) bármelyik pont lehet. Az A és B paraméterek értékei tetszőlegesen választhatók, itt B ki van fejezve A segítségével, A értékét pedig lehet például 1-nek választani, ez csak a végeredményként kapott függvény amplitúdóját befolyásolja. Az ω_n sajátfrekvenciák számítása során c_L , ami a rúdban létrejött longitudinális rezgések terjedési sebességét (ez valójában a hangsebesség a rúdban) jellemzi, felírható az anyag rugalmasságát jellemző Young-modulus (E), és rúd sűrűségének (ρ) hányadosaként kapott eredmény gyökének kiszámításával. A K inerciasugar meghatározható a rúd magassága (h) osztva $\sqrt{12}$ -vel.



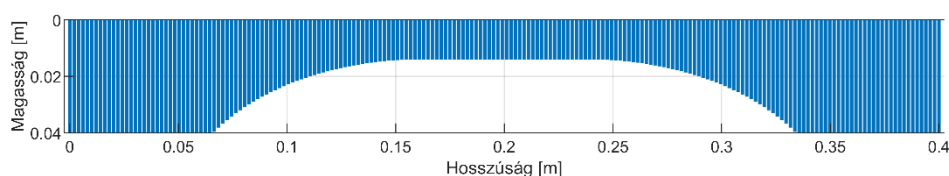
3.1. ábra: Módusalakok és frekvenciák analitikus módszerrel

A 3.1. ábrán a módszer alkalmazásaként kapott eredmény látható. Az első öt módus látszik a grafikonokon, és a frekvenciák, amiken az egyes módusok rezegnek leütéskor. A grafikonokat úgy kell értelmezni, hogy a hossz mentén egy bizonyos pozícióban a hanglapra való leütésnél, le lehet olvasni az adott módusalak értékét, ami többnyire azt határozza meg, hogy milyen erősséggel vesz részt a teljes hanglap rezgésében. Végössoron tehát a rúd deformációja felírható a módusalakok szuperpozíciójaként. A szuperpozícióban szereplő súlyok a modális koordináták, melyek szabadrezgés esetén a sajátfrekvenciáknak és a csillapításértékeknek megfelelően exponenciálisan csillapodó harmonikus időfüggvények.

Ennek a módszernek az előnye, hogy létezik analitikus megoldása a test rezgését leíró differenciálegyenletnek, így könnyedén meg lehet határozni a hajlító rezgéseket a rúd fizikai jellemzőinek ismeretében. A probléma ezzel az, hogy a valódi marimba hanglapok nem szabályos téglatestek. Ennek az az oka, hogy a konstans vastagságú szabályos rúd esetén a sajátfrekvenciák arányai nem lesznek harmonikusak egymással, és így zenei szempontból nem fog szép hangot eredményezni a rúd szabadrezgése. Ezt a problémát a valóságban a hanglapok alsó részén íves bevágással oldják meg, így érhető el a rudak megfelelő hangolása. Ennek következtében a rúd alakja inhomogén lesz, amely esetén a szabadrezgéseket leíró differenciálegyenletet nem lehetséges megoldani analitikus módszerekkel, így a fenti képletek nem lesznek alkalmasak ennél a modellenél történő használatra.

3.1.2 Végeelem módszer

A hanglap rezgéseit más úton kellett jellemezni, amire végül az egydimenziós végeelem-módszert használtam. A módszer lényege, hogy a teljes rúd fizikai jellemzésére a rudat sok kisebb elemekre darabolom fel a hossza mentén, ezután az egyes elemek tömeg- és merevségmátrixainak felhasználásával kiszámolhatók a teljes rúdra a sajátértékek és sajátvektorok. Az így kapott sajátértékek és sajátvektorok tartalmazzák a rúd módusalakjait, és a módusok sajátfrekvenciáit. Ez a módszer nem teljesen pontos eredményeket ad, de a marimba hanglapjának fizikai jellemzőivel számolva jó közelítést ad a módusok valódi értékeire. Ezt a módszert használtam a végső modell során is.



3.2. ábra: Végeelem módszer hanglap oldalnézetben

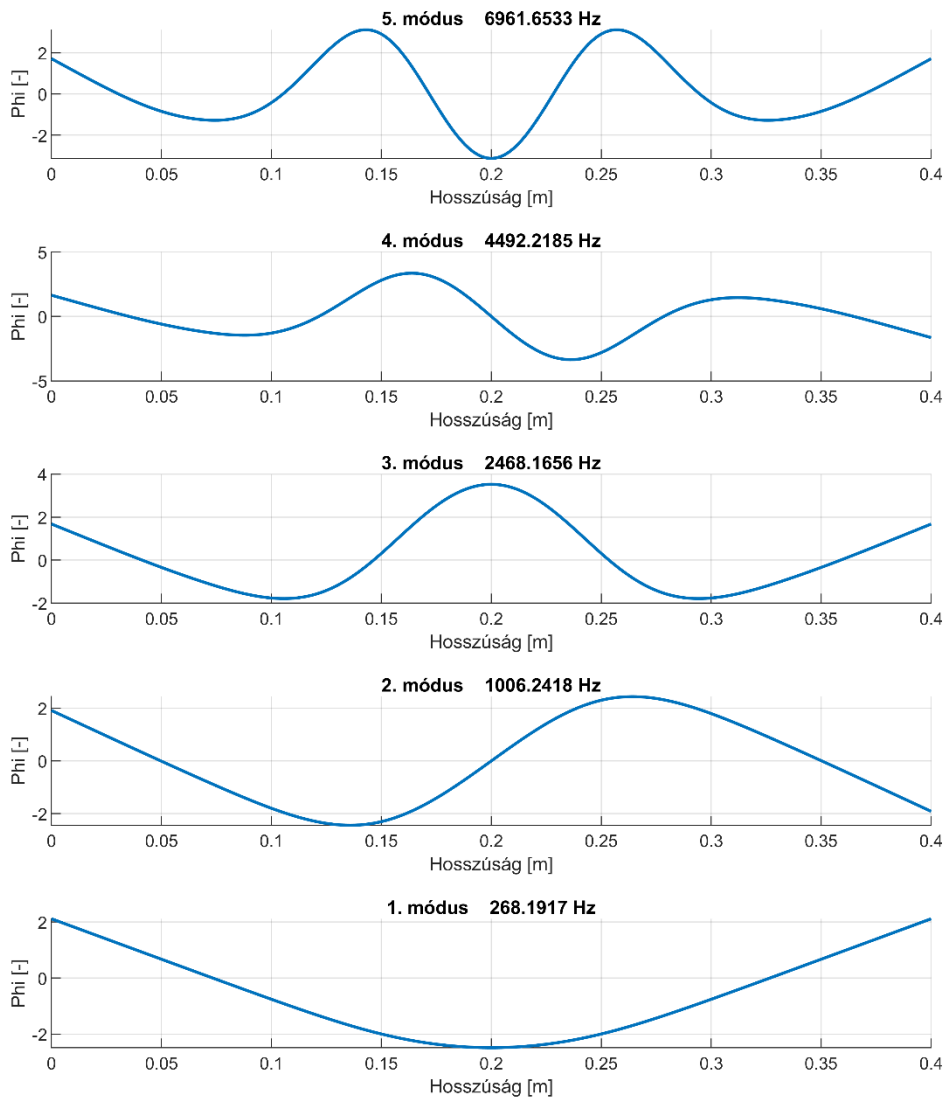
A 3.2. ábra szemlélteti a módszer alkalmazását egy konkrét esetben. Az ábrán egy 40 cm hosszú, 4 cm magas, és 6 cm széles hanglap látható oldalnézetben. Az is kivehető a grafikomból, hogy a hanglapnak nem egységes a magassága, ami a korábban említett, a megfelelő hangolás elérésére alkalmazott bevágás miatt van. A rudat hossza mentén 200 részre osztottam, ezek a végeelem módszernél használt elemek, gyakorlatilag kis téglatestek, ezek segítségével számolhatók a teljes testre jellemző módusalakok.

További fizikai jellemzők, amik a számoláshoz szükségesek:

$$\rho = 830 \text{ kg/m}^3$$

$$E = 1 \cdot 10^{10} \text{ Pa}$$

A leírt paraméterek felhasználásával kiszámított módusalakok a 3.2. ábrán láthatók:



3.3. ábra: Módusalakok és frekvenciák végelem módszerrel

A 3.3. ábrán található a végelem módszer alkalmazása során kapott módusalakok és sajátfrekvenciájuk. Érdekes összehasonlítani a 3.1. ábrán szereplő függvényekkel és frekvenciákkal. Mindkét esetben azonos paraméterekkel számoltam, annyi különbséggel, hogy a végelem módszer esetén a rúd vastagsága (magassága) nem

konstans 4 cm, hanem a bevágás mértékének függvényében középen vékonyítva van. Észrevehető a 3.1. esetében, hogy a módusalakok szabályosabbak és szinuszos alakúak, ami nem meglepő, hiszen tényleg ilyen függvényekből épülnek fel a fenti képletek alapján. A végeelem módszernél a módusok alakjai már nem emlékeztetnek egyszerű függvényekre, változik az amplitúdójuk is a hossz mentén.

További különbségek jelennek meg a két eset között a frekvenciákban. A 3.3. ábrán egyrészt alacsonyabbak lettek az egyes módusok sajátfrekvenciái, másrészt az egymáshoz vett arányaik is különböznek. Ez a rúdon ejtett bevágás következménye, ami azt jelenti, hogy megfelelő bevágással behangolható a hanglap úgy, hogy annak megfelelő zenei hangzása legyen, például az 1:4:10 arányok beállításával.

A hanglap modellezését tehát a végeelem módszerrel oldottam meg, az implementációnál még lesz szó a konkrét megvalósításról. A fent említett konkrét paraméterek csak példaként szolgáltak, ezek természetesen konfigurálhatók úgy, hogy egy valódi behangolt marimba hanglapot nagy pontossággal lehessen modellezni.

3.2 Az ütő-hanglap kölcsönhatás

Az előző fejezetben bemutattam a hanglap hangjának modellezésére felírható rendszert. A következő lépés, hogy ezt a rendszert valamilyen erővel gerjeszteni kell, hogy megkapjuk a rendszer válaszát. Ez nem meglepő, mivel a valóságban is a hanglapra kell ráütni az ütővel, hogy a hangszer megszólaljon.

Néhány feltételezést alkalmaztam az ütő-hanglap kölcsönhatásnál, annak érdekében, hogy a modell ne legyen túlságosan bonyolult, ugyanakkor az egyszerűsítésekkel együtt is minél pontosabb maradjon a modell. Az egyik feltételezés szerint a leütés közben az ütő feje rugalmasan összenyomódik, miközben a hanglap nem mozdul el a kölcsönhatás végéig. A másik egyszerűsítés, hogy a kölcsönhatási erő pontszerűen hat a hanglap egy meghatározott pontján. A valóságban ezek a megkötések nem teljesülnek, ugyanakkor nem is befolyásolják számottevően a kölcsönhatás során kialakult erőfüggvényt, és alkalmazásuk lényeges leegyszerűsíti a folyamatot.

A leütés során létrejött erő kiszámításához szükség van néhány fizikai paraméter ismeretére. Ilyen például az ütő fejének anyagára jellemző értékek, mint a tömege és az anyagának keménysége, de ezeken kívül még a leütés sebessége is fontos szerepet játszik a hangszer megszólaltatásánál [8].

A kalapácsütéseknél, ami a marimba megszólaltatását is jellemzi, a létrejött gerjesztő erő nemlineáris jellegű. Leütésnél az ütő fejének összenyomódása határozza meg a hanglaphra ható erőt, ezek között azonban nem tekinthető lineárisnak az összefüggés, aminek következménye, hogy analitikus módszerekkel nehézkes az erő előállítás. Ebben az esetben is közelítést alkalmaztam, a Newmark-módszerrel számoltam ki az erőt.

A Newmark-módszer időben másodrendű differenciálegyenletek megoldására használható, pontosabban az egyenletek egy kezdetiérték-problémáját lehet megoldani a segítségével. A megoldás során bizonyos, tetszőlegesen sűrű időközönként lehet végig menni a meghatározott időtartományban, és a jelenlegi elmozdulás, sebesség és gyorsulás értékek ismeretében kiszámíthatók a következő időpontban felvett értékek. Ezzel a módszerrel modellezhetők a teljes függvények jó megközelítéssel, viszont fontos a kellő mintavételezési frekvencia.

A Newmark módszer által meghatározott egyenletek [9]:

$$u_{n+1} = u_n + \Delta t \dot{u}_n + \frac{\Delta t^2}{2} ((1 - 2\beta)\ddot{u}_n + 2\beta\ddot{u}_{n+1})$$

$$\dot{u}_{n+1} = \dot{u}_n + (1 - \gamma)\Delta t \ddot{u}_n + \gamma\Delta t \ddot{u}_{n+1}$$

$$M\ddot{u}_{n+1} + C\dot{u}_{n+1} + Ku_{n+1} = F_{n+1}$$

Az erő-összenyomódás összefüggés képlete [6]:

$$F_M = K_M \delta_M^{3/2}$$

A fenti képletekben u_n , \dot{u}_n és \ddot{u}_n jelöli az adott n időpontban az elmozdulás, sebesség és gyorsulás függvények értékeit. Δt a két időpont közötti időkülönbség (leggyakrabban ez egy adott mintavételi időköz, amely akár változhat is a szimuláció során). M , C és K mátrixok a tömeg, csillapítás és keménység mátrixai. δ_M az ütő fejének összenyomódását jelöli, ami jellemezhető a hanglaphoz képest történő elmozdulásával. β és γ konstansok, ezeknek választható a pontos értéke, én $\beta = 0.25$ és $\gamma = 0.5$ -el számoltam. Ezzel a választással a Newmark-módszer tetszőleges Δt időlépés mellett stabil. F_M az ütő által a hanglaphra kifejtett erőt jelöli.

A Newmark-módszer általános megoldásának lépései [10] alapján:

Segédváltozók felvétele (prediktor lépés):

$$\tilde{u} = u_n + \Delta t \dot{u}_n + \frac{\Delta t^2}{2}(1 - 2\beta)\ddot{u}_n$$

$$\tilde{\dot{u}} = \dot{u}_n + (1 - \gamma)\Delta t \ddot{u}_n$$

ebből $M\ddot{u}_{n+1} + C\dot{u}_{n+1} + Ku_{n+1} = F_{n+1}$ összefüggés alapján

$$\ddot{u}_{n+1} = (M^*)^{-1}(F_{n+1} - C\tilde{\dot{u}} - K\tilde{u})$$

ahol

$$M^* = M + \gamma\Delta t C + \Delta t^2 \beta K$$

Végül a gyorsulás segítségével kiszámolható a sebesség és elmozdulás is a következő időpontban:

$$\dot{u}_{n+1} = \tilde{\dot{u}} + \gamma\Delta t \ddot{u}_{n+1}$$

$$u_{n+1} = \tilde{u} + \Delta t^2 \beta \ddot{u}_{n+1}$$

A kalapács esetén ennek egy speciális esetét oldom meg, amely néhány lépésben különbözik az általános megoldástól. Kiszámolható az F_M erőfüggvény értéke a következő időpontra:

$$F_M = \begin{cases} K_M |\tilde{u}^{3/2}|, & \text{ha } \tilde{u} < 0 \\ 0, & \text{egyébként} \end{cases}$$

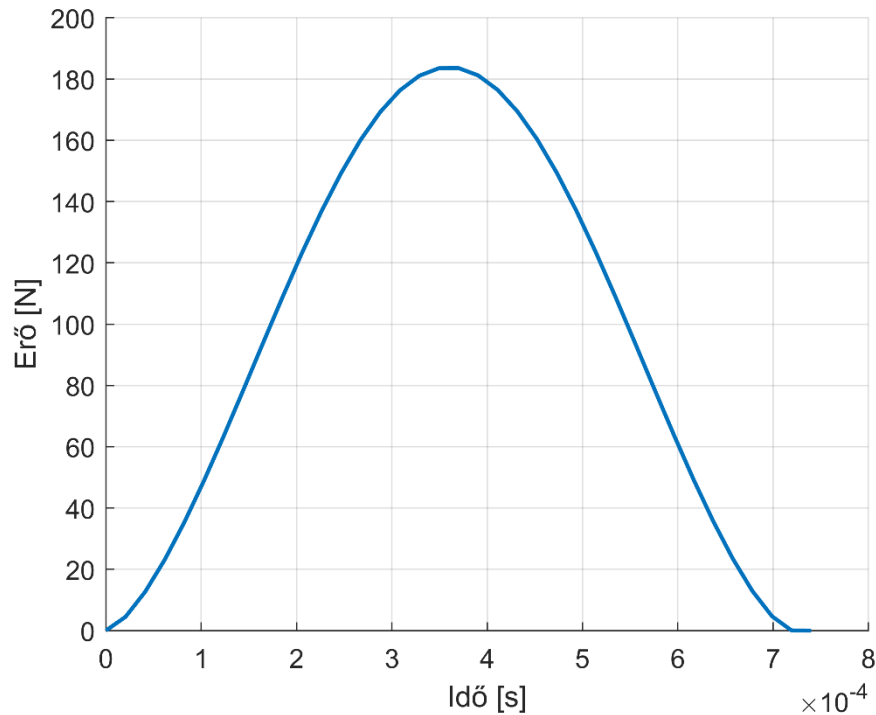
Az ütő esetén a gyorsulás függvény értékét a következő időpontban az alábbi egyenlet írja le. Ez egy egyszerűbb eset, mivel K és C mátrixok értékeit 0-nak veszem. A többi lépés megegyezik az általános megoldással.

$$\ddot{u}_{n+1} = F_M (M_M)^{-1}$$

A módszer segítségével és kellően sűrű időközönként számolva, valamint a teljes időtartományban végig iterálva elég pontosan ki lehet számolni az erőfüggvény értékeit a meghatározott fizikai jellemzőket figyelembe véve. A 3.3. ábra egy kiszámolt erőfüggvényt ábrázol, ahol 1 m/s volt az ütő kezdősebessége, valamint a következők voltak az ütő anyagának jellemzői:

$$M_M = 35 \times 10^{-3} \text{ Kg}$$

$$K_M = 5 \times 10^7 \text{ Nm}^{-3/2}$$



3.3. ábra: Az erőfüggvény

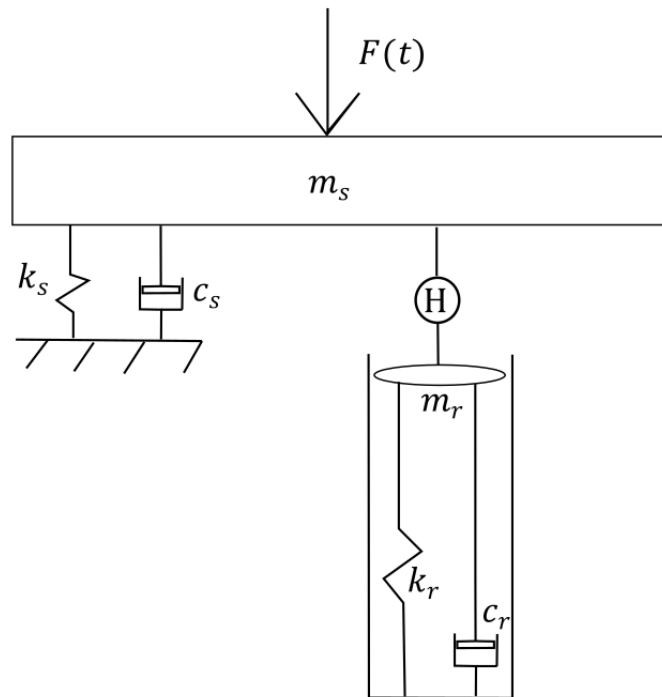
A létrejött erőfüggvény ránézésre egy fél szinusz függvényre hasonlít, valójában azonban a nemlinearitás miatt ettől némileg eltér. Azt is fontos megemlíteni, hogy az erőhatás hossza változhat attól függően, hogy milyen sebességgel történik a leütés, ezzel pedig változik a megszólaló hang hangszíne is.

3.3 Rezonátor hatása a rendszerre

Az utolsó főbb része a modell felépítésének a hanglap-rezonátor kölcsönhatás figyelembevétele. A rezonátor olyan módon hat a hanglaptól lesugárzott hangra, hogy a rezgés első módusának hangját felerősíti és visszasugározza. A rezonátor általában cső alakú, amely az egyik végén zárt, így önmagában egy nagyon hatékony, monopólus sugárzóként viselkedik. A hanglap ezzel ellentétben, egy gyengébb dipólus sugárzó. Az erősebb lesugárzott hangnyomás mellett egy fontos hatása még, a rezonátorral megszólaló hanglap lecsengése gyorsabb lesz az önmagában álló hanglapéhoz képest, amivel az összehatás is megváltozik, ezért fontos ezt az elemet is figyelembe venni a modellezésnél.

A rezonátornak is vannak különböző fontos fizikai tényezői, amelyek meghatározzák, hogy miként fogják befolyásolni a hanglaptól lesugárzott hangot. Ilyen például a csőbe zárt légoszlop tömege, a cső átmérője, csillapítási tényezője, vagy az,

hogy milyen frekvenciára hangolták. A hangolást általában úgy állítják be, hogy a hozzá közel eső hanglap alapfrekvenciájával majdnem megegyező legyen. Kis mértékben el szokták hangolni a megfelelő hangzás eléréséhez, de általában nagyon közel van a hanglap frekvenciájához. A hanglap 1:4:10 frekvenciaarányaihoz képest az egyik végén nyitott rezonátorra az 1:3:5: ... arányok jellemzők, ezért a hanglap és a rezonátor első módusa között van számottevő kölcsönhatás [11].



3.4. ábra: Csatolt rendszer szemléltetése

Soares és társai egy új, 2022 elején megjelent tanulmányukban [12] egy egyszerűsített csatolt rendszert mutatott be a hanglap–rezonátor kölcsönhatás modellezésére. Én ezt használtam fel, mert jól használható a Newmark módszerrel kombinálva.

A csatolt rendszert leíró képlet:

$$\begin{bmatrix} m_s & \Pi \\ \Pi & m_r \end{bmatrix} \begin{Bmatrix} \dot{Y}(t) \\ \dot{\gamma}(t) \end{Bmatrix} + \begin{bmatrix} c_s & 0 \\ 0 & c_r \end{bmatrix} \begin{Bmatrix} \dot{Y}(t) \\ \dot{\gamma}(t) \end{Bmatrix} + \begin{bmatrix} k_s & 0 \\ 0 & k_r \end{bmatrix} \begin{Bmatrix} Y(t) \\ \gamma(t) \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \end{Bmatrix}$$

ahol

$$k_s = m_s \omega_s^2 \qquad k_r = m_r \omega_r^2$$

$$c_s = 2\zeta_s m_s \omega_s \quad c_r = 2\zeta_r m_r \omega_r$$

$$\Pi = \rho_0 S H \varphi(L)$$

$$S = \pi a^2$$

$$\varphi(x) = -\left(\frac{c}{\omega_r}\right) \cos\left(\frac{\omega_r}{c} x\right)$$

A képletben Y és deriváltjai a hanglap, γ és deriváltjai pedig a rezonátor elmozdulását és annak deriváltjait jelentik. A hanglap esetében Y annak a pontnak az elmozdulása, amelyik a rezonátor közepe fölött van. Ennek a pontnak az elmozdulását pedig a módusalaknak az ebben a pontban felvett értéke és a modális koordináta szorzata határozza meg. m_s , m_r , k_s , k_r , c_s és c_r jelölik a hanglaphoz és a rezonátorhoz tartozó tömeget, merevséget és csillapítást. ζ_s és ζ_r csillapítási arányokat határoznak meg. ρ_0 a levegő sűrűsége, S a cső keresztmetszetének területe. ω_s és ω_r a sajátfrekvenciák körfrekvenciái, c a hangsebesség.

A H függvény jelentése még nem ismert, ehhez azonban némi magyarázat szükséges. H az átviteli függvényt jelöli, amely befolyásolja a hanglap és a rezonátor közötti csatolást. Az átvitelt a hanglap gyorsulása és a rezonátor nyílt végénél lévő nyomás hányadosaként értjük, és ugyanilyen átviteli függvény szerint hat vissza a rezonátor nyomása a hanglapra egy erővel. Ez a függvény egy közelítésen alapul, amely egy dimenzió nélküli távolságtól, a d/a hányadostól függ. A következő képlet írja le a H függvény kiszámításának módját:

$$H\left(\frac{d}{a}\right) = \left(1 + \sum_{p=1}^P \delta_p \left(\frac{d}{a}\right)^p\right)^{-1}$$

$$\delta_p = [2.78, -1.03, 3.34, -1.15, 0.13]$$

H függvény abból adódik, hogy a hanglapot a rezonátor pozíciójában merev dugattyúként közelíti a modell, és a hengerszimmetrikus végeselem-módszerrel kiszámított frekvenciafüggő átvitel kisfrekvenciás közelítése lesz a H . P értéke határozza meg a polinom fokszámát, [10] szerint a legjobb közelítést ötödfokú polinommal lehet elérni. d/a szabadon választható bizonyos keretek között, én az alkalmazás során 0.5 értékkel számoltam, de ez változtatható a végső modellben.

A képletek segítségével ki lehet számolni a rezonátor választ a hanglap első módusának rezgésére egy adott időpontban. A teljes időtartományra a Newmark-módszert használtam az erőfüggvény számításához hasonlóan. A végén egy 3 másodperces tartományban egy adott mintavételezési gyakorisággal előállítottam a mintákat, ezek is változtatható paraméterek az implementált modellben.

3.4 A teljes modell elkészítése

Az előzőkben ismertettem, hogy a modell milyen elemekből épül fel. Az maradt hátra, hogy ezeket az elemeket felhasználva elő lehessen állítani a végső hangmintát, amit már meg lehet hallgatni.

Ennek módja nem teljesen magától értetődő, mivel a rendszernek két hangforrása is van, a hanglapból lesugárzott hangot is hallja az emberi fül, valamint a rezonátor is visszasugározza a hangot. A végső hang meghatározásához ki kell számolni a hanglapból és a rezonátorból érkező hangnyomást.

A rezonátor egy monopólus sugárzóként működik, a lesugárzott hangnyomást a rezonátor alakjának jellemzői, egyes csillapító jellemzők és a lehallgatási pozíció távolsága határozza meg. A következő képlet adja meg a hangnyomást:

$$p_r = \frac{-Sp\dot{v}_r}{4\pi r}$$

ahol

$$p = \sin\left(\frac{\omega_r}{c}L\right)$$

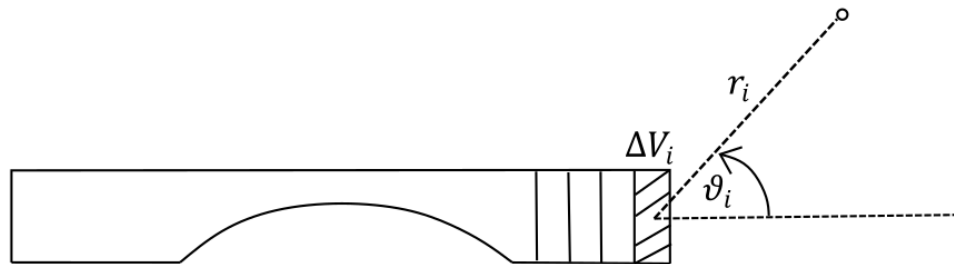
A hanglapon ezzel ellentétben az egyes elemei dipólus sugárzóként viselkednek, ezért ebben az esetben a hangnyomást kicsit eltérően kell számolni. Itt már számít a lehallgatási pozíció és a hanglap elemei által bezárt szög is, amelyek befolyásolják a lesugárzott hangot.

A hanglap i -ik eleme által lesugárzott hangnyomás egy adott lehallgatási pozícióban az időkézés figyelembevétele nélkül [13]:

$$\Delta p_i = \frac{3}{8\pi} \rho_0 \Delta V_i \cos(\vartheta_i) \left[\frac{1}{r_i^2} \dot{v}_i + \frac{1}{r_i c_0} \ddot{v}_i \right]$$

$$p_b = \sum_{i=1}^N \Delta p_i$$

Ezekben a képletekben szereplő paraméterek nagy részének értéke tetszőlegesen beállítható az implementációban, célszerű valódi méreteket, anyagtulajdonságokat, frekvenciát választani a valódihoz hasonló hangzás elérésére. A hanglap esetén a végelem módszerből adódóan a hallgatási pozíciótól vett távolságot és szöget minden egyes elemre ki kell számolni.

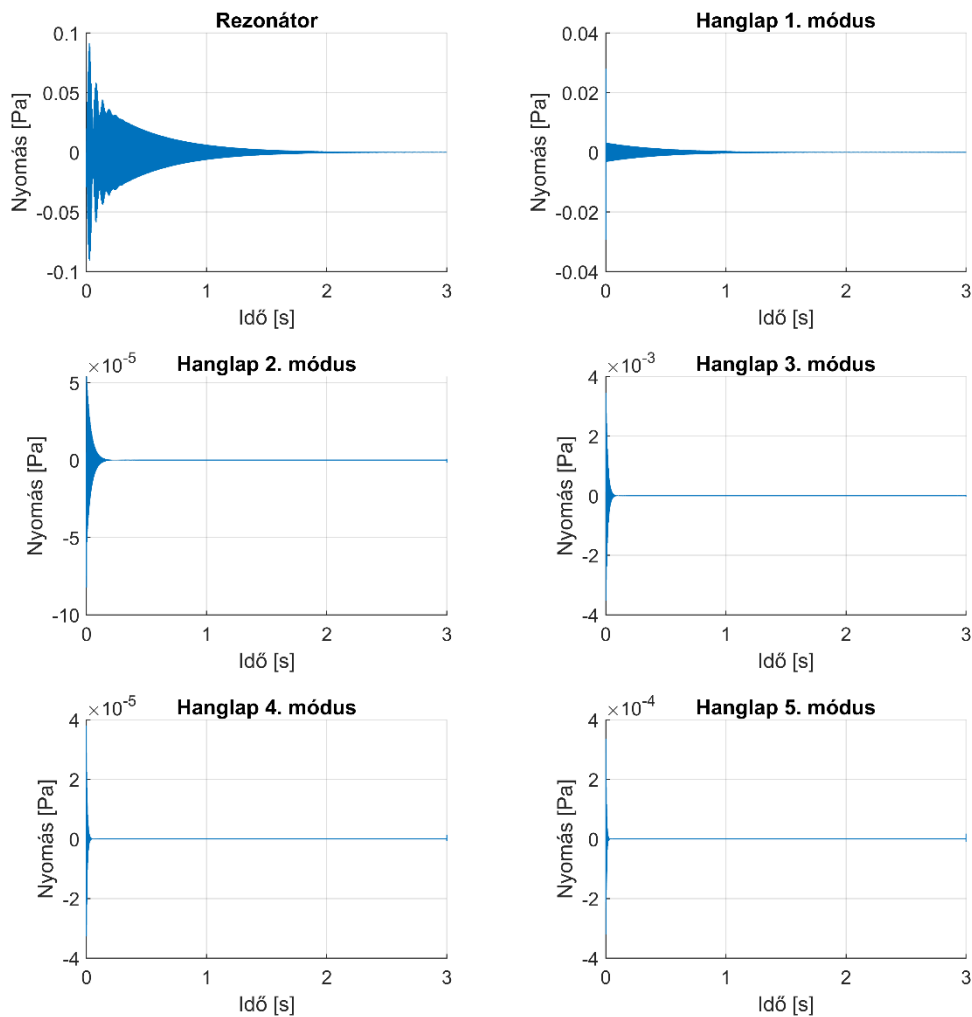


3.5. ábra: Hanglap, mint dipólussugárzó szemléltetése

Ezután csak a gyorsulás és a gyorsulás derivált függvénye hiányzik ahhoz, hogy kiszámolható legyen a hangnyomás. Ezeket két féle módon lehet megkapni, de mindkettőhöz szükséges a korábban ismertetett gerjesztő erőfüggvény:

- A rezonátor és a hanglap első módusánál a hanglap-rezonátor kölcsönhatást leíró csatolt rendszer és a Newmark módszer alkalmazásával kiszámolhatók az elmozdulás, sebesség és gyorsulás függvények. A hanglaponál ezen kívül a gyorsulás deriváltját is ki kell számolni.
- A másik esetben, a hanglap magasabb módusainak esetében már nincs csatolás a hanglap és a rezonátor között, így érdemesebb kevésbé számításgényes módszert alkalmazni a Newmark helyett. Ennél az esetben frekvenciatartományba kell alakítani a gerjesztő erőt, majd kiszámolni a magasabb módusok válaszát. A gyorsulás és gyorsulás derivált előállítás után vissza kell alakítani időtartományba ezeket a függvényeket.

Így már kiszámolhatók egyenként a p_b és p_r függvények, amelyek leírják a hangnyomást az idő függvényében. p_b -t minden módus esetén ki kell számítani. Ezek összege lesz a végső hallható hang, amit az emberi fül is hallana egy megadott távolságra és pozícióban a hangszerhez képest. A következő ábrákon 3.6. megmutatom az egyes módusok hangnyomás függvényeit.



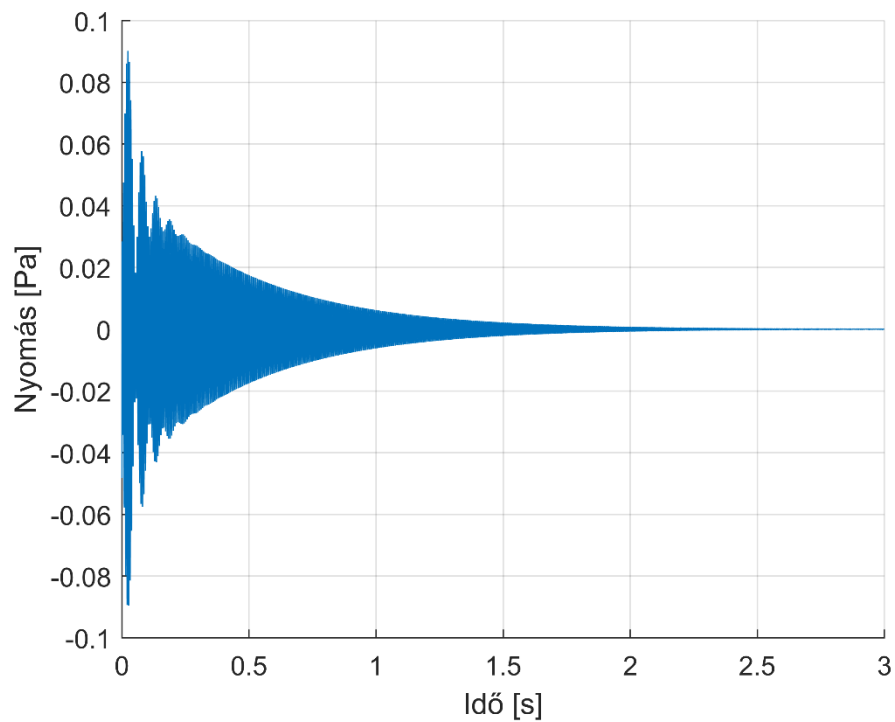
3.6. ábra: Hangnyomás függvények

Jól látható, hogy a rezonátor hangja a legerősebb, és a leghosszabb ideig is tart, a hanglap esetében a magasabb módusok egyre kisebb intenzitással vannak jelen (ez a leütési pozíciótól is függ), és egyre gyorsabb a lecsengési idejük. Ez annak köszönhető, hogy a magasabb módusoknak erősebb a csillapítása.

$$\tau = \frac{1}{\omega_0 \xi}$$

Csillapított sajátrezgéseknél egy τ időállandót határoznak meg [6], így általánosságban elmondható, hogy minél kisebb ez az időállandó, annál gyorsabb lesz a lecsengés. Ebből következik, hogy nagyobb frekvencia esetén kisebb lesz az időállandó értéke, ez magyarázza a módusok esetén is, hogy a magasabb frekvenciájú módusok hamarabb lecsengenek.

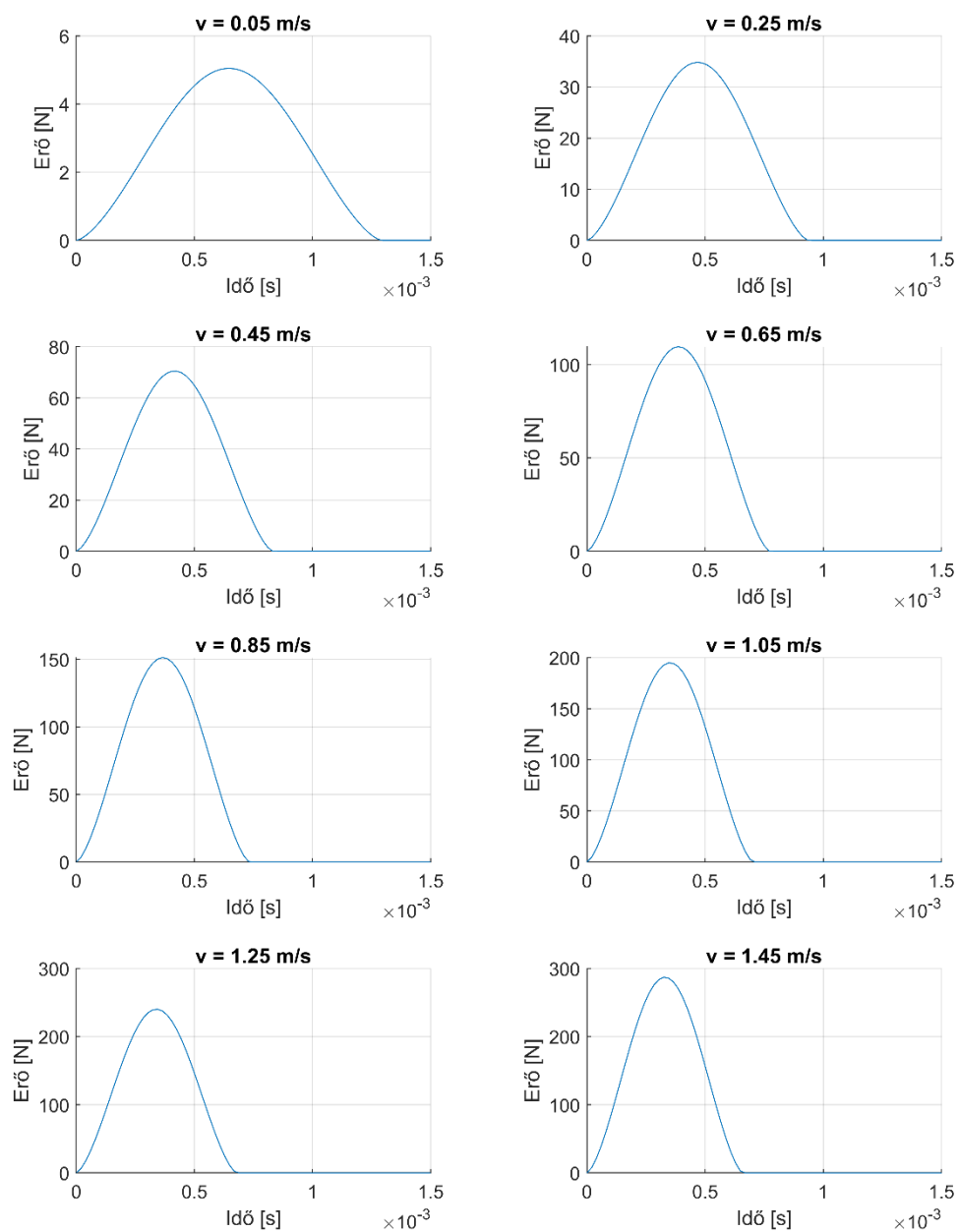
Végül a 3.6-es ábrán látható függvények összegeként megkapható az adott pozícióban hallható hang, ezt a 3.7. ábra mutatja.



3.7. ábra: A végső hang függvénye

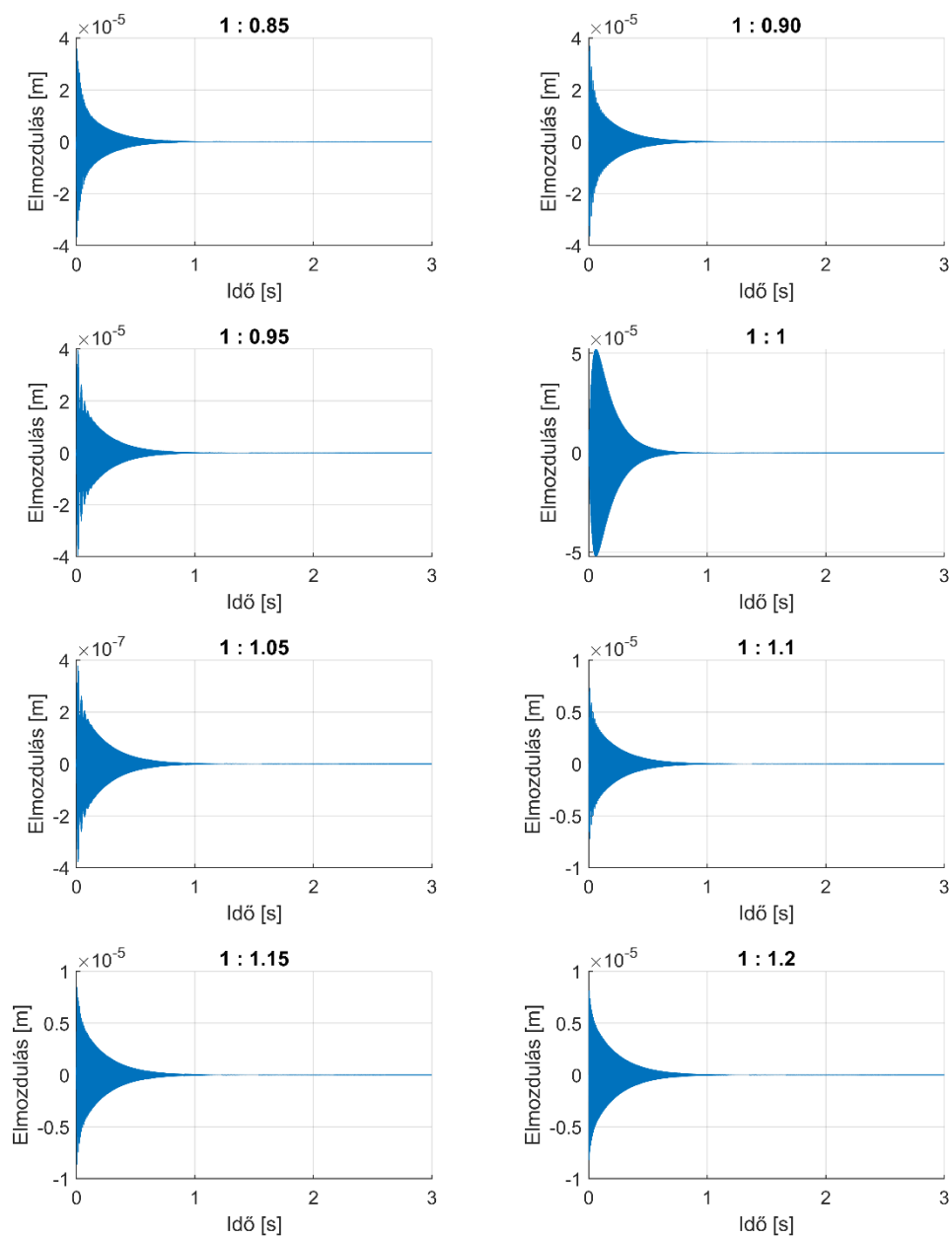
3.5 Példák a modell tesztelésére

A 3.8. ábrán különböző kezdősebességű leütések szimulációja során kapott erőfüggvények kerülnek bemutatásra. Itt megfigyelhetők az erősségek és a leütési idők változásai.



3.8. ábra: Erőfüggvények összehasonlítása különböző sebességek esetén

A 3.9. ábra bemutatja a rezonátor elmozdulásait a hanglap alulfrekvenciájához képest történő hangolás arányainak változtatása során.



3.9. ábra: A rezonátor elmozdulása különböző hangolások esetén

Ezzel elkészült a modell, ezt kell előállítani digitálisan is, egy VST plugin implementációként, amit a következő fejezetben fogok megmutatni.

4 A szintetizátor VST implementációja

A modell sikeres megalkotása után a szükséges módszereket át kell ültetni szoftveres formába is. Egy teljes szintetizátor program megvalósítása valamilyen kiindulási alap nélkül nagyon komplex feladat. Szerencsére ma már létezik többféle szabvány és fejlesztőkörnyezet ennek megoldására, így számomra is ez volt a kedvezőbb opció.

A következőkben bemutatom a megvalósításhoz szükséges technológiai hátteret, a fejlesztéshez használt szoftveres környezetet, valamint a konkrét implementáció folyamatát és a végső szoftver felépítését.

4.1 Virtual Studio Technology

A Virtual Studio Technology (VST) egy audio plug-in szoftver interfész, amely segíti a szoftveres szintetizátorok és effektek beintegrálását digital audio workstation-nek (DAW) nevezett applikációkba. A VST plugin-ok tehát általában önmagukban nem működnek, hanem ezekben a bizonyos DAW programokba épülnek be és ott futtathatóak. A VST plugin-oknak két fő kategóriája van, az egyik a hangszerek (VSTi), a másik az effektek (VSTfx). A VST-t a Steinberg nevű cég fejlesztette.

Sok esetben grafikus felülettel is rendelkeznek, amely a valódi vezérlő elemek kinézetét is megpróbálja utánozni, például fizikai kapcsolók, gombok, vagy állítható potenciométerek. Manapság már sok jól ismert hardveres szintetizátornak létezik VST változata, amely az eredeti hangzását szimulálja, és a kinézetben is igyekszik hasonlítani hozzá. Ez jelentősen segítheti a zenészek, hangmérnökök munkáját, hiszen egyes valós eszközök már nehezen elérhetőek, vagy nagyon drágák lennének.

A VST hangszerek a lejátszandó hangokat MIDI segítségével kapják meg a bemenetükre, ami egy szabványos interfész elektronikus hangszerek vezérlésére. Így akár lehetséges a számítógép billentyűzetével vagy grafikus felületen történő gombok lenyomásával vezérelni a hangszert. Léteznek külön erre a célra készített MIDI vezérlő eszközök is, amelyeken például zongorabillentyűk vannak, ezek is kiválóan alkalmasak VST plugin-ok vezérlésére. A lejátszandó hangon kívül egyéb paramétereket is lehet vezérelni MIDI segítségével, például a billentyűlenyomás sebességét is figyelembe tudja

venni a plugin, ha támogat ilyen funkciót. Kimenatként digitális hangot állítanak elő a VST hangszerek.

VST effektek esetén bemenatként is digitális hangot vezetnek rá, ezt dolgozza fel a plugin, majd a kimenetén visszaadja a hangot. Lehetséges akár egymás után kapcsolni több különböző plugin-t is a DAW applikáció segítségével [14].

4.2 Digital Audio Workstation

A Digital Audio Workstation (DAW) egy elektronikus eszköz vagy szoftveres alkalmazás, amelyet hangfájlok felvételére, módosítására és készítésére terveztek. Többféle variációban előfordulnak, ez terjedhet egy laptopon futó szoftveres programtól, egészen komplex eszközökig is, amelyek sok különböző komponensből és egy központi vezérlő egységből állnak. A DAW applikációkat elsősorban zenék, TV vagy rádióadás, podcast-ok, hangeffektjeinek elkészítésére.

Korábban léteztek hardver alapú DAW eszközök is, azonban manapság a szoftveres alkalmazások terjedtek el. Ez nem is meglepő, hiszen ma már személyi számítógépeken, mobiltelefonokon, tableteken is használhatóak ezek a programok, így nincs szükség egy külön eszköz vásárlására.

A DAW működéséhez néhány alap komponensre van szükség: egy számítógépre, hangkártyára, vagy egyéb audio interfészre, egy hang módosító szoftverre (manapság ezt szokták DAW alatt érteni, de tradicionálisan ez is egy részegysége volt a DAW-nek), és legalább egy bemeneti eszközre, ami lehet akár egér és billentyűzet, vagy például egy zongora stílusú MIDI vezérlő [15].

Sokféle szoftveres DAW létezik különböző funkciókkal, vannak egyszerűbb, csak limitált funkcionalitással rendelkezők, és vannak komplexebb, professzionális munkavégzésre tervezett, általában fizetős megoldások is. A dolgozatomban szempontjából fontos tulajdonságuk, hogy a legtöbb DAW támogatja a VST plugin-ok futtatását is, ezzel kiegészítve saját funkcionalitását. Ennek a szabványos módszernek az az előnye, hogy a VST hangszerek tervezésénél nem kell a bemeneti és kimeneti eszközök kezelésével foglalkozni, ezeket a DAW program kezeli.

Az általam fejlesztett VST szoftver futtatásához az FL Studio [16] nevű DAW alkalmazás ingyenesen használható verzióját használtam, amely alkalmas volt a hangzás

és a grafikus felület tesztelésére. Természetesen más program is használható, ami támogatja a VST szabványt.

4.3 Steinberg VST SDK

Az előző alfejezetekben bemutatam a fejleszteni kívánt szoftver alapjait nyújtó VST szabványt és használatát, azonban arról még nem beszéltem, hogy milyen módon lehetséges ilyen programokat fejleszteni.

VST plugin fejlesztéséhez általában érdemes valamilyen fejlesztőkörnyezetet használni, amely a program alapműködéséhez szükséges funkciókat megvalósítja, és felépíti a szoftver vázát. Ilyen fejlesztőkörnyezetből több is létezik, én a Steinberg VST SDK (Software Development Kit) [17] környezetet használtam a VST szintetizátor elkészítéséhez.

A Steinberg cég által készített SDK ingyenesen letölthető és használható, és egy jó keretet ad digitális hangszerek vagy effektek fejlesztéséhez, ezért az én céljaimnak is megfelelt. A fejlesztőkörnyezet C++ programozási nyelv segítségével íródott, így az implementáció során is ezt a nyelvet kellett használnom. A CMake [18] program segítségével létrehozhatók a projektfájlok a fejlesztéshez, többféle programozási IDE-t (Integrated Development Environment) is támogat, és platformfüggetlen.

A VST plugin futtatásához a projekt generálása után fordítani kell a programot, ennek hatására létrejön egy .vst3 kiterjesztésű bináris fájl, és egyéb forrásfájlok is mellé kerülnek, amelyek a grafikus felülethez tartozó képeket és leírófájlt tartalmaznak. Ezt követően már futtatható a plugin, egy DAW alkalmazás segítségével megnyitható és kipróbálható.

A Steinberg SDK tartalmaz példaként szolgáló kész projekteket is, ezek különféle effekteket, szintetizátorokat valósítanak meg, és jól bemutatják a működési elvet, amivel saját plugin-okat is lehet tervezni. Lehetőség van üres projekt létrehozására is, ilyenkor csak a program vázát képező alapsztyályokat tartalmazza, ez abban az esetben hasznos, ha teljesen előlről, az alapoktól szeretnénk kezdeni a fejlesztést.

Az implementációnál az egyik példaprojektet vettem alapul, ami egyrészt hasznos volt a plugin felépítésének és működésének megértéséhez, másrészt így nem kellett minden alap funkcionális megvalósítanom, hanem a feladat szempontjából fontosabb modell implementálására tudtam fókuszálni.

A következőkben a kiindulásként felhasznált projekt segítségével mutatom be általánosan egy VST plugin alapműködését és tulajdonságait.

4.3.1 Note Expression Synth

Az SDK példaprojektjei között szereplő Note Expression Synth nevű plugin alapján indultam el az implementáció során. Azért ezt a projektet választottam, mert ez volt az egyetlen a példák között, amely egy teljes szintetizátort valósított meg.

A plugin alapvetően egy additív elveken alapuló szintetizátort alkot, amelyben egy oszcillátor szolgáltatja a hangot, szinusz, háromszög vagy négyzetjel alapján. Ezt a hangot egy filter segítségével megszűrjük. További beállítási paraméterei a hangerő, sebesség, hangolás, és lecsengési idő. A legtöbb paraméter a szintetizátor felhasználói felületén állítható, az egér segítségével lehet a megfelelő állásba hozni a potenciométereket. A 4.1. ábra mutatja a teljes felhasználói felületét beállítható paraméterekkel és egy zongorabillentyűket tartalmazó vezérlővel.



4.1. ábra: Note Expression Synth felhasználói felülete

Ez a plugin jó alapot szolgáltatott az általam tervezett szintetizátor megvalósításához és a fejlesztőkörnyezet működésének megértéséhez, azonban sok

mindenben különbözött is. Mivel én fizikai modell alapján, és nem additív szintetizátort készítettem, ezért sok dolog felesleges volt a plugin-ban. Nem volt szükség oszcillátorokra, filterre, néhány beállítási paraméterre, mert teljesen más módon állítom elő a digitális hangmintákat, így a programkód ezen részeit eltávolítottam.

Végül tehát így a szintetizátor váza maradt meg a példa plugin módosításával, ahonnan el tudtam kezdeni a modell implementálását. Összességében ez előnyösebb volt, mint egy üres projektből indulni, volt néhány hasznosítható része, nem kellett minden alapfunkciót elkészítenem.

4.4 További felhasznált szoftverek

A fejlesztőkörnyezeten kívül felhasználtam néhány külső C++ könyvtárat a fejlesztéshez. Erre azért volt szükség, mert a C++ programozási nyelv alapból nem támogatja a mátrixműveleteket, sajátérték megoldókat, és a Fourier-transzformációt sem, amelyek szükséges műveletek voltak a modellben. Ezeknek manuálisan történő implementálása bonyolult és hosszadalmas lenne, ezért döntöttem a már létező könyvtárak használata mellett.

4.4.1 Eigen

Az Eigen [19] egy C++ könyvtár, amit lineáris algebrai számítások segítésére terveztek. Támogatja a mátrix- és vektorműveleteket, valamint többféle algoritmust is implementál. További előnye, hogy ingyenesen használható, és nincsenek egyéb függőségei a standard C++ könyvtáron kívül.

Az Eigen könyvtár hasznos volt a szintetizátor implementációjánál, mivel a modellben sok mátrix- és vektorműveletet kell végezni a rendszert leíró egyenletek megoldása során. Ezen felül további két fontos funkcióját használtam a könyvtárnak, ami elengedhetetlen volt a megoldáshoz.

Az egyik a mátrixok sajátértékeinek kiszámítására használható `GeneralizedSelfAdjointEigenSolver` osztály volt, amit a hanglap esetén a végeselem módszer alkalmazása után használtam a felépített keménység és tömegmátrixok sajátértékeinek és sajátvektorainak kiszámolására.

Egy másik hasznos funkciója volt a könyvtárnak, hogy képes lineáris egyenletek automatikus megoldására, amely a hanglap-rezonátor kölcsönhatás és a Newmark módszer alkalmazásánál kialakult egyenlet megoldásánál volt előnyös.

4.4.2 FFTW

A feladat megoldásához szükség volt még egy külső könyvtár használatára, ami az FFTW [20] volt. Az FFTW egy ingyenes C szubrutin könyvtár, amit diszkrét Fourier-transzformáció elvégzésére hoztak létre. Többféle módban működik lehetséges egy vagy több dimenzióban elvégezni, illetve valós-valós, valós-komplex és komplex-valós műveletként is elvégezni a diszkrét Fourier-transzformációt. Ezeken felül vannak még egyéb beállítási lehetőségek, például a művelet végrehajtása során mekkora pontossággal számoljon. Ez egy olyan megfontolás, amelyet felhasználónak kell eldöntenie attól függően, hogy melyik a nagyobb prioritás, a művelet gyorsasága, vagy a pontossága.

Az FFTW használatához egy `fftw_plan` típusú változót kell definiálni a kódban, amely leírja a Fourier-transzformáció konkrét típusát, beállításait, valamint a bemenet és a kimenetet tartalmazó tömböket és az elemszámot is meg kell adni. Ezt követően az `fftw_execute()` függvényhívással lehet végrehajtani a műveletet, az előbbi `plan`-t megadva paraméterként. Végül, ha már nincs szükség rá, az `fftw_destroy_plan()` függvénnyel törölni lehet a létrehozott `plan`-t.

A modell szempontjából a magasabb módusok esetében a gerjesztő erőre adott válasz kiszámolásához volt szükséges frekvenciatartományba transzformálni az erőfüggvényt, majd az eredményt inverz Fourier-transzformáció segítségével visszaalakítani időtartományba.

Ebből következően a frekvenciatartományba alakításhoz egydimenziós transzformációt alkalmaztam valósból komplex formába, a visszaalakításhoz pedig ugyanúgy egydimenziós, de komplexből valós irányba végeztem az inverz műveletet. Mindkét esetben az `FFTW_ESTIMATE` beállítást használtam.

4.5 A szintetizátor megvalósítása

A szükséges háttérismeretek, és fejlesztéshez használt szoftverek, fejlesztőkörnyezet ismertetése után rátérek az implementáció konkrét lépéseire, folyamatára. Ismertetni fogom a program felépítésére vonatkozó legfontosabb tulajdonságokat, valamint a grafikus felület megtervezésére is kitérek.

A Steinberg VST fejlesztőkörnyezet lényegében sok C++ nyelven íródott több osztály együttműködésével kialakított program. A legtöbb háttérfunkciót már elkészítették, amelyek a VST plugin működéséhez feltétlenül szükségesek, így már a

fejlesztéshez csak néhány osztályban kell implementálni a hang digitális feldolgozását, és a vezérlést megvalósító függvényeket. A kiindulásként használt projektben ezeket a részeket is elkészítették, nekem azonban jelentősen módosítanom kellett ahhoz, hogy a céljaimnak megfelelően tudjam implementálni a fizikai modellt.

4.5.1 Controller

A Controller osztály valósítja meg a szintetizátor vezérlését, valamint a felhasználói felületet is, azonban ebben az esetben a grafikus felület ki lett szervezve egy különálló egységként egy másik osztályban. A Controller osztályban sokféle dolgot be lehet állítani, ami a vezérléssel kapcsolatos, az implementáció során én a felhasználói felületen állítható paraméterek létrehozását és értékeinek meghatározását végeztem el.

Amikor a felhasználó a grafikus felületen megnyom egy gombot, vagy elteker egy potenciómétert, a programnak tudnia kell, hogy megváltozott egy paraméter értéke, és ennek megfelelően beállítani az értékét. Meg lehet adni továbbá, hogy milyen értékek között változzon a paraméter, és milyen mértékegységgel. Ez a felhasználónak ad információt a paraméter értékeiről, a program a valóságban egy 0 és 1 közötti számként értelmezi alapértelmezetten.

Egy konkrét példaként a programban felvettem egy „kParamPosition” nevű paramétert, amely a marimba hanglapjának hossza mentén leütött pozícióra vonatkozik. A paraméter értékeinek beállítása a következő kódrészlettel lehetséges:

```
Parameter* param = new RangeParameter(USTRING("Position"), kParamPosition,
USTRING("m"), 0, 0.4, 0.2);
param->setPrecision(2);
parameters.AddParameter(param);
```

A példában egy RangeParameter objektumot hozok létre, amely esetén a paraméter egy meghatározott intervallumban lévő értékeket vehet fel. Ebben a kódrészletben be van állítva, hogy a paraméter 0 és 0.4 közötti értékeket vehet fel, a kezdőértéke 0.2 és kettő tizedesjegy pontossággal jelezze ki az értékét. Ez a fajta megoldás potencióméterek esetén hasznos, így állítható az értéke egy megadott intervallumban.

4.5.2 UI

A grafikus felületet kezelő UI állomány a Controller osztállyal ellentétben nem csak egy osztályra vonatkozik, tulajdonképpen egy speciális Controller-t valósít meg, és

egyéb segéd osztályokat definiál a grafikus felhasználói felület kezelésének érdekében. Itt jön létre a program futtatása során a grafikus ablak egy leírófájl alapján (erről lesz szó részletesebben a grafikus felület elkészítésénél), valamint különböző vezérlők is elindulnak az egyes feladatok kezelésére, például a MidiLearnController, amely használatával MIDI segítségével is lehet vezérelni egyes paramétereket. Az előző alfejezetben ismertetett Controller osztálynak egy példányát is létrehozza, így az ott létrehozott paraméterek elérhetőek lesznek a felhasználói felületen.

A feladatom során nem változtattam nagy mértékben az UI kódján a kiindulásként használt Note Expression Synth projekthez képest, azonban néhány részt eltávolítottam belőle. Az eredeti forráskód viszonylag hosszú és bonyolult volt, így a számomra kevésbé fontos részeket eltávolítottam, illetve a grafikus zongorabillentyűket tartalmazó vezérlőt is kivettem a programból, mivel nem működött megfelelően az esetben.

4.5.3 Processor

A Processor osztály, ahogy a neve is utal rá, a feldolgozást valósítja meg. A feldolgozás során a plugin a bemenetek hatására digitális módon előállítja a kimeneti hangot, amit a kimeneten egy pufferbe tölt be. A Processor osztálynak külön paraméterei vannak, így a Controller-ben meghatározott paraméterek változásaira a saját paraméterek értékeit is frissíti. A Controller osztályhoz hasonlóan itt is ki van szervezve a feladatának egy része, ebben az esetben a feldolgozás konkrét megvalósítását egy Voice osztály végzi el. Ennek ellenére a Processor osztály is ellát fontos feladatokat, viszont inkább közvetítő szerepe van a be- és kimenetek, valamint a Voice osztály között.

Fontosabb függvényei és szerepük:

```
MarimbaSynthProcessor()
```

Ez a programomban a Processor osztály konstruktora, itt állítja be a paraméterek alapértelmezett értékeit. Ezek a paraméterek mind 0 és 1 közötti számok, egy intervallumot írnak le.

```
tresult PLUGIN_API MarimbaSynthProcessor::setActive(TBool state)
```

Ebben a függvényben hozza létre a feldolgozásra használt VoiceProcessor osztály példányát, amely létrehoz egy Voice objektumokból álló tömböt.

```
tresult PLUGIN_API MarimbaSynthProcessor::process(Vst::ProcessData& data)
```

A process függvény egyrészt lekezeli a paraméterek változását, másrészt a Voice osztály által létrehozott hangmintákat a kimenetre teszi, így lejátszható lesz az előállított hang.

4.5.4 Voice

Ez az osztály végzi a hangminták digitális feldolgozását és előállítását. A feladat szempontjából ez az osztály kapja a legnagyobb szerepet, itt történik a teljes fizikai modell implementálása. A következőkben bemutatom a Voice osztály legfontosabb függvényeit, majd a modell megvalósításának főbb részleteit.

4.5.4.1 Függvények

```
Voice()
```

A Voice osztály konstruktora, itt történik a paraméterek alapértelmezett értékeinek beállítása, illetve a konfigurációs fájlból történő beolvasása. A tömbök elemeit is a konstruktorban foglalja le a program, a legtöbb tömb a modellben használt függvények értékeit tárolja egy megadott mintavételi frekvencia szerinti időlépéssel. Ez akár több százezer elemű tömböket is eredményezhet.

```
~Voice()
```

A destruktornál történik a tömböknek lefoglalt memóriaterület felszabadítása.

```
bool process(SamplePrecision* outputBuffers[2], int32 numSamples)
```

A process függvényben a lejátszandó hangmintákat kimeneti pufferekbe tölti. A Processor osztály process függvénye is ezt a függvényt hívja meg, így kerül ki a hang a kimenetre.

```
void noteOn(int32 pitch, ParamValue velocity, float tuning, int32 sampleOffset, int32 nId) SMTG_OVERRIDE
```

A noteOn függvény a MIDI bemenet, például egy zongorabillentyű lenyomásának hatására fut le, ennek hatására lehet egy hangszer megszólaltatását szimulálni. Paraméterként több információ is rendelkezésre áll, például a hangmagasság vagy a leütés erőssége. Ezt a függvényt használtam a marimba esetén az ütővel való leütés szimulációjának elkezdésére, amelyet egy másik függvényben valósítottam meg.

```
void noteOff(ParamValue velocity, int32 sampleOffset) SMTG_OVERRIDE
```

Hasonlít a noteOn függvényhez, azzal a különbséggel, hogy itt a billentyű elengedése váltja ki ezt az eseményt. Ezzel lehet egyes hangszerek, például az orgona esetén a

lecsengést szimulálni. A marimba esetén nem használtam, mert ennél a hangszernél a lecsengést nem az ütő felengedése, hanem a megütés után a szabadrezgés csillapítása határozza meg.

```
void reset() SMTG_OVERRIDE
```

Egyes értékek, például a minták számlálójának alaphelyzetbe állítására használtam.

A következő függvények nem képezik a Voice osztály részét alapesetben, a modell implementálásának segítésére hoztam létre ezeket:

```
void calculateFrequencies()
```

A hanglap frekvenciáinak és módusalakjainak kiszámítására hoztam létre. Itt történt a végeselem-módszer implementálása. Ezt a függvényt a program indulásánál kell lefuttatni, valamint abban az esetben, ha a felhasználó változtatott a hanglap paraméterein.

```
void assemble_beam_matrices(unsigned int n, sparse_matrix_t& K,  
sparse_matrix_t& M, double* p_t1, double* p_t2)
```

A végeselem-módszer alkalmazásánál a mátrixok összerakására használom.

```
void beam_element_matrices(double l, elem_matrix_t& k, elem_matrix_t& m)
```

Az egyes elemek mátrixainak kialakításában segít a végeselem-módszernél.

```
void malletHit(int32 pitch, double velocity, int pos)
```

Az ütővel történő leütéstől a hang kialakulásáig zajló folyamat szimulációja történik ebben a függvényben. A noteOn esemény hatására hívja meg a program a szükséges paraméterek beállítása után.

4.5.4.2 Modell megvalósítása

A konkrét megvalósítás a 3. fejezetben ismertetett modell alapján történt, amely fizikai egyenletek segítségével leírja a hangszer megszólaltatásától a hang keletkezéséig lezajló folyamatot. A szoftveres implementációnál azonban néhány dolgot figyelembe kell venni a valós fizikai jelenségeket leíró rendszerekkel szemben. Egy szoftverben digitális értékekkel lehet számításokat végezni, a fizikai, matematikai függvények pedig sok esetben folytonosak, így valamekkora információveszteséggel jár ezek digitális reprezentálása. Megfelelő mintavételezési gyakorisággal az információveszteség elhanyagolható mértékűre csökkenthető, ennek segítségével az eredetihez nagyban hasonló függvénygörbéket lehet kapni. Ezt a szempontot modell alkotásánál figyelembe

vettem, a legtöbb alkalmazott módszer nem folytonos függvényeket használ, hanem diszkrét időpillanatokban lehet előállítani a függvényértékeket (például Newmark-módszer).

A megvalósítás a végeelem-módszerrel kezdődik, amivel a hanglap módusalakjait és a sajátfrekvenciáit számolom ki. Ezt a számítást még a leütés előtt érdemes elvégezni két szempontból is, egyrészt ez egy időigényes folyamat és lassítaná a hang kialakulásának szimulációját. A másik szempont, hogy a módusalakok és frekvenciák nem változnak minden egyes leütésnél, így felesleges lenne minden alkalommal újra kiszámolni. Abban az esetben kell csak újra elvégezni, ha a felhasználó változtatja a hanglap geometriájának, vagy a lehallgatási pozíció paramétereit.

A végeelem szimulációt a 3.1.2 fejezetben ismertetett módszer alapján végzi. A hanglap geometriáját meghatározó, és a felhasználó által konfigurálható fizikai paraméterek alapján a rudat 200 elemre felosztja, majd az egyes elemek merevség- és tömegmátrixainak előállítása után felépíti a teljes hanglapot jellemző mátrixokat, ezek után pedig az Eigen sajátérték megoldó függvényét használva kiszámolja a sajátértékeket, sajátvektorokat. A szimulációt lehetne 200-nál nagyobb elemre is elvégezni, azonban hatékonyság szempontból nem növeltem. Túl magas elemszám esetén nagyon időigényes lenne a számítás, túl kevés elemre viszont nem lehetne elég pontos eredményeket kapni (a 200-as felbontás már elég jónak bizonyult ahhoz, hogy az elemszám növelésével ne változzon jelentősen az első néhány, meghatározó sajátfrekvencia).

Érdemes megjegyezni, hogy a kiszámított sajátfrekvenciák esetén ezeknek csak az egymáshoz viszonyított arányait használom fel. A konkrét frekvenciákat egy tömbben tárolom el minden egyes MIDI hang esetén. Ebben az esetben 128 zenei hang esetén kiszámolom az alapfrekvenciát, majd a sajátfrekvenciák arányszámaival felszorozom, így megkapva az első 5 módus frekvenciáit mind a 128 MIDI hangra. Ez a valóságot ugyan nem tükrözi, mert valójában egy hanglap csak egyféle alapfrekvencián tudna rezegni, ugyanakkor így lehetségessé válik dallamok lejátszása a szintetizátoron. A valós megoldás az lenne, ha minden egyes zenei hangra egy-egy különböző geometriával rendelkező és megfelelően behangolt hanglapon elvégezni a végeelem szimulációt, azonban ez egy hosszadalmas számítási folyamat lenne, ezért használtam az említett egyszerűsített megoldást az implementáció során. Több különböző alakú hanglap nem jelentene plusz bonyolultságot az implementáció szempontjából, viszont lassabb lenne a plugin betöltése

Egy másik fontos művelet, a marimbarúd által lesugárzott hang kiszámításához (p_b) szükséges konstans együtthatók kiszámítása is ugyanitt zajlik, a végeelem szimulációval párhuzamosan. Ezt azért itt kell elvégezni, mert a hangnyomás kiszámításához szükséges Δp_i értékeit a hanglap összes elemére ki kell számolni. Hiányzik azonban a képletből a hanglap módusainak gyorsulása és a gyorsulás deriváltja, így ezen a helyen csak a konstans együtthatók értékei kaphatók meg.

Az eddig leírtak a **calculateFrequencies()** nevű függvényben történnek, a leütés és a hang keletkezésének szimulációja pedig a **malletHit()** függvényben fognak zajlani, aminek a következő a folyamata:

Az ütővel történő leütés szimulációjához először a hanglap-rezonátor rendszert gerjesztő erőfüggvény kiszámítására van szükség. Ezt a 3.2. fejezetben leírt módon a Newmark-módszerrel oldottam meg az ütő fejének összenyomódásából kiszámítva. A Newmark-módszert 3 másodperces intervallumban számoltam a DAW által meghatározott mintavételezési frekvencián, ami alapértelmezetten 44100 Hz. Ez viszonylag nagy mennyiségű számítást és adatot eredményez, a Newmark-módszer alkalmazása során az egyenleteket egy ciklusban 132300 alkalommal kell kiszámolni és az eredményként kapott függvényértékeket ugyanekkora elemszámú tömbben kell eltárolni. Az erőfüggvény esetén azonban nem szükséges a teljes időtartamra kiszámolni az értékeit, ugyanis az erő csak az interakció elején rövid ideig hat, utána konstans 0 lesz az értéke.

Az erőfüggvény előállításával párhuzamosan a hanglap-rezonátor kölcsönhatást is leíró egyenletek megoldása is elvégezhető a Newmark-módszerrel (3.3. fejezet), így azonos ciklus segítségével mindkét számítás eredménye megkapható. Az egyenletrendszer megoldásával kiszámolom a hanglap első módusának és a rezonátor választ az adott erőfüggvényre. Ezek segítségével a rezonátor és a hanglap első módusa által lesugárzott hangnyomás is kiszámolható.

Ezek után a lesugárzott hang meghatározása maradt hátra a hanglap magasabb módusai esetén. Ehhez először frekvenciatartományban kiszámolom a módusok választ a korábban kiszámolt gerjesztő erőre. Az erőfüggvényt a Fourier-transzformációval frekvenciatartományba alakítottam az FFTW könyvtár segítségével. Frekvenciatartományban kiszámoltam a módusok gyorsulás és a gyorsulás deriváltjának függvényeit, majd a két függvényt visszaalakítottam időtartományba inverz Fourier-

transzformációval. Így a korábban kiszámolt hangnyomás együtthatókkal és a most megkapott függvényekkel kiszámolható a lesugárzott hangnyomás a magasabb módusok esetén is.

A végső hangminták a korábban kiszámolt hangnyomás függvények összegeként állnak elő, illetve ezeket egy hangerőt szabályzó paraméter segítségével skálázom a megfelelő erősség elérése érdekében. Ezzel teljes a modell implementációja, a hangmintákat a kimeneti pufferbe betöltve le lehet játszani a hangszórón a szintetizátor használata során.

4.5.5 Grafikus felület elkészítése

A grafikus felület feladata az, hogy lehetőséget nyújtson a szintetizátor egyes paramétereinek beállítására, valamint néhány esetben visszajelzés megjelenítése bizonyos állapotváltozások esetén.



4.2. ábra: A VST szintetizátor grafikus felülete

A 4.2. ábrán látható az általam készített szintetizátor grafikus felülete. A felület tervezése során a kiindulásként használt Note Expression Synth projekt grafikus felületét módosítottam. A felület viszonylag egyszerű, néhány potenciómétert tartalmaz, amelyek egyes paraméterek állítására szolgálnak egy megadott intervallumon belül. A paraméterek konkrét funkcióit a következő fejezetben ismertetem.

A grafikus felület kinézetére és tartalmára vonatkozó információkat egy *.uidesc* kiterjesztésű leírófájl tartalmazza, ennek változtatásával lehet változtatni a plugin megjelenését. Létezik erre azonban egy kényelmesebb lehetőség is, a DAW segítségével a betöltött plugin esetén egy „Editor” üzemmódba lehet váltani, ahol a plugin kinézetét grafikus formában az egér és billentyűzet segítségével lehet kialakítani. Az eredmény megegyező lesz, ugyanúgy a leírófájl tartalmát lehet módosítani ezzel a módszerrel is, de jelentősen megkönnyíti a felület tervezésének folyamatát.

5 Konfigurálható paraméterek

A szintetizátor program készítésénél ügyeltem arra, hogy a modellt befolyásoló fizikai tényezők nagy része konfigurálható, vagy futás közben állítható legyen. Ennek az előnye, hogy így nem csak egy meghatározott esetre lesz működőképes a modell, többféle hanghatás is elérhető a szimuláció során. A paraméterek egy része a felhasználói felületen állítható a program futása közben a felhasználó által, míg a többi fizikai tényező értékét egy konfigurációs fájlban lehet megadni, amely a program betöltésekor beolvasásra kerül. Ha egy paraméter nincs beállítva a konfigurációs fájlban, akkor egy alapértelmezett értéket kap a kérdéses változó, így működőképes marad a program.

5.1 A felhasználói felületen beállítható értékek

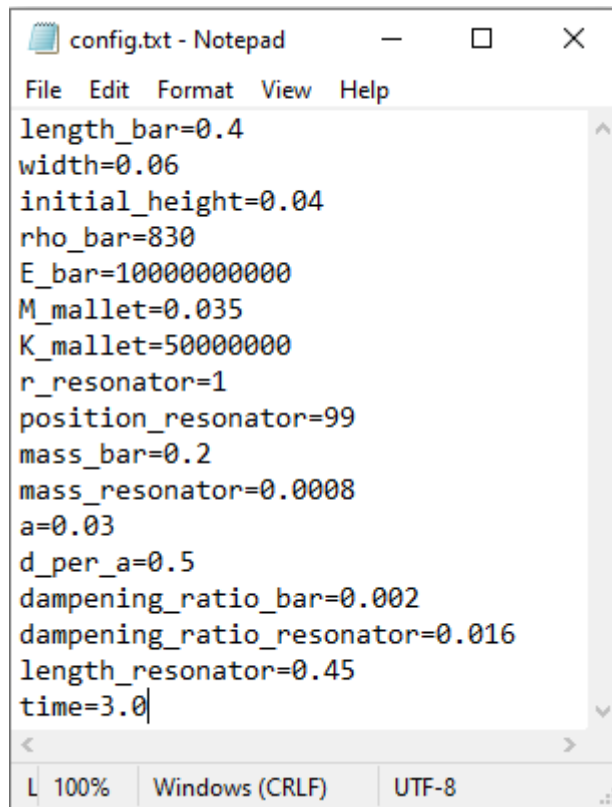
A felhasználói felületen található állítható paraméterek a következők:

- **Volume:** A hangerőt lehet vele szabályozni. Erősítő, valamint csillapító hatása is lehet. Alapértelmezett értéke 0, ilyenkor nem változtatja a hangerőt. Balra tekerve a potenciométert a negatív irányba a hangerő gyengébb lesz, egészen -100%-ig állítható. Jobbra tekerve a hangot erősíteni lehet, maximális értéke 100%.
- **Radius (Listening Position):** A lehallgatási pozíció sugara, amely a hanglaptól és a rezonátortól való távolságot jelenti, ahonnan a hangszer hangját hallgatjuk. 0 és 10 méter között változtatható, alapértelmezetten 1 méter a távolság.
- **Angle:** A lehallgatási pozíció szöge, ahol 0 radián a hanglappal egy síkban van horizontálisan, és egészen $\pi/2$ -ig állítható az értéke, ami alapesetben $\pi/6$.
- **Position:** A kalapáccsal való leütés pozíciója a hanglap hossza mentén. 0-tól a hanglap hosszáig terjedhet, alpból középén történik a leütés.
- **Tuning Ratio:** A rezonátor hangolásának aránya a hanglap alaphangfrekvenciájához képest. Kezdetben 0.9-szerese a hanglap frekvenciájának, de 0.7 és 1.3 között állítható.

- **Radius (Bar Tuning Cut):** A hanglap hangoló vajatál a vágás körívének sugara. 0 méter és a hanglap magasságának kétszerese között állítható.
- **Cut Start:** A bevágás kezdetének helye. A hanglap szélétől egészen a közepéig állítható, ahol 0% a szélét jelenti. A vágás szimmetrikus, így a rúd mindkét végétől ugyanolyan kezdési pozícióval indul a bevágás.
- **Cut Depth:** A bevágás mélysége. A bevágás mértékének maximumát határozza meg, 0 méter és a hanglap magassága között változtatható.

5.2 Konfigurációs fájl tartalma

A konfigurációs fájlban tetszőleges értékek beállíthatók az egyes fizikai tényezőknek, célszerű a valóságnak megfelelő értékek beállítása. A fájlban a paraméterek külön sorban szerepelhetnek, valamint a következő formában kell megadni őket: <paraméter neve>=<paraméter értéke>



```

config.txt - Notepad
File Edit Format View Help
length_bar=0.4
width=0.06
initial_height=0.04
rho_bar=830
E_bar=10000000000
M_mallet=0.035
K_mallet=50000000
r_resonator=1
position_resonator=99
mass_bar=0.2
mass_resonator=0.0008
a=0.03
d_per_a=0.5
dampening_ratio_bar=0.002
dampening_ratio_resonator=0.016
length_resonator=0.45
time=3.0
L 100% Windows (CRLF) UTF-8

```

5.1. ábra: Konfigurációs fájl tartalma

A megadható paraméterek, és mértékegységeik a következők:

- **length_bar:** A hanglap hossza (m)

- **width:** A hanglap szélessége (m)
- **initial_height:** A hanglap kezdő magassága/vastagsága (m)
- **rho_bar:** A hanglap anyagának sűrűsége (kg/m^3)
- **E_bar:** Young-modulus (Pa)
- **M_mallet:** Az ütő fejének tömege (Kg)
- **K_mallet:** Az ütő fejének keménysége ($\text{Nm}^{(-3/2)}$)
- **position_resonator:** A rezonátor pozíciója a hanglap hosszhoz képest (0 és 200 között)
- **mass_bar:** A hanglap tömege (kg)
- **mass_resonator:** A rezonátorban lévő levegőoszlop tömege (kg)
- **a:** dimenzió nélküli távolság (m)
- **d_per_a:** dimenzió nélküli hányados (0 és 1 között)
- **dampening_ratio_bar:** Csillapítási tényező a hanglapnál (%)
- **dampening_ratio_resonator:** Csillapítási tényező a rezonátornál (%)
- **length_resonator:** A rezonátorcső hossza (m)
- **time:** Az előállítandó hang időintervalluma (s)

6 Összegzés

A munkám során egy szintetizátor szoftvert valósítottam meg, amely egy létező hangszer, a marimba hangképzésén alapult. A feladat megvalósításához részletesen megismerkedtem a hangszer fizikai tulajdonságaival, majd hangjának mesterséges módon történő előállításához egy modellt készítettem el. A kész modell felhasználásával sikerült elkészítenem a hangzás szoftveres szimulációját is, így végeredményként egy zenei vagy tudományos felhasználásra alkalmas VST plugin-t hoztam létre.

6.1 Értékelés

Az elkészült VST szintetizátor egy DAW szoftver segítségével futtatható és tesztelhető. Induláskor bizonyos fizikai paramétereket egy konfigurációs fájlból beolvas, amelyeket a felhasználó tetszőlegesen megszabhat az indítás előtt. A szintetizátor futás közben is vezérelhető, valamint néhány paraméter érték egy megadott intervallumon belül állítható a grafikus felületen található potenciométerek segítségével, amelyek egér vagy billentyűzet segítségével vezérelhetők. A hangok leütése MIDI segítségével irányítható, ehhez egy MIDI vezérlőre van szükség, ami sok formában megtalálható, az operációs rendszerekben általában található valamilyen szoftveres implementáció, amit szintén egérrel és billentyűzettel lehet használni. A bejövő MIDI jeleket a szintetizátor értelmezni tudja, így például egy billentyű leütése esetén megkapható a billentyűhöz rendelt hangmagasság és a leütés sebessége. Ezek felhasználásával a program elvégzi a marimba megszólaltatásának szimulációját, és az elkészült hangmintákat a kimeneten visszaadja.

Összességében a kész plugin jól modellezi a marimba hangkeltését, ehhez hasonló, pontosabban ilyen részletes szimulációt végző szoftver létezéséről eddig nem szereztem tudomást. A szintetizátor hangja megfelelő beállításokkal nagyban hasonlít az eredeti hangszerhez. Bizonyos funkciókban többet is nyújt a valós marimbánál, például az eredeti hangszer esetén nem lehet könnyedén állítani az egyes komponensek anyagára és geometriájára vonatkozó fizikai tényezőket.

6.2 Továbbfejlesztési lehetőségek

A jelenlegi modell a hangszer egyes elemei, illetve ezek kölcsönhatása esetén bizonyos egyszerűsítésekkel vagy feltételezések mellett valósult meg. Természetesen a

legtöbb modell esetén alkalmaznak valamilyen egyszerűsítést, de lehet egy továbbfejlesztési lehetőségnek tekinteni a modell pontosabbá tételét, vagy a számításigény szempontjából hatékonyabb módszerek használatát.

Néhány további fejlesztési lehetőség az implementált szoftverben található. A mostani megvalósításban a hangmintát gyakorlatilag a leütés hatására előre elkészíti a plugin, majd ezt a hangmintát abban az ütemben tölti be a kimeneti pufferbe, ahogy a DAW szoftver ezeket a mintákat a lejátszáshoz elkéri a plugin-tól. Ezt a folyamatot lehetne gyorsítani azzal, ha nem előre készítenénk el a hangmintákat, hanem abban az ütemben, ahogy szükség van rá. Ennek megvalósítására a Newmark-módszer alkalmas, hiszen egyforma mintavételezési frekvenciával gyártja az egyes mintákat. A magasabb módusok hangját viszont frekvenciatartományban számoltam, ami nem alkalmazható a minták egyesével történő előállítására, így a jövőben más módszert kell alkalmazni ennek a továbbfejlesztésnek a megvalósításához.

6.3 Köszönetnyilvánítás

Szeretnék köszönetet mondani témavezetőmnek és egyben konzulensemnek, Dr. Rucz Péternek, a modell tervezésében és a szoftver megvalósításában nyújtott segítségéért, valamint a hasznos meglátásaiért és tanácsaiért, folyamatos támogatásáért, amivel jelentősen hozzájárult a munkámhoz.

Irodalomjegyzék

- [1] Yamaha: *Anatomy of a marimba*,
<https://hub.yamaha.com/drums/percussion/anatomy-of-a-marimba/>, Utolsó hozzáférés: 2022.11.23.
- [2] Vienna Symphonic Library: *Marimba*,
<https://www.vsl.info/en/academy/percussion/marimba>, Utolsó hozzáférés: 2022.11.23.
- [3] Yamaha: *How a Marimba is Made Tuning the tone plates*,
https://www.yamaha.com/en/musical_instrument_guide/marimba/manufacturing/manufacturing002.html, Utolsó hozzáférés: 2022.11.23.
- [4] Yamaha: *The Structure of the Marimba Inside and outside the resonator pipes*
https://www.yamaha.com/en/musical_instrument_guide/marimba/mechanism/mechanism002.html, Utolsó hozzáférés: 2022.11.24.
- [5] Modern Drummer: *What You Need to Know About... Mallets*,
<https://www.moderndrummer.com/2015/02/need-know-mallets-percussion/>,
Utolsó hozzáférés: 2022.11.24.
- [6] Antoine Chaigne and Vincent Doutaut, *Numerical simulations of xylophones. I. Time-domain modeling of the vibrating bars*, J. Acoust. Soc. Am. 101 (1), January 1997: 539-557, <https://doi.org/10.1121/1.418117>
- [7] Fiala Péter: *A hangszerek fizikája - jegyzet*,
<https://last.hit.bme.hu/sites/default/files/documents/hangfiz.pdf>, Utolsó hozzáférés: 2022.11.30.
- [8] Ingolf Bork, *Measuring the Acoustical Properties of Mallets*, Applied Acoustics 30 (1990): 207-218, [https://doi.org/10.1016/0003-682X\(90\)90044-U](https://doi.org/10.1016/0003-682X(90)90044-U)
- [9] Newmark Nathan M. (1959), "A method of computation for structural dynamics", Journal of the Engineering Mechanics Division, 85 (EM3) (3): 67–94,
<https://doi.org/10.1061/JMCEA3.0000098>
- [10] Peter Rucz, Mihaly Adam Ulveczki, Judit Angster, and Andras Miklos, *Simulation of mallet percussion instruments by a coupled modal vibroacoustic finite element model*, J. Acoust. Soc. Am. 149 (5), May 2021: 3200-3212,
<https://doi.org/10.1121/10.0004216>
- [11] Ingolf Bork, *Practical Tuning of Xylophone Bars and Resonators*, Applied Acoustics 46 (1995): 103-127, [https://doi.org/10.1016/0003-682X\(95\)93953-F](https://doi.org/10.1016/0003-682X(95)93953-F)
- [12] Filipe Soares, Jose Antunes, Vincent Debut, *A minimal model of the bar-resonator vibroacoustic interaction in mallet percussion instruments*, Applied Acoustics Volume 199, October 2022, 109049: 1-13,
<https://doi.org/10.1016/j.apacoust.2022.109049>

- [13] Vincent Doutaut, Denis Matignon, Antoine Chaigne, *Numerical simulations of xylophones. II. Time-domain modeling of the resonator and of the radiated sound pressure*, J. Acoust. Soc. Am. 104 (3), Pt. 1, September 1998: 1633-1647, <https://doi.org/10.1121/1.424376>
- [14] Wikipedia, *Virtual Studio Technology*, https://en.wikipedia.org/wiki/Virtual_Studio_Technology, Utolsó hozzáférés: 2022.11.30.
- [15] Wikipedia, *Digital Audio Workstation*, https://en.wikipedia.org/wiki/Digital_audio_workstation, Utolsó hozzáférés: 2022.12.01.
- [16] Image Line, *FL Studio*, <https://www.image-line.com/fl-studio/>, Utolsó hozzáférés: 2022.12.01.
- [17] Steinberg, *VST 3 SDK*, <https://www.steinberg.net/developers/>, Utolsó hozzáférés: 2022.12.01.
- [18] CMake, *CMake*, <https://cmake.org/>, Utolsó hozzáférés: 2022.12.01.
- [19] Eigen, *Eigen*, https://eigen.tuxfamily.org/index.php?title=Main_Page, Utolsó hozzáférés: 2022.12.02.
- [20] FFTW, *FFTW*, <https://www.fftw.org/>, Utolsó hozzáférés: 2022.12.02.

Függelék

Az elkészült szintetizátor tesztelése során felvett minta-hangfájlok:

- **hangolasi_arany.wav** - A rezonátor hangolási arányának változtatása azonos hangmagasság mellett
- **hangmagassag.wav** - Egy hangsor bemutatása
- **lehallgatasi_szog.wav** - Lehallgatási pozíció szögének változtatása azonos hangmagasság mellett
- **pozicio.wav** - A leütés pozíciójának változtatása azonos hangmagasság mellett