



Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Méréstechnika és

Nika Márton Péter

BASSZUSGITÁR HANGJÁNAK FIZIKAI ALAPÚ SZINTÉZISE

KONZULENS

Dr. Bank Balázs

BUDAPEST, 2014

SZAKDOLGOZAT-FELADAT

Nika Márton Péter (G8R1XP)

szigorló villamosmérnök hallgató részére

Basszusgitár hangjának fizikai alapú szintézise

A hallgató feladata egy fizikai modellen alapuló, valós időben futtatható basszusgitár-szintetizátor megvalósítása. A fizikai modellek jellegzetessége, hogy azok a hangszer mechanikai kialakításából, fizikai tulajdonságaiból adódó nemlineáris tulajdonságokat is képesek figyelembe venni. Ilyen nemlineáris viselkedés basszusgitár esetén a "slapping", amikor a zenész a húrokat a gitár bundjaihoz csapja. Ez esetben a húr rezgését a bundok korlátozzák, ez pedig a hangban megnövelt felharmónikustartalomként jelentkezik.

A basszusgitár-szintetizátor legfontosabb eleme a húrmodell. Húrok modellezésére az irodalom több megoldást is bemutat. Valós idejű szintézis esetén ezek közül leginkább a digitális hullámvezető, ill. a modális szintézis jöhet szóba, kérdés azonban, hogy a "slap bass" technika melyikkel modellezhető hatékonyabban. Az ez alapján választott húrmodellt természetesen ki kell egészíteni a pengetés és a pickup megfelelő modelljeivel.

A fizikai alapú húrmodellek a hagyományos, pl. felvett minták visszajátszásán alapuló módszerekhez képest jelentősen több számítást igényelnek, azonban ez a személyi számítógépek kellően nagy számítási kapacitásának köszönhetően mára már nem okoz gondot. Szoftveres megvalósítás esetén a Steinberg cég VST környezetét érdemes használni. Ennek előnye, hogy így az elkészített plugin szinte bármely zeneszerkesztő programmal kompatibilis, ill. a pluginként történő megvalósítás leveszi a fejlesztő válláról a hangkártyakezelés nehézségeit is.

A hallgató munkájának a következőkre kell kiterjednie:

- Tekintse át a különböző húrmodellezési eljárások irodalmát!
- Valósítsa meg Matlabban mind digitális hullámvezetőn, mind pedig modális szintézisen alapuló húrmodellt, és vizsgálja meg, melyik milyen hatékonysággal képes a húr bundoknak csapódását (slap bass technika) modellezni!
- A megvalósított gerjesztésmodell a pengetés és a hüvelykujjal történő megütés lehetőségét is tartalmazza!
- A pickup modelljét az irodalom alapján valósítsa meg oly módon, hogy a pickup nemlineáris karakterisztikáját is vegye figyelembe!
- A teljes hangszermodellt VST pluginként valósítsa meg, és adjon lehetőséget a hangszer főbb paramétereinek megváltoztatására!

Tanszéki konzulens: Dr. Bank Balázs, docens

Budapest, 2014. március 12.

.....
Dr. Jobbágy Ákos
tanszékvezető

Tartalomjegyzék

Összefoglaló	6
Abstract.....	7
1 Bevezetés	8
2 Hangszintézis módszerek	10
2.1 Absztrakt algoritmusok.....	10
2.2 Előre felvett minták feldolgozása	10
2.3 Spektrális modellek.....	11
2.4 Fizikai alapú szintézis	11
3 A húr modellezése	14
3.1 A véges differenciák módszere	15
3.2 A digitális waveguide modell	15
3.3 A modális szintézis	16
3.4 A slap bass játékmód jellegzetességei	18
3.5 A gerjesztés modellezése	26
3.5.1 Pengető modellezése.....	26
3.5.2 Slap bass modellezése.....	30
4 A pickup modellezése.....	35
4.1 A mágneses pickupok tulajdonságai	36
4.2 A hangszedő helyzete és szélessége	38
4.3 A hangszedő nemlinearitása	40
4.4 A hangszedő deriváló hatása.....	40
4.5 A hangszedő áramkörének átvitele	41
5 Analízis.....	42
5.1 Mérések.....	44
5.1.1 Hangok felvétele	44
5.1.2 A módusfrekvenciák megkeresése.....	45
5.1.3 A lecsengési idők kiszámítása	46
5.2 A gerjesztési paraméterek meghatározása	48
5.2.1 Pengetővel való gerjesztés	48
5.2.2 Slap bass gerjesztés.....	49
5.3 A kinyert adatok implementációja	49

6 VST Implementáció.....	51
6.1 VST alapok.....	51
6.2 A MIDI kezelés	52
6.3 A szintetizátor megvalósítása.....	53
6.3.1 A változók	53
6.3.2 A függvények	53
6.3.3 Felhasználói paraméterek	56
6.4 Hibák	57
7 Értékelés, továbbfejlesztési lehetőségek	59
7.1 Értékelés	59
7.2 Továbbfejlesztési lehetőségek.....	60
Irodalomjegyzék	61

HALLGATÓI NYILATKOZAT

Alulírott **Nika Márton Péter**, 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, 2014. 05. 22.

.....
Nika Márton Péter

Összefoglaló

Az első szintetizátorok megjelenése óta foglalkoztatja a felhasználókat a szintetikus hangok előállítása mellett egyéb hangszerek, pl. orgonák, zongorák, gitárok hangjának minél jobb minőségű modellezése is. A hanggenerálás leggyakrabban előre eltárolt minták lejátszásában, jelalakformálásában és különböző szűrésekben merül ki, azonban a virtuális szintetizátorok elterjedésével és a processzorok számítási kapacitásának növekedésével manapság egyre nagyobb teret kapnak a magasabb számítási igényű, fizikai alapokon nyugvó szintézismódszerek.

Mára már sok fizikai alapú elektromos és akusztikus gitármodell érhető el akár ingyen is, azonban a basszusgitár által lefedett hangtartományra ezek általában nem használhatók megfelelően, elsősorban az elektromos gitárok magasabb, melegebb tónusa miatt. Emellett basszusgitárt modellező programokból jóval kevesebb lelhető fel.

További problémát jelent, hogy a basszusgitár hangtartományában is kellemes hangzású gitárszintetizátorok nincsenek felkészítve a kifejezetten basszusgitáros technikákra, pl. a slap bass játéktílusra, amikor a zenész a húrokat „megcsapja” a hüvelykujjával, vagy a magasabb húrokat „feltépi”, ezzel létrehozva a más hangszereknél nem felfedezhető, egyéni hangzást.

A húr fizikai alapú modellezésére többféle bevált módszer létezik, ezek egyike a modális szintézis. E szakdolgozat keretein belül a húrt ezzel a módszerrel modellezem, majd a hangszer többi részét a fellelhető irodalom alapján implementálom. Céлом a saját basszusgitárom hangjának fizikai alapú modellezése és valósidejű futtatása VST (Virtual Studio Technology) környezetben, különös hangsúlyt fektetve a slap bass játéktílusra. A VST a Steinberg cég által kifejlesztett de facto szabvány környezet, ami lehetővé teszi virtuális szintetizátorok és ún. digitális audió munkaállomások, DAW-ok kapcsolatát. A plugint egy ilyen munkaállomásba betöltve egy külső MIDI vezérlővel – pl. billentyűzettel – valós időben lehet vezérelni.

Abstract

Since the first synthesizers came out, users have wanted – besides creating new, synthetic sounds – to accurately generate the sounds of other instruments, such as organs, pianos, or guitars. The most often used method is playing back pre-recorded samples from an onboard memory, while shaping the waveforms and filtering the output. The spread of virtual synthesizers and the increasing computing power of CPUs have allowed more demanding, physics based synthesis methods to gain ground.

Nowadays many physics based electric and acoustic guitar models are available even for free, but usually these cannot be used in the sounding range of the bass guitar, mainly due to the warmer tone of electric guitars. In addition, there are far less synthesizers modeling the bass guitar.

Another problem is that the guitar synthesizers that sound good even in the bass range are not prepared for special bass guitar playing techniques, like the slap bass style, where the musician either slaps the lower strings or pops the higher ones, creating a unique sound not to be found in any other instrument.

Numerous methods have been introduced to model the vibrating string from a physical aspect, one of these is the modal based approach, which I use in this thesis. After modeling the string, the other parts of the instrument are implemented by methods found in the available literature. My objective is to create a physics-based model of my own bass guitar and to use it as a VST (Virtual Studio Technology) plugin in real time, focusing on the modeling of the slap bass style. VST is a de facto standard environment developed by Steinberg that allows DAWs (Digital Audio Workstations) to access virtual synthesizers. After loading the plugin in one of these workstations, the user is able to control the synthesizer with an external MIDI controller, e.g. a keyboard, in real time.

1 Bevezetés

A basszusgitár viszonylag újkori hangszer, a jelenleg ismert formáját az 1930-as években fejlesztették ki. Elődjének a nagybögő tekinthető (szokás is a basszusgitárt egyszerűen bögőnek nevezni), azonban ezt inkább vonós hangszerként ismerték a zenészek-zeneszerzők, bár lehet pengetve is játszani rajta.

Az elektromos basszusgitár egyértelműen az elektromos gitár mintájára készült, leggyakrabban a nagybögőknél és csellóknál megszokott négy húrral rendelkezik, de népszerűek az öthúros modellek is. Léteznek akár 10-12 húros verziók is, ezek vagy 2-3 húros csoportokban vannak egymáshoz képest oktávra hangolva, vagy pedig hatalmas fogólapú, gitárnak már nehezen nevezhető, külön technikát igénylő hangszerek. A leggyakoribb azonban a mai napig a négyhúros verzió.

Az elektromos basszusgitár nem annyira népszerű, mint hathúros testvére, ennek ellenére gyakorlatilag az összes zenei stílus rendkívül fontos tagja: a dobbal, vagy egyéb ütőhangszerekkel karöltve szolgáltatja a ritmust és az alapot a többi zenész számára. Gyakori mondás, hogy a basszusgitár hangját nem hallani a zenében, de az rögtön észrevehető, ha nincs ott.

Általában háromféle pengetési módszert használnak a basszusgitárosok: a leggyakrabban ujjal pengetik meg a húrokat, ezzel szinte bármilyen zenei stílushoz megfelelő hangzást lehet létrehozni. Keményebb rockzenékben gyakori a pengetővel való játék, ami a hangszernek karakteresebb, fémesebb hangzásvilágot kölcsönöz. Végül a legtechnikásabb, legnagyobb ügyességet igénylő módszer az ún. slap bass, amikor a zenész a mélyebb húrokat a hangszertest felé üti, a magasabbakat a testhez képest felfelé tépi. Ez a játéktípus könnyen felismerhető, a tépések átütő, fémes hangzást hoznak létre, míg a csapások által gerjesztett hanghullámokat a hallgató inkább a testében érzi, mint hallja. Az elsősorban szólóikról ismert, híres basszusgitárosok általában ezzel a technikával tűnnek ki a tömegből.

A számítástechnika térnyerésével és a számítási kapacitás növekedésével együtt jár, hogy a felhasználók részéről mind nagyobb igény jelentkezik szintetizátorszoftverekre. Míg 10-15 éve csak méregdrága stúdióknak volt megfelelő felszerelésük jó minőségű felvételek készítésére, manapság már bármilyen asztali PC alkalmas lehet ilyen célokra, elég egy jó minőségű hangkártya, egy DAW (Digital Audio

Workstation) szoftver és néhány szoftveres szintetizátor ahhoz, hogy bárki zenét szerezhessen. További jó hír az amatőrök számára, hogy egyre több ingyenes, vagy viszonylag olcsó termék elérhető mind DAW-ból, mind szintetizátorból és pluginekből, amik akár méregdrága, híres szoftverek tudását, hangminőségét is elérhetik.

A szakdolgozat célja egy valós időben jól működő, adott hangszer és gerjesztés fizikáját modellező basszusgitar-szintetizátor létrehozása. A megvalósított program modális szintézis alkalmazásával, mérési eredményekre, illetve a gerjesztés és a hangszedők fizikai tulajdonságaira alapozva állítja elő a hangot, mindezt VST környezetben, ami segítségével bármilyen, azt támogató DAW-ban felhasználható lesz.

A dolgozat következő fejezetében bemutatom az alapvető hangszintézis módszereket, majd ezután a 3. fejezet témája a húr fizikáján alapuló modell létrehozása és a slap bass technika különlegességeinek feltérképezése, számítása. A 4. fejezet a hangszedők tulajdonságait vizsgálja, végül a modellezendő tulajdonságok analízisét és az előző fejezetek eredményeinek implementációját ismerhetjük meg.

2 Hangszintézis módszerek

Az első szintetizátor az 1897-ben bejegyzett Telharmonium volt [Wikipedia 1], ami még elektromechanikus elven működött. Azóta hatalmas fejlődésen mentek keresztül mind az analóg, mind a digitális szintetizátorok, a fő feladatuk mégis ugyanaz maradt: a felhasználó számára szükséges hangok létrehozása és lejátszása.

Sokféle különböző hangszintézis módszer létezik, melyek nagyban eltérhetnek egymástól a modellezendő tulajdonságban és a modellezés tartományában is. Ezen módszerek egy alapvető csoportosítása [Bank 2000] és [Smith 1991] alapján:

2.1 Absztrakt algoritmusok

Az absztrakt algoritmusok lényege, hogy egyszerű matematikai műveletek segítségével szintetizálnak hangot. A módszer előnyei közé tartozik az egyszerűsége, általánossága és a viszonylag kevés, vezérléshez szükséges paraméter, azonban különböző, az analízis során felmerülő nehézségek miatt valós hangszerek hangját közel lehetetlen előállítani a módszerrel, inkább újfajta hangok, „zajok” létrehozásához hasznosak.

Az FM (frekvenciamodulációs) hangszintézist Chowning [1973] fejlesztette ki. Alapgondolata szerint a rádióátvitelhez hasonló modulációval, azonban azonos nagyságrendbe eső vivő- és modulálófrekvenciák használatával adott frekvenciájú, akár inharmonicitással is rendelkező hangot lehet előállítani.

A waveshaping kifejllesztői Le Brun [1979] és Arfib [1979] voltak. Módszere egyszerű: egyetlen bejövő jel nemlineáris torzításával felharmonikusokban gazdag jelet kaphatunk. Ha ez a bemeneti jel szinuszos, a létrejövő harmonikusok amplitúdói a Chebyshev polinomokkal számolhatók. Ezzel a módszerrel a magasabb harmonikusoknál jelentkező inharmonicitásokat nem tudjuk modellezni.

2.2 Előre felvett minták feldolgozása

A manapság leggyakrabban használt módszer. A szintetizátor által létrehozandó hangokat jó minőségben felveszik, eltárolják, majd a szintetizátor a hangot burkológörbékkel és szűrőkkel dolgozza fel a játék függvényében. Elvéből adódóan bármilyen hangszer hangját nagyon pontosan adja vissza, azonban kizárólag azt az

egyetlen, felvételnél használt modell hangját játssza le és a felhasználó által módosítható paraméterek száma igen szűkös. Alapvető szűrőkkel, burkológörbékkel módosítható a jel, de komplexebb tulajdonságok, pl. gitáron a pengetés helye, vagy vonós hangszernél a vonó dinamikája nem állíthatók.

A szintetizálendő hangot az analízis során általában három részre, a hang kezdetére, egy rövid, ismétlődő mintára, illetve a hang végi lecsengésre bontják, majd ezeket tárolják el. A középső szakaszt addig ismételteti (akár egyre csökkenő amplitúdóval) a szintetizátor, amíg azt vezérlik, így egyszerűen lehet memóriát spórolni.

Ez a módszer számításigénnyel alig rendelkezik, manapság a háttértárak is kellően olcsók, emiatt belépő szinten népszerű, a felsorolt hibái ellenére is.

2.3 Spektrális modellek

A fentebb felsorolt technikák mind időtartományban vizsgálják a hangot, a spektrális modellek – nevükből adódóan – a frekvenciatartomány felől közelítik azt. Ezzel a megközelítéssel figyelembe lehet venni az emberi hallásérzékelést és különböző pszichoakusztikus hatásokat is, de általában sok paraméter szükséges egy hangszer megfelelő leírásához, illetve a tranziensek szimulációja is problémás [Bank 2000].

Az additív szintézis a legegyszerűbb spektrális megoldás, egyszerűen különböző frekvenciájú és burkolójú szinuszok összeadásából állítja elő a hangot, amit ezek után különböző zajmodellekkel és tranziensekkel lehet kiegészíteni. Ez a módszer pl. orgonahang előállítására nagyon jól használható. A témáról bővebb információkat tartalmaz [Márkus 1999].

2.4 Fizikai alapú szintézis

A fizikai modellezés nem a létrehozandó hang vizsgálatából indul ki, hanem az azt létrehozó hangszer rezgéseit, csillapításait, gerjesztését veszi figyelembe. Mivel a jel helyett a hang forrását írja le, elméletileg jóval valósághűbb modellt tud létrehozni, hiszen az állítható paraméterei a hangszer fizikai tulajdonságából adódnak.

A modellezés három lépésből áll: először a hangszer fizikáját írjuk le, tipikusan differenciálegyenletekkel, majd ezután az egyenletek numerikus megoldása, illetve futtatható algoritmus létrehozása következik. Mivel ezek igen számításigényes műveletek, harmadik lépésként beiktatnak pszichoakusztikai egyszerűsítéseket, ami

egyszerűen azt jelenti, hogy az emberi fül számára érzékelhetetlen változásokat kihagyjuk a számításból.

Ezekkel az egyszerűsítésekkel is megmarad a fizikai alapú modellezés legnagyobb hátránya: a nagy számításigény. Az elmúlt 5-10 évben azonban elért arra a szintre a végfelhasználói szintű számítástechnika is, hogy komolyabb számítási kapacitást igénylő modellek is valós időben futtathatók legyenek bárki által elérhető processzorokon, ezért várhatóan a jövőben egyre több ilyen szintetizátor fog megjelenni.

Húros hangszerek esetén a húr mozgásának leírására három fő megoldás létezik:

Az első fizikai alapú modell a húr differenciálegyenletének diszkrétidejű megoldásán alapul, ezt hívják véges-differenciás modellnek [Hiller és Ruiz 1971]. Az egyenletekben közvetlenül szerepelnek a paraméterek, emiatt jól átlátható módszer, viszont egyben a leginkább számításigényes is, különösen többdimenziós esetben.

A leggyakrabban használt, legkisebb számításigényű módszer az ún. digitális hullámvezetős, azaz waveguide modellezés [Smith 1983, 1987, 1992]. A módszer a hullámegyenlet egydimenziós diszkrétidejű megoldásán alapul, a húr pontjait késleltetővonalakkal köti össze, majd az összes veszteséget egyetlen szűrővel veszi figyelembe, ezzel különösen DSP-ken lehet hatékonyan implementálni a modellt.

A modális szintézist Adrien [1991] egymódusú, lecsengő rendszerek lineáris kombinációjaként írja le. Legfőbb előnye, hogy minden rezgő rendszert általánosan le lehet írni ugyanazzal a formulával. Hátrányai közé tartozik, hogy bár a paraméterei fizikai tulajdonsággal bírnak, azok nem a közvetlenül látható paraméterek (pl. a húr hossza), hanem a felhasználó számára keveset mondó tulajdonságok, mint a húr tömegsűrűsége. Ez a módszer implementációjában hasonlít az additív szintézishez, azonban ebben az esetben csak a húrt modellezzük harmonikusok szuperpozíciójaként, a gerjesztést és a további szűrőket külön-külön implementáljuk [Bank 2000].

Másik megközelítés a Trautmann és Rabenstein [2002] által kidolgozott Functional Transformation Method (FTM). A módszer szintén a rezgő húr elhajlását leíró parciális differenciálegyenletből indul ki, a húr egyenletét időtartománybeli transzformációkkal oldja meg, így valamivel bonyolultabb rendszereket leírására is alkalmazható.

A modális szintézis legfőbb hátránya, hogy csak olyan, egyszerű geometriákra tudjuk alkalmazni, amiknek létezik analitikus megoldása. Ilyenek pl. a húr, illetve a kör, vagy négyzet alakú kétdimenziós membrán egyenletei.

A megvalósított programomban végül modális szintézist használok. A módszer kiválasztásánál közrejátszott, hogy a szükséges paraméterek megfelelő mérések elvégzése után könnyen előállíthatók, finomhangolhatók. Ezeket a módszereket, a hullámegyenletek megoldásait, illetve a rendszer implementációját a következő fejezetekben részletesen elemzem.

3 A húr modellezése

Mint azt az előző fejezetben említettem, a fizikai alapú hangszintézis húros hangszer esetén a húr hullámegyenletéből indul ki, ami [Morse és Ingard 1968] alapján:

$$\mu \frac{\partial^2 y}{\partial t^2} = T_0 \frac{\partial^2 y}{\partial x^2} - ES\kappa^2 \frac{\partial^4 y}{\partial x^4} - 2R(\omega)\mu \frac{\partial y}{\partial t} + d_y(x, t), \quad (3.1)$$

ahol:

μ a húr egység hosszra eső tömege

T_0 a húr megfeszítettsége

$ES\kappa^2$ a húr merevségét leíró tényező

$R(\omega)$ a frekvenciafüggő súrlódási ellenállás, ami a húr veszteségeit írja le

$d_y(x, t)$ a külső gerjesztésből adódó erő

Gitárhúr modellezésekor első közelítésben a fenti egyenletből elhagyhatjuk a húr merevségével és a veszteségével kapcsolatos tagokat, illetve egy kezdeti húralak megadása után szabadon hagyva a rendszert zérus külső gerjesztést feltételezhetünk. Ekkor az egyenlet az alábbira egyszerűsödik:

$$\mu \frac{\partial^2 y}{\partial t^2} = T_0 \frac{\partial^2 y}{\partial x^2} \quad (3.2)$$

Ezt tekintjük az ideális húr egyenletének. Tudjuk még továbbá, hogy a húrban terjedő transzverzális hullám sebessége:

$$c_t = \sqrt{\frac{T_0}{\mu}} \quad (3.3)$$

Így a (3.2) és (3.3) egyenletek összevonásából megkapjuk az ideális húr differenciálegyenletét:

$$\frac{\partial^2 y}{\partial t^2} = c_t^2 \frac{\partial^2 y}{\partial x^2} \quad (3.4)$$

A differenciálegyenlet megoldására és diszkretizálására az előző fejezetben említett három módszert alkalmazzák, illetve megemlítenő még a húr tömeg-rugó modellje is, amiről belátható, hogy a véges-differenciás modellel azonos eredményre

vezet. Ez utóbbi elméletéről e dolgozat keretében bővebben nem esik szó, levezetése Garamvölgyi [2006] TDK-munkájában található meg. A következőkben a véges-differenciás és a digitális waveguide modelleknek csak az eredményeit közlöm, mivel az általam készített program modális szintézist használ. Ezekről bővebben [Bank 2000] és [Bank 2006] tanulmányokban lehet olvasni.

3.1 A véges differenciák módszere

A módszer közvetlenül a hullámegyenletről indul ki, a deriváltakat véges differenciákkal helyettesíti. $x_m = m\Delta x$, $t_n = n\Delta t$ használatával (3.4) diszkrétizációja:

$$\left. \frac{\partial^2 y}{\partial x^2} \right|_{x_m, t_n} \approx \frac{y_{m-1,n} - 2y_{m,n} + y_{m+1,n}}{\Delta x^2} \quad (3.1.1a)$$

$$\left. \frac{\partial^2 y}{\partial t^2} \right|_{x_m, t_n} \approx \frac{y_{m,n-1} - 2y_{m,n} + y_{m,n+1}}{\Delta t^2}, \quad (3.1.1b)$$

ahol $y_{m,n} = y(x_m, t_n)$.

A fenti helyettesítést visszavezetve (3.4)-be:

$$y_{m,n+1} = \frac{c_t^2 \Delta t^2}{\Delta x^2} (y_{m-1,n} - 2y_{m,n} + y_{m+1,n}) - y_{m,n-1} + 2y_{m,n}, \quad (3.1.2)$$

ami a húr m pontban vett helyzetét számolja ki a következő időpillanatra, m pont és szomszédos pontjai aktuális és korábbi időpillanatban vett értékeiből. A rendszer numerikusan stabil, ha $\frac{\Delta x}{\Delta t} \geq c_t$. Hiller és Ruiz [1971] szerint, ha Δx és Δt értékét úgy választjuk, hogy a stabilitás határhelyzetére érjünk, azaz a hányados értéke megegyezzen c_t értékével, akkor a következő egyenletet kapjuk:

$$y_{m,n+1} = y_{m-1,n} + y_{m+1,n} - y_{m,n-1} \quad (3.1.3)$$

Ezzel a választással egyszerűen összekapcsolható a módszer a digitális waveguide megoldásával, mivel ott mindig fennáll a határhelyzet [Karjalainen és Erkut 2004].

3.2 A digitális waveguide modell

A digitális waveguide a haladó hullám leírását diszkrétizálja térben és időben. Alapgondolata, hogy a haladóhullám felbontható két, egymással ellentétes irányba (jobbra és balra) haladó hullámra [Smith 1983].

$$y(x, t) = y^+ \left(t - \frac{x}{c_t} \right) + y^- \left(t + \frac{x}{c_t} \right), \quad (3.2.1)$$

ahol y^+ és y^- a két haladó hullám. $x_m = m \cdot \Delta x$, $t_n = n \cdot \Delta t$ diszkretizáció alkalmazásával a fenti egyenlet a következő alakot veszi fel:

$$y(x_m, t_n) = y^+ \left(n\Delta t - \frac{m\Delta x}{c_t} \right) + y^- \left(n\Delta t + \frac{m\Delta x}{c_t} \right) \quad (3.2.2)$$

Mivel a $\Delta x = c_t \Delta t$ határhelyzet mindig fennáll, egyszerűsíthetünk:

$$y_{m,n} = y_{m,n}^+ + y_{m,n}^- \quad (3.2.3)$$

ahol $y_{m,n}^+, y_{m,n}^-$ a két haladó hullám aktuális helyzete x_m pontban, t_n időpillanatban. Ezzel a leírással, illetve a ténnyel, hogy veszteségmentes haladó hullám esetén a hullám adott pontján, a következő időpillanatban felvett értéke megegyezik az előző ponton, aktuális időpillanatban vett értékkel, azaz:

$$y_{m,n+1}^+ = y_{m-1,n}^+ \quad (3.2.4a)$$

$$y_{m,n+1}^- = y_{m+1,n}^- \quad (3.2.4b)$$

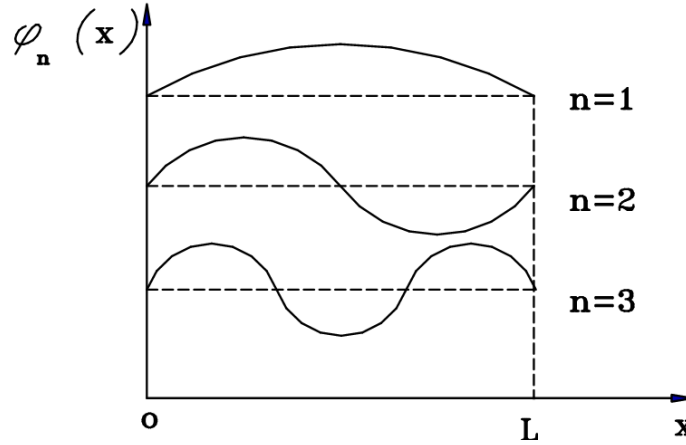
Ez a módszer különösen hatékony DSP-ken megvalósítva, mivel csak eltolni kell a két hullám értékeit, majd a veszteségeket modellezni egyetlen, összevont szűrővel, a DSP-k pedig ehhez hasonló műveletek párhuzamos végzésére lettek kifejlesztve.

E hatások modellbe építésére ez a dolgozat nem tér ki, bővebb levezetést tartalmaz [Smith 1983].

3.3 A modális szintézis

A modális szintézis az előző technikákkal szemben a frekvenciatartományban vizsgálja a hullámegyenletet. A teljes modellt szétbontja másodfokú lecsengő rezonátorok összességére, amikkel minden módust külön-külön kezelhetünk.

A módszer kiindulási alapja, hogy egy, mindkét végén rögzített húr leírható az alapharmonikusa és felharmonikusai szuperpozíciójával (3.1. ábra):



3.1. Ábra: Mindkét végén rögzített húr első három módusa [Fizipedia 1]

Az ilyen húr alakját leíró egyenlet a következő:

$$y(x, t) = \sum_{k=1}^{\infty} y_k(t) \sin\left(\frac{k\pi x}{L}\right), \quad (3.3.1)$$

ahol $y_k(t)$ a k -adik módus amplitúdója a vizsgált t időpillanatban, illetve $x \in [0; L]$.

Mivel a digitális technikában nem tudunk végtelen módusszámot alkalmazni és jól látható módon a számításigény a számolt módusok mennyiségétől függ, meg kell keresni azt az optimális módusszámot, ahol a fülünk már nem tudja megkülönböztetni a generált hang minőségét egy jóval nagyobb módusszámú változattól, de még valós időben tudja számolni azt a processzor. Elektromos basszusgitár esetén szerencsére egyszerű dolgunk van: a legmagasabb frekvenciák néhány kHz környékén jelennek meg, így viszonylag kevés, a megvalósított programban 50 módussal is megfelelő hangzást lehet elérni.

(3.1.1)-be behelyettesítve (3.3.1)-et megkapjuk a hullámgömb egyenlet megoldását. Ennek levezetése megtalálható [Szabó 2013] és [Bank 2006]-ban, az egyenlet végeredménye:

$$\frac{\partial^2 y_k}{\partial t^2} + a_{1,k} \frac{\partial y_k}{\partial t} + a_{0,k} y_k = b_{0,k} F_{y,k}(t), \quad (3.3.2)$$

ahol:

$$a_{1,k} = 2R_k \quad (3.3.2a)$$

$$a_{0,k} = \frac{T_0}{\mu} \left(\frac{k\pi}{L}\right)^2 + \frac{ES\kappa^2}{\mu} \left(\frac{k\pi}{L}\right)^4 \quad (3.3.2b)$$

$$b_{0,k} = \frac{2}{L\mu} \quad (3.3.2c)$$

$$F_{y,k}(t) = F(t) \sin\left(\frac{k\pi x_g}{L}\right) \quad (3.3.2d)$$

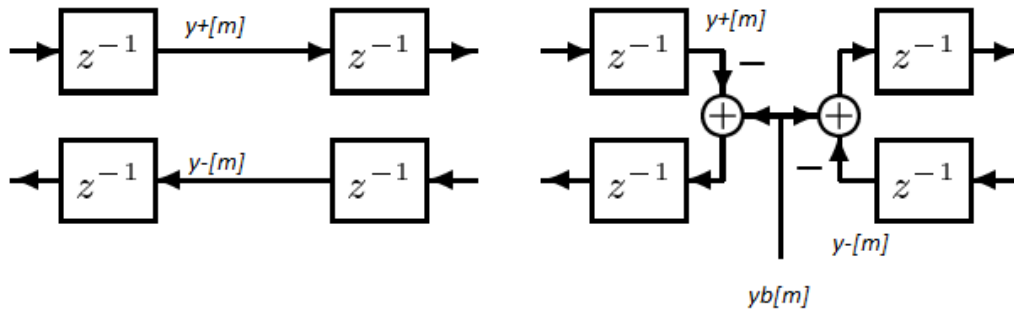
Az általános leírásnál (3.1)-ben $R(\omega)$ frekvenciafüggő veszteség volt jelen, ez a külön-külön számított módusok miatt könnyen implementálható.

(3.3.2d) egyenletben az egyetlen x_g pontban létrejövő gerjesztés esetén érvényes gerjesztőerő képlete látható. Mivel ez a gerjesztés megfelelő eredményhez vezet kis számításigény mellett, a megvalósítás során ilyen módszert alkalmaztam. $F(t)$ értéke a gerjesztés módjától függ, ez húros hangszereknél általában vagy pengető-, vagy kalapácsmodell jelent. Ennek a modellezéséről a későbbiekben lesz szó.

3.4 A slap bass játékmód jellegzetességei

Amikor a zenész slap bass technikával játszik, alapvetően két különböző pengetési technikát alkalmaz felváltva: a mélyebb húrokat a hüvelykujjával a fogólap felé csapja, míg a magasabb húrokat valamelyik ujjával a fogólappal ellentétes irányban, „felfelé” húzza, majd elengedi azt. Mindkét esetben teljesül, hogy – a közel oldalirányú, pengetővel, vagy ujjal való pengetéssel ellentétben – a húr nem rezeghet szabadon, a fogólapnak, vagy valamelyik bundnak fog csapódni. Ez a jelenség azzal jár, hogy a húr egy rövid időre az eredetileg lefogott bundhoz képest egy másik távolságánál lezáródik, így tulajdonképpen leírható lesz két darab, egymás melletti kifeszített húrral, amiknek egy pontjuk közös. ezzel az így létrejövő új hullámhossz frekvenciája is belekerül a spektrumba, illetve a húr energiája jelentősen csökken. Emiatt az így gerjesztett húrok rezgésében jóval magasabb, az eredeti hanghoz nem feltétlenül tartozó felharmonikusok is megjelennek. Ez végeredményben egy fémesen csengő, a sima pengetéshez képest jóval hamarabb lecsengő hangot eredményez.

A módszerről igen kevés irodalom található, egy kivételként digitális hullámvezetőt alkalmazó implementációt ír le Rank és Kubin [1997]. A módszer lényege, hogy a húr és a bund ütközésének helyén két részre vágja a hullámvezetőt, a húr adott pontbeli értékét pedig az ütközés pozíciójában tartja egészen addig, amíg a waveguide-ban haladó hullámok helyének összege meg nem haladja a fenti értéket (3.2. ábra).



3.2. Ábra: a waveguide modell, ha a húr kitérése nem éri el a bundot (balra), illetve ha eléri (jobbra). y_+ és y_- a haladó hullámokat, y_b a bund pozícióját jelképezi m időpillanatban

Ez a megközelítés akkor működik, ha a hullámvezetőben kitérési értékeket tárolunk, nem sebesség- vagy gyorsulásértékeket, mint azt általában szokták. Ez néhány extra művelettel jár, de a működést nem befolyásolja.

Modális szintézis esetén a bemeneti gerjesztés alapján indulunk ki. Minden időpillanatban figyeljük a húr pozícióját a bundok helyén, majd ha az átlépné a bundok magassága által megszabott határt, abban a pontban akkora ellentétes irányú gerjesztőerőt kell bejuttatni a rendszerbe, hogy az az adott pozícióban tartsa a húr, amíg a szabad mozgása onnan vissza nem húzza.

A húr egyenletét ismerjük, diszkretizáció után a következőképp néz ki:

$$y_k[n] = -a_{1,k}y_k[n-1] - a_{2,k}y_k[n-2] + w_k b_{0,k}F_{in}[n], \quad (3.4.1a)$$

ahol F_{in} a bemeneti gerjesztőerő, w_k pedig az adott ponthoz tartozó, módusonként különböző bemeneti súlyozás, általános esetben:

$$w_{x,k} = \sin\left(\frac{k\pi x}{L}\right). \quad (3.4.1b)$$

(3.4.1a) egyenletből a szükséges erőn kívül minden ismert, összevonásokkal a következő egyszerűsített alak érhető el:

$$y_k = y_{free,k} + c_{in,k}F_{in,k}, \quad (3.4.1c)$$

ahol $y_{free,k}$ a szabadon rezgő húr k . módusának amplitúdója, $c_{in,k}$ pedig a bemeneti súlyozás és $b_{0,k}$ szorzata.

(3.4.1b)-ből tudjuk továbbá, hogy

$$w_{bund,k} y_k = y_{bund,k}, \quad (3.4.2)$$

ahol $w_{bund,k}$ az aktuálisan figyelt bund pozíciójához tartozó súlyozás, azaz $\sin(\frac{k\pi x_{bund}}{L})$, míg $y_{bund,k}$ a húr amplitúdója ebben a pontban. Így, ha egy pontot előre meghatározott amplitúdójúra szeretnénk gerjeszteni, a következő egyenletet kapjuk:

$$y_{c,bund} = \sum_{k=1}^K w_{bund,k} (y_{free,k} + c_{in,k} F_{bund,k}) \quad (3.4.3a)$$

$$y_{c,bund} = \sum_{k=1}^K w_{bund,k} y_{free,k} + \sum_K w_{bund,k} c_{in,k} F_{bund,k} \quad (3.4.3b)$$

$$y_{c,bund} = \sum_{k=1}^K (w_{bund,k} (-a_{1,k} y_k[n-1] - a_{2,k} y_k[n-2] + \\ + w_{bund,k} w_{bund,k} b_{0,k} F_{bund,k}[n])), \quad (3.4.3c)$$

ahol $y_{c,bund}$ konstans, az épp vizsgált bund előre meghatározott pozíciója. Átrendezéssel megkapjuk $F_{bund}[n]$ értékét:

$$F_{bund}[n] = -1 \sum_{k=1}^K \frac{y_{c,bund} - y_{bund-1}}{b_w} \sin(k\pi x_{bund}), \quad (3.4.5)$$

ahol a -1-szeres szorzás az erőbevitel irányából adódik, illetve a változók értékei:

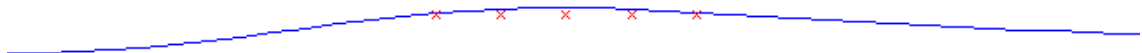
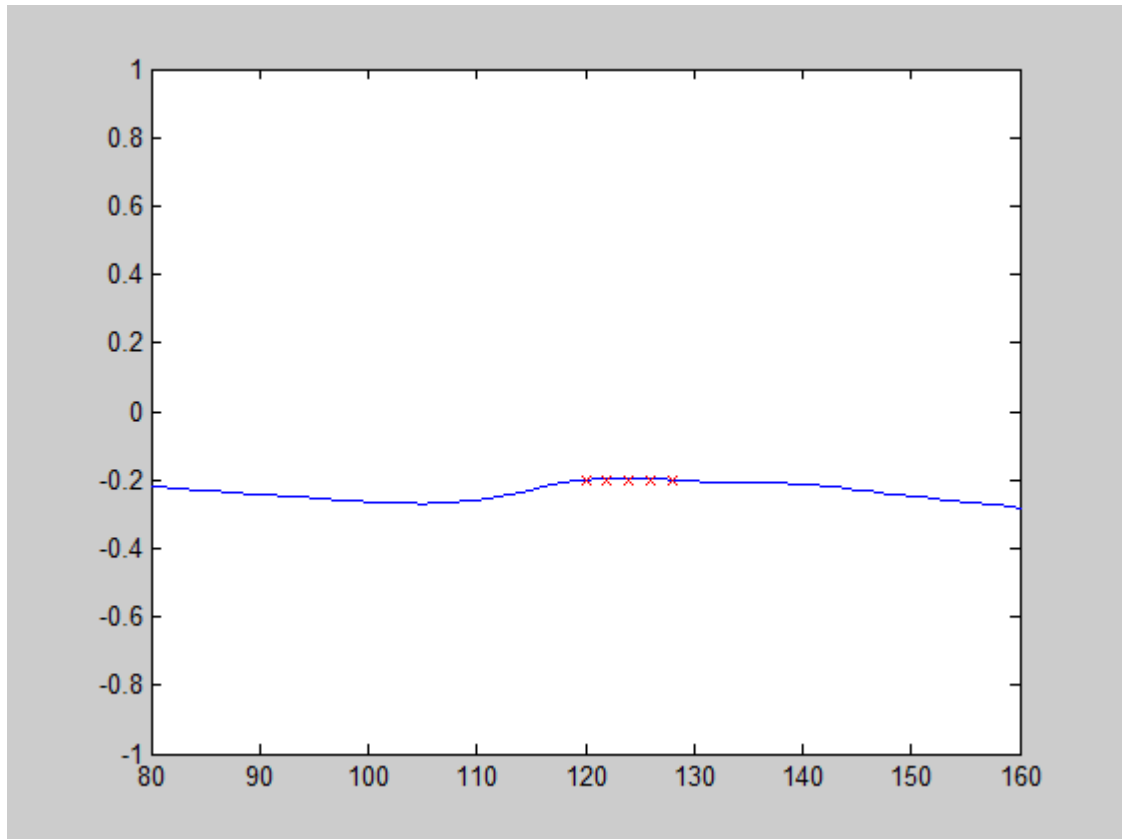
$$y_{bund-1} = \sum_{k=1}^K \sin(k\pi x_{bund}) y_k[n-1] \quad (3.4.5a)$$

$$b_w = \sum_{k=1}^K \sin^2(k\pi x_{bund}) b_{0,k}, \quad (3.4.5b)$$

A megvalósított algoritmus a következő módon működik: minden időpillanatban kiszámolja az éppen aktuális bemeneti erők és a húr mozgása hatására létrejövő $y_k[n]$ értéket, amiből a vizsgált pozícióhoz tartozó súlyozás után megkapjuk a húr amplitúdóját abban a pontban. Ha ez az érték kisebb, mint a bund helyéből adódó határ, szükség van akkora ellenerőre, ami erre a határértékre kényszeríti a húralakot. Ekkor (3.4.5) alapján kiszámoljuk azt, majd hozzáadjuk az algoritmus elején kiszámított y_k értékekhez. Ezután végzi a program a kimenet generálását a kimeneti súlyokkal. (3.4.1a) egyenlet alapján korrektebb eljárás lenne F_{in} értékét növelni a kapott eredménnyel, majd újraszámoltatni azt, de ez a megközelítés túlságosan erőforrásigényes volt a végül alkalmazotthoz képest.

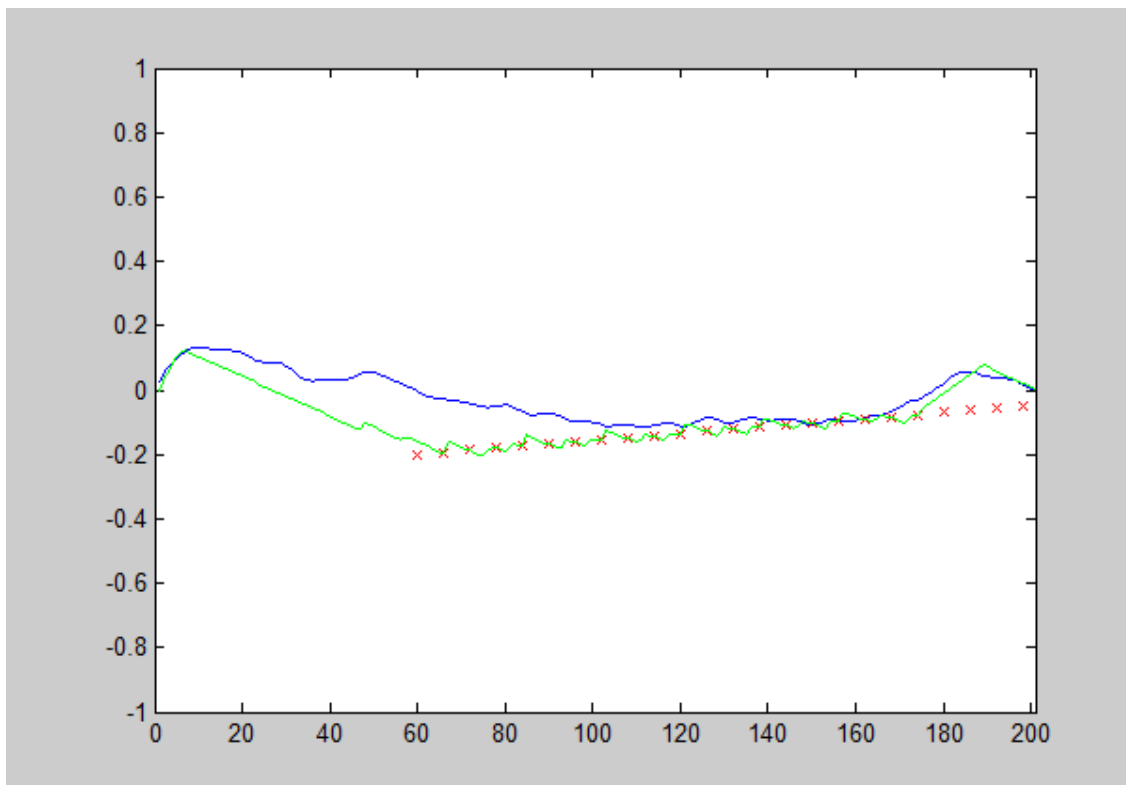
Ezt az erőt természetesen minden egyes bundra ki kell számolni, amelyekhez hozzáér a húr adott időpillanatban. Problémaként merülhet fel, hogy ha két egymáshoz igen közel lévő pontban is kell gerjesztőerőt számolni, a végül így kapott eredmény közel

a duplája lenne a valójában szükségesnek. A megvalósított algoritmus ezt úgy küszöböli ki, hogy miután kiszámolta F_{in} értéket egy bundra, azt továbbviszi a következőhöz is. Mivel az eredeti képletet minden különböző pontra újra kell számolni, ilyenkor már tartalmazza a képlet az előző pontokra számolt bemeneti erőket is, ezzel megszüntetve a problémát. Ezt a sejtést 5 db, egymáshoz igen közeli ponttal teszteltem, a végeredmény a vártak megfelelően alakult, azaz az algoritmus megfelelően működik (3.4. ábra).



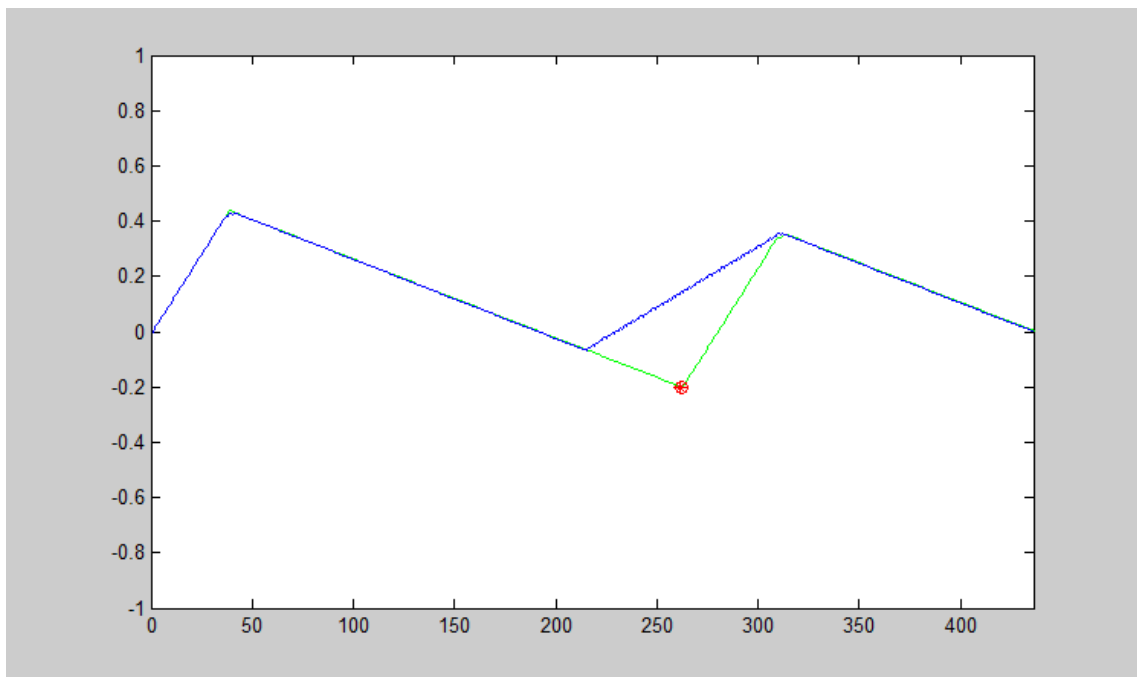
3.4. Ábra: Egymáshoz igen közeli pontok gerjesztése (x: a „bundok” helye, felül: a húr egy rövid szakasza, alul: a bundokkal való érintkezés kinagyítva)

A két, előzőekben bemutatott megoldásnak elvileg azonosan kéne működnie, azonban a gyakorlatban ez nem valósult meg, a waveguide „beragadt” a figyelt pontoknál, ezzel jelentős különbség alakult ki a két módszer között mind húralakban, mind a kimeneti hangban (3.5. ábra):

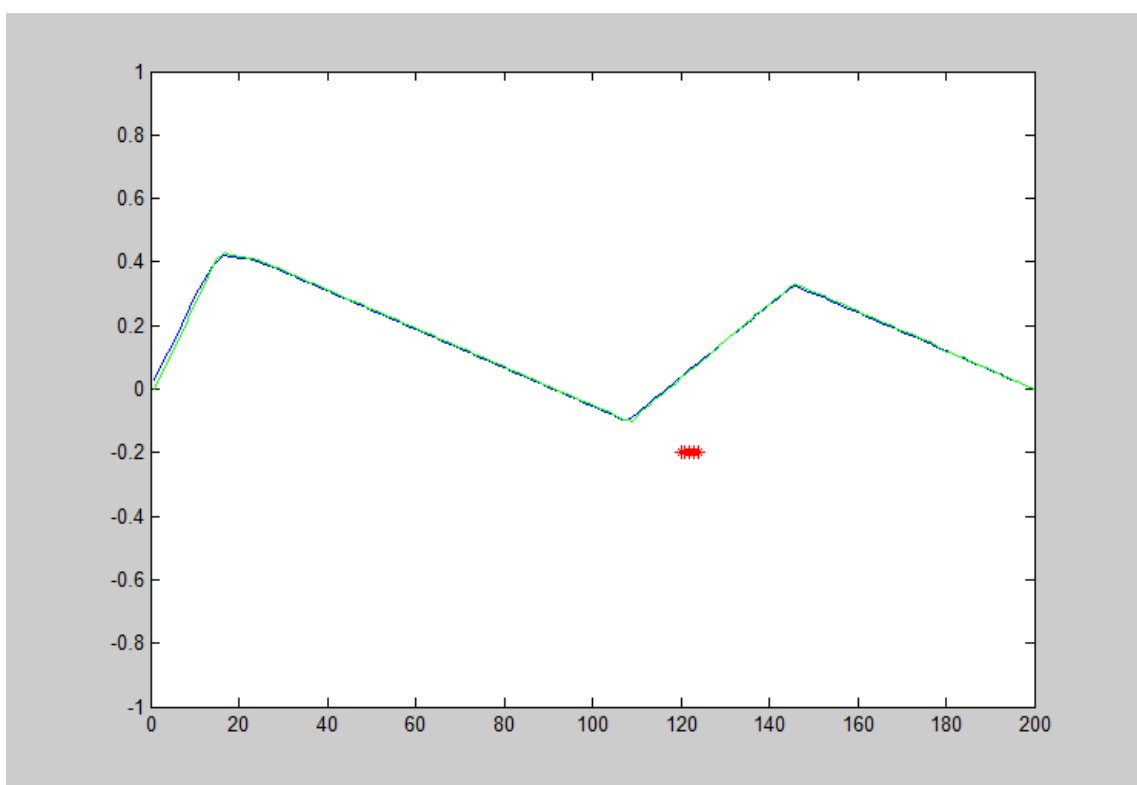


3.5. Ábra: Különbség a modális és a waveguide megoldás húralakjai között (kék: modális, zöld: waveguide)

Mivel ugyan ránézésre a modális szintézis eredménye tűnt logikusnak, de ezt biztosan nem lehetett kijelenteni, szükség volt egy harmadik megoldás leprogramozására is, amivel eldönthető a fenti kérdés. Az egyszerűség kedvéért a húr tömeg-rugó modelljét hoztam létre, kihasználva, hogy a megoldás azonos eredményre vezet a véges differenciák módszerével [Garamvölgyi 2006]. A modell igen egyszerű, a rugókkal összekapcsolt tömegpontok két szélső tagját rögzítjük, a bundokat jelképező helyeken pedig egyszerűen a határ értékére kényszerítjük az adott pontot, ha átlépné azt.



3.6. Ábra: A tömeg-rugó modell (kék) és a waveguide (zöld) húralakjai közti különbség



3.7. Ábra: A tömeg-rugó modell (kék) és a modális szintézis (zöld) húralakjai, 5 egymáshoz igen közeli bund esetén

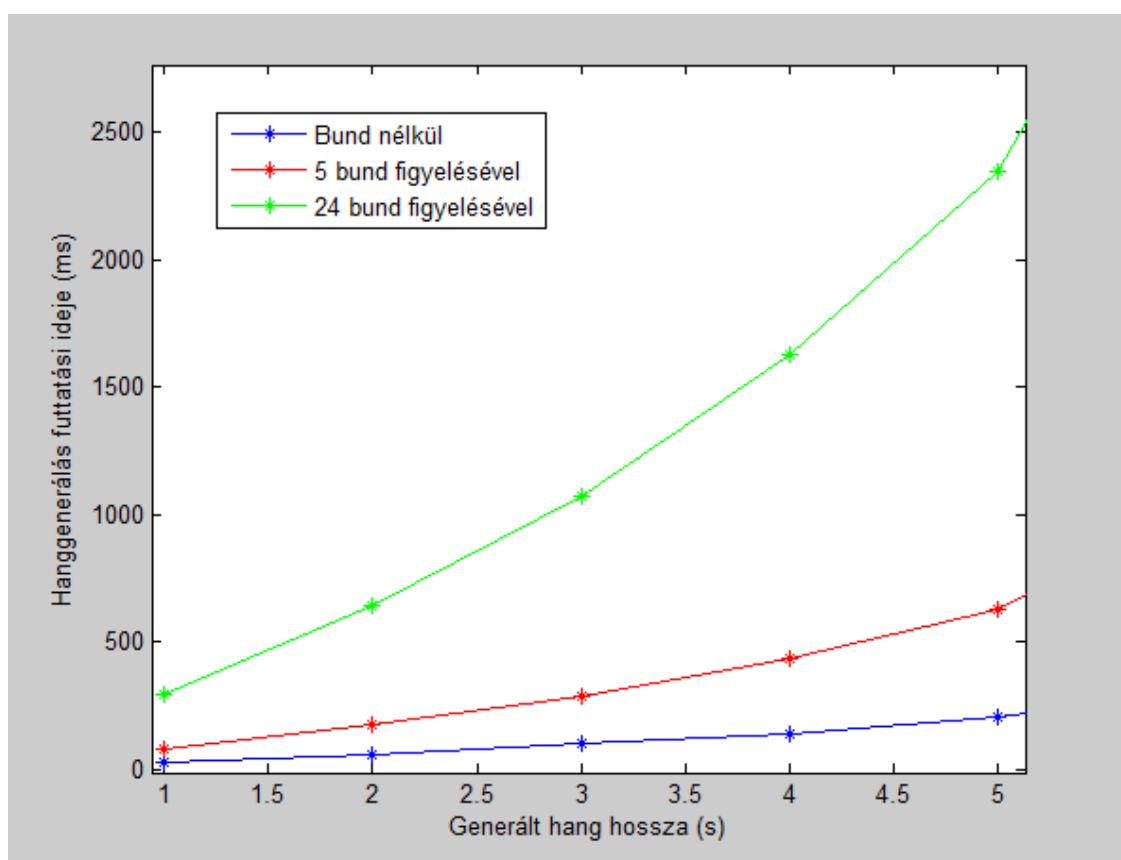
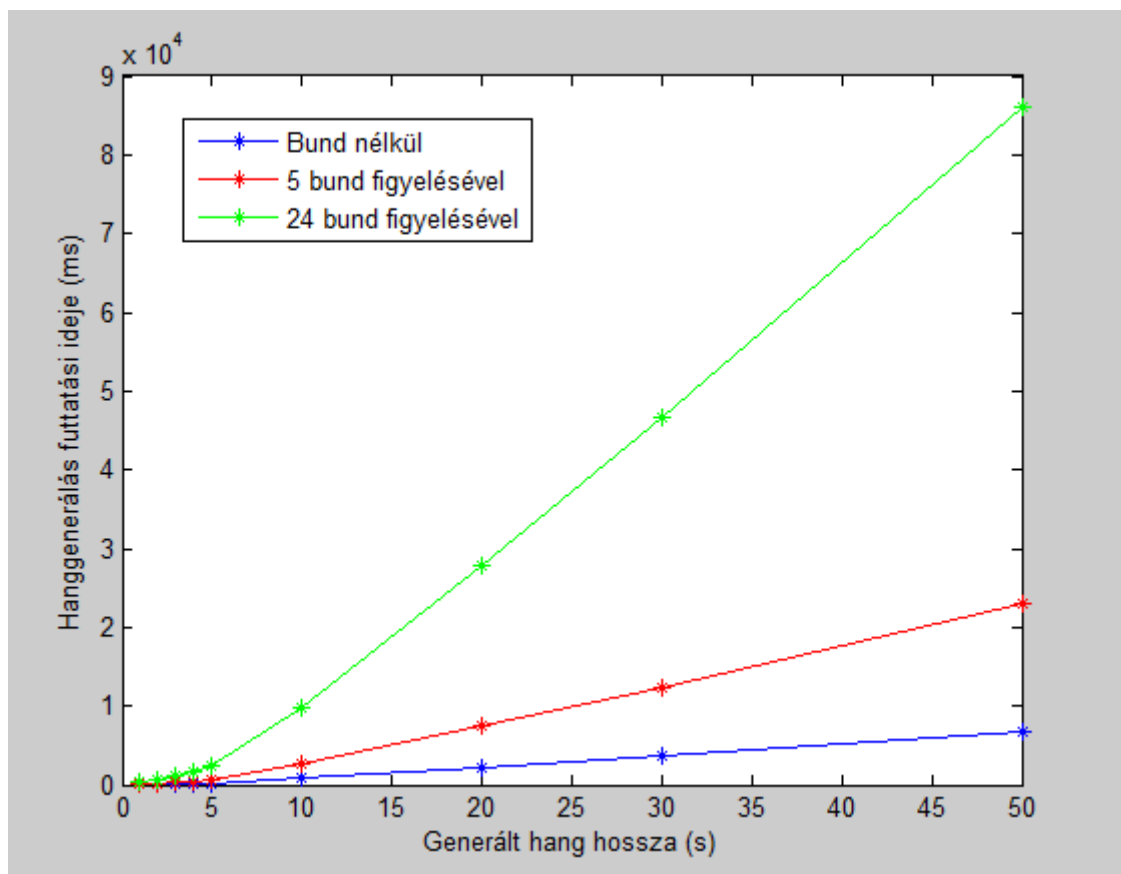
Mint az 3.6. és 3.7. ábrákon is jól látszik, a sejtés beigazolódott, a waveguide megoldása valóban beragad. A modális szintézis viszont még az egymáshoz közeli

gerjesztőerők problémájával szimulálva is teljesen együtt fut a tömeg-rugó modellel, így bebizonyosodott, hogy a megoldás fizikailag helyes.

Fontos megjegyezni, hogy nem tudjuk biztosan, hogy a Rank és Kubin [1997] által ismertett módszer hibás-e, vagy csak az implementációja nem sikerült megfelelően. Hosszas, eredménytelen tesztelés és hibajavítási kísérlet után végül eltekintettünk e megoldás alkalmazásától.

További kérdés a lecsengési idők megváltozása. Az analízis során egyszerű pengetésből számolt lecsengést használ a szintetizátor, amik slap esetén megváltoznak. Ez azonban nem probléma, mivel a húrmodellhez eredetileg számolt lecsengési idők a húr saját veszteségeinek tekinthetők, míg a bundoknál jelentkező külső erőbejuttatás – miután a húr eredeti mozgási irányával ellentétesen fejt ki hatását – a húr teljes energiáját, és ezzel együtt a lecsengési időket is lecsökkenti. Fontos megjegyezni, hogy a valós életben valószínűleg nem csak a bundoknak, hanem magának a fogólapnak is csapódhat a húr. Ez nyilván más hatásokkal rendelkezhet, elsősorban a faanyag miatt, ami több energiát nyelhet el, mint a fém bundok. A teljes fogólap jelenlétének modellezését azonban annak jóval nagyobb számítási igénye miatt nem lehet valós idejű szintézishez felhasználni. Ekkor ugyanis nem csak a bundokat jelképező pontokban, hanem azok között legalább 3-4, de inkább több további helyen kellene kiszámolni az erőket, ez pedig túl sok erőforrást igényel.

A modellbe építés szempontjából alapvető elvárás volt, hogy a lehetséges maximális 24 bund jelenléténel is futhasson az algoritmus valós időben, akár több húrra is, azaz adott hang generálása ideálisan legalább egy nagyságrenddel rövidebb ideig tartson, mint az így generált hang hossza. A húrmodell első megvalósítása során ez még a bundokat nem figyelve sem valósult meg, végül a kód optimalizálásával ez a probléma megoldódott. Az optimalizáció utáni futási idők láthatók a 3.8. ábrán, különböző hosszúságú hangok generálása esetén.



3.8. Ábra: A generált hang hossza és a futtatási idők közötti kapcsolat. Felül: 1-50s tartomány, alul: 1-5 s tartomány

A 3.8. ábrából két fontos tanulság vonható le: egyrészt hiába futtatunk egyetlen ciklust, amiben folyamatosan ugyanazokkal a változókkal dolgozunk, a kód futási ideje nem lineáris. Valószínűleg a processzor gyorsítótár használatával kapcsolatos a jelenség, de pontos magyarázatot nem találtam a problémára. A másik, fontosabb probléma, hogy 24 bund figyelése esetén már egy öt másodperces hangot is majdnem 2,5 másodpercig számol, ennél hosszabb hangok esetében pedig már több idő létrehozni a kimenetet, mint lejátszani azt. A bundok figyelése és a bemeneti erők számolása nélkül viszont teljesül, hogy nagyságrendekkel gyorsabb az algoritmus. Emiatt szükség volt egy kiegészítés bevezetésére: nyilvánvaló, hogy ha egy bizonyos ideig nem ér a húr semelyik bundhoz, akkor már lecsökkent a maximális kitérése annyira, hogy a továbbiakban sem fog.

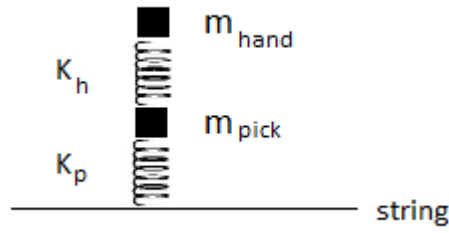
Így a modellbe építettem egy számlálót, ami az első húr-bund kölcsönhatástól kezdve minden bundérintés során alaphelyzetbe áll, egyébként pedig növelem az értékét. Ha ez a számláló elér egy határértéket, onnantól kezdve valószínűsítjük a fenti állítást és a továbbiakban nem figyeli a program az ütközéseket. Ezzel a módosítással az 50 másodperces hang eredetileg több, mint 80 másodpercig tartó futási ideje a tizedére csökkent, így pedig már akár mind a négy húr számításait egyszerre futtatva is tartható a valós idejű szintézis.

3.5 A gerjesztés modellezése

A megvalósított programban kétféle játéktílus közül választhat a felhasználó, ehhez három különböző gerjesztés modellezésére van szükség. Pengetős játék esetén egy pengetőmodellre, a slap bass stílushoz viszont kétféle gerjesztés tartozik: egy a magasabb húrok feltépéséhez, egy a mélyebb húrok csapásához.

3.5.1 Pengető modellezése

A pengető egy egyszerű, kéttagú tömeg-rugó modell használatával valósul meg: külön vesszük figyelembe a pengető, és külön az azt tartó kéz tulajdonságait.



3.9. Ábra: A pengető modellezése

A 3.9. ábrán m tömegpontok jelentik a zenész kezének, illetve a pengetőnek a súlyát, illetve K_h a pengető lehajlásából, K_p pedig annak a felületi rugalmasságából adódó erőt [Szabó 2013].

A modellezés során a következő feltételeket kell kielégíteni:

- a pengető hajlékony, annak a lehajlása határozza meg a húrra ható erőt
- e hajlékonyságtól függ, mekkora erő kifejtés után válik el a pengető a kifeszített húrtól
- az implementáció során egyetlen pontban hat gerjesztőerő a húrra
- a pengető és a húr között létrejön rugalmas kölcsönhatás
- a pengető és a kéz közötti erőket is modellezzük

A rugóerők alapvetően nemlineáris karakterisztikával rendelkeznek, a megvalósított modellben az egyszerűség kedvéért azonban lineárisnak vesszük azt.

Egy l_0 kezdeti hosszal rendelkező, K rugóállandójú rugó által kifejtett erő, ha azt l hosszra nyomjuk össze:

$$F_s = -K_s(l - l_0) \quad (3.5.1)$$

A modellünkben feltételezhetjük, hogy l_0 értéke mind a kéz és a pengető, mind pedig a pengető és a húr között zérus, mivel a pengetés pillanatában a húr közel merőlegesen áll a húrra, így a húr függőleges síkjára vett vetületén a kéz, a pengető és a húr pozícióját leíró értékek megegyeznek, azaz $y_{hand}(0) = y_{pick}(0) = y_{string}(0) = 0$. Ezzel a választással, a két összekötött rugó egyenletei leírhatók:

$$F_h = -K_h(y_{hand} - y_{pick}) \quad (3.5.2a)$$

$$F_p = -K_p(y_{pick} - y_{string}) \quad (3.5.2b)$$

$$F_{string,x0} = -F_p \quad (3.5.2c)$$

Tudjuk, hogy $F = ma$, illetve:

$$\Delta y = \int_{t_1}^{t_2} v(t) dt \quad (3.5.3a)$$

$$\Delta v = \int_{t_1}^{t_2} F(t) dt \frac{1}{m} \quad (3.5.3b)$$

Ezek a newtoni egyenletek egy lépésben diszkrétizálhatók és megtehetjük ezt (3.5.2b) és (3.5.2c) egyenlettel is. Ezek eredménye [Szabó 2013] alapján:

$$F_h[n] = -K_h(y_{hand}[n-1] - y_{pick}[n-1]) \quad (3.5.4a)$$

$$y_h[n] = y_h[n-1] + \frac{1}{f_s} v_h[n] \quad (3.5.4b)$$

$$v_h[n] = v_h[n-1] + \frac{1}{m_h} \frac{1}{f_s} F_h[n], \quad (3.5.4c)$$

illetve

$$F_p[n] = -K_p(y_{pick}[n-1] - y_{string}[n-1]) \quad (3.5.5a)$$

$$y_p[n] = y_p[n-1] + \frac{1}{f_s} v_p[n] \quad (3.5.5b)$$

$$v_p[n] = v_p[n-1] + \frac{1}{m_h} \frac{1}{f_s} F_p[n]. \quad (3.5.5c)$$

A húrt gerjesztő erő (3.5.2c) alapján tehát (3.5.5a) -1-szerese. A teljes húrmodellt gerjesztő erő végül (3.3.2) szerint, annak diszkrét alakját véve az alábbi:

$$F_y[n] = \sum_K K_p(y_{pick}[n-1] - y_{string}[n-1]) \sin(\frac{k\pi x_0}{L}). \quad (3.5.6)$$

Szükség van még a kezdeti értékekre is, a különböző tömegpontok helyzetét fentebb zérusnak vettük, így a kezdeti erőket és a tömegpontok kezdeti sebességeit kell megadni. Ezek az alábbiak:

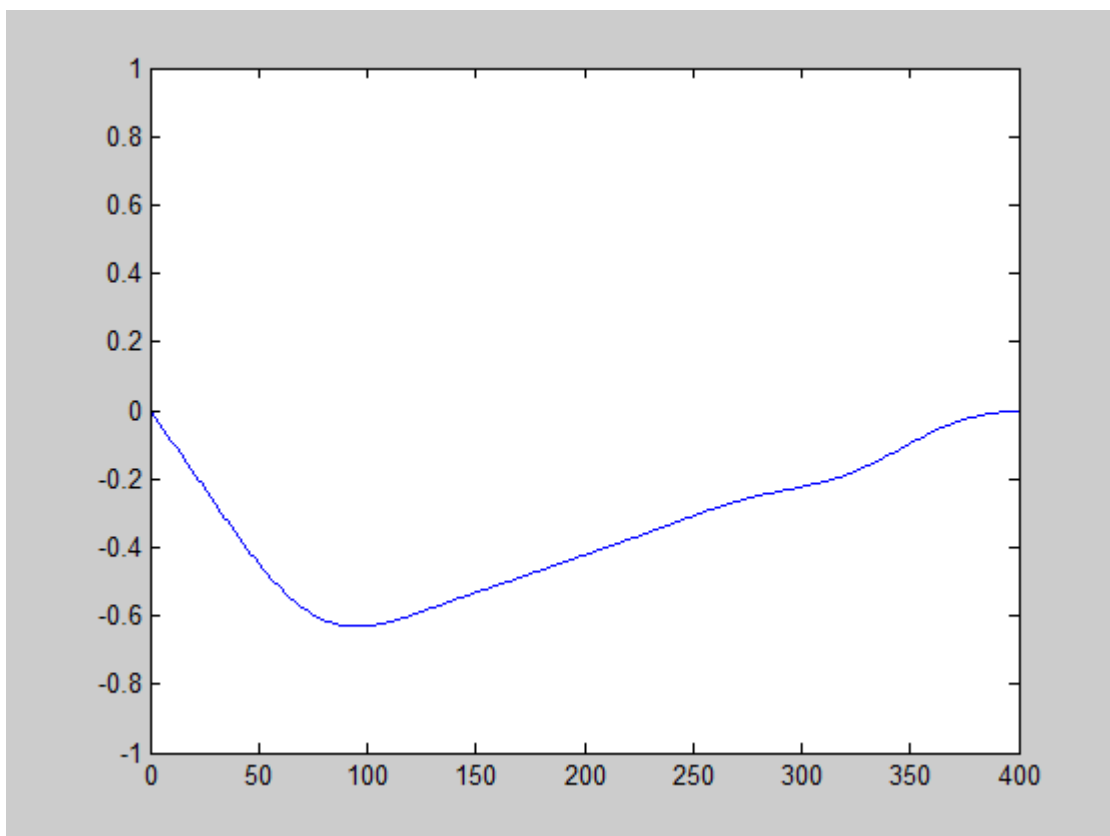
$$F_h[0] = F_p[0] = F_s[0] = 0 \quad (3.5.7a)$$

$$v_{hand}[0] = v_{pick}[0] = v_0 \quad (3.5.7b)$$

Ezekkel az összefüggésekkel leírható a pengető és a húr kölcsönhatása, amíg azok együtt mozognak. Hogy ez meddig történik, az attól függ, mekkora erő hatására hajlik meg a pengető annyira, hogy az lecsússzon a húrról, szabadon engedve azt. Ezt, és a további paramétereket becsléssel, illetve mérésekkel állapíthatjuk meg.

A pengető tömege egyszerű méréssel végezhető, a kéz tömegének meghatározása azonban nem triviális. A rugóállandóknál figyelembe kell venni, hogy basszusgitár esetén a használt pengetők általában jóval keményebbek, mint amit a szólógitárosok használnak, mivel nagyságrendekkel vastagabb húrokat kell megpengetniük. A pengetés sebessége nagyban függ az éppen játszott zenétől, értelemszerűen minél gyorsabb a tempója, annál nagyobb sebességgel mozog a pengető is, hogy a következő hangot időben megpengethesse a zenész. Azonban ez lassú tempójú zeneszámok esetén kevésbé triviális, ugyanis mindenképp szükség van egy minimális sebességre is, különben a húrt nem lehetne ésszerűen megpengetni.

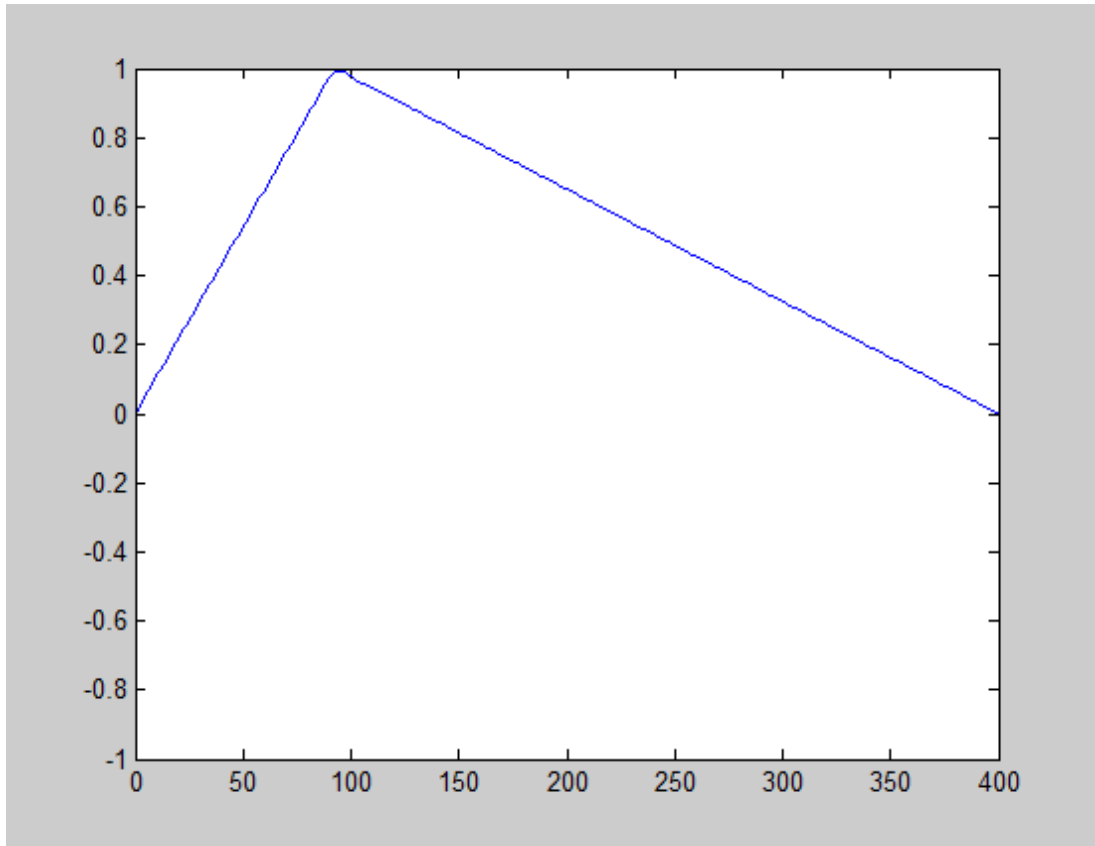
A pengetőmodell implementációja kezdetiérték-beállítással történik a megvalósított modellben, a hang létrehozásánál az első időpillanatnak azt vesszük, amikor a húr lecsúszik a pengetőről és szabad rezgésbe kezd. Ezért az algoritmus a hanggenerálás megkezdése előtt addig fut, amíg a pengető és a húr között létrejövő erő meg nem haladja azt az értéket, ami már elegendő ahhoz, hogy lecsússzon a pengető a húrról, majd ezután kezdődik a szűrés. Ekkor a húr alakja a 3.10. ábrán látható. A használt paraméterekről az analízis fejezetben lesz szó.



3.10. Ábra: A húr kezdeti alakja a pengetőmodellel létrehozva, a pengető lecsúszásának pillanatában

3.5.2 Slap bass modellezése

Slap esetén kétféle gerjesztési mód váltakozik, ebből a magasabb húrok feltépése az egyszerűbb. Ekkor a módusokat úgy állítjuk be, hogy az egy háromszög alakban kifeszített húralakot adjon, aminek a csúcsai a húr két végén, illetve a gerjesztés pozíciójánál vannak. Ezután hagyjuk a húrt szabadon rezegni, viszont ezután a szűrés során figyelni kell a bundokra a 3.4. részben bemutatott módszer szerint. Használható lenne a pengetésmodellhez hasonló kezdetérték-beállítás is más paraméterekkel, ám a kezdetiérték-beállítással jóval könnyebb relatív kitérési értéket beállítani, mint abszolútértékből visszavezetni azt. A bundok helyének számolása miatt így ebben az esetben ezt a módszert alkalmaztam.



3.11. Ábra: A húr kezdeti alakja háromszög gerjesztést beállítva, 50 módus használatával ($x_0 = 0.2L$ pontban gerjesztve, $A=1$ amplitúdóval)

Háromszögalak beállításához a következő képletet kell kielégítenie y_k -nak, [Szabó 2013] alapján:

$$\sum_{k=1}^{\infty} y_k \sin\left(\frac{k\pi x}{L}\right) = \varepsilon(x) \left(\frac{x}{x_0}\right) + \varepsilon(x - x_0) \left(\frac{L-x}{L-x_0} - \frac{x}{x_0}\right) - \varepsilon(x - L) \left(\frac{L-x}{L-x_0}\right) \quad (3.5.8)$$

Mindkét oldalt megszorozva $\sin\left(\frac{n\pi x}{L}\right)$ taggal, majd a húr hossza mentén x szerint integrálva, illetve y_k ortogonális mivolta következtében, a következő végeredményt kapjuk y_k kezdeti értékére:

$$y_k[0] = \frac{4L\left(\frac{x_0}{x} \sin(k\pi) - \sin\left(k\pi \frac{x_0}{x}\right)\right)}{\frac{x_0}{x} k\pi \left(\frac{x_0}{x} - 1\right) 2\pi kL} \quad (3.5.9)$$

A rendszer másodfokú, ezért szükség van az egy időpillanattal ezelőtti kezdeti értékre is. Mivel kezdeti értéket számolunk, nem szükséges pontosan tudnunk, mi történik a „negatív időben”, feltételezhetjük, hogy addig csillapítatlanul rezgett a húr, illetve hogy

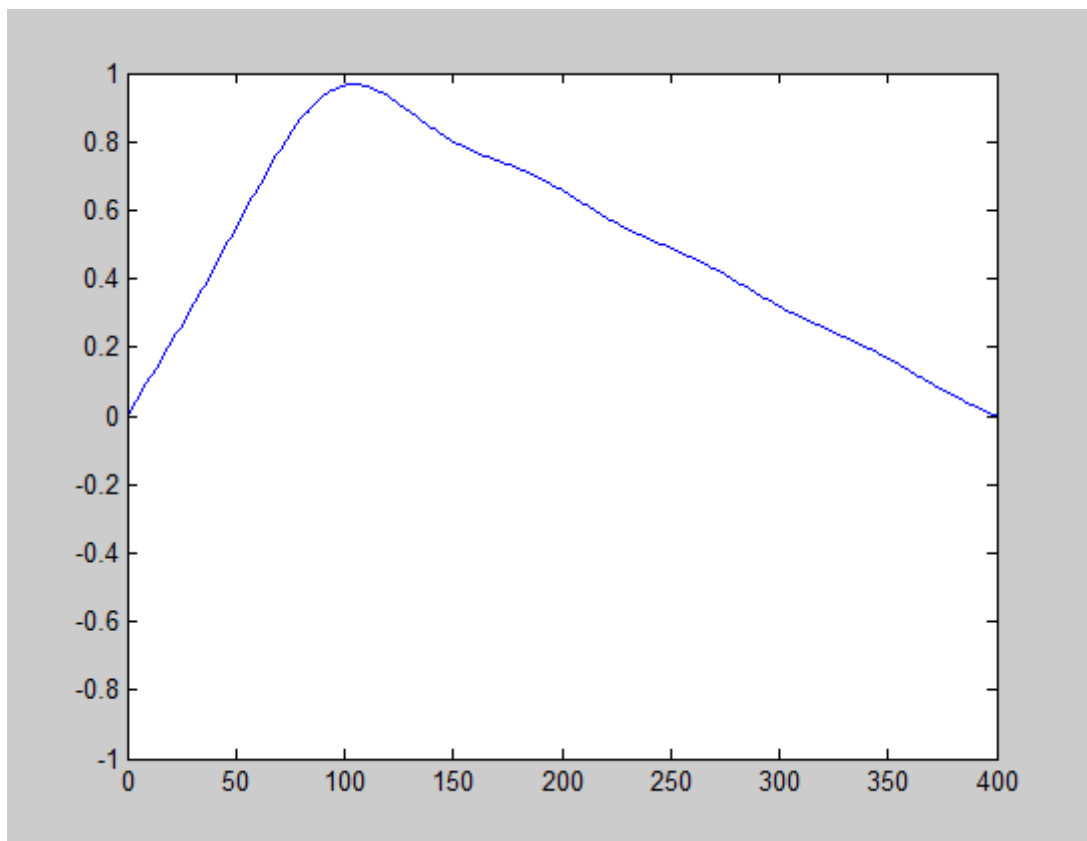
a legnagyobb amplitúdót $t=0$ pillanatban éri el. Ekkor a -1. időpillanathoz tartozó érték egyszerű $\cos$ függvénnyel való szorzással redukálódik:

$$y_k[-1] = y_k[0]\cos(-2\pi f_k dt) \quad (3.5.10)$$

Tudjuk, hogy $dt=1/f_s$, így az egyenlet végleges alakja:

$$y_k[-1] = y_k[0]\cos(-2\pi \frac{f_k}{f_s}) \quad (3.5.11)$$

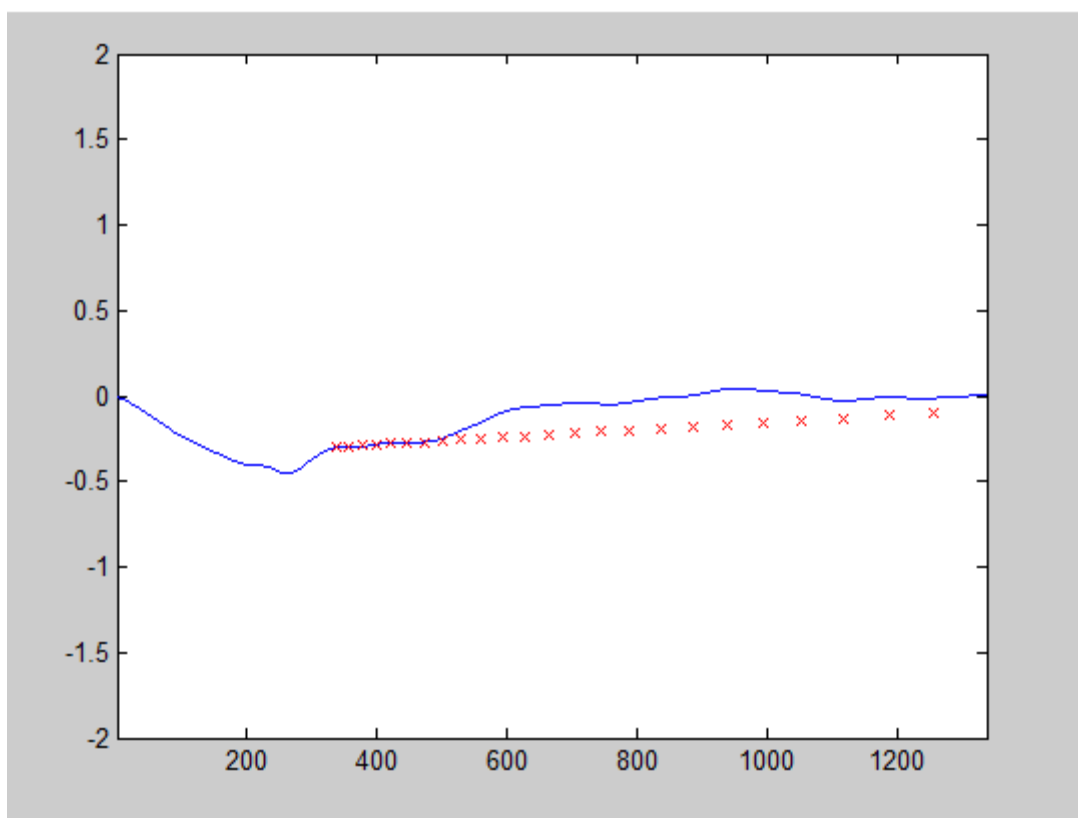
(3.5.8)-ból látszik, hogy tényleges háromszögforma létrehozásához végtelen módusra kellene kiszámolni az egyenletet. Ez értelemszerűen nem megoldható, ám az a 3.11. ábra alapján elmondható, hogy már 50 módus esetén is megfelelően közelítő alakú lesz a kifeszített húr. Minél magasabb frekvenciákat alkalmazunk, annál élesebb átmenetek érhetők el. Ebből következik, hogy minél kevesebb módusra hozzuk létre, annál lekerekítettebb, hullámosabb alakot kapunk. A 3.12. ábra ezt a jelenséget mutatja be.



3.12. Ábra: Háromszög gerjesztés alakja 10 módussal számolva ($x_0 = 0.2L$ pontban gerjesztve, $A=1$ amplitúdóval)

Ebben az esetben nincs beleszólása a húr alakjába a kezdeti alak beállításakor a bundoknak, mert azokkal ellentétes irányba feszítjük ki a húr. Azonban a mély húrokon használt technika esetén ez már nem igaz. Ekkor ugyanis, mivel a hangszertest felé csapja a zenész a húr, az már a gerjesztéskor is nekiütközhet valamelyik bundnak.

Ezt a csapást szintén tömeg-rugó modellel lehet modellezni, azonban ez egyszerűbb, mint a pengető esete, ugyanis itt csak a hüvelykujjat jellemző tömegpont és a húr közti kapcsolatot jelentő rugó szükséges a leírásához. Így egy kapcsolattal kevesebbet kell modellezni. A pengetővel való gerjesztéshez képest itt nem kezdetiérték-megadással dolgozunk, ugyanis mire a gerjesztés megszűnik, akár több bundnak is nekicsapódhat a húr, ami fontos hangszínbeli változást hoz. Így ebben az esetben a rendszer – az eddigiekkel ellentétben – csak néhány száz minta után kerül gerjesztésmentes állapotba a pengetés helyén. A húr alakját ekkor a 3.13. ábra mutatja be.

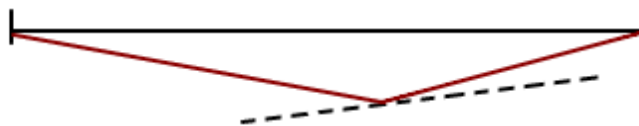


3.13. Ábra: Húralak a gerjesztés megszűnésekor, hangszer felé csapott húr esetén

($x_0 = 0.2L$ pontban gerjesztve)

Utóbbi két módszernél a programnak jelezzük, hogy figyelnie kell a bundokhoz csapódást is. További problémát vet fel a lefogott hangok esete: ekkor a húr egyik végpontja a lefogott bundon rögzül, megváltoztatva annak az alakját és feszességét is. A

hatást a 3.14. ábra szemlélteti. Így ilyenkor két eset lehetséges: vagy rögzítjük a bundok abszolút pozícióját, ekkor a húr egyik végpontja nem 0 értéket vesz fel, amit igen nehéz egyenletekkel hatékonyan leírni, vagy a húr kifeszített alakját tekintjük etalonnak, ekkor a koordináta-rendszerünk, és ezzel együtt a bundok pozíciói is változnak. Az implementáció nehézségei miatt ezt a változást a programban nem teljesen veszem figyelembe, a megvalósított módszerben a bundok x tengelyen való pozícióját a lefogott bundtól figyelem, y tengelybeli értéküket, azaz a magasságukat pedig úgy számolom újra, hogy pozíciójukat feljebb tolom a lefogott bund magasságával. Tehát ha pl. az 5. bundhoz tartozó hangot kell megszólaltatni, a modell a 6-24. bundok eredeti pozíciójával számol, értékükből pedig kivonja az 5. bund magasságát. A lefogásból eredő egyéb fizikai változásokat, pl. a feszesség hatását nem veszem figyelembe.

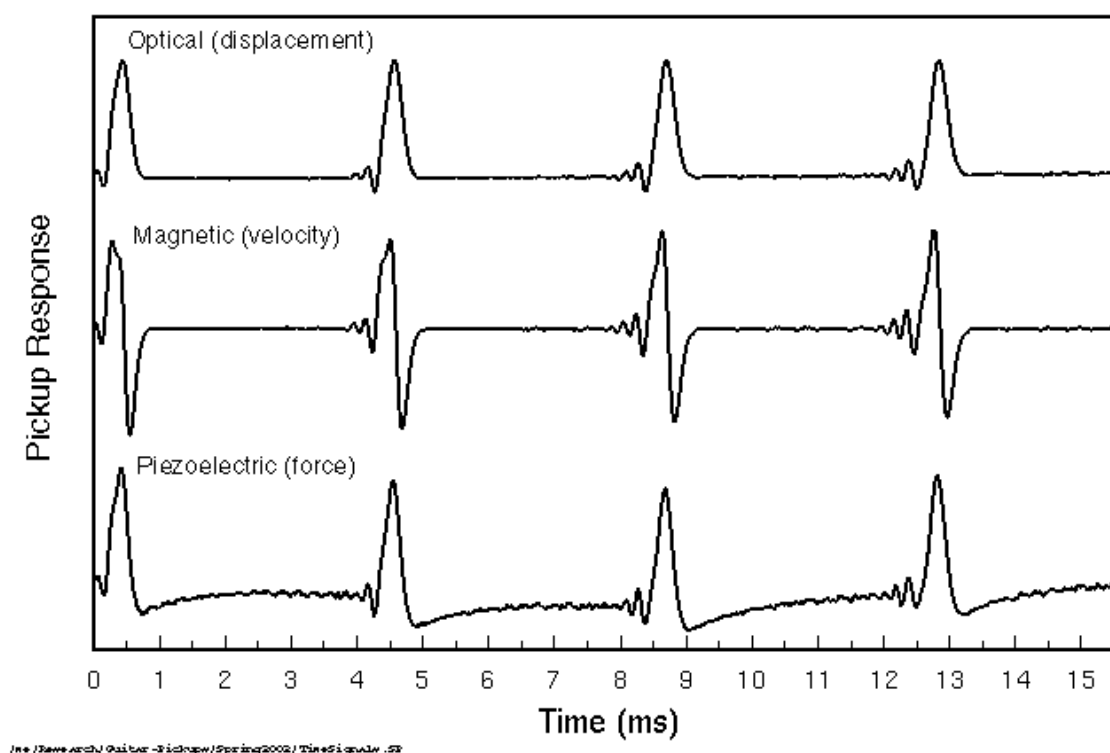


3.14. Ábra: A húr és a bundok pozícióinak megváltozása üresen pengetett húr (fekete) és lefogott hang (piros) esetén

4 A pickup modellezése

Míg akusztikus hangszereknél a hangot leginkább befolyásoló tényező a hangszer teste, főképp annak anyaga(i), illetve formája, elektromos gitárok esetén nem ez a legfontosabb. Bár két, a test anyagán kívül azonos hangszer között jól hallható tonális eltérések is lehetnek, a hangzást legfőképp meghatározó elemek a hangszedők, azaz pickupok. Ezek minősége, elhelyezkedése, felépítése mind befolyásolja a hangszer kimenetét.

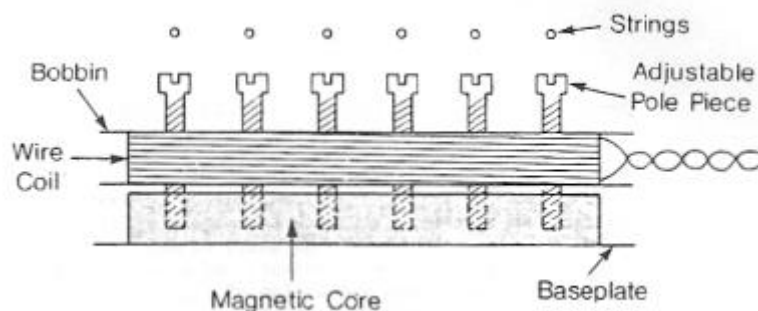
Többféle működési elvű hangszedőt gyártanak, leggyakrabban mágneses, főleg elektroakusztikus hangszereken pedig piezoelektromos pickupok találhatók. Létezik még ezeken kívül optikai alapú is, ám ezek nem gyakoriak.



4.1. Ábra: Különböző elvű hangszedők kimenetei (forrás: [Russell 2012])

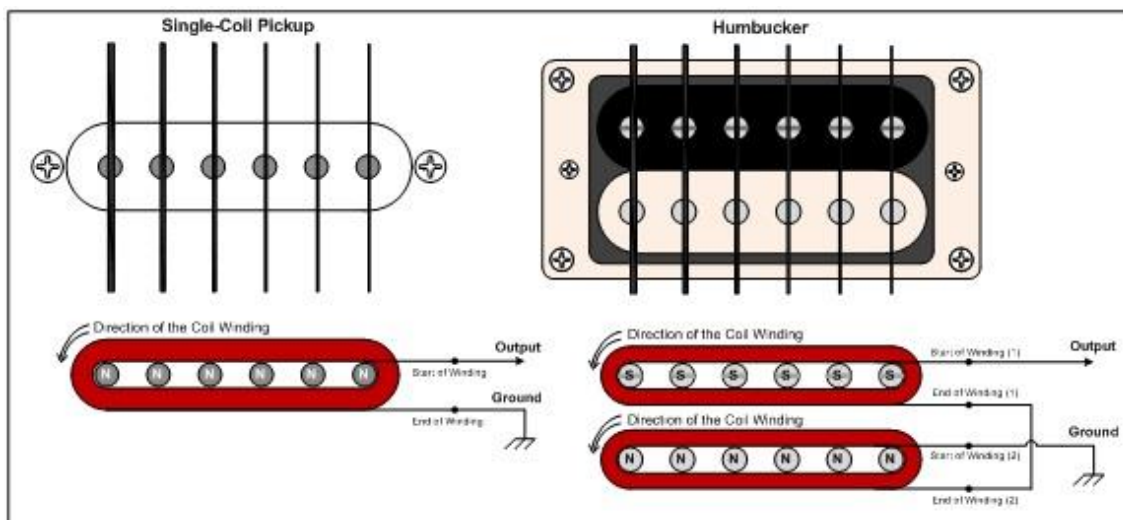
Mint az a 4.1. ábrán látható, az optikai elven működő hangszedő a húr kitérését érzékeli, míg a mágneses társa annak a sebességét. Ezekből már elvben is eltér a piezoelektromos pickup, ami nem a húr valamelyik belső pontjának a változását méri, hanem a húrlábon keletkező erőt, ami egyébként az akusztikus hangszer testének is átadódik.

4.1 A mágneses pickupok tulajdonságai



4.2. Ábra: Mágneses hangszedő felépítése (forrás: <http://www.guitarsinsync.com>)

A mágneses hangszedők felépítése igen egyszerű, egy mágneses mag és a körülötte lévő tekercselés az alapja, amit bizonyos típusoknál kiegészítenek a húrok alatt található, állítható magasságú csavarokkal (4.2. ábra). Ezekkel húronként szabályozható a mágneses tér távolsága és egyben a kimeneti feszültség nagysága, míg a többi pickup esetén ez csak a tokozás két oldalán lévő csavarokkal állítható. Egy hangszedő állhat ennyiből, ezeket hívják single coilnak, vagy összeköthetnek két ilyen, a fázisaikat eltolva, ezek az ún. humbuckerek. Előbbi típus magasabb, felharmonikusokban gazdagabb, ámde zajosabb kimenettel rendelkezik, ilyenből található pl. három darab a Fender Stratocaster modelljén. A humbucker típusú hangszedők kevésbé zajosak, jóval nagyobb kimenetük van, cserébe „sötétebb” hangzásúak, dominánsabb a mély frekvenciatartomány. E tulajdonságok miatt előszeretettel használják ezeket keményebb, jobban torzított rockzenéhez (4.3. ábra)



4.3. Ábra: single coil és humbucker típusú hangszedők (forrás: <http://revolutiondeux.blogspot.com>)

Basszusgitárok esetében létezik egy harmadik típusú, split-coilnak nevezett pickup is: ez egy darab single-coil pickup, amit „kettévágnak”, majd a két részt két-két húr alá teszik (lásd 4.4. ábra), ilyen elrendezésű pl. a híres Fender Precision Bass. Elmondható, hogy általában single coil hangszedőket használnak a legtöbb basszusgitáron.

Fizikailag a hangszedők mágneses fluxust hoz létre a térben, amit a húr elmozdulása megváltoztat. Ez a fluxusváltozás a tekercselésben közvetlenül elektromos feszültséget indukál [Fodor 1996], amiből végül – a pickup átvitelén keresztül – a kimeneti jel lesz.

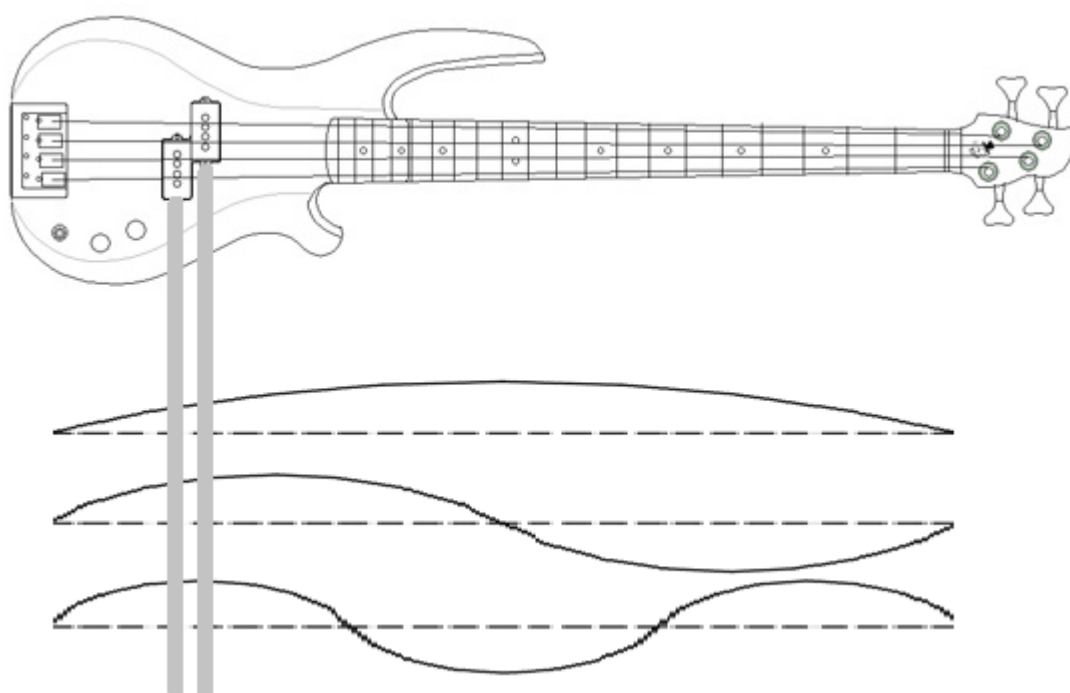
Mivel a feszültséget a húr elmozdulása indukálja, nemlineáris elem is várható a modellben. Ez nagyban játéktílus- és hangszerbeállítás-függő, így a modellezése nem triviális, sok paramétertől függ. Ezért a megvalósítás során egy egyszerű, nagyon általános modell implementációját céloztuk meg.

A mágneses pickupok modellezéséről máig kevés irodalom lelhető fel, ez egy viszonylag új terület, azonban a hangszedők felépítése és működése ismert, így alapszintű modellezési módszerek implementálhatók. A hangszedők négy lényeges tulajdonsága határozza meg a hatásukat: a helyzetük és szélességük, a nemlinearitásuk, a deriváló hatásuk és az áramkörük átvitele [Paiva et. al. 2012]

Ezek a hatások a hangszedő különböző részein jönnek létre, továbbá mivel nemlineáris tulajdonságok is találhatók ezek közt, ezért számításuk az alábbi sorrendhez kötött:

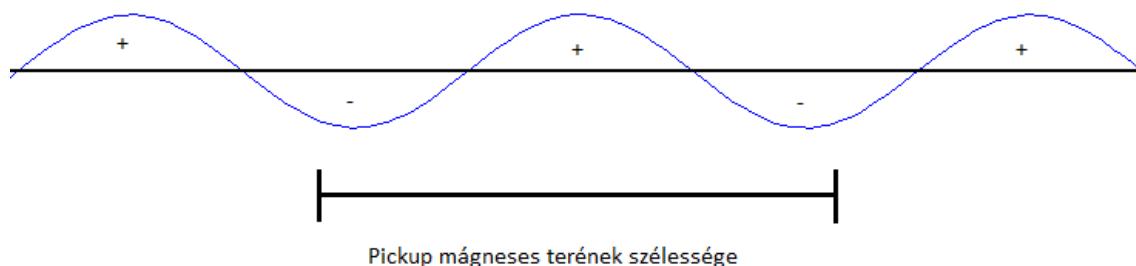
1. A hangszedő helyzetének hatása
2. A hangszedő szélességéből adódó tulajdonságok meghatározása
3. A nemlinearitás hatásának figyelembe vétele
4. A deriváló hatás
5. Az áramköri átvittel való szűrés

4.2 A hangszedő helyzete és szélessége



4.4. Ábra: hangszedők helyzete és a harmonikusok amplitúdója

Mint az a 4.4. ábrán is jól látható, a hangszedők elhelyezkedése alapvetően határozza meg a kimeneti jelet – minden felharmonikus esetén más-más amplitúdó van jelen azon a helyen, ahol a pickupok találhatók. Mivel a hangszedők mágnesesek és a húr által gerjesztett mágneses tér változását érzékelik, sem a pickupok, sem azok mágneses tere nem tekinthető pontszerűnek.



4.5. Ábra: A pickup átlagoló hatása

Bizonyos frekvenciájú rezgés felett a hangszedő húrral párhuzamos irányú mágneses térbe beérkező, illetve abból távozó hullámok hullámhossza rövidebb lesz, mint a mágneses tér szélessége (4.5. ábra). Ezek a nagyon rövid hullámok főként a magasabb módusfrekvenciákra érvényesek, illetve a térből adódóan kiátlagolódnak, így lényegében aluláteresztő szűrőként értelmezhető ez a tulajdonság [Csaba 2002].

Az átlagoló hatás többféleképpen implementálható: legegyszerűbb, még fizikai alapokon nyugvó módszer a kimenet aluláteresztő szűrése.

Ezek specifikációja nem egyszerű, mivel minden húrra más vágási frekvenciájú szűrőt kellene alkalmazni. Ez abból adódik, hogy bár a hűrok ugyanolyan hosszúak, a vastagságuk, megfeszítettségük és ezek miatt az alaphfrekvenciájuk különböző. Mivel $\lambda_k = 2L/k$, ahol λ_k a k . módus hullámhossza, L pedig a húr hossza, ha tudjuk a mágneses tér szélességét, kiszámolható, melyik felharmonikustól kezdve jelenik meg egyre nagyobb elnyomás a kimeneten, erre a frekvenciára kell tervezni aluláteresztő szűrőt.

Pontosabb módszer a hangszedők teljes szélességében figyelni a húr pontjait, majd ezt átlagolni. A modális szintézis előnye, hogy ez egyszerűen elvégezhető feladat, csupán a pickupok szélességére kell integrálni a húralak megfelelő tartományát:

$$p(t) = \frac{1}{x_2 - x_1} \int_{x_1}^{x_2} y(x, t) dx, \quad (4.2.1)$$

ahol $p(t)$ a pickup kimenete, x_1 és x_2 a hangszedő által lefedett tartomány végpontjai, $y(x, t)$ pedig a húralak. (4.1.)-be behelyettesítve a húregyenlet modális megoldását:

$$p(t) = \frac{1}{x_2 - x_1} \int_{x_1}^{x_2} \sum_{k=1}^K y_k(t) \sin\left(\frac{k\pi x}{L}\right) dx \quad (4.2.2)$$

Tudjuk, hogy az integrálás és az összegzés felcserélhető, a konstans szorzás pedig az integrálból kiemelhető. Ezzel a pickup kimenete:

$$p(t) = \sum_{k=1}^K y_k(t) \frac{1}{x_2 - x_1} \int_{x_1}^{x_2} \sin\left(\frac{k\pi x}{L}\right) dx \quad (4.2.3)$$

A szintetizátoromban ez utóbbi megoldást használom.

4.3 A hangszedő nemlinearitása

Ideális esetben a pickup mágneses tere homogén, a húr pedig annak valamelyik tengelyével párhuzamos mozgást végez. Az első kikötés fizikailag lehetetlen, hiszen ehhez a húr teljes tartományát le kellene fednie a mágneses térnek. A húr ideális mozgása elvileg létrehozható, gyakorlatilag a legritkább esetben történik meg. Valós körülmények között megpengetés után közvetlenül a húr pályája kör-ellipszis alakot vesz fel a húrra merőlegesen, majd néhány (tized) másodperc után beáll egy, a hangszedőre se nem merőleges, se nem párhuzamos, egyenes pályára. Így a húr és a pickup között folyamatosan változik a távolság és a közöttük lévő mágneses kölcsönhatás is. E mozgások és a nem homogén mágneses tér miatt a húr kitérése és a pickup kimeneti feszültsége közti kapcsolat nemlineáris [Csaba 2002].

A dolgozat keretein belül a saját gitárom hangszedőivel nem folytattam méréseket, az implementált algoritmus [Csaba 2002] alapján történt. Eszerint a pickup kimenete függ a megpengetett húr amplitúdójától, minél nagyobb a kitérése, annál jobban torzít a kimenet. A pontos mérték függ a hangszedő felépítésétől, minőségétől, de valószínűleg még a gyártási szórástól is. Ennek ellenére közelítésként jól használható a tangensfüggvény -1 és $+1$ közti tartománya. A programomban is ezt a megoldást használom.

4.4 A hangszedő deriváló hatása

A deriváló hatás abból fakad, hogy a hangszedő kimenete nem a húr pozíciójától függ, amit a modellben számolunk, hanem annak a sebességétől [Csaba 2002]. A modellben a húr aktuális kitérésének a helyzetét kapjuk meg, emiatt egy deriváló szűrőt be kell építeni a kimeneti jel létrehozása előtt.

4.5 A hangszedő áramkörének átvitele

Mivel a pickup hálózata egy RLC kör, ami ráadásul több hangszedő esetén sorba, vagy párhuzamosan van kötve (néhány esetben ez is a felhasználó által változtatható), az átvitel meghatározása nem triviális, sokféle paramétertől függ. Emiatt, és mert a többi hatás jóval alacsonyabb vágási frekvenciával szerepel, ezt a hatást a programomban nem modelleztem. Bővebb ismertetés [Paival et. al. 2012]-ben található.

5 Analízis

A modell pontos leírásához szükségesek a hangszer fizikai paraméterei. A modális megközelítés tulajdonságai miatt elegendő az egyes módusok lecsengési idejét és frekvenciáját tudnunk, a húr egység hosszára eső tömege ugyan szerepel a képletben, de csak konstans szorzásként, ezért ennek nagy pontosságú meghatározása nem feltétlenül szükséges [Szabó 2013]. A húr hosszát tudnunk kell a bundok pozíciójának meghatározásához, illetve a gerjesztés és a kimenet számolásakor, ám, mivel a bundok pozícióját sem nagy pontossággal mértük, kielégítő eredményt ad, ha egyszerűen a gitár névleges menzúrájával számolunk. Ez 34 inch, azaz kb. 86,36 cm.

A húrmodell paraméterei:

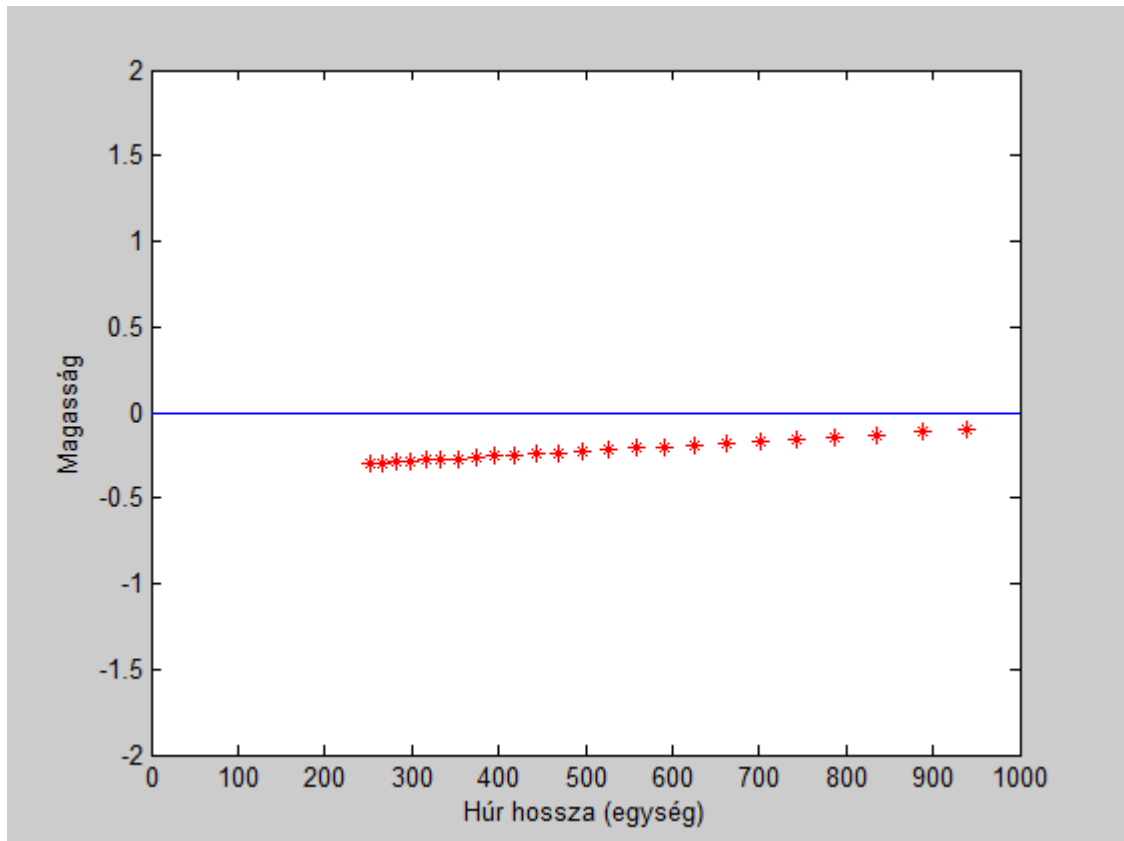
- μ húr egység hosszára eső tömege [kg/m]
- L húrhossz [m]
- f_k módusfrekvenciák [Hz]
- τ_k a fenti frekvenciához tartozó lecsengési idő [s]

A gerjesztés paraméterei:

- Pengetővel való gerjesztés:
 - m_{h1} zenész kezének tömege [kg]
 - m_p pengető tömege
 - K_{h1} pengető lehajlását jellemző rugóállandó [N/m]
 - K_p pengető felületi rugalmasságát jellemző rugóállandó [N/m]
 - v_1 pengetési sebesség
- Slap bass technika:
 - m_{h2} zenész hüvelykujjának tömege [kg]
 - K_{h2} a hüvelykujjal húrra csapást jellemző rugóállandó [N/m]
 - v_2 a hüvelykujj sebessége

Szükség van még a paramétereken kívül a modellezendő basszusgitár bundjainak húrlábtól való távolságára (x tengely), illetve a bundok legmagasabb pontjainak húrtól

való távolságára (y tengely). Ez utóbbi értékek elégséges pontossággal mérhetők egyszerű mérőszalag segítségével, így részletesebb kifejtésüktől eltekintek. Hasonló paraméter μ is, a különböző húrok adatait a [D’Addario 1] ismertetőjéből olvastam ki, majd ezekből egyetlen, általánosnak tűnő értéket használtam: $\mu = 0.00145$ kg/m. A gerjesztetlen, kifeszített húr és a bundok mérések során meghatározott pozíciója az 5.1. ábrán látható:



5.1. Ábra: a húr és a bundok elhelyezkedése gerjesztetlen állapotban

A gitártest átvitelét az egyszerűség kedvéért elektromos hangszer lévén elhanyagoljuk, míg a pickupokét nem tudjuk kellő pontossággal meghatározni. Emiatt előbbivel egyáltalán nem foglalkozunk, utóbbi hatásait pedig a korábban említett általános függvényekkel implementáljuk.

Az analízis MATLAB segítségével történik, mivel a program igen sok funkcióval jelentkezik jelanalízis, jelfeldolgozás és szűrőtervezés témakörben, ami jelentősen megkönnyíti a munkát.

5.1 Mérések

A mérések 2013 tavaszán történtek a Budapesti Műszaki és Gazdaságtudományi Egyetem Méréstechnika és Információs Rendszerek tanszékének DSP laborjában. A modell paramétereikhez szükséges f_k és τ_k értékek meghatározásához szükség volt az etalon gitár hangjaira. Ez a hangszer a saját basszusgitárom, egy Warwick Corvette Std. típusú elektromos basszusgitár, a mérés idején 4 db, 2-3 hetes, alig használt D'Addario EXL húrral felhúrozva. A gitárom speciális hangolása miatt ezek rendre .045, .065, .085 és .125 inch átmérőjűek, tehát a három magasabb húr az általában használt húroknál kicsivel vastagabb, míg az E húr helyén egy, az általánosnál vékonyabb mély H húr van. A hangszer így a megszokott EADG hangolás helyett CGCF-re van húrozva, azaz a legmélyebb húr 2, míg a többi 1 egész hanggal mélyebb a standardnál.

5.1.1 Hangok felvétele

A mérési elrendezés egyszerű volt: a gitár kimenete közvetlenül egy jó minőségű USB-s erősítő-hangkártya segítségével lett összekötve a számítógéppel. A teljes fogólap által lefedett tartományt, azaz 4x24 hangot felvettünk, a modellben azonban nem használok mindet, csak a következőket:

- Mély C húr: 7 hang, C1 – F#1 tartomány
- G húr: 5 hang, G1 – H1 tartomány
- Magas C húr: 5 hang, C2 – D#2 tartomány
- F húr: 24 hang, F2 – F4 tartomány

A felvételt úgy állítottuk be, hogy még véletlenül se legyen túlvezérelve a bemenet, a lehetséges tartomány 60-70%-ára méreteztük a mély C húr üresen pengetett hangját, ami jellemzően a legnagyobb kimeneti jellel rendelkezik. Minden hangot addig fogtam le, amíg hallható hangot és a számítógépen jól látható jelet adott. Ez a legmélyebb hangoknál akár 10 másodpercet is jelentett, míg a legmagasabb bundoknál már 2-3 másodperc is elegendő volt. A felvételt 44100Hz mintavételi frekvenciával végeztük. Az így felvett hangokat ezután hangonként külön vágtam, majd .wav kiterjesztésű fájlkként letároltam.

Mivel a modellparaméterek ezekből a hangokból lettek meghatározva, a szintetizátor így legfeljebb 42 hangot tud lejátszani. A legmélyebb hang C1 (33 Hz

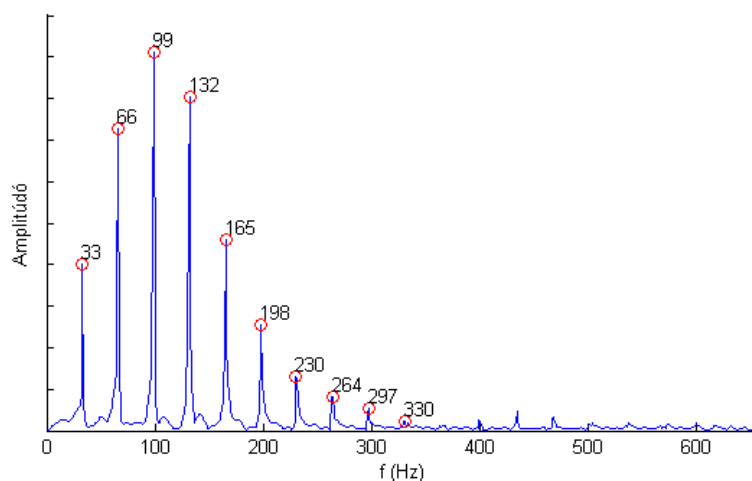
alapfrekvencia), a legmagasabb F4 (350 Hz alapfrekvencia). A felvétel során a húrt pengetővel gerjesztettem.

Később ezeket a hangmintákat MATLAB-bal beolvastatva, megfelelő függvényekkel kinyerhetők a húrmodell paraméterei.

5.1.2 A módusfrekvenciák megkeresése

Az első és legfontosabb analízis feladat a létrehozandó hang módusfrekvenciáinak megtalálása. Erre az információra csak a hangok vizsgálatánál, a heterodin szűréshez van szükség. Ha a felvételkor nem az adott hang névleges alapfrekvenciáján rezgett a húr, a spektrumbeli eltolódás hibás lecsengési idők kiolvasásához vezethet. Később, a szintézis során ezekkel a lecsengési időkkel már a 440 Hz-es hangmagasság-szabvány szerinti frekvenciát számol a program.

Kiindulási alapként tudjuk, hogy a gitárhang szinuszos frekvenciák összegéből áll, illetve ideális szinuszjel spektruma egy darab vonal a jel frekvenciájánál. Tehát a vizsgálandó jel spektruma vonalas lesz, ezeken a helyeken lesznek a módusok frekvenciái.



5.2. Ábra: üresen pengetett mély C húr (33Hz alapfrekvencia) spektruma és első 10 módusának frekvenciái

Az 5.2. ábra alapján megállapítható, hogy egyrészt a felvétel jó minőségű lett, a spektrumban zaj alig van jelen, másrészt pedig hogy hiába számoljuk 50 módusra a szintetizált hangot, már 9-10 módus után alig láthatók az eredeti jel lokális amplitúdómaximumai. Ettől függetlenül természetesen mind az 50 harmonikust feldolgozzuk, ezek a gyors lecsengésük ellenére is fontos elemei a hangzásnak.

További fontos jelenség, hogy diszperzió nem látható a spektrumban, az összes maximum az alapfrekvencia egész számú sokszorosánál található (± 1 Hz). Ezek együtt azt hordozzák magukban, hogy a spektrum elemzése során elég csak az alapfrekvenciát megkeresni, ami könnyen meghatározható zajmentes környezet lévén, majd a lecsengési idők megállapításakor annak a k -szorosai lesznek a magasabb módusok frekvenciái. Az analízist végző függvény még akkor is megfelelő értékeket olvashat ki, ha esetleg 1-2Hz különbség lenne a tényleges és a vizsgált csúcs frekvenciája között, mivel bár az ekkor használt ablakozás igen keskeny, ekkora sávot még átereszt.

A függvény működése a következő: bemenetként szüksége van az vizsgálandó adatsorra, az eredetileg várt f_0 alapfrekvenciára, a módusszámra és az adatsor mintavételi frekvenciájára. Ezekkel az adatokkal rendelkezvén a függvény egy egyszerű FFT transzformációt hajt végre az adatsoron, majd ennek abszolútértékét véve megkapjuk az 5.2. ábrán is látható spektrumot. Az FFT függvény után a spektrum olyan hosszú, amilyen a bemeneti adatvektor volt. Ebből következik, hogy minél hosszabb mintán hajtjuk végre, annál pontosabb lesz a kapott spektrum. Alapvetően problémás lehet a későbbi, már lecsengett módusok következtében megnövekedett jel-zaj viszony, azonban a felvett hangok minősége és alacsony zajszintje miatt ebben az esetben ez a hatás számottevően nem befolyásolja a pontosságot.

Ezután a függvény a MATLAB beépített *findpeaks* függvényét használva megkeresi a spektrum lokális maximumait, majd a talált értékeket kimenti egy vektorba. A lecsengési idők számítása során az így kinyert alapharmonikust használom.

5.1.3 A lecsengési idők kiszámítása

Miután megvannak a módusfrekvenciák, ki kell számolni azok lecsengési időit. Az erre használt függvény első feladata a módusfrekvenciák lecsengéseinek megmérése. Ehhez az FZ-ARMA (Frequency Zooming Auto-Regressive Moving Average) módszert alkalmazom [Karjalainen et al., 2002]. Az analízis a következőképp működik:

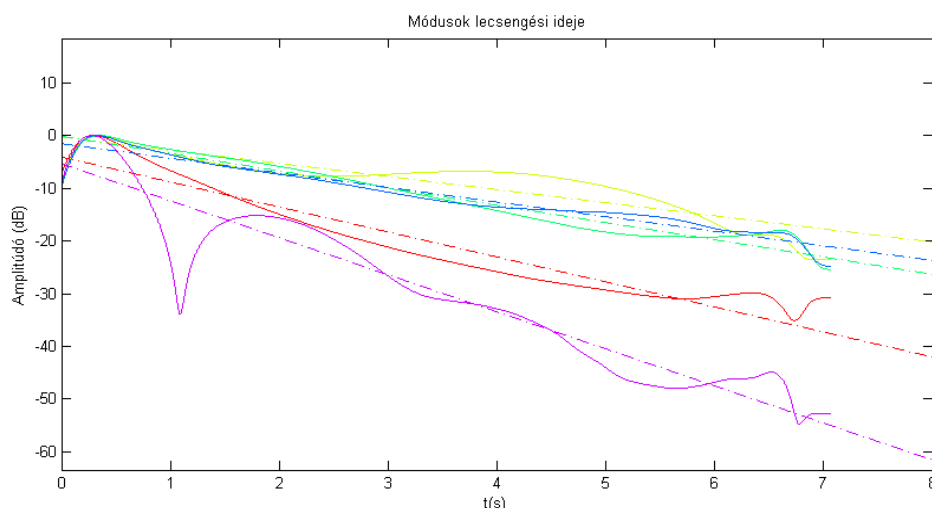
1. A függvény bemenetként megkapja az vizsgálandó hangot, és annak alapfrekvenciáját
2. Az éppen keresett módus frekvenciáját DC-re keveri
3. Igen keskeny sávú aluláteresztő szűrővel oda-vissza megszüri a kapott jelet, ezzel megtartva a fázisát

4. A megmaradt adat az aktuális módus amplitúdómenete, erre egyenest illeszt
5. Megkeresi, mikor lépi át a határfrekvenciát.

A bemenet a felvett és hangokra vágott minta. A vágásnál figyeltem, hogy jó közelítéssel a pengetés megkezdése legyen a minták elején. Ezután következik a DC-re keverés. Tudjuk, hogy a spektrum f_0 -al való eltolásához az időtartományban $e^{j2\pi f_0 t}$, azaz $e^{j2\pi \frac{f_0}{f_s} n}$ taggal kell megszorozni. Ezt a szorzást elvégezve a bemeneti mintánkra, a vizsgált módus frekvenciája DC-be kerül. Ezután aluláteresztő szűrni kell a jelet olyan vágási frekvenciájú szűrővel, hogy a következő módus már ne zavarjon a jelbe, de az aktuális módus frekvenciacsúcsát a lehető legkisebb mértékben torzítsa. Mivel a csúcsok kb. f_0 távolságra vannak egymástól, ami a legmélyebb hang esetén alig 33 Hz, igen szűk sávot szabad csak átengedni. Több próbálkozás után végül $\frac{1}{25}f_0$ széles szűrő megfelelőnek bizonyult. Egy egyszerű másodfokú Butterworth szűrőt terveztettem a MATLAB szűrőtervezőjével, majd a beépített *filtfilt* utasítással megsűrtem a jelet. Utóbbi azért szükséges, mert a sima *filter* utasítás nem lineáris, ami problémát jelentene a spektrumban. Ezt a *filtfilt* úgy küszöböli ki, hogy miután megsűrte a jelet, visszafele is elvégzi ugyanezt, így az eredeti szűrőhöz képest nulla fázistolású, ám négyzetes amplitúdómenetű szűrést hajt végre.

Az így kapott komplex adatsor amplitúdója megegyezik a vizsgált módus időbeli lecsengésével. Ezt a függvényt átkonvertáljuk dB-be, mivel az exponenciális lecsengés logaritmikus skálán lineáris lesz. A valós lecsengés általában nem ténylegesen exponenciális volta miatt még nem számolhatjuk közvetlenül ebből a lecsengési időket, előtte még egyenest illesztünk a kapott amplitúdómenetre a MATLAB beépített *polyfit()* és *polyval()* függvényei segítségével (5.3. ábra).

A felvett minták tisztasága és a basszusgitárom jó hangkitartása következtében a megvágott hangminták megfelelőnek bizonyultak, az illesztő egyenest nem torzította komolyabb mértékű zaj.



5.3. Ábra: Az első öt módus lecsengése és a rájuk illesztett egyenesek (mély C húr üresen pengetve, 33 Hz alaphfrekvencia)

Az algoritmus erre az illesztett egyenesre megkeresi azt az időértéket, amikor az egyenes amplitúdója eléri kezdeti értékének $\frac{1}{e}$ szeresét, azaz -8.7 dB-t. Hibás analízis során előfordulhat, hogy az egyenes sosem metszi ezt a pontot, ilyenkor a függvény 50 másodperc értéket határoz meg. Ha bármelyik vizsgált módusra ezt kapnánk, akkor valószínűleg frekvenciacsúcs helyett zajt talált az algoritmus, ilyenkor kézzel külön meg kell vizsgálni az adott hangot.

Ez a függvény végül egy .txt fájlba írja az így kiszámított lecsengési időket, melyeket később a paraméterek offline generálását végző MATLAB szkript használ fel.

5.2 A gerjesztési paraméterek meghatározása

5.2.1 Pengetővel való gerjesztés

A pengető és a kéz tömege viszonylag egyszerűen mérhető, ha eltekintünk attól, hogy a kézfej és a kar tömege összefügg. m_p értéke 5-10g körül változik, míg m_{hl} nagyjából 0.4-0.6 kg, az egyszerűség kedvéért a használt érték 0.5 kg lett. A rugóállandók meghatározása már nem triviális feladat, becslés és MATLAB szimulációk révén határoztam meg ezeket. $K_p = 100$ N/m, $K_{hl} = 500$ N/m adódott, míg a pengetés sebessége gyors játékot véve alapul kb. $v_{01} = 0.2-0.25$ m/s. Ehhez feltételeztem, hogy átlagosan 5 cm-t mozdul a kezem és benne a pengető a kitérés végpontjai között. 120 BPM sebességű zenében nyolcadokat játszva ez percenként 240 pengetést jelent. $240 \cdot 0.05/60 = 0.2$ m/s,

ez a tempó pedig közepesnek számít a rockzenék között. Végül a fenti intervallum közepét, azaz 0.225 m/s értéket használok.

A húr-pengető kapcsolat megszakadásához szükséges erőértéket szimulációkkal állapítottam meg. Mivel a pickup nemlinearitását tangensfüggvénnyel modellezem, a maximális megengedhető amplitúdó $\tan(1) = 0.7854$. A szimulációk során ehhez kb. 0.6N határértéket találtam megfelelőnek. A VST plugin vezérlésekor a MIDI adatok között található *velocity* érték azt jelzi, milyen intenzitással szólaltatták meg a hangot. Pengetős gerjesztés esetén ezt a 0.6N határértéket osztom le a bejövő *velocity* függvényében.

5.2.2 Slap bass gerjesztés

Ilyenkor a húrral csak a hüvelykujj kerül kölcsönhatásba, de természetesen az egész kéz közrejátszik a folyamatban. Itt is becslés és szimulációk alapján tudtam meghatározni a használt értékeket. Kiindulási alapként megfigyeltem, pontosan hogyan csapok a húrra. A játékmódom során a hüvelykujjam alsó és középső ujjpercének találkozásánál, a felső ujjperc csontjának kiszélesedésénél érek a húrhoz. Ez a pont hozza létre a legkisebb erő kifejtéssel a legnagyobb húrkitérést, így feltételeztem, hogy a K_{h2} rugóállandó a pengetős modellénél nagyobb lesz. Értékét végül 600-800 N/m nagyságrendben állapítottam meg, a modellbe az alsó határát építettem be.

A hüvelykujj tömegét a már korábban megállapított kéz tömegéből vezettem vissza, ám az alkarból és a csuklóból eredő forgatónyomaték és az izmok feszsége komoly mértékben megváltoztatják a hüvelykujj modellezendő tulajdonságait. Az ujjak nagyjából a teljes kéz tömegének negyedét-ötödét adhatják, de a fentiek miatt nem vehetünk a gerjesztés során egyetlen „levágott” ujjat. Végül szimulációk után m_{h2} értékét 0,1 kg-nak vettem. A gerjesztés sebessége a másfajta kézmozgások miatt gyorsabb, mint pengetős esetben, szintén szimulációkra támaszkodva maximum $v_{02} = 0.5$ m/s került a programba. A *velocity* üzenet ebben az esetben ezt a sebességet változtatja, míg a magas húrok feltételezése során a kezdetiérték amplitúdóját.

5.3 A kinyert adatok implementációja

Az analízis során μ , L , f_k és τ_k értékeket kaptunk, ezeket nevezhetjük a „legmélyebb szintű” paramétereknek, amik a húr differenciálegyenletében közvetlen szerepelnek. Azonban a (3.3.2) egyenletben látszik, hogy ezek az értékek közvetlenül

nem szerepelnek a húr számításaiért felelő algoritmusban, csak az $a_{1,k}$, $a_{2,k}$, b_k paraméterek. Ezek a „második szintű” paraméterek, amikkel a program a számításokat végzi. Végül a „harmadik szintű” paraméterek a felhasználó által vezérelhetők, a programomban ezek a gerjesztés tulajdonságai, illetve a hangszedők pozíciói. Ezeket a plugin működése közben is változtathatja a felhasználó, ilyenkor a program a következő fejezetben ismertetett függvények segítségével újraszámolja a szükséges változókat.

A „második szintű” paraméterek kiszámítása offline módon történik. Mivel adott hangszer hangját állítom elő, nem pedig általános modellt implementálok, így számítási kapacitást spórolhatok a valós idejű futtatás során. Ezeket MATLAB segítségével számítom, az ehhez írt függvény bemenetén megkapja az elemmezendő hangfájlt, a hozzá tartozó lecsengési idő értékeit, illetve azoknak a fájloknak az adatait, amibe a kiszámított értékeket menti.

Miután a program lefutott, minden hanghoz 3 generált fájl tartozik, különböző mappákban, *<MIDI_note>.txt* névvel. Ezek a fájlok tartalmazzák rendre a_1 , a_2 és b paraméterek 50 módusra kiszámított értékeit, pl. a C1 hanghoz (MIDI note 24) tartozó b paramétereket a *..\b\24.txt* fájlba menti. A szintetizátor program által használt könyvtárszerkezet a következő: *Plugins\VST\BassSynth\<paraméter>\<MIDI_note>.txt*. Így minden fájl 50 sort tartalmaz, a sorszámnak megfelelő módus megfelelő paraméterének értékével.

Ezen paraméterek kiszámítása az előző fejezetekben leírt módszerek szerint történik, a VST plugin pedig ezeket indításkor dolgozza fel, amikor meghívja az egyes hangokhoz tartozó konstruktorokat.

6 VST Implementáció

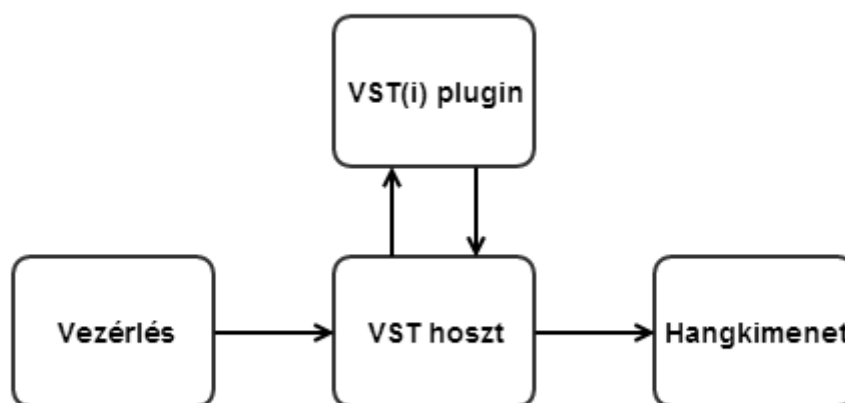
6.1 VST alapok

A Virtual Studio Technology nevű szoftveres interfészt a Steinberg cég fejlesztette ki. A C vagy C++ nyelven írt VST(i) pluginek a hoszt program és az effektek (VST) vagy szintetizátorok (VST Instrument, VSTi) közötti kapcsolatot biztosítják. A hoszt program bármi lehet, ami fel van készítve VST kezelésre, leggyakrabban ezek a bevezetőben említett DAW-ok.

Ezek a munkaállomások lényegében szoftveres hangstúdiók, ezekkel és egy jó minőségű hangkártyával hangszerek vehetők fel, szintetizátorok segítségével MIDI sávok programozhatók, illetve az elkészült mű teljes utómunkája, keverése is elvégezhető a használatukkal.

Alapvetően háromféle VST plugin létezik, effektek, MIDI effektek és hangszerek pluginjei. A MIDI effekt pluginek a beérkező MIDI adatokat dolgozzák fel és továbbítják, a VST effektek egy, már meglévő jelet dolgoznak fel funkciójuk szerint, a VSTi hangszerek pedig a MIDI vezérlés alapján hoznak létre új hangokat. Ezek általában szintetizátorok vagy samplerek, ilyen plugin a szakdolgozatom programja is.

A virtuális hangszerek vezérlése egyszerű, vagy hoszt által generált, vagy külső forrásból kapott MIDI adatokat feldolgoz a plugin, majd a hoszt felé visszaadja a kimeneti jelalakot. A folyamat fekete dobozként látszódik a felhasználó részéről, csak adott számú be- és kimenet, illetve az állítható paraméterekhez van hozzáférésük. A plugin csak a hanggenerálással foglalkozik, minden mást, beleértve a hangkártyával való kommunikációt is, a hoszt végez (6.1. ábra).



6.1. Ábra: VST pluginek működése

6.2 A MIDI kezelés

Szintetizátor vezérléséhez a hoszt program a beérkező MIDI adatokat használja fel. Alapvetően három fontos üzenettípust használunk: a Note On, Note Off és All Notes Off üzeneteket. Egy nem rendszerszintű MIDI esemény 2 vagy 3 byte hosszú, az első a státuszbájt, a többi a státusz típusához tartozó további adatokat szolgáltatja (6.2. ábra).

Státusz D7----D0	Adatbájt(ok) D7----D0	Leírás
1000nnnn	0kkkkkkk 0vvvvvvvv	Note Off Hang kikapcsolását jelentő üzenet. k a hangot határozza meg, v a hangerőt (velocity érték).
1001nnnn	0kkkkkkk 0vvvvvvvv	Note On Hang bekapcsolását, a vezérlőbillentyű lenyomását jelenti. k a hang azonosítója, v a hangerő.
1011nnnn	0ccccccc 0vvvvvvvv	All Notes Off Minden hangot kikapcsol, ha $122 < c < 128$, illetve $v=0$.

6.2. Ábra: Szükséges MIDI események és leírásuk (forrás:

<http://www.midi.org/techspecs/midimessages.php>)

Létezik még másfajta üzenet is, pl. ami modulációs kerék állását jelzi, vagy a pitch bend, ami a hang nyújtására, azaz a lefogott hang kismértékű frekvenciaváltoztatására vonatkozó adatokat tartalmaz, de ezeket a megvalósított program nem kezeli.

A fenti adatok észlelésekor a program meghívja a szükséges belső függvényeket a bemeneti információk alapján. Fontos, hogy néhány hoszt a Note Off üzenetet úgy határozza meg, hogy az egy 0 hangerejű Note On üzenetként kerül továbbításra, ezért ezt az esetet külön meg kell vizsgálni. All Notes Off esemény hatására értelemszerűen az összes aktuálisan működő szintetizátorra kell végigfuttatni a hang kikapcsolását vezérlő függvényt.

6.3 A szintetizátor megvalósítása

Mivel polifóniát szeretnénk megvalósítani, szükség van egy objektumra, ami az aktuálisan játszott hang adatait kezeli, illetve egy tömbre, ami ezeket tartalmazza. Két megközelítési mód lehetséges, vagy az éppen generált szintetizátorokat tároljuk egy négyelemű tömbben (ennyi húr – és emiatt hang – szólhat egyszerre), vagy a plugin betöltésekor létrehozunk egy tömböt, ami az összes lehetséges hang adatait tárolja, majd ezekből választjuk ki, melyek szólhatnak éppen. A megvalósítás során utóbbi módszert alkalmazom, mivel ilyenkor elég egyszer beolvasni a paramétereket, nem szükséges minden meghívott hangnál ezzel is terhelni a processzort.

6.3.1 A változók

Mindegyik szintetizátorobjektumnak tartalmaznia kell az adott hang helyes működéséhez szükséges információkat, illetve a modellhez tartozó globális értékeket is, mint a vezérlés információi. Ezeken kívül két fontos *bool* típusú változó jelöli, hogy éppen működik-e az adott hang szintetizátora (*synthIsOn*), illetve hogy a kezdeti érték beállítása után vagyunk-e, azaz a húr szabadon rezeg, vagy sem (*noteIsOn*).

6.3.2 A függvények

A VST rendszer a MIDI vezérlés kezelése és a hoszttal való kommunikáció mellett három fő függvényt alkalmaz, ezek, és a funkciók helyes működéséhez szükséges, fejlesztők által írt további függvényekkel dolgozik a plugin. A következőkben ezeket mutatom be.

- `void initProcess()`

A hoszt egyszer hívja meg, a plugin betöltésekor. Ez a függvény végzi el az összes fontos, egyszeri beállítást, létrehozza a szintetizátorobjektumok tömbjét, a globális konstansokat, valamint beállítja a program gyári paramétereit.

- `void process(float* *inputs, float* *outputs, VstInt32 sampleFrames)`

VST-k fő függvénye, ebben történik a bemeneti adatok feldolgozása. Fő tulajdonsága, hogy a kimenetéhez hozzáadja a feldolgozott adatokat, nem pedig felülírja azt. Mivel a szintetizátoroknak nincs audiobemenetük, ezt a függvényt nem implementáljuk.

- `void processReplacing(float* *inputs, float* *outputs, VstInt32 sampleFrames)`

VSTi-k jelgenerálásért felelős függvénye, a *process()* „testvére”. Legfőbb tulajdonsága, hogy a kimenetet felülírja a jelgenerálás, jelfeldolgozás során kapott kimeneti adatokkal, így nincs feltétlen szüksége audiobementre. Ha egy bemeneti jellel dolgozó plugint fejlesztünk, a VST2 szabványa szerint a *process()* függvényt mindenképp implementálni kell, a *processReplacing()* függvényt pedig ajánlott. Általában ez a *process()* függvény egyszerű másolásából és néhány sor módosításából megkapható.

A többi, belső kommunikációt megvalósító függvény ismertetésétől eltekintek, érdekesebb a programom megvalósításához szükséges adatfeldolgozó függvények működése:

- `Synth(VstInt32 note)`

A konstruktor függvény. A plugin betöltésekor hívja meg az *initProcess()*, a szintetizátorokat tároló tömbökbe kerülnek. Bemeneti paramétere a MIDI note értéke, ami a létrehozandó hangot határozza meg. Elsőként az előző fejezetben bemutatott könyvtárszerkezetből olvassa be a hanghoz tartozó paramétereket, majd betölti a bundok helyzetét és értékeit, illetve kiszámolja a számítási igényt csökkentő, bonyolultabb konstans szorzókat.

A slap játékszerűséhez tartozó kétféle gerjesztés közül az alapján választunk, hogy melyik húron lévő hangot hívja meg a *noteOn()* függvény. A program – a paraméterkezeléséből adódóan – jelenlegi állapotban azt figyeli, hogy *note* értéke meghaladja-e 35-öt. Ez ugyanis a G húr 4. bundja, ami a megvalósított modellben a két mélyebb húr legmagasabb hangja. Eddig a hüvelykujj húrnak csapásával gerjesztjük a húr, az értéket meghaladva pedig a húr feltépésével. Egy belső bool típusú változó jelzi az algoritmusnak, melyik megoldást kell számítani.

- `void noteOn(int velocity, int dFrames, int dBundszám)`

A megfelelő MIDI üzenet hatására bekapcsolja az adott szintetizátort a *synthIsOn* paraméterrel, illetve átalakítja a 7 bites egész *velocity* értéket 0 és 1 közötti paraméterre. Az így kapott, 0 és 1 közé eső értékkel az aktuális gerjesztés megfelelő változóját skálázza. A *dFrames* változó a generálandó blokk méretét határozza meg.

- `void noteOff()`

A hang kikapcsolásakor nullázza, illetve alaphelyzetbe állítja a húr és a gerjesztés változóit, illetve a *synthIsOn* paramétert kikapcsolja.

- `void fExcMod(float newfExc)`
- `void fOut1Mod(float newfOut1)`
- `void fOut2Mod(float newfOut2)`

A gerjesztés, illetve a hangszedők pozícióját, majd a relatív értékeit állítja be. A paraméterek változtatásért felelős függvény hívja meg ezeket, de mivel több, belső változót is módosítanak, külön implementációt választottam.

- `void generate(float* out, float sampleRate)`

Itt történik a húrmodell futtatása. Először ellenőrzi az objektum paramétereit, majd annak megfelelően viselkedik. Új hang esetén a kezdeti érték-beállítást hajtja végre, majd a pengetés módjától függően futtatja a fő algoritmust, azaz vagy figyeli a húr bundnak csapódásait, vagy szabadon rezeg. Kimenete a húr módusainak a hangszedők elhelyezkedésével és átlagoló hatásával súlyozott értéke.

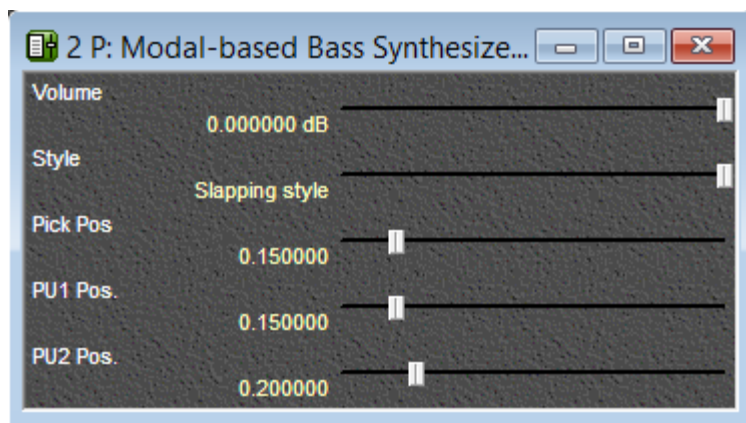
Fontos megjegyezni, hogy ilyenkor a húr kitérése a fizikai mennyiségekből adódóan méter dimenziójú lesz, nagyságrendileg néhányszor 10^{-3} értékekkel. Mivel a pickup hatásai során a tangensfüggvény 0-1 tartományára van szükségünk, illetve a hangkártya is ekkora jelet vár, a kapott kimenetet át kell váltani a legnagyobb kitéréshez viszonyított relatív értékre. A modellemben maximális kitérésnek 5 mm-t vettem, ez a programban egyszerűen a kimenet $5 \cdot 10^{-3}$ értékkel való osztásában nyilvánul meg.

- `void pickup(float* out, float sampleRate)`

A pickup hatásait modellezi, bemenete a húrmodell által generált kimenet. E függvény kimenete lesz végül a plugin kimenete is.

6.3.3 Felhasználói paraméterek

A felhasználó a szintetizátor négy paraméterét állíthatja: A gerjesztés fajtáját (pengető vagy slap), annak a pozícióját (x/L arányban), illetve a hangszedők pozícióját (szintén x/L arányban). A program jelen állapotában saját kezelőfelületet nem kapott, a hoszt által biztosított lehetőségekkel (csúszka, tekerő) lehet ezeket beállítani (6.1. ábra).



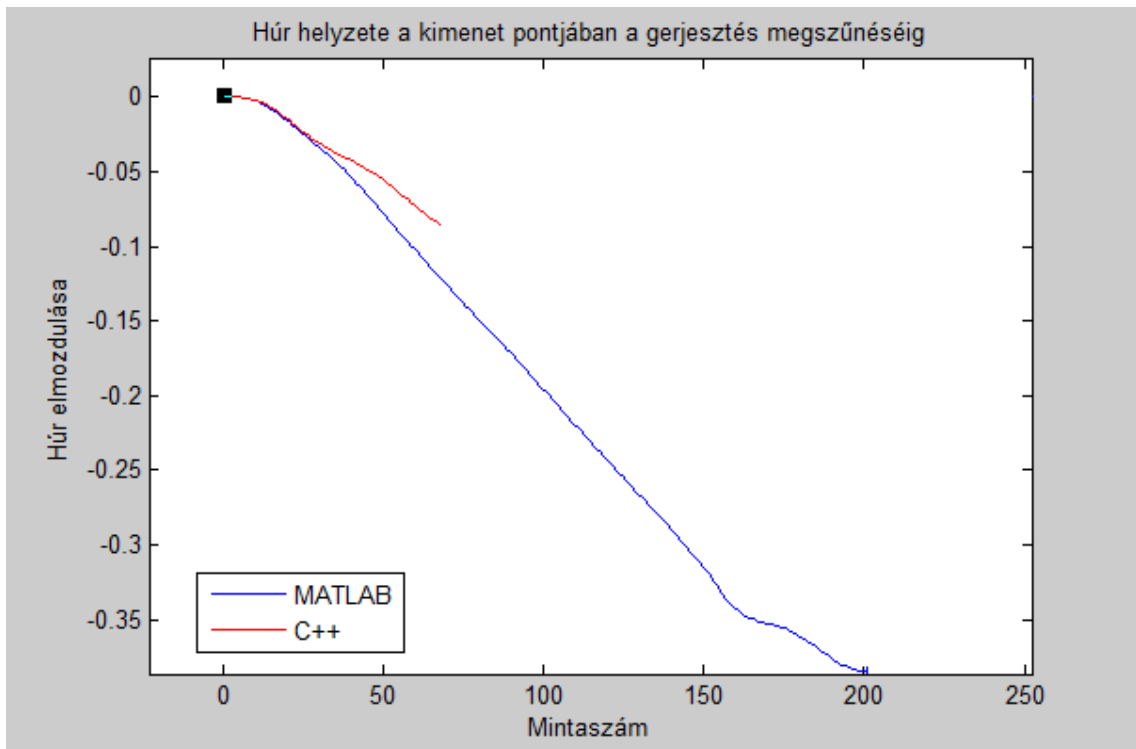
6.1. Ábra: Paraméterek a VST Host programban

A VST kezelés főbb alkotóelemei a fentiek, a program működéséhez C++ nyelvre kell portolni a MATLAB-ban megírt függvényeket. A plugin vázát a Steinberg által készített *VstXSynth* nevű mintaszintetizátor adta, teszteléshez az ingyenes MiniVSTHost és VST Host nevű programokat használtam. A pluginnak felépítéséből adódóan elvileg bármilyen DAW-ból meghívva működnie kell.

6.4 Hibák

A VST implementáció nem sikerült problémamentesen, maradt néhány eddig kijavítatlan hiba az elkészült programban.

A legnagyobb és legfontosabb hiba slap bass esetén az alsó húrok gerjesztésénél jelentkezik. A plugin sokkal hamarabb éri el a gerjesztés erejének határértékét, mint a MATLAB szimuláció.



6.2. Ábra: A szimuláció (kék) és a plugin (piros) kimenetének értéke amíg eléri a gerjesztés határát

A 6.2. ábrán jól látható, hogy a plugin és a szimuláció néhány mintáig együtt fut, majd a program kimenete elmarad a pluginétól, a gerjesztés határát pedig kb. harmadannyi idő alatt éri el.

A hiba okát nem sikerült kideríteni, a MATLAB és a C++ kód az előbbiben lévő, főleg az újraszámolások kivételével teljesen megegyezett, mintáról mintára vizsgálva sem találtam eltérést a futásaikban. A plugin float, míg a MATLAB double típusú számokkal dolgozik, de ez ekkora eltérést nem jelenthet a két eredmény között.

Emiatt a húr sokkal kisebb kezdőamplitúdójú lesz a szabad rezgésekor, így a kimenetre olyan kicsi érték jut, hogy az gyakorlatilag nem hallható.

További hiba szintén slap modellezésnél, hogy a magasabb húrok gerjesztésekor néhány hang a megfelelő *velocity* érték mellett egyszerűen túlvezérli a kimenetet, fehérzaj jelenik meg a kimeneten. Ez némely hangnál a *velocity* egy határa felett jön létre, azonban van, hogy akár két fehérzajt létrehozó érték között is létezik olyan tartomány, ahol megfelelően működik a szintézis. Ezeket a hangokat mintánként elemezve sem találtam meg, melyik változó(k) vesz(nek) fel hibás értéket a folyamat során.

Végül megemlítendő még a hangok elengedésekor jelentkező kattanás, ami a hirtelen megszűnt kimenet miatt hallatszik. Ez a jövőben a *noteOff* esemény bejövetelekor pl. gyors elhalkulás implementálásával orvosolható.

7 Értékelés, továbbfejlesztési lehetőségek

A szakdolgozat elsődleges célja a basszusgitárom hangjának fizikai alapú modellezése volt, amit valós időben, VST pluginként lehet futtatni. További célként a slap bass technika implementációja volt kitűzve, amiről eddig igen kevés irodalom található.

7.1 Értékelés

A kész program a C1-F4 tartományban képes hang létrehozására a basszusgitárom felvett hangjaiból kinyert adatok által, a kitűzött célokat azonban nem sikerült tökéletesen elérni. Pengetővel való gerjesztés esetén a hangminőséget megfelelőnek találtam, ugyan nem adja vissza teljesen a felvett hangok élményét, de egy kellemes hangú, jól működő szintetizátor készült.

Slap bass technika modellezésekor két problémába is ütköztem, amik MATLAB szimuláció során nem jöttek elő: a mély húrok hangszer felé csapása nem olyan gerjesztést hozott létre a pluginben, mint a szimulációban, illetve a magas húrok feltépésekor néhány hang bizonyos *velocity* értékek mellett begerjed, a kimeneten fehérzaj jelenik meg. Mindezek ellenére azok a hangok, amelyek megfelelően szólnak, visszaadják a slap hangzását, élményét.

A dolgozat elkészítése során sikerült kifejleszteni egy modális alapokon nyugvó modellt, ami figyeli a húr útjába kerülő akadályok, jelen esetben bundok esetén fellépő erőhatásokat, és képes valós időben kiszámolni az ezek által megváltoztatott húrmozgásokat. Ezek szintén fizikai alapokon nyugszanak, de sok helyen szükség volt egyszerűsítésekre, vagy a számítások bonyolultsága, vagy azok erőforrásigénye miatt.

A piacon fellelhető basszusgitár szintetizátorok között tudtommal nincs olyan, ami nem előre felvett minták segítségével dolgozna és képes lenne a slap technika modellezésére. Az elkészült programom hangminősége kielégítő, hangzása még nem éri el az általam ismert szintetizátorokét, de további finomításokkal és a hibák kijavítása után akár jobb is lehet ezeknél.

Az általam ismert irodalomban fizikai alapú slap bass modellezésről egyetlen cikket találtam, ami a probléma megoldását waveguide alapokon mutatja be. Ebből

kifolyólag úgy tűnik, hogy az ebben a dolgozatban közölt módszer az első modális szintézisen alapuló slap bass modell. Érdekes lehet mélyebben megismerni a technika használata során végbemenő folyamatokat és kidolgozni egy minden fontosabb részletre kitérő megoldást.

7.2 Továbbfejlesztési lehetőségek

Elsősorban az előzőekben említett hibák kijavítása a fő feladat, különös tekintettel a slap bassnél felmerülő furcsaságokra. Ezek kijavítása után lehet fontolóra venni a fejlesztési lehetőségeket.

A legfőbb továbbfejlesztés elsősorban a bundok (illetve a későbbiekben a fogólap) hatásainak modellezésénél található. Figyelembe lehetne venni a fellépő kölcsönhatás nem ideálisan rugalmas voltát, a fogólap jelenlétét, a húr útjába kerülő különböző anyagok eltérő hatásait. Megoldandó probléma még a lefogott hangok esetén a húr és a bundok relatív pozíciójának megváltozása.

Másik fejlesztési irány lehet a húr rezgésének két, transzverzális polarizációs volta. A megalkotott modellben a húr egy síkban rezeg, ami vagy párhuzamos a hangszedőkkel (pengetős gerjesztés), vagy merőleges arra (slap bass). Valós esetben azonban ez nem így van, ami pl. a pickupok kimenetének nemlinearitását is okozza.

A hangszedők pontos mérése, átvitelük pontosabb meghatározása szintén hangminőségbeli fejlődést vonhat maga után. Ezzel a dolgozat megalkotásakor nem foglalkoztunk, csak a fellelhető irodalom alapján építettem be a hatásokat a modellbe. Ugyanígy eltekintettünk a hangszertest rezgésétől, ezek a húrlábon és a nulladik bundként értelmezhető nyergen keresztül visszajuthatnak a húrmodellbe.

Végül, de nem utolsó sorban, egy jó és hatékony bundmodell létrehozása után el lehet gondolkodni a tapping játéktílus – amikor a zenész erősen a fogólapnak nyomja a húrt valamelyik hangnál, majd elengedi azt, a megszokott pengetés helyett ezzel gerjesztve a rendszert – megvalósításának lehetőségein, amiről szintén nem találtam még irodalmat. Ennek a megalkotása után az összes lényegesebb játéktílus modellezhetővé válna, a modális szintézissel problémásan megvalósítható, folyamatos frekvenciaváltozással járó technikák (pl. nyújtás és slide) kivételével.

Irodalomjegyzék

- Arfib, D. (1978). *Digital Synthesis of Complex Spectra by Means of Multiplication of Non Linear Distorted Sine Waves*, AES Convention 59 (February 1978), Paper number: 1319
- Adrien, J.M. (1991). The missing link: *Modal synthesis*, in G. De Poli, A. Piccialli and C. Roads (eds), *Representations of Musical Signals*, The MIT Press, Cambridge, Massachusetts, USA, pp. 269–297.
- Bank, B. (2000). *Physics-Based Sound Synthesis of the Piano*, Master's thesis, URL: <http://home.mit.bme.hu/~bank/thesis/pianomod.pdf>
- Bank, B. (2006). *Physics-based Sound Synthesis of String Instruments Including Geometric Nonlinearities*, Ph.D. Thesis, URL: <http://home.mit.bme.hu/~bank/phd/phd.pdf>
- Csaba, A.Z. (2002). *Elektromos gitár hangjának valós idejű szintézise*, diplomamunka, URL: <http://dsp.mit.bme.hu/userfiles/diploma/csabadiploma02.pdf>
- D'Addario 1. *String Tension Guide*, URL: <http://www.daddario.com/DAStringtensionguide.Page?AxPageID=2371&Mode=0&ActiveID=2871>
- Fizipedia 1. *Állóhullámok megfeszített, rugalmas húrban*, URL: http://fizipedia.bme.hu/index.php/%C3%81ll%C3%B3hull%C3%A1mok_megfesz%C3%ADtett,_rugalmas_h%C3%BArban
- Fodor, Gy. (1996). *Elektromágneses terek*, Műegyetemi kiadó, 55019
- Garamvölgyi, Zs. (2006). *Dobhang fizikai alapú szintézise*, TDK dolgozat, URL: <http://dsp.mit.bme.hu/userfiles/tdk/garamvolgyitdk06.pdf>
- Hiller, L. and Ruiz, P. (1971). *Synthesizing musical sounds by solving the wave equation for vibrating objects: Part 1 & Part 2*, J. Audio Eng. Soc. 19(6) & 19(7)
- Karjalainen, M. and Erku, C. (2004). *Digital waveguides versus finite difference structures: Equivalence and mixed modeling*, EURASIP J. on Appl. Sign. Proc. 2004(7): 978-989.
- Karjalainen, M. et al. (2002). *Optimization techniques for parametric modeling of acoustic systems and materials*, Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing, Finland
- Le Brun, M. (1979). *Digital Waveshaping Synthesis*, JAES Volume 27 Issue 4 pp. 250-266.
- Márkus, J. (1999). *Orgonasípok hangjának jelmodell alapú szintézise*, diplomaterv, URL: <http://dsp.mit.bme.hu/userfiles/diploma/markusdiploma99.pdf>

- Morse, P.M. and Ingard, K.U. (1968). *Theoretical Acoustics*, McGraw-Hill
- Paiva, R.C.D., Pakarinen, J. and Välimäki, V. (2012). *Acoustics and modeling of pickups*, J. Audio Eng. Soc. 60(10), p. 768-782.
- Rank, E. and Kubin, G. (1997). *A waveguide model for slapbass synthesis*, Acoustics, Speech, and Signal Processing, 1997 IEEE International Conference, p. 443-446. vol. 1, Munich
- Russell, D.A. (2012). *Physics of Guitar Pickups*, URL: <http://www.acs.psu.edu/drussell/guitars/pickups.html>
- Smith, J.O. (1983). *Techniques for Digital Filter Design and System Identification with Application to the Violin*, PhD thesis, Stanford University, California, USA
- Smith, J.O. (1987). *Music Applications of Digital Waveguides*, Technical Report STAN-M-39, CCRMA, Dept. of Music, Stanford University, California, USA, p. 181.
- Smith, J.O. (1991). *Viewpoints on the history of digital synthesis*, Proc. Int. Computer Music Conf., Montreal, Canada, pp. 1-10.
- Smith, J.O. (1992). *Physical modeling using digital waveguides*, Computer Music J. 16(4)
- Trautmann, L. and Rabenstein, R. (2002). *Digital Sound Synthesis of String Instruments with the Functional Transformation Method*, University of Erlangen-Nürnberg, Multimedia communications and Signal Processing, Erlangen, Germany
- Wikipedia 1. *Telharmonium*, URL: <http://en.wikipedia.org/wiki/Telharmonium>