



## DIPLOMATERV-FELADAT

**Konyári András Márton (J9MYM0)**

szigorló villamosmérnök hallgató részére

### Hammond orgona hangjának modellalapú szintézise

A harmincas években megalkotott Hammond orgonát eredetileg templomi orgonák helyettesítésére szánták, de hamar a könnyűzenei műfajok kedvelt hangszere lett. A Hammond orgona egy elektromechanikus hangszer, amelyben az egyes harmonikusok jele forgó fogaskerekek által tekercsekben indukált feszültségként áll elő. Az így kapott szinuszos jellegű jelekből egy keverőrendszer állítja elő a kívánt hangzást, tkp. additív szintézist megvalósítva. A hangzásban a hanggenerátoron és a keverőrendszeren kívül a hangszer elektroncsöves előerősítője, a billentyűlételek dinamikájáért felelős „percussion áramkör”, továbbá az elektromechanikus vibrato effekt is fontos szerepet játszik.

Bár a templomi orgonáknál kétségtelenül jóval könnyebben mozgítható, a Hammond orgona így is nehezen szállítható, manapság nehezen beszerezhető, ill. a bonyolult elektromechanikus konstrukcióból adódóan a szervizelése sem triviális. Ennek megfelelően hamar felmerült az igény a legendás Hammond hangzás tisztán elektronikus előállítására is. A korai digitális szintetizátorok esetében ez az orgona tipikus beállításai mellett felvett hangok visszajátszását jelentette, ezzel azonban pont az eredeti hangszer sokoldalúsága veszett el. A modern orgonaszintetizátorok a Hammond orgona minden fontosabb alkotóelemét tartalmazzák, így ezekben az egyes harmonikusok aránya ugyanúgy állítható, és gyakran a kapcsolók kialakítása is az eredeti hangszert tükrözi.

A hallgató feladata egy olyan szintetizátor megalkotása, amely a Hammond orgona minden fontos alkotóelemének modelljét tartalmazza. A hallgató munkájának a következőkre kell kiterjednie:

- Ismerje meg a Hammond orgonák felépítését, tekintse át a modellezési lehetőségeket és az elérhető digitális szintetizátorokat!
- Mérések segítségével tárja fel a hanggenerátorok kimenőjelének jellegzetességeit, a hangzás szempontjából nem elhanyagolható, ideálistól eltérő tulajdonságait, majd alkosson Matlab modellt a hanggenerátorok jelének előállítására!
- A keverőrendszer Matlab modelljét az elérhető áramkörüi rajzok és szükség esetén mérések segítségével készítse el!
- Mérések alapján tárja fel az előerősítő és a „percussion áramkör” viselkedését, különös hangsúlyt fektetve az esetleges nemlineáris hatásokra, és készítse el ezek modelljét!
- A megalkotott Matlab modellek alapján készítsen valós időben futtatható orgonaszintetizátort PC-s vagy beágyazott környezetben!

**Tanszéki konzulens:** Dr. Bank Balázs, docens

Budapest, 2019. március 10.

.....  
Dr. Dabóczi Tamás  
tanszékvezető



M Ű E G Y E T E M 1 7 8 2

**Budapesti Műszaki és Gazdaságtudományi Egyetem**  
Villamosmérnöki és Informatikai Kar  
Méréstechnika és Információs Rendszerek Tanszék

Konyári András Márton

# **HAMMOND ORGONA HANGJÁNAK MODELLALAPÚ SZINTÉZISE**

KONZULENS

**Dr. Bank Balázs**

BUDAPEST, 2020

# Tartalomjegyzék

<b>Összefoglaló .....</b>	<b>6</b>
<b>Abstract.....</b>	<b>7</b>
<b>1 Bevezetés .....</b>	<b>8</b>
1.1 A választott téma aktualitása .....	8
1.2 A klasszikus Hammond modellek és helyettesítőik .....	9
1.3 A felhasznált eszközök .....	10
1.4 A dolgozat felépítése .....	11
<b>2 A Hammond orgona működése .....</b>	<b>12</b>
2.1 Additív hangszintézis.....	12
2.2 Az orgona felépítése .....	13
2.2.1 A <i>tonewheel generator</i> működése .....	14
2.2.2 A keverőrendszer felépítése és működése .....	15
2.2.3 Az opcionális effektek .....	16
2.2.4 Az előerősítő szerepe .....	18
<b>3 A jelgenerátor-egység modellje.....</b>	<b>21</b>
3.1 A generátorok belső ellenállásának mérése .....	21
3.2 A generátorok jelének modellezése .....	23
3.2.1 Jelanalízis.....	23
3.2.2 Jelmodell.....	28
3.2.3 Jelszintézis .....	31
<b>4 A keverőrendszer modellje .....</b>	<b>33</b>
4.1 A harmonikusok súlyozása .....	33
4.2 A <i>key click</i> effektus.....	35
<b>5 A <i>percussion</i> funkció modellje .....</b>	<b>39</b>
<b>6 Az előerősítő modellje.....</b>	<b>43</b>
6.1 Modelltípus kiválasztása.....	43
6.2 A kiterjesztett Wiener-Hammerstein modell .....	44
6.2.1 A modell felépítése .....	44
6.2.2 A modell adaptálása a modellezendő rendszerhez .....	46
6.3 Átviteli függvények mérése .....	48
6.3.1 A logaritmikus sweep mérőjel .....	48

6.3.2 A mérőjel előállítás	49
6.3.3 A mérések kiértékelése	51
6.3.4 Az előerősítő átvitelének mérése	54
6.4 Az erősítőmodell elkészítése	54
6.4.1 Az előszűrő és az utószűrő elkészítése	54
6.4.2 A nemlineáris függvény paramétereinek hangolása	55
6.4.3 Eredmények értékelése	56
6.5 Párhuzamos IIR szűrőrendszer tervezése	57
6.6 Az <i>expression control</i> pedál működése	59
<b>7 Valós idejű alkalmazás fejlesztése</b>	<b>63</b>
7.1 A <i>JUCE</i> platform	63
7.2 Az szoftver felépítése	65
7.3 Az elkészült alkalmazás tesztelése, értékelése	69
<b>8 Összefoglalás, kitekintés</b>	<b>73</b>
8.1 Az elvégzett feladatok	73
8.2 Továbbfejlesztési lehetőségek	75
<b>Irodalomjegyzék</b>	<b>76</b>
<b>Köszönetnyilvánítás</b>	<b>80</b>

# HALLGATÓI NYILATKOZAT

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

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy 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, 2020. 05. 26.

.....  
Konyári András Márton

# Összefoglaló

Egyedi hangzásvilágának köszönhetően a *Hammond orgona* mára a legkülönbözőbb zenei irányzatok kultikus hangszerévé vált, ám szállításának és karbantartásának körülményessége miatt jelentős igény mutatkozik a digitális eszközökkel történő kiváltására. A hangszer sajátosságai okán – a hagyományos szintetizátorok hangfelvételek visszajátszásán alapuló megoldásai helyett – a *modell alapú megközelítés* tűnik célravezetőbbnek, ami figyelembe veszi az orgona működési elvét.

Jelen dolgozatban az elektromechanikus Hammond orgonák modellezésének folyamatát járom körbe a hangszer felépítésének megismerésétől kezdve, a mérésekre alapozott jelanalízis és -szintézis feladatain keresztül, a valós idejű hangszermodell-alkalmazás fejlesztéséig. A modellalkotás folyamán egyrészt egy Hammond A-100 típusú orgonával rögzített hangmintákra, másrészt internetes forrásokra támaszkodom.

Legelőször ismertetem az orgona szerkezeti egységeinek (*tonewheel generator*, *keverőrendszer*, *percussion áramkör* és *előerősítő*) szerepét, illetve működését. Ezután az alkatrészek MATLAB modelljeinek elkészítéséről lesz szó, kezdve a generátorok jelkomponenseinek analízise alapján felépített *jelmodellel*. Ezt követi a keverőrendszert reprezentáló *súlyozó-összegző* és *sztochasztikus key click modell*, majd pedig a percussion funkció szerepét betöltő *burkológenerátor* ismertetése. Folytatásként az előerősítő lineáris és nemlineáris átviteli tulajdonságainak szoftveres megvalósításáról adok leírást: ismertetem a *kiterjesztett Wiener-Hammerstein modell* felépítését, valamint beszámolok a modell identifikációját célzó kísérleteimről. Eközben kitérek a *logaritmikus sweep* jellel végzett átvitelmérési eljárásra is. Végül az erősítő átvitelét megvalósító *párhuzamos IIR szűrőrendszer*ről, illetve az *expression control* pedálról lesz szó.

A MATLAB-beli modelleket követően az ezek alapján konstruált *valós idejű alkalmazást* ismertetem, amit a *JUCE keretrendszerben* implementáltam. Leírást adok a fejlesztőkörnyezetről, a felépített szoftver szerkezetéről, majd pedig tesztek eredményei nyomán értékelem az applikációt. Legvégül összefoglalom az elvégzett feladatokat, valamint javaslatokat teszek a lehetséges továbbfejlesztési irányokra vonatkozóan.

# Abstract

Thanks to its unique sound, the *Hammond organ* has reached an outstanding popularity in several musical genres up to now. However, due to the difficulties of its transport and maintenance, a significant need has arisen for its substitution by digital devices. Owing to the specialities of the instrument – instead of the sample-replaying solutions used in ordinary synthesizers – a *model based approach* seems to be more suitable that considers the working principle of the organ.

In this thesis, the modelling method of the electromechanical Hammond organ is discussed from studying the structure of the instrument, through the tasks of signal analysis and synthesis based on measurements, until the development of the real-time Hammond simulator application. My work has been founded both on sound samples recorded from a Hammond A-100 model, and on the available literature on-line.

Firstly, the function and operation of the organ components – such as *tonewheel generator*, *harmonic mixing unit*, *percussion circuit* and *preamplifier* – are introduced. After that, the software models of the above units are discussed, which have been developed in MATLAB. First, a *signal model* based on the analysis of the generator signals is shown. It is followed by the *weight-and-sum algorithm* and *stochastic key click model* representing the harmonic mixing unit, and the *envelope generator* of the percussion system. Thereafter, the software based modelling of the linear and nonlinear transfer characteristics of the preamplifier is presented: the description of the *extended Wiener-Hammerstein model* is followed by the experiments aiming the identification of model parameters. Meanwhile, a technique is introduced which has been used to measure the linear frequency response applying *logarithmic sweep* signal. Finally, the *parallel IIR filter system* is described that implements the transfer function of the preamplifier unit, which is followed by the *expression control* pedal.

After that, the real-time application based on the previous MATLAB models is outlined, which has been constructed in *JUCE application framework*. A description is given on the framework itself and the structure of the software. Then the evaluation of test results are presented, demonstrating the suitability of the application. Finally, the performed tasks are summarized, and suggestions are given about the possible directions of future development.

# 1 Bevezetés

A villamosmérnöki mesterképzés kezdetén elhatároztam, hogy a projekttárgyak – vagyis az Önálló laboratórium és a Diplomatervezés című kurzusok – keretein belül olyan témával fogok foglalkozni, amely a digitális hangfeldolgozás területén szerzett ismereteim bővítésén kívül a zene, az elektronikus hangszerek iránti érdeklődésemet is kielégíti. Így került képbe a *Hammond orgona* mint a zenetörténet elmúlt 85 évének egyik meghatározó, rendkívül széles körben alkalmazott hangszere, egyszersmind mai és/vagy mérnöki szemmel izgalmas tárgy.

Az alábbi fejezet 1.1. pontjában körbejárom a téma létjogosultságának a kérdéskörét. Az 1.2. pontban röviden írok a kutatásom tárgyát képző klasszikus Hammond modellek néhány jellemzőjéről, valamint napjaink azon digitális orgonáiról, amelyeket elterjedten használnak az idős Hammondok helyettesítésére. Az 1.3. pontban bemutatom a munkám során felhasznált hardveres és szoftveres eszközöket. Végül a 1.4. pontban röviden ismertetem a dolgozat további részeinek tartalmát.

## 1.1 A választott téma aktualitása

Az 1930-as évek Amerikájában, a gazdasági válság idején *Laurens Hammond* óragyártó cége az órák iránti csökkenő kereslet okán profilváltásra kényszerült. Felfedezték, hogy a korábban időmérésre használt, igen stabil fordulatszámú szinkronmotorjukat fel tudják használni elektromechanikus hangszerek készítéséhez. Így született meg a Hammond orgona [1], amellyel a templomoknak próbálták a hagyományos sípos orgonákhoz képest olcsó és hordozható alternatívát kínálni. A Hammond orgonák sajátos hangzásvilága később nagy sikert aratott a jazz-, majd a rockzenészek körében; ma már etalonnak mondható. Az eredeti hangszerek karbantartása, szállítása viszont nehézkes, költséges feladat. Ennek köszönhetően jelenleg is komolyabb fejlesztések zajlanak, amelyek a Hammond orgonák hangszínének pontos digitális reprodukálását célozzák.

Az általános célú, más hangszereket imitáló szintetizátorok működése általában hangminták periodikus visszajátszásán, vagyis a *sampling technikán* alapszik. Ebben az esetben az egyes billentyűk lenyomásakor olyan rövid hangfelvételek szólalnak meg, amelyeket az eredeti hangszer zenei hangok frekvenciáin történő megszólaltatásakor

rögzítettek. Az sampling-et alkalmazó szintetizátorok több szempontból alkalmatlannak bizonyultak a Hammond orgonák helyettesítésére: Egyrészt az eredeti hangszer igen sokféle beállítási lehetőséget kínál, ráadásul ezek közt előfordulnak (közel) folytonos skálán állítható paraméterek is. Ha a hangszer felépítésének ismeretében felrajzolnánk a lehetséges beállítások „állapotterét”, túl sok dimenzióval kéne számolnunk, vagyis a hozzájuk tartozó hangminta-sorok tárolása a memóriaigény szempontjából rendkívül gazdaságtalan lenne. Másrészt a hangszer mechanikai, elektronikai kialakítása olyan zajokat, pontatlanságokat okoz a létrejött hangjelben, amelyeket nem lehet ilyen módszerrel imitálni, viszont a zenészek számára fontosak, kíváncsok. Épp ezért került előtérbe a Hammond orgona hangjának *modell alapú szintézise*, amely a mechanikai és áramkörti kialakítás sajátosságait is figyelembe veszi. Így tehát a kimenő, kész hangjel rögzítése helyett a hangszer működésének vizsgálata került a középpontba.

## 1.2 A klasszikus Hammond modellek és helyettesítőik

A *Hammond Organ Company* 1935-ben mutatta be az első modelljét, amely az additív hangszintézis elvén állított elő harmonikusokban gazdag, intenzíven színezett hangot egyszerű, szinuszos formájú jelek összegzésével. A Hammond orgonák egyrészt abban voltak úttörők, hogy az említett harmonikusokat elektronikusan keverték össze, másrészt pedig a szinuszos jeleket egy elektromechanikus jelforrás, az ún. *tonewheel generator* segítségével állították elő. Az egymást követő modelljein végzett ugyan apróbb módosításokat a gyártó, de a hangszerek alapvető felépítése egészen az 1970-es évek közepéig nem változott [2]. A nagy keresletnek köszönhetően a tonewheel generator-os típusokból valaha gyártott példányok száma a milliós nagyságrendben mozog. A legkedveltebb, klasszikus modellek a *B-3* és a *C-3*.

A klasszikus, Hammond tonewheel orgonák hangzásának reprodukcióját célzó, általában digitális billentyűs hangszereket – szellemes szójátékkal – szokás *clonewheel organ*-nek is nevezni [3]. A clonewheel orgonák kezdetben analóg áramkörti kapcsolásokat, később sampling technikát és a digitális jelfeldolgozás eljárásait alkalmazták az tonewheel hangzás imitálására. A '80-as évek első analóg clonewheel orgonái – mint a *Korg CX-3* és a *Roland VK-1* – nagy sikert arattak a könnyűzenészek körében, ami sok hangszeripari gyártónak megadta a kezdő lökést ahhoz, hogy egyre precízebben modellezzék a Hammond orgonákat. Napjaink digitális, hangszermodell

alapú clonewheel orgonái esetében már gyakran felismerhetetlen a hangzásbeli különbség egy eredeti Hammond-hoz viszonyítva.

A ma kapható virtuális Hammond orgonákat két csoportra oszthatom aszerint, hogy kézzelfogható hangszerről vagy számítógépes plug-in-ról beszélünk-e. Az első csoport jelentős képviselőire példa a közkezdvelt *Clavia Nord C2D* [4], vagy a *Hammond Suzuki SK2* [5], amelyek a klasszikus Hammond modellekhez hasonlóan két 5 oktávós billentyűmanuállal rendelkeznek. A *Viscount Legend* szériájának [6] modelljei – mint a *Legend Classic* vagy a *Legend Live* – különösen nagy népszerűségnek örvendenek, ami a magas fokú hangzáshűségükön túl annak is köszönhető, hogy megjelenésük részletekbe menően idézi az eredeti Hammond orgonákét. A *Ferrofish B4000+ Authentic Organ Emulator* [7] egy MIDI és USB interfészekkel ellátott, kompakt megoldást kínál billentyűzet nélkül, viszont élethű Hammond B-3 hangzással. Napjaink zeneiparában a hagyományos, fizikai hangszerek helyett gyakran szoftveres plug-in-okat használnak, mivel így mind az élőzene, mind pedig a stúdiómunka eszközei variábilissá tehetők, logisztikai hátulütők nélkül. Hatványozottan igaz ez azon hangszerek esetében, amelyek MIDI billentyűzettel kényelmesen vezérelhetők. A piac legjobb virtuális Hammond plug-in-jai közé tartozik – a teljesség igénye nélkül – az *Acousticsamples B-5 Organ V2* [8], az *Arturia B-3 V* [9], illetve az *IK Multimedia Hammond B-3X* [10].

### 1.3 A felhasznált eszközök

A jelen dolgozat további fejezeteiben tárgyalt, Hammond orgona részegységeiről készített szoftvermodellek felépítéséhez az interneten elérhető irodalmi forrásokon kívül volt szerencsém egy valódi hangszert közelebbről is megismerni. Két alkalommal végeztem méréseket egy *Hammond A-100* típusú orgonán, amely műszaki tartalmát tekintve megegyezett a leghíresebb B-3 (és C-3) modellekkel.

*Megjegyzés:* Egy eredeti A-100 modell abban különbözik a másik kettőtől, hogy végerősítővel és hangszórókkal is fel van szerelve, így nem igényel külső hangforrást ahhoz, hogy megszólaljon. Az általam vizsgált A-100 orgonából a tulajdonosa kiszerelte az említett extrákat, mivel Ő is – mint a Hammond-játékosok általában – jobban kedveli a forgótölcsekes *Leslie* hangsugárzót az orgona eredeti hangforrásánál. Említésre méltó még, hogy a mérések alapját képző orgonát a tulajdonosa nem kis munka árán felújította és lelkiismeretesen karbantartja, így a hangszer a fennmaradt darabokhoz képest is kiemelkedően jó állapotnak örvend.

A mérések során a hangminták rögzítését egy *M-Audio FastTrack Pro* típusú külső hangkártyával és az *Audacity 2.2.2* verziójával végeztem. A mérési eredmények kiértékeléséhez, illetve ábrázolásához a *MathWorks MATLAB* szoftverrendszer *R2015a* kiadását használtam.

Az kiválasztott, legfontosabb részegységek szoftvermodelljét előbb MATLAB-ban építettem fel, majd csak ezután ültettem át a valós idejű szimulátorba. A valós idejű szoftvermodellt a *JUCE* elnevezésű fejlesztőkörnyezet *4.3.1* verziójának felhasználásával, C++ nyelven készítettem el. A JUCE-ban megírt programkódot a *Microsoft Visual Studio 2015* integrált szoftverfejlesztői környezetben fordítottam és futtattam.

## 1.4 A dolgozat felépítése

A 2. fejezetben előbb az additív hangszintézis Hammond-féle megvalósítását, majd pedig a Hammond A-100 szerkezeti felépítését, illetve az egyes részegységek működését tekintem át.

A 3. fejezettől a 6. fejezetig tartó szekcióban az orgonán végzett mérésekről és a MATLAB-ban végrehajtott modellépítési folyamatokról lesz szó. A 3. fejezet a jelgenerátorok belső ellenállásával, valamint kimenőjeleinek analízisével és jelmodell alapú szintézisével foglalkozik. Az ezt követő 4. fejezet a keverőrendszer egyszerű, jelfolyam elvű megvalósításáról, illetve a „*key click*” jelenségről szól. Az 5. fejezetben a *percussion* rendszer mérések alapján létrehozott szoftvermodelljét ismertetem. A 6. fejezet az előerősítőt reprezentáló modelltípus kiválasztásával, a modellparaméterek identifikációjával, és az erősítőn végzett átvitelmérések módszertanával foglalkozik. Itt esik majd szó az *expression control* pedál szoftveres megvalósításáról is.

A 7. fejezetben a valós idejű alkalmazás fejlesztésének szoftverkörnyezetét ismertetem, majd ezt követően a C++ modell felépítéséről, működéséről és értékeléséről lesz szó.

A 8. fejezetben összefoglalom az eddig elért eredményeket, végül pedig ismertetem a továbbfejlesztési lehetőségeket.

## 2 A Hammond orgona működése

Az alábbi fejezetben a klasszikus Hammond modellek – mint a B-3, C-3, vagy az általam vizsgált A-100 – szerkezeti felépítését, valamint működését tekintem át. A 2.1. pontban szó lesz az additív hangszintézisről mint alapparadigmáról, illetve annak Hammond-féle megvalósításáról. A 2.2. pont a hangszer felépítésével és a jelen dolgozat szempontjából releváns részegységek azonosításával foglalkozik.

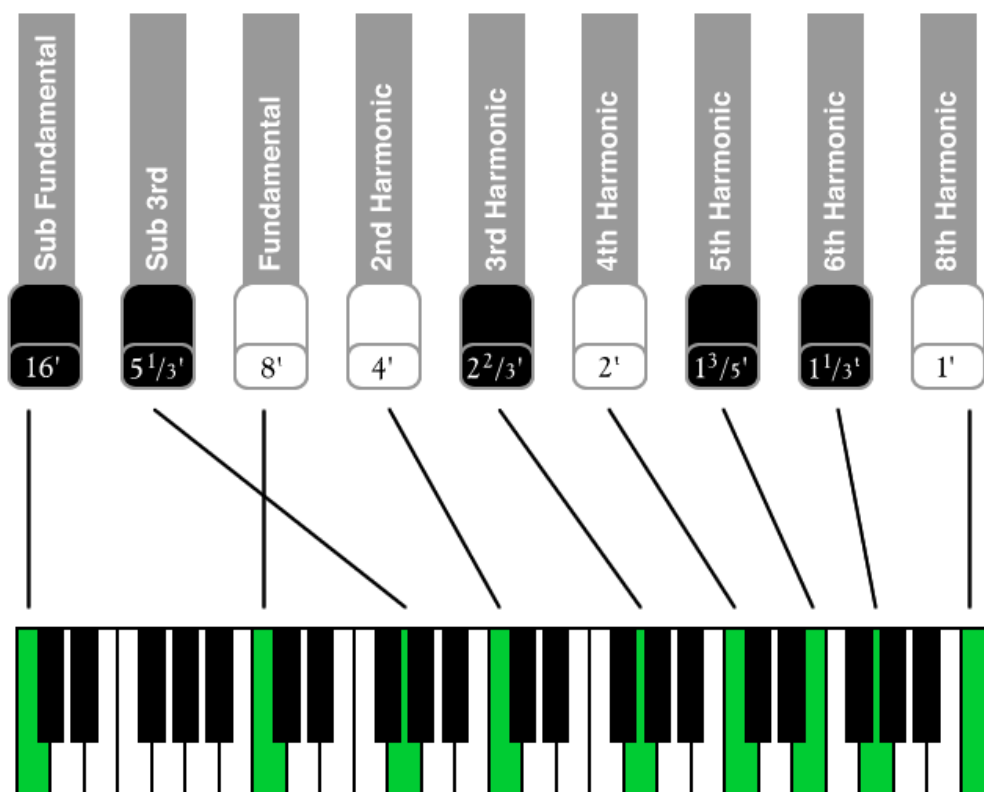
### 2.1 Additív hangszintézis

Az Hammond orgona működése az *additív hangszintézis* elvén alapul. Ez annyit takar, hogy a benne található tonewheel generator egység közel szinuszos formájú jeleket állít elő, majd ezek keverésével kaphatunk harmonikusokban gazdag hangot. Bármely billentyű lenyomásakor egyszerre 9 harmonikus szólal meg a hangszerjátékos által beállított súllyal.

Mivel a hangszer billentyűi 5 oktávot fognak át, valójában 549 szinuszjel előállítására lett volna szükség, ha minden billentyűhöz 9 különálló generátort rendelnek. Ez túlságosan nagy hardverigényt jelentett volna, ezért a Hammond gyár fejlesztőmérnökeinek takarékoskodni kellett a generátorok számával. Rájöttek, hogy ugyanazon generátor adhatja egyszerre az egyik billentyű alaphangját, egy másik billentyű második harmonikusát, és így tovább. Ezzel a módszerrel, valamint további egyszerűsítésekkel végül 91 darab szinuszgenerátor beépítése mellett döntöttek annak ellenére, hogy így több helyen előfordulnak pontatlan frekvenciájú harmonikusok.

Az 1. ábra olyan alaphangokat mutat, amelyek egyetlen – a képen balról a második – C billentyű lenyomásakor harmonikusokként szólalnak meg. A hangszer kimenetén megjelenő hangban hallható harmonikusok részarányát az orgonajátékos az ábra felső részén látható tolókapcsolókkal, az ún. *drawbar*-okkal tudja beállítani. (Valamely *drawbar* teljesen „betölt” állapotában a hozzá tartozó harmonikus néma. A *drawbar* fokozatos „kihúzásával” növelhető a harmonikus részaránya. Egy *drawbar* teljesen „kihúzott” állapotában az adott harmonikus hangereje maximális súllyal szerepel a mixben.) A Hammond orgonák által használt harmonikusok az ábrán is jelölve vannak: A lenyomott billentyű alapharmonikusán (*Fundamental*) kívül a második, harmadik, negyedik, ötödik, hatodik és nyolcadik harmonikusa is megszólal. Ezeken kívül szerepel

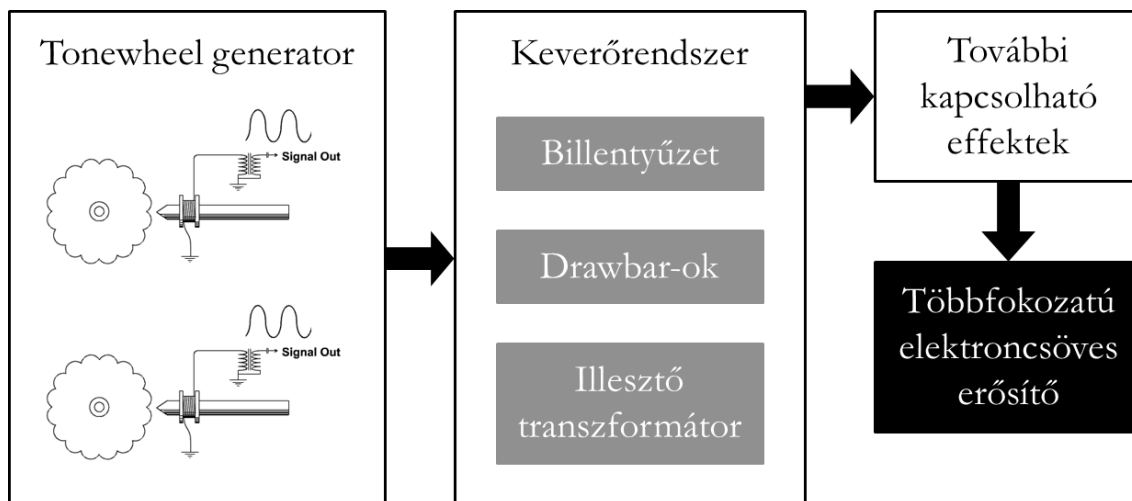
még az alaphang szubharmonikusa (*Sub Fundamental*) az alapfrekvencia felén, illetve ennek harmadik harmonikusa (*Sub 3rd*), ami tkp. az alaphang kvintjének felel meg. Természetesen több billentyű együttes lenyomásakor a harmonikusaik drawbar-onként összegződnek, így egy adott drawbar állása minden nyomva tartott billentyű adott harmonikusát egyszerre befolyásolja.



1. ábra: Egy C billentyűhöz tartozó harmonikusok [11]

## 2.2 Az orgona felépítése

A XX. század első felében, amikor a Hammond orgonákat tervezték, nem volt ritka az összetett mechanikus rendszerek, vagy mai szemmel nézve egyszerű elektronikus elemekből felépített, bonyolult analóg hálózatok használata. Az elemzésünk tárgyát képező hangszer sem kivétel ez alól, ezért – minden igyekezetem ellenére – előfordulhat, hogy a kedves Olvasó talál nehezen érthető részeket az alábbi leírásban. Erre az esetre ajánlom a *North Suburban Hammond Organ Society* honlapján található lapozható leírást [12], ami egyszerre közérthető, látványos, és viszonylag részletes. Emellett a Hammond A-100 kapcsolási és huzalozási rajzai könnyedén elérhetők a *theatreorgans.com* honlapon [13], vagy némi keresés után az eredeti, Hammond gyár által kiadott szervizelési útmutatóban [14].



2. ábra: A hangszer egyszerűsített blokkvázlata (tonewheel-ek rajza a [12] honlapról)

A 2. ábra blokkvázlata a Hammond orgonák főbb részegységeit, illetve alapvető jelútját mutatja, a könnyű átláthatóság érdekében nagymértékben leegyszerűsítve. Az ábrán szereplő építőelemek működését az alábbi alfejezetekben ismertetem.

### 2.2.1 A *tonewheel generator* működése

A klasszikus Hammond modellekben az ún. *tonewheel generator* – a továbbiakban *jelgenerátor-egység* – tölti be a jelforrás szerepét, amely az additív hangszintézishez szükséges jelek előállítását végzi.

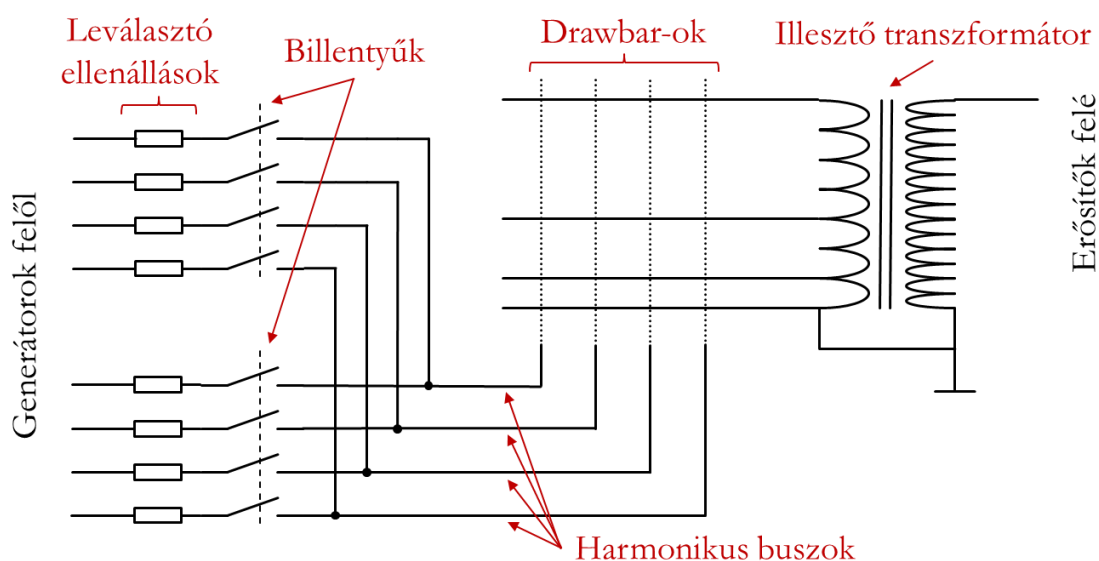
Az elektromechanikus jelgenerátor-egységben egy szinkronmotor összetett, fogaskerékes áttételrendszeren keresztül 91 darab, közel körlap alakú fémlemezt forgat állandó fordulaton. A lemezek pereme úgy van megmunkálva, hogy a mellékük helyezett vasmagos hangszedő tekercsekben közel szinuszos formájú jeleket indukáljanak. Minden egyes jelgenerátor frekvenciáját az adott lemez fordulatszáma és „fogainak” darabszáma határozza meg. A hangszedők jelei transzformátorokból és kondenzátorokból kialakított rezonáns szűrőkön keresztül jutnak a jelgenerátor-egység kimeneteire, amelyek a nem kívánt felharmonikusok és egyéb zajok elnyomásáért felelősek. Az orgona villamos hálózata távolról sem visszahatásmentes, ezért a felharmonikusok eliminálásán túl a szűrőkörök lényeges feladata a hangszedők kimenőfeszültségeinek stabilizálása, függetlenítése a keverőrendszer által okozott terhelésváltásoktól. (Valójában nem minden generátor kimeneti szűrőáramköre azonos; a generátorok frekvenciáinak három intervallumában háromféle különböző struktúrájú szűrőkört alkalmaztak.)

*Megjegyzés:* A 12 legkisebb alaphfrekvenciájú (1...12. számú) generátor szinuszos formájú jel helyett eleve intenzív harmonikustartalommal rendelkező jelet állít elő. Ezeket a jeleket kizárólag a lábbal kezelhető manuál hasznosítja, ami kívül esik a modellezési törekvéseimen, ezért a továbbiakban csak a szinuszos (13...91. számú) generátorokról lesz szó.

A jelgenerátorok kimeneteinek mérés alapú analíziséről és modellezéséről a 3. pontban írok majd bővebben.

## 2.2.2 A keverőrendszer felépítése és működése

A jelgenerátor-egységben létrehozott szinuszos jelek útja a keverőrendszerbe vezet, ahol már a billentyűkhöz tartozó harmonikusokként funkcionálnak. A keverőrendszer fontosabb részegységei maga a billentyűzet, a harmonikusok keverési arányait szabályzó drawbar-ok, és az illesztőtranszformátor. A keverőrendszer felépítését az alábbi 3. ábra mutatja egyszerűsítve.



3. ábra: A keverőrendszer felépítése

A keverőrendszer egyik fontos építőeleme a billentyűsor. Minden billentyű alatt 9 darab független, kétállású kapcsoló van elhelyezve, amelyek lenyomásakor többé-kevésbé egyszerre záródnak. Adja tehát magát, hogy a 9 kapcsoló az adott billentyűhöz tartozó 9 harmonikusért felelős. Az összes billentyű  $n$ -edik harmonikusához tartozó kapcsoló kimenete az  $n$ -edik *harmonikus busz*hoz csatlakozik; ennél fogva az  $n$ -edik harmonikus buszon a lenyomott billentyűk  $n$ -edik harmonikusai keverednek össze. A

hanggenerátor-egység kimeneteiről érkező jelek leválasztó ellenállásokon keresztül vannak a kapcsolók bemeneteire kötve. Ezek az ellenállások egyrészt leválasztják a generátorok kimeneteit a harmonikus buszokról, másrészt a buszokra kevert jelek amplitúdóinak nagyságát (súlyozási arányát) is meghatározzák.

Amint azt láttuk, a 9 harmonikus buszon a lenyomott billentyűkhöz tartozó harmonikusok jelennek meg keverve. A 9 drawbar és az illesztőtranszformátor alkotják a keverőrendszer második szekcióját, amely a harmonikus buszok jeleinek súlyozott keverését végzi. Az illesztőtranszformátor alapvetően a keverőrendszer és az előerősítő bemenete közti impedanciaillesztést szolgálja, ám ezen túl egy másik funkciót is betölt: A transzformátor primer tekercse 9 helyen van megcsapolva, méghozzá logaritmikus léptékben. Minden drawbar egy-egy kilencállású kapcsoló, amely a hozzá tartozó harmonikus busz jelét az illesztőtranszformátor primer oldalának valamelyik kivezetéséhez kapcsolja. (A legalsó kivezetés a földpontra van kötve, ahogy az ábrán is látható.) Ezzel az összeállítással egyszerre válik lehetővé a harmonikus buszok jeleinek keverése és súlyozása, hiszen a drawbar-ok állásai szerint alakulnak az áttételi arányok a transzformátor szekunder tekercse – vagyis az előerősítő – felé. (A drawbar-ok harmonikuskiosztásáról már volt szó a 2.1. pontban.)

*Megjegyzés:* A Hammond orgonák két darab 5 oktáv széles kézi manuállal rendelkeznek, így az előbb megismert keverőrendszerből is két komplett példányt tartalmaznak. A két manuálhoz külön huzalokkal csatlakoznak a tonewheel generator kimenetei, a két illesztőtranszformátor szekunder oldali kivezetése pedig közösítve csatlakozik az előerősítő bemenetére.

*Megjegyzés:* Egy valóságos hangszeren egy manuálhoz két drawbar-sor, valamint 9 rögzített harmonikusarányú ún. *preset* is tartozik, amelyek között a manuálok bal szélén elhelyezett „inverz festésű” billentyűk segítségével lehet váltani. A fix beállítások segítenek a hangszerjátékosnak a gyors hangzásváltásban. Mivel a preset-ek megvalósítása csupán „huzalozás” kérdése, funkciójuk pedig megegyezik a drawbar-okéval, ezért a továbbiakban nem foglalkozom velük. (A hangszer már így is elég komplikált.)

### **2.2.3 Az opcionális effektek**

A jelút következő állomásait azok a be- és kikapcsolható effektek (*percussion* és *vibrato*) adják, amelyek az eddigi részegységek által előállított „stacionárius

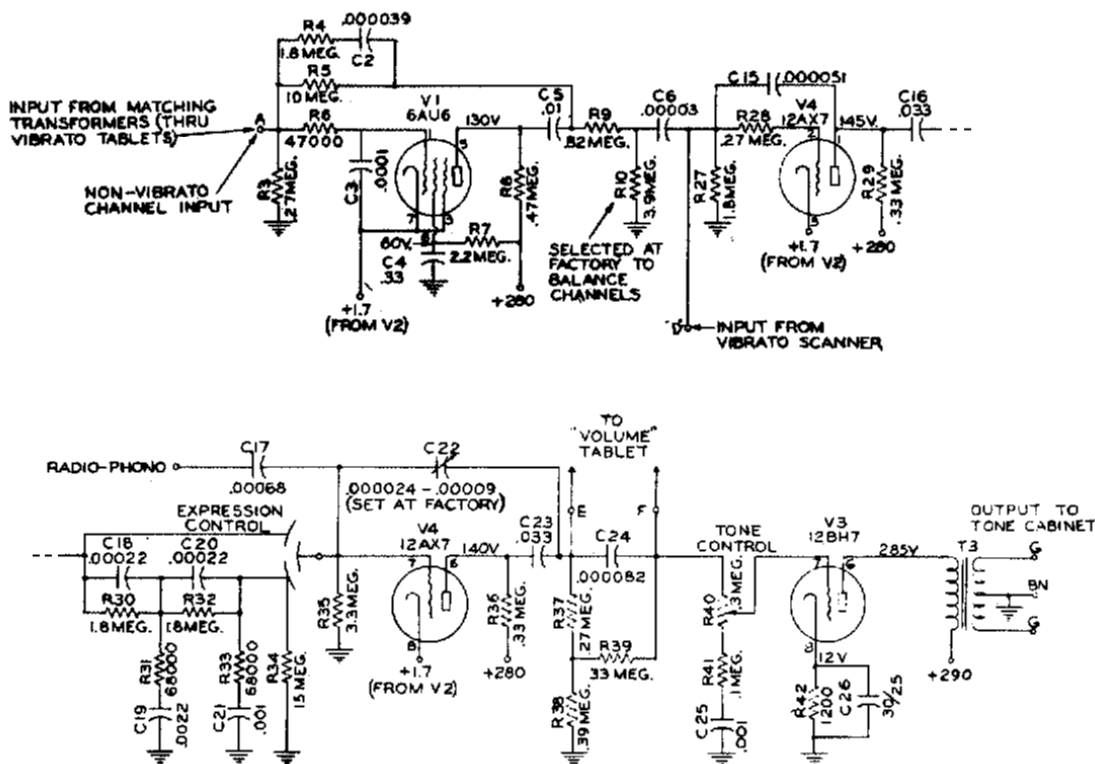
multiszinusz” jeleket az idő függvényében variálják, így egyfajta „dinamikával” látják el a hangjelet.

A *percussion* funkció a billentyűleütéseknek kölcsönöz hangzásbeli dinamikát. Ha a hangszerjátékos bekapcsolja a *percussion* effektet, a felső manuál bármely billentyűjének lenyomásakor – az alapvető harmonikusok mellett – megjelenik egy additív, tranzien্স módon lecsengő komponens is az orgona hangjában. Fontos megjegyezni, hogy az említett additív tranzien্স hang csak akkor szólal meg, ha korábban semelyik billentyű nem volt lenyomott állapotban. (A zenei hangszintézis nyelvén ezt a fajta működést „single triggered” elvű vezérlésként is emlegetik [15].) Az orgonán található választókapcsolók segítségével beállítható a *percussion* effekt néhány tulajdonsága: egyrészt eldönthetjük, hogy a második vagy a harmadik harmonikus adja-e a *percussion* jelforrását (*second* vagy *third*), másrészt két lehetőség közül választhatunk az effektus hangerejét illetően (*normal* vagy *soft*), harmadrészt szintén két opciónk van lecsengés gyorsaságának beállítására (*slow* vagy *fast*).

*Megjegyzés:* A *percussion* áramkör – jelbemeneteinek és vezérlésének specialitása okán – nem egyszerűen a keverőrendszer után foglal helyet a jelútban. A *percussion* jelforrását a második vagy a harmadik harmonikus adja, ezért a felső manuál megfelelő harmonikus buszai a *percussion* rendszer kapcsolóihoz csatlakoznak, majd a jelek onnan jutnak vissza drawbar-okhoz. A hangszer tervezői a triggerjel előállítását a nyolcadik harmonikus buszának „újrahasznosításával” oldották meg, ezért az effekt bekapcsolt állapota mellett a legfelső harmonikus nem szól.

A 4. ábra a *percussion* effektust megvalósító áramkörü kapcsolást mutatja, amely része a leggyakoribb Hammond modellekbe szerelt AO-28 szériaszámú előerősítő áramkörnek, ám maguktól az erősítőfokozatoktól mégis jól elkülöníthető. A kapcsolásban elrejtett burkológenerátor időállandóját a C31 jelölésű kondenzátor, valamint az R57 és R58 jelölésű ellenállások határozzák meg. Amikor a hangszerjátékos lenyom egy billentyűt, akkor a V6 dióda vezetni kezd, így áram indul meg R58-on keresztül, amit C31 állandó értéken tart. Eközben a T5 transzformátor középső megcsapolásánál mérhető feszültség közel exponenciális léptékben csökken, amivel egy időben a V7 jelölésű kettős triódából kialakított differenciálerősítő rácsain is csökken a feszültség egyenkomponense, vagyis lecsökken az erősítés mértéke. Az erősítés csökkenési üteme meghatározza az áramkör kimenetén megjelenő jel burkológörbéjét. Mivel a burkológenerátor áramkör





5. ábra: A Hammond AO-28 típusú előerősítő kapcsolása (két részletben) [16]

A Hammond orgonák tervezésének korában az elektronikai ipar a ma létező aktív eszközök közül csupán az elektroncsövet ismerte, ezért az előerősítőt egy négyfokozatú elektroncsöves kapcsolással valósították meg (5. ábra). A jelszint-konverzió mellett az előerősítő egy másik lényeges funkciót is ellát: A tonewheel generator-ban található, szinuszos jelek előállításáért felelős fogaskereknek nincsenek „mereven” rögzítve a tengelyükön, hanem egyfajta mechanikus szabadonfutóval vannak ellátva. A szabadonfutók használatának célja az volt, hogy valamely kerék véletlen lefékeződése ne okozzon hibát a többi kerék fordulatszámában, azaz ne okozzon frekvenciahibát a többi generátor-kimeneten. A megoldás hátránya az, hogy a generátorban előállított, egymással harmonikus viszonyban lévő szinuszos jelek relatív kezdőfázisai nem állandó értékűek, vagyis az egyes harmonikusok keverésének eredményeként létrejött jel csúcstényezője sem determinisztikus. Szélsőséges esetben a generátor indítása után a fogaskereknek forgása úgy állandósul, hogy a harmonikus szinuszosok épp egyazon pillanatban veszik fel a csúcserőértéküket. Ebben az esetben az előerősítő feladata a létrejött jelcsúcs „simítása”, aminek hatására torzítás jön létre a rendszerben. Az előerősítő modellezésének kérdéskörével a 6. fejezet foglalkozik majd.

A Hammond orgonákon található lábpedált és a mögötte működő – előerősítőbe rejtett – áramköri kapcsolást abból a célból fejlesztették ki, hogy a zenészek minél finomabban tudják szabályozni az orgonajáték dinamikáját. (Jelen esetben hangerőt, hangzásbeli „teltséget” értek a dinamika alatt.) Annak ellenére, hogy a dinamika változtatására a legkézenfekvőbb megoldást általában az egyszerű hangerőszabályzás jelenti, a Hammond tervezői máshogyan oldották meg a kérdést. A pedál működése ugyanis a hangszín variálásával biztosítja a kívánt hatást, azaz valójában fiziológiai hangerőszabályzásról beszélhetünk, ami kihasználja az emberi hallás sajátosságait. Az emberek halló szervrendszere jóval érzékenyebb a középfrekvenciás – 1 kHz körüli – tartományra, mint a kis- vagy nagyfrekvenciákra, így a hangosságérzet befolyásolása megoldható a középfrekvenciás tartomány szűrésével anélkül, hogy jelentősen csökkenteni kéne a jelszintet az orgona hangzásának „keretét” adó kis- és nagyfrekvenciás sávokban. Mivel a középfrekvenciák jelentős súllyal bírnak az emberi beszédhangban is, ezért a lábpedált szokás *expression control*-ként emlegetni. A pedálról, illetve a mögöttes szűrőáramkőről a 6.6. pontban lesz szó bővebben.

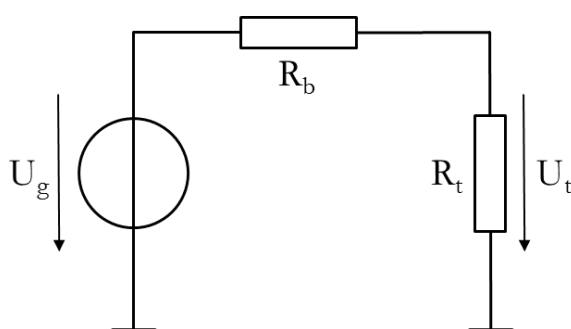
### 3 A jelgenerátor-egység modellje

Amint azt a 2. pontban már láthattuk, a Hammond orgonák „spektrálisan színes” hangjelet állítanak elő additív hangszintézis segítségével, amihez a tonewheel generator egység 91 darab kimenete szolgáltatja az alapjeleket. Mivel ezek a jelek adják minden további feldolgozás alapját, így szükség van a pontos ismeretükre.

Az alábbi fejezet a Hammond orgona jelgenerátor-egységén végzett méréseket, és az ezekből leszűrt eredményeket ismerteti. A 3.1. pontban a generátorok belső ellenállásának méréséről lesz szó. A 3.2. alfejezetben belül a 3.2.1. pont a generátorok kimenőjeleinek analízisével, a 3.2.2. pont a jelmodell felépítésével, a 3.2.3. pont pedig a jelszintézissel foglalkozik.

#### 3.1 A generátorok belső ellenállásának mérése

A hangszer felépítéséből fakadóan (lásd: 2.2. pont) a jelgenerátor-egység kimeneteinek aktuális terhelését a keverőrendszer határozza meg. A jelgenerátorok kimenetei a leválasztó ellenállásokon (és a billentyűkön) keresztül látják az illesztőtranszformátor primer oldali impedanciájának drawbar-ok által meghatározott részét. Felmerült a kérdés, hogy mennyire befolyásolja ez a változó terhelés a generátorok kimenőjeleinek nagyságát. A válasz keresésének első lépése a kimenetek belső impedanciáinak, vagy az egyszerűség kedvéért belső ellenállásainak mérése volt.



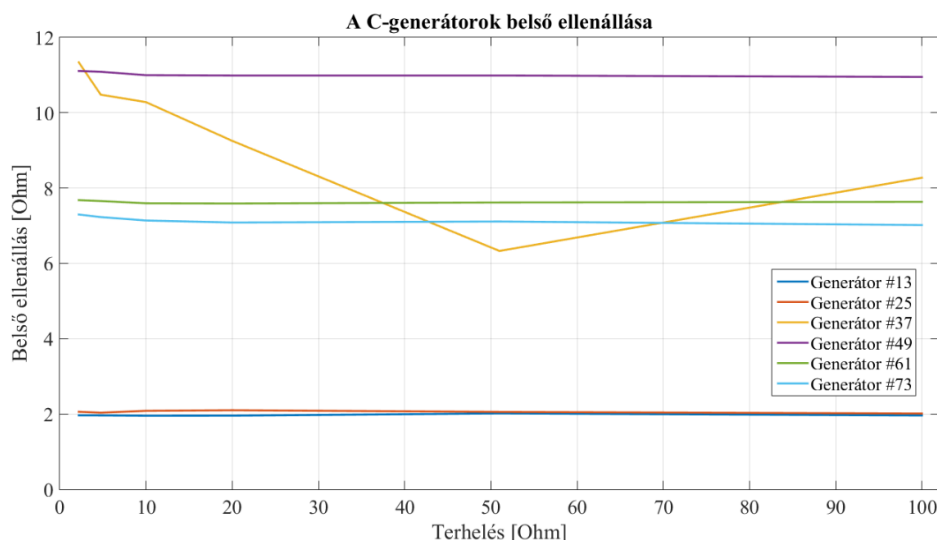
6. ábra: A belső ellenállás mérése

A jelgenerátorok belső ellenállását a 6. ábra szerinti összeállításban mértem. Többféle, különböző értékű terhelő ellenállást ( $R_t$ ) kapcsoltam a generátor adott kimenetére, miközben kb. 20 másodperc hosszú hangmintákat rögzítettem a külső hangkártya azonos beállításai mellett. A hangmintákból kalkuláltam a különböző

terhelések esetén megjelenő kimeneti feszültségek effektív értékét ( $U_t$ ), majd az (1) képlet alapján kiszámoltam a generátor belső ellenállását (minden terhelésnél):

$$R_b = |Z_b| = R_t \cdot \frac{U_g - U_t}{U_t}, \quad (1)$$

ahol  $R_b$  a belső ellenállás, ( $Z_b$  a belső impedancia),  $R_t$  a terhelő ellenállás értéke,  $U_g$  az üresjárási feszültség effektív értéke,  $U_t$  pedig a terhelő ellenálláson eső feszültség effektív értéke. (Az üresjárási feszültség mérésekor nem volt terhelés a generátor kimenetén.) A többféle terhelés használata, illetve a többszöri számítás csupán biztonsági okokból történt; elegendő lett volna csak az üresjárás és egyetlen terhelés vizsgálata. A mérések során csak a belső ellenállással foglalkoztam, mivel a belső impedancia mérése jóval komplikáltabb lett volna és valószínűleg lényegtelen a modellezés szempontjából.



7. ábra: A C-generátorok belső ellenállása

A 7. ábra a zenei C hangot adó generátor-kimenetek belső ellenállásait mutatja. A számításokat a következő terhelésekre végeztem el: 2.2  $\Omega$ , 4.75  $\Omega$ , 10  $\Omega$ , 20  $\Omega$ , 51  $\Omega$  és 100  $\Omega$ . Látható, hogy az ellenállásértékeket összekötő görbék közel konstansok, tehát az eredmények nagy valószínűséggel helyesek. (A 37-es számú generátor esetében vélhetően a mérőcsipesz és a kivezetés közötti rossz kontaktus miatt adódtak eltérő értékek.)

Mivel a dolgozat készítésének időpontjáig nem került sor az illesztőtranszformátor mérésére, ezért a belső ellenállásokat érintő mérési eredmények egyelőre irrelevánsak a hangszermodell szempontjából.

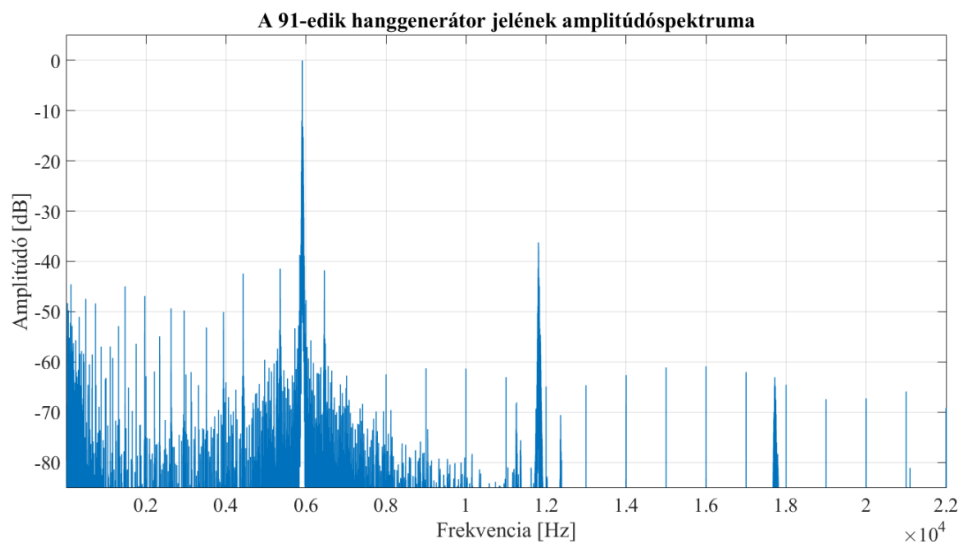
## 3.2 A generátorok jelének modellezése

A generátorok kimenőjelének alaposabb vizsgálata elengedhetetlen volt a jelgenerátor-egység szoftvermodelljének elkészítéséhez, hiszen a jelformák alapján befolyásolják a létrehozott hang tónusát. A következő pontok logikai sorrendben ismertetik a jelmodell építésének állomásait.

### 3.2.1 Jelanalízis

Az elemzések első szakaszát a 91-edik generátorból származó hangmintán mutatom be, mert ezen viszonylag jól szemléltethető a jelek néhány alaptulajdonsága.

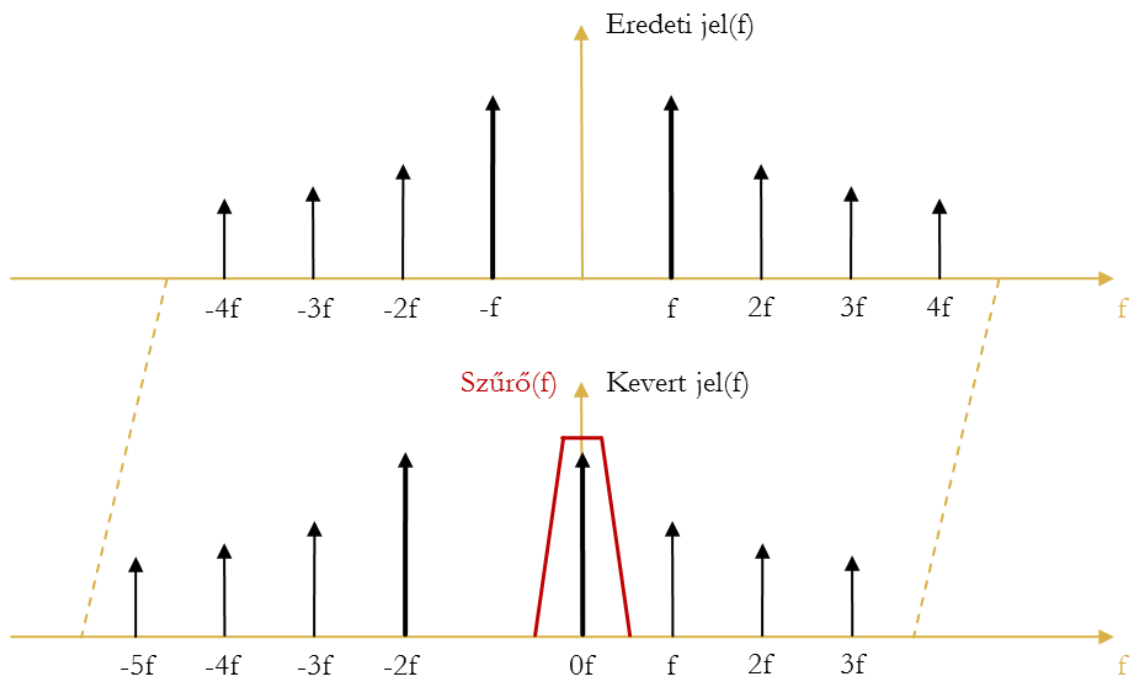
Első lépésben a generátor jeléről készített, nagyjából 20 másodperc hosszú mintát Hanning-ablakkal szoroztam, majd Fourier-transzformáltam, aminek eredményeként igen jó frekvencia-felbontású, részletgazdag spektrális képet kaptam. (Az amplitúdóspektrum vizsgálata általában jó kiindulási alapot jelent, hiszen így megállapíthatjuk, hogy melyek azok a komponensek, amelyek hosszabb idő átlagában az adott jel karakterét adják.) A 8. ábra szemlélteti az amplitúdóspektrumot, amelyen a kb. 5924 Hz frekvenciájú alapharmonikus mellett megjelenik a 35 dB-lel gyengébb második harmonikus és a -60 dB-es harmadik harmonikus is. Ezeken kívül a jel szélessávú (hallható) zajt is tartalmaz.



8. ábra: A 91. generátor jelének amplitúdóspektruma

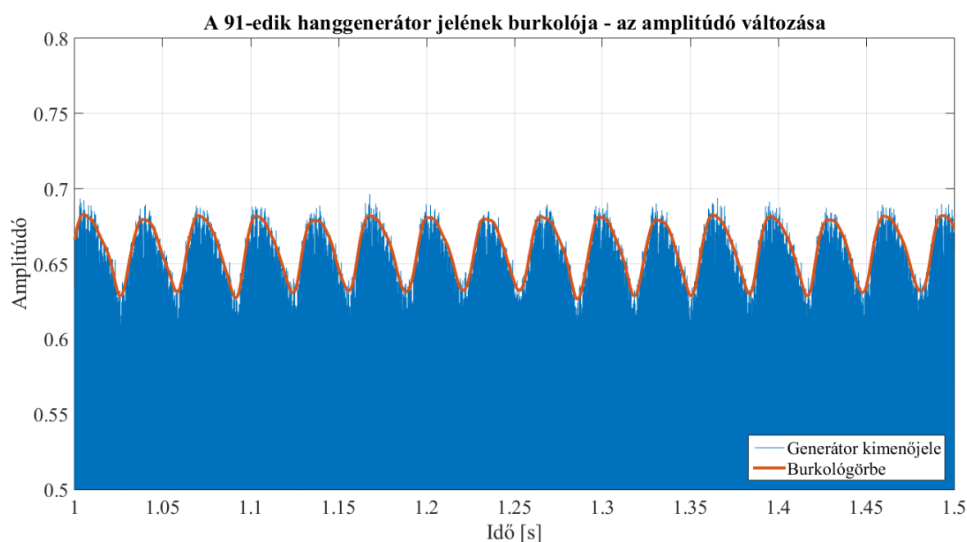
Látható, hogy a jel nem tisztán szinuszos, ezért szükségesnek látszott a harmonikusok szétválasztása a további vizsgálatokhoz. Erre a feladatra egy heterodin analizátort tartalmazó MATLAB függvényt készítettem. Az analizátor (9. ábra) előbb

lekeveri a jelet az alapharmonikus frekvenciájával, majd egy aluláteresztő szűrő segítségével kiszűri a felharmonikusokat és a zaj jelentős részét.



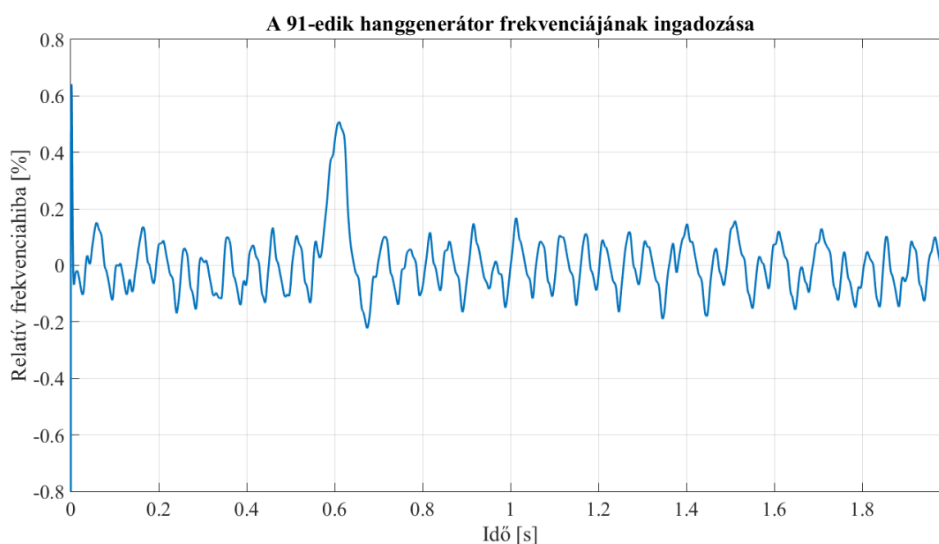
**9. ábra: A heterodin analízátor működése**

A példánkban szereplő 91-es számú generátor jelének vizsgálatához az analízátor keverési frekvenciáját a jel alapharmonikusára állítottam, a felhasznált szűrő pedig negyedfokú Butterworth szűrő volt, 100 Hz vágási frekvenciával. Az analízátorral készített komplex időfüggvény abszolút értéke megadja az alapharmonikus jel burkológörbéjét, amelyet a 10. ábra mutat. A burkolón jól megfigyelhető egyfajta amplitúdómoduláció-jellegű hatás, amely hallható is az eredeti jelen.



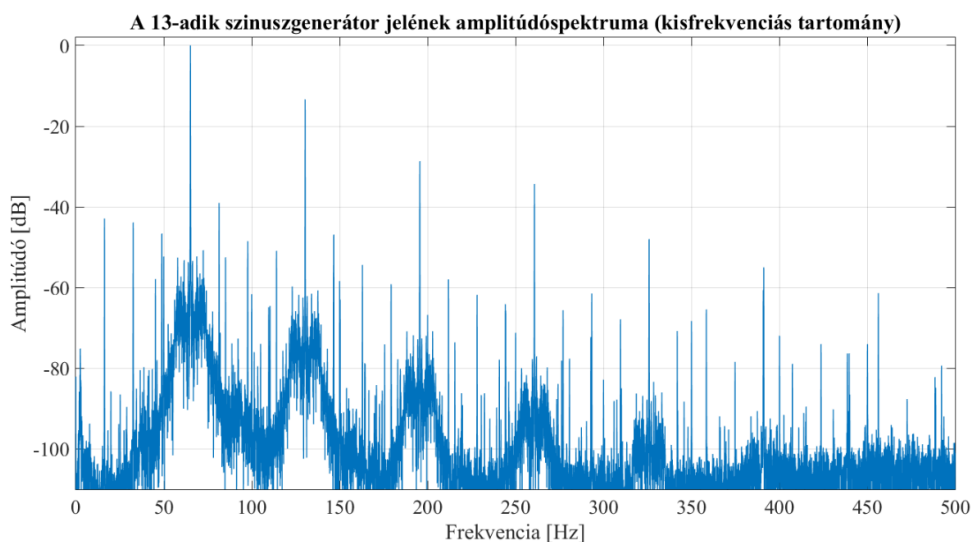
**10. ábra: A 91. generátor jelének amplitúdóbeli változásai**

Az analizátorból nyert komplex időfüggvény szögének deriváltja megadja a generátor frekvenciájának időbeli ingadozását. Az alapharmonikushoz viszonyított relatív frekvenciahiba időfüggvényét szemlélteti a 11. ábra. Az ábrán látható legnagyobb eltérés is csupán 0.5%-os, ami az emberi fül számára gyakorlatilag nem észlelhető. Ebből következően a generátorok modelljének összeállításakor a frekvenciahibákat figyelmen kívül hagyhatjuk. (A tonewheel generator „hangkerekei” a fordulatszám tekintetében mechanikusan csatolva vannak, ezért jelentősebb frekvenciahiba más generátorok esetében sem állhat fenn.)



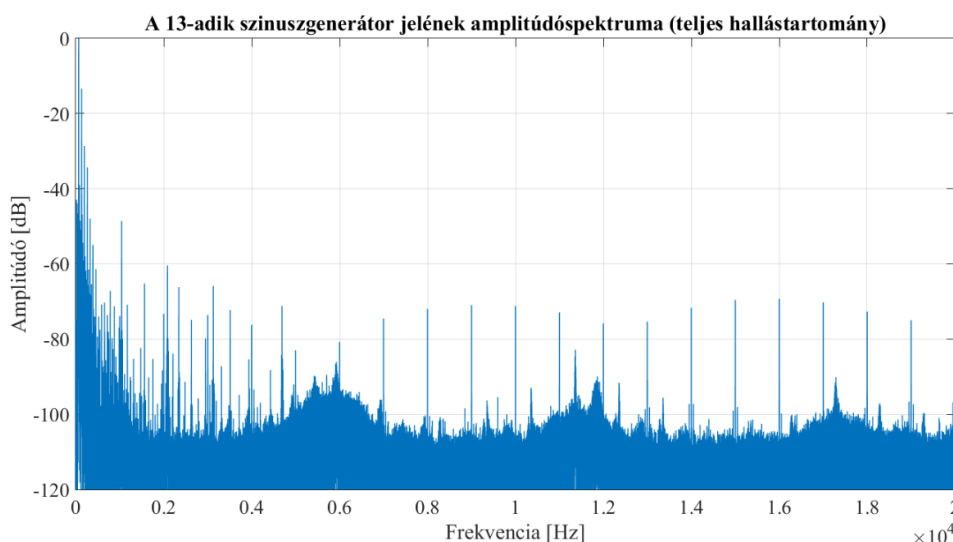
**11. ábra: A 91. generátor frekvenciájának ingadozása**

Az elemzések második szakaszát a 13-as számú generátorból származó hangmintára alapozom, mert ennek jóval kisebb az alapfrekvenciája, így kivethetőbbé válik a harmonikus-összetétel, valamint a háttérzaj is.



**12. ábra: A 13. generátor jelének amplitúdóspektruma a kisfrekvenciás tartományban**

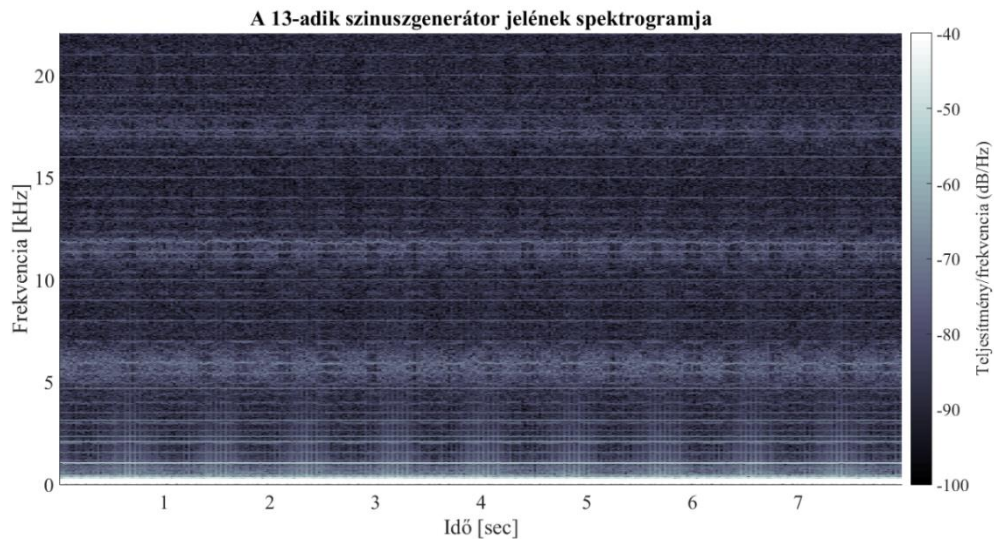
A 12. ábrán a 13-as generátor jelének amplitúdóspektruma látható a kisfrekvenciás tartományban. A 65.22 Hz-en, valamint annak többszörösein jelentkező, magasan kiemelkedő csúcsok jelölik a „rendes” harmonikusokat. Az is jól kivehető, hogy ezek között – némileg kisebb amplitúdóval – megjelennek a szubharmonikusok alkotta csúcshármasok. Ezen szubharmonikusok az említett alapfrekvencia negyedén és annak többszörösein helyezkednek el. Jelenlétük csupán a kisebb alapfrekvenciájú jelgenerátoroknál szignifikáns. Nagyobb alapfrekvenciájú generátorok – mint például a 91-es számú (8. ábra) – esetében „elvesznek a zajban” vagy létre sem jönnek.



**13. ábra: A 13. generátor jelének amplitúdóspektruma a teljes hallástartományban**

Az előzővel szemben a 13. ábra már a teljes hallástartományra kiterjedően mutatja ugyanazon jel amplitúdóspektrumát. Az imént látott részletek így az ábra bal szélére zsugorodnak, miközben kirajzolódik a jel szélessávú összetétele. Először is feltűnő, hogy 1 kHz-en és annak többszörös frekvenciáin jelentős amplitúdójú csúcsok jelentkeznek, amelyek az összes többi generátor esetében is jelen vannak, legfeljebb más arányokban. Ez a hatás valószínűleg nem a jelgenerátor mechanikai kialakításból származik, hanem valamilyen csatolt zavar okozza, illetve a generátorok terhelésével jelentősen csökken. A kHz-es csúcsokon kívül a szélessávú spektrumon látható még három, a zajból „domb jelleggel” kiemelkedő szakasz. Ezen dombok időbeli viselkedése a 14. ábrán jobban megfigyelhető.

Amíg az előző amplitúdóspektrumok csupán a jelben megjelenő komponensek hosszabb időre átlagolt teljesítményét szemléltetik, addig a spektrogram (14. ábra) az időbeli változásokról is információval szolgál. A 8 másodperc hosszú mintából előállított spektrogramon látható, hogy az előbb látott dombok teljesítménye periodikusan változik, méghozzá egymással szinkronban. A hangfelvételen ez a hatás egyfajta „halk nyikorgásként” érhető tetten, ami arra utal, hogy ez az effektus a fogaskerekes áttételrendszer vagy a szinkronmotor aszimmetriájából származik. A dombok időbeli periodicitása – vagy másképpen a dombokat moduláló jel hullámformája és frekvenciája – minden generátor esetében egyforma, ami a jelmodell készítésekor jelentős könnyebbséget okozott.



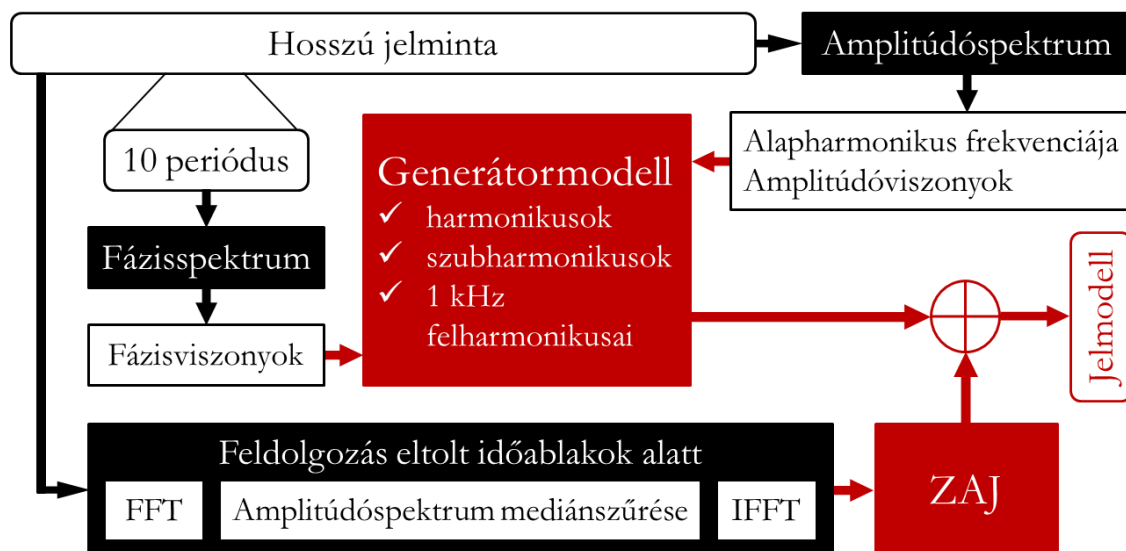
14. ábra: A 13. generátor jelének spektrogramja

### 3.2.2 Jelmodell

Az előző pontban megismert jellemzők alapján, több iterációs ciklusban sikerült elkészítenem a jelgenerátorok egy lehetséges jelmodelljét. Az elsődleges célom az volt, hogy a generátormodell által előállított jelek és az eredeti hangminták között ne legyen lényeges, hallható különbség. Egy „Sinusoids plus Noise” jellegű [17], egyszerű modellt képzeltem el, amihez a hangminták spektrális reprezentációjából nyerek ki a fontosabb komponensek paramétereit.

A jelmodell tehát két, jól elkülöníthető összetevőből áll: Egyrészt szükség van a generátorspecifikus hatások előállítására, úgymint a „rendes” harmonikusok, szubharmonikusok és a kHz-es harmonikusok. Másrészt ezekhez hozzá kell még adni azt a szélessávú „nyikorgó” zajt, amely minden generátor esetén hasonló. A generátorspecifikus hatások keresendő paramétereit azok pontos frekvenciája, amplitúdója, valamint relatív fázishelyzete.

A generátorok jeleinek MATLAB-beli analízise, a generátorspecifikus jelparaméterek keresése és a jelmodell építése (jelszintézis) párhuzamosan zajlott. A folyamat végén született modell optimalizálása főként a hallgatásos tesztek segítségével, illetve a jelek idő- és frekvenciatartománybeli megfeleltetésével, az ezeken tapasztalható különbségek minimalizálásával volt lehetséges.



15. ábra: A jelmodell paramétereinek keresése

A fenti, 15. ábrán láthatjuk, hogy a generátorspecifikus paramétereket, valamint a közös zajt miként sikerült kinyernem a felvett mintákból.

Legelőször a hosszú, nagyjából 20 másodperces hangmintából nyert amplitúdóspektrumot használtam fel az alapharmonikus frekvenciájának megállapítására, mivel ennek volt megfelelően nagy frekvencia-felbontása, illetve a rengeteg periódusból számított FFT jól kiátlagolta az alapfrekvencia változásait, hibáit – bár a 3.2.1. pontban láthattuk, hogy a frekvenciabeli változások elhanyagolhatók. A legnagyobb amplitúdójú csúcs minden felvétel esetében az alapfrekvenciára esett, ezért kézenfekvő volt egyszerű maximumkeresést alkalmazni. A felharmonikusok és szubharmonikusok frekvenciáit mindig az alapharmonikuséból számítottam ki. (Eleinte próbálkoztam más megoldással is: az alapharmonikus frekvenciájának ismeretében kiszámítottam a további harmonikusok várható frekvenciáit. Ezek szűk környezetében lokális maximumkeresést végeztem az amplitúdóspektrumon, majd pedig a lokális maximumok helyeit tekintettem a harmonikusok érvényes frekvenciáknak. Ekkor az értékek diszkrét frekvenciatengelyről történő leolvasása – a harmonikus frekvenciák nem egész arányai okán – gyakran a komponensek diszharmonióijához vezetett. A diszharmonikus komponensek összegzése aperiodikus jelet adott, ami az időfüggvényen lineáris egyenszint növekedés formájában volt tetten érhető. Ennek folytán a módszert elvetettem.)

Az egyes komponensek relatív amplitúdóviszonyának megállapításához szintén a nagy felbontású amplitúdóspektrumot hívtam segítségül, amin a már ismert frekvenciák szűk környezetében végeztem maximumkeresést. A decibelben kapott eredményeket

később visszaalakítottam lineáris skálájú mennyiségekké. (Mivel kiderült, hogy a 3.2.1. pontban megismert, amplitúdómoduláció jellegű effektust az alapharmonikusknál kisebb frekvenciájú szubharmonikusok jelenléte okozza, ezért a korábban szükségesnek látszó megoldással – vagyis a harmonikusok amplitúdóinak modulálásával – szemben elég volt az szubharmonikusokat a többi jelkomponenshez adni és minden komponens amplitúdóját állandónak tekinteni.)

A hosszú mintából előállított fázisspektrum túl részletesnek, vagyis a jelanalízis szempontjából túl zajosnak bizonyult. Ezt a problémát úgy sikerült orvosolnom, hogy egy 10 periódus hosszú szakaszból készült fázisspektrumban kerestem a vizsgált jelalkotók relatív fázishelyzetét. Ennek eredményeként a generátormodell kimeneti jelalakja (időtartományban) majdnem megegyező az eredeti felvétel jelformájával.

Mivel a „nyikorgás jellegű” szélessávú zaj minden generátor jelében nagyjából egyformán jelentkezik, ezért elég azt egyetlen hangmintából kiszűrni, majd pedig a jelszintézis során minden generátormodell jeléhez hozzákeverni. A zaj kinyeréséhez a legkisebb frekvenciájú szinuszos hangmintát (13. generátor) használtam, mert ebben a „rendes” harmonikusok csupán a kisfrekvenciás tartomány szűk sávjába esnek, tehát könnyű azokat eliminálni. A hangmintán ún. idő-frekvencia feldolgozást végeztem, hogy rövid időablakonként frekvenciatartománybeli mérést, illetve transzformációt végezhessenek a jelen. Ehhez periodikusan eltolt, átlapolódó Hanning-ablakok alatt Fourier-transzformációt végeztem. Az így kapott rövid szakaszok amplitúdóspektrumait nagy foksámú (200-adfokú) mediánszűrő segítségével simítottam, hogy a jel nagy amplitúdójú harmonikus komponenseit elnyomjam, viszont a szélessávú zaj – és a zajküszöb alatt megbúvó „nyikorgás” – jellege megmaradjon. A simított amplitúdóspektrumokból és az eredeti fázisspektrumokból inverz Fourier-transzformációval alakítottam vissza a jelet időtartományba. Ezután szükség volt még a jel felüláteresztő szűrésére (500 Hz vágású, harmadfokú Butterworth IIR szűrővel), mert kisfrekvenciás tartományban – a nagy amplitúdójú harmonikus komponensek környezetében – a mediánszűrés nem volt elég hatékony. A simítás és a szűrés eredményeként a szélessávú zaj karaktere megmaradt, viszont a nagy teljesítményű kisfrekvenciás komponensek eltűntek a jelből.

Az előzőekben ismertetett jelanalízist automatizáltam, így a MATLAB képes bármely generátor jelparamétereit kiszámítani a felvételek alapján. A „rendes” harmonikusok és szubharmonikusok számát beállíthatóvá tettem, aminek köszönhetően a

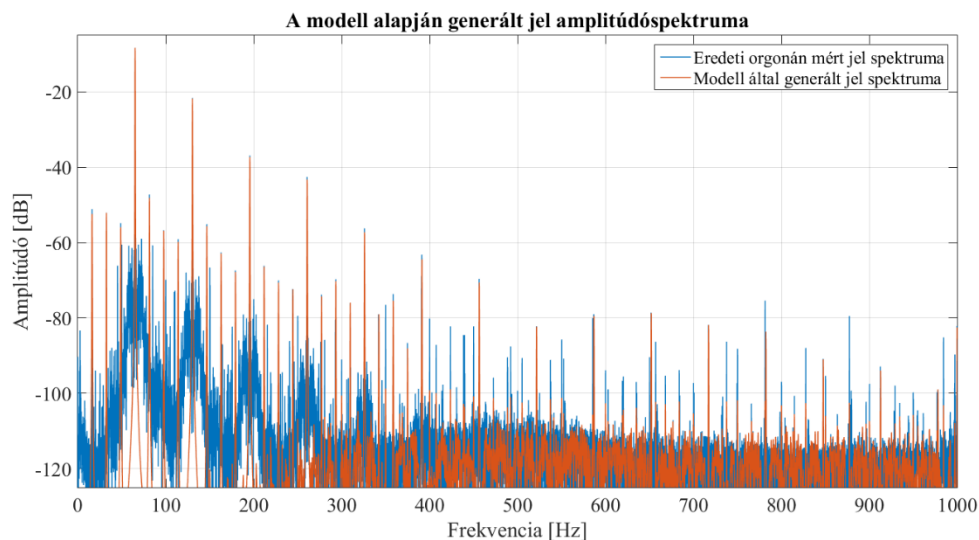
jelmodell részletessége variálható lett. A jelparaméterek és a kapott zajminta felhasználásával a generátorok jele szintetizálhatóvá vált.

### 3.2.3 Jelszintézis

A jelszintézis szintén MATLAB-ban zajlott, még hozzá a legegyszerűbb, számítástakarékos formában. A „rendes” harmonikusokat, szubharmonikusokat és a kHz-es harmonikusokat mind a koszinusz függvény diszkrét időpontokban vett értékeiként állítottam elő az alábbi (2) formában:

$$x[k] = A \cdot \cos\left(2\pi \cdot f \cdot \frac{k}{f_s} + \varphi\right), \quad (2)$$

ahol  $k$  a diszkrét időváltozó,  $x[k]$  az adott komponens  $k$ -adik időpillanatban vett értéke,  $A$  az amplitúdója,  $f$  a frekvenciája,  $\varphi$  a relatív fázishelyzete,  $f_s$  pedig a mintavételi frekvencia. A mintavételi frekvencia a jelforrás és -szintézis során egyaránt 44.1 kHz volt, ami az audioteknikában szabványosnak mondható. A jelalkotó komponensek összegzésével megkaptam a generátormodell kimeneti jelét. Minden generátormodell jeléhez olyan mértékben kevertem hozzá a „nyikorgó” zajt, hogy az eredeti hangfelvétel és a generált jel zajteljesítménye megegyezzen. (Megjegyzés: A dolgozat mellékletébe illesztettem egy hangfájlt, ami a jelmodell komponenseinek fokozatos egymásra építésével szemlélteti a modell „evolúcióját”, illetve az egyes összetevők szerepét.)



16. ábra: A generált jel amplitúdóspektruma a kisfrekvenciás tartományban

A fenti, 16. ábrán a kék görbe a 13-as számú generátorról készült eredeti hangfelvétel, a piros görbe pedig a modell által generált, zajjal ellátott jel

amplitúdóspektrumát mutatja a kisfrekvenciás tartományban. A két görbén a vizsgált, legfontosabb komponensek csúcsai fedik egymást, a zaj pedig a kb. 300 Hz feletti frekvenciasávban jelen van. Bár az eredeti jel spektrumán a „rendes” harmonikusok környezetében megjelenő „vállak” hiányoznak a generált jelből, ez mégsem jelent hallható különbséget, mivel az ottani jelteljesítmény nagy része mindkét esetben pontosan a harmonikusokra koncentrálódik.

Az állandónak tekintett frekvencia-, amplitúdó-, és fázisparaméterek megkönnyítik a 91 jelgenerátor modelljének implementálását valós idejű környezetben is. Az eddigi eredmények alapján a „nyikorgó” zaj rövid szakaszának visszajátszása adja a legjobb eredményt, mivel ez felhasználható mindegyik generátorhoz. Ebből fakadóan a valós idejű orgonamodellben elegendő a „nyikorgás” egy periódusát (az zajjal együtt, vagy további szűréseket követően) eltárolni, majd azt a generátormodellek jeleihez súlyozottan hozzákeverni.

A valós idejű alkalmazás jelenlegi verziójában (7. fejezet) a jelgenerátorokat wavetable szintetizátorok valósítják meg. Ez a megoldás jóval hatékonyabbnak bizonyult, mint a generátorok jelkomponenseinek valós idejű előállítása és keverése, ráadásul a szoftverben alkalmazott lineáris interpolációnak köszönhetően a létrehozott hang minősége is kiváló. Az applikációhoz felhasznált, 4000 elem széles hullámtáblákat a MATLAB modell harmonikus és szubharmonikus összetevői alapján állítottam elő úgy, hogy a legkisebb frekvenciájú szubharmonikus éppen kitöltse azokat. (Így mindig az alapharmonikus komponens négy periódusa szerepel egy wavetable-ben.) A hullámtáblákat minden generátor esetén külön generáltam figyelve arra, hogy a jel minden olyan felharmonikus komponens tartalmazzon, ami valamelyest a zajküszöb felett van.

*Megjegyzés:* A valós idejű alkalmazás jelenlegi verziójának jelgenerátor modellje nem tartalmazza a zajküszöb környéki „periodikus nyikorgást”, sem pedig a kHz-es harmonikusokat. Az előbbi főként azért nem építettem az alkalmazásba, mert nem ismerem a generátorok fázisbeli viszonyait. Ez a harmonikus komponensek alkotta periodikus jelek esetében – a tonewheel-ek 2.2.4. pontban említett szabadonfutói miatt – nem jelent problémát, viszont gyanús, hogy a mechanikai eredetű nyikorgás azonos fázisban vagy konstans fáziskülönbségekkel van jelen a generátorok jelében. A kHz-es harmonikusok nagy valószínűséggel egy közeli tápegységről csatolódtak a jelekre a mérések során, ezért ezeket nem tekintettem hasznos komponenseknek.

## 4 A keverőrendszer modellje

A Hammond orgonákban magát az additív szintézist a billentyűzet, drawbar-sor és illesztőtranszformátor alkotta keverőrendszer valósítja meg, amint arról a 2.2.2. pontban már szó volt. Az alábbi fejezetben a keverőrendszer működésének MATLAB-beli – és egyben valós idejű – implementációját (4.1. pont), valamint a *key click* effektus jellegzetességeit (4.2. pont) ismertetem.

### 4.1 A harmonikusok súlyozása

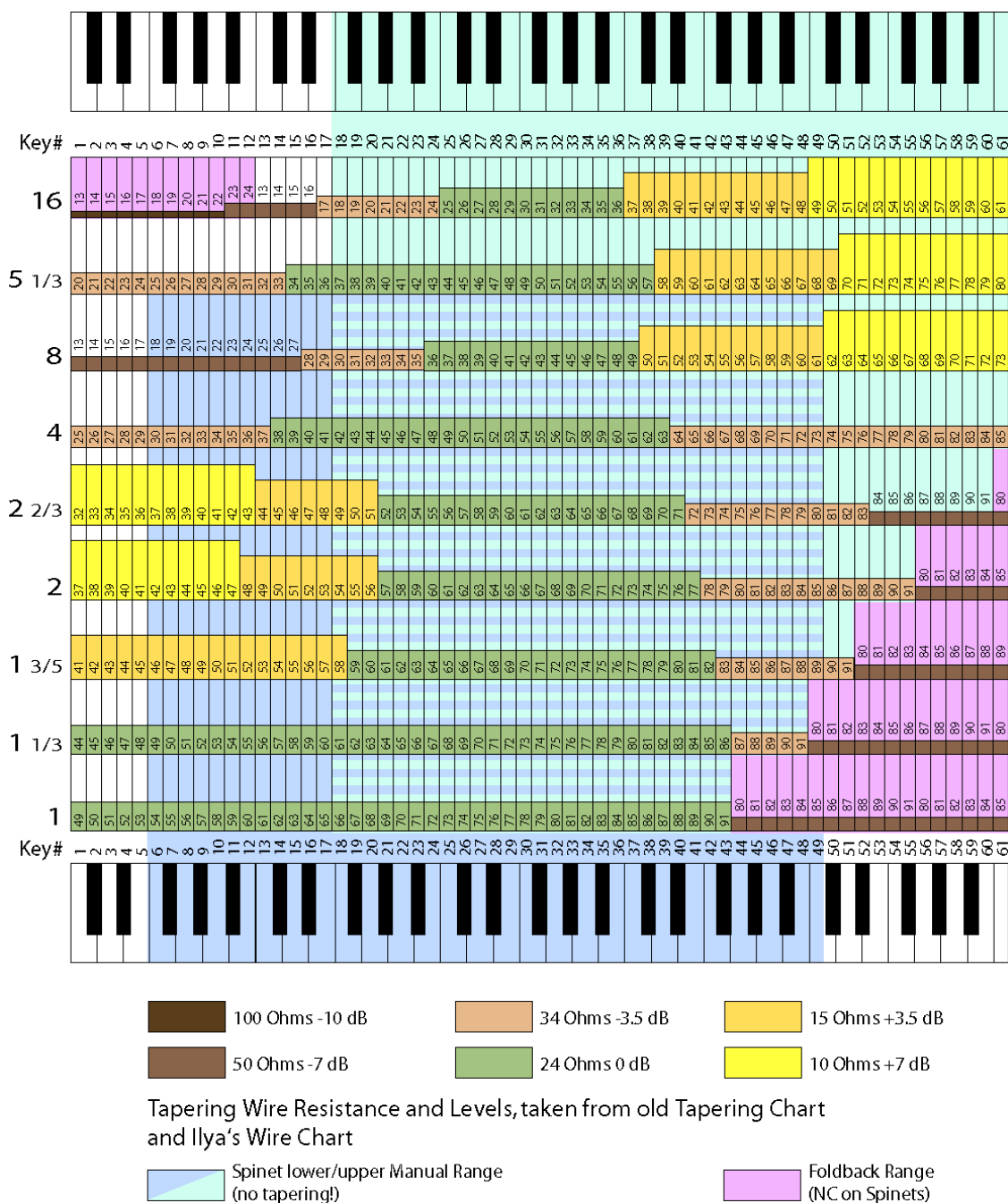
A keverőrendszer szoftveres modellezésének lehetőségei az egyszerű, jelfolyam elvű megoldásoktól kezdve a komplex, elektronikus paramétereket is figyelembe vevő rendszerekig széles skálán mozognak. Egy idealisztikus, minden részletre kiterjedő modell a billentyűk és a drawbar-ok állapota alapján, valamint az illesztőtranszformátor primer kivezetéseinek mérhető impedanciák ismeretében folyamatos számításokat végezne a generátor-kimenetek terhelésére vonatkozóan. Tekintve, hogy egy-egy generátor kimenőjelét több billentyű is felhasználja, illetve az egyazon harmonikus buszra kapcsolt generátorok – a leválasztó ellenállásokon keresztül – egymás kimeneti impedanciáit is látják, a helyzet korántsem egyszerű. (A 4.2. pontban szereplő *key click* jelenség csak ront a helyzeten.) Talán a legfontosabb kérdés az, hogy megéri-e ilyen bonyolultságú modellt használni. Mivel nem végeztem méréseket erre vonatkozóan, sajnos nem tudok kielégítő választ adni. Mindenesetre megkíséreltem a keverőrendszer implementációját az egyszerűbb úton, és az eredmény hanghatás szempontjából megfelelőnek bizonyult.

Mind a MATLAB-beli kísérletek során, mind pedig a valós idejű szoftverben ugyanazt az egyszerű algoritmust használtam, amit a billentyűsor huzalozási diagramja alapján készítettem el (17. ábra). A diagram egyszerre mutatja a billentyűk harmonikusaihoz tartozó generátor-kimenetek számát (számok), illetve a leválasztó ellenállások értékét (színek). Az ábra alján a leválasztó ellenállások értékei mellett az is szerepel, hogy azok milyen erősítést vagy csillapítást okoznak az áthaladó hangjelekre nézve. A diagram alapján már könnyű volt létrehozni azt a súlyozó-összegző algoritmust, ami a billentyűk „bemenetein” megjelenő hangjelekből előállítja a harmonikus buszok jeleit. A drawbar-ok és az illesztőtranszformátor által alkotott rendszert is kézenfekvő volt egy egyszerű összegzőként implementálni, ami a harmonikus buszok jeleit a drawbar-ok állása szerint súlyozza. Az eredeti hangszer illesztőtranszformátorának

primer köre – az emberi fül hangosságérzetéhez igazodva – exponenciális léptékben van megcsapolva, amit a modellem súlyozó tényezői is figyelembe vesznek. (Kézenfekvő volt a súlyozókat MATLAB-ban vektor- és mátrixszorzásokkal, valós időben (C++) pedig egymásba ágyazott ciklusokon belül, tömbértékek szorzásával megvalósítani.

## Hammond Console Manual Wiring and Tapering Scheme

by Carsten Meyer, [www.tonewheel.de](http://www.tonewheel.de)



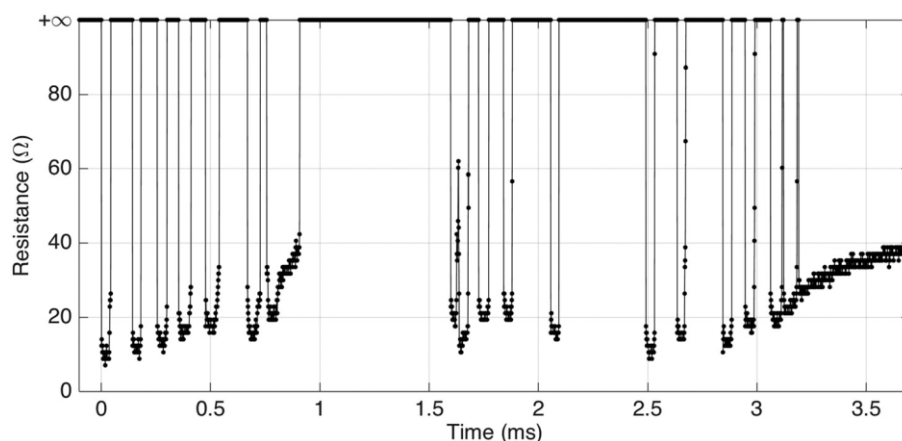
17. ábra: A billentyűzet huzalozási és súlyozási diagramja [18]

## 4.2 A *key click* effektus

A Hammond orgona hangzásvilágának egyik nélkülözhetetlen összetevőjét az ún. *key click* effektus adja, amely a billentyűzet tökéletlen működéséből fakad. Az alábbi alfejezetben részletesebben körüljáróm a *key click* jelenség létrejöttét és hatását, valamint bemutatok egy egyszerű szoftvermodellt, ami – közelítőleg – képes reprodukálni az effektust.

Amint azt korábban láthattuk, az egyes billentyűk alatt 9-9 mechanikus kontaktus található, amelyek egy billentyű lenyomásakor az adott zenei hanghoz tartozó harmonikus összetevőket a harmonikus buszokra kapcsolják. Ezek a kontaktusok természetesen nem az ideális kapcsolóktól elvárt módon viselkednek, vagyis nem egyszerre záródnak és – bár rövid ideig, de – peregnek. (Olyannyira szignifikáns jelenségről van szó, hogy ha egy billentyűt lassan nyomunk le, akkor az egyes harmonikus összetevők megjelenése külön-külön hallhatóvá válik.) Az időben eltolódott érintkezés és a pergések hatására minden billentyű lenyomása alkalmával rövid, ám összetett tranziens jelenség játszódik le, amelyet a hangszerjátékos enyhe „puffannásként” érzékel a létrejött hangban [19], [20], [21]. (Innen származik a „*key click*” elnevezés is.) Az előbbiekből következik, hogy a billentyűk leütésének időbeli lefolyása – különös tekintettel az akár változó leütési sebességre – alapvetően befolyásolja a tranziens milyenségét, így a létrejövő hanghatást is [22]. (A [23] videón Giulio Moro szemlélteti a billentyűütés dinamikájának változtatásával létrehozható hangeffektusokat.)

A *key click* effektus két legfontosabb tényezője a billentyűk alatti kontaktusok záródásának időbeli eltolódása, illetve a pergésük lefolyása. A kontaktusok záródását alapvetően az érintkezők felfüggesztéséért felelős rugalmas fémszalagok geometriája, beállítása, elhasználódása és egyéb sajátosságai határozzák meg. A 18. ábra egy kontaktus ellenállásának változását szemlélteti pergés közben. Szembetűnő, hogy a folyamat során a végtelen ellenállás (nyitott kapcsoló) és az állandósult állapotbeli kb. 20 – 40 ohmos érték (zárt kapcsoló) közti átmeneti állapotok időtartama – az 1 MHz-es mintavételi frekvencia ellenére – gyakorlatilag mérhetetlenül rövid [22]. Ez a tulajdonság jelentősen leegyszerűsíti a pergési jelenség modell alapú közelítését.



18. ábra: Egy kontaktus ellenállásának változása a pergési folyamat közben [22]

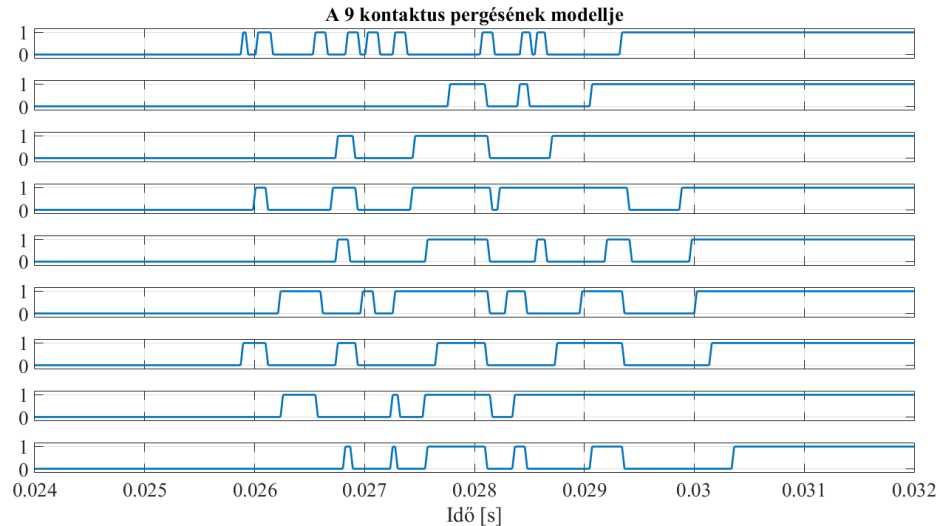
Az [22] cikk a billentyűütések sebessége és a létrejövő key click effektus tulajdonságai között keres összefüggéseket. Mivel nem volt alkalmam saját méréseket végezni a key click hatás körülményeinek felderítésére, ezért az említett cikk alapján kíséreltem meg a saját szoftvermodellem összeállítását, amely a jelenség lefolyását igyekszik közelíteni. A cikk eredményei nyomán kidolgozott egyszerű valószínűségi modell a következő valószínűségi változókat tartalmazza:

- $T_{offset}$  (kontaktusra jellemző) – Az ujjbegy és a billentyű érintkezésétől kezdve az adott kontaktus pergésének kezdetéig (az első záródás időpontjáig) eltelt időtartam átlagos leütési sebesség mellett. Normális eloszlású valószínűségi változó 26.31 ms várható értékkel és 0.41 ms szórással.
- $T_{bounce}$  (kontaktusra jellemző) – A pergés időtartama, vagyis a pergés kezdetétől az zárt állapot állandósulásáig eltelt időtartam átlagos leütési sebesség mellett. Normális eloszlású valószínűségi változó 3.03 ms várható értékkel és 0.65 ms szórással.
- $T_{temp\_on}$  (pergési ütemre jellemző) – Pergés közben egy-egy zárt állapotú szakasz várható időtartama. 0.036 ms és 0.143 ms között egyenletes eloszlású valószínűségi változó.
- $T_{temp\_off}$  (pergési ütemre jellemző) – Pergés közben egy-egy nyitott állapotú szakasz várható időtartama. 0.020 ms és 0.696 ms között egyenletes eloszlású valószínűségi változó.

*Megjegyzés:* A cikkben nem szerepeltek a valószínűségi eloszlásokra vonatkozó konkrétumok, és én sem végeztem méréseket ezzel kapcsolatban, ám szükségem volt

kiindulási adatokra a modell elkészítéséhez, ezért „kontár” eljáráshoz folyamodtam. A modellben használt valószínűségi változók eloszlásait csak a sejtéseimre, azok paramétereit pedig a [22] cikk ábráinak időtengelyein vonalzóval végzett arányításaimra alapoztam. Kézenfekvőnek tűnt, hogy az orgona billentyűi alatt található flexibilis fémlemezeket a gyártáskor valamilyen pontossággal, a méretezéshez képest valamilyen szórással megmunkálták. Valószínűleg ilyen – általában normál eloszlással modellezhető – gyártásból fakadó eltérések befolyásolják a  $T_{offset}$  késleltetést és a  $T_{bounce}$  pergési időt. A  $T_{temp\_on}$  és a  $T_{temp\_off}$  pedig csupán azért lett egyenletes eloszlású, mert egyszerű volt az ábrákon az ideiglenesen zárt és nyílt időtartamok legrövidebb, illetve leghosszabb értékeit leolvasni. Ezen felül a nagyfrekvenciás pergés során sokkal kevésbé lényeges ezek eloszlása, mint nagyságrendje. (A törtszámok a vonalzón mért hosszúságadatok és a referencia időegység szorzása miatt adódtak.)

A MATLAB-ban implementált key click modell működése: A modell egy billentyű leütésekor a 9 kontaktus viselkedésére jellemző burkológörbéket állít elő, amelyekkel a megfelelő – generátorokból érkező – jeleket modulálva megkapjuk a harmonikus buszokon megjelenő hangjeleket. A burkolók konstruálását megelőzően szükséges a  $T_{offset}$  és a  $T_{bounce}$  időtartamok ismerete, amelyeket normális eloszlású véletlenszám generátorok állítanak elő minden egyes kontaktus számára a megadott várható érték és szórás paraméterek alapján. Egy kontaktus burkolója a hozzá tartozó  $T_{offset}$  ideig 0 értékű mintákat tartalmaz (nyílt kapcsoló), majd ezután  $T_{bounce}$  ideig zajlik a pergés folyamata, amit az állandósult állapotbeli 1 értékű minták követnek (zárt kapcsoló). A pergés folyamán zárt és nyílt állapotú szakaszok váltakoznak egymást követően, amelyek időtartamát egyenletes eloszlású randomgenerátorok szolgáltatják (a korábban ismertetett időkorlátok között): ha a kontaktus korábban zárt állapotban volt véletlen  $T_{temp\_on}$  (ideiglenesen zárt) ideig, akkor működésbe lép a véletlenszám generátor és megadja a következő ideiglenesen nyílt állapot  $T_{temp\_off}$  időtartamát, majd pedig – ha ez letelt – újabb  $T_{temp\_on}$  értéket generálunk, és így tovább. Ha a pergés folyamata elérte vagy meghaladta a  $T_{bounce}$  időtartamot, akkor beáll az állandósult zárt állapot, ahogy az korábban is szerepelt. A modell egyik futtatása alkalmával létrejött 9 darab burkológörbét a 19. ábra szemlélteti.



**19. ábra: A key click effektus modelljének működése egy billentyűleütéskor**

Az imént ismertetett key click modell alkalmazhatóságát csupán szubjektív, hallgatásos tesztekkel tudtam ellenőrizni, viszont azok alapján az eredmény megfelelőnek bizonyult. Az eredeti orgona billentyűütést követő tranziense teljesen hasonló „puffanást” okozott a modell által generált effektushoz. Annak ellenére, hogy a valósághoz viszonyítva a pergés folyamatát a modellünk jóval egyszerűbben képezi le, az elért hanghatás alapján nem érdemes ennél összetettebb algoritmust alkalmazni. (*Megjegyzés:* A dolgozat mellékletébe illesztettem egy hangfájlt, ami lehetővé teszi az eredeti hangszerrel rögzített billentyűleütések, a MATLAB modellel generált hangminták, illetve egy key click effektus nélkül generált minta összehasonlítását.)

*Megjegyzés:* Az előbb ismertetett key click modellt érdemi módosítás nélkül átültettem a JUCE rendszerben fejlesztett valós idejű alkalmazásba is (7. fejezet). Sajnos a kontaktusonként intenzíven használt véletlenszám generátorok túl nagy számítási igényt támasztottak a számítógépem – egyébként korszerű, nagysebességű – processzorával szemben. A key click modell működtetéséhez kevésnek bizonyult a hangminták előállításához rendelkezésre álló idő, aminek következtében a kimeneti hangjel szaggatottá vált. A modell egyszerűsítése lehetséges lenne, ha például sokféle előre generált pergési folyamatot tárolnák tömbökben, amelyek közül minden billentyűütéskor egyetlen véletlengenerátor választana. Sajnálatos módon az említett alternatív megoldás megvalósítására még nem került sor.

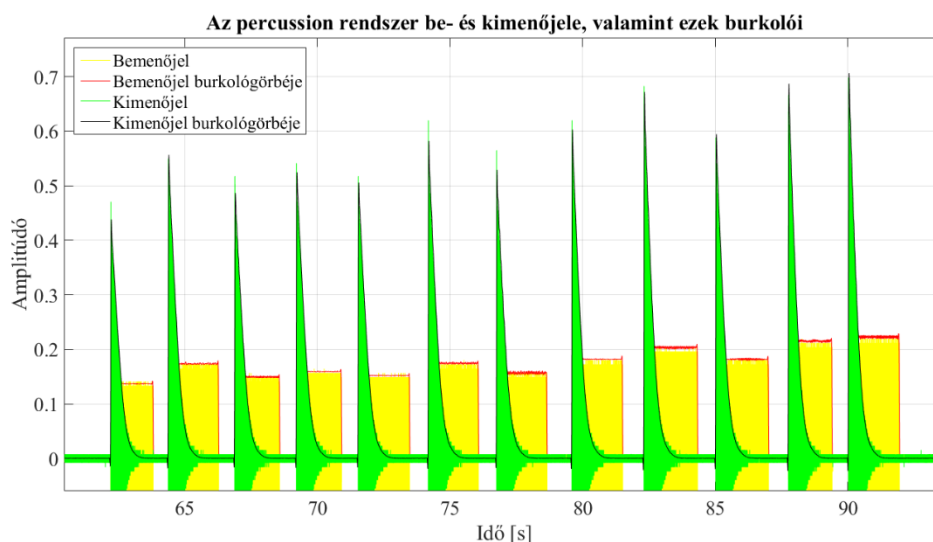
## 5 A *percussion* funkció modellje

Az alábbi fejezet a *percussion* rendszer mérésalapú elemzésével foglalkozik. Az itt ismertetett eredmények egy az egyben felhasználásra kerültek a valós idejű alkalmazás fejlesztése során.

A *percussion* áramkör működésének felderítése céljából a külső hangkártya segítségével végeztem méréseket az eszközön. A hangkártya egyik csatornáját a *percussion* jelforrására (bemenetére), a másik csatornáját pedig a *percussion* áramkör kimenetére kapcsoltam. Ezt követően működtettem az effektet, vagyis a drawbar-ok teljesen „betolt”, néma állása mellett egymás után lenyomtam a billentyűket néhány másodpercig. A mérést megismételtem a választókapcsolók más beállításai mellett, így információhoz jutottam azok hatásáról is.

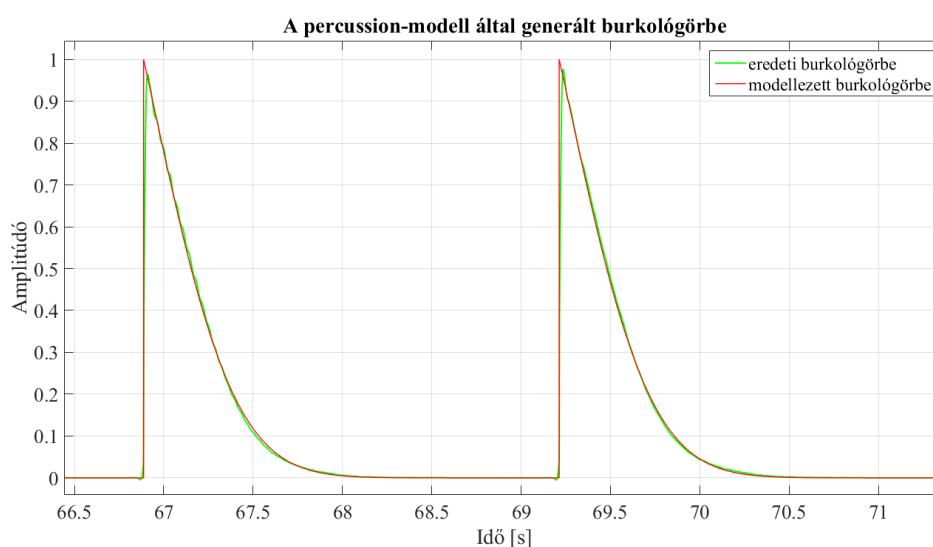
*Megjegyzés:* A most következő elemzés ábrái a *percussion* rendszer *normal* hangereje és *slow* lecsengése mellett készültek; forrásjelként a harmadik harmonikus (*third*) volt beállítva. Az analízis során természetesen megvizsgáltam a többi beállítást is, de a nyilvánvaló különbségeken kívül hasonló eredményekre jutottam. (Például a *soft* hangerő kb. hatodakkora jelszintet jelent, mint a *normal*. A *fast* beállítás mellett a lecsengés nagyjából háromszor gyorsabb, mint a *slow* esetében, de a burkolók jellege egyforma.)

A MATLAB-beli vizsgálatok során nem volt más dolgom, mint előállítani a rögzített be- és kimeneti hangminták burkolóit, majd pedig elemezni azok tulajdonságait. A burkológörbék elkészítéséhez a jelek abszolút értékét egy 25 Hz vágási frekvenciájú, másodfokú, Butterworth IIR aluláteresztő szűrővel szűrtem. A szűréshez a *filtfilt* függvényt használtam, hogy a jel fázisviszonyai változatlanok maradjanak. A szűrt jel amplitúdójának csökkenését úgy kompenzáltam, hogy a burkolók fedjék az eredeti jel körvonalát. A 20. ábrán az orgona billentyűsorának középső oktávjához tartozó jelek, valamint ezek burkolói figyelhetők meg.



20. ábra: A burkológörbék létrehozása

*Megjegyzés:* A mérések során az eredeti jelek „darabosságán” is tetten érhető furcsa számábrázolási hibát produkálta a hangkártya, amire csak később derült fény. Ennek ellenére a burkolók előállítása sikeres volt, mivel ez a feladat nem igényelt különösebb pontosságot.



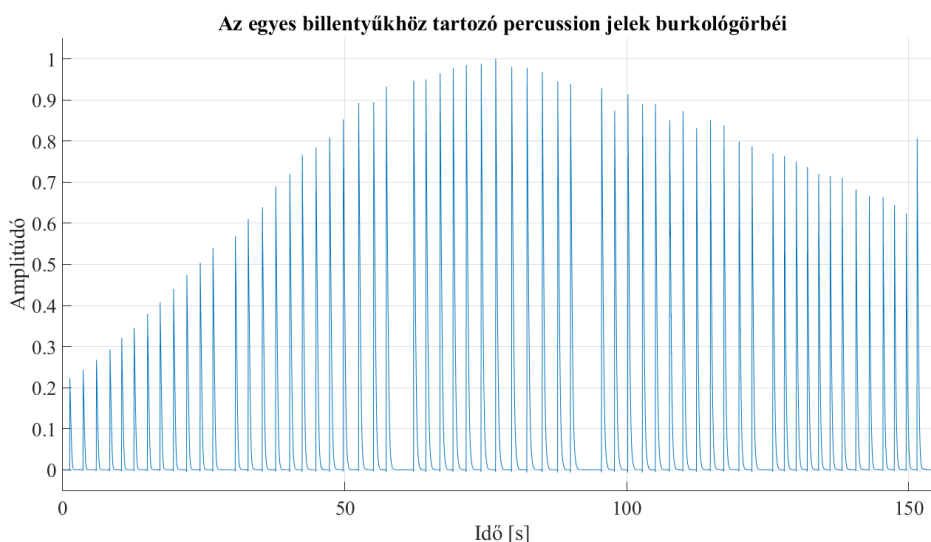
21. ábra: A percussion burkológörbéjének közelítése

A 21. ábra zöld görbéje billentyűleütésekhez tartozó burkolókat mutat a percussion rendszer kimenetén. A lecsengés időbeli lefolyásának analízise céljából – egy ilyen rövid szakaszon – a mért jel burkolójának természetes alapú logaritmusát idő szerint deriváltam, majd a görbét ábrázoltam. A várt, tisztán exponenciális lecsengéshez tartozó közel konstans érték helyett lineárisan csökkenő görbét kaptam. Az említett lineárisan csökkenő függvény az  $k$  diszkrét időváltozó  $\sigma[k]$  együtthatójának felel meg, ha

konvencionális módon, exponenciális formában (3) írjuk fel a lecsengés diszkrét időfüggvényét. Az eredmények alapján elkészítettem a percussion rendszer digitális modelljét:

$$a[k] = A \cdot e^{\sigma[k] \cdot k}, \quad (3)$$

ahol  $k$  a billentyűlenyomás óta eltelt diszkrét idő,  $a[k]$  a generált burkolójel diszkrét időfüggvénye,  $A$  a *percussion volume* beállítástól függő konstans erősítés,  $\sigma[k]$  pedig az diszkrét időváltozó együtthatója, amit a megismert,  $k$  szerint lineárisan csökkenő függvény reprezentál. A percussion forrásaként szolgáló harmonikus busz jelét a képlet alapján generált  $a[k]$  burkolóval szorozva megkapjuk a modell kimenőjelét. Az így kapott eredményt a 21. ábrán szereplő piros görbe szemlélteti, amelyről elmondható, hogy igen jól közelíti a mért lecsengő jel burkolóját. Ezt a modellt használtam később a valós idejű alkalmazásban (7. fejezet) is: A modell  $\sigma[k]$  együtthatóját megvalósító kétféle lineáris függvény közül a lecsengés gyorsaságát beállító *slow* és *fast* gombok választják ki a megfelelőt. Ugyanígy az  $A$  erősítés kétféle értéke közül a *normal* és *soft* gombokkal választhatunk. Természetesen a percussion jelforrása is hasonló módon kiválasztható. (A valós idejű modell annyival képes többre ennél, hogy a billentyűzet állapotának változása alapján előállítja a saját triggerjelét, valamint a triggerből számítja a  $k$  időt.)



**22. ábra: Az egyes billentyűkhöz tartozó percussion jelek burkológörbéi**

A 22. ábra azt mutatja, hogy a percussion generátor áramkörének átvitele is frekvenciafüggő, ugyanis az egyes billentyűk balról jobbra – vagyis legmélyebbtől legmagasabb hangig történő – egymás utáni lenyomása különböző amplitúdójú lecsengő

jeleket eredményez. (Az ábrán látható szituációban a percussion generátor kimenetének burkolóit már a „forrás” oldalán mért bemenőjeleknek megfelelően normalizáltam, így nem okozhatja a jelenséget az egyes billentyűkhöz tartozó bemeneti jelszintek eltérése.) Ezt a jelenséget a valós idejű applikáció fejlesztésekor figyelmen kívül hagytam, mivel csak kis mértékben befolyásolja a létrejött hanghatást, így egy dedikált percussion szűrő tervezése nem lenne kifizetődő.

## 6 Az előerősítő modellje

Az 2.2.4. alfejezetben leírtakból következik, hogy az előerősítő – jobb esetben nemlineáris viselkedését is figyelembe vevő – modellezése mindenképpen szükséges, amennyiben a Hammond orgonák hangzását igazán hűen szeretnénk visszaadni. Jelen fejezet 6.1. pontjától a 6.4. pontig az előerősítő lineáris átvitelének és nemlineáris karakterisztikájának integrált modellezéséről lesz szó. A 6.5. pontban röviden ismertetem azt az eljárást, amivel az erősítő MATLAB modelljében használt, nagy foksámú digitális szűrőt erőforrás-hatékonyabb struktúrájára cseréltem. Végül a 6.6. alfejezet az erősítő részét képező *expression control* pedál működésének vizsgálatával és szoftveres megvalósításával foglalkozik.

### 6.1 Modelltípus kiválasztása

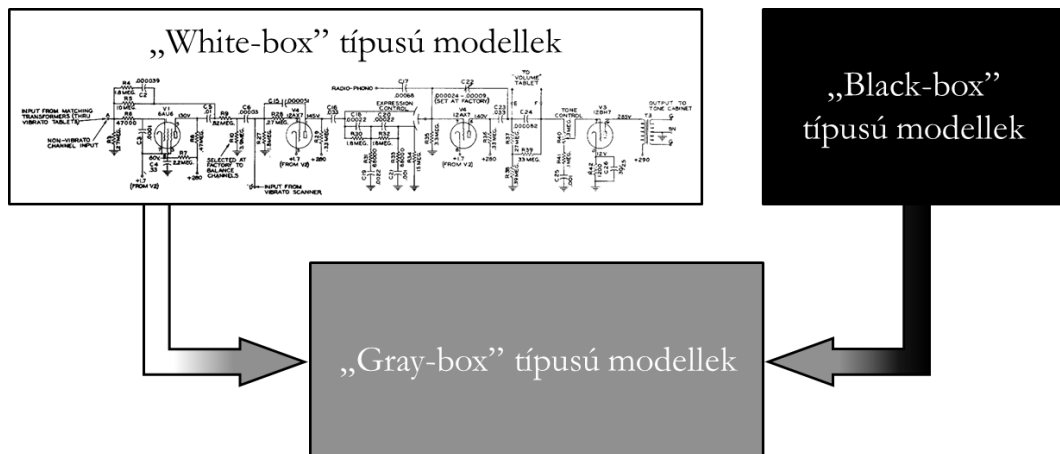
A nemlineáris rendszerek modellezési módszerei rendkívül szerteágazóak [24]. A céljainknak megfelelő struktúra kiválasztása [25] során figyelembe kell vennünk, hogy:

- Milyen mélységben rendelkezünk előzetes ismeretekkel a modellezendő rendszerről?
- Rendelkezésünkre áll-e egy vagy több referenciapéldány?
- Milyen típusú és milyen részletes mérésekre van lehetőségünk?
- Mik a korlátaink a modell későbbi implementálása során?

A modellezési módszereket általában két főbb paradigma mentén szoktuk megkülönböztetni: A „*white-box*” (*fehér doboz*) modellezési elv abból indul ki, hogy pontosan ismerjük az eredeti rendszer felépítését és működését. Ezzel szemben a „*black-box*” (*fekete doboz*) modellek esetén csak „a sötétben tapogatózva”, pusztán mérések alapján tudunk következtetéseket levonni a rendszer működését illetően.

Esetünkben a fehér doboz elv megfeleltethető annak, hogy létrehozzuk az előerősítő áramköri kapcsolásának diszkrét idejű leírását, majd a rendszeregyenletek numerikus, iteratív megoldásával működtetjük a modellt [26]. Egyrészt a modellezendő előerősítő bonyolultsága, másrészt az áramköri elemek karakterisztikáinak nehéz fellelhetősége, harmadrészt pedig az így implementált modell erőforrásigénye miatt elvettem a fehér doboz elv alkalmazását.

A tisztán fekete doboz elvű megközelítések – például a Volterra-sor alkalmazása – előnyt jelentenek abból a szempontból, hogy a modellillesztés viszonylag egyszerű mérések alapján kivitelezhető, illetve az elkészült modell könnyen implementálható [27], viszont hátrányt jelenthetnek amiatt, hogy a modell túl általános, nem specifikusan az adott alkalmazásra készült, így nem közelíti „jó hatásfokkal” a rendszerünk viselkedését.



23. ábra: Modellezési paradigmák fúziója

Az előző két bekezdésben leírtakból kiindulva kompromisszumos megoldást kerestem, amely ötvözi a két alapelv előnyeit, így jutottam az ún. *kiterjesztett Wiener-Hammerstein modellig*, amelyet *Felix Eichas* és *Udo Zölzer* fejlesztett ki a hamburgi *Helmudt Schmidt Egyetemen* [28]. A kiterjesztett Wiener-Hammerstein modell a fekete doboz típusra emlékeztet abból a szempontból, hogy nem igényli az áramkör pontos ismeretét és az adaptációja csupán átvitelméréseket vesz igénybe, viszont fehér doboz modellnek is mondható, mivel kifejezetten elektroncsöves erősítők lineáris és nemlineáris hatásainak modellezésére fejlesztették ki. (A fejlesztők az előbb említett kettőssége miatt – és mivel jól is hangzik – *szürke doboz* típusú (amerikaiában „gray-box”) modellként hivatkoznak rá (23. ábra).

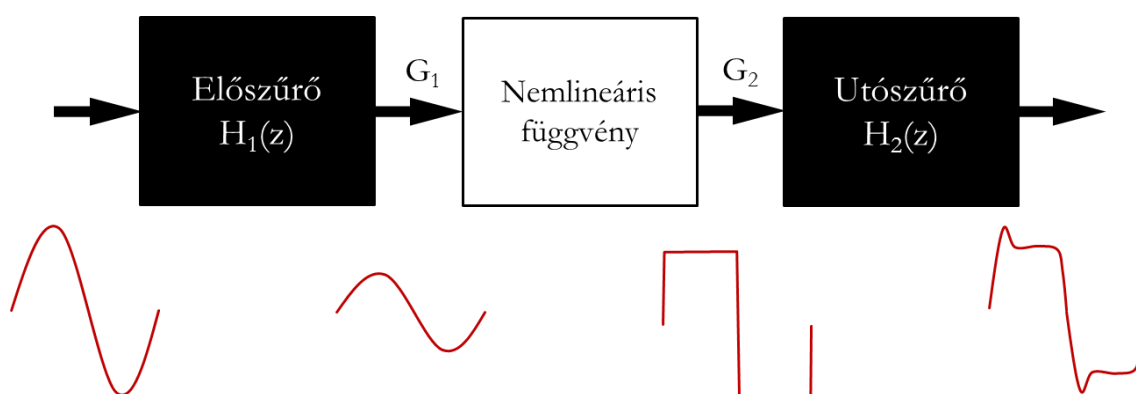
## 6.2 A kiterjesztett Wiener-Hammerstein modell

### 6.2.1 A modell felépítése

A nemlineáris rendszerek viselkedésének szimulálására elterjedten használják a blokkalapú Wiener- és Hammerstein-modellt. Mindkét modell egy lineáris szűrőből és egy statikus nemlinearitásból épül fel [28]. A Wiener-modell esetén a bejövő jel előbb a lineáris szűrőn halad keresztül, majd pedig a nemlineáris függvény torzítja azt; a

Hammerstein-modellnél mindez fordítva történik. A két modell vitathatatlan előnye az egyszerűség, viszont nem minden esetben vezetnek megfelelő eredményre, mivel a modellezendő rendszerek nemlineáris viselkedése gyakran több részegység – pl. egy erősítő esetén több aktív áramköri elem, telítésbe vitt transzformátor, stb. – komplex torzító hatásaként jön létre.

A kiterjesztett Wiener-Hammerstein modell az egyszerű Wiener-modell és Hammerstein-modell adta lehetőségeket azok ötvöztetésével bővíti ki. A kiterjesztett modell két lineáris szűrőtagból és a közöttük elhelyezett statikus nemlineáris függvényből áll, ahogy azt a 24. ábra is mutatja.



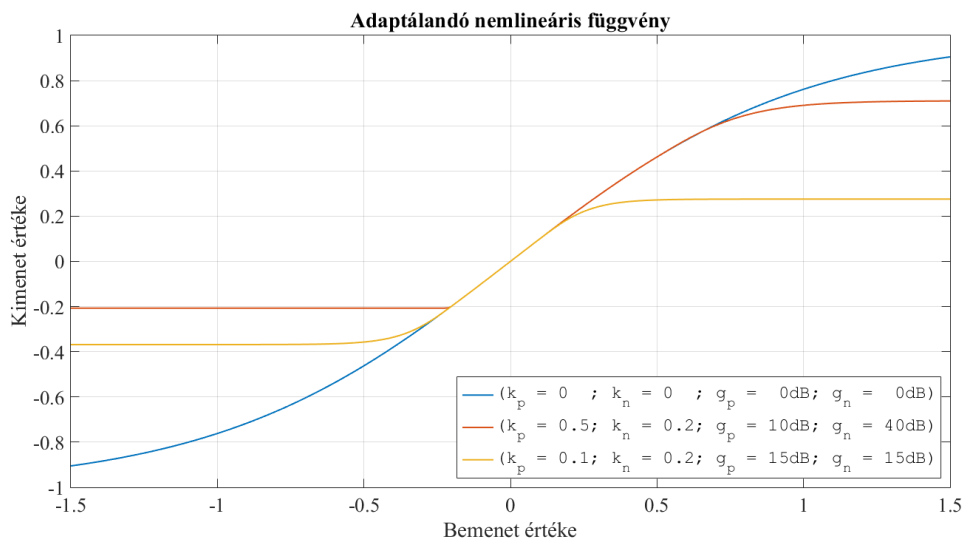
24. ábra: A kiterjesztett Wiener-Hammerstein modell felépítése és működése

A modell működési elvét a 24. ábrán szereplő blokkvázlat alatt feltüntetett szimbolikus hullámformákon keresztül érthetjük meg a legkönnyebben: A beérkező jel keresztülhalad a  $H_1(z)$  átviteli függvénnyel leírható *előszűrőn*, ami meghatározza, hogy a jel egyes frekvenciakomponensei milyen súllyal kerülnek a nemlineáris függvény bemenetére. Ennélfogva a modellezett rendszer által létrehozott harmonikus, illetve intermodulációs torzítások frekvenciafüggő mértéke döntően az előszűrő átvitelétől és az ábrán is feltüntetett  $G_1$  erősítés értékétől függ.

Az előszűrőből kilépő,  $G_1$ -szerezésre erősített jelet a hiperbolikus tangens jellegű nemlineáris függvény torzítja, amely az alábbi (4) képlettel írható le [28]:

$$y(x) = \begin{cases} \tanh(k_p) - \left[ \frac{\tanh^2(k_p) - 1}{g_p} \tanh(g_p(x - k_p)) \right] & \text{ha } x > k_p \\ \tanh(x) & \text{ha } -k_n \leq x \leq k_p \\ -\tanh(k_n) - \left[ \frac{\tanh^2(k_n) - 1}{g_n} \tanh(g_n(x + k_n)) \right] & \text{ha } x < -k_n \end{cases} \quad (4)$$

ahol  $x$  a nemlineáris függvény argumentuma (szűrt, erősített bemenőjel aktuális mintájának értéke),  $y$  a függvény értéke (függvény által torzított jel aktuális mintájának értéke),  $k_p$ ,  $k_n$ ,  $g_p$  és  $g_n$  pedig a függvény hangolható paraméterei, amelyek az alsó indexükben szereplő  $p$  és  $n$  jelöléseknek megfelelően a pozitív és negatív szaturációs régiók jellegét határozzák meg. A  $g_p$ , illetve  $g_n$  paraméterek növelésével a pozitív, illetve negatív félhullámok egyre „meredekebb, durvább töréssel” lépnek a szaturációs régiókba, így a létrehozott harmonikus torzítás mértéke jelentősen megnő. A  $k_p$  és  $k_n$  paraméterekkel beállíthatjuk, hogy ezek a töréspontok mekkora bemeneti jelszintjénél (az  $x$  argumentum mekkora értékénél) helyezkedjenek el. (A  $k$  paraméterek gyakorlatilag a hiperbolikus tangens függvény töréspontjain – a  $g$  paraméterek által okozott – szintcsökkenést kompenzálják.) A 25. ábrán a nemlineáris függvény alakját ábrázoltam a hangolható  $k_p$ ,  $k_n$ ,  $g_p$  és  $g_n$  paraméterek néhány konfigurációja esetén.



25. ábra: Az adaptálandó nemlineáris függvény

A nemlineáris függvény által torzított jel végül  $G_2$  értékű erősítést követően az *utósűrőn* halad át, amelynek átviteli függvényét  $H_2(z)$ -vel jelöltem a 24. ábrán. Az utósűrő a torzítás következtében létrejött felharmonikusok átvitelét szabályozza, korlátozza.

## 6.2.2 A modell adaptálása a modellezendő rendszerhez

A kiterjesztett Wiener-Hammerstein modell a következő módon adaptálható a modellezendő rendszerhez [28]: Először meg kell határoznunk az elősűrő és az utósűrő

átviteli függvényeit, amelyekhez célszerűen átvitelmérések révén juthatunk. Ezt követi a nemlineáris függvény paramétereinek hangolása.

A két szűrő átviteli függvényeinek meghatározásához két különböző jelszinten kell megmérnünk a modellezendő rendszer lineáris átvitelét. Az első mérés alkalmával a gerjesztés jelszintjét olyan kicsire kell választanunk, hogy ne lépjen fel jelentős torzítás a kimeneten. Ennek eredményeképp megkapjuk a  $H_{low}(z)$ -vel jelölt kisjelű átviteli függvényt. A második mérés során akkora jelszinten kell gerjesztenünk a rendszert, amekkora a bemenetén maximálisan előfordulhat. Ebben az esetben már jelentős a torzítás mértéke, ezért az itt kapott átviteli függvényből eliminálnunk kell a harmonikus torzítás hatását; így megkapjuk a  $H_{high}(z)$ -vel jelölt nagyjelű, lineáris átvitelt. Az alábbi két képlet alapján meghatározható az előszűrő (5) és az utószűrő (6) átvitele a két mérés során számított átviteli függvények felhasználásával:

$$H_1(z) = \frac{H_{low}(z)}{H_{high}(z)}, \quad (5)$$

$$H_2(z) = H_{high}(z), \quad (6)$$

ahol  $H_1(z)$  az előszűrő,  $H_2(z)$  pedig az utószűrő átvitele, ahogy azt az 6.2.1. pontban is jelöltük. A kiterjesztett Wiener-Hammerstein modell sokoldalúsága – és az egyszerűbb Wiener- vagy Hammerstein-moddellel szembeni előnye – éppen abban rejlik, hogy maguk az (5) és (6) alapján kapott szűrők határozzák meg a rendszer nemlinearitásának jelűbeli elhelyezkedését. Ha a mért  $H_{low}(z)$  és  $H_{high}(z)$  átviteli függvények megegyeznek, akkor az előszűrő átvitele egységnyi lesz, vagyis a rendszerünk szimpla Hammerstein-moddellel helyettesíthető. Ha viszont a kimenőjel akkora torzítást szenved, hogy szinte frekvenciafüggetlenül szaturál (ekkor a mért  $H_{high}(z)$  átvitel közel egységnyi), akkor Wiener-modellt kapunk. Ebből látható, hogy a kétszeri mérést, valamint a modellben alkalmazott két szűrő alkalmazását a nemlinearitás gyakori frekvenciafüggése indokolja. (A képletek alapján az is feltűnhet, hogy nemlineáris függvény nélkül a két szűrő eredő átvitele megegyezne a  $H_{low}(z)$  kisjelű, lineáris átvittel.)

Miután megkaptuk az előszűrő és az utószűrő átvitelét, össze kell állítanunk a teljes modellt, majd pedig meg kell határoznunk a  $k_p$ ,  $k_n$ ,  $g_p$ ,  $g_n$ , valamint a  $G_1$  és  $G_2$  paraméterek értékét. A kiterjesztett Wiener-Hammerstein modell kifejlesztői azt javasolják, hogy paraméterek hangolását egy hibakritérium-függvény minimalizálásával végezzük. A kritériumfüggvényt az eredeti rendszer nagyjelű gerjesztés hatására adott

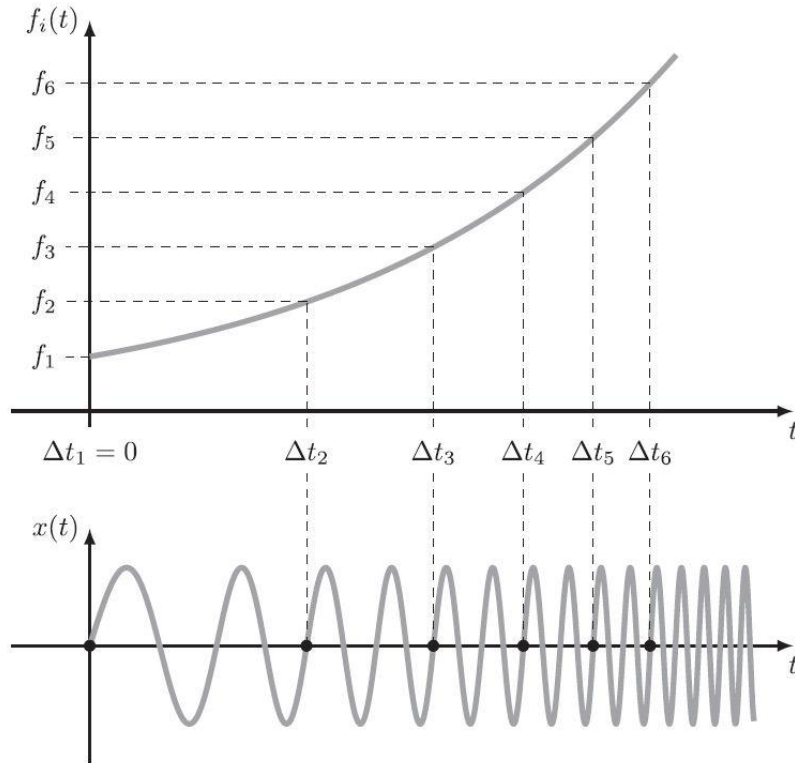
kimenetének és az ugyanígy gerjesztett modell kimenetének időablakonkénti spektrális felbontásainak négyzetes hibái alapján javasolják kiszámítani. Ezzel ellentétben én egy egyszerűbb kritériumfüggvénnyel próbálkoztam, amelyről majd az 6.4.2. pontban ejtek szót.

## 6.3 Átviteli függvények mérése

### 6.3.1 A logaritmikus sweep mérőjel

Az előerősítő kisjelű és nagyjelű lineáris átvitelének méréséhez olyan mérési eljárásra volt szükségem, amely egyrészt a teljes hallástartományban pontos képet ad a rendszer átviteléről, másrészt egyszerűen kezelhető, determinisztikus jeleket alkalmaz, így gyorsan kiértékelhetővé teszi az eredményeket, harmadrészt lehetőséget biztosít a rendszer lineáris átvitelének és a harmonikus torzításból származó hatásoknak a szétválasztására.

A leírt követelményeknek az ún. *logaritmikus sweep* jel [29], [30], [31] felel meg leginkább, amely az emberi hallástartomány egészét „végigpásztázza” és ez alatt – az emberi hallás sajátosságainak megfelelően – logaritmikus frekvenciaskála mellett egyenletesen adja le az energiáját. A logaritmikus sweep elnevezés egy exponenciális léptékben növekvő frekvenciájú szinuszhullámot takar. Az exponenciális növekedés hatására oktávnyi frekvenciaváltozás mindig egységnyi idő alatt következik be, így a mérőjel követi a hallásunk frekvenciafelbontását. A log. sweep jelek természetét jól szemlélteti a 26. ábra.



26. ábra: A log. sweep jel időfüggvénye és frekvenciájának változása [31]

### 6.3.2 A mérőjel előállítása

A logaritmikus sweep típusú mérőjel előállítása során Novak, Lotton és Simon publikációja [31] alapján jártam el, amely ismerteti az általuk *synchronized swept-sine* néven emlegetett gerjesztőjel használatának előnyeit a hagyományos log. sweep jelekkel [30] szemben. A [31] cikk módszere abban tér el a korábbiaktól, hogy a mérőjel előállításakor ún. fázisbeli szinkronizációt alkalmaz. Ez abban nyilvánul meg, hogy a jel kezdeti frekvenciájának többszöröseit elérve mindig nullátmenetet produkál, ami hasznos lesz majd számunkra a lineáris és a felharmonikus átvitelek pontos szétválasztásában. A jel előállításához szükséges képletek (7, 8, 9) levezetés nélkül:

$$x(t) = \sin \left[ 2\pi f_1 L \cdot \exp \left( \frac{t}{L} \right) \right] ; \quad (7)$$

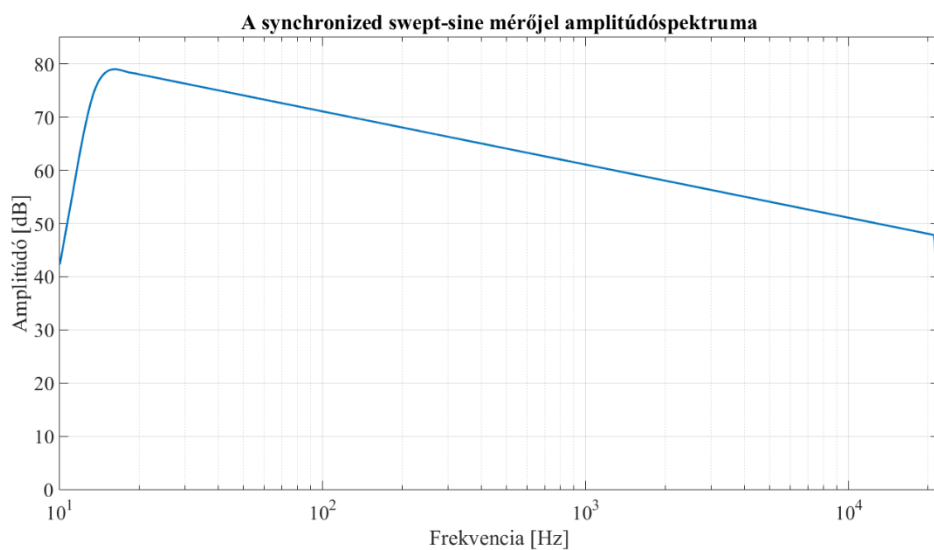
$$L = \frac{1}{f_1} \cdot \text{round} \left[ \frac{f_1}{\ln \left( \frac{f_2}{f_1} \right)} \cdot \tilde{T} \right] ; \quad (8)$$

$$T = L \cdot \ln \left( \frac{f_2}{f_1} \right) . \quad (9)$$

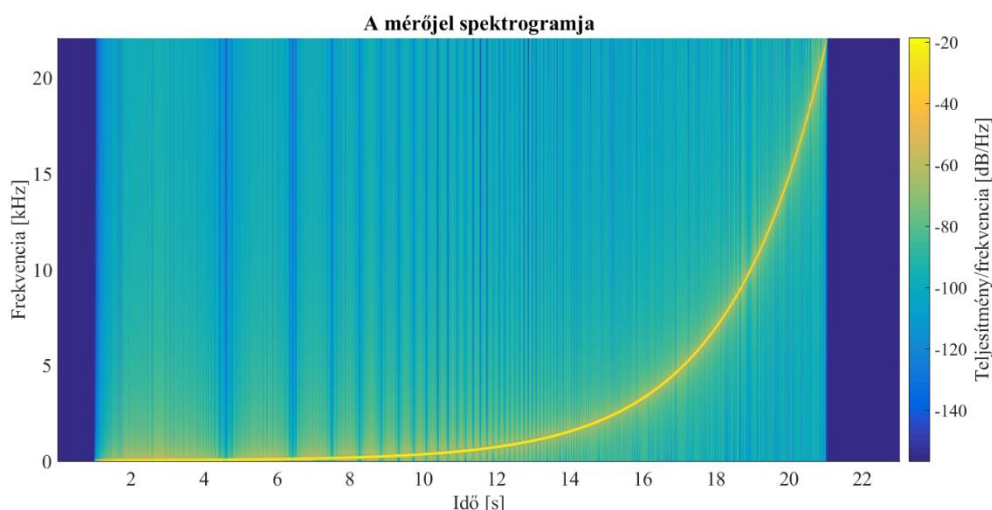
A képletekben megjelenő változók a következőket jelentik:

- $f_1$  – a jel kezdeti frekvenciája
- $f_2$  – a jel végső frekvenciája
- $\tilde{T}$  – a jel szándékolt időtartama
- $T$  – a szinkronizáció miatt megváltozott, valós időtartam
- $L$  – a frekvencianövekedés mértéke
- $x(t)$  – a létrejött mérőjel

Az erősítő átviteléhez használt logaritmikus sweep mérőjelet MATLAB segítségével állítottam elő. A jel időtartama kicsivel több, mint 20 másodperc, miközben a kezdő- és végfrekvenciái 11 Hz és 22 kHz voltak (44.1 kHz mintavételi frekvencia mellett). Egyetlen 20 szekundum hosszú mérés nagyjából megfelel 20 darab 1 másodperces mérésnek [30]. A relatíve hosszú mérőjel használata tehát annak átlagoló hatása miatt volt indokolt. Az így előállított sweep-en 1 másodperces felkeverést és 1/8 másodperces lekeverést alkalmaztam, hogy ezzel a jel amplitúdóspektrumának „szélein” jelentkező ingadozásokat kiküszöböljem. Ezek mellett a jel elé 1 szekundum hosszúságú „csendet” szúrtam be, hogy a mérés elindításakor gyakran jelentkező, hangkártyából származó „kattanás” később kivágásra kerülhessen. A válaszjelek lecsengésének időt hagyva a sweep mögé is illesztettem 2 másodpercnyi szünetet. Végül az így előállt mérőjelet 24 bites felbontású, 'wav' típusú fájlba exportáltam. A mérőjel amplitúdóspektrumát a 27. ábra, spektrogramját pedig a 28. ábra szemlélteti.



27. ábra: A mérőjel amplitúdóspektruma



28. ábra: A mérőjel spektrogramja

### 6.3.3 A mérések kiértékelése

A log. sweep mérőjelet mint gerjesztést egy rendszer bemenetére adva és közben a kimenetén megjelenő válaszjelet rögzítve lehetőségünk nyílik a rendszer átviteli jellemzőinek pontos mérésére. A gerjesztő- és válaszjelek dekonvolválására a legegyszerűbb módszer, ha a MATLAB *fft* függvényének segítségével a jeleken Fourier-transzformációt hajtunk végre, majd a két transzformált jelet elosztjuk egymással. Így kapjuk meg a rendszer  $H(z)$  átviteli függvényét (10):

$$H(z) = \frac{Y(z)}{X(z)}, \quad (10)$$

ahol  $Y(z)$  a válaszjel,  $X(z)$  pedig a gerjesztőjel diszkrét Fourier-transzformáltja,  $H(z)$  pedig az ezekből számított átviteli függvény.

A (10) képletben szereplő egyszerű dekonvolúciós eljárás használata jelen esetben azért előnytelen, mert a számított átviteli függvényben jelentős csúcsok keletkeznek azokon a tartományokon, ahol a bemenőjel amplitúdóspektruma kis értékeket tartalmaz (vagyis a nagyon kicsi és nagyon nagy frekvenciák tartományán). A probléma megoldása céljából az erősítőn végzett átvitméréseknél – és később minden olyan esetben, ahol átviteli függvények osztásáról van szó – a Tyihonov-féle regularizációs eljárást [32] alkalmaztam (11) és a dekonvolúciót az így létrehozott inverz szűrő segítségével végeztem (12). Az *inverz szűrő* gyakorlatilag a mérőjel spektrumának a frekvenciatengelyre vetített „tükörképét” adja, így a válaszjel Fourier-transzformáltját az inverz szűrő spektrumával megszorozva kaphatjuk meg a kívánt átviteli függvényt. A regularizációs eljárás használatának előnyeit a 30. ábra is mutatja.

$$X_{inverz}(z) = \frac{X^*(z)}{|X(z)|^2 + \lambda(z)}, \quad (11)$$

ahol  $X^*(z)$  a mérőjel spektrumának komplex konjugáltja,  $\lambda(z)$  a regularizációs paraméter,  $X_{inverz}(z)$  pedig az így kapott inverz szűrő spektruma. Esetünkben a  $\lambda(z)$  paraméter gyanánt egy olyan frekvenciatartományban definiált függvényt választottam, aminek segítségével a gerjesztőjel amplitúdóspektrumához tudtam viszonyítani a regularizáció szintjét. Az inverz szűrés alkalmazásával a (12) képletben leírt módon számíthatjuk az átviteli függvényt, amiből inverz Fourier-transzformációval az impulzusválasz is kinyerhető:

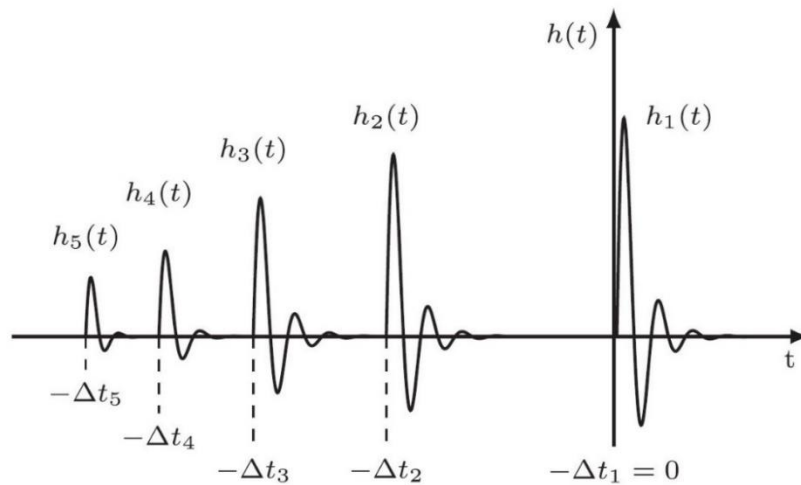
$$H(z) = X_{inverz}(z) \cdot Y(z), \quad (12)$$

ahol  $X_{inverz}(z)$  a (11) alapján számított inverz szűrő spektruma,  $Y(z)$  a válaszjel diszkrét Fourier-transzformáltja,  $H(z)$  pedig az ezekből számított átviteli függvény.

Az impulzusválaszból megkapható a rendszer felharmonikus átvitele is, ha az impulzusválaszon „negatív időben” megjelenő felharmonikus-válaszokat egyesével különválasztjuk a lineáris átvitelből adódó „pozitív idejű” impulzusválasztól. A 29. ábra az egyes harmonikusválaszok ( $h_1, h_2, h_3, h_4, h_5$ ) megjelenését mutatja a vizsgált rendszer impulzusválaszában. A 6.3.2. pontban említett *synchronized swept-sine* használata esetén a harmonikusválaszok közti időbeli eltolásokat az (13) képlet alapján kapjuk meg:

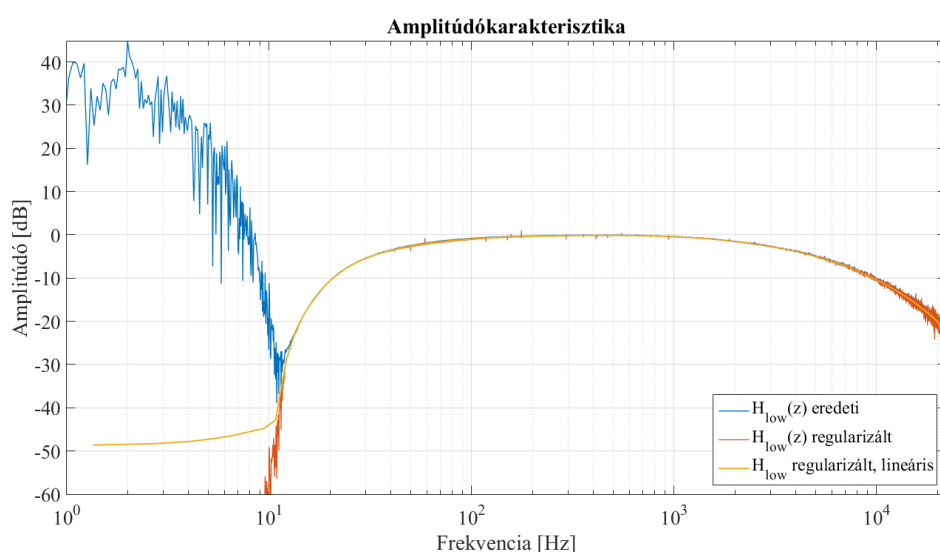
$$\Delta t_n = L \cdot \ln(n). \quad (13)$$

(Az említett virtuális „negatív idő” abból adódik, hogy a mérőjel  $\Delta t_n$  idővel való eltolása megfelel az  $n$ -edik harmonikus gerjesztésének. Ezek az eltolások jelennek meg az impulzusválaszban is.)



29. ábra: A harmonikusválaszok szétválasztása [31]

A 29. ábrán szemléltetett harmonikusválaszok közül  $h_1$  a lineáris átvitelhez tartozik, amíg  $h_2, h_3, h_4, h_5, \text{ stb.}$  a harmonikus torzítás miatt jönnek létre. Ezeket – például az impulzusválasz ablakozásával – szétválasztva, majd külön-külön Fourier-transzformálva lehetővé válik az átviteli függvény és a harmonikus torzítási jelleggörbék egyenkénti vizsgálata. Jelen esetben a torzítási jellemzők nem relevánsak számunkra, az viszont lényeges, hogy a felharmonikus-válaszok ne okozzanak hibát a lineáris átvitel mérésében. Ennek érdekében a mérések alapján számított impulzusválaszokból mindig kivágtam ezeket. (Más mérőjelek – például zajok – használata esetén ilyen szeparációra nincs lehetőség, így ott a nemlineáris hatások torzítják a lineáris átvitel mérését is.)



**30. ábra: A Tyihonov-féle regularizáció hatása kisjelű, lineáris átviteli függvény számításánál**

A 30. ábra szemlélteti a regularizáció útján végzett inverz szűrés (11, 12) előnyeit az egyszerű dekonvolúcióval (10) szemben. Azt láthatjuk, hogy a kisfrekvenciás tartományban a válaszjel és a gerjesztőjel Fourier-transzformáltjainak osztása esetén a kapott amplitúdómenet (kék görbe) erős kiemelést tartalmaz a kisfrekvenciás tartományban, ami lehetetlenné teszi az átviteli függvényből számított impulzusválasz direkt alkalmazását szűrési célokra. Ezzel szemben a regularizáció útján kapott inverz szűrő alkalmazásával az amplitúdókarakterisztikából (piros görbe) sikerült kiküszöbölnünk a kiemelést. Az ábrán szereplő sárga görbe az impulzusválasz csonkításával kapott lineáris átvitelhez tartozó amplitúdómenetet mutatja.

### 6.3.4 Az előerősítő átvitelének mérése

Az előerősítőn végzett átvitelmerések alkalmával legelőször leforrasztottam az erősítő bemenetére kapcsolt vezetékét, hogy az illesztőtranszformátor kimeneti impedanciája ne torzítsa a mérést. Ezt követően a külső hangkártya segítségével adtam a különböző jelszintekre beállított mérőjelet az erősítő bemenetére, miközben a hangkártya bemenetén keresztül a számítógépbe jutattam és rögzítettem az erősítő kimenőjelét. A felvételek elkészítéséhez az *Audacity* hangszerkesztő szoftvert használtam, majd pedig a felvételeket MATLAB-ba importáltam és a további feldolgozást itt végeztem el.

*Megjegyzés:* Az átvitelmerések során sajnos elkövettem egy komoly hibát: Általában véve egy olyan mérés esetén, ahol a gerjesztő- és a válaszjelet egyaránt felhasználjuk a kiértékelés folyamán, érdemes odafigyelni a két jel időbeli szinkronjára. Esetünkben a hangkártya késleltetése jelentette a problémát, amit elfelejtettem figyelembe venni. Ha a gerjesztőjelet az egyik, a válaszjelet pedig a másik bemeneti csatornán rögzítettem volna, akkor később nem kellett volna a késleltetés okozta időbeli eltolódás MATLAB-beli kompenzációjával foglalkozni. Végül az egymáshoz képest időben eltologatott jelek közti keresztkorrelációs csúcs keresése (*xcorr*) jelentette a megoldást, ami megadta, hogy hány mintával késett a válaszjel a gerjesztéshez képest.

## 6.4 Az erősítőmodell elkészítése

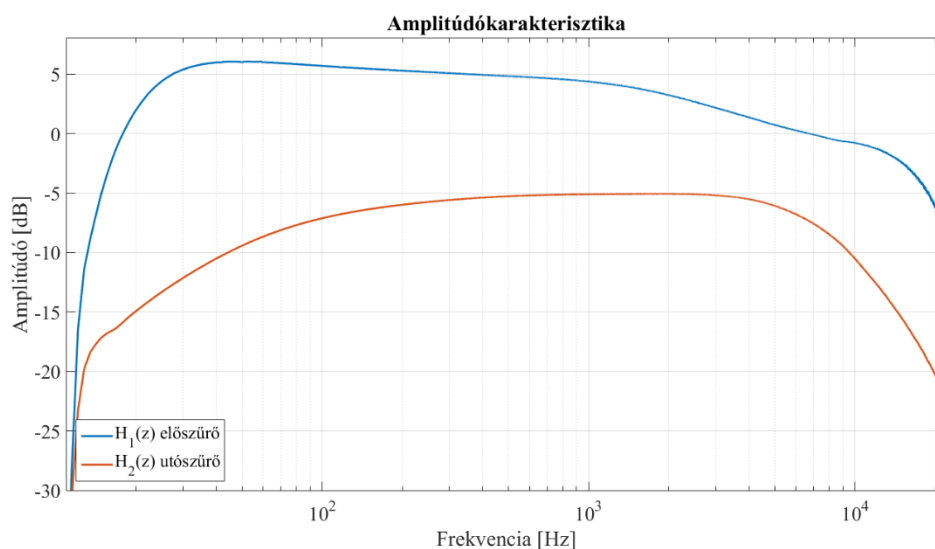
Az Hammond előerősítőjének kiterjesztett Wiener-Hammerstein modelljét MATLAB használatával készítettem el. A következő pontokban összefoglalom a modell összeállításának lépéseit és néhány ábrával szemléltetem az elért eredményeket.

### 6.4.1 Az előszűrő és az utószűrő elkészítése

Az 6.3. pontban ismertetett mérési eljárás alkalmazásával elvégeztem az erősítő átviteli függvényének mérését 7 különböző jelszintű gerjesztés mellett. A legnagyobb szintet az erősítő bemenetén – normál használat mellett – előforduló legnagyobb amplitúdójú jelhez igazítottam, majd a következő jelszintet mindig az előző felére csökkentettem. A 7. mérés eredménye olyan kismértékű harmonikus torzítást mutatott, hogy az ebből számított lineáris átvitelt feleltettem meg az 6.2.2. pontban  $H_{low}(z)$ -vel jelölt kisjelű átviteli függvénynek. Ehhez hasonló módon a  $H_{high}(z)$  nagyjelű átvitelt a 3. mérésből származó lineáris átvitel adta, mivel itt már megfelelően magas torzítás lépett fel. A kisjelű és nagyjelű lineáris átviteli függvényből (5) és (6) alapján előállítottam az

előszűrő  $H_1(z)$  és az utószűrő  $H_2(z)$  átviteli függvényét. A  $H_1(z)$  függvény számításakor a korábbiakhoz hasonlóan regularizációt alkalmaztam, hogy ne jöhessen létre nagy erősítés azokon a frekvenciákon, ahol  $H_{high}(z)$  kis értékeket tartalmaz.

Az elkészült előszűrő és utószűrő amplitúdókarakterisztikáit a 31. ábra szemlélteti. A görbék azt mutatják, hogy az erősítőmodell a kisfrekvenciás tartományban jelen lévő hangkomponenseket torzítja a legnagyobb mértékben, viszont a torzított jeleket a középtartományban emeli ki a legerősebben. Bár a görbék alapján a hallási tartományban a szűrők kiemelése – vagy más szempontból elnyomása – nem túl jelentős, de a torzító függvény „frekvenciafüggő munkapontját” ezek a csekély hatások is erősen befolyásolják.



31. ábra: Az előszűrő és az utószűrő amplitúdókarakterisztikái

## 6.4.2 A nemlineáris függvény paramétereinek hangolása

Az nemlineáris függvény paramétereinek hangolása az elkészült szűrők megvalósításával és a nemlineáris függvény implementációjával kezdődött. Mivel a paraméter-optimalizálás a modell iteratív működtetésével jár, ezért különös figyelmet kellett fordítanom arra, hogy a torzító függvény futása és a szűrők működése minél gyorsabb legyen. Ezen kívül meg kellett határoznom azt a költségfüggvényt (hibakritérium-függvényt), amelyet minimalizálni szeretnék a paraméterek hangolása során. Az egyszerűség kedvéért első körben a modell kimenete és az erősítő eredeti nagyjelű kimenete közötti, időtartománybeli négyzetes hibát (négyzetes különbséget) választottam hibafüggvénynek.

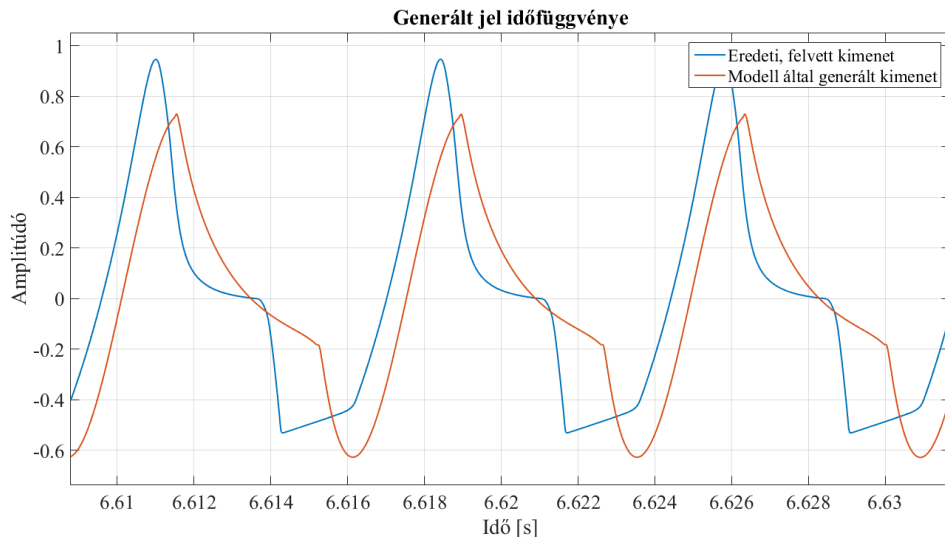
Mivel a szűrőket egyelőre FIR szűrők formájában valósítottam meg, vagyis az impulzusválaszuk jelentette egyben az együtthatók sorát is, ezért a szűrők működtetésére a MATLAB *fftfilt* nevű függvényét használtam. Ez a függvény frekvenciatartománybeli szűrést végez: a bemenőjelet átlapolt időablakok alatt Fourier-transzformálja, az egyes időszakaszok spektrumait megszorozza a szűrő átviteli függvényével, majd visszatranszformálja a jelet időtartományba. Az erőforrás-igényes időtartománybeli konvolúciót alkalmazó, hagyományos *filter* parancshoz képest az *fftfilt* futási ideje nagyjából ötödasharmada.

A modell hangoláskor az előszűrőt megvalósító *fftfilt* függvényt csak egyszer kellett futtatnom, mivel ez nem tartalmazott a hangolandó paramétert. Az előszűrőből kilépő jel jelentette a hangolandó modellt megvalósító függvény – vagyis az optimalizációs függvény – bemenőjelét. Ez a függvény tartalmazta a  $G_1$  erősítést, a  $k_p$ ,  $k_n$ ,  $g_p$  és  $g_n$  paraméterek által meghatározott nemlineáris torzító függvényt, a  $G_2$  erősítést és az utószűrőt. A függvény minden futtatás során előállította a modell kimenetét és az előerősítő eredeti kimenőjelének ismeretében kiszámította a költségfüggvény értékét minden mintára. Az optimalizációs függvényt a MATLAB *Optimization Toolbox* könyvtárában található *lsqnonlin* függvénynek adtam át, amely a 6 szabad paraméter értékét hangolta a legkisebb négyzetes hiba eléréséig. Azért esett a választásom az függvényre, mert ezt pontosan olyan nemlineáris optimalizálási problémákra tervezték, ahol nem számítható ki explicit módon a költségfüggvény egyes paraméterek szerinti parciális deriváltja. A paraméterek kezdőértékét modell leírásában [28] javasoltak szerint állítottam be, illetve később kipróbáltam más értékeket is.

Az optimalizációs folyamat eredményeit az 6.4.3. pontban részletezem.

### 6.4.3 Eredmények értékelése

A paraméterek optimalizálására alkalmazott *lsqnonlin* függvényt sok alkalommal futtattam. Annak ellenére, hogy a paraméterek kezdőértékeit, legkisebb és legnagyobb elérhető értékeit, valamint a költségfüggvény értékének elfogadási küszöbét több alkalommal igyekeztem megfelelően beállítani, az optimalizálás nem vezetett teljes sikerre. Ez abban nyilvánult meg, hogy az optimalizációs folyamat végén egyszer sem sikerült pontosan az erősítő eredeti kimenőjelének megfelelő jelet előállítanom a modellel. Az egyik legközelebbi eredményt a 32. ábra görbéi szemléltetik.



**32. ábra: A modellel eddig elért legjobb időfüggvény és az erősítő eredeti kimenőjele**

Az modellel elért eddigi eredmények – mint láthattuk – nem kielégítőek. Ennek oka egyfelől lehet a mért átviteli függvények pontatlansága, bár ezt nem tartom valószínűnek. Az is lehetséges, hogy a költségfüggvény választása nem volt megfelelő. Ez a hibaforrás sokkal valószínűbb, és az is bizonyos, hogy egy frekvenciatartományban előírt hibafüggvény minimalizálása sokkal inkább alkalmas lehetne a hallható különbségek enyhítésére, mint az eddig alkalmazott időtartománybeli görbeillesztés. A legrosszabb eshetőség, hogy ez a modell egyáltalán nem alkalmas az erősítő szimulálására. Ebben az esetben más modelltípusok között kell keresni a megfelelőt.

*Megjegyzés:* Mivel jelen dolgozat elkészültéig nem sikerült jobb eredményt elérni az időtartománybeli optimalizálással, a frekvenciatartománybeli kritériumfüggvény kidolgozására pedig már nem maradt idő, ezért a megismert nemlineáris modellt nem építettem bele a valós idejű alkalmazásba. Ehelyett a kisjelű lineáris átvitelnek megfelelő szűrő formájában valósítottam meg az előerősítő modelljét. Erről a szűrőről a következő, 6.5. alfejezetben lesz szó.

## 6.5 Párhuzamos IIR szűrőrendszer tervezése

A nemlineáris modell sikertelen identifikációja következtében kénytelen voltam egyszerűsíteni az előerősítő modelljén. Kézenfekvőnek tűnt, hogy az erősítőt a mért, kisjelű lineáris átvitelével (korábban  $H_{low}(z)$ -ként jelölve) szimuláljam valós időben. Mivel az eddigi MATLAB modellben használt, mérések nyomán kapott FIR szűrő fokszáma túl nagynak bizonyult a valós idejű használathoz, így egy erőforrás-hatékonyabb megoldást kellett találnom az eredeti  $H_{low}(z)$  átvitelű szűrő helyett.

A párhuzamosan kapcsolt, másodfokú IIR szűrőkből felépített struktúra [33] kiválóan alkalmas tetszőleges átviteli jelleggörbe modellezésére. Egy ilyen szűrőrendszer frekvenciafelbontása a megvalósítani kívánt átvitel szerint optimalizálható, valamint valós idejű használata sem jár nagy számítási igénnyel. A másodfokú tagok ráadásul könnyen stabilan tarthatók, így nem valószínű gerjedések előfordulása. Minden egyes szűrő a teljes frekvenciatartomány egy szűkebb sávjában végez érdemi módosítást, bár az egész átvitelbe „beleszól”, így érdekesebb a szűrők hatását együttesen vizsgálni. A párhuzamos szűrőrendszer átviteli függvényét a (14) képlet írja le:

$$H(z) = \sum_{n=1}^N \frac{b_{n,0} + b_{n,1}z^{-1}}{1 + a_{n,1}z^{-1} + a_{n,2}z^{-2}}, \quad (14)$$

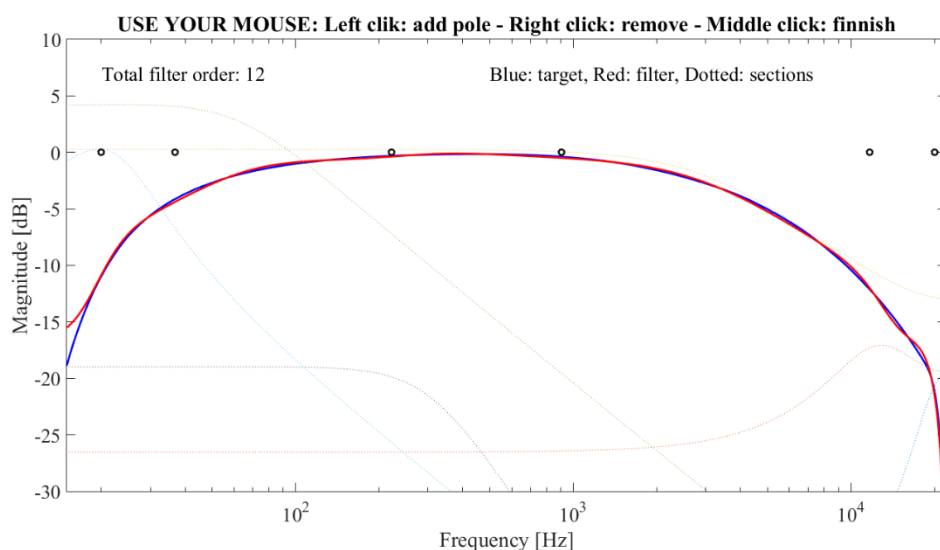
ahol  $N$  a másodfokú IIR szűrők száma,  $b_{n,0}$ ,  $b_{n,1}$ ,  $a_{n,1}$ ,  $a_{n,2}$  pedig az  $n$ -edik szűrő együtthatói.

A párhuzamos szűrőrendszer tervezéséhez a Bank Balázs honlapján [34] elérhető *parfiltdemo.m* MATLAB-ban írt függvényrendszert használtam némi módosítással. A függvény használata lehetővé teszi az egyes szűrők pólusainak manuális elhelyezését a modellezendő átviteli függvény mentén, miközben látható az így kialakult szűrőstruktúrával elért átvitel.

A *parfiltdemo.m* működése nagy vonalakban a következő: betáplálom a modellezendő szűrő impulzusválaszát, illetve beállítom a kezdeti póluspárok számát, amiket a függvény egyenletesen szétoszlat a hallási tartományban. Ha a grafikus felületen beviszek egy komplex konjugált póluspárt a rendszerbe – vagyis újabb ( $n$ -edik) másodfokú szűrővel bővítem azt –, akkor ebből a függvény kiszámolja  $a_{n,1}$  és  $a_{n,2}$  együtthatók értékét, így előáll az hozzáadott szűrő átviteli függvényének nevezője. Az ezekhez tartozó  $b_{n,0}$  és  $b_{n,1}$  értékeket az alkalmazás úgy számítja ki, hogy a módosult (bővült) szűrőrendszer hatására létrejövő átviteli függvény és a modellezendő átvitel között a legkisebb négyzetes hiba adódjon. A póluspárok száma és frekvenciája tetszőlegesen beállítható, ezért a MATLAB függvény jól használhatónak bizonyult.

A párhuzamos IIR szűrőstruktúra tervezésére alkalmas függvényt használtam az előerősítő kisjelű átviteli függvényének modellezésére. A manuális póluspár-elhelyezés során a 33. ábrán látható elrendezést kaptam. Az eredeti, modellezendő átvitelt az ábra kék görbéje, miközben a létrejött párhuzamos IIR szűrőrendszer frekvenciamenetét a

piros görbe szemlélteti. A MATLAB függvénnyel megtervezett elrendezésben 6 darab másodfokú IIR szűrő volt szükséges a kívánt hasonlóság eléréséhez.



33. ábra: A párhuzamos IIR szűrőrendszer tervezése a [34] honlapon található programmal

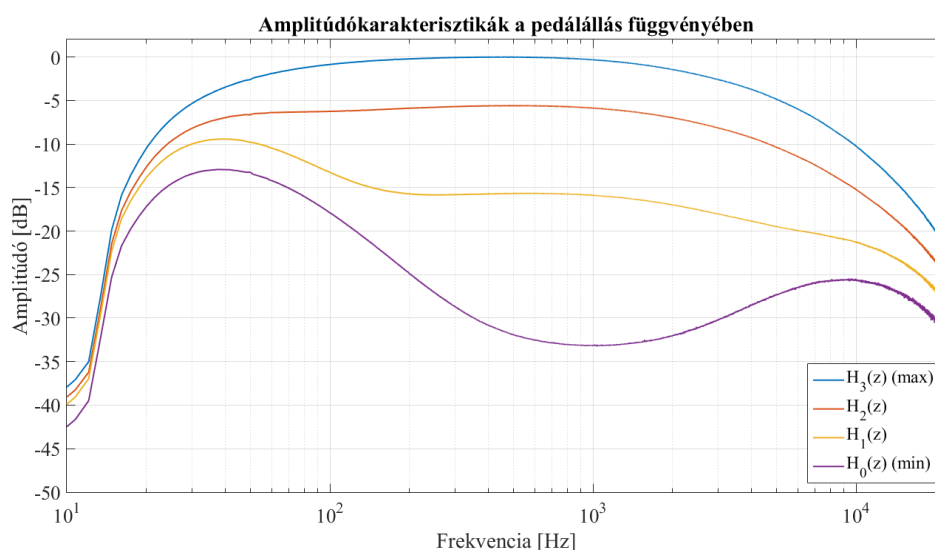
Az így kapott szűrőegyütthatók felhasználásával valósítottam meg a párhuzamos rendszert a valós idejű alkalmazásban (lásd: 7.2. pont), amihez 6 másodfokú IIR blokkra és egy összegző egységre volt szükség.

## 6.6 Az *expression control* pedál működése

A lábpedál által vezérelt szűrőáramkör az AO-28 típusú előerősítőbe van ágyazva, ezért az eszköz működését csak a teljes erősítőláncsal együtt tudtam vizsgálni. A méréseket megelőzően az előerősítő bemenetéről leforrasztottam az illesztőtranszformátor szekunder oldali kivezetését. Ennek a helyére csatlakozott a méréshez használt külső hangkártya kimenete, miközben az erősítő kimenete egy rezisztív feszültségosztón keresztül hangkártya bemenetére volt kapcsolva. A mérések során a megszokott logaritmikus sweep gerjesztőjelet használtam, amellyel lehetővé vált a lineáris átvitel harmonikus torzításmentes mérése. Az átviteli mérést négy különböző pedálállás esetében végeztem el: a pedál teljesen lenyomott (lehangosabb) és teljesen leszabályzott (leg halkabb) helyzetén kívül még két köztes pontban mértem meg az erősítő átviteli függvényét. (Az említett köztes beállítások nagyjából az 1/3-os és 2/3-os pedálállásokat jelentették a teljes skálát tekintve.)

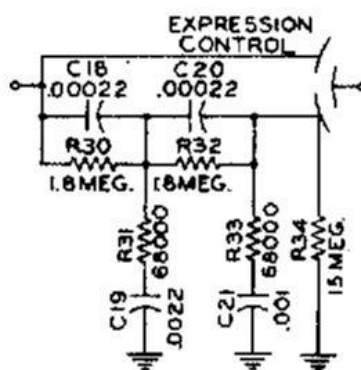
A 34. ábra mutatja a mérések eredményeként kapott amplitúdókarakterisztikákat. Az ábrán látható  $H_0(z)$ ,  $H_1(z)$ ,  $H_2(z)$  és  $H_3(z)$  jelölésekben szereplő alsó indexbeli szám

arra utal, hogy hány harmadnyira volt lenyomva a lábpedál. ( $H_3(z)$  a maximálisan lenyomott, lehangosabb,  $H_0(z)$  pedig a teljesen felengedett, leghalkabb állás.)



34. ábra: Az előerősítő amplitúdókarakterisztikája a pedál állásának függvényében

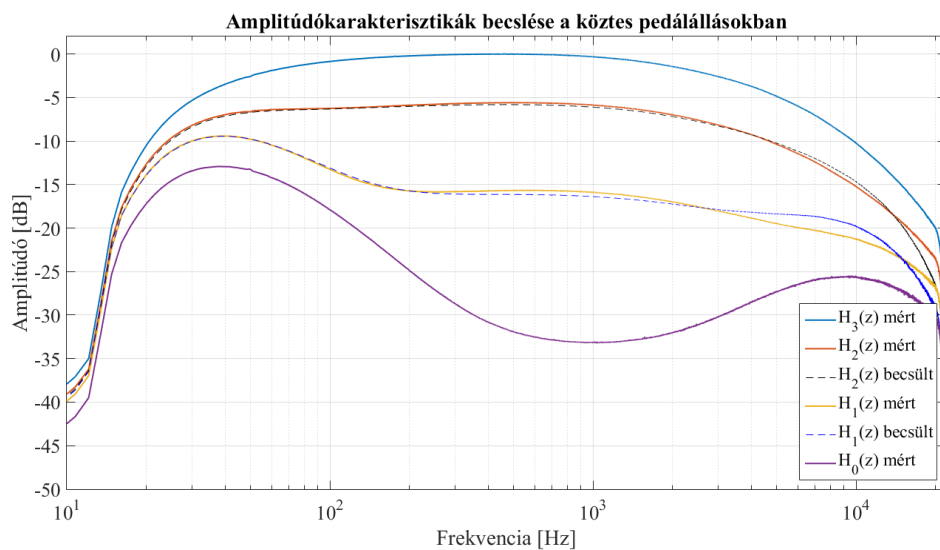
A pedál működése igen egyszerű: Az erősítő első két fokozatát követően kettéágazik a jelút. Az egyik ágban a jel érintetlen marad, a másik ágban pedig keresztülhalad a – 2.2.4. pontban már említett – középfrekvenciás szűrőn. A lábpedál kapacitív csatolást használva keveri össze a két ágból származó jelet, aminek eredményeként létrejön a folytonosan állítható mértékű szűrés. A jelút kettéválását, a középfrekvenciás szűrőáramkör kapcsolását, illetve a pedál kapacitív csatolási elvét a 35. ábra szemlélteti. (Az ellenállások értékei ohmban, a kapacitások értékei pedig mikrofardban értendők.)



35. ábra: A pedál szűrőáramkörének kapcsolása [16]

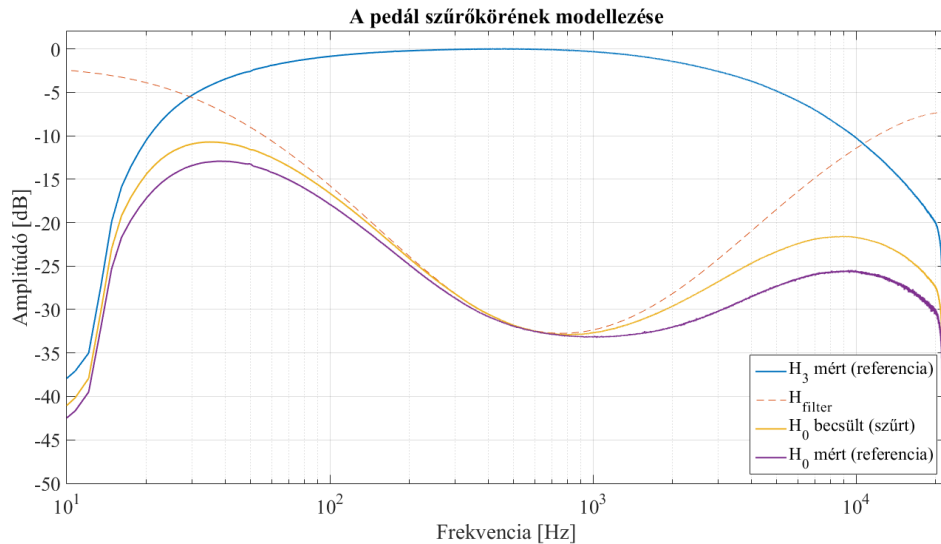
A lábpedál működésének szoftveres modellezése, illetve a későbbi valós idejű implementáció szempontjából felmerült a kérdés, hogy elég-e csupán a két szélső

helyzetben ismerni az átvitelt. Ennek eldöntése céljából megkíséreltem kikeverni a köztes állások átviteli függvényeit a szélső állásokéból. A folyamat eredményét a 36. ábra mutatja, amelyen a folytonos görbék továbbra is a mért karakterisztikának felelnek meg, amíg a szaggatott görbék a szélső állásokhoz tartozó átviteli függvények keveréseinek eredményei. A szaggatott görbék – főleg a kisfrekvenciás tartományban – igen jól közelítik a mért átviteleket. Érdekes tapasztalatot jelent, hogy a keverési arányok nem lineárisan függnek a pedál állásától.



**36. ábra: A köztes pedálállásokhoz tartozó amplitúdókarakterisztikák közelítése**

Az utolsó kísérletet az jelentette, hogy összehasonlítottam a pedál leszályzott állása mellett mért, vagyis leginkább szűrt jelet a középfrekvenciás szűrő elvi –, a kapcsolási rajz alapján levezetett – átviteli függvényéből konstruált digitális szűrővel szűrt jellel. Az elvi átviteli függvényt s-tartományban vezettem le, majd az így kapott folytonos idejű együtthatókat bilineáris transzformációnak [35] vettem alá. Az elkészült digitális IIR szűrő együtthatóival konvolváltam a teljesen lenyomott pedálállásnál mért válaszjelet. A 37. ábra az összehasonlítás eredményét szemlélteti. Az ábrán a lila görbe tartozik a mért karakterisztikához, a sárga pedig a kalkulált átvitel amplitúdómenetéhez. A két függvény közti eltérés adódhat az alkatrészek öregedés folytán bekövetkezett elhangolódásából. Az is lehetséges, hogy az érintetlen jel a pedál leginkább felengedett állásában is valamelyest a szűrt jelhez keveredik, vagyis a leszályzott pedál mellett végzett mérések sem önmagában a szűrőn áthaladó jelet adják vissza. Mindenesetre a legfeljebb kb. 4 dB-es eltérés nem olyan jelentős, ezért a levezetett átviteli függvényt használtam a továbbiakban.



**37. ábra: A pedál szűrőkörének modellezése**

Később, a valós idejű alkalmazás készítése során (7. fejezet) az *expression control* pedál középfrekvenciás szűrőkörét egy hetedfokú, „Direct-Form I” struktúrájú IIR szűrővel valósítottam meg. Ennek együtthatói megegyeztek a MATLAB-beli kísérletek során is használtakkal, amik az áramköri kapcsolás alapján számított folytonos idejű átviteli függvény bilineáris transzformáltjából származnak. (Gyakorlatilag egy az egyben a 37. ábra piros szaggatott görbéjének megfelelő szűrőt használtam.) Az így kapott szűrőt a párhuzamos IIR szűrőrendszer után kapcsoltam. A pedálállás és a kapacitív jelösszegző keverési aránya közti összefüggést a 36. ábrán szemléltetett kísérletek alapján tett becsléseim alapján kódoltam le. A szoftvermodell összegzője a párhuzamos szűrőrendszer „nyers” kimenőjeléből és a középfrekvenciás szűrőn áthaladó jelből keveri ki az alkalmazás végső kimenőjelét.

## 7 Valós idejű alkalmazás fejlesztése

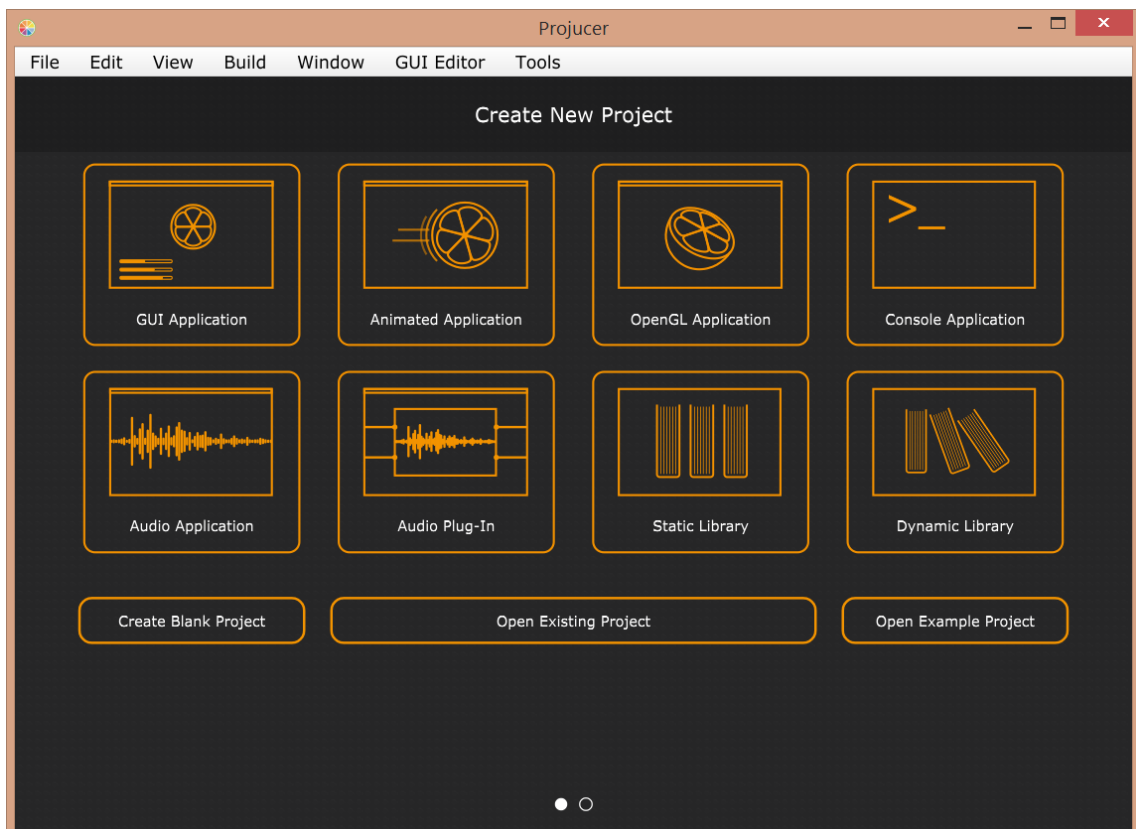
Általában egy hangszermodell fejlesztése önmagában is nagy volumenű projektnek számít, viszont a befektetett munka a modell alkalmazása esetén nyer valódi értelmet. Ennélfogva fontosnak tartottam a MATLAB-ban elkészített orgonamodell-részek integrálását és valós idejű környezetbe történő átültetését. A valós idejű alkalmazás elkészítése – azon túl, hogy kézzelfoghatóvá (füllel foghatóvá) teszi az eredményeket – katalizálja az algoritmusok erőforrás-hatékonyságát serkentő intézkedéseket.

### 7.1 A JUCE platform

A JUCE egy nyílt forráskódú keretrendszer, amely elsősorban zenei alkalmazások, valamint plug-in-ok fejlesztésének támogatására szolgál [36]. A JUCE által nyújtott támogatások az audio szoftverfejlesztés igényeinek igen széles spektrumát lefedik: A projektek forrásfájl-menedzsmentjétől kezdve, a saját függvénykönyvtárakon és példakódokon keresztül, egészen a grafikus interfészszerkesztőig minden megtalálható a JUCE rendszerben ahhoz, hogy akár egy gyakorlatlan fejlesztő is megtalálhassa benne a számítását. A fejlesztés nyelve a C++, ami a szoftver absztrakciós szintjének skálázhatósága szempontjából előnyös, hiszen a fejlesztő vegyítheti a szekvenciális és az objektumorientált szemléletet a programkód írása során. A keretrendszer nem rendelkezik ugyan saját fordítóprogrammal, viszont többféle integrált szoftverfejlesztői környezettel – mint a *Microsoft Visual Studio*, *Code::Blocks*, *Xcode*, *Android Studio*, stb. – képes együttműködni, így a JUCE-ban írt kód gyakorlatilag bármely elterjedt operációs rendszer felett fordítható. A keretrendszer saját függvénykönyvtárai – az ún. modulok – megoldást nyújtanak a leggyakrabban szükséges funkciókra, amelyek egy zenei alkalmazás esetén szóba kerülhetnek. Ilyenek például a hangkártya kezelését, MIDI eszközök illesztését, fájlkezelést, grafikus felület tervezését, plug-in-ok fejlesztését szolgáló osztályok, függvények. Egyes könyvtári elemek meglepően komplex rendszereket reprezentálnak, mint például a „Synthesiser” osztály.

A *Projucer* nevezetű alkalmazás a JUCE központi interfésze, amelyen keresztül a rendszer funkciói elérhetőek. A 38. ábra mutatja Projucer kezdőfelületét, amely a program indítását követően megjelenik. Ha a fejlesztő az itt látható projektípusok valamelyikét választja, akkor az új projekthez létrejön egy konfigurációs menü, illetve

néhány kiindulási forrásfájl. A menüben néhány projekttypus-specifikus beállítás mellett megadhatjuk, hogy a projekten belül mely forrásfájlokat és könyvtármodulokat szeretnénk fordítani, és melyik IDE környezetben. A kiindulási forrásfájlok tartalmazzák a működéshez szükséges minimális programkódot, ami nagyban megkönnyíti a fejlesztés korai szakaszát. Példaként egy „üres” zenei alkalmazás esetén a létrejövő két fájlban tárolt kód létrehoz egy üres ablakot és előkészíti a hangkártya kezelését végző függvényeket.



38. ábra: A JUCE Projucer felülete

*Megjegyzés:* A JUCE keretrendszer ingyenesen hozzáférhető, letölthető bárki számára, ám a használata webes regisztrációhoz kötött. Az azonosítás a Projucer legelső indításakor történik. Ekkor a program átirányít a böngészőbe, hogy ott a felhasználó bejelentkezzen a saját fiókjába. Amikor elkezdtem használni a JUCE-t (jelen dolgozat születése előtt kb. fél évvel), a legfrissebb verzió (JUCE 5.4.7) egy hibának köszönhetően nem tudott kommunikálni az *Internet Explorer*-rel, így a bejelentkezés sikertelen volt. A problémát egy régebbi verzió (JUCE 4.3.1) telepítésével orvosoltam, viszont a legfrissebb könyvtármodulokat használtam a fejlesztés során. Tudtommal az említett hiba mindmáig fennáll.

## 7.2 Az szoftver felépítése

A MATLAB-ban felépített orgonamodell-részek alapján – néhol azok jelentős egyszerűsítésével – elkészítettem a valós idejű Hammond alkalmazásomat. Mivel egy grafikus interfésszel ellátott hangfeldolgozó szoftverre volt szükségem, ezért a JUCE *Audio Application* projekttypusát választottam. A *HammondSimulatorApp* névre keresztelt projekt kezdetben két automatikusan generált forrásfájlt tartalmazott. Ezek közül a *Main.cpp* a *HammondSimulatorAppApplication* nevű objektum inicializálásáért felelős, és az applikáció – valamint a hozzá tartozó ablak – indításához, kezeléséhez és bezárásához szükséges komplett programkódot tartalmazza. Ezt a fájlt a későbbiekben sem módosítottam. A második fájl, a *MainComponent.cpp* az alkalmazás gerincét jelentő *MainContentComponent* osztály definícióját tartalmazza. Esetünkben a *MainContentComponent* az *AudioAppComponent* osztály leszármazottja, amely kifejezetten grafikus interfésszel ellátott audio alkalmazások alapjául szolgál. A *MainContentComponent* osztály hangkártyakezelő tagfüggvényei a *prepareToPlay*, a *getNextAudioBlock*, és a *releaseResources*. A *prepareToPlay* a kimenetek beállításait adja meg az alkalmazás indulásakor, a *getNextAudioBlock* függvény pedig rendszeresen meghívódik, amikor kiürül a hangkártya buffertára. A grafikus interfészhez való hozzáférést a *paint*, valamint a *resized* függvények biztosítják. Az említett függvények módosításával, felülírásával személyre szabhatjuk a zenei alkalmazást, így elérve a kívánt funkcionalitást.

A Hammond szimulátor alkalmazás vezérlését külső MIDI billentyűzet használatával képzeltem el. A MIDI üzenetek kezelésének problémájára a JUCE *HandlingMidiEventsTutorial* című példaprojektjében [37] találtam gyors és hatékony megoldásra, amit kevés módosítással átvettem. Ennek lényegi működése nagy vonalakban: Az *AudioDeviceManager* osztály függvényeivel felderítjük, hogy milyen külső MIDI eszközök vannak a számítógéphez csatlakoztatva, majd egy legördülő ablakot hozunk létre, hogy a felhasználó választhasson közülük. A *MidiInputCallback* osztály függvényeivel lekezeljük a kiválasztott bemenetről érkező MIDI üzeneteket és a billentyűzet állapotát egy *MidiKeyboardState* típusú objektumban tartjuk számon. Létrehozunk egy képernyőn megjelenő billentyűzetet is (*MidiKeyboardComponent* típusú objektum), amelyet szinkronizálunk a billentyűzet mindenkorai állapotával. Ennek köszönhetően a képernyőn is megjelenik, ha egy külső eszközön lenyomnak egy billentyűt. A példaprojekt tartalmaz még üzenetek dekódolására és kiírására alkalmas

függvényeket, ám ezeket csupán tesztelésre használtam a fejlesztés kezdetén. Mivel a billentyűk állapotát egy saját készítésű adatstruktúrában szerettem volna tárolni (*KeyState* osztály), ezért a *MidiKeyboardStateListener* osztály eredeti *handleNoteOn* és *handleNoteOff* függvényeibe építettem a saját *KeyState* osztályom *handle\_key\_on* és *handle\_key\_off* függvényeinek hívását. (Erre csupán azért volt szükség, mert az alkalmazás továbbfejlesztése során sokkal egyszerűbbé teszi majd a billentyűzet változásainak időbélyeghez kötött tárolását. Erre például a key click jelenség szoftvermodelljének megvalósítása esetén lehet szükség.)

A zenei alkalmazás alapvető forrásfájljai (*Main.cpp* és *MainComponent.cpp*) mellé létrehoztam a *hammond\_sound\_processing.h* és *hammond\_sound\_processing.cpp* fájlokat, amelyek az orgonamodell részegységeit megvalósító osztályok definícióit tartalmazzák. Ezek az osztályok a következők:

- ***SineGenerator***: A jelgenerátor egyszerűsített verziója, amely egy tiszta szinuszjel mintáinak előállítását végzi a standard C++ könyvtár *sin* függvényének felhasználásával. Ez az osztály csak a fejlesztés korai szakaszában kapott szerepet; az alkalmazás jelenlegi verziójában már a *WavetableSineGenerator* helyettesíti. Tagfüggvények: A *set\_sampling\_frequency*, *set\_amplitude*, *set\_frequency*, és *set\_phase* függvények – ahogy a nevükből is látszik – a mintavételi frekvenciának, a jel amplitúdójának, frekvenciájának, és kezdőfázisának beállítását teszik lehetővé. A *generate\_sample* függvény a soron következő mintát állítja elő.
- ***WavetableSineGenerator***: Szintén a jelgenerátor funkcióját tölti be, viszont a *SineGenerator*-hoz képest ez már a *wavetable szintézis* elvén működik. Ez azt jelenti, hogy a generátor egy tömbben tárolja az előállítandó jel egy periódusának mintáit. Mivel a tömb hossza erősen korlátozza a jel fázisbeli felbontását, így a pontosabb eredmény érdekében az aktuális hangminta értékét két szomszédos tömbelem közti *lineáris interpolációval* számítjuk ki. A tagfüggvények nevükben és funkciójukban megegyeznek a *SineGenerator*-ével, viszont itt megjelenik a *set\_wavetable* is, amely a hullámot tartalmazó tömb megadására szolgál.
- ***KeyState***: Egy billentyű állapotának („lenyomva” és „felengedve”) nyilvántartására szolgál. (Jelenleg csupán egy „intelligens” *bool* változóról van szó.) Tagfüggvények: A *handle\_key\_on*, illetve *handle\_key\_off* függvények

hívásával állíthatjuk be az aktuális állapotot, ami a *get\_key\_state*-tel kérdezhető le.

- ***HarmonicMixingUnitOutput***: Erre a struktúrára csupán azért volt szükség, hogy a keverőrendszert megvalósító *HarmonicMixingUnit* osztály feldolgozást végző függvénye (*mix\_samples*) egyszerre térhessen vissza az előerősítő és a percussion rendszer felé továbbítandó hangmintákkal.
- ***HarmonicMixingUnit***: A keverőrendszert valósítja meg. Egyetlen tagfüggvénye a *mix\_samples*, ami a generátorok által előállított minták összegzését végzi a billentyűk és drawbar-ok aktuális állapota alapján, a billentyűzet huzalozásának ismeretében, valamint a leválasztó ellenállások okozta súlyozási tényezők figyelembe vételével. A *mix\_samples* visszatérési típusa a *HarmonicMixingUnitOutput* struktúra, ami egyszerre tartalmazza az előerősítő és a percussion rendszer felé továbbítandó hangmintákat.
- ***PercussionGenerator***: A percussion funkciót valósítja meg annak minden beállításával, kompakt formában. Tagfüggvények: A *set\_sampling\_frequency* a mintavételi frekvencia beállítására szolgál. A *set\_percussion\_main*, *set\_percussion\_volume*, *set\_percussion\_decay*, és *set\_percussion\_harmonic* függvényekkel a percussion rendszerhez tartozó választókapcsolók állapotai adhatók meg (főkapcsoló, hangerő, lecsengési idő, és harmonikusválasztó). A *generate\_percussion\_sample* függvény a percussion rendszer aktuális kimeneti mintáját állítja elő. Ez utóbbi függvény egyszerre végzi az időzítést a billentyűzet állapota alapján, a változó időállandójú burkolójel számítását, valamint a kimenő minta létrehozását a burkoló és a megfelelő harmonikus felhasználásával.
- ***SecondOrder\_IIR\_Filter***: Az előerősítő lineáris átvitelét reprezentáló párhuzamos szűrőrendszer másodfokú IIR tagjait valósítja meg. Tagfüggvények: A *set\_filter\_coefficients* függvény a szűrőegyütthatók beállítására szolgál. A *filter* függvény végzi magát az IIR szűrést „Direct-Form I” struktúrában.
- ***Pedal\_IIR\_Filter***: Az *expression control* pedál középfrekvenciás tartományt elnyomó szűrőjét valósítja meg. Felépítése teljesen megegyezik a *SecondOrder\_IIR\_Filter*-ével azzal a különbséggel, hogy ez a szűrő hetedfokú és az együtthatói állandó értékűek.

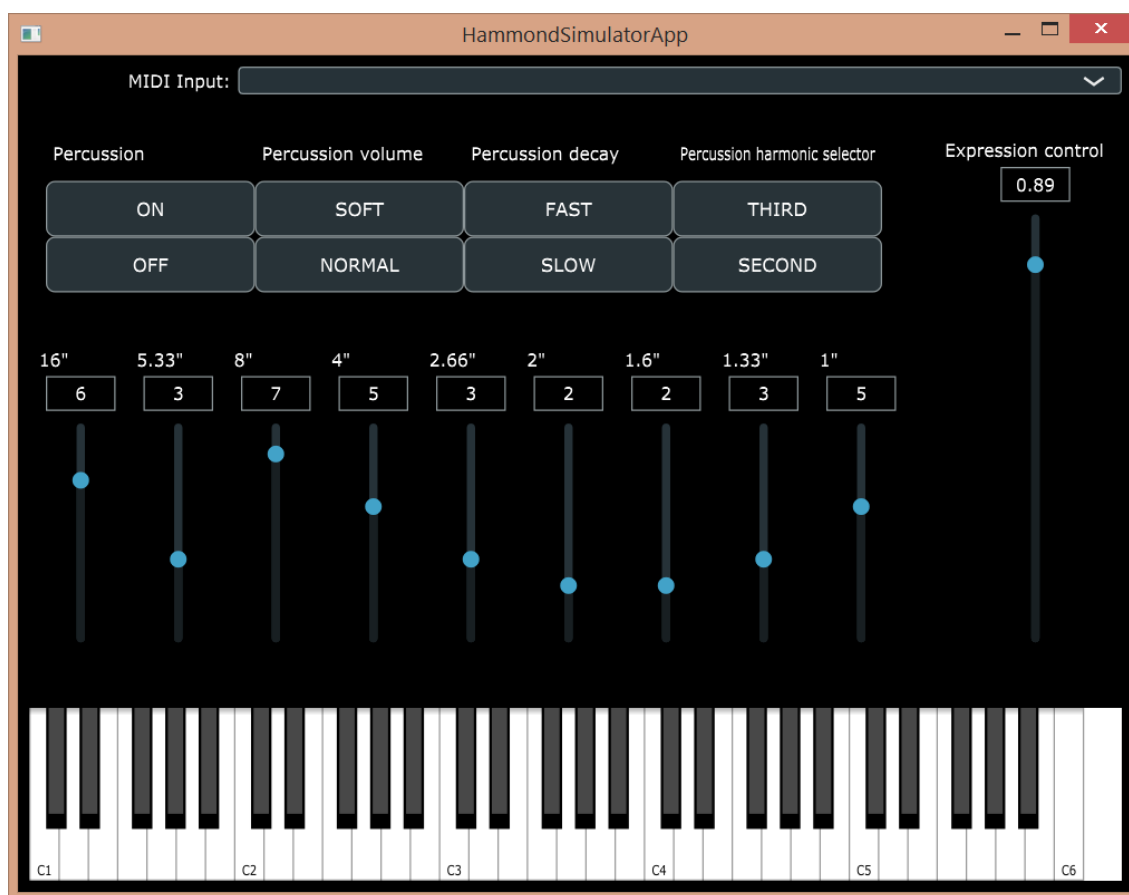
- **Preamplifier:** Az előerősítő szerepét látja el. Ebben az osztályban kerülnek példányosításra az erősítő lineáris átvitelét reprezentáló párhuzamos szűrőrendszer másodfokú IIR tagjai (*SecondOrder\_IIR\_Filter*), valamint az expression control pedál középfrekvenciás tartományt elnyomó szűrője (*Pedal\_IIR\_Filter*) is. Az egyetlen tagfüggvény az *amplify\_sample*, ami meghívja a szűrők *filter* függvényeit, majd a megfelelő súllyal összegzi azok kimenőjeleit.

A *hammond\_sinegenerator\_wavetables.h* nevű fájl a *WavetableSineGenerator* objektumok működéséhez szükséges hullámtáblákon kívül a generátorok legmélyebb szubhamonikusainak frekvenciáit tároló tömböt tartalmazza. Az említett frekvenciaértékek határozzák meg a hullámtáblák „olvasási sebességét” a működés során.

A *MainComponent.cpp* nevű forrásfájlban példányosítottam az osztályok listájában szereplő komponenseket, amik az orgona alkatrészeit valósítják meg. A kezdeti beállításokért felelős tagfüggvényeket a *prepareToPlay* függvény törzsében helyeztem el, így ezek mindig lefutnak az alkalmazás indításakor. A hangminták előállításáért és feldolgozásáért felelős, iteratív futtatást igénylő függvények a *getNextAudioBlock* törzsében kaptak helyet, tehát ezek a hangkártya bufferének töltése közben végzik a dolgukat.

Szintén a *MainComponent.cpp*-ben találhatók a grafikus interfészt megvalósító JUCE könyvtári függvények. Ezek közül a kezdeti beállításokért felelős példányok a *MainContentComponent* konstruktorában, a periodikus hívást igénylők pedig a *resized* függvényben kaptak helyet. Az elkészült alkalmazás felhasználói interfésze a 39. ábrán látható.

*Megjegyzés:* A dolgozat melléklete tartalmazza a szoftver teljes, „beszélő” változónevekkel és kommentekkel ellátott forráskódját, illetve az alkalmazás Win32 platformon futtatható verzióját is.



39. ábra: A valós idejű alkalmazás felhasználói interfésze

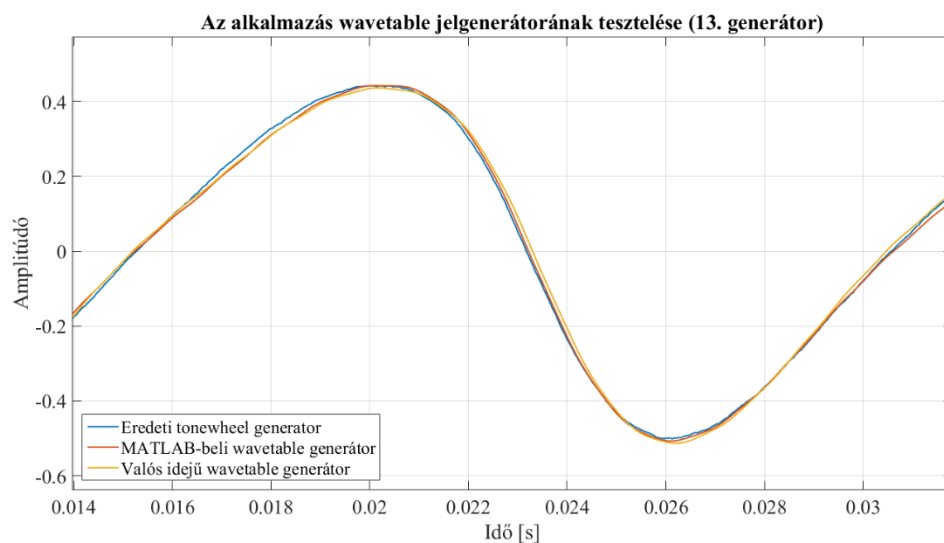
## 7.3 Az elkészült alkalmazás tesztelése, értékelése

Egy termék fejlesztésének – legyen szó akár hardverről vagy szoftverről, esetleg diplomatervről – általában a határidő eljövetele vagy a projekt tervezésekor felállított kritériumrendszer teljesítése szab határt. Esetünkben a végkritériumok szubjektív jellegűek, mivel a „hangzásbeli hasonlóság” összességében nem mérhető egzakt módon. Ennek következtében kénytelenek vagyunk beérni azzal, hogy a valós idejű alkalmazás részegységeinek mérhető jellemzőit összehasonlítjuk az eredeti hangszeren mért megfelelőikkel, valamint a MATLAB-beli modellekével.

*Megjegyzés:* A JUCE fájlírást és -olvasást célzó könyvtári függvényeit sokszori próbálkozást követően sem sikerült működésre bírni, ezért az alábbi teszteknel egy alternatív megoldáshoz nyúltam: a számítógép hangkimenetét felvételek formájában rögzítettem, majd ezeket importáltam a MATLAB-ba, hogy ellenőrizhessem a valós idejű orgonaszimulátor helyes működését.

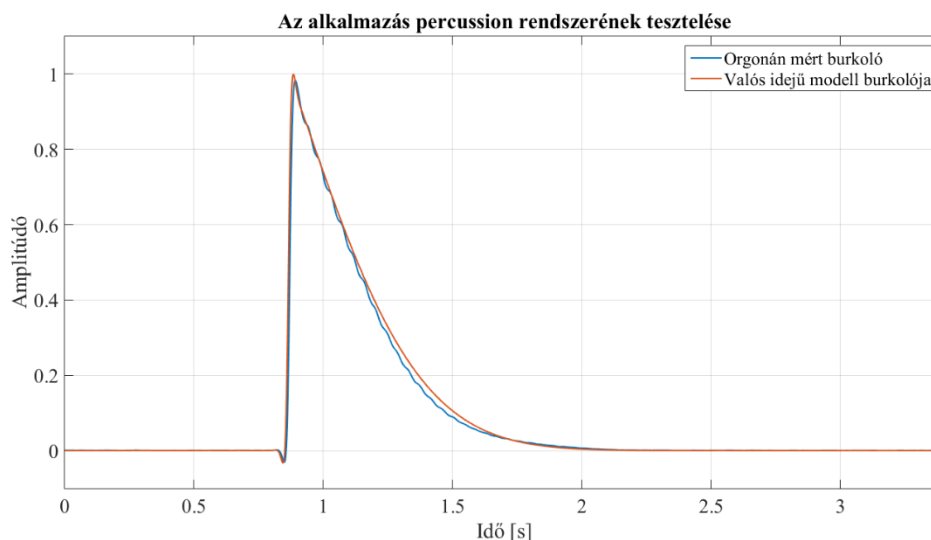
A tesztek során elsőként a wavetable jelgenerátorok megfelelőségét vizsgáltam. Ehhez úgy módosítottam az alkalmazás programkódját, hogy az előerősítő-egység

kimenőjele helyett egy-egy generátor kimeneti mintáit kapja meg a hangkártya buffere. Az így rögzített hangjelet beolvastam a MATLAB-ba, majd időbeli szinkronba hoztam az orgona tonewheel generátorának jeléről készült felvétellel, valamint az eredeti, MATLAB-ban létrehozott hullámtáblával. Az így kapott, 13-as generátorhoz tartozó időfüggvények hasonlósága a 40. ábrán látható. (A jelek Fourier-transzformációját követően összehasonlítottam a spektrumokat is, amik szintén egyezést mutattak a jelmodellben szereplő harmonikus és szubharmonikus komponenseket illetően.) Az első teszt eredménye az alkalmazás wavetable generátorának helyes működését bizonyítja.



**40. ábra: A valós idejű alkalmazás tesztelése – wavetable jelgenerátor**

Második teszt gyanánt a valós időben implementált percussion-modell működését vizsgáltam. Ehhez nem kellett módosítanom az alkalmazás rendes programkódját, csak a virtuális keyboard billentyűit leütni a drawbar-ok némított állapota mellett. A rögzített hangjel burkológörbáját az 5. fejezetben megismert módon állítottam elő MATLAB-ban, majd egy-egy lecsengés burkolóját összehasonlítottam az eredeti hangszeren mért megfelelő görbével. Ennek eredményét a 41. ábra mutatja egy-egy véletlenszerűen kiválasztott billentyűleütés esetén, amiket időben nagyjából szinkronba hoztam. A lecsengések időbeli lefolyásai között tapasztalt eltérés igen csekély, füllel nem érzékelhető. (Hasonlóan jó eredményeket kaptam más percussion-beállítások mellett is.) Elmondhatjuk, hogy a rendszer szoftveres modelljének működése elég jól közelíti a valódi orgonáét.



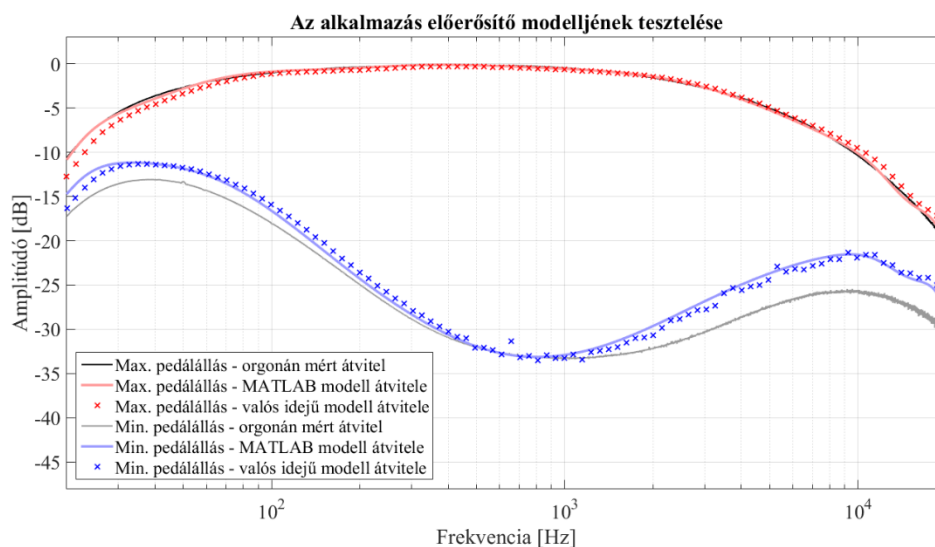
**41. ábra: A valós idejű alkalmazás tesztelése – percussion-modell**

Az utolsó tesztet az előerősítő modelljén végeztem. Mivel a modell mind az erősítő átviteli függvényét megvalósító párhuzamos szűrőrendszert, mind pedig az expression control pedálhoz tartozó középtartományi szűrőt tartalmazza, ezeket együttesen vizsgáltam. Ez a teszt igényelte a legösszetettebb szoftverbeli módosítást, mert a szűrők frekvenciamenetét csak azok szélessávú gerjesztésével lehetett mérni. Ezt nehezítette az adatok fájlból való olvasásának lehetetlensége, hiszen így nem tudtam a logaritmusos sweep mérőjelet használni. Végül a méréshez a tiszta szinuszelet előállító generátormodellt használtam, ami a saját készítésű C++ osztályok listáján *SineGenerator* néven szerepelt a 7.2. pontban.

Első lépésben a MATLAB *logspace* függvényével készítettem egy tömböt, ami 20 Hz és 20 kHz között 100 darab, exponenciálisan növekvő frekvenciaértéket tartalmazott. (Ez azért tűnt jó megoldásnak, mert – mint arról már szó volt – a fülünk érzékenysége logaritmusos frekvenciaskálán nézve egyenletes.) Az alkalmazásban létrehoztam 100 darab *SineGenerator* típusú objektumot, amelyek kimenőjeleit összegezve – diszkrét frekvenciájú komponensekből álló – szélessávú gerjesztőjelet kaptam. A szinuszgenerátorok frekvenciáit a MATLAB-ban generált 100 érték adta, kezdőfázisaikat pedig egyenletes eloszlású véletlenszám generátorokkal 0 és  $2\pi$  közé állítottam, hogy a létrehozott gerjesztőjelet csúcsstényezőjét minimalizáljam.

Az átvitelmérések során az előerősítő-modell bemeneti mintáit a keverőrendszer kimenete helyett az előbb ismertetett gerjesztőjelet adta. A gerjesztés mintáit a hangkártya bal csatornájára, az erősítő kimeneti mintáit pedig a jobb csatornára juttattam, majd a

jelek 1 perces regisztrátumait rögzítettem. A mérést megismételtem az expression control több állása mellett is. Ezt követően a MATLAB-ba importált jeleket frekvenciatartományban dekonvolváltam, és az így kapott átviteli függvényeket a gerjesztés diszkrét frekvenciáin ábrázoltam.



**42. ábra: A valós idejű alkalmazás tesztelése – előerősítő-modell**

A 42. ábrán a lábpedál teljesen lenyomott és teljesen felengedett állásához tartozó átviteli görbehármasok láthatók, mivel ezekben a helyzetekben volt kézenfekvő a valós idejű előerősítő-modellt összehasonlítani a MATLAB-beli megfelelőjével, valamint az eredeti hangszeren mért frekvenciamenetekkel. A piros és kék diszkrét pontok a valós idejű modell átvitelét rajzolják ki, a folytonos halványpiros és -kék görbék a MATLAB-beli szűrőket reprezentálják, a fekete és szürke görbék pedig a hangszeren végzett eredeti mérésekhez tartoznak. A maximális (teljesen lenyomott) pedálállás esetén a modellekben csak a párhuzamos szűrőrendszer van a jelútba kapcsolva. Ekkor csak kismértékű eltérés van az átviteli függvények között. Ezzel szemben a minimális pedálállásnál a szoftvermodellek viselkedése jelentősen eltér az valódi orgonáétól, viszont az itt tapasztalt hiba megfelel a 6.6. pontbeli eltéréseknek. (A valós időben alkalmazott hetedfokú szűrő megegyezik azzal, amit az áramköri komponensek értékei alapján MATLAB-ban készítettem. Ennek működése már korábban is eltért az orgonán mérttől.) Összességében az alkalmazás és a MATLAB modell átvitele közti eltérés sehol nem nagyobb néhány dB-nél, így a JUCE applikáció előerősítő-modellje megfelelőnek mondható.

## 8 Összefoglalás, kitekintés

Az alábbi fejezet 8.1. pontja a modellezési folyamat során elvégzett feladatokat foglalja össze, a 8.2. pont pedig a továbbfejlesztési irányokra vonatkozóan ad tájékoztatást.

### 8.1 Az elvégzett feladatok

Az előző félévek során a projektárgyak keretein belül a Hammond orgona szoftveres modellezésére tettem kísérletet. A munkám kezdetén főként interneten elérhető forrásokra támaszkodva tanulmányoztam az elektromechanikus Hammond orgonák felépítését, és az egyes alkatrészek működését. Ezt követően a hangszer részegységeinek MATLAB-beli modellezésével foglalkoztam, amihez a szükséges esetekben méréseket végeztem egy Hammond A-100 típusú orgonán.

Elsőként a *tonewheel generator*-t vettem alaposabban górcső alá. Ehhez hangfelvételeket készítettem a hangszer generátoregységének kimenőjeleiről, majd ezeket felhasználva egyrészt kiszámítottam a generátorok *belső ellenállását*, másrészt pedig elkészítettem a generátorok *jelmodelljét*: Szoftveres heterodin analízátor használatával, valamint spektrális vizsgálatok útján analizáltam a rögzített kimenőjeleket, majd ennek eredménye nyomán szétválasztottam a jeleket alkotó fontosabb komponenseket („*rendes*” *harmonikusok*, *szubharmonikusok*, *kHz-es harmonikusok*, „*periodikus nyikorgás*” *a szélessávú zajban*). Olyan algoritmust készítettem MATLAB-ban, aminek segítségével a generátorspecifikus komponensek paraméterei (frekvencia, amplitúdó, relatív fázishelyzet) automatikusan kinyerhetők a felvételekből. A jelalkotók immár ismert paramétereinek felhasználásával végzett jelszintézis eredményét összehasonlítottam az eredeti hangfelvétellel. Az iteratív optimalizálás eredményeként az eredeti és szintetizált jelek közti különbség elhanyagolhatóvá törpült. Végül a harmonikus és szubharmonikus jelalkotók alapján hullámtáblákat generáltam a később készített valós idejű alkalmazás számára.

A *keverőrendszer* szoftveres helyettesítőjét annak alkotói – vagyis a *billentyűzet*, a *drawbar-ok* és az *illesztőtranszformátor* – által megvalósított áramköri struktúra nyomán készítettem el. A billentyűsor huzalozási diagramja alapján, illetve az drawbar-transzformátor rendszer felépítésének ismeretében implementáltam a harmonikusok

keveréséért felelős *súlyozó-összegző algoritmust*. Ezen felül egyszerű, ám hatásos valószínűségi modellt készítettem a „*key click*” jelenségről, amit a valóságban a hangszer billentyűkontaktusainak pergése okoz.

A *percussion rendszer* kimenetén (többféle beállítás mellett) rögzített hangminták alapján modelleztem annak viselkedését: MATLAB-ban ábrázoltam a jelek burkolóit, majd megvizsgáltam azok időállandóit, amiből kiderült, hogy ezek – a várakozások szerinti konstansok helyett – időfüggőek. Ennek figyelembevételével elkészítettem azt a *szoftveres burkológenerátort*, ami az eredeti hangszerével megegyező lecsengésű burkolójeleket képes létrehozni.

Ezt követően az orgonában található *elektroncsöves előerősítő* lineáris és nemlineáris átviteli tulajdonságainak modellezésével foglalkoztam. Először kiválasztottam egy – az erősítő jeltorzító hatásának szimulálására alkalmasnak ígérkező – modell típust: a *kiterjesztett Wiener-Hammerstein modellt*. A modell felépítésének megismerését követően megkíséreltem alkalmazni azt az orgona erősítőjére, amihez – különböző amplitúdójú *logaritmikus sweep* jeleket használva – átvitméréseket végeztem az eszközön. Ezek alapján kiszámítottam a modellben szereplő elő- és utószűrő átviteli függvényét, majd pedig a modellben alkalmazott statikus nemlineáris függvény paramétereinek adaptációjára tettem kísérletet. Ennek sikertelensége okán az előerősítő korábbi nemlineáris modelljét egy egyszerű szűrőre cseréltem, ami annak lineáris átvitelét reprezentálja. A mérések alapján számított impulzusválaszt felhasználva *párhuzamos IIR szűrőrendszert* terveztem, ami erőforrás-hatékony megoldást jelentett a későbbi valós idejű implementáció során. Kiszámítottam még az – előerősítő részét képező – *expression control pedál* áramkörében alkalmazott szűrőkör folytonos idejű átviteli függvényét, majd ez alapján digitális szűrőt terveztem. Ennek működését összehasonlítottam a valódi orgonán mérttel.

A munka legvégső fázisát a *valós idejű alkalmazás* felépítése jelentette, amelyet a korábbi MATLAB modellek alapján konstruáltam. Az applikációt C++ nyelven írtam, a *JUCE fejlesztői keretrendszer* használatával. A JUCE megismerését követően elkészítettem a *HammondSimulatorApp* névre keresztelt zenei alkalmazást, ami külső MIDI billentyűzettel vagy a képernyőn megjelenő billentyűsorral vezérelhető. A szoftverbeli jelgenerátorok a *wavetable szintézis* elvén működnek, amíg a keverőrendszer- és percussion-modellek megegyeznek a MATLAB-beli verziókkal. Az előerősítő szoftvermoduljában *párhuzamos IIR szűrőrendszer* valósítja meg annak

átviteli függvényét, illetve egy *hetedfokú IIR szűrő* felelős az expression control pedál középfrekvenciás vágóhatásáért. Az alkalmazás moduljainak működése az elvégzett tesztek alapján elhanyagolható különbséggel közelíti a MATLAB modellekét.

## 8.2 Továbbfejlesztési lehetőségek

A Hammond tonewheel orgonák komplexitásából fakadóan még rengeteg lehetőség kínálkozik az általam felépített modellek finomítására vagy módosítására. Ezek közül néhányról már szó volt a dolgozatban.

Talán a legszembetűnőbb, hogy mennyit jelent a key click effektus, vagy a generátorok jelében tapasztalt „periodikus nyikorgó hang” jelenléte. Mivel az említett jelenségeket már valamelyest ismerem, ezért kézenfekvőnek tűnik ezek valós idejű alkalmazása, amihez viszont érdemes lehet további méréseket végezni. (Például a key click valószínűségi modelljéhez szeretnék valódi statisztikát készíteni a kontaktusok pergéséről. Megfontolandó a generátorok kimenőjelei közt fennálló fázisviszonyok vizsgálata is a „nyikorgás” szempontjából.) A 3.1. fejezetben már említettem a jelgenerátorok keverőrendszer általi terhelésének kérdéskörét. Szeretném a jövőben ezt is alaposabban megvizsgálni, ami elsősorban az illesztőtranszformátor impedanciájának mérését igényli.

Fontosnak tartom még az előerősítő nemlineáris modelljének feljavítását és valós idejű megvalósítását. A kiterjesztett Wiener-Hammerstein modell rengeteg lehetőséget rejt magában: elképzelhető, hogy a nemlineáris függvény módosításával már jobban közelíthetők a valódi erősítő kimenetén megjelenő időfüggvények is.

A fentiekén kívül szeretném a hangszerhez elterjedten használt *Leslie típusú forgótölcséres hangsugárzó* modelljét is a rendszerhez csatolni, ami nagyban befolyásolja a létrehozott hanghatást. Ez például abban mutatkozik meg, hogy az előerősítő kimenetén rögzített hangminták visszahallgatása és a hangsugárzó élő hangja között szembetűnő a különbség. Általában a Leslie-kről az emberek csak a pörgő tölcsérekre asszociálnak, pedig maguk a hangszórók is nyilvánvalóan fontos szerepet töltenek be: hallhatóan finomítják, tompítják a nagyfrekvenciás tartomány jelenlétét, ami a key click okozta meredek időfüggvényű, ugrásszerű jelek „elkenéséért” is felelős lehet.

# Irodalomjegyzék

- [1] Wikipedia (2020). *Hammond organ*.  
[https://en.wikipedia.org/wiki/Hammond\\_organ](https://en.wikipedia.org/wiki/Hammond_organ)  
(Utolsó megtekintés: 2020.05.26.)
- [2] Wikipedia (2020). *List of Hammond organs*.  
[https://en.wikipedia.org/wiki/List\\_of\\_Hammond\\_organ](https://en.wikipedia.org/wiki/List_of_Hammond_organ)  
(Utolsó megtekintés: 2020.05.26.)
- [3] Wikipedia (2020). *Clonewheel organ*.  
[https://en.wikipedia.org/wiki/Clonewheel\\_organ](https://en.wikipedia.org/wiki/Clonewheel_organ)  
(Utolsó megtekintés: 2020.05.26.)
- [4] Clavia DMI (2019). *Nord C2D*.  
<https://www.nordkeyboards.com/products/nord-c2d>  
(Utolsó megtekintés: 2020.05.26.)
- [5] Hammond USA (2020). *Sk2*.  
<http://hammondorganco.com/products/portable-organs/sk2/>  
(Utolsó megtekintés: 2020.05.26.)
- [6] Viscount (2020). *Legend*.  
<https://www.viscountinstruments.com/legend.html>  
(Utolsó megtekintés: 2020.05.26.)
- [7] Ferrofish Germany (2020). *B4000+*.  
<https://www.ferrofish.com/product/b4000-virtual-organ/>  
(Utolsó megtekintés: 2020.05.26.)
- [8] acousticsamples (2020). *B-5 Organ V2*.  
<https://www.acousticsamples.net/B5>  
(Utolsó megtekintés: 2020.05.26.)
- [9] Arturia (2020). *B-3 V Overview*.  
<https://www.arturia.com/products/analog-classics/b-3-v/overview>  
(Utolsó megtekintés: 2020.05.26.)
- [10] IK Multimedia (2020). *Hammond B-3X*.  
<https://www.ikmultimedia.com/products/hammondb3x/>  
(Utolsó megtekintés: 2020.05.26.)
- [11] Electric Druid (2019). *Technical aspects of the Hammond Organ*.  
<https://electricdruid.net/technical-aspects-of-the-hammond-organ/#drawbars>  
(Utolsó megtekintés: 2020.05.26.)
- [12] North Suburban Hammond Organ Society (2020). *The Hammond Organ*.  
<http://www.nshos.com/HAMMOND1.htm>  
(Utolsó megtekintés: 2020.05.26.)

- [13] theatreorgans.com (2020). *Theory of operation; Series A-100*.  
<http://theatreorgans.com/hammond/faq/a-100/a-100.html>  
(Utolsó megtekintés: 2020.05.26.)
- [14] Hammond Instrument Company (1987). *Service manual H000-000495*.  
<https://archive.org/details/HammondOrganServiceManual-EarlyModelsabcSeriesB3A-100RtDG>  
(Utolsó megtekintés: 2020.05.26.)
- [15] Dairiki, Geoffrey T. (2002). *Percussion*.  
<http://www.dairiki.org/HammondWiki/Percussion>  
(Utolsó megtekintés: 2020.05.26.)
- [16] Benton Electronics (2020). *Servicing the Hammond B-3 Type Pre-Amp*.  
<https://bentonelectronics.com/servicing-the-hammond-b-3-type-pre-amp/>  
(Utolsó megtekintés: 2020.05.26.)
- [17] Serra, Xavier (1997). *Musical Sound Modeling with Sinusoids plus Noise*.  
In: Musical Signal Processing. Szerk.: Roads, Pope, Piccilli, De Poli.  
ISBN 9026514832. Downington, Swets & Zeitlinger. 1997. 91-122. oldal.
- [18] Dairiki, Geoffrey T. (2002). *Manual Tapering*.  
<http://www.dairiki.org/HammondWiki/ManualTapering>  
(Utolsó megtekintés: 2020.05.26.)
- [19] Dairiki, Geoffrey T. (2018). *Key Click*.  
<http://www.dairiki.org/HammondWiki/KeyClick>  
(Utolsó megtekintés: 2020.05.26.)
- [20] Mercier, Marc A. (2001). *The Hammond-Leslie FAQ*.  
[http://theatreorgans.com/hammond/faq/hammond-faq\\_toc.html](http://theatreorgans.com/hammond/faq/hammond-faq_toc.html)  
(Utolsó megtekintés: 2020.05.26.)
- [21] Wiltshire, Tom. (2008). *Technical aspects of the hammond organ*.  
<https://electricdruid.net/technical-aspects-of-the-hammond-organ/>  
(Utolsó megtekintés: 2020.05.26.)
- [22] Moro, Giulio; McPherson, Andrew P.; Sandler, Mark B. (2017). *Dynamic temporal behaviour of the keyboard action on the Hammond organ and its perceptual significance*. The Journal of the Acoustical Society of America, 142. évfolyam, 5. szám, 2808-2822. oldal.
- [23] Moro, Giulio. (2016). *The effect of touch on the Hammond organ*.  
<https://www.youtube.com/watch?v=kcRtyvQxbgU&feature=youtu.be>  
(Utolsó megtekintés: 2020.05.26.)
- [24] Czupy, Gergely János (2017). *Nemlineáris rendszerek identifikációja gyakorlati megközelítésben*. Szakdolgozat. Budapesti Műszaki és Gazdaságtudományi Egyetem.

- [25] Pakarinen, Jyri; Yeh, David T. (2009). *A Review of Digital Techniques for Modeling Vacuum-Tube Guitar Amplifiers*. Computer Music Journal, 33. évfolyam, 2. szám, 85-100. oldal.
- [26] Yeh, David T. (2011). *Automated Physical Modeling of Nonlinear Audio Circuits For Real-Time Audio Effects - Part II: BJT and Vacuum Tube Examples*. Institute of Electrical and Electronics Engineers, Transactions on Audio, Speech, and Language Processing, 20. évfolyam, 4. szám, 1207-1216. oldal.
- [27] Reed, M. J.; Hawksford, Malcolm J. (1996). *Practical Modeling of Nonlinear Audio Systems Using the Volterra Series*. In: Proc. Audio Engineering Society 100th Convention, May 1996, Copenhagen, Denmark, Convention Paper No. 4264.
- [28] Eichas, Felix; Zölzer, Udo (2018). *Gray-Box Modeling of Guitar Amplifiers*. Journal of the Audio Engineering Society, 66. évfolyam, 12. szám, 1006-1015. oldal.
- [29] Guy-Bart, Stan; Jean-Jacques, Embrechts; Dominique, Archambeau (2002). *Comparison of different impulse response measurement techniques*. Journal of the Audio Engineering Society, 50. évfolyam, 4. szám, 249-262. oldal.
- [30] Farina, Angelo (2007). *Advancements in impulse response measurements by sine sweeps*. In: Proc. Audio Engineering Society 122nd Convention, May 2007, Vienna, Austria, Convention Paper No. 7121.
- [31] Novak, Antonin; Lotton, Pierrick; Simon, Laurent (2015). *Synchronized Swept-Sine: Theory, Application, and Implementation*. Journal of the Audio Engineering Society, 63. évfolyam, 10. szám, 786-798. oldal.
- [32] Dabóczi, Tamás (2017). *Új jelút-kompenzációs eljárások*. MTA doktori értekezés. Budapesti Műszaki és Gazdaságtudományi Egyetem, Méréstechnika és Információs Rendszerek Tanszék.
- [33] Bank, Balázs (2011). *Logarithmic Frequency Scale Parallel Filter Design with Complex and Magnitude-Only Specifications*. Institute of Electrical and Electronics Engineers, Signal Processing Letters, 18. évfolyam, 2. szám, 138-141. oldal.
- [34] Bank, Balázs (2018). *Transfer function modeling and equalization by fixed-pole parallel filters*.  
<http://home.mit.bme.hu/~bank/parfilt/>  
 (Utolsó megtekintés: 2020.05.26.)
- [35] Parks, Thomas W.; Burrus, C. Sidney (1987). *Digital Filter Design*. New York, John Wiley & Sons. ISBN 978-0471828969.
- [36] JUCE (2019). *JUCE*.  
<https://juce.com/>  
 (Utolsó megtekintés: 2020.05.26.)

- [37] JUCE (2019). *Tutorial: Handling MIDI events*.  
[https://docs.juce.com/master/tutorial\\_handling\\_midi\\_events.html](https://docs.juce.com/master/tutorial_handling_midi_events.html)  
(Utolsó megtekintés: 2020.05.26.)

## Köszönetnyilvánítás

Először is szeretnék óriási köszönetet mondani a konzulensemnek, Bank Baláznak az elmúlt négy év során nyújtott minden szakmai és szakmán kívüli segítségéért. Kívánok neki felhőtlen zenélést és stabil IIR szűrőket!

Szeretném nagyon megköszönni Márkus Jánosnak, hogy kitűnő állapotú veterán hangszerét azonnal, kérdés nélkül a rendelkezésemre bocsátotta, és hogy a mérések alkalmával szívesen fogadott az otthonában.

Rendkívüli köszönettel tartozom a Szüleimnek odaadó támogatásukért, ösztönzésükért, valamint azért, hogy minden helyzetben kiálltak mellettem!

Köszönöm Kazella Ádámnak, hogy egyetemi éveink alatt is sűrűn randomkodott velem! Köszönöm Bodó Baláznak, hogy sok-sok mosolyt vitt a mesterképzésem éveibe és elviselte a hajnali ténykedéseimet.