



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

Gyurász Gábor Tamás

CSÖVES TORZÍTÓPEDÁL VALÓSIDEJŰ SZIMULÁCIÓJA

KONZULENS

dr. Bank Balázs

BUDAPEST, 2015

Tartalomjegyzék

Összefoglaló	1
Abstract.....	2
1 Bevezetés	3
2 A modellezendő áramkör: B. K. Butler Tube Driver	5
3 Elméleti összefoglaló	7
3.1 Nemlineáris rendszer modellezése.....	7
3.2 Diszkretizálási módszerek	8
3.2.1 Előrelépő Euler módszer.....	8
3.2.2 Hátralépő Euler módszer	10
3.2.3 Impulzusinvariáns transzformáció.....	11
3.2.4 Bilineáris transzformáció, trapéz módszer.....	12
3.2.5 Runge-Kutta módszerek	13
3.3 Gyökkereső algoritmusok	14
3.3.1 Newton-Raphson módszer.....	14
3.3.2 Szekáns módszer.....	15
3.4 Lineáris egyenletrendszert megoldó algoritmusok	16
3.4.1 Egyenletrendszer megoldása Inverz számításával.....	16
3.4.2 Egyenletrendszer megoldása LU felbontással	17
4 A folytonos idejű modell létrehozása	18
4.1 Az elektroncső modell	19
4.2 Lineáris áramköri blokkok.....	23
4.2.1 Az előerősítő	23
4.2.2 A parametrikus hangszínszabályzó.....	24
4.3 Nemlineáris áramköri blokkok	28
4.3.1 A túlvezérelt műveleti erősítő és környezete.....	29
4.3.2 A csöveket tartalmazó részáramkör	32
5 Diszkrét idejű modell.....	35
5.1 A túlvezérelt műveleti erősítő és környezete	36
5.2 Hangszínszabályzó.....	37
5.3 Cső körüli részáramkör	41
5.3.1 Explicit módszerek	41

5.3.2 Implicit módszer	46
6 A Matlab modell megvalósítása C++-ban VST pluginként	54
6.1 Mi az a VST	54
6.2 A Tube Driver VST plugin felépítése	54
6.2.1 Jelfeldolgozás szempontjából fontos függvények	54
6.3 Matlab kód portolása C++ kódra	55
6.4 Fejlesztő eszközök	55
6.5 Tesztelésnél használt eszközök	56
6.6 Hatékonyság mérése	58
6.7 Hatékonyság javítása	58
7 Az áramkör megépítése, mérése	60
7.1 Az elektroncső karakterisztikájának mérése	60
7.1.1 A puffer erősítők ellenállásainak megválasztása	62
7.1.2 Az árammérő ellenállások megválasztása	63
7.1.3 A mérési eredmény feldolgozása	63
7.2 A megépített Tube Driver áramkör kimeneti jelének mérése	70
8 A végleges modell kiválasztása	72
9 Tesztek eredményei.....	73
9.1 Matlabban szimulált modell összehasonlítása a Spice modellel	75
9.2 A VST plugin összehasonlítása a Matlab modellel	79
9.3 A VST plugin összehasonlítása a valódi áramkörrel	83
10 Összefoglalás.....	86
Irodalomjegyzék.....	88

HALLGATÓI NYILATKOZAT

Alulírott **Gyurász Gábor Tamás**, szigorló hallgató kijelentem, hogy ezt a szakdolgozatot/ 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, 2015. 05. 23.

.....
Gyurász Gábor Tamás

Összefoglaló

A dolgozat a B.K. Butler Tube Driver torzítópedál áramkör valós időben futtatható modelljének létrehozásáról, C++ nyelven való implementálásáról és működésének valós hardverrel történő ellenőrzéséről szól. A dolgozat az egyes lépéseknél felmerülő problémákat, és a lehetséges megoldásokat is ismerteti. A fontosabb módszerekhez tartozó elméleti alapok az elméleti bevezetőben találhatóak, mely a diszkretizálási eljárásokról, gyökkereső algoritmusokról, lineáris egyenletrendszerek megoldásáról és a munka során felhasznált nemlineáris rendszer leírási módjáról szól.

Az analóg áramkör vizsgálatához az *LTSpice* áramkörszimulációs programot használtam. A modellt *Matlab* környezetben fejlesztettem. A C++ nyelven implementált modellt VST pluginként valósítottam meg, mely valós időben tesztelhető egy tetszőleges VST host alkalmazással. Az áramkört egyszerű prototípusok építésére szolgáló *breadboardon* építettem meg.

A dolgozat bemutatja az áramköri kapcsoláson és az áramköri elemek modelljein alapuló folytonos idejű modell megalkotásának menetét, a kiválasztott elektroncső modell leírását és a folytonos idejű modell négyféle diszkretizációs eljárással létrehozott, diszkrét idejű változatainak egyenleteit és származtatásainak részleteit. A diszkrét modellek közül kiválasztottam azt, mely az erőforrás igény, stabilitás és hangzásbeli hűség szempontok alapján a legoptimálisabbnak bizonyult. A VST plugin erre a diszkrét modellre épül. A dolgozat betekintést nyújt a VST pluginok hatékony C++ nyelvű megvalósításának alapvető irányelveibe és a hardverrel való összehasonlításhoz szükséges elektroncső karakterisztika mérés elméleti és gyakorlati megvalósításába.

A kiválasztott *Matlab* algoritmus és VST bővítmény program minősítése olyan szimulációkon és méréseken alapul, mely során tipikusan előforduló, valós gitárhang által gerjesztett kimeneti értékeket vizsgálok. Az összehasonlítás során az *LTSpice* programban és *Matlab* környezetben futtatott szimulációk eredményeit, a VST plugin eltárolt kimeneti jelét és a megépített áramkőron mért kimeneti jelet hasonlítottam össze. Az *LTSpice*, *Matlab* szimulációk és VST plugin eredményei között minimális eltérés mutatkozik, a hardveren végzett mérés eredménye viszont eltér.

Abstract

This thesis deals with the process of building a real-time model of the B. K. Butler Tube Driver distortion pedal, implementing the algorithm in C++ and comparing the resulting algorithm and program against the real hardware. The thesis describes the problems at each step and gives the most viable solutions. The literature for the most relevant methods in the field of system discretization, root finding algorithms, solving of linear equation systems and the method of describing this particular nonlinear network can also be found in the first chapters.

For the analysis of the analog circuit *LTSpice* has been used. The model has been developed in *Matlab*. The C++ implementation of the model is a VST plugin, which can be tested in real-time with a VST host program. The analog circuit was built on a breadboard panel

The thesis describes how the continuous time model was created based on the circuit schematic and the models of the electronic components. It introduces the chosen vacuum tube model and presents the details behind four different methods used to discretize the continuous time model. I have chosen the optimal discrete time model in terms of resource requirements, stability and sound fidelity. The VST plugin implements this chosen forward Euler model. In addition, the thesis gives guidelines in the topic of efficient C++ based VST plugin implementation and vacuum tube characteristic measurement in theory and in practice.

The fidelity of the chosen Matlab algorithm and VST plugin was evaluated based on simulations and measurements, where the input signal was a typical guitar signal. Comparison was made between the output of the *LTSpice* simulation, the final *Matlab* algorithm, the VST plugin and the real hardware. The *LTSpice*, *Matlab* simulations and VST plugin measurements show little difference in the results. The results of the hardware measurements differ.

1 Bevezetés

A torzító effekt az 1950-es évekbeli megjelenése óta az egyik legmeghatározóbb és legkarakteresebb effektek egyike. Az effekt megszületése az elektromos jelek erősítésére szolgáló berendezések zenészkörökben történő elterjedéséhez köthető. Az első ilyen effektek az áramkör sérülésétől vagy nem rendeltetésszerű használatából alakultak ki. Ez többnyire azt jelentette, hogy az áramkör bemeneteire nagyobb feszültségű jeleket vezettek, mint amit a specifikáció megengedett, ezért a kimeneti jel torzult. Ez a hatás az elektromos jelútban lévő erősítőkomponensek vagy a hangszórók túlvezérlése miatt jött létre. Ezekben az erősítőberendezésekben a jel erősítését akkoriban elektroncsövek végezték. Az elektroncsöveket tartalmazó áramkör túlvezérlése során az elektroncsöveket a lineáris tartományukon kívül használták, ezért az elektroncsövek a karakterisztikájukra jellemző módon torzították az áramkör kimeneti jeleit. Az elektroncsövek által okozott torzítás nagyon jellegzetes, nem hasonlít a hangtechnikában előforduló más torzítási típusokra. A zenei élet hamar felkapta az új hangzásvilágot, új zenei stílusok alakultak ki, melyek elképzelhetetlenek torzításeffekt nélkül. A torzításeffekt más fajtái is léteznek más karakterisztikájú áramköri elemekre épülve, de alapvetően a csövek által okozott torzítás számít etalonnak a zenész társadalom számára. Rendkívül sokrétű torzítási mód érhető el vele.

Az elektroncsöveken alapuló torzítóeffekt rendkívül érzékeny az áramkörben lévő elektroncső karakterisztikájára. Nagyon nehéz két olyan elektroncsövet találni, melyek azonos torzítást eredményeznek. Az elektroncsövek karakterisztikai időben változnak. Erre a hatásra a zenészek azt a kifejezést szokták használni, hogy az elektroncsövek *előregednek*. Ha ez bekövetkezik, romlik az effekt minősége, az elektroncsövet ki kell cserélni. Az elektroncsövek fűtése komoly teljesítményt kívánhat, nagy a méretük, emellett a zajuk sem mindig elhanyagolható.

Az elektroncső fenti hátrányai miatt, és a számítógépek számítási kapacitásának rohamos növekedése miatt érdemes az elektroncsöveken alapuló torzítási effekt szoftveres megoldásával foglalkozni. A kifejlesztett algoritmus egyszerűbb változata később akár beágyazott rendszerben is megvalósítható egy alacsonyabb költségű elektroncsöves torzítást szimuláló torzító pedálban.

Az elektroncsöves torzító pedálok közül a *B. K. Butler Tube Driver* nevű torzító pedálra esett a választásom, mert ez a pedál a leghíresebb a kategóriáján belül. Az áramkört B.K. Butler fejlesztette a 70-es és 80-as években. Egy elektroncsöves lemezjátszó erősítőfokozaton alapul. A legnépszerűbb verziója sok iteráció által alakult ki, melyet többek között Eric Johnson, David Gilmour, Billy Gibbons is előszeretettel használt. Az effekt manapság is nagy népszerűségnek örvend [6].

Feladatom egy nemlineáris viselkedést mutató analóg áramkör szimulációs algoritmusának megtervezése, implementációja és validálása volt. Munkám során meg kellett találnom az optimális megoldást az algoritmus különböző paramétereinek között az algoritmus stabilitásának garantálása mellett. Ezen paraméterek az eredeti áramkör működéséhez való hűség és az erőforrásigény voltak. Az algoritmus valósidejű futtatásához magasak a hardverkövetelmények, ezért a létrehozott algoritmusnak a lehető leghatékonyabb módon kell a hardvert használnia. A dolgozat bemutatja, hogy milyen mérnöki eszközök felhasználásával lehet a fenti problémát megoldani az analóg áramköri analízis, folytonos idejű rendszerek diszkretizálása, nemlineáris rendszerek leírása, iteratív numerikus módszerek, egyenletrendszerek hatékony megoldása, vákuumcsövet tartalmazó áramkörök bemérése és hatékony C++ implementáció területén.

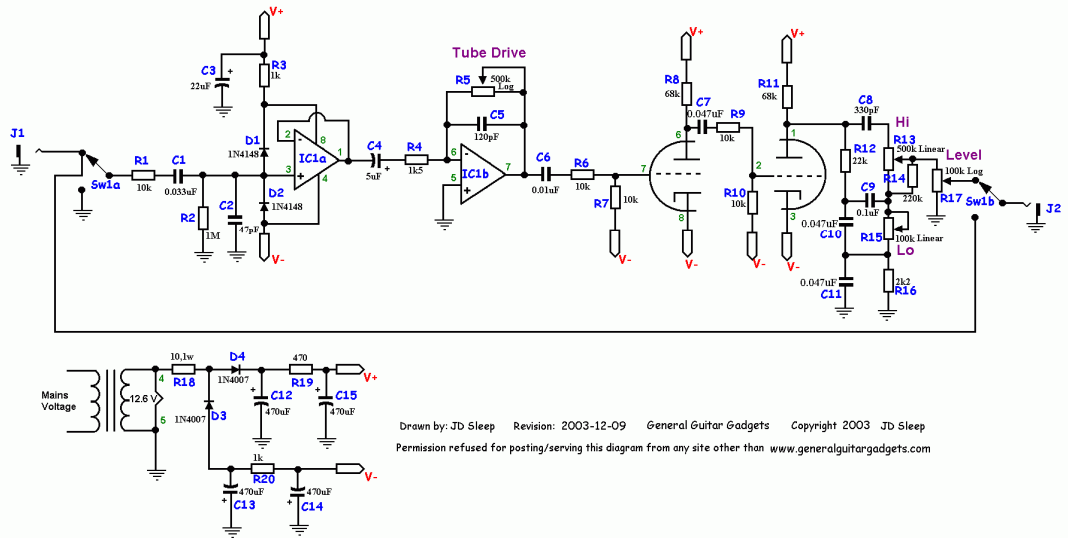
2 A modellezendő áramkör: B. K. Butler Tube Driver

A kiválasztott áramkör egy elektroncsövet tartalmaz, melyen belül két erősítő trióda található. Az áramkör egy bemenettel és egy kimenettel rendelkezik, a négy paramétere potenciométerekkel állítható be. Az áramkör paraméterei a következők: *hangerősség, magas, mély, torzítás mértéke*. A második és a harmadik paraméter a hangszínszabályzó áramköri részt vezérli. A *torzítás mértéke* paraméter a jelútban második műveleti erősítő erősítését befolyásolja. Ez a műveleti erősítő kellően nagy erősítés esetén *hard clipping* módon torzítja a bemenő jelét. A *hangerősség* paraméter arra szolgál, hogy a *torzítás mértékének* beállítását után kompenzálni lehessen a második erősítő erősítésének hatását a teljes rendszer erősítésére.



1. ábra: Tube Driver torzító pedál előnézeti képe

Tube Driver



2. ábra: Tube Driver torzító pedál kapcsolási rajza

Az áramkörnek létezik egy módosított változata, mely egy új állítható paramétert ad a rendszerhez. Ezzel a paraméterrel az elektroncsőben lévő triódák katódjai és a negatív tápfeszültség közötti impedanciát lehet befolyásolni. Az 1. ábrán épp egy ilyen változat látható. Az új paramétert a kép jobb felső sarkában lévő potenciométerrel lehet beállítani. A modell az eredeti áramkörön alapul, melynek áramköri kapcsolása a 2. ábrán látható.

3 Elméleti összefoglaló

Ebben a fejezetben azokat az elméleti, matematikai alapokat foglalom össze, amik feltétlenül szükségesek voltak a feladat megoldásához.

3.1 Nemlineáris rendszer modellezése

Nemlineáris rendszerek modellezésére sok módszer létezik. A két legelterjedtebb megközelítési módszer a fekete doboz alapú modellezés és az áramkör alapú szimuláció. A fekete doboz alapú modellt rendszerint akkor érdemes alkalmazni, amikor a rendszer belső működése nem ismert. Ilyen esetekben, további információ hiányában, a rendszer adott bemeneti jelekre adott válaszából történik a rendszer becslése. A modellezendő áramkör nemlinearitása, és paramétereinek nagy száma miatt rendkívül sok mérésre lenne szükség ahhoz, hogy elegendő információ álljon rendelkezésre a rendszer kielégítő pontosságú becsléséhez. A modell illesztése, és valósidejű futásra való optimalizálása egyáltalán nem lenne triviális feladat.

Jelen esetben ismert az áramkör felépítése és paramétereinek értéke, ezért a modellalkotást alapozhattam áramköri szimulációra. A teljes modell a diszkrét áramköri elemek modelljeiből és az összekapcsolásukból adódó kényszerekből épül fel. A nemlinearitásokat statikus nemlinearitásként kezelhettem, mert a nemlineáris elemek dinamikája elhanyagolható volt a *Spice* szimuláció alapján. A csövek karakterisztikáit leíró egyenletek túl bonyolultak ahhoz, hogy valós időben futtatható modellt eredményezzen a közvetlenül rájuk épülő modell, ezért a statikus nemlinearitásokat leíró függvények értékeit táblázatokba mentettem, melyből a konkrét értékeket interpolációval számoltam. A nemlineáris részrendszereket gyakorlatilag olyan módon modelleztem ebben a munkában, hogy áramköri analízissel meghatároztam a folytonos idejű rendszert leíró egyenleteket, majd biztosítottam, hogy az egyenletrendszereket minden időpontban a lehető legjobban kielégítsék a modell állapotváltozói és bemeneti értékei. Ezzel a módszerrel lehet a legpontosabban modellezni az áramkört, és a legegyszerűbben megtalálni a modellben vétett hibákat, mert a felvett állapotváltozóknak jelen esetben meg kell egyezniük a *Spice* programban szimulált rendszer állapotváltozóinak értékeivel. Az eltérésekből egyszerűen következtethető a hiba helye. Fekete doboz alapú modellezésnél a modell javítása nem lenne ilyen

egyszerű. A modellnek három állítható paramétere van, melyek nagyban befolyásolják a rendszer működését.

Az alfejezetben felsorolt tények valóban azt indokolják, hogy érdemes áramkör analízis alapján és nem mérésen alapuló modell identifikációs eljárással, azaz nem fekete doboz modellel megközelíteni a problémát.

3.2 Diszkretizálási módszerek

Diszkretizálási eljárások során folytonos idejű dinamikus rendszerek diszkrét idejű dinamikus rendszerekkel való legjobb közelítése a feladat. Folytonos idejű dinamikus rendszerekben az állapotváltozók deriváltjai az állapotváltozók aktuális értékétől és a bemeneti jeltől függenek.

$$\frac{\partial \mathbf{x}}{\partial t} = \mathbf{A} \cdot \mathbf{x} + \mathbf{B} \cdot \mathbf{u} \quad \mathbf{1}$$

\mathbf{x} : állapotváltozók

\mathbf{u} : bemeneti jelek

\mathbf{A} , \mathbf{B} : állapotváltozós leírás mátrixai

Diszkrét időben nem áll rendelkezésünkre minden folytonos idejű időpillanatban az állapotváltozók értéke. A probléma megoldásához az alábbi módszerek állnak rendelkezésre [5].

3.2.1 Előrelépő Euler módszer

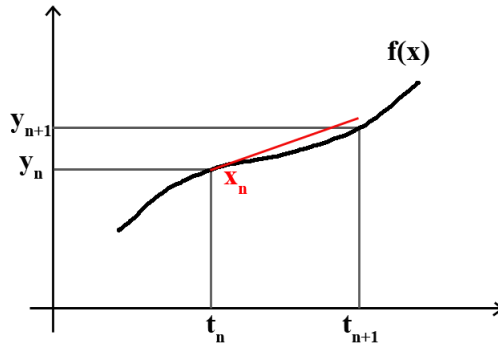
Az előre lépő Euler módszer során a t_n és t_{n+1} időpillanatok közötti időintervallumban az állapotváltozók deriváltjának számolásához a t_n időpillanathoz tartozó állapotváltozó értékeket használjuk (3. ábra) [5]. Egyenlettel kifejezve:

$$y[n+1] - y[n] = T \cdot x[n] \quad \mathbf{2}$$

y : állapotváltozó értéke

x : állapotváltozó idő szerinti deriváltja

T : mintavételi frekvencia reciproka



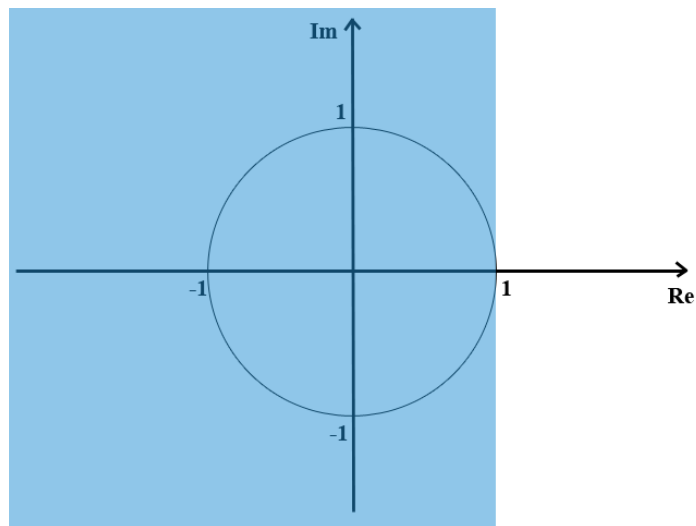
3. ábra: Előrelépő Euler módszer

Ez a módszer nemlineáris rendszerek modellezésénél jól alkalmazható, mert az állapotváltozók változásának számolásakor a változás nagysága az állapotváltozók jövőbeli értékétől nem függ, ezért az egyenletrendszer gyökeinek keresése során az állapotváltozók értékei már rendelkezésünkre állnak. A módszer hátránya, hogy a diszkrét idejű rendszer stabilitása nem garantált stabil folytonos idejű rendszerek transzformációja esetén sem.

A rendszer pólusai az alábbi módon módosulnak:

$$z = s \cdot T + 1$$

3



4. ábra: Előrelépő Euler módszer, pólusok transzformációja

A 4. ábrán a folytonos idejű stabil pólusok transzformációja látható az előre lépő Euler módszer során. A kék színnel besatírozott félsíkra transzformálódnak a folytonos időben stabil pólusok. Azok a pólusok maradnak diszkrét időben is stabilak, melyek a transzformáció után az egységkörön belül lesznek. Mindig létezik olyan kellően nagy

mintavételi frekvencia, mellyel biztosítható a diszkrét idejű pólusok stabilitása. A kérdés az, hogy mennyire nagy ez a mintavételi frekvencia, és elegendő számítási kapacitás áll-e rendelkezésre a valós idejű futtatáshoz.

3.2.2 Hátralépő Euler módszer

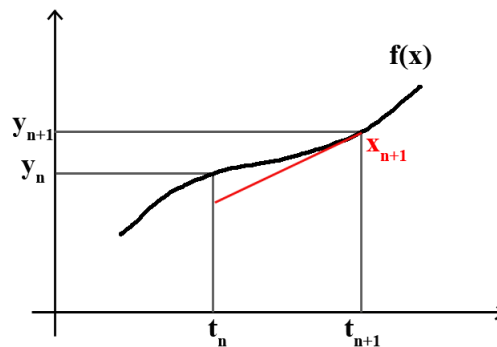
A hátralépő Euler módszer során a t_n és t_{n+1} időpillanatok közötti időintervallumban az állapotváltozók deriváltjának számolásához a t_{n+1} időpillanathoz tartozó állapotváltozó értékeket használjuk (5. ábra) [5]. Ez azt jelenti, hogy az állapotváltozók jövőbeli értékét használjuk a deriváltak számolásához. Ugyanez egyenlettel leírva:

$$y[n] - y[n-1] = T \cdot x[n] \quad 4$$

y : állapotváltozó értéke

x : állapotváltozó idő szerinti deriváltja

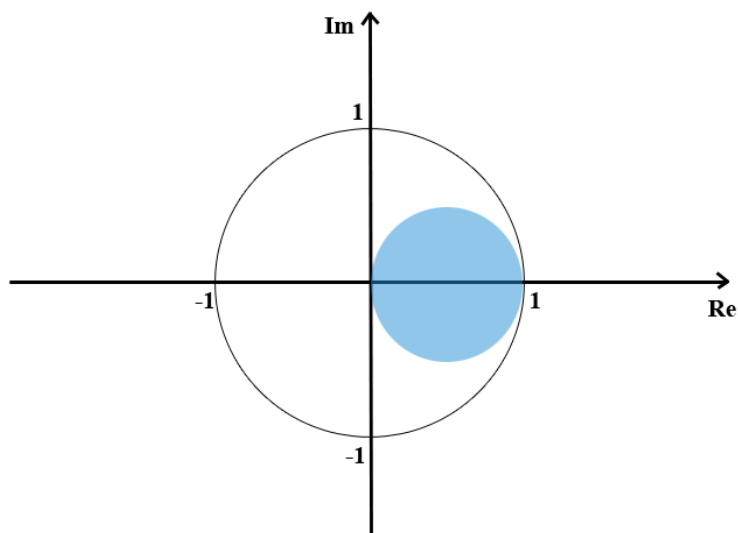
T : mintavételi frekvencia reciproka



5. ábra: Hátralépő Euler módszer

A módszer a rendszer pólusait az alábbi módon transzformálja:

$$z = \frac{1}{1 - s \cdot T} \quad 5$$



6. ábra: Hátralépő Euler módszer, pólusok transzformációja

A 6. ábrán a hátralépő Euler módszer pólusokra kifejtett hatása látható. A módszer előnye, hogy stabil folytonos idejű rendszerből mindig stabil diszkrét idejű rendszert hoz létre, mert a pólusok mindig az egységkörön belülre transzformálódnak. A módszer hátránya, hogy a diszkrét rendszer frekvenciamenete kevésbé hasonlít a folytonos idejű rendszer frekvenciamenetére, mint az előrelépő Euler módszer esetében, és implicit egyenletekhez vezet nemlineáris rendszerek leírása esetén. Ez azt jelenti, hogy a gyökkereső algoritmus során a dinamikus állapotváltozók t_n időpillanatbeli értékeit is keresni kell.

3.2.3 Impulzusinvariáns transzformáció

Ez a diszkrétizálási eljárás a rendszer ismert impulzusválaszát veszi alapul. Az eljárás során a rendszer impulzusválaszából mintákat veszünk diszkrét időpontokban, majd a kapott értékek alapján végezzük el a digitális szűrő tervezését. Az így kapott diszkrét idejű rendszerben a mintavételi frekvencia felénél nagyobb frekvenciájú komponensek átlapolódnak frekvenciatartományban, ezért aluláteresztő jellegű rendszerek diszkrétizálására alkalmazható leginkább. A paraméterek változása esetén újra el kell végezni a folytonos idejű átviteli függvény részlettörtekre bontását. Emiatt és a frekvenciatartománybeli átlapolódás miatt ezt a diszkrétizálási módszert nem alkalmaztam.

3.2.4 Bilineáris transzformáció, trapéz módszer

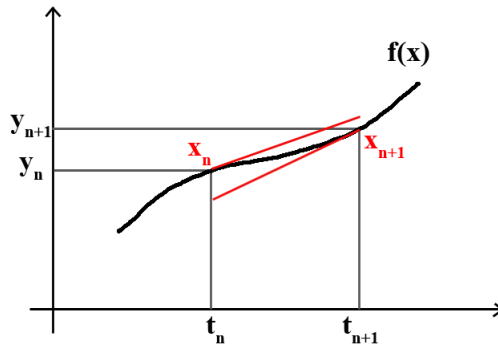
A bilineáris transzformáció az előrelépő és hátralépő Euler módszerek kombinációja [5]. A módszer során a t_n és t_{n+1} időpillanatok közötti időintervallumban az állapotváltozók deriváltjának számolásához a t_n és t_{n+1} időpillanatokhoz tartozó állapotváltozó értékek átlagát használjuk.

$$y[n] - y[n-1] = T \cdot \left(\frac{x[n] + x[n+1]}{2} \right) \quad 6$$

y : állapotváltozó értéke

x : állapotváltozó idő szerinti deriváltja

T : mintavételi frekvencia reciproka



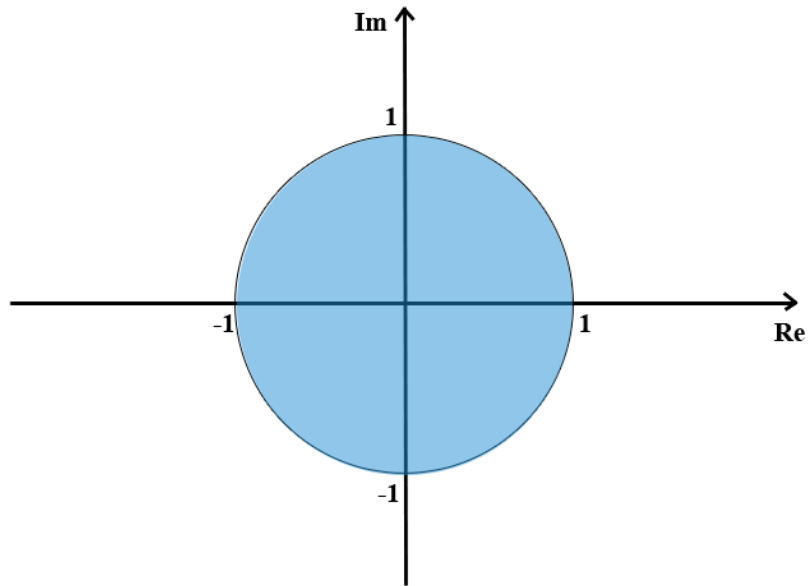
7. ábra: Bilineáris transzformáció

A bilineáris transzformációt a Z transzformáció elsőfokú Taylor-soros közelítésének is tekinthetjük. Az elsőfokú közelítés során az exponenciális függvényt az elsőfokú Taylor polinomjával közelítjük.

$$z = e^{s \cdot T} = \frac{e^{s \cdot T/2}}{e^{-s \cdot T/2}} \approx \frac{1 + s \cdot T/2}{1 - s \cdot T/2} \quad 7$$

A módszer során a rendszer pólusai az alább módon transzformálódnak:

$$z = \frac{1 + s \cdot T/2}{1 - s \cdot T/2} \quad 8$$



8. ábra: Bilineáris transzformáció, pólusok transzformációja

A 8. ábrán a bilineáris transzformáció pólusokra kifejtett hatása látható. A módszer nagy előnye, hogy a folytonos időben stabil pólusok félsíkját a diszkrét időben stabil pólusok egységkörére transzformálja, ez azt jelenti, hogy folytonos idejű, stabil rendszereket diszkrét idejű stabil rendszerekké transzformál a stabilitás határhelyzetének megtartásával. A nagyfrekvenciás frekvenciakomponensek nem lapolódnak át frekvenciatartományban. Az eljárás hátránya, hogy a 0 Hz és ∞ Hz közötti analóg frekvenciatartománybeli viselkedést a 0 Hz és a mintavételi frekvencia $\frac{1}{2}$ -szerese közé transzformálja, ezáltal torzítja a szűrő nagyfrekvenciás viselkedését. Magasabb mintavételi frekvencia esetében ez a hatás jelentősen csökken.

3.2.5 Runge-Kutta módszerek

A Runge-Kutta módszerek differenciálegyenletek numerikus közelítő megoldását teszik lehetővé. A módszert Carl David Tolmé Runge és Martin Wilhelm Kutta dolgozta ki 1900 körül. A módszernek sok változata létezik. A dolgozatban csak az alkalmazott módszerekről lesz szó. A Runge-Kutta módszerekről bővebben a [7]. forrásban olvashatunk.

Elsőfokú, explicit Runge-Kutta módszer:

Azonos a 3.2.1-es fejezetben leírt előrelépő Euler módszerrel.

Másodfokú, explicit, felezőpont (midpoint) Runge-Kutta módszer:

A másodfokú, explicit, felezőpont (midpoint) Runge-Kutta módszer két lépésben becsli az állapotváltozók idő szerinti deriváltját az aktuális időbeli lépésben. Az időtartomány középső időpontbeli deriváltját az állapotváltozók első időpontbeli értékeinek segítségével becsli az adott időtartományra vonatkozóan. Az így számolt középső időpontbeli differenciákat használja az algoritmus az egész időtartományra. Az algoritmust leíró egyenletek a következők:

$$f(y, u) := \frac{\partial y}{\partial t} = \mathbf{A} \cdot y + \mathbf{B} \cdot u \quad 9$$

$$y_{n+1/2} = y_n + \frac{\Delta T}{2} \cdot f(y_n, u_n) \quad 10$$

$$y_{n+1} = y_n + \Delta T \cdot f(y_{n+1/2}, u_n) \quad 11$$

y : állapotváltozó értéke

$y_{n+1/2}$: segédváltozó

ΔT : mintavételi frekvencia reciproka

3.3 Gyökkereső algoritmusok

Munkám során olyan egyenletrendszerek megoldására volt szükségem, melyeknek nem létezik explicit megoldása, ezért iteratív gyökkereső algoritmusokat kellett használnom [8].

3.3.1 Newton-Raphson módszer

Iteratív gyökkereső módszer, mely a függvény deriváltjának segítségével számítja ki a gyökhely következő iterációbeli értékét. Az algoritmus minimum négyzetesen konvergál a megoldáshoz a gyökhely közelében. Az algoritmus konvergenciája kérdéses, ha a függvény deriváltja a gyökhely környezetében nem folytonos vagy nullával egyenlő. A kezdeti érték megválasztásától nagyban függ a módszer sikeressége, főleg nem szigorúan monoton függvényeknél.

Az elektroncsövek karakterisztikáját szigorúan monoton függvény adta meg. Az iteráció minden esetben konvergens volt a használt időlépték mellett.

A Newton-Raphson módszer egyenlete:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad 12$$

$f(x)$: függvény, melynek a gyökhelyét keressük

$f'(x)$: függvény deriváltja az x pontban

A módszer egyenlete egyenletrendszerekre kiterjesztve:

$$\mathbf{J}_f(\mathbf{x}_n) \cdot (\mathbf{x}_{n+1} - \mathbf{x}_n) = -\mathbf{F}(\mathbf{x}_n) \quad 13$$

\mathbf{J}_f : az $f(\mathbf{x})$ egyenletrendszer Jacobi-mátrixa, melyet az alábbi módon lehet meghatározni az egyenletek és változók segítségével:

$$\mathbf{J} = \begin{bmatrix} \frac{\partial F}{\partial x_1} & \dots & \frac{\partial F}{\partial x_n} \end{bmatrix} \quad 14$$

3.3.2 Szekáns módszer

A szekáns módszer a Newton-Raphson módszer véges differenciákkal történő közelítése, mely a Newton-Raphson módszerben szereplő deriváltat véges differenciákkal helyettesíti [8].

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad 15$$

$$f'(x_{n-1}) \approx \frac{f(x_{n-1}) - f(x_{n-2})}{x_{n-1} - x_{n-2}} \quad 16$$

Ebben a munkában ezt a változatot használtam, mert a nemlineáris függvényt bonyolultsága miatt nem az egyenletekből számoltam, hanem egy előre kiszámolt táblázatban tároltam, és abból interpoláltam kétdimenziós lineáris interpolációval. Ahhoz, hogy a Newton-Raphson módszert használhassam munkám során, el kellett volna tárolni a deriváltak táblázatait is, ami még négy táblázatot jelentett volna, amiben

szintén interpolálni kellett volna. Ez többlet memóriaigényt, több munkát és több számítást igényelt volna, bár a módszer konvergenciája valószínűleg javult volna.

3.4 Lineáris egyenletrendszert megoldó algoritmusok

Az áramköri szimulációs leírás végeredménye többismeretlenes egyenletrendszer, melynek a rendszer állapotváltozóit, bemeneti értékeit és paramétereit behelyettesítve minden időpillanatban igaznak kell lennie a hibahatáron belül.

A megoldandó probléma a következő:

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{b} \quad 17$$

Ezt az egyenletrendszert kell a lehető legkevesebb számítással megoldani az \mathbf{x} vektor értékére.

3.4.1 Egyenletrendszer megoldása Inverz számításával

$$\mathbf{A}^{-1} \cdot \mathbf{A} \cdot \mathbf{x} = \mathbf{A}^{-1} \cdot \mathbf{b} \quad 18$$

$$\mathbf{x} = \mathbf{A}^{-1} \cdot \mathbf{b} \quad 19$$

Az inverz számításának egyik módja:

Ha $\det(\mathbf{A}) \neq 0$:

$$\mathbf{A}^{-1} = \frac{1}{\det(\mathbf{A})} \cdot \text{adj}(\mathbf{A}) \quad 20$$

$$\text{adj}(\mathbf{A})_{ij} = \left((-1)^{i+j} \cdot \mathbf{M}_{ij} \right)^T \quad 21$$

\mathbf{M}_{ij} : \mathbf{A}_{ij} mátrix elemhez tartozó minor

Ezt a módszert a Jacobi-mátrixot tartalmazó egyenletrendszer számításánál alkalmaztam. A Jacobi-mátrix sok nulla elemet tartalmazott, ezért az inverz számítás kevesebb számítást igényelt, mint általános esetben. A visszacsatolást elhanyagoló implicit, bilineáris modell ezt a módszert alkalmazza az egyenletrendszer megoldására. Az erőforrások szempontjából érzékenyebb, tökéletesen csatolt, bilineáris modell esetében az LU felbontással történő egyenletrendszer megoldást használtam.

3.4.2 Egyenletrendszer megoldása LU felbontással

A fenti egyenletrendszer sokkal gyorsabban megoldható az \mathbf{A} mátrix alsó és felső háromszög mátrixra bontásával. A felbontást követően a változók behelyettesítésével megoldható az egyenletrendszer.

Az LU felbontásnak sok fajtája létezik, munkám során a Crout-féle LU felbontást alkalmaztam [9].

Az \mathbf{L} és \mathbf{U} mátrixok 3x3-as \mathbf{A} mátrix esetén:

$$\mathbf{A} = \mathbf{L} \cdot \mathbf{U} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \cdot \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \quad 22$$

A behelyettesítés módszere:

$$\mathbf{A} \cdot \mathbf{x} = \mathbf{L} \cdot \mathbf{U} \cdot \mathbf{x} = \mathbf{b} \quad 23$$

$$\mathbf{y} := \mathbf{U} \cdot \mathbf{x} \quad 24$$

$$\mathbf{L} \cdot \mathbf{y} = \mathbf{b} \quad 25$$

Mivel az \mathbf{L} mátrix alsó háromszög mátrix, ezért az \mathbf{y} vektor értékei behelyettesítéssel kiszámolhatók.

Például
$$y_1 = \frac{b_1}{l_{11}}, y_2 = \frac{b_2 - l_{21} \cdot y_1}{l_{22}} \dots$$

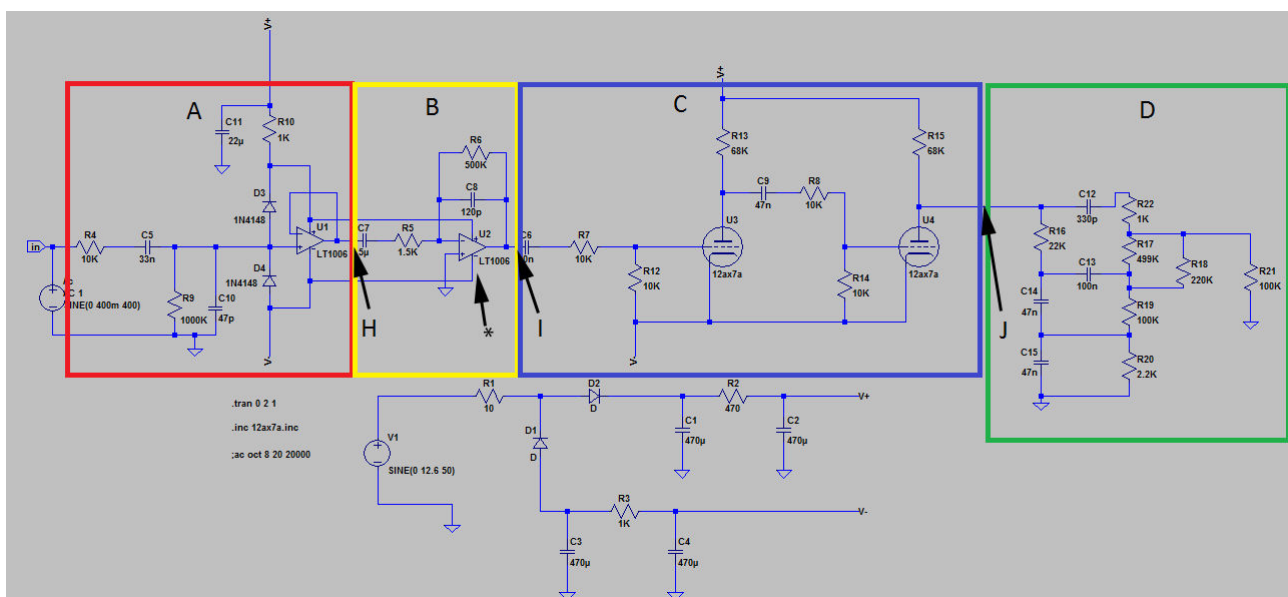
Ezután a 26. egyenlet hasonlóképpen megoldható.

$$\mathbf{U} \cdot \mathbf{x} = \mathbf{y} \quad 26$$

Ezt a módszert a tökéletesen csatolt, implicit modellben használtam a gyökkereső iterációs ciklusban.

4 A folytonos idejű modell létrehozása

A modellezendő rendszer valódi, diszkrét elemekből összeépített áramkör, melynek az áramköri rajza és elemei ismertek. Ésszerű volt az áramköri rajzból, és az áramköri elemek modelljeiből kiindulni. Első lépésként érdemes volt a rendszert olyan kisebb, egyszerűbb részáramkörökre felbontani, melyek elhanyagolhatóan kis mértékben hatnak vissza az előző fokozatok működésére. Ez a felbontás egyszerűbbé teszi a modellt, és külön ellenőrizhetővé teszi az egyes részelemeket, így lecsökkentve a hibajavításra fordított időt.



9. ábra: Az áramkör részekre osztása

Ezt a fajta elválasztást leginkább a 9. ábrán látható 'H' és 'I' pontokon tehetjük meg, mert a műveleti erősítő nagy bemeneti ellenállása miatt elhanyagolhatóan kis árammal terhelődik az előző fokozat kimenete mindkét esetben. A 'J' pontban ez az állítás nem igaz, mert a parametrikus hangszínszabályzó bemeneti árama hatással van a második elektroncső anódjának feszültségére, viszont az elektroncsöveket tartalmazó részáramkör modelljének bonyolultságát nagymértékben csökkenti ez az egyszerűsítés. Az egyszerűsítés hatása a 5.3.2.1-es fejezetben olvasható. A modellalkotás során különböző csatolási módokat is kipróbáltam az elektroncsöveket tartalmazó

részáramkör és a hangszínszabályzó között. Az így keletkezett különböző modelleket összehasonlítottam erőforrás felhasználás és minőség szempontjából.

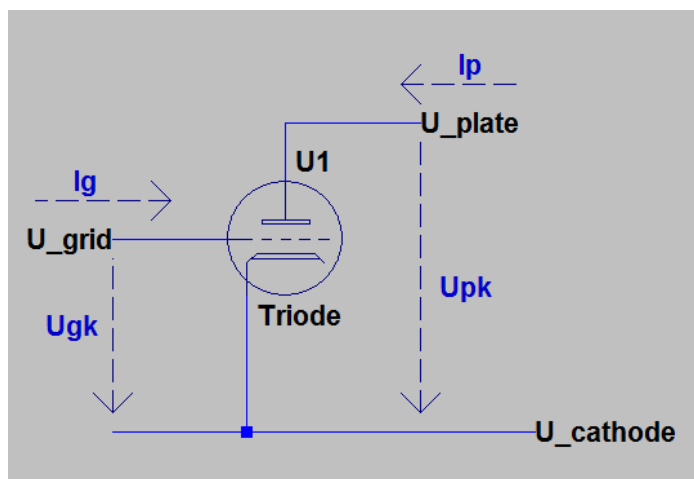
Az így keletkezett részáramköröket két csoportra kellett osztanom: a lineárisan leírható csoportra és a nemlineáris csoportra. Mivel a lineárisnak tekinthető áramköri részek leírhatók kevesebb számítási kapacitást felhasználó explicit módszerekkel, ezért a csoportosítás során arra kellett törekednem, hogy a nemlineáris módon modellezett rész a lehető legkisebb legyen. Emiatt bontottam szét az áramkört az 9. ábrán látható 'J' pontban.

Ebben az alkalmazásban az elektroncsövek a lineáris tartományokon kívül is működnek. Az áramkörre jellemző különleges hangzás az ebből származó torzításnak köszönhető, ezért az elektroncsöveket nem tekinthetjük lineáris működésűnek, a nemlineáris modelljük felhasználása szükséges, mely a lineáris működési tartományon kívül is jól leírja viselkedésüket. Ugyanez igaz a kapcsolásban szereplő '*' jelzésű műveleti erősítőre, mely túl van vezérelve. A fent leírtak alapján a 'B' és 'C' áramköri rész, azaz az elektroncsövek környezete és a nemlineáris tartományban működő műveleti erősítő környezete lesz a nemlineáris része, az 'A' és 'D' áramköri rész pedig a lineáris része a modellnek.

4.1 Az elektroncső modell

Az elektroncsöveket statikus nemlinearitásként modelleztem. Ezt azért tehettem meg, mert a csöveket jellemző időállandók elegendően kicsik ahhoz, hogy ebben az alkalmazásban ne legyenek hatással a működésre. Sok csőmodell azt feltételezi, hogy az anód-katód feszültség mindig pozitív, ezért hibás értékek adódnak negatív anód-katód feszültségek esetén. A Norman Koren elektroncső modell a teljes tartományon megfelelően működött, ezért ezt a modellt alkalmaztam. [10]

Az elektroncsőnek, mint triódának, a működését leíró áramok és feszültségek a 10. ábrán láthatók.



10. ábra: Elektroncsőt jellemző mennyiségek (rács-katód feszültség, anód-katód feszültség, rácsáram és anódáram)

Az anódáramot leíró egyenletek:

$$E_1 = \frac{U_{pk}}{k_p} \cdot \ln \left(1 + e^{k_p \cdot \left(\frac{1}{\mu} + \frac{U_{gk} + U_{ct}}{\sqrt{k_{VB} + U_{pk}^2}} \right)} \right) \quad 27$$

$$I_p = \frac{E_1^{E_x}}{K_g} \cdot (1 + \text{sgn}(E_1)) \quad 28$$

I_p : Az elektroncső anódárama

U_{gk} : Az elektroncső rács-katód feszültsége

U_{pk} : Az elektroncső anód-katód feszültsége

$\mu, E_x, K_g, K_p, U_{ct}, K_{vb}$: az elektroncső modell paramétere

Az elektroncső rács-katód kapcsolata gyakorlatilag egy diódát alkotnak. A rácsáram egy soros ellenállással bíró dióda áramával modellezhető. A modellben a dióda zárótartományán belül a diódán átfolyó áram értéke nulla amper. A karakterisztika azon része, ahol az ellenállás hatása a meghatározó, egy lineáris egyenlettel leírható. A két lineáris szakasz közötti tartományban másodfokú egyenlet írja le a dióda áramának nagyságát a diódára kapcsolt feszültség függvényében. A függvény deriváltja folytonos mindkét csatlakozási pontban.

A rácáramot leíró egyenletek:

$$I_g = \left\{ \begin{array}{ll} 0 & , ha \quad U_{gk} < U_\gamma - K_n \\ \frac{U_{gk} - U_\gamma}{R_{gk}} & , ha \quad U_{gk} > U_\gamma + K_n \\ a \cdot U_{gk}^2 + b \cdot U_{gk} + c & , \quad egyébként \end{array} \right\} \quad 29$$

$$a = \frac{1}{4 \cdot K_n \cdot R_{gk}} \quad 30$$

$$b = \frac{K - U_\gamma}{2 \cdot K_n \cdot R_{gk}} \quad 31$$

$$c = -a \cdot (U_\gamma - K_n)^2 - b \cdot (U_\gamma - K_n) \quad 32$$

I_g : Az elektroncső rácárama

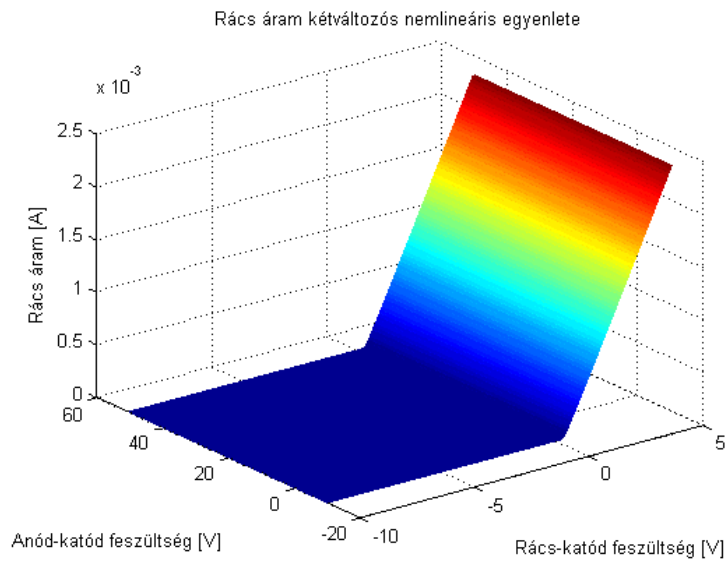
a, b, c : a másodfokú függvény paraméterei

R_{gk} : a rácás áramot leíró dióda modell soros ellenállása

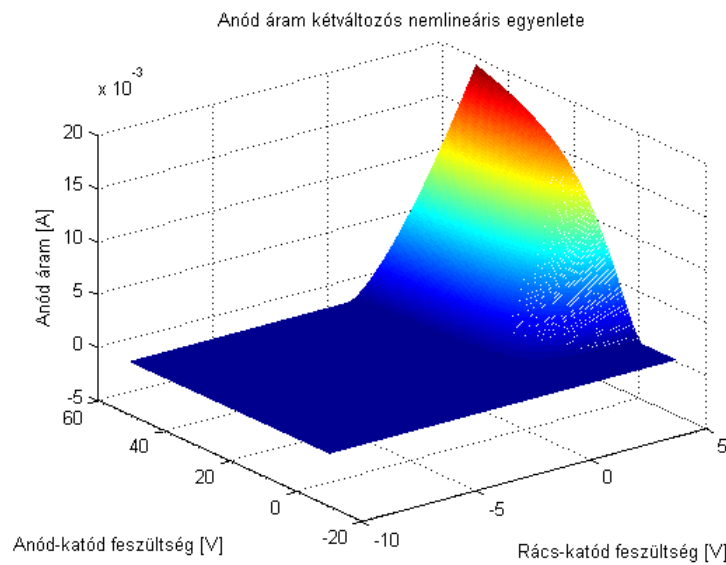
U_γ : a feszültség, ahonnan a rácáram lineárisnak tekinthető

K_n : a másodfokú tartomány szélességének a fele

Az elektroncső működését leíró táblázatok a 11. ábrán és 12. ábrán láthatóak.



11. ábra: A rácsáramot leíró táblázat



12. ábra: Az anódáramot leíró táblázat

$$I_g = f_1(U_{gk}) \quad I_p = f_2(U_{gk}, U_{pk})$$

33

A modellben használt I_g és I_p táblázat nagyon pontosan megfelel a *Spice* programban használt trióda karakterisztikájának, mert mindkettő a Norman Koren modell alapján működik azonos paraméterekkel. Ez feltétele a *Spice* alapú és Matlab alapú szimulációk összehasonlításának.

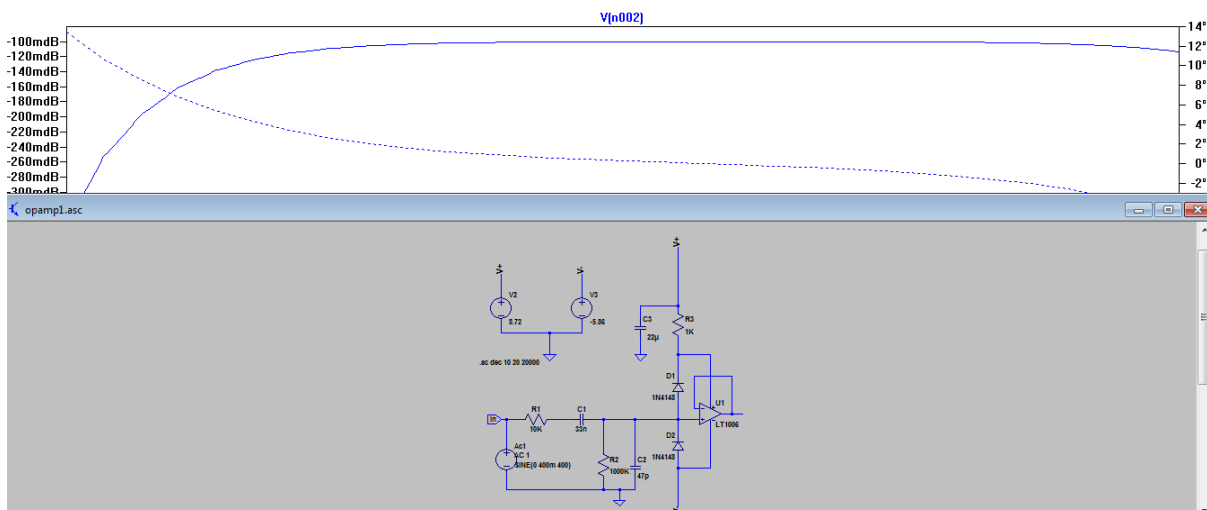
4.2 Lineáris áramköri blokkok

4.2.1 Az előerősítő

Az elektromos szólógitárok mágneses hangszedői rendszerint pár száz millivolt amplitúdójú jelet képesek létrehozni a kimenetükön. Fender és Gibson márkájú hangszedőket megvizsgálva 400 kOhm és 1 MOhm között mozog ezeknek az eszközöknek a kimeneti impedanciája. A megfelelő impedanciacsatolás érdekében nagy bemeneti ellenállású előerősítő áramkör jelenléte szükséges. Az 'A' jelzésű áramköri rész épp ezt a funkciót látja el.

Az áramköri kapcsolásból látható, hogy az előerősítő áramkörben található műveleti erősítő egy egyszeres erősítésű puffere erősítőként működik. Ezt a komponens csak passzív alkatrészek és a védődiódák előzik meg a jelútban. A védődiódákon csak akkor folyik áram, ha a bemenetre az üzemi tartományon kívül eső feszültség kerül, ami károsíthatná a berendezést. Mivel a modell szoftverben fog futni és üzemszerű működésnél a diódák nem befolyásolják az áramkört, ezért ez a védelem elhagyható. Az elektromos gitárok kimeneti jelének maximális amplitúdója rendszerint 400 mV-nál nem nagyobb. Basszusgitárok esetében ez az érték 2-3 V is lehet, de ennek a műveleti erősítőnek a bemeneteire még ebben az esetben sem juthat olyan jel, ami a két tápfeszültség által meghatározott feszültségtartományon kívülre esik, tehát a műveleti erősítő lineáris tartományban üzemel, ezért a viselkedése leírható az átviteli függvényével.

Mivel ez egy audio alkalmazás, ezért azok az áramköri részek elhanyagolhatóak, melyek az átviteli függvényt elhanyagolhatóan kis mértékben befolyásolják a hangfrekvenciás frekvenciatartományon belül. A SPICE szimuláció alapján az 9. ábrán pirossal bekeretezett áramköri rész a hangfrekvenciás frekvenciatartományban maximálisan -0.34 dB-el (20 Hz közelében) változtatja meg az átviteli karakterisztikát.



13. ábra: Az előerősítő kapcsolás áramköri rajza és átviteli függvénye

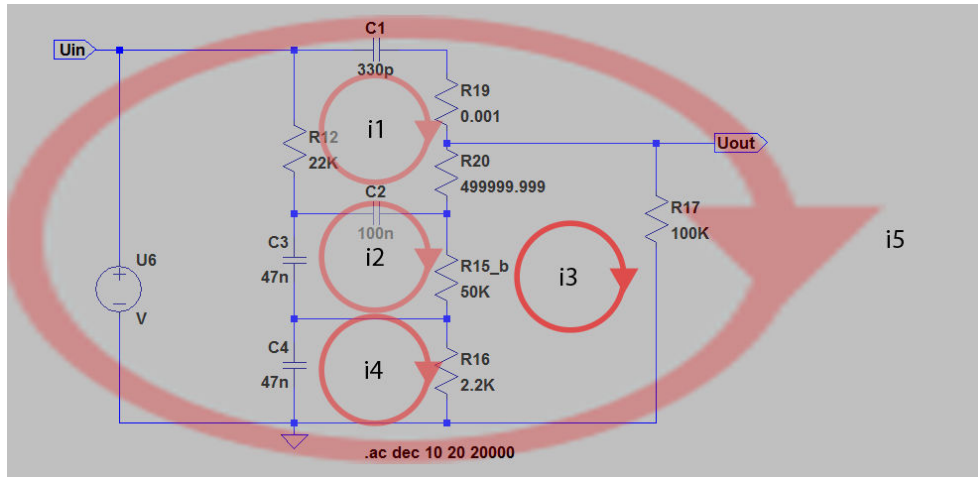
Az áramkör ezen része tehát a modellben elhanyagolható, mert elhanyagolhatóan kis mértékben változtat csak a hangzáson. A valódi áramkörben mindenképpen indokolt a jelenléte, mert a hangszer és az áramkör közötti impedanciacsatolást teszi megfelelővé, és kiszűri a bemenő jel egyenáramú és hangfrekvenciás sávnál nagyobb frekvenciájú komponenseit, melyek többnyire zajforrásokból származnak. VST plugin esetében, amiben a modell megvalósult, a bemenő jel eleve sávkorlátozott a mintavételi frekvenciának megfelelően, az egyenáramú komponens 0 V.

4.2.2 A parametrikus hangszínszabályzó

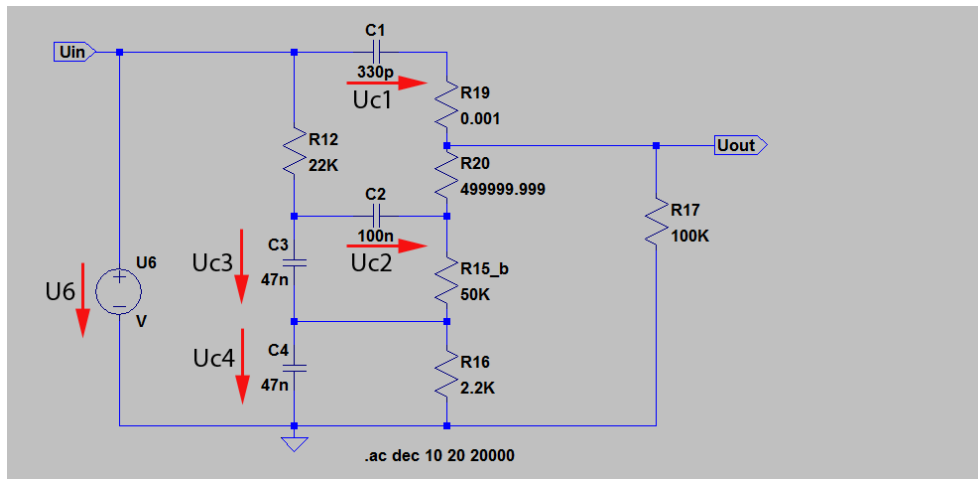
A parametrikus hangszínszabályzó lineáris működésű passzív alkatrészekből áll, ezért lineáris áramkörnek tekinthető, és parametrikus állapotváltozós egyenletekkel leírható. Az áramkör topológiája miatt a hurokáramok módszerével volt egyszerűbb megoldani a problémát. A hurokáramok száma a következő képlet alapján számolható:

$$e = a - cs + 1 \quad 34$$

Az egyenletben e a hurokáramok és felírandó egyenletek száma, a az ágak és cs a csomópontok száma. Ebben az áramkörben a csomópontok száma hat, az ágak száma tíz, tehát a felírandó egyenletek száma öttel egyenlő. Az áramkörben felvett hurokáramok és állapotváltozók az 14. ábrán és 15. ábrán láthatóak.



14. ábra: A hangszínszabályzó áramkörben felvett hurokegyenletek



15. ábra: A hangszínszabályzó áramkörben felvett állapotváltozók

A részáramkör két paraméterrel rendelkezik: magas, mély. A dolgozatban a *magas* paramétert t és a *mély* paramétert b szimbólum jelöli. A két paraméter értéke a $[0;1]$ intervallumban vehet fel értéket.

Az R_{19} és R_{20} ellenállások eredő ellenállások, melyeket az alábbi egyenletek határoznak meg:

$$R_{19} = R_{13} \cdot (1 - t) \quad 35$$

$$R_{20} = \frac{R_{13} \cdot R_{14} \cdot t}{R_{13} \cdot t + R_{14}} \quad 36$$

A fenti áramköri kapcsolásból, állapotváltozókból és hurokáramokból az alábbi lineáris egyenletrendszer vezethető le Kirchoff második törvénye alapján:

$$U_{c1} + (i_1 + i_5) \cdot R_{19} + (i_1 - i_3) \cdot R_{20} - U_{c2} + i_1 \cdot R_{12} = 0 \quad 37$$

$$U_{c2} + (i_2 - i_3) \cdot R_{15} \cdot b - U_{c3} = 0 \quad 38$$

$$(i_3 + i_5) \cdot R_{17} + (i_3 - i_4) \cdot R_{16} + (i_3 - i_2) \cdot R_{15} \cdot b + (i_3 - i_1) \cdot R_{20} = 0 \quad 39$$

$$(i_4 - i_3) \cdot R_{16} - U_{c4} = 0 \quad 40$$

$$U_{c1} + (i_5 + i_1) \cdot R_{19} + (i_5 + i_3) \cdot R_{17} - U_{c6} = 0 \quad 41$$

$$\begin{bmatrix} R_{19} + R_{20} + R_{12} & 0 & -R_{20} & 0 & R_{19} \\ 0 & R_{15} \cdot b & -R_{15} \cdot b & 0 & 0 \\ -R_{20} & -R_{15} \cdot b & R_{17} + R_{16} + R_{15} \cdot b + R_{20} & -R_{16} & R_{17} \\ 0 & 0 & -R_{16} & R_{16} & 0 \\ R_{19} & 0 & R_{17} & 0 & R_{19} + R_{17} \end{bmatrix} \cdot \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \end{bmatrix} = \begin{bmatrix} -U_{c1} + U_{c2} \\ -U_{c2} + U_{c3} \\ 0 \\ U_{c4} \\ -U_{c1} + U_{c6} \end{bmatrix} \quad 42$$

16. ábra: A hangszínszabályzót leíró parametrikus lineáris egyenletrendszer

A hangszínszabályzó állapotváltozós leírása:

$$\mathbf{x}' = \mathbf{A}_{EQ} \cdot \mathbf{x} + \mathbf{B}_{EQ} \cdot u \quad 43$$

$$y = \mathbf{C}_{EQ}^T \cdot \mathbf{x} + \mathbf{D}_{EQ} \cdot u \quad 44$$

\mathbf{x} : állapotváltozókból álló vektor

u : bemeneti jel

y : kimeneti jel

\mathbf{A}_{EQ} , \mathbf{B}_{EQ} , \mathbf{C}_{EQ} , \mathbf{D}_{EQ} : lineáris rendszert meghatározó mátrixok

A hangszínszabályzó állapotváltozós leírása a következő egyenletekből és a 16. ábrán látható egyenletrendszerből számolható ki.

$$U'_{c1} = \frac{i_1 + i_5}{C_1} \quad U'_{c2} = \frac{i_2 - i_1}{C_2} \quad 45$$

$$U'_{c3} = \frac{-i_2}{C_3} \quad U'_{c4} = \frac{-i_4}{C_4} \quad 46$$

$$U_{ki} = (i_3 + i_5) \cdot R_{17} \quad 47$$

A végleges állapotváltozós leírás mátrixaihoz a hurokáramokat ki kellett fejezni az állapotváltozók függvényében, és behelyettesíteni a fenti öt egyenletbe. Ezeket a számításokat a Matlab Symbolic Toolbox segítségével végeztem el.

A parametrikus hangszínszabályzó folytonos idejű állapotváltozós leírását a *SPICE* modellel összevetve ellenőriztem *Matlabban*. A hangszínszabályzó áramkörön átviteli függvény méréseket végeztem az *LTSpice* programmal, majd szöveges fájlba exportáltam azokat. Ezeket a szövegfájlokat később *Matlabbal* dolgoztam fel. A két modell frekvenciatartománybeli viselkedését hasonlítottam össze a paraméterek különböző értékeire. A paraméter értékeket úgy választottam meg, hogy a paraméterek bármely megengedett kombinációjára következtethessünk az alábbi összehasonlításból.

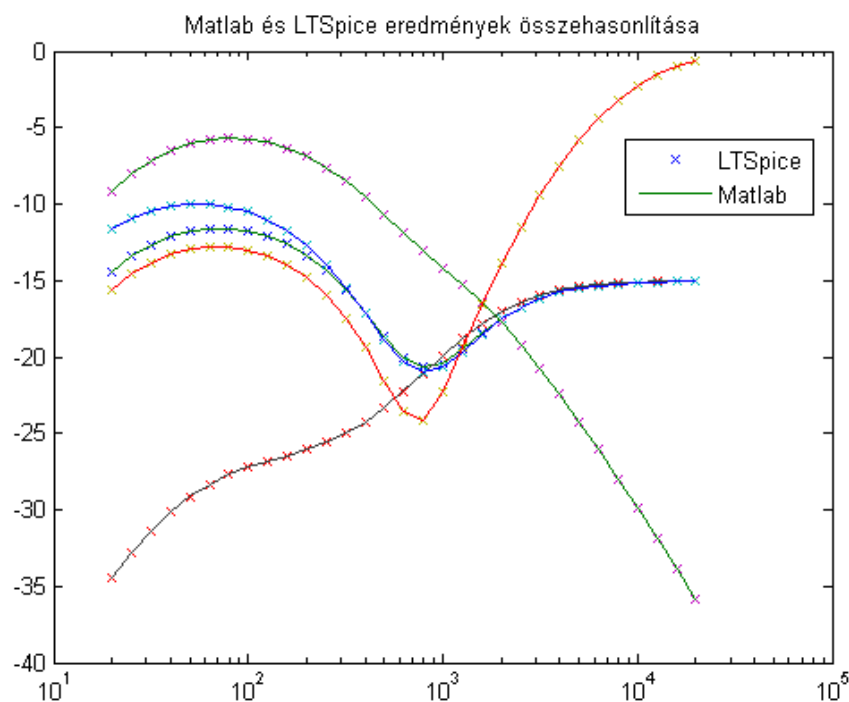
A paramétereket a $[0,1]$ intervallumban definiáltam. Fizikailag ez azt jelenti, hogy az adott beállításban a potméter maximális ellenállásának hanyad része az aktuális ellenállása. Logaritmikus potméterek esetében fontos szempont, hogy ugyanúgy viselkedjenek, mint a fizikai áramkörben. Ha nem logaritmikus a hangerősséget szabályzó paraméter, akkor érezhetően különböző a szoftveres modell vezérelhetősége az eredeti áramköréhez képest.

A tesztelést a paraméterek alábbi pontjaiban végeztem el:

1. táblázat: A hangszínszabályzó-mérés paramétertáblája

Mérési pont száma	Mély (b paraméter)	Magas (t paraméter)
1	0.5	0.5
2	0.0001	0.5
3	1	0.5
4	0.5	0.0001
5	0.5	1

A tesztelés eredménye:



17. ábra: Hangszínszabályzó folytonos idejű modelljének ellenőrzése LTSpice-szal

Az összehasonlítás eredményéből arra következtettem, hogy a hangszínszabályzó folytonos idejű állapotváltozós leírásának mátrixai helyesek.

4.3 Nemlineáris áramköri blokkok

Az áramkör modellezése során célszerű volt a nemlineárisként kezelt áramköri részek bonyolultságát minimálisnak tartani. Ez azért van így, mert a nemlineárisként kezelt rész mindenképpen nagyobb mintavételi frekvencián kell futtatni azért, hogy a nemlinearitás miatt keletkező felharmonikusok ne lapolódjanak át frekvenciatartományban és a modell jobban közelítse az áramkör valódi működését. A parametrikus szűrőt is érdemes túlmintavételezve futtatni, mert a bilineáris transzformáció és változatai a 0 Hz és ∞ Hz közötti analóg frekvenciatartománybeli viselkedést a 0 Hz és a mintavételi frekvencia $\frac{1}{2}$ -szerese közé transzformálja, ezáltal torzítja a szűrő nagyfrekvenciás viselkedését. Magasabb mintavételi frekvencia esetében ez a hatás jelentősen csökkenthető számunkra hasznos frekvenciatartományban.

A másik indok, ami amellet szól, hogy a nemlineárisként kezelt részt minimális méretűre válasszuk az az, hogy a számítás során minden egyes bemeneti mintára több gyökkereső algoritmus-iterációt kell lefuttatni az implicit egyenletek megoldására, ami sokkal több erőforrást igényel, mint a lineáris módszerrel leírt részeken végzett számítások. A minimális méretű nemlineáris részrendszer két részre osztható: a túlvezérelt műveleti erősítőt tartalmazó részáramkörre és az elektroncsöveket tartalmazó részáramkörre.

4.3.1 A túlvezérelt műveleti erősítő és környezete

A túlvezérelt műveleti erősítő és környezete a 18. ábrán látható. Az áramkör két dinamikus komponens tartalmaz. A részáramkör egy paraméterrel rendelkezik. Ez a paraméter a *torzítás mértéke*, melynek szimbóluma d . Értékét a $[0;1]$ intervallumon belül veheti fel. Az alábbi egyenletek alapján az áramkör mindkét pólusa negatív és valós, a pólusok és a zérus frekvenciája a következő:

$$A = -\frac{(R_5 \cdot d) \times \left(\frac{1}{s \cdot C_5}\right)}{R_4 + \frac{1}{s \cdot C_4}} = \frac{s \cdot R_5 \cdot d \cdot C_4}{(s \cdot R_5 \cdot d \cdot C_5 + 1) \cdot (s \cdot R_4 \cdot C_4 + 1)} \quad 48$$

A rendszer pólusai:

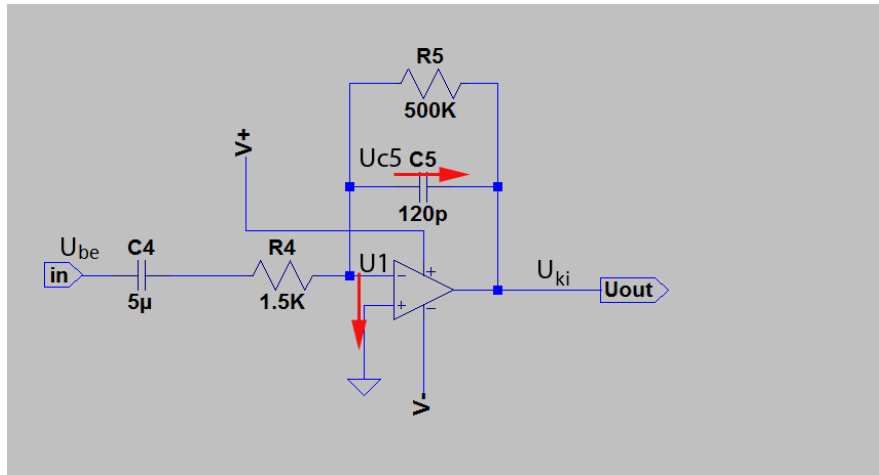
$$p_1 = -\frac{1}{R_4 \cdot C_4} \Rightarrow f_1 = 21,22Hz \quad 49$$

$$p_2 = -\frac{1}{d \cdot R_5 \cdot C_5} \Rightarrow f_{2\min} = \frac{1}{d \cdot R_5 \cdot C_5 \cdot 2 \cdot \pi} = 2.6kHz \quad 50$$

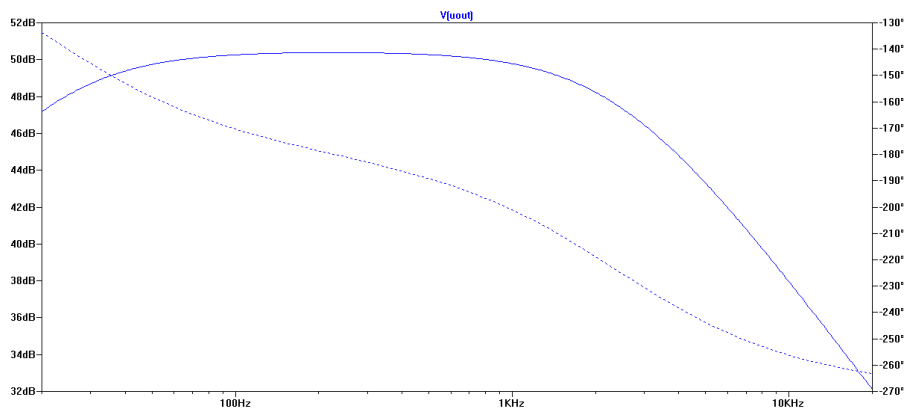
$$f_{2\max} \rightarrow \infty kHz, \text{ ha } d \rightarrow 0$$

A rendszer zérusa:

$$z=0$$



18. ábra: A túlvezérelt műveleti erősítő és környezete

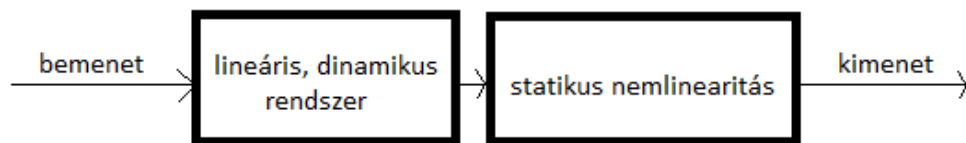


19. ábra: A túlvezérelt műveleti erősítő átviteli karakterisztikája lineáris tartományban

Az egyik pólus 3 dB-es törésponti frekvenciája 21,22 Hz, a zérus 3 dB-es törésponti frekvenciája 0 Hz-en található. Ez a pólus és zérus felelős az egyenáramú komponensek kiszűréséért. A modell bemenetére mindig audio jel fog jutni, ami nem tartalmaz egyenáramú komponens, ezért ez a pólus és zérus elhagyható, tehát az áramkör egyszerűsíthető. A pólus-zérus pár elhagyása elhanyagolható mértékben változtatja meg a középfrekvenciás átvitel amplitúdóját. (A *Matlab* és a *SPICE* modellek erősítésének pontos egyezése rendkívül fontos. Az erősítés eltérése esetén a túlvezérelt műveleti erősítő más időpontban korlátozza be a kimenő jelet, ezáltal különböző lesz a keletkező négyszögösített jel kitöltési tényezője, ezáltal változik a felharmonikus tartalma is.)

Az új állapotváltozós leírásban így csak egy állapotváltozó szerepel. A részrendszer nemlinearitását az okozza, hogy a műveleti erősítő nem képes a pozitív tápfeszültségnél magasabb feszültséget, a negatív tápfeszültségnél alacsonyabb feszültséget kiadni a kimenetén, emiatt a műveleti erősítőnek telítéses a

karakterisztikája. A telítéses karakterisztika miatt a műveleti erősítő bemenetén jelen lévő feszültség elvileg nem tekinthető nullának, mert telítésben a műveleti erősítő erősítése drasztikusan lecsökken. Ennek hatása gyakorlatilag nem észlelhető a részrendszer kimeneti jelén valós gitárjellel több paraméter állásban szimulálva, ezért elhanyagoltam. A kapcsolásban olyan műveleti erősítő szerepel, melynek terhelhetősége meghaladja a következő fokozat maximális bemeneti áramát, ezért a műveleti erősítő kimeneti ellenállása tekinthető nullának. *Spice* alapú szimuláció alapján még abban az esetben is fennáll ez az állítás, amikor a műveleti erősítő kimenete korlátozva van a tápfeszültség által. *SPICE* alapú szimulációra alapozottan a műveleti erősítő bemenő áramát 0 ampernek tekintettem, és a bemeneti ellenállását végtelennek. A műveleti erősítőnek és környezetének nemlineáris Wiener modelljét használtam, melynek a blokkdiagramja az alábbi ábrán látható.



20. ábra: A túlvezérelt műveleti erősítő nemlineáris Wiener modellje

A lineáris, dinamikus részrendszert leíró egyenletek a következők:

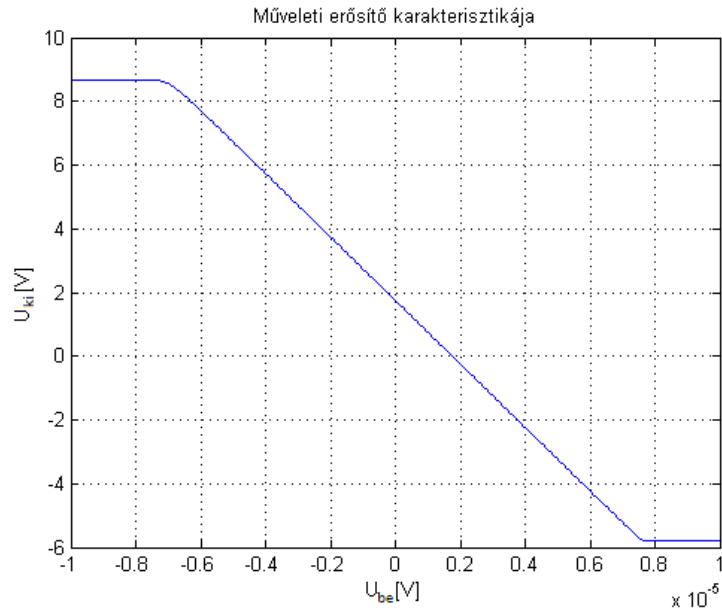
$$U'_{c5} = -\left(\frac{1}{R_5 \cdot C_5 \cdot d}\right) \cdot U_{c5} + U_{be} \quad 51$$

$$U_{ki_id} = \frac{1}{R_4 \cdot C_5} \cdot U_{c5} \quad 52$$

A nemlineáris karakterisztikát leíró egyenletek:

$$-V_p \leq U_{ki_id} \leq -V_n \quad 53$$

$$U_{ki} = -U_{ki_id} \quad 54$$



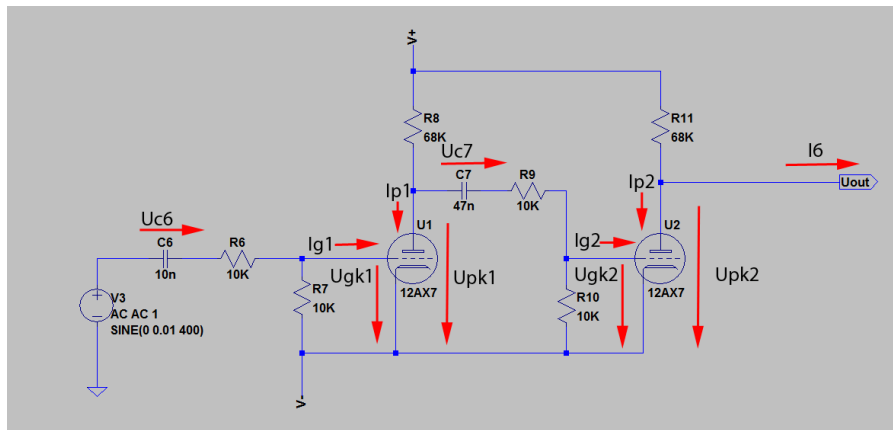
21. ábra: A túlvezérelt műveleti erősítő Wiener modelljében lévő statikus nemlinearitás karakterisztikája a SPICE program alapján

A részrendszer diszkrét modelljét ezekre az egyenletekre és karakterisztikára alapoztam, mely az 5.1-es fejezetben található.

4.3.2 A csöveket tartalmazó részáramkör

Ebben a fejezetben a csöves torzítópédál legfontosabb részéről lesz szó, ami azt a hangzást és felharmonikustartalmat alakítja ki, ami a leginkább jellemző ezekre az áramkörökre.

Az elektroncsöveket tartalmazó részáramkör kapcsolási rajza:



22. ábra: Az elektroncsöveket tartalmazó áramköri rész kapcsolási rajza

Az ábrán a kiválasztott állapotváltozókat és az I_6 segédváltozót piros nyíl jelöli. Az állapotváltozók két csoportra oszthatók: a rendszer dinamikáját leíró állapotváltozók (\mathbf{X}), melyek ebben az áramkörben a kondenzátorok feszültségei, és a nem-lineáris állapotváltozók (\mathbf{W}), melyek a nemlineáris elektroncsövek rácscatód és anódcatód feszültségei. A V_n feszültség az áramkör negatív tápfeszültsége.

$$\mathbf{U} = \begin{bmatrix} U_2 \\ I_6 \end{bmatrix} \quad \mathbf{X} = \begin{bmatrix} U_{c6} \\ U_{c7} \end{bmatrix} \quad \mathbf{W} = \begin{bmatrix} U_{gk1} \\ U_{gk2} \\ U_{pk1} \\ U_{pk2} \end{bmatrix}$$

$$Y = V_n + U_{pk2} \quad 55$$

Az áramkör bemenetei az U_2 feszültség és a következő áramköri rész által felvett áram: I_6 . Az áramkör kimenete a második elektroncső anódfeszültsége: $V_n + U_{pk2}$.

Kirchhoff első törvényét a fenti áramkör csomópontjaira felírva az alábbi egyenletrendszer adódik:

$$0 = -I_{g1} + \frac{U_2 - U_{c6} - U_{gk1} - V_n}{R_6} - \frac{U_{gk1}}{R_7} \quad 56$$

$$0 = -I_{p1} + \frac{V_p - V_n - U_{pk1}}{R_8} + \frac{U_{gk2} - U_{pk1} + U_{c7}}{R_9} \quad 57$$

$$0 = -I_{g2} + \frac{U_{pk1} - U_{c7} - U_{gk2}}{R_9} - \frac{U_{gk2}}{R_{10}} \quad 58$$

$$0 = -I_{p2} + \frac{V_p - V_n - U_{pk2}}{R_{11}} - I_6 (V_n + U_{pk2}) \quad 59$$

Az áramkör dinamikáját a következő egyenletek írják le a kondenzátorok feszültségének deriváltjára vonatkozó összefüggés alapján.

$$U'_{c6} = \frac{I_{c6}}{C_6} = \frac{U_2 - U_{c6} - V_n - U_{gk1}}{R_6 \cdot C_6} \quad 60$$

$$U'_{c7} = \frac{I_{c7}}{C_7} = \frac{U_{pk1} - U_{c7} - U_{gk2}}{R_9 \cdot C_7} \quad 61$$

$$\begin{bmatrix} U'_{c6} \\ U'_{c7} \end{bmatrix} = \mathbf{A}_{\text{TubeC}} \cdot \begin{bmatrix} U_{c6} \\ U_{c7} \end{bmatrix} + \mathbf{B}_{\text{TubeC}} \cdot \begin{bmatrix} U_{gk1} \\ U_{gk2} \\ U_{pk1} \\ U_{pk2} \\ U_2 \\ V_n \end{bmatrix} \quad 62$$

$$Y = V_n + U_{pk2} \quad 63$$

Az $\mathbf{A}_{\text{TubeC}}$ és $\mathbf{B}_{\text{TubeC}}$ mátrixokat a 62. egyenlet definiálja az 60. és 61. egyenlet alapján. Az I_{g1} , I_{g2} , I_{p1} , I_{p2} változók az elektroncsövek anód- és katódáramai, melyek az elektroncsövek vezérlőfeszültségeitől függenek. Az f_1 és f_2 függvényeket a 11. ábrán és 12. ábrán látható táblázatok írják le.

$$I_{g1} = f_1(U_{gk1}) \quad I_{p1} = f_2(U_{gk1}, U_{pk1}) \quad 64$$

$$I_{g2} = f_1(U_{gk2}) \quad I_{p2} = f_2(U_{gk2}, U_{pk2}) \quad 65$$

5 Diszkrét idejű modell

A diszkrétizálás alapvető módszereit a 3.2-es fejezetben ismertettem. Ez a fejezet arról szól, hogy munkám során hogyan alkalmaztam ezeket a módszereket, és melyik módszer milyen eredményt adott.

Az elsődleges szempont a rendszer stabilitása volt. Mivel nemlineáris rendszerről van szó, ezért ez egyáltalán nem triviális feladat. A rendszer nemlinearitása a bejövő jel jelszintjének korlátozásából áll (*hard clipping*, *soft clipping*), ezért feltételeztem, hogy a rendszer stabil marad, ha a stabil folytonos idejű modellt megfelelően diszkrétizálom, és a numerikus algoritmusokat megfelelően alkalmazom. A diszkrétizált rendszer stabilitásáról úgy győződtem meg, hogy ahol lehetett leellenőriztem, hogy a lineáris részrendszer stabil legyen a pólusainak helyzete alapján. Ezután teszteltem a modult olyan bemeneti jelekkel, melyek működés során nagy valószínűséggel előfordulnak. A stabilitási kritérium sérülhet azért, mert a folytonos idejű rendszer rosszul lett diszkrétizálva, vagy azért, mert a nemlineáris részrendszert számító gyökkereső iteráció divergens, és nagymértékű oszcilláció lép fel. A stabilitás megtartása érdekében a bilineáris transzformációt használtam, ahol lehetőség volt rá, mert a bilineáris transzformáció stabil folytonos idejű rendszer esetén stabil diszkrét idejű rendszert eredményez. Ez a választás sajnos azt vonta magával, hogy a bilineárisan transzformált, nemlineáris egyenletrendszereknek nincs explicit megoldása, ezért iteratív gyökkereső algoritmusokat kellett használnom, például a Newton-Raphson iterációs eljárást. Ez a módszer sajnos nagyon megnöveli a számítási igényt. Ez a tény más diszkrétizálási módszerek tesztelésére adott okot, például az előrelépő Euler módszer és magasabb fokú, explicit Runge-Kutta módszerek tesztelésére.

A másik két szempont a modell hasonlósága az eredeti áramkörhöz és az erőforrás igény volt. A feladat az optimális megoldás megtalálása volt.

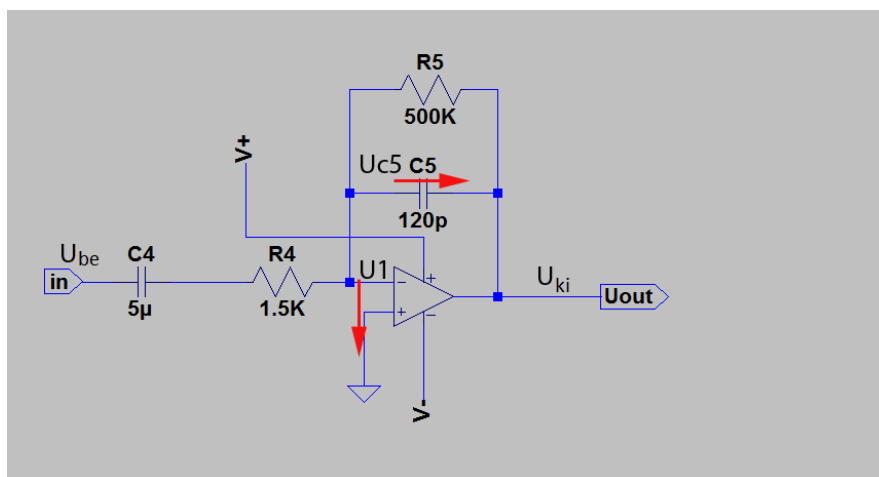
A modell stabilitásának szempontjából rendkívül fontos paraméter a rendszer mintavételi frekvenciája, mert ettől függ, hogy előrelépő Euler módszer esetében a folytonos idejű, stabil pólusok stabilak maradnak-e diszkrét időben is. A bilineáris transzformációval diszkrétizált modellek esetében is lényeges a mintavételi frekvencia, mert a bilineáris transzformáció a 0 Hz és ∞ A számítógép hangkártyája általában 44.1

kHz-es mintavételi frekvenciát használ a be- és kimenetein. Ezt a mintavételi frekvenciát, vagy többszörösét kell használni ahhoz, hogy ne kelljen minden egyes mintát interpolálni a meglévő mintákból.

5.1 A túlvezérelt műveleti erősítő és környezete

A diszkrét idejű modell az 4.3.1-es fejezetben megtalálható folytonos idejű modellre épül.

A diszkrétizálás során bilineáris transzformációt használtam, mert a dinamikus rendszer pólusa paraméterfüggő, és előfordul olyan paraméter érték, hogy előrelépő Euler módszerrel való diszkrétizáció esetén a rendszer nem stabil a 2x44.1 kHz-es mintavételi frekvenciával működő VST plugin esetében. A diszkrét rendszer frekvenciamenete jól közelítette a folytonos idejű rendszer frekvenciamenétét *Matlabban* 4x44.1 kHz-es és VST pluginként 2x44.1 kHz-es mintavételi frekvencia alkalmazásával.



23. ábra: A túlvezérelt műveleti erősítő és környezete

A következő algoritmus szerint működik a modell:

$$A = -\frac{1}{R_5 \cdot d \cdot C_5} \quad B = 1 \quad 66$$

$$C = \frac{1}{R_4 \cdot C_5} \quad D = 0 \quad 67$$

$$A_d = \frac{1 + \frac{\Delta T}{2} \cdot A}{1 - \frac{\Delta T}{2} \cdot A} \quad B_d = \frac{\Delta T \cdot B}{1 - \frac{\Delta T}{2} \cdot A} \quad 68$$

$$U_{c5} = A_d \cdot U_{c5} + B_d \cdot U_{be} \quad 69$$

$$U_{ki_id} = C \cdot U_{c5} \quad 70$$

$$-V_p \leq U_{ki_id} \leq -V_n \quad 71$$

$$U_{ki} = -U_{ki_id} \quad 72$$

U_{be} : rendszer bemeneti értéke

U_{c5} : a fokozat kondenzátorának feszültsége

U_{ki} : a fokozat kimeneti feszültsége

U_{ki_id} : a fokozat kimeneti feszültségének segédváltozója

ΔT : mintavételi frekvencia reciproka

A, B, C, D: folytonos idejű állapotváltozós leírás mátrixai

A_d, B_d, C, D : diszkrét idejű állapotváltozós leírás mátrixai

5.2 Hangszínszabályzó

Ebben az alkalmazásban a legfontosabb a rendszer stabilitása mellett, hogy frekvenciatartományban a lehető legpontosabban kövesse a diszkrét idejű modell a folytonos idejű modellt. A stabilitás megtartására a bilineáris transzformáció a legalkalmasabb, ezért először ezt a módszert alkalmaztam. A hangszínszabályzó diszkrét idejű, parametrikus állapotváltozós leírását az 4.2.2-es fejezetben meghatározott folytonos idejű állapotváltozós leírásra alapoztam.

A bilineárisan diszkrétizált állapotváltozós leírás mátrixait az alábbi módon lehet kiszámítani a 3.2.4-es fejezet alapján:

$$\mathbf{A}_{dEQ} = \frac{\mathbf{I} + \frac{T}{2} \cdot \mathbf{A}_{EQ}}{\mathbf{I} - \frac{T}{2} \cdot \mathbf{A}_{EQ}} \quad \mathbf{B}_{dEQ} = \frac{T \cdot \mathbf{B}_{EQ}}{\mathbf{I} - \frac{T}{2} \cdot \mathbf{A}_{EQ}} \quad 73$$

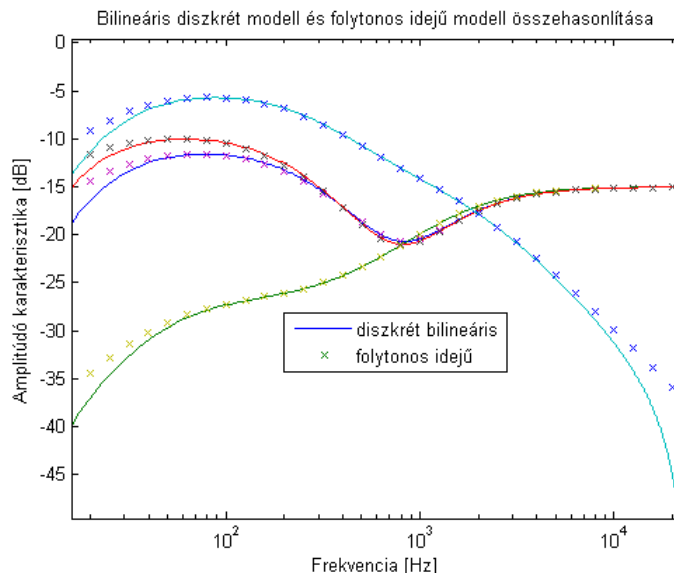
$$\mathbf{C}_{dEQ} = \mathbf{C}_{EQ} \quad \mathbf{D}_{dEQ} = \mathbf{D}_{EQ} \quad 74$$

A diszkrét rendszer állapotváltozós leírásának mátrixai segítségével az alábbi egyenletek adják a modell válaszát:

$$\mathbf{x}[n+1] = \mathbf{A}_d \cdot \mathbf{x}[n] + \mathbf{B}_d \cdot \left(\frac{u[n] + u[n+1]}{2} \right) \quad 75$$

A transzformációt elvégezve b és t paramétereiktől függő mátrixokhoz jutottam. Az állapotváltozós leírás mátrixait csak paraméter változás esetén szükséges újraszámolni. A bilineárisan transzformált mátrixok számítása a fenti alakban túl sok erőforrást igényel. Emiatt ez a módszer csak úgy használható valós idejű alkalmazásban, ha a folytonos idejű mátrixokba először behelyettesítjük a két paraméter értékét, majd ezt követően végezzük el a bilineáris transzformációt. Ez a módszer is több erőforrást igényelt a megengedettnél, ezért végül az előrelép Euler módszert alkalmaztam.

A bilineárisan diszkrétizált állapotváltozós leírás és a folytonos idejű modell átviteli függvényének összehasonlítása a következő ábrán látható:



24. ábra: A hangszínszabályzó bilineárisan diszkrétizált modelljének és folytonos idejű modelljének összehasonlítása frekvenciatartományban

A *Matlabban* szimulált diszkrétizált rendszer működése kellő mértékben közelíti a folytonos idejű rendszer működését.

A C++ nyelven létrehozott modell diszkrétizálása során az előrelépő Euler módszert alkalmaztam végül azért, hogy paraméter változtatása esetén se lassuljon le a működés. A módszerről a 3.2.1–es fejezet szól.

Az előrelépő Euler módszer egyenlete a következő:

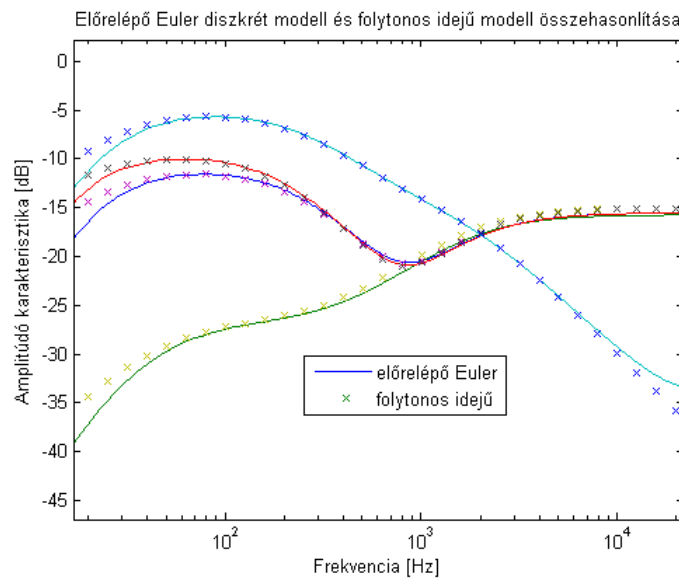
$$y[n + 1] - y[n] = T \cdot x[n] \quad 76$$

y : állapotváltozó értéke

$x[n]$: állapotváltozó idő szerinti deriváltja n -edik mintához tartozó időpillanatban

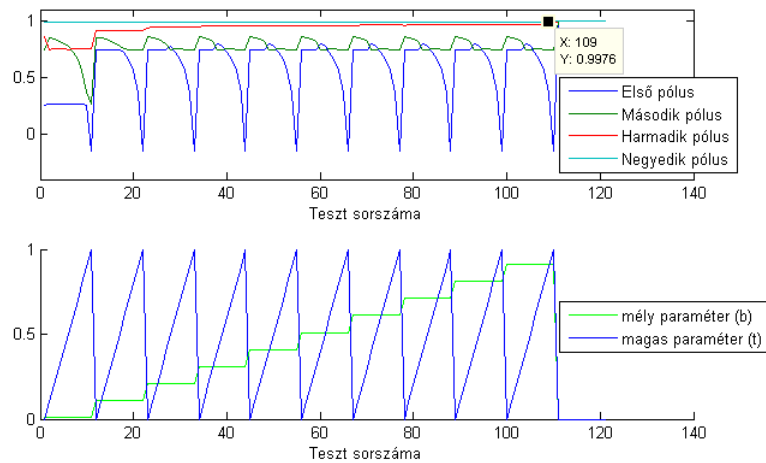
T : mintavételi frekvencia reciproka

A módszer eredményét összehasonlítottam a folytonos idejű modell eredményével frekvenciatartományban:

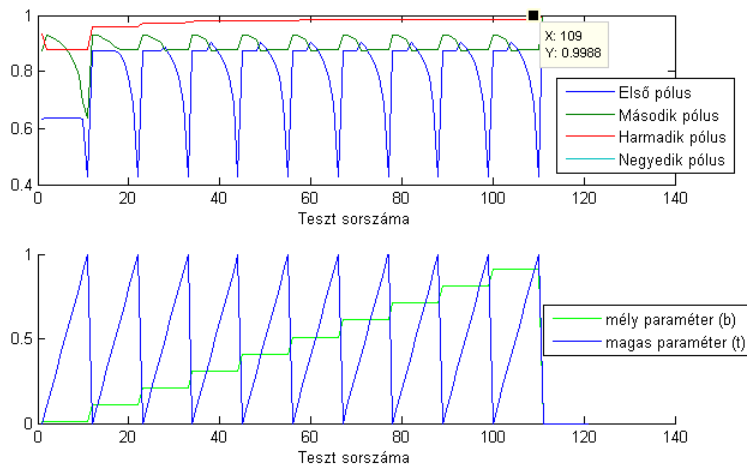


25. ábra: A hangszínszabályzó előrelépő Euler (elsőfokú Runge-Kutta) módszerrel diszkrétizált modelljének és folytonos idejű modelljének összehasonlítása frekvenciatartományban

Az előrelépő Euler módszerrel diszkretizált modell stabilitása a 3.2.1-es fejezet alapján:



26. ábra: A hangszínszabályzó diszkrét pólusainak változása a paraméterek függvényében 44100 Hz-es mintavételi frekvencia esetén előrelépő Euler diszkretizáció mellett



27. ábra: A hangszínszabályzó diszkrét pólusainak változása a paraméterek függvényében 2x44100 Hz-es mintavételi frekvencia esetén előrelépő Euler diszkretizáció mellett

Az előrelépő Euler módszerrel diszkretizált modell 44.1 kHz-es és 2x44.1 kHz-es mintavételi frekvencián is stabil maradt. A 26. ábrán és 27. ábrán szereplő pólusok valós értékűek. A számításuk a 3. egyenlet alapján történt a folytonos idejű pólusokból.

A bilineárisan transzformált állapotváltozós leírás jobban követte a *SPICE* programmal szimulált áramkör működését nagyfrekvenciás jelekre olyan paraméter beállítások esetén, amikor a *magas* címkéjű potenciométer nagy értékűre volt állítva, mégis az előrelépő Euler módszerrel diszkretizált modell került bele a végleges VST

verzióba, mert más esetben paraméterváltozás esetén az új állapotmátrixok számítása sok számítást igényel, veszélyezteti a valós idejű működést.

5.3 Cső körüli részáramkör

Az elektroncsövek körül lévő áramköri rész diszkretizálási módja nagy hatással van az egész diszkrét modell erőforrásigényére. Ettől a diszkretizálási módtól az is függ, hogy érdemes-e az 4.3.2-es fejezetben leírt I_6 áram elhanyagolását megtenni a számítási igény csökkenésének érdekében.

Az első diszkretizálási módok, melyeket kipróbáltam explicit sémán alapulnak. Ez azt jelenti, hogy nem vezetnek implicit egyenletrendszerekhez, melyeket nagy erőforrásigényű gyökkereső algoritmusok alkalmazásával lehet csak megoldani nemlineáris rendszerek esetében. A letesztelt explicit diszkretizálási módok: előrelépő Euler módszer, másodfokú, explicit, középértéken alapuló Runge-Kutta módszer.

Később egy implicit sémán alapuló diszkretizálási módot is kipróbáltam: a bilineáris transzformációt. A bilineáris transzformáció esetében két modellt építettem fel attól függően, hogy elhanyagoltam-e a hangszínszabályzó és a csöves részáramkör közötti visszacsatolást az erőforrásigény csökkentése érdekében. Az explicit modellek esetében szimuláltam a visszacsatolást az előző mintához tartozó áram értékkel. Ez egy minta nagyságú késleltetést tesz a visszacsatolt rendszerbe.

5.3.1 Explicit módszerek

5.3.1.1 Előrelépő Euler módszer

A módszer részletes leírása a 3.2.1 fejezetben olvasható. A módszer előnye, hogy ebben az esetben minimális az implicit, nemlineáris egyenleteket megoldó gyökkereső iterációban jelen lévő egyenletek száma, és a Jacobi-mátrix nagysága. Ez jelentősen csökkenti a felhasznált számítási kapacitást, viszont nem garantált a diszkrét modell stabilitása.

Az 4.3.2-es fejezet alapján a gyökkereső iterációban jelen lévő megoldandó egyenletek:

$$0 = -I_{g1} + \frac{U_2 - U_{c6} - U_{gk1} - V_n - U_{gk1}}{R_6} - \frac{U_{gk1}}{R_7} \quad 77$$

$$0 = -I_{p1} + \frac{V_p - V_n - U_{pk1}}{R_8} + \frac{U_{gk2} - U_{pk1} + U_{c7}}{R_9} \quad 78$$

$$0 = -I_{g2} + \frac{U_{pk1} - U_{c7} - U_{gk2}}{R_9} - \frac{U_{gk2}}{R_{10}} \quad 79$$

$$0 = -I_{p2} + \frac{V_p - V_n - U_{pk2}}{R_{11}} - I_6 \quad 80$$

A Jacobi-mátrix:

$$Jacobi = \begin{bmatrix} \frac{\partial I_{g1}}{\partial U_{gk1}} - \frac{1}{R_6} - \frac{1}{R_7} & 0 & 0 & 0 \\ -\frac{\partial I_{p1}}{\partial U_{gk1}} & \frac{1}{R_9} & -\frac{\partial I_{p1}}{\partial U_{pk1}} - \frac{1}{R_8} - \frac{1}{R_9} & 0 \\ 0 & -\frac{\partial I_{g2}}{\partial U_{gk2}} - \frac{1}{R_9} - \frac{1}{R_{10}} & \frac{1}{R_9} & 0 \\ 0 & -\frac{\partial I_{p2}}{\partial U_{gk2}} & 0 & -\frac{\partial I_{p2}}{\partial U_{pk2}} - \frac{1}{R_{11}} - (D_{EQZ}) \end{bmatrix} \quad 81$$

A Jacobi-mátrix legutolsó elemében szerepel a D_{EQZ} változó. Ez a változó írja le az I_6 áram függését a második trióda katód-anód feszültségétől, ezért szerepel a Jacobi-mátrix negyedik sorának negyedik elemében.

A rendszer dinamikáját leíró állapotváltozók számítása:

Az alábbi egyenletek az 4.2.2 és az 4.3.2 fejezetben lévő egyenleteken alapulnak.

Az állapotváltozók változását és a rendszer kimenetét az alábbi egyenletek adják meg:

$$\begin{bmatrix} \Delta U_{c6} \\ \Delta U_{c7} \end{bmatrix} = \Delta T \cdot \left(\mathbf{A}_{TubeC} \cdot \begin{bmatrix} U_{c6} \\ U_{c7} \end{bmatrix} + \mathbf{B}_{TubeC} \cdot U_{dynTube} \right) \quad 82$$

$$\begin{bmatrix} \Delta U_{c1} \\ \Delta U_{c2} \\ \Delta U_{c3} \\ \Delta U_{c4} \end{bmatrix} = \Delta T \cdot \left(\mathbf{A}_{EQ} \cdot \begin{bmatrix} U_{c1} \\ U_{c2} \\ U_{c3} \\ U_{c4} \end{bmatrix} + \mathbf{B}_{EQ} \cdot (V_n + U_{pk2}) \right) \quad 83$$

$$Y = \mathbf{C}_{EQ}^T \cdot \begin{bmatrix} U_{c1} \\ U_{c2} \\ U_{c3} \\ U_{c4} \end{bmatrix} + \mathbf{D}_{EQ} \cdot (V_n + U_{pk2})$$

ΔT : időlépték, mintavételi frekvencia reciproka

A modell létrehozását tesztek sorozata követte. A kérdés az, hogy mekkora mintavételi frekvencia szükséges ahhoz, hogy a modell stabil legyen, és megfelelően kövesse az eredeti áramkör működését, és eközben mennyi erőforrást igényel az algoritmus.

Ez a modell 2x44.100 kHz-es mintavételi frekvencia mellett is stabilnak bizonyult, ezért ezen a modellen alapuló VST plugin lett a végleges a viszonylag kisebb erőforrásbeli igénye miatt.

5.3.1.2 Másodfokú Runge-Kutta módszer

A módszer részletes leírása a 3.2.5 fejezetben olvasható. A módszer előnye az előző módszerhez hasonlóan, hogy ebben az esetben minimális az implicit, nemlineáris egyenleteket megoldó gyökkereső iterációban jelen lévő egyenletek száma, és a Jakobi mátrix nagysága. Ez jelentősen csökkenti a felhasznált számítási kapacitást, viszont itt sem garantált a diszkrét modell stabilitása. Az előző módszerhez képest a másodfokú Runge-Kutta módszerrel diszkrétizált modell elvileg jobban konvergál a megoldáshoz, viszont közel kétszer akkora a szükséges számítási kapacitás.

Az algoritmus:

Az itt használt másodfokú, explicit Runge-Kutta módszer két lépésben becsli az állapotváltozók idő szerinti deriváltját az aktuális időbeli lépésben. Az időtartomány középső időpontbeli deriváltját becsli az állapotváltozók első időpontbeli értékeivel az adott időtartományra vonatkozóan. Az így számolt középső időpontbeli differenciákat használja az algoritmus az egész időtartományra.

A gyökkereső iterációs ciklus nem változott az előrelépő Euler módszert alkalmazó modellhez képest.

Az 4.3.2-es fejezet alapján a gyökkereső iterációban jelen lévő megoldandó egyenletek:

$$0 = -I_{g1} + \frac{U_2 - U_{c6} - U_{gk1} - V_n - U_{gk1}}{R_6} - \frac{U_{gk1}}{R_7} \quad 85$$

$$0 = -I_{p1} + \frac{V_p - V_n - U_{pk1}}{R_8} + \frac{U_{gk2} - U_{pk1} + U_{c7}}{R_9} \quad 86$$

$$0 = -I_{g2} + \frac{U_{pk1} - U_{c7} - U_{gk2}}{R_9} - \frac{U_{gk2}}{R_{10}} \quad 87$$

$$0 = -I_{p2} + \frac{V_p - V_n - U_{pk2}}{R_{11}} - I_6 \quad 88$$

A Jacobi-mátrix:

$$Jacobi = \begin{bmatrix} \frac{\partial I_{g1}}{\partial U_{gk1}} - \frac{1}{R_6} - \frac{1}{R_7} & 0 & 0 & 0 \\ -\frac{\partial I_{p1}}{\partial U_{gk1}} & \frac{1}{R_9} & -\frac{\partial I_{p1}}{\partial U_{pk1}} - \frac{1}{R_8} - \frac{1}{R_9} & 0 \\ 0 & -\frac{\partial I_{g2}}{\partial U_{gk2}} - \frac{1}{R_9} - \frac{1}{R_{10}} & \frac{1}{R_9} & 0 \\ 0 & -\frac{\partial I_{p2}}{\partial U_{gk2}} & 0 & -\frac{\partial I_{p2}}{\partial U_{pk2}} - \frac{1}{R_{11}} - (D_{EQZ}) \end{bmatrix} \quad 89$$

A rendszer dinamikáját leíró állapotváltozók számolása különböző a két lépésben. Az alábbi egyenletek az 4.2.2 és az 4.3.2 fejezetben lévő egyenleteken alapulnak.

Első lépésben a középső időpontbeli nemlineáris állapotváltozó értékek számolása történik a gyökkereső iterációs ciklus lefutása után. Második lépésben a rendszer dinamikáját leíró állapotváltozók számítása történik a középső időpontra vonatkozóan:

$$\begin{bmatrix} U_{c6mid} \\ U_{c7mid} \end{bmatrix} = \begin{bmatrix} U_{c6} \\ U_{c7} \end{bmatrix} + \frac{\Delta T}{2} \cdot \left(\mathbf{A}_{TubeC} \cdot \begin{bmatrix} U_{c6} \\ U_{c7} \end{bmatrix} + \mathbf{B}_{TubeC} \cdot U_{dynTube} \right) \quad 90$$

$$\begin{bmatrix} U_{c1mid} \\ U_{c2mid} \\ U_{c3mid} \\ U_{c4mid} \end{bmatrix} = \begin{bmatrix} U_{c1} \\ U_{c2} \\ U_{c3} \\ U_{c4} \end{bmatrix} + \frac{\Delta T}{2} \cdot \left(\mathbf{A}_{EQ} \cdot \begin{bmatrix} U_{c1} \\ U_{c2} \\ U_{c3} \\ U_{c4} \end{bmatrix} + \mathbf{B}_{EQ} \cdot (V_n + U_{pk2}) \right) \quad 91$$

Harmadik lépésben a középső időpontbeli értékeken alapuló gyökkereső iterációs ciklus fut le. Negyedik lépésben a kiszámolt középső időpontbeli állapotváltozó értékeken alapuló differencia számítása történik:

$$\begin{bmatrix} \Delta U_{c6} \\ \Delta U_{c7} \end{bmatrix} = \Delta T \cdot \left(\mathbf{A}_{TubeC} \cdot \begin{bmatrix} U_{c6mid} \\ U_{c7mid} \end{bmatrix} + \mathbf{B}_{TubeC} \cdot U_{dynTube} \right) \quad 92$$

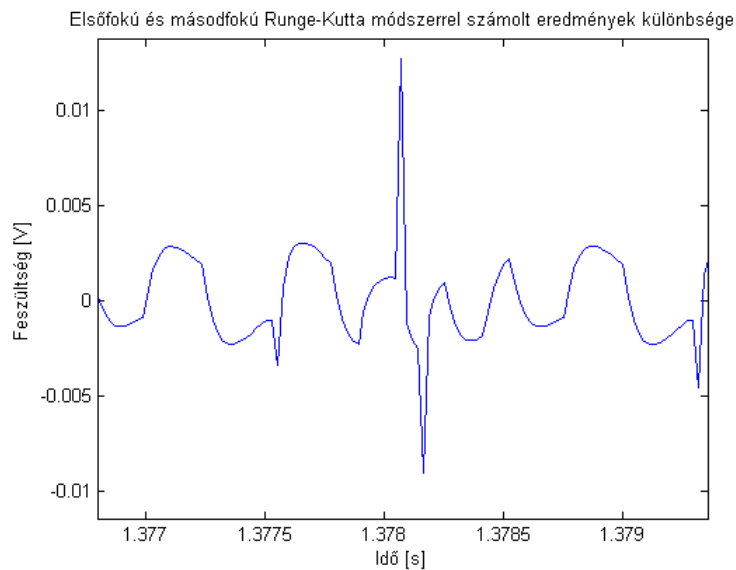
$$\begin{bmatrix} \Delta U_{c1} \\ \Delta U_{c2} \\ \Delta U_{c3} \\ \Delta U_{c4} \end{bmatrix} = \Delta T \cdot \left(\mathbf{A}_{EQ} \cdot \begin{bmatrix} U_{c1mid} \\ U_{c2mid} \\ U_{c3mid} \\ U_{c4mid} \end{bmatrix} + \mathbf{B}_{EQ} \cdot (V_n + U_{pk2}) \right) \quad 93$$

ΔT : időlépték, mintavételi frekvencia reciproka

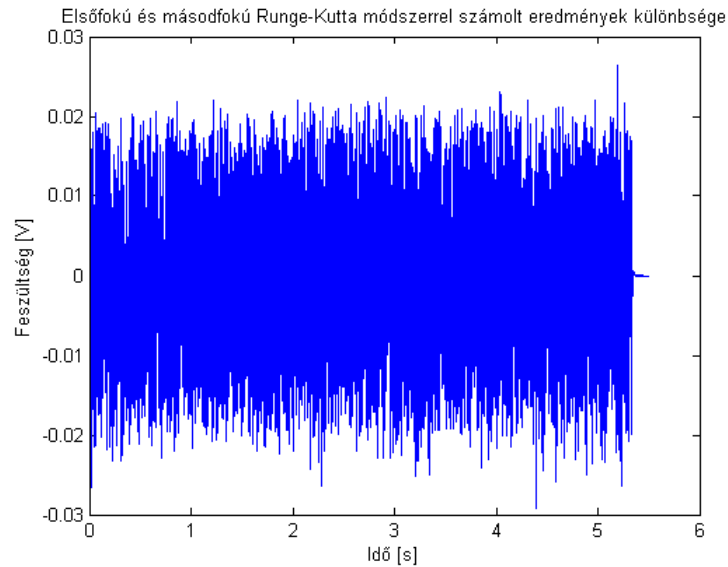
A rendszer kimenete az alábbi egyenlettel számítható:

$$Y = \mathbf{C}_{EQ}^T \cdot \begin{bmatrix} U_{c1} \\ U_{c2} \\ U_{c3} \\ U_{c4} \end{bmatrix} + \mathbf{D}_{EQ} \cdot (V_n + U_{pk2}) \quad 94$$

A modell létrehozását tesztek sorozata követte. A teszt célja kideríteni, hogy vajon javul-e a nagyfrekvenciás működés az előző modellhez képest, és ez a javulás megéri-e a többlet erőforrás igényt.



28. ábra: Különbség az előrelépő Euler és a másodfokú Runge-Kutta módszerek eredményei között



29. ábra: Különbség az előrelépő Euler és a másodfokú Runge-Kutta módszerek eredményei között

A teszt során alkalmazott gerjesztőjel és a hibajel teljesítményének hányadosa 1%, a különbség elhanyagolható. A másodfokú modell által nyújtott hangminőség nem volt jobb az elsőfokú explicit modellhez képest, ezért a másodfokú explicit modell elvetettem.

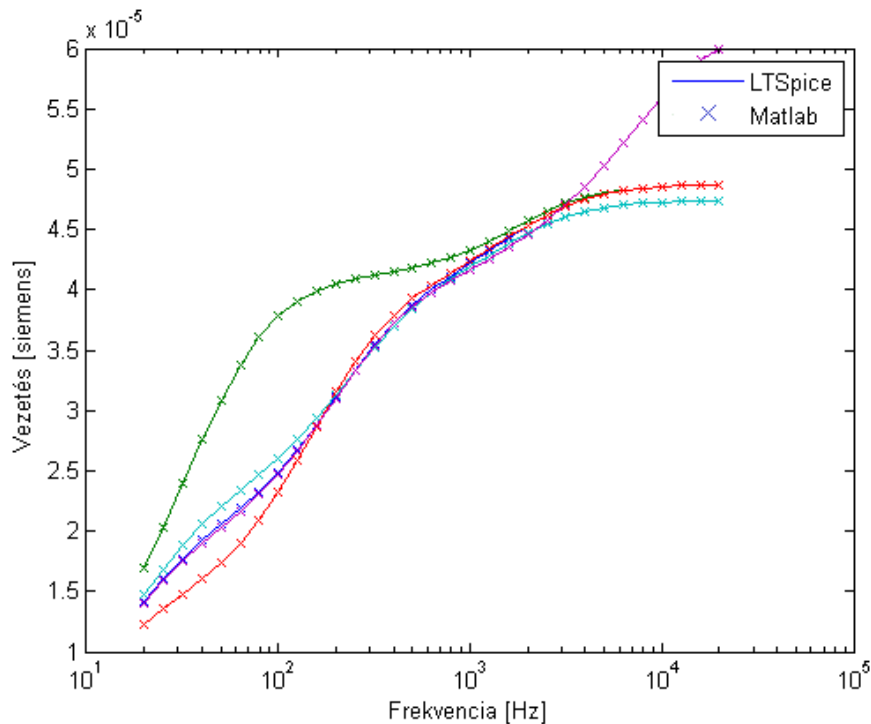
5.3.2 Implicit módszer

Implicit módszerek közül a bilineáris diszkrétizálási módszert alkalmaztam, mert stabil folytonos idejű rendszert stabil diszkrét idejű rendszerré alakít. A módszerről bővebben a 3.2.4 fejezetben olvasható.

A bilineáris transzformáció alkalmazásának előnye, hogy az így képzett diszkrét modell stabilis marad, ha a gyökkereső iterációs ciklus konvergens. Ennek azért van nagy jelentősége, mert a modell paraméteres, ezért nehéz garantálni, hogy az explicit módon diszkrétizált modell stabil legyen minden bemeneti értékre minden paraméter érték esetén. A módszer hátránya, hogy az algoritmusnak nincs explicit alakja nemlineáris rendszerek esetében. A nemlineáris egyenletek megoldása mellett még a rendszer dinamikus viselkedésének számítása is bekerül a gyökkereső iterációs ciklusba, ami jelentősen megnöveli a számításokra fordított időt. Ez a növekedés olyan mértékű, hogy valós időben nem fut a C++ nyelven megvalósított modell, ezért felépítettem egy olyan implicit modellt is, melyben a csöveket tartalmazó részáramkör és a parametrikus hangszínszabályzó közötti visszacsatolást elhanyagoltam.

5.3.2.1 A visszacsatolás elhanyagolásának hatása

A hangszínszabályzó modul, mint terhelés, a csöveket tartalmazó áramkör kimenetén:

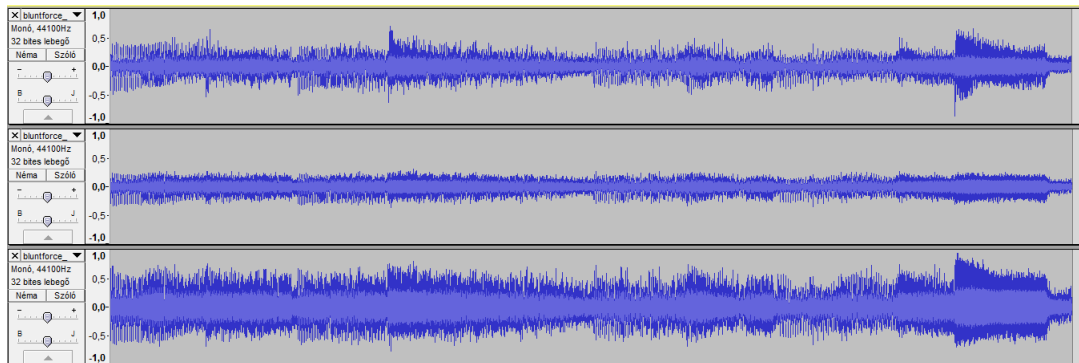


30. ábra: A hangszínszabályzó bemeneti áramának és bemeneti feszültségének hányadosa frekvenciatartományban a mély és magas paraméterek függvényében

Ez az ábra egy *LTSpice* szimuláció és a *Matlabban* kiszámolt hangszínszabályzó-impedancia modell összehasonlításának eredménye. Az 4.3.2-es fejezetben ezt a terhelő áramot az I_6 változó adta meg. A teszt során beállított paraméterértékek az 1. táblázatban találhatóak.

Az elhanyagolás hatása nehezen meghatározható. A visszacsatolt áram hatására más feszültség áll be az elektroncsöves fokozat kimenetén, mint a tökéletesen csatolt esetben, ami hatást gyakorol a hangszínszabályzóra. Végül beáll egy egyensúly a két részrendszer állapotváltozói között. A visszacsatolásban elhanyagolt áram nagyságát könnyű meghatározni, de az elhanyagolás nélküli esetben a visszacsatolt áram hatását a hangszínszabályzó kimenetére nem lehet explicit módon kiszámolni, csak konkrét esetek tesztelésével lehet vizsgálni.

Megvizsgáltam a különbséget valódi gitárjel gerjesztés esetén. A beállított paraméterek: magas: 0.5, mély: 1, hangerősség:1. A tesztet az *LTSPICE* programmal végeztem oly módon, hogy a modell bemenetére kapcsolt feszültséggenerátor egy *wav* formátumú fájlból vette a gerjesztőjeleket, majd egy *wav* formátumú fájlba történt kimenő jel regisztrálása.



31. ábra: Teszt valódi gitár jellel (fent: bemenő jel, közepen: tökéletesen csatolt, lent: csatolatlan)

A felvett hangfájlok a mellékletben megtalálhatóak. Az elhanyagolás érzékelhető változásokat okoz a modellben, viszont jelentős erőforrásbeli megtakarítás érhető el vele a tökéletesen csatolt modellhez képest. Az erről a modelltől készült VST plugin valós időben hibamentesen fut.

5.3.2.2 A csatolás nélküli implicit eset diszkrét idejű leírása

Ebben a modellben a csöveket tartalmazó részáramkör és a hangszínszabályzó áramkör két külön modulként kezelhető. A hangszínszabályzó modul parametrikus, lineáris áramkör, ezért egy parametrikus negyedfokú digitális szűrővel megvalósítható a csöveket tartalmazó modul kimenetére csatlakoztatva.

A rendszer állapotváltozói:

$$\begin{bmatrix} U_{gk1} \\ U_{gk2} \\ U_{pk1} \\ U_{pk2} \\ U_{c6} \\ U_{c7} \end{bmatrix}$$

A rendszert leíró egyenletek az 4.3.2-es fejezet alapján:

$$0 = -I_{g1} + \frac{U_2 - U_{c6} - U_{gk1} - V_n}{R_6} - \frac{U_{gk1}}{R_7} \quad 95$$

$$0 = -I_{p1} + \frac{V_p - V_n - U_{pk1}}{R_8} + \frac{U_{gk2} - U_{pk1} + U_{c7}}{R_9} \quad 96$$

$$0 = -I_{g2} + \frac{U_{pk1} - U_{c7} - U_{gk2}}{R_9} - \frac{U_{gk2}}{R_{10}} \quad 97$$

$$0 = -I_{p2} + \frac{V_p - V_n - U_{pk2}}{R_{11}} \quad 98$$

$$0 = \begin{bmatrix} -U_{c6it} \\ -U_{c7it} \end{bmatrix} + \mathbf{A}_{\text{TubeD}} \cdot \begin{bmatrix} U_{c6} \\ U_{c7} \end{bmatrix} + \mathbf{B}_{\text{TubeD}} \cdot \begin{bmatrix} U_{gk1} \\ U_{gk2} \\ U_{pk1} \\ U_{pk2} \\ U_2 \\ V_n \end{bmatrix} + \begin{bmatrix} U_{gk1pr} \\ U_{gk2pr} \\ U_{pk1pr} \\ U_{pk2pr} \\ U_{2pr} \\ V_n \end{bmatrix} \quad 99$$

A gyökkereső iterációs ciklus a fenti egyenleteket kielégítő állapotváltozó értékeket keresi meg. A modul kimenetét az alábbi egyenlet adja meg:

$$Y = V_n + U_{pk2} \quad 100$$

A változók, melyek pr indexet kaptak, az előző időpillanatbeli értéket jelentik. Az it indexszel rendelkező dinamikus állapotváltozóhoz kötődő változók azok az értékek, melyeket a gyökkereső iteráció éppen keres.

A modellnek a következő egyenletet kell megoldania:

$$\mathbf{Jacobi} \cdot \Delta \mathbf{X} = -\mathbf{f}(\mathbf{x}) \quad 101$$

Jacobi: Az egyenletrendszernek és ismeretleneinek Jacobi-mártixa

$\Delta\mathbf{X}$: állapotváltozók változtatása az adott iterációs ciklusban a gyökkereső algoritmuson belül

$f(\mathbf{x})$: a modellt leíró tíz egyenlet

A fenti egyenletrendszer megoldását a Jacobi-mátrix inverzének használatával végeztem. A módszer leírása a 3.4.1-es fejezetben található.

A Jacobi-mátrix sok nulla elemet tartalmaz. Valószínűleg megérné áttérni az LU felbontáson alapuló egyenlet megoldási módszerre, de a modul számítási igénye megengedi, hogy ezt ne tegyem meg. A 5.3.2.3-as fejezetben leírt modul Jacobi-mátrixa 10x10-es nagyságú, ezért abban az esetben már a Jacobi-mátrix LU felbontásával oldottam meg a fenti egyenletet.

A modell létrehozását követően tesztekkel ellenőriztem a működést.

5.3.2.3 Tökéletesen csatolt implicit eset diszkrét idejű leírása

Gyökkereső iteráció egyenletrendszere

Az 4.2.2-es és 4.3.2-es fejezetek alapján az egyenletrendszer állapotváltozói a következők:

$$\begin{bmatrix} U_{gk1} \\ U_{gk2} \\ U_{pk1} \\ U_{pk2} \\ U_{c6} \\ U_{c7} \\ U_{c1} \\ U_{c2} \\ U_{c3} \\ U_{c4} \end{bmatrix}$$

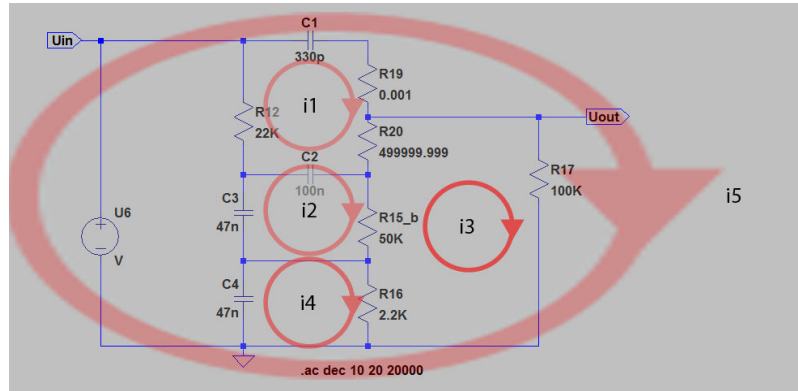
A tökéletesen csatolt esetben meg kellett határozni, hogy a hangszínszabályzó mekkora árammal terheli a csöveket tartalmazó áramkörü rész kimenetét az állapotváltozóinak és a bemenetének állapotától függően. Ez az áram az alábbi egyenletrendszerrel számolható ki:

$$\begin{bmatrix} U'_{c1} \\ U'_{c2} \\ U'_{c3} \\ U'_{c4} \end{bmatrix} = \mathbf{A}_{EQ} \cdot \begin{bmatrix} U_{c1} \\ U_{c2} \\ U_{c3} \\ U_{c4} \end{bmatrix} + \mathbf{B}_{EQ} \cdot U_{beEQ} \quad 102$$

$$I_6 = \mathbf{C}_{EQG}^T \cdot \begin{bmatrix} U_{c1} \\ U_{c2} \\ U_{c3} \\ U_{c4} \end{bmatrix} + \mathbf{D}_{EQG} \cdot U_{beEQ} \quad 103$$

A felvett hurokáramok alapján a \mathbf{C}_{EQG} és \mathbf{D}_{EQG} változók az alábbi képlettel írhatók le:

$$I_6 = i_5 \quad 104$$



32. ábra: A hangszínszabályzó áramkörben felvett hurokáramok

A \mathbf{C}_{EQG} és \mathbf{D}_{EQG} mátrixok kiszámolásához az előző egyenlet és az 16. ábrán látható egyenletrendszer megoldása szükséges.

A tökéletesen csatolt megoldáshoz vezető egyenletrendszer:

$$0 = -I_{g1} + \frac{U_2 - U_{c6it} - U_{gk1} - V_n}{R_6} - \frac{U_{gk1}}{R_7} \quad 105$$

$$0 = -I_{p1} + \frac{V_p - V_n - U_{pk1}}{R_8} + \frac{U_{gk2} - U_{pk1} + U_{c7it}}{R_9} \quad 106$$

$$0 = -I_{g2} + \frac{U_{pk1} - U_{c7it} - U_{gk2}}{R_9} - \frac{U_{gk2}}{R_{10}} \quad 107$$

$$0 = -I_{p2} + \frac{V_p - V_n - U_{pk2}}{R_{11}} - I_6 \quad 108$$

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} -U_{c6it} \\ -U_{c7it} \end{bmatrix} + \mathbf{A}_{\text{TubeD}} \cdot \begin{bmatrix} U_{c6} \\ U_{c7} \end{bmatrix} + \mathbf{B}_{\text{TubeD}} \cdot \left(\begin{bmatrix} U_{gk1} \\ U_{gk2} \\ U_{pk1} \\ U_{pk2} \\ U_2 \\ V_n \end{bmatrix} + \begin{bmatrix} U_{gk1pr} \\ U_{gk2pr} \\ U_{pk1pr} \\ U_{pk2pr} \\ U_{2pr} \\ V_n \end{bmatrix} \right) \quad 109$$

$$I_6 = \mathbf{C}_{EQG}^T \cdot \begin{bmatrix} U_{c1it} \\ U_{c2it} \\ U_{c3it} \\ U_{c4it} \end{bmatrix} + \mathbf{D}_{EQG} \cdot (V_n + U_{pk2}) \quad 110$$

111

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = - \left(\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \frac{T_tube}{2} \cdot \mathbf{A}_{EQ} \right) \cdot \begin{bmatrix} U_{c1it} \\ U_{c2it} \\ U_{c3it} \\ U_{c4it} \end{bmatrix} + \left(\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} + \frac{T_tube}{2} \cdot \mathbf{A}_{EQ} \right) \cdot \begin{bmatrix} U_{c1} \\ U_{c2} \\ U_{c3} \\ U_{c4} \end{bmatrix} + \frac{T_tube}{2} \cdot \mathbf{B}_{EQ} \cdot (2 \cdot V_n + U_{pk2} + U_{pk2pr})$$

A gyökkereső iterációs ciklus a fenti egyenleteket kielégítő állapotváltozó értékeket keresi meg. Az egyenletrendszerben lévő I_6 változó csupán segédváltozó, nem állapotváltozó. A modul kimenetét az alábbi egyenlet adja meg:

$$Y = \mathbf{C}_{EQ}^T \cdot \begin{bmatrix} U_{c1} \\ U_{c2} \\ U_{c3} \\ U_{c4} \end{bmatrix} + \mathbf{D}_{EQ} \cdot (V_n + U_{pk2}) \quad 112$$

A változók, melyek pr indexet kaptak, az előző időpillanatbeli értéket jelentik. Az it indexszel rendelkező dinamikus állapotváltozóhoz kötődő változók azok az értékek, melyeket a gyökkereső iteráció éppen keres.

A rendszer Jacobi-mátrixa:

$$Jacobi_A = \begin{bmatrix} -\frac{\partial I_{g1}}{\partial U_{gk1}} - \frac{1}{R_6} - \frac{1}{R_7} & 0 & 0 & 0 & -\frac{1}{R_6} & 0 \\ -\frac{\partial I_{p1}}{\partial U_{gk1}} & \frac{1}{R_9} & -\frac{\partial I_{p1}}{\partial U_{pk1}} - \frac{1}{R_8} - \frac{1}{R_9} & 0 & 0 & \frac{1}{R_9} \\ 0 & -\frac{\partial I_{g2}}{\partial U_{gk2}} - \frac{1}{R_9} - \frac{1}{R_{10}} & \frac{1}{R_9} & 0 & 0 & -\frac{1}{R_9} \\ 0 & -\frac{\partial I_{p2}}{\partial U_{gk2}} & 0 & -\frac{\partial I_{p2}}{\partial U_{pk2}} - \frac{1}{R_{11}} - (DSolMatZ) & 0 & 0 \\ BTubeD(1,1) & 0 & 0 & 0 & -1 & 0 \\ 0 & BTubeD(2,2) & BTubeD(2,3) & 0 & 0 & -1 \end{bmatrix}$$

$$Jacobi_B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ C_{EQZ}^T \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad 114$$

$$Jacobi_C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{T_tube}{2} \cdot B_{EQ} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad 115$$

$$Jacobi_D = \begin{bmatrix} \left(\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \frac{T_tube}{2} \cdot A_{EQ} \right) \end{bmatrix} \quad 116$$

$$Jacobi = \begin{bmatrix} Jacobi_A & Jacobi_B \\ Jacobi_C & Jacobi_D \end{bmatrix} \quad 117$$

A modellnek a következő egyenletet kell megoldania a 3.3.2-es fejezetben leírt gyökkereső algoritmussal:

$$\mathbf{Jacobi} \cdot \Delta \mathbf{X} = -\mathbf{f}(\mathbf{x}) \quad 118$$

Jacobi: az egyenletrendszernek és ismeretleneinek Jacobi-mátrixa

$\Delta \mathbf{X}$: állapotváltozók változtatása az adott iterációs ciklusban a gyökkereső algoritmuson belül

$\mathbf{f}(\mathbf{x})$: a modellt leíró tíz egyenlet

A fenti egyenletrendszert nem hatékony a Jacobi-mátrix inverzének segítségével megoldani. A Jacobi-mátrix LU felbontását használva sokkal kevesebb erőforrás is elegendő a számítások elvégzésére. A Crout-féle LU felbontás algoritmusát alkalmaztam (lásd 3.4.2-es fejezet).

6 A *Matlab* modell megvalósítása C++-ban VST pluginként

6.1 Mi az a VST

A VST (Virtuális Stúdió Technológia) egy olyan technológia, melyet virtuális hangszerek és effektek létrehozására fejlesztettek ki. Ez a technológia már-már szabvánnyá vált ezen a területen, és a *Steinberg* nevű céghez fűződik.

A VST programok az alábbi két csoportra bonthatóak: az első a VST gazdaprogram, mely egy futtatható állomány. Az egyes modulok összeköttetéseit, paramétereit és az audio rendszer beállításait lehet általa megadni. Ebbe a programba tölthetők be a másik csoportba tartozó VST programok: a bővítmények.

A VST bővítmények (pluginok) olyan különálló könyvtárak, melyek egy-egy effektet, effekt csoportot vagy hangszert valósítanak meg. A moduláris felépítés és az elérhető pluginok nagy száma miatt tetszőleges audio alkalmazás felépíthető a rendszerrel szem előtt tartva a hardveres erőforrásigényeket.

6.2 A Tube Driver VST plugin felépítése

A modult egy *AGain* nevű osztály implementálja, melyet az *AudioEffectX* őszosztályból származtattam. Az *AudioEffectX* osztály a *VST 2.x* verziójához tartozó absztrakt, bővített VST effekt osztály, mely meghatározza, hogy milyen függvényeket kell az effektet megvalósító osztálynak implementálnia. Ezek közül a függvények közül azokra fogok kitérni, melyek a jelfeldolgozás szempontjából fontosak.

6.2.1 Jelfeldolgozás szempontjából fontos függvények

AGain (audioMasterCallback audioMaster):

Az *AGain* osztály konstruktor, melyben meghívom az *AudioEffectX* őszosztály konstruktorát a paraméterek számával és inicializálom az osztály tagváltozóit.

~AGain ():

Az *AGain* osztály destruktora. Mivel a konstruktorban nem hoztam létre dinamikusan allokált változókat, ezért ez a függvény üres.

virtual void processReplacing (float inputs, float** outputs, VstInt32 sampleFrames),**

virtual void processDoubleReplacing (double inputs, double** outputs, VstInt32 sampleFrames)**

A jelfeldolgozást végző függvények. A különbség közöttük a ki és bemeneti értékük típusa. Az *AudioEffectX* őssztály megköveteli, hogy a leszármazott osztály mindkettő függvényt implementálja, de gyakorlatban csak az egyik verzió végzi a valós működést. A *AGain* modulban a *processReplacing()* függvény hajtja végre a jelfeldolgozást.

6.3 *Matlab* kód portolása C++ kódra

A *Matlab* kódot eleve úgy írtam meg, hogy könnyedén át lehessen alakítani C++ kódra. A kétdimenziós lineáris interpolációt végző függvényt és az LU dekompozíciós algoritmust végrehajtó függvényt is ennek a szempontnak megfelelően kódoltam le. A *Matlab* környezetet arra használtam, hogy ellenőrizzem a megtervezett algoritmus helyességét, majd minimális változtatásokkal C++ környezetbe ültessem.

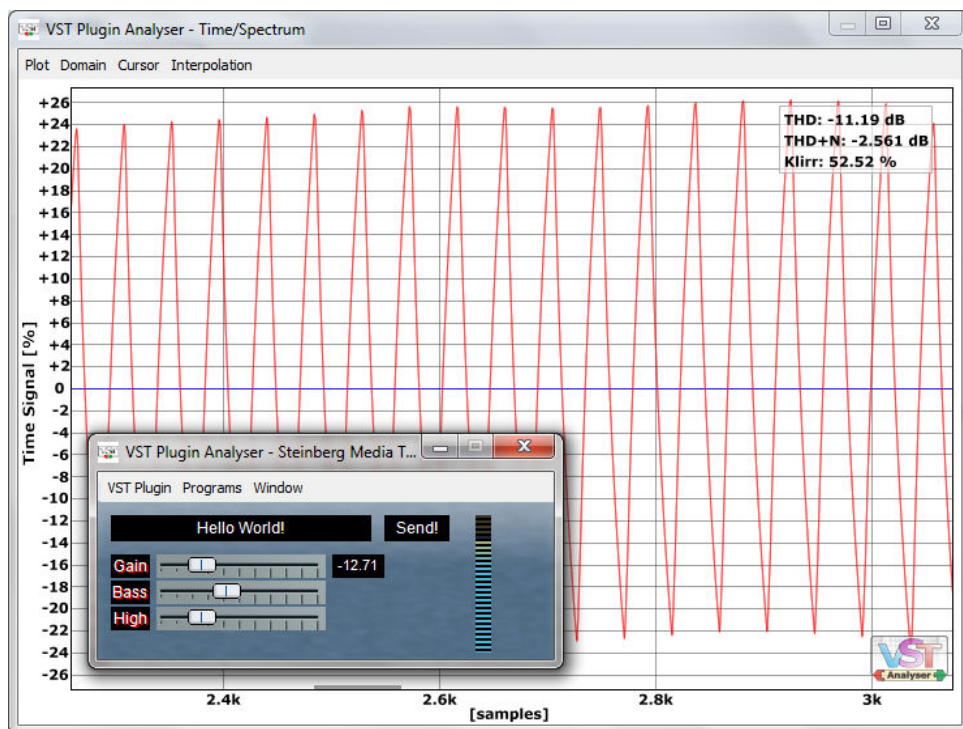
6.4 Fejlesztő eszközök

A fejlesztés során *Microsoft Visual C++ 2010 Express* fejlesztőkörnyezetet használtam. Azért erre a fejlesztőkörnyezetre esett a választás, mert ehhez a rendszerhez találhatóak a legtöbb és legjobban használható teljesítménymérő programok. Kezdetben *Code::Blocks* fejlesztőkörnyezetben fejlesztettem a modult *MinGW* fordítóval, de a

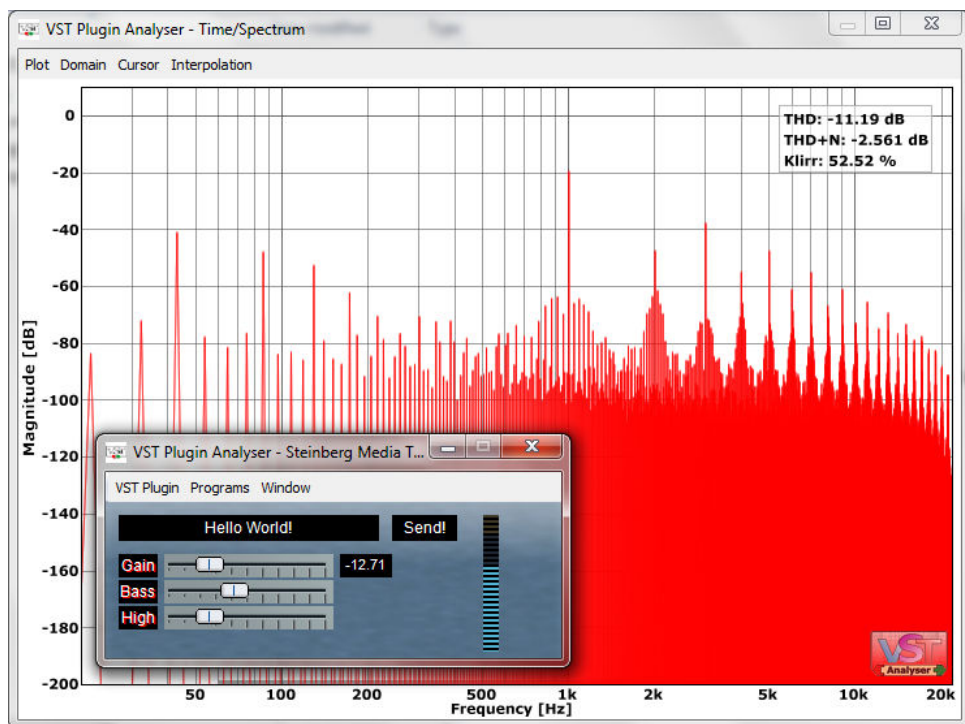
kezdetleges teljesítménymérő programok miatt át kellett térnem az *MSVC* fordítóra és a *Microsoft Visual C++ 2010 Expressre*.

6.5 Tesztelésnél használt eszközök

Az első tesztek a *VST Plugin Analyser* programmal végeztem. Ez egy egyszerű gazda program (*host*), mely betölti a kiválasztott könyvtárat (*library*) és egy 1.0 amplitúdójú szinuszjellel gerjeszti a bemenetén. A paramétereket csúszkákkel lehet állítani. A modul kimenete időtartományban és frekvenciatartományban is ellenőrizhető. Amennyiben pontosabb összehasonlításra volt szükség, vagy a belső változók értékeire voltam kíváncsi, egy fájlba írtam és *Matlabbal* dolgoztam fel a kapott adatokat.



33. ábra: Időtartománybeli analízis

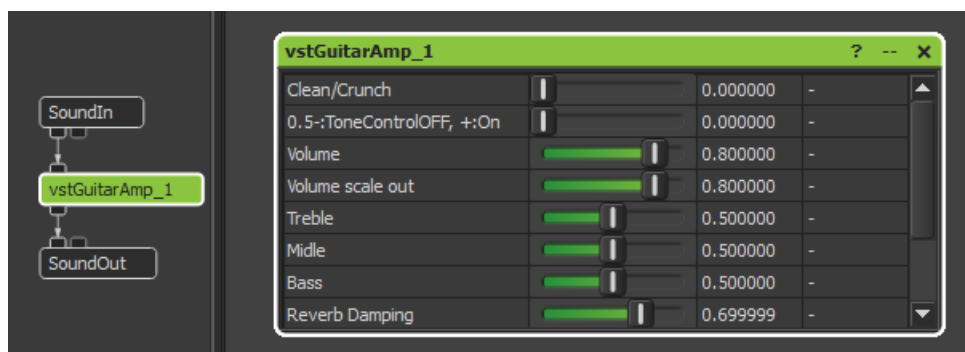


34. ábra: Frekvenciatartománybeli analízis

Amennyiben a program működött, nem adott futás idejű hibát és a jelek is helyesnek tündek, egy valódi VST gazdaprogramban folytatódott a tesztelés.

Erre a feladatra az AudioMulch programot használtam.

Az AudioMulch nagyon könnyen, egyszerűen használható program. A modul működtetéséhez egy jelforrásra és egy jel nyelőre volt szükség a megfelelő kimenetein. A jelforrás általában egy gitár vagy egy hang fájl volt. A modul kimeneti jeleit a hangszóróra irányítottam vagy fájlba mentettem.

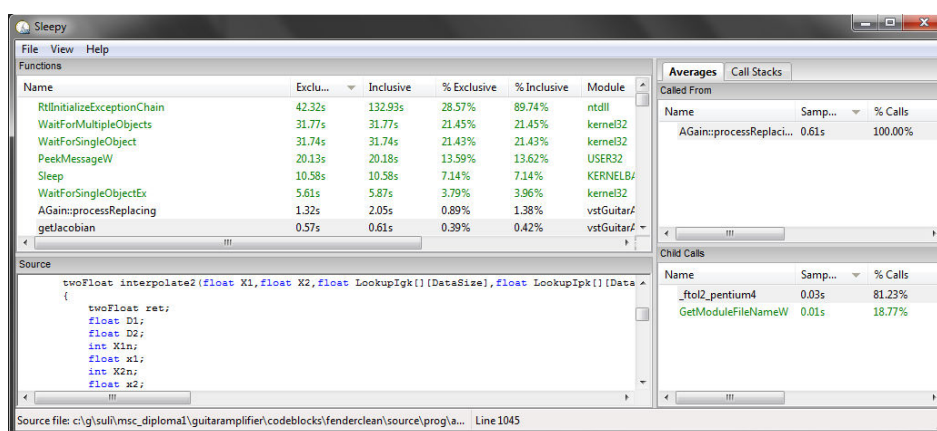


35. ábra: Tesztelés AudioMulch-csal

6.6 Hatékonyság mérése

A kód hatékonysága rendkívül fontos ebben az alkalmazásban. Optimális esetben négyszeres mintavételi frekvencia mellett és mintánként négyszeres gyökkereső iteráció során másodpercenként $4 \times 44100 \times 4 = 705600$ alkalommal fut le a gyökkereső iterációban lévő kód, mely két darab két dimenziós táblázatban végez lineáris interpolációt, majd egyenletrendszert old meg. Ez rendkívül erőforrásigényes feladat, ezért az iterációban lévő számítások számát minimalizálni kellett, a számításokat a leghatékonyabban kellett végrehajtani.

A hatékonyság mérésére a *Very Sleepy* nevű programot használtam.



36. ábra: Modul teljesítményének mérése a Very Sleepy nevű programmal

A program azt méri, hogy mennyi időt töltött a program az egyes utasítások végrehajtásával. Információt ad azzal kapcsolatban, hogy az egyes függvényekben összesen mennyi időt töltött a program és ez hány százaléka a teljes futásidőnek. Ha nincs erősen optimalizálva a C++ kód, soronkénti vizsgálat is lehetséges.

6.7 Hatékonyság javítása

A hatékonyság javítására a következő alapelveket vettem figyelembe:

Adatmozgatás

Nagyobb struktúrák vagy adatok esetében a mutatókkal történő megvalósítás javasolt, ha elkerülhető az adatok másolása.

Függvényhívások

Függvényhívások esetén a nagyobb struktúrákat, mint az előző pontban is említettem, mutatókkal érdemes átadni a függvény argumentumaként.

Memóriához való hozzáférés

A processzor könnyebben fér hozzá egy blokkban lévő memóriaterület sorban következő elemeihez. Bizonyos esetekben érdemes az adatokat módosított sorrendben tárolni a memóriában, ha az algoritmusnak úgy optimálisabb. (Pl. LU felbontás algoritmus)

Az osztás művelet

Az osztás művelet időben rendkívül költséges művelet. Még a lebegőpontos számokkal dolgozó matematikai segédprocesszor segítségével is többszöröse a szorzás műveletének. Ez a különbség processzorfüggő. Intel i3-2310M típusú processzorban 4-5-ször volt lassabb egy osztás művelet, mint egy szorzás művelet. Amennyiben lehetséges, az érték reciproka legyen eltárolva, ha később sok osztás történne az adott számmal.

7 Az áramkör megépítése, mérése

A feladat részét képezi a meghatározott modell és a valódi áramkör összehasonlítása. A SPICE szimuláció kiforrott, jól használható eszköz az áramkörök szimulációjára, viszont az eredmény jelentősen függ a SPICE programban használt csőmodelltől. Az, hogy egy valódi áramkörrel is összehasonlításra került a kifejlesztett modell, ezt a hibalehetőséget kiküszöböli, ugyanakkor biztosít arról, hogy a SPICE programban használt csőmodell megfelelő ebben az alkalmazásban. A munka végső célja a valódi áramkörhöz való hasonlóság, nem pedig a SPICE szimulációhoz.

Ahhoz, hogy a szoftveres modellt össze lehessen hasonlítani az áramkörön mért jelekkel, a szoftveres modellben lévő csövek karakterisztikájának meg kell egyeznie az áramkörben lévő elektroncső karakterisztikájával. A szoftveres modell által használt csőkarakterisztika táblázatot létre lehet hozni csőparaméterek megadásával, és a csőnek adott pontokban kimért karakterisztikájával. A csőparaméterekkel való megadás esetében szintén egy csőmodellt kell alkalmazni, mely a paraméterek alapján kiszámolja a táblázat pontjait, ezért a második módszerrel kívántam megadni az elektroncső táblázatát. Utólag ebből a mérésből a csőparaméterek is meghatározhatóak, és megvizsgálható a csőmodellek pontossága is.

7.1 Az elektroncső karakterisztikájának mérése

A feladat az elektroncső anód- és rácsáramának mérése volt a anód-katód feszültség és a rács-katód feszültség függvényében. A mérés során a cső statikus paramétereinek mérése volt a cél, ezért a csőre adott feszültségek annyi ideig hajtották a vizsgált eszközt, hogy az esetleges tranziensek lecsengjenek. A 12AX7-es típusú elektroncső kivezetései között rendszerint 1 pF-nál nem nagyobb a parazita-kapacitás. A két mérendő mennyiség két statikus kétváltozós nemlineáris egyenlet. Az áramkör tápfeszültsége + 8,72 V és – 5,86 V.

Az áramkörben lévő kondenzátorok miatt a tápfeszültségnél nagyobb feszültségű váltóáram is kialakulhat, de a *Spice* programban történt szimuláció alapján

15 V-nál nagyobb feszültségkülönbség nem alakul ki a csövek elektródái között, ezért a két bemeneti feszültséget az alábbi tartományban állítottam be:

$$U_{pk}: -5V - 15 V$$

$$U_{gk}: -5V - 10 V$$

A mért áramok ebbe a tartományba esnek a csőtípusra vonatkozó átlagos értékek alapján:

$$I_g: 0-3 \text{ mA}$$

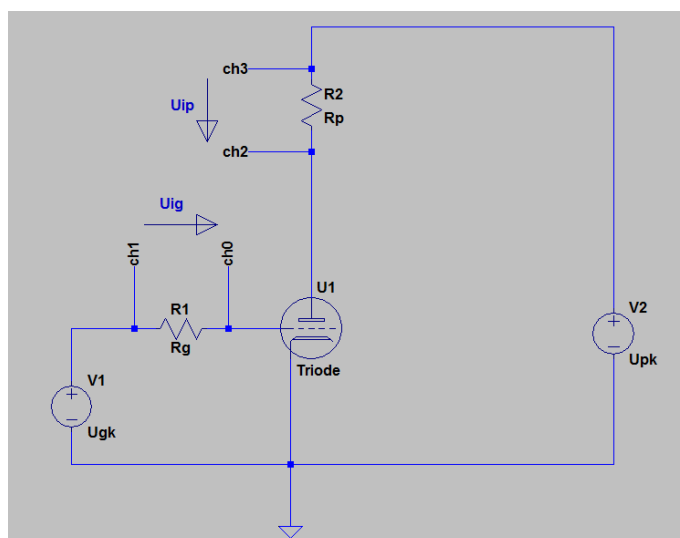
$$I_p: 0-13 \text{ mA}$$

A rác- és anódáramokat leíró táblázat felbontása: 150x150

Egy táblázatbeli érték mérésének időtartama: 10 ms

A mérés bemeneti mintavételi frekvenciája: 8 kHz

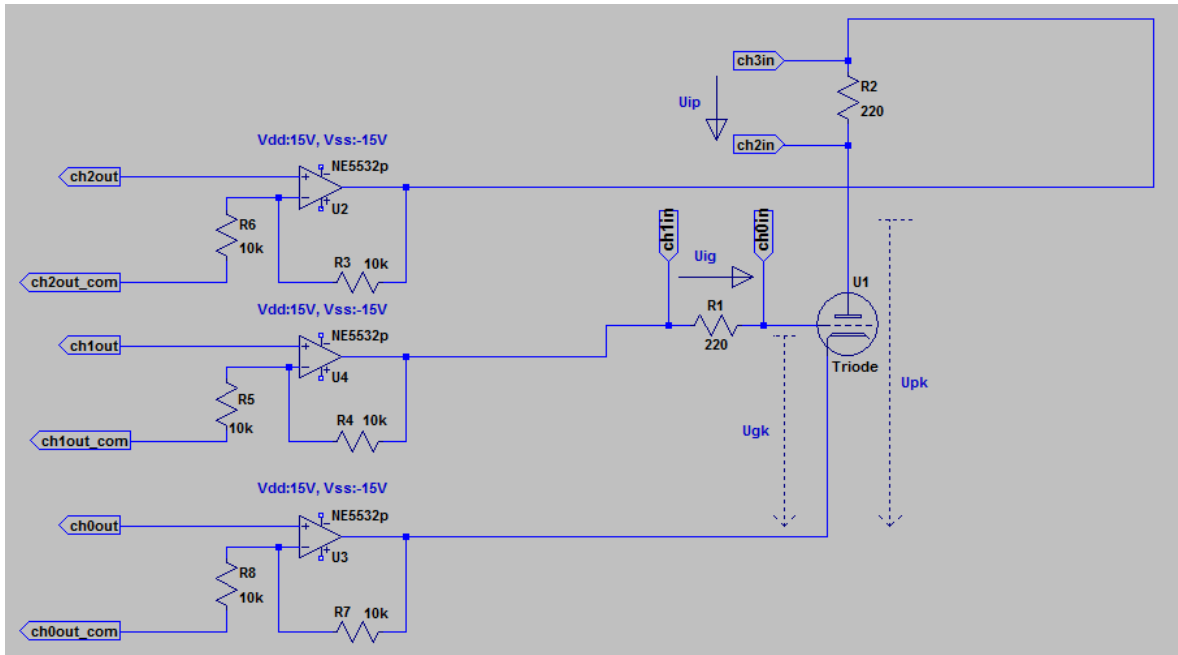
A fenti paraméterek alapján a táblázat egy pontjáról 80 minta áll rendelkezésre. A mérést a DSP laborban végeztem az NI9263 D/A átalakító és az NI9201 A/D átalakító segítségével. A D/A átalakító ± 10 V-os feszültséget képes kiadni 1 mA áramterhelhetőség mellett, ezért puffer erősítőt kellett alkalmaznom. A választás az NE5532p típusú műveleti erősítőre esett, mert 16 mA a maximális áramterhelhetősége és a tápfeszültsége ± 15 V. A D/A átalakító egyszerre négy kimeneten tudta vezérelni az áramkört. Az A/D átalakító 8 csatornán tudott mérni. A mérés elvi vázlatát az alábbi ábrán látható.



37. ábra: A statikus csőkarakterisztika mérésének elvi vázlatát

A mérési csatornák megválasztását az indokolta, hogy a DSP labor elektromágneses zajokkal erősen terhelt. A mérés során az összetartozó csatornákat egymáshoz közel vezettem, ezért közel azonosan hatott rájuk a laborban jelen lévő zavaró elektromos tér, mely főleg az 50 Hz-es hálózati frekvencián és annak felharmonikusain volt jelen. A két csatorna különbsége adja a hasznos jelet, ezért a közös zajt nagymértékben el lehetett nyomni.

A mérés tényleges vázlatja a puffer-erősítővel:



38. ábra: A mérés puffer-erősítővel

7.1.1 A puffer erősítő ellenállásainak megválasztása

Az R_3 , R_4 , R_5 , R_6 ellenállásokat úgy választottam meg, hogy a puffer erősítőknek az erősítése elérje a nekünk szükséges minimális értéket, ne terhelje túl nagy árammal a visszacsatolás a puffer erősítő kimenetét, és ne legyenek túl nagy értékűek az ellenállások, mert a vezetett zaj-áramok által keltett zajfeszültségek nagysága egyenesen arányos az ellenállás nagyságával, ugyanakkor az ellenállások termikus zaja is egyenesen arányos az ellenállás értékének négyzetgyökével.

A szükséges minimális erősítés:

$$\frac{15V}{10V} = 1.5$$

119

A puffererősítő erősítésének egyenlete:

$$A = 1 + \frac{R_2^*}{R_1^*} = 1 + \frac{R_3}{R_6} = 1 + \frac{R_4}{R_5} = 2 \quad 120$$

Ebből adódóan az ellenállásokat 10 k Ω -nak választottam, ekkor a visszacsatolások árama 1 mA. A műveleti erősítő áramterhelhetősége 16 mA, a mért elektroncső maximális áramfelvétele egy csatornán 13 mA. A puffer erősítőket még a legnagyobb várt terhelőáram se fogja túlterhelni.

A műveleti erősítők kimeneti feszültségének felső határa 15 V.

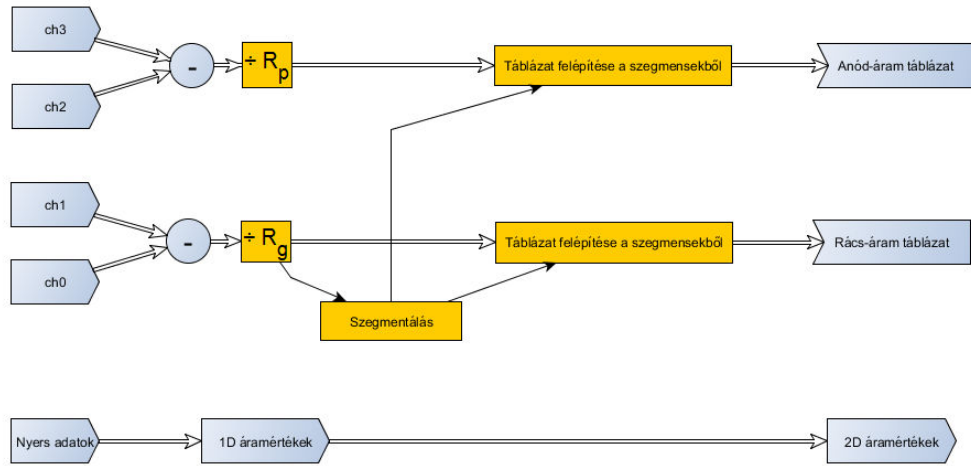
7.1.2 Az árammérő ellenállások megválasztása

Az áram-mérő ellenállások kiválasztásánál az alábbi hatásokat vettem figyelembe: nagyobb ellenállás-érték esetén nagyobb feszültség esik adott áram hatására, ezért a zavarokra kevésbé érzékeny a hasznos jel. Ha nagyobb feszültség esik a mérő-ellenálláson, csökken a mért trióda kivezérelhetőségi tartománya. Az előző pontban felvázolt módon a nagyobb ellenállás-érték negatívan befolyásolja az ellenállás zaját.

A mérő ellenállásokat 220 Ohm értékűnek választottam, ekkor a kivezérelhetőségi tartomány a maximális 13 mA esetén: 0V - 12 V

7.1.3 A mérési eredmény feldolgozása

A LabView projekt TDMS formátumban tárolja az érkező adatokat. A feldolgozás Matlabban történt, az adatok beolvasásához a *TDMS_getStruct* függvényt használtam. A feldolgozó algoritmust a következő blokkvázlat mutatja be:



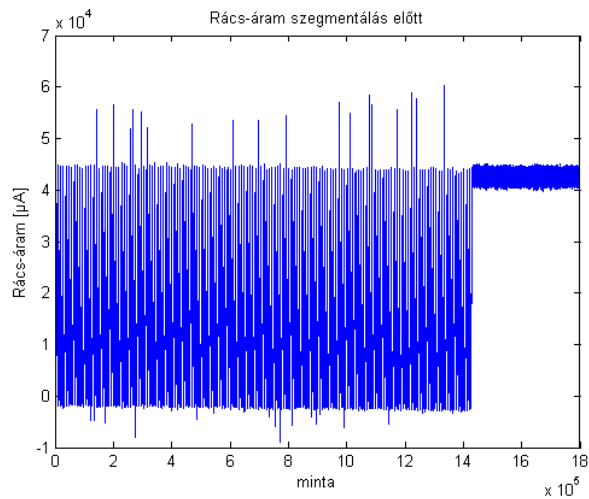
39. ábra: Cső karakterisztika mérés eredményének feldolgozása

Csatorna adatokból az áram adatok az alábbi módon számolhatók:

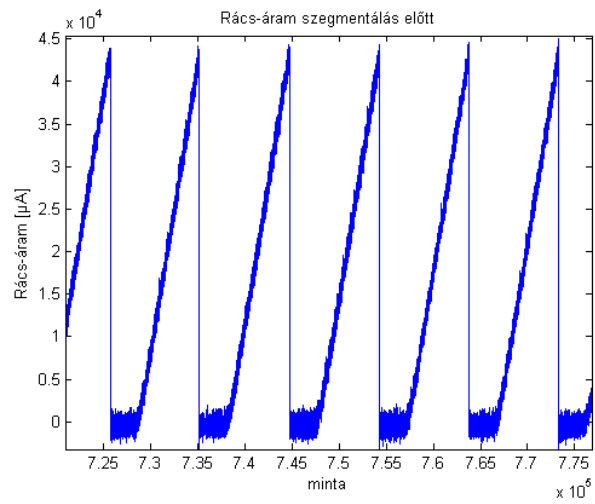
$$I_{g1D} = \frac{(ch_1 - ch_0)}{R_g} \quad 121$$

$$I_{p1D} = \frac{(ch_3 - ch_2)}{R_p} \quad 122$$

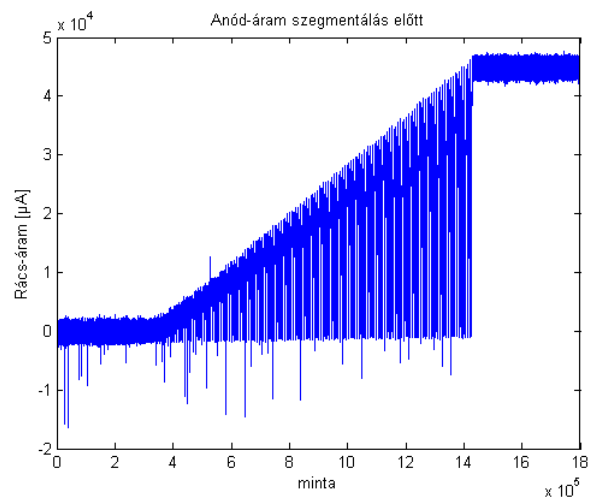
A következő ábrákon a mért rácsáram és anódáram látható. Az áramadatok jellegre megfelelnek a várt adatoknak.



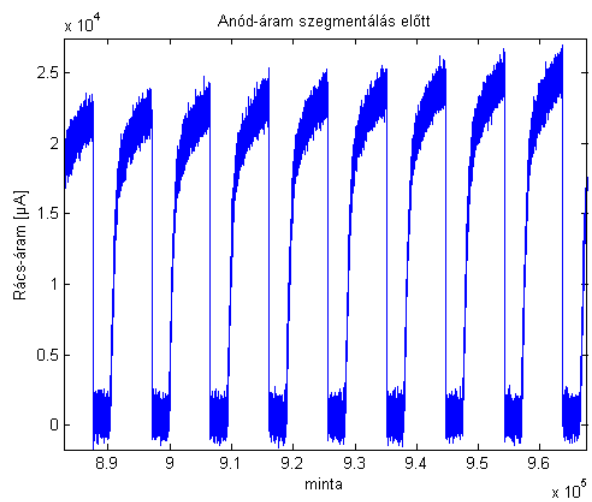
40. ábra: Rácsáram szegmentálás előtt



41. ábra: Rácsáram szegmentálás előtt, nagyítva



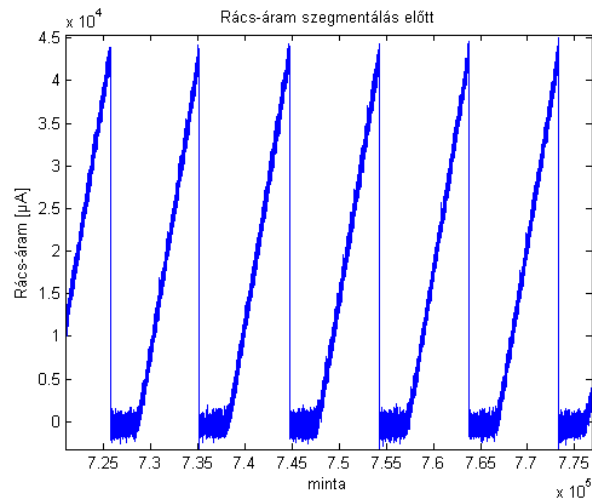
42. ábra: Anódáram szegmentálás előtt



43. ábra: Anódáram szegmentálás előtt, nagyítva

7.1.3.1 Szegmentálás

A rácsáram csak a rács-katód feszültségtől függ. A rács-katód feszültség 150 szegmens alatt végighalad a teljes mérési tartományon, majd a következő anód-katód feszültségérték beállása után megismétli ezt a műveletet. Emiatt a rácsáram 150 szegmensnyi értéke 150-szer ismétlődik a felvett adatokban. A rácsáram a 150 szegmensnyi minta végén egy minta alatt drasztikusan lecsökken. Emiatt a rácsáram felhasználható a szegmensek közötti határok meghatározásához.

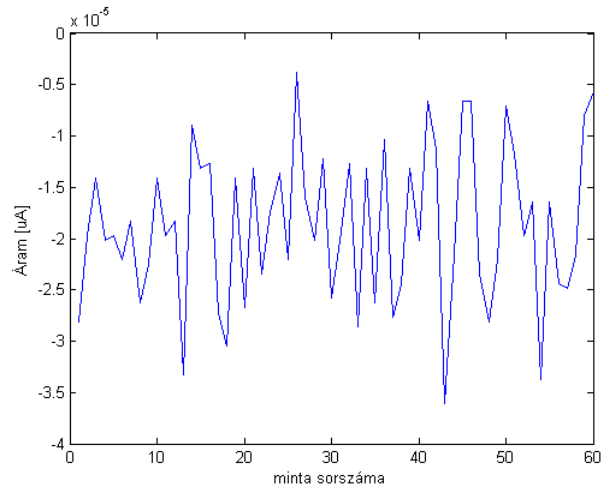


44. ábra: A szegmenshatárok kijelölése

150 x 150 pontban történt a karakterisztika mérése. A rendszer gerjesztése úgy történt, hogy beállítottam a legalacsonyabb értékű anód-katód feszültséget, majd beállítottam a legalacsonyabb értékű rács-katód feszültséget. Ebben az állapotban mértem 80 mintát. Ezután megnöveltem a rács-katód feszültséget a következő, nagyobb értékre és mértem újabb 80 mintát. 150 x 80 minta kimérése után megnöveltem az anód-katód feszültségértéket a következőre. Ezt a folyamatot hajtottam végre, amíg 150 x 150 x 80 mintát regisztráltam.

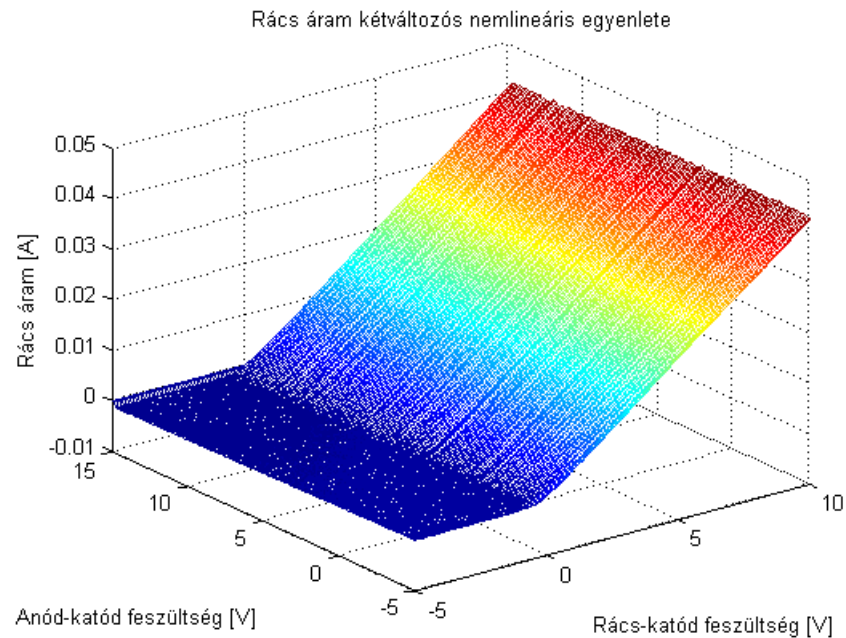
Szegmentálás során el kellett különítenem az egyes rácspontokhoz tartozó méréseket. Az anód-katód feszültség szerinti szegmentálás egyszerű volt, mert a rácsáram nagy változásai épp a szegmenshatáron vannak. A rács-katód feszültség szerinti szegmentálás a jelalak alapján nem lehetett meghatározni a jelenlévő zaj miatt, ezért a szegmentálandó 150 x 80 hosszú mintákat 150 részre osztottam a mintaszám alapján.

Az így keletkező 80 mintás blokkoknak az első és utolsó 10 mintáját eldobtam, hogy a tranziensek ne befolyásolják a keletkező táblázatot. A maradék 60 mintát átlagoltam, és ez adta a táblázat adott pozícióban lévő értékét. Egy ilyen 60 mintából álló tömb látható a következő ábrán:

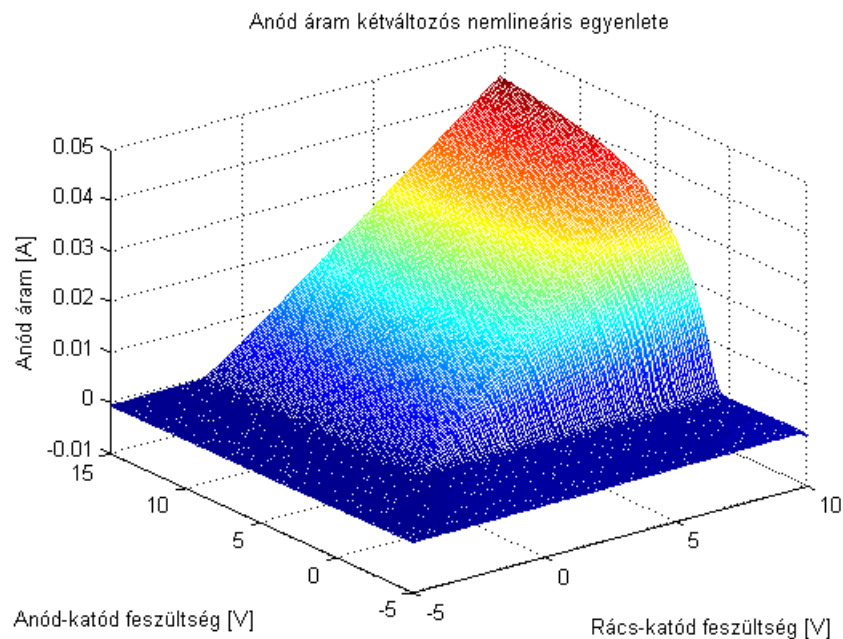


45. ábra: A táblázat egy pontjához tartozó 60 minta hosszú tömb

A kiszámolt táblázat:



46. ábra: A rács áramot leíró táblázat a mérés alapján



47. ábra: Az anód áramot leíró táblázat a mérés alapján

A táblázat értékeinek mérése során a mérőellenállásokon a gerjesztéshez képest nem elhanyagolható nagyságú feszültség esett. Ez módosította a trióda kivezetéseire jutó feszültséget, emiatt a táblázatot korrigálni kellett.

7.1.3.2 A kapott táblázatok korrekciója

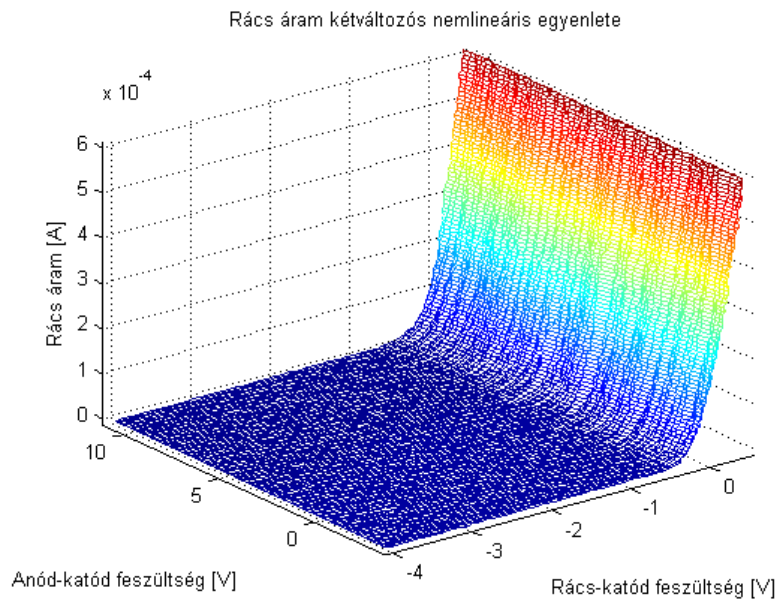
A mérés során a trióda anód-katód feszültsége nem egyezik meg a DA átalakító által generált vezérlőfeszültséggel, mert a mérőellenálláson feszültség esik. Ez kompenzálható, mert ismerjük, hogy mekkora feszültség esik a mérőellenálláson. Ez azt jelenti, hogy a táblázat kiszámolt pontjai nem épp a meglévő pontokhoz tartoznak. A táblázat pontjait transzformálni kellett az alábbi képlet szerint:

$$U_{gk} = U_{gk_idealis} - U_{ig}, \quad U_{pk} = U_{pk_idealis} + U_{ip}, \quad 123$$

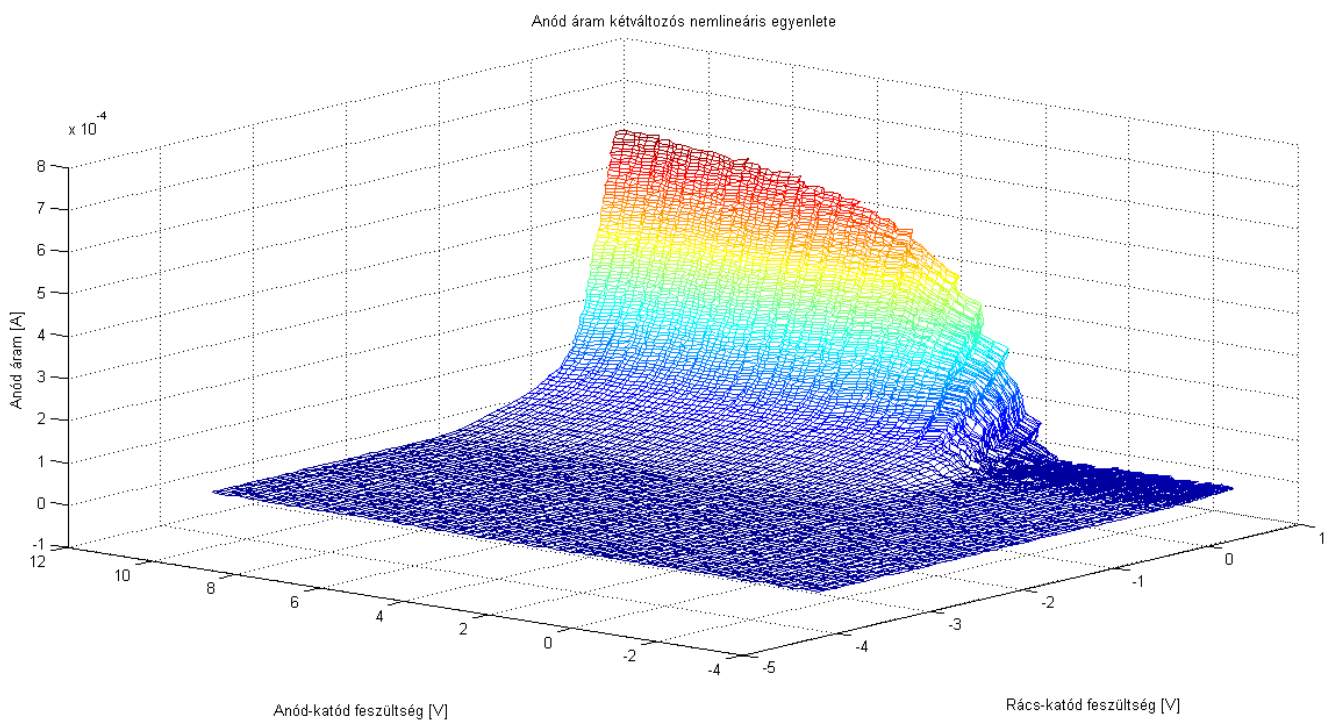
minden U_{gk} , U_{pk} pontra.

Ezután interpolációval kiszámolhatók a táblázat értékei a korábbi pontokra.

A korrigált rác-katód és anód-katód feszültségek esetén adódó rác- és anódáramok:



48. ábra: Korrigált rácsáram



49. ábra: Korrigált anódáram

A probléma: a mérőellenállásokon túl nagy feszültség esik a kivezérlehetőségi tartományhoz képest. A mérést megismételtem 10 Ohmos mérőellenállásokkal. A probléma ekkor a rossz jel-zaj viszony volt.

A mérés a mérőellenállásokon történő feszültség esés miatt csak erősen lecsökkent mérési tartományon adott hasznos információt az elektroncső karakterisztikájáról. A mérés ismételt végrehajtását javaslom a kivezérlési tartomány megkésztetésével. Ez a trióda katód feszültségének minimálisra csökkentésével érhető el a legegyszerűbben. Ehhez az alábbi beállítások szükségesek:

A mérőellenállás értéke maradjon 220 Ohm. A D/A átalakító nulladik csatornájának puffer erősítőjén a kimeneti feszültség legyen állandóan -10 V. Ez legyen a mérendő rendszer föld potenciálja, ami a katód mérőponthoz csatlakozik. A gerjesztőjeleket generáló LabView projektben a gerjesztőjelek az új földponthoz képest vegyék fel eddigi értékeiknek kétszeresét. Ezáltal a puffer erősítő és a trióda mérőpontja közötti feszültség nagy része nem a mérőellenálláson fog esni, és a jel-zaj viszony is megfelelő lesz.

Jelenleg a rács-katód és az anód-katód feszültség mérési tartománya -5 V és 10 V helyett -5 és 0.3 V körül van. A modell csőparamétereit nem tudtam a hardverben lévő elektroncső paramétereikhez illeszteni. A hardver és elkészített modell összehasonlítását a cső paraméterek illesztése nélkül kellett végrehajtanom.

7.2 A megépített Tube Driver áramkör kimeneti jelének mérése

Annak ellenére, hogy az elektroncső paramétereit nem állnak rendelkezésre, érdemes volt a megalkotott modell és a hardver kimeneti jeleit összehasonlítani adott gerjesztésre. A gerjesztőjel frekvenciatartománybeli alakja a 9. fejezetben található.

A működő hardver kimeneti jelét hangkártya segítségével rögzítettem a lentebb megadott paraméterállításokban. A gerjesztőjelet a számítógép hangkártyája szolgáltatta. A mérés során olyan paraméterértékeket választottam, melyeket viszonylag könnyen be lehetett állítani. A torzítás mértékének megválasztásánál olyan értékeket állítottam be, melyek hangzás szempontjából zeneileg hasznos kimenetet adtak. A torzítás paramétert megadó potenciométer logaritmikus skálázású. A későbbi kiértékelés során a logaritmus alapja nem állt rendelkezésre, ezért a becsült értékkel azonos beállítású *Spice*, *Matlab* és VST plugin szimulációkat végeztem, ami még egy bizonytalanságot ad az összehasonlításhoz. A hangerő paraméter olyan értékű volt, hogy a hangkártya

bemenetét ne terhelje túl a vizsgált hardver kimeneti jele. A mérést az alábbi paraméterállásokban készítettem:

mérés sorszáma	1	2	3	4
paraméterek				
torzítás	0.1	0.1	0.2	0.5
magas	1.0	0.5	1.0	1.0
mély	1.0	0.5	1.0	1.0

A mérést két paraméter bizonytalansága befolyásolta: torzítás mértéke, hangerő.

Utólag átgondolva precízen beállított, visszamért potenciométer beállításokkal az ebből a bizonytalanságból adódó hibákat ki lehet küszöbölni. A felvett adatokkal való összehasonlítás eredményei a 9.3-as fejezetben találhatóak.

8 A végleges modell kiválasztása

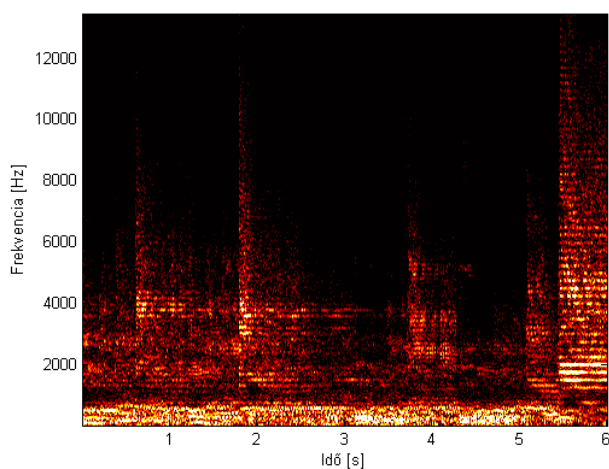
A meglévő négy modell közül az alábbi kritériumok alapján választottam ki a végleges modellt. A kritériumok fontossági sorrendben: valós időben futtatható lassulások nélkül, stabil, hangbeli hasonlóság az eredeti áramkörhöz képest.

A diszkretizációs mód relatív egyszerűségéből adódóan az elsőfokú explicit módszernek a számítási igénye a legalacsonyabb a többi modellhez képest. Az explicit séma stabilnak bizonyult kétszeres mintavételi frekvencia esetén is. Amennyiben ez a stabilitás a fejlesztés során hamarabb bebizonyult volna, a többi diszkretizációs mód C++ nyelven való megvalósítására nem lett volna szükség. A minőség összehasonlításához elegendő lett volna a *Matlab* modell.

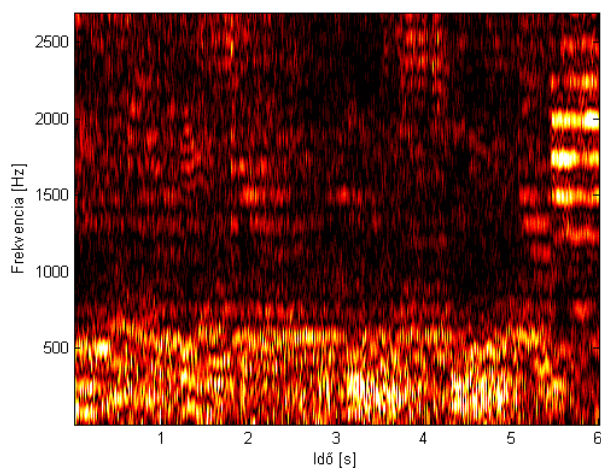
A végleges modell kiválasztásához szükség volt a VST pluginként való megvalósításhoz, hogy a valós időben futtathatóság kritériumot ellenőrizni lehessen. Az eredmények azt mutatják, hogy az implicit modellek közül a csatolást elhanyagoló implicit módszer valós időben futtatható, viszont a csatolás elhanyagolása miatt a hangzás kevésbé hű az eredeti áramkörhöz, mint az explicit, tökéletesen csatolt modell esetében. Az implicit, tökéletesen csatolt modell sajnos nem fut valós időben egy 4 GB RAM-mal és egy négymagos i3-as Intel processzorral rendelkező számítógépen.

9 Tesztek eredményei

A tesztek a VST plugin alapjául szolgáló kiválasztott modellen végeztem, ami az előrelépő Euler módszerrel diszkretizált modelleket jelenti *Matlab* és VST plugin esetében egyaránt. A végleges VST plugin az elsőfokú explicit modell alapján működik, ami az előrelépő Euler módszerrel diszkretizált modellt jelenti. A tesztek egy olyan gerjesztőjellel végeztem, mely valós használat közben tipikusan előfordul. Valódi gitárjelet használtam. A gerjesztőjel spektrumának időbeli lefolyása a következő:



50. ábra: Gerjesztőjel spektrumának időbeli lefolyása



51. ábra: Gerjesztőjel spektrumának időbeli lefolyása alacsony frekvenciákon

Az alábbi teszteseteket definiáltam az áramkör paramétereinek megfelelően. A táblázatban tartalmazza a pontos paraméter beállításokat és a rendelkezésre álló minták

forrását attól függően, hogy a mérés *Spice* programban, *Matlabban*, VST plugin-on vagy a hardveren történt. A cella bal felső sarkában a teszt eset száma látható.

hangszínszabályzó beállítás, torzítás mértéke	Mély	Közép	Magas
Nagy	1 Spice Matlab VST torzítás: 0.2 magas: 0.0 mély: 1.0	2 Spice VST Hardver torzítás: 0.3162 magas: 1.0 mély: 1.0	3 Spice Matlab VST torzítás: 0.2 magas: 1.0 mély: 0.1
Alacsony	4 Spice VST torzítás: 0.05 magas: 0.1 mély: 0.5	5 Spice Matlab VST Hardver torzítás: 0.01 magas: 1.0 mély: 1.0	6 Spice VST torzítás: 0.05 magas: 1.0 mély: 0.1

A rendelkezésre álló minták alapján az összehasonlítás egyes fázisai a következő tesztesetekben történtek meg:

Spice – Matlab: 1, 3, 5

Matlab – VST plugin: 1, 3, 5

VST plugin – Hardver: 2, 5

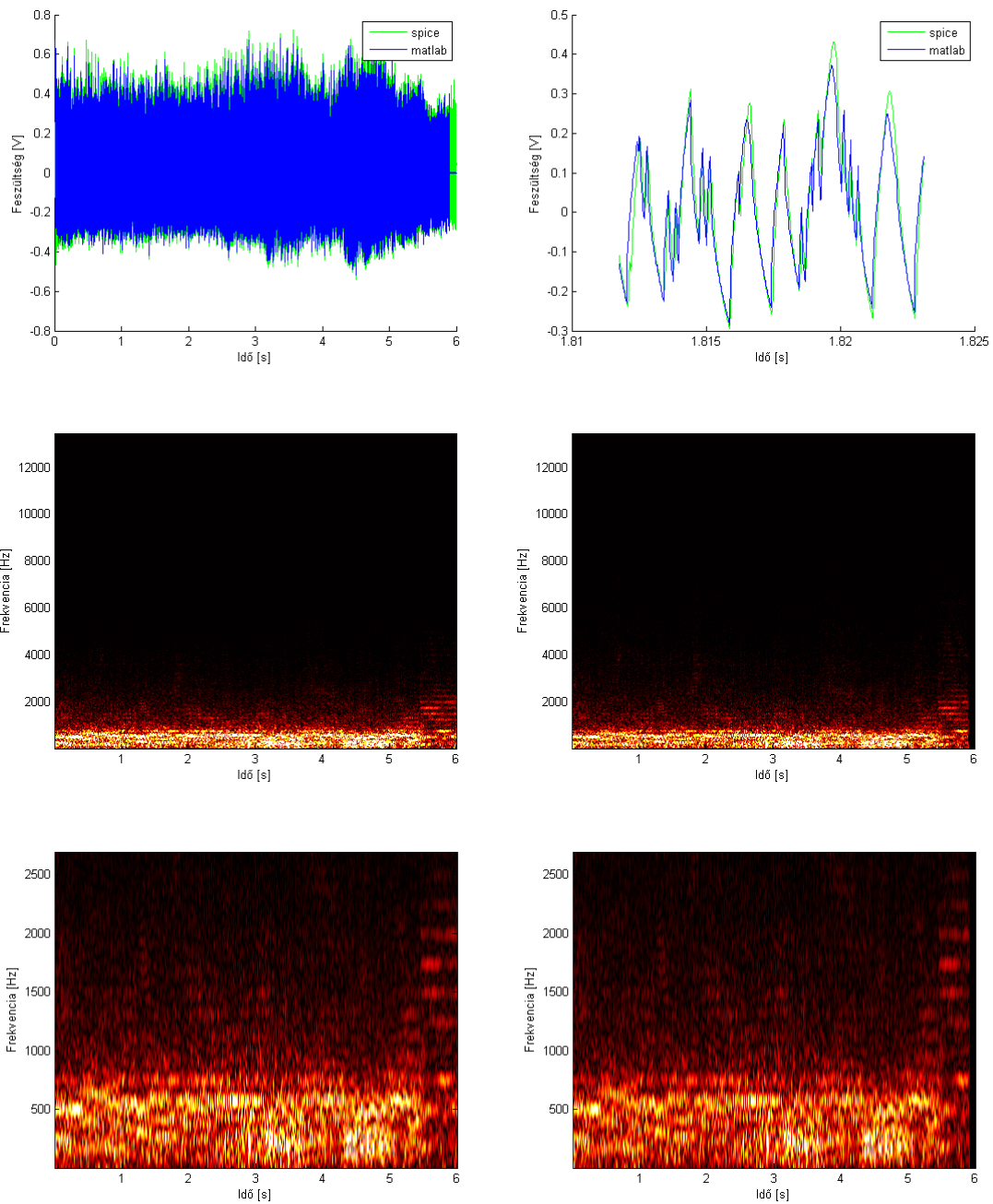
Spice – VST plugin: 1-6

A felvételek készítése során a bemenő jelek nem haladhatták meg az ábrázolási tartományt, ezért néhány esetben a jelek skálázására volt szükség. A frekvenciatartománybeli összehasonlítás az időtartománybeli jel időben ablakozott Fourier-transzformáltja alapján történt.

9.1 *Matlabban* szimulált modell összehasonlítása a *Spice* modellel

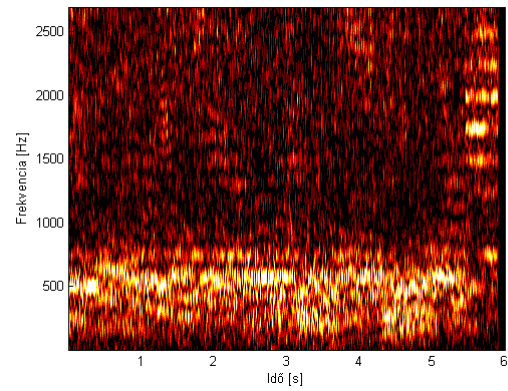
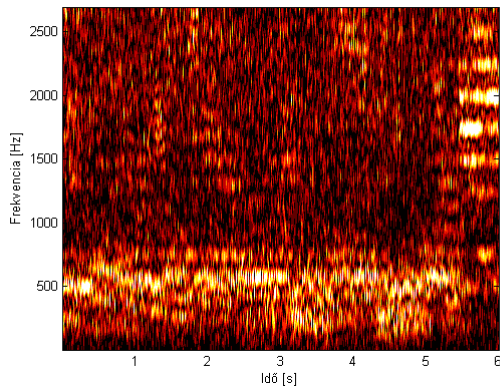
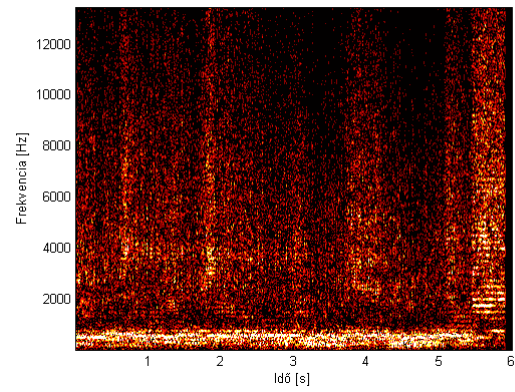
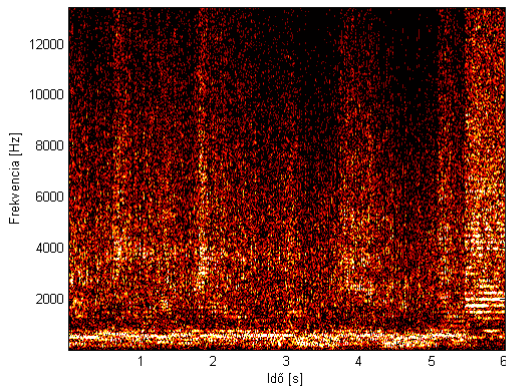
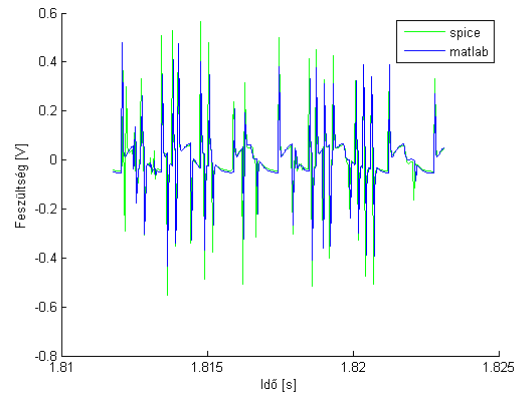
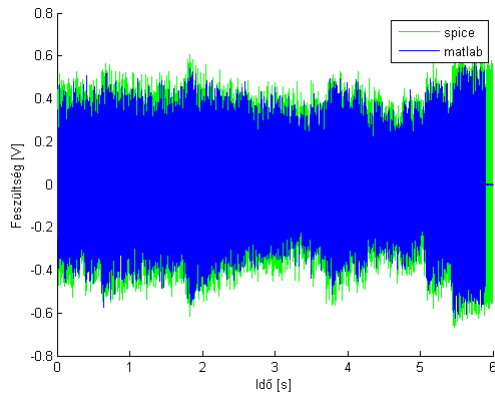
Ez az összehasonlítás azt mutatja meg, hogy mennyire pontosan sikerült az áramköri rajz, az áramköri elemek és a kiválasztott csőmodell alapján egy diszkrét idejű modellt létrehozni. A *Matlab* szimuláció során 176,400 kHz-es mintavételi frekvenciával végeztem az áramkör modelljének kiértékelését. A nemlineáris egyenletrendszert megoldó iterációs algoritmus ciklusa minden mintára négyszer fut le. Az időtartománybeli és frekvenciatartománybeli összehasonlítás eredménye a következő a megadott tesztesetekben:

Első teszt eset: torzítás: 0.2, magas: 0.0, mély: 1.0



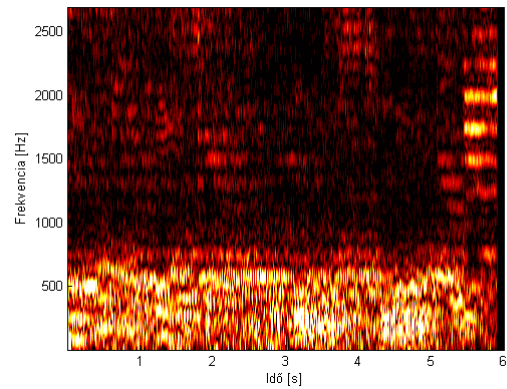
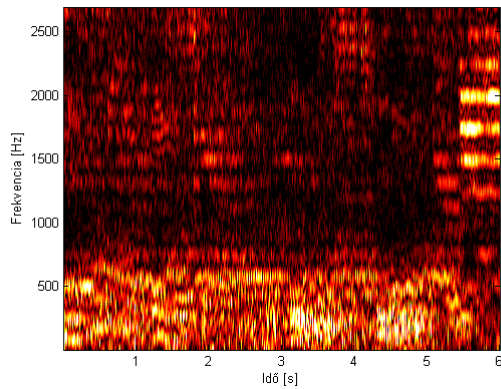
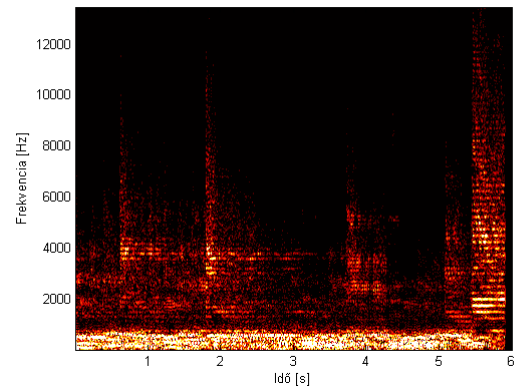
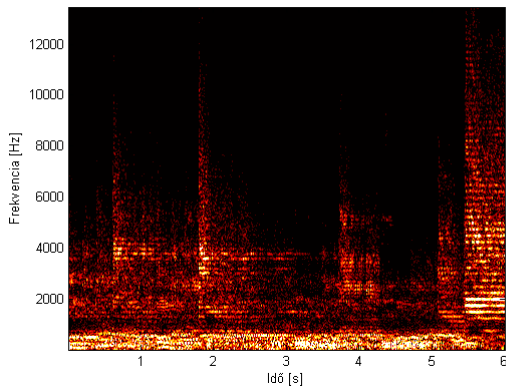
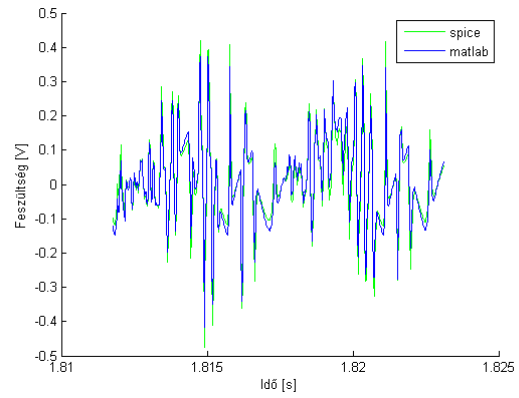
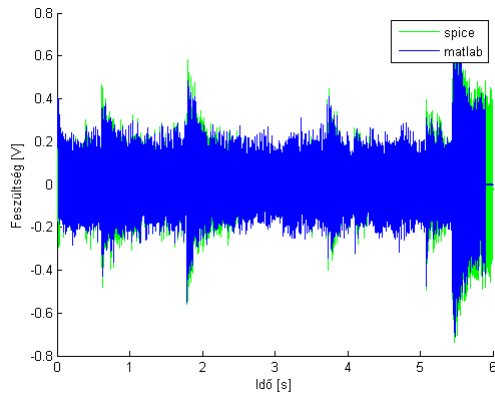
52. ábra: Összehasonlítás: a, időtartománybeli b, időtartománybeli részletes c, frekvenciatartománybeli *Spice* d, frekvenciatartománybeli *Matlab* e, frekvenciatartománybeli *Spice* részlet f, frekvenciatartománybeli *Matlab* részlet

Harmadik teszt eset: torzítás: 0.2, magas: 1.0, mély: 0.1



53. ábra: Összehasonlítás: a, időtartománybeli b, időtartománybeli részletes c, frekvenciatartománybeli *Spice* d, frekvenciatartománybeli *Matlab* e, frekvenciatartománybeli *Spice* részlet f, frekvenciatartománybeli *Matlab* részlet

Ötödik teszt eset: torzítás: 0.01, magas: 1.0, mély: 1.0



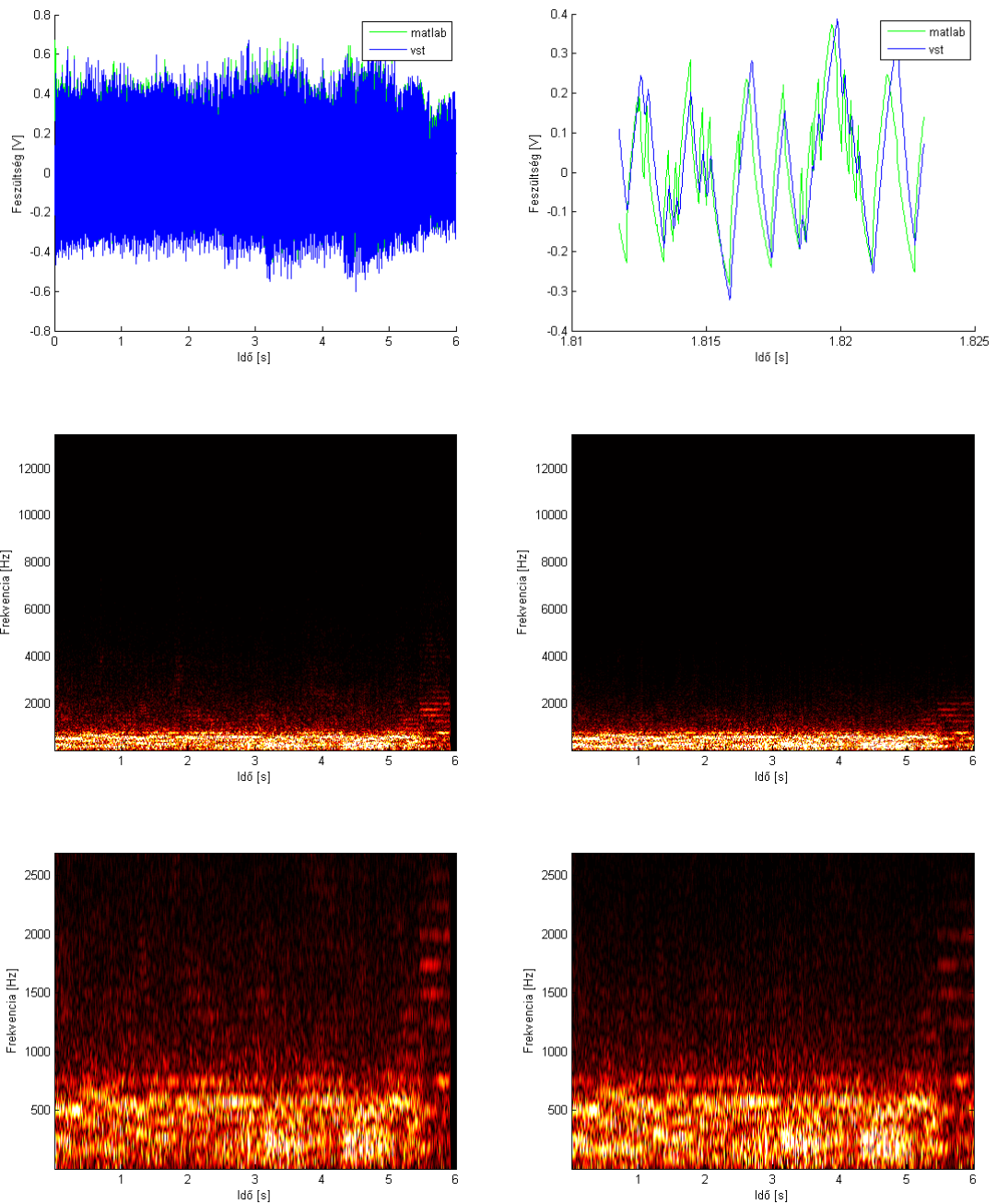
54. ábra: Összehasonlítás: a, időtartománybeli b, időtartománybeli részletes c, frekvenciatartománybeli *Spice* d, frekvenciatartománybeli *Matlab* e, frekvenciatartománybeli *Spice* részlet f, frekvenciatartománybeli *Matlab* részlet

Mindhárom esetben pontosan követi a *Matlab* modell a *Spice* program által szimulált áramkört. A *Matlab* modell az elvárt módon működik.

9.2 A VST plugin összehasonlítása a *Matlab* modellel

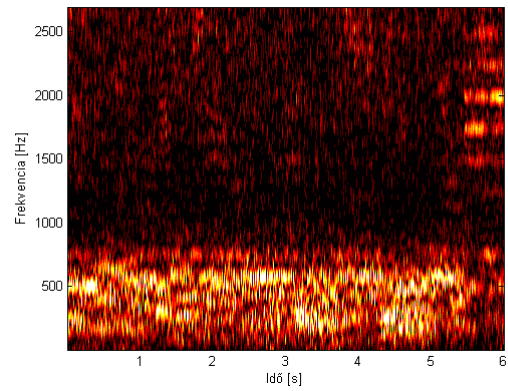
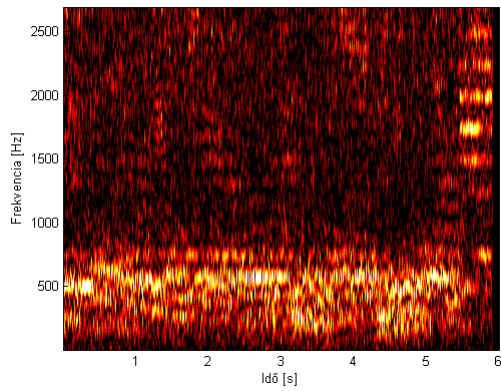
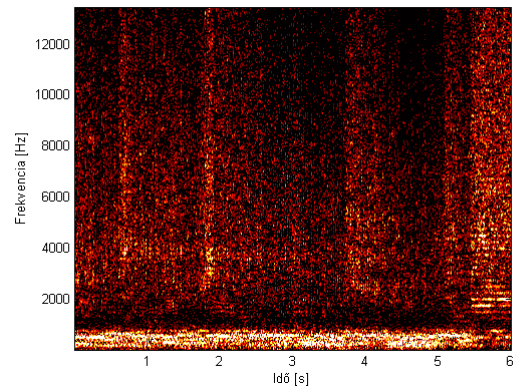
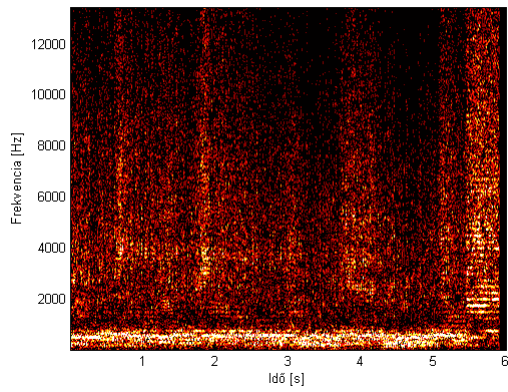
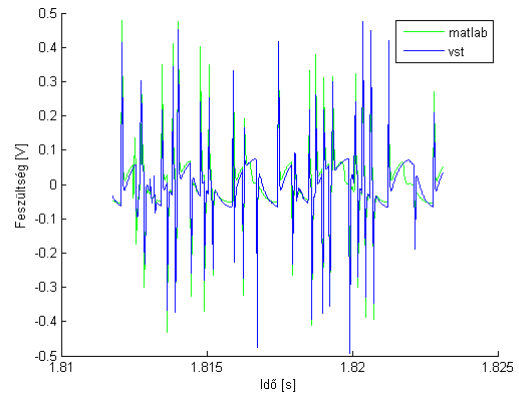
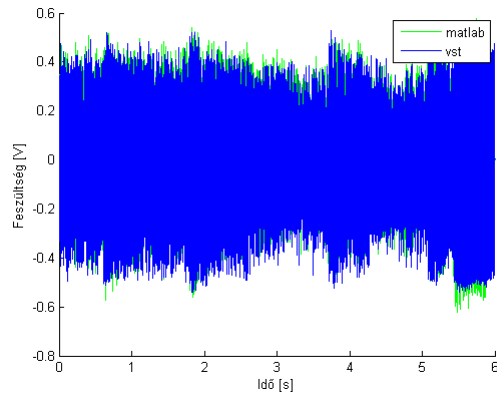
Ez az összehasonlítás a *Matlab* modell VST pluginként való megvalósításának minőségéről és a szükséges egyszerűsítések hatásáról ad információt. A nemlineáris egyenletrendszer megoldását végző iterációs ciklus minden mintára négyszer fut le. A *Matlab* szimuláció során 176,400 kHz-es mintavételi frekvenciával végeztem a modell kiértékelését. A valós időben futó VST plugin esetében ekkora számítási kapacitás nem áll rendelkezésre a megadott időre vonatkoztatva, ezért a mintavételi frekvenciát 88.2 kHz-re kellett csökkenteni. A VST plugin és a *Matlab* modell közötti különbségek ebből adódnak. A gerjesztőjel frekvenciájának növelésével növekedik a két modell válaszában a különbség. Az összehasonlítás eredménye a következő:

Első teszt eset: torzítás: 0.2, magas: 0.0, mély: 1.0



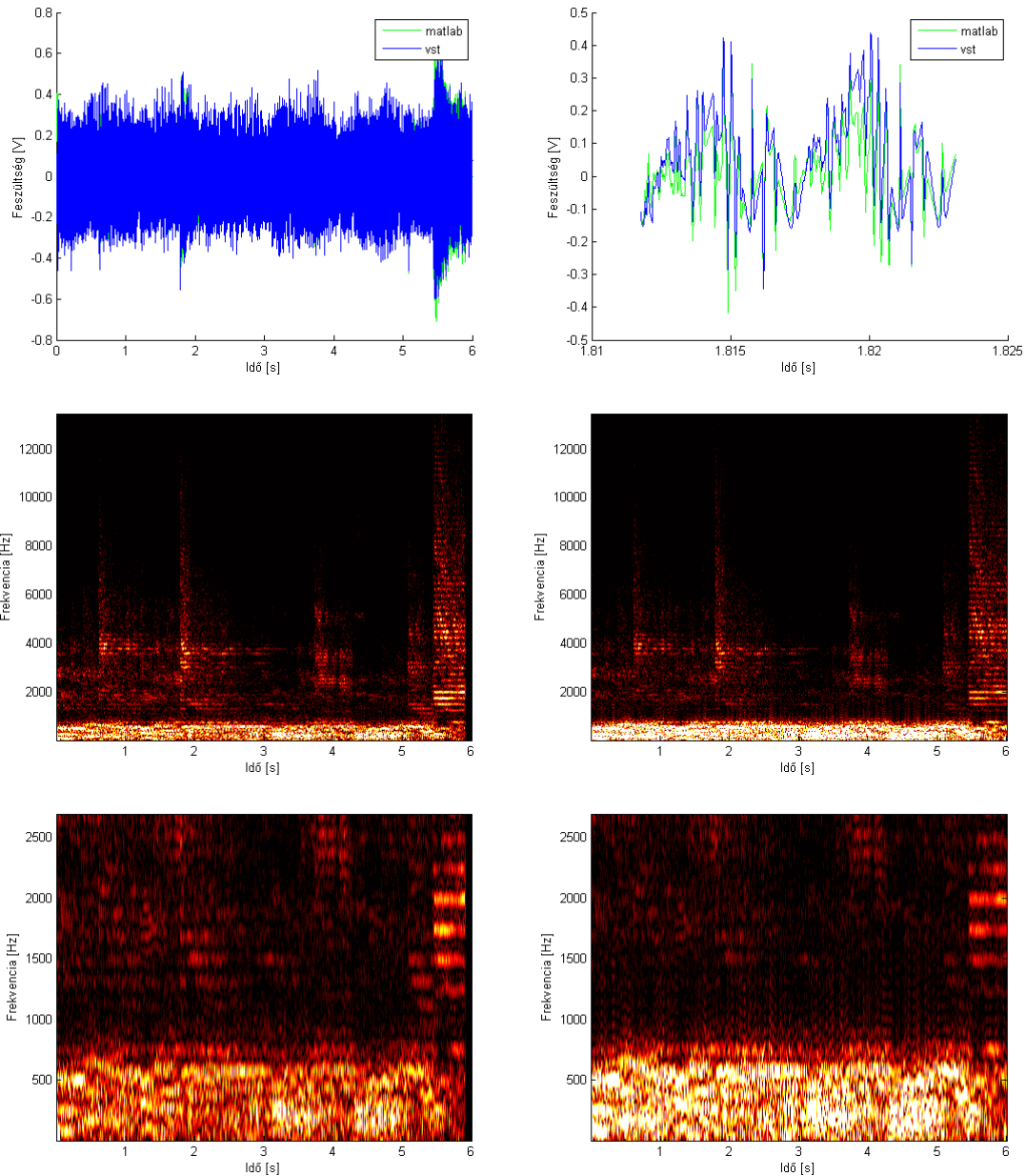
55. ábra: Összehasonlítás: a, időtartománybeli b, időtartománybeli részletes c, frekvenciatartománybeli *Matlab* d, frekvenciatartománybeli *VST* e, frekvenciatartománybeli *Matlab* részlet f, frekvenciatartománybeli *VST* részlet

Harmadik teszt eset: torzítás: 0.2, magas: 1.0, mély: 0.1



56. ábra: Összehasonlítás: a, időtartománybeli b, időtartománybeli részletes c, frekvenciatartománybeli *Matlab* d, frekvenciatartománybeli *VST* e, frekvenciatartománybeli *Matlab* részlet f, frekvenciatartománybeli *VST* részlet

Ötödik teszt eset: torzítás: 0.01, magas: 1.0, mély: 1.0



57. ábra: Összehasonlítás: a, időtartománybeli b, időtartománybeli részletes c, frekvenciatartománybeli *Matlab* d, frekvenciatartománybeli *VST* e, frekvenciatartománybeli *Matlab* részlet f, frekvenciatartománybeli *VST* részlet

Mindhárom teszt esetében az tapasztalható, hogy az 1 kHz és 2 kHz közötti felharmonikusok intenzitása csökkent a jel teljes intenzitásához képest. Az ötös teszt esetében az időtartománybeli és frekvenciatartománybeli jeleken látható, hogy a magas és mély hangok aránya kissé megváltozott. A mély hangok intenzitása nőtt a magas hangok intenzitásához képest. A két modell közötti egyetlen különbség az eltérő mintavételi frekvencia. Az 1 kHz és 2 kHz közötti frekvenciatartományban tapasztalt különbség adódhat abból, hogy a nemlineáris tartományban működő műveleti erősítő

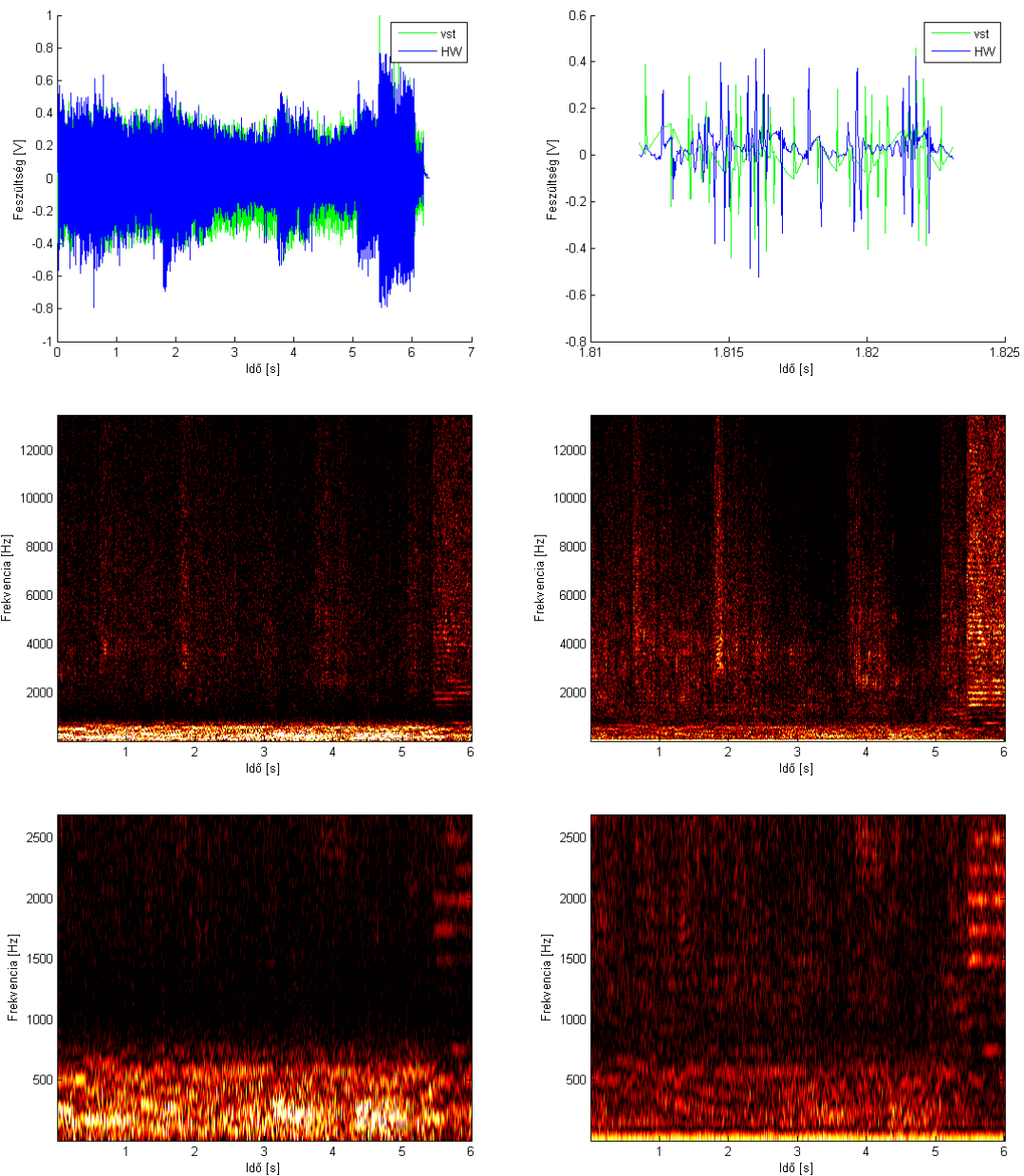
dinamikus viselkedését egy bilineáris transzformációval diszkretizált aluláteresztő típusú szűrő valósítja meg. A bilineáris transzformáció a folytonos idejű szűrő 0 és ∞ Hz közötti viselkedését 0 Hz és a mintavételi frekvencia fele közé transzformálja. A mintavételi frekvenciák eltérése miatt enyhén eltérhet a bemeneti szűrő átvitele a két esetben. Ez a kis különbség okozhat a kimeneti jelben észlelhető különbséget, mert az áramkör nagyon érzékeny a kis változásokra a nemlineáris részrendszerek vágási amplitúdójának közelében.

A mély hangok intenzitásának relatív növekedése történhet amiatt, hogy a VST plugin esetében az alacsonyabb mintavételi frekvencia miatt nagyobb jelteljesítmény lapolódhatna át nagyfrekvenciás jelekből. Ez akkor történhet meg, ha a rendszeren belül egy ponton nagyon megnő a mintavételi frekvencia felénél nagyobb frekvenciájú jelkomponensek teljesítménye. A pontos ok meghatározásához további vizsgálat szükséges.

9.3 A VST plugin összehasonlítása a valódi áramkörrel

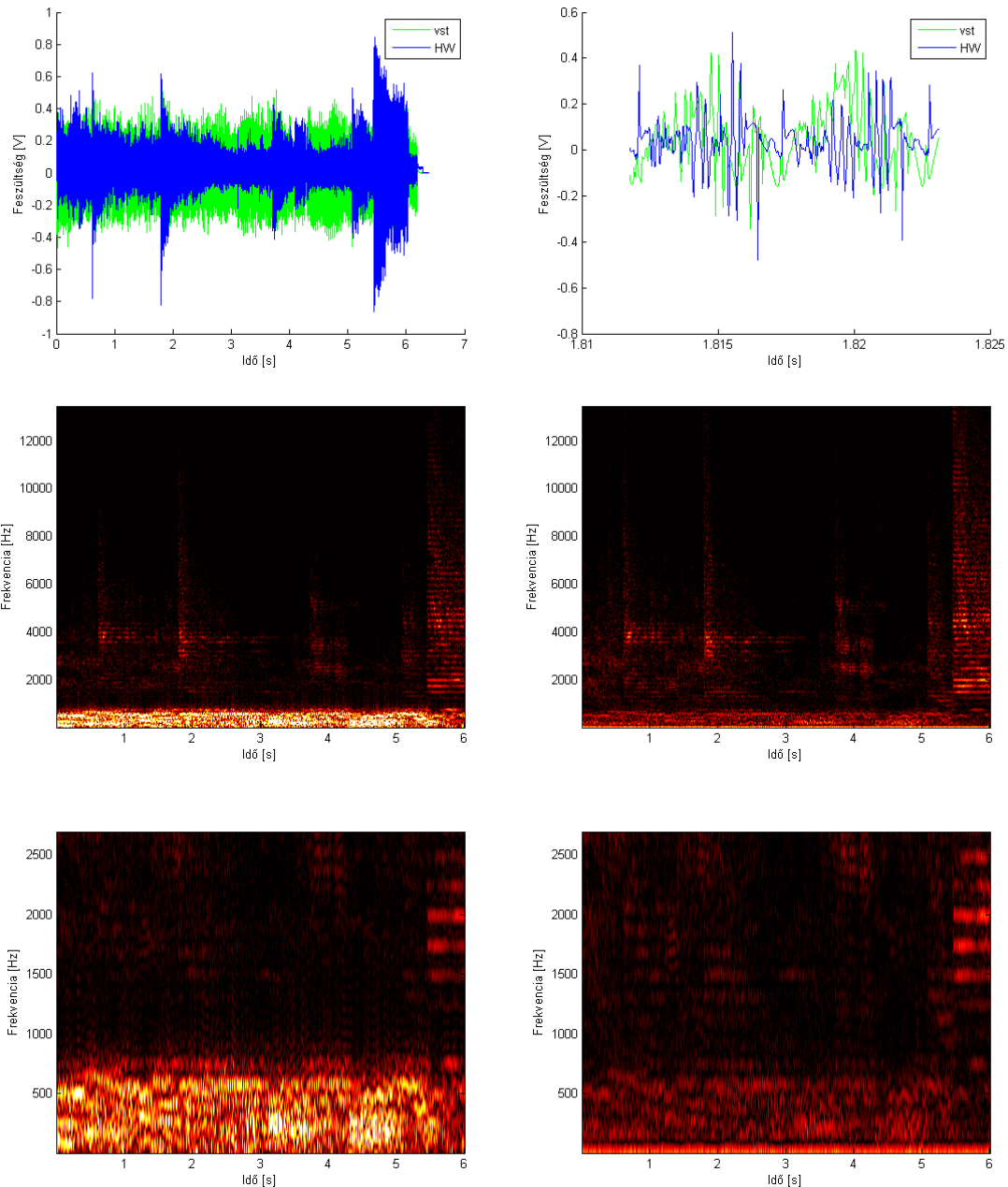
A munka végső célja, hogy a megvalósított VST plugin a valódi áramkör hangzását a lehető legkisebb eltéréssel képes legyen reprodukálni. Az áramkör mérése során a felhasznált elektroncső karakterisztikájának mérése nem sikerült kellő minőségben, ezért a hardverben lévő elektroncső karakterisztikája nem ismert. Ez hangzásbeli különbséget ad a két rendszer között még akkor is, ha a megalkotott modell egyébként tökéletes lenne. Ennek tükrében az összehasonlítás eredménye a következő:

Második teszt eset: torzítás: 0.3162, magas: 1.0, mély: 1.0



58. ábra: Összehasonlítás: a, időtartománybeli b, időtartománybeli részletes c, frekvenciatartománybeli VST d, frekvenciatartománybeli hardver e, frekvenciatartománybeli VST részlet f, frekvenciatartománybeli hardver részlet

Negyedik teszt eset: torzítás: 0.01, magas: 1.0, mély: 1.0



59. ábra: Összehasonlítás: a, időtartománybeli b, időtartománybeli részletes c, frekvenciatartománybeli VST d, frekvenciatartománybeli hardver e, frekvenciatartománybeli VST részlet f, frekvenciatartománybeli hardver részlet

A két rendszer jellegre hasonló torzítást ad, de a különbség erősen hallható. A legnagyobb különbség mély hangtartományban tapasztalható. A pontos összehasonlításhoz az elektroncső paraméterek pontos illesztésére van szükség. Nagyvonalakban mindenesetre hasonló a hardver és a megvalósított VST plugin működése.

10 Összefoglalás

A munka során a leegyszerűsített, folytonos idejű rendszer négy különböző módon diszkrétizált modelljét alkottam meg. A két implicit modell és az elsőfokú explicit modell alapján három VST programot hoztam létre. Az elsőfokú explicit modell volt időrendben az utolsó, amit sikerült stabilan lefuttatni. Ez a modell minőség és műveleti igény alapján optimális a másik kettő modellhez képest. A tökéletesen csatolt implicit modell nagyfrekvenciás viselkedés tekintetében jobb, viszont jelen állapotban egy átlagos számítógépen nem futtatható valós időben a nagy műveletigénye miatt. A tökéletesen csatolt implicit modellt tovább optimalizálva, esetleg az egyenletrendszerek megoldását párhuzamosítva, GPU támogatással alátámasztva meg lehetne oldani, hogy az implicit, tökéletesen csatolt modell valós időben fusson.

A modellek eredeti áramkörhöz való hűségét úgy lehetne növelni, hogy változó lépésközű algoritmusra váltom a fix lépésközű egyenletmegoldó algoritmust, vagy az implicit egyenletmegoldó algoritmus futásának számát nem fixre állítom be, hanem a maradó hiba értékétől teszem függővé. Ez azért javítaná a rendszer minőségét, mert nagyfrekvenciás jeleknél nagyobb a maradó hiba az egyenletmegoldó algoritmus futtatása után. Ezek a változtatások azt eredményezik, hogy a bemeneti jeltől függ a VST modul teljesítménye. Ez a hatás nem hátrány, ha elegendő erőforrás áll rendelkezésre. A bemeneti és kimeneti jeleken való számítása blokkosítva van. Ez a teljesítménybeli ingadozás kiegyenlítésében segítene.

A hardverrel való hasonlóságról csak részben tudtam megbizonyosodni a csőkarakterisztikamérés során adódott probléma miatt. A megalkotott VST plugin jól közelíti a *SPICE* szimuláció eredményeit, a torzítás típusa hasonló a hardveren tapasztalttal, de a különbség jelentős. Érdeemes lenne újra mérni az elektroncső karakterisztikáját, és azt felhasználva összehasonlítani őket.

A VST plugin helyes működésének ellenőrzésére további nagy frekvenciájú gerjesztőjelet használó tesztek javasoltak. A program kis változtatásokkal képes lenne különböző elektroncső karakterisztika táblázatokat beolvasni azért, hogy a felhasználó több közül választhasson. A program csak egy processzort használ, és egy szálon fut. Szükség esetén párhuzamosítani lehetne a legtöbb erőforrást felhasználó függvényeket. A tökéletesen csatolt implicit modellben a gyökkereső iterációban megoldandó

egyenletrendszer felbontható kisebb olyan egyenletrendszerekre, melyek egymás után megoldhatók. Ez azért lehetséges, mert a triódák anód-katód feszültsége és anód árama nem befolyásolja a rács-katód feszültségét és rács áramát. A kisebb Jacobi-mátrixok LU felbontását és a kevesebb egyenletről álló egyenletrendszerek megoldását kevesebb számítással el lehet végezni, mint a 10×10 es Jacobi-mátrixét és 10 egyenletrendszerét. Ez felgyorsítaná az implicit modell működését. Az implicit modellre való áttérés javíthatja a VST plugin nagyfrekvenciás működését.

Irodalomjegyzék

- [1] Ivan Cohen, Thomas Helie: *Simulation of a guitar amplifier stage for several triode models: examination of some relevant phenomena and choice of adapted numerical schemes*, AES 127th Convention, New York NY, USA, 2009 October 9–12, Preprint No.:7929
- [2] Ivan Cohen, Thomas Helie: *Measures and parameter estimation of triodes, for the real-time simulation of a multi-stage guitar preamplifier*, AES 129th Convention, San Francisco, CA, USA, 2010. November 4–7, Preprint No.:8219
- [3] David T. Yeh, Julius O. Smith: *Discretization of the '59 Fender Bassman tone stack*, Proc. of the 9th Int. Conference on Digital Audio Effects (DAFx-06), Montreal, Canada, September 18-20, 2006
- [4] Ivan Cohen, Thomas Helie: *Real-time simulation of a guitar power amplifier*, Proc. of the 13th Int. Conference on Digital Audio Effects (DAFx-10), Graz, Austria, September 6-10, 2010
- [5] Endre Süli: *Numerical Solution of Ordinary Differential Equations*, Mathematical Institute, University Oxford, October 18, 2014
- [6] Tube Driver Review: http://www.ultimate-guitar.com/reviews/guitar_effects/bk_butler/tube_driver (2014-11-12)
- [7] John Butcher: *Runge-Kutta methods for ordinary differential equations*, CUE Workshop on Numerical Analysis, Kyushu University, May 2005
- [8] Doron Levy: *Introduction to Numerical Analysis*, Department of Mathematics and Center for Scientific Computation and Mathematical Modeling (CSCAMM), University of Maryland
- [9] David Bindel, Jonathan Goodman: *Principles of Scientific Computing Linear Algebra II, Algorithms*, March 2, 2006
- [10] Norman Koren: *Improved vacuum tube models for SPICE simulations*, http://www.normankoren.com/Audio/Tubemodspice_article.html, (2014-11-12)