



Budapesti Műszaki és Gazdaságtudományi Egyetem
Villamosmérnöki és Informatikai Kar
Mesterséges Intelligencia és Rendszertervezés Tanszék

Gitárerősítő modellezése és megvalósítása VST környezetben

SZAKDOLGOZAT

Készítette
Jedla Martin

Konzulens
Dr. Bank Balázs

2024. december 8.

Tartalomjegyzék

Kivonat	i
Abstract	ii
1. Bevezetés	1
1.1. A dolgozat célja	1
2. Analóg áramkörök digitális modellezése	3
2.1. Lineáris áramkörök modellezése	3
2.1.1. Előrelépő Euler-módszer	3
2.1.2. Hátralépő Euler-módszer	4
2.1.3. Bilineáris transzformáció	5
2.1.4. Impulzus invariáns transzformáció	6
2.1.5. Diszkretizálás utáni szimuláció	7
2.2. Nemlineáris elemek	7
2.2.1. Előrelépő Euler-módszer	8
2.2.2. Hátralépő Euler-módszer	8
2.2.3. Negyedfokú Runge-Kutta módszer	8
3. K-módszer	10
3.1. A K-módszer alap gondolata	10
3.2. A K-módszer elemzése	13
3.3. A nemlineáris rész	14
3.3.1. Dini-tétel	14
3.3.2. A Dini-tétel alkalmazása a K-módszerben	15
3.4. A K-módszer alkalmazása	15
3.5. Példa	16
3.5.1. Megvalósítás	17
3.5.2. Eredmény	18
4. A K-módszer összehasonlítása más állapotteres módszerekkel	19
4.1. Az állapotteres módszerek	19
4.1.1. Euler-módszerek	19
4.1.2. K-módszer	20
4.2. Rugalmasabb diszkretizáció a K-módszerben	20
4.3. Számítási igény	20
4.3.1. Hátralépő Euler-módszer	21
4.3.2. K-módszer	22
4.4. Az összehasonlítás összefoglalása	22
5. Gitárerősítő modellezése	24

5.1.	Az áramköri részek modellezése	25
5.1.1.	Bemeneti szűrő	25
5.1.2.	Erősítő rész	26
5.1.2.1.	A műveleti erősítőt tartalmazó rész	27
5.1.2.2.	Passzív hálózat	28
5.1.3.	Torzító rész	29
5.1.3.1.	A műveleti erősítő levágásának megvalósítása	32
5.1.4.	Tone stack	34
5.1.5.	Aktív szűrő	37
5.1.6.	Összeadó áramkör	38
5.1.7.	Hangerőszabályozó	39
5.1.8.	Végfokozat	40
5.2.	A torzítás okozta belapolódó felharmonikusok kiszűrése	41
5.2.1.	Shannon-féle mintavételi tétel	41
5.2.2.	Harmonikus torzítás	42
5.2.3.	Újramintavételezés	42
5.2.3.1.	Kapcsolat a folytonos és diszkrét idejű jel között	43
5.2.3.2.	Interpoláció	43
5.2.3.3.	Decimálás	45
5.2.4.	Megvalósítás a pluginban	45
5.2.4.1.	Paraméterek megválasztása	47
5.3.	Paraméterváltozás kezelése	49
5.3.1.	Lehetséges megoldások	49
5.3.2.	Megvalósítás	50
5.4.	Eredmények	50
6.	Eredmények, tapasztalatok, jövőbeli tervek	52
6.1.	Összehasonlítás	52
6.2.	Az Orange Crush 20L erősítő valós idejű szimulálása K-módszerrel	53
6.3.	Jövőbeli tervek	53
	Köszönetnyilvánítás	54
	Irodalomjegyzék	55
	Függelék	57
F.1.	K-módszer a hátralépő Euler-módszerben használt diszkrétizálással	57
F.2.	A túlmintavételezéshez használt szűrő összehasonlítása különböző paramé- terek esetén	58
F.2.1.	A szűrő együtthatószáma	58
F.2.2.	A különböző β együtthatós Kaiser-ablakok összehasonlítása	60
F.3.	Az elkészült plugin tesztelése és összehasonlítása LTSpice szimulációkkal	61

HALLGATÓI NYILATKOZAT

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

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

Budapest, 2024. december 8.

Jedla Martin
hallgató

Kivonat

Napjainkban a zeneiparban és a zenészek körében a hangzás megváltoztatása céljából alkalmazott hardver-effektek helyett egyre gyakrabban használnak számítógépen futtatható pluginokat. Ezek a pluginok valós időben futnak, és gyakran a fizikai effekt áramkörök alapján valósítják meg őket, lehetőleg úgy, hogy ugyanazt a hangzásélményt nyújtsák.

Ezeknek a pluginoknak az elkészítése nem triviális: a zenei eszközök áramkörében gyakran találhatóak nemlineáris elemek (diódák, elektroncsövek stb.), amelyek nehezítik digitális modellezésüket. A valós idejűség követelménye miatt olyan módszereket kell találni, amelyek elfogadható számítási igény mellett is megfelelő pontossággal oldják meg a nemlineáris differenciálegyenlet rendszert. Ezek a módszerek jellemzően nem triviális egyszerűsítésekkel (akár az egyenletrendszer másképpen való felírásával, akár algebrai átalakítások segítségével) próbálják ezt elérni.

Nemlineáris differenciálegyenletek megoldására a legegyszerűbb és leggyakrabban használt eljárások az úgynevezett Euler-módszerek. Azonban az előrelépő Euler-módszer esetén a stabilitás nem garantált, a hátralépő Euler-módszer pedig egy nemlineáris differenciálegyenlet numerikus megoldását igényli, ezzel nagy számítási igényt előidézve.

A kevésbé ismert K-módszer a számítási igényt azzal igyekszik csökkenteni, hogy az áramkör lineáris és nemlineáris részáramkörökre bontása után először a nemlineáris rész kimenetét számolja ki geometriai módszerek segítségével, majd csak ezután számolja ki a további szükséges változókat. A dolgozat egyik célja a K-módszer és a hátralépő Euler-módszer kapcsolatának vizsgálata, valamint hasonlóságai és különbségeik feltérképezése. Emellett a dolgozat azt is részletezi, hogy milyen előnyei és hátrányai vannak a K-módszernek a hátralépő Euler-módszerrel szemben.

A dolgozatban az Orange Crush 20L gitárerősítő modelljét készítettem el, valamint ennek alapján valós időben futó VST3 plugint fejlesztettem a JUCE fejlesztői környezetben. A fejlesztés célja a fizikai erősítő hangzásának minél pontosabb megvalósítása volt valós idejű szoftveres környezetben.

A fejlesztést MATLAB környezetben végeztem, az eredményeket az LTSpice program segítségével teszteltem. Ezt követően a MATLAB-ban megvalósított algoritmusokat átültettem a JUCE fejlesztői környezetbe, és elkészítettem a VST3 plugint. A fejlesztési folyamatot és a plugin tesztelését részletesen bemutatom.

Az erősítő lineáris részeit bilineáris transzformáció segítségével modelleztem, a nemlineáris részek modellét pedig a K-módszer segítségével valósítottam meg. Az áramkörben található műveleti erősítők levágását figyelembe vettem, ha a valóságnak megfelelő bemeneti jel esetén levágnak, egyébként pedig ideálisnak tekintettem azokat. A műveleti erősítők slew rate-jét és a frekvenciafüggő viselkedését elhanyagoltam.

Az erősítő áramkörét részekre bontottam, amelyeket külön-külön modelleztem. Az erősítőn található potenciométerek számos paraméter megjelenését eredményezték az egyenletekben. Ezek állíthatóságával járó nehézségeket és azok megoldásait részletezem. A nemlineáris torzítások okozta átlapolódás csökkentésére túlmintavételezést alkalmaztam.

A dolgozat végén összefoglalom a fejlesztés során szerzett tapasztalatokat és az elért eredményeket, valamint javaslatot teszek néhány lehetséges továbbfejlesztési irányra.

Abstract

In today's music industry and among musicians, instead of using hardware effects to alter sound, digital plugins are increasingly being used. These plugins are run in real time and are mostly designed to emulate the physical circuit of the desired effect, aiming to deliver the same listening experience.

The design of these plugins is not trivial: the circuits of musical devices often contain nonlinear components (diodes, valves, etc.), complicating their digital modeling. Due to the real-time requirement, appropriate methods must be applied that can solve the ordinary differential equation caused by the nonlinearity with sufficient accuracy while having an acceptable computational demand. These methods typically achieve this through non-trivial simplifications (by defining the equation system differently or algebraic transformations).

One of the simplest and most common methods used for solving nonlinear differential equations are the so-called Euler-methods. However, the forward Euler method does not guarantee system stability, and the backward Euler method requires numerically solving nonlinear equations.

The lesser-known K-method reduces computational demand by dividing the circuit into linear and nonlinear subcircuits. It first computes the nonlinear output geometrically, then calculates the other necessary variables. One of the goals of this thesis is to examine the connection between the K-method and the backward Euler-method, as well as to detail their similarities and differences. Additionally, the thesis details the advantages and disadvantages of the K-method against the backward Euler-method.

In this thesis, I have modeled the Orange Crush 20L guitar amplifier and developed a real-time VST3 plugin using the JUCE framework. The goal was to replicate the sound of the analog circuit in a real-time software environment.

The development process was carried out in MATLAB, and the results were tested using LTSpice. The MATLAB algorithms were then transferred to JUCE to complete the VST3 plugin. The development process and the plugin testing are presented in detail.

The linear components of the amplifier were modeled using bilinear transformation, while the nonlinear components were implemented using the K-method. For the operational amplifiers in the circuit, their clipping behavior was modeled when the input signal would realistically cause it; otherwise, they were treated as ideal. The slew rate and frequency-dependent behavior of the operational amplifiers were neglected.

The amplifier circuit was divided into subcircuits, which were modeled separately. Due to the potentiometers on the amplifier, a large number of parameters appeared in the equations. The challenges posed by these adjustable parameters and their solutions are detailed. Oversampling was applied to reduce aliasing caused by nonlinear distortion.

The thesis concludes with a summary of the development insights, results, and potential future improvements.

1. fejezet

Bevezetés

A digitális technológia fejlődésével a zeneipar egyre inkább a szoftveres megoldások felé fordult, és ennek egyik legjelentősebb következménye a digitális pluginok térnyerése. Az olyan fizikai eszközök, mint az analóg keverőpultok, erősítők, kompresszorok, zengetők vagy különféle effektpedálok egyre gyakrabban helyettesíthetők digitális változataikkal, amelyek akár egyetlen laptop segítségével is alkalmazhatóak, így a zenei produkció mobilisabbá, költséghatékonyabbá és hozzáférhetőbbé vált.

Az elmúlt évtizedekben a pluginok fontossága drámai módon növekedett, ami részben annak köszönhető, hogy a digitális audio munkaállomások (DAW: Digital Audio Workstation) népszerűsége megugrott, és ma már az otthoni stúdióktól a professzionális felvételekig minden területen jelen vannak. A DAW-ok olyan szoftverkörnyezetek, amelyek lehetővé teszik a felhasználók számára, hogy különféle hangszerek, ének és egyéb hangforrások sávjait rögzítsék, szerkesszék és effektezzék. Ezek a platformok pluginok által biztosítják azt a lehetőséget, hogy a korábban csak fizikai hardver formában elérhető effekteket alkalmazhassák a felvételeken.

A pluginok hatalmas előnye, hogy a fizikai eszközökkel szemben rugalmasabbak és dinamikusabbak. Emellett a legtöbbjük könnyen beilleszthető bármelyik modern DAW környezetbe. A különböző plugin-formátumok, mint például a VST, AU, AAX vagy a VST3 biztosítják, hogy ezek az eszközök széles körben elérhetők legyenek, kompatibilisek legyenek a különböző operációs rendszerekkel és platformokkal, és lehetővé teszik, hogy mind a professzionális, mind a hobbi zenészek elérjék és használhassák őket saját produkcióikban.

1.1. A dolgozat célja

A fizikai áramköröket valós időben modellező pluginok létrehozása gyakran komoly kihívást jelent: ezek bonyolultsága megköveteli, hogy a készítő optimalizálja a számítási igényt, hogy a szimuláció minél kisebb erőforrásigénnyel valósulhasson meg valós idejű futása alatt.

A modern elektronikai áramkörök szimulációjában gyakran alkalmaznak állapotteres módszereket. Az állapotváltozós leírással történő szimuláció különösen hasznos, ha a rendszer viselkedését komplex, nemlineáris elemek határozzák meg. Az ilyen típusú elemek, mint például tranzisztorok vagy egyéb aktív eszközök, jelentős nehézségeket okoznak a szimulációs algoritmusokban, mivel a nemlinearitásokat megfelelő módon kell kezelni a pontos és hatékony eredmények eléréséhez.

A dolgozat célja, hogy bemutassa az Orange Crush 20L tranzisztoros gitárerősítő áramkörének valós idejű szimulációját, amely VST környezetben valósul meg. A célom,

hogy a valós idejű szimuláció révén a fizikai, analóg erősítő viselkedése reprodukálható legyen egy digitálisan, így a fejlesztett plugin képes legyen annak helyettesítésére.

A dolgozat első részében áttekintem az áramkörök modellezéséhez leggyakrabban alkalmazott állapotterez módszereket, részletezve azok alapvető jellemzőit és alkalmazhatóságukat. Ezt követően a két, valós idejű szimuláció esetén gyakran használt eljárást, a hátralépő Euler-módszert és a K-módszert részletesen összehasonlítom, figyelembe véve a számítási igényt, a módszerek előnyeit és hátrányait, valamint az egyes eljárások alkalmazhatóságának feltételeit.

Az erősítő modellezéséhez a bilineáris transzformációt és a K-módszert használom. Az áramkört először részáramkörökre bontom, majd az egyes részáramkörök modellezését részletesen bemutatom. A nemlineáris elemek hatása, különösen a torzítások és az abból következő belapolódó felharmonikusok, komoly kihívást jelentenek, így ezek kezelésére interpolálás és decimálás eljárásokat alkalmazok. Az interpolálás során a bemenő jelet a kívánt mintavételi frekvencia növeléséhez módosítom. A decimálás célja pedig az, hogy a jelet a nemlineáris elemek feldolgozása után visszakonvertáljam az eredeti mintavételi frekvenciára, minimalizálva a belapolódott komponenseket. Mindkét folyamatnak jelentős hatása van a rendszer teljesítményére és a jel minőségére, így fontos, hogy a megfelelő szűrők legyenek alkalmazva a hibák minimalizálására, miközben a számítási igény is megfelelő maradjon. A dolgozatban részletesen elemzem az interpolálás és decimálás hatásait. Bemutatom a véglegesen megválasztott szűrőt, amely a két eljárás alkalmazásához szükséges, és ismertetem a megvalósítás részleteit.

A szimuláció során tesztelem az egyes áramköri részeket, majd az egész plugin kimenetét is összevetem az LTSpice szimulációval. Az LTSpice kimenetével való összehasonlítás indokolt, mivel az LTSpice egy széles körben használt szimulátor, amelyet az iparban és kutatásokban is gyakran referenciaértékként alkalmaznak. Az LTSpice kimenetével történő összevetés segít a plugin pontosságának validálásában, és megerősíti annak alkalmazhatóságát a gyakorlatban.

A dolgozat végén különböző hangfájlokat generálok az LTSpice program segítségével, hogy meghallgatásos tesztek alapján is megállapíthassam, hogy a végleges plugin képes-e a fizikai, analóg erősítő hangzását reprodukálni.

2. fejezet

Analóg áramkörök digitális modellezése

2.1. Lineáris áramkörök modellezése

Először a lineáris áramkörök modellezésének leggyakoribb módszereit ismertetem [1, 2] alapján.

Egy lineáris, időinvariáns rendszert pontosan jellemez az átviteli függvénye, amely a kimenet és a bemenet Laplace-transzformáltjainak hányadosaként definiálható. A legtöbb, általunk vizsgált rendszer megadható s -beli polinomok hányadosaként.

$$H(s) = \frac{Y(s)}{U(s)} \quad (2.1)$$

Cél ebből a hányadosból olyan végtelen impulzusválaszú szűrő kifejezése, ami ezt a viselkedést valósítja meg.

Lehetséges az a megközelítés is, hogy az állapotváltozós leírás alapján szimuláljuk a rendszert. Ebben az esetben minden belső változó is kiszámítható lenne, azonban a legtöbb esetben ez nem szükséges, mivel csupán a rendszer kimenete a lényeges. Az állapotváltozós leírás szerinti szimuláció több egyenlet megoldását igényli. Emiatt lineáris hálózatok esetén, a számítási igény csökkentése érdekében, érdemes kerülni.

2.1.1. Előrelépő Euler-módszer

Az előrelépő (explicit) Euler-módszer a derivált közelítéséből indul ki. Ezt a következő és az aktuális állapotváltozóból fejezi ki, a többi tagnak az ezelőtti állapotát használja.

$$\dot{\mathbf{w}} = \frac{\mathbf{w}[n+1] - \mathbf{w}[n]}{\Delta t} \quad (2.2)$$

$$\dot{\mathbf{w}} = \frac{\mathbf{w}[n+1] - \mathbf{w}[n]}{\Delta t} = \mathbf{A}\mathbf{w}[n] + \mathbf{B}\mathbf{u}[n] \quad (2.3)$$

ahol \mathbf{A} az állapotmátrix, \mathbf{w} az állapotvektor és \mathbf{u} a gerjesztésvektor. A (2.2)-t Laplace-transzformálva:

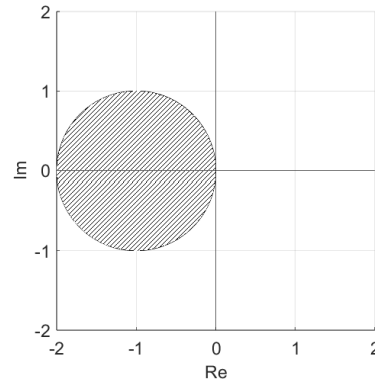
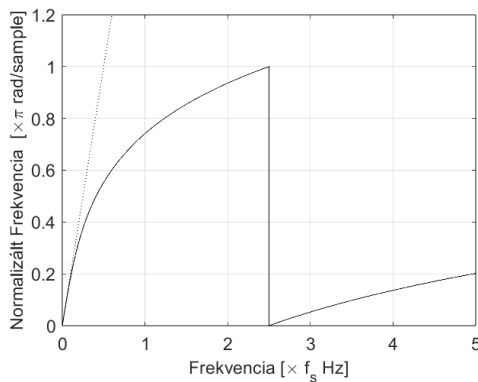
$$s = \frac{z-1}{T} = \frac{1-z^{-1}}{Tz^{-1}} \quad (2.4)$$

Az előrelépő Euler-módszer nem mindig fog stabil folytonos idejű rendszerből stabil diszkrét idejű rendszert eredményezni, ami 2.4-ből egyértelműen kifejezhető [1]. A stabilitás függeni fog a mintavételi frekvenciától, és ha az nem elég nagy, akkor instabil lesz a diszkrétizált rendszer. A (2.4) egyenlet átrendezve:

$$z = sT + 1 \quad (2.5)$$

Az előrelépő Euler-módszerrel diszkrétizált rendszer stabilitási feltétele a 2.1b ábrán látható. Ahhoz, hogy az előrelépő Euler-módszerrel diszkrétizált rendszer stabil legyen, a pT szorzat, ahol p a folytonos idejű rendszer pólusa, az ábrán jelölt tartományon belül legyen.

A 2.1a ábrán látható, hogy a folytonos idejű rendszer ω körfrekvenciái melyik θ diszkrét körfrekvenciára képződnek le a transzformáció során. Érdeemes megjegyezni, hogy a frekvenciatorzítást az átviteli függvény nagy frekvenciáinak átlapolódása okozza.



(a) Az előrelépő Euler-módszer frekvenciatorzítása

(b) Az előrelépő Euler-módszer alkalmazásával létrehozott diszkrét rendszer stabilitásának feltétele

2.1. ábra. Az előrelépő Euler-módszerrel diszkrétizált rendszer frekvenciatorzítása és stabilitási feltétele

2.1.2. Hátralépő Euler-módszer

A hátralépő (implicit) Euler-módszer a deriváltat az aktuális és az ezelőtti állapotváltozóból fejezi ki, a többi tagnak az aktuális állapotát használja.

$$\dot{\mathbf{w}} = \frac{\mathbf{w}[n] - \mathbf{w}[n-1]}{\Delta t} \quad (2.6)$$

$$\dot{\mathbf{w}} = \frac{\mathbf{w}[n] - \mathbf{w}[n-1]}{\Delta t} = \mathbf{A}\mathbf{w}[n] + \mathbf{B}\mathbf{u}[n] \quad (2.7)$$

ahol \mathbf{A} az állapotmátrix, \mathbf{w} az állapotvektor és \mathbf{u} a gerjesztésvektor. A (2.6)-t Laplace-transzformálva:

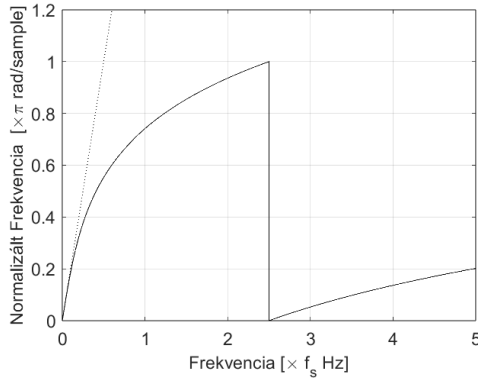
$$s = \frac{1 - z^{-1}}{T} = \frac{z - 1}{Tz} \quad (2.8)$$

A hátralépő Euler-módszerrel modellezett folytonos idejű rendszer mindig stabil diszkrét idejű rendszerhez vezet, ami 2.8 alapján belátható [1]. A (2.8) egyenlet átrendezve:

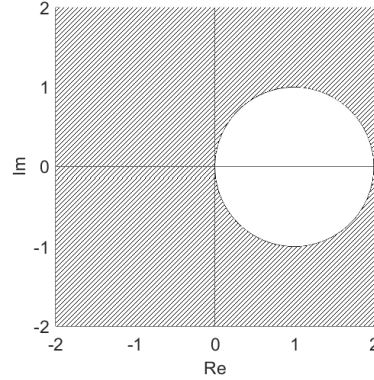
$$z = \frac{1}{1 - sT} \quad (2.9)$$

A hátralépő Euler-módszerrel diszkrétizált rendszer stabilitási feltétele a 2.2b ábrán látható. Ahhoz, hogy az hátralépő Euler-módszerrel diszkrétizált rendszer stabil legyen, a pT szorzat, ahol p a folytonos idejű rendszer pólusa, az ábrán jelölt tartományon belül legyen.

A 2.2a ábrán látható, hogy a folytonos idejű rendszer ω körfrekvenciái melyik θ diszkrét körfrekvenciára képződnek le a transzformáció során. Ugyanúgy, mint az előrelépő Euler-módszer esetén, a frekvenciatorzítást az átviteli függvény nagy frekvenciáinak átlapolódása okozza.



(a) A hátralépő Euler-módszer frekvenciatorzítása



(b) A hátralépő Euler-módszer alkalmazásával létrehozott diszkrét rendszer stabilitásának feltétele

2.2. ábra. A hátralépő Euler-módszerrel diszkrétizált rendszer frekvenciatorzítása és stabilitási feltétele

2.1.3. Bilineáris transzformáció

Folytonos idejű rendszerek diszkrétizációjához az egyik leggyakrabban használt módszer a bilineáris transzformáció, amely az s -síkból a z -síkra transzformál a trapézsabály, azaz az előre- és a hátralépő Euler-módszer átlaga szerint.

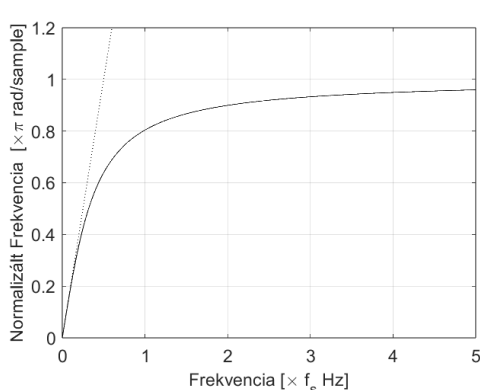
$$s = \frac{2}{T} \frac{z - 1}{z + 1} = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}} \quad (2.10)$$

A transzformáció után a rendszer stabilitása megmarad, amit 2.10-ből ki lehet fejezni [1]. A (2.10) egyenlet átrendeve:

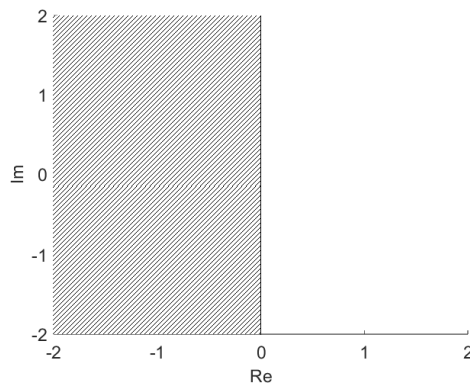
$$z = -\frac{4}{sT - 2} - 1 \quad (2.11)$$

Az Euler-módszerekkel ellentétben a bilineáris transzformáció során nem lapolódik át az átviteli függvény. Emiatt a tulajdonság miatt használják olyan széles körűen a gyakorlatban. Ezt szemlélteti a 2.3a ábra, ami megmutatja, hogy a folytonos idejű rendszer ω körfrekvenciái melyik θ diszkrét körfrekvenciára képződnek le a transzformáció során.

A bilineáris transzformáció során, ha a folytonos rendszer pólusainak a valós része kisebb, mint nulla, akkor stabil diszkrét rendszer lesz az eredmény, ami a 2.3b ábrán látható. Azaz a stabilitáshoz a pT szorzatnak, ahol p a folytonos idejű rendszer pólusa, az ábrán jelölt részen belül kell lennie.



(a) A bilineáris transzformáció frekvenciatorzítása



(b) A bilineáris transzformáció alkalmazásával létrehozott diszkrét rendszer stabilitásának feltétele

2.3. ábra. A bilineáris transzformációval diszkrétizált rendszer frekvenciatorzítása és stabilitási feltétele

2.1.4. Impulzus invariáns transzformáció

Az impulzus invariáns transzformációhoz ugyanúgy szükség van a transzformálni kívánt rendszer átviteli függvényére.

$$H(s) = \frac{B(s)}{A(s)} \triangleq \frac{b_0 s^{N-1} + b_1 s^{N-2} + \dots + b_{N-2} s + b_{N-1}}{s^N + a_1 s^{N-1} + \dots + a_{N-1} s + b_N} \quad (2.12)$$

Az átviteli függvényt részlettortekre bontva, inverz Laplace-transzformálva, majd mintavételezve:

$$H(s) = \sum_{i=1}^N \frac{K_i}{s - s_i} \quad (2.13)$$

$$h(t) = \sum_{i=1}^N K_i e^{s_i t} \quad (2.14)$$

$$h[n] = \sum_{i=1}^N K_i e^{s_i n T} \quad (2.15)$$

Ezt z-transzformálva kapjuk meg az impulzus-invariáns transzformáció segítségével megtervezett digitális szűrőt.

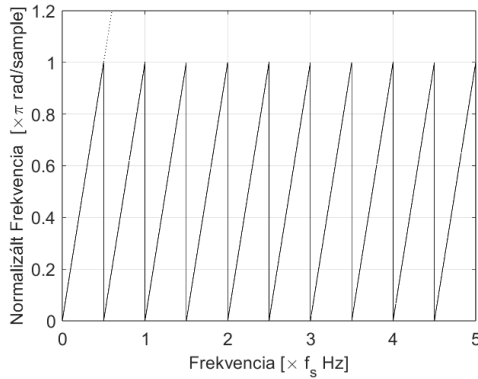
$$H(z) = \sum_{i=1}^N \frac{K_i}{1 - e^{s_i T} z^{-1}} \quad (2.16)$$

A transzformáció után a rendszer stabilitása megmarad, hiszen a transzformáció során s és z között a pontos összefüggést használjuk, és nem közelítünk [1].

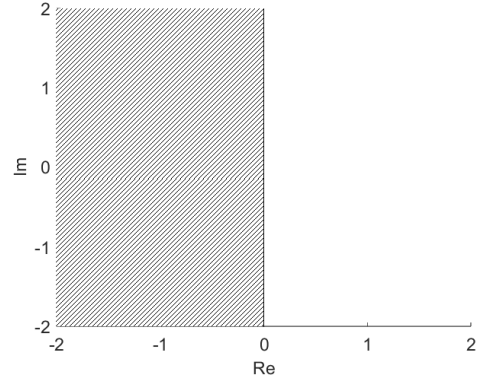
$$z_i \triangleq e^{s_i T} \quad (2.17)$$

Más szóval mondva, ha a folytonos idejű rendszernek a pólusainak a valós részei kisebbek, mint nulla, akkor az impulzus-invariáns transzformáció által eredményezett rendszer is stabil lesz. Ezt szemlélteti a 2.4b ábra, ami megmutatja, hogy a folytonos idejű rendszer ω körfrekvenciái melyik θ diszkrét körfrekvenciára képződnek le a transzformáció során.

Az impulzus-invariáns transzformáció során az átviteli függvény nagy frekvenciái átlapolódnak. Emiatt a tulajdonsága miatt csak bizonyos átvitelek esetén alkalmazzák ezt a diszkrétizálási módszert, amelyeknél az átlapolódás nem okoz problémát. Ezt szemlélteti a 2.4a ábra.



(a) Az impulzus-invariáns transzformáció frekvenciatorzítása



(b) Az impulzus-invariáns transzformáció alkalmazásával létrehozott diszkrét rendszer stabilitásának feltétele

2.4. ábra. Az impulzus invariáns transzformációval diszkrétizált rendszer frekvenciatorzítása és stabilitási feltétele

2.1.5. Diszkrétizálás utáni szimuláció

A bemutatott diszkrétizálási módszerek alkalmazásakor behelyettesítés után az átviteli függvény az alábbi alakot veszi fel:

$$\frac{Y(z)}{U(z)} = \frac{\sum_{i=0}^N b_i z^{-i}}{1 + \sum_{j=1}^M a_j z^{-j}} \quad (2.18)$$

Átrendezve és inverz z-transzformálva kifejezhető a rendszeregyenlet, ami tulajdonképpen egy végtelen impulzusválaszú (IIR) szűrő:

$$y[n] = \sum_{j=1}^M -a_j y[n-j] + \sum_{i=0}^N b_i u[n-i] \quad (2.19)$$

A szimuláció során ezt az egyenletet kell implementálni. Az y változó (válasz) utolsó M értékét, valamint az u változó (gerjesztés) utolsó N értékét kell tárolni a válasz következő értékének kiszámításához. A számítási igény csökkentése érdekében a buffereléshez célszerű cirkuláris buffer használata.

2.2. Nemlineáris elemek

Ha a rendszerbe nemlinearitás kerül, akkor a rendszer átviteli függvénye nem értelmezett. Ekkor a rendszer az állapotváltozós leírásával jellemezhető, viszont léteznek olyan rendszerek is, amelyek nem írhatóak le ezzel.

Az állapotváltozós leírás egy differenciálegyenlet-rendszert reprezentál, amely pontosan meghatározza a hálózatot. Mivel az átviteli függvény a nemlinearitás miatt nem használható, ezért az állapotváltozós leírás segítségével kell modellezni a hálózatot. Ebben

az esetben a nemlineáris függvény az állapotváltozós leírásban fog megjelenni.

$$\dot{\mathbf{w}}(t) = \mathbf{A}\mathbf{w}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{C}\mathbf{f}(\mathbf{w}(t), \mathbf{u}(t)) \quad (2.20)$$

ahol \mathbf{A} az állapotmátrix, \mathbf{w} az állapotvektor, \mathbf{u} a gerjesztés és \mathbf{f} a nemlinearitást leíró függvény.

Leggyakrabban a megoldáshoz az Euler-módszerek és a Runge-Kutta módszerek használatosak [2, 3, 4].

2.2.1. Előrelépő Euler-módszer

Az előrelépő Euler-módszer használatánál az egyenletrendszer gyakran egyszerűen megoldható lesz, átrendezés után az előző állapotokat behelyettesítve kapható meg a következő állapot.

Diszkrétizáláshoz a (2.8) egyenletet használandó, ezt behelyettesítve, és rendezve:

$$\mathbf{w}[n+1] = \mathbf{A}'\mathbf{w}[n] + \mathbf{B}'\mathbf{u}[n] + \mathbf{C}'\mathbf{f}(\mathbf{w}[n], \mathbf{u}[n]) \quad (2.21)$$

Ezt átrendezve:

$$\mathbf{w}[n] = \mathbf{A}'\mathbf{w}[n-1] + \mathbf{B}'\mathbf{u}[n-1] + \mathbf{C}'\mathbf{f}(\mathbf{w}[n-1], \mathbf{u}[n-1]) \quad (2.22)$$

Az átalakítások után sok esetben csak a korábbi értékeinket be kell helyettesítenünk az explicit egyenletbe a következő minta számolásához.

Ahogy a 2.1.1 alfejezetben látható, lehetséges, hogy a diszkrétizált rendszer instabil lesz. Ez a mintavételi frekvenciától függ, és gyakran nagyon nagy mintavételi frekvencia esetén lesz csak stabil a rendszer. Ez a feltétel valós idejű szimulációk során általában nem teljesíthető, ezért az előrelépő Euler-módszer ritkán használatos valós idejű szimulációra.

2.2.2. Hátralépő Euler-módszer

A hátralépő Euler-módszer használatával a stabilitási kérdés megoldódik (ld. 2.1.2 alfejezet), viszont új problémák merülnek fel. Az állapotváltozós leírás transzformálása és rendezése ugyanis implicit egyenletrendszerhez vezet.

$$\mathbf{w}[n] = \mathbf{A}'\mathbf{w}[n-1] + \mathbf{B}'\mathbf{u}[n] + \mathbf{C}'\mathbf{f}(\mathbf{w}[n], \mathbf{u}[n]) \quad (2.23)$$

Látható, hogy az egyenletrendszer bal és jobb oldalán is (jobb oldalon a függvény argumentumaként) megtalálható a $\mathbf{w}[n]$ állapotváltozó. Az implicit egyenlet valamely zérushely-kereső algoritmussal megoldandó (intervallumfelezés, Newton-módszer, szelő-módszer, stb. [5]). A nehézségeket ellensúlyozza az, hogy az implicit egyenlet megoldása révén egy stabil folytonos idejű rendszerből stabil diszkrét idejű rendszer keletkezik.

2.2.3. Negyedfokú Runge-Kutta módszer

A Runge-Kutta módszereket gyakran alkalmazzák nemlineáris rendszerek offline szimulációjára. Alapötletük hasonló az Euler-módszerekéhez, azonban itt több pontból közelítik a deriváltat. Ha a cél a rendszer nagyon pontos modellezése, akkor a Runge-Kutta módszerek jobb eredményeket nyújtanak, mint az Euler-módszerek. Ugyanakkor a számítási igényük nagyobb, és nem egyszerű a valós idejű alkalmazásuk, mivel a számításához szükséges gerjesztést olyan pontokban is ismerni kell, ahol ez általában egy valós idejű alkalmazás esetén nem ismert. A leggyakrabban alkalmazott változat a negyedfokú Runge-Kutta (RK4) módszer [3, 4].

A negyedfokú Runge-Kutta (RK4) módszerre a matematikusok a leghatékonyabb numerikus közönséges differenciálegyenlet-megoldó módszerként hivatkoznak [6], mivel a magasabb fokú változatok csak kis mértékben növelik a pontosságot a hozzáadott nagy számítási igény mellett. A módszer alapfeltétele, hogy a nemlineáris differenciálegyenlet ismert legyen, és a kezdeti értékek adottak legyenek.

$$\begin{cases} \dot{\mathbf{w}} = f(\mathbf{w}(t), \mathbf{u}(t)) \\ \mathbf{w}(t_0) = \mathbf{w}_0 \end{cases} \quad (2.24)$$

Ekkor legyen $h = T$, és felírható:

$$\begin{cases} w[n+1] = w[n] + u[n] + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4) \\ K_1 = f(w[n], u(nT)) \\ K_2 = f(w[n] + h\frac{K_1}{2}, u(nT + \frac{h}{2})) \\ K_3 = f(w[n] + h\frac{K_2}{2}, u(nT + \frac{h}{2})) \\ K_4 = f(w[n] + hK_3, u(nT + h)) \end{cases} \quad (2.25)$$

Ebben az egyenletrendszerben:

- K_1 a függvény meredeksége az intervallum kezdetén y alapján számítva
- K_2 a függvény meredeksége a intervallum közepén y és K_1 alapján számolva
- K_3 a függvény meredeksége a intervallum közepén y és K_2 alapján számolva
- K_4 a függvény meredeksége a intervallum végén y és K_3 alapján számolva

Mint látható, K_2 és K_3 számolásához szükség lenne a gerjesztés két mintavételezési pont közötti értékére. Ez valós időben nem, vagy csak nehezen alkalmazható.

Digitális jelfeldolgozásban gyakran a bemeneti jel és a kimeneti jelnek követelmény, hogy ugyanannyi mintavételi pontból álljanak. A negyedfokú Runke-Kutta módszer alkalmazásához a bemenet mintavételi frekvenciájának kétszer nagyobbának kellene lennie. Erre egy lehetőség a köztes pont kiszámolása interpolációval, viszont növeli a számításigényt igényel.

3. fejezet

K-módszer

A K-módszer szétválasztja a rendszer lineáris részét és a késleltetés nélküli utat tartalmazó részt. Ezután a késleltetés nélküli részt geometriai módszerekkel oldja meg.

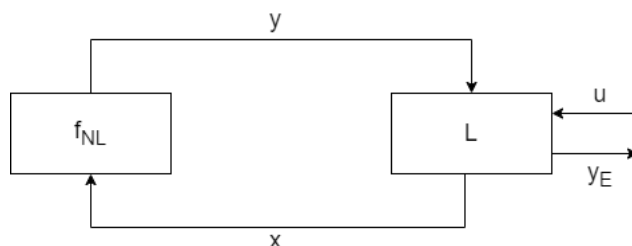
3.1. A K-módszer alap gondolata

A K-módszer alapfeltevése, hogy a modellezendő rendszer felbontható egy lineáris dinamikus és egy nemlineáris memória nélküli részre [7, 8, 9]. Ennek szemléltetése a 3.1 ábrán látható.

$$\begin{cases} \mathbf{x}(t) = \mathbf{L}(\mathbf{y}(t), \mathbf{u}(t)) \\ \mathbf{y}(t) = \mathbf{f}_{NL}(\mathbf{x}(t)) \end{cases} \quad (3.1)$$

ahol $\mathbf{x}(t)$ a lineáris dinamikus rendszer kimenete és egyben a nemlineáris memória nélküli rész bemenete, $\mathbf{y}(t)$ pedig a nemlineáris memória nélküli rész kimenete és egyben a lineáris dinamikus rendszer egyik bemenete.

Így, hogy a rendszer két részre van bontva, azokat a bemeneteket és kimeneteket fogják érinteni a számítási problémák, amelyek függenek a nemlineáris résztől.

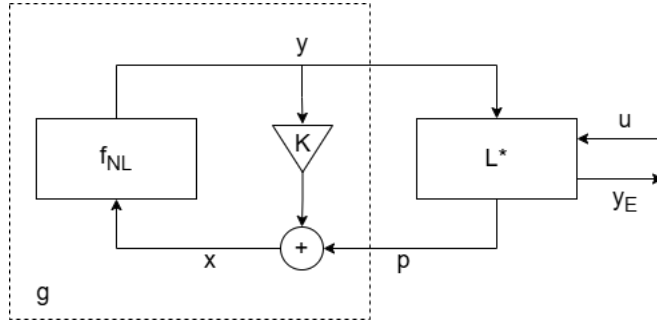


3.1. ábra. A K-módszer alapfeltétele; \mathbf{u} a bemenet, és \mathbf{y}_E a kimenet a külvilág felé. \mathbf{L} és \mathbf{f}_{NL} között késleltetés nélküli hurok van, hiszen az \mathbf{y} változó kiszámolásához szükség van az \mathbf{x} változóra, és az \mathbf{x} változó kiszámításához szükség van az \mathbf{y} változóra.

Ha a rendszert a hátralépő Euler-módszer, akár a bilineáris transzformáció segítségével diszkrétizáljuk, akkor a következő összefüggéshez jutunk [7]:

$$\mathbf{x}[n] = \mathbf{K}\mathbf{y}[n] + \mathbf{p}[n] \quad (3.2)$$

ahol a rendszernek M darab „belső” bemenete ($y_1 \dots y_M$), P darab „külső” bemenete ($u_1 \dots u_P$) és N darab „belső” kimenete ($x_1 \dots x_N$) van.

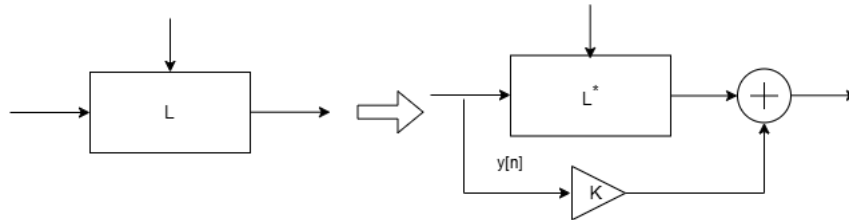


3.2. ábra. A rendszer felbontva; \mathbf{L}^* -nak az \mathbf{y} bemenete és \mathbf{p} kimenete között nincs késleltetés nélküli összeköttetés. A szaggatott vonallal bekeretezett rész fog megfelelni $\mathbf{g}(\mathbf{p})$ -nek.

Ebben az összefüggésben \mathbf{p} egy \mathbf{L}^* rendszer kimenete, amelyet csak a definiált (\mathbf{u}) és a múltbéli jelek határoznak meg. Ekkor felírható a rendszer az alábbi formában [7], amit a 3.2 ábra szemléltet:

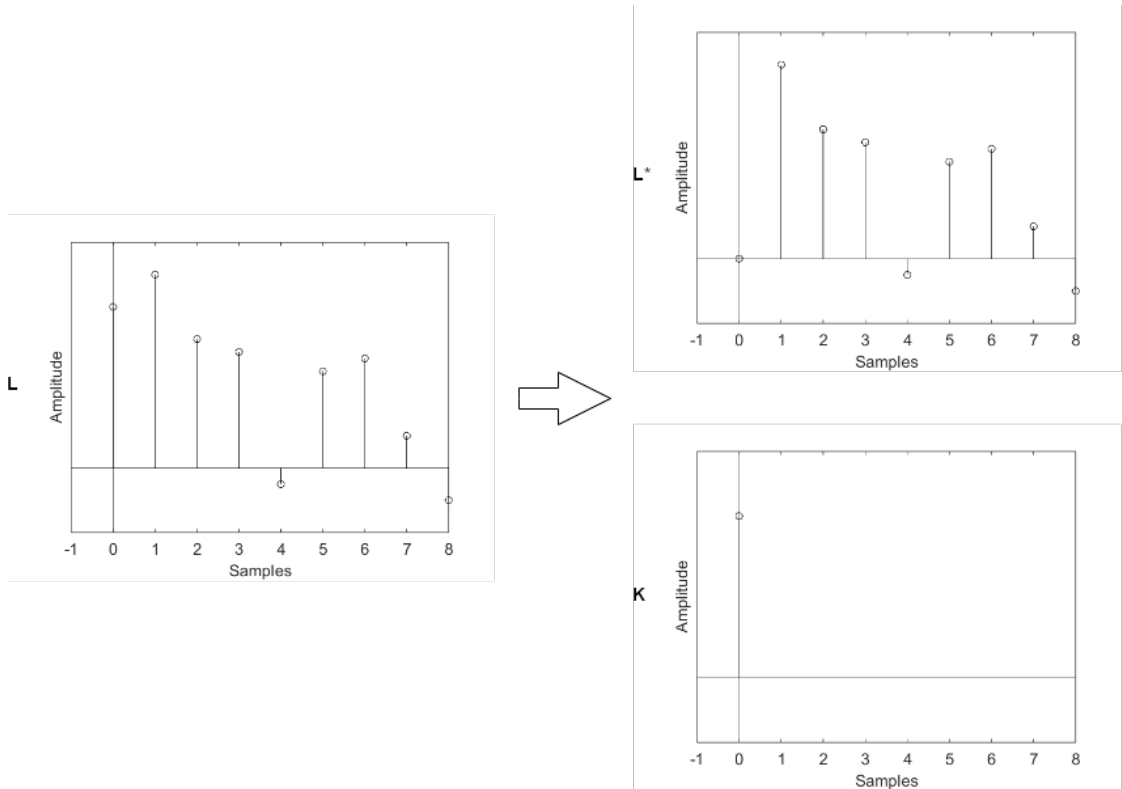
$$\begin{cases} \mathbf{p}[n] = \mathbf{L}^*(\mathbf{u}[n], \mathbf{y}[n]) \\ \mathbf{y}[n] = \mathbf{f}_{NL}(\mathbf{p}[n] + \mathbf{K}\mathbf{y}[n]) \end{cases} \quad (3.3)$$

Tulajdonképpen az \mathbf{L} lineáris rendszernek az egyenletében lévő \mathbf{y} aktuális értékét reprezentáló változót különválasztjuk, és hozzávesszük a nemlineáris rendszerhez, ezzel eliminálva a késleltetés nélküli hurkot. Ezt azért tehetjük meg, mert \mathbf{L} lineáris, így az \mathbf{y} aktuális értéke csak egy konstans \mathbf{K} szorzóval van jelen a kimenetében (3.2). A különválasztást a 3.3 ábra mutatja.



3.3. ábra. Az \mathbf{L} lineáris dinamikus rendszer felbontva a K-módszer szerint; lineáris tulajdonsága miatt az $\mathbf{y}[\mathbf{n}]$ kiemelhető a rendszerből.

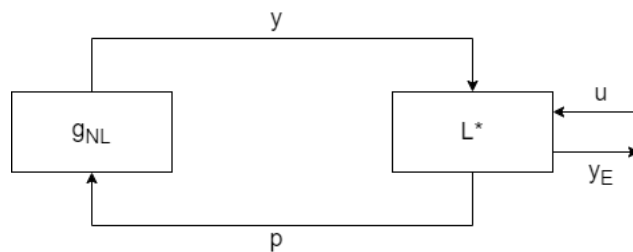
Egy másik értelmezés, hogy az \mathbf{L} rendszernek az \mathbf{y} változóra vonatkozó impulzusválaszának első és a többi értékét különválasztjuk. Ezt ábrázolja a 3.4 ábra. Mivel \mathbf{L} egy lineáris MIMO (Multiple Input Multiple Output) rendszer, így a kimenetei a bemenetek superpozíciójával kiszámolható. Az $\mathbf{u} = 0$ esetén megkapható az \mathbf{y} bemenethez és \mathbf{x} kimenethez tartozó impulzusválasz. Ennek az impulzusválasznak a legelső mintáját és a többi mintáját szétválasztva eliminálható a késleltetésmentes hurok.



3.4. ábra. Az **L** lineáris dinamikus rendszer felbontva a **K**-módszer szerint; Az impulzusválaszt szétválasztva kapható meg az **L*** rendszer és a **K** konstans.

Fontos megjegyezni, hogy az **L*** rendszernek nincs semmilyen késleltetés nélküli kapcsolata **y** és **p** között. Ez látható a 3.5 ábrán. Így a számolási probléma az $\mathbf{y}[n] = \mathbf{f}_{NL}(\mathbf{p}[n] + \mathbf{K}\mathbf{y}[n])$ részre korlátozódik, amely explicit megoldásához a Dini-tételt alkalmazandó, vagy megoldható valamelyik numerikus módszerrel.

Amennyiben az $\mathbf{y} = \mathbf{g}_{NL}(\mathbf{p})$ explicit függvény létezik és meghatározható, a számítási probléma megszűnik.



3.5. ábra. A rendszerből az átalakítások miatt megszűnt a késleltetés nélküli rész; ebben a formában már nincs számítási probléma.

3.2. A K-módszer elemzése

Egy általános kauzális, idő-invariáns rendszernek az általános állapotváltozós leírása az alábbi alakban felírható [7, 8, 9]:

$$\begin{cases} \dot{\mathbf{w}} = \mathbf{A}\mathbf{w} + \mathbf{B}\mathbf{u} + \mathbf{C}\mathbf{y} & (3.4a) \\ \mathbf{x} = \mathbf{D}\mathbf{w} + \mathbf{E}\mathbf{u} + \mathbf{F}\mathbf{y} & (3.4b) \\ \mathbf{y} = \mathbf{f}(\mathbf{x}) & (3.4c) \end{cases}$$

ahol \mathbf{w} az állapotvektor, \mathbf{u} a gerjesztés és \mathbf{f} a nemlineáris részt leíró függvény. Laplace-transzformálva, majd bilineáris transzformációt alkalmazva (3.4a) egyenletre [7]:

$$s\mathbf{W}(s) = \mathbf{A}\mathbf{W}(s) + \mathbf{B}\mathbf{U}(s) + \mathbf{C}\mathbf{Y}(s) \quad (3.5)$$

$$h \frac{1 - z^{-1}}{1 + z^{-1}} \mathbf{W}(s) = \mathbf{A}\mathbf{W}(s) + \mathbf{B}\mathbf{U}(s) + \mathbf{C}\mathbf{Y}(s) \quad (3.6)$$

ahol $h = \frac{2}{T}$. Ezt rendezve:

$$h(1 - z^{-1})\mathbf{W}(z) = \mathbf{A}(1 + z^{-1})\mathbf{W}(z) + (1 + z^{-1})(\mathbf{B}\mathbf{U}(z) + \mathbf{C}\mathbf{Y}(z)) \quad (3.7)$$

$$h\mathbf{W}(z) - \mathbf{A}\mathbf{W}(z) = z^{-1}h\mathbf{W}(z) + z^{-1}\mathbf{A}\mathbf{W}(z) + \mathbf{B}\mathbf{U}(z) + \mathbf{C}\mathbf{Y}(z) + z^{-1}(\mathbf{B}\mathbf{U}(z) + \mathbf{C}\mathbf{Y}(z)) \quad (3.8)$$

$$(h\mathbf{I} - \mathbf{A})\mathbf{W}(z) = z^{-1}(h\mathbf{I} + \mathbf{A})\mathbf{W}(z) + \mathbf{B}\mathbf{U}(z) + z^{-1}\mathbf{B}\mathbf{U}(z) + \mathbf{C}\mathbf{Y}(z) + z^{-1}\mathbf{C}\mathbf{Y}(z) \quad (3.9)$$

Inverz z-transzformálva:

$$(h\mathbf{I} - \mathbf{A})\mathbf{w}[n] = (h\mathbf{I} + \mathbf{A})\mathbf{w}[n - 1] + \mathbf{B}\mathbf{u}[n] + \mathbf{B}\mathbf{u}[n - 1] + \mathbf{C}\mathbf{y}[n] + \mathbf{C}\mathbf{y}[n - 1] \quad (3.10)$$

Ha $(h\mathbf{I} - \mathbf{A})$ invertálható (ami igaz lesz mindig, ha h az \mathbf{A} mátrix sajátértékeitől különbözik [7]), akkor balról az inverzével beszorozva a két oldalt:

$$\mathbf{w}[n] = (h\mathbf{I} - \mathbf{A})^{-1}((h\mathbf{I} + \mathbf{A})\mathbf{w}[n - 1] + \mathbf{B}\mathbf{u}[n] + \mathbf{B}\mathbf{u}[n - 1] + \mathbf{C}\mathbf{y}[n] + \mathbf{C}\mathbf{y}[n - 1]) \quad (3.11)$$

A könnyebb felírás érdekében új segédváltozók bevezetése ajánlott.

$$\mathbf{G} \triangleq (h\mathbf{I} - \mathbf{A})^{-1}\mathbf{C} \quad (3.12)$$

$$\mathbf{H} \triangleq (h\mathbf{I} - \mathbf{A})^{-1}(h\mathbf{I} + \mathbf{A}) \quad (3.13)$$

$$\mathbf{J} \triangleq (h\mathbf{I} - \mathbf{A})^{-1}\mathbf{B} \quad (3.14)$$

Ezeket behelyettesítve:

$$\mathbf{w}[n] = \mathbf{H}\mathbf{w}[n - 1] + \mathbf{J}(\mathbf{u}[n] + \mathbf{u}[n - 1]) + \mathbf{G}\mathbf{y}[n - 1] + \mathbf{G}\mathbf{y}[n] \quad (3.15)$$

Egy újabb segédváltozó bevezetése:

$$\hat{\mathbf{p}} \triangleq \mathbf{H}\mathbf{w}[n - 1] + \mathbf{J}(\mathbf{u}[n] + \mathbf{u}[n - 1]) + \mathbf{G}\mathbf{y}[n - 1] \quad (3.16)$$

Ezt behelyettesítve:

$$\mathbf{w}[n] = \hat{\mathbf{p}}[n] + \mathbf{G}\mathbf{y}[n] \quad (3.17)$$

Az átalakítások segítségével a (3.4b) egyenlet felírható:

$$\mathbf{x}[n] = \mathbf{D}\hat{\mathbf{p}}[n] + \mathbf{D}\mathbf{G}\mathbf{y}[n] + \mathbf{E}\mathbf{u}[n] + \mathbf{F}\mathbf{y}[n] = \mathbf{p}[n] + \mathbf{K}\mathbf{y}[n] \quad (3.18)$$

ahol

$$\mathbf{K} \triangleq \mathbf{D}\mathbf{G} + \mathbf{F} \quad (3.19)$$

$$\mathbf{p}[n] \triangleq \mathbf{D}\hat{\mathbf{p}}[n] + \mathbf{E}\mathbf{u}[n] \quad (3.20)$$

Összességében [7]:

$$\begin{cases} \mathbf{p}[n] = \mathbf{D}\mathbf{H}\mathbf{w}[n-1] + \mathbf{D}\mathbf{J}(\mathbf{u}[n] + \mathbf{u}[n-1]) + \mathbf{D}\mathbf{G}\mathbf{y}[n-1] + \mathbf{E}\mathbf{u}[n] \\ \mathbf{y}[n] = \mathbf{f}_{NL}(\mathbf{p}[n] + \mathbf{K}\mathbf{y}[n]) \\ \mathbf{w}[n] = \mathbf{H}\mathbf{w}[n-1] + \mathbf{J}(\mathbf{u}[n] + \mathbf{u}[n-1]) + \mathbf{G}(\mathbf{y}[n] + \mathbf{y}[n-1]) \end{cases} \quad (3.21)$$

A (3.21) egyenletrendszerben az egyetlen problémát okozó rész a második egyenlet. Ezen kívül minden változó kiszámolható, hiszen $\mathbf{p}[n]$ csak a definiált és a múltbéli változóértékektől függ, a $\mathbf{w}[n]$ egyenletébe szintén csak be kell helyettesíteni az $\mathbf{y}[n]$ kiszámolása után.

Érdemes megjegyezni, hogy ezt a módszert másféle folytonos időből diszkrét időbe transzformációval is lehet vezetni (például hátralépő Euler-módszerrel, ld. F.1 függelék) [7].

3.3. A nemlineáris rész

Ahogy korábban említettük, a számítási probléma a (3.21) második egyenletére redukálódik:

$$\mathbf{y}[n] = \mathbf{f}_{NL}(\mathbf{p}[n] + \mathbf{K}\mathbf{y}[n]) \quad (3.22)$$

Az egyenletek átrendezésének célja az, hogy meg lehessen vizsgálni, hogy teljesül-e rá a Dini-tétel. Ha teljesül, akkor található legyen egy olyan $g(p)$ implicit egyenlet, ami meg egyezik az $f_{NL}(x)$ függvénnyel.

3.3.1. Dini-tétel

A Dini-tételnek csak a SISO rendszerekre vonatkoztatott alakját tárgyalja ez a dolgozat (MISO és MIMO rendszerekre is meg lehet fogalmazni [7]). Adott egy implicit függvény:

$$f(x, y) = 0 \quad (3.23)$$

Ha ennek a függvénynek létezik olyan pontja, amelyre igaz egy $P_0(x_0, y_0)$ pontban, hogy

$$\left. \frac{\partial f}{\partial y} \right|_{(x_0, y_0)} \neq 0 \quad (3.24)$$

akkor létezik olyan $g(x)$ függvény, amelyre

$$f(x, g(x)) = 0 \quad (3.25)$$

igaz a P_0 pont körül. Ha (3.24) nem csak P_0 -ra, hanem az egész $f(x, y)$ függvényre igaz, akkor $f(x, y)$ nem csak lokálisan, hanem globálisan is kifejezhető $f_{NL}(x, g(x))$ függvénnyel [7].

Összefoglalva a Dini-tétel azt mondja ki, hogy ha egy implicit $y = f(x, y)$ függvényre teljesül (3.24), akkor létezik $g(x) = f(x, y)$ explicit függvény.

3.3.2. A Dini-tétel alkalmazása a K-módszerben

Ebben az esetben az $f(x, y)$ függvény a (3.22) nullára rendezett alakja:

$$f_{NL}(p + ky) - y = 0 \quad (3.26)$$

Ha az y szerinti parciális deriváltjára adott a feltétel, azaz

$$\frac{\partial f}{\partial y} = f'_{NL}(p + ky)k - 1 \neq 0 \quad (3.27)$$

$$f'_{NL}(p + ky) \neq \frac{1}{k} \quad (3.28)$$

akkor létezik $g(p)$, amelyre

$$g(p) = y \quad (3.29)$$

Mivel olyan $g(p)$ függvényt keresünk, amely megegyezik a $f(p + ky)$ függvénnyel, ezért geometriai megfontolások miatt [7] a $g(p)$ függvény az $f(x)$ függvény lineáris transzformációjával kapható meg, mégpedig a

$$\begin{cases} y = y_s \\ p = x - ky \end{cases} \quad (3.30)$$

lineáris transzformáció segítségével, ami az (x, y) síkból a (p, y_s) síkba transzformál. A transzformációt a 3.6 ábra szemlélteti.

A $g(p)$ függvény csak úgy lehet egyenlő az $f(p + ky)$ függvénnyel, ha az egész p tengelyt eltoljuk ky -nal. Így minden p helyhez az az érték fog kerülni, ami eredetileg a $p + ky$ helyen volt (azaz a $g(p)$ függvényben minden p helyen az az érték található, ami az $f(p + ky)$ függvényben a $p + ky$ helyen). Ezzel a lineáris transzformációval ez az implicit függvény kifejezhető egy explicit függvénnyel, ha teljesül a Dini-tétel.

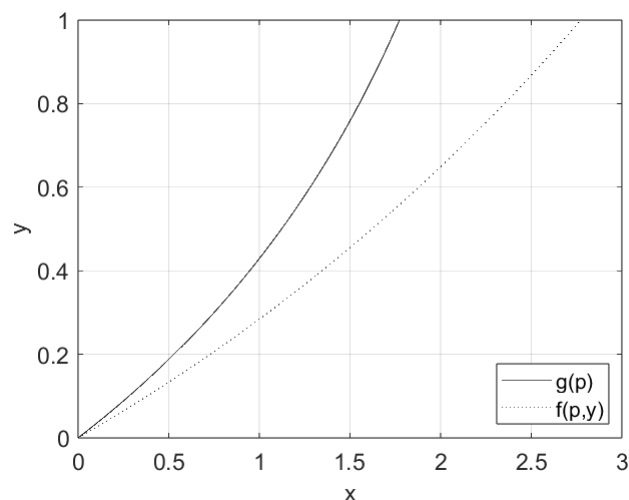
3.4. A K-módszer alkalmazása

A korábban már levezetett (3.21) szerint:

$$\begin{cases} \mathbf{p}[n] = \mathbf{D}(\mathbf{H}\mathbf{w}[n-1] + \mathbf{J}(\mathbf{u}[n] + \mathbf{u}[n-1])) + \mathbf{G}\mathbf{y}[n-1] + \mathbf{E}\mathbf{u}[n] \\ \mathbf{y}[n] = \mathbf{f}_{NL}(\mathbf{p}[n] + \mathbf{K}\mathbf{y}[n]) \\ \mathbf{w}[n] = \mathbf{H}\mathbf{w}[n-1] + \mathbf{J}(\mathbf{u}[n] + \mathbf{u}[n-1]) + \mathbf{G}\mathbf{y}[n-1] + \mathbf{G}\mathbf{y}[n] \end{cases} \quad (3.31)$$

A korábban bevezetett változók felhasználásával megfogalmazott egyenleteket egymás után megoldva a nemlineáris rendszer egyenleteinek megoldásához vezet [7].

$$\begin{cases} \hat{\mathbf{p}}[n] = \mathbf{H}\mathbf{w}[n-1] + \mathbf{J}(\mathbf{u}[n] + \mathbf{u}[n-1]) + \mathbf{G}\mathbf{y}[n-1] \\ \mathbf{p}[n] = \mathbf{D}\hat{\mathbf{p}}[n] + \mathbf{E}\mathbf{u}[n] \\ \mathbf{y}[n] = \mathbf{g}(\mathbf{p}[n]) \\ \mathbf{w}[n] = \mathbf{G}\mathbf{y}[n] + \hat{\mathbf{p}}[n] \end{cases} \quad (3.32)$$



3.6. ábra. Olyan $g(x)$ explicit függvényt keresünk, amely minden p -re megegyezik az $f(p, y)$ függvénnyel. Mivel az $f(p, y) = f(p + Ky)$, azért $g(p)$ meghatározásához az $f(p, y)$ függvényt el kell tolni úgy, hogy minden p -hez Ky -nal kevesebb érték tartozzon.

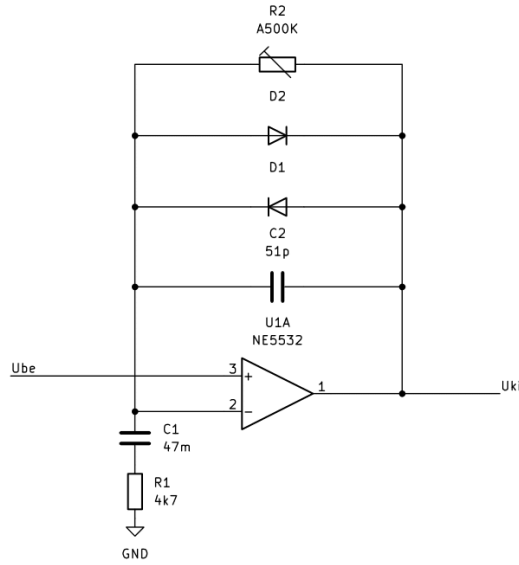
A (3.32) egyenletrendszer harmadik egyenlete például egy előre legenerált táblázat segítségével megoldható, amely a (p, y) értékpárokat tárolja. Ezek közül az értékpárok közvetlenül kikereshetőek. Ez a lépés a K-módszer hatékonyságának az alapja, mivel így nem szükséges egy implicit egyenletrendszert valós időben megoldani; elegendő az előre legenerált táblázatból kikeresni az eredményt.

Összefoglalva az algoritmus lépéseit [9]:

1. $\hat{\mathbf{p}}[n]$ kiszámolása (3.16) szerint
2. $\mathbf{p}[n]$ kiszámítása $\hat{\mathbf{p}}[n]$ és $\mathbf{u}[n]$ felhasználásával
3. $\mathbf{y}[n]$ kiszámítása a letárolt táblázatban való kereséssel
4. Az állapotváltozók (\mathbf{w}) frissítése (3.17) szerint
5. A kimenet kiszámolása $\mathbf{w}[n]$, $\mathbf{u}[n]$, $\mathbf{y}[n]$ segítségével

3.5. Példa

A módszer bemutatása céljából az Ibanez Tube-Screamer gitártorzító pedál áramkörének a „clipping stage” részét valósítottam meg K-módszer segítségével.



3.7. ábra. A Tube-Screamer áramkör „clipping stage” részének áramköre

3.5.1. Megvalósítás

Először meg kell határozni a rendszer állapotváltozós leírását. Az állapotváltozók a kondenzátorok feszültségei lesznek, a diódákat érdemes összevonni, és egy nemlineáris elemként kezelni. A nemlineáris függvény pedig a két dióda karakterisztika összege lesz. A Kirchhoff-törvényekből felírt egyenleteket átrendezve:

$$\begin{aligned}
 u_{C1} \dot{C}_1 &= -\frac{1}{R_1 C_1} u_{C1} + \frac{1}{R_1 C_1} u_{be} \\
 u_{C2} \dot{C}_2 &= -\frac{1}{C_2 R_1} u_{C1} - \frac{1}{C_2 R_2} u_{C2} + \frac{1}{C_2 R_1} u_{be} - \frac{1}{C_2} f(u_{C2})
 \end{aligned} \tag{3.33}$$

$$f(u) = I_{S0} \left(e^{\frac{u}{u_T}} - e^{\frac{-u}{u_T}} \right)$$

$$u_{ki} = u_{C2} + u_{be}$$

A K-módszerhez szükséges mátrix alakban felírva:

$$\begin{cases}
 \dot{\mathbf{w}} = \begin{pmatrix} -\frac{1}{R_1 C_1} & 0 \\ -\frac{1}{C_2 R_1} & -\frac{1}{C_2 R_2} \end{pmatrix} \mathbf{w} + \begin{pmatrix} \frac{1}{R_1 C_1} \\ \frac{1}{C_2 R_1} \end{pmatrix} u + \begin{pmatrix} 0 \\ -\frac{1}{C_2} \end{pmatrix} y \\
 x = \begin{pmatrix} 0 & 1 \end{pmatrix} \mathbf{w} \\
 y = f(x)
 \end{cases} \tag{3.34}$$

Az ebből meghatározott mátrixok:

$$\mathbf{A} = \begin{pmatrix} -\frac{1}{R_1 C_1} & 0 \\ -\frac{1}{C_2 R_1} & -\frac{1}{C_2 R_2} \end{pmatrix} \quad \mathbf{B} = \begin{pmatrix} \frac{1}{R_1 C_1} \\ \frac{1}{C_2 R_1} \end{pmatrix} \quad \mathbf{C} = \begin{pmatrix} 0 \\ -\frac{1}{C_2} \end{pmatrix} \quad (3.35)$$

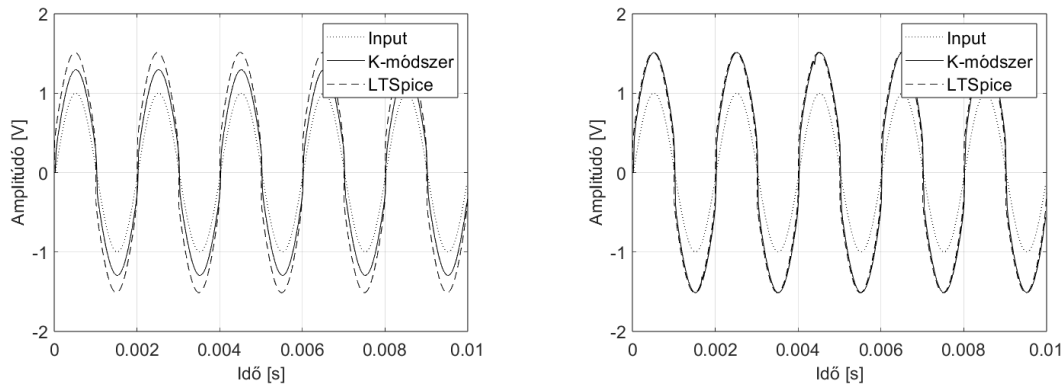
$$\mathbf{D} = \begin{pmatrix} 0 & 1 \end{pmatrix} \quad E = 0 \quad F = 0$$

A segédváltozókat kiszámoltam a (3.12), (3.13), (3.14), (3.19) egyenletekkel.

Legeneráltam az $f(u)$ függvény táblázatát, majd végrehajtottam rajta az $y_s = y$, $p = x - Ky$ lineáris transzformációt. Lehetőség szerint érdemes kizárólag a valóságban előforduló két szélsőérték határán belüli értékeket legenerálni, továbbá, ha a függvény szimmetrikus az y tengelyre nézve, akkor elegendő csupán az egyik oldal értékeit tárolni (ennél a példánál ez éppen lehetséges). Ezt követően már csak a (3.32) pontban leírt algoritmust kell futtatni.

3.5.2. Eredmény

Az áramkört az LTSpice programban is elkészítettem összehasonlítás céljából, és teszteltem a K-módszert a fent említett függvénnyel, illetve LTSpice-ből kiexportált karakterisztikával. Ezt szemlélteti a 3.8 ábra.



(a) A (3.33) egyenletben leírt karakterisztikával (b) LTSpice-ből exportált karakterisztikával

3.8. ábra. A megvalósított modell válaszjele egy 500 Hz frekvenciájú, 1 V amplitúdójú szinuszjelre

A (3.33) egyenletrendszerben említett karakterisztikával az eredmény eltér az LTSpice program által meghatározott kimenettől, ami a 3.8a ábrán látható. A jelalakok hasonlóak, viszont amplitúdóban különböznek. Ez azért van, mert az LTSpice nem ugyanezzel a karakterisztikával számol. Erre abból lehet következtetni, hogy ha az LTSpice-ből exportáljuk a karakterisztikánkat, akkor a kimenetek nagyon hasonlóak lesznek, ahogy látható a 3.8b ábrán is.

4. fejezet

A K-módszer összehasonlítása más állapotteres módszerekkel

Áramkör-szimulációt többféle módon is meg lehet közelíteni. Ezek a megközelítések három nagy csoportba sorolhatóak: állapotteres módszerek, hullámdigitális szűrő módszer és a fekete doboz módszerek [10]. Ebben a fejezetben az állapotteres módszerek közül a K-módszert és a valós idejű szimulációnál leggyakrabban használt állapotteres módszereket fogom összehasonlítani.

4.1. Az állapotteres módszerek

Az állapotteres módszerek kiindulási alapja az áramkör állapotváltozós leírása:

$$\dot{\mathbf{w}} = \mathbf{A}\mathbf{w}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{C}\mathbf{f}(\mathbf{D}\mathbf{w}(t) + \mathbf{E}\mathbf{u}(t) + \mathbf{F}\mathbf{f}(\dots)) \quad (4.1)$$

Az egyenletrendszerben a \mathbf{w} az állapotvektor, \mathbf{u} a gerjesztés, \mathbf{f} pedig egy nemlineáris függvény. Bizonyos áramkörök állapotváltozós leírásában megfigyelhető a nemlineáris függvény egymásbaágyazódása. Ez általában bonyolultabb áramkörökben fordul elő, ahol visszacsatolás van jelen, és annak része a nemlineáris rész. Ezt az egymásbaágyazódást jelöli a (4.1) egyenletben az $\mathbf{f}(\dots)$ változó, aminek az argumentumában ugyanazok a változók találhatóak, mint a külső függvény argumentumában.

Ezután diszkretizálásra van szükség: a legtöbb módszer a deriváltat egyből helyettesíti az azt közelítő kifejezéssel, más módszerek az általuk definiált módon átírják az egyenletet.

Leggyakrabban az állapotteres módszerek a diszkretizálásban fognak különbözni, hiszen kulcsfontosságú, hogy hogyan fogják a deriváltat közelíteni. Ez fogja meghatározni a diszkrét rendszer stabilitását, befolyásolja a számítási igényt (a kifejezés bonyolultsága), valamint a hiba nagyságát, amellyel eltér a folytonos idejű rendszertől.

4.1.1. Euler-módszerek

Az előre- és hátralépő Euler-módszereket gyakran alkalmazzák a mindennapi gyakorlatban [2]. Valós idejű szimuláció esetén leggyakrabban a hátralépő Euler-módszert célszerű alkalmazni, mivel az előrelépő Euler-módszer stabilitási feltételének (2.1.1) teljesítéséhez legtöbbször nagy mintavételi frekvencia szükséges.

A hátralépő Euler-módszerben a diszkretizálás után az egyenlet mindig implicit lesz, így azt valamilyen numerikus módszerrel kell megoldani (2.1.2). Ekkor az egyenlet alakja:

$$\mathbf{w}[n] = \mathbf{H}\mathbf{w}[n-1] + \mathbf{J}\mathbf{u}[n] + \mathbf{G}\mathbf{f}(\mathbf{D}\mathbf{w}[n] + \mathbf{E}\mathbf{u}[n] + \mathbf{F}\mathbf{f}(\dots)) \quad (4.2)$$

Látható, hogy az egyenlet implicit, hiszen az egyenlet bal oldalán és a függvény argumentumaként is szerepel $\mathbf{w}[n]$.

4.1.2. K-módszer

A K-módszer a lineáris dinamikus, és a memória nélküli nemlineáris rész szétválasztásával kezd, függetlenül az \mathbf{F} változótól. Szétválasztás után ugyanúgy a nemlineáris rész kimenetét számolja ki először, de ehhez nem numerikus zérushelykereső módszert használ, hanem alkalmazza a Dini-tételt [7], ami kimondja, hogy egy implicit függvényhez található egy olyan explicit függvény, ami néhány feltétel teljesülése esetén megegyezik vele (3.3.1).

Ezt az explicit függvényt ki lehet offline számolni, letárolni egy tömbben, és így meg lehet oldani az egyenletet egy tömbben való kereséssel a numerikus zérushelykereső módszer helyett.

Érdemes megjegyezni, hogy mivel a Dini-tétel alkalmazásának ugyanaz a feltétele, mint a Newton-módszernek, így csak akkor lehet alkalmazni, ha a Newton-módszerrel is meg lehetne oldani az egyenletet [9].

4.2. Rugalmasabb diszkretizáció a K-módszerben

A K-módszer lehetőséget ad, hogy bármilyen z -beli racionális tört használható legyen a diszkretizáláshoz, és nem definiálja azt úgy, mint az Euler-módszerek.

A K-módszer Laplace-transzformáció után az s tartományból behelyettesítéssel áttérünk a z tartományba, majd rendezés után inverz- z transzformálva megkapjuk a diszkretizált állapotegyenletet. Ezzel lehetőséget ad arra, hogy különböző, akármilyen z -beli polinom/polinom alakú kifejezéssel közelítsük az s változót, ha erre szükség lenne. Laplace-transzformáció esetén a (4.1) egyenletrendszer alakja:

$$s\mathbf{W}(s) = \mathbf{A}\mathbf{W}(s) + \mathbf{B}\mathbf{U}(s) + \mathbf{C}\mathbf{f}(\mathbf{D}\mathbf{W}(s) + \mathbf{E}\mathbf{U}(s) + \mathbf{F}\mathbf{f}(\dots)) \quad (4.3)$$

$$s \cong l(z) \quad (4.4)$$

Ezt behelyettesítve:

$$l(z)\mathbf{W}(z) = \mathbf{A}\mathbf{W}(z) + \mathbf{B}\mathbf{U}(z) + \mathbf{C}\mathbf{f}(\mathbf{D}\mathbf{W}(z) + \mathbf{E}\mathbf{U}(z) + \mathbf{F}\mathbf{f}(\dots)) \quad (4.5)$$

Inverz z -transzformáció és rendezés után az egyenlet alakja:

$$\mathbf{w}[n] = \sum_{i=1}^M \mathbf{H}_i \mathbf{w}[n-i] + \sum_{j=0}^N \mathbf{J}_j \mathbf{u}[n-j] + \sum_{k=0}^P \mathbf{G}_k \mathbf{f}(\mathbf{D}\mathbf{w}[n-\mathbf{k}] + \mathbf{E}\mathbf{u}[n-\mathbf{k}] + \mathbf{F}\mathbf{f}(\dots)) \quad (4.6)$$

4.3. Számítási igény

Valós idejű szimuláció esetén kritikus fontossággal bír a választott módszer számítási igénye, hiszen megadott időkereten belül kell a kimenetet kiszámolni. Általánosan elmondható az állapotteres módszerekről, hogy a számításigényük az állapottér mérete szerint négyzetesen függenek, azaz N állapotváltozóval rendelkező rendszer esetén a módszer számításigénye N^2 [10].

A számítási igények meghatározásához feltételezem, hogy a szorzás és összeadás műveletek számításigénye konstans, ami a korszerű processzorok architektúrájának megfelelően megtehető [11, 12, 13].

4.3.1. Hátralépő Euler-módszer

A hátralépő Euler-módszer számítási igénye a lineáris részt tekintve N^2 , hiszen az N elemű \mathbf{w} állapotvektort minden iteráció során meg kell majd szorozni egy $N \times N$ méretű, \mathbf{H} mátrixszal.

Érdemes ezen felül megvizsgálni az implicit nemlineáris egyenlet megoldásának számítási igényét, hiszen érdemben ebben fog különbözni a többi állapotterres módszertől.

A hátralépő Euler-módszerhez leggyakrabban Newton-módszert szokás használni az implicit nemlineáris egyenlet megoldásához. A Newton-módszer az egyenlet gyökét $\log(n)$ iteráció után találja meg, ha a keresett eredmény n számjegy pontosságú [14]. Másszóval a Newton-módszer futtatása közben a pontosságunk három tizedesjegy, akkor a következő iteráció után hat tizedes jegy.

A Newton-módszer alkalmazásához átrendezett egyenletrendszer:

$$\mathbf{P}(\mathbf{w}[n]) = 0 = -\mathbf{w}[n] + \mathbf{H}\mathbf{w}[n-1] + \mathbf{J}\mathbf{u}[n] + \mathbf{G}\mathbf{f}(\mathbf{D}\mathbf{w}[n] + \mathbf{E}\mathbf{u}[n] + \mathbf{F}\mathbf{f}(\dots)) \quad (4.7)$$

A Newton-módszer alkalmazásához szükség van a (4.7) egyenlet $\mathbf{w}[n]$ szerinti parciális deriváltjára:

$$\mathbf{J}_{\mathbf{P}}(\mathbf{w}[n]) = \frac{\partial \mathbf{P}(\mathbf{w}[n])}{\partial \mathbf{w}[n]} = -\mathbf{I} + \mathbf{G}\mathbf{D} \frac{\partial \mathbf{f}}{\partial \mathbf{w}[n]} \quad (4.8)$$

ahol \mathbf{I} az egységmátrix, $\mathbf{J}_{\mathbf{P}}$ pedig a Jacobi-mátrix.

Ezután minden egyes iterációnál kiszámítandó:

$$\mathbf{w}[n]^{(k+1)} = \mathbf{w}[n]^{(k)} - \mathbf{J}_{\mathbf{P}}(\mathbf{w}[n]^{(k)})^{-1} \mathbf{P}(\mathbf{w}[n]^{(k)}) \quad (4.9)$$

amíg az iteráció el nem éri az eredmény legalább n számjegy pontosságát.

Valós időben a Jacobi-mátrix és inverzének kiszámolása az állapotváltozók számától köbösen (N^3) függ, hiszen minden iteráció során be kell helyettesíteni a Jacobi-mátrixba, aminek $N \times N$ méretű mátrix esetén N^2 a számítási igénye, és ezután azt invertálni kell azt, aminek N^3 a számítási igénye.

Emellett van lehetőség a Jacobi-mátrix inverzének paraméteres kiszámolására:

$$\mathbf{J}_{\mathbf{P}}^{-1} = \frac{\text{Adj}(\mathbf{J}_{\mathbf{P}})}{\det(\mathbf{J}_{\mathbf{P}})} \quad (4.10)$$

Ebbe a mátrixba való behelyettesítés az állapotváltozók számától négyzetesen (N^2) függ.

Összefoglalva, a hátralépő Euler-módszerben a lineáris rész kiszámolása az állapotváltozóktól négyzetesen függ (N^2).

A Newton-módszer során érdemes előre, offline kiszámolni a Jacobi-mátrixot és annak inverzét paraméteresen. Iteráció során szükség van a 4.7 függvény kiértékelésére, ami N állapotváltozó esetén N függvény kiértékelése.

Ezután szükség van az inverz Jacobi-mátrixba való behelyettesítéshez. Mivel a Jacobi-mátrix itt egy $N \times N$ méretű mátrix, így a behelyettesítések és kiértékelések négyzetesen függenek az állapotváltozók számától (N^2). Az inverz Jacobi-mátrixot végül meg kell szorozni a kiértékelt 4.7 N elemű vektorral, ami összességében N^2 művelet.

Így a nemlineáris rész négyzetesen (N^2) függ az állapotváltozók számától.

Ezekhez a számítási igényekhez feltételeztem, hogy függvények kiértékelésének számítási igénye konstans [12].

Érdemes megjegyezni, hogy ez a számítási független a nemlineáris rész bemeneteinek és kimeneteinek a számától.

4.3.2. K-módszer

A K-módszernél a lineáris részek kiszámítása a számításigény szempontjából négyzetesen nő az állapotváltozók számával, mivel minden iterációban egy N méretű \mathbf{w} állapotvektort egy $N \times N$ méretű \mathbf{H} mátrixszal kell szorozni.

A nemlineáris rész megoldása esetén a tömb mérete határozza meg a számításigényt. Mivel a tömböt futtatás előtt generáljuk, lehetőség van azt növekvő sorrendben tárolni, így bináris kereséssel találhatjuk meg a keresett értéket. Ha a tömb mérete m , akkor a legrosszabb esetben $\log_2 m$ iteráció alatt találjuk meg.

Ha a nemlineáris résznek több be- és kimenete van (MISO vagy MIMO), akkor a nemlineáris függvényt mátrixban kell tárolni. A tárolt mátrix dimenziója megegyezik a nemlineáris rész bemeneteinek számával. Ebben a struktúrában való keresés abban tér el a tömbben való kereséstől, hogy annyi bináris keresést kell végrehajtani, ahány bemenete van a nemlineáris résznek.

Ezért a nemlineáris egyenlet megoldása a nemlineáris rész bemeneteinek számától lineárisan függ.

Összefoglalva, a K-módszerben a lineáris rész számításigénye négyzetesen függ az állapotváltozók számától (N^2), míg a nemlineáris rész megoldásának számításigénye a nemlineáris függvény bemeneteinek számától lineárisan függ.

Érdemes megjegyezni, hogy a nemlineáris rész számításigénye független az állapotváltozók számától.

4.4. Az összehasonlítás összefoglalása

A K-módszer előnyei a hátralépő Euler-módszerrel szemben:

- A K-módszerben a nemlineáris egyenlet megoldása csak a nemlineáris rész bemeneteinek számától lineárisan, a hátralépő Euler-módszernél pedig az állapotváltozók számától négyzetesen függ.
- A K-módszer pontosabb, mivel a derivált közelítésében pontosabb eredményt ad, és lehetőséget nyújt magasabb rendű közelítések alkalmazására is.
- A hátralépő Euler-módszer működése az állapotváltozók matematikai leírásától függ, míg a K-módszer mindig egységes sémát alkalmaz.
- A K-módszerben a letárolt mátrix mérete optimalizálható: például y_s tengelyre szimmetrikus függvény esetén elegendő csak a pozitív p értékeket eltárolni. A mátrix méretének csökkentéséből fakadó pontatlanság interpolációval kompenzálható, például lineáris interpoláció esetén a lineáris szakaszokat ritkábban tárolhatjuk.

A hátralépő Euler-módszer előnyei a K-módszerrel szemben:

- A hátralépő Euler-módszer futás közben kevesebb memóriát igényel, mivel a K-módszerhez szükséges letárolt mátrixot a teljes futás idejére memóriában kell tartani.
- Ha a modellezett áramkörben változtatható paraméterek (például potenciométerek, érzékelők stb.) találhatók, akkor a K-módszer esetén minden paraméterváltozáskor újra kell számolni a segédváltozókat és a letárolandó mátrixokat.
- A K-módszer alkalmazása előtt a mátrix generálásakor meg kell határozni a nemlineáris rész reális minimum- és maximumértékeit.

A két módszer közös tulajdonságai:

- Mindkét módszer alkalmazása feltételhez kötött.

A gyakorlatban a K-módszer számításigénye általában kisebb, mint a hátralépő Euler-módszeré. Ennek oka, hogy komplexebb áramkörök esetén a nemlineáris rész bemeneteinek száma jellemzően kevesebb, mint az állapotváltozók száma. Ezért a hátralépő Euler-módszer alkalmazása gyakran nagyobb számításigényt eredményez, mint a K-módszer használata.

5. fejezet

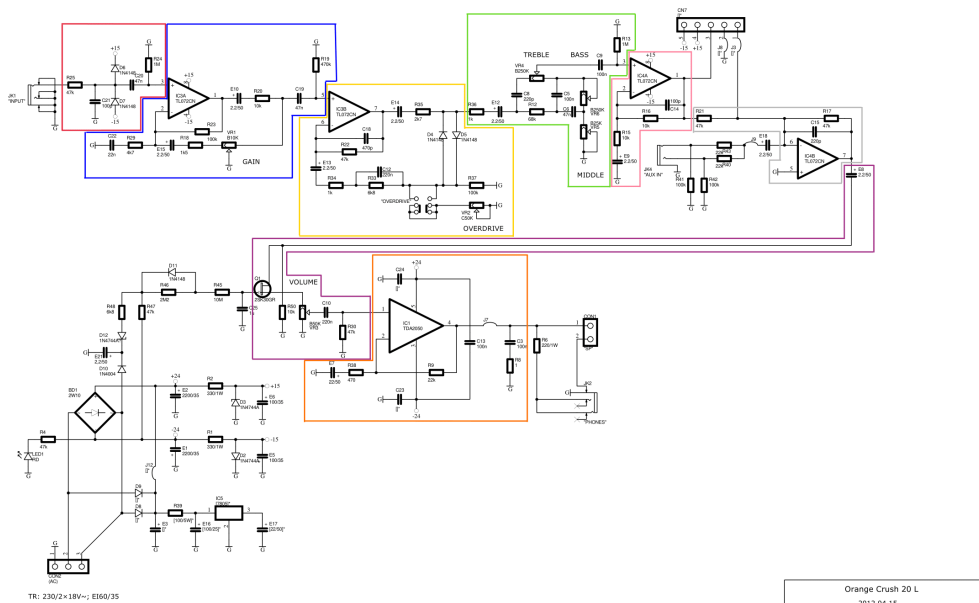
Gitárerősítő modellezése

A dolgozatom egyik célja, hogy az Orange Crush 20L gitárerősítő áramkörét modellezem VST3 plugin formájában. A szimulációt először offline MATLAB környezetben valósítottam meg, ezután írtam meg a valós idejű plugint C++ programnyelven JUCE környezetben. Az áramkört az LTSpice segítségével teszteltem.

A modellezés célja az volt, hogy a szimuláció ugyanolyan hangzásélményt biztosítson, mint a valóságban az analóg áramkör, és nehezen lehessen a kettőt hallgatás esetén megkülönböztetni.

Ideális műveleti erősítőkkel számoltam a MATLAB-ban és a C++-ban programozott szimulációban ott, ahol a valóságnak megfelelő jel esetén nem tapasztaltam levágást. Ahol pedig tapasztaltam, ott egy egyszerű levágó algoritmussal modelleztem ezt a jelenséget [15]. A műveleti erősítők frekvenciafüggő átvitelétől és a slew rate-jétől eltekintettem, hiszen modellezésük ilyen komplex rendszer esetén nem fog nagy hangzásbeli különbséget eredményezni.

Az áramkört a műveleti erősítők bemeneténél és kimeneténél részekre bontottam, hiszen a műveleti erősítőnek nagyimpedanciás a bemenete és kisimpedanciás a kimenete, ami miatt az előtte lévő áramköri elemeket nem terheli meg, illetve az utána lévő áramkörök nem terhelik meg a műveleti erősítő köré épített részt. A szétválasztás az 5.1 ábrán látható.

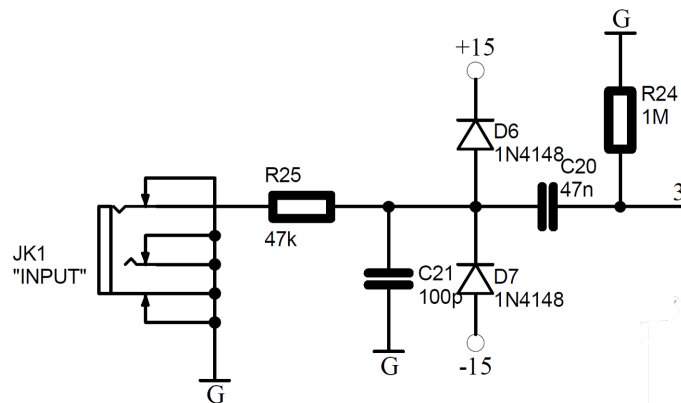


5.1. ábra. Az Orange Crush 20L erősítő áramköre részekre bontva

K-módszert használtam a torzító rész megvalósításához, hiszen abban található két dióda, ami miatt a részáramkör nem lineáris. A többi áramköri részt bilineáris transzformáció segítségével modelleztem. Az LTSpice szimuláció alapján az erősítő, a torzító és a végfokozat részénél volt szükség a műveleti erősítő levágásának figyelembevételéhez, hiszen valóságnak megfelelő bemeneti jel esetén a kimenetén gyakran nagyobb feszültség jelent meg, mint a műveleti erősítők tápfeszültsége.

5.1. Az áramköri részek modellezése

5.1.1. Bemeneti szűrő



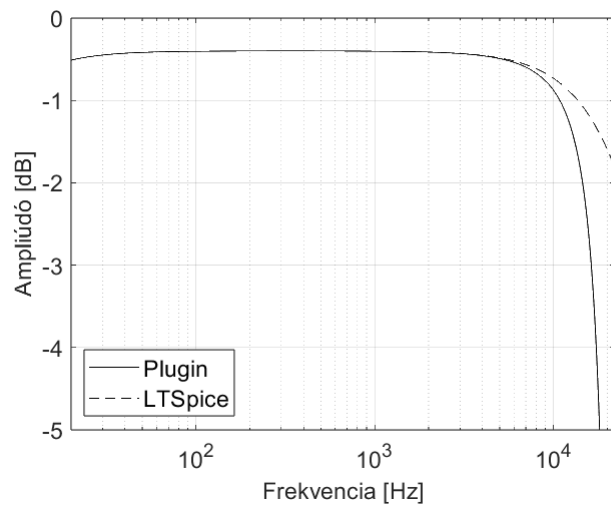
5.2. ábra. A bemeneti szűrők

A bemeneti szűrő áramköre az 5.2 ábrán látható, ami egy alul- és egy felüláteresztő RC szűrőből áll. A diódáknak védelmi célja van. Ezek megakadályozzák, hogy a műveleti erősítő bemenetére több feszültség kerüljön, mint a tápfeszültsége. Ezért a modellezésben elhanyagolom ezek hatását.

A részáramkör átviteli függvénye:

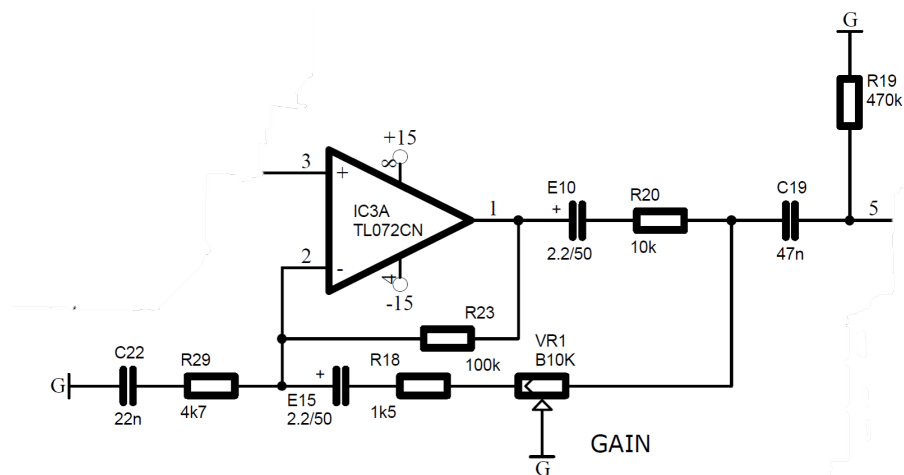
$$H(s) = \frac{C_{20}R_{24}s}{C_{20}C_{21}R_{24}R_{25}s^2 + (C_{20}(R_{24} + R_{25}) + C_{21}R_{25})s + 1} \quad (5.1)$$

A hálózatot bilineáris transzformációval (2.10) modellezve és az átviteli függvényét LTSpice szimulációval összehasonlítva az 5.3 ábrán látható. A különbség a bilineáris transzformáció frekvenciatorzításának köszönhető, ami nem okoz problémát a nagy törésponti frekvenciának köszönhetően.



5.3. ábra. Az LTSpice szimuláció eredménye összehasonlítva a megvalósított szűrővel

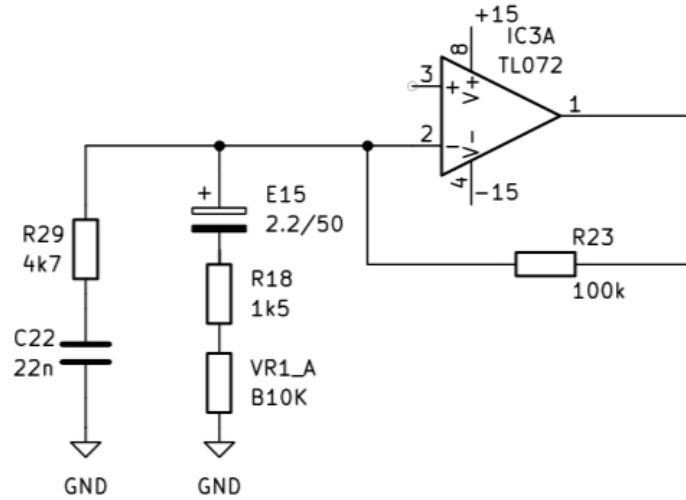
5.1.2. Erősítő rész



5.4. ábra. A bemeneti szűrők

Az erősítő rész áramköre az 5.4 ábrán látható. A műveleti erősítő kimeneténél felbontottam az áramkört, a műveleti erősítőt tartalmazó részt és a passzív hálózatot külön vizsgáltam. Mind a két részt bilineáris transzformáció (2.10) segítségével modelleztem.

5.1.2.1. A műveleti erősítőt tartalmazó rész



5.5. ábra. Az erősítő részről leválasztott műveleti erősítő rész

A műveleti erősítőt tartalmazó rész átviteli függvénye:

$$H(s) = \frac{(AB + BC + AD)s^2 + (A + B + C + D)s + 1}{ABs^2 + (A + B)s + 1} \quad (5.2)$$

$$A = R_{29}C_{22} \quad (5.3)$$

$$B = (V_{R_{1_A}} + R_{18})E_{15} \quad (5.4)$$

$$C = R_{23}C_{22} \quad (5.5)$$

$$D = R_{23}E_{15} \quad (5.6)$$

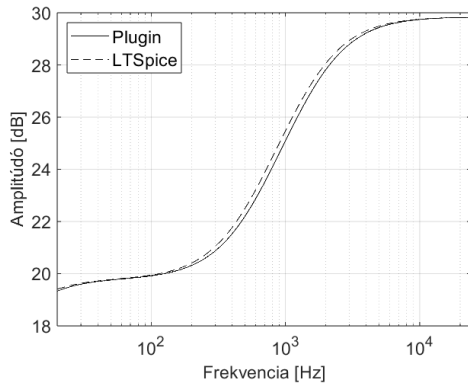
LTSpice szimuláció segítségével megállapítottam, hogy a valóságnak megfelelő bemeneti jel esetén is a GAIN potméter adott állásaiban le fog vágni a műveleti erősítő a tápfeszültségénél, mert a műveleti erősítő kimenetén ideális esetben nagyobb feszültség jelenne meg, mint a tápfeszültsége.

A kisjelű, lineáris átviteli függvény az LTSpice szimulációval összehasonlítva különböző paraméterértékek esetén az 5.6 ábrán látható

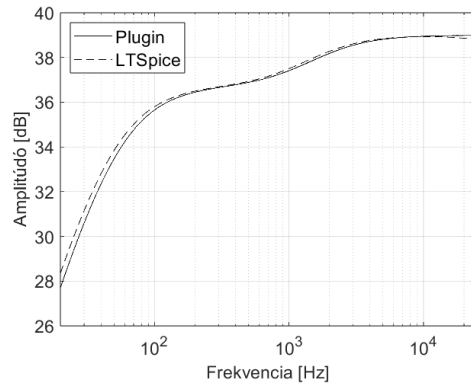
Az 5.7 ábrán a jelalak minimálisan eltér az általam megvalósított modell és az LTSpice szimuláció eredménye között.

Ez valószínűleg annak köszönhető, hogy az LTSpice-ban lévő műveleti erősítő modell a műveleti erősítő frekvenciafüggését és slew rate-jét is figyelembe veszi, illetve sokkal bonyolultabb karakterisztikát használhat.

A műveleti erősítő levágását egyszerű levágó algoritmussal valósítottam meg [15, 16]. Eredetileg az ideális műveleti erősítőt tartalmazó modell fut, és ha annak a kimenete túllépné az LTSpice alapján meghatározott levágási feszültséget, akkor a szaturációs modellel visszszámolom a bemenetet (ekkor a kimenete a levágási feszültség). A rendszeregyenlet segítségével egyszerűen visszszámolható a bemenet, azt csak át kell rendezni $u[n]$ -re, $y[n]$ adott. Ezután a bufferekben ezek alapján kell frissíteni az értékeket.

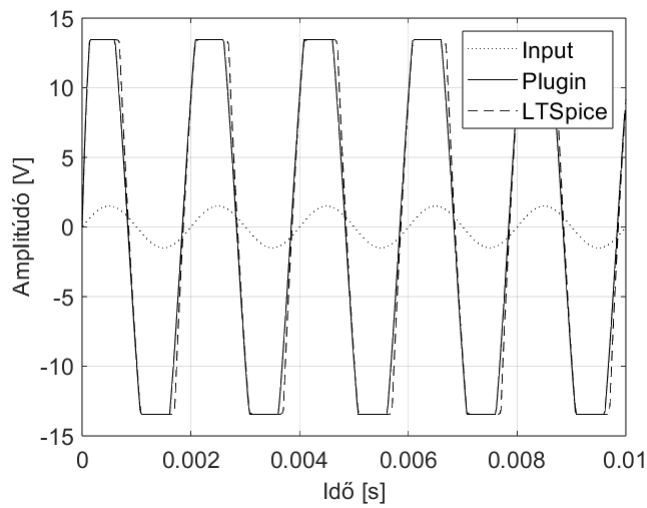


(a) GAIN: 5 k Ω – 5 k Ω



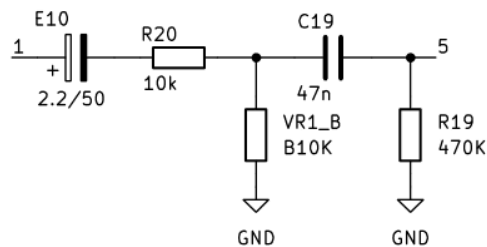
(b) GAIN: 0 Ω – 10 k Ω

5.6. ábra. A műveleti erősítőt tartalmazó rész átvitele az LTSpice szimulációval összehasonlítva



5.7. ábra. A műveleti erősítőt tartalmazó rész válasza egy 500 Hz-es 1.5 V amplitúdós szinusz jelre

5.1.2.2. Passzív hálózat



5.8. ábra. Az erősítő részből leválasztott passzív hálózat

A passzív hálózat átviteli függvénye:

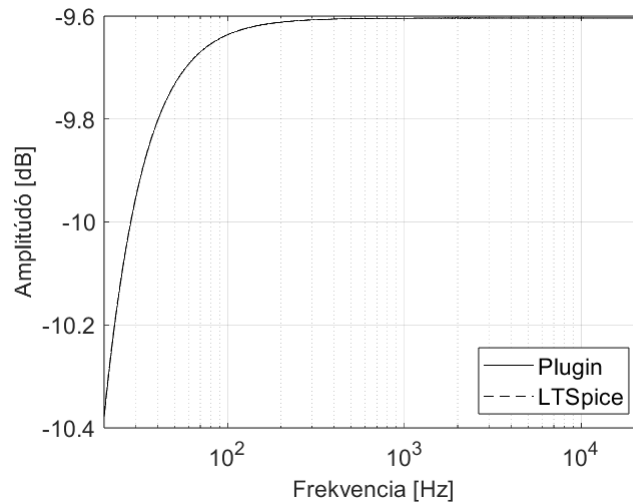
$$H(s) = \frac{As^2}{Cs^2 + Bs + 1} \quad (5.7)$$

$$A = C_{19}E_{10}V_{R_{1_B}}R_{19} \quad (5.8)$$

$$B = C_{19}V_{R_{1_B}} + E_{10}V_{R_{1_B}} + C_{19}R_{19} + E_{10}R_{20} \quad (5.9)$$

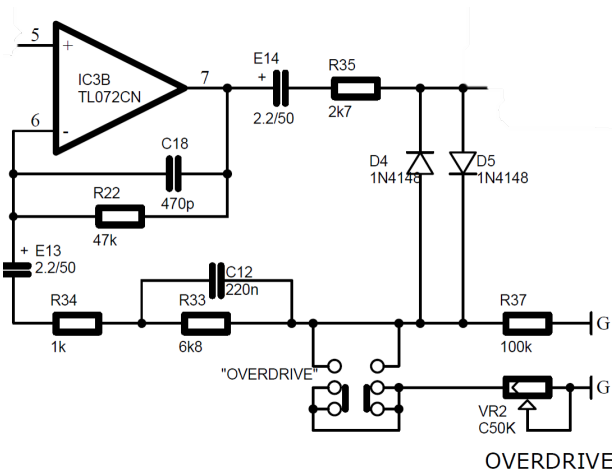
$$C = C_{19}E_{10}V_{R_{1_B}}R_{19} + C_{19}E_{10}V_{R_{1_B}}R_{20} + C_{19}E_{10}R_{19}R_{20} \quad (5.10)$$

A passzív hálózat felüláteresztő jellege miatt a bilineáris transzformációval létrehozott modell jól követi a frekvenciamenetet, ez az 5.9 ábrán látható.



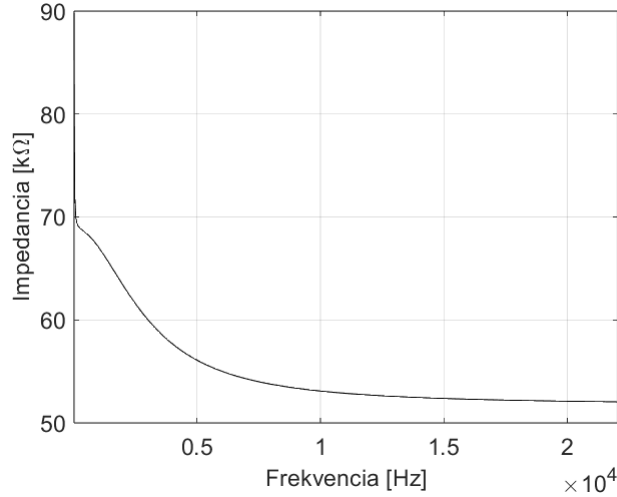
5.9. ábra. A műveleti erősítő utáni passzív hálózat tesztelése. A GAIN potméter 5 kΩ–5 kΩ értékre volt állítva.

5.1.3. Torzító rész



5.10. ábra. A torzító rész áramköre. A nemlinearitást a két dióda és a műveleti erősítő levágása okozza.

A torzító rész utáni áramkörreszt leválasztottam, és külön részként kezeltem. Ezt azért tehettem meg, mert annak a bemeneti impedanciája sokkal nagyobb, mint a torzító rész kimeneti impedanciája, azaz a torzító rész feszültséggenerátorosan hajtja meg az utána következő részeket (ezt LTSpice-ban egy követő erősítő beiktatásával is teszteltem).



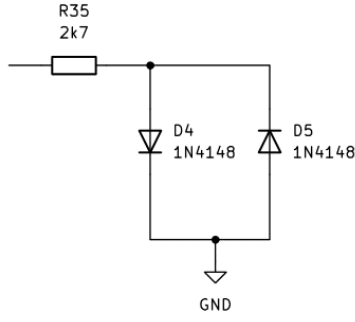
5.11. ábra. A torzító részt követő áramkör bemeneti impedanciája

A torzító rész modellezéséhez a K-módszert alkalmaztam, ahol a nemlinearitást a diódák okozták. Amikor csak a diódákat vettem a nemlineáris résznek, nem tudtam egyértelműen meghatározni az áramkör állapotváltozós leírását. A problémát úgy oldottam meg, hogy az R_{35} soros ellenállást is bevontam a nemlineáris részbe, így tudtam létrehozni a táblázatot. Az áramra egy implicit egyenlet jött ki, míg a feszültségre egy explicit, ezért a táblázatot a feszültség alapján generáltam.

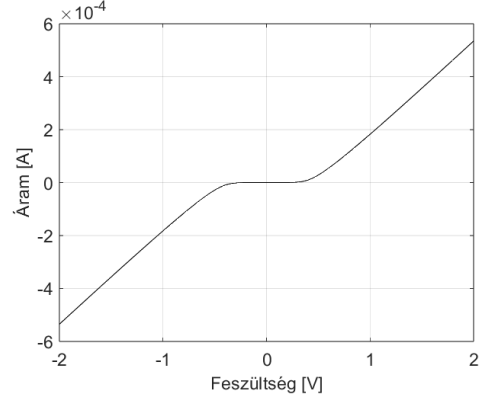
$$f(u) = i = I_{S0} \sinh \frac{u - R_{35}i}{u_T} \quad (5.11)$$

$$f^{-1}(i) = u = u_T \sinh^{-1} \frac{i}{2I_{S0}} + R_{35}i \quad (5.12)$$

Pontosabb eredmény elérése érdekében kiexportáltam az LTSpice-ban megépített modellből a nemlineáris karakterisztikát, és a plugin végleges modelljében ezt alkalmaztam, ami körülbelül ~ 250 értékpárból állt. Ezért volt elegendő ilyen kevés értékpár, hiszen a diódák csak kis feszültség esetén határozzák meg az áramot. $[-1.5, 1.5]$ V intervallumon elég sűrűn generáltam értékpárokat, $[-2, -1.5] \cup [1.5, -2]$ V intervallumon már csak fele olyan ritkán, és $[-500, -2] \cup [2, 500]$ V felett pedig körülbelül csak három-három értékpárt helyeztem el, hiszen az áramkörben ekkor már az ellenállás dominál, aminek köszönhetően lineárisnak mondható a rész, és lineáris interpoláció segítségével elég nagyon ritkán értékpárokat letárolni. Az áramkört, amely segítségével kiexportáltam a karakterisztikát és a kiexportált karakterisztikát az 5.12 ábra szemlélteti.



(a)



(b)

5.12. ábra. A nemlineáris rész, ami alapján a karakterisztikát generáltam le, és a kiexportált karakterisztika

Az állapotváltozós leírás alapján kiszámolt mátrixok:

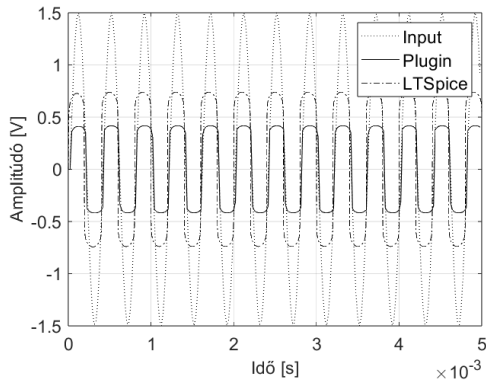
$$\mathbf{A} = \begin{pmatrix} -\frac{1}{C_{12}(R_{37} + R_{34})} - \frac{1}{C_{12}R_{33}} & 0 & -\frac{1}{C_{12}(R_{37} + R_{34})} & 0 \\ \frac{C_{18}(R_{37} + R_{34})}{1} & -\frac{1}{R_{22}C_{18}} & \frac{C_{18}(R_{37} + R_{34})}{1} & 0 \\ -\frac{E_{13}(R_{37} + R_{34})}{0} & 0 & -\frac{E_{13}(R_{37} + R_{34})}{0} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} -\frac{1}{C_{12}(R_{37} + R_{34})} \\ \frac{C_{18}(R_{37} + R_{34})}{1} \\ -\frac{E_{13}(R_{37} + R_{34})}{0} \\ 0 \end{pmatrix} \quad \mathbf{C} = \begin{pmatrix} \frac{R_{37}}{C_{12}(R_{37} + R_{34})} \\ -\frac{R_{37}}{C_{18}(R_{37} + R_{34})} \\ \frac{R_{37} + R_{34}}{1} \\ -\frac{E_{14}}{1} \end{pmatrix} \quad (5.13)$$

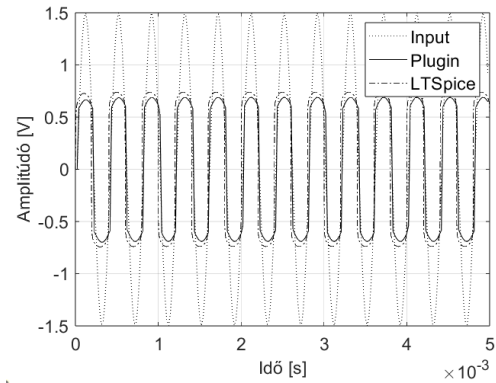
$$\mathbf{D} = \begin{pmatrix} -\frac{R_{37}}{R_{37} + R_{34}} & 1 & -\frac{R_{37}}{R_{37} + R_{34}} & 1 \end{pmatrix}$$

$$E = \frac{R_{34}}{R_{37} + R_{34}} \quad F = -\frac{R_{34}R_{37}}{R_{37} + R_{34}}$$

A (3.12), (3.13), (3.14), (3.19) egyenletekkel kiszámoltam a segédváltozókat, és a (3.32) lépései szerint programoztam be az algoritmust. Az általam számolt karakterisztikával kapott eredmény az 5.13a ábrán, az LTSpice-ből kiexportált karakterisztikával kapott eredmény pedig az 5.13b ábrán látható.



(a)



(b)

5.13. ábra. A plugin és az LTSpice által adott válasz, 5.13a ábrán a (5.12) függvényt használva, és 5.13b ábrán az LTSpice-ből exportált karakterisztikát használva. A gerjesztés egy 2500 Hz frekvenciájú, 1.5 V amplitúdójú szinusz jel. Az OVERDRIVE paraméter mindkét tesztnél a maximális beállításon ($R_{37} = 0 \Omega$) volt.

A megvalósítás során kihívást jelentett az OVERDRIVE paraméter, hiszen minden egyes változásánál módosulnak a segédváltozók, azaz minden egyes paraméterváltozásnál újra ki kell számolni a (3.12), (3.13), (3.14), (3.19) egyenleteket, majd el kell végezni az $f(u)$ függvényre a lineáris transzformációt (hiszen függ K értékétől).

5.1.3.1. A műveleti erősítő levágásának megvalósítása

Az LTSpice szimuláció során észrevettem, hogy a műveleti erősítő kimenetén olyan feszültség szintek fordulhatnak elő valóságban bemenetek esetén, amelyeknél meghatározó lesz a műveleti erősítő szaturációs feszültség szintje, és le fogja vágni, ha afelé menne.

Mivel az áramkör részt nem a rendszeregyenlete, hanem az állapotváltozós leírása alapján modellezem, így nem volt olyan egyszerű ezt megoldani, mint ahogy az 5.1.2.1 fejezetben tárgyaltam.

Ilyen esetben meg kell határozni a szaturációs modell állapotváltozós leírását. Ekkor az eredeti u_{be} változó ismeretlen lesz, és a bemenet a műveleti erősítő kimenetén található (az 5.10 ábrán a 7-es pont), aminek feszültsége pedig mindig a szaturációs feszültség lesz.

A szaturációs modell állapotváltozós leírása kifejezhető közvetlenül a (5.13) egyenletrendszerből: a műveleti erősítő kimenete a C_{18} kondenzátor feszültségével tér el az eredeti gerjesztéstől.

Az így kiszámolt állapotváltozós leírás:

$$\mathbf{A} = \begin{pmatrix} \frac{1}{C_{12}(R_{37} + R_{34})} - \frac{1}{C_{12}R_{33}} & \frac{1}{C_{12}(R_{37} + R_{34})} & \frac{1}{C_{12}(R_{37} + R_{34})} & 0 \\ \frac{C_{18}(R_{37} + R_{34})}{1} & -\frac{R_{22}C_{18}}{1} - \frac{C_{18}(R_{37} + R_{34})}{1} & \frac{C_{18}(R_{37} + R_{34})}{1} & 0 \\ \frac{E_{13}(R_{37} + R_{34})}{0} & \frac{E_{13}(R_{37} + R_{34})}{0} & -\frac{E_{13}(R_{37} + R_{34})}{0} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{B} = \begin{pmatrix} \frac{1}{C_{12}(R_{37} + R_{34})} \\ \frac{C_{18}(R_{37} + R_{34})}{1} \\ -\frac{E_{13}(R_{37} + R_{34})}{0} \\ 0 \end{pmatrix} \quad \mathbf{C} = \begin{pmatrix} \frac{R_{37}}{C_{12}(R_{37} + R_{34})} \\ -\frac{C_{18}(R_{37} + R_{34})}{R_{37}} \\ \frac{R_{37} + R_{34}}{1} \\ -\frac{1}{E_{14}} \end{pmatrix}$$

$$\mathbf{D} = \begin{pmatrix} -\frac{R_{37}}{R_{37} + R_{34}} & \frac{R_{37}}{R_{37} + R_{34}} & -\frac{R_{37}}{R_{37} + R_{34}} & 1 \end{pmatrix}$$

$$E = \frac{R_{34}}{R_{37} + R_{34}} \quad F = -\frac{R_{34}R_{37}}{R_{37} + R_{34}}$$

(5.14)

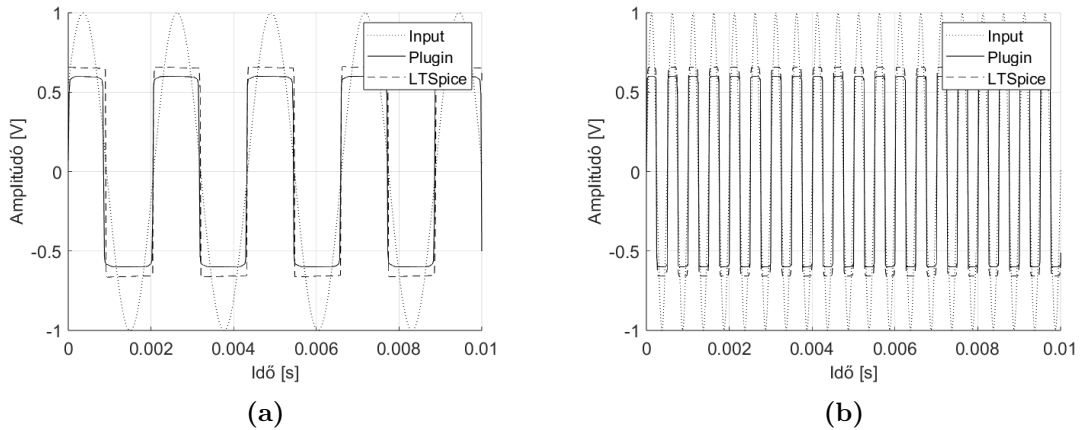
Hasonló módon kezelem a szaturációt, mint az 5.1.2.1 fejezetben: ha az állapotváltozós segítségével kifejezett feszültség, ami a műveleti erősítő kimenetén található, nagyobb, mint a szaturációs feszültség, akkor a K-módszert arra az iterációra az 5.14 változók alapján számolom ki.

Mivel ez az állapotváltozós leírás a műveleti erősítő kimenetét veszi bemenetnek, így az iteráció végén vissza kell számolni az eredeti bemenetet, és az alapján frissíteni a buffereket. Ez egyszerűen megtehető, a C_{18} kondenzátor feszültségét a szaturációs feszültségből kivonva megkapjuk az u_{be} feszültséget.

Ez a megoldás viszont azzal a hátránnyal jár, hogy ha gyakran történik meg a levágás, akkor a két modell fut le egy iteráció alatt egy helyett. Paraméterváltáskor a letárolt értékpárokat a szaturációs modellhez tartozó K szerint kell lineárisan transzformálni az eredeti transzformáció mellett. Így kétszer akkora ebben az esetben a számításigénye az áramkör rész szimulációjának.

Emellett szükséges eltárolni a szaturációs paramétereket is. Így a modell állandóan, függetlenül a szaturációs eset bekövetkezésétől, kétszer annyi memóriát használ.

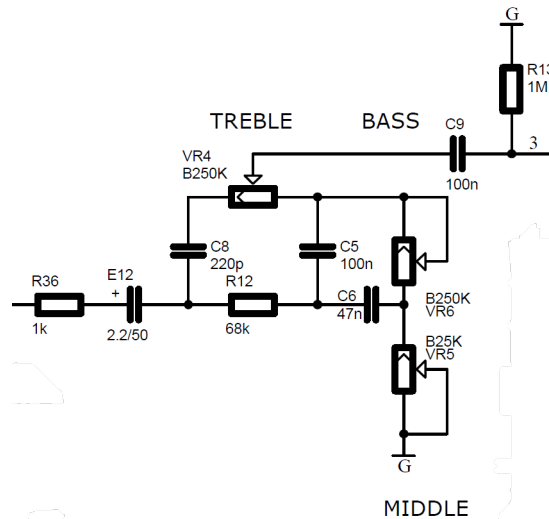
A műveleti erősítő levágásának megvalósítása után a plugin és az LTSpice összehasonlítása az 5.14 ábrán látható.



5.14. ábra. A plugin és az LTSpice által adott válasz; 5.14a ábrán 440 Hz frekvenciájú 1 V amplitúdójú szinusz, 5.14b ábrán 2000 Hz frekvenciájú 1 V amplitúdójú szinusz a gerjesztő jel. Az OVERDRIVE paraméter mindkét tesztnél a maximális beállításon ($R_{37} = 0 \Omega$) volt.

Érdemes megjegyezni, hogy az LTSpice-ban használt modell számításba veszi a műveleti erősítő parazita tulajdonságait: a slew rate-et, frekvenciafüggő viselkedést, véges bemeneti impedanciát, kimeneti impedanciát, közös módusú elnyomást stb., ami az általam megvalósított modelltől hiányzik.

5.1.4. Tone stack



5.15. ábra. Az erősítő tone stack áramköre

Az áramkör rész átviteli függvénye:

$$H(s) = \frac{N(s)}{D(s)} \quad (5.15)$$

Számláló $N(s)$:

$$N(s) = C_9 E_{12} R_{13} (K_1 s^2 + K_2 s^3 + K_3 s^4) \quad (5.16)$$

ahol:

$$K_1 = V_{R_6} (A_1 + A_2 + A_3 + A_4 + A_5 + A_6 + A_7) \quad (5.17)$$

$$\begin{aligned} A_1 &= C_5 C_6, & A_2 &= C_5 C_8, & A_3 &= C_6 C_8, \\ A_4 &= C_5 V_{R_5}, & A_5 &= C_6 V_{R_5}, & A_6 &= C_8 V_{R_5}, & A_7 &= C_8 V_{R_4_B} \end{aligned} \quad (5.18)$$

$$K_2 = V_{R_6} (B_1 + B_2 + B_3 + B_4 + B_5 + B_6 + B_7 + B_8) \quad (5.19)$$

$$\begin{aligned} B_1 &= C_5 C_6 V_{R_5}, & B_2 &= C_6 C_8 V_{R_5}, & B_3 &= C_5 C_8 R_{12}, \\ B_4 &= C_5 C_8 V_{R_4_A}, & B_5 &= C_5 C_8 V_{R_4_B}, & B_6 &= C_6 C_8 R_{12}, \\ B_7 &= C_5 C_6 C_8 V_{R_5}, & B_8 &= C_5 C_6 C_8 V_{R_5} R_{12} \end{aligned} \quad (5.20)$$

$$K_3 = V_{R_6} (D_1 + D_2 + D_3 + D_4 + D_5 + D_6 + D_7) \quad (5.21)$$

$$\begin{aligned} D_1 &= C_5 C_6 C_8 V_{R_5} R_{12}, & D_2 &= C_5 C_6 C_8 V_{R_5} V_{R_4_A}, \\ D_3 &= C_5 C_6 C_8 R_{12} V_{R_4_B}, & D_4 &= C_5 C_6 C_8 V_{R_5} V_{R_4_B}, \\ D_5 &= C_5 C_6 C_8 V_{R_5} R_{13}, & D_6 &= C_5 C_8 C_9 V_{R_5}, \\ D_7 &= C_6 C_8 C_9 R_{12} \end{aligned} \quad (5.22)$$

Nevező $D(s)$:

$$D(s) = P_1 + P_2 s + P_3 s^2 + P_4 s^3 + P_5 s^4 \quad (5.23)$$

ahol:

$$P_1 = C_5 + C_6 + C_8 + E_{12} \quad (5.24)$$

$$P_2 = F_1 + F_2 + F_3 + F_4 + F_5 + F_6 \quad (5.25)$$

$$\begin{aligned} F_1 &= C_5 C_6, & F_2 &= C_5 C_9, & F_3 &= C_6 C_8, \\ F_4 &= C_6 C_9, & F_5 &= C_8 C_9, & F_6 &= C_5 E_{12} \end{aligned} \quad (5.26)$$

$$P_3 = G_1 + G_2 + G_3 + G_4 + G_5 + G_6 + G_7 \quad (5.27)$$

$$\begin{aligned} G_1 &= C_5 C_6 C_9 V_{R_5}, & G_2 &= C_5 C_8 C_9 R_{12}, & G_3 &= C_6 C_8 E_{12} R_{12}, \\ G_4 &= C_5 C_8 E_{12} V_{R_4_A}, & G_5 &= C_5 C_9 V_{R_5} V_{R_4_B}, & G_6 &= C_6 C_9 R_{13}, \\ G_7 &= C_8 C_9 V_{R_5} R_{13} \end{aligned} \quad (5.28)$$

$$P_4 = H_1 + H_2 + H_3 + H_4 + H_5 + H_6 + H_7 \quad (5.29)$$

$$\begin{aligned} H_1 &= C_5 C_6 C_8 V_{R_5} R_{12}, & H_2 &= C_5 C_8 C_9 V_{R_5} V_{R_4_A}, \\ H_3 &= C_6 C_8 C_9 V_{R_5} R_{13}, & H_4 &= C_5 C_6 C_8 R_{12} R_{13}, \\ H_5 &= C_6 C_8 C_9 V_{R_5} V_{R_4_B}, & H_6 &= C_5 E_{12} C_8 V_{R_5} R_{13}, \\ H_7 &= C_5 C_8 V_{R_5} R_{12} V_{R_4_B} \end{aligned} \quad (5.30)$$

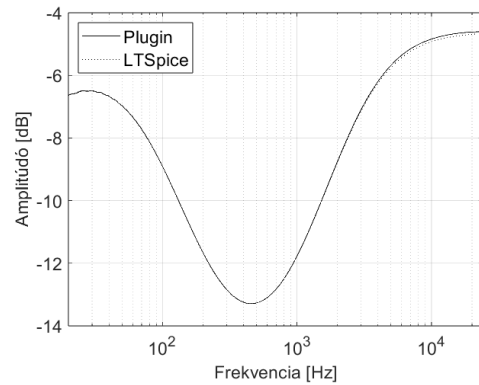
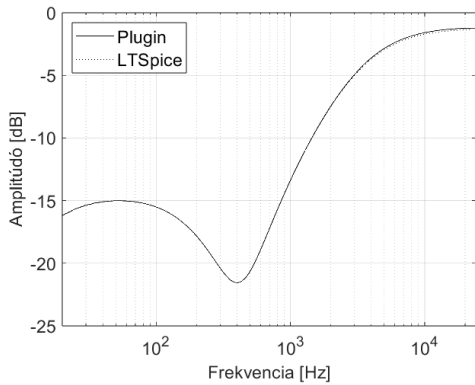
$$P_5 = J_1 + J_2 + J_3 + J_4 + J_5 \quad (5.31)$$

$$\begin{aligned}
J_1 &= C_5 C_6 C_8 C_9 V_{R_5} R_{12}, & J_2 &= C_5 C_6 C_8 C_9 V_{R_5} V_{R_{4_A}}, \\
J_3 &= C_5 C_6 C_8 C_9 V_{R_5} V_{R_{4_B}}, & J_4 &= C_5 C_6 C_8 E_{12} V_{R_{4_B}}, \\
J_5 &= C_5 C_6 C_8 V_{R_5} R_{13}
\end{aligned}
\tag{5.32}$$

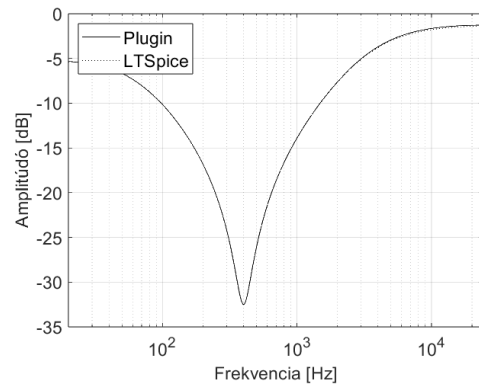
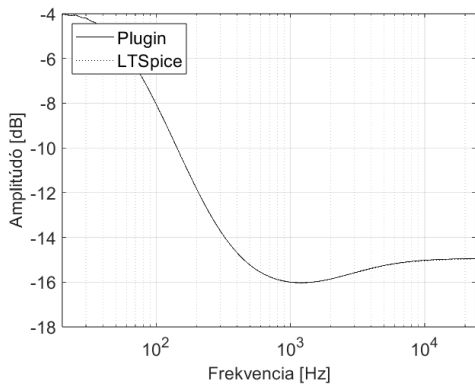
ahol $V_{R_{4_A}}$ és $V_{R_{4_B}}$ a TREBLE potenciométer által szabályozott ellenállások.

Az erősítő tone stack részét szintén bilineáris transzformáció (2.10) segítségével modelleztem. Kihívást okozott a sok paraméter (itt három is van, BASS, MID, TREBLE, ráadásul a mintavételi frekvencia is).

A modellezett áramkör rész összehasonlítása LTSpice szimulációval az 5.16 ábrán látható.



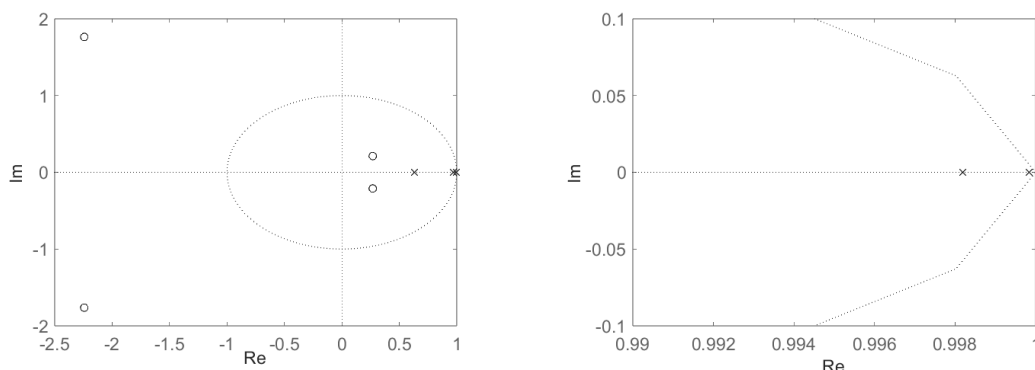
(a) BASS 25 kΩ, MID 2.5 kΩ, TREBLE 225 kΩ (b) BASS 100 kΩ, MID 20 kΩ, TREBLE 25 kΩ



(c) BASS 200 kΩ, MID 12 kΩ, TREBLE 0 Ω (d) BASS 250 kΩ, MID 0 Ω, TREBLE 125 kΩ

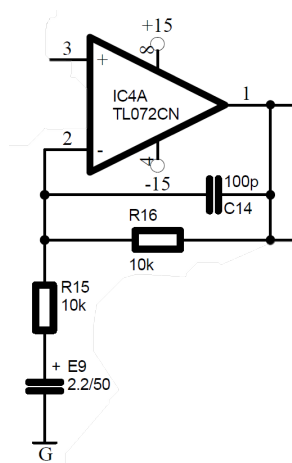
5.16. ábra. A tone stack áramkör rész átvitele különböző paraméter beállítások esetén

A szűrőegytárhók számításához nem adott elég nagy pontosságot a float (32 bit) típus, emiatt double (64 bit) típust kellett használnom. A transzformált rendszer néhány pólusa olyan közel volt az egységkörhöz bizonyos paraméter beállításoknál, hogy a potencióterek bizonyos állásainál a legközelebbi pólus az egységkörön kívülre esett float típus esetén. Ezt a double típusban való tárolás megoldotta. Egy adott paraméterállítás esetén a pólusok elhelyezkedését mutatja az 5.17 ábra.



5.17. ábra. A bilineáris transzformáció segítségével meghatározott átviteli függvény zérusainak és pólusainak pozíciója az 5.16a ábrán használt potméter beállítások mellett (BASS 25 kΩ, MID 2.5 kΩ, TREBLE 225 kΩ)

5.1.5. Aktív szűrő

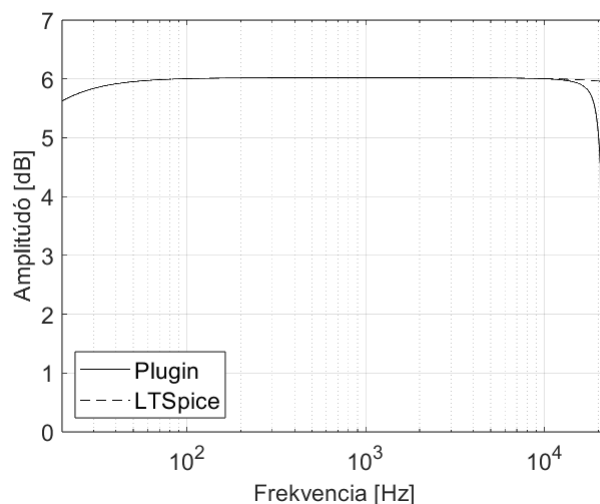


5.18. ábra. Az erősítő tone stack után lévő aktív szűrő áramkör

A részáramkör átviteli függvénye:

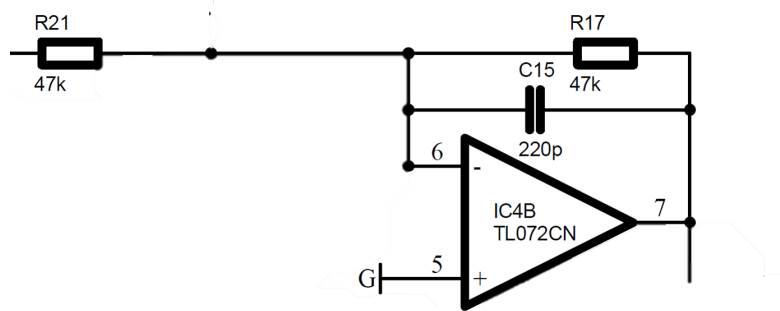
$$H(s) = \frac{C_{14}R_{16}R_{15}E_9s^2 + (R_{15}E_9 + E_9R_{16} + 1 + C_{14}R_{16})s + C_{14}R_{16}}{C_{14}R_{16}s + 1} \quad (5.33)$$

Egyszerű aktív szűrő, bilineáris transzformáció segítségével modelleztem. Célja az áramkörben, hogy a kis- és nagyfrekvenciás komponenseket kiszűrje. A bilineáris transzformáció miatt a mintavételi frekvenciákhoz közeli frekvenciákon nem pontos, hiszen a bilineáris transzformáció frekvenciatorzítást okoz nagy frekvenciákon, ami a nagy törésponti frekvencia miatt nem okoz problémát. Az eredmény az 5.19 ábrán látható.



5.19. ábra. Az erősítő tone stack után lévő aktív szűrő áramkörének átvitele

5.1.6. Összeadó áramkör

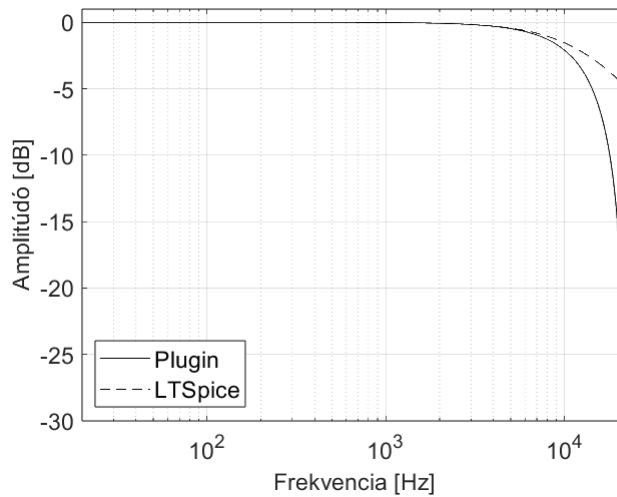


5.20. ábra. Az áramkör, ami az erősítő jack bemenetén beadott jelet összeadja az erősítő jelével

Az áramkör rész átviteli függvénye:

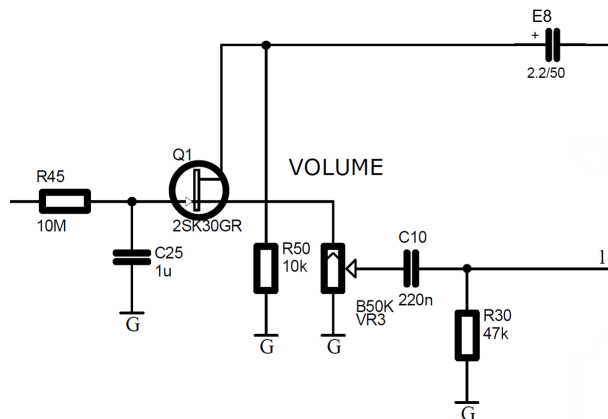
$$H(s) = -\frac{R_{17}}{C_{15}R_{17}R_{21}s + R_{21}} \quad (5.34)$$

Ez az áramköri rész azért felel, hogy az erősítőbe jack csatlakozón beadott jelet összeadja az erősítő jelével. Mivel a jack csatlakozón beadott jelet nem akarjuk módosítani, ezért minden hangszínváltoztatás és torzítás után van ez az áramkör. Az erősítése egységnyi, és egy aluláteresztő szűrőt valósít meg, hogy a jack bemenetről bejövő nagy frekvenciájú komponenseket kiszűrje. Bilineáris transzformáció (2.10) segítségével modelleztem, ami megfelelő eredményt biztosít a szűrő nagy törésponti frekvenciája miatt. Az eredmény az 5.21 ábrán látható.



5.21. ábra. Az áramkör, ami az erősítő jack bemenetén beadott jelet összeadja a gitár jelével

5.1.7. Hangerőszabályozó



5.22. ábra. Az erősítő hangerőszabályzó áramköre

Az áramkörben található egy MOSFET tranzisztor a hangerőszabályzó potméter előtt. Ez a tranzisztor a végfokozat védelmét biztosítja. A MOSFET gate elektródáján lévő $1 \mu F$ kapacitású kondenzátor lassan fog feltöltődni az előtte található $10 \text{ M}\Omega$ -os ellenálláson. Miután feltöltődött teljesen, utána a MOSFET gate-jén konstans feszültség lesz (nyitva lesz), és az impedanciája az alapján fog változni, amennyi feszültség esik rajta. Ez $\text{m}\Omega$ nagyságrendekben lesz, ami elhanyagolható az utána lévő $10 \text{ k}\Omega$ -os osztó miatt.

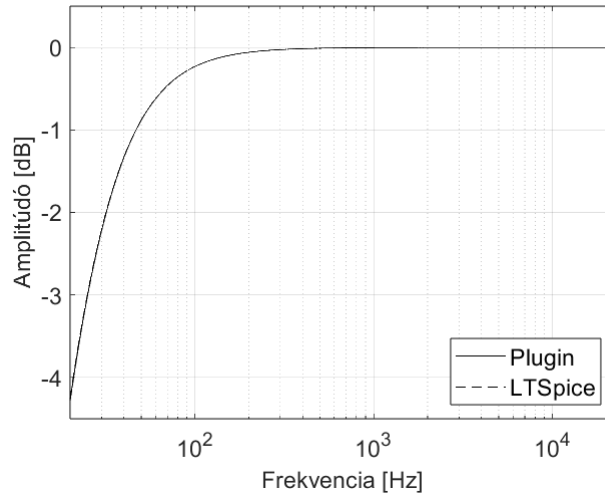
Az áramkör rész átviteli függvénye:

$$H(s) = \frac{A_1 s^2}{A_2 + (A_3 + A_4)s + A_5 s^2} \quad (5.35)$$

$$\begin{aligned}
A_1 &= C_{10} E_8 R_{30} R_{50} V_{R_{3_B}} \\
A_2 &= R_{50} + V_{R_{3_A}} + V_{R_{3_B}} \\
A_3 &= C_{10} R_{30} R_{50} + C_{10} R_{30} V_{R_{3_A}} + C_{10} R_{30} V_{R_{3_B}} \\
A_4 &= C_{10} R_{50} V_{R_{3_B}} + E_8 R_{50} V_{R_{3_A}} + E_8 R_{50} V_{R_{3_B}} \\
A_5 &= C_{10} E_8 R_{30} R_{50} V_{R_{3_A}} + C_{10} E_8 R_{30} R_{50} V_{R_{3_B}} + C_{10} E_8 R_{50} V_{R_{3_A}} V_{R_{3_B}}
\end{aligned} \tag{5.36}$$

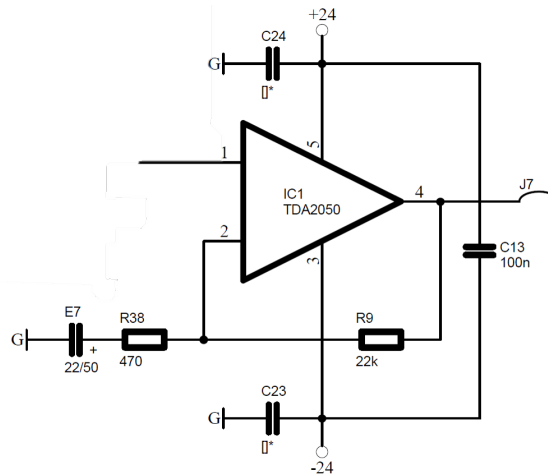
ahol $V_{R_{3_A}}$ és $V_{R_{3_B}}$ a VOLUME potenciométer által szabályozott ellenállások.

A részáramkört szintén bilineáris transzformáció (2.10) segítségével modelleztem. Az eredmény az 5.23 ábrán látható.



5.23. ábra. A részáramkör átvitele a (VOLUME 10 k Ω)

5.1.8. Végfokozat



5.24. ábra. Az erősítő kimenő fokozatának áramköre

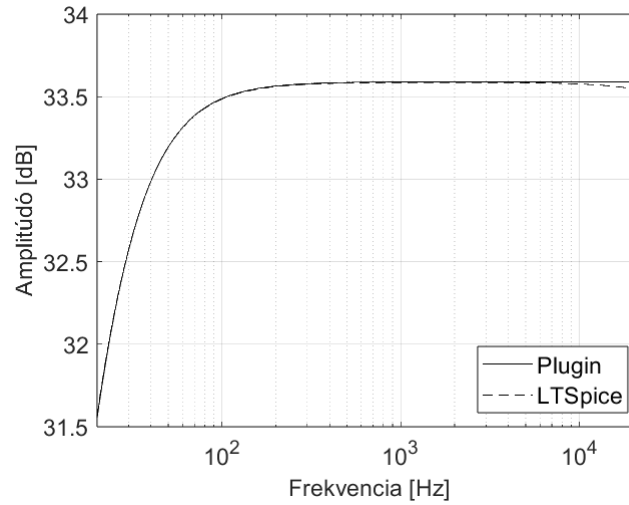
Ez az áramköri rész gondoskodik az erősítő jelének olyan szintre való felerősítéséről, hogy azt közvetlenül a hangszóróra lehessen vezetni. A műveleti erősítő tápján lévő kondenzátorok feladata a tápfeszültség stabilizálása, hogy védelmet nyújtsanak mind a közös módusú, mind a differenciál módusú zavarok ellen. Az E_7 kondenzátornak köszönhetően az átvitel-

nek felüláteresztő jellege lesz, hogy a hangszóróra ne kerüljön nagyon alacsony frekvenciás jel.

Az áramkör rész átviteli függvénye:

$$H(s) = \frac{1 + (E_7 R_9 + E_7 R_{38})s}{1 + E_7 R_{38}s} \quad (5.37)$$

Ezt az áramkört is bilineáris transzformáció (2.10) segítségével modelleztem. Az eredmény az 5.25 ábrán látható. A műveleti erősítő levágását az 5.1.2.1 fejezetben bemutatott módon valósítottam meg annak érdekében, hogy könnyebben tudjam a plugin hangerőszintjeit skálázni (megfelelő legyen digitális szinteknek).



5.25. ábra. Az erősítő kimenő fokozatának áramköre

A nagy frekvenciáknál az LTSpice szimuláció és az én megvalósításom között valószínűleg azért észlelhető különbség, mert az LTSpice nem hanyagolja el a műveleti erősítő frekvenciamenetét. Nagy erősítéseknél az átvitel frekvenciafüggővé válik, ami minimális eltérést okozhat a szimulált és a megvalósított rendszer között.

5.2. A torzítás okozta belapolódó felharmonikusok kiszűrése

5.2.1. Shannon-féle mintavételi tétel

A Shannon-féle mintavételi tétel azt mondja ki, hogy egy folytonos jelet pontosan lehet rekonstruálni, ha azt egy olyan frekvencián mintavételezzük, amely legalább kétszer akkora, mint a jel legnagyobb frekvencia komponense [17]. Ez a frekvencia a mintavételi frekvencia.

$$f_s > 2f_{\max} \quad (5.38)$$

ahol:

- f_s a mintavételi frekvencia
- f_{\max} a jel legnagyobb frekvenciája

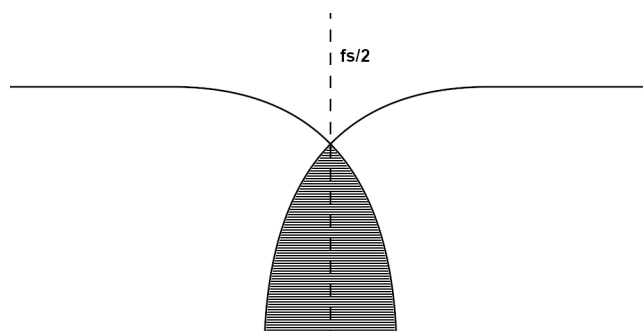
Aliasing (belapolódás) akkor következik be, amikor olyan mintavételi frekvencián mintavételezzük a jelet, amivel megsérti a mintavételi tételt. Ilyenkor a nagyobb frekvenciájú komponensek az alsóbb frekvenciákra belapolódnak. Az aliasing gyakran úgy jelenik meg, mint egy nem kívánt „zaj” vagy torzítás.

5.2.2. Harmonikus torzítás

A harmonikus torzítás egy olyan jelenség, amely akkor fordul elő, amikor egy rendszer (például egy erősítő vagy egy áramkör) a bemeneti jel mellett további, a bemeneti jel frekvenciájának egész számú többszörösein jelentkező komponenseket (harmonikusokat) hoz létre a kimenetén. Ezek a harmonikusok nincsenek jelen az eredeti bemeneti jelben, hanem az eszköz nemlineáris viselkedése miatt keletkeznek.

A harmonikus torzítás sok esetben jelen van a gitárerősítőkben, mert az alapvetően hozzájárul a gitárhangzás gazdagításához. Ez a torzítás olyan felharmonikusokat ad a kimeneti jelhez, amelyek az alapfrekvencia egész számú többszörösei, így a gitár hangja teltebb, melegebb és dinamikusabb lesz. Egyes zenei műfajokban ez a hangzás alapvető.

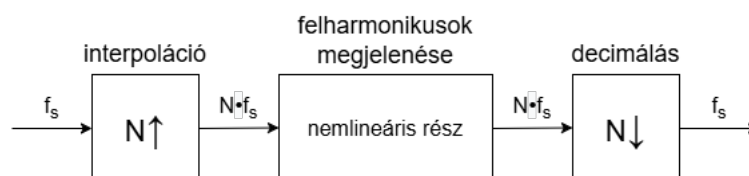
A digitális rendszerekben a harmonikus torzítás különösen problémás lehet, mert a nemlineáris elemek által generált felharmonikusok jelentősen befolyásolhatják a kimeneti jel minőségét, és olyan nem kívánt jelenségekhez vezethetnek, mint a spektrális belapolódás (aliasing). A belapolódást szemlélteti az 5.26 ábra.



5.26. ábra. Belapolódás történik, ha a mintavételi frekvencia felénél ($f_s/2$) nagyobb frekvenciakomponenseket is tartalmaz a jel. A belapolódott komponensek a vízszintes vonalakkal jelölt rész. Az x tengely frekvencia, az y tengely amplitúdó.

5.2.3. Újramintavételezés

A harmonikus torzítás okozta belapolódás csökkenthető úgy, hogy olyan mintavételi frekvenciájú jellel újramintavételezzük (interpoláljuk) a jelet a nemlineáris rész előtt, hogy az új mintavételi frekvencián a nemlineáris rész kimenetén megjelenő jel ne sértse meg a mintavételi tételt. Ezután ezt egy olyan aluláteresztő szűrővel, amelynek törésponti frekvenciája az eredeti mintavételi frekvencia felénél található, megszűrve a jel már nem fog tartalmazni az eredeti mintavételi frekvencia felénél nagyobb frekvenciájú komponenseket. A jelet az eredeti mintavételi frekvenciára visszakonvertálva (decimálva) megszabadultunk a belapolódás problémájától. Ezt a folyamatot szemlélteti az 5.27 ábra.



5.27. ábra. A felharmonikusok által okozott belapolódás csökkentése: először interpoláció, a nemlineáris rész lefutása után pedig decimálás

A valóságban viszont nem lehetséges a tökéletes aluláteresztő szűrő megvalósítása, így decimálás előtt nem lehet teljesen kiszűrni a mintavételi tételt megsértő, nagy frekvenciájú komponenseket.

A dolgozatban azt az esetet tárgyalom, amikor a nagyobb frekvencia az alacsonyabb pozitív egész többszöröse (az 5.27 ábrán látható N változó pozitív egész szám).

5.2.3.1. Kapcsolat a folytonos és diszkrét idejű jel között

Egy sávkorlátos, folytonos idejű $x(t)$ jel rekonstruálható a $x(nT)$ mintáiból az alábbi egyenlet alapján [18]:

$$x(t) = \sum_{n=-\infty}^{\infty} x(nT) \text{sinc}(\pi f_c(t - nT)) \quad (5.39)$$

ahol f_c a diszkrét jel mintavételi frekvenciája, T a mintavételi idő, illetve:

$$\text{sinc}(t) \triangleq \frac{\sin(t)}{t} \quad (5.40)$$

Ha a (5.39) egyenletbe behelyettesítjük t helyére az új mintavételi pontjaink idejét (lT'), akkor az alábbi egyenletet kapjuk [18]:

$$x'[l] = x'(lT') = \sum_{n=-\infty}^{\infty} x[n] \text{sinc}(\pi f_c(lT' - nT)) \quad (5.41)$$

ahol $x'[l]$ az új, $f'_s = 1/T'$ mintavételi frekvenciával rendelkező jel.

Az lT' változót kifejezve T változóval:

$$x'[kT + \Delta T] = \sum_{n=-\infty}^{\infty} x[n] \text{sinc}(\pi f_c(kT + \Delta T - nT)) = \sum_{n=-\infty}^{\infty} x[n] \text{sinc}(\pi f_c((k - n)T + \Delta T)) \quad (5.42)$$

ahol k egy pozitív, egész szám, lT' változót T egész számú többszöröséből és egy törtrészből állítjuk elő.

Ekkor tekinthetünk a $\text{sinc}(\pi f_c((k - n)T + \Delta T))$ függvényre úgy, mint egy kétváltozós, polifázisú függvényre:

$$\text{sinc}(\pi f_c(nT + \Delta T)) \triangleq h(n, \Delta) \quad (5.43)$$

amelynek Δ paraméterétől függ az együtthatókészlete. Behelyettesítve $h(n, \Delta)$ jelölést:

$$x'[kT + \Delta T] = \sum_{n=-\infty}^{\infty} x[n] h(k - n, \Delta) \quad (5.44)$$

Mivel az összefüggés a konvolúció alakját veszi fel, és a konvolúció asszociatív, így az egyenlet az alábbi formára rendezhető:

$$x'[kT + \Delta T] = \sum_{n=-\infty}^{\infty} x[k - n] h(n, \Delta) \quad (5.45)$$

5.2.3.2. Interpoláció

Interpoláció esetén $f'_s > f_s$. Ekkor $f_c = f_s/2$, hiszen a kisebb mintavételű frekvenciájú jelet nagyobb mintavételi frekvenciává konvertálás során nem akarunk a jeltől információt veszteni (ugyanazt a jelet szeretnénk, csak nagyobb mintavételi frekvencián).

Mivel $f'_s = N f_s$, N pedig egy pozitív egész szám, így:

$$T' = \frac{T}{N} \quad (5.46)$$

$$lT' = l \frac{T}{N} = kT + \Delta T \quad (5.47)$$

$$\frac{l}{N} = k + \Delta \quad (5.48)$$

Összességében:

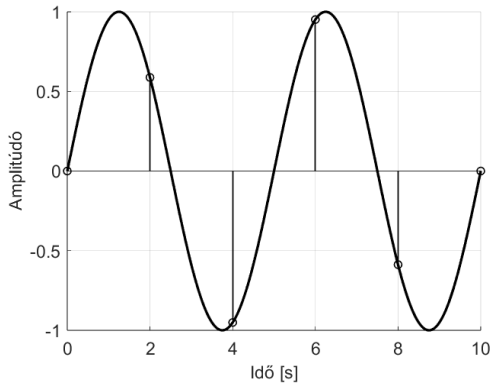
$$k = \lfloor l/N \rfloor \quad (5.49)$$

$$\Delta = l/N - \lfloor l/N \rfloor \quad (5.50)$$

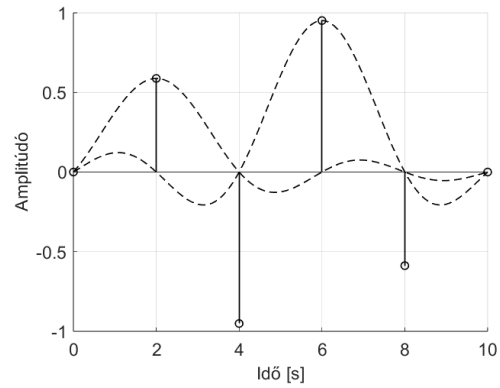
$$x'[l] = \sum_{n=-\infty}^{\infty} x[\lfloor l/N \rfloor - n]h(n, \Delta) \quad (5.51)$$

Az (5.50) egyenlet alapján Δ értéke $0, 1/N, 2/N, \dots, (N-1)/N$ értékeket vesz fel. Az ilyen szűrőstruktúrákat hívják polifázisú szűrőknek. A P együtthatójú szűrőként akarjuk megtervezni a (5.2.3.1) szűrőt, akkor ehhez NP elemet kell eltárolnunk, hiszen a paraméterkészlete a Δ változótól függ.

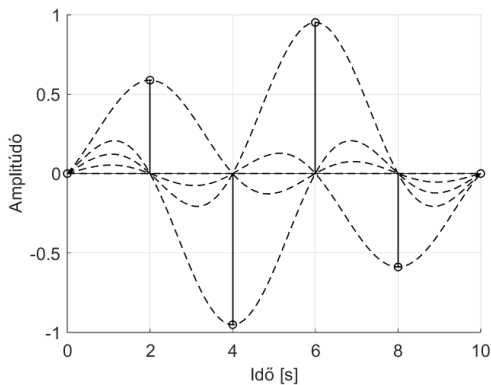
Az interpoláció értelmezését az 5.28 ábra szemlélteti.



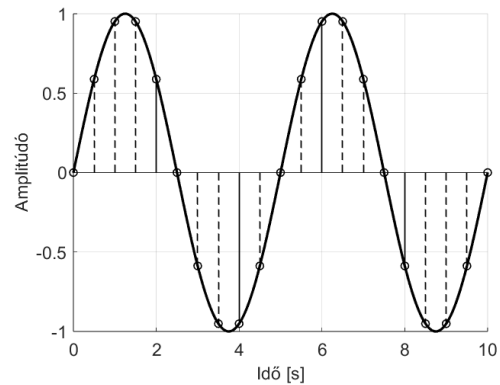
(a) Az eredeti jel, f_s mintavételi frekvenciával mintavételezve



(b) A sinc függvény illesztése



(c) Az összes mintára sinc függvény illesztése



(d) Az új mintavételi pontokon az összes sinc függvényértéket összeadva az f'_s mintavételi frekvenciás jel az eredmény.

5.28. ábra. Az interpoláció értelmezése (5.2.3.1) egyenlet szerint.

5.2.3.3. Decimálás

Decimálás esetén $f'_s < f_s$. Ekkor $f_c = f_s/2$, hiszen az átlapolódás elkerülése érdekében az új, egyben kisebb mintavételi frekvencia szerint kell a sávszélességet meghatározni.

Mivel $Nf'_s = f_s$, N pedig egy pozitív egész szám, akkor:

$$T' = NT \quad (5.52)$$

$$lT' = lNT = kT + \Delta T \quad (5.53)$$

$$Nl = k + \Delta \quad (5.54)$$

Összességében:

$$k = Nl \quad (5.55)$$

$$\Delta = 0 \quad (5.56)$$

$$x'[l] = \sum_{n=-\infty}^{\infty} x[Nl - n]h(n, \Delta) \quad (5.57)$$

Tulajdonképpen így a (5.2.3.1) egyenletben megadott h nem fog függeni már Δ értéktől, így csak egy egyszerű, aluláteresztő szűrő lesz.

A decimálás értelmezését az 5.29 ábra szemlélteti.

5.2.4. Megvalósítás a pluginban

A pluginban külön írtam egy osztályt a megfelelő sinc függvény legenerálásához. Ez az osztály abban is segítségemre volt, hogy később a konvolúció számolásánál nem kellett az indexeléssel foglalkoznom, hiszen az operator `[]` függvényét úgy valósítottam meg, hogy a sinc közepe a 0 indexen legyen, és tudjak negatívan indexelni. Emellett még megvalósítottam az operator `()` függvényét is, ami két értéket vár az argumentumaként: az indexelő, n paramétert, és a Δ paramétert.

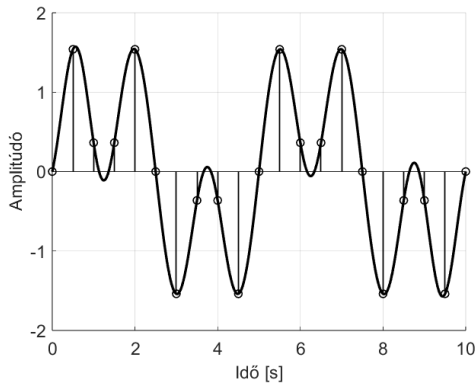
Az interpolálás és decimálás során ugyanazokat az előre legenerált szűrő együtthatókat használom, így elég egyetlen sinc sorozatot letárolnom. Kevesebb memóriahasználat érdekében csak a sinc pozitív felét tárolom el.

Erre azért van lehetőség, mert az interpolálásnál és a decimálásnál is ugyanazt, f_s töréspontú szűrőt akarjuk alkalmazni. Mivel interpolálásnál az alap mintavételi frekvenciára adjuk meg a polifázisú szűrőt az időtartományban, akkor ennek a szűrőnek az impulzusválasza $Nf_s = f'_s$ frekvencián N -szer annyi együtthatóból fog állni, mint a kisebb f_s frekvencián. A polifázisú szűrőhöz pedig kiszámoltuk az időtartománybeli értékeket $1/N, 2/N \dots (N-1)/N$ törtrészeknél, ami azért szerencsés, mert az új mintavételi frekvencián ugyanehhez az szűrőhöz pont ezekre a pontokra fognak esni az minták. És mivel ekkor (5.56) igaz, így nem is kell ennél több mintát legenerálni.

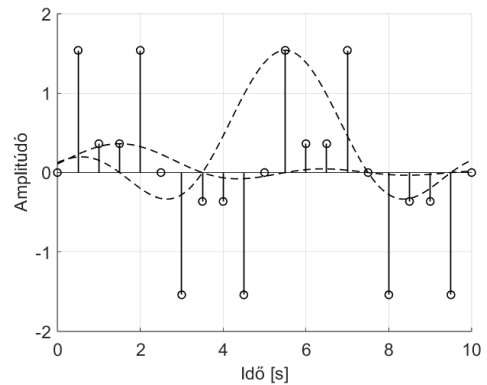
Egyedül a szűrő amplitúdóját kell korrigálni, ami pedig az időtartományban is megoldható, ha minden végső mintát elosztunk N -nel (ha „kinyújtjuk” a szűrőt időtartományban az N -szer olyan hosszúra, akkor egy egységimpulzust megszűrve vele N -szer akkora lesz az amplitúdója, mintha az N -szer „rövidebbel” szűrnénk meg).

Ezután külön osztályban írtam meg az interpoláció és decimáló függvényeimet. Az osztály tartalmazott egy tömböt, amelyik tárolta a mintákat az interpoláció után, még a decimálás előtt.

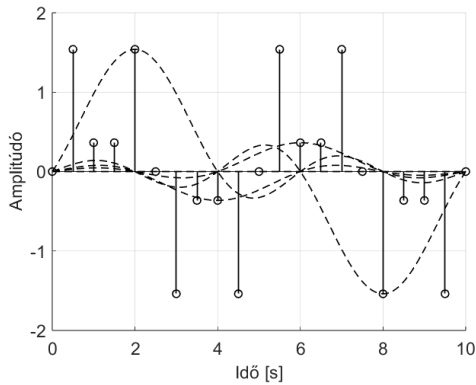
Az osztály emellett rendelkezett egy tagfüggvénnyel, ami egy függvény pointert vett át argumentumként, és az interpolált tömbön lefuttatta.



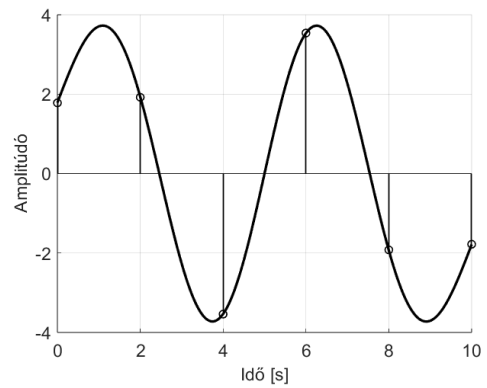
(a) Az eredeti jel, f_s' mintavételi frekvenciával mintavételezve (tartalmaz $f_s/2$ -nél nagyobb frekvenciájú komponens)



(b) A sinc függvény illesztése



(c) Minden N . mintára sinc függvény illesztése



(d) Az új mintavételi pontokon az összes sinc függvényértéket összeadva és elosztva N -nel az f_s mintavételi frekvenciás jel az eredmény.

5.29. ábra. A decimálás értelmezése (5.2.3.1) egyenlet szerint.

Összefoglalva az általam megírt osztály használata a pluginban:

- A plugin inicializáló függvényének futása során generálódik le az argumentumban átvett mintavételi frekvencia alapján a sinc függvény értékei (JUICE környezetben a `void prepareToPlay(double sampleRate, int samplesPerBlock)` függvény)
- A nemlineáris részek előtt meghívom az osztályom `void interpolate(juce::AudioBuffer<float>& buffer)` függvényét, ami segítségével az argumentumban átvett buffert interpolálja, és letárolja az osztály belső változójába
- Meghívom a nemlinearitást modellező algoritmusokat a belső változóra az osztály `void process(std::function<void(juce::AudioBuffer<float>&)> processBlock)` függvénye segítségével
- Decimálom a belső változóban tárolt mintákat az argumentumban átvett, eredeti bufferbe a `void decimate(juce::AudioBuffer<float>& buffer)` függvény segítségével

5.2.4.1. Paraméterek megválasztása

A modellezés célja az eredeti áramkörrel hangzásban való egyezés. Természetesen a legkényesebb objektív megoldás erre az lenne, hogy teljes egyezést valósítsunk meg minden egyes bemeneti jelalak esetén a válasza, viszont ezt a valós idejűség korlátozza. Így nem lehet teljes mértékben objektívan meghatározni a megfelelő paramétereket.

Lehetőség lenne a maszkolási jelenséget vizsgáló kutatások alapján megállapítani az elnyomás mértékét, hogy egy belapolódott komponens ne lehessen hallani a legrosszabb esetben sem. Viszont a belapolódás jelensége összetett, különböző alapharmonikusoknak különböző helyekre lapolódnak be a felharmonikusai, így nehéz a problémára egyetlen számmal leírni a szükséges elnyomás mértékét.

Leggyakrabban ilyenkor meghallgatásos tesztek alapján szokás meghatározni a paramétereket, több, független hallgató véleményének a feldolgozásával. Ez végeredményben egy szubjektív vélemények alapján megállapított paraméterértékeket fog eredményezni.

A dolgozatomban a paraméterek meghatározására a saját szubjektív véleményemet használtam, amelyet több, különböző meghallgatásos teszt alapján formáltam.

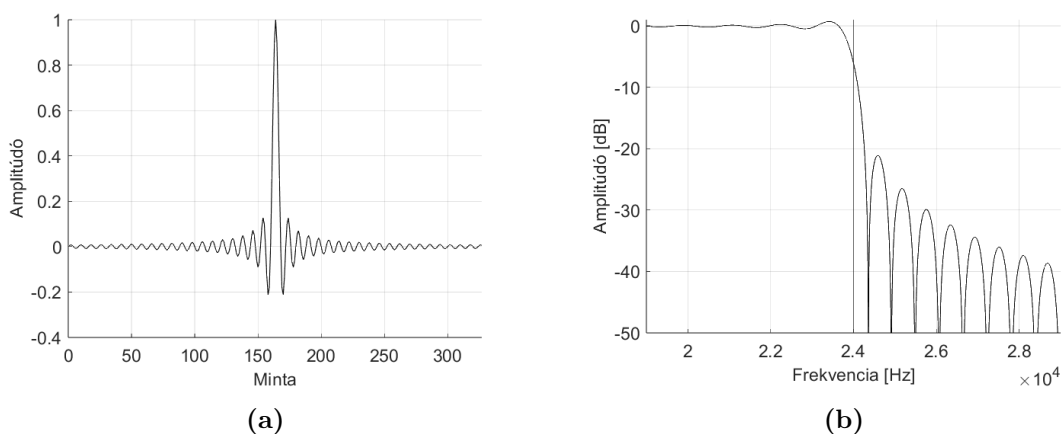
Az általam írt osztály két paraméterrel rendelkezik: az egyik az interpoláció és decimálás mértéke (5.27 ábrán az N változó), és a használt szűrő együtthatóinak száma.

Az N változó meghatározásához az LTSpice szimulációban gerjesztésnek egy 10 kHz szinusz jelet adtam meg, és ez alapján vizsgáltam a belapolódásokat maximális torzítás beállítások mellett.

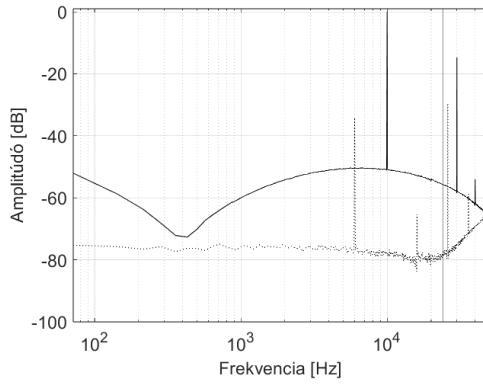
Azért 10 kHz frekvenciájú jelet választottam, hiszen a gitár és a gitár hangszedő általában nem képes ennél nagyobb frekvenciájú hangot átvinni. Átlagos elektromos gitár hangszedők átvitelének a maximum amplitúdója 6–8 kHz környéken van, és ezután meredeken, másodfokú aluláteresztő szűrőként levág [19].

Az 5.30 ábrán a belapolódás látható különböző N értékek esetén. A végső pluginban $N = 4$, amit meghallgatásos tesztek alapján választottam. Ebben az esetben az alapharmonikushoz képest a belapolódott felharmonikusok elnyomása körülbelül -50 dB.

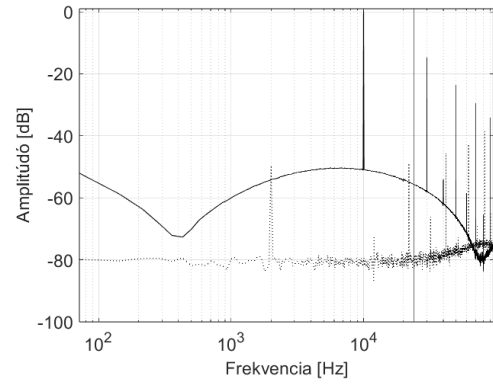
A szűrőegyütthatók számait is meghallgatásos tesztek alapján választottam meg. A szűrőnek 330 együtthatót generáltam le, amelyek az 5.31 ábrán látható, különböző együtthatós szám különbsége az F.2.1 függelékben látható.



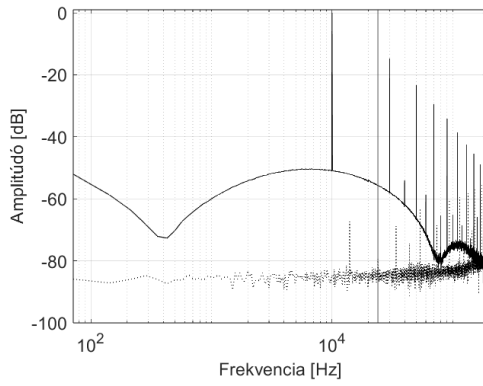
5.31. ábra. A 330 együtthatós szűrő impulzusválasza az 5.31a ábrán, és átvitele az 5.31b ábrán látható.



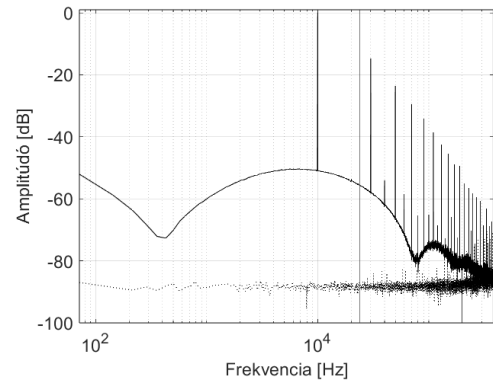
(a) $N = 2$



(b) $N = 4$



(c) $N = 8$



(d) $N = 16$

5.30. ábra. Egy 10 kHz frekvenciájú szinusz jelre az erősítő kimenete maximális torzítás beállítások mellett. Az amplitúdók a kimeneten lévő alapfrekvenciához vannak viszonyítva. A végső pluginban az 5.30b ábrán szemléltetett, $N = 4$ paramétert használtam. Decimálás után az új mintavételi frekvencia f_s , így a jel maximális frekvenciája $f_s/2$.

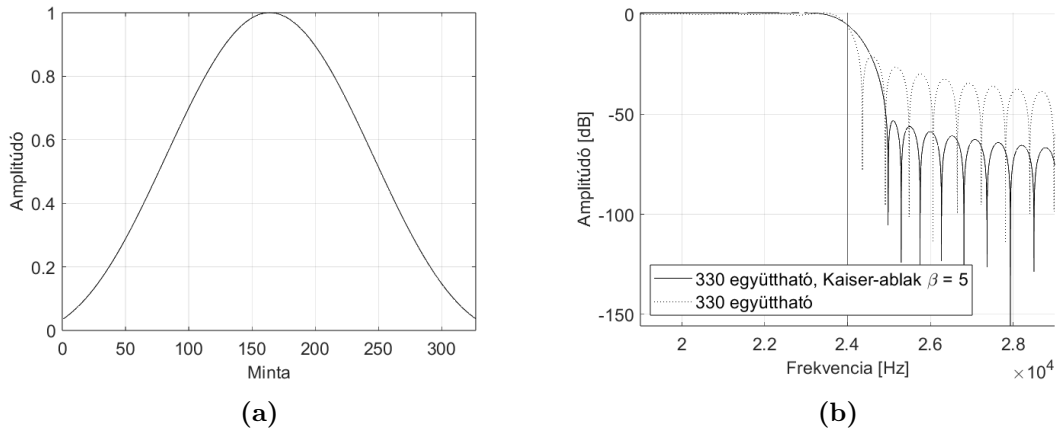
Ahhoz, hogy az $f_s/2$ frekvencia felett még nagyobb elnyomást kapjak [18, 20], a megtervezett sinc szűrőre Kaiser-ablakot alkalmaztam. A Kaiser-ablakot az alábbi képlet definiálja [20]:

$$w_K[n] = \begin{cases} \frac{I_0\left(\beta\sqrt{1 - \left(\frac{n}{M/2}\right)^2}\right)}{I_0(\beta)}, & -\frac{M-1}{2} \leq n \leq \frac{M-1}{2} \\ 0, & \text{egyébként} \end{cases} \quad (5.58)$$

ahol I_0 a nulladrendű módosított Bessel-függvény:

$$I_0(x) \triangleq \sum_{k=0}^{\infty} \left(\frac{(x/2)^k}{k!}\right)^2 \quad (5.59)$$

Így a Kaiser-ablaknak van egy β paramétere, ami $\beta = 0$ esetén megegyezik a négy-szögablakkal. Ezt a paramétert meghallgatásos teszt alapján $\beta = 5$ értéknek választottam meg. A Kaiser-ablak és az alkalmazása után a szűrő az 5.32 ábrán látható.



5.32. ábra. A $\beta = 5$ paraméterű Kaiser-ablak az 5.32a ábrán, és az alkalmazása után a szűrő átvitele az 5.32b ábrán látható .

Látható, hogy a Kaiser-ablak alkalmazása esetén a szűrő nagyobb elnyomást biztosít a zárótartományban, viszont növeli az átmeneti sáv szélességét. További β paraméter értékekkel való tesztelés az F.2.2 függelékben található.

5.3. Paraméterváltozás kezelése

A pluginok paraméterváltozásának kezelése során a választott megoldás jelentős hatással van a számításigényre. Egyszerűbb, kevésbé komplex pluginoknál megengedhető, hogy a paramétereket minden ciklus során újraszámoljuk, függetlenül attól, hogy történt-e változás. Ez a megközelítés egyszerűsíti a struktúrát, viszont számításigényes, és nem mindig praktikus.

Az ebben a dolgozatban bemutatott áramkör összetettebb felépítésű, és jelentősebb számítási igényvel rendelkezik. Ezért a fentebb említett egyszerűsített struktúra nem alkalmazható, mivel nem skálázható hatékonyan.

Emellett a paraméterek közvetlen, azonnali frissítése zajokat okozhat. Ennek oka, hogy a szűrő együtthatóinak hirtelen megváltozása egy ugrást generál az új szűrő bemenetén, amit a kimenet is tükröz, kellemetlenül hallható zaj formájában.

5.3.1. Lehetséges megoldások

A problémák csökkentésére az egyik megközelítés az, hogy a paramétereket nem azonnal frissítjük az új értékekre, hanem fokozatosan, lineárisan változtatjuk őket a célérték felé. Ez a módszer akkor működik hatékonyan, ha a paraméterek csak a szűrő törésponti frekvenciáját, Q jellemzőjét, stb. érintik. A kis lépésekben történő változtatás miatt az energiakülönbség a váltások között csökken, az új szűrő bemenetén kisebb amplitúdójú lesz a váltáskor keletkezett ugrás, így a hiba mértéke is jelentősen kisebb lesz.

A jelenlegi esetben ez a megoldás nem alkalmazható, mivel a paraméterek egyszerre több szűrőegyütthatót befolyásolnak. Ezért a lineáris közelítés nemcsak a szükséges zajmentességet nem garantálja, hanem a szűrőegyütthatók fokozatos, lineáris módosítása a célérték felé instabil rendszerműködést is eredményezhet.

Egy másik megközelítés a crossfade technika alkalmazása. Ebben az esetben a paraméterváltozáskor párhuzamosan futtatjuk a régi és az új szűrőt. Az átmenet során crossfade-et használunk, hogy fokozatosan váltsunk át az új szűrő kimenetére. Ezzel a módszerrel csökkenthetőek az ugrásból adódó hallható zajok, viszont a szűrők futtatásából adódó számítási igény megkétszereződik.

5.3.2. Megvalósítás

A pluginban a crossfade megközelítést alkalmaztam. Minden áramkörészhez példányosítottam egy egyedi osztályt, amely a `juce::AudioProcessorValueTreeState::Listener` osztályból származik. Ez az osztály paraméterváltozás esetén meghívja a konfigurációs függvényt (`void configure(double sampleRate)`), amely kiszámítja az új paraméterértékeket az adott áramkörészhez.

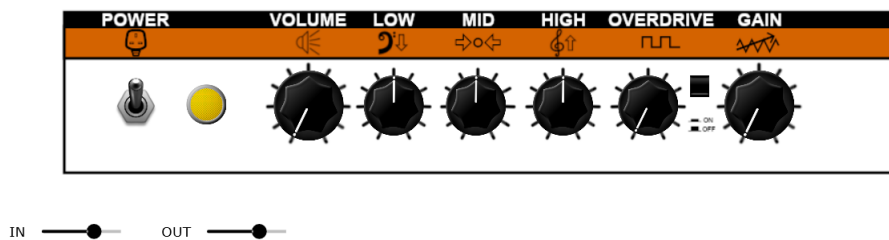
Az új paraméterek kezelésére egy saját osztályt (`ParameterQueue`) használtam, amely biztosítja az olvasási és írási műveletek biztonságát attól függetlenül, hogy ezek éppen melyik szálon történnek. Ez megakadályozza, hogy az éppen végrehajtás alatt álló paraméterfrissítéseket megzavarják az író műveletek, amelyek inkonzisztens adatokat eredményezhetnek.

A crossfade hatást a JUCE `juce::SmoothedValue` osztályának segítségével valósítottam meg. Ez az osztály fokozatosan, lineárisan változó értékeket biztosít, amelyek minden iteráció után közelednek a célérték felé. A crossfade-et csak akkor indítottam el, amikor az előző paraméterváltozás teljesen befejeződött. Ezáltal a közben érkező új értékek nem indítottak újabb crossfade-et, csökkentve a terhelést.

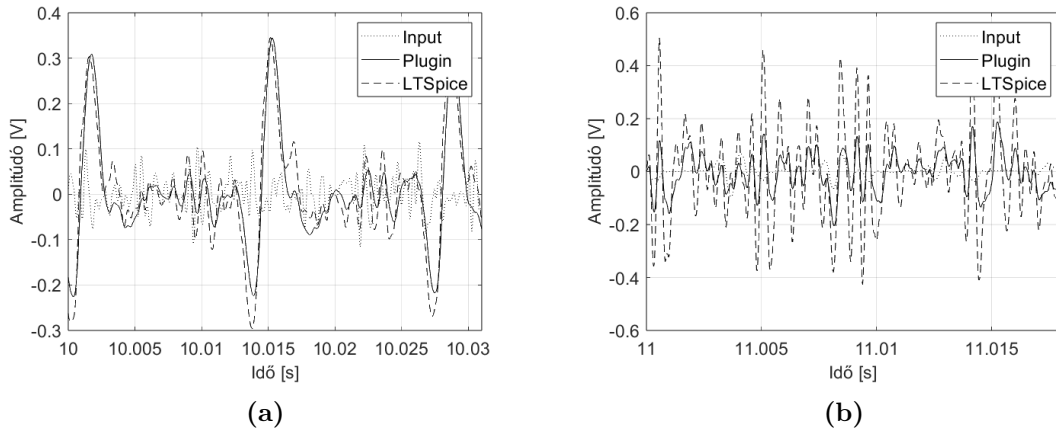
Ez a megoldás biztosította, hogy a plugin paraméterváltozásai zajmentesen történjenek.

5.4. Eredmények

Az erősítő áramkör minden része elkészült, és a plugin többféle jelre vonatkozó tesztelésen esett át, amelyeket Spice szimulációkkal hasonlítottam össze. Egy ilyen összehasonlítás eredménye látható az 5.34 ábrán látható. További összehasonlítások az F.3 függelékben találhatóak, az elkészült plugin felhasználói felülete az 5.33 ábrán látható.



5.33. ábra. Az elkészült plugin felhasználói felülete



5.34. ábra. A plugin és az LTSpice által adott válasz egy elektromos gitár jelére; GAIN: $5\text{ k}\Omega - 5\text{ k}\Omega$, OVERDRIVE kapcsoló kikapcsolva

A plugin elkészítése során megszerzett tapasztalatok:

- A torzító részben, amelyet a K-módszerrel valósítottam meg, található egy OVERDRIVE paraméter, amelynek változása esetén mindig újra kell számolni (a 3.12), (3.13), (3.14) és (3.19) segédváltozókat. Ezen felül az eredeti nemlineáris egyenletet újra lineárisan kell transzformálni, újra kell generálni a táblázatot a K-módszer futtatásához.
- A K-módszerben először a `juce::dsp::Matrix<type>` előre megírt osztályt használtam a mátrixműveletek elvégzéséhez. Viszont ennek az osztálynak a használatával a plugin nem működött megfelelően, folyamatosan recsegő hangok voltak hallhatóak a kimenet mellett. A saját mátrix osztályom megírása és használata után ez a probléma megoldódott.
- A tone stack résznél a sok paraméter miatt nagyon körülményes az átvitel felírása.
- A modell a valóságot szimulálja, így a valóság feszültségszintjei szerint kell a bemenő és a kimenő jelet értelmezni. Ezt úgy oldottam meg, hogy megmértem oszcilloszkóppal egy gitárjel maximum amplitúdóját, és ehhez képest skáláztam a bemenő jelet (körülbelül $[-1.5, 1.5]$ V). A kimeneten megvalósított feszültséglevágás miatt a kimenetet pedig csak egyszerűen leosztottam, hogy minden esetben a digitális jelszintnek feleljen meg $[-1, 1]$.
- A kimenetet előállító műveleti erősítőnek is figyelembe vettem a levágását a kimenet egyszerűbb skálázása érdekében.
- A K-módszer táblázatába ≈ 250 pontpárt generáltattam le, és ez a mennyiség egyáltalán nem okozott problémát a futás közben. A táblázat elemei között elég volt lineárisan interpolálni, nem volt szükséges magasabb fokú interpolációra.
- A legenerált ≈ 250 pontpárt nem egyenletes lépésközzel generáltam le. Mivel a nemlineáris rész egy ellenállás és két diódának soros kapcsolatából állt, így a diódáknak az átvitelre való hatása a $[-2, 2]$ V intervallumra korlátozódott. Itt a pontpárokat sűrű lépésközzel generáltam le, a tartományon kívül pedig csak néhány, $\approx 3-3$ pontpárt generáltam.
- Szükség volt a torzítás okozta felharmonikusok miatt túlmintavételezésre az átlapolódás elkerülése érdekében. Az átlapolódás további csökkentése érdekében érdemes volt Kaiser-ablak alkalmazása a túlmintavételezéshez használt szűrőre.

6. fejezet

Eredmények, tapasztalatok, jövőbeli tervek

A dolgozat célkitűzése az Orange Crush 20L gitárerősítő valós idejű digitális modellezése, és ennek alapján egy VST3 plugin írása volt. Emellett cél volt a K-módszer ismertetése, és más állapotteres módszerek kapcsolatának, hasonlóságainak és különbségeiknek feltárása, illetve összehasonlítása.

6.1. Összehasonlítás

A K-módszer összehasonlítása a leggyakoribb állapotteres módszerrel, a széleskörűen használt hátralépő Euler-módszerrel, megállapításra kerültek az alábbiak:

- A K-módszer számításigénye a gyakorlati alkalmazások esetén gyakran kisebb, mint a hátralépő Euler-módszer számításigénye
- A K-módszer kisebb számításigény mellett pontosabb, mint a hátralépő Euler-módszer (bilineáris transzformáció Euler-féle közelítéssel szemben)
- A K-módszer lehetőséget ad bármilyen polinom hányados használatára a derivált közelítéséhez
- A még kisebb számítási igény eléréséhez lehetőség van az előre kiszámolt tömb tárolásának optimalizálására

A K-módszer hátrányai a hátralépő Euler-módszerrel szemben:

- Ha az áramkörben találhatóak változtatható paraméterek, akkor minden egyes változásnál a K-módszer esetén le kell újra generálni az egyenletmegoldáshoz használt tömböt
- A K-módszer esetén a memóriában a futás során végig tárolni kell az offline legenerált tömböt
- Az offline generálás előtt meg kell vizsgálni a függvényt reális nagyságú bemenetekre, hogy a legenerált tömb értelmezési tartománya megfelelő nagyságú legyen

Így összességében elmondható, hogy az előrelépő Euler-módszer alkalmazása helyett javasolt K-módszert alkalmazni a pontosság javulása és a számítási igény csökkenése érdekében.

6.2. Az Orange Crush 20L erősítő valós idejű szimulálása K-módszerrel

A K-módszer demonstrálásához az Orange Crush 20L gitárerősítő áramkörét modelleztem VST3 környezetben a JUCE keretrendszer segítségével. Az erősítő minden része elkészült, és a Spice szimulációval összehasonlítva hasonló az eredmény, hallgatásra nehezen megkülönböztethető. A dolgozatban kitűzött célokat elértnek tekintem.

Az elkészült plugin különböző bemenetekre való tesztelése, és összehasonlítása LTSpice szimulációkkal az F.3 fejezetben található. A dolgozat mellékletében találhatóak hangfájlok, amelyek a plugin és az LTSpice által lettek generálva, összehasonlítás céljából.

A plugin forráskódja, telepítője és VST3 forrásfájlja megtalálható a <https://github.com/jedlamartin/Crush20L> oldalon.

6.3. Jövőbeli tervek

- A K-módszer vizsgálata a derivált magasabb fokszámú közelítése esetén
- A K-módszer különböző változatainak vizsgálata (Nodal K-method, DK-method), majd összehasonlításuk a K-módszerrel
- Az állapotteres módszerektől különböző megközelítésű módszerek megismerése (hullám digitális szűrő módszer, Wiener-Hammerstein módszer)
- Az állapotteres módszerek összehasonlítása egyéb módszerekkel (hullám digitális szűrő módszer, fekete doboz módszerek)
- A K-módszer különböző felhasználási területeken alkalmazásának kutatása
- A pluginban megvalósítani hangláda impulzusválaszának importálását *.wav* fájlformátumból
- További áramkörök (erősítők, effekt pedálok) valós idejű digitális modellezése

Köszönetnyilvánítás

Szeretnék köszönetet mondani **Dr. Bank Balázsnak**, a Budapesti Műszaki és Gazdaságtudományi Egyetem Mesterséges Intelligencia és Rendszertervezés Tanszék egyetemi docensének, a dolgozatom elkészítéséhez nyújtott motivációjáért és támogatásáért.

A heti rendszerességű személyes és online konzultációk jelentős mértékben hozzájárultak a cél eléréséhez, ötletei és tanácsai pedig mindig új perspektívákat nyitottak meg, ösztönözve a kitartó és lelkes munkára. Egyetlen általa ajánlott cikk vagy akár egy rövid megjegyzés is olyan útra terelt, amely folyamatosan rávilágított arra, milyen izgalmas új tudást felfedezni és elsajátítani.

Valószínűleg az ő segítségével ez a dolgozat nem jöhetett volna létre, és a kutatási munka sem valósulhatott volna meg.

Irodalomjegyzék

- [1] Julius O. Smith. *Physical Audio Signal Processing*. <http://ccrma.stanford.edu/jos/-pasp/>, 2010.
- [2] David T. Yeh, Jonathan S. Abel, Andrei Vladimirescu, and Julius O. Smith. Numerical methods for simulation of guitar distortion circuits. *Computer Music Journal*, 32(2):23–42, 2008.
- [3] Andrés Granados. Implicit runge-kutta algorithm using newton-raphson method. In *IV Congreso Internacional de Métodos Numéricos en Ingeniería y Ciencias Aplicadas, CIMENICS'98*, volume IV, 03 1998.
- [4] David Medine. Dynamical systems for audio synthesis: Embracing nonlinearities and delay-free loops. *Applied Sciences*, 6(5), 2016.
- [5] J. E. Dennis and Robert B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Society for Industrial and Applied Mathematics, 1996.
- [6] Zheng Tan, Delin; Chen. On a general formula of fourth order runge-kutta method. *Journal of Mathematical Science & Mathematics Education*, 7(2), 2012.
- [7] Gianpaolo Borin. Metodi per la sintesi per modelli fisici. Technical report, Università degli Studi di Pavia, October 1998.
- [8] G. Borin, G. De Poli, and D. Rocchesso. Elimination of delay-free loops in discrete-time models of nonlinear acoustic systems. In *Proceedings of 1997 Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 4 pp.–, 1997.
- [9] David T. Yeh, Jonathan S. Abel, and Julius O. Smith. Automated physical modeling of nonlinear audio circuits for real-time audio effects—part i: Theoretical development. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(4):728–737, 2010.
- [10] Rafael Paiva. *Circuit modeling studies related to guitars and audio processing*. PhD thesis, 11 2013. Thesis for: Doctor in Science, Advisor: Vesa Välimäki.
- [11] Intel Corporation. *Intel® 64 and IA-32 Architectures Software Developer’s Manual, Volume 2 (Instruction Set Reference)*. Intel Corporation, 2021.
- [12] John L. Hennessy and David A. Patterson. *Computer Architecture: A Quantitative Approach*. Elsevier, Burlington, MA, 6th edition, 2019.
- [13] David A. Patterson and John L. Hennessy. *Computer Organization and Design: The Hardware/Software Interface*. Morgan Kaufmann, Burlington, MA, 5th edition, 2014.
- [14] Citizendium. Newton’s method. http://www.citizendium.org/wiki/Newton's_method, 2009.

- [15] Leete Timothy, Tarr Eric, and Ko Doyuen. Non-ideal operational amplifier emulation in digital model of analog distortion effect pedal. *Journal of the Audio Engineering Society*, 05 2022.
- [16] Khaled Abdelfattah and Behzad Razavi. Modeling op amp nonlinearity in switched-capacitor sigma-delta modulators. In *IEEE Custom Integrated Circuits Conference 2006*, pages 197–200, San Jose, CA, USA, 2006.
- [17] Julius O. Smith. *Mathematics of the Discrete Fourier Transform (DFT)*. W3K Publishing, <http://www.w3k.org/books/>, 2007.
- [18] Julius O. Smith. *Digital Audio Resampling Home Page*. <http://www-ccrma.stanford.edu/jos/resample/>, January 28 2002.
- [19] Peter D. Hiscocks. The frequency response of guitar pickups. <https://www.syscompdesign.com/wp-content/uploads/2018/09/guitar-pickups.pdf>, 2018. Accessed: 2024-11-30.
- [20] Julius O. Smith. *Spectral Audio Signal Processing*. <http://ccrma.stanford.edu/~jos/sasp/>, 2024. online book, 2011 edition.

Függelék

F.1. K-módszer a hátralépő Euler-módszerben használt diszkretizálással

K-módszer esetén az egyenletrendszert (3.4) alakban kell felírni.

$$s\mathbf{W}(s) = \mathbf{A}\mathbf{W}(s) + \mathbf{B}\mathbf{U}(s) + \mathbf{C}\mathbf{Y}(s) \quad (\text{F.1.1})$$

$$\frac{1 - z^{-1}}{T}\mathbf{W}(s) = \mathbf{A}\mathbf{W}(s) + \mathbf{B}\mathbf{U}(s) + \mathbf{C}\mathbf{Y}(s) \quad (\text{F.1.2})$$

$$\frac{1}{T}\mathbf{W}(z) - \mathbf{A}\mathbf{W}(z) = \frac{z^{-1}}{T}\mathbf{W}(z) + \mathbf{B}\mathbf{U}(z) + \mathbf{C}\mathbf{Y}(z) \quad (\text{F.1.3})$$

$$\left(\frac{1}{T}\mathbf{I} - \mathbf{A}\right)\mathbf{W}(z) = z^{-1}\frac{1}{T}\mathbf{W}(z) + \mathbf{B}\mathbf{U}(z) + \mathbf{C}\mathbf{Y}(z) \quad (\text{F.1.4})$$

Ezt z-tartományból visszaalakítva:

$$\left(\frac{1}{T}\mathbf{I} - \mathbf{A}\right)\mathbf{w}[n] = \frac{1}{T}\mathbf{w}[n-1] + \mathbf{B}\mathbf{u}[n] + \mathbf{C}\mathbf{y}[n] \quad (\text{F.1.5})$$

Új változókat bevezetve:

$$\mathbf{H} \triangleq \left(\frac{1}{T}\mathbf{I} - \mathbf{A}\right)^{-1}\frac{1}{T} \quad (\text{F.1.6})$$

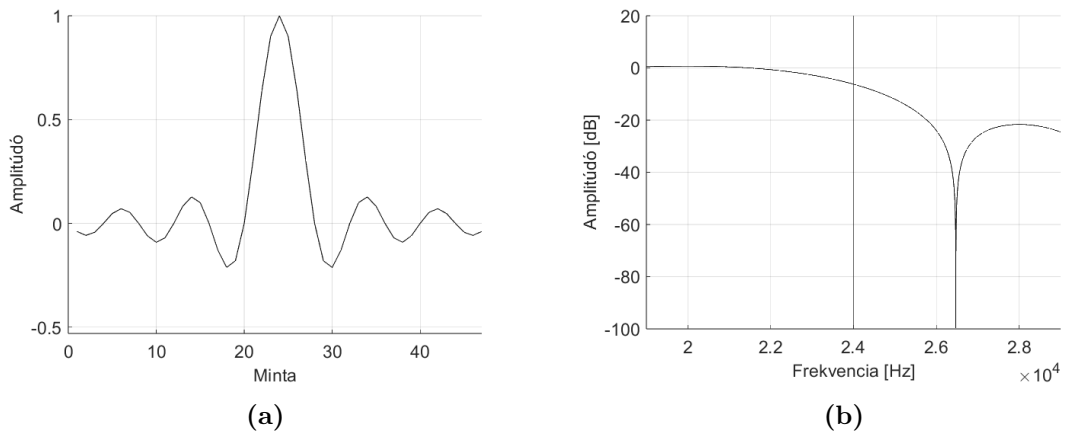
$$\mathbf{J} \triangleq \left(\frac{1}{T}\mathbf{I} - \mathbf{A}\right)^{-1}\mathbf{B} \quad (\text{F.1.7})$$

$$\mathbf{G} \triangleq \left(\frac{1}{T}\mathbf{I} - \mathbf{A}\right)^{-1}\mathbf{C} \quad (\text{F.1.8})$$

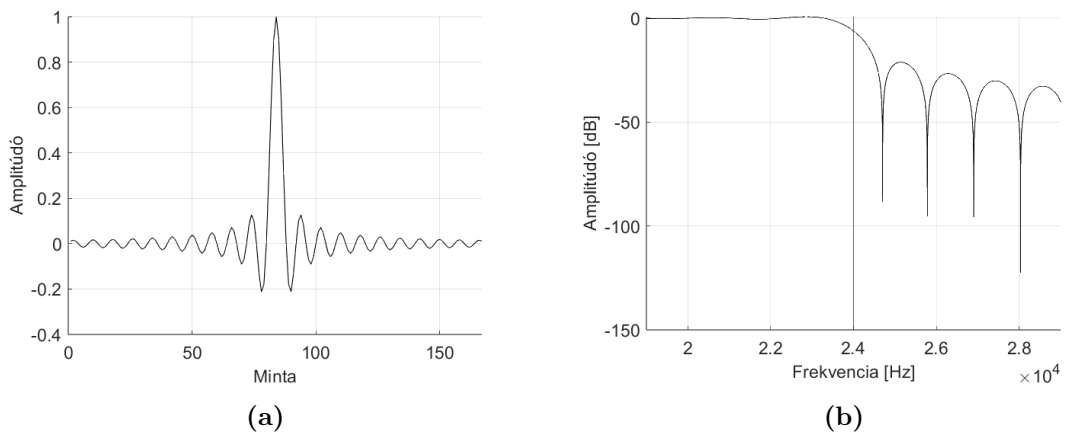
$$\mathbf{w}[n] = \mathbf{H}\mathbf{w}[n-1] + \mathbf{J}\mathbf{u}[n] + \mathbf{G}\mathbf{y}[n] \quad (\text{F.1.9})$$

F.2. A túlmintavételezéshez használt szűrő összehasonlítása különböző paraméterek esetén

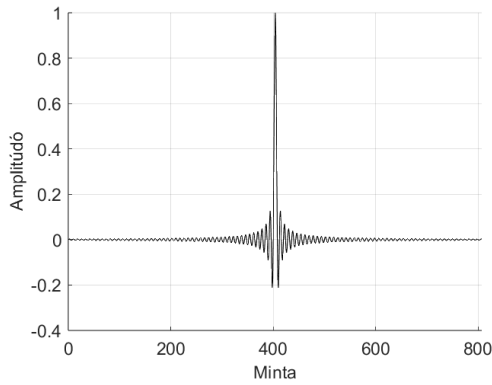
F.2.1. A szűrő együtthatószáma



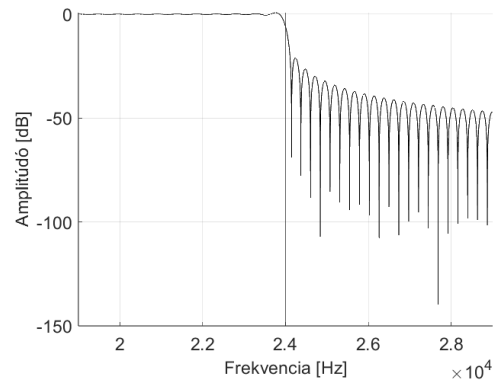
F.2.1. ábra. 50 együtthatós szűrő impulzusválasza az F.2.1a ábrán, átviteli függvénye az F.2.1b



F.2.2. ábra. 170 együtthatós szűrő impulzusválasza az F.2.2a ábrán, átviteli függvénye az F.2.2b

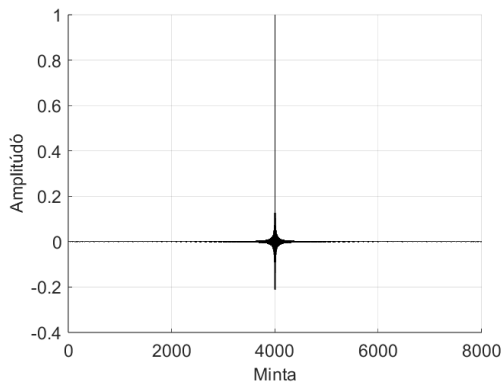


(a)

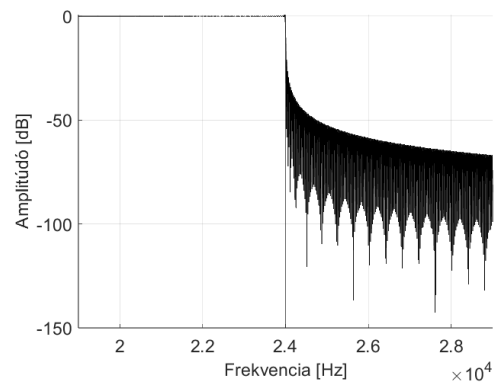


(b)

F.2.3. ábra. 810 együtthatós szűrő impulzusválasza az F.2.3a ábrán, átviteli függvénye az F.2.3b



(a)

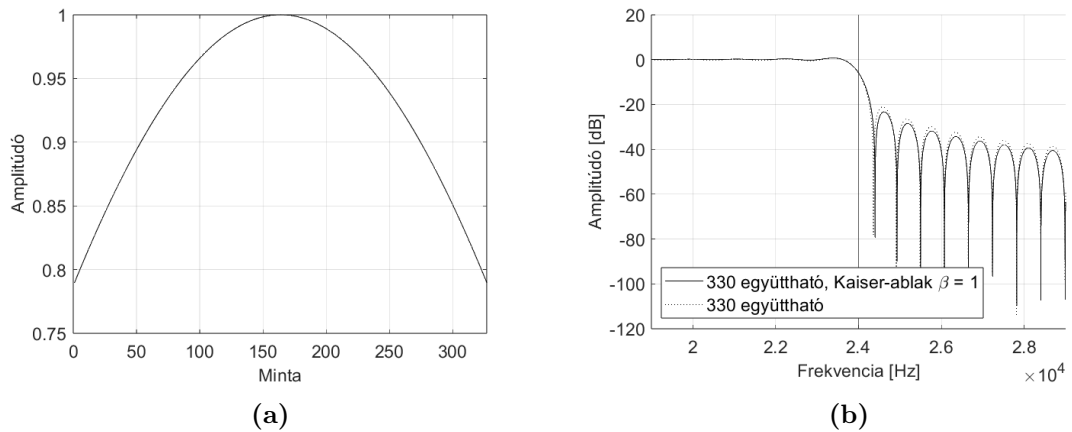


(b)

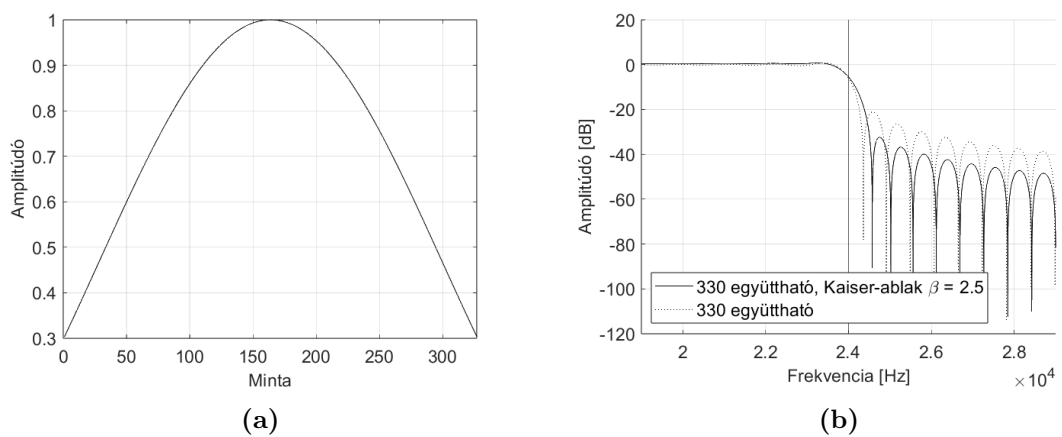
F.2.4. ábra. 8010 együtthatós szűrő impulzusválasza az F.2.4a ábrán, átviteli függvénye az F.2.4b

Látható, hogy a szűrő együtthatóinak számának növelésével a szűrő egyre meredekebben vág le, és a zárótartományban nagyobb elnyomást eredményez. A valós idejűség miatt szükséges olyan optimális együtthatószámot találni, amelynek elég nagy az elnyomása a zárótartományban ahhoz, hogy a nem kívánt felharmonikusokat ne lehessen hallani, miközben az együtthatószáma minimális.

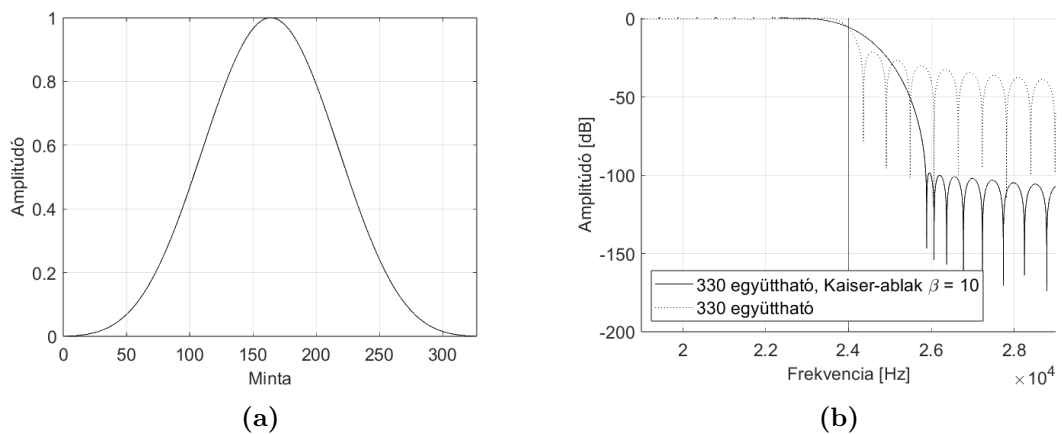
F.2.2. A különböző β együtthatós Kaiser-ablakok összehasonlítása



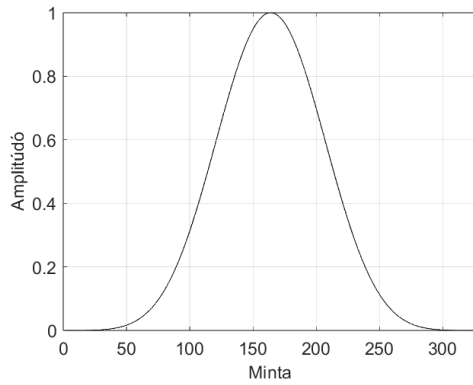
F.2.5. ábra. Kaiser-ablak $\beta = 1$ paraméterrel F.2.5a ábrán, és annak alkalmazása a választott 330 együtthatós szűrőre F.2.5b ábrán.



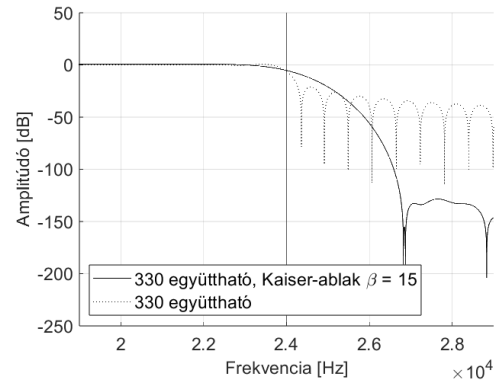
F.2.6. ábra. Kaiser-ablak $\beta = 2.5$ paraméterrel F.2.6a ábrán, és annak alkalmazása a választott 330 együtthatós szűrőre F.2.6b ábrán.



F.2.7. ábra. Kaiser-ablak $\beta = 10$ paraméterrel F.2.7a ábrán, és annak alkalmazása a választott 330 együtthatós szűrőre F.2.7b ábrán.



(a)

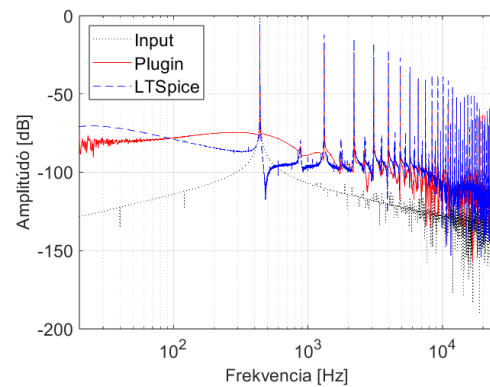
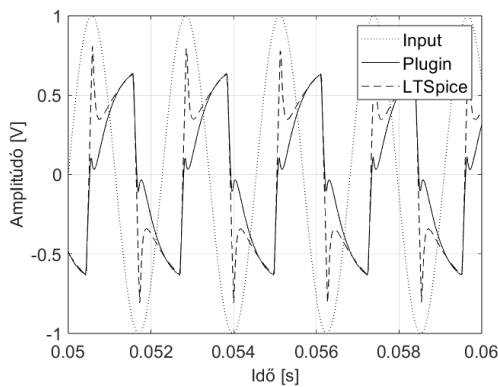


(b)

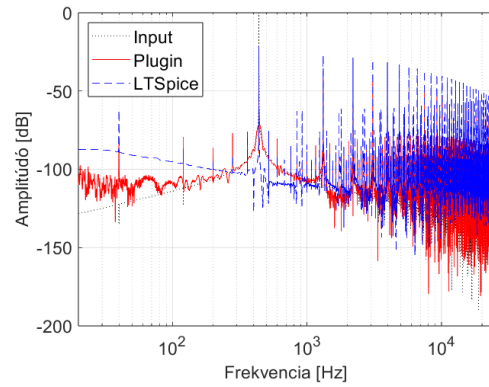
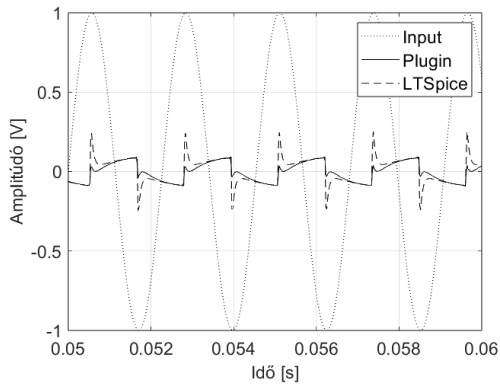
F.2.8. ábra. Kaiser-ablak $\beta = 15$ paraméterrel F.2.8a ábrán, és annak alkalmazása a választott 330 együtthatós szűrőre F.2.8b ábrán.

Látható, hogy a β paraméter növelésével a túlmintvételezéshez használt szűrő zárótartományában nagyobb az elnyomás, viszont az átmeneti sáv szélesebb.

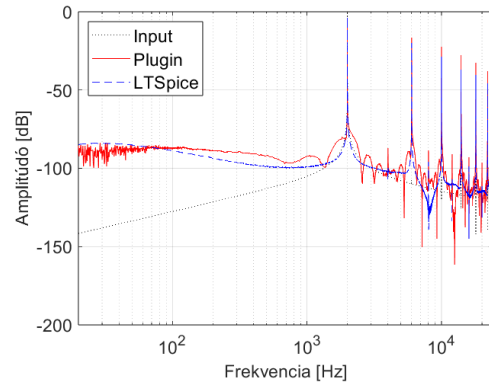
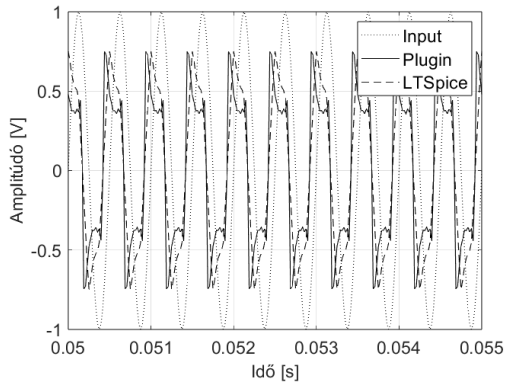
F.3. Az elkészült plugin tesztelése és összehasonlítása LTSpice szimulációkkal



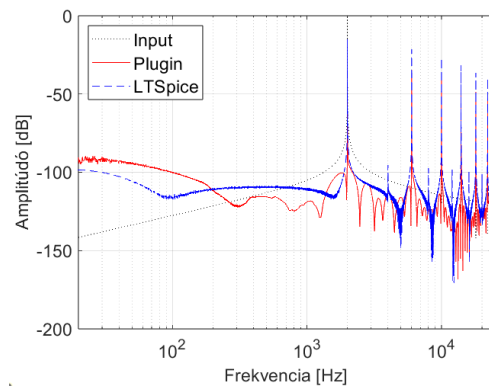
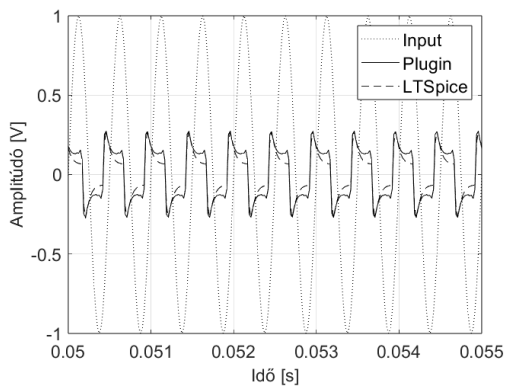
F.3.1. ábra. 1 V amplitúdójú, 440 Hz frekvenciájú szinusz jel gerjesztés, GAIN: 5 k Ω – 5 k Ω , OVERDRIVE kapcsoló kikapcsolva



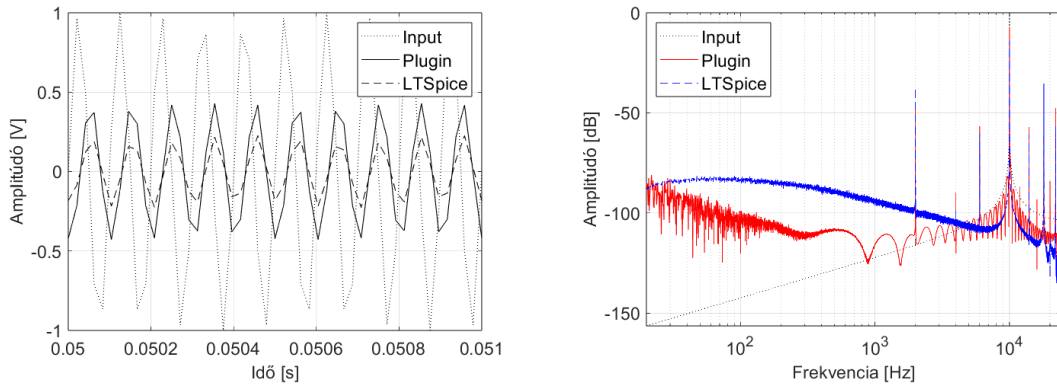
F.3.2. ábra. 1 V amplitúdójú, 440 Hz frekvenciájú szinusz jel gerjesztés, GAIN: maximum, OVERDRIVE: maximum, kapcsoló bekapcsolva



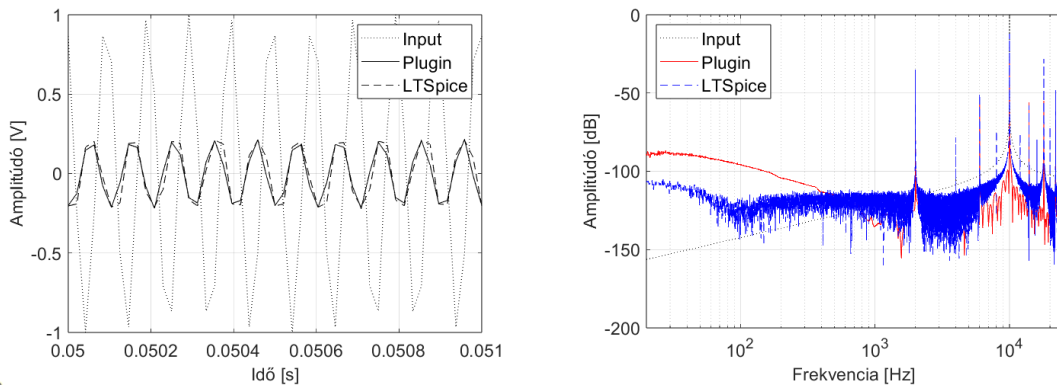
F.3.3. ábra. 1 V amplitúdójú, 2000 Hz frekvenciájú szinusz jel gerjesztés, GAIN: 5 k Ω – 5 k Ω , OVERDRIVE kapcsoló kikapcsolva



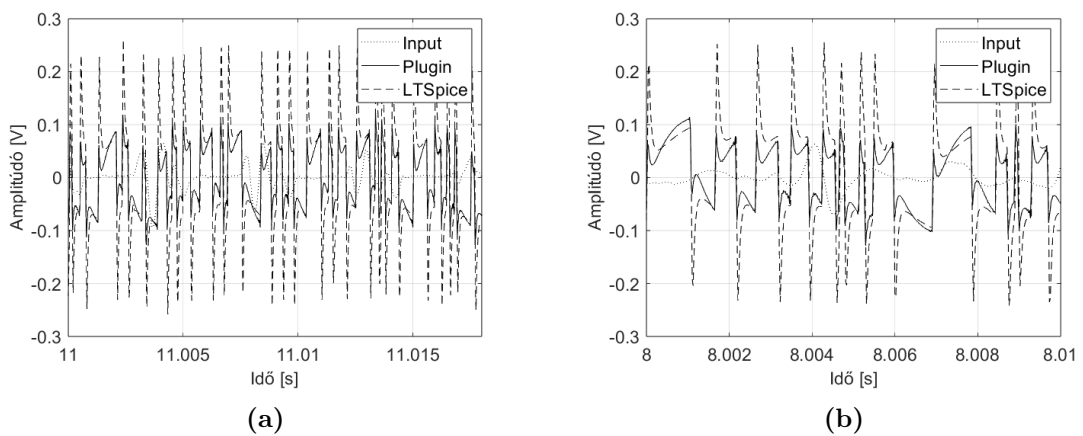
F.3.4. ábra. 1 V amplitúdójú, 2000 Hz frekvenciájú szinusz jel gerjesztés, GAIN: maximum, OVERDRIVE: maximum, kapcsoló bekapcsolva



F.3.5. ábra. 1 V amplitúdójú, 10000 Hz frekvenciájú szinusz jel gerjesztés, GAIN: $5\text{ k}\Omega$ – $5\text{ k}\Omega$, OVERDRIVE kapcsoló kikapcsolva



F.3.6. ábra. 1 V amplitúdójú, 10000 Hz frekvenciájú szinusz jel gerjesztés, GAIN: maximum, OVERDRIVE: maximum, kapcsoló bekapcsolva



F.3.7. ábra. A plugin és az LTSpice által adott válasz egy elektromos gitár jelére; GAIN: maximum, OVERDRIVE: maximum, OVERDRIVE kapcsoló bekapcsolva

A plugin kimenete és az LTSpice szimulációk között láthatóak különbségek. A legnagyobb eltérések azoknál a részeknél láthatóak, amelyeken meredek a felfutása vagy lefutása a jelnek.

Emellett az LTSpice szimulációs eredményekben megfigyelhető az abban használt műveleti erősítők nemideális viselkedése, például a slew rate hatása (F.3.3 ábrán), valamint a frekvenciafüggő viselkedésük (F.3.5 ábrán).

A plugin kimenete és az LTSpice szimuláció eredménye közötti különbség nem hallható.