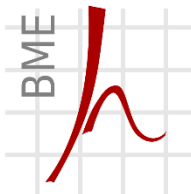


Parametrikus equalizer megvalósítása C++ környezetben

Készítette: Bárány Bálint

Konzulens: Rucz Péter



Híradástechnikai Tanszék



M Ű E G Y E T E M 1 7 8 2

Tartalomjegyzék

1. Bevezetés.....	3
1.1 Feladat kitűzése	3
2. Az EQ működése	3
2.1 Equalizer típusok	
[http://www.homerecordingconnection.com/news.php?action=view_story&id=144].....	3
2.1.1 Fix EQ.....	3
2.1.2 Sweepelhető középsávval rendelkező EQ	3
2.1.3 Parametrikus EQ.....	3
2.1.4 Grafikus EQ	4
2.2 Szükséges szűrők [http://www.musicdsp.org/files/Audio-EQ-Cookbook.txt]	4
3. A C++ és VST környezet.....	5
3.1 VST működése	5
3.2 VST effekt	5
3.3 Az effekt kódjának felépítése.....	6
3.3.1 Controller	6
3.3.2 Editor	6
3.4 DSP filters	6
3.4.1 A szűrők függvényei	7
4. A plugin használata	8
4.1 A plugin működés közben.....	8
4.1.1 Tesztelés hangokkal.....	9
4.1.2 Tesztelés VST plugin analyserrel	10
5. Összegzés	10
6. Hivatkozás jegyzék.....	11

1. Bevezetés

A hangfelvételek készítése során szeretnénk egy hangforrást úgy rögzíteni, hogy annak mintavett formája minél hűebben reprezentálja az eredeti jelet. Az egyes felvevő rendszerek azonban sokszor torzíthatják a rajtuk áthaladó jeleket, így megjelenik az igény az átviteli karakterisztika egyfajta korrigálására, amit praktikusán megtehetünk egy – a jelútba iktatott - processzási folyamat hozzáadásával - egy általunk paraméterezett szűrő formájában.

Persze a fenti alkalmazási forma nem az egyetlen, amikor hasonló jellegű jelfeldolgozást szeretnénk végezni. A hangszínszabályozás fontos eleme a zenei produkcióknak, ahol a hangszerek egymáshoz képesti megfelelő megszólalást szeretnénk elérni. Ugyanígy fontos egy rendezvény alkalmával a helyszínen egyedileg összeállított hangrendszer megfelelő hangzásának elérése, a helyszín egyedi akusztikáját figyelembe véve.

1.1 Feladat kitűzése

A témám kiválasztásakor egy, a hangszínszabályozást alapszinten elvégző szűrő megvalósítását terveztem. A szűrő szoftveresen megvalósított, VST plugin formátumú. Ez azt jelenti, hogy különböző audio szoftverek (DAW – Digital Audio Workstation) képesek felhasználni. Ez a fent említett felhasználási területek közül a szoftveres hangeditálás – zenei produktum készítés körében teszi használhatóvá.

2. Az EQ működése

2.1 Equalizer típusok [1]

2.1.1 Fix EQ

Ez a legegyszerűbb típusa az equalizereknek. Analóg eszközökbe olcsóság és egyszerűség miatt használják fel, ugyanis ez a legkevesebb paraméter állítást lehetővé tevő EQ. Előre definiált frekvenciatartományok erősítését állíthatjuk benne. Tipikus példája a zenekari keverőkön található mély-közép-magas hangszínszabályzó.

2.1.2 Sweepelhető középsávval rendelkező EQ

Ebben a kategóriában olyan eszközök vannak, amelyeknél az előző típus képességein felül egy vagy két középső frekvenciatartománybeli szűrőnek a frekvenciáját is állíthatjuk, ami egy precízebb korrekciót tesz lehetővé.

2.1.3 Parametrikus EQ

Az általam választott feladatban egy ilyen került megvalósításra. Itt minden szűrőnek állítható a frekvenciája, és például bandpass szűrő esetén a sáv szélessége is. Ez az egyik legprecízebb fajta.

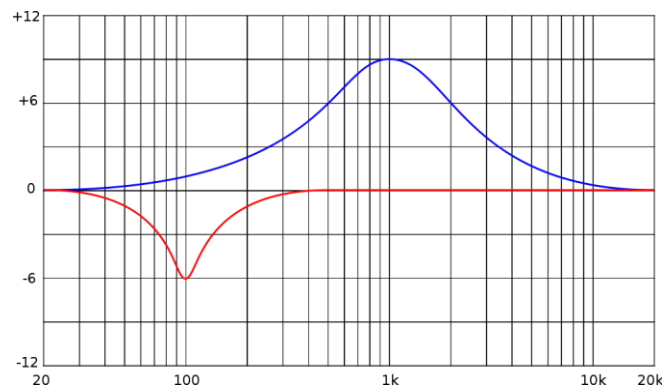
2.1.4 Grafikus EQ

Ezek az eszközökön rengeteg slider található, amik egy-egy specifikus frekvencián állítják az erősítést. Gyakran használják keverő konzol kimenetén, a hang karakterének apró változtatásához.

2.2 Szükséges szűrők [2]

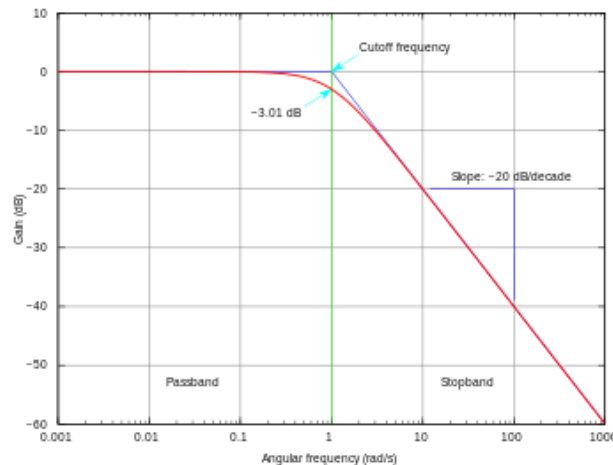
Egy parametrikus EQ ideális használatához háromféle szűrőre van szükség. Aluláteresztő (lowpass), felüláteresztő (highpass), és egy úgynevezett band shelf szűrőre, amiknek a karakterisztikája az ábrákon látható módon alakul.

$$H(s) = \frac{s^2 + \frac{\sqrt{A}}{Q} * s + A}{A * s^2 + \frac{\sqrt{A}}{Q} * s + 1}$$



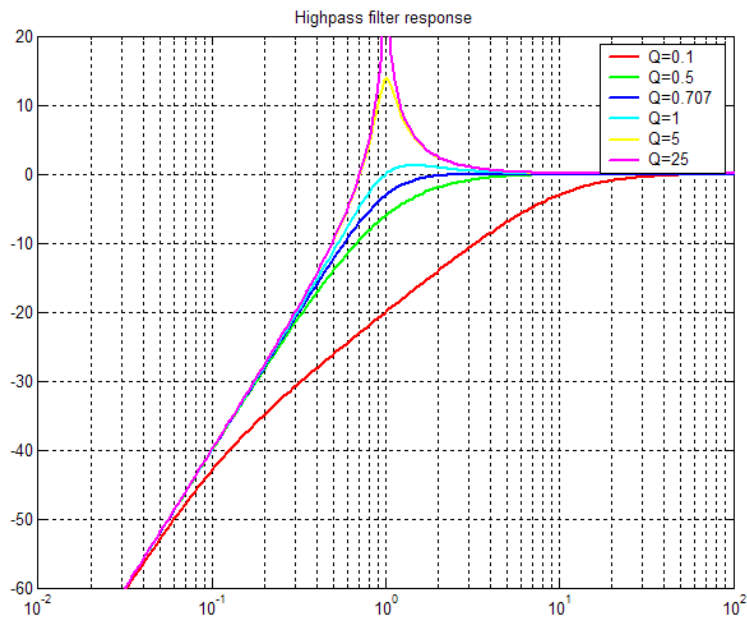
2.1 ábra: Band shelf szűrő átviteli karakterisztikája, és az amplitúdó karakterisztika ábrázolva. (késsel 9 dB-es erősítés 1 kHz-en, pirossal 6 dB-es vágás 100 Hz-en, nagyobb jósági tényezővel)[3]

$$H(s) = \frac{1}{s^2 + \frac{s}{Q} + 1}$$



2.2 ábra: Lowpass szűrő átviteli függvény és karakterisztika [4]

$$H(s) = \frac{s^2}{s^2 + \frac{s}{Q} + 1}$$



2.3 ábra: Highpass szűrő átviteli függvény és amplitudó karakterisztika többféle jóági tényezőre [5]

3. A C++ és VST környezet

A jelfeldolgozás működését leíró programkód C++ nyelven íródott, Microsoft Visual Studio 2010-el. A program lefordítását követően a kimenet egy vst3 (VST 3.0 formátum esetén), vagy dll (vst 2.0 formátum esetén) kiterjesztésű fájl lesz.

3.1 VST működése

A VST (Virtual Studio Technology) egy olyan technológia, amit a Steinberg nevű cég fejlesztett ki, és lehetővé teszi, hogy virtuális hangszerek és effektek használhatóak legyenek minden olyan audio szoftverrel, amelyek ezt támogatják. A félév során megvalósított plugin az effekt kategóriába tartozik.

3.2 VST effekt

A plugin használó program (host) a VST betöltésekor beiktatja a processzási folyamatba, a diszkrét idejű mintavett jelet továbbítja neki, majd ugyanilyen diszkrét mintákat vár vissza a kimeneten. Az audio bemenet(ek)en kívül lehetőség van MIDI jelet is küldeni az effekt modulok számára, ezzel még több funkciót megvalósítva. A pluginok által tartalmazott paraméterek egy teljesen tetszőlegesen testreszabható kezelőfelületen jelennek meg. Ezeket a paramétereket a DAW szoftver számára is láthatóvá tehetjük, ezzel lehetővé téve, hogy a host

alkalmazás vezérelje – így valósítható meg például az automatizálás, vagy definiálható egy olyan virtuális potnméter, amivel egyszerre több, különböző paramétert változtatunk.

3.3 Az effekt kódjának felépítése

Egy VST alkalmazás működésének megértéséhez kiváló segítséget nyújt a VST3 SDK csomag tartalma, ugyanis mellékeltek néhány minta alkalmazást is [6]. A plugin nevét viselő cpp fájlban található az a programrész, ami végzi a jelfeldolgozást – azaz a hangmintákat a megfelelő átviteli függvények segítségével processzálja, mindezt úgy, hogy figyelembe veszi a felhasználó által a kezelőfelületen beállított paraméter értékeket.

3.3.1 Controller

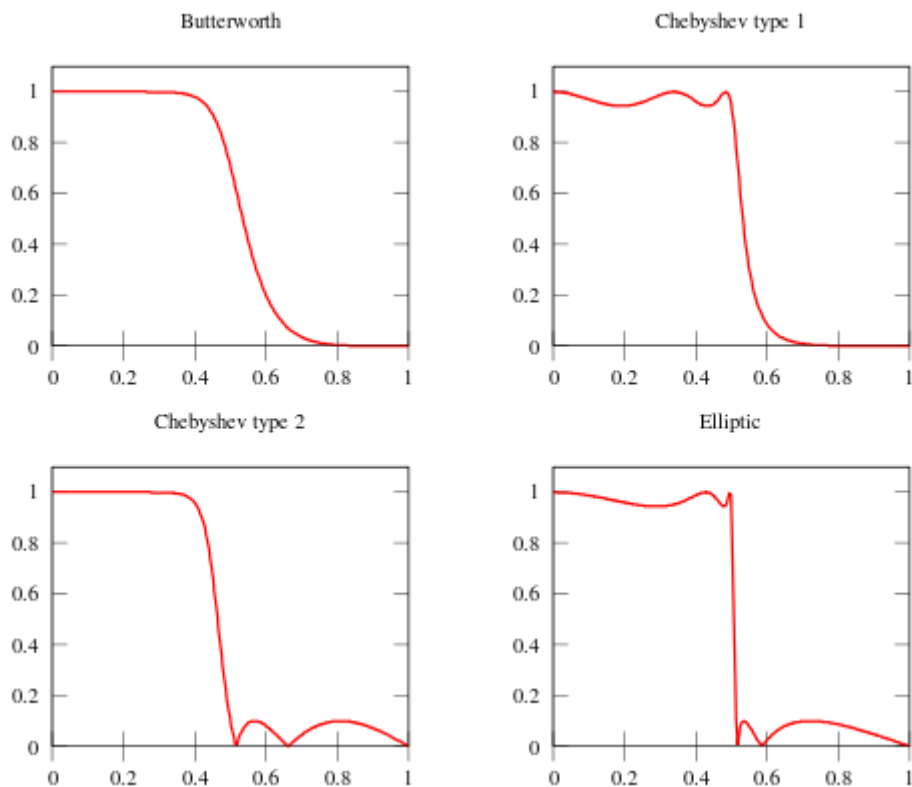
A controller nevű forrásfájl tartalmazza a paraméterek állításával kapcsolatos feladatokat. Amikor betöltjük a plugint, akkor definiálja a paramétereket, és inicializálja az értékeiket, valamint a beérkező MIDI információ is itt kerül feldolgozásra.

3.3.2 Editor

A grafikus megjelenés, és az ott beállított értékek értelmezése az editor feladata. A források közé tetszőleges grafikai elemet csatolhatunk, amit itt konfigurálhatunk, pixel pontossággal meghatározva a megjelenési helyét és funkcióját.

3.4 DSP filters

A szűrést elvégző programrészt egy interneten szabadon elérhető könyvtár[7] tartalmazza. 2009 óta van online, és többféle IIR szűrőtípust használhatunk vele, azokat tetszés szerint paraméterezve. (Butterworth, Chebyshev, Elliptic).



1.ábra: Különböző típusú aluláteresztő szűrők[8]

Különböző osztályok segítségével hozhatunk létre egy-egy szűrő objektumot. Például ha szeretnénk egy aluláteresztő Butterworth szűrőt, akkor a Butterworthból származó lowpass típusú objektumot hozzuk létre.

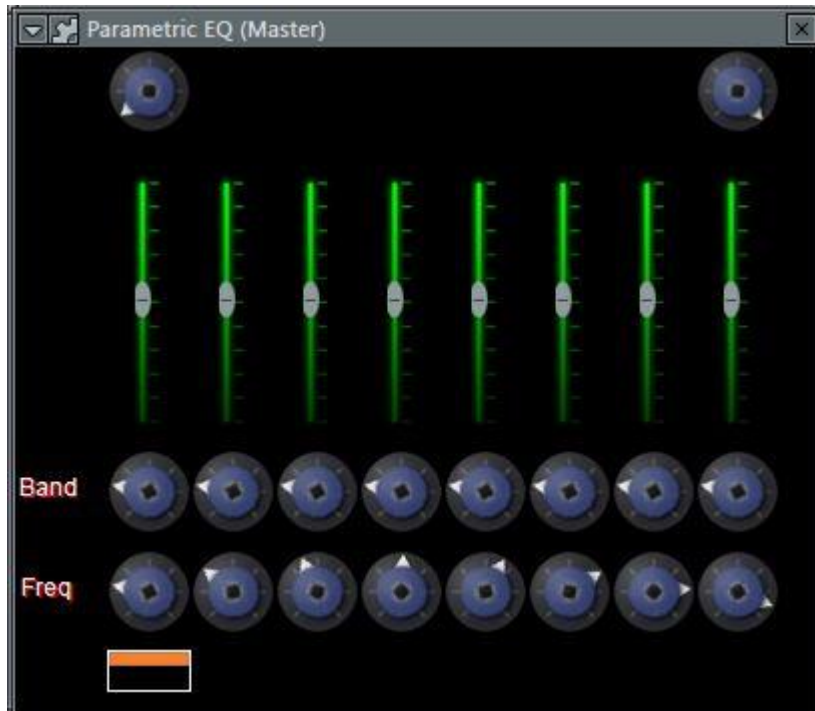
3.4.1 A szűrők függvényei

Ha létrehoztunk egy objektumot, azt a függvényeivel tudjuk használni, amik közül az EQ megvalósítása során kettő fontos van.

A setup függvény állítja be a szűrés paramétereit, ami alapján ki lesz számolva az átviteli karakterisztika. Itt szükség van megadni a szűrő fokszámát, mintavételi frekvenciát, vágási frekvenciát, és sáváteresztő szűrő esetén a sáv szélességet is. Ez a függvény először akkor kap értéket, mikor betöltjük a plugint, ezzel az általunk kívánt kezdeti állapotot beállítja az alkalmazás. Majd meghívódik minden egyes alkalommal, amikor a szűrőnek valamelyik paraméterét állítottuk.

A process függvény argumentumai a feldolgozni kívánt hangminták száma, és egy tömb, ami tartalmazza ezeket a hangmintákat. Ezek az értékek közvetlenül elérhetőek a processzási folyamatot leíró programrészben. Ez a függvény folyamatosan kapja az értékeket a host felől, amiket a setup függvény segítségével meghatározott módon feldolgoz.

4. A plugin használata



2. ábra: A kezelőfelület

A 2. ábrán látható a plugin alapállapota, amint egy példányt megnyitunk a DAW alkalmazással belőle. A zöld színű sávokon haladó csúszkák állítják az erősítés/csillapítás mértékét. Ezek mind egy-egy bandpass szűrőt paramétereznek. Felfelé tolva a maximumig az erősítés mértékét állítjuk egészen +25 dB-ig, lefelé ugyanez -25 dB értékig. A csúszka oszlopában közvetlen alatta található az adott szűrő sévszélességét állító paraméter, ami a szűrő jósági tényezőjének (Q) felel meg. Legalul helyezkednek el a frekvencia paramétert állító potméterek, ezekkel lehet a vágás/kiemelés helyét beállítani a spektrumban. A 8 db szűrő frekvenciája inicializáláskor a spektrumban egyenletesen elosztva kerül beállításra, így a jobbra elhelyezkedő paraméterekkel egyből a magasabb frekvenciákat állíthatjuk, a balra elhelyezkedőkkel pedig a mélyeket – ezt persze manuálisan akár meg is cserélhetjük miközben paramétereket állítunk.

Látható még két paraméter a jobb és bal felső sarokban. Ezek a hallható spektrum két széléről induló alul és felüláteresztő szűrők frekvenciáit állítják. Az emberi intuíciónak megfelelően a legkisebb frekvenciától induló felüláteresztő a baloldali, a nagy frekvenciáktól induló a jobboldali lett.

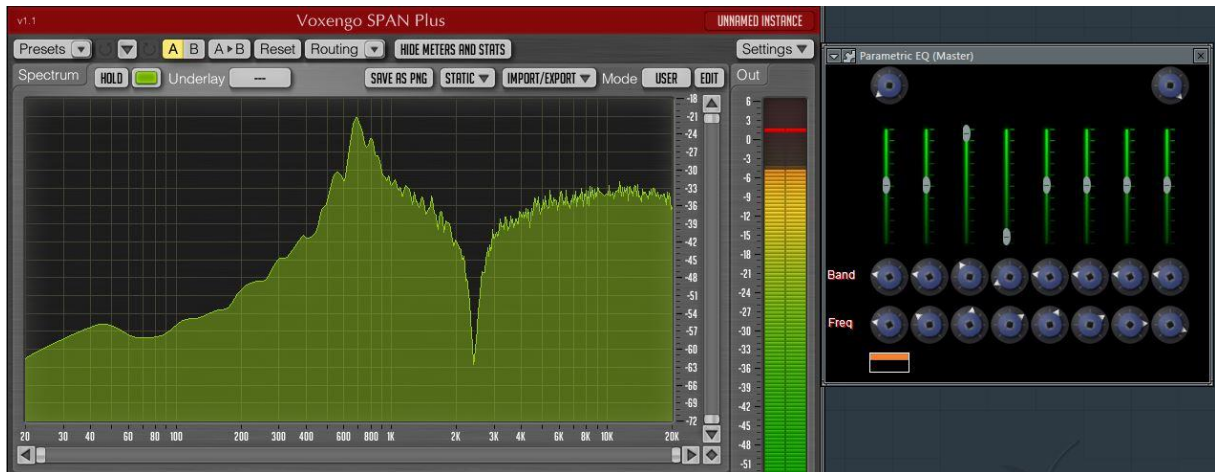
4.1 A plugin működés közben

A működés tesztelésére két módszert használtam. Az egyik alkalmával hangmintákon használtam a plugint, és figyeltem a paraméterek állításának hatását, amit nem csak hallgattam, hanem spektrum analízátoron vizuálisan is megfigyeltem. A második módszer egy kifejezetten

VST effektek számára készült analízáló program, ami impulzusok segítségével meghatározza egy effekt aktuális bode diagramját.

4.1.1 Tesztelés hangokkal

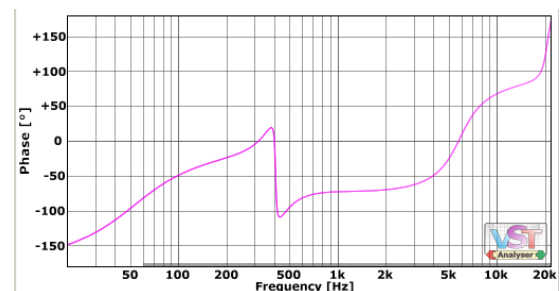
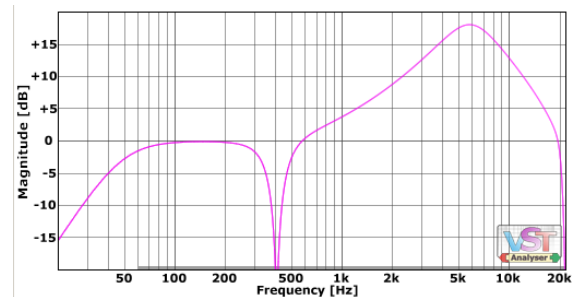
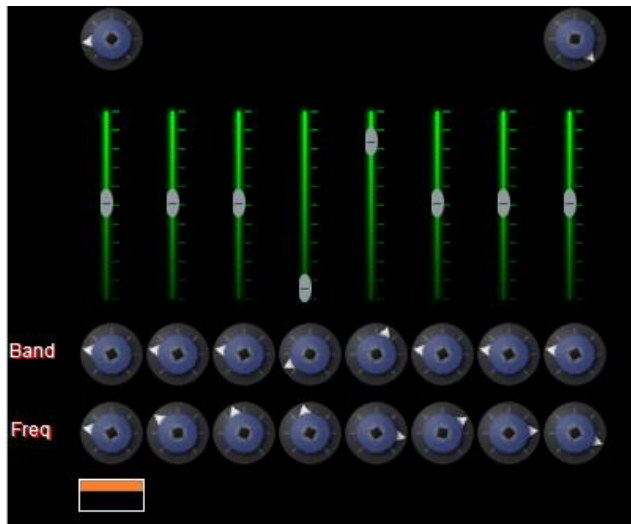
Ez esetben a működést hangminták mellett praktikusán fehérzajjal is vizsgáltam, így a spektrum analízátort figyelve könnyen megállapítható, ha egyenetlenséget eredményez a zaj spektrumában a szűrés.



4.1 ábra: Szűrés egy lehetséges beállítása és a hatása fehérzajra

4.1.2 Tesztelés VST plugin analyserrel [9]

A VST Plugin Analyser egy hostként működő alkalmazás, amibe betöltve egy VST effektet, annak Bode diagramja mérhető, és figyelhető, hogy milyen paraméter változások milyen hatással lesznek a karakterisztikára.



4.2 ábra: Egy lehetséges szűrő beállítás és a hozzá tartozó Bode diagram

5. Összegzés

A félév során bővült az ismeretem az IIR szűrőkről, itt újdonságnak számított a Band shelf szűrő, amit általában nem említettek tananyagokban az alap szűrőtípusoknál, viszont a hangszínszabályozásban nagy szerepet játszott. Felidéztem a programozással kapcsolatos ismereteket, ami kapcsán megismertem a VST fejlesztés, és a benne rejlő lehetőségek egy részét. Az elkészült effekt legutolsó verziója működése során nem hozott nem várt, hibás eredményt. A processzort nem használta aránytalan mértékben, az effekt hangszínszabályozásra való használatát kivitelezhetőnek tartom.

6. Hivatkozás jegyzék

- [1] http://www.homerecordingconnection.com/news.php?action=view_story&id=144
- [2] <http://www.musicdsp.org/files/Audio-EQ-Cookbook.txt>
- [3] [https://en.wikipedia.org/wiki/Equalization_\(audio\)#/media/File:Peaking-eq.svg](https://en.wikipedia.org/wiki/Equalization_(audio)#/media/File:Peaking-eq.svg)
- [4] https://upload.wikimedia.org/wikipedia/commons/thumb/6/60/Butterworth_response.svg/350px-Butterworth_response.svg.png
- [5] http://www.swarthmore.edu/NatSci/echeeve1/Ref/FilterBkgrnd/highpass1_2.gif
- [6] http://www.steinberg.net/en/company/press/archive/2008/steinberg_vst3_sdk.html
- [7] <https://github.com/vinniefalco/DSPFilters>
- [8] https://en.wikipedia.org/wiki/Butterworth_filter
- [9] <http://www.pcv.de/applications/measurement-programs/>