

デジタルチャンネルディバイダ channeldividerF Ver0.43 及び Ver050 について

1. 概要

FIR(直線位相)型の2Wayのチャンネルディバイダとして機能する、音楽再生ソフト foobar2000用のDSPプラグインです。

foobar2000については、以下の公式サイトをご参照下さい。

<http://www.foobar2000.org/>

foobar2000の日本語の解説については、こちらをご参照ください。

<http://foobar.s53.xrea.com/fbwiki/>

2. foobar2000のバージョンとプラグインのバージョンについて

foobar2000はVer083からVer090に変わって、構造がかなり変化しました。Ver083はWindows 95、98、ME、NT、2000、XPで動作しますが、Ver09はWindows 95、98、ME、NTでは動作しません。また、プラグインもVer08用のものとVer09用のものでは互換性が無くなりました。foobar2000とchanneldividerFのバージョンの対応は以下の通りです。

foobar2000 Ver083	channeldividerF Ver044
foobar2000 Ver09	channeldividerF Ver050

なお、公式サイトでは、既に、旧バージョンであるVer083は既にダウンロードできなくなっているため、Ver083を使いたい方は、以下のミラーサイトなどからダウンロードしてください。

<http://www.clubic.com/telecharger-fiche11022-foobar2000.html>

3. 配布ファイル

- ・foo_dsp_channeldividerF.dll プラグインです。
- ・FirGain2.exe Tap数と周波数特性の関係を示すプログラムです。
- ・ReadMe2Way.pdf 本文書です。

4. 仕様

現状の仕様は以下の通りです。

- ・フィルタの種類：カイザー窓を用いたFIR(直線位相)型
- ・遮断特性：Tap数で指定(Tap数と遮断特性の関係については別プログラム(FirGain2.exe)参照)
- ・阻止帯域減衰量(Stop Band Attenuation): -144dB
- ・内部演算精度：64bit(倍精度浮動小数点)
- ・出力ch：1ch 低域左、2ch 低域右、3ch 高域左、4ch 高域右

- ・出力 bit 数 : Output プラグインで指定
- ・対象音源 : 音源が 2ch (ステレオ) の時のみチャンネルディバイダとして動作。その他の ch 数の場合は音声を加工せずそのまま出力

5 . 動作環境

- ・foobar2000 が動作する環境
- ・CPU : Pentium 1GHz 以上推奨
- ・サウンドカード : 4ch 出力をサポートしているもの。24bit 出力推奨

なお、Ver044 を動かすためには msxcr70.dll が別途必要です。msxcr70.dll は <http://www.vector.co.jp/vpack/filearea/win/util/runtime/index.html> の [VC++ .NET ランタイムインストーラー 1.0](#) などで手に入れることができます。

6 . 使い方

(1) インストールなど

(1) インストールなど

(a) Ver044 の場合

foobar2000 がインストールされているフォルダの中の components フォルダに foo_dsp_channeldividerF.dll を入れてください。

もし、システムに msxcr70.dll がインストールされていない場合は windows フォルダの system32 ディレクトリか、foobar2000.exe と同じフォルダに msxcr70.dll を入れてください。

メニューの foobar2000 から preferences を選び DSP Manager で channeldividerF を有効にします。

DSP Manager の下の channeldividerF を選択し、希望のクロス周波数などを入力します。設定は、一旦曲を終了させ、曲の再生が開始されるときに変更されます。

(b) Ver050 の場合

foobar2000 がインストールされているフォルダの中の components フォルダに foo_dsp_channeldividerF.dll を入れてください。

メニューの foobar2000 から preferences を選び DSP manager で channeldividerF を有効 (Active DSPs) にします。

Configure selected を押し、セッティング画面を出し、希望のクロス周波数などを入力し OK ボタンを押します。OK を押した時点で設定は変更されます。

(2) 周波数の指定について

クロスオーバー周波数はローパスとハイパスについて同一の周波数のみ指定可能です。

(3) 遮断特性

Tap 数で間接的に指定します。Tap 数は必ず奇数にしてください。(一応計算時には奇数に修正するようにはしていますが。)クロスオーバー周波数が低域になるほど、同一の遮断特性を得るためには大きな Tap 数が必要です。

(4) 遅延

遅延はサンプリング周波数 44100Hz のときのサンプル数 (Tap) 単位で与えてください。サンプル数を入力すると距離を計算し表示します。なお、クロス周波数、遮断特性、遅延の設定については、次に再生ボタンを押し再生が開始された時点で変更されます。

(5) レベル調整

0.5dB 単位でアッテネートできます。Ver044 は曲を再生中にレベルを変動させると瞬時に音量も変更されますが、Ver050 では、OK ボタンを押さないと音量は変化しません。音量は単純な掛け算で音量を絞っていますので、あまり、減衰させることは bit 落ちを招くので好ましくありません。ただし、フィルタ演算の結果が 1 を超えた場合の悪影響を防ぐために高域、

低域とも - 3 dB 以下にすることをお勧めします。(基準とするほうを - 3dB とし他をそれ以上に減衰させる。) 詳細は、10 . 測定(7)を参照してください。なお、全体の音量調整は、bit 落ちのことを考えれば、DA コンバータの後のアナログボリュームで調整したほうが良いです。多チャンネルのアナログボリュームになかなか安価で良いのがないのが悩ましいところですが。

7 . いいわけ

作者はデジタル処理に関してもプログラム作成に関してもド素人です。FIR フィルタを为了能にデジタル処理の入門書を読んで勉強し(見よう見まねで中身を良く理解せずに) PASCAL (delphi) でチャンドバを作って遊んでいました。これは waveFile 限定、DirectSound 限定だったので、CD を直接再生し ASIO で出力したいために(これを 1 から作るのには私には不可能です) foobar のプラグインの作り方及び C++ の勉強を始めました。というわけで、hello world の類以外では作者が C++ を使用して作ったものとしては最初のソフトです。(まだ C++ をよく理解していません。) ということで、ひょっとしたらとんでもないミスをしている可能性があります。また、ここに書いていることも間違いがたくさんあると思います。その場合は、なにとぞご容赦ください。

8 . 使用上の注意及び免責

本プラグインは作成途上であり、思わぬところで雑音が出る可能性があります。従って、ツイータ保護のために、ツイータとアンプを直結せず、必ず保護用のコンデンサを直列につないでください。

また、Windows の警告音を出さないようにするため、コントロールパネルの「サウンドとオーディオデバイス」を選択し「サウンド」タグのサウンド設定から「サウンドなし」を選択するか、Windows の警告音や他のアプリケーションの音は別のサウンドカードから出力させることをお勧めします。(これをやらないと警告音が大音量で出てビックリすることがあります)

なお、本プログラムを使用することにより生じたいかなる損害についても、作者は一切の責任を負わないものとさせていただきます。

9 . 仕様などの詳細

(1) 遮断特性

遮断特性指定方法など

このプラグインで用いているフィルタ係数は教科書通りのカイザー窓を用いたものです。カイザー窓で FIR フィルタを設計する場合、通常、通過帯域 (pass band)、阻止帯域 (stop band)、阻止帯域減衰量 (stop band attenuation) を指定すると、指定性能を満足する必要 Tap 数 (フィルタ長) が計算され、計算された Tap 数分のフィルタ係数が計算されます。チャンドバの場合、CD のダイナミックレンジや AD コンバータのダイナミックレンジ、サウンドカードの出力などを考えると阻止帯域減衰量は - 144dB あれば十分なので、これで固定とすると、あとは、通過帯域と阻止帯域を指定すればよいこととなりますが、二つの数字を指定するのは面倒? (プログラムのにも) だったので、Tap 数を指定することにより間接的に両者を指定することにしました。

遮断特性と遅延の指定はサンプリング周波数 44100Hz の場合の Tap 数を指定することにより行いますが、他のサンプリング周波数の音源の場合は、44100Hz とほぼ同様の減衰率、遅延となるよう内部で簡易的なやりかたでフィルタの再構成を行います。具体的には、指定した Tap 数を内部でサンプリング周波数に比例して増加させています。従って画面で 4001Tap の Tap 数を指定した場合でも、88200Hz の音源を再生する場合は 8003Tap の計算を行うこととなります。現在個人で手に入れられるソースはサンプリング周波数 44100Hz

のものに限られていると思うので、ここまでやる必要があるのかどうか疑問でしたが、異なるサンプリング周波数のものを間違っ再生した場合のツイータの保護などを考慮しこのような構成にしました。

FirGain2.exe について

Tap 数から遮断特性を想像するのは結構難しいので Tap 数と周波数を指定すると周波数特性を描画するソフト FirGain2.exe を別に作成しました。(前に delphi で作ったチャンデバの表示部分をそのまま使いました。手抜きソフトですので描画にやたら時間がかかることがあります)

プログラムを立ち上げると、以下のような画面(一部)がでますので、カットオフのところに希望のクロスオーバー周波数を入れ Tap 数を入力し、フィルタ特性計算ボタンを押すと、指定したクロスオーバー周波数と Tap 数の周波数特性が出ます。カイザー窓推薦 Tap 数(参考)の欄は、プラグインの周波数特性推測には関係ありません。(手抜きプログラムですみません。)



(2) 演算速度

FIR フィルタの畳込みには FFT を利用していますので、現在発売されている PC であれば、2Way のチャンデバとして必要な演算速度は確保できると思います。演算量は Tap 数 + デレイの Tap 数が 2 の累乗を超えた時(例えば 1023 から 1025 へ)段階的に増加します。

チャンデバとしては 8000Tap 前後で動けば必要十分だと思いますが、当方で確認したところでは Celelon500MHz のノートパソコンでハードディスク上の wave ファイルを再生した場合で 1万6千 Tap でも音飛びしませんでした。athlon2600+ のマシンでハードディスク上の wave ファイルを再生した場合は 26 万 Tap (無意味ですけど)まで音飛びはしませんでした。(26 万 Tap の場合遅延はすごく大きくなります。再生ボタンを押してから音が出るまで約 16 秒かかりました。遅延を少なくするためには、多分、分割畳込みをすれば良いんでしょうが、チャンデバ用途としてはここまで大きな Tap 数は不必要なのでそのままにしてあります。)(いずれも Ver032 の場合)

もし、音が飛ぶようであれば、サウンドカードのレイテンシを大きくして、Output プラグインのバッファのサイズを大きくし、playback で Full file buffering に大きい値を入れれば改善できるかもしれません。また、Visualization をオフにすると改善できる可能性があります。

(3) 遅延 (Time Alignment)

遅延については、サンプリング周波数単位の簡易的な方法を用いています。指定は Tap 数単位で行います。(1Tap の遅延は 343.5m/s (音速) \div 44100 (サンプリング周波数) = 7.8mm) プログラム的にはフィルタ係数の前にゼロを入れることによって実現しています。従って、計算上の Tap 数は指定した Tap 数 + 遅延 Tap 数となります。遅延の指定はサンプリング周波数 44100Hz の時の Tap 数の距離を前提に行いますが、異なるサンプリング周波数のものを再生する場合は、遅延についてもサンプリング周波数に比例して変更していますので、表示される距離との差は、1 サンプルあたりの距離の差程度にはなっていると思います。

(4) 出力 bit 数

本プラグインは内部演算を倍精度浮動小数点(64bit)で計算し output プラグインあるいは次の DSP プラグインにデータを引き継いでいます。出力 bit 数は output プラグインで決めることとなりますが、可能であれば 24bit 以上を推薦します。計算結果にはディザなどは加えずそのまま output プラグインに渡しています。大多数の音楽ソースは 16bit ですが、16bit 信号を演算し 16bit に丸めて出力した時にディザを加えていない場合は、フィルタ計算の結果に原信号と相関関係のある雑音がのるため、微少音がザラつくと言われていました。24bit 出力にすれば、誤差が十分小さくなること、A/D コンバータやアナログ回路の SN 比に隠れることなどの理由でこの問題は小さくなるとも言われています。また、後述するように出力は -3dB 程度落とした方がよいので、これによる bit 落ちを防ぐためにも出力は 24bit 以上とすることを勧めます。

10. 測定

素人が個人で測定できることには限りがありますが、とりあえず測定した結果を示します。サウンドカードの Prodigy192 には、Direct Wire という機能があり、任意の ch の出力をデジタルデータのまま入力に戻すことができます。これを利用し測定をしてみました。測定には efu 氏作成の WaveGene、WaveSpectra、Sigeboh 氏作成の WaveAnalyzer32 を使用させていただきました。ご両人に感謝いたします。

(1) 遮断特性

クロスオーバー周波数を 1000Hz、Tap 数を 501Tap に設定し、ホワイトノイズを再生し、出力を WaveSpectra に入力して FFT にかけたものが、図 1、図 2 です。それなりの遮断特性になっていると思います。

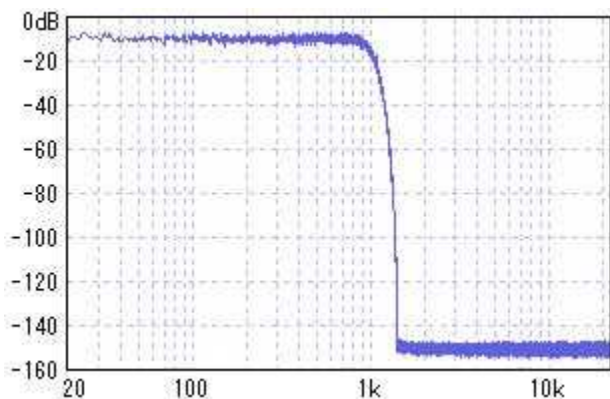


図 1 低域

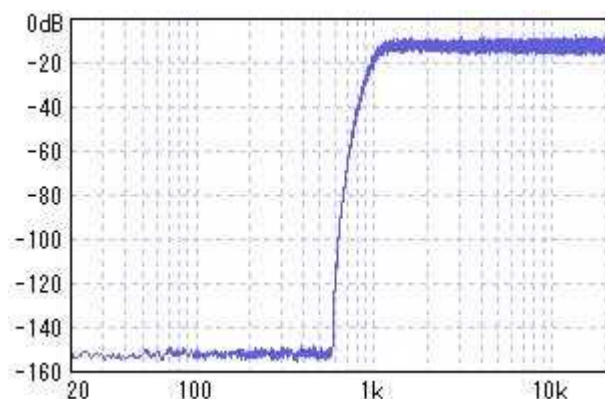


図 2 高域

図 3、図 4 は、クロスオーバー周波数を 100Hz、Tap 数を 8001 に設定し、スイープ音を再生し WaveSpectra でピークホールド表示させたものです。FIR フィルタの場合クロス周波数が低いほど大きな Tap 数が必要になります。(同等の遮断特性を得ようとするくとクロス周波数に反比例した Tap 数が必要です。)

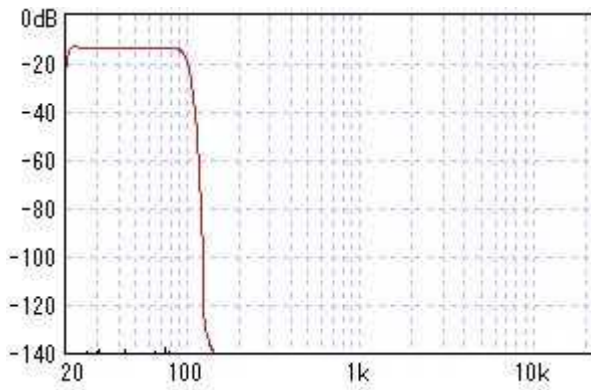


図3 低域

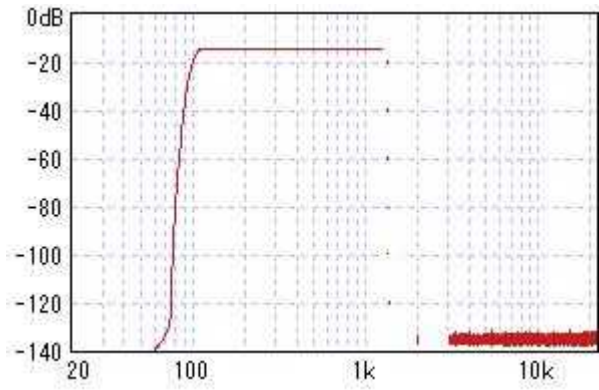


図4 高域

(2) 合成周波数特性

クロスオーバー周波数を 1000Hz、Tap 数を 501Tap に設定し、ホワイトノイズを再生し、WaveAnalyzer32 の左チャンネルに低域を、右チャンネルに高域を入力し、両者を合成した周波数特性のグラフが図5です。合成の周波数特性はフラットになっています。

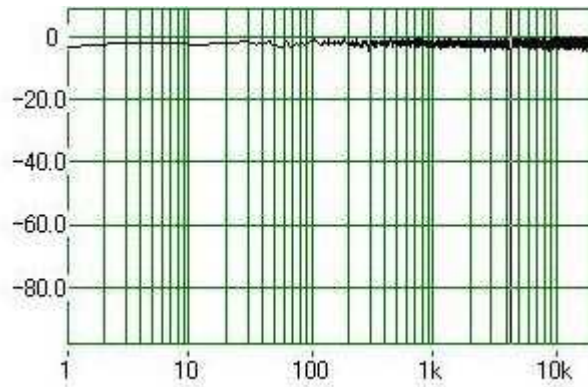


図5 合成周波数特性

(3) 合成波形

クロスオーバー周波数を 1000Hz、Tap 数を 501Tap に設定し、1000Hz の単発サイン波を再生し、WaveAnalyzer32 の左チャンネルに低域を、右チャンネルに高域を入力し、両者を合成した波形を表示したものが図6です。ChanneldividerF は直線位相型 FIR フィルタなので、単発サイン波が再合成されています。もっとも、これはデジタルデータでの合成なので、空气中で異なるスピーカから再生される音波がどのように合成されるかという議論のあるところでしょう。

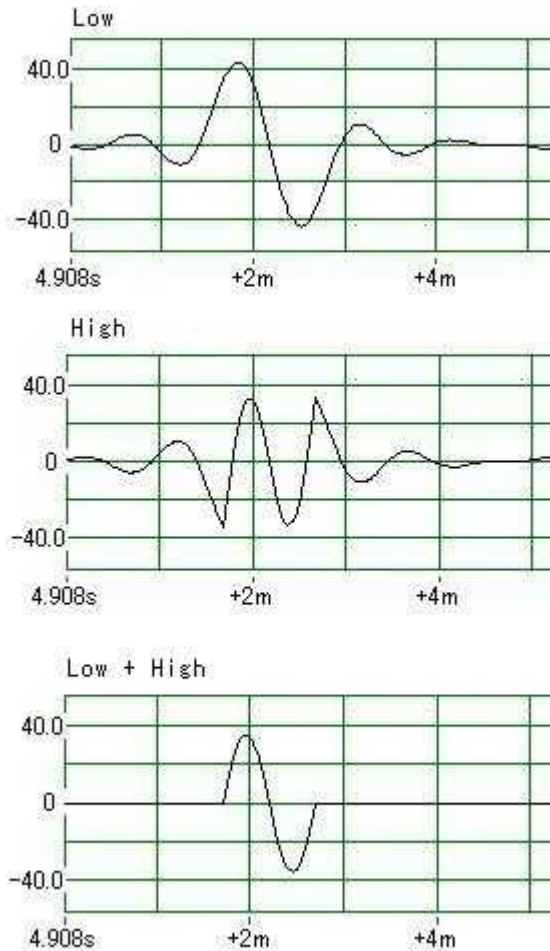


図6 単発サイン波の合成

(4) 1000Hz sin 波 純音の再生

クロスオーバー周波数を 1000Hz、Tap 数を 501Tap に設定し、1000Hz の sin 波を再生し WaveSpectra に入力し、周波数特性を調べたのが図7、図8です。変な歪は出ていないようです。

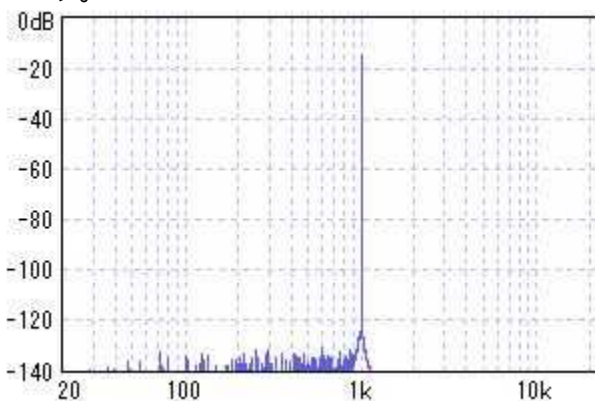


図7 低域

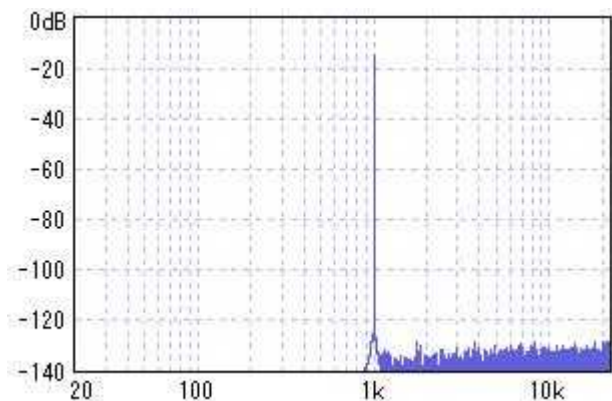


図8 高域

(5) 遅延

クロスオーバー周波数を 1000Hz、Tap 数を 501Tap に設定し、1000Hz (サンプル周波数 44100Hz) の sin 波を再生し、WaveSpectra に低域を左、高域を右に入力し、リサージュを描かしてみました。図9が遅延無し、図10が高域を半波長：180度 (22Tap) 遅延させたもの、図11が高域を1/4波長：90度 (11Tap) 遅延させたものです。理論通りのリサージュ

ユが描けていると思います。

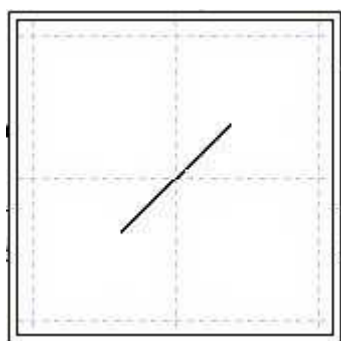


図9 遅延無し

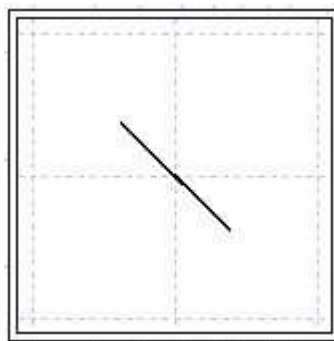


図10 高域半波長遅延

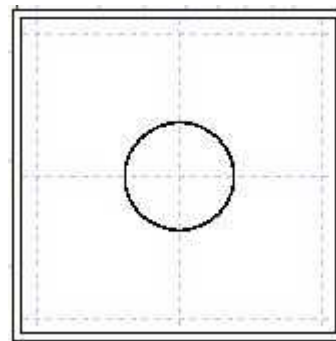


図11 高域(1/4 波長)遅延)

(6) 遅延をさせた場合の合成周波数特性

図12は、クロスオーバー周波数を1000Hz、Tap数を501Tapに設定し、高域を半波長22Tap遅延させ、(2)と同様ホワイトノイズを再生し低域と高域を合成したものです。半波長(180度)の位相遅れがあるのでクロス周波数で鋭いディップが起こっています。世の中、遮断特性が急峻であれば位相についてはさほど考慮しなくとも周波数特性は影響を受けないという俗説がありますが、そうならない場合もあるというが示されています。

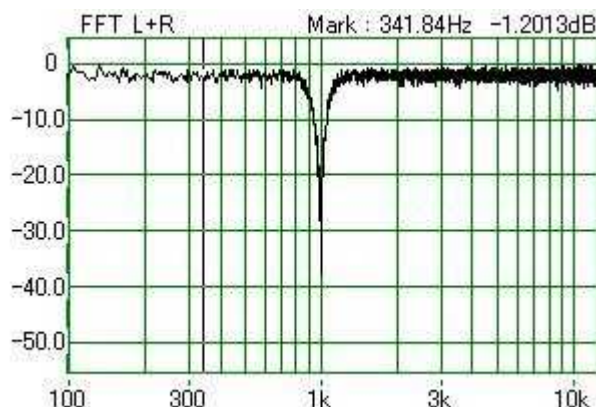


図12 高域半波長遅延

(7) 演算結果が1を超えた場合の影響

DSP プラグインにはデータは1に正規化された(最大値が1の)倍精度(foobar2000Ver091の場合は単精度)浮動小数点で渡されます。渡されたデータは DSP プラグインで計算され Output プラグインに渡され、ここで出力 Bit 数に応じて適切な値に変換されハードウェアにデータが渡されます。(らしいです。)

ところで、デジタルチャンデバの場合、最大ビットの音が連続した場合(コンプレッサーを多用する J P O P のような音楽の場合)演算結果が1(最大値)を超える場合があります。

通常、最大 bit 数を超えた値がサウンドカードに送られるとバチッという反転ノイズがでますが、output プラグインでは指定された最大 bit 数より大きい数値があった場合それを最大 bit 数にあわせる、いわゆるリミッターが働くようになっています。(らしいです。)

リミッターが働くことは働かないで反転ノイズが出る(ツイータが壊れる危険性有り)よりはるかに良い状況ですが、チャンデバとしては遮断特性に影響が出ます。

リミッターがかかるということは、リミッターに係る直前のデータとリミッターがかかったデータとの間でデータとして不連続な点が現れるということで高周波が発生したことと同じですし、リミッターがかかった値が連続した場合、データは直線状態となり、直流が現れたと同じこととなります。

フィルタ処理が終わった後の output プラグイン処理でこのようなことが起こるということはハイパスフィルタで直流域が出る可能性があり、ローパスフィルタで高周波が出るということなのです。

実際、最大値が 0 dB であるホワイトノイズを波形生成ソフト (Wavegen) で作成したものを、1 KHz クロスで再生したところ、ハイパスフィルタの周波数特性は図 13 のとおりで、リミッターが作動し低周波が発生したせいか？ 阻止帯域減衰量が非常に少なく (-40dB) なっています。一方、最大値が -6 dB であるホワイトノイズを再生したところ、このような現象は起こらず、阻止帯域減衰量は期待通りとなっています。この理由は上記のとおりであると考えられます。

そこで、これを避けるために最大値 0 dB のホワイトのイズを再生し、ディバイダ内部のボリュームで各音域を -3 dB に下げて、同じファイルを再生したところ周波数特性は、図 14 のとおり問題ないものとなりました。

以上よりチャンデバのボリューム設定の初期値は -3dB 程度絞ったものにするほうが良いのではないかと思います。

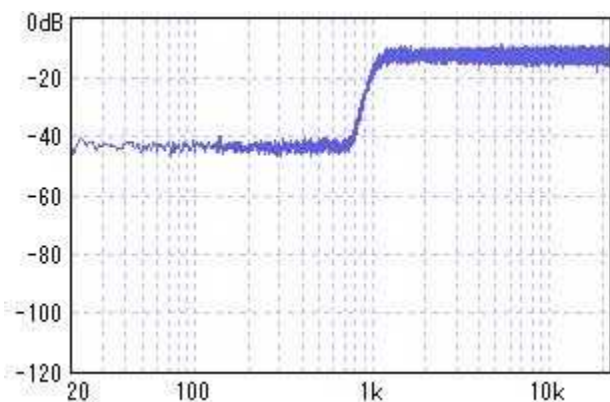


図 13 ゲイン調整無し

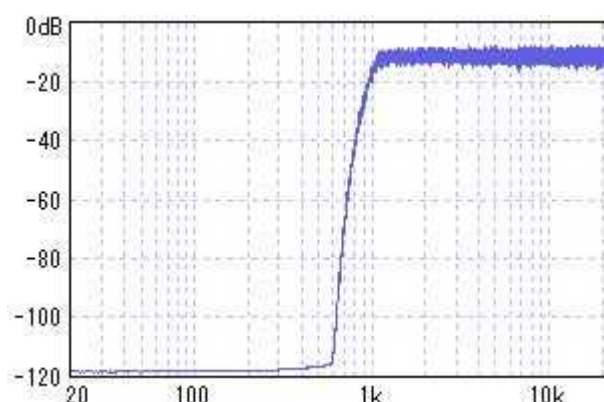


図 14 内部で -3dB ダウン

1.1 . 著作権など

本プログラムはフリーソフトです。非営利の場合は (商業用に使う人がいるとは思えませんが) 自由に使って頂いてかまいません。再配布もご自由にどうぞ。ただし、再配布する場合はプラグインと FirGain2.exe と本文書を同時に配布してください。

著作権は minn にあります。(日本の法律では著作権の放棄はできないらしいので)

1.2 . 謝辞

本プログラムを作成するに当たり、いろいろご指導を頂いた a 氏、s 氏、n 氏、i 氏に感謝いたします。また、FFT については、大浦拓哉氏作成のソフトを使わせていただきました。すばらし

いソフトを公開されている大浦氏に感謝いたします。

General Purpose FFT(Fast Fourier Transform) Package
Copyright (C) 1996-2001 by Takuya OOURA
<http://momonga.t.u-tokyo.ac.jp/~ooura/index-j.html>

1 3 . バージョン履歴

- 050 2006/05/29 foobar2000Ver09 対応
- 044 2006/05/24 関数名 (flush) のミスを修正
- 043 2005/04/24 042 で生じた長大なタップが指定できないバグを修正。ベッセル関数の精度をほんの少し上げた。
- 0421 フィルタ係数を代入する動的配列の確保の方法を変更。DSP マネージャーを開くときの安定性が増したはず。
- 042 44100Hz 以外のサンプリング周波数のものを再生した場合にフィルタを再構成することとした。長大な Tap を指定した場合に DSP マネージャーを開くときに時間がかかることを修正した。
- 041 バッファクリア関数を定義、ソースコードを整理した。
- 040 データの引継ぎ方法を改善し FFT の回数を半減した。
- 032 終了時に最後のデータが破棄されるのを修正した。
- 031 ogg ファイルなど可変の chunk->sampleNo の音源を再生した場合に音がとぎれていたので修正した。
- 030 デイレイを追加した。
- 020 ゲイン調整を追加した。
- 010 クロス周波数、Tap 数の指定を追加した。
- 001 最初のバージョン