

デジタルチャンネルディバイダ channeldivierF2B Ver064 について

1. 概要

channeldivierF2B は、直線位相 FIR 型の 2Way 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。channeldivierF2B は、channeldivierF にスピーカのバッフルステップ補正用として直線位相 FIR 型の疑似シェルビングフィルタを加えたものです。

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

channeldivier2FB Ver064 は、foobar2000 Ver1.00 以降で使用できます。2010/08/29 現在、Ver1.0、Ver1.03、Ver1.1 での動作を確認しています。

3. 配布ファイル

- ・foo_dsp_channeldivierF2B.dll プラグインです。
- ・FirGain2B.exe フィルタの Tap 数と周波数特性の関係および疑似シェルビングフィルタの周波数特性を示すプログラムです。
- ・ReadMe2WayBSC.pdf 本文書です。

4. 仕様

現状の仕様は以下の通りです。

- ・動作条件：入力 2ch (ステレオ) のときのみチャンネルディバイダとして動作
- ・フィルタの種類：カイザー窓を用いた直線位相 FIR 型
- ・遮断特性：フィルタ長 (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 (FirGain2B.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
- ・疑似シェルビングフィルタ：低域側 (下限 40Hz) と高域側の周波数を指定すると両者を結んで直線的に低域を 6dB まで上昇可
- ・出力 bit 数：Output プラグインで指定

5. 動作環境

- ・foobar2000Ver1.00 以上が動作する環境
- ・Windows XP、VISTA、7

- CPU : PentiumⅢ 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

6. 使い方

(1) インストールなど

① 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_channeldividerF2B.dll を入れてください。

foobar2000 Ver1.1 の場合は、上記の方法の他、次の方法でもインストールできます。File メニューより Preferences を選び、Preferences 画面の Components 画面を選択します。その画面の「Install」ボタンを押し、foo_dsp_channeldividerF2B.dll を選択し、「開く」ボタンを押すと、プラグインがインストールされます。

インストールされる場所は、c:\¥Documents and Settings¥User¥Application Data¥foobar2000¥user-components になります。

② メニューの foobar2000 から preferences を選び (図 1 参照) Available DSPs より channeldividerF2B をダブルクリックし、channeldividerF2B を Active DSPs に移し、有効にします。

③ Configure selected を押し、セッティング画面を出し (図 2 参照)、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) クロスオーバー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) シェルビングフィルタの指定

指定した 2 つの周波数を結んで、直線的に低域の周波数特性を増幅させることができます。上昇が終了する周波数 (40Hz が下限)、上昇を始める周波数、増幅させる量 (6dB が上限) を指定します。

(5) 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。

(6) レベル調整

0.5dB 単位でアッテネートできます。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 -3dB 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、9. 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ものがないのが悩ましいところですが。

(7) 各種設定の反映

クロス周波数、フィルタ長 (Tap 数)、ディレイ、シェルビングフィルタの設定、レベルは、いずれも設定画面の「OK」ボタンを押した後有効になります。

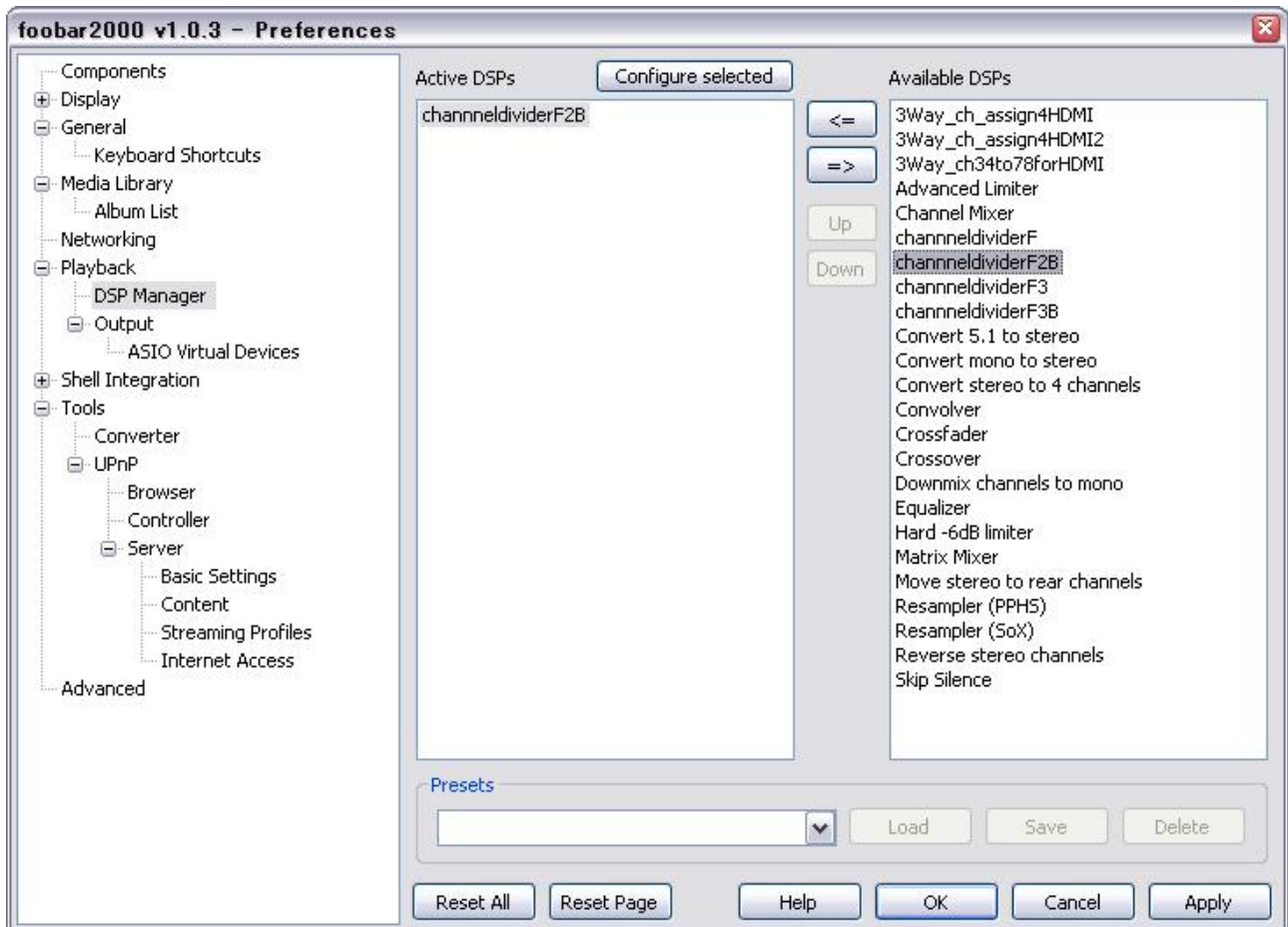


図 1 foobar2000 Preferences 画面

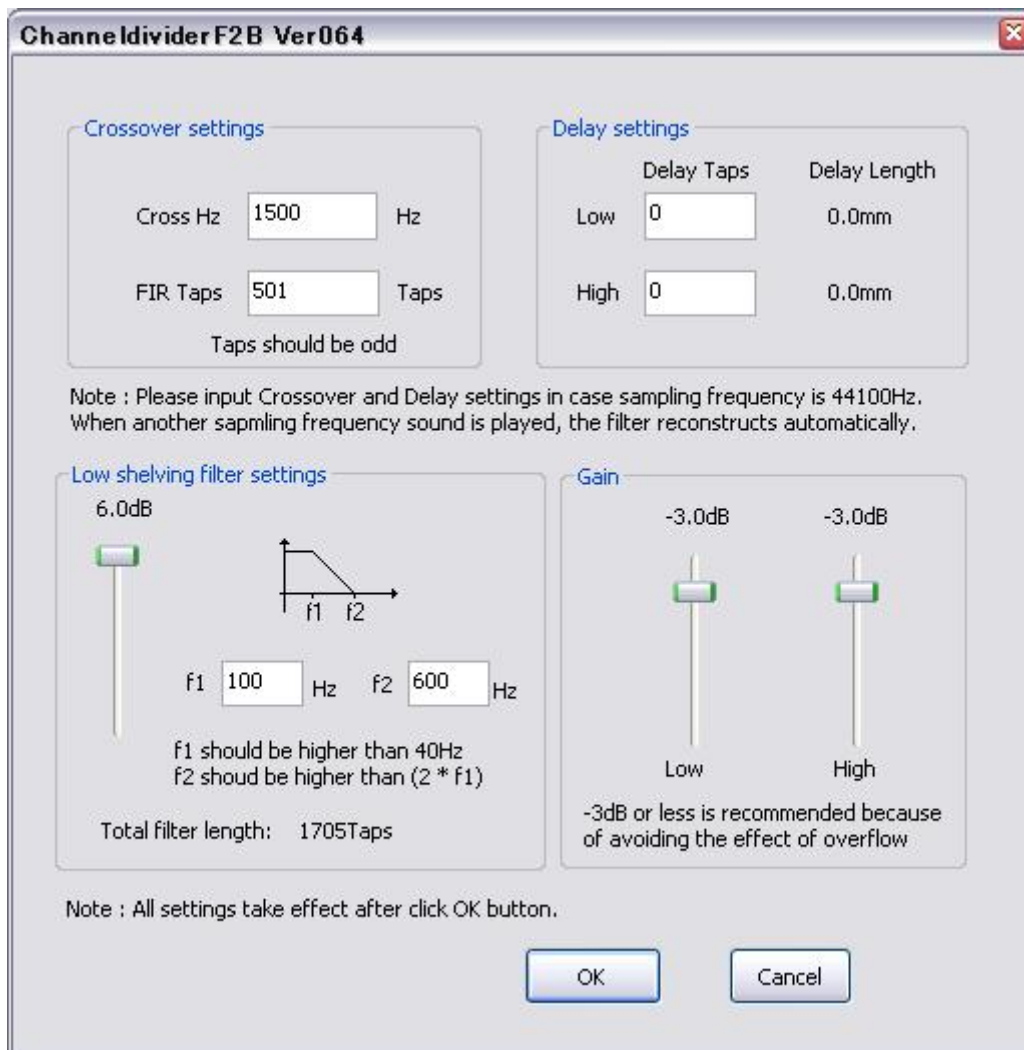


図2 channeldividerF2B Configure selected 画面

7. 使用上の注意及び免責

本プラグインは作成途中であり、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奨めします。(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)

本プラグインは、再生する音源が 2ch の時のみチャンネルディバイダとして動作します。4ch や 5.1ch の音源を再生した場合は、音に何も加工せずそのまま出力します。このためツイータに損傷を与える場合がありますのでこれらの音源を再生しないで下さい。モノラルの音源を再生した時もチャンドバとして動作させたい場合は、標準でついてくる DSP の「Convert mono to stereo」を channeldividerF2B の上に置いておけば、チャンドバとして動作します。「Convert mono to stereo」はモノラルの音源にのみ作用しますので、モノラルとステレオとを演奏する場合もこれを入れっぱなしにしておいてかまいません。

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。また、本文書にも誤りのある可能性があることについてご容赦下さい。

8. チャンネルディバイダの仕様の詳細

(1) 遮断特性

① 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。チャンネルデバの場合、CD のダイナミックレンジや DA 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えて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。

② FirGain2B.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 FirGain2B.exe を別に作成しました。

プログラムを立ち上げると、図 3 の画面 (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。)

バッフルステップ補償用の疑似シェルビングフィルタについても、シェルビングフィルタとして動作させる低域の周波数 (40Hz が下限)、高域の周波数、および、持ち上げる量を指定すると周波数特性を計算し表示します。

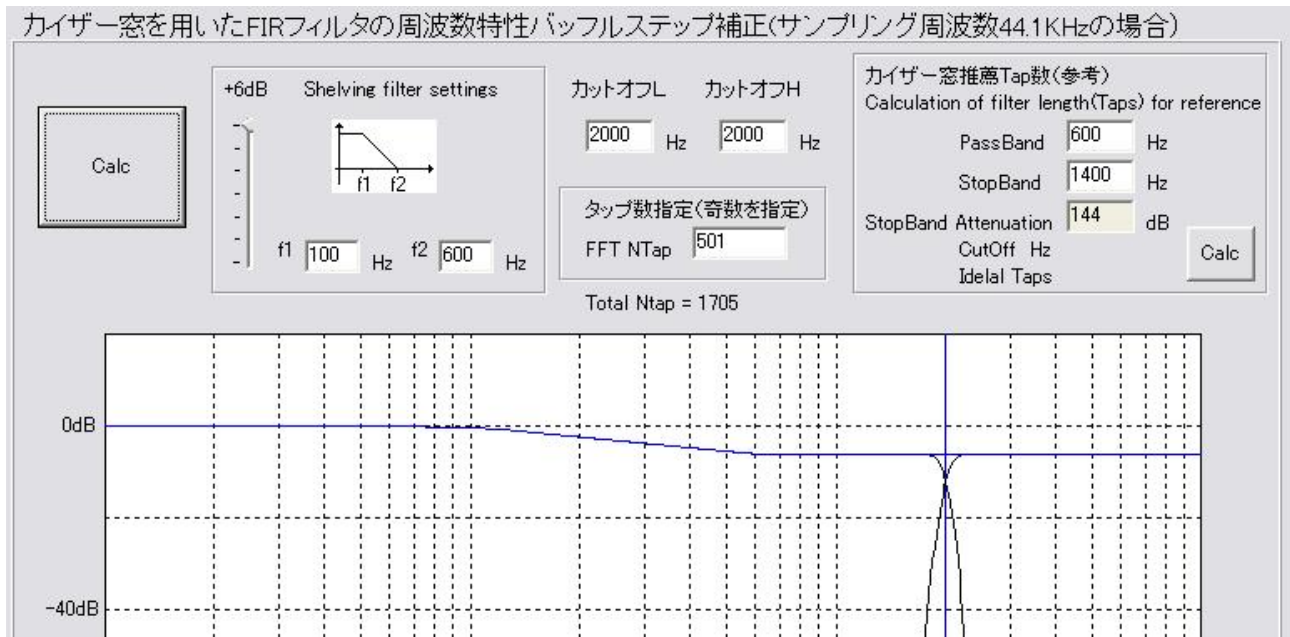


図3 FirGain2B 設定画面 (一部)

(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 のチャンド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+ディレイの 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。

チャンドバとしては 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、チャンド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 Ver044 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は倍精度浮動小数点 (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、DA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。

9. チャンネルディバイダの測定

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 4、図 5 です。それなりの遮断特性になっていると思います。

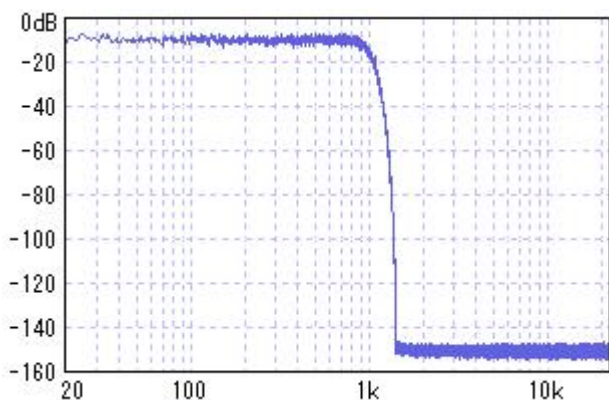


図 4 低域

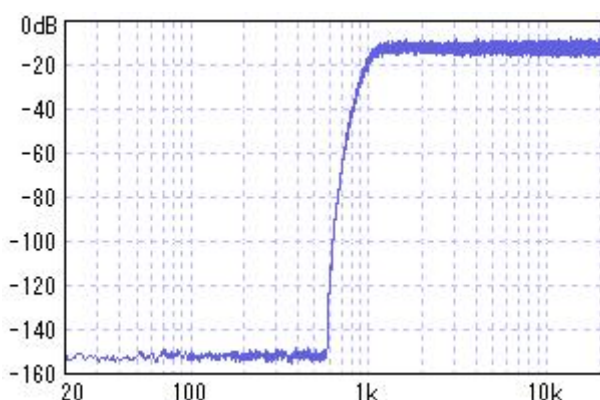


図 5 高域

図 6、図 7 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とクロス周波数に反比例した Tap 数が必要です。)

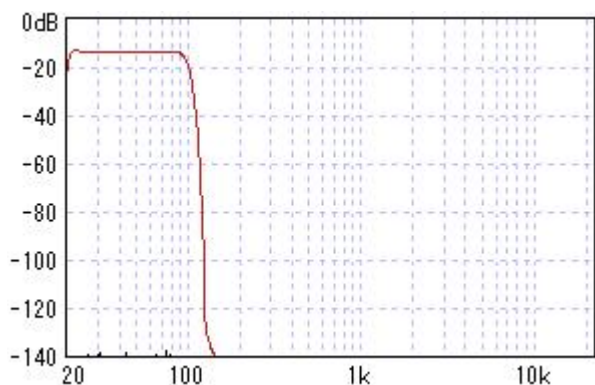


図 6 低域

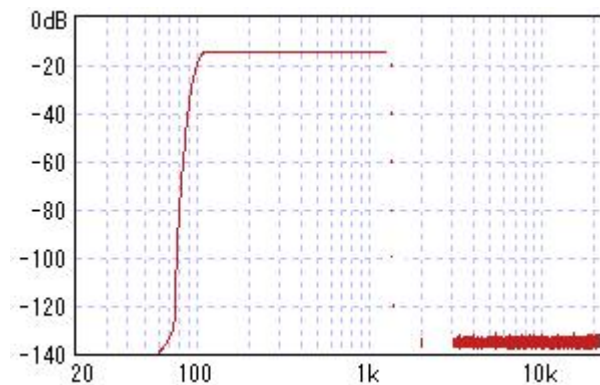


図 7 高域

(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 8 です。合成の周波数特性はフラットになっています。

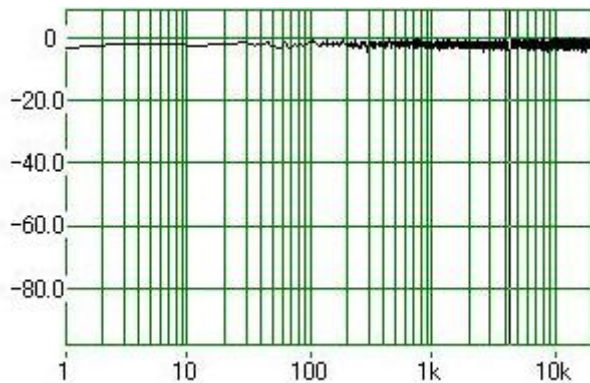


図 8 合成周波数特性

(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 9 です。ChanneldividerF2B 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、空気中で異なるスピーカから再生される音波がこ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。

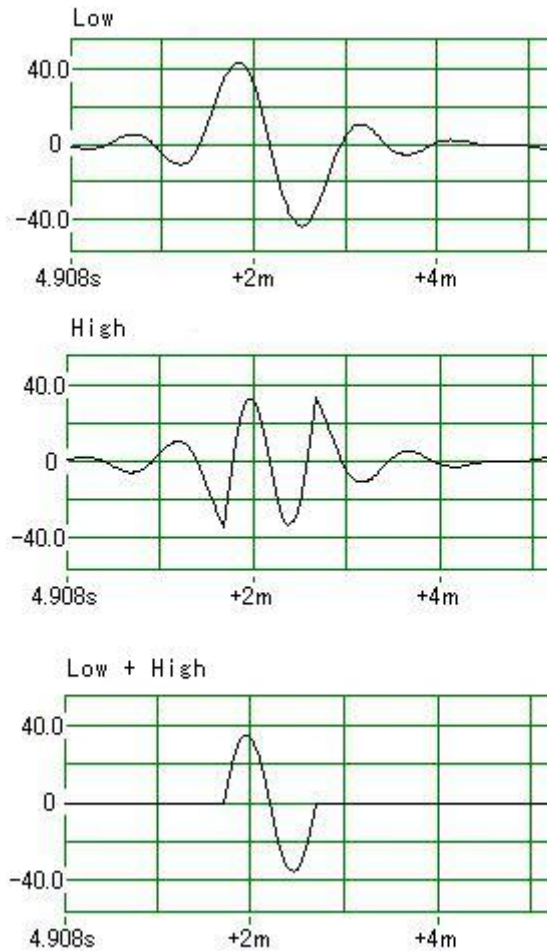


図9 単発サイン波の合成

(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 10、図 11 です。変な歪は出ていないようです。

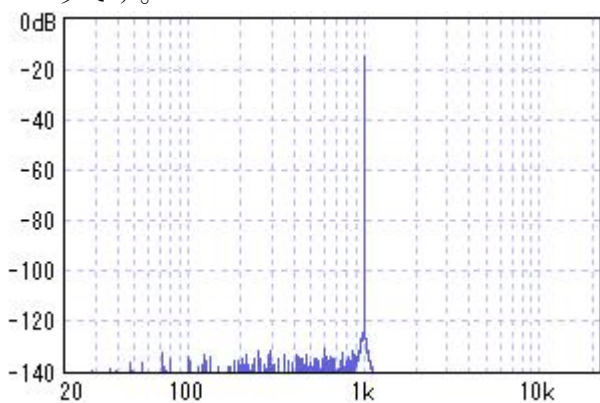


図 10 低域

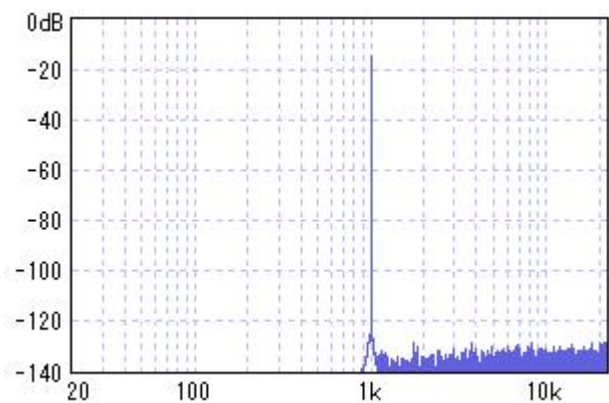


図 11 高域

(5) 遅延

クロスオーバー周波数を 1000Hz、Tap 数を 501Tap に設定し、1000Hz (サンプリング周波数 44100Hz) の sin 波を再生し、WaveSpectra に低域を左、高域を右に入力し、リサーチを描かしてみました。図 12 が遅延無し、図 13 が高域を半波長：180 度 (22Tap) 遅延させたもの、図 14 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

ジュが描けていると思います。

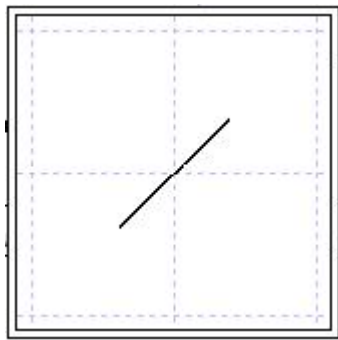


図 1 2 遅延無し

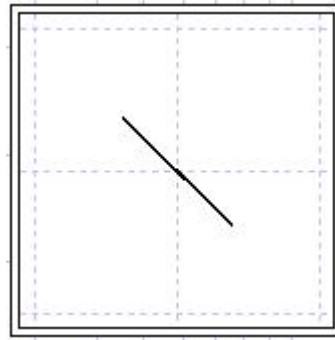


図 1 3 高域半波長遅延

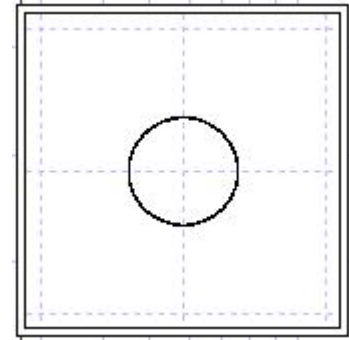


図 1 4 高域(1/4 波長)遅延

(6) 遅延をさせた場合の合成周波数特性

図 1 5 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。

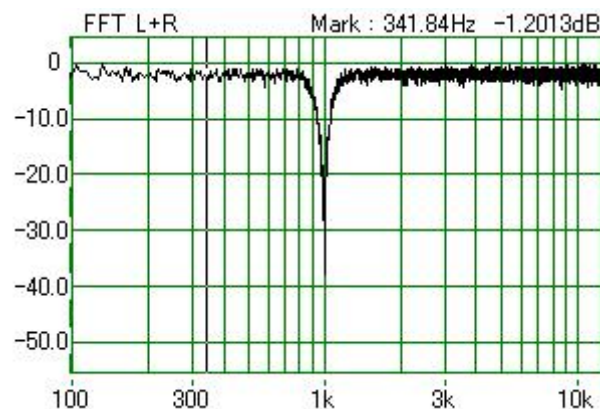


図 1 5 高域半波長遅延

(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) 32bit 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。

ところで、デジタルチャンドバの場合、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 JPOP 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 (ツィータが壊れる危険性有り) よりはるかに良い状況ですが、チャンド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 1 6 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 1 7 のとおり問題ないものとなりました。

以上よりチャンド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。

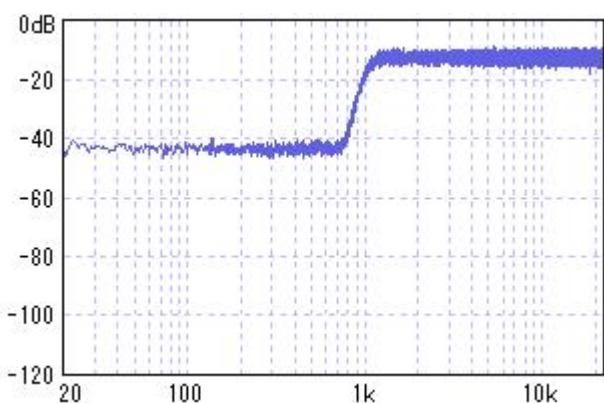


図 1 6 ゲイン調整無し

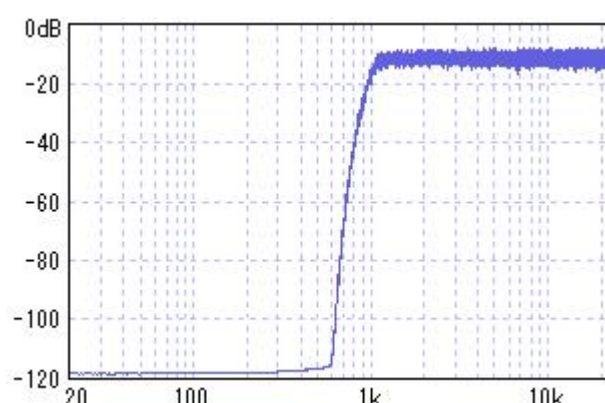


図 1 7 内部で-3dB ダウン

10. バッフルステップの補正とシェルビングフィルタの仕様について

スピーカを壁と床から離して設置すると、バッフルの大きさが有限であることから、波長の長い低域はスピーカボックスの後ろに回り込み反射せず、波長の短い高域はバッフル表面から反射します。この結果、低域は高域より -6dB ほど低下します。この現象をバッフルステップといいます。

図 1 8、図 1 9 に例を示します。四角のバッフルでは 1KHz 前後で音圧のピークがありますが、これは反射波の影響で、バッフルの形状を工夫するとかバッフルの角をラウンド型にするとかスピーカの周りを吸音処理することによりかなり押さえることができます。

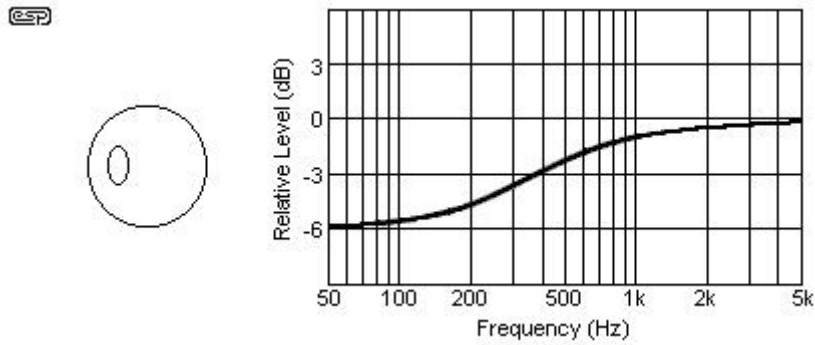


図 1 8 30cm の球

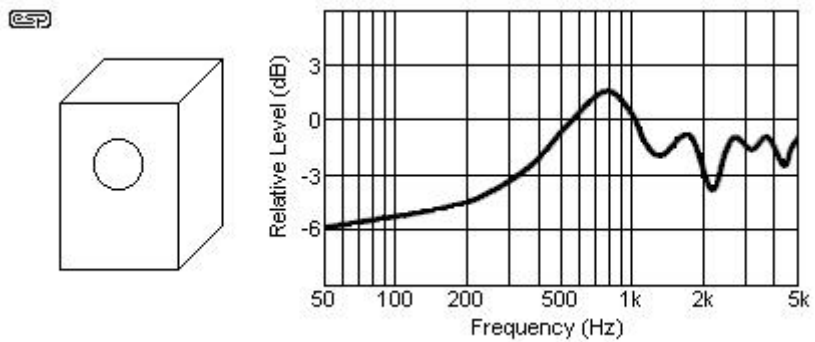


図 1 9 幅 30cm×高さ 45cm の直方体

バッフルステップの影響は、たとえば、edge.exe というプログラムで計算できます。ただし、edge.exe は、バッフルが直角に曲がっている場合を計算しており、角を丸めたラウンドバッフルなどの場合は 1KHz 前後のピークやディップの値は実際とは異なった値となります。

<http://www.tolvan.com/edge/>

図 2 0 に、幅 20cm、高さ 30cm の箱に直径 12cm のスピーカーを入れた場合の edge.exe の計算結果を示します。

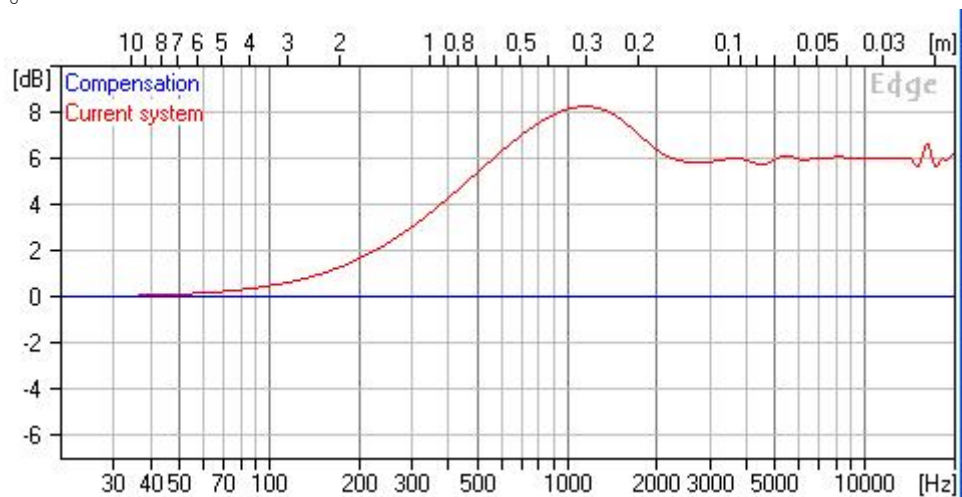


図 2 0 20cm×30cm の箱に 12cm のスピーカーを入れた場合の edge の計算結果

バツフルステップの影響は、スピーカボックスが小型であるほど、高域側から現れます。通常の小型スピーカでは、600Hz位から100Hz位にかけてなだらかに音圧が-6dBほど低下します。よくできたネットワークであれば、この影響を考慮した設計がなされています。

一方、マルチアンプシステムの場合、アンプとスピーカユニットを直結にするため、この補正は困難です。アンプとスピーカの間抵抗とコイルを用いたフィルタをいれれば補正はできますが、これでは、せっかくのマルチアンプの良さが生かせません。

これを解決するために、直線位相 FIR 型のシェルビングフィルタを加えました。バツフルステップの補正を直線位相 FIR 型フィルタで行うことは正当なやり方ではないと思いますが、周波数特性をフラットにすること、使用方法を簡単にすること、プログラムの作成が容易であること、を優先し、直線位相 FIR 型フィルタとしました。

なお、前述したように、四角いバツフルなどでは1KHz前後で音圧のピークが発生しますが、これは、反射波の影響で、バツフルの形状を工夫するかスピーカの周りを吸音処理することによりかなり押さえることができますので、今回の補正の対象とはしませんでした。(物理的に対処できるものは、物理的に対処すべきだと思います。)

具体的には、音圧が低下を始める周波数と低下が終了する周波数と音圧低下量を指定し、これに沿った形で周波数特性が変化するようにフィルタ係数を定めています。フィルタ長は、サンプリング周波数 44.1KHz の場合、1205Tap としました。Tap 数はサンプリング周波数に比例し変化させています。(周波数特性を指定したデータ列に対して逆 FFT をかけ順番を並び替えることでフィルタ係数を求めています。)

シェルビングフィルタで低域を見かけ上増大させることにはなりますが、デジタルフィルタですので、実質は高域を低下させることにはなります。

今回のシェルビングフィルタにおいて下降を始める周波数の最低値は 40Hz です。また、下降を終了する点は、開始する周波数の 2 倍より小さくはできません。

バツフルステップの補正用であるので、それほど低い周波数を指定することはなく、傾斜も急にすることはなく、また、フィルタ長をある程度短くしたかったことから、このような制限を設けました。

最終的には、求めたフィルタと、チャンネルディバイダ用のフィルタを畳みこむことにより、使用するフィルタを求めています。従って、フィルタ長は、サンプリング周波数が 44.1KHz の場合、

$$\text{最終フィルタ Tap 数} = \text{チャンネルディバイダ用に指定したフィルタ長} + 1205 - 1 + \text{遅延 Tap 数}$$

となります。なお、シェルビングフィルタにより上昇する量を 0dB にした場合はシェルビングフィルタは生成されません。

9. (1) ②で述べたように、シェルビングフィルタを含めたチャンネルディバイダの特性を計算するソフト FirGain2B.exe を作りましたので、使用する際の参考としてください。edge.exe などバツフルステップによる低域での低下量を計算した後、設置場所等を考慮し、必要に応じ補正をすれば良いと思います。

11. シェルビングフィルタを含めた特性の測定

シェルビングフィルタを利用した場合のチャンネルディバイダの周波数特性と、合成周波数特性を以下に示します。図 21、図 22 は、クロス 2000Hz、800Hz から 100Hz にかけて 6dB 上昇さ

せた場合、図 2 3、図 2 4は、クロス 200Hz、800Hz から 100Hz にかけて 6dB 上昇させた場合です。後者の条件はバッフルステップ補正が重要となる小型スピーカではあまりない条件ですが、動作確認のため測定しました。いずれも、指定通りの特性が得られています。

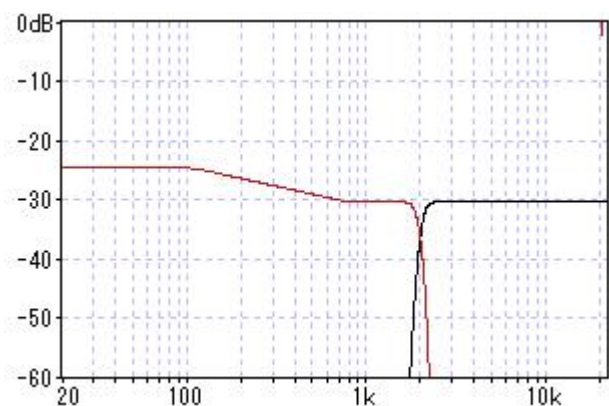


図 2 1 シェルビングフィルタの特性

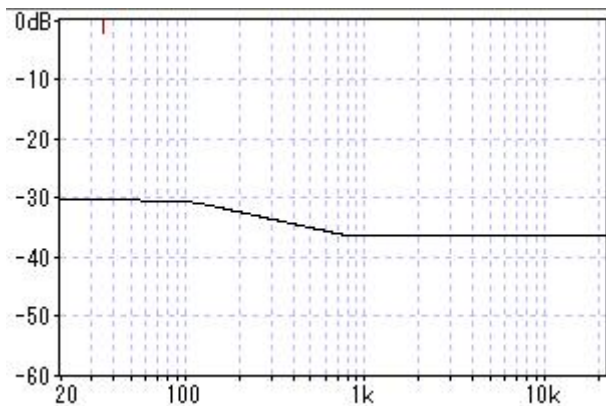


図 2 2 2way の合成周波数特性

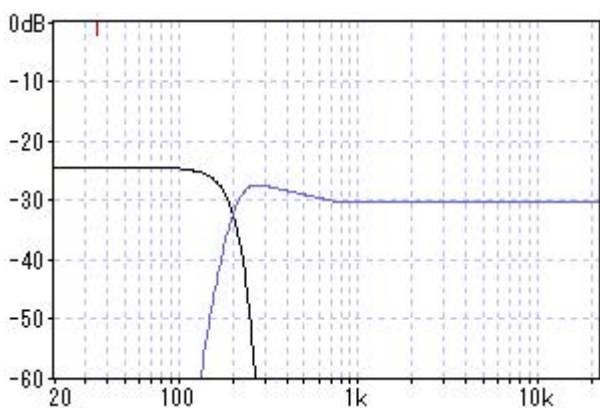


図 2 3 シェルビングフィルタの特性

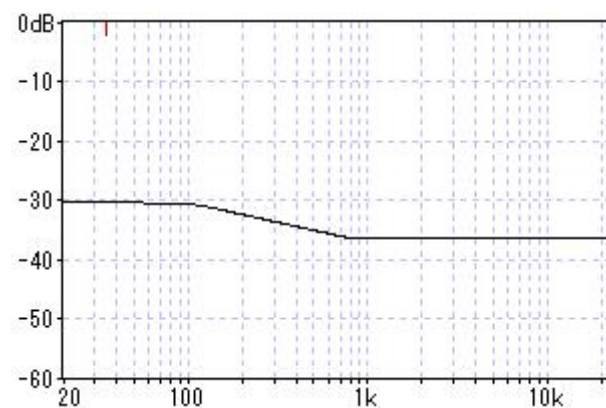


図 2 4 2way の合成周波数特性

1 2. 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 FirGain2B.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 3. 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。素晴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 4. バージョン履歴

(1) channeldividerF2B foobar2000 Ver100 以降用

- 064 2ch ステレオ以外の音を再生し、シークなどを行ったときにクラッシュするバグを修正。(プラグイン自身は 2ch の音声に対してのみ機能することに注意)
- 063 foobar2000 Ver1.00 用 SDK (2010/05/21 版) でコンパイル。この結果、foobar2000 Ver1.0 以降でのみ使用可能となった。

(2) channeldividerF2B foobar2000 Ver096 以降用

- 062b 2010/05/05 フィルタ tap 数を変えたときにトータルフィルタ長をすぐに表示するように変更
- 062 2009/10/25 tempBufDim を前回の計算より大きい場合のみ変更することとし、chunk の大きさが小さくなったときのノイズ発生の可能性に対応。指定 Tap 数より chunk サイズが大きい場合の時の計算式を変更しノイズの発生の可能性に対応。Configure selected 画面の表示を一部変更、初期値を 2000Hz401tap へ。
- 061a 2009/10/25 on_endofplayback() の処理を cleannnBuf() にした。MN を cleannnbuf() でゼロにした。
- 061 2009/10/19 09 用への最初の移植バージョン。バージョン番号を他と合わせた。
- 0441 2006/??/? シェルビングフィルタを加えた最初のバージョン (foobar2000 Ver083 用)

(3) channeldividerF foobar2000 Ver096 以降用

- 062 2009/10/25 tempBufDim を前回の計算より大きい場合のみ変更することとし、chunk の大きさが小さくなったときのノイズ発生の可能性に対応。指定 Tap 数より chunk サイズが大きい場合の計算式を変更しノイズの発生の可能性に対応。
- 061 2009/10/18 曲間でノイズが発生する可能性を改善。(need_track_change_mark() を true にし on_endoftrack() に制御が行くようにした。) Configure selected 画面の英語を一部変更。
- 060 2009/10/12 foobar2000Ver095 用 SDK で作成、vista 対応。
- 050 2006/05/29 foobar2000Ver091 対応。

(4) channeldividerF foobar2000 Ver083 用

- 045 2009/10/25 tempBufDim を前回の計算より大きい場合のみ変更することとし、chunk の大きさが小さくなったときのノイズ発生の可能性に対応。指定 Tap 数より chunk サイズが大きい場合の時の計算式を変更しノイズの発生の可能性に対応。
- 044 2006/05/20 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を修正した。
- 030 ディレイを追加した。
- 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