オーディオ信号処理で学ぶDSP

第3回　高速なIIRフィルタの解説と実装

堀江 誠一

今回は，前回設計を行ったFIRフィルタと反対の特徴を持つIIR（Infinite Impulse Response）フィルタを設計する．IIRフィルタはアナログ・フィルタと同様の特性を持ち，演算時間が短くて済むなどの利点がある反面，固定小数点DSPでは実装が難しいという問題がある．

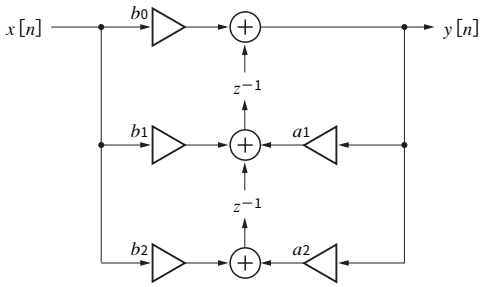
そこで今回は，IIRフィルタの設計法を解説したあと，工夫を凝らして固定小数点DSPに実装する方法を解説する．（編集部）

前回（2007年2月号，pp.143-149），設計方法を示したFIRフィルタには，次のような特徴がありました．

- 位相直線のような，アナログ・フィルタでは実現しにくい特性を実装できる
- その代償として，演算に時間がかかる

今回紹介するIIRフィルタはこれと相反する特性を持っています．つまり，アナログ・フィルタと同様の特性を持ち，演算時間が少なくて済みます．常に位相直線フィルタが必要なわけではないので，演算時間が少ないという点でIIRフィルタは魅力的です．しかし，固定小数点DSPでIIRフィルタを実装する場合には，いろいろと面倒なことがあります．

結論からいえば，16ビット固定小数点DSPで4次や6次のチェビシェフ・フィルタを作るのはなかなか大変です．そのため，筆者はIIRフィルタではなく，FIRフィルタを専ら使用しています．けれども，浮動小数点DSPや，もっとビット数の多いDSP，あるいはFPGAといったものでの実装を考えると，IIRフィルタのことを知っておいて損はありません．



（a）直接型IIの転置形式

$$H(z)=\frac{b_0+b_1z^{-1}+b_2z^{-2}}{1-a_1z^{-1}-a_2z^{-2}}$$

（b）伝達関数

図1　バイクワッド構成

そこで今回は少々無理をして固定小数点DSPでIIRフィルタを実装する方法を解説します．

1. IIRフィルタの原理

● IIRフィルタはインパルス応答が無限に続く

IIRフィルタは，FIRフィルタと対比して紹介されます．FIRフィルタと異なり，IIRフィルタはフィードバック経路を内部に持つため，インパルス応答が無限に続きます．これがIIR（Infinite Impulse Response）という名前の由来です．

IIRフィルタは，一般にはバイクワッド構成と呼ばれる形式で実装されます（**図1**）．この形式にも複数の形があるのですが，詳細はここでは書きません．大事なことは，このバイクワッド形式のフィルタは分子と分母がそれぞれ2次の多項式であるような伝達関数を持つということです．ということは，バイクワッド・フィルタ単体では2次のフィルタしか実装できません．しかし，バイクワッド形式のフィルタを縦列接続することで，高次の伝達関数を持ったフィルタを実現できます．その場合，全体の伝達関数はそれぞれのバイクワッド・フィルタの伝達関数の積になります．

さらに，この分子と分母が多項式であるような伝達関数は，チェビシェフ・フィルタやバターワース・フィルタを初めとするアナログ・フィルタの分野ですでに研究されています(1)．こういったフィルタの設計手法を用いることにより，ディジタル・フィルタでもチェビシェフやバターワースといった特性のフィルタを使うことができます．

もちろん，アナログ・フィルタの伝達関数はラプラス変換を基にするものであり，それに対してディジタル・フィルタの伝達関数はZ変換が基になっています．しかし，この両者は機械的に変換できることがわかっています(2)．そ

KEYWORD ── DSP，信号処理，IIRフィルタ，固定小数点処理，バイクワッド・フィルタ，スケーリング

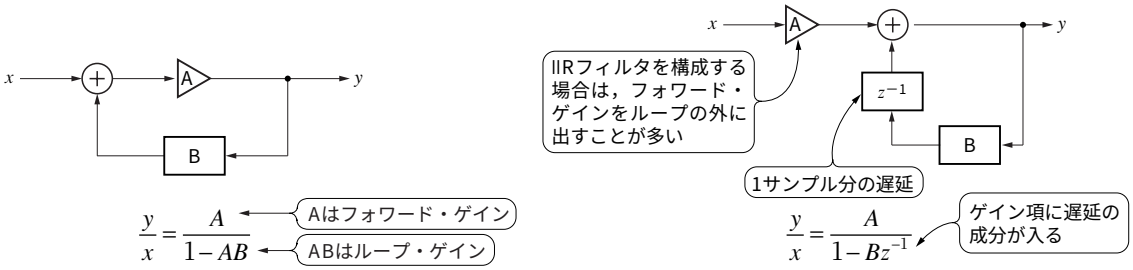


図2
フィードバック回路

（**a**）アナログ・フィードバック回路　　（**b**）ディジタル・フィードバック回路

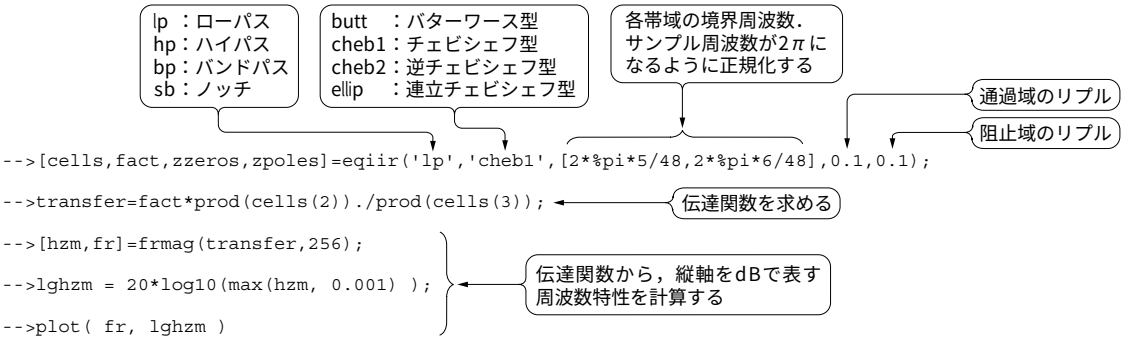


図3
eqiir( )関数による設定

もそもアナログ回路とディジタル回路のフィードバック回路の伝達関数は，本質的にはほとんど同じなのですから，変換できるのは不思議ではありません（**図2**）．

繰り返しになりますが，複雑な伝達関数はバイクワッド・フィルタの縦列接続によって実装します．そこで与えられた伝達関数から，それぞれのバイクワッド・フィルタの伝達関数を求めるために，全体の伝達関数を因数分解しなければなりません．伝達関数をいったん因数分解してしまえば，分子，分母の2次関数の未知数に付く係数をそのままバイクワッド・フィルタの係数として利用することができます．

前回説明したようにFIRフィルタは伝達関数の「波形」をそのまま実装したものです．それに対して，IIRフィルタはよく知られているアナログ・フィルタの伝達関数の「式」をそのまま実装したものだと言えます．

2. IIRフィルタの設計

● 今回もSciLabで設計

IIRフィルタもFIRフィルタと同じようにSciLabを利用して設計できます．FIRフィルタの設計を行うにはSciLabのeqfir()関数を用いてインパルス応答を求めると前回は説明しました．今回のIIRフィルタの設計にはSciLabのeqiir()関数を用います．

念を押しますが，IIRフィルタの設計結果は伝達関数になります．SciLabはこれを因数分解まで含めて全部行ってくれるので，設計者が行うべき作業はフィルタの通過特性，遮断特性，チェビシェフかバターワースか，次数はどこまで許すかといった仕様を決めることです．もちろん，設計結果はSciLab上でグラフとして確認できます．

図3では，グラフを描画するところまで説明しています．描画に関する部分は前回のFIRフィルタの設計とほとんど同じです．1点だけ違うのは，周波数帯域の計算関数であるfrmag()に対して渡す引き数がインパルス応答ではなく，伝達関数そのものになっていることです．これまで見たように，フィルタの特性表現には，時間軸でインパルス応答として表現するか，伝達関数を数式として表現するかという二つの選択肢があります．SciLabのfrmag()は，そのどちらにも応えることができるように設計されています．

描画の前に行っているtransferの計算は重要です．これは伝達関数をSciLab形式で準備しています．eqiir()の結果としては，IIRフィルタの設計に使う数値がそのま