

D A S P 応 用 入 門

Introduction of how to use Digital Audio Signal Processor

高 島 淳 行⁽¹⁾ 伊 藤 辰 男⁽²⁾
Atsuyuki Takashima Tatsuo Ito

要 旨

近年、オーディオの分野でも注目を集めているデジタル信号処理技術は、従来のアナログ技術では実現が難しかったさまざまな機能を実現できる可能性を持っている。

本稿では、当社が開発したデジタル・オーディオ・シグナル・プロセッサ FT8800 (DASP) を初めて使おうとする人のために、必要最小限の知識とプログラム例を紹介する。

第二章では、基礎知識として、デジタル信号の説明から、伝達関数の求め方、Z変換の方法について、アナログシステムと比較しながら述べる。第三章では、デジタル信号処理で多く使用されるデジタルフィルタについて述べ、第四章では、その設計法について具体例を用いて説明する。また、第五章では、第四章で設計したデジタルフィルタをDASPのプログラムで表現し、その内容について述べる。

In recent years, Digital Signal Processing Technology that is paid attention to in the field of audio equipment. Because various functions being hardly realized by analog technology can be realized by digital technology.

This paper introduces basic knowledges and a program example for beginners who try to use Digital Audio Signal Processor FT8800 (DASP) developed by Fujitsu Ten.

In chapter 2, fundamental knowledge necessary to use DASP is explained, including digital signals, how to lead transfer functions and a method of Z-transformation. Chapter 3 explains digital filters used many times in digital signal processing technology, and in chapter 4 its design method are discussed with a concrete example. Chapter 5 explains a DASP program example of digital filter designed in chapter 4.

1. ま え が き

ディジタル信号処理技術は、かなり古くから存在し、ハードウェアのレベルに応じて音声処理、通信処理など、様々な分野に利用されてきた。近年、LSI技術が高度に発達し、高速化を実現したことによって、可聴周波数帯域での処理が可能となり、オーディオの分野でも注目を集めるようになってきた。これは、音場制御のように従来のアナログ処理では回路規模が大きく実用が不可能な機能が、ディジタル処理によって、実用化が期待できるためである。

当社においても、ハードウェア、ソフトウェアの両面から開発を進めてきた。ハードウェアに関しては、当社独自のディジタル・オーディオ・シグナル・プロセッサFT8800（以下DASPという）の開発が完了した。また、ソフトウェアに関しても、音場制御、イコライザなどの開発を完了し、製品化を推進している。

しかし、DASPハードウェア自身はあくまで実現手段であり、そこに搭載されるソフトウェアがなければ、何の役にも立たない。したがって、今後、有効にDASPを活用するためには、ハードウェアの改良もさることながら、アルゴリズム開発、ソフトウェア開発を強力に推進し、さまざまなユーザ・ニーズに柔軟に対応したアプリケーションを搭載していく必要がある。

本稿では、当社が開発したDASPを活用する上で、必要最小限の知識とプログラム例を紹介する。

2. DASP応用の基礎知識

2.1 デジタル信号

アナログ信号は一般に時間に対して連続であるが、ディジタル信号処理で扱う信号は時間に対して不連続となる。このような信号を離散的信号と

よぶ。図-1(a)で示される時間に対して連続的な信号を、等間隔に測定した時のデータの列が離散的信号となり、図-1(b)のようになる。このように、ある時間ごとに値を測定していくことを標本化（サンプリング）とよぶ。

さらに、標本化された信号は一般に連続的な値であるが、有限のビット長で近似表現することを量子化とよび、それにより得た信号をディジタル信号とよぶ。（図-1(c)参照）

アナログ信号を標本化して離散的信号にし、さらにそれを量子化してディジタル信号にすることをA/D変換といい、それを行う機構をA/Dコンバータとよぶ。この逆であるディジタル信号からアナログ信号を得ることをD/A変換といい、そ

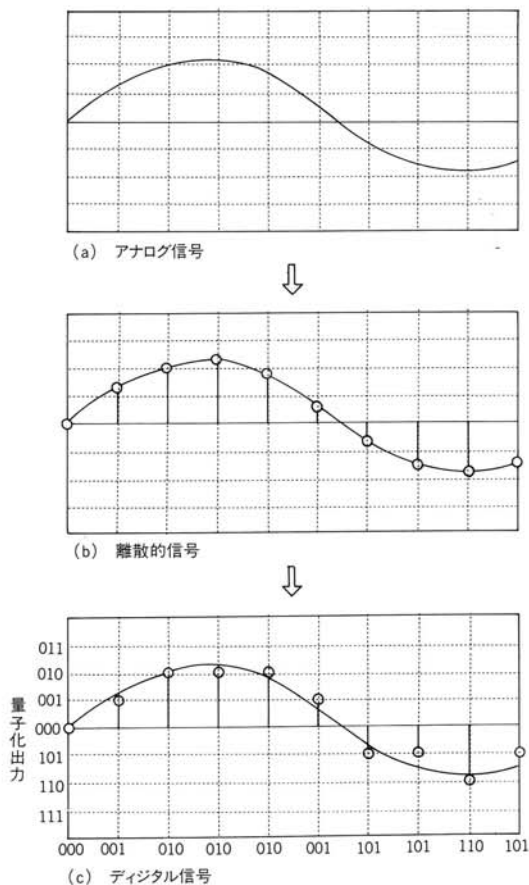


図-1 アナログ信号からディジタル信号への流れ

Fig. 1 Flow of analog signal to digital signal.

れを行う機構をD/Aコンバータとよぶ。

2.2 サンプリング定理

離散的信号を作る場合、アナログ信号からサンプリング間隔 T で標本化するが、このサンプリング間隔の逆数、つまり $1/T$ をサンプリング周波数とよぶ。この時、 T が大きければアナログ信号の細かな時間変動、つまり周波数の高い成分は離散的信号には現れない。したがって、アナログ信号に含まれる最高の周波数が高いほど T を小さくする必要がある。この T の値を理論的に与えるのが「ナイキストのサンプリング定理」とよばれるもので、以下のとおりである。

「連続信号が周波数 f_0 以上の成分を含まない時に、連続信号のすべての周波数成分についての情報を保ったまま、これをサンプリングして離散的信号を作るには、

$$T \leq \frac{1}{2f_0} \dots\dots\dots (1)$$

の間隔 T でサンプリングしなければならない。」

ここで、サンプリング周波数を fs とすると、(1)式は、

$$fs \geq 2f_0 \dots\dots\dots (2)$$

と表現できる。また、サンプリング周波数の $1/2$ の周波数をナイキスト周波数とよび、これを fn とすると、

$$fn \geq f_0 \dots\dots\dots (3)$$

と書くことができる。

ナイキスト周波数より高い周波数成分が含まれている場合、これをサンプリングして作った離散的信号中では、別の周波数成分があるように見なされてしまう。この現象を「エイリアシング」とよぶ。たとえば、サンプリング周波数 $fs = 20\text{kHz}$ (ナイキスト周波数 $fn = 10\text{kHz}$)として8 kHzと12 kHzの正弦波で離散的信号を作った場合、図-

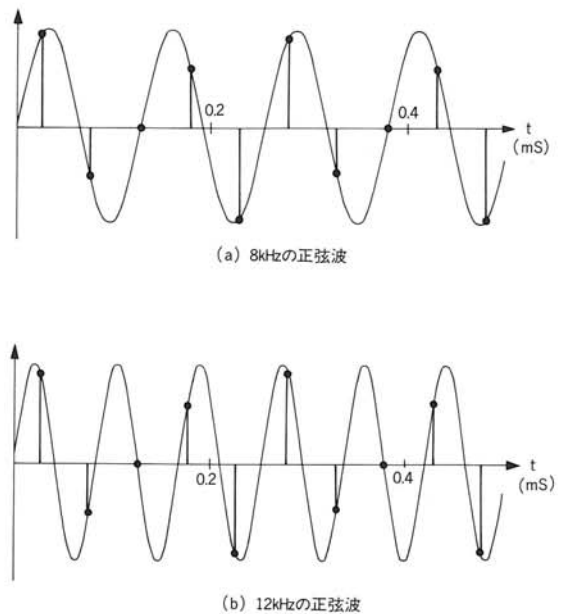


図-2 正弦波のサンプリング
(サンプリング周波数20 kHz)

Fig. 2 Sampling state of sine wave (sampling frequency: 20 kHz).

2のように全く 同一になってしまい、どちらの信号か見分けがつかなくなる。つまり、原信号が12 kHzの正弦波であるにもかかわらず、8 kHzの正弦波と見なしてしまうことになる。このため、ナイキスト周波数以下のカットオフ周波数のローパスフィルタで、原信号の高周波成分を除いてからサンプリングを行う必要がある。このローパスフィルタを「アンチ・エイリアスフィルタ」とよび、できるだけ高い周波数成分までサンプリングするために、急峻な遮断特性が要求される。

実際にサンプリングを行う場合、サンプリング周波数は以下の理由によって、サンプリング定理で理論的に決められる周波数よりも高く設定する必要がある。

①一般にフィルタのカットオフ周波数は、振幅特性が-3 dBになる点を指すため、アンチ・エイリアスフィルタを用いても、カットオフ周波数以上の周波数成分が残る。

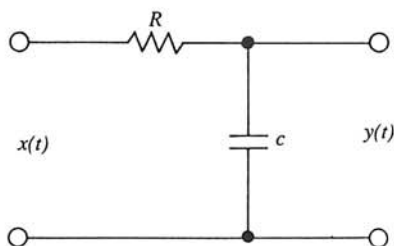


図-3 アナログ素子による積分回路
Fig. 3 An integrator using analog devices.

②ナイキスト周波数近辺をサンプリングした離散的信号は、無限データ長のサンプリングであれば、原信号の情報を全て含むが、実際のデータは有限長であるため、原信号を完全には含まない。

これらのことから、20 kHzまでの信号を含むコンパクトディスクでは、サンプリング周波数が44.1 kHzと決められている。

2. 3 離散システム

アナログシステムの時間領域での入力と出力の関係は、微分方程式を用いて表すことができる。一方、離散的信号を扱う離散システムでは、入力と出力の関係を差分方程式で表す。

たとえば、図-3に示す積分回路（ローパスフィルタ）の入力信号 $x(t)$ と出力信号 $y(t)$ の関係は(4)式で表すことができる。

$$\frac{dy(t)}{dt} + \frac{1}{CR}y(t) = \frac{1}{CR}x(t) \dots\dots\dots (4)$$

これに対応する離散システムの入力信号 $x[n]$ と出力信号 $y[n]$ の関係は(5)式で表せる。

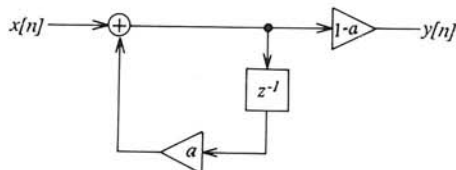


図-4 (5)式に対応する離散システムのブロック線図
Fig. 4 Block diagram of discrete system corresponded to expression (5).

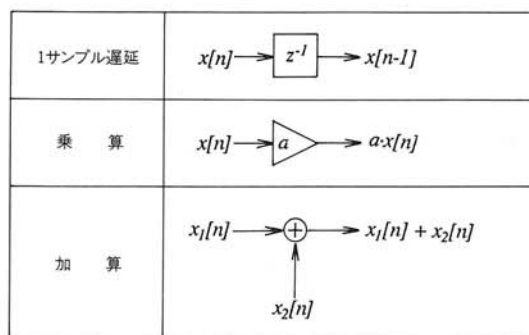


図-5 ブロック線図の要素
Fig. 5 Factors of block diagram.

$$y(n) = \frac{CR}{CR + 1}y(n-1) + \frac{1}{CR + 1}x[n]$$

$$= a \cdot y[n-1] + (1-a) \cdot x[n] \dots\dots (5)$$

$$\left(a = \frac{CR}{CR + 1} \right)$$

これをブロック線図で表すと図-4となる。ブロック線図は、アナログシステムの回路図に相当し、その要素には図-5のようなものがある。

図-3の積分回路に単位インパルス信号を入力した時の出力 $y(t)$ は(4)式を解くと求められて(6)式となり、図-6のような応答を示す。

$$y(t) = \frac{1}{CR} \cdot e^{-\frac{t}{CR}} \dots\dots\dots (6)$$

つぎに(5)式の差分方程式に単位インパルス信号を入力した場合には、つぎのようになる。

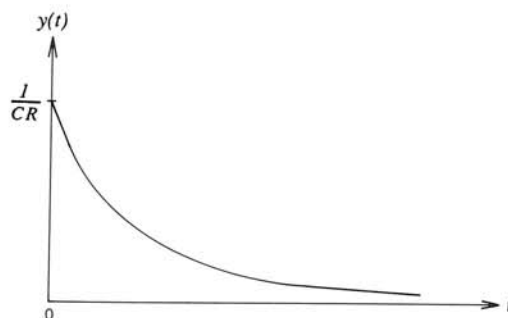


図-6 積分回路のインパルス応答
Fig. 6 Impulse response of an integrator.

$x[0] = 1, n \neq 0$ のとき $x[n] = 0$

$n < 0$ のとき $y[n] = 0$ となるから

$$y[0] = a \cdot y[0-1] + (1-a) \cdot x[0]$$

$$= a \cdot y[-1] + (1-a) \cdot 1$$

$$= 1-a$$

$$y[1] = a \cdot y[1-1] + (1-a) \cdot x[1]$$

$$= a \cdot y[0] + (1-a) \cdot 0$$

$$= a(1-a)$$

$$y[2] = a \cdot y[2-1] + (1-a) \cdot x[2]$$

$$= a \cdot y[1] + (1-a) \cdot 0$$

$$= a^2 \cdot (1-a)$$

⋮

$$y[n] = a \cdot y[n-1] + (1-a) \cdot x[n]$$

$$= a^n \cdot (1-a) \dots\dots\dots (7)$$

これは図-7のようになり、図-6の曲線のある時間間隔ごとにサンプリングして得た値に相当する。

2.4 伝達関数

前節で述べたように、アナログシステムの入出力の関係は微分方程式で表すことが出来るが、ラプラス変換を用いて伝達関数で表現することも出来る。一方、離散システムの場合も伝達関数を用いて表現することが出来るが、これには、 z 変換を用いる。 z 変換の詳しい理論的な説明は専門書に譲ることとして、ここでは、設計する上で必要

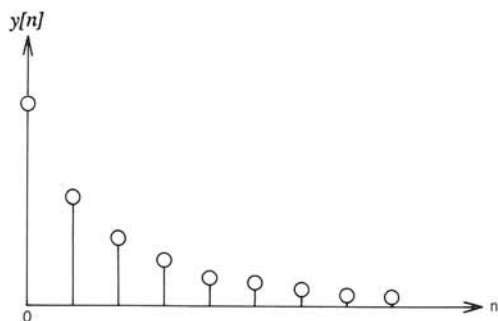


図-7 一次の離散システムのインパルス応答
Fig. 7 Impulse response of simple discrete system.

な伝達関数の求め方について述べる。

まず、差分方程式から伝達関数を求める場合には、次の置き換えを行えばよい。

「離散的信号 $x[n]$ の z 変換を $X(z)$ とすると、1 サンプル遅れた信号 $x[n-1]$ の z 変換は、 $X(z) \cdot z^{-1}$ となる。また、 m サンプル遅れた信号 $x[n-m]$ の z 変換は $X(z) \cdot z^{-m}$ となる。」

例として、(5)式の伝達関数を求める。まず、(5)式の両辺を上記の法則にしたがって z 変換する。 $x[n]$ の z 変換を $X(z)$ 、 $y[n]$ の z 変換を $Y(z)$ とすると、 $y[n-1]$ は $Y(z) \cdot z^{-1}$ となるから、

$$Y(z) = a \cdot Y(z) \cdot z^{-1} + (1-a) \cdot X(z) \dots\dots\dots (8)$$

伝達関数 $H(z)$ は、入力と出力の比であるから、

$$H(z) = \frac{Y(z)}{X(z)} = \frac{1-a}{1-a \cdot z^{-1}} \dots\dots\dots (9)$$

となる。

また、アナログシステムの s 領域での伝達関数が判っている場合には、表-1のような標準 z 変換または双一次 z 変換を用いて離散システムの伝達関数を求めることになる。例として、図-3の積分回路の伝達関数 $H(s)$ から $H(z)$ を求める。

(4)式から

$$\begin{aligned} H(s) &= \frac{1}{CRs + 1} \\ &= \frac{1}{CR} \cdot \frac{1}{s + \frac{1}{CR}} \end{aligned}$$

ここで標準 z 変換を用いて、 $(\alpha = 1/CR)$

$$H(z) = \frac{1}{CR} \cdot \frac{1}{1 - e^{-\alpha T} \cdot z^{-1}} \dots\dots\dots (10)$$

となる。

以上、デジタルシステムの特徴を、アナログシステムと比較しながら述べてきたが、これらをまとめると、表-2、表-3のようになる。

表-1 各種 $s-z$ 変換の特性と特徴

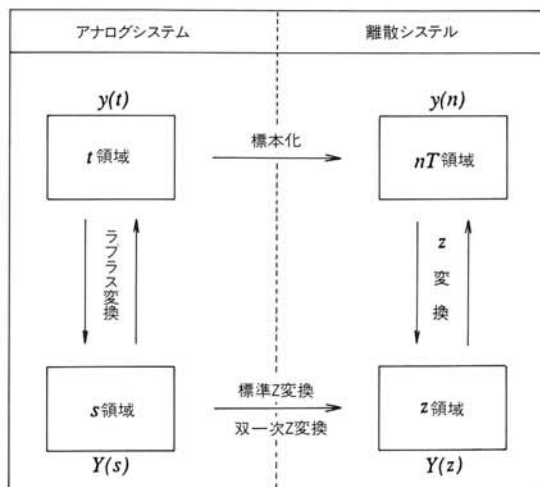
	標準 z 変換	双一次 z 変換
変換の形式	指数変換 $z = e^{sT}$	代数的変換 $s = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}$
伝達関数 $H(s)$ の変形	部分分数展開 $\frac{1}{s + \alpha} \rightarrow \frac{1}{1 - z^{-1} \exp(-\alpha T)}$	有理関数のままでよい 周波数歪みの補正が必要 $H(s) \mid x = \frac{2}{T} \frac{1 - z^{-1}}{1 + z^{-1}}$
適用可能な $H(s)$ の形式	帯域制限のある $H(s)$ のみに有効 低域、帯域フィルタ	全ての $H(s)$ に有効、とくに広帯域フィルタの設計に有効
変換の性質	インパルス応答の標準値をそのまま保存 (インパルス不変)	振幅周波数特性はそのまま保存される 非線形な周波数変換を受ける $\Omega = \frac{2}{T} \cdot \tan \frac{\omega T}{2}$
サンプリング間隔 T	$\omega_s/2$ の点で $ H(j\omega) $ が十分減衰するように選ぶ	自由に選ぶことができる

3. デジタルフィルタ

デジタル信号処理で最も基本的な操作は、離散的フーリエ変換(Discrete Fourier Transform 略して DFT) とフィルタリングである。DFT に関しては、専門書に譲り、ここでは、DSP 応用の王者とも言え、オーディオ信号処理に無くてはならないデジタルフィルタについて述べる。

3.1 デジタルフィルタとは

表-2 各種変換領域の関係



デジタルフィルタとは何かという問いに対しては、つぎのように考えることができる。

「図-8のアナログ回路がある周波数特性をもつとき、その特性と一致するように設計された図-8のデジタル回路の部分をデジタルフィルタという。」

表-3 アナログフィルタとデジタルフィルタの比較

	アナログフィルタ	デジタルフィルタ
数 式	微分方程式	差分方程式
表 現	回路図	ブロック線図
変 換	ラプラス変換	z 変換
伝達関数	$H(s) = \frac{Y(s)}{X(s)}$	$H(z) = \frac{Y(z)}{X(z)}$
固有関数	$e^{j\omega t}$	$e^{jn\omega T}$: サンプリング間隔
具 体 例	 $H(s) = \frac{1}{CRs + 1}$ $y(t) = \frac{1}{CR} e^{-\frac{t}{CR}}$	 $H(z) = \frac{1}{CR} \cdot \frac{1}{1 - e^{-\frac{T}{CR}} z^{-1}}$ $y[n] = \frac{1}{CR} \cdot e^{-\frac{nT}{CR}}$



図-8 デジタルフィルタの概念

Fig. 8 Concept of a digital filter.

このことから、デジタルフィルタを実現するためには、ICを使って組み立てたデジタル回路でも良く、またコンピュータの演算処理でも良い。したがって、コンピュータのプログラムでデジタルフィルタを実現した場合には、その実体は存在しないことになる。

つぎに、デジタルフィルタの長所としては、以下のようなものがある。

①同一のハードウェアでも、制御命令の与え方により、フィルタの特性を容易に変えることができる。

②信号やフィルタの係数を2進数表現する場合、そのビット長を長くすればするほど高精度のフィルタを実現することができる。

③アナログ回路のような温度変動、製品ごとのバラツキ、経年変化による品質劣化がない。

④LSI化による小型化、経済化、高信頼性化が図れる。

⑤回路接続に際しては、インピーダンス整合の問題がない。

一方、短所として、つぎのようなものがある。

①アナログ信号に対して、標本化、量子化、符号化を行うためにクロック周波数が高くなり、扱えるアナログ信号の周波数上限が低い。

②処理が簡単な場合でも、A/Dコンバータ、D/Aコンバータなど、周辺回路規模が増大し、コスト高になる。

③フィルタの係数、信号が有限ビットで近似さ

れるために特性の誤差が生じたり、演算結果の切り捨てやオーバーフローによる雑音の発生に注意する必要がある。

3.2 デジタルフィルタの分類

デジタルフィルタの入出力関係は、一般に、(11)式の差分方程式で表現される。ここで、帰還路がない($b_k=0, k=1, 2, \dots, N$) 場合、(11)式は(12)式となる。伝達関数 $H(z)$ は(13)式となり、その構成は図-9となる。このようなデジタルフィルタを非巡回形とよび、インパルス応答はフィルタの係数に一致し、有限の時間域に収まる。また、(12)式および(13)式に現れる M をフィルタの次数とよぶ。

$$y[n] = \sum_{k=0}^M a_k \cdot x[n-k] + \sum_{k=0}^N b_k \cdot y[n-k] \dots (11)$$

$$y[n] = \sum_{k=0}^M a_k \cdot x[n-k] \dots (12)$$

$$H(z) = \sum_{k=0}^M a_k \cdot z^{-k} \dots (13)$$

一方、帰還路がある場合には(11)式の差分方程式となる。また、伝達関数 $H(z)$ は(14)式となり、

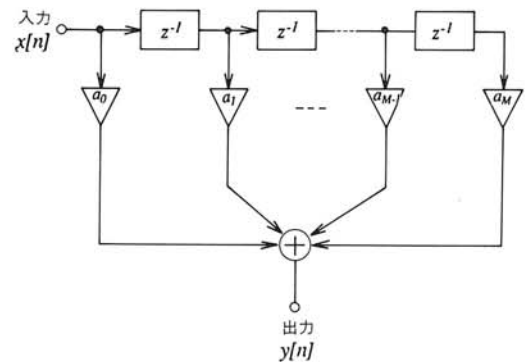


図-9 非巡回形フィルタ

Fig. 9 Non-recursive filter.

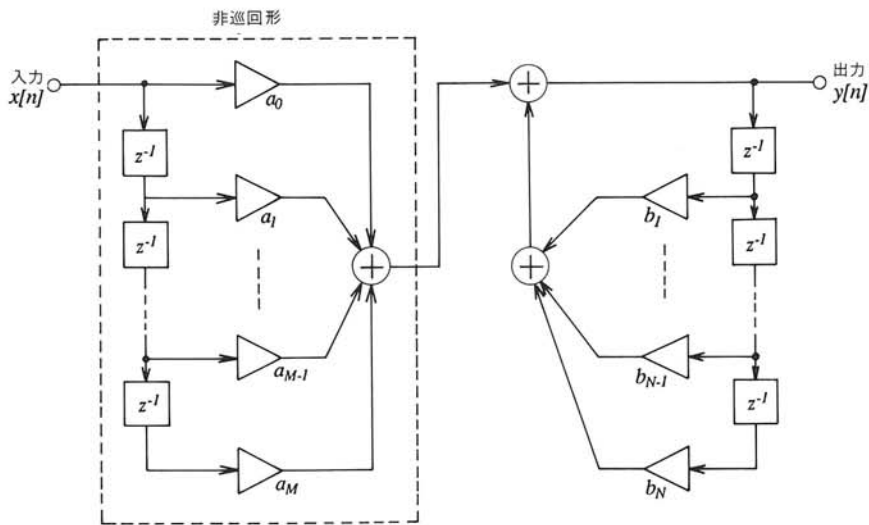


図-10 巡回形フィルタ

Fig. 10 Recursive filter.

その構成は図-10で表せる。このようなデジタルフィルタを巡回形とよび、インパルス応答は多くの場合、無限に継続する。また、巡回形フィルタの次数は、 N つまり伝達関数の分母の次数と

なる。

$$H(s) = \frac{\sum_{k=0}^M a_k \cdot z^{-k}}{1 - \sum_{k=1}^N b_k \cdot z^{-k}} \dots\dots\dots (14)$$

表-4 デジタルフィルタの分類

フィルタの種類	特 徴	フィルタの種類	特 徴
F I R	- インパルス応答が有限に収まる z平面上、零点のみ存在	非巡回形	帰還路なし
		巡回形	
I I R	- インパルス応答が無限に続く z平面上、零点と極が存在		帰還路あり
		巡回形	

以上のことから、インパルス応答の継続時間に着目して、有限なインパルス応答を持つフィルタをFIR (Finite Impulse Response) フィルタ、無限なインパルス応答を持つフィルタをIIR (Infinite Impulse Response) フィルタとよんで区別することも多い。通常、FIRフィルタは非

表-5 デジタルフィルタの長所と短所

	I I R フィルタ	F I R フィルタ
長 所	- アナログフィルタの伝達関数をそのまま利用できる - 少ない演算量で急峻な特性を実現できる	- 完全な直線位相特性をもつフィルタが構成できる - 非巡回形構成では発振がない。そのため、係数が自由に換えられ、適応フィルタリングに適する - I I Rに比べ、丸め雑音が少ない - 目的に応じた各種の構成法が存在し、選択の余地が広い
短 所	- 帰還部をもつため、発振の可能性がある - 係数値の有限語長の影響を受け易く、特性劣化を起こすことがある	- 急峻な周波数特性を得るためには相当高い次数のフィルタが必要となるため、演算量が多くなり、処理速度が低下する - アナログ領域での設計ができない

巡回形、IIRフィルタは巡回形に対応するが、例外的に巡回形でFIRフィルタとなるものもあるので、デジタルフィルタは表-4のように分類する。また、FIRフィルタ、IIRフィルタの長所、短所をまとめると、表-5になる。

4. デジタルフィルタの設計

4.1 フィルタの仕様

デジタルフィルタを設計する場合、まずフィルタの満足すべき仕様を決める必要がある。この仕様の与え方には、周波数領域と時間領域の二つがあり、さらに細分化するとつぎのようになる。

①周波数領域——振幅特性・位相特性

②時間領域——インパルス応答

ここでは、もっとも一般的な、振幅特性が指定された場合のデジタルフィルタの設計手順を紹介する。

一般的な振幅特性の与え方は、フィルタの特性により、図-11の四種に分類される。これらはすべて、通過域遮断周波数 ω_p と通過域許容減衰量 A_p 、阻止域遮断周波数 ω_s と阻止域最小減衰量 A_s を指定する必要がある。

4.2 IIRフィルタの設計

一般的に、IIRフィルタの設計は、図-12のような手順に従って行われる。双一次 z 変換は、変換のための制約が少なく、標準 z 変換に比べて計算が容易であるため、よく用いられる。

いま、設計すべきデジタルフィルタの仕様として、つぎのような低域通過型フィルタが与えられたとする。

- ・サンプリング周波数 ($1/T$): 20 kHz
- ・通過域周波数 : 0 Hz~1.5 kHz
(通過域遮断角周波数: $\omega_p = 2\pi \times 1500$ (rad/s))
- ・通過域許容減衰量: $A_p = 3.0$ dB
- ・阻止域遮断周波数: 3 kHz
(阻止域遮断角周波数: $\omega_s = 2\pi \times 3000$ (rad/s))
- ・阻止域最小減衰量: $A_s = 10$ dB

まず最初に、対応するアナログフィルタの設計

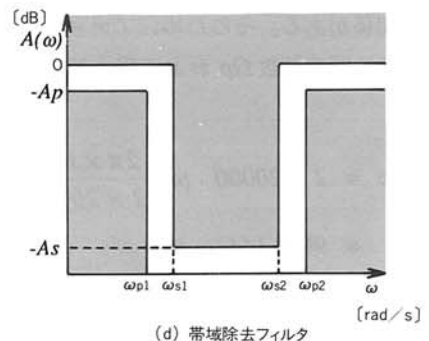
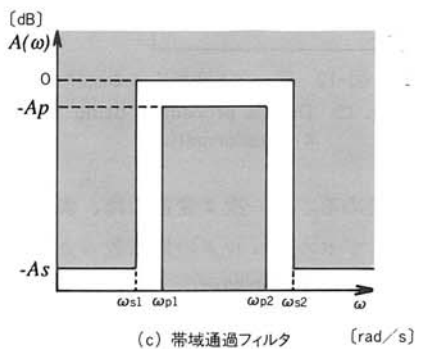
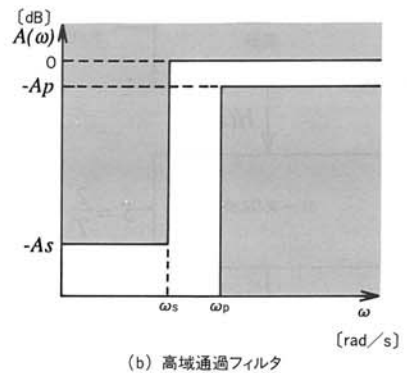
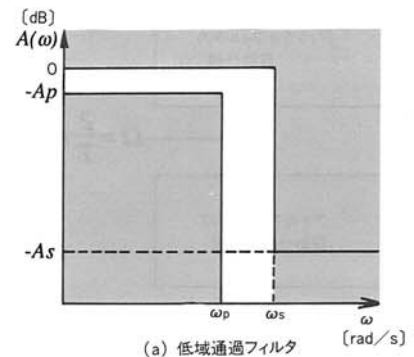


図-11 デジタルフィルタの振幅特性の仕様
Fig. 11 Frequency response of digital filters.

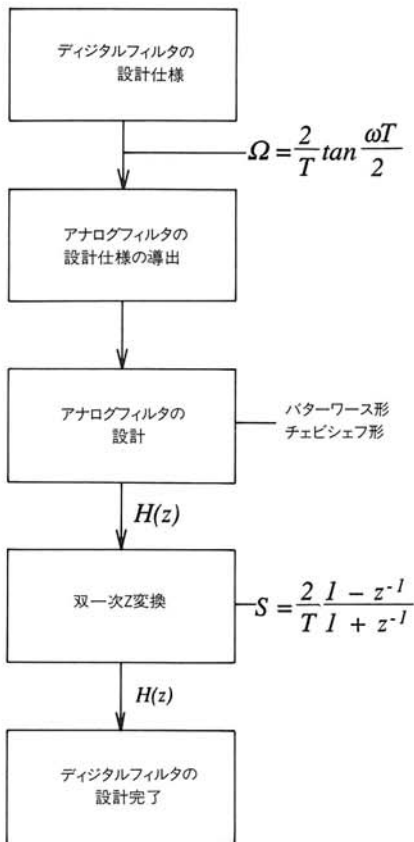


図-12 双一次 z 変換による設計手順

Fig. 12 Design procedure using bilinear z-transformation.

仕様を決める。双一次 z 変換では、表-1にあるようにデジタルフィルタの周波数 ω とアナログフィルタの周波数 Ω の間に、

$$\Omega = \frac{2}{T} \cdot \tan \frac{\omega T}{2}$$

となる関係がある。そのため、アナログフィルタの通過域遮断周波数 Ω_p および阻止域遮断周波数 Ω_s は、

$$\begin{aligned} \Omega_p &= 2 \times 20000 \cdot \tan \frac{2\pi \times 1500}{2 \times 20000} \\ &= 9603.144 \text{ (rad/s)} \\ & (= 1528.4 \text{ (Hz)}) \end{aligned}$$

$$\Omega_s = 2 \times 20000 \cdot \tan \frac{2\pi \times 3000}{2 \times 20000}$$

$$= 20381.008 \text{ (rad/s)}$$

$$(= 3243.7 \text{ (Hz)})$$

となる。一方、アナログフィルタの振幅特性の仕様はデジタルフィルタの場合と同じであり、

- 通過域許容減衰量: $A_p = 3.0 \text{ dB}$
- 阻止域最小減衰量: $A_s = 10 \text{ dB}$

となる。

つぎに、仕様を満足するアナログフィルタの伝達関数を求める。ここでは、バターワースフィルタを用いて設計する。

バターワースフィルタの設計手順はつぎのとおりである。さらに高次数のフィルタ設計やチェビシェフフィルタ設計手順については、専門書を参照して頂きたい。

まず、(16)式を用いて次数 N を計算する。この値は通常、整数とはならないので、大きい方の整数値に切り上げて次数 N とする。この場合、 $N = 2$ となる。

$$N = \frac{\log \left[\frac{10^{\frac{A_s}{10}} - 1}{10^{\frac{A_p}{10}} - 1} \right]}{2 \log \frac{\Omega_s}{\Omega_p}} \dots\dots\dots (16)$$

$$\begin{aligned} &= \frac{\log \left[\frac{10^1 - 1}{10^{0.3} - 1} \right]}{2 \log \frac{20381.0}{9603.1}} \\ &= 1.463 \quad N = 2 \end{aligned}$$

つぎに、(17)式を用いて遮断周波数 Ω_0 を求める。

$$\Omega_0 = \frac{\Omega_p}{\left(10^{\frac{A_p}{10}} - 1 \right)^{\frac{1}{2N}}} \dots\dots\dots (17)$$

$$\begin{aligned} &= \frac{9603.144}{(10^{0.3} - 1)^{0.25}} \\ &= 9614.55 \text{ (rad/s)} \dots\dots\dots (18) \\ & (= 1530.2 \text{ (Hz)}) \end{aligned}$$

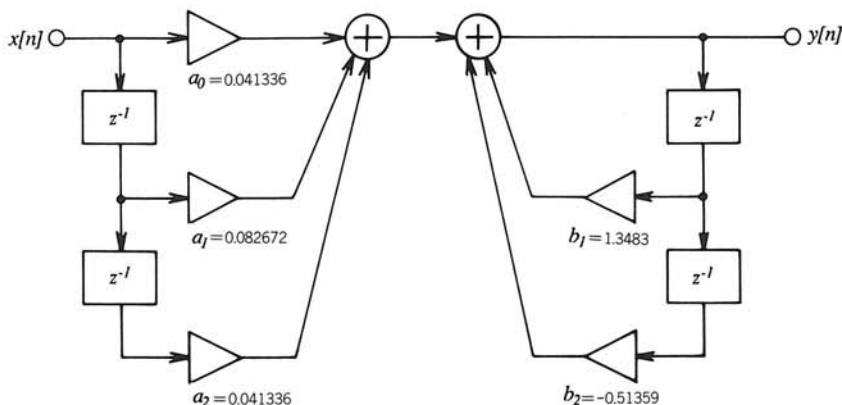


図-13 設計されたデジタルフィルタのブロック線図

Fig. 13 Block diagram of a digital filter.

最後に、バターワース多項式を用いて得られる伝達関数の式に、求めた値を代入する。この場合、次数 $N=2$ であるから伝達関数 $H(s)$ は、

$$\begin{aligned}
 H(s) &= \frac{1}{\left[\frac{s}{\Omega_0}\right]^2 + \sqrt{2}\left[\frac{s}{\Omega_0}\right] + 1} \\
 &= \frac{\Omega_0^2}{s^2 + \sqrt{2}\Omega_0 \cdot s + \Omega_0^2} \\
 &= \frac{9.2440 \times 10^7}{s^2 + 1.3597 \times 10^4 s + 9.2440 \times 10^7}
 \end{aligned}
 \quad \dots\dots\dots (19)$$

以上で、アナログフィルタの伝達関数が求められる。

アナログフィルタの伝達関数が求まると、いよいよ双一次 z 変換を行って、デジタルフィルタの伝達関数 $H(z)$ を求める。これには、(19) 式に (21) 式を代入すればよい。

$$S = \frac{2}{T} \cdot \frac{1 - z^{-1}}{1 + z^{-1}} \quad \dots\dots\dots (20)$$

$$= 4 \times 10^4 \cdot \frac{1 - z^{-1}}{1 + z^{-1}} \quad \dots\dots\dots (21)$$

したがって、 $H(z)$ は、

$$H(z) = \frac{0.041336 (1 + z^{-1})^2}{1 - 1.3483 z^{-1} + 0.51359 z^{-2}}$$

$$\begin{aligned}
 &= \frac{0.041336 + 0.082672 z^{-1} + 0.041336 z^{-2}}{1 - 1.3483 z^{-1} + 0.51359 z^{-2}} \\
 &\dots\dots\dots (22)
 \end{aligned}$$

となる。

以上で、デジタルフィルタの設計は完了であり、ブロック線図は図-13になる。また、このデジタルフィルタのシミュレーション結果は図-14のようになり、仕様を満足している。

4. 3 F I R フィルタの設計

F I R フィルタの設計法には、フーリエ級数展開による設計法、窓関数による設計法、周波数サンプリングによる設計法、チェビシェフ近似による設計法などがあるが、アナログ領域での設計は

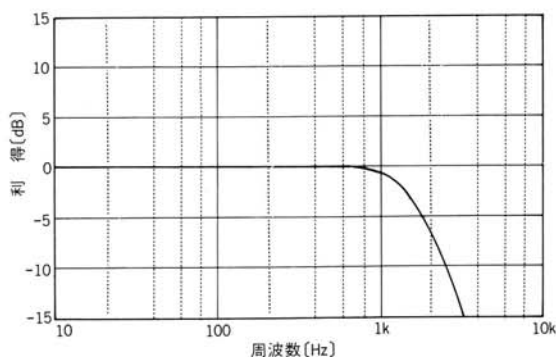


図-14 設計したフィルタのシミュレーション結果

Fig. 14 Simulation results of a digital filter.

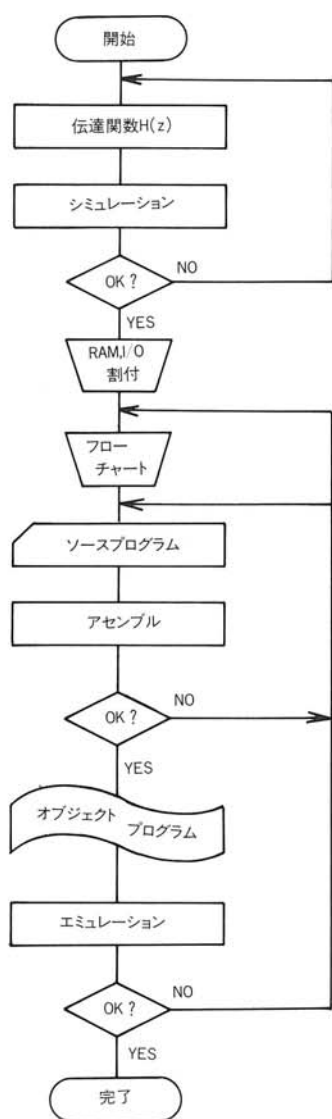


図-15 プログラム開発手順

Fig. 15 Programming procedure of DASP.

できない。実際に設計する場合には、そのためのプログラムが準備されており、それを用いれば簡単に設計できるため、詳しい内容は専門書に譲り、本稿では割愛する。

5. DASPを用いたフィルタの実現

5.1 DASPでのプログラム作成手順

伝達関数 $H(z)$ が求まり、ブロック線図が出来

れば、いよいよDASPにマスク化するプログラムを作成することになる。

プログラムの開発手順は、図-15のとおりである。

①シミュレーションを行って仕様通りの設計になっているかを確認する。

②I/Oピン、内部RAMに機能を割り当てる。

③フローチャートの設計を行い、ニーモニックを使用してフローチャートのコーディングを行う。

④エディタを用いてソースプログラムを作成する。

⑤ソースプログラムをクロスアセンブラでアセンブルし、オブジェクトプログラムを作成する。

⑥開発ツールを用いてオブジェクトプログラムの動作確認（エミュレーション）を行う。

動作確認で仕様通りの特性が得られれば、プログラムの開発は完了する。あとは、プログラムをDASPにマスク化すればよい。

5.2 DASPでのプログラム

前章第二節で設計した二次IIR低域通過フィルタのDASPでのソース・プログラムは図-16のようになる。プログラムの実行は、DASP初期設定⇒内部RAMの初期設定⇒入力データの転送⇒フィルタ計算⇒演算結果の出力という手順となる。

0000番地から0008番地まではDASPの初期設定を行う部分で、0009番地から000D番地までは入力ポートのシリアルデータをレジスタへ転送する部分である。0011番地から0018番地までが主演算部で、内部メモリから係数を呼び出してフィルタの計算を行い、0019番地から001B番地で演算結果を出力する。001C番地から0026番地までは内部RAMの初期設定を行う部分で、サブルーチンとなっており、演算の前に実行される。

```

*****
= 000000 ch_in      equ $00
*****
= 000001 x1_stack   equ $01
= 000002 b1_stack   equ $02
*****
= 000010 a0         equ $10      : 4.1336e-002
= 000011 a1         equ $11      : 8.2672e-002
= 000012 a2         equ $12      : 4.1336e-002
= 000013 b1         equ $13      : 1.3483e+000
= 000014 b2         equ $14      : -5.1359e-001
*****
= 000020 plus_one   equ $20
= 000021 minus_one  equ $21
= 000022 zero       equ $22
*****

0000      org      $000
0000 reset:      clm      port:rnf
0001          stm      hsl1:lsen:btcl:btco:blud
0002          clr0     a:b:d:p:c0:cl:sft:sp
0003          clr1     zrd:sgd:ovd:dvzd:unm:lt:ovm
0004          ldi      H$0.vsm
0005          ldi      H$0100.b1
0006          nop
0007          nop
0008          bs1f      alw:ram_clr

*****
0009      *****      MAIN ROUTINE *****
000A wait1:      brul     hsl:wait1
wait2:      br1f      hsl:wait2
*****
000B      *****      DATA INPUT *****
000C      mov      s1fl1.d
000D      mov:cif   a,a
000E      mov      d,ch_in
*****
000F      *****      VERTIAL SHIFT *****
0010      mov      x1_stack,x1
shift:      lab      zero(a),[bx1+1](b)
x1_push:    mov      x1,x1_stack
*****
0011      *****      OPERATION *****
0012      mov      ch_in(a),[bx1](b)
0013      mov      a0(a),[bx1+1](b)
0014      Lab:fml   a1(a),[bx1+1](b)
0015      Lab:fsp   a2(a),[bx1+2](b)
0016      Lab:fms   b1(a),[bx1+1](b)
0017      Lab:fms   b2(a),[bx1+2](b)
0018      mov:fms   a,a
mov:fsm     a,a
*****
0019      *****      DATA OUTPUT *****
001A      mov:cif   d,[bx1+0](b)
001B      mov      d,soel1
brif        alw:wait1
*****
001C      *****      SUB ROUTINE *****
001D      ram_clr:  lia     $0
001E      mov      a,x1
001F      a_ram_clr: rep     $0
0020      mov      b,(x1+1)
0021      rep      $0
0022      b_ram_clr: mov     a,(x1+1)
0023      set_cof:  lld      H$3e0000
0024      mov      d,plus_one
0025      lld      H$be0000
0026      mov      d,minu_one
rt1f        alw

end

```

メモリ割付け部
↑
実行部

RAM AREA

[プログラム上で用いるラベルを定義する]

***** DSPの初期設定 *****

[クロックモードの設定、通信モードの設定]
[フラグ、レジスタ等の設定を行う]

***** 演算部 *****

[入力ポートにデータが入るまで待つ]

***** データ入力 *****

[固定小数点データを浮動小数点データに]
[変換し、「ch_in」へ転送する]

***** バーチャル・シフト *****

[アドレスポインタのシフトを行う]

***** 主演算部 *****

[差分方程式に従った演算を行う]

***** データ出力 *****

[浮動小数点データを固定小数点データに]
[変換し、出力ポートへデータを転送する]

***** 内部RAMの初期設定 *****

[内部RAMの各メモリの値を設定する]

番地 ラベル ハーモニック オペランド

図-16 設計したフィルタのプログラム

Fig. 16 Program lists of a digital filter.

6. あ と が き

以上、DASP応用のための最も基本的な事項について紹介した。この手順にしたがえば、誰でもディジタルフィルタを設計し、DASPを用い

て実現することが可能となる。すでにディジタル信号処理技術を知っておられる方々にとっては、物足りない面も多々あると思うが、本稿の主旨をご理解頂き、ご諒解を願いたい。

DASPのハードウェアおよびソフトウェアの

開発環境が整備された今、我々は、デジタル信号処理技術を積極的に取込み、アナログの置き換えだけでなく、デジタルならではのアプリケーションを製品に反映して行かなければならない。

本稿により、デジタル技術がより身近なものとなり、DASPの応用範囲がさらに拡大し、普及することを願って止まない。

参考文献

- 1) 三上直樹：『技術者のためのデジタル信号処理入門』トランジスタ技術、CQ出版社、443-450 (1987. 12)
- 2) 山崎芳男：『音響信号とデジタル信号処理』富士通テン技報、Vol. 5 No. 1 p.4 (1987)
- 3) 辻井重雄ほか：『デジタル信号処理技術』日本工業技術センター、38-40 (1982)
- 4) 三谷政昭：『デジタルフィルタデザイン』昭晃堂、37-44, 136-138 (1987)